

VMware and Oracle Databases Software Solutions Deployment Guide



By Desmond Chan
Senior ISV Solutions Engineer
Partner Solutions & Marketing,
VMware, Inc.



Table of Contents

| | | |
|-----------|--|-----------|
| 1. | Introduction | 5 |
| 1.1. | Oracle Database 10g / RAC Software Suite..... | 5 |
| 1.2. | VMware® Infrastructure 3 Software Suite | 6 |
| 1.3. | Oracle Database-VMware Solutions..... | 9 |
| 1.4. | Summary of Deployment Approach..... | 12 |
| 2. | Deployment of Single-instance Oracle Database 10g..... | 14 |
| 2.1. | Solution Architecture..... | 15 |
| 2.2. | VMware Infrastructure 3 Installation | 19 |
| 2.3. | ESX Server Host Time Synchronization | 19 |
| 2.3.1. | Configuration File Modification | 20 |
| 2.3.2. | Open the NTP Client Port..... | 21 |
| 2.3.3. | Restarting and Monitoring the NTP Service | 24 |
| 2.4. | Add Virtual Machine to a Host | 24 |
| 2.4.1. | Installation of Linux Guest Operating System | 25 |
| 2.4.2. | Installation of VMware® Tools | 27 |
| 2.4.3. | Virtual Machine Time Synchronization | 28 |
| 2.5. | Linux Configuration for Oracle Database Installation..... | 29 |
| 2.6. | Oracle ASM Installation and Configuration..... | 31 |
| 2.6.1. | Virtual Disk Creation | 31 |
| 2.6.2. | Format the Virtual Disks in Linux | 32 |
| 2.6.3. | ASM Packages and Configuration..... | 34 |
| 2.7. | Installation of Oracle Database Software / Template | 36 |
| 2.8. | Creation of a Gold Image Template for a Single-instance Oracle Database..... | 39 |
| 2.9. | Deployment of a Single-Instance Oracle Database on a Virtual Machine..... | 40 |
| 2.10. | Using the VMware HA Feature for Single-Instance Oracle Database Failover | 45 |
| 2.10.1. | Network Card Redundancy..... | 45 |
| 2.10.2. | VMware HA Cluster Creation..... | 46 |
| 2.10.3. | VMware HA Benefits..... | 47 |
| 2.11. | VMware VMotion for Single-Instance Oracle Database | 48 |
| 2.11.1. | VMKernel Port Configuration | 48 |
| 2.11.2. | Live Migration with VMware VMotion | 50 |
| 3. | Deployment of Oracle Database 10g RAC | 51 |
| 3.1. | Solution Architecture..... | 53 |
| 3.2. | VMware Environments for RAC Software..... | 55 |
| 3.2.1. | Network Configuration..... | 55 |
| 3.3. | Storage Configuration | 58 |

| | | |
|----------|---|-----|
| 3.3.1. | Add New Physical Shared Storage..... | 58 |
| 3.3.2. | Shared Disk Creation | 59 |
| 3.4. | Gold Image for Oracle RAC..... | 61 |
| 3.4.1. | Add Virtual Machine to a Host | 61 |
| 3.5. | Gold Image Creation for RAC Deployment..... | 64 |
| 3.5.1. | Installation of Linux Guest Operating System | 64 |
| 3.5.2. | OCFS2..... | 67 |
| 3.5.3. | Oracle RAC Installation Prerequisites..... | 73 |
| 3.5.4. | Installation of Oracle Clusterware | 73 |
| 3.6. | Installation of Oracle Database 10g Release 2..... | 81 |
| 3.7. | Database Template Preparation | 82 |
| 3.8. | Gold Image Virtual Machine Template Preparation | 83 |
| 3.9. | RAC Gold Image Deployment | 85 |
| 3.9.1. | Virtual Machine Cloning..... | 85 |
| 3.9.2. | Second RAC Node Customization..... | 86 |
| 3.10. | VMware HA for RAC..... | 94 |
| 3.11. | VMware VMotion for RAC | 94 |
| 4. | Summary..... | 95 |
| Appendix | | 97 |
| | rootaddnode.sh | 97 |
| | rootconfig.sh | 100 |

Note from the author

This guide is based on the deployment of Oracle Database products on VMware® Infrastructure performed in the VMware lab located in Palo Alto, CA. It provides the groundwork for more powerful and complicated deployments of Oracle Database products on VMware Infrastructure.

The author would like to thank the following team members for their valuable support in rolling out this guide:

Rajagopal Ramanujam, Solution Architect, Partner Solutions and Marketing

Christopher Rimer, Director, Oracle Alliances, Product Marketing

Johanna Holopainen, Senior Manager of ISV Alliances, Product Marketing

Tamao Nakahara, Alliances Marketing Specialist, Alliances

Christine Holland, ISV Alliance Marcom Manager, Alliances

Matthew Portnoy, Senior Systems Engineer, Systems Engineers

Rodney Rock, Staff Systems Engineer, Systems Engineers

Desmond Chan
Senior ISV Solutions Engineer
Partner Solutions & Marketing,
VMware, Inc.

1. Introduction

The Oracle Database (Single-instance / RAC) software system is one of the most widely deployed database environments in the IT industry today. It is known for its high performance, reliability, availability and scalability features. The VMware® Infrastructure 3 software suite provides powerful and robust infrastructure virtualization solutions to the IT industry, enabling high levels of fault-tolerance, isolation, rapid provisioning, change management, and availability.

This guide provides an overview of the deployment steps for Oracle Database 10g Single-instance and RAC deployments on VMware Infrastructure 3 (VI3) platform. It showcases the key benefits that VMware solutions provide to Oracle Database deployments.

NOTE: Oracle RAC is not officially supported by Oracle on VMware Infrastructure at the time of publishing this guide. Please refer to Oracle documentation for the latest support information.

1.1. Oracle Database 10g / RAC Software Suite

The Oracle Database 10g (Single-instance / RAC) software suite has multiple components that fit into the solution architecture. Each component provides significant functionality to ensure that the databases deployed have high performance, reliability, availability, and scalability. This deployment guide discusses four components in particular:

- **Oracle Database 10g**
 - Oracle Database 10g is a relational database management system (RDBMS) that delivers record-breaking performance and scalability on different platforms.
- **Oracle Automatic Storage Management (ASM)**
 - Oracle ASM provides the performance of asynchronous I/O with the easy management of a file system. It also provides capability that saves database administrators time and provides flexibility to manage a dynamic database environment with increased efficiency.
- **Oracle Real Application Clusters (RAC)**
 - Oracle RAC is a cluster database with a shared cache architecture that overcomes the limitations of traditional shared-nothing and shared-disk approaches to provide highly scalable and available database solutions.
- **Oracle Cluster File System (OCFS2)**
 - OCFS2 presents a consistent file system image across the servers in a cluster. It eases the administration of Oracle RAC.

1.2. VMware® Infrastructure 3 Software Suite

VMware Infrastructure 3 simplifies IT environments so that customers can leverage their storage, network, and computing resources to control costs and respond quickly to changing business needs. The VMware Infrastructure approach to IT management creates virtual services out of the physical infrastructure, enabling administrators to allocate these virtual resources quickly to the business units that need them most.

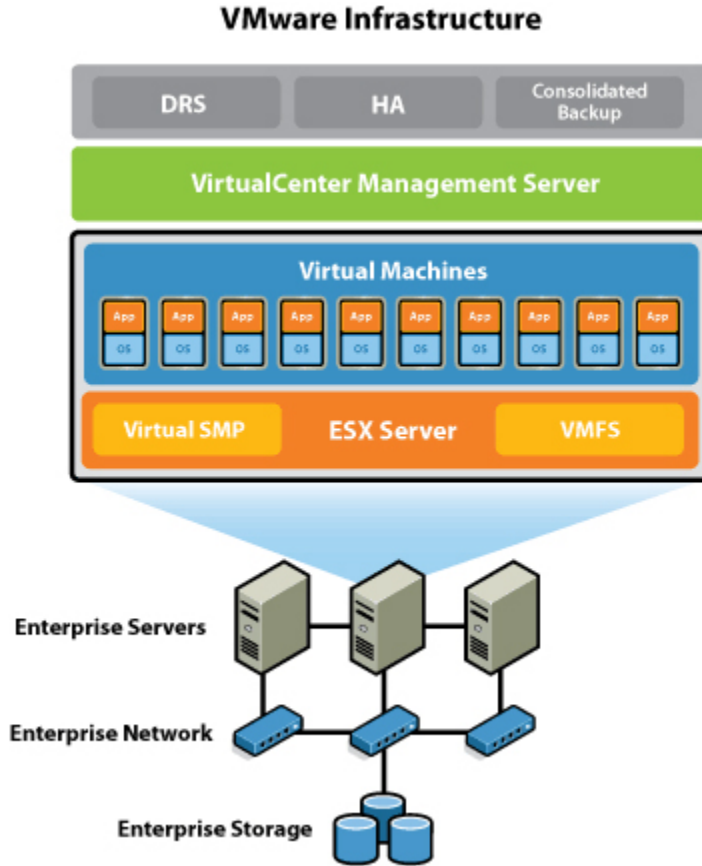


Figure 1. VMware Infrastructure

VMware Infrastructure 3 (VI3) is the next generation of industry-leading infrastructure software that virtualizes servers, storage, and networking, allowing multiple unmodified operating systems and their applications to run independently in virtual machines while sharing physical resources. This deployment guide discusses the installation and configuration of two VMware Infrastructure software components:

- **VMware® ESX Server**
 - VMware ESX Server abstracts processor, memory, storage, and networking resources into multiple virtual machines, giving IT greater hardware utilization and flexibility.

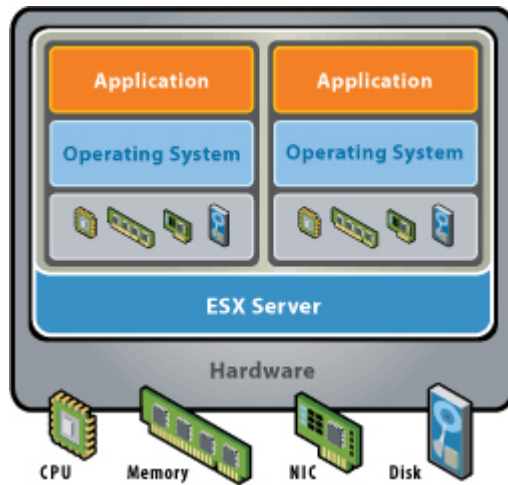


Figure 2. VMware Infrastructure 3 - ESX Server

- **VMware® VirtualCenter**
 - VirtualCenter enables rapid provisioning of virtual machines and performance monitoring of physical servers and virtual machines. VirtualCenter intelligently optimizes resources, ensures high availability to all applications in virtual machines, and makes IT environments more responsive with virtualization-based distributed services such as VMware® High Availability (HA) and VMware® vMotion™.

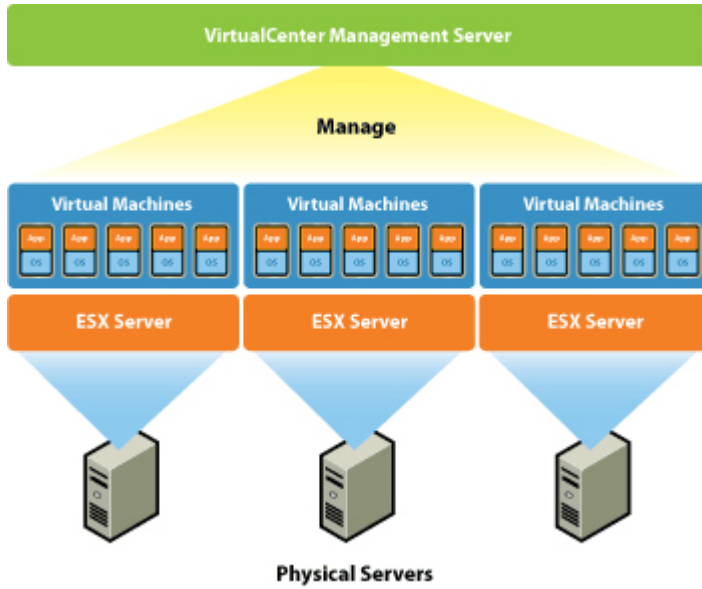


Figure 3. VMware VirtualCenter Management Server

1.3. Oracle Database-VMware Solutions

Oracle Database-VMware Infrastructure 3 deployments discussed in this document will cover the following benefits that the VMware Infrastructure suite brings to enterprise Oracle Database deployments, including single-instance and RAC.

Server Containment

Typically there is a one-to-one relationship between the databases and the physical servers in an Oracle environment. Consolidation, when it happens, usually takes the form of deploying all Oracle Databases in one operating system on a physical server. In this configuration, the databases become unavailable when the server hardware or operating system is down. With VMware Infrastructure, it is possible to run multiple databases on the same physical hardware. The databases run in separate operating system environments on isolated virtual machines, providing a high level of server containment and enhanced availability and manageability.

Key benefits of using VMware Infrastructure for server containment:

1. Consolidate dedicated and isolated Oracle Database environments to a few physical systems.
2. Run multiple Oracle Database environments on the same physical system, providing significant reduction in server numbers and lowering total cost of ownership (TCO).
3. Allow multiple test environments to share the same physical system and eliminate the need for dedicated test systems.



Figure 4. Isolated Virtual Machines

Availability

Traditionally, third-party vendor solutions ensure the availability of single-instance Oracle Databases by allowing for application failover (e.g. clustering) in case of hardware failures. Oracle provides a robust, scalable and available solution that can balance workloads across multiple instances within a Real Application Cluster (RAC). Business-critical databases with high service level agreement targets are perfect candidates for Oracle RAC. However, for customers who run single-instance environments on x86 hardware environments or who have databases that can accept lower levels of availability, VMware HA and VMotion offer a cost-effective alternative to expensive third-party clustering and replication solutions. With VMware HA, failed Oracle instances and virtual machines on one ESX Server host can be restarted on another ESX Server host within minutes. With VMware VMotion, it is possible to migrate live Oracle Database virtual machines between ESX Server hosts, and to move Oracle Database instances off failing hardware, with minimal interruption to the end users.

Key benefits of using VMware Infrastructure to achieve high availability:

1. Provide a cost-effective failover alternative to expensive third-party clustering and replication solutions. VMware software makes it possible to implement enhanced availability without the cost of identical servers, the complexity of rebuilding clusters when physical hardware changes, and the difficulty associated with testing the clustering of physical systems.
2. Employ a time-efficient mechanism to restart an Oracle Database within minutes on an available ESX Server host in case of failures.
3. Cause minimal interruption to end users during live migration of Oracle Database virtual machines from one ESX Server host to another, and minimize downtime during hardware maintenance.

Rapid Provisioning

VMware virtualization solutions significantly reduce the time to provision Oracle Databases and Oracle RAC nodes. For a new deployment, administrators must ordinarily procure hardware, install the operating systems, and perform necessary configurations before the application can be properly installed. This process consumes significant time, IT resources, and dedicated hardware. By using VMware infrastructure, Oracle Database customers can take advantage of virtual machine templates to provision new pre-configured database environments in minutes on virtualized infrastructure hardware. VMware Infrastructure provides sophisticated pre-configured features (e.g. Oracle ASM), and pre-designed database templates that enable rapid provisioning of Oracle Databases.

Cloning ensures a controlled virtual machine configuration so deployment is less error-prone and time-consuming. Administrators can roll out a production-ready environment in a very short period of time.

Key benefits of using VMware Infrastructure for provisioning:

1. Allow rapid provisioning of Oracle Databases (single-instance / RAC) from virtual machine templates.
2. Pass Oracle Database virtual machine images easily from developers directly to testers.
3. Pass Oracle Database virtual machine images easily from test back to development for problem replication and resolution.
4. Recreate Oracle Database instances in the production environment on a single “virtualized” physical system for test purposes.
5. Move test Oracle Database instances to production in minutes.
6. Reset test images from templates after test completion, cutting down on test setup and reset time.
7. Instantly provision different versions of Oracle Database software by storing them in virtual machines.

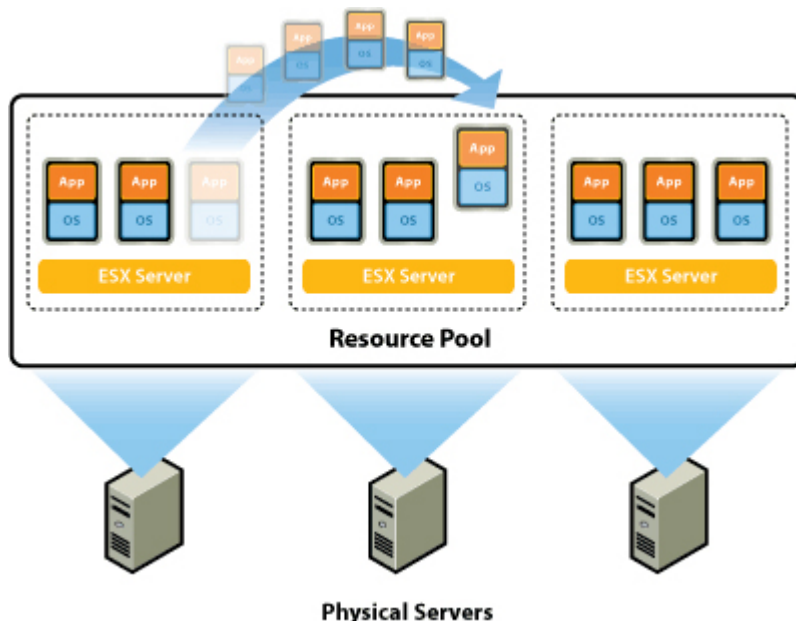


Figure 5. Rapid Provisioning using VMware Infrastructure 3.

1.4. *Summary of Deployment Approach*

This deployment guide covers the following deployment processes:

- a. Installing VMware Infrastructure 3 software, including ESX Server and VirtualCenter.
- b. Configuring the ESX Server hosts to be time-synchronized using NTP.
- c. Creating a virtual machine.
- d. Installing a guest operating system in the virtual machine. Linux is the guest operating system of choice for this deployment guide.
- e. Customizing the kernel parameters and user account configurations in the Linux guest OS.
- f. Installing the Oracle Database 10g / RAC software.
- g. Configuring the Oracle Database 10g / RAC software
- h. Creating a template from the virtual machine installed in the previous steps.
- i. Using the virtual machine template created as a Gold Image to deploy more instances of Oracle single or RAC virtual machines.
- j. Performing customization to enable the new Oracle virtual machines for production.
- k. Configuring VMware HA and/or VMotion to add or enhance high availability for Oracle Database environments.

This guide describes two deployment scenarios:

- 1. Deployment of Single-instance Oracle Database 10g:** Use VMware Infrastructure 3 features to streamline the deployment of single-instance Oracle Database 10g, thereby reducing the cost and time to deployment and providing enhanced availability and manageability.
- 2. Deployment of Oracle Database RAC 10g:** Use VMware Infrastructure 3 features to deploy Oracle Database 10g RAC nodes to reduce the cost and time to deployment.

Figure 6 illustrates the flow of the guide.

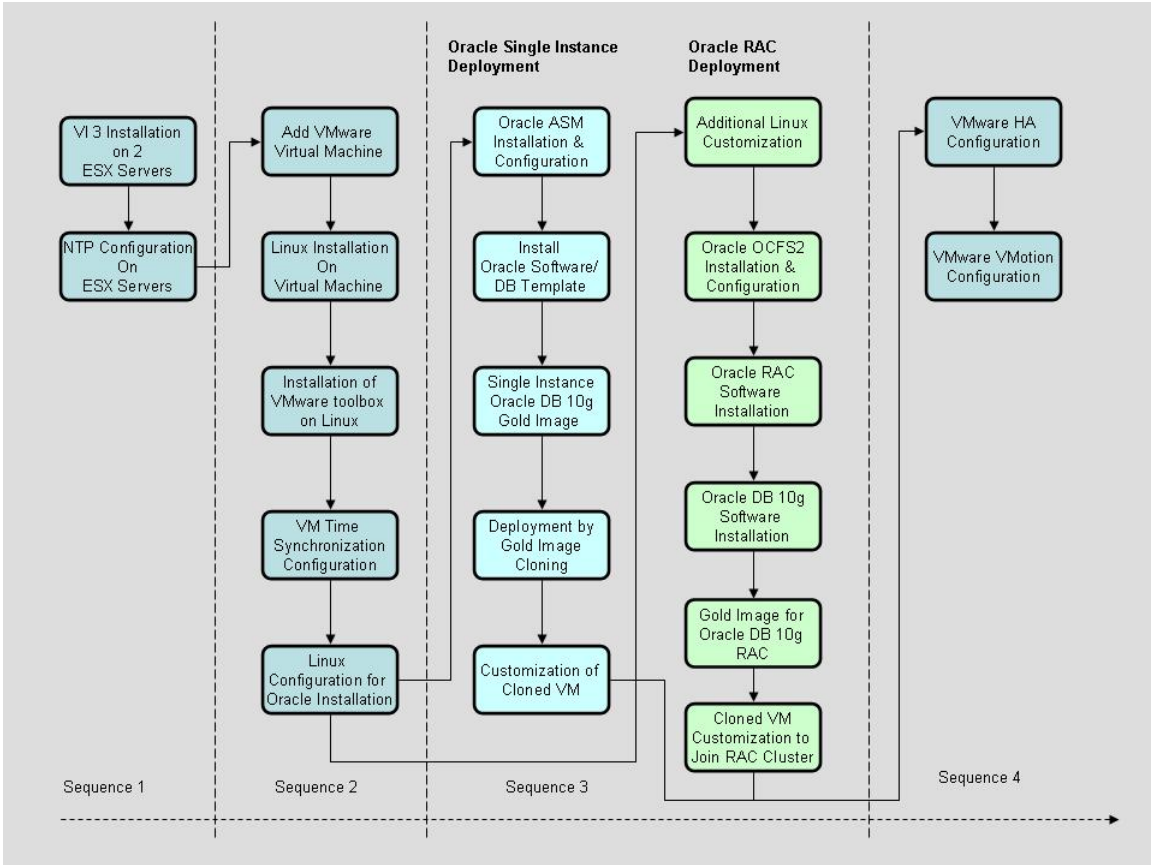


Figure 6. Flow Chart Illustration of the two deployment scenarios: Single-instance Oracle Database and Oracle RAC.

2. Deployment of Single-instance Oracle Database 10g

Oracle Database 10g is a RDBMS that provides high levels of reliability, scalability, availability, and performance. When coupled with the built-in reliability of ASM support, the reliability of Oracle Database 10g increases. This section shows how to provision single-instance Oracle Database 10g in a virtual machine on the VMware Infrastructure platform (VI3), configure Oracle ASM, create a database template, and create a virtual machine template. The virtual machine template, referred to as a **Gold Image**, can be used to rapidly provision a new single-instance Oracle Database environment. This section also showcases the **VMware HA feature and VMware VMotion feature**, which provide **high availability and fault tolerance** for single-instance Oracle Database 10g deployments. Figure 7 shows the deployment process flow for single-instance Oracle Database 10g on VMware Infrastructure 3.

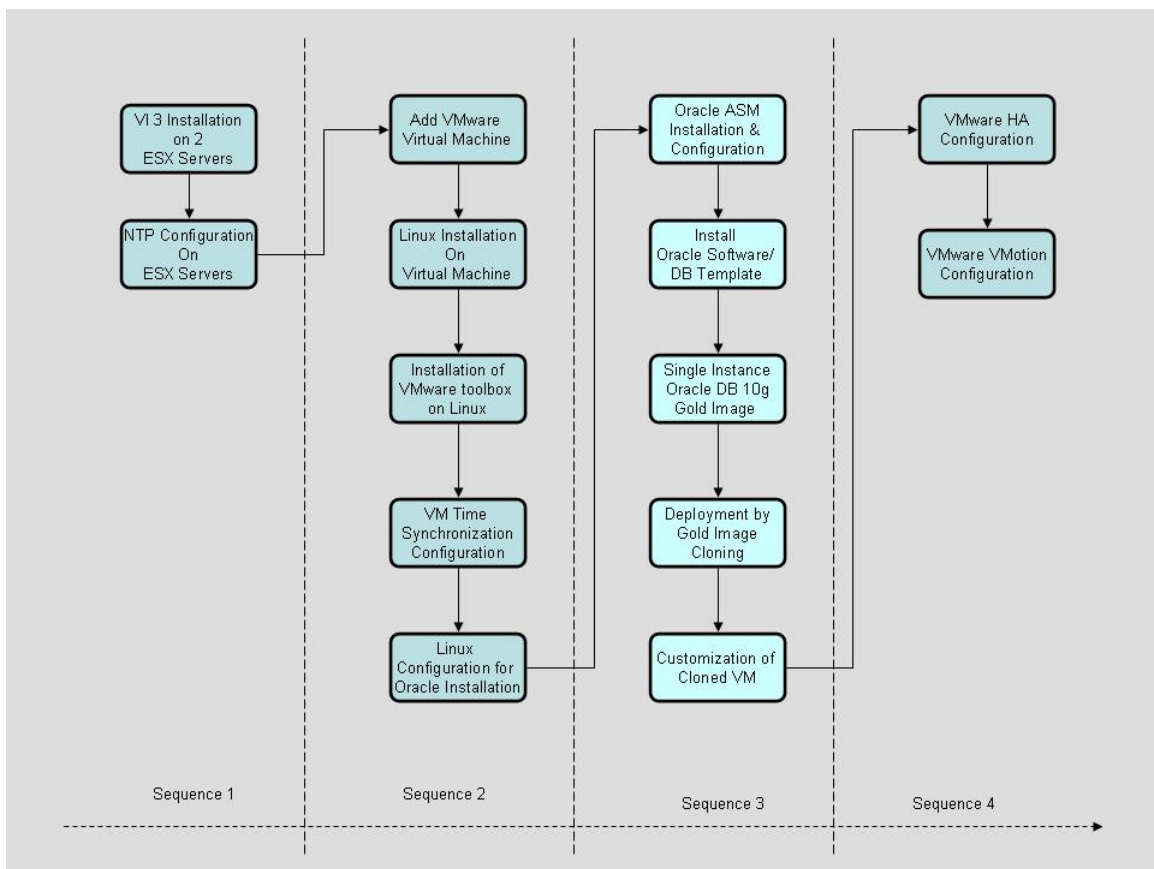


Figure 7. Flow Chart of Single-Instance Oracle Database 10g Deployment on VMware Infrastructure 3.

2.1. Solution Architecture

This procedure provides an **Install-Once-Run-Everywhere (IORE)** model that leverages a virtual machine template with all the installed components and configurations to help users clone and deploy Oracle virtual machines. With the Gold Image, it is also easier to ensure software deployment consistency. Users can be certain that the software and configuration are consistent with the original template.

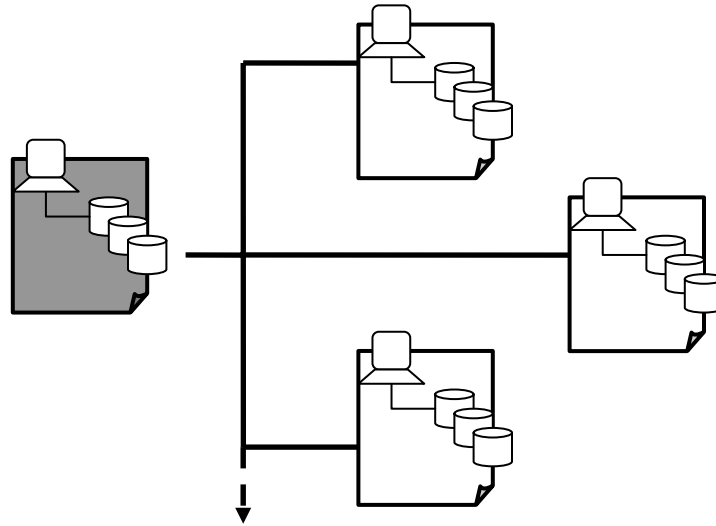


Figure 8. Illustration of cloning from a virtual machine template to form many virtual machines and achieve Install-Once-Run-Everywhere (IORE).

This solution also provides enhanced availability for **Oracle Database 10g** users who only need a **single-instance** deployment. In current industry practices, IT professionals who deploy single-instance Oracle Database usually opt for an **active/passive model**, enabling the Oracle Database to fail over to another host server when the original host fails. To implement the failover, IT often depends on costly third-party solutions. In lieu of such solutions, VMware Infrastructure 3 **HA and VMotion** features provide a cost-effective alternative.

The scenario in this section requires the following hardware and software:

- **Two servers**
 - These two servers need very similar, if not exactly the same, configurations: CPU, memory and NIC cards. The CPUs need to be compatible in order for VMware VMotion to perform properly. The administrator guide http://www.vmware.com/pdf/vi3_admin_guide.pdf provides more details on compatible server configurations and CPUs.
 - For a more complete list of documents on hardware compatibility, refer to http://www.vmware.com/support/pubs/vi_pubs.html.
 - In our lab, the two servers have two CPUs, two gigabytes of memory, and two NIC cards.

- **VMware Infrastructure 3 software**
 - **VMware ESX Server** is installed on each of the two servers.
 - **VMware VirtualCenter 2** is installed on a separate Windows machine to manage the virtual machines on the two ESX Server host systems.

- **Linux OS**
 - **Red Hat Enterprise Linux AS** is the guest operating system covered in this section.
 - This deployment guide is applicable to a variety of Linux guest operating systems, including Red Hat Enterprise Linux AS and Oracle Enterprise Linux. For a list of guest operating systems supported by VMware Infrastructure, refer to http://www.vmware.com/pdf/GuestOS_guide.pdf.
 - **Oracle Enterprise Linux (OEL)** provides the convenience of installing OCFS2 and ASM software packages during operating system installation. OEL can be downloaded via <http://edelivery.oracle.com/linux>.
 - Oracle provides an on-line guide for deploying Oracle Database RAC on Oracle Enterprise Linux (OEL): <http://www.oracle.com/technology/pub/articles/chan-ubl-vmware.html>.

- **Oracle Database 10g**
 - The Oracle Database 10g software is deployed on the Linux guest operating system in a VMware virtual machine.

- **VMware High Availability**
 - In the case of server failure, VMware HA ensures rapid, automated restart of virtual machines. VMware HA automatically and intelligently restarts affected virtual machines on other production servers.
 - This guide shows how to configure a cluster of ESX Server hosts to take advantage of the VMware HA feature.

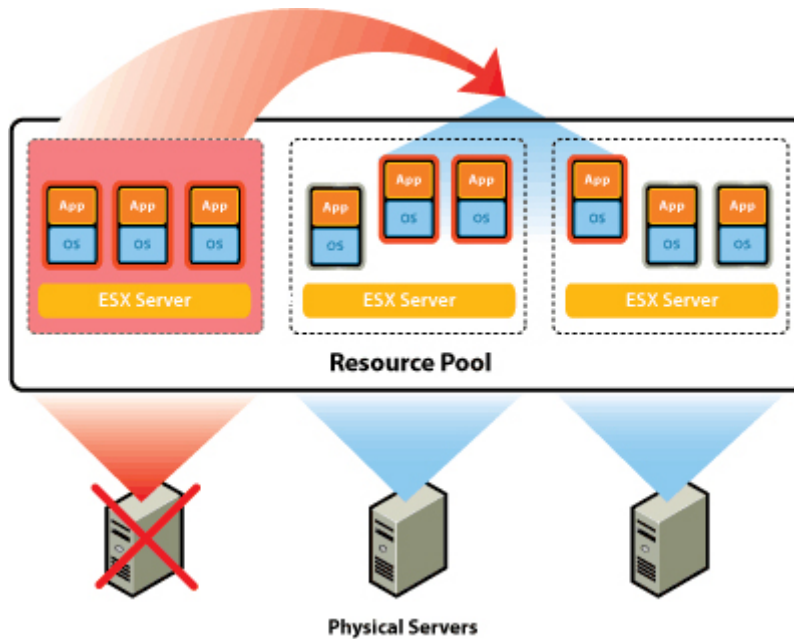


Figure 9. VMware High Availability

- **VMware VMotion**

- VMware VMotion enables live migration of virtual machines from one ESX Server host to another and, as a result, helps organizations to reduce planned downtime dramatically. Because workloads in a VMware Infrastructure environment can be dynamically moved to different physical servers with no service interruption, maintenance can be performed without requiring application and service downtime.
- This guide shows how to configure VMware VMotion in VMware VirtualCenter and how to use VMware VMotion to live-migrate a virtual machine, causing zero interruption to end users.
- Besides live migration, VMware VMotion provides cold migration for virtual machines being moved to another ESX Server host after a shutdown. This guide shows how to enable and use cold migration.

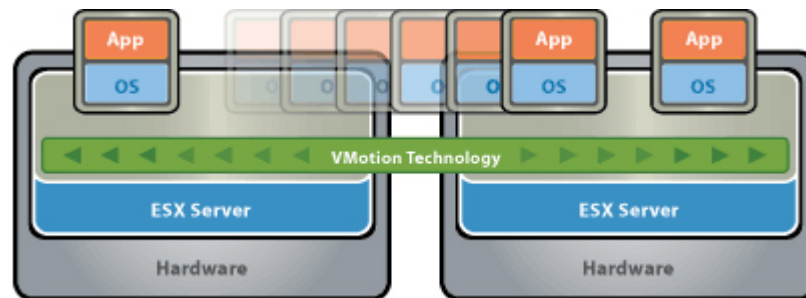


Figure 10. VMware VMotion

2.2. VMware Infrastructure 3 Installation

The VMware Infrastructure 3 installation steps are covered in detail in the document http://www.vmware.com/pdf/vi3_installation_guide.pdf.

2.3. ESX Server Host Time Synchronization

In general it is a good practice to synchronize the clock of an ESX Server host with a trusted source. The host clock can then be used as the source to synchronize the clocks on the virtual machines.

This section explains how to configure the ESX Server host machine to have synchronized clock. Section 2.4.3 shows how to use it as the source to synchronize the clocks on the virtual machines.

NTP is installed by default on ESX Server 3.0.1. To enable it, perform these steps (note: some of these steps require log-on to the service console):

- Edit the configuration files (/etc/ntp.conf, /etc/ntp/step-tickers, /etc/hosts) on the service console.
- Allow NTP client to go through the firewall.
- Restart NTPd.

These steps are elaborated in the following sections.

2.3.1. Configuration File Modification

This section explains which files to modify and the suggested content of each modified file. The modification requires log-on to the service console of the ESX Server host that is being configured.

| File Name | Suggested Content | Reference |
|-----------------------|--|---|
| /ect/ntp.conf | restrict 127.0.0.1 restrict default kod nomodify notrap server 0.pool.ntp.org server 1.pool.ntp.org server 2.pool.ntp.org driftfile /var/lib/ntp/drift | <ul style="list-style-type: none"> • http://www.pool.ntp.org/use.html for using NTP server pools • http://www.eecis.udel.edu/~mills/ntp/html/acopt.html for a full description of the access control commands. • http://www.eecis.udel.edu/~mills/ntp/html/notes.html for a more complete definition of the drift file. |
| /etc/ntp/step-tickers | 0.pool.ntp.org 1.pool.ntp.org 2.pool.ntp.org pool.ntp.org | |
| /etc/hosts | IP_ADDR 0.pool.ntp.org IP_ADDR 1.pool.ntp.org IP_ADDR 2.pool.ntp.org IP_ADDR pool.ntp.org | Adding these entries to /etc/hosts saves unnecessary DNS queries. It is for sake of optimization. |

2.3.2. Open the NTP Client Port

The second step is to open the NTP client port on the ESX Server Service Console. There are two ways to open the NTP client port:

- ESX Server configuration command on the Service Console
- VirtualCenter

ESX Server Service Console command

Run the following command on the ESX Server Service Console:

```
# esxcfg-firewall --enableService ntpClient
```

VirtualCenter

Use VirtualCenter 2.0 to open the NTP client port on the firewall, without having to log onto the Service Console. Follow the procedure below:

1. ESX Server host > Configuration tab
2. Click on Security Profile.

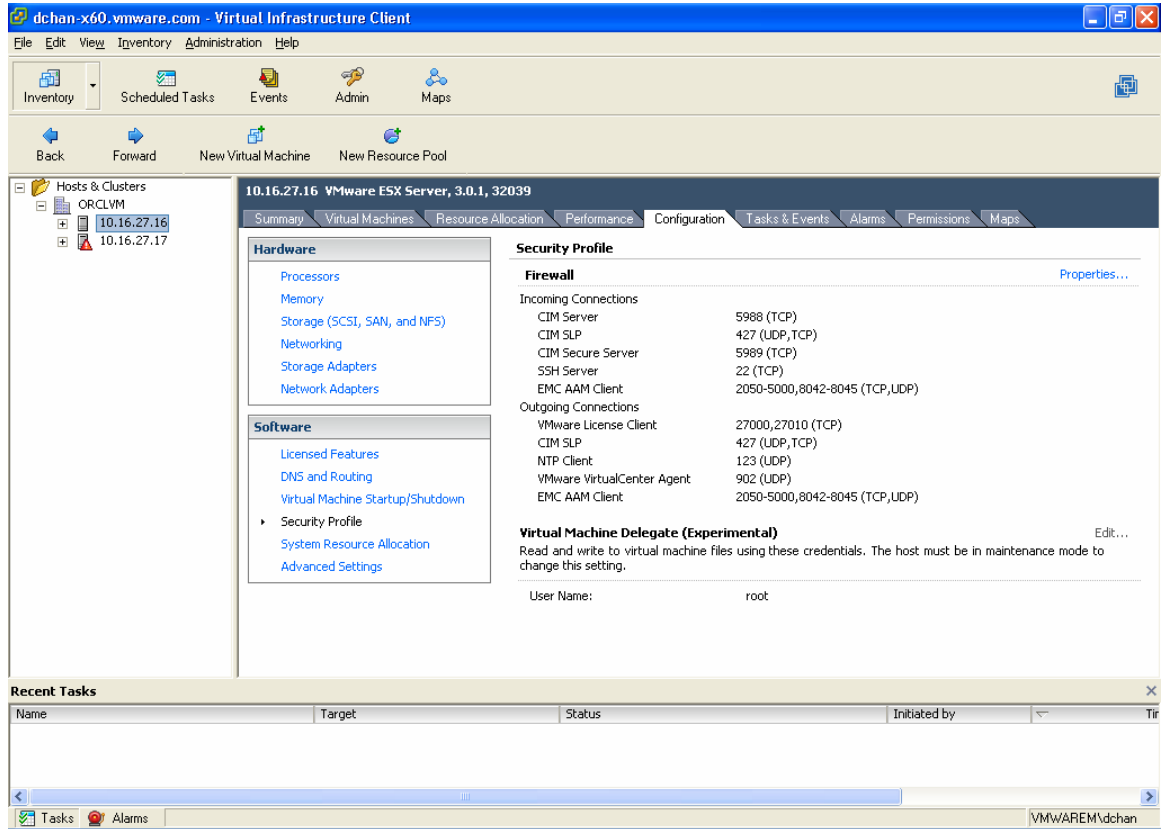


Figure 11. Security Profile for Configuration

3. Click on Properties in the upper right corner of the screen.
4. Check NTP Client to enable it.

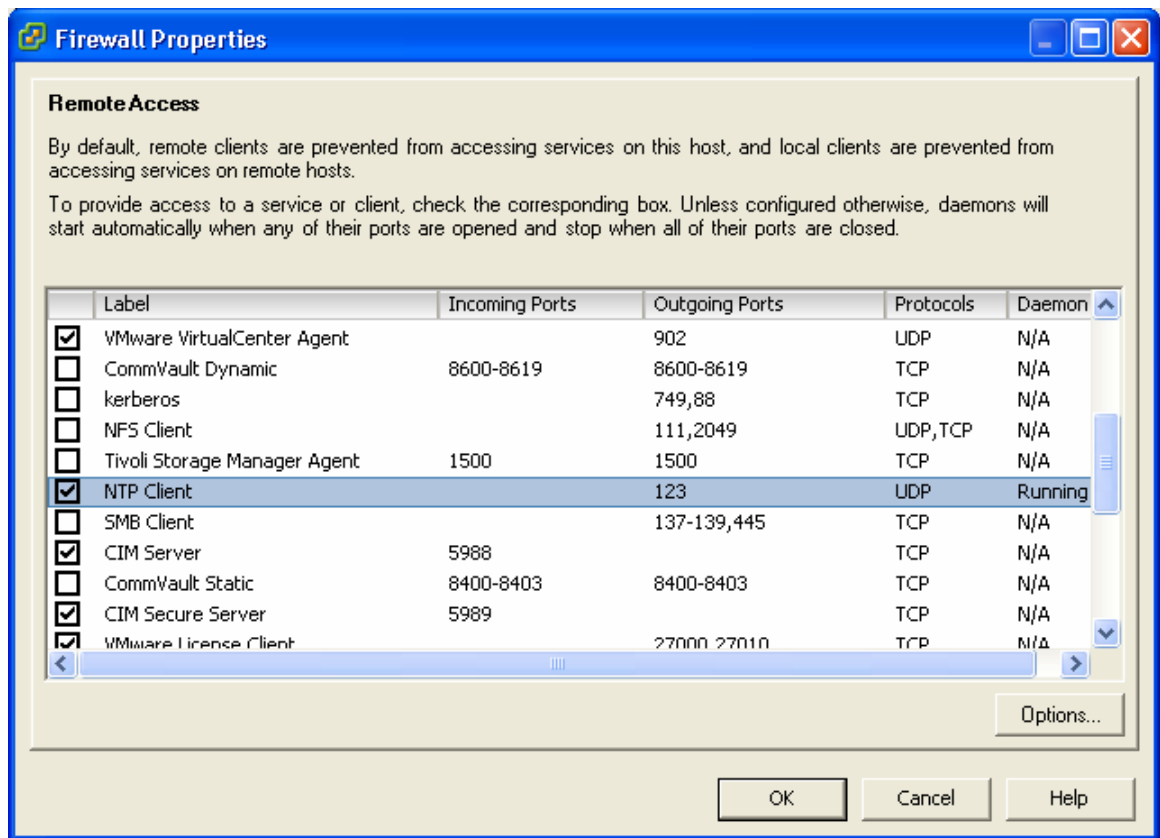


Figure 12. Firewall Properties Sheet for enabling NTP Client

2.3.3. Restarting and Monitoring the NTP Service

After the NTP service has been configured, it is necessary to restart the NTP daemon for the synchronization to take effect.

As root, do the following on the Service Console:

| | |
|---------------------------------|--|
| # service ntpd restart | restart the service |
| # chkconfig --level 345 ntpd on | To enable the NTP daemon to autostart when the server is rebooted |
| # hwclock --systohc | set the local hardware clock to the NTP synchronized local system time |

2.4. Add Virtual Machine to a Host

After ESX Server has been successfully installed on the host, virtual machines can be added to it. The following procedure shows how to add a virtual machine to a host using VMware VirtualCenter:

- Host > New Virtual Machine
- In the New Virtual Machine Wizard, enter the following:

| Wizard Type | Virtual Machine Configuration |
|-------------------------------|---|
| | Typical |
| Name and Folder | Virtual Machine Name: |
| | Oracle |
| Datstore | Select a datstore that is visible to both ESX Server host machines. In our scenario, we chose [ORCL] |
| Guest Operating System | Guest Operating System: |
| | Linux |
| | Version: Red Hat Enterprise Linux 4 |
| CPUs | Number of Virtual Processors: |
| | 2 |
| Memory | 700 MB. |
| | NOTE: 700MB is not enough for production deployment of Oracle RAC nodes. For production deployment, specify at least 922 MB. |
| Network | Number of NICs: |
| | 1 |
| | NIC 1: |
| | VM Network/ Connect at Power On |
| Virtual Disk Capacity | Disk Size: |
| | 20 GB |
| Ready to Complete | Click Finish |

2.4.1. Installation of Linux Guest Operating System

A new virtual machine needs to have a guest operating system installed. Installing Red Hat Enterprise Linux (RHEL 4 AS) is covered in this section. Follow the procedure below:

- Power on the virtual machine named Oracle.
- At the prompt for boot, enter the desired Linux version to install: **rhel4u2as**
- Connect the virtual CD ROM to an ISO image or the local CD ROM drive.
 - Click on the button Virtual CDROM (ide0:0) and follow the instructions.
- Select the language/keyboard type according to your settings.
- When prompted for installation method, select "Local CDROM"
- In the installation wizard, enter the following:

| | |
|------------------------------|--|
| Welcome Page: | Click Next |
| Disk Partition Setup: | Manually partition with Disk Druid |
| | Allocate disk space on sda drive by double-clicking on /dev/sda free space for the mount points (/ and /u01) and swap space. |
| | Mount Point: / File System Type: ext3 Start Cylinder: 1 End Cylinder: 910 |
| | File System Type: Swap Start Cylinder: 911 End Cylinder: 1170 |
| | Mount Point: /u01 File System Type: ext3 Start Cylinder: 1171 End Cylinder: 2610 |

| | |
|---------------------------------------|---|
| Boot Loader Configuration | Leave as default. |
| Network Configuration | For the HA scenario, only one Ethernet card is needed. |
| | Etho0: 10.16.27.26/255.255.0.0 |
| | Host Name: OracleVM |
| | DNS settings: Enter proper DNS servers and default gateway. |
| Firewall Configuration: | No Firewall Disable SELinux. (These configurations are needed for OCFS2/ASM to function) |
| Additional Language Support | Leave as default. |
| Time Zone Selection | Select desired time zone. |
| Set Root Password | Select desired root password |
| Package Installation Selection | <p>Customize software packages to be installed.</p> <ul style="list-style-type: none"> • Select X Window System. • Select GNOME Desktop Environment. • Select Editors. <ul style="list-style-type: none"> ○ Click on Details and select preferred text editor. • Select Graphical Internet. • Select Text-based Internet. • Select Office/Productivity. • Select Sound and Video. • Select Graphics. • Select Server Configuration Tools. • Select FTP Server. • Select Legacy Network Server. <ul style="list-style-type: none"> ○ Click on Details. ○ Select rsh-server. ○ Select telnet-server. • Select Development Tools. • Select Legacy Software Development. • Select Administration Tools. • Select System Tools. <ul style="list-style-type: none"> ○ Click on Details. ○ Select sysstat. • Select Printing Support. |
| About to Install | Click Next. |

2.4.2. Installation of VMware® Tools

After Linux has been installed on the virtual machine, VMware Tools should be installed using the following procedure:

- Select the virtual machine to install the VMware Tools.
- Right click on the virtual machine and select Install VMware Tools. This mounts the VMware tools image on CD-ROM.
- Mount CD-ROM (e.g. `mount /dev/cdrom /mnt/cdrom`).
- In `/mnt/cdrom`, there is a rpm file and a tar.gz file.
- `VMwareTools-3.0.1-32039.i386.rpm`
- `VMwareTools-3.0.1-32039.tar.gz`
- Use the command "`rpm -Uvh VMwareTools-3.0.1-32039.i386.rpm`" to install the package.
- After the package has been installed, run `/usr/bin/vmware-config-tools.pl` and follow the prompted instructions to install the network drivers as necessary.
- For more information about VMware Tools, refer to http://www.vmware.com/pdf/server_vm_manual.pdf.

2.4.3. Virtual Machine Time Synchronization

The clock of a virtual machine needs to be synchronized for proper Oracle Database installation. This procedure uses VMware tools to synchronize the clock on the Linux guest operating system with the clock on the ESX Server host:

- In the ESX Server Service Console, run the following command as root:

```
[root@rac1]#/usr/bin/vmware-toolbox&
```

- In the options tab, check the box labeled "Time Synchronization between the virtual machine and the console operating system."

NOTE: This synchronization mechanism works only if the clock on the guest operating system has been set to a previous time.



Figure 13. Use VMware Tools to Synchronize Time.

2.5. Linux Configuration for Oracle Database Installation

In preparation for Oracle Database installation, several Linux configuration steps need to be performed beforehand. These steps are outlined in the table below:

| | |
|--|---|
| <p>Create the user "oracle".</p> | <pre>As the root user, execute # groupadd oinstall # groupadd dba # mkdir -p /export/home/oracle /ocfs # useradd -d /export/home/oracle -g oinstall -G dba -s /bin/ksh oracle # chown oracle:dba /export/home/oracle /u01 # passwd oracle New Password: Re-enter new Password: passwd: password successfully changed for Oracle</pre> |
| <p>Create the environment file for user "oracle".</p> <p>Depending on the preferred shell, the commands on the right will need to be added to an appropriate file. One suggested candidate for bash is /export/home/oracle/.bash_profile.</p> | <pre>export ORACLE_BASE=/u01/app/oracle export ORACLE_HOME=\$ORACLE_BASE/product/10.2.0/db_1 export ORA_CRS_HOME=\$ORACLE_BASE/product/10.2.0/crs_1 export LD_LIBRARY_PATH=\$ORACLE_HOME/lib export PATH=\$ORACLE_HOME/bin:\$ORA_CRS_HOME/bin:/bin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/X11R6/bin umask 022</pre> |
| <p>Create the file system directory structure.</p> | <pre>As the user "oracle", execute # mkdir -p \$ORACLE_BASE/admin # mkdir -p \$ORACLE_HOME # mkdir -p \$ORA_CRS_HOME</pre> |
| <p>Increase the shell limits for the user "oracle".</p> | |
| <p>/etc/security/limits.conf</p> | <pre>oracle soft nproc 2047 oracle hard nproc 16384 oracle soft nofile 1024 oracle hard nofile 65536</pre> |
| <p>/etc/pam.d/login</p> | <pre>session required /lib/security/pam_limits.so</pre> |
| <p>/etc/profile</p> | <pre>if [\$USER = "oracle"]; then if [\$SHELL = "/bin/ksh"]; then ulimit -p 16384 ulimit -n 65536</pre> |

| | |
|---|---|
| | <pre> else ulimit -u 16384 -n 65536 fi fi </pre> |
| Required package versions (or later): | <ul style="list-style-type: none"> • binutils-2.15.92.0.2-10.EL4 • compat-db-4.1.25-9 • control-center-2.8.0-12 • gcc-3.4.3-9.EL4 • gcc-c++-3.4.3-9.EL4 • glibc-2.3.4-2 • glibc-common-2.3.4-2 • gnome-libs-1.4.1.2.90-44.1 • libstdc++-3.4.3-9.EL4 • libstdc++-devel-3.4.3-9.EL4 • make-3.80-5 • pdksh-5.2.14-30 • sysstat-5.0.5-1 • xscreensaver-4.18-5.rhel4.2 • libaio-0.3.96 • openmotif21-2.1.30-11.RHEL4.2 |
| Configure the kernel parameters. | |
| Modify the /etc/sysctl.conf file. | <pre> kernel.shmall = 2097152 kernel.shmmax = 2147483648 kernel.shmmni = 4096 kernel.sem = 250 32000 100 128 fs.file-max = 65536 net.ipv4.ip_local_port_range = 1024 65000 net.core.rmem_default = 1048576 net.core.rmem_max = 1048576 net.core.wmem_default = 262144 net.core.wmem_max = 262144 </pre> |
| Make changes effective. | <pre># /sbin/sysctl -p</pre> |
| Modify the /etc/hosts file. | <pre> # Do not remove the following line, or various programs. # that require network functionality will fail. 127.0.0.1 localhost 10.16.27.26 OracleVM OracleVM.vmware.com </pre> |
| Configure the hangcheck timer kernel module. | |
| Add to /etc/modprobe.conf | <pre> options hangcheck-timer hangcheck_tick=30 hangcheck_margin=180 </pre> |
| Load the module immediately. | <pre>modprobe -v hangcheck-timer</pre> |

2.6. Oracle ASM Installation and Configuration

Oracle ASM manages multiple storage devices, providing asynchronous I/O and disk redundancy. In order to take advantage of the disk redundancy provided by Oracle ASM, users either have to have multiple physical storage devices or have to create different partitions on a storage device. With VMware Infrastructure, users can create virtual disks that can be used by Oracle ASM for disk redundancy. This method provides boundary protection between the virtual disks without the rigid limits imposed by disk partitioning.

2.6.1. Virtual Disk Creation

This section explains how to add virtual disks in preparation for ASM deployment. These virtual disks are fully integrated to leverage the **Snapshot** capabilities provided by VMware Infrastructure 3.

In VMware VirtualCenter, perform the following:

- VM > Edit Settings.

In the Add Hardware Wizard, enter the following:

| | |
|--------------------------|--|
| Device Type | Hard Disk |
| Select a Disk | Create a new virtual disk. |
| Disk Capacity | Disk Capacity |
| | 5 GB |
| | Location |
| | Specify a datastore that is visible to both ESX Server hosts. In this scenario [ORCL] is chosen. |
| Advanced Options | Virtual Device Node |
| | SCSI (0:1) |
| | Mode |
| | Leave as default. |
| Ready to Complete | Finish. |

- Continue to add another three virtual disks of similar configuration. For Advanced Options, choose SCSI (0:X) to use the same controller.

2.6.2. Format the Virtual Disks in Linux

After adding the disks, power on the virtual machine. Follow the procedure below to format each of the virtual disks:

As root, format the disks using fdisk for /dev/sdb, /dev/sdc, /dev/sdd and /dev/sde:

```
[root@OracleVM ~]# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI
or OSF disklabel
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that, of course, the
previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be
corrected by w(rite)

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-652, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-652, default 652):
Using default value 652

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

Use the following command to check for proper configuration of the disk partitions:

```
[root@OracleVM ~]# fdisk -l

Disk /dev/sda: 21.4 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1           910     7309543+  83  Linux
/dev/sda2                911        1170     2088450   82  Linux
swap
/dev/sda3                1171        2610    11566800   83  Linux

Disk /dev/sdb: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1                1           652     5237158+  83  Linux

Disk /dev/sdc: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1                1           652     5237158+  83  Linux

Disk /dev/sdd: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdd1                1           652     5237158+  83  Linux

Disk /dev/sde: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sde1                1           652     5237158+  83  Linux
```

2.6.3. ASM Packages and Configuration

ASM installation requires the following RPM packages:

```
[root@OracleVM ~]# rpm -qa | grep asm
oracleasm-support-2.0.3-1
oracleasm-lib-2.0.2-1
oracleasm-2.6.9-22.ELsmp-2.0.3-1
```

Configure Oracle ASM:

```
[root@OracleVM ~]# /etc/init.d/oracleasm configure
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 [oracle]:
Default group to own the driver interface [dba]:
Start Oracle ASM library driver on boot (y/n) [y]:
Fix permissions of Oracle ASM disks on boot (y/n) [y]:
Writing Oracle ASM library driver configuration: [ OK ]
Scanning system for ASM disks: [ OK ]
```

Each of the new ASM disks needs to be configured as a raw device in **/etc/sysconfig/rawdevices**:

```
# This file and interface are deprecated.
# Applications needing raw device access should open regular
# block devices with O_DIRECT.
# raw device bindings
# format: <rawdev> <major> <minor>
#         <rawdev> <blockdev>
# example: /dev/raw/raw1 /dev/sda1
#         /dev/raw/raw2 8 5
/dev/raw/raw1      /dev/sdb1
/dev/raw/raw2      /dev/sdc1
/dev/raw/raw3      /dev/sdd1
/dev/raw/raw4      /dev/sde1
```

All the raw ASM disks need to have proper ownership and permission:

```
As root, execute:
# /sbin/service rawdevices restart
# chown oracle:dba /dev/raw/raw[1-4]
# chmod 660 /dev/raw/raw[1-4]
```

The following soft links are required. As the user "oracle", execute:

```
# ln -sf /dev/raw/raw1 /u01/oradata/devdb/asmdisk1
# ln -sf /dev/raw/raw2 /u01/oradata/devdb/asmdisk2
# ln -sf /dev/raw/raw3 /u01/oradata/devdb/asmdisk3
# ln -sf /dev/raw/raw4 /u01/oradata/devdb/asmdisk4
```

The permissions of the raw ASM disks need to be properly configured in /etc/udev/permissions.d/50-udev.permissions:

- Comment out the line that reads "raw/*:root:disk:0660"
- Add a new line that reads "raw/*:oracle:dba:0660"

The section for raw devices should read:

```
# raw devices
ram*:root:disk:0660
#raw/*:root:disk:0660
raw/*:oracle:dba:0660
```

Create the ASM disks for use in the Oracle Database deployment as below:

```
[root@OracleVM ~]# /etc/init.d/oracleasm createdisk VOL1 /dev/sdb1
Marking disk "/dev/sdb1" as an ASM disk: [ OK ]
[root@OracleVM ~]# /etc/init.d/oracleasm createdisk VOL2 /dev/sdc1
Marking disk "/dev/sdc1" as an ASM disk: [ OK ]
[root@OracleVM ~]# /etc/init.d/oracleasm createdisk VOL3 /dev/sdd1
Marking disk "/dev/sdd1" as an ASM disk: [ OK ]
[root@OracleVM ~]# /etc/init.d/oracleasm createdisk VOL4 /dev/sde1
Marking disk "/dev/sde1" as an ASM disk: [ OK ]
```

2.7. Installation of Oracle Database Software / Template

This section shows how to install the Oracle Database 10g software as well as a database template.

- Download the latest Oracle Database 10g software from www.oracle.com.
- For the purpose of this scenario, 10201_database_linux32.zip is used.
- Unzip 10201_database_linux.zip.
- As root, execute "xhost +".
- As the user "oracle", execute ./database/runInstaller in the directory where the software is unzipped.

| | |
|--|--|
| Select Installation Method | Advanced Installation |
| Specify Inventory directory and credentials | Full path of the inventory directory |
| | /u01/app/oracle/orainventory |
| | Operating System group name |
| | oinstall |
| Select Installation Type | Enterprise Edition |
| Specify Home Details | Name |
| | OraDb10g_home1 |
| | Path |
| | /u01/app/oracle/product/10.2.0/db_1 |
| Product-Specific Prerequisite Checks | All checks should pass. NOTE: In the case that the physical memory requirement does not pass, the customer can choose to ignore it. In best practices for Oracle RAC deployment, it is suggested that sufficient virtual memory be allocated to the virtual machine. |
| Select Configuration Option | Create a database. |
| Select Database Configuration | Select Advanced . |
| Summary | Click on Install . |
| DBCA | |
| Database Templates | Select General Purpose . |
| Database Identification | Global Database Name |
| | devdb |
| | SID Prefix |
| | devdb |
| Management options | Select Configure the Database with Enterprise Manager . |
| Database Credentials | Use the same password for all accounts. |

| | |
|----------------------------------|---|
| Storage Options | Select Automatic Storage Management (ASM) . NOTE: At this point, the customer will be prompted to run the script "/u01/app/oracle/product/10.2.0/db_1/bin/localconfig add". This is to make sure that the CSS service is configured to run properly. |
| Create ASM Instance | SYS password: <enter SYS password> Select Create initialization parameter file (IFILE) . |
| ASM Disk Groups | Click on Create New . |
| Create Disk Group | Disk Group Name DG1 Redundancy Normal Select Member Disks. /dev/raw/raw1 /dev/raw/raw2 Disk Group Name DG2 Redundancy Normal Select Member Disks. /dev/raw/raw3 /dev/raw/raw4 |
| Database File Locations | Select Use Oracle-Managed Files . Database Area: +DG1 |
| Recovery Configurations | Select Specify Flash Recovery Area . Flash Recovery Area: +DG2 Flash Recovery Area Size 2048M Select Enable Archiving . |
| Database Content | Select or deselect the sample schemas. |
| Initialization Parameters | Select Custom . Shared Memory Management: Automatic SGA Size: 200 MB PGA Size: 25 MB Modify the rest of the configuration as necessary. |
| Database Storage | Click on Next . |
| Creation Options | Select Create Database . Select Save as a Database Template . Name: asm_template NOTE: This asm_template can then be used to deploy other databases. |
| Confirmation | Click on OK. |

The last steps execute the following scripts as root:

```
/u01/app/oracle/oraInventory/orainstRoot.sh  
/u01/app/oracle/product/10.2.0/db_1/root.sh
```

2.8. *Creation of a Gold Image Template for a Single-instance Oracle Database*

The previous section explained how to install a database template that uses Oracle ASM in a virtual machine. This virtual machine has all of the configurations that are needed to deploy a database. It can be used as is, have snapshots taken for data protection, or serve as a Gold Image to deploy more ASM-enabled Oracle Database virtual machines.

The following procedure shows how to create a virtual machine template from this virtual machine by a simple conversion:

Use VMware VirtualCenter to do the following:

- Power down the virtual machine.
- VM > Convert to Template

Besides converting a virtual machine directly to a template, customers can also create a template from a virtual machine and leave the original virtual machine intact:

Use VMware VirtualCenter to do the following:

- Power down the virtual machine.
- VM > Clone to Template.

A progress bar on the bottom of the VirtualCenter display shows when template creation finishes. The next section shows how to use this template to create other virtual machines that contain a single-instance Oracle Database.

2.9. Deployment of a Single-Instance Oracle Database on a Virtual Machine

This section shows how to use a Gold Image template for rapid provisioning of one or more virtual machines with a single-instance Oracle Database:

In VMware VirtualCenter, do the following:

- OracleVM > Deploy Virtual Machine from this Template.
- In the Deploy Template Wizard, enter the following:

| | |
|----------------------------|--|
| Name and Folder | Virtual Machine Name |
| | OracleVM_A |
| | Virtual Machine Inventory Name |
| | <Choose the datacenter for this virtual machine> |
| Host/Cluster | <Choose the host to deploy this virtual machine in> |
| Datastore | <Choose a datastore that is visible to both ESX machines> |
| Guest Customization | Select Do not Customize . The guest operating system will be customized manually as described later. |
| Ready to Complete | Select Finish . |

- Power On OracleVM_A.
- When Linux is up, perform the following configurations:

| | |
|--|---|
| As root, execute: # system-config-network | <ul style="list-style-type: none"> - Make sure that the IP address and subnet mask are configured properly. - Make sure that the MAC address reflects the new hardware. Press the probe button in the GUI to perform a hardware probe. - Make sure that the hostname does not collide with any existing hosts. |
| As root, execute: # /etc/init.d/network restart | <ul style="list-style-type: none"> - Make sure that the new settings take effect. |
| Modify /etc/hosts. | <ul style="list-style-type: none"> - Make sure that the new hostname is entered in this file. An example is shown below: <pre>127.0.0.1 localhost 10.16.27.26 OracleVM OracleVM.vmware.com</pre> |

- Run DBCA.

As the user "oracle", execute the following to launch DBCA:

```
# /u01/app/oracle/product/10.2.0/db_1/bin/dbca&
```

In the DBCA wizard, enter the following as parameters:

| | |
|----------------------------------|--|
| Welcome | Select Next . |
| Operations | Select Create a Database . |
| Database Templates | Select asm_template . (This is the database template created in the Gold Image.) |
| Database Identification | Global Database Name: |
| | Devdb |
| | SID |
| | Devdb |
| Management Options | Select Configure the Database with Enterprise Manager . |
| Database Credentials | Use the Same Password for All Accounts. |
| ASM Credentials | <Specify SYS password specific to ASM> |
| ASM Disk Groups | Select DG1 and DG2. |
| Database File Locations | Leave as default. The configuration should be exactly the same as the asm_template created in the Gold Image. |
| Recovery Configuration | Leave as default. The configuration should be exactly the same as the asm_template created in the Gold Image. |
| Database Scripts | Select No scripts to run . |
| Initialization Parameters | Leave as default. The configuration should be exactly the same as the asm_template created in the Gold Image. |
| Database Storage | Next |
| Creation Options | Select Create a Database . |

- Check /etc/oratab.

In /etc/oratab, entries are of the form:

```
$ORACLE_SID:$ORACLE_HOME: <N|Y>
```

Make sure that the newly added database is enabled during startup. A sample /etc/oratab has the following:

```
+ASM:/u01/app/oracle/product/10.2.0/db_1:Y
devdb:/u01/app/oracle/product/10.2.0/db_1:Y
```

- Modify listener.ora.

Modify /u01/app/oracle/product/10.2.0/db_1/network/admin/listener.ora so that the listener knows about the devdb database. In order to do so, add the following to the SID_LIST:

```
(SID_DESC =
  (GLOBAL_DBNAME=devdb)
  (ORACLE_HOME = /u01/app/oracle/product/10.2.0/db_1)
  (SID_NAME = devdb)
)
```

A sample listener.ora reads as follows:

```
# listener.ora Network Configuration File:
/u01/app/oracle/product/10.2.0/db_1/network/admin/listener.ora
# Generated by Oracle configuration tools.

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME=devdb)
      (ORACLE_HOME = /u01/app/oracle/product/10.2.0/db_1)
      (SID_NAME = devdb)
    )
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /u01/app/oracle/product/10.2.0/db_1)
      (PROGRAM = extproc)
    )
  )
)

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
      (ADDRESS = (PROTOCOL = TCP)(HOST = OracleVM)(PORT = 1521))
    )
  )
)
```

- Manual start the listener and the database.

In the case that **the listener is not running**, run the following command:

```
# u01/app/oracle/product/10.2.0/db_1/bin/lsnrctl start
```

In the case that **the database is not running**, run the following command:

```
# u01/app/oracle/product/10.2.0/db_1/bin/dbstart
```

- Auto start the listener and database during a reboot.

It is useful in a failover scenario to configure the listener and database to start during boot time. Follow the procedure below to do so:

1. Make sure that “/etc/init.d/init.cssd enable” has been run.
2. Modify /etc/inittab file moving the cssd line to before the init 3 servicing:

Original /etc/inittab reads:

```
l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
(...)
h1:35:respawn:/etc/init.d/init.cssd run >/dev/null 2>&1 </dev/null
```

Modified /etc/inittab reads:

```
l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
h1:35:respawn:/etc/init.d/init.cssd run >/dev/null 2>&1 </dev/null
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
(...)
```

3. Edit /etc/rc.local to include the following lines:

```
su – oracle –c “lsnrctl start”  
su – oracle –c “dbstart”
```

After this procedure, a single-instance Oracle Database 10g is deployed on a separate virtual machine. In case of operating system failures, the database is automatically restarted when the virtual machine reboots. Customers who would like to leverage the power of the VMware High Availability feature can use the information provided in the next section. When the VMware HA feature is used during system failure, both the virtual machine and the Oracle environment are restarted automatically on an available ESX Server host.

2.10. Using the VMware HA Feature for Single-Instance Oracle Database Failover

This setup uses two ESX Server hosts, one as the original server and the other as the failover target. This section explains how to use VMware High Availability to provide a failover mechanism for single-instance Oracle Databases.

2.10.1. Network Card Redundancy

The VMware HA feature depends on the heartbeats reported by VMware Tools via a virtual switch connected to a particular NIC card. If the hardware profile allows, it is desirable to have redundancy built into the network configuration of a host, so that there are two NIC cards in each of the two ESX Server hosts. There are two options when building network card redundancy for the ESX Server hosts to minimize the effects of a failed NIC card:

- NIC teaming
- Redundant service console port on a separate switch

2.10.1.1. NIC Teaming

NIC teaming groups NIC cards together to serve one virtual switch. In order to add another NIC card to a virtual switch, do the following:

1. Desired Host > Configuration Tab > Networking
2. Click on Properties for vSwitch0.
3. Network Adaptors Tab
4. Click on Add.

In the Add Adaptor Wizard, enter the following:

| | |
|------------------|---|
| Adaptor | Select the network adaptor. |
| NIC order | Have the new NIC card moved down to be a Standby adapter. |
| Summary | Finish. |

2.10.1.2. Redundant Service Console Port

An alternative for creating network card redundancy is to build a redundant Service Console port that is connected to a separate NIC card in case the other one fails.

In VMware VirtualCenter, do the following:

- Desired Host > Configuration Tab > Networking
- Click on **Add Networking**.
- In the **Add Network Wizard**, enter the following:

| | |
|----------------------------|---|
| Connection Type | Select Service Console |
| Network Access | Create a virtual switch using a NIC that has not been connected to any virtual switch. In this scenario, it is vmnic1. |
| Connection Settings | Port Group Properties |
| | Network Label: Service Console 2 |
| | IP Settings |
| | IP Address: 10.16.27.29/Subnet Mask: 255.255.0.0 |
| Summary | Finish. |

2.10.2. VMware HA Cluster Creation

This section explains how to create a cluster with the VMware HA feature by using the following procedure in VMware VirtualCenter:

- Data Center > New Cluster
- In the New Cluster Wizard, enter the following:

| | |
|--------------------------|--|
| Cluster Features | Name |
| | ORACLE CLUSTER |
| | Cluster Features |
| | VMware HA |
| VMware HA | Host Failures |
| | Number of host failures allowed: 1 |
| | Admission Control |
| | Select Do not start virtual machines if they violate availability constraints . |
| Ready to Complete | Finish. |

- Drag-and-drop both of the ESX Server hosts to ORACLE CLUSTER. After this step, ORACLE CLUSTER will contain both ESX Server hosts for failover purposes.

2.10.3. VMware HA Benefits

With the VMware HA cluster deployed, the virtual machine is restarted on another ESX Server host in the cluster if the system running the Oracle virtual machine stops due to hardware failure. This configuration provides a cost-effective high availability solution.

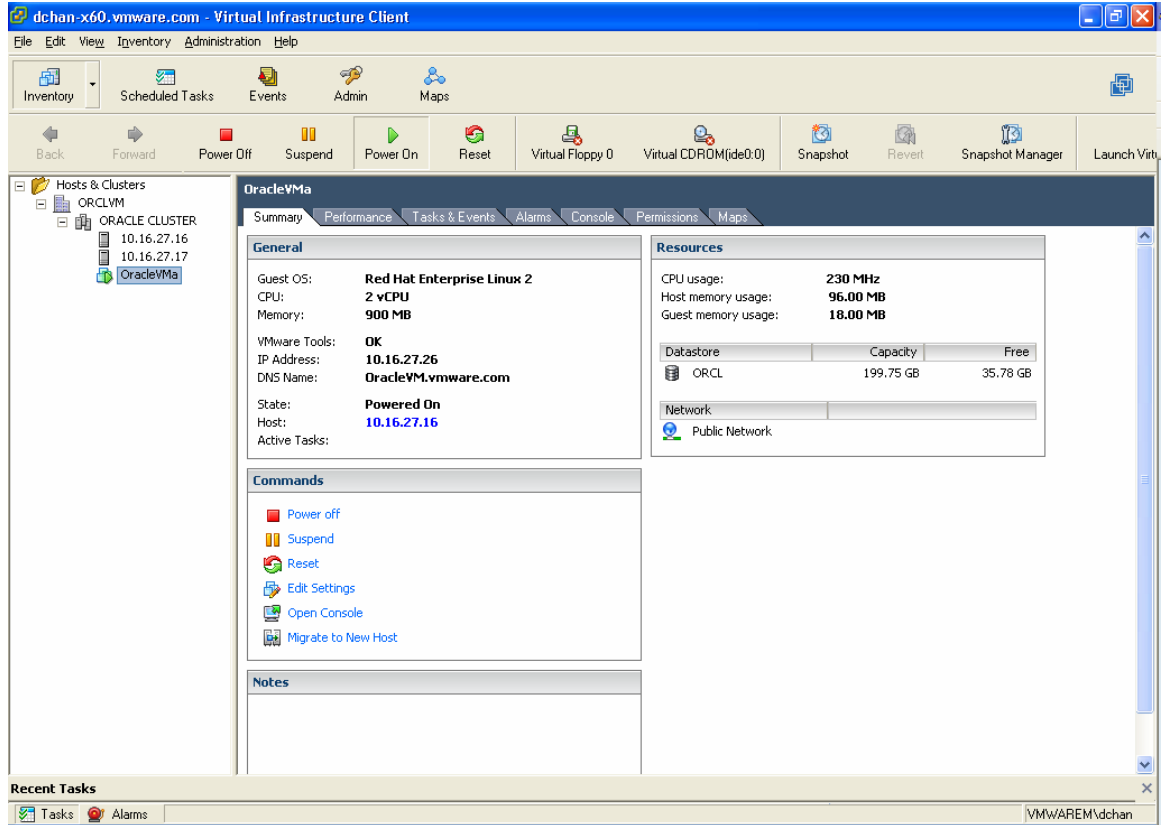


Figure 14. VMware HA cluster

2.11. VMware VMotion for Single-Instance Oracle Database

The VMware HA feature can be leveraged to provide failover availability across the machines in the ESX Server cluster in case of unplanned downtime. VMotion can migrate live Oracle virtual machines from one ESX Server host to another with no disruption to end users. In case of scheduled host maintenance, perform a live migration with VMware VMotion to ensure that the Oracle Database continues to serve all user requests.

For VMotion to function properly, make sure the following conditions are met:

- The hardware of the two ESX Server hosts, in particular the CPU, is compatible.
- All the virtual switches are configured in the same way for both ESX Server hosts.
- A VMKernel port is configured on both ESX Server hosts.

2.11.1. VMKernel Port Configuration

A VMKernel port has to be configured on a virtual switch for VMware VMotion to function. Make sure that both ESX Server host machines have the same port configuration by using the following procedure:

- Desired Host > Configuration Tab > Networking
- Click on **Add Networking**.
- In the **Add Network Wizard**, enter the following:

| | |
|----------------------------|---|
| Connection Type | Select VMKernel . |
| Network Access | Use vSwitch0. |
| Connection Settings | Port Group Properties |
| | Network Label: VMKernel |
| | Check the box for "Use this port group for VMotion™." |
| | IP Settings |
| | IP Address: 10.16.27.30/Subnet Mask: 255.255.0.0 |
| Summary | Finish. |

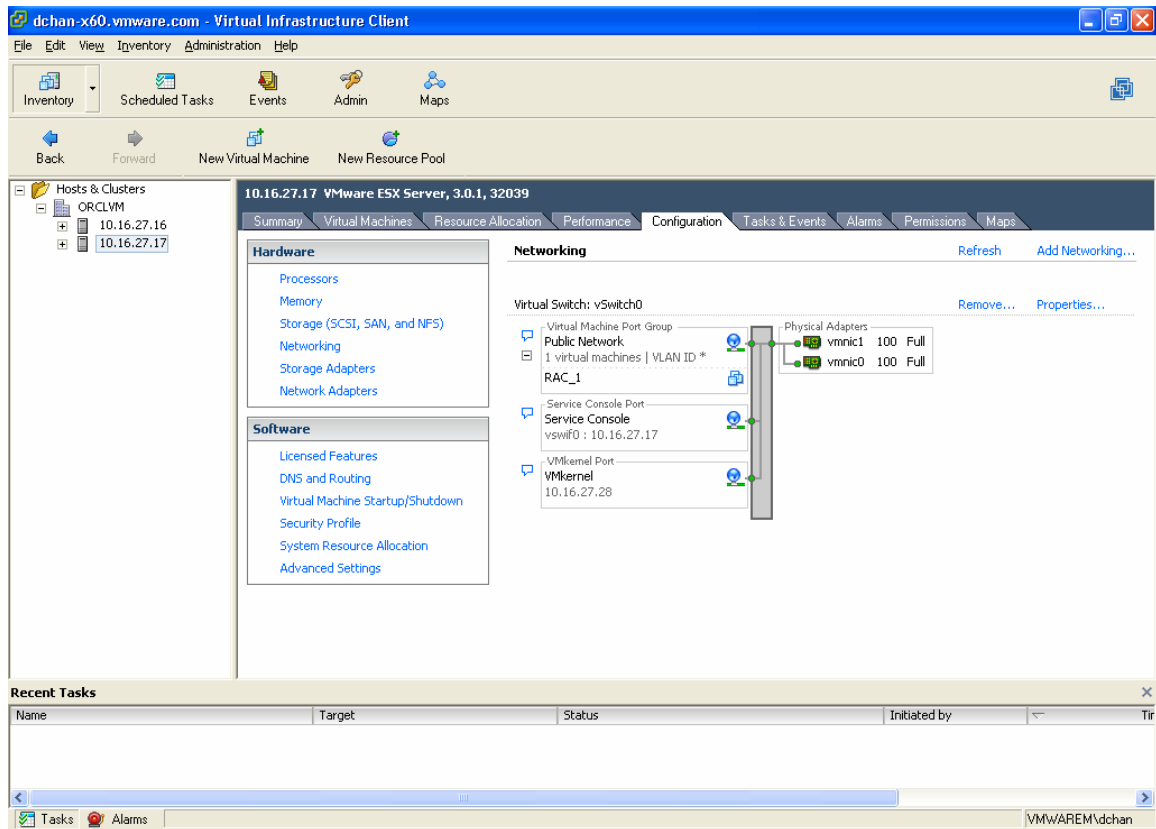


Figure 15. VMKernel Port Configuration

2.11.2. Live Migration with VMware VMotion

Once VMotion is configured properly, customers can perform a live migration of any or all of the virtual machines from the active server to the stand-by server. With this feature, customers can perform server maintenance with minimal service interruption.

To perform a live migration on a virtual machine, follow this procedure:

- VM > Migrate
- In the Migrate Virtual Machine Wizard, enter the following:

| | |
|-----------------------------|--|
| Select Destination | Select the redundant ESX host. 10.16.27.17 is chosen in this scenario. |
| Select Resource Pool | Leave as default. |
| Migration Priority | Select High Priority . |
| Ready to Complete | Finish. |

Once the virtual machine has been migrated to the stand-by host, it is safe to put the original host in maintenance mode and perform maintenance work on it.

3. Deployment of Oracle Database 10g RAC

This section explains how to deploy a VMware Infrastructure 3 solution to run the Oracle Database 10g RAC software.

Oracle Database 10g RAC has two major components: Clusterware and Database engine. By deploying both components together, IT professionals create a cluster of nodes that can work on the same databases via different instances.

In a nutshell, Clusterware is responsible for maintaining the cluster configuration, monitoring node health, and moving applications to another node in the cluster when a node fails. With Clusterware, databases can be deployed in a highly available, load-balanced, and clustered environment.

This section explains how to use VMware Infrastructure 3 to achieve the Install-Once-Run-Everywhere (**IORE**) deployment of Oracle Database 10g RAC. Included here are step-by-step instructions for creating and using a Gold Image to deploy Oracle Database 10g RAC. The VMware Infrastructure 3 solution can make the deployment of Oracle Database 10g RAC more efficient and cost-effective while enhancing the scalability, availability and reliability of the Oracle RAC environment.

The flow of deployment steps for Oracle RAC on VMware Infrastructure 3 is very similar to the one for single-instance Oracle Database, as illustrated in Figure 16.

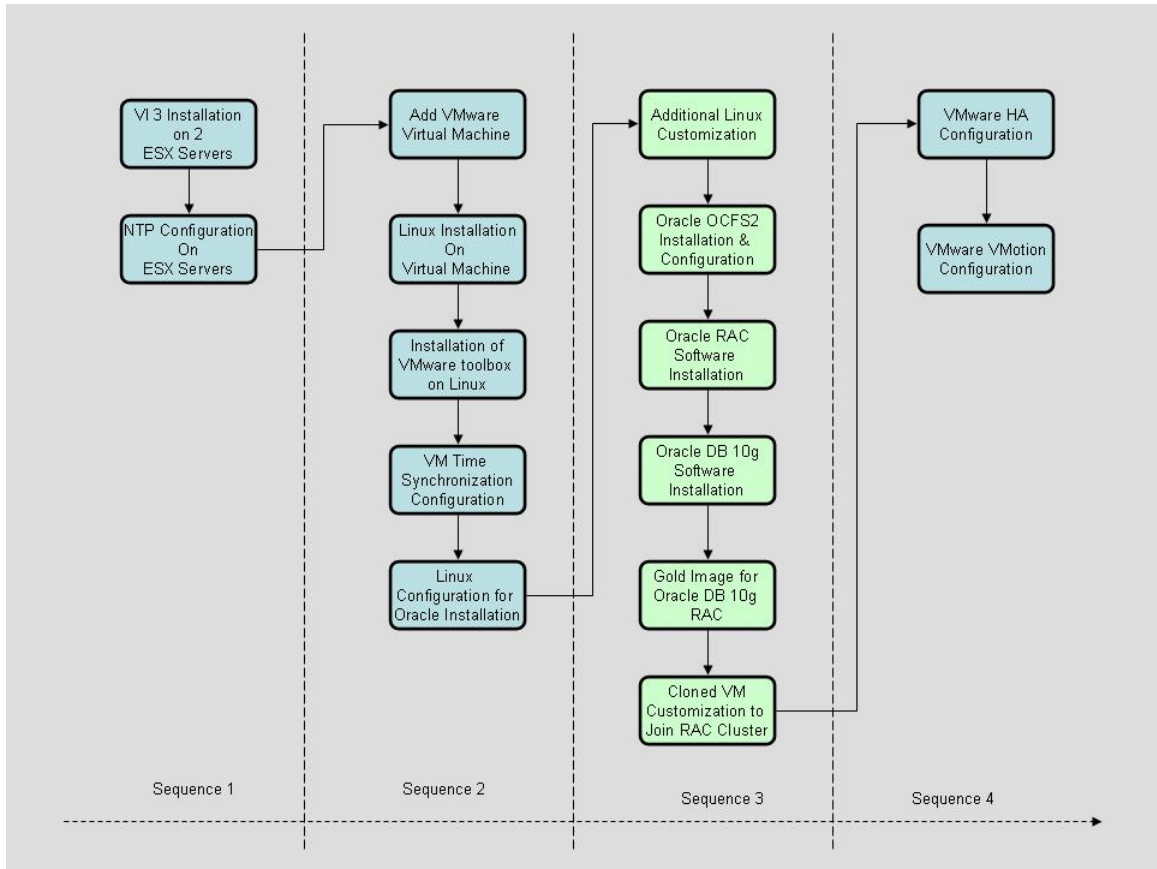


Figure 16. Flow Chart of Oracle RAC Deployment on VMware Infrastructure 3.

3.1. *Solution Architecture*

The physical environment for this solution is exactly the same as the one discussed in section 2.1. In this scenario, both ESX Server host machines are active, and each has one or more virtual machines that have Oracle Database 10g RAC software installed. The Oracle Database 10g RAC nodes in the virtualized environment work in exactly the same way they would if they were deployed in a physical environment, with the additional benefits that VMware Infrastructure 3 features provide.

NOTE: For an Oracle Database 10g RAC production deployment on VMware Infrastructure, it is recommended that each RAC node be deployed on a separate ESX Server host.

NOTE: The RAC nodes need a shared storage system (e.g. SAN) for database access. The shared storage system is provisioned for shared data access from all the available RAC nodes.

Figure 17 shows a sample of recommended solution architecture for an Oracle RAC Deployment on VMware Infrastructure. The sample solution architecture comprises two ESX Server hosts and a shared storage system. One RAC node is installed on each ESX Server host, forming a 2-node RAC cluster. Furthermore, the ESX Server hosts are configured for VMware HA Clustering.

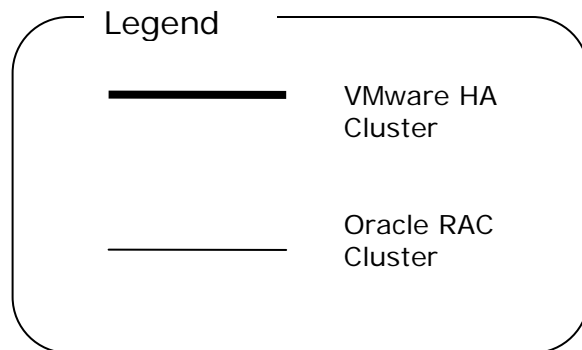
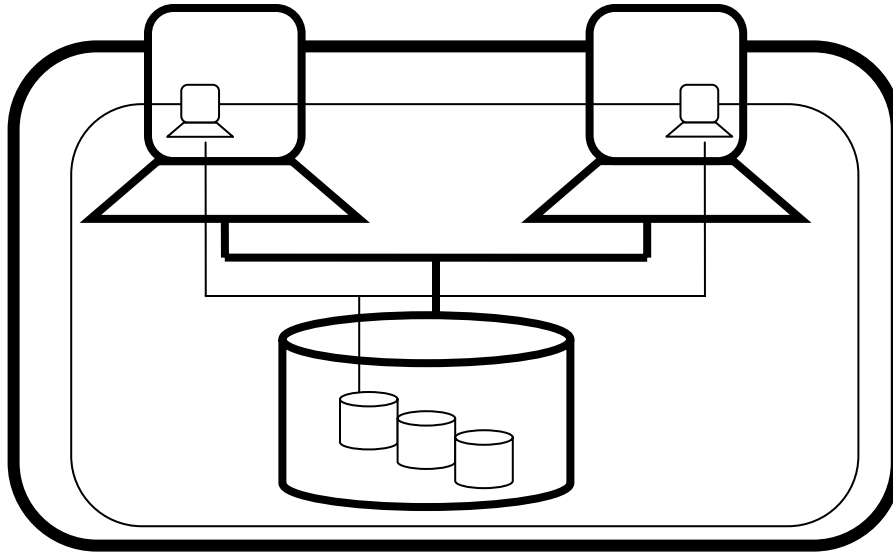


Figure 17. Illustration of deployment of Oracle RAC cluster on a VMware HA cluster.

3.2. *VMware Environments for RAC Software*

In order for the Oracle RAC software to work properly, several server requirements must be met. This section explains them in detail.

3.2.1. Network Configuration

3.2.1.1. NIC Cards

For the Oracle Clusterware deployment, it is essential to have a public interface and a private interface. There should be at least **two functional NIC cards** installed in each of the ESX Server hosts. One NIC should be connected to the public network and the other should be on an internal network that both of the ESX Server hosts can access.

In addition, two virtual switches need to be configured: one for the public network and one for the private network. Each of them is mapped to the corresponding physical network cards. The following section details the process for configuring the virtual switches.

3.2.1.2. Virtual Switch

This deployment needs **two virtual switches**. By default, a virtual switch called **vSwitch0** connecting VM Network and the Service Console is installed. This vSwitch0 will connect to the **public network** for the Oracle RAC deployment. It is important to ensure that vSwitch0 is connected to the public NIC card.

The virtual switches have to be **configured in exactly the same way** on both ESX Server host machines in order for VMware features for high availability to function properly.

Follow these steps to add another virtual switch for the **private network** using VMware VirtualCenter:

- In the Hosts & Clusters View, select the ESX Server host where the virtual switch will be added.
- Go to the Configuration Tab.

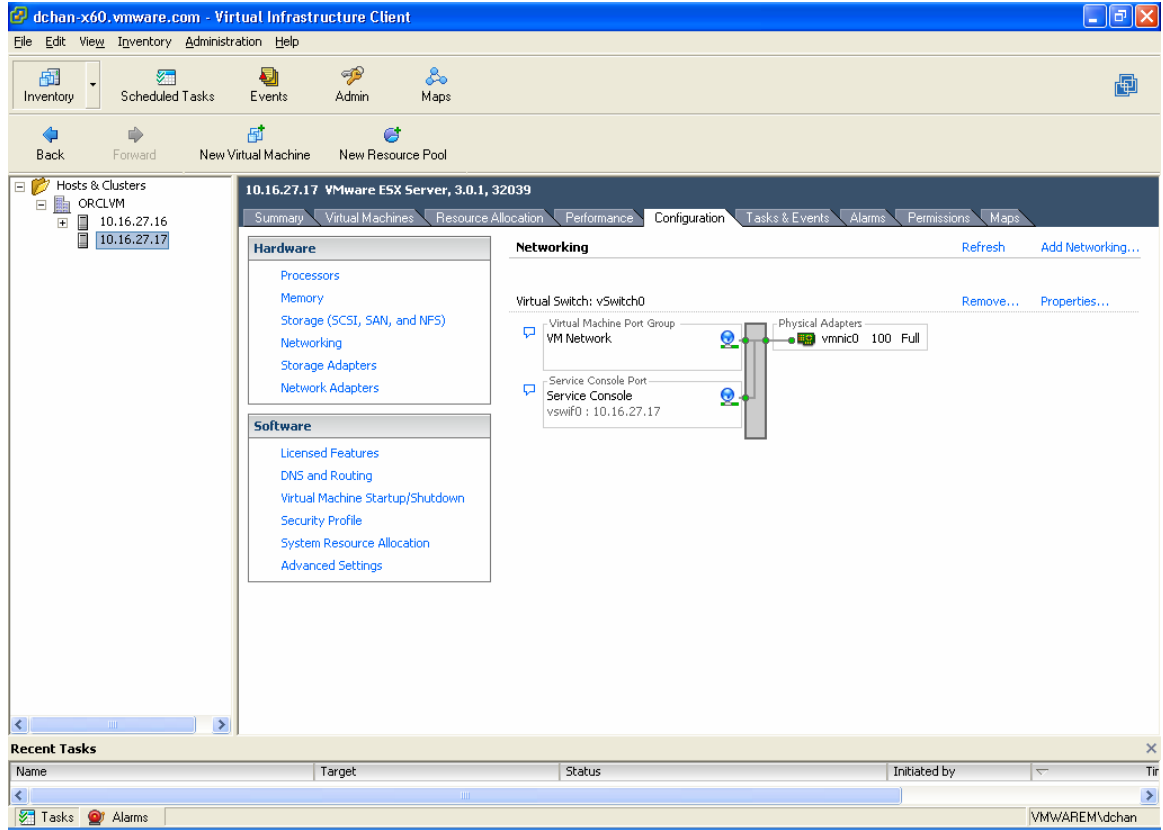


Figure 18. Virtual Switch Configuration

- Click on **AddNetworking** in the upper-right-hand corner of the screen. The Add Network Wizard will open.
- Use the following parameters for configuration:

| | | |
|----------------------------|--|-----------------|
| Connection Type | Virtual Machine | |
| Network Access | Create a new virtual switch (make sure that the private NIC is selected for this purpose). | |
| Connection Settings | Network Label | Private Network |
| | VLAN ID | Leave as blank. |
| Summary | Click Finish. | |

- There are now two virtual switches after this setup is completed.

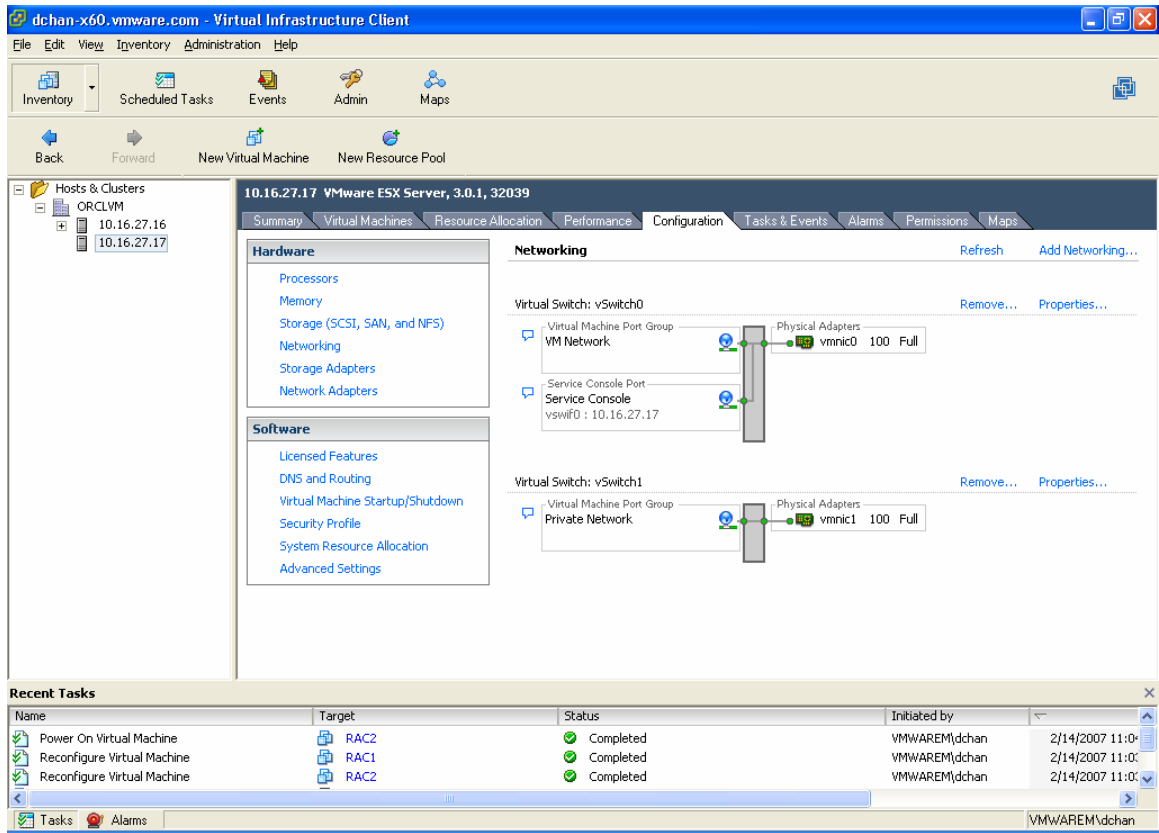


Figure 19. Networking view of 2 Virtual Switches.

3.3. Storage Configuration

Oracle Database 10g RAC requires shared storage that each of the Oracle RAC nodes can access for the data files. This section explains the shared storage configuration for the RAC deployment.

3.3.1. Add New Physical Shared Storage

To add a shared storage system (e.g. new LUN) after ESX Server has been installed, follow the procedure below to rescan and add it to the infrastructure using VMware VirtualCenter.

- Select the desired host server.
- Click on the Configuration tab.
- Click on **Add Storage...** in the top right corner.
- In the Add Storage Wizard, enter the following:

| | |
|----------------------------|---|
| Disk/LUN | Select Disk/LUN. |
| Device Location | A list of available LUNs should show on the window to the right. Select the device to add. |
| Current Disk Layout | (Indication that the disk is blank should be shown) Click Next. |
| Properties | Datastore Name ORCL |
| Formatting | Keep the defaults. Make sure that the checkbox for Maximize Capacity has been checked. |
| Ready to Complete | Click Finish. |

After this configuration, a datastore called ORCL is ready for consumption.

3.3.2. Shared Disk Creation

There are several ways to create a shared disk in VMware Infrastructure 3. This section covers shared virtual disk creation and raw disk mapping. Both methods call for a shared storage system (e.g. SAN). Depending on their requirements, customers can use either method to create a shared virtual disk.

3.3.2.1. Shared Virtual Disk Creation

A shared LUN can be formatted as VMFS3. This formatting can be done during ESX Server installation or by VirtualCenter 2.0 as described in section 3.3.1. After the LUN has been formatted as VMFS3, a utility called `vmkfstools` can be invoked on the ESX Server Service Console to create virtual disks.

A virtual disk needs to be created using **thick mode** in order to be used as a shared disk. Use the command listed below for creating a virtual disk of 20G.

```
[root@rac1]# vmkfstools -c 20G -d eagerzeroedthick -a lsilogic  
/vmfs/volumes/DATASTORE_NAME/VM_NAME/shared_disk.vmdk.
```

NOTE: Use the `-a` switch to specify lsilogic as the adaptor type. The default buslogic cannot be used in this scenario.

NOTE: Customers can substitute the desired size specification in place of 20G using the `-c` switch.

NOTE: Customers can use the desired path and name of the vmdk file created for this virtual disk.

3.3.2.2. Raw Disk Mapping

After a LUN has been added to VMware Infrastructure 3, it is possible to map it directly to the system as a virtual disk. This method is known as Raw Disk Mapping (RDM). A virtual disk created using this method can be used in the same way as a virtual disk created using vmkfstools, as described in section 3.3.2.

This section discusses how to configure raw disk mapping on VMware VirtualCenter. After the following procedure, a shared virtual disk is added to the hardware profile of the corresponding virtual machine.

In VMware VirtualCenter, do the following:

- VM > Edit Settings
- Click on **Add**.
- In the Add Hardware wizard, enter the following:

| | |
|---------------------------|---|
| Device Type | Select Hard Disk. |
| Select a Disk | Select Raw Disk Mapping. |
| Select Target LUN | Select the LUN desired. |
| Select Datastore | Select Store with Virtual Machine . |
| Compatibility Mode | Select Physical (for the database to access it directly) . |
| Advanced Options | Node <select a node that will be using a separate controller, e.g. SCSI (1:0)> |
| Ready to Complete | Click on Finish . |

3.4. Gold Image for Oracle RAC

The next step is to create a virtual machine. The following procedure shows how to install an initial RAC node on a virtual machine that can later be converted to a Gold Image for future deployment.

3.4.1. Add Virtual Machine to a Host

3.4.1.1. Add Virtual Machine

In VMware VirtualCenter, do the following:

- Host > New Virtual Machine
- In the New Virtual Machine Wizard, enter the following:

| Wizard Type | Virtual Machine Configuration |
|---------------------------------------|---|
| | Typical |
| Name and Folder | Virtual Machine Name: |
| | RAC_1 |
| Datstore | Select a datstore that is preferably visible to both ESX host machines. In this scenario, [ORCL] is chosen. |
| Guest Operating System | Guest Operating System: |
| | Linux |
| | Version: Red Hat Enterprise Linux 4 |
| CPUs | Number of Virtual Processors: |
| | 2 |
| Memory | 700 MB |
| | NOTE: 700MB is not enough for production deployment of Oracle RAC nodes. For production deployment, specify at least 922 MB. |
| Network | Number of NICs: |
| | 2 |
| | NIC 1: |
| | vSwitch0 / Connect at Power On |
| | NIC 2: |
| Private Network / Connect at Power On | |
| Virtual Disk Capacity | Disk Size: |
| | 20 GB |
| Ready to Complete | Click Finish. |

3.4.1.2. Add Virtual Shared Disk

A shared virtual disk is essential for the success of this deployment. In the following steps, a shared virtual disk will be created in a shared datastore called ORCL.

As root, run the following in the Service Console of one of the ESX Server host machines:

```
# vmkfstools -c 5G -d eagerzeroedthick -a lsilogic
/vmfs/volumes/ORCL/RAC_1/ocfs_disk.vmdk
```

NOTE: Customers can also opt to create a shared virtual disk using Raw Device Mapping (RDM). Refer to section 3.3.2.2 for the corresponding procedure. The following procedure is for adding such a shared virtual disk that has been created beforehand.

In VMware VirtualCenter, do the following:

- VM > Edit Settings
- In Add Hardware Wizard, enter the following:

| | |
|-----------------------------|---|
| Device Type | Hard Disk |
| Select a Disk | Use an existing virtual disk. |
| Select Existing Disk | Browse to the vmdk file that has been created in thick mode or using RDM. |
| Advanced Options | Virtual Device Node |
| | SCSI (1:0) |
| | Mode |
| | Independent/Persistent |
| Ready to Complete | Finish |

- Enable SCSI Bus Sharing by choosing **Physical** as indicated in the screenshot below:

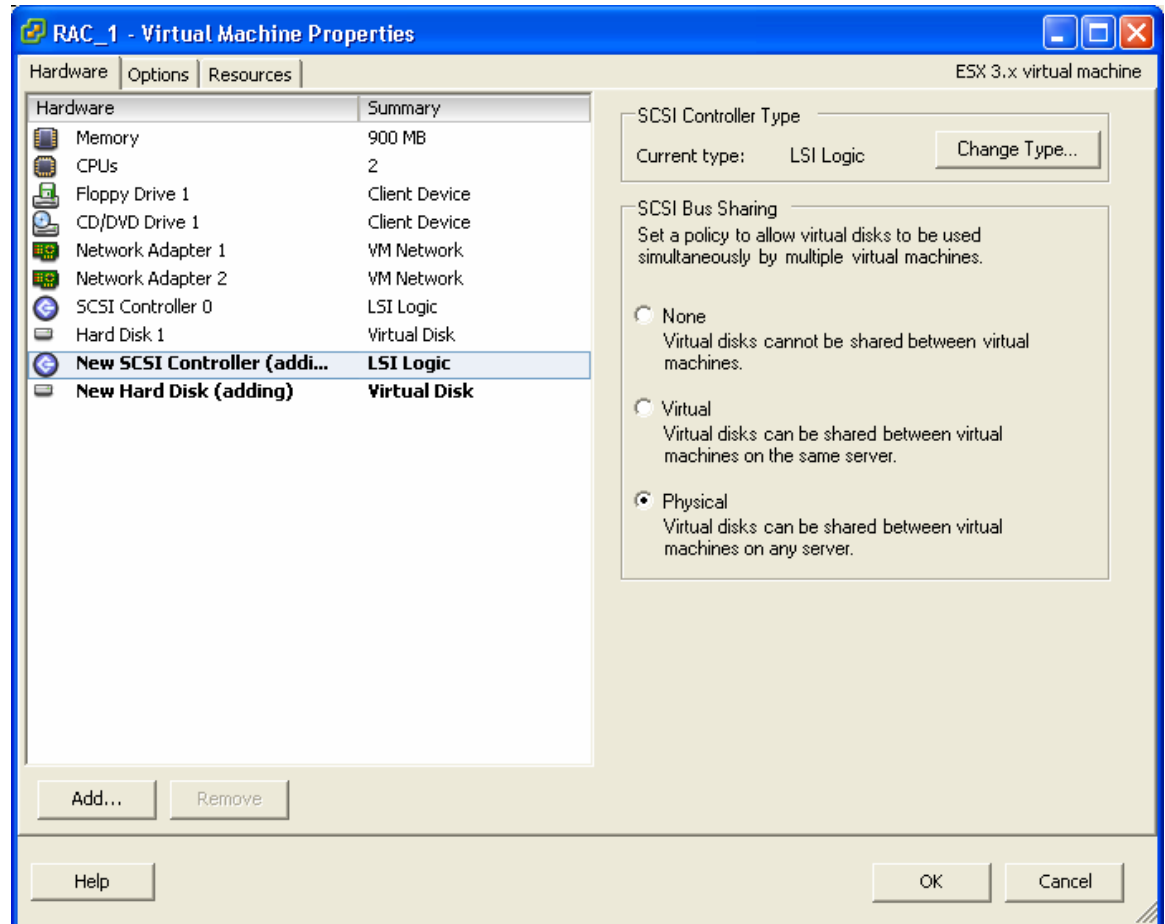


Figure 20. Add a Hard Disk to a Virtual Machine

3.5. Gold Image Creation for RAC Deployment

3.5.1. Installation of Linux Guest Operating System

Use the following procedure to install the Linux guest operating system. The steps use RHEL 4 AS for the sake of illustration:

- Power on the virtual machine rac_1.
- At the prompt for boot, enter the desired Linux version to install.
- Connect the Virtual CD ROM to an ISO image or the local CD ROM drive.
 - Click on the button Virtual CDRom (ide0:0) and follow the instructions.
- Select the language/keyboard type according to desired settings.
- When prompted for the installation method, select "Local CDRom"
- In the installation wizard, enter the following:

| | |
|------------------------------|--|
| Welcome Page: | Click Next |
| Disk Partition Setup: | Manually partition with Disk Druid. |
| | Allocate disk space on sda drive by double-clicking on /dev/sda free space for the mount points (/ and /u01) and swap space. |
| | Mount Point: / File System Type: ext3 Start Cylinder: 1 End Cylinder: 910 |
| | File System Type: Swap Start Cylinder: 911 End Cylinder: 1170 |
| | Mount Point: /u01 File System Type: ext3 Start Cylinder: 1171 End Cylinder: 2610 |

| | |
|---|--|
| Boot Loader Configuration | Leave as default. |
| Network Configuration | Configure the two interfaces to have static IPs and appropriate subnet masks: |
| | Etho0: 10.16.27.18/255.255.0.0 |
| | Etho1: 192.168.2.31/255.255.255.0 |
| | Host Name: rac_1 |
| DNS settings: Enter proper DNS servers and default gateway. | |
| Firewall Configuration: | No Firewall Disable SELinux. (These configurations are needed for OCFS2 to function.) |
| Additional Language Support | Leave as default. |
| Time Zone Selection | Select time zone. |
| Set Root Password | Select root password. |
| Package Installation Selection | <p>Customize software packages to be installed.</p> <ul style="list-style-type: none"> • Select X Window System. • Select GNOME Desktop Environment. • Select Editors. <ul style="list-style-type: none"> ◦ Click on Details and select your preferred text editor. • Select Graphical Internet. • Select Text-based Internet. • Select Office/Productivity. • Select Sound and Video. • Select Graphics. • Select Server Configuration Tools. • Select FTP Server. • Select Legacy Network Server. <ul style="list-style-type: none"> ◦ Click on Details. ◦ Select rsh-server. ◦ Select telnet-server. • Select Development Tools. • Select Legacy Software Development. • Select Administration Tools. • Select System Tools. <ul style="list-style-type: none"> ◦ Click on Details. ◦ Select sysstat. • Select Printing Support. |
| About to Install | Click Next. |

3.5.1.1. Installation of VMware Tools

Please refer to Section 2.4.2.

3.5.1.2. VM Time Synchronization

Please refer to Section 2.4.3.

3.5.2. OCFS2

This section shows how to use OCFS2 instead of ASM as shown in Section 2.6.

3.5.2.1. Prerequisites for OCFS2 Installation

Ensure the following is configured properly in order for OCFS2 to function. Launch `/usr/bin/system-config-securitylevel` to check:

- The Firewall should be disabled.

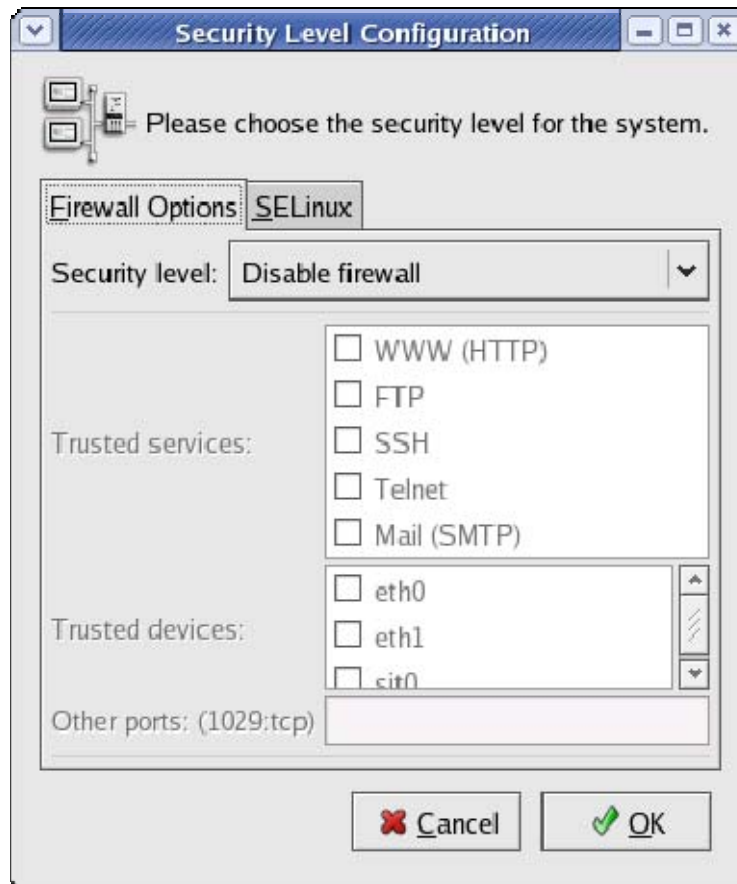


Figure 21. Firewall disabled in Linux.

- SELinux should be disabled.

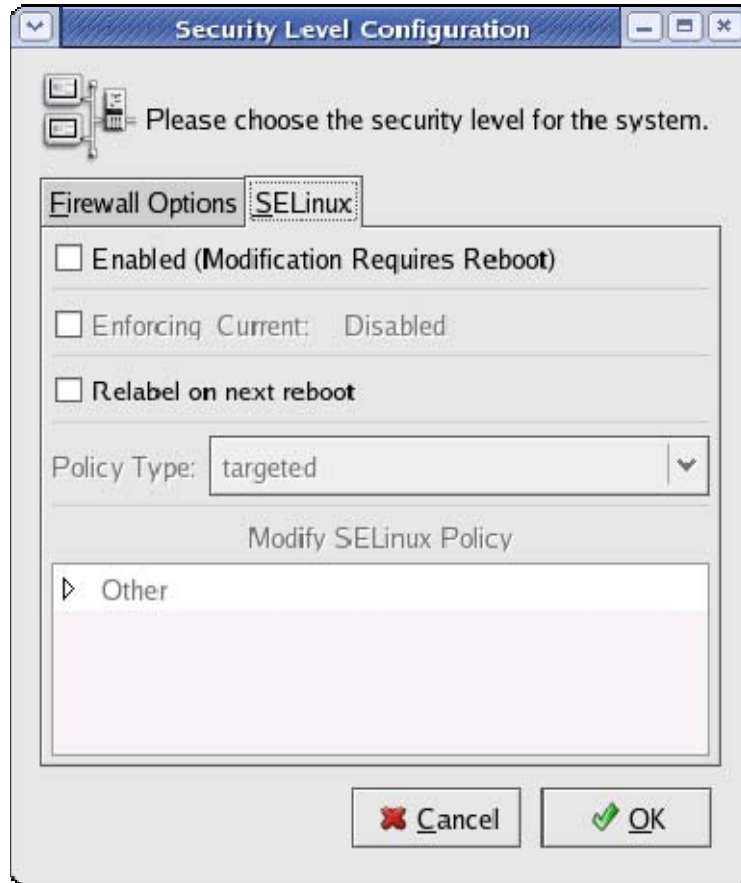


Figure 22. SELinux disabled on Linux.

3.5.2.2. OCFS2 Packages

OCFS2 comes in the following RPM packages:

| | |
|--|---------------|
| ocfs2-2.6.9-22.ELsmp-1.2.3-1.i686.rpm | Necessary |
| ocfs2console-1.2.2-1.i386.rpm | Necessary |
| ocfs2-tools-1.2.2-1.i386.rpm | Necessary |
| ocfs2-tools-debuginfo-1.2.2-1.i386.rpm | Not necessary |
| ocfs2-tools-devel-1.2.2-1.i386.rpm | Not necessary |

To install, execute the following command:

```
#rpm -Uvh ocfs2*.rpm
```

To verify that all the ocfs packages have been installed, execute the following command:

```
#rpm -qa | grep ocfs
```

3.5.2.3. OCFS2 Configuration

3.5.2.3.1. Virtual Disk Addition

If a virtual disk has not been added at this point, refer to section 3.3 for the corresponding procedure.

3.5.2.3.2. OCFS2 Configuration

After the virtual disk has been added, follow the procedure below to format it as OCFS2.

- Power on the VM.
- Log in as root.
- Start X windows.
 - # startx
- Configure the O2CB driver. Use **61** as the heartbeat dead threshold. The heartbeat dead threshold cannot be set to too low if the disk response time is not optimal. The nodes may crash in such a case.
- As root, execute:

```
# /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

# /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 without
typing an answer will keep that current value. Ctrl-C will abort.

Load O2CB driver on boot (y/n) [y]: y
Cluster to start on boot (Enter "none" to clear) [ocfs2]:
Specify heartbeat dead threshold (>=7) [7]: 61
Writing O2CB configuration: OK
Loading module "configfs": OK
Mounting configfs filesystem at /config: OK
Loading module "ocfs2_nodemanager": OK
Loading module "ocfs2_dlm": OK
Loading module "ocfs2_dlmfs": OK
Mounting ocfs2_dlmfs filesystem at /dlm: OK
Starting O2CB cluster ocfs2: OK
```

- Verify that the o2cb is online:

```
# /etc/init.d/o2cb status
Module "configs": Loaded
Filesystem "configs": Mounted
Module "ocfs2_nodemanager": Loaded
Module "ocfs2_dlm": Loaded
Module "ocfs2_dlmfs": Loaded
Filesystem "ocfs2_dlmfs": Mounted
Checking O2CB cluster ocfs2: Online
Checking O2CB heartbeat: Not active
```

- # fdisk -l

```
Disk /dev/sdb: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

The device /dev/sdb is the disk to format as OCFS2.

- In one of the terminal windows, run the following:
 - # ocfs2console
- In ocfs2console, go to Tasks->Format

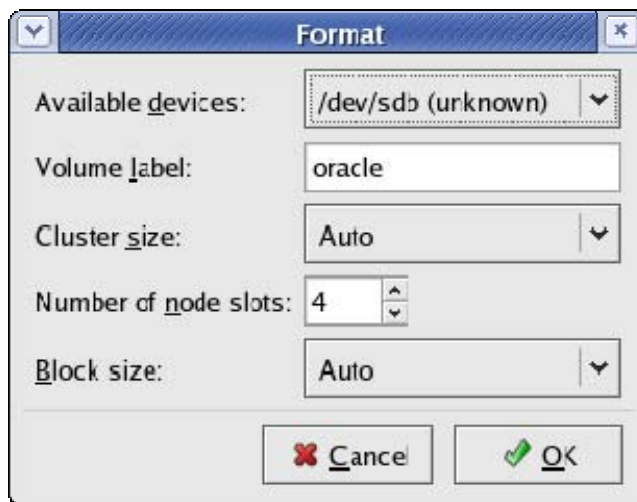


Figure 23. Format /dev/sdb as OCFS2

- Click OK. See that /dev/sdb shows up one of the devices in the console.

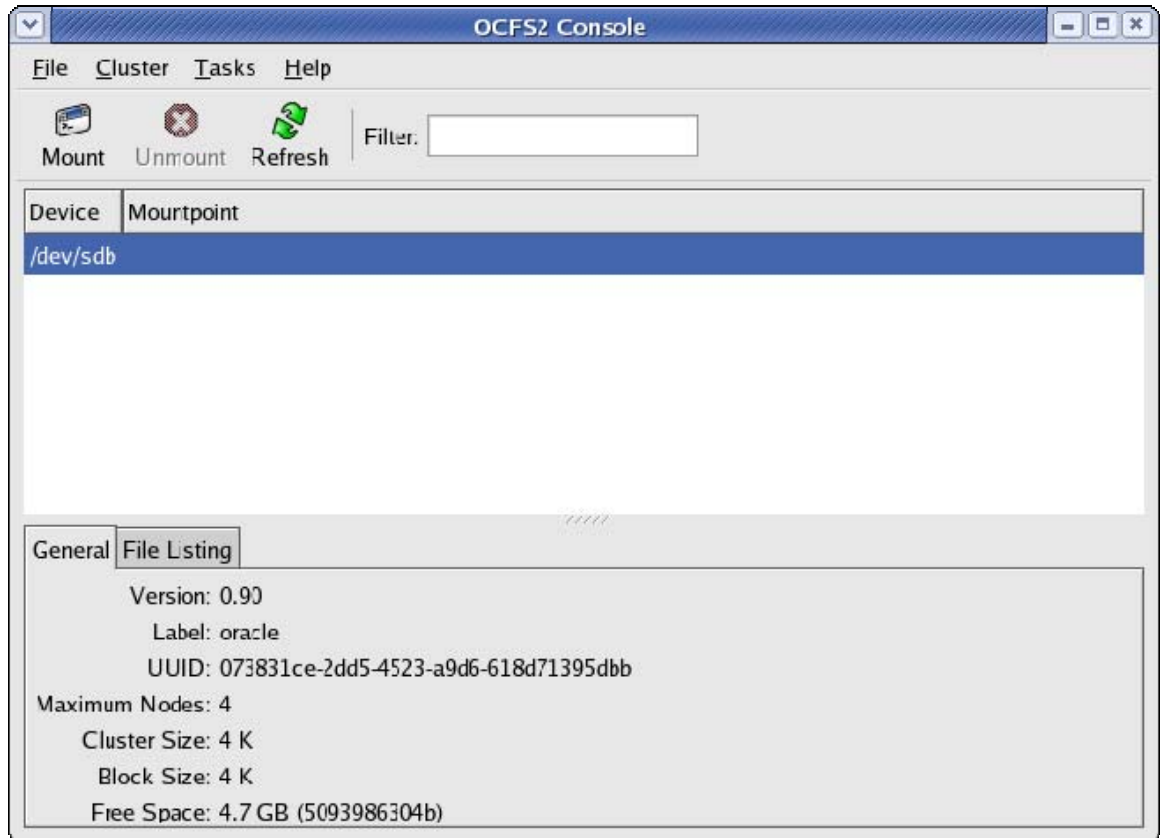


Figure 24. /dev/sdb formatted as OCFS2

- Configure the cluster for OCFS2. To configure the cluster, at least one node should be in it.
 - Go to Cluster->Configure Nodes.
 - Add rac_1 in the cluster.

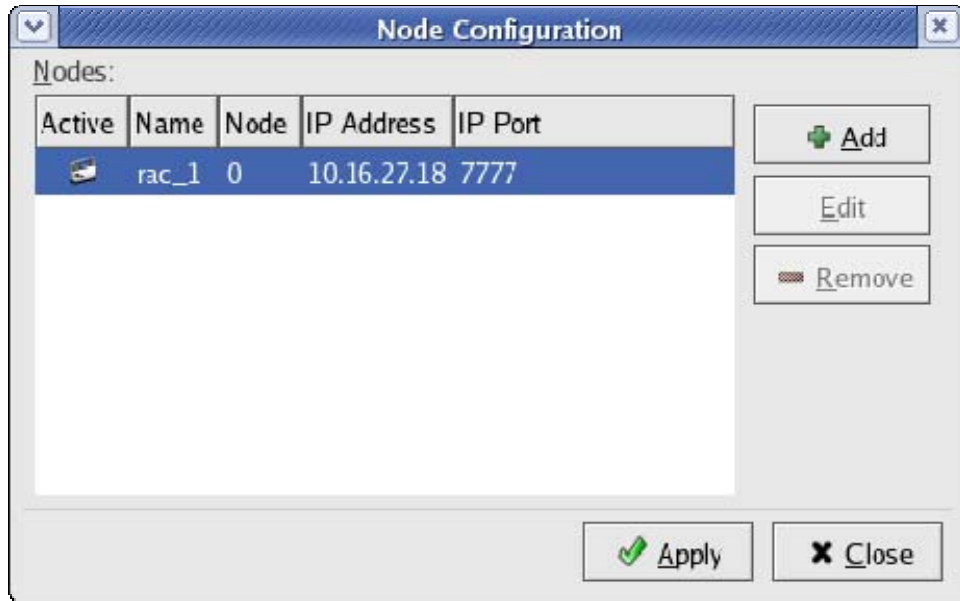


Figure 25. Node Configuration for the OCFS2 cluster.

- Mount the file system. To mount the file system, execute the command below on both nodes. Make sure that /ocfs exists before performing the mount.

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

- Add the following line in /etc/fstab in order to mount the file system on boot.

```
/dev/sdb /ocfs ocfs2 _netdev,datavolume,nointr 0 0
```

3.5.3. Oracle RAC Installation Prerequisites

Refer to section 2.5 for the pre-requisites, except for the /etc/hosts file:

| | | | |
|--|--------------|-----------------------|------------|
| Modify the /etc/hosts file. | 127.0.0.1 | localhost | |
| | 10.16.27.18 | rac_1.vmware.com | rac_1 |
| | 10.16.27.19 | rac_1-vip.vmware.com | rac_1-vip |
| | 192.168.2.31 | rac_1-priv.vmware.com | rac_1-priv |

3.5.4. Installation of Oracle Clusterware

This section shows how to install Oracle Clusterware on an OCFS volume.

- Download the installation zip file from the Oracle website. For the purpose of this deployment, 10201_clusterware_linux32.zip is used.
- As root, execute the following commands to set up the /ocfs directory for Oracle Clusterware installation:

```
# mkdir /ocfs/clusterware
# chown -R oracle:dba /ocfs
# xhost +
```

- Establish User Equivalence:
 - Note that even though only one node is configured in this cluster, the OUI still uses ssh / rsh to copy the software. Therefore, before proceeding to install Clusterware, ensure that user equivalence is established.
 - As the user "oracle", do the following:

```
# mkdir ~/.ssh
# chmod 700 ~/.ssh
# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/export/home/oracle/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /export/home/oracle/.ssh/id_rsa.
Your public key has been saved in /export/home/oracle/.ssh/id_rsa.pub.
The key fingerprint is:
87:54:4f:92:ba:ed:7b:51:5d:1d:59:5b:f9:44:da:b6
oracle@rac_1.vmware.com

# ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/export/home/oracle/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /export/home/oracle/.ssh/id_dsa.
Your public key has been saved in /export/home/oracle/.ssh/id_dsa.pub.
The key fingerprint is:
31:76:96:e6:fc:b7:25:04:fd:70:42:04:1f:fc:9a:26
oracle@rac_1.vmware.com

# cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
# cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys

# ssh rac_1.vmware.com date
# ssh rac_1-priv.vmware.com date
# ssh rac_1 date
# ssh rac_1-priv date
```

- As the user "oracle", do the following:
 - Unzip 10201_clusterware_linux32.zip in a directory /export/home/oracle/downloads.
 - After the zip file has been unzipped, there will be a directory clusterware in /export/home/oracle/downloads.
 - # cd clusterware
 - # ./runInstaller

- In the Oracle Universal Installer, enter the following configuration:

| | |
|---|---|
| Welcome: | Click on Next . |
| Inventory directory and credentials: | |
| The full path of the inventory directory: | /u01/app/oracle/orainventory. |
| Operating System group name: | oinstall |
| Home Details: | |
| Name: | OraCrs10g_home |
| Path | /u01/app/oracle/product/10.2.0/crs_1 |
| Product-Specific Prerequisite Checks | All checks should pass. NOTE: In the case that the physical memory requirement does not pass, the customer can choose to ignore it. In best practice for Oracle RAC deployment, it is suggested that sufficient virtual memory be allocated to the virtual machine. |
| Cluster Configuration | |
| Cluster Name: | crs |
| Cluster Nodes: | Make sure that rac_1 is in the cluster. <ul style="list-style-type: none"> • Public Node Name: rac_1.vmware.com • Private Node Name: rac_1-priv.vmware.com • Virtual Host Name: rac_1-vip.vmware.com |
| Network Interface Usage | |
| | Make sure that eth0 is used as the public interface. This configuration should be identical across all the nodes in the cluster. <ul style="list-style-type: none"> • Interface Name: eth0 • Subnet: 10.16.0.0 • Interface Type: Public • Interface Name: eth1 • Subnet: 192.168.2.0 • Interface Type: Private |
| Oracle Cluster Registry (OCR) Location | |
| OCR Configuration | In this scenario, the OCR is not mirrored for simplicity's sake. However, it is advisable to have the OCR multiplexed in a |

| | |
|-----------------------------|---|
| | production environment. Select External Redundancy . |
| OCR Location | /ocfs/clusterware/ocr |
| Voting Disk Location | |
| Voting Disk Configuration | In this scenario, the voting disk is not mirrored for simplicity's sake. It is advisable to have the voting disk mirrored in a production environment. Select External Redundancy . |
| Voting Disk Location: | /ocfs/clusterware/votingdisk |
| Summary: | Click on Install . |

- Execute configuration scripts:

Execute the scripts below as the root user sequentially, one at a time. Do not proceed to the next script until the current script completes.

- Execute `/u01/app/oracle/orainventory/orainstRoot.sh` on `rac_1`.
- Execute `/u01/app/oracle/product/10.2.0/crs_1/root.sh` on `rac_1`.

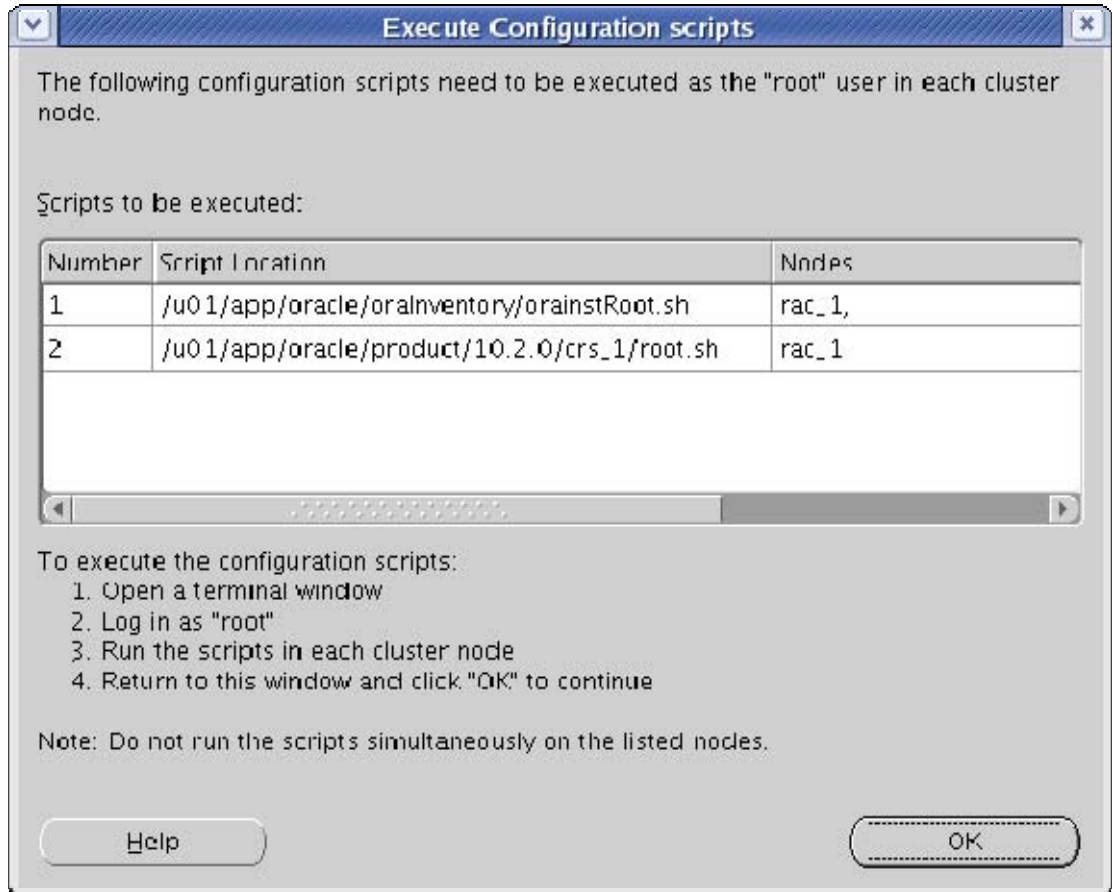


Figure 26. Final Configuration Scripts to run for Oracle Clusterware Installation.

The following shows the output of the two shell scripts. Note the following points:

1. The warnings about the directories not owned by root can be ignored.
2. The Oracle configuration utility vipca decides that IP addresses in the subnets 192.168.0.0 and 10.16.0.0 are not public IP addresses. As 10.16.0.0 subnet is used for this deployment scenario, vipca does not like to run in silent mode. The utility vipca needs to be run in order to configure the vip.

```
[root@rac_1 ~]# /u01/app/oracle/oraInventory/orainstRoot.sh
Changing permissions of /u01/app/oracle/oraInventory to 770.
Changing groupname of /u01/app/oracle/oraInventory to oinstall.
The execution of the script is complete
[root@rac_1 ~]# /u01/app/oracle/product/10.2.0/crs_1/root.sh
WARNING: directory '/u01/app/oracle/product/10.2.0' is not owned
by root
WARNING: directory '/u01/app/oracle/product' is not owned by root
WARNING: directory '/u01/app/oracle' is not owned by root
WARNING: directory '/u01/app' is not owned by root
WARNING: directory '/u01' is not owned by root
Checking to see if Oracle CRS stack is already configured
/etc/oracle does not exist. Creating it now.

Setting the permissions on OCR backup directory
Setting up NS directories
Oracle Cluster Registry configuration upgraded successfully
WARNING: directory '/u01/app/oracle/product/10.2.0' is not owned
by root
WARNING: directory '/u01/app/oracle/product' is not owned by root
WARNING: directory '/u01/app/oracle' is not owned by root
WARNING: directory '/u01/app' is not owned by root
WARNING: directory '/u01' is not owned by root
assigning default hostname rac_1 for node 1.
Successfully accumulated necessary OCR keys.
Using ports: CSS=49895 CRS=49896 EVMC=49898 and EVMR=49897.
node <nodenumber>: <nodename> <private interconnect name>
<hostname>
node 1: rac_1 rac_1-priv rac_1
Creating OCR keys for user 'root', privgrp 'root'..
Operation successful.
Now formatting voting device: /ocfs/clusterware/votingdisk
Format of 1 voting devices complete.
Startup will be queued to init within 90 seconds.
ksh: /etc/profile[58]: ulimit: exceeds allowable limit
Adding daemons to inittab
Expecting the CRS daemons to be up within 600 seconds.
CSS is active on these nodes.
    rac_1
CSS is active on all nodes.
Waiting for the Oracle CRSD and EVMD to start
Waiting for the Oracle CRSD and EVMD to start
Oracle CRS stack installed and running under init(1M)
Running vipca(silent) for configuring nodeapps
The given interface(s), "eth0" is not public. Public interfaces
should be used to configure virtual IPs.
```

- As root, run /u01/app/oracle/product/10.2.0/crs_1/bin/vipca.

In the VIPCA wizard, enter the following:

| | |
|--|---------------|
| Welcome | Click Next. |
| Network Interfaces | Select eth0. |
| Virtual IPs for Clustered Nodes | |
| Node Name | rac_1 |
| IP Alias Name | rac_1-vip |
| IP address | 10.16.27.19 |
| Subnet Mask | 255.255.255.0 |
| Summary | Click Finish. |

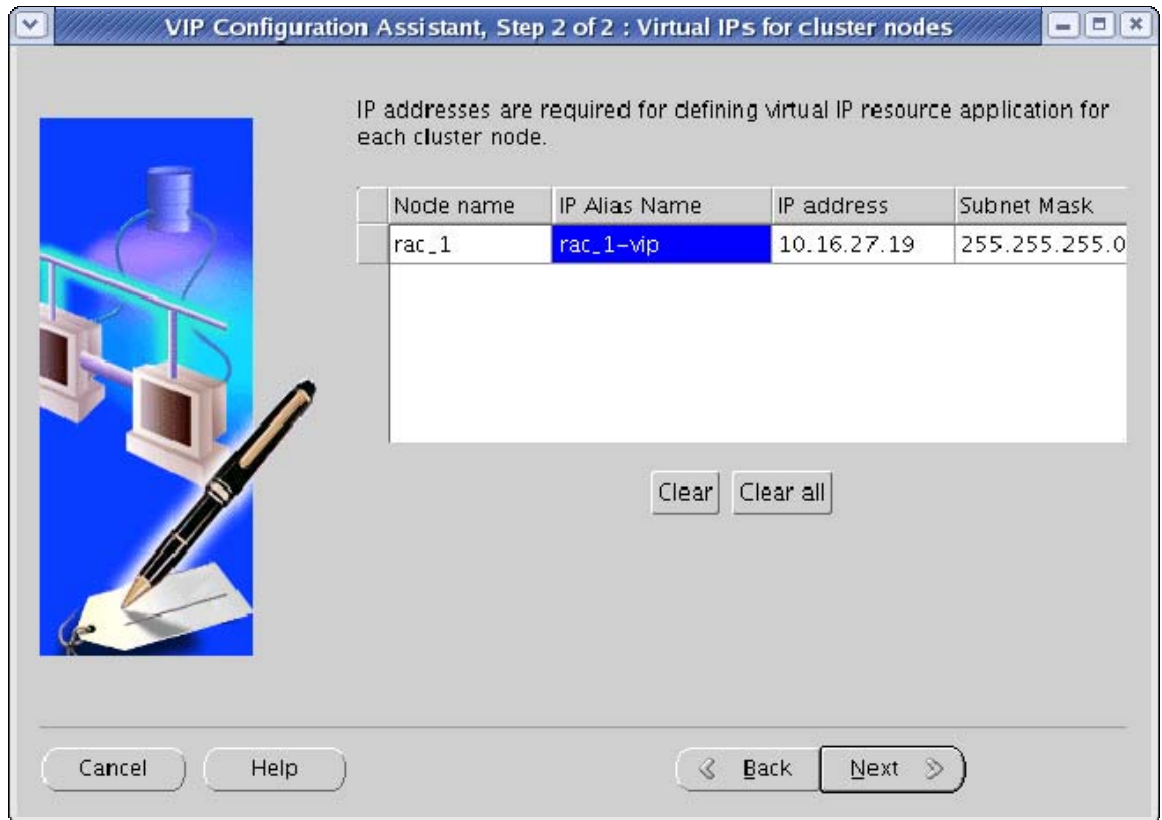


Figure 27. Configuration of VIP in vipca

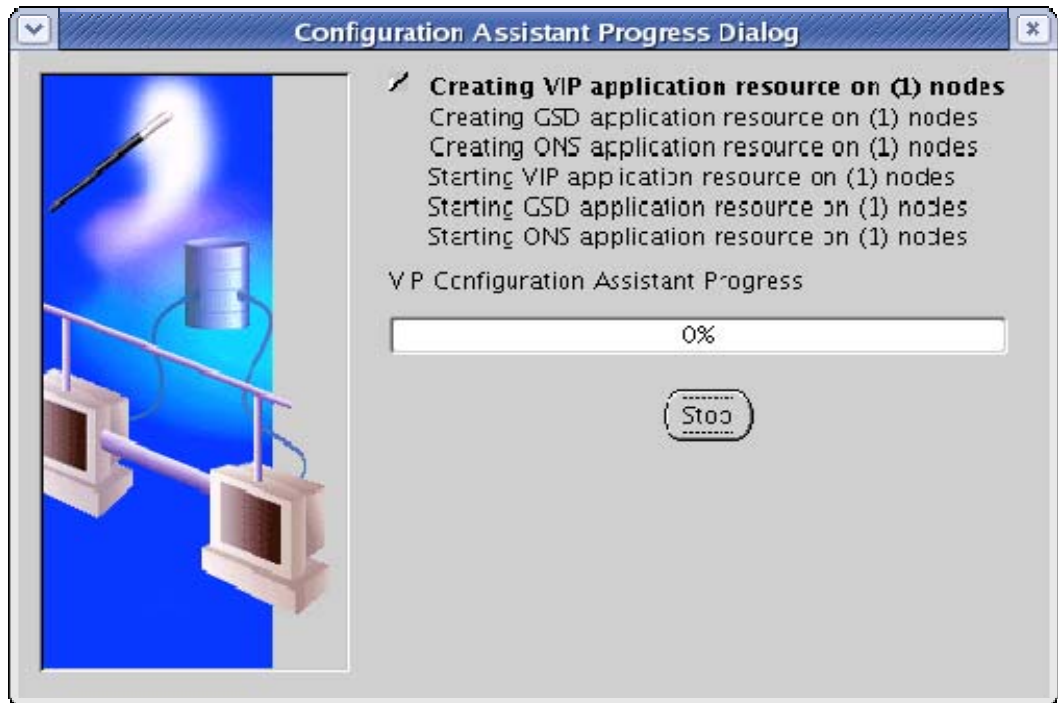


Figure 28. VIPCA Configuration Progress

- At this point, the following services should be running on rac_1:
 - ONS
 - GSD
 - VIP
- Check the status of each of the services by running crs_stat.

```
[root@rac_1 ~]#
/u01/app/oracle/product/10.2.0/crs_1/bin/crs_stat -t
Name           Type           Target         State         Host
-----
ora.rac_1.gsd  application    ONLINE        ONLINE        rac_1
ora.rac_1.ons  application    ONLINE        ONLINE        rac_1
ora.rac_1.vip  application    ONLINE        ONLINE        rac_1
```

3.6. *Installation of Oracle Database 10g Release 2*

- Download the installation zip file from the Oracle website. For the purpose of this deployment, 10201_database_linux32.zip is used.
- As the user "oracle", do the following:
 - Unzip 10201_database_linux32.zip in a directory /export/home/oracle/downloads.
 - After the zip file has been unzipped, there will be a directory database in /export/home/oracle/downloads.
 - # cd database
 - # runInstaller
- In the Oracle Universal Installer, enter the following configuration:

| | |
|--|--|
| Welcome: | Click on Next . |
| Installation Type: | Select Enterprise Edition . |
| Home Details: | <ul style="list-style-type: none"> • Name: OraDb10g_home1 • Path: /u01/app/oracle/product/10.2.0/db_1 |
| Hardware Cluster Installation Mode: | <ul style="list-style-type: none"> • Select Cluster Installation. • The node rac_1 should be selected by default. |
| Product-Specific Prerequisite Checks: | <p>All checks should pass.</p> <p>NOTE: If the physical memory requirement does not pass, the customer can choose to ignore it. In best practice for Oracle RAC deployment, it is suggested that sufficient virtual memory be allocated to the virtual machine.</p> |
| Configuration Option: | Install Database Software Only. |
| Summary | Click Install. |

- Execute configuration scripts: execute the scripts below as the root user.
 - Execute /u01/app/oracle/product/10.2.0/db_1/root.sh on rac_1.

The following is a listing of the output of root.sh. Take the default configuration for the local bin directory.

```
[root@rac_1 ~]# /u01/app/oracle/product/10.2.0/db_1/root.sh
Running Oracle10 root.sh script...

The following environment variables are set as:
  ORACLE_OWNER= oracle
  ORACLE_HOME= /u01/app/oracle/product/10.2.0/db_1

Enter the full pathname of the local bin directory:
[/usr/local/bin]:
  Copying dbhome to /usr/local/bin ...
  Copying oraenv to /usr/local/bin ...
  Copying coraenv to /usr/local/bin ...

Creating /etc/oratab file...
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.
```

3.7. Database Template Preparation

With the RAC database software installed, customers can choose to install a database template as illustrated in the single-instance database installation in section 2.7.

3.8. Gold Image Virtual Machine Template Preparation

With the virtual machine successfully installed and configured, it is possible to convert it to a template for further deployment.

NOTE: If the intention is to deploy other nodes in an existing RAC cluster, remove the shared disk that has been used for OCFS before creating this template. If the shared disk is not removed, it will be included in this procedure and take up space as virtual machines are deployed from this template.

NOTE: The virtual disk should not be removed from the physical disk because it will need to be re-added during deployment.

NOTE: Include the shared disk if the intention is to use this template to deploy a RAC cluster from scratch in another datacenter.

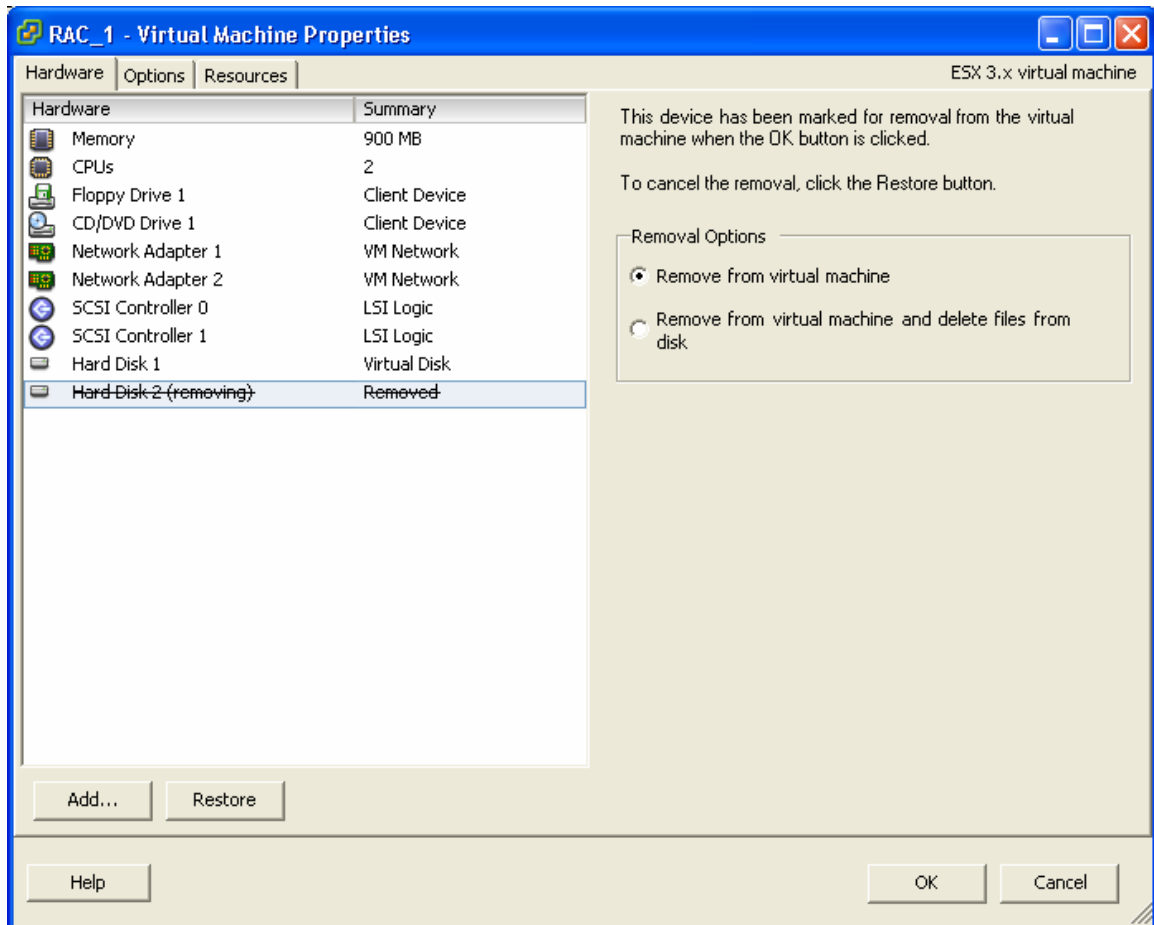


Figure 29. Removing a Hard Disk from a Virtual Machine.

To create the template, perform the following steps in VMware VirtualCenter, :

- Select the virtual machine in the inventory.
- Clone to Template.
- In the wizard, follow the parameters specified below:

| | |
|--------------------------|---|
| Name and Folder | Template Name |
| | RAC_GOLD_IMAGE |
| Host / Cluster | Select the host desired for the template to reside in. 10.16.27.16 is used in this scenario. |
| Datastore | Pick a datastore that is preferably visible to both ESX Server hosts. |
| Disk Format | Select Compact. This saves space and will help when transferring to a different host. |
| Ready to Complete | Click Finish. |

3.9. RAC Gold Image Deployment

The Gold Image for RAC can be used for the following purposes:

- Deploy the first RAC node by providing the template to another datacenter.
- Deploy other RAC nodes by performing minimal customization steps.

This section shows how to achieve the above goals.

3.9.1. Virtual Machine Cloning

With the template, one can clone another RAC_2 node in another datacenter. Follow the procedure below:

In VMware VirtualCenter, perform the following:

- In the Virtual Machines and Templates view, select RAC_GOLD_IMAGE.
- Select "Deploy Virtual Machine from this Template."
- In the Deploy Template Wizard, enter the following:

| | |
|----------------------------|---|
| Name and Folder | Virtual Machine Name: RAC_2 |
| Host / Cluster | Choose the host to deploy RAC_2: For this scenario, 10.16.27.16 is used. |
| Datastore | Preferable, pick a datastore that is visible to both ESX Server hosts. |
| Guest Customization | Pick Do not customize. |
| Ready to Complete | Click Finish. |

- Add the virtual disk that has been formatted for OCFS2 as an independent and persistent disk. Make sure that bus sharing has been configured as physical.

3.9.2. Second RAC Node Customization

After a virtual machine has been deployed from the RAC virtual machine template, it needs to be customized to function as another RAC node in the RAC cluster. This section shows the customization steps.

3.9.2.1. Linux Network Settings

When the second RAC node is powered on, customers are prompted to configure the network cards. Follow the instructions to configure the new IPs, subnet masks, etc.

After the Linux operating system is launched, launch X-windows and use system tool **system-config-network** to make sure that the network cards are configured properly.

Check the following:

1. eth0 is configured to be the public interface and eth1 the private one.
2. hostname is configured to be rac_2.
3. the DNS server and default gateway are both configured properly.

Execute the following command as root for the configurations to take effect:

```
# /etc/init.d/network restart
```

Check /etc/hosts

On both rac_1 and rac_2, make sure that the /etc/hosts file reads as follows:

| | | |
|--------------|-----------------------|------------|
| 127.0.0.1 | localhost | |
| 10.16.27.18 | rac_1.vmware.com | rac_1 |
| 10.16.27.19 | rac_1-vip.vmware.com | rac_1-vip |
| 192.168.2.31 | rac_1-priv.vmware.com | rac_1-priv |
| 10.16.27.20 | rac_2.vmware.com | rac_2 |
| 10.16.27.21 | rac_2-vip.vmware.com | rac_2-vip |
| 192.168.2.32 | rac_2-priv.vmware.com | rac_2-priv |

Establish user equivalence

As the user "oracle", do the following:

On rac_2, execute the following:

```
# rm -rf ~/.ssh
# mkdir ~/.ssh
# chmod 700 ~/.ssh
# ssh-keygen -t rsa
# ssh-keygen -t dsa
```

On rac_1, execute the following:

```
# ssh rac2 cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
# ssh rac2 cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
```

On both nodes, make sure that user equivalence has been set up properly.

```
ssh rac_1 date
ssh rac_2 date
ssh rac_1-priv date
ssh rac_2-priv date
ssh rac_1.vmware.com date
ssh rac_2.vmware.com date
ssh rac_1-priv.vmware.com date
ssh rac_2-priv.vmware.com date
```

3.9.2.2. OCFS2 Settings

There are two ways to make rac_2 join the cluster:

- OCFS2 console
- Edit OCFS2 cluster configuration file.

3.9.2.2.1. OCFS2Console

The utility program ocfs2console can be used to configure rac_2 to join the ocfs cluster.

```
# ocfs2console
```

1. In OCFS2 Console, select **Cluster, Configure Nodes**.
2. "The cluster stack has been started": Click on **Close**.
3. In Node Configuration: Click on **Add**.
4. Add Node: Add the following nodes and then click on **Apply**.

- Name: rac_2
- IP Address: 10.16.27.20
- IP Port: 7777

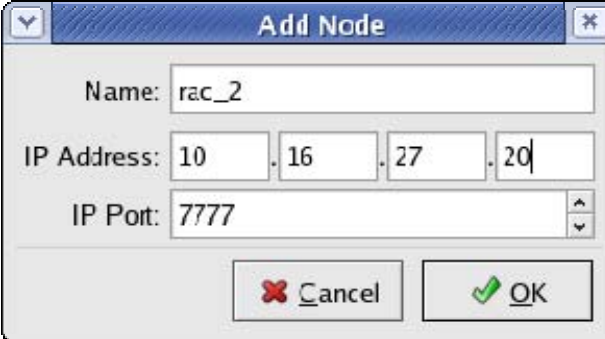


Figure 30. Add rac_2 in the OCFS2 cluster.

5. Propagate the configuration file to rac_2. The steps above can be re-run on rac_2 to generate the configuration file or select **Cluster, Propagate Configuration** in the OCFS2 Console on rac_1 to propagate the configuration file to rac_2.

3.9.2.2.2. *Edit Cluster Configuration File*

Another alternative is to edit the cluster configuration file `/etc/ocfs2/cluster.conf` to read as follows on both nodes:

```
node:
  ip_port = 7777
  ip_address = 10.16.27.18
  number = 0
  name = rac_1
  cluster = ocfs2

node:
  ip_port = 7777
  ip_address = 10.16.27.20
  number = 1
  name = rac_2
  cluster = ocfs2

cluster:
  node_count = 2
  name = ocfs2
```

After the configuration file has been edited, run the following command as root:

```
# /etc/init.d/o2cb restart
```

3.9.2.3. RAC Customization

Now it is time to make rac_2 join the RAC cluster. In order to do so, a number of Oracle scripts need to be modified. The steps are described below:

- On rac_1, copy
u01/app/oracle/product/10.2.0/crs_1/install/rootaddnode.sbs to
u01/app/oracle/product/10.2.0/crs_1/install/rootaddnode.sh
- On rac_1, edit
/u01/app/oracle/product/10.2.0/crs_1/install/rootaddnode.sh. See the Appendix for a full listing of the shell script. An example for adding rac_2 is shown below (The changes are highlighted):

```
#!/bin/sh
# Node addition script. Must be run from an existing node of the
# cluster *prior* to starting any of the Cluster Ready Services
# daemons on the new nodes.

ORA_CRS_HOME=/u01/app/oracle/product/10.2.0/crs_1
export ORA_CRS_HOME
CH=$ORA_CRS_HOME
ORACLE_HOME=$CH
export ORACLE_HOME
ORACLE_OWNER=oracle
CRS_NEW_HOST_NAME_LIST=rac_2,2
CRS_NEW_NODE_NAME_LIST=rac_2,2
CRS_NEW_PRIVATE_NAME_LIST=rac_2-priv,2
CRS_NEW_NODEVIPS=rac_2-vip,2
( ... )
```

- On rac_2, edit u01/app/oracle/product/10.2.0/crs_1/install/rootconfig.sh. See the Appendix for the full listing of the shell script. The changes are highlighted in the listing below:

```
#!/bin/sh
#
# rootconfig.sh for Oracle CRS homes
#
# This is run once per node during the Oracle CRS install.
# This script does the following:
# 1) Stop if any GSDs are running from 9.x oracle homes
# 2) Initialize new OCR device or upgrade the existing OCR device
# 3) Setup OCR for running CRS stack
# 4) Copy the CRS init script to init.d for init process to start
# 5) Start the CRS stack
# 6) Configure NodeApps if CRS is up and running on all nodes
#
# NOTE: Use sample paramfile in
$ORA_CRS_HOME/srvm/admin/paramfile.sample for
# setting CRS parameters
# The following commands need to be run before Oracle Cluster Registry is
# populated.
# This is run during CRS installation and not during RAC

SILENT=false
ORA_CRS_HOME=/u01/app/oracle/product/10.2.0/crs_1
CRS_ORACLE_OWNER=oracle
CRS_DBA_GROUP=oinstall
CRS_VNDR_CLUSTER=false
CRS_OCR_LOCATIONS=/ocfs/clusterware/ocr
CRS_CLUSTER_NAME=crs
CRS_HOST_NAME_LIST=rac_1.vmware.com,1,rac_2.vmware.com,2
CRS_NODE_NAME_LIST=rac_1.vmware.com,1,rac_2.vmware.com,2
CRS_PRIVATE_NAME_LIST=rac_1-priv.vmware.com,1,rac_2-
priv.vmware.com,2
CRS_LANGUAGE_ID='AMERICAN_AMERICA.WE8ISO8859P1'
CRS_VOTING_DISKS=/ocfs/clusterware/votingdisk
CRS_NODELIST=rac_1,rac_2
CRS_NODEVIPS='rac_1/rac_1-vip/255.255.0.0/eth0','rac_2/rac_2-
vip/255.255.0.0/eth0'

( ... )
```

- On rac_2, run as root the command below:

```
# u01/app/oracle/product/10.2.0/crs_1/root.sh
```

- On rac_2, invoke VIPCA as the root user:

```
# /u01/app/oracle/product/10.2.0/crs_1/bin/vipca
```

- In VIPCA wizard, enter the following:

| | |
|---------------------------------------|--|
| Welcome: | Click on Next. |
| Network Interfaces: | Select eth0. |
| Virtual IPs for cluster nodes: | Node name: rac_1 IP Alias Name: rac_1-vip IP address: 10.16.27.19 Subnet Mask: 255.255.255.0 Node name: rac_2 IP Alias Name: rac_2-vip IP address: 10.16.27.21 Subnet Mask: 255.255.255.0 |
| Summary: | Click on Finish. |

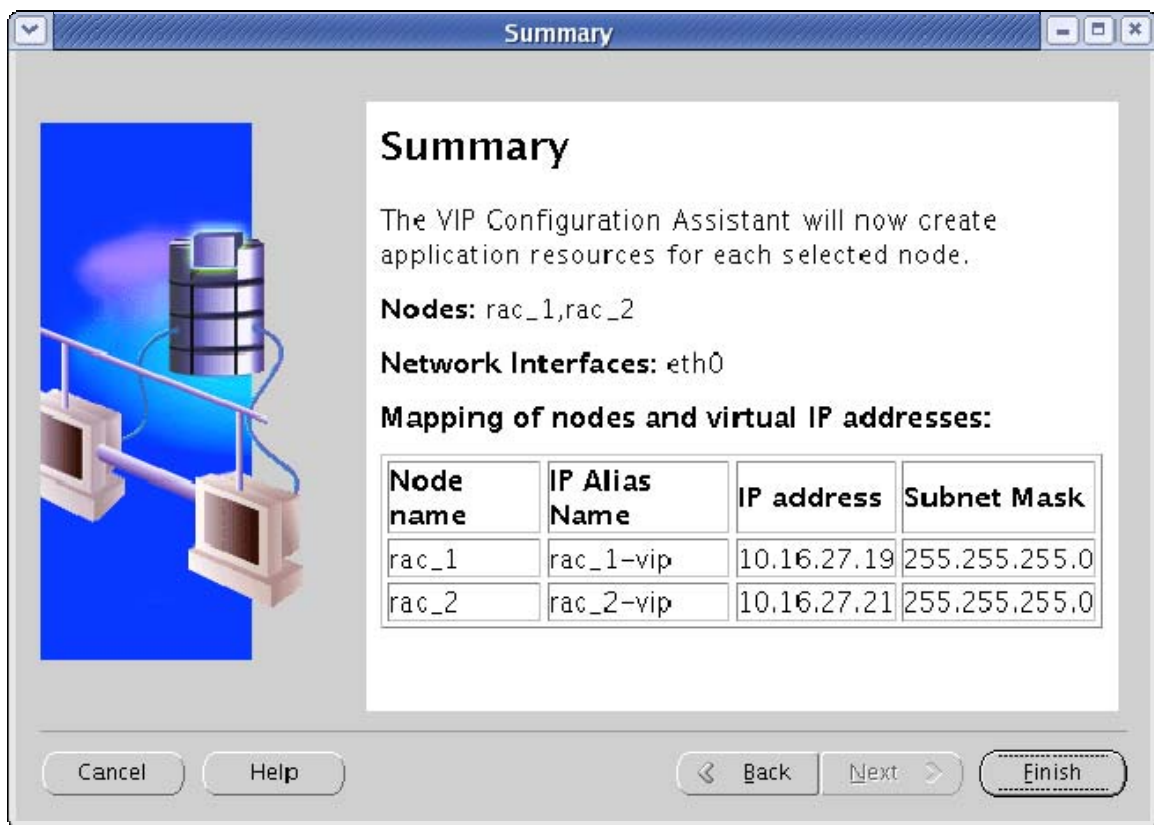


Figure 31. Summary page of the RAC cluster configuration.

- After rac_2 has been successfully deployed, check the status of the node applications by issuing the following command:

```
[root@rac_2 bin]# ./crs_stat -t
Name                Type                Target              State              Host
-----
ora.rac_1.gsd       application         ONLINE             ONLINE             rac_1
ora.rac_1.ons       application         ONLINE             ONLINE             rac_1
ora.rac_1.vip       application         ONLINE             ONLINE             rac_1
ora.rac_2.gsd       application         ONLINE             ONLINE             rac_2
ora.rac_2.ons       application         ONLINE             ONLINE             rac_2
ora.rac_2.vip       application         ONLINE             ONLINE             rac_2
```

- On rac_1, make sure that the database inventory is aware of the node list change. Execute the following command in \$ORACLE_HOME/oui/bin as the user "oracle":

```
./runInstaller -UpdateNodeList ORACLE_HOME=$ORACLE_HOME
ORACLE_HOME_NAME="OraDb10g_home1" CLUSTER_NODES=rac_1,rac_2
```

- On rac_1, make sure that the CRS inventory is aware of the node list change. Execute the following command in \$ORA_CRS_HOME/oui/bin as the user "oracle":

```
./runInstaller -UpdateNodeList ORACLE_HOME=$ORA_CRS_HOME
ORACLE_HOME_NAME="OraCrs10g_home" CLUSTER_NODES=rac_1,rac_2
```

- On rac_1, use an Oracle utility program "cluvfy" to verify the cluster membership of rac_1 and rac_2. As the user "oracle", run:

```
/u01/app/oracle/product/10.2.0/crs_1/bin/cluvfy stage -post crsinst -n
rac_1,rac_2
```

- Now a 2-node RAC cluster has been configured. The next step is to use the **dbca** utility program to create a database on the two nodes.

3.10. VMware HA for RAC

To take advantage of VMware HA for Oracle RAC deployment, customers can follow the configuration steps listed in section 2.10. In the case of ESX Server host failures, Oracle RAC provides high availability when the RAC nodes are distributed across multiple ESX Server hosts. Using VMware HA, customers can automatically restart the failed RAC node (on the failed ESX Server host) on another online ESX Server host. This will help restore the original RAC configuration within minutes while enhancing availability and performance. Virtual machines on a failed ESX Server host can be restarted on an available ESX Server host in the HA cluster automatically.

3.11. VMware VMotion for RAC

VMware VMotion supports live migration and cold migration of virtual machines. During system maintenance performed on an ESX Server host running Oracle virtual machines, live migration moves the Oracle virtual machines to another ESX Server host with zero or minimal disruption to end users. Unlike the single-instance Oracle environment, RAC deployment cannot benefit from the live migration feature because of the shared iscsi controllers for the shared virtual disks. In this case, customers can rely on cold migration instead. Even though it entails a virtual machine shutdown, the cold migration feature restarts an Oracle RAC virtual machine node in a time-efficient manner so that the end users are minimally disrupted.

In order to perform a cold migration on a virtual machine, follow the procedure below:

- Shut down the RAC databases in the virtual machine that is going to be migrated.
- Power off the virtual machine.
- VM > migrate
- In the Migrate Virtual Machine Wizard, enter the following:

| | |
|-----------------------------|--|
| Select Destination | Select the redundant ESX Server host. In this scenario, it is 10.16.27.17. |
| Select Resource Pool | Leave as default. |
| Select Datastore | Select Keep virtual machine configuration files and virtual disks in their current locations. |
| Ready to Complete | Click Finish. |

4. Summary

The procedures outlined above for the deployment of Oracle Database software on VMware Infrastructure 3 platform show that **Install-Once-Run-Everywhere (IORE) provides rapid provisioning capabilities**. The Gold Image is very useful in terms of software image consistency and rapid Oracle Database deployments. It is pre-configured and thus provides standardized corporate IT deployments and efficient change management.

Besides IORE, VMware Infrastructure 3 also complements the Oracle software features by providing enhanced levels of high availability and server consolidation. Users of single-instance Oracle Database 10g can leverage **VMware HA** and **VMotion™** to ensure cost effective availability solutions, which serve as a viable alternative to expensive 3rd party failover and clustering solutions. Even RAC users can benefit from the VMware Infrastructure 3 HA feature and add more availability options for the Oracle deployment. This feature powerfully complements the scale out, load balancing, and availability features of Oracle RAC.

VMware Infrastructure 3 and Oracle Database software capabilities complement each other in streamlining an Oracle deployment. IT users can also leverage other features of VMware Infrastructure such as VMware® Distributed Resource Scheduling (DRS) and VMware® Consolidated Backup to make their deployment of Oracle Database RAC even more manageable, efficient and cost effective.

For more details about VMware and Oracle products, here are some useful links:

VMware official website: www.vmware.com

Oracle official website: www.oracle.com

VMware Infrastructure 3 product website:

http://www.vmware.com/products/data_center.html

Oracle Database Enterprise Edition Website:

http://www.oracle.com/database/enterprise_edition.html

VMware Infrastructure 3 download website:

<http://www.vmware.com/download/vi/eval.html>

Oracle Database/RAC 10g download website:

<http://www.oracle.com/technology/software/products/database/oracle10g/index.html>

VMware support website

<http://www.vmware.com/vmtn/>

Oracle support website

<http://www.oracle.com/technology/about/index.html>

Oracle Enterprise Linux download website:

<http://edelivery.oracle.com/linux>

The following link shows other VMware products that can complement the provisioning of Oracle software:

<http://www.vmware.com/products/>

Appendix

This appendix includes full listings of two shell scripts: `rootaddnode.sh` and `rootconfig.sh`. These shell scripts are customized for RAC node deployments in the step to configure the RAC cluster and include a new node (i.e. `rac_2`). More details are given in section 3.9.2.

rootaddnode.sh

Below is a sample `rootaddnode.sh` script (`u01/app/oracle/product/10.2.0/crs_1/install/rootaddnode.sh`) for RAC node customization:

```
#!/bin/sh
# Node addition script. Must be run from an existing node of the
# cluster
# *prior* to starting any of the Cluster Ready Services daemons on the
# new nodes.

ORA_CRS_HOME=/u01/app/oracle/product/10.2.0/crs_1
export ORA_CRS_HOME
CH=$ORA_CRS_HOME
ORACLE_HOME=$CH
export ORACLE_HOME
ORACLE_OWNER=oracle
CRS_NEW_HOST_NAME_LIST=rac_2,2
CRS_NEW_NODE_NAME_LIST=rac_2,2
CRS_NEW_PRIVATE_NAME_LIST=rac_2-priv,2
CRS_NEW_NODEVIPS=rac_2-vip,2

UNAME=/bin/uname
PLATFORM=`$UNAME`

if [ -z "$AWK" ]; then AWK=/bin/awk; fi
if [ -z "$ECHO" ]; then ECHO=/bin/echo; fi
if [ -z "$CUT" ]; then CUT=/usr/bin/cut; fi
if [ -z "$ID" ]; then ID=/usr/bin/id; fi
if [ -z "$SU" ]; then SU=/bin/su; fi
if [ -z "$GREP" ]; then GREP=/bin/grep; fi
if [ -z "$CAT" ]; then CAT=/bin/cat; fi

# This script is intended to be run by root.

RUID=`$ID|$AWK -F\ ( '{print $2}'|$AWK -F\ ) '{print $1}'`
if [ ${RUID} != "root" ];then
    $ECHO "You must be logged in as root to run $0."
    $ECHO "Log in as root and restart $0 execution."
    exit 1
fi

case $PLATFORM in
Linux) LD_LIBRARY_PATH=$ORA_CRS_HOME/lib
      OCRCONFIG=/etc/oracle/ocr.loc
      SU="/bin/su -l"
      export LD_LIBRARY_PATH
```

```

    ;;
HP-UX) MACH_HARDWARE=`/bin/uname -m`
    if [ "$MACH_HARDWARE" = "ia64" ]; then
        SO_EXT=so
        NMAPIDIR_64=/opt/nmapi/nmapi2/lib/hpux64
        NMAPIDIR_32=/opt/nmapi/nmapi2/lib/hpux32
    else
        SO_EXT=sl
        NMAPIDIR_64=/opt/nmapi/nmapi2/lib/pa20_64
        NMAPIDIR_32=/opt/nmapi/nmapi2/lib
    fi
    OCRCONFIG=/var/opt/oracle/ocr.loc

LD_LIBRARY_PATH=$ORA_CRS_HOME/lib:$NMAPIDIR_64:/usr/lib:$LD_LIBRARY_PATH
H
    SHLIB_PATH=$ORA_CRS_HOME/lib32:$NMAPIDIR_32:$SHLIB_PATH
    export LD_LIBRARY_PATH
    export SHLIB_PATH
    ;;
SunOS) case ` /bin/uname -i ` in
    i86pc)

LD_LIBRARY_PATH=$ORA_CRS_HOME/lib:/opt/ORCLcluster/lib:/usr/lib:/usr/uc
bllib:$LD_LIBRARY_PATH
        export LD_LIBRARY_PATH
        ;;
    *)

LD_LIBRARY_PATH=$ORA_CRS_HOME/lib32:/opt/ORCLcluster/lib:/usr/lib:/usr/
ucbllib:$LD_LIBRARY_PATH

LD_LIBRARY_PATH_64=$ORA_CRS_HOME/lib:/opt/ORCLcluster/lib:/usr/lib:/usr
/ucbllib:$LD_LIBRARY_PATH_64
        export LD_LIBRARY_PATH
        export LD_LIBRARY_PATH_64
        ;;
    esac
    OCRCONFIG=/var/opt/oracle/ocr.loc
    ;;
AIX) LIBPATH=$ORA_CRS_HOME/lib:$ORA_CRS_HOME/lib32:/usr/lib:$LIBPATH
LD_LIBRARY_PATH=$LIBPATH:$LD_LIBRARY_PATH
OCRCONFIG=/etc/oracle/ocr.loc
    export LIBPATH
    export LD_LIBRARY_PATH
    ;;
OSF1) LD_LIBRARY_PATH=$ORA_CRS_HOME/lib:/shlib:/usr/lib
    export LD_LIBRARY_PATH
    OCRCONFIG=/var/opt/oracle/ocr.loc
    ;;
*) /bin/echo "ERROR: Unknown Operating System"
    exit -1
    ;;
esac

CLSCFG=$CH/bin/clscfg

# clscfg will modify the current configuration stored in the

```

```

# Oracle Cluster Repository.

$CLSCFG -add -nn $CRS_NEW_NODE_NAME_LIST -pn $CRS_NEW_PRIVATE_NAME_LIST
-hn $CRS_NEW_HOST_NAME_LIST

## extract node names from node1,node1-number,node2,node2-number,...
NODES_LIST=`$ECHO $CRS_NEW_NODE_NAME_LIST | $AWK '{
    nElems = split($1, nodeList, ",");
    for (i = 1; i < nElems; i++)
    {
        print nodeList[i];
        i += 2;
    }
}'`

LOCALNODE=`$CH/bin/olsnodes -l`
VIP_STRING=`$CH/bin/srvctl config nodeapps -a -n $LOCALNODE`
NETMASK=`$ECHO $VIP_STRING | $AWK -F"/" '{ print $(NF-1)}'`
NETIFS=`$ECHO $VIP_STRING | $AWK -F"/" '{ print $NF}'`

Ni=1
for i in ` $ECHO $NODES_LIST `
do
    NODE_NAME=$i
    NODE_VIP=`$ECHO $CRS_NEW_NODEVIPS | $CUT -d',' -f$Ni`
    NODEVIP=$NODE_VIP/$NETMASK/$NETIFS

    $ECHO $CH/bin/srvctl add nodeapps -n $NODE_NAME -A $NODEVIP -o $CH
    $CH/bin/srvctl add nodeapps -n $NODE_NAME -A $NODEVIP -o $CH
    Ni=`expr $Ni + 1`
done

. $CH/install/paramfile.crs

PARAM_VALUE=`eval echo \\$CRS_ORACLE_OWNER`
valid=`$ECHO $PARAM_VALUE | $AWK '/^%/ { print "false"; }'`
if [ "$valid" = "false" ];
then
    CRS_ORACLE_OWNER=$ORACLE_OWNER
fi

##Copy ocr.loc file to newly added nodes to sync ocr location
$SU $CRS_ORACLE_OWNER -c "$CH/bin/cluutil -sourcefile $OCRCONFIG -
destfile $CH/srvm/admin/ocr.loc -nodelist $NODES_LIST"

```

rootconfig.sh

Below is a sample rootconfig.sh script (u01/app/oracle/product/10.2.0/crs_1/install/rootconfig.sh) for RAC node customization:

```
#!/bin/sh
#
# rootconfig.sh for Oracle CRS homes
#
# This is run once per node during the Oracle CRS install.
# This script does the following:
# 1) Stop if any GSDs are running from 9.x oracle homes
# 2) Initialize new OCR device or upgrade the existing OCR device
# 3) Setup OCR for running CRS stack
# 4) Copy the CRS init script to init.d for init process to start
# 5) Start the CRS stack
# 6) Configure NodeApps if CRS is up and running on all nodes
#
# NOTE: Use sample paramfile in
$ORA_CRS_HOME/srvm/admin/paramfile.sample for
# setting CRS parameters
# The following commands need to be run before Oracle Cluster Registry
is
# populated.
# This is run during CRS installation and not during RAC

SILENT=false
ORA_CRS_HOME=/u01/app/oracle/product/10.2.0/crs_1
CRS_ORACLE_OWNER=oracle
CRS_DBA_GROUP=oinstall
CRS_VNDR_CLUSTER=false
CRS_OCR_LOCATIONS=/ocfs/clusterware/ocr
CRS_CLUSTER_NAME=crs
CRS_HOST_NAME_LIST=rac_1.vmware.com,1,rac_2.vmware.com,2
CRS_NODE_NAME_LIST=rac_1.vmware.com,1,rac_2.vmware.com,2
CRS_PRIVATE_NAME_LIST=rac_1-priv.vmware.com,1,rac_2-priv.vmware.com,2
CRS_LANGUAGE_ID='AMERICAN_AMERICA.WE8ISO8859P1'
CRS_VOTING_DISKS=/ocfs/clusterware/votingdisk
CRS_NODELIST=rac_1,rac_2
CRS_NODEVIPS='rac_1/rac_1-vip/255.255.0.0/eth0','rac_2/rac_2-
vip/255.255.0.0/eth0'

CRS_PARAMS="ORA_CRS_HOME CRS_ORACLE_OWNER CRS_DBA_GROUP
CRS_VNDR_CLUSTER CRS_OCR_LOCATIONS CRS_CLUSTER_NAME CRS_HOST_NAME_LIST
CRS_NODE_NAME_LIST CRS_PRIVATE_NAME_LIST CRS_LANGUAGE_ID
CRS_VOTING_DISKS CRS_NODELIST CRS_NODEVIPS"

if [ -z "$CP" ]; then CP=/bin/cp; fi
if [ -z "$MV" ]; then MV=/bin/mv; fi
if [ -z "$CHOWN" ]; then CHOWN=/bin/chown; fi
if [ -z "$CHOWNH" ]; then CHOWNH="/bin/chown -h"; fi
if [ -z "$CHGRP" ]; then CHGRP=/bin/chgrp; fi
if [ -z "$CHMOD" ]; then CHMOD=/bin/chmod; fi
if [ -z "$ECHO" ]; then ECHO=/bin/echo; fi
if [ -z "$LS" ]; then LS=/bin/ls; fi
```

```

if [ -z "$KILL" ]; then KILL=/bin/kill; fi
if [ -z "$SLEEP" ]; then SLEEP=/bin/sleep; fi
if [ -z "$LN" ]; then LN=/bin/ln; fi
if [ -z "$LNS" ]; then LNS="/bin/ln -s"; fi
if [ -z "$RM" ]; then RM=/bin/rm; fi
if [ -z "$RMF" ]; then RMF="$RM -f"; fi
if [ -z "$CAT" ]; then CAT=/bin/cat; fi
if [ -z "$MKDIRP" ]; then MKDIRP="/bin/mkdir -p"; fi
if [ -z "$GREP" ]; then GREP=/bin/grep; fi
if [ -z "$CAT" ]; then CAT=/bin/cat; fi
if [ -z "$TOUCH" ]; then TOUCH=/bin/touch; fi
if [ -z "$AWK" ]; then AWK=/bin/awk; fi
if [ -z "$SYNC" ]; then SYNC=/bin/sync; fi
if [ -z "$ID" ]; then ID=/usr/bin/id; fi
if [ -z "$HOSTN" ]; then HOSTN=/bin/hostname; fi
if [ -z "$EXPRN" ]; then EXPRN=/usr/bin/expr; fi
if [ -z "$CUT" ]; then CUT=/usr/bin/cut; fi

###Function for displaying usage of this script
usage()
{
    printf "Usage:"
    printf "$0 [-silent]\n"
    printf "    [-crshome <Oracle CRS home path>]\n"
    printf "    [-paramfile <Path of file specifying CRS parameter\n"
values>]\n"
}

##Function for parsing the command line arguments
parseArgs()
{
    USER_ARGS=$*

    while [ $# -gt 0 ];
    do
        if [ $1 = "-silent" ]; then
            SILENT=1;
        elif [ $1 = "-paramfile" ]; then
            USE_FILE=true
            if [ $# -ge 2 ]; then
                CRS_ENV_FILE=$2;
                shift;
            else
                usage;
                exit 2
            fi
        elif [ $1 = "-crshome" ]; then
            if [ $# -ge 2 ]; then
                ORA_CRS_HOME=$2;
                shift;
            else
                usage;
                exit 2;
            fi
        else
            usage;
            exit 2
        fi
    done
}

```

```

        fi
        shift;
    done
}

##Function for reading CRS parameters from the given param file
readparamfile()
{
    $ECHO "Reading $CRS_ENV_FILE.."
    ## Make sure that this input file has only "LHS=RHS" and empty lines
    ## Any other lines will be considered improper input for root script
    awkOutput=`$AWK 'BEGIN { Ni=0 } \
                    /^#/    { next } \
                    /[A-Z]\=\`*\`\/ { printf("\`%s\`\\n", $0); next } \
                    /[A-Z_]\/=\/ { next } \
                    /^$/    { next } \
                    { printf("\`%s\`\\n", $0); next }'
$CRS_ENV_FILE`

    if [ "x$awkOutput" != "x" ]; then
        $ECHO "The following unexpected input found in CRS parameter file:
$awkOutput"
        exit 1
    fi

    . $CRS_ENV_FILE

    $ECHO "File ($CRS_ENV_FILE) is used for setting CRS variables."
    $ECHO "CRS_ORACLE_OWNER=$CRS_ORACLE_OWNER"
    $ECHO "CRS_DBA_GROUP=$CRS_DBA_GROUP"
    $ECHO "CRS_VNDR_CLUSTER=$CRS_VNDR_CLUSTER"
    $ECHO "CRS_VNDR_LOCATIONS=$CRS_OCR_LOCATIONS"
    $ECHO "CRS_CLUSTER_NAME=$CRS_CLUSTER_NAME"
    $ECHO "CRS_HOST_NAME_LIST=$CRS_HOST_NAME_LIST"
    $ECHO "CRS_NODE_NAME_LIST=$CRS_NODE_NAME_LIST"
    $ECHO "CRS_PRIVATE_NAME_LIST=$CRS_PRIVATE_NAME_LIST"
    $ECHO "CRS_LANGUAGE_ID=$CRS_LANGUAGE_ID"
    $ECHO "CRS_VOTING_DISKS=$CRS_VOTING_DISKS"
    $ECHO "CRS_NODELIST=$CRS_NODELIST"
    $ECHO "CRS_NODEVIPS=$CRS_NODEVIPS"
}

validateInput()
{
    ##validates if any value is assigned to the script variables
    for i in `echo $CRS_PARAMS`;
    do
        PARAM_NAME=$i
        PARAM_VALUE=`eval echo \\$${i}`
        valid=`$ECHO $PARAM_VALUE | $AWK '/^%/ { print "false"; }'`
        if [ "$valid" = "false" ];
        then
            $ECHO "No value set for the CRS parameter $PARAM_NAME. Use
parameter file to set values";
            usage;
            return 1;
        fi
    done
}

```

```

done

return 0;
}

validateVDisks()
{

#Checks for validity of voting disks
vDisks=`$ECHO $1 | $AWK -F',' '{ for(n=1;n<=NF;n++) print $n; }'`;
set ` $ECHO $vDisks `;

ret=0;
while [ $# -gt 0 ];
do
    vdisk=$1;
    if [ ! -r $vdisk ]; then
        $ECHO "\"$vdisk\" does not exist. Create it before proceeding."
        $ECHO "Make sure that this file is shared across cluster nodes."
        ret=1
    else
        $CHOWN $CRS_ORACLE_OWNER $vdisk || { $ECHO $?; exit 1; }
        $CHGRP $CRS_DBA_GROUP $vdisk || { $ECHO $?; exit 1; }
        $CHMOD 644 $vdisk || { $ECHO $?; exit 1; }
    fi;
    shift;
done;

return $ret;
}

validateOCRDisks()
{
    if [ -f $OCR_SYNC_FILE ]; then

        ##Checking the OCR locations used by existing nodes in the cluster
        NEW_OCR_FILE=`$AWK -F= '/ocrconfig_loc/ {print $2}' <
$OCR_SYNC_FILE`
        NEW_OCRMIRROR_FILE=`$AWK -F= '/ocrmirrorconfig_loc/ {print $2}' <
$OCR_SYNC_FILE`

        CRS_OCR_LOCATIONS=$NEW_OCR_FILE
        if [ ! -z "$NEW_OCRMIRROR_FILE" ];
        then
            CRS_OCR_LOCATIONS=$CRS_OCR_LOCATIONS,$NEW_OCRMIRROR_FILE
        fi

        $ECHO "OCR LOCATIONS = " $CRS_OCR_LOCATIONS
    else
        ##Syncing of OCR disks is not required
        $ECHO ""
    fi

return 0;
}

validateSICSS()

```

```

{
  SICSS=false;
  if [ -f "$OCRCONFIG" ]; then
    # ocr.loc file exists and ocr location set here is
    OCRCONFIG_LOC=`$AWK -F= '/ocrconfig_loc/ {print $2}' < $OCRCONFIG`

    # ocr.loc already has a location specified. Check if it is used for
    # single instance CSS/ASM
    LOCAL_FLAG=`$AWK -F= '/local_only/ {print $2}' < $OCRCONFIG`
    LOCAL_ONLY=`$ECHO $LOCAL_FLAG | $AWK '/[[fF]][aA][lL][sS][eE]/ {
print "FALSE" }
                                /[tT][rR][uU][eE]/ { print
"TRUE" }'`
    # Previous installation of 10g single instance
    if [ "$LOCAL_ONLY" = "TRUE" ];then
      SICSS=true;
    fi
  fi

  ret=0;
  if $SICSS;
  then
    $ECHO "CSS is configured for single instance Oracle Databases.
Delete this configuration using the command 'localconfig delete' before
proceeding with RAC configuration."
    ret=1;
  fi

  return $ret;
}

NSSetup()
{
  $ECHO "Setting up NS directories"

  if [ ! -d /var/tmp/.oracle ];
  then
    $MKDIRP /var/tmp/.oracle || { return 1; };
  fi

  $CHMOD 01777 /var/tmp/.oracle || { return 1; }
  $CHOWN root /var/tmp/.oracle || { return 1; }

  if [ ! -d /tmp/.oracle ];
  then
    $MKDIRP /tmp/.oracle || { return 1; }
  fi

  $CHMOD 01777 /tmp/.oracle || { return 1; }
  $CHOWN root /tmp/.oracle || { return 1; }

  return 0;
}

# This script is intended to be run by root.
RUID=`$ID|$AWK -F\('{'print $2}'|$AWK -F\) '{print $1}'`

```

```

if [ ${RUID} != "root" ];then
    $ECHO "You must be logged in as root to run $0."
    $ECHO "Log in as root and restart $0 execution."
    exit 1
fi

#If the hostname is an IP address, let hostname
#remain as IP address
HOST=`$HOSTN`
len1=`$EXPRN "$HOST" : '.*'`
len2=`$EXPRN match $HOST '[0-9]*\.[0-9]*\.[0-9]*\.[0-9]*'`

# Strip off domain name in case /bin/hostname returns
# FQDN hostname
if [ $len1 != $len2 ]; then
    HOST=`$ECHO $HOST | $CUT -d'.' -f1 | $AWK '{ print tolower($0); }'`
else
    HOST=`$ECHO $HOST | $AWK '{ print tolower($0); }'`
fi

# File that $LOGMSG in init.cssd (and other places) will write
MSGFILE=/var/adm/messages

DISABLE_OPROCd=false;

USE_FILE=false

##Parse command line arguments passed to this script
parseArgs $*

if $USE_FILE;
then
    if [ ! -f "$CRS_ENV_FILE" ]; then
        $ECHO "The input file($CRS_ENV_FILE) specifying the parameter
values for CRS variables is not found"
        exit 1
    fi

    ###Read CRS parameter values from param file
    readparamfile
fi

ORACLE_HOME=$ORA_CRS_HOME
export ORA_CRS_HOME
export ORACLE_HOME
CH=$ORA_CRS_HOME

OCR_SYNC_FILE=$CH/srvm/admin/ocr.loc

##validating the parameter values
validateInput || { $ECHO $?; exit 1; }

CRSCTL=$CH/bin/crsctl

#make sure Voting disks exists
validateVDisks $CRS_VOTING_DISKS || { $ECHO $?; exit 1; }

```

```

# Definitions to Add/Remove inittab entries

ADDITAB="$CAT \${IT} | $GREP -v init.evmd | $GREP -v init.cssd | $GREP -v
init.crsd > \${IT}.no_crs || { $ECHO $?; exit 1; } ; \
    $CAT \${IT}.no_crs $CH/crs/admin/inittab > \${IT}.tmp || { $ECHO $?;
exit 1; }; \
    $CP \${IT}.tmp \${IT}.crs || { $ECHO $?; exit 1; }; \
    \${SYNC} ; \${SYNC} ; \${SYNC} ; \
    $MV \${IT}.tmp \${IT} || { $ECHO $?; exit 1; }; \
    \${SYNC} ; \${SYNC} ; \${SYNC} "

RMITAB="$CAT \${IT} | $GREP -v init.evmd | $GREP -v init.cssd | $GREP -v
init.crsd > \${IT}.tmp || { $ECHO $?; exit 1; } ; \
    $CP \${IT}.tmp \${IT}.no_crs || { $ECHO $?; exit 1; }; \
    \${SYNC} ; \${SYNC} ; \${SYNC} ; \
    $MV \${IT}.tmp \${IT} || { $ECHO $?; exit 1; }; \
    \${SYNC} ; \${SYNC} ; \${SYNC} "

CLSCFG_MISCNT=""

UNAME=/bin/uname
PLATFORM=`$UNAME`

case $PLATFORM in
Linux) LD_LIBRARY_PATH=$ORA_CRS_HOME/lib
    export LD_LIBRARY_PATH
    SO_EXT=so
    DIRNAME=/usr/bin/dirname
    ID=/etc/init.d
    RCSDIR="/etc/rc.d/rc3.d /etc/rc.d/rc5.d"
    RCKDIR="/etc/rc.d/rc0.d /etc/rc.d/rc1.d /etc/rc.d/rc2.d
/etc/rc.d/rc4.d /etc/rc.d/rc6.d"
    RC_START=S96
    RC_KILL=K96
    IT=/etc/inittab
    INIT=/sbin/init
    OCRCONFIGDIR=/etc/oracle
    SRVCONFIG=/var/opt/oracle/srvConfig.loc
    OCRCONFIG=/etc/oracle/ocr.loc
    OPROCDDIR=/etc/oracle/oprocd
    SCRBASE=/etc/oracle/scls_scr
    DISABLE_OPROCDDIR=true
    CLSCFG_MISCNT="-misscount 60"
    ;;
HP-UX) MACH_HARDWARE=`/bin/uname -m`
    if [ "$MACH_HARDWARE" = "ia64" ]; then
        SO_EXT=so
        NMAPIDIR_64=/opt/nmapi/nmapi2/lib/hpux64
        NMAPIDIR_32=/opt/nmapi/nmapi2/lib/hpux32
    else
        SO_EXT=s1
        NMAPIDIR_64=/opt/nmapi/nmapi2/lib/pa20_64
        NMAPIDIR_32=/opt/nmapi/nmapi2/lib
    fi

LD_LIBRARY_PATH=$ORA_CRS_HOME/lib:$NMAPIDIR_64:/usr/lib:$LD_LIBRARY_PATH
SHLIB_PATH=$ORA_CRS_HOME/lib32:$NMAPIDIR_32:$SHLIB_PATH

```

```

export LD_LIBRARY_PATH
export SHLIB_PATH
DIRNAME=/bin/dirname
ID=/sbin/init.d
RCSDIR="/sbin/rc3.d"
RCKDIR="/sbin/rc2.d"
RC_START=S960
RC_KILL=K001
IT=/etc/inittab
INIT=/sbin/init
if [ ! -f $ORA_CRS_HOME/lib/libskgxns.${SO_EXT} ]; then
    $MV $ORA_CRS_HOME/lib/libskgxns2.${SO_EXT}
$ORA_CRS_HOME/lib/libskgxns.${SO_EXT}
fi
if [ ! -f $ORA_CRS_HOME/lib32/libskgxns.${SO_EXT} ]; then
    $MV $ORA_CRS_HOME/lib32/libskgxns2.${SO_EXT}
$ORA_CRS_HOME/lib32/libskgxns.${SO_EXT}
fi
if [ -f ${NMAPIDIR_64}/libnmap2.${SO_EXT} ]; then
    $LN -sf ${NMAPIDIR_64}/libnmap2.${SO_EXT}
$ORA_CRS_HOME/lib/libskgxns2.${SO_EXT}
    $LN -sf ${NMAPIDIR_32}/libnmap2.${SO_EXT}
$ORA_CRS_HOME/lib32/libskgxns2.${SO_EXT}
else
    $LN -sf $ORA_CRS_HOME/lib/libskgxns.${SO_EXT}
$ORA_CRS_HOME/lib/libskgxns2.${SO_EXT}
    $LN -sf $ORA_CRS_HOME/lib32/libskgxns.${SO_EXT}
$ORA_CRS_HOME/lib32/libskgxns2.${SO_EXT}
fi

    $CHOWNH $CRS_ORACLE_OWNER $ORA_CRS_HOME/lib/libskgxns2.${SO_EXT}
$CHOWNH $CRS_ORACLE_OWNER
$ORA_CRS_HOME/lib32/libskgxns2.${SO_EXT}

OCRCONFIGDIR=/var/opt/oracle
SRVCONFIG=/var/opt/oracle/srvConfig.loc
OCRCONFIG=/var/opt/oracle/ocr.loc
OPROCDIR=/var/opt/oracle/oprocd
SCRBASE=/var/opt/oracle/scls_scr
;;
SunOS) MACH_HARDWARE=`/bin/uname -i`
case $MACH_HARDWARE in
i86pc)
    CLUSTERDIR=/opt/ORCLcluster

LD_LIBRARY_PATH=$ORA_CRS_HOME/lib:$CLUSTERDIR/lib:/usr/lib:/usr/ucblib:
$LD_LIBRARY_PATH
    export LD_LIBRARY_PATH
    ;;
*)
    CLUSTERDIR=/opt/ORCLcluster

LD_LIBRARY_PATH=$ORA_CRS_HOME/lib32:$CLUSTERDIR/lib:/usr/lib:/usr/ucblib:
b:$LD_LIBRARY_PATH

LD_LIBRARY_PATH_64=$ORA_CRS_HOME/lib:$CLUSTERDIR/lib:/usr/lib:/usr/ucblib:
ib:$LD_LIBRARY_PATH_64

```

```

        export LD_LIBRARY_PATH
        export LD_LIBRARY_PATH_64
        ;;
    esac
    SO_EXT=so
    DIRNAME=/bin/dirname
    ID=/etc/init.d
    RCSDIR="/etc/rc3.d"
    RCKDIR="/etc/rc0.d /etc/rc1.d /etc/rc2.d /etc/rcS.d"
    RC_START=S96
    RC_KILL=K96
    IT=/etc/inittab
    INIT=/sbin/init
    if [ "$MACH_HARDWARE" != "i86pc" ]; then
        if [ ! -f $ORA_CRS_HOME/lib/libskgxn2.${SO_EXT} ]; then
            $MV $ORA_CRS_HOME/lib/libskgxn2.${SO_EXT}
$ORA_CRS_HOME/lib/libskgxn2.${SO_EXT}
        fi
        if [ ! -f $ORA_CRS_HOME/lib32/libskgxn2.${SO_EXT} ]; then
            $MV $ORA_CRS_HOME/lib32/libskgxn2.${SO_EXT}
$ORA_CRS_HOME/lib32/libskgxn2.${SO_EXT}
        fi
        if [ -f $CLUSTERDIR/lib/libskgxn2.${SO_EXT} ]; then
            $LN -sf $CLUSTERDIR/lib/libskgxn2.${SO_EXT}
$ORA_CRS_HOME/lib/libskgxn2.${SO_EXT}
            $LN -sf $ORA_CRS_HOME/lib32/libskgxn2.${SO_EXT}
$ORA_CRS_HOME/lib32/libskgxn2.${SO_EXT}
        else
            $LN -sf $ORA_CRS_HOME/lib/libskgxn2.${SO_EXT}
$ORA_CRS_HOME/lib/libskgxn2.${SO_EXT}
            $LN -sf $ORA_CRS_HOME/lib32/libskgxn2.${SO_EXT}
$ORA_CRS_HOME/lib32/libskgxn2.${SO_EXT}
        fi

        $CHOWNH $CRS_ORACLE_OWNER $ORA_CRS_HOME/lib/libskgxn2.${SO_EXT}
        $CHOWNH $CRS_ORACLE_OWNER
$ORA_CRS_HOME/lib32/libskgxn2.${SO_EXT}
    fi
    OCRCONFIGDIR=/var/opt/oracle
    SRVCONFIG=/var/opt/oracle/srvConfig.loc
    OCRCONFIG=/var/opt/oracle/ocr.loc
    OPROCDDIR=/var/opt/oracle/oprocd
    SCRBASE=/var/opt/oracle/scls_scr
    ;;
AIX) LIBPATH=$ORA_CRS_HOME/lib:$ORA_CRS_HOME/lib32:/usr/lib:$LIBPATH
LD_LIBRARY_PATH=$LIBPATH:$LD_LIBRARY_PATH
export LIBPATH
export LD_LIBRARY_PATH
SO_EXT=a
DIRNAME=/bin/dirname
ID=/etc
RCSDIR="/etc/rc.d/rc2.d"
RCKDIR="/etc/rc.d/rc2.d"
RC_START=S96
RC_KILL=K96
IT=/etc/inittab
INIT=/usr/sbin/init

```

```

ADDITAB="$ECHO \"Adding daemons to inittab\";
    while read line; \
    do \
        rflds=`echo \$line | $CUT -d ':' -f 1-3`; \
        lfld=`echo \$line | $CUT -d ':' -f 4`; \
        mkitab \$rflds:\\"$lfld\"; \
    done < $CH/crs/admin/inittab"
RMITAB="while read line; \
    do \
        identifier=`echo \$line | $CUT -d ':' -f 1`; \
        rmitab \$identifier 2>/dev/null; \
    done < $CH/crs/admin/inittab"
if [ -f /usr/sbin/cluster/utilities/cldomain ]; then
    CLDOMAIN=`/usr/sbin/cluster/utilities/cldomain 2>/dev/null`
    if [[ -n $CLDOMAIN ]]; then
        CLUSTER="true"
    fi
fi
if [ -f /usr/lpp/ssp/bin/spget_syspar ]; then
    SYSPAR=`/usr/lpp/ssp/bin/spget_syspar -n 2>/dev/null`
    if [[ -n $SYSPAR ]]; then
        CLUSTER="true"
    fi
fi

if [ ! -f $ORA_CRS_HOME/lib/libskgxns.${SO_EXT} ]; then
    $MV $ORA_CRS_HOME/lib/libskgx2.${SO_EXT}
$ORA_CRS_HOME/lib/libskgxns.${SO_EXT}
fi
if [ ! -f $ORA_CRS_HOME/lib32/libskgxns.${SO_EXT} ]; then
    $MV $ORA_CRS_HOME/lib32/libskgx2.${SO_EXT}
$ORA_CRS_HOME/lib32/libskgxns.${SO_EXT}
fi
if [[ -n $CLUSTER ]]; then
    $LN -sf $ORA_CRS_HOME/lib/libskgxnr.${SO_EXT}
$ORA_CRS_HOME/lib/libskgx2.${SO_EXT}
    $LN -sf $ORA_CRS_HOME/lib32/libskgxnr.${SO_EXT}
$ORA_CRS_HOME/lib32/libskgx2.${SO_EXT}
else
    $LN -sf $ORA_CRS_HOME/lib/libskgxns.${SO_EXT}
$ORA_CRS_HOME/lib/libskgx2.${SO_EXT}
    $LN -sf $ORA_CRS_HOME/lib32/libskgxns.${SO_EXT}
$ORA_CRS_HOME/lib32/libskgx2.${SO_EXT}
fi

$CHOWNH $CRS_ORACLE_OWNER $ORA_CRS_HOME/lib/libskgx2.${SO_EXT}
$CHOWNH $CRS_ORACLE_OWNER
$ORA_CRS_HOME/lib32/libskgx2.${SO_EXT}

OCRCONFIGDIR=/etc/oracle
SRVCONFIG=/var/opt/oracle/srvConfig.loc
OCRCONFIG=/etc/oracle/ocr.loc
OPROCDIR=/etc/oracle/oprocd
SCRBASE=/etc/oracle/scls_scr
;;
OSF1) LD_LIBRARY_PATH=$ORA_CRS_HOME/lib:/shlib:/usr/lib
export LD_LIBRARY_PATH

```

```

SO_EXT=so
DIRNAME=/usr/bin/dirname
ID="/sbin/init.d"
RCSDIR="/sbin/rc3.d"
RCKDIR="/sbin/rc3.d"
RC_START=S96
RC_KILL=K96
IT=/etc/inittab
INIT=/sbin/init
if [ ! -f $ORA_CRS_HOME/lib/libskgxns.${SO_EXT} ]; then
    $MV $ORA_CRS_HOME/lib/libskgxns2.${SO_EXT}
$ORA_CRS_HOME/lib/libskgxns.${SO_EXT}
fi
if [ -f /shlib/libdlm.${SO_EXT} ]; then
    $LN -sf /$ORA_CRS_HOME/lib/libskgxnr.${SO_EXT}
$ORA_CRS_HOME/lib/libskgxns2.${SO_EXT}
else
    $LN -sf $ORA_CRS_HOME/lib/libskgxns.${SO_EXT}
$ORA_CRS_HOME/lib/libskgxns2.${SO_EXT}
fi
$CHOWNH $CRS_ORACLE_OWNER $ORA_CRS_HOME/lib/libskgxns2.${SO_EXT}
OCRCONFIGDIR=/var/opt/oracle
SRVCONFIG=/var/opt/oracle/srvConfig.loc
OCRCONFIG=/var/opt/oracle/ocr.loc
OPROCDIR=/var/opt/oracle/oprocd
SCRBASE=/var/opt/oracle/scls_scr
DISABLE_OPROCD=true;
;;

*) /bin/echo "ERROR: Unknown Operating System"
exit -1
;;

esac

SCRDIR=$SCRBASE/$HOST
FATALFILE=$SCRDIR/$CRS_ORACLE_OWNER/cssfatal

OCR_CDATA_DIR=$ORA_CRS_HOME/cdata
OCR_BACKUP_DIR=$ORA_CRS_HOME/cdata/$CRS_CLUSTER_NAME

LOGDIR=$ORA_CRS_HOME/log/$HOST
CRS_LOGDIR=$ORA_CRS_HOME/log/$HOST/crsd
EVM_LOGDIR=$ORA_CRS_HOME/log/$HOST/evmd
CSS_LOGDIR=$ORA_CRS_HOME/log/$HOST/cssd
CLSMON_LOGDIR=$ORA_CRS_HOME/log/$HOST/cssd/oclsmon
OCR_LOGDIR=$ORA_CRS_HOME/log/$HOST/client
RES_LOGDIR=$ORA_CRS_HOME/log/$HOST/racg
RES_COREDIR1=$RES_LOGDIR/racgmain
RES_COREDIR2=$RES_LOGDIR/racgeut
RES_COREDIR3=$RES_LOGDIR/racgevtf
ADMIN_LOGDIR=$ORA_CRS_HOME/log/$HOST/admin

# some directories for OPROCD to store the domain
# sockets we use for client communication
OPROCDCHECKDIR=$OPROCDIR/check
OPROCDSTOPDIR=$OPROCDIR/stop
OPROCDFATALDIR=$OPROCDIR/fatal

```

```

###Validate if single instance CSS is configured
validateSICSS || { $ECHO $?; exit 1; }

##Checking if CRS has already been configured
$ECHO "Checking to see if Oracle CRS stack is already configured"
if [ -f $FATALFILE ] && [ -f $OCRCONFIG ];
then
    $ECHO "Oracle CRS stack is already configured and will be running
under init(1M)"
    exit 0
fi

LSDB=$ORA_CRS_HOME/bin/lsdb

if $CRS_VNDR_CLUSTER;
then

    $ECHO "Checking to see if any 9i GSD is up"
    GSDNODE=`$LSDB -g`
    GSDCHK_STATUS=$?

    if [ $GSDCHK_STATUS != 0 ];then
        $ECHO "9i GSD is running on node '$GSDNODE'. Stop the GSD and rerun
root.sh"
        exit 1
    fi
fi

if [ ! -d "$OCRCONFIGDIR" ];then
    $ECHO "$OCRCONFIGDIR does not exist. Creating it now."
    $MKDIRP $OCRCONFIGDIR || { $ECHO $?; exit 1; }
fi

$CHOWN root $OCRCONFIGDIR || { $ECHO $?; exit 1; }
$CHGRP $CRS_DBA_GROUP $OCRCONFIGDIR || { $ECHO $?; exit 1; }
$CHMOD 755 $OCRCONFIGDIR || { $ECHO $?; exit 1; }

# create a number of subdirectories for use by OPROCD
# These should be owned by root and only allow for
# root and group access hence the 770 permissions
if [ "$DISABLE_OPROCD" = "false" ]
then
    if [ ! -d "$OPROCDDIR" ]; then
        $MKDIRP $OPROCDDIR || { $ECHO $?; exit 1; }
        $MKDIRP $OPROCDCHECKDIR || { $ECHO $?; exit 1; }
        $MKDIRP $OPROCDSTOPDIR || { $ECHO $?; exit 1; }
        $MKDIRP $OPROCDFATALDIR || { $ECHO $?; exit 1; }

        $CHOWN root $OPROCDDIR || { $ECHO $?; exit 1; }
        $CHOWN root $OPROCDCHECKDIR || { $ECHO $?; exit 1; }
        $CHOWN root $OPROCDSTOPDIR || { $ECHO $?; exit 1; }
        $CHOWN root $OPROCDFATALDIR || { $ECHO $?; exit 1; }

        $CHMOD 775 $OPROCDDIR || { $ECHO $?; exit 1; }
        $CHMOD 770 $OPROCDCHECKDIR || { $ECHO $?; exit 1; }
        $CHMOD 770 $OPROCDSTOPDIR || { $ECHO $?; exit 1; }
    fi
fi

```

```

    $CHMOD 770 $OPROCDFATALDIR || { $ECHO $?; exit 1; }
fi
fi # End "DISABLE_OPROCD=false"

# Create SCR Home dir, owned by root 755
$MKDIRP $SCRDIR || { $ECHO $?; exit 1; }
$CHOWN root $SCRBASE || { $ECHO $?; exit 1; }
$CHOWN root $SCRDIR || { $ECHO $?; exit 1; }
$CHMOD 755 $SCRBASE || { $ECHO $?; exit 1; }
$CHMOD 755 $SCRDIR || { $ECHO $?; exit 1; }

$CHMOD 644 $IT $IT.orig $IT.no_cssd $IT.cssd $IT.crs $IT.no_crs
2>/dev/null

# Create CRS log directories

for d in $LOGDIR $CRS_LOGDIR $EVM_LOGDIR $CSS_LOGDIR $OCR_LOGDIR
$RES_LOGDIR $ADMIN_LOGDIR $RES_COREDIR1 $RES_COREDIR2 $RES_COREDIR3
$CLSMON_LOGDIR
do
    $MKDIRP $d || { $ECHO $?; exit 1; }
done

for d in $LOGDIR $CRS_LOGDIR
do
    $CHOWN root $d || { $ECHO $?; exit 1; }
    $CHGRP $CRS_DBA_GROUP $d || { $ECHO $?; exit 1; }
done

for d in $EVM_LOGDIR $CSS_LOGDIR $OCR_LOGDIR $RES_LOGDIR $ADMIN_LOGDIR
$RES_COREDIR1 $RES_COREDIR2 $RES_COREDIR3 $CLSMON_LOGDIR
do
    $CHOWN $CRS_ORACLE_OWNER $d || { $ECHO $?; exit 1; }
    $CHGRP $CRS_DBA_GROUP $d || { $ECHO $?; exit 1; }
done

$CHMOD 1755 $LOGDIR || { $ECHO $?; exit 1; }
$CHMOD 1775 $RES_LOGDIR || { $ECHO $?; exit 1; }
$CHMOD 1777 $RES_COREDIR1 || { $ECHO $?; exit 1; }
$CHMOD 1777 $RES_COREDIR2 || { $ECHO $?; exit 1; }
$CHMOD 1777 $RES_COREDIR3 || { $ECHO $?; exit 1; }

for d in $CRS_LOGDIR $EVM_LOGDIR $CSS_LOGDIR $OCR_LOGDIR $ADMIN_LOGDIR
do
    $CHMOD 750 $d || { $ECHO $?; exit 1; }
done

#create EVM auth dir
$MKDIRP $CH/evm/auth
$CHOWN $CRS_ORACLE_OWNER $CH/evm/auth
$CHMOD 1755 $CH/evm/auth

# Create an empty alert log file
ALERTFILE=$LOGDIR/alert$HOST.log
$TOUCH $ALERTFILE
$CHMOD 664 $ALERTFILE
$CHGRP $CRS_DBA_GROUP $ALERTFILE

```

```

$CHOWN root $ALERTFILE

##Checking the repository location used by 9.x installations
SRVCONFIG_LOC=""
if [ -f "$SRVCONFIG" ]; then
  # srvConfig.loc file exists and repository location is
  SRVCONFIG_LOC=`$AWK -F= '/srvconfig_loc/ {print $2}' < $SRVCONFIG`

  if [ "$SRVCONFIG_LOC" = "/dev/null" ]; then
    # 9.x srvconfig_loc is already invalidated. So ignore it
    # take the location entered by user to polulate ocr.loc
    SRVCONFIG_LOC=""
  fi
fi

##Checking the OCR locations used by 10gR1 or previous 10gR2
installations
OCRCONFIG_LOC=""
OCRMIRRORCONFIG_LOC=""
if [ -f "$OCRCONFIG" ]; then
  # ocr.loc file exists and ocr location set here is
  OCRCONFIG_LOC=`$AWK -F= '/ocrconfig_loc/ {print $2}' < $OCRCONFIG`
  OCRMIRRORCONFIG_LOC=`$AWK -F= '/ocrmirrorconfig_loc/ {print $2}' <
$OCRCONFIG`
fi

##If the ocrconfig_loc property is not set in ocr.loc then
##we will check if srvconfig_loc property is set in srvConfig.loc file.
##if set the we set ocrconfig_loc = srvconfig_loc
OCRFILE=$OCRCONFIG
if [ -z "$OCRCONFIG_LOC" ];
then
  ##ocrconfig_loc is empty
  ##Lets set this value to srvconfig_loc from 9.x if exists
  if [ ! -z "$SRVCONFIG_LOC" ];
  then
    #set ocrconfig_loc = srvconfig_loc
    OCRCONFIG_LOC=$SRVCONFIG_LOC
    OCRFILE=$SRVCONFIG
  fi
fi

validateOCRDisks || { $ECHO $?; exit 1; }

CRS_OCR_LOCATION=`$ECHO $CRS_OCR_LOCATIONS | awk -F',' '{print $1}'`
CRS_OCR_MIRROR_LOCATION=`$ECHO $CRS_OCR_LOCATIONS | awk -F',' '{print
$2}'`

####Verify current OCR settings with user entered values
if [ ! -z "$OCRCONFIG_LOC" ];
then
  if [ "$CRS_OCR_LOCATION" != "$OCRCONFIG_LOC" ];
  then
    $ECHO "Current Oracle Cluster Registry location '$OCRCONFIG_LOC'
in '$OCRFILE' and '$CRS_OCR_LOCATION' does not match"
  fi
fi

```

```

    $ECHO "Update either '$OCRFILE' to use '$CRS_OCR_LOCATION' or
variable CRS_OCR_LOCATIONS in rootconfig.sh with '$OCRCONFIG_LOC' then
rerun rootconfig.sh"
    exit 1
fi
else
    #set ocrconfig_loc = CRS_OCR_LOCATION
    OCRCONFIG_LOC=$CRS_OCR_LOCATION
fi

if [ ! -z "$OCRMIRRORCONFIG_LOC" ];
then
    if [ "$CRS_OCR_MIRROR_LOCATION" != "$OCRMIRRORCONFIG_LOC" ];
    then
        $ECHO "Current Oracle Cluster Registry mirror location
'$OCRMIRRORCONFIG_LOC' in '$OCRCONFIG' and '$CRS_OCR_MIRROR_LOCATION'
does not match"
        $ECHO "Update either '$OCRCONFIG' to use
'$CRS_OCR_MIRROR_LOCATION' or variable CRS_OCR_LOCATIONS in
rootconfig.sh with '$OCRMIRRORCONFIG_LOC' then rerun rootconfig.sh"
        exit 1
    fi
else
    #set the mirror location = user entered value for
CRS_OCR_MIRROR_LOCATION
    OCRMIRRORCONFIG_LOC=$CRS_OCR_MIRROR_LOCATION
fi

$ECHO ocrconfig_loc=$OCRCONFIG_LOC > $OCRCONFIG || { $ECHO $?; exit 1;
}
if [ ! -z "$OCRMIRRORCONFIG_LOC" ];
then
    $ECHO ocrmirrorconfig_loc=$OCRMIRRORCONFIG_LOC >> $OCRCONFIG || {
$ECHO $?; exit 1; }
fi
$ECHO local_only=FALSE >> $OCRCONFIG || { $ECHO $?; exit 1; }
$CHOWN root $OCRCONFIG || { $ECHO $?; exit 1; }
$CHGRP $CRS_DBA_GROUP $OCRCONFIG || { $ECHO $?; exit 1; }
$CHMOD 644 $OCRCONFIG || { $ECHO $?; exit 1; }

if [ -f $OCR_SYNC_FILE ]; then
    $RMF $OCR_SYNC_FILE
fi

# First Activate the Volume Groups containing Oracle Cluster Registry
# and the Voting disk.
$CH/css/admin/init.cssd activatevg

##if ocr file does not exist
if [ ! -f "$OCRCONFIG_LOC" ];then
    $CP /dev/null $OCRCONFIG_LOC || { $ECHO $?; exit 1; }
fi

$CHOWN root $OCRCONFIG_LOC || { $ECHO $?; exit 1; }
$CHGRP $CRS_DBA_GROUP $OCRCONFIG_LOC || { $ECHO $?; exit 1; }
$CHMOD 640 $OCRCONFIG_LOC || { $ECHO $?; exit 1; }

```

```

if [ ! -z "$OCRMIRRORCONFIG_LOC" ];
then
##if ocr mirror file does not exist
if [ ! -f "$OCRMIRRORCONFIG_LOC" ];then
  $CP /dev/null $OCRMIRRORCONFIG_LOC || { $ECHO $?; exit 1; }
fi

$CHOWN root $OCRMIRRORCONFIG_LOC || { $ECHO $?; exit 1; }
$CHGRP $CRS_DBA_GROUP $OCRMIRRORCONFIG_LOC || { $ECHO $?; exit 1; }
$CHMOD 640 $OCRMIRRORCONFIG_LOC || { $ECHO $?; exit 1; }
fi

if [ ! -d "$OCR_BACKUP_DIR" ];
then
  $ECHO "OCR backup directory '$OCR_BACKUP_DIR' does not exist.
  Creating now"
  $MKDIRP $OCR_BACKUP_DIR
fi

$ECHO "Setting the permissions on OCR backup directory"
$CHMOD 775 $OCR_CDATA_DIR
$CHMOD 775 $OCR_BACKUP_DIR

NSSetup || { $ECHO $?; exit 1; }

# GSD should be down when this script is called.
# 1. Calls ocrconfig -upgrade to upgrade Oracle Cluster Registry
contents from 9.2 to 10i format.
if $CH/bin/ocrconfig -upgrade $CRS_ORACLE_OWNER $CRS_DBA_GROUP;
then
  $ECHO "Oracle Cluster Registry configuration upgraded successfully"
else
  $ECHO "Failed to upgrade Oracle Cluster Registry configuration"
  exit 1
fi

##Invalidate the existing srvConfig.loc file if it was existing
if [ -f "$SRVCONFIG" ];then
  $ECHO srvconfig_loc=/dev/null > $SRVCONFIG || { $ECHO $?; exit 1; }
  $CHOWN root $SRVCONFIG || { $ECHO $?; exit 1; }
  $CHGRP $CRS_DBA_GROUP $SRVCONFIG || { $ECHO $?; exit 1; }
  $CHMOD 644 $SRVCONFIG || { $ECHO $?; exit 1; }
fi

##-----

# set ownership to root, but give execute permissions to all

$CHOWN root $CH/bin/crs*
$CHMOD 555 $CH/bin/crs*

# only owner(root) should ever invoke crsd

$CHMOD 744 $CH/bin/crsd

if [ ! -d $CH/crs/init ]; then

```

```
$MKDIRP $CH/crs/init
fi

for d in log racg/dump srvn/log
do
if [ ! -d $CH/$d ]; then
$MKDIRP $CH/$d
$CHOWN $CRS_ORACLE_OWNER $CH/$d
$CHGRP $CRS_DBA_GROUP $CH/$d
$CHMOD 775 $CH/$d
fi
done

for d in . lib bin crs crs/init crs/profile crs/script crs/template
crs/auth
do
$CHOWN root $CH/$d
$CHMOD 755 $CH/$d
done

# These are writeable by all
for d in crs/public
do
$CHOWN $CRS_ORACLE_OWNER $CH/$d
$CHMOD 1777 $CH/$d
done

for d in evm evm/init
do
$CHMOD 750 $CH/$d
done

# Writable by the oinstall group but can only be deleted by the crshome
owner
$CHMOD 1770 $CH/evm/log
$CHMOD 1750 $CH/crs/log
$CHMOD 1750 $CH/crs/trace

for d in css css/init css/log css/auth
do
$CHMOD 711 $CH/$d
done

# check directories above $CH are owned by root
# If any are not owned by root, then it allows root escalation
# by the owner of the directory.
d=$CH
while [ "$d" != "/" ]
do
d=`$DIRNAME $d`
set -x- `ls -ld $d`
case $3 in
root) ;; # ok
*) $ECHO "WARNING: directory '$d' is not owned by root";;
esac
done
```

```

# Initialize the SCR settings.
$CRSCTL create scr $CRS_ORACLE_OWNER
if [ "$?" != "0" ]; then
    $ECHO Failure initializing entries in $SCRDIR.
    exit 1
fi

# Initialize the Oracle Cluster Registry the first time we come through
the script.
# The invoker is required to run this script on the install node before
the
# other cluster nodes.

CLSCFG_INIT=false
#
# clscfg - Initialize the Oracle Cluster Registry for the cluster.
Should be
#     done once per cluster install. Overwriting a configuration
#     while any CRS daemon is running can cause serious issues.
#
HOSTNAMES_LIST=$CRS_HOST_MAME_LIST
if [ -z "$HOSTNAMES_LIST" ]; then
    HOSTNAMES_ARGS=""
else
    HOSTNAMES_ARGS="-hn $HOSTNAMES_LIST"
fi

$CH/bin/clscfg -install -nn $CRS_NODE_NAME_LIST -pn
$CRS_PRIVATE_NAME_LIST \
    $HOSTNAMES_ARGS -o $CH -c $CRS_CLUSTER_NAME -l "$CRS_LANGUAGE_ID"
-q $CRS_VOTING_DISKS \
    $CLSCFG_MISCNT

case $? in
    0) CLSCFG_INIT=true;;
    105) $ECHO "Oracle Cluster Registry for cluster has already been
initialized";;
    *) $ECHO "Failed to initialize Oracle Cluster Registry for cluster";
$ECHO $?; exit 1;;
esac

# basic copies first...
$CP $CH/evm/admin/init.evmd $ID/init.evmd || { $ECHO $?; exit 1; }
$CP $CH/css/admin/init.cssd $ID/init.cssd || { $ECHO $?; exit 1; }
$CP $CH/crs/admin/init.crsd $ID/init.crsd || { $ECHO $?; exit 1; }
$CP $CH/crs/admin/init.crs $ID/init.crs || { $ECHO $?; exit 1; }

for rc in $RCSDIR
do
    $RMF $rc/"$RC_START"init.crs
    $LNS $ID/init.crs $rc/"$RC_START"init.crs || { $ECHO $?; exit 1; }
done
for rc in $RCKDIR
do
    $RMF $rc/"$RC_KILL"init.crs
    $LNS $ID/init.crs $rc/"$RC_KILL"init.crs || { $ECHO $?; exit 1; }

```

```
done

SCP $IT $IT.orig || { $ECHO $?; exit 1; }

# Prepare to start the daemons.
$ID/init.crs start

# Check to see if they are going to start.
$ID/init.cssd startcheck CSS
if [ "$?" != "0" ]; then
    $ECHO CRS daemons not set to start. See $MSGFILE for details.
    exit 1
fi

$ECHO "Adding daemons to inittab"
eval $RMITAB
$INIT q
$SLEEP 10
eval $ADDITAB
$INIT q

ALLDONE=false
# Wait until the daemons actually start up
$CRSCTL check install -wait 600
STATUS=$?
case $STATUS in
0)
    # Complete success. This is the last node of the install.
    works=false
    # Wait for CRSD and EVMD to start up
    for i in 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100
105 110 115 120
    do
        if $CH/bin/crs_stat > /dev/null 2>&1
        then
            works=true
            break
        fi

        $ECHO "Waiting for the Oracle CRSD and EVMD to start"
        $SLEEP 5
    done

    if $works
    then
        $ECHO "Oracle CRS stack installed and running under init(1M)"
        ALLDONE=true
    else
        $ECHO "Timed out waiting for the CRS stack to start."
        exit 1
    fi
    ;;
2)
    # Current node is successful so far. Some nodes are not yet
    installed.
    # CRSCTL will have printed out the names of the remaining nodes.
    ;;
```

```
*)
  $ECHO "Failure at final check of Oracle CRS stack."
  $ECHO $STATUS
  exit 1
  ;;
esac

# Startup was successful. Continue with the remainder.
##VIPCA is run by the last node that joins the CSS groups
if $ALLDONE
then
  CRS_PARAMS="CRS_NODEVIPS"

  if validateInput > /dev/null 2>&1
  then
    $ECHO "Running vipca(silent) for configuring nodeapps"
    $CH/bin/vipca -silent -nodelist $CRS_NODELIST -nodevips
  $CRS_NODEVIPS
  else
    $ECHO "Run VIPCA for configuring nodeapps"
    $ECHO "The command to use 'vipca -nodelist <list of cluster
nodes>'"
    fi
  fi

fi

exit 0
```



VMware, Inc. 3401 Hillview Ave Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

© 1998-2007 VMware, Inc. All rights reserved. Protected by one or more U.S. Patents Nos. 6,397,242, 6,496,847, 6,704,925, 6,711,672, 6,725,289, 6,735,601, 6,785,886, 6,789,156, 6,795,966, 6,880,022, 6,944,699, 6,961,806, 6,961,941, 7,069,413, 7,082,598, 7,089,377, 7,111,086, 7,111,145, 7,117,481, 7,149, 843 and 7,155,558; patents pending

