BUILDING A RAC TEST ENVIRONMENT ON VMWARE FOR FREE

Dan Norris, dnorris(at)piocon.com, Piocon

INTRODUCTION AND TERMINOLOGY

This paper offers a recipe for creating a RAC environment suitable for testing DBA skills and even running an application or two for functional (not stress) testing. Many DBAs are interested in RAC and want to add to their skills, but they lack the extra hardware needed to build a multi-node cluster. Enter VMWare. VMWare is virtualization software that allows you to create virtual machines. That is, using VMWare, you can run multiple virtual servers on a single computer. Of course, you must still have enough memory, disk space, and CPU to drive multiple machines, but many of today's workstations have this capability already. Please be sure to review the references found at the end of this document for more information. The installation guides contain most of the information on installation and a few internet resources (cited in the references section) provide the necessary extra steps you need to complete the setup.

To support the discussion that follows, these definitions may be helpful:

Term	Definition
Database	the control files, data files, and online redo logs
Instance	the shared memory and background processes that operate on a database
Clusterware	software that manages cluster membership and monitors the nodes and networks in a cluster
Storage Area Network (SAN)	a storage environment where multiple servers can utilize a single storage array; a storage network is commonly implemented using fiber channel technology
Local Storage	disk space that is available to exactly one node in a cluster; this storage may be part of a SAN or may be direct-attached to the server
Shared Storage	disk space that is available to more than one node in a cluster at the same time; this storage is commonly part of a SAN
Raw Device	the character (unbuffered) special device presented by the operating system
Cluster Filesystem	a special type of filesystem that can be accessed by multiple cluster nodes at the same time
Oracle Automatic Storage Management (ASM)	software that will manage disks directly and provide volume management functions like striping and mirroring; this feature was new with Oracle Database 10g
Single-instance Database	the traditional "stand alone" database configuration used by most Oracle environments today
Multi-instance Database	a single database serviced by more than one instance
Guest OS	in a VMWare environment, the operating system that is installed in the virtual machine
Host OS	in a VMWare environment, the operating system that is installed on the host running the VMWare software

In the sections that follow, you will learn:

- An architectural overview of RAC
- System requirements for a VMWare Cluster
- VMWare product overview
- Configuration and installation processes

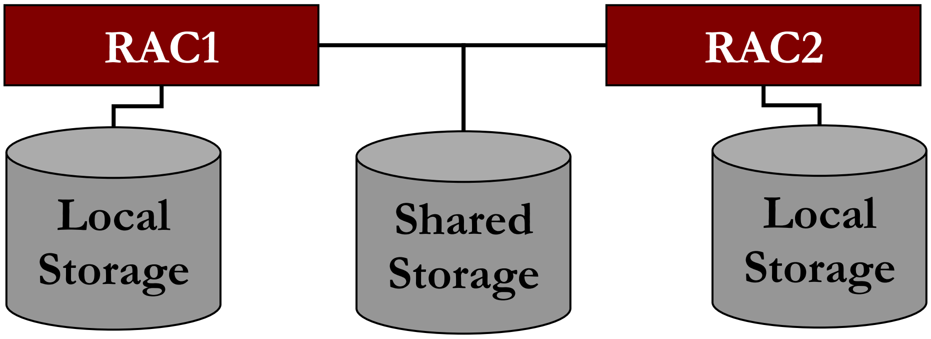
- Challenges and common mistakes

This paper is intended to be used by Oracle DBAs that have at least a little bit of Linux knowledge and have installed some version of Linux at least once. RAC experience is also helpful, but most DBAs will probably be able to complete the installation without having done it before. This document is just a recipe on how to install the products, not how to use them. If you're new to RAC and need a more basic introduction to RAC, check out the "RAC For Beginners: The Basics" whitepaper and presentation found online at http://www.dannorris.com/professional-activities/ under the Presentations section.

Oracle RAC Architecture Overview

The obvious differences between RAC and a traditional single-instance database are apparent on several different levels. The sections that follow describe these differences in more detail.

System Architecture



The diagram above shows a simple RAC cluster architecture. A minimal cluster consists of multiple servers, a private network, a public network, and shared storage. Most DBAs are familiar with the single-instance configuration in which storage is not shared between nodes and all access to the database comes from the single instance.

Sharing the same storage poses several challenges. First, the operating system must be able to handle accesses from other nodes at the same time as it accesses the storage. In the early days of Oracle Parallel Server, this was addressed by using raw devices. The operating systems were accustomed to file systems that supported access from exactly one host. When only one host was involved, the host could manage all locking and synchronization from a single operating system kernel. With multiple hosts, the filesystem access must be coordinated between all hosts sharing the filesystem. This introduces some latency and, until fairly recently, cluster-wide filesystems were very uncommon. Today, several vendors offer cluster filesystems that are supported by Oracle RAC. Oracle also offers its OCFS filesystem for Linux and Windows with reported plans to make it available for more platforms in the future.

Sharing a database also creates challenges. Being able to handle the cross-host locking required to ensure database integrity is the first and most important challenge. Previously, that challenge was addressed by exchanging blocks that required serialization via disk. This was known as a block ping and was a major limiting factor for OPS. With Cache Fusion technology introduced in Oracle9i, the block ping was put on the endangered species list and was only required for a small, infrequently seen set of circumstances.

The additional hardware required for a RAC cluster usually consists of a few additional networks to support the private interconnect. The private interconnect can be made up of multiple physical network interfaces. It is common to utilize two or more gigabit Ethernet interfaces for the interconnect. In past versions, expensive and proprietary interconnect technologies were used. These proprietary interconnects were required because gigabit Ethernet created too much latency and did not have the bandwidth necessary to support some environments. With advancements in Ethernet technology and the ability to utilize multiple interconnects, gigabit Ethernet is the most common interconnect in use today. Infiniband is a relatively new technology that was recently (10.2.0.3) certified for use as an Oracle private interconnect. Infiniband promises to be a significant player in the RAC environment because of its impressive technical specifications.

Finally, before we move on, let's review some points about databases and instances. Oracle RAC is, by definition, a multi-instance database. There is one database. That's o-n-e, as in 1, uno. That (one) database resides on shared storage and is accessed concurrently (that's at the same time) by multiple instances that run on separate hosts. Read that sentence again—it is critical that you understand what you're building and that last sentences sums it all up. Database and instance are defined in the table above. It's a good idea to review their meanings since oftentimes in "normal" Oracle environments, the terms database and instance tend to be used interchangeably. That's fine, but it will benefit you to try to use those words properly in the RAC environment context while you're learning how the environment works.

As we move on to the cluster implementation processes, keep in mind the terminology and "big picture" of what we're trying to build. For me, it always helps to refer back to the overall architecture design we're trying to create frequently during complex installations.

The implementation topics covered in the remainder of this document are:

- System requirements for a VMWare Cluster
- VMWare product overview
- Configuration and installation processes
- Challenges and common mistakes

SYSTEM REQUIREMENTS FOR A VMWARE CLUSTER

In order to build a complete cluster using freely available tools and software, you'll need hardware capable of handling at least two virtual machines running full copies of the OS in addition to an Oracle database instance. The following sections describe some of the configuration that has worked for others.

SUGGESTED MINIMUMS FOR CPU, RAM, DISK

If you really have patience and can wait for extended periods for the installations to complete, you can generally complete the installations and successfully build a cluster with very modest hardware. However, if you put any stress on such a system, it is likely to produce undesirable results as some CPU is required to maintain cluster heartbeats and overall cluster health.

Note that since RAC is not officially supported on VMWare, there are no published or documented minimum requirements. The minimum requirements for a database installation could just be doubled, but that may not be feasible and there's enough padding in those minimums to create some waste. With that in mind, the following minimum hardware requirements are my personal minimums:

Suggested Minimums	
CPU	1 x 2 Ghz CPU
Memory	2.5 Gb RAM
Disk	Minimum is 250 Gb, less is uncomfortable

RECOMMENDED CONFIG

More is almost always better when it comes to CPU and memory on the system. Of course, disk space is usually easy to come by—I don't know that you can easily buy a 200 GB (or smaller) drive these days, so you'll likely have more. My test server has about 1 Tb of disk space and 4 GB of RAM.

For my testing, I used a small Dell server, similar in configuration to a high-end desktop system. Its specifications are in the table below:

My server configuration	
CPU	1 x 3 Ghz CPU with HyperThreading (appears as two CPUs in OEL 4)
Memory	4 Gb RAM
Disk	931 Gb formatted

More resources are always welcome, but I found this little server serves my needs pretty well.

VMWare product overview

VMWare Player

VMWare Player is a free tool that can be used to run existing virtual machines. It cannot create virtual machines, only run existing machines. Regardless of the guest OS in the virtual machine, the player can run on Windows or Linux host OSes.

VMWare Workstation

VMWare Workstation provides the capability to create and run virtual machines on a workstation. It is a licensed product and also has support available for purchase. Virtual machines created with VMWare Workstation can be played with VMWare Player. It is available for evaluation and purchase via the VMWare website. VMWare Workstation has a limitation related to disk sharing, so that's one reason we're using VMWare Server (which has no such restriction).

VMWare Server

VMWare Server is a free product capable of creating and running multiple virtual machines concurrently (up to the limits of the hardware). VMWare Server has some additional server-side features and management tools that make it an attractive solution for server-based VM management. With some web-based tools and easy-to-use GUI management tools as well, multiple people can utilize a server for VMs for testing. Advanced features like multiple snapshots, teaming, automated cloning, and clustering are not available with VMWare Server. Support can be purchased for VMWare Server.

This is the product we'll use to build our RAC cluster.

VMWare ESX Server

VMWare ESX Server is an operating system that provides advanced VM capabilities. Since it does not install on top of a host OS, it resides much closer to hardware resources and, as a result, can optimize access to those resources much better. It also does not typically require as much overhead as a host OS might when using other VMWare products that are installed on top of a Host OS.

Configuration and installation processes

This process is mostly derived from a great article written by Tim Hall at Oracle-Base. The article can be found online at http://www.oracle-base.com/articles/11g/OracleDB11gR1RACInstallationOnOEL5UsingVMware.php (not the easiest URL to remember). You can also find it by going to http://www.oracle-base.com/ and using the "Oracle RAC Articles" link in the left-hand menu. Thanks to Tim for graciously allowing me to use the information in that article for this paper as well. One last note—Tim also wrote and hosts articles on building the same setup using the possible combinations of 10g, 11g, Linux, and Windows.

I will forego the myriad screenshots of each installation step as Tim's article covers that nicely. Instead, I'll describe each step and try to keep you on course to complete the installation to end up with a successful cluster. You don't need anything other than this document to guide you, but reading Tim's article may be helpful.

Prepare host OS

There's not much to do in this step other than install the host OS. I didn't see any need to change much from a default installation. The minimum requirements are buried in the VMWare Server documentation (it isn't easily possible to link directly to them due to the way that VMWare's online documentation is organized). See the earlier section for hardware configuration recommendations. No special kernel parameters, groups, or users are required to support the VMWare software.

If you have support, it's always a good idea to install the latest updates available.

Install VMWare Server

You can download the VMWare Server software from http://www.vmware.com/download/server/. It comes packaged as an RPM. To install the RPM, run:

```
rpm -Uvh VMware-server-1.0.4-56528.i386.rpm
```

Once that's installed (should take just a few seconds), you next need to configure the virtual networks that VMWare Server will provide to the guest OSes. Note that we're using version 1.0.4. The menus and wizards changed slightly from version 1.0.3 to 1.0.4, so while either version should work okay, you might notice differences if you use a version other than 1.0.4 on Linux. The virtual network setup happens as part of the server settings configuration in the next step.

Configure VMWare Server Settings

To configure the VMWare Server, you will need to first obtain a serial number. The product is free, but you still have to register and obtain a serial number in order to use the product. Registration is automated and starts by going to http://register.vmware.com/content/registration.html. With serial number in hand, the next step is to run the interactive vmware-config.pl utility to configure VMWare Server. Most questions can be answered as defaults. Specifically, the only non-default answers I gave were for the following questions:

```
Do you want to be able to use host-only networking in your virtual machines? [yes] no
Do you want this program to set up permissions for your registered virtual
machines?  This will be done by setting new permissions on all files found in
the "/etc/vmware/vm-list" file. [no] yes
In which directory do you want to keep your virtual machine files?
[/var/lib/vmware/Virtual Machines] /u01/vmware/vms
Do you want to enter a serial number now? (yes/no/help) [no] yes
```

At this point, you should be able to start the graphical VMWare Server console. There are several ways to start the console:

- On the server console, you can run "`vmware`" at a command prompt.
- On the server console, you can run the run the shortcut found on the Applications -> System Tools -> VMWare Server Console.
- Or, you can download the "VMware Server Windows client package" from the same download site where you obtained the VMWare Server software and install the Windows VMWare Console to manage your VMWare Server virtual machines over the network from your Windows desktop (this is what I typically do).

Create VM

With the server console fired up, next step is to create the first virtual machine that will serve as the first RAC node. I've found it much easier to install the software if you are able to download the ISO images for Linux to the VMWare Server system directly. You can obtain the Oracle Enterprise Linux (OEL) download at http://edelivery.oracle.com/linux. We're using OEL 5 for this installation.

Create a new virtual machine by following these steps:

- Click on the New Virtual Machine button on the console or going to File -> New -> Virtual Machine. This starts the New Virtual Machine Wizard.
- We choose the Custom configuration and then choose Linux -> Red Hat Enterprise Linux 4 (current VMWare Server doesn't list version 5). OEL is based on Red Hat and that's the reason that this choice is the best one.
- For the name, enter RAC1 (the location automatically defaults appropriately)
- Uncheck the "Make this virtual machine private" checkbox.
- Choose defaults for the Startup/Shutdown options (these settings can be changed later).
- Choose One CPU
- Allocate no more than 40% of the host RAM to this VM. In my case, I have approximately 4 Gb available on my host, so I'm going to allocate 1024 Mb to this VM.
- Choose "Use bridged networking" for the network type for now—we'll add another NIC later, after the VM is created.
- Accept the default I/O adapter settings
- Choose "Create a new virtual disk"
- Choose SCSI for the disk type
- Choose a Disk size of 10.0 GB and uncheck the "Allocate all disk space now" to conserve a little more disk space. Leave the split option checked for ease of maintenance.
- Choose RAC1.vmdk as the disk file name

- Finish the wizard
- Back on the main screen for this VM, choose "Edit virtual machine settings"
- Remove the Floppy device
- Add a new device, type Ethernet Adapter. Choose Bridged for network type.
- Choose OK to exit the VM settings dialog.
- Now we're ready to install the guest OS into this VM.

INSTALL GUEST OS IN VM

I prefer to use ISOs loaded on the VMWare Server. To do that, you just need to edit the CD-ROM virtual device to reference the location of the first ISO file. Edit the virtual machine settings, choose the CD-ROM and then select the "Use ISO image" radio button and browse for the .iso image representing the first CD. Check the "connect at power on" box.

With the CD in place, boot the new virtual machine by clicking "Start this virtual machine". Once booted, follow these steps to install the guest OS:

- Install as you would normally install any server. For detailed instructions on server installation, Tim Hall has another fine article posted at http://www.oracle-base.com/articles/linux/OracleEnterpriseLinux5Installation.php that walks through the installation step-by-step.
- When prompted, choose the following package groups:
  - GNOME Desktop Environment
  - Editors
  - Graphical Internet
  - Text-based Internet
  - Development Libraries
  - Development Tools
  - Server Configuration Tools
  - Administration Tools
  - Base
  - System Tools
  - X Window System
- In my sample installation, I used the following information for host identification:
  - hostname: rac1.dannorris.net
  - IP Address eth0: 192.168.169.51 (public address)
  - Default Gateway eth0: 192.168.169.1 (public address)
  - IP Address eth1: 192.168.150.51 (private address)
  - Default Gateway eth1: none
- For our testing, it's easiest to disable the built-in firewall and also disable the SELinux subsystem. Disabling these components will make it easier to configure and use the system.
- Once the initial installation is complete, install the following packages from the installation media (you need to mount/umount the proper CD images as indicated:

```
# From Enterprise Linux 5 Disk 1
cd /media/cdrom/Server
rpm -Uvh binutils-2.*
rpm -Uvh elfutils-libelf-0.*
rpm -Uvh glibc-2.*
rpm -Uvh glibc-common-2.*
rpm -Uvh libaio-0.*
rpm -Uvh libgcc-4.*
rpm -Uvh libstdc++-4.*
rpm -Uvh make-3.*
cd /
eject

# From Enterprise Linux 5 Disk 2
cd /media/cdrom/Server
rpm -Uvh compat-libstdc++-33*
rpm -Uvh elfutils-libelf-devel-*
rpm -Uvh glibc-headers*
rpm -Uvh glibc-devel-2.*
rpm -Uvh libgomp*
rpm -Uvh gcc-4.*
rpm -Uvh gcc-c++-4.*
rpm -Uvh libaio-devel-0.*
rpm -Uvh libstdc++-devel-4.*
rpm -Uvh unixODBC-2.*
rpm -Uvh unixODBC-devel-2.*
cd /
eject

# From Enterprise Linux 5 Disk 3
cd /media/cdrom/Server
rpm -Uvh sysstat-7.*
cd /
eject
```

- With the installation complete and packages installed, we can move on to configuring the Oracle prerequisites.

CONFIGURE ORACLE PREREQUISITES

The Linux prerequisites for installing Oracle RAC are all the same ones for installing a single instance environment, plus a few additional requirements. The following steps are the "cheat sheet" for configuring the necessary parts of the OS to prepare for the Oracle installations.

- While not an official prerequisite, I usually change the default runlevel from 5 to 3 in order to disable X windows on the console. This saves a sizeable amount of RAM on the server for something that is rarely needed anyway. To change this, modify the initdefault setting in the /etc/inittab file from 5 to 3. Then, as root, run "`init q; telinit 3`" to make the change effective immediately. If you prefer a GUI console, don't make this change—it is optional, but since it frees up memory, performance may improve by following this recommendation.

- Modify the /etc/hosts file to look like the one below (replace the whole file with these contents):

```
127.0.0.1 localhost.localdomain localhost
192.168.169.51 rac1.dannorris.net rac1
192.168.169.52 rac2.dannorris.net rac2
192.168.150.51 rac1-priv.dannorris.net rac1-priv
192.168.150.52 rac2-priv.dannorris.net rac2-priv
192.168.169.151 rac1-vip.dannorris.net rac1-vip
192.168.169.152 rac2-vip.dannorris.net rac2-vip
```

- Add the following lines to the /etc/sysctl.conf file (Note that on OEL 5, many of the parameters that you had to add in previous versions are now in the file by default):

```
kernel.shmmni = 4096
# semaphores: semmsl, semmns, semopm, semmni
kernel.sem = 250 32000 100 128
net.ipv4.ip_local_port_range = 1024 65000
net.core.rmem_default=4194304
net.core.rmem_max=4194304
net.core.wmem_default=262144
net.core.wmem_max=262144
```

- Once the /etc/sysctl.conf file is modified, force the new parameters into effect immediately with "`sysctl -p`"
- Add the following lines to /etc/security/limits.conf:

```
@dba     soft    nproc   2047
@dba     hard    nproc   16384
@dba     soft    nofile  1024
@dba     hard    nofile  65536
```

- Add the following lines to the /etc/pam.d/login file:

```
session    required     /lib/security/pam_limits.so
session    required     pam_limits.so
```

- Add the necessary groups and user to the system with the following commands:

```
groupadd -g 500 oinstall
groupadd -g 501 dba
groupadd -g 502 asmadmin
useradd -u 500 -g oinstall -G dba,asmadmin -s /bin/bash oracle
passwd oracle
mkdir -p /u01/app/oracle /u01/crs/oracle
chown -R oracle:dba /u01
```

- Set default shell variables by adding the following to the /home/oracle/.bash_profile file:

```
ORACLE_BASE=/u01/app/oracle; export ORACLE_BASE
PATH=/usr/sbin:$PATH; export PATH

if [ $USER = "oracle" ]; then
  if [ $SHELL = "/bin/bash" ]; then
    ulimit -p 16384
    ulimit -n 65536
  else
    ulimit -u 16384 -n 65536
  fi
```

- Once these changes are in place, login to the oracle account and run "`ssh-keygen -t dsa`" to create an SSH key. Do not specify a pass phrase for the key.
- After the key is created, run "`cp .ssh/id_dsa.pub .ssh/authorized_keys`" to prepare us for the host equivalency required later for the installations.
- Once the prerequisites are configured, you may wish to install the VMWare Tools. It is an optional step that won't affect the success of the Oracle installations, but may make things easier to manage and may make console usage easier since it will install video and mouse drivers that are generally better than the Linux defaults.

CONFIGURE VIRTUAL SHARED DISKS IN THE VM

Next, we need to configure our "virtual" shared storage that we will use to store the RAC database. To start this process, you need to shutdown the virtual RAC node (rac1) that we just finished configuring. I usually use "`init 0`" to shut down the system. Once it is down, follow these steps to set up the shared storage.

- On the host (VMWare Server host), create a directory to hold the virtual shared disk files: `mkdir -p /u01/vmware/vms/RAC-shared-disks`
- Using the VMWare Server Console, edit the virtual machine settings. Add a device of type "Hard Disk" -> Create a new virtual disk -> SCSI -> Disk size 10 Gb (uncheck "Allocate all disk space now") -> Name: `/u01/vmware/vms/RAC-shared-disks/ocr1.vmdk` -> Click Advanced button
- On the Advanced dialog, Choose SCSI 1:0 as the node for this disk and select Independent and Persistent for the mode.
- Repeat this process for additional disks, until you've configured all disks as follows (settings not listed in the table should match the example above:

Name	Size	Node
`/u01/vmware/vms/RAC-shared-disks/ocr1.vmdk`	10 Gb	SCSI 1:0

Name	Size	Node
/u01/vmware/vms/RAC-shared-disks/voting1.vmdk	10 Gb	SCSI 1:1
/u01/vmware/vms/RAC-shared-disks/asm1.vmdk	10 Gb	SCSI 1:2
/u01/vmware/vms/RAC-shared-disks/asm2.vmdk	10 Gb	SCSI 1:3
/u01/vmware/vms/RAC-shared-disks/asm3.vmdk	10 Gb	SCSI 1:4

- Next step is to hand edit the virtual machine's configuration file to add some parameters. The resulting file should contain these settings (as well as some others). Some of the following lines may already exist. If they already exist, their values should be set as shown below and if they don't exist, add them. Leave any other lines not shown below as they are found.

```
disk.locking = "FALSE"
diskLib.dataCacheMaxSize = "0"
diskLib.dataCacheMaxReadAheadSize = "0"
diskLib.dataCacheMinReadAheadSize = "0"
diskLib.dataCachePageSize = "4096"
diskLib.maxUnsyncedWrites = "0"

scsi1.sharedBus = "VIRTUAL"
scsi1.present = "TRUE"
scsi1:0.present = "TRUE"
scsi1:0.fileName = "/u01/vmware/vms/RAC-shared-disks/ocr1.vmdk"
scsi1:0.mode = "independent-persistent"
scsi1:0.deviceType = "plainDisk"
scsi1:0.redo = ""
scsi1:1.present = "TRUE"
scsi1:1.fileName = "/u01/vmware/vms/RAC-shared-disks/voting1.vmdk"
scsi1:1.mode = "independent-persistent"
scsi1:1.deviceType = "plainDisk"
scsi1:1.redo = ""
scsi1:2.present = "TRUE"
scsi1:2.fileName = "/u01/vmware/vms/RAC-shared-disks/asm1.vmdk"
scsi1:2.mode = "independent-persistent"
scsi1:2.deviceType = "plainDisk"
scsi1:2.redo = ""
scsi1:3.present = "TRUE"
scsi1:3.fileName = "/u01/vmware/vms/RAC-shared-disks/asm2.vmdk"
scsi1:3.mode = "independent-persistent"
scsi1:3.deviceType = "plainDisk"
scsi1:3.redo = ""
scsi1:4.present = "TRUE"
scsi1:4.fileName = "/u01/vmware/vms/RAC-shared-disks/asm3.vmdk"
scsi1:4.mode = "independent-persistent"
scsi1:4.deviceType = "plainDisk"
scsi1:4.redo = ""
scsi1.virtualDev = "lsilogic"
```

- Next, you need to boot the virtual machine again so that you can partition the new disks. Once booted, you can partition the disks with these commands for each disk (`/dev/`sdb through `/dev/sdf`): `fdisk /dev/sdb`, then enter "`n`", "`p`", "`1`", <return>, <return>, and "`w`". That will create a single partition that is the size of the whole disk on each disk.
- In order for the oracle user to be able to access these partitions, you need to add these lines to the `/etc/rc.local`:

```
chown oracle:oinstall /dev/sdb1
chown oracle:oinstall /dev/sdc1
chown oracle:oinstall /dev/sdd1
chown oracle:oinstall /dev/sde1
chown oracle:oinstall /dev/sdf1
chmod 600 /dev/sdb1
chmod 600 /dev/sdc1
chmod 600 /dev/sdd1
chmod 600 /dev/sde1
chmod 600 /dev/sdf1
```

See the notes later in this document about using udev to configure device access. udev is a better and more supportable method and the notes mentioned in that section will provide instructions on how to configure and use udev on your system. For this installation, it isn't needed to get our environment working and stable.

- There are no raw devices to create on OEL 5, so we're done configuring our shared storage at this point.
- Next, we will create our second cluster node by cloning the first node. So, shut down rac1 now ("`init 0`") and follow the next section to clone the rac1 VM to create the rac2 VM.

Clone the VM

In our process, the next step is to create the second virtual machine. We will do this by copying the first one to clone it. To perform the clone, follow these steps:

- Shut down the rac1 virtual machine cleanly (`init 0`) if not already shut down
- Copy all virtual machine file to create a new directory on the VMWare Server host: `cp –R /u01/vmware/vms/RAC1 /u01/vmware/vms/RAC2`
- Edit the `/u01/vmware/vms/RAC2/RAC1.vmx` file to change `displayName = "RAC1"` to be "`RAC2`"
- Using the VMWare Console, open the new VM's configuration file. Then start the new rac2 VM. Leave the rac1 VM down.
- During RAC2 power on, answer "Create" when prompted about handling the identifier.

Prepare the 2nd VM for use

- Once RAC2 is booted, login to the console and update the following files to set the correct network information for the rac2 node:
  - `/etc/sysconfig/network` (change hostname)
  - `/etc/sysconfig/network-scripts/ifcfg-eth0` (replace with following contents—substitute your `HWADDR` as appropriate):
    ```
    DEVICE=eth0
    ONBOOT=yes
    HWADDR=00:0c:29:e8:4f:5e
    NETMASK=255.255.255.0
    IPADDR=192.168.169.52
    TYPE=Ethernet
    ```
  - `/etc/sysconfig/network-scripts/ifcfg-eth1` (replace with following contents—substitute your `HWADDR` as appropriate):
    ```
    DEVICE=eth1
    ONBOOT=yes
    HWADDR=00:0c:29:e8:4f:68
    NETMASK=255.255.255.0
    IPADDR=192.168.150.52
    TYPE=Ethernet
    ```
- Once the files are changed, you need to restart the networking by running "`service network restart`".
- Now rac2 should have the correct hostname (check with the "`hostname`" command).
- It is now safe to restart the RAC1 virtual machine.
- Once both machines are up again, login to each of them and make sure that you can `ping` **each** of the following hostnames **from each node**: rac1 rac1-priv rac2 rac2-priv

Establish Host Equivalency

Our next steps will establish host equivalency for the oracle user so that the oracle account is free to login to the remote node without being prompted for a password. In our case, we're not as concerned with security as we are with keeping the configuration very simple, so we won't protect our private key with a password (not a best practice for production).

Note that once the installations are completed, there is no requirement for host equivalency (other than convenience for maintenance operations), so you may disable the host equivalency established in this section once all installations are complete.

To establish host equivalency, follow these steps:

- Login to rac1 as the oracle user.
- Run "`ssh-keygen –t dsa`". Take all default settings, do not enter any passphrase.
- Run "`cp .ssh/id_dsa.pub .ssh/authorized_keys`"
- For each host in the cluster, ssh to that host to connect by running the following commands in succession from the oracle account. Note that you will be prompted to accept each host's key; respond "yes" when prompted.

  ```
  ssh rac1 date
  ssh rac1-priv date
  ssh rac2 date
  ssh rac2-priv date
  ```
- With all settings in place, now run these commands as the oracle user from rac1: `cd ; scp –rp .ssh rac2:.`
- Now, when you run the ssh statements above, you should see nothing but the output from the "`date`" command—no prompts of any kind.
- At this point, we've done all the OS preconfiguration and have all the necessary items in place to begin installing software. If you're cautious and want to save some time in the event of a catastrophe, this is a good point to shut down the virtual machines and take a copy of them for backup purposes.

INSTALL CLUSTERWARE

Finally, we can start installing some software. In a production installation, we'd usually take some extra time to run the cluster verification utility (CVU) and double-check that all prerequisites are in place. If you've followed the recipe in this document up until this point, we can save a little time and not run that utility. We'll move right in to running the clusterware installer. Follow these steps to install Oracle Clusterware on both nodes.

- Make a Oracle Clusterware 11g installation CD (or the download of the CD contents) available to the rac1 virtual machine.
- If you've followed along carefully to this point, you won't have a graphical console. If that's true, then you'll probably want to start VNC in order to perform the graphical installations. To start VNC, while logged in to rac1 as the oracle user, create the following file named `$HOME/.vnc/xstartup` (you might have to create the `$HOME/.vnc` directory first):

  ```
  xsetroot -solid grey
  gnome-session &
  ```
- Then, you can start the VNC server by running "`vncserver`" from the command prompt. The first time you run it, you'll be prompted to set a password for your desktop. I usually use the same password as the login password for the oracle user.
- Once VNC is running, connect to the server with the a VNC Viewer. Then, ensure your environment has the `ORACLE_BASE` environment variable set to `/u01/app/oracle`.
- After confirming the proper environment settings, run these commands: `cd <download_location>/clusterware; ./runInstaller`
- On the Specify Inventory directory screen, take the defaults (`/u01/app/oraInventory` and oinstall).
- On the Specify Home Details screen, take the defaults of OraCrs11g_home and `/u01/app/11.1.0/crs`
- All prerequisite checks should succeed.
- On the Specify Cluster Configuration screen, the local node (rac1) settings should already be displayed. Add the information for rac2 to the screen using the appropriate names for Public, Private and Virtual.
- On the Specify Network Interface Usage screen, set eth0 to be Public and leave eth1 as Private.
- On the Specify OCR Location screen, choose External Redundancy and enter `/dev/sdb1` for the OCR location.

- On the Specify Voting Disk Location screen, choose External Redundancy and enter `/dev/sdc1` for the voting disk location.
- On the Summary screen, ensure that the Cluster Nodes list includes both nodes and then click Install.
- Once the installations finish, run the root scripts as prompted. Remember to run them in sequence—not concurrently.
- After the root scripts are run, click OK and then the final assistants run. All assistants should run without issue or error.

INSTALL ASM & RDBMS

In our case, we're interested in a test cluster, not necessarily interested in how much it matches any production environment. So, we're going to run ASM from the RDBMS `ORACLE_HOME` in our test environment. That makes one less installation (and a little less disk space) plus also saves time. During our installation, we're also going to have the installer configure ASM for us. Here are the steps to install and configure ASM and RDBMS.

- Once again, we'll use VNC to perform this installation from the rac1 node. Make the RDBMS 11g software CDs or downloads available on rac1 (either locally or via an NFS mount). We'll assume that the software is available at `/mnt/cdrom/db11gr1-linux/database` in our case.
- Within the VNC session that is running as the oracle user account (ensure ORACLE_BASE is set in the environment), run the installer by running these commands: "`cd /mnt/cdrom/db11gr1-linux/database ; ./runInstaller`"
- On the Select Installation Type screen, choose Enterprise Edition.
- On the Install Location screen leave the defaults of Oracle Base: `/u01/app/oracle`; Name: OraDb11g_home1; Path: `/u01/app/oracle/product/11.1.0/db_1`
- On the Specify Hardware Cluster Installation Mode screen, choose Cluster Installation and click "Select All" to choose all nodes.
- All prerequisite checks should be successful.
- On the Select Configuration Option screen, choose Configure Automatic Storage Management (ASM) and enter an ASM SYS password.
- On the Configure Automatic Storage Management screen, you may first have to "Change Disk Discovery Path" to "`/dev/sd*`" in order to see your candidate drives. Then, create one External Redundancy disk group named `DATA` and it uses all three drives (`/dev/sdd1`, `/dev/sde1`, and `/dev/sdf1`), so check the boxes next to all of these drives.
- On the Privileged Operating System Groups screen, take the defaults of OSDBA: dba, OSOPER: oinstall, OSASM: asmadmin.
- On the Oracle Configuration Manager Registration screen, click next to bypass registration.
- Click Install on the summary screen to start the installation. Go get some caffeine and wait. ☺
- Once the installation finishes, you'll be prompted to run the `root.sh` script on each node. Run them sequentially (either node first). Take the default for the local bin directory as prompted by the `root.sh` script.

At this stage, the ASM instances should be running on each node (you should see lots of `asm_XXXX_+ASM?` processes running on each node. The "`crs_stat -t`" command should also show ASM and listener resources configured for each node.

CONFIGURE NEW DATABASE

Next on the task list is to configure a RAC database. We'll use the quick and easy DBCA to create the new database. For this, we'll use the VNC connection once again since DBCA is a GUI tool. Connect to a VNC session on one of the two servers and follow the steps here to create a database instance.

- Set your environment variables as shown here:

```
export ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1
export PATH=$ORACLE_HOME/bin:$PATH
```

- Then run the dbca tool by entering: `dbca &`
- On the opening screen, choose database type "Oracle Real Application Clusters database"
- Choose the operation "Create a Database"
- On the node selection screen, choose Select All
- On the templates screen (step 3 of 19), choose General Purpose or Transaction Processing
- For the Global Database Name enter "RAC.world" and let the SID Prefix default accordingly (default should be RAC)
- On the configuration assistant screen (step 5 of 18), choose the default of "Configuration Enterprise Manager" and you do not need to enable alert notifications nor daily disk backups
- For simplicity (instead of security), choose "Use the Same Administrative Password for All Accounts" on step 6 of 18 and enter a password.
- On the storage selection screen, choose Automatic Storage Management (ASM)
- Select the DATA disk group and click next on the disk group selection screen.
- Use Oracle-Managed Files on the Database File Locations screen (step 9 of 17)
- No need to configure a flash recovery area nor archiving at this time
- You may install the sample schemas if you would like—I didn't.
- I chose to use the default and suggested Typical memory configuration. You can use whatever you'd like, but ensure a minimum SGA size of 400 Mb.
- On the Character Set tab, I prefer to Use Unicode.
- Keep the enhanced 11g security settings on step 13 of 16
- It is recommended to leave the automatic maintenance tasks enabled
- There are no required changes on the Database Storage screen
- On the final screen, click Finish, then click OK on the final summary. Wait.
- Once the DBCA is complete, it will display the summary confirmation screen. Make a note of the URL for the DB Control website.
- When you click Exit on the summary screen, the instances are started and it takes a while before any messages are displayed, so be patient.
- All done! RAC database is created, services created and everything should be running.

Challenges and common mistakes

Firewall settings in guest (Linux) OS

If you fail to disable the Linux firewall or leave SELinux in any setting other than disabled, you will likely encounter security-related issues and warnings. The clusterware installation may have errors due to network ports not being available or open for communication.

Shared storage misconfiguration

One of the common setup issues is configuration of shared storage. On Linux servers, devices are not persistently bound to the same device name. That is, Linux may decide to make `/dev/sdb` into `/dev/sdc` on subsequent reboots, based on a number of factors. As this is a known issue, there is also a known, stable, accepted way to address the issue. With RAC clusters, this is especially daunting, since the device not only needs to be the same name on that node each time, but it needs to be the same name on all nodes in the cluster.

The solution used for most recent Linux versions is to use a subsystem called "udev." The udev subsystem creates a layer of device names on top of the real devices. The names used by udev are controlled by rules and all mappings are done using device identitifers unique to each device in order to be certain that the right devices are identified from each node. This configuration document didn't use udev as vmware keeps the devices fairly stable.

If you do not configure udev or device mapper, the cluster verification utility may report success, but after rebooting the nodes (especially when adding or reconfiguring storage), you may find that the cluster no longer comes up because the device names all changed during the reboot cycle.

To learn more about how to configure udev, see the following notes on Metalink:

- 443996.1 – How to map raw device on RHEL5 and OEL5
- 414897.1 – How to setup /etc/udev/rules.d for OCR and Voting for RAC on SLES10/RHEL5 and EL5
- 371814.1 – Using udev with Oracle Architecture (RAC & ASM) – Red Hat 4.0
- 456239.1 – Understanding Device-mapper in Linux 2.6 Kernel
- 357472.1 – Configuring device-mapper for CRS/ASM
- 465001.1 – Configuring raw devices for Oracle 10gR2 (10.2.0) Clusterware on Red Hat/Enterprise Linux 5 (RH/EL5)

Networking setup issues

It is sometimes difficult to configure the multiple networks required for a RAC deployment. Usually that's because it's hard to know which physical port on the server corresponds to eth0, eth1, etc. To address this, I find it best to configure one network interface at a time instead of connecting them all and then trying to sort things out. If necessary, even going as far as physically disconnecting the wire can help. That's because most Ethernet drivers will report when the link is up or down on the console buffer (which you can see by running "dmesg" on a server). By disconnecting and reconnecting a wire, you should be able to see messages indicating which physical interface lost its link and that should tell you which physical port corresponds to that physical interface.

It's also wise to include the HWADDR in the `/etc/sysconfig/network-scripts/ifcfg-ethX` file to ensure that Linux doesn't renumber the interfaces (which can happen when hardware configurations change). By locking that device to a physical hardware address (MAC address), you ensure that it will always remain "tied" together.

To establish redundancy for network connections, it is best practice to configure network bonding. While not part of this document, you can find more information about using and configuring network bonding on Linux from these articles on Metalink:

- 456055.1 – How to create two different bonding configurations in OEL 4
- 298891.1 – Configuring Linux for the Oracle 10g VIP or private interconnect using bonding driver
- 434375.1 – Linux Ethernet bonding driver

VMWare and VM setup

The VMWare and VM setup steps are straightforward. The primary hang-ups are usually related to network configuration. While we can't educate everyone on all they need to know about networking here, it's lucky that misconfiguring the networking doesn't usually affect the setup in this document. However, if you attempt to access your VMs from another remote host (instead of from their virtual consoles), you may not be able to access them.

The interconnect network should always be the host-only network. During VM setup, you should configure a host-only network. For the purposes of this document, the network choices don't generally make much difference in the ability to use the environment for testing and functional demos. However, you may notice some impact if you perform a lot of network traffic as part of your demo or testing since the interconnect may be using the same network if you have misconfigured the environment. Again, this shouldn't impact the ability to make things "work," but they will perform better and be more reliable if you use the host-only network for your interconnect.

For more information about configuring and using VM networks, see the following links:

- http://media.techtarget.com/searchNetworking/downloads/Book_VMware_Chapter_8.pdf
- http://www.carbonwind.net/VMware/VMwareServerNetworkingOptions/VMwareServerNetworkingOptions.htm

The only other setup that can be somewhat challenging is the shared disk setup. Without going in to much detail, you'll see in this document that you must manually edit the VM's configuration file to allow the disks to be accessible to two VMs at the same time. The parameters added and changed may not be documented well by VMWare, but they have been tested and work well for RAC environments on VMWare. Forgetting to make these manual edits generally results in a completely non-functional RAC environment since the clusterware requires access to shared disks from all nodes.

Linux installation challenges

If you haven't installed Linux before, you may fear that it will be complex or daunting. Many people find that Linux installation is not nearly as difficult as they thought it may have been.

The most common issues with Linux installation usually involve disk partitioning. Some try to size the partitions too small so as to make the best use of space and others try to create separate partitions for many different areas of the filesystem tree. For the "normal" installations I perform, I usually have partitions for / (root), swap, /tmp and /u01 (for Oracle software). With disk sizes being quite large these days, I usually size things this way:

Purpose	Size	Notes
/	15 Gb	
swap	>= 2 Gb	More if RAM size is larger, but typically no larger than 8 Gb
/tmp	2 Gb	If there is a lot of space available, I may make this 4 Gb, but no larger than that.
/u01	the rest	Everything else is allocated here—minimum of 20 Gb, usually much larger.

Linux User Equivalency

Configuring user equivalency on Linux (or any UNIX-based platform) is the process of setting up the user accounts on two or more hosts so that they won't require a password to login and/or run commands on each other's hosts. For many, this configuration is challenging and they don't understand the reasons why their configuration doesn't work since they don't know how it really works.

There are quite a few references online to find help with this configuration and generally, if you follow the right steps, even if you don't really understand how it works, you'll end up with a working configuration. That said, it's always better to understand how it works and especially so in this case since it has to do with system security and messing up the server configuration may result in a less secure system. For our purposes in this document (getting a working RAC configuration), following the steps for user equivalency shown herein is sufficient and does not compromise system security.

- http://www.ardentperf.com/2007/03/27/dbca-slow-check-host-equivalence/
- 372795.1 – How to configure SSH for user equivalence
- 300548.1 – How to configure SSH for a RAC installation

References

- Tim Hall's article on installing 11g RAC on VMWare using Oracle Enterprise Linux 5 at http://www.oracle-base.com/articles/11g/OracleDB11gR1RACInstallationOnOEL5UsingVMware.php
- Oracle Metalink notes (specific notes cited elsewhere in this document)

From the Lawyers

The information contained herein should be deemed reliable but not guaranteed. The author has made every attempt to provide current and accurate information. If you have any comments or suggestions, please contact the author at dnorris(at)piocon.com.