

# Using Oracle Clusterware to Protect Oracle VM Manager

*An Oracle Technical White Paper*  
*July 2009*

# Using Oracle Clusterware to Protect Oracle VM Manager

This document describes the steps that can be used to set up a clustered environment to provide fail-over capabilities for Oracle VM Manager. Oracle VM Manager is the management component of Oracle VM. It is a J2EE application that provides a browser-based management interface and uses an Oracle 10g or higher database repository to store information about every Oracle VM Server and its Virtual Machines (VMs). Oracle VM Manager is also responsible for all communications with the Oracle VM Servers and controls the various tasks to be executed by the servers such as start, stop, pause, create guest VM and so on.

In a default setup, Oracle VM Manager is installed on a single system running both the application container on top of OC4J and the database repository (Oracle Database Express Edition). To make the management components have fail-over and high availability capabilities, there are a few options.

**Protecting the database repository:** the Oracle Real Application Cluster option can be used for the database repository to make the database back-end highly available and scalable. It is possible to install Oracle VM Manager and point to a remote database cluster instance as a repository target. If any of the database nodes fail, the service remains up and the manager application remains available.

**Protect the J2EE container in case of a server or application failure:** using Oracle Clusterware, we can monitor and manage the J2EE container that runs the Oracle VM Manager itself. We register the application with the clusterware, and in case of an application or server failure, it automatically detects and restarts the application on a secondary server.

In this paper we assume that the database service will run on the same server as the Oracle VM Manager J2EE container, which is the default installation mode.

The entire software stack is available from Oracle. The components used as examples are:

- Oracle Enterprise Linux 4(OEL4)
- Oracle VM Manager 2.1.5
- Oracle 11g 11.1.0.6 Clusterware (CRS)

Customers with an Unbreakable Linux support subscription get a use-license for the Oracle Clusterware as part of the subscription. Oracle Enterprise Linux and Oracle VM Manager are freely downloadable and free to use products. All the software can be retrieved from <http://www.oracle.com>.

The shared storage required for this setup can be NFS- or SAN-based. For SAN-based storage we use OCFS2, as the cluster file system to store the binaries and data. OCFS2 is part of OEL4.

The hardware used to prepare this paper was two servers to create a two node cluster with each server having the following specifications:

- Two network interface cards (NICs) (public network eth0, and private network eth1)
- 2GB RAM (minimum requirement 1GB, however 2GB or more recommended)
- Local storage (SATA, IDE, or SCSI) (at least 10GB in size – in our case */dev/hda*)
- Shared storage device (NFS volume or SAN volume) at least 10GB in size (NFS mount point or */dev/sda* in this document)
- Three hostnames and three static IP addresses (public IP, private IP, virtual interface IP) (node1, node1-priv, node1-inter)

For the cluster, one separate hostname and static IP address are required. This is the hostname that would be used for the browser to connect to the application. It is called the service hostname and IP address, for example, ovmm-server.example.com.

The installation steps start from installing the operating system on the servers from scratch, installing Oracle VM Manager, installing and configuring Oracle Clusterware and registering the appropriate application profiles with the clusterware to monitor and restart failed services.

The rest of this paper guides you through installing and configuring the clustered environment to provide fail-over capabilities for Oracle VM Manager.

## **INSTALL ORACLE ENTERPRISE LINUX 4 X86 ON BOTH NODES**

To install Oracle Enterprise Linux 4 on both nodes, download the Oracle Enterprise Linux 4 CD or DVD set from <http://edelivery.oracle.com/Linux>, burn the software on the media and boot the server(s). We used OEL4.7. During the operating system installation:

1. Configure the two network interfaces with a static IP and activate on boot (installer).
2. Configure eth0 with the public IP address and eth1 with the private IP address.

3. Install with **SELinux** disabled.
4. Choose the **No firewall** option.
5. Choose **Customize software** packages to be installed.
6. At the package selection screen, also select **System Tools** under the **System** category and in the details section select the **oracle-validated** and **ocfs2** packages.
7. Start the OEL installation and reboot the server after installation completes.
8. At first boot there is no need to create another user (first boot welcome screen), the oracle-validated RPM already created the user `oracle`. You can change the password of the `oracle` account by using `passwd oracle` as the root user.
9. Edit `/etc/hosts` to include all the public, private and virtual IP addresses and node names

At this point, both servers should have a fully configured OEL4 operating system installed, and we are ready for the next step, configuring shared storage as either:

- SAN storage with OCFS2
- NAS storage with NFS

The following sections describe how to set up both options.

## CONFIGURE SAN STORAGE WITH OCFS2

To configure SAN storage with OCFS2, you need to:

1. Install OCFS2 file system on both nodes.
2. Install Oracle Clusterware on both nodes.

### Install OCFS2 File System on Both Nodes

To install OCFS2 file system, on both nodes:

1. As the `root` user create and format two partitions on the shared storage device on node 1. The first partition is a small partition used for the Oracle Clusterware configuration files, for example, 1GB would be sufficient in size. The second partition is a large partition which will contain the Oracle Clusterware binaries shared home, the Oracle Database XE database binaries and data (when using Oracle VM Manager with the local pre-configured repository) and the Oracle VM Manager binaries.

```
# fdisk /dev/sda
```

Create partition `sda1` as 1GB. Create partition `sda2` as the rest of the volume.

2. As the root user, start `ocfs2console` on node 1:
  - a. Click on **Cluster** and configure (**Add**) node 1, and node 2 (using the public hostname and the private network IP address).
  - b. Click on **Cluster and Propagate configuration**. This asks for the root password for node 1, and node 2, and copies over `/etc/ocfs2/cluster.conf` to the nodes in the cluster.

3. Enable OCFS2 at boot up on both nodes:

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

4. In `ocfs2console` go to **Tasks > Format** and format the two partitions `/dev/sda1` and `/dev/sda2`. You can name the volumes anything you want, for example `sda1` as `oraclecrs`, and `sda2` as `oraclehome`.
5. Mount the partitions and update `/etc/fstab` on both nodes (mount points are `/opt` and `/crs`):

- a. On both nodes execute:

```
# mkdir /crs
```

- b. On both nodes update `/etc/fstab` with following entries (example):

```
/dev/sda1 /crs ocfs2 datavolume 0 0
/dev/sda2 /opt ocfs2 defaults 0 0
```

- c. On both nodes execute:

```
# mount -a
```

- d. Verify the setup by executing the `mount` command again on each node and you should see the following two lines in the output:

```
/dev/sda1 on /crs type ocfs2
(rw,_netdev,datavolume,heartbeat=local)
/dev/sda2 on /opt type ocfs2
(rw,_netdev,heartbeat=local)
```

### Install Oracle Clusterware on Both Nodes

Download Oracle 11g Clusterware for Linux x86 (11.1.0.6) from <http://www.oracle.com> (zip file). To install Oracle Clusterware, on both nodes:

1. As the root user create the directory `/opt/oracle` and provide `oracle` user access on node 1:

```
# mkdir /opt/oracle
# chown oracle:dba /opt/oracle
```

2. Provide `oracle` user access to the `/crs` directory:

```
# chown oracle:dba /crs
```

3. Configure your operating system environment for Oracle Clusterware. Generate SSH keys and copy over the public keys to the other node for both the `oracle` and `root` accounts. For example, as both the `root` and `oracle` users:

```
$ ssh-keygen -t dsa
```

This generates a public and private key in the user home directory `~/.ssh`. Append the public key (the `id_dsa.pub` file) to the `~/.ssh/authorized_keys` file, for example, on node 1 execute:

```
$ ssh node2 cat ~/.ssh/id_dsa.pub >>
~/.ssh/authorized_keys
```

You have to execute the above command for both the `root` and `oracle` accounts to each node (including itself, that is node 1 to node 1, node 2 to node 2).

4. Unzip the CRS product zip file on node 1 and run the installation script to install as the `oracle` user (only do this on node 1).

```
$ cd clusterware ; ./runInstaller
```

The installer requires you enter a number of parameters and perform some configuration:

- a. Enter `/opt/oracle/OraInventory` as the location for the inventory.
  - b. Enter `/opt/oracle/crshome` as the destination for the clusterware Home directory.
  - c. Choose a value for **clustername**, for example, `oraclevm`.
  - d. Edit the **cluster nodes** for node 1 and add node 2.
  - e. Make sure the network interfaces are correct (public and private).
  - f. Choose **external redundancy** for the OCR location and provide `/crs/ocrconfig` as the file name.
  - g. Choose **external redundancy** for the voting disk location and provide `/crs/votingdisk` as the file name.
  - h. Start the installation.
5. Run the `oraInstRoot.sh` and `root.sh` scripts on both node 1 and node 2 as the `root` user:

```
# oraInstRoot.sh
# root.sh
```

6. Verify Oracle Clusterware is configured correctly. As user `root` or `oracle` set the path (`$PATH`) to include the `crs/bin` directory.

```
$ export PATH=$PATH:/opt/oracle/crshome/bin
```

Execute:

```
$ crs_stat -t
```

This should generate output similar to:

```
[oracle@mycomputer ~]# crs_stat -t
Name Type Target State Host
ora....gc1.gsd application ONLINE ONLINE node1
ora....gc1.ons application ONLINE ONLINE node1
ora....gc1.vip application ONLINE ONLINE node1
ora....gc2.gsd application ONLINE ONLINE node2
ora....gc2.ons application ONLINE ONLINE node2
ora....gc2.vip application ONLINE ONLINE node2
```

## CONFIGURE NAS STORAGE WITH NFS

To install NAS storage with NFS, you need to:

1. Mount an NFS partition.
2. Install Oracle Clusterware on both nodes.

### Using NFS as Shared File System for Oracle Clusterware

To configure NFS as the shared file system for Oracle Clusterware, perform the following steps on both nodes:

1. As the root user mount an NFS partition under */opt* and add the following entry in */etc/fstab*:

```
nfsservername:/sharename /opt nfs
rw,bg,hard,nointr,tcp,vers=3,timeo=300,rsize=32768
,wsiz=32768,actimeo=0 0 0
```

2. Make sure you have write permission on the partition mounted, this can easily be checked running the following command

```
# touch /opt/testfile
```

3. Create a directory for the Clusterware configuration files:

```
# mkdir /opt/crs
```

## Install Oracle Clusterware on Both Nodes

To install Oracle Clusterware on both nodes:

1. Download Oracle 11g Clusterware for Linux x86 (11.1.0.6) from <http://www.oracle.com> (zip file).
2. As user root create the directory */opt/oracle* and provide oracle user access on node 1:

```
# mkdir /opt/oracle
```

```
# chown oracle:dba /opt/oracle
```

3. Provide oracle user access to the `/opt/crs` directory:

```
# chown oracle:dba /opt/crs
```

4. Configure your operating system environment for Oracle Clusterware. Generate SSH keys and copy over the public keys to the other node for both the oracle and root accounts. For example, as both root and oracle users run:

```
$ ssh-keygen -t dsa
```

This generates a public and private key in the user home directory `~/.ssh`. Append the public key (the `id_dsa.pub` file) to the `.ssh/authorized_keys` file, for example, on node 1 execute:

```
$ ssh node2 cat .ssh/id_dsa.pub >>
.ssh/authorized_keys
```

You have to execute the above command for both the root and oracle accounts to each node (including itself, that is, node 1 to node 1, node 2 to node 2).

5. Unzip the CRS product zip file on node 1 and run the installation script to install as the oracle user (only do this on node 1).

```
$ cd clusterware ; ./runInstaller
```

The installer requires you enter a number of parameters and perform some configuration:

- a. Enter `/opt/oracle/OraInventory` as the location for the inventory.
  - b. Enter `/opt/oracle/crshome` as the destination for the clusterware Home directory.
  - c. Choose a value for **clustername**, for example, `oraclevm`.
  - d. Edit the **cluster nodes** for node 1 and add node 2.
  - e. Make sure the network interfaces are correct (public and private).
  - f. Choose **external redundancy** for the OCR location and provide `/opt/crs/ocrconfig` as the file name.
  - g. Choose **external redundancy** for the voting disk location and provide `/opt/crs/votingdisk` as the file name.
  - h. Start the installation.
6. Run the `oraInstRoot.sh` and `root.sh` scripts on both node 1 and node 2 as the root user:

```
# oraInstRoot.sh
# root.sh
```

7. Verify Oracle Clusterware is configured correctly. As user root or oracle set the path (\$PATH) to include the *crs/bin* directory:

```
$ export PATH=$PATH:/opt/oracle/crs/bin
```

Execute:

```
# crs_stat -t
```

This should generate output similar to:

```
[oracle@mycomputer ~]# crs_stat -t
Name Type Target State Host
ora....gc1.gsd application ONLINE ONLINE node1
ora....gc1.ons application ONLINE ONLINE node1
ora....gc1.vip application ONLINE ONLINE node1
ora....gc2.gsd application ONLINE ONLINE node2
ora....gc2.ons application ONLINE ONLINE node2
ora....gc2.vip application ONLINE ONLINE node2
```

## INSTALL ORACLE VM MANAGER ON NODE 1 ONLY

To install Oracle VM Manager from the Oracle VM Manager CD or ISO image on node 1 only:

1. Download Oracle VM Manager from <http://edelivery.oracle.com/oraclevm> and burn the ISO image to media or use the ISO image directly.
2. On node 1, insert the Oracle VM Manager CD, or mount the ISO image:

```
# losetup /dev/loop0 OracleVM-Manager-2.1.5.iso ;
mount /dev/loop0 /mnt
# mount /dev/cdrom /mnt
```

3. Run the `runInstaller.sh` script on the CD mount point as the root user to install, uninstall or upgrade Oracle VM Manager:

```
# sh runInstaller.sh
```

Install either using the locally installed Oracle Database XE (which is part of Oracle VM Manager installation) or using an external repository database.

Do not choose to autostart the database on startup of the operating system.

## SYNCHRONIZE THE NODES FOR MANAGER FAILOVER

You can manually run the commands below from node 1 only. The argument \$1 is the hostname of node 2; and the argument \$2 is the hostname of the public service name (`servicenodename`) used for the manager - this is the hostname that will failover with the cluster.

The commands copy over the required files to node 2 and prepares node 2 to be able to start the services, in particular `/etc/init.d/oc4j-ha`, `/etc/init.d/oracle-xe-ha`, `/etc/sysconfig/oracle-xe`, and creates the `/var/log/ovm-manager` directory.

The commands also move the Oracle Database XE instance from `/usr/lib/oracle` to `/opt/oracle/xes` on the shared volume and modifies the `listener.ora` and `tnsnames.ora` files to use the `servicenodename`.

1. Shut down `oc4j` and `oracle-xe`

```
# /etc/init.d/oc4j stop
# /etc/init.d/oracle-xe stop
```

2. Move Oracle-xe to `/opt/oracle/xes` (shared volume)

```
# mv /usr/lib/oracle/xes /opt/oracle
# ln -sf /opt/oracle/xes /usr/lib/oracle/xes
```

3. Copy the original scripts `/etc/init.d/oc4j` and `/etc/init.d/oracle-xe` and copy them to `-ha` in the same directory then add a new entry "check". By default, init scripts provide start stop status.

Modify the `/etc/init.d/oc4j-ha` by adding the check entry:  
(check is essentially the same as `oc4j_status` function)

```
case "$1" in
.....
  check) # for CRS
    oc4j_status
    ;;
  *)
    echo $"Usage: `basename $0`
{start|stop|status|check}"
    exit 1
esac
```

Modify the `/etc/init.d/oracle-xe-ha` by adding the check entry:

```
docheck() {
  xe_running=`ps auxww|grep -w xe_pmon_XE |grep
-v grep`
  if [ -z "$xe_running" ]
  then
    echo "Oracle XE is down."
    exit 1
  else
    echo "Oracle XE is running."
    exit 0
  fi
  return
}
case "$1" in
```

```

.....
check)
docheck
    ;;
.....
*)
    echo $"Usage: $0 {start|stop|restart|force-
reload|configure|status|enable|disable|check}"
    exit 1
esac

```

#### 4. Copy files to node 2

```

# ssh root@$1 "mkdir /usr/lib/oracle"
# ssh root@$1 "ln -sf /opt/oracle/xe
/usr/lib/oracle/xe"
# ssh root@$1 "mkdir /var/log/ovm-manager"
# scp /etc/init.d/oracle-xe* $1:/etc/init.d
# scp /etc/init.d/oc4j* $1:/etc/init.d
# scp /etc/sysconfig/oracle-xe $1:/etc/sysconfig
# scp /etc/oratab $1:/etc
# cd
/opt/oracle/xe/app/oracle/product/10.2.0/server/netwo
rk/admin
# sed -i "s/localhost/$2/g" listener.ora
# sed -i "s/localhost/$2/g" tnsnames.ora

```

### REGISTER THE ORACLE VM MANAGER PROFILE WITH CRS ON NODE 1

To register the Oracle VM Manager profile with CRS on node 1:

1. Create resources for the three Oracle VM Manager resources (virtual IP, DB, OC4J). As user root execute the following commands (192.168.3.3 in this example is the IP address of the servicenodename).

```

# crs_profile -create ovmm -t application -a
/opt/oracle/crshome/bin/usrvip -o
oi=eth0,ov=192.168.3.3,on=255.255.252.0
# crs_register ovmm
# crs_profile -create ovmm db -t application -r
ovmm -a /etc/init.d/oracle-xe-ha -o ci=20,ra=5
# crs_register ovmm_db
# crs_profile -create ovmm_oc4j -t application -r
ovmm -a /etc/init.d/oc4j-ha -o ci=20,ra=5
# crs_register ovmm_oc4j

```

Note: If you have created a separate database repository on a different server, you do not have to create the *ovmm\_db* service.

2. Verify the above services registered. Execute:

```
# crs_stat -t
```

Output similar to the following is displayed:

```
ovmm application OFFLINE OFFLINE
```

```
ovmm_db application OFFLINE OFFLINE  
ovmm_oc4j application OFFLINE OFFLINE
```

3. Start the service by using the following command as the root user:

```
# crs_start ovmm  
# crs_start ovmm_db  
# crs_start ovmm_oc4j
```

At this point your cluster is configured, your services are registered, monitored and managed by the clusterware. When you shut down, kill or crash node 1, the services auto-restart on node 2 (and vice versa). If the application were to unexpectedly fail, it will automatically restart.

You now have a full high availability install of Oracle VM Manager.



Using Oracle Clusterware to Protect Oracle VM Manager  
July 2009, Version 1.1  
Author: Wim Coekaerts

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
[oracle.com](http://oracle.com)

Copyright © 2009, Oracle and/or its affiliates. All rights reserved.  
This document is provided for information purposes only and the contents hereof are subject to change without notice.  
This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. 0408