



VM

ORACLE
TECHNOLOGY NETWORK

Virtual Technology Summit

Hands-On Learning With Oracle and Community Experts

Where Technology and Community Meet



Build a Complete Highly-Available Oracle VM Architecture from Server to App

Oracle VM, Oracle Linux, KSplice, Oracle Clusterware and MySQL

BY: SIMON COTER

CONTRIBUTORS: DOAN NGUYEN, CHRISTOPHE PAULIAT, OLIVIER CANONGE, BRUNO BOTREAU

ORACLE®

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Table of Contents

Disclaimer	1
Lab objective & Introduction	1
Preparation (done before lab)	2
Summary of steps	3
Start the servers (VirtualBox VMs)	3
Connect to the Oracle VM Manager 3.3.1 and become familiar with the product	5
Detailed architecture of the Lab	10
Connect to the Oracle VM Servers and verify that everything is ready to accommodate the lab	11
Connect to the Oracle VM guests and become familiar with Oracle Linux and Oracle Clusterware (Grid Infrastructure)	12
High availability general concepts: understand what we are going to demonstrate	14
The architecture proposed in this lab is based on:	14
 Oracle VM	14
 Oracle Linux	14
Oracle Ksplice 	15
 Oracle Clusterware	15

Demonstrate high-availability features covered by Oracle Ksplice	17
Demonstrate high-availability features covered by Oracle Clusterware	20
Demonstrate high-availability features covered by Oracle VM	29
Oracle VM Live-Migrate	29
Oracle VM High-Availability	29
Upshot: Oracle VM High Availability	37
Appendix A: preparation of the environment before the lab	38
Purpose	38
Download required binaries	38
Installation of Oracle VM VirtualBox	38
Installation of Oracle VM Server (2 servers to install)	39
Installation of Oracle VM Manager	40
Discover Oracle VM Servers, add the file server by Oracle VM Manager, create the Server Pool	42
Import the “Oracle VM Template” Oracle Linux 6.5 x86-64	45
Modify the network configuration	46
Create three new guest(s) servers	47
Install Oracle Grid Infrastructure 12c on nodes “vdb01.oow.local” and “vdb02.oow.local”	57
Install Oracle Grid Infrastructure standalone Agents on clustered nodes	65
Download and install Oracle MySQL 5.6 Community Edition	65
Create ASM Clustered Filesystem (ACFS) for application-tier	66

Apply custom changes to the environment for the future resource clustering	67
On node “vdb01.oow.local” create a demo MySQL database	67
On node “vdb01.oow.local” install a demo web-app	68
Create an user app-vip clustered with Oracle Grid Infrastructure 12c	68
Configure MySQL to be managed by Oracle Clusterware 12c	69
Configure Apache “httpd” to be managed by Oracle Clusterware 12c	69
Configure Clusterware resources permission(s) to allow “oracle” account to manage them	71
Add a wrapped script to see clusterware resources status	71
Oracle Ksplice subscription and activation	72
Document references.	72

Lab Objective & Introduction

"In this lab, you will learn from our field experts on best practices of implementing and using a complete Oracle VM high-available solution. This lab helps to demonstrate how Oracle products such as Oracle VM, Oracle Linux, Ksplice and Oracle Clusterware (all products included with Oracle Linux and Oracle VM Support) allows to create a solution that is able to cover all the high-availability requirements. This lab also walks you through the managing, by Oracle Clusterware, of a custom web-application. The implementation of this lab helps to deploy an enterprise-proven high-available infrastructure software layer at zero license cost to your virtualization or Linux environment."

This hands-on lab takes you through the best practices on how to exploit all products mentioned above to build up a clustered solution without license costs.

Oracle VM is a free license product and it's the only virtualization x86 software solution certified for all Oracle products; Oracle server virtualization is designed to enable rapid enterprise application deployment and simplify lifecycle management.

Oracle Linux, a Linux free provides the latest innovations, tools, and features that enable you to innovate, collaborate, and create solutions across traditional, cloud-based, and virtual environments.

Oracle Linux is easy to download, free to use, free to distribute, and free to update.

Oracle Linux offers zero-downtime kernel updates with **Ksplice**.

Linux management and high availability (by **Oracle Clusterware**) are provided to Oracle Linux support customers at no additional charge.

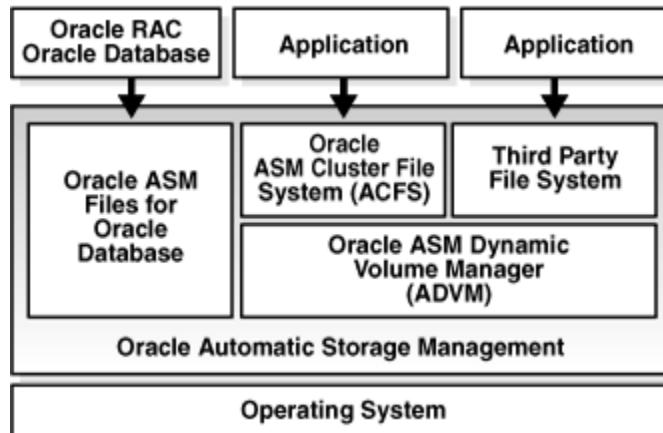
Ksplice is available for Oracle Linux, free of charge, for Oracle Linux customers with a Premier support subscription.

Ksplice lets you apply 100% of the important kernel security updates without rebooting. You don't need to stop any running applications and you don't need to reboot to install.

Oracle Clusterware 12c Release 1 is the integrated foundation for Oracle Real Application Clusters (RAC) and the High Availability (HA) and resource management framework for all applications on any major platform.

Oracle Grid Infrastructure Agents (XAG) is the framework that provides a complete, ready to use application HA solution that contains pre-defined Oracle Grid Infrastructure resource configurations and agents to integrate applications for complete application HA.

Oracle Automatic Storage Management Cluster File System (Oracle ACFS) is a multi-platform, scalable file system, and storage management technology that extends Oracle Automatic Storage Management (Oracle ASM) functionality to support customer files maintained outside of Oracle Database. Oracle ACFS supports many database and application files, including executables, database trace files, database alert logs, application reports, BFILEs, and configuration files:



MySQL is the world's most popular open source database, enabling the cost-effective delivery of reliable, high-performance and scalable Web-based and embedded database applications.

Preparation (Must be done before lab)

In order to exercise this lab in the one hour time slot, there are pre-lab set up steps that must be performed. There are 2 options to perform the pre-lab set up.

Option 1: Download the pre-configured *.OVA files for the lab. The *.OVA files are VMs that have been pre-configured and pre-installed with Oracle VM Server, Oracle VM Manager and guest operating system Oracle Linux. There are 3 OVA files for this lab. They can be downloaded from here:

<http://www.oracle.com/technetwork/server-storage/vm/downloads/index.html>

OVA1: ovmm.oow.local.ova 24GB

OVA2: ovs01.oow.local.ova 540MB

OVA3: ovs02.oow.local.ova 483 MB

Option 2: Download the software components and follow the Pre-lab set up steps below:

Pre-lab set up:

- Install Oracle Linux 6.5 (64 bits) on the laptop.
- Install Oracle VM VirtualBox 4.2.x + extensions on the laptop.
- Add host-only Networks vboxnet0 and vboxnet1 on VirtualBox configuration.
- Install and configure an Oracle VM Manager 3.3.1 server in a VirtualBox virtual machine.
- Install and configure two Oracle VM Server 3.3.1 servers in two VirtualBox virtual machines.
- Create a clustered Oracle VM server pool.
- Configure networks dedicated for each role.
- Create an Oracle VM repository that will guest virtual machines (60gb in our example).
- Create two Oracle Linux 6.5 virtual-machines starting from Oracle VM Templates.
- Configure Oracle Linux 6.5 virtual machines.
- Configure the storage (virtual-disks) on Oracle VM virtual machines.
- Install and configure Oracle Clusterware 12c.
- Install and configure Oracle Clusterware 12c agents to manage MySQL database.
- Install and configure Oracle Ksplice.
- Install and configure Oracle MySQL Enterprise 5.6.
- Configure Oracle ACFS Cluster filesystem.
- Install Mediawiki demo application.
- Proceed with clustering all applications such as VIP, MySQL database and custom web-app (mediawiki)

Note: to run this lab at home of office

- Requirements:
 - Have an X86 machine with at least 16GB of RAM and 4 cores CPU.
 - Any X86 Operating System supported by Oracle VM VirtualBox is OK (Microsoft Windows, Most linux distributions, Oracle Solaris X86, Apple Mac OSX, ...)
- Read appendix A



Summary of steps

In this lab, we will execute the following steps:

- 1) Connect to Oracle VM Manager and become familiar with the product.
- 2) Connect to Oracle VM Servers and verify that everything is ready to accommodate the lab.
- 3) Connect to Oracle VM guests and become familiar with Oracle Linux and Oracle Clusterware.
- 4) High availability general concepts: understand what we are going to demonstrate.
- 5) Demonstrate high-availability features covered by Oracle Ksplice
- 6) Use Ksplice to update UEK, rollback and re-update while verifying Kernel release online.
- 7) Demonstrate high-availability features covered by Oracle Clusterware.
- 8) Use Oracle Clusterware to execute a managed service switchover (MySQL and WebApps).
- 9) Simulate a process failure within the guest (managed by Oracle Clusterware)
- 10) Simulate a virtual-guest fault (managed by Oracle Clusterware)
- 11) Understand which high-availability features are covered by Oracle VM.
- 12) Use Oracle VM to execute a guest live-migration.
- 13) Simulate a physical server fault (managed by Oracle VM and Oracle Clusterware)

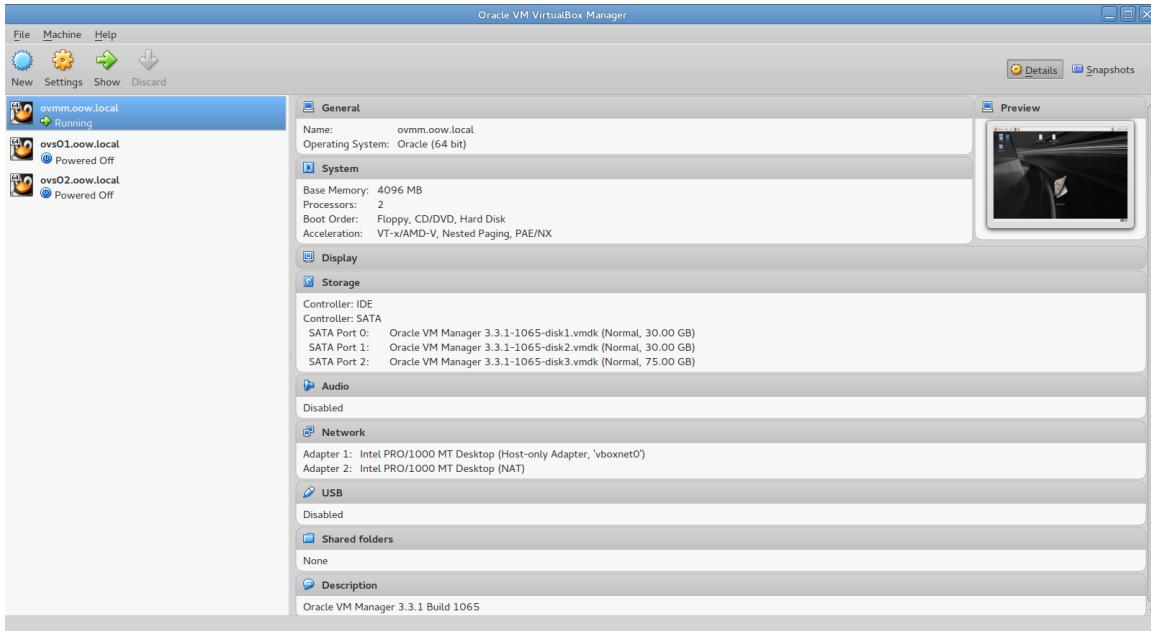
Start the servers (VirtualBox VMs)

As previously explained, we will use Oracle VM VirtualBox to host the 3 servers (Oracle VM Manager and two Oracle VM Servers) on a single laptop.

Those 3 servers were pre-installed and preconfigured before this lab to same time. Thus, you just have to start them here.

IMPORTANT: Since the VMs startup can take time, we advise you to do this as soon as possible following the steps below and then take time to read this documentation.

- a) Start the Oracle VM VirtualBox console if not yet started by clicking on icon 
- b) In this console, you should see the 3 VMs we will use in this lab.



- c) Select the VM called “**ovmm.0ow.local**” and click on the icon to start it
 d) Verify its console and wait that the network and nfs-services are started as show in the figure below (press ESC to obtain console details):

ovmm.0ow.local [Running] – Oracle VM VirtualBox

```

Machine View Devices Help
Starting irqbalance: [ OK ]
Starting rpcbind: [ OK ]
Starting NFS statd: [ OK ]
Starting kdump: [ FAILED ]
Starting system message bus: [ OK ]
Starting cups: [ OK ]
Mounting filesystems: [ OK ]
Starting acpi daemon: [ OK ]
Starting HAL daemon: [ OK ]
Retrigger failed udev events [ OK ]
Loading autofs4: [ OK ]
Starting automount: [ OK ]
Starting NFS services: [ OK ]
Starting NFS quotas: [ OK ]
Starting NFS mountd: [ OK ]
Starting NFS daemon: [ OK ]
Starting RPC idmapd: [ OK ]
Starting the VirtualBox Guest Additions [ OK ]
Starting VirtualBox Guest Addition service [ OK ]
Starting SCSI target daemon: [ OK ]
Starting mcelog daemon
Starting sshd: [ OK ]
Starting oraclevm-template...
Starting OVMM MySQL..._

```

The output shows several services starting, with some like kdump and NFS services failing. A red circle highlights the NFS services section.

- e) Select the VM called “**ovs01.0ow.local**” and click on the icon to start it

- f) Select the VM called “**ovs02.oow.local**” and click on the icon  to start it
- g) Wait for the 3 VMs to be ready
- o Wait for the prompt (desktop started) on “**ovmm.oow.local**” VM console
 - o When this prompt is displayed, all VMs are ready (since Oracle VM Manager is the longest to start)

Connect to the Oracle VM Manager 3.3.1 and become familiar with the product

- a) On your Linux physical desktop open a Firefox browser and connect to the **Oracle VM Manager 3.3.1** console using URL <https://192.168.56.200:7002/ovm/console>

In the case that you receive some warnings proceed as described in the following screens:

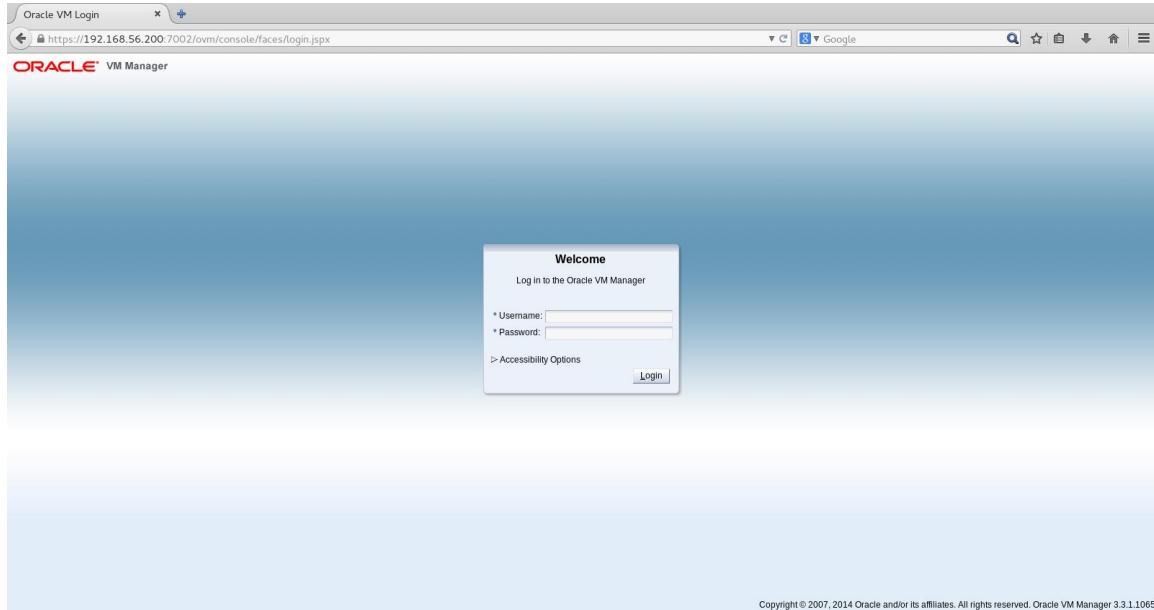
Click on “I Understand the Risks” and on “Add exception”:



Finally click on “Confirm Security Exception”:



After security exception managed you should get the following login window:



- b) Log in using the following credentials:
 Login : **admin** (default Oracle VM Manager Administrator)
 Password: **Welcome1** (W is uppercase)

- a) Once logged in the Oracle VM Manager console, go to the “**Servers and VMs**” tab, expand and select **ovmpool.0ow.local**, select it, change perspective view to “**Virtual Machines**” and verify the status of Oracle VM pool and Server; everything should be as in this picture.

You need to see also that virtual-guests **vdb01**, **vdb02** and **lvtest** are active and running.

The screenshot shows the Oracle VM Manager interface with the following details:

- URL:** https://192.168.56.200:7002/ovm/console/faces/resource/resourceView.jspx
- User:** admin
- Perspective:** Virtual Machines
- Table Headers:** Name, Status, Tag(s), Event Severity, Server, Max. Memory (MB), Memory (MB), Max. Processors, Processors, Keymap, Operating System
- Table Data:**

Name	Status	Tag(s)	Event Severity	Server	Max. Memory (MB)	Memory (MB)	Max. Processors	Processors	Keymap	Operating System
lvtest.0ow.local	Running		Informational	ov01.0ow.local	1	1	1	1	en-us	Oracle Linux 6
vdb01.0ow.local	Running		Informational	ov01.0ow.local	2560	2560	2	2	en-us	Oracle Linux 6
vdb02.0ow.local	Running		Informational	ov02.0ow.local	2560	2560	2	2	en-us	Oracle Linux 6
- Left Sidebar:** Shows Server Pools (ovmpool.0ow.local, ov01.0ow.local, vdb01.0ow.local, Unassigned Servers, Unassigned Virtual Machines).

- b) Click on “**Networking**” tab and verify defined networks and their roles:

192.168.56.0 = VirtualBox host-only Network used for default management, heartbeat and live-migrate network; also guests servers use this network to obtain access from our laptop

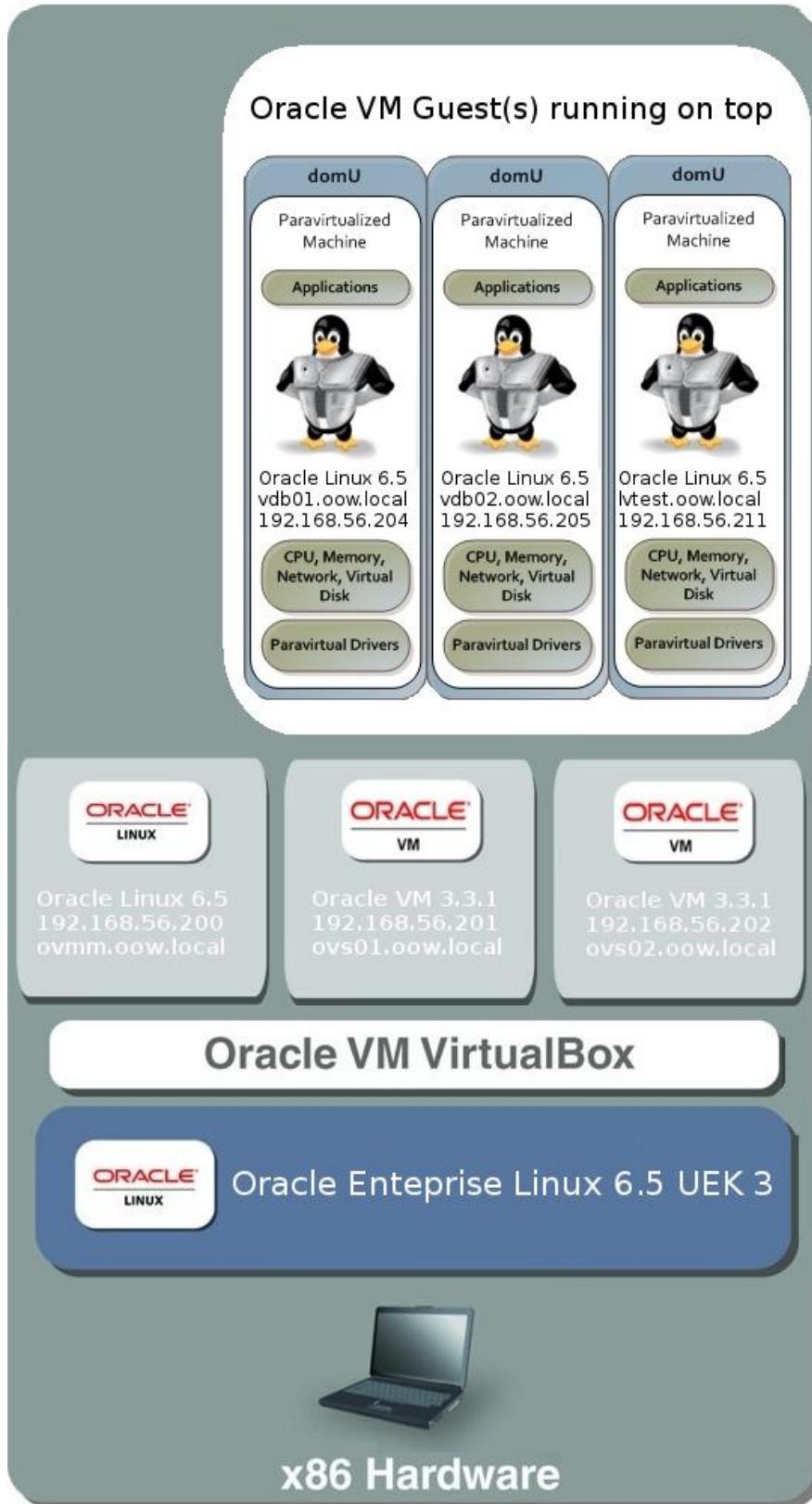
Internet = VirtualBox NAT network to allows guests to reach public and external sites

Intracluster = VirtualBox host-only Network used for intracluster on Oracle Clusterware

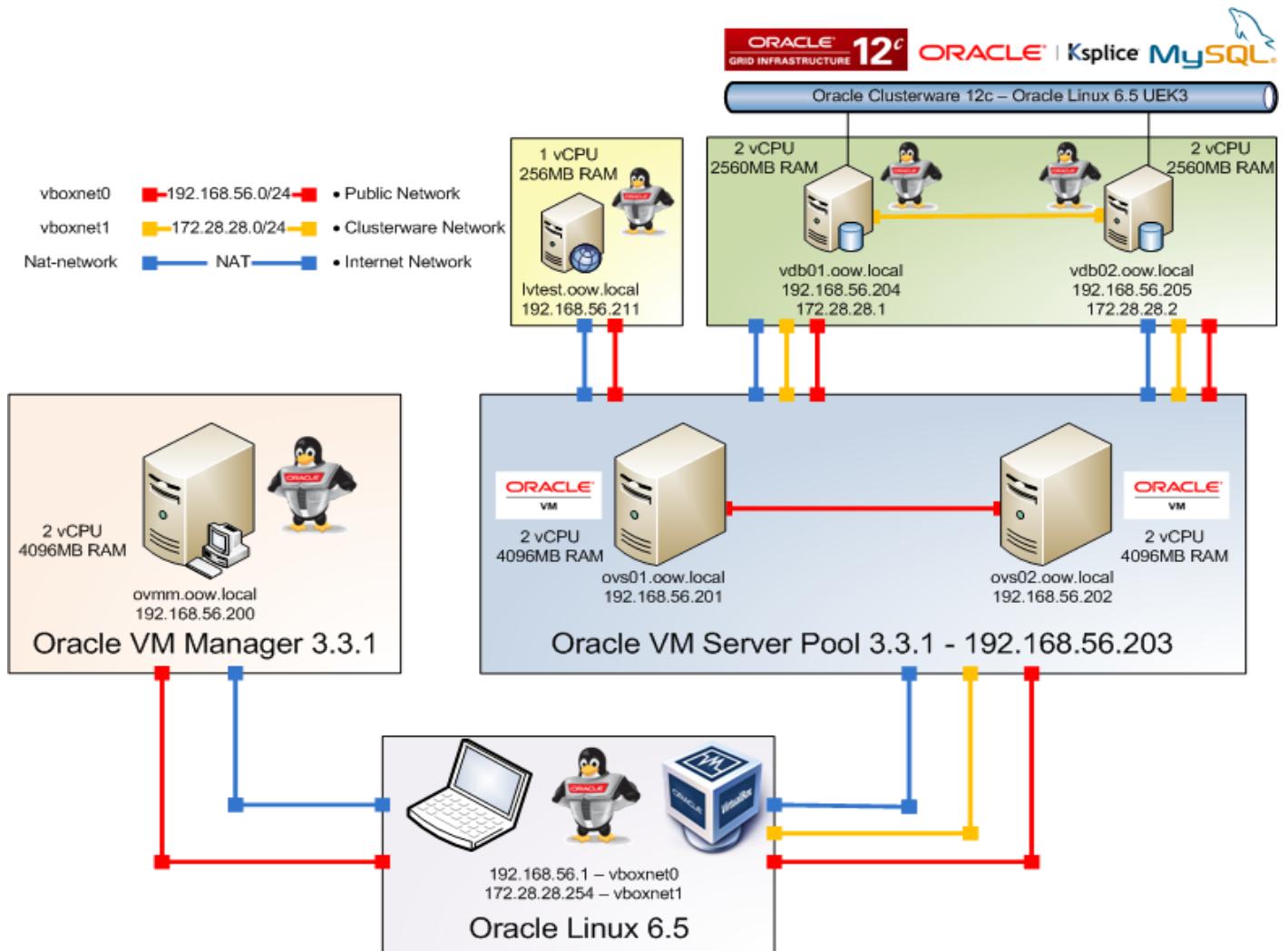
The screenshot shows the Oracle VM Manager interface with the Networking tab selected. The page displays a table of network interfaces and their connectivity status across various channels.

Name	ID	Intra-Network Server	Network Channels				
			Server Management	Cluster Heartbeat	Live Migrate	Storage	Virtual Machine
192.168.56.0	c0a83800		√	√	√		√
internet	103c790e25						√
intracluster	100a266cbe						√

The architecture built for this lab :



Detailed architecture of the Lab



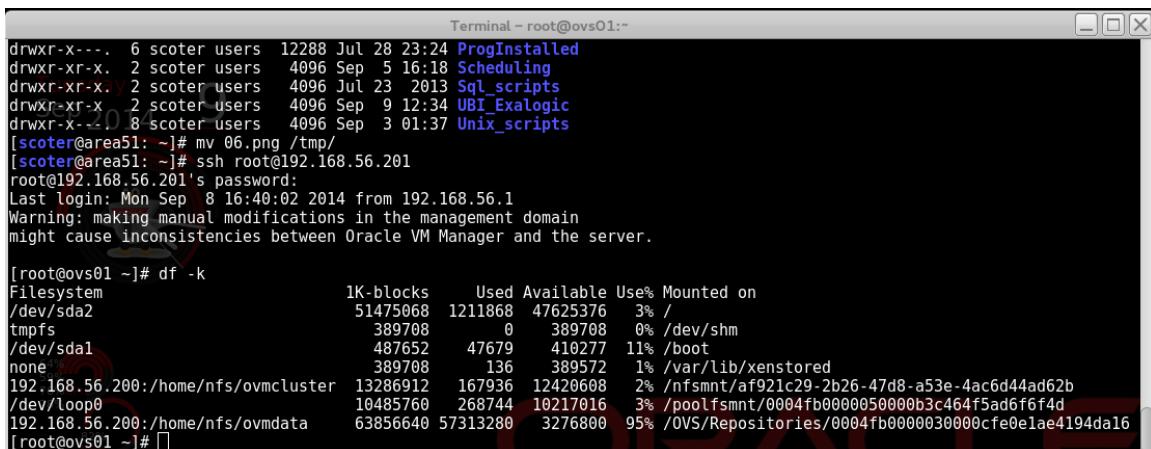
Connect to the Oracle VM Servers and verify that everything is ready to accommodate the lab

Connect by ssh to **ovs01.oow.local / ovs02.oow.local** and verify that Oracle VM guests are really running:

Open a terminal on your linux desktop and execute:

```
ssh root@192.168.56.201 ( password is ovsroot )
ssh root@192.168.56.202 ( password is ovsroot )
```

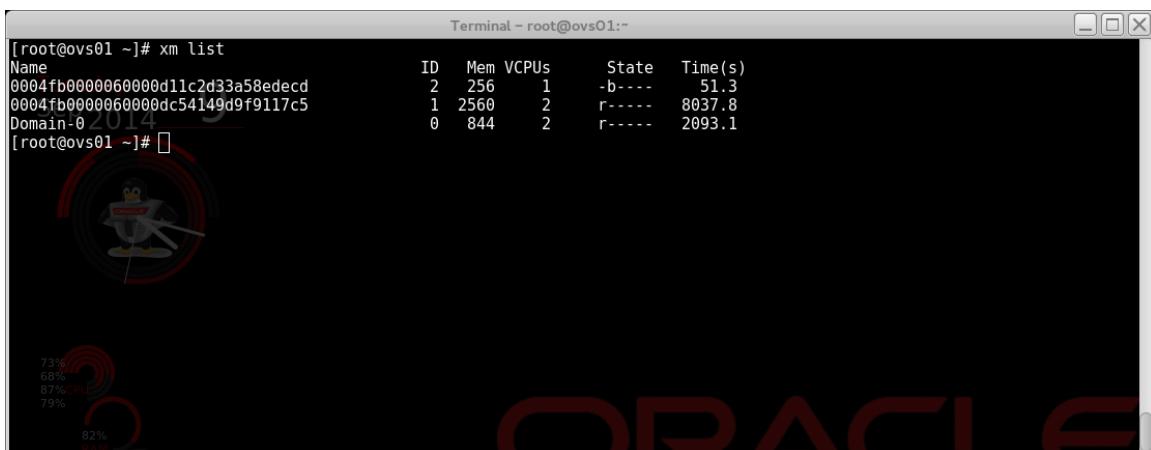
Once connected to both Oracle VM Servers, verify that Oracle VM repository is correctly mounted with the command “**df -k**”; the output should be the same as the figure below (verify on both Oracle VM servers):



```
Terminal - root@ovs01:~[scoter@area51: ~]# mv 06.png /tmp/
[scoter@area51: ~]# ssh root@192.168.56.201
root@192.168.56.201's password:
Last login: Mon Sep  8 16:40:02 2014 from 192.168.56.1
Warning: making manual modifications in the management domain
might cause inconsistencies between Oracle VM Manager and the server.

[root@ovs01 ~]# df -k
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda2              51475068  1211868   47625376  3% /
tmpfs                  389708       0   389708  0% /dev/shm
/dev/sdal                487652    47679   410277 11% /boot
none                   389708     136   389572  1% /var/lib/xenstored
192.168.56.200:/home/nfs/ovmcluster  13286912   167936  12420608  2% /nfsmnt/af921c29-2b26-47d8-a53e-4ac6d44ad62b
/dev/loop0               10485760  268744  10217016  3% /poolfsmnt/0004fb0000050000b3c464f5ad6f6f4d
192.168.56.200:/home/nfs/ovmdata    63856640  57313280   3276800 95% /OVS.Repositories/0004fb0000030000cf0e1ae4194da16
[root@ovs01 ~]#
```

Verify that Oracle VM guests are running by executing the command “**xm list**” (on both Oracle VM Servers). You will find two guests running on server ovs01 and one guest running on ovs02.



```
Terminal - root@ovs01:~[root@ovs01 ~]# xm list
Name                           ID  Mem VCPUs  State   Time(s)
0004fb0000060000d11c2d33a58edecd  2   256    1 -b----  51.3
0004fb0000060000dc54149d9f9117c5  1   2560   2 r----- 8037.8
Domain-0                         0   844    2 r----- 2093.1
[root@ovs01 ~]#
```

Connect to the Oracle VM guests and become familiar with Oracle Linux and Oracle Clusterware (Grid Infrastructure)

Connect by ssh to **vdb01.oow.local / vdb02.oow.local** and verify all Oracle Cluster managed resources are up and running:

Open a terminal on your linux desktop and execute:

```
ssh root@192.168.56.204 ( password is ovsroot )
ssh root@192.168.56.205 ( password is ovsroot )
```

Once connected to both Oracle VM guests, verify that Oracle Clusterware is up and running and, at the same time, that all resource managed by Oracle Clusterware are correctly working (execute the commands on both guests):

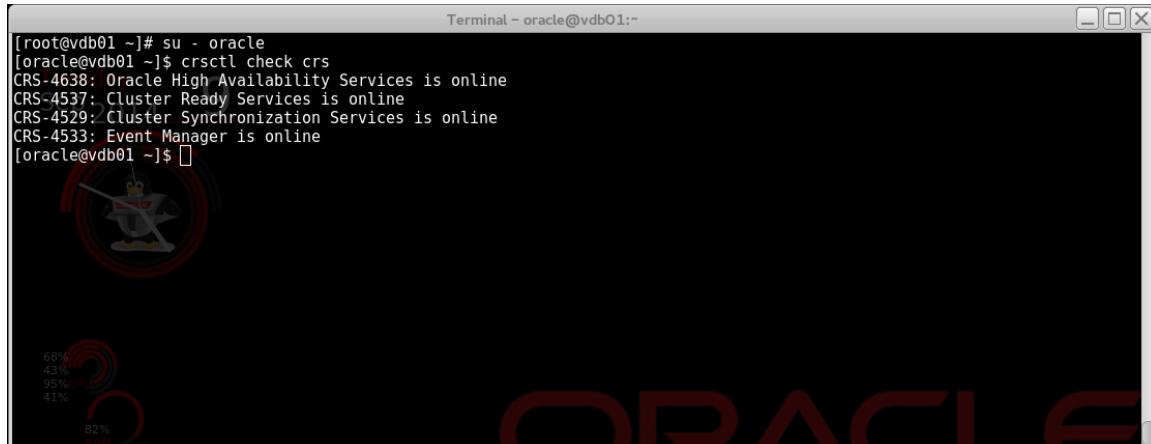
- 1) Switch user to Oracle Cluster owner (oracle):

```
# su - oracle
```

- 2) Verify that Oracle Clusterware services are up and running

```
# crsctl check crs
```

The output should be the same as shown in the figure below:

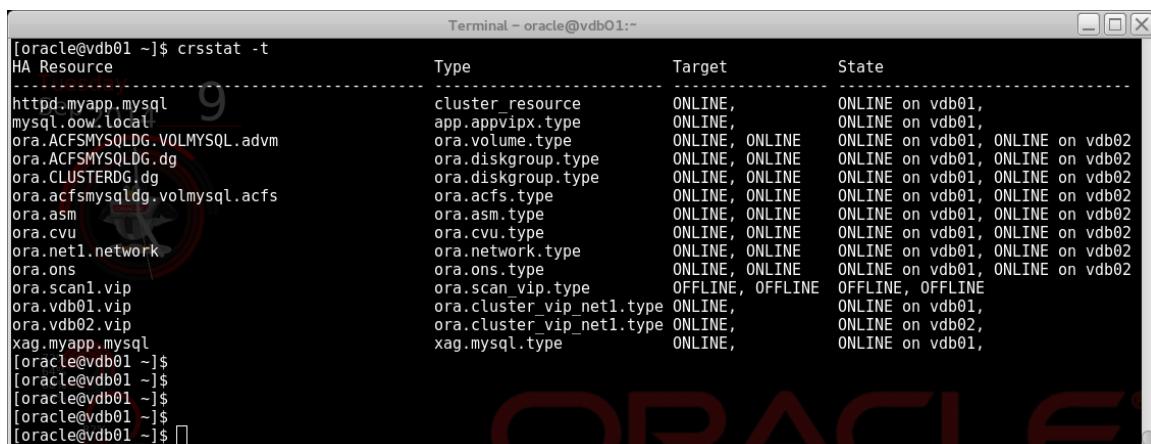


```
[root@vdb01 ~]# su - oracle
[oracle@vdb01 ~]$ crsctl check crs
CRS-4638: Oracle High Availability Services is online
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
[oracle@vdb01 ~]$
```

- 3) Verify that all Oracle Clusterware managed resource are correctly running

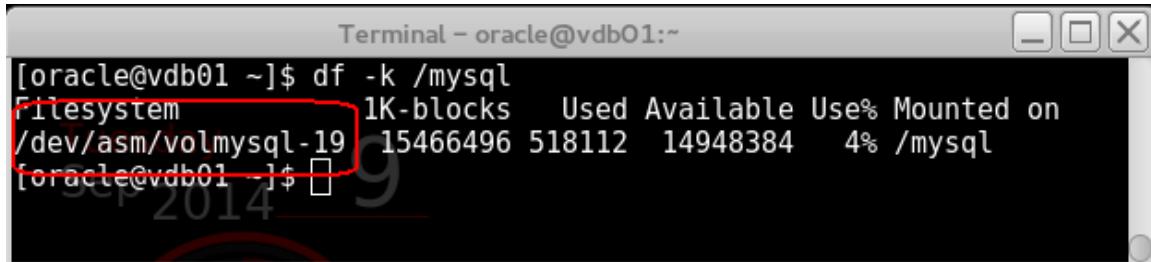
```
# crsctl status resources
```

You will find a script that wrap the output of the command above; execute "**crsstat -t**".



HA Resource	Type	Target	State
httpd.myapp.mysql	cluster_resource	ONLINE,	ONLINE on vdb01,
mysql.oow.local	app.appvipx.type	ONLINE,	ONLINE on vdb01,
ora.ACFSMySQLLDG.VOLMYSQL.advm	ora.volume.type	ONLINE, ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.ACFSMySQLLDG.dg	ora.diskgroup.type	ONLINE, ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.CLUSTERDG.dg	ora.diskgroup.type	ONLINE, ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.acfsmysqld.volmysql.acfs	ora.acfs.type	ONLINE, ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.asm	ora.asm.type	ONLINE, ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.cvu	ora.cvu.type	ONLINE, ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.net1.network	ora.network.type	ONLINE, ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.ons	ora.ons.type	ONLINE, ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.scan1.vip	ora.scan_vip.type	OFFLINE, OFFLINE	OFFLINE, OFFLINE
ora.vdb01.vip	ora.cluster_vip_net1.type	ONLINE,	ONLINE on vdb01,
ora.vdb02.vip	ora.cluster_vip_net1.type	ONLINE,	ONLINE on vdb02,
xag.myapp.mysql	xag.mysql.type	ONLINE,	ONLINE on vdb01,

- 4) Verify that ACFS cluster filesystem is mounted on both servers
df -k /mysql



A screenshot of a terminal window titled "Terminal - oracle@vdb01:~". The command "df -k /mysql" is run, and the output shows a single entry for the "/dev/asm/volmysql-19" device, which is mounted at the "/mysql" directory. The output includes columns for Filesystem, 1K-blocks, Used, Available, Use%, and Mounted on. A red box highlights the "Filesystem" column.

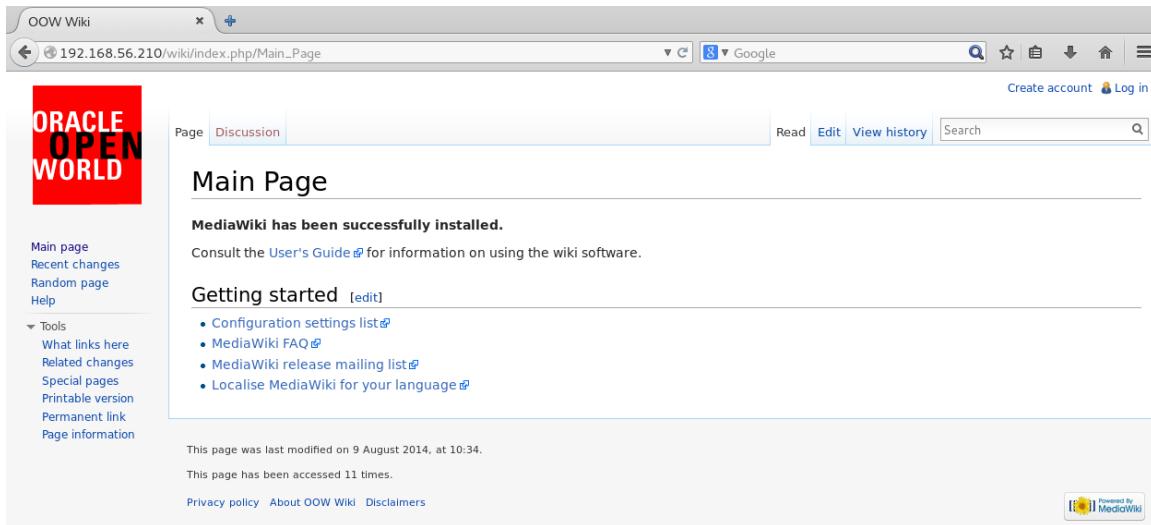
```
[oracle@vdb01 ~]$ df -k /mysql
Filesystem      1K-blocks   Used   Available Use% Mounted on
/dev/asm/volmysql-19  15466496 518112  14948384   4% /mysql
[oracle@vdb01 ~]$
```

- 5) Verify that clustered demo web application is running

On your laptop open a browser and connect to the following URL:

<http://192.168.56.210/wiki>

The web page should be similar to the picture below



High availability general concepts: understand what we are going to demonstrate

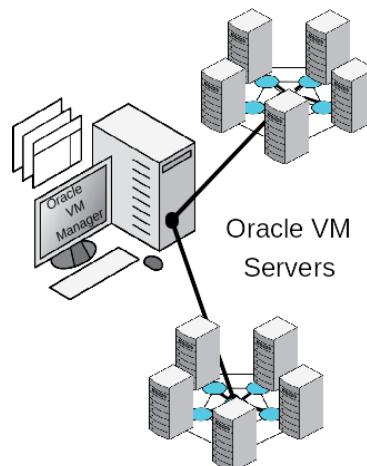
The architecture proposed in this lab is based on:

- Oracle VM 3.3.1
- Oracle Linux 6.x
- Oracle Ksplice
- Oracle Grid Infrastructure 12c (Clusterware)
- Oracle Grid Infrastructure Agents 5.1 (Clusterware agents to manage MySQL)
- Oracle MySQL Enterprise 5.6
- Demo web application

VM Oracle VM

Oracle VM is an enterprise-class server virtualization solution comprised of Oracle VM Server for x86, Oracle VM Server for SPARC and Oracle VM Manager. Oracle VM 3 reflects Oracle strategic commitment to deliver Application Driven Virtualization, the Virtualization that makes the entire enterprise software and hardware stack easier to deploy, manage, and support so IT and business can be more agile.

Oracle VM Server is based on the Xen hypervisor. Oracle VM Server can be managed using Oracle VM Manager, or as a standalone product with OpenStack.



Oracle Linux



Oracle Linux is an open-source operating system available under the GNU General Public License (GPLv2). Suitable for general purpose or Oracle workloads, it benefits from rigorous testing of more than 128,000 hours per

day with real-world workloads and includes unique innovations such as Ksplice for zero-downtime kernel patching, DTrace for real-time diagnostics, the powerful Btrfs file system, and more.

Oracle Ksplice |

Ksplice Utrack lets you apply 100% of the important kernel security updates released by your Linux vendor without rebooting.

Ksplice Utrack is available for Oracle Linux, free of charge, for Oracle Linux customers with a Premier support subscription.

Running Red Hat Enterprise Linux? Get a taste of one of the many features Oracle Linux Premier Support has to offer with our free 30-day Ksplice trial for RHEL systems. Give it a try and bring your Linux kernel up to date without rebooting (not even once to install it!)

Oracle Clusterware

Oracle Clusterware 12c Release 1 is the integrated foundation for Oracle Real Application Clusters (RAC) and the High Availability (HA) and resource management framework for all applications on any major platform. Oracle Clusterware 12c builds on the innovative technology introduced with Oracle Clusterware 11g by providing comprehensive multi-tiered HA and resource management for consolidated environments. The idea is to leverage Oracle Clusterware in the cloud to provide enterprise-class resiliency where required and dynamic, online allocation of compute resources where needed, when needed.

Oracle Clusterware 12c Release 1 is the integrated foundation for Oracle Real Application Clusters (RAC) and the High Availability (HA) and resource management framework for all applications on any major platform. Oracle Clusterware 12c builds on the innovative technology introduced with Oracle Clusterware 11g by providing comprehensive multi-tiered HA and resource management for consolidated environments. The idea is to leverage Oracle Clusterware in the cloud to provide enterprise-class resiliency where required and dynamic, online allocation of compute resources where needed, when needed.

Oracle MySQL Enterprise

MySQL Enterprise Edition includes the most comprehensive set of advanced features, management tools and technical support to achieve the highest levels of MySQL scalability, security, reliability, and uptime. It reduces the risk, cost, and complexity in developing, deploying, and managing business-critical MySQL applications.

All this components, from the virtualization layer (Oracle VM) to the managed application (Oracle MySQL) are supplied by Oracle and, at the same time, are included in Oracle Premier Support.

These components allows to build up a complete high-available architecture that is able to grant an high service-level with a reduced TCO; at the same time, these products are able to supply enterprise features such as:

- Virtualization Layer that allows to create guest servers on x86 platforms (Microsoft, Linux, Solaris)

- Virtualization layer features such as hot-clone, snapshots, live-migrate, distributed resource scheduling and dynamic power management
- Enterprise Linux Operating System and Unbreakable Linux Kernel
- Kernel updates without disruptive server reboots
- Active resource monitoring and high-availability Oracle Clusterware

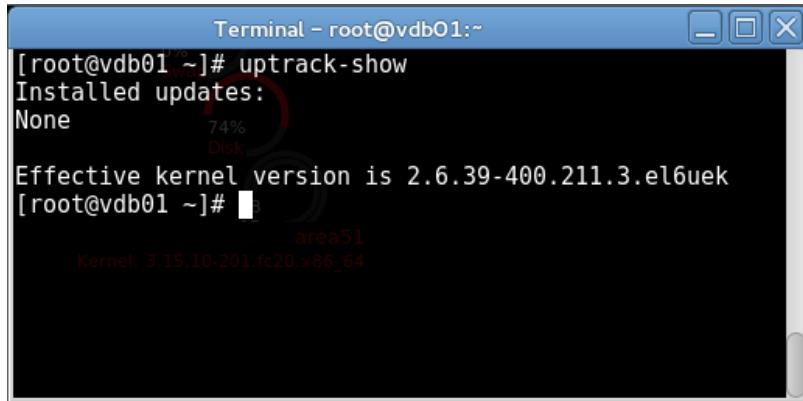
Demonstrate high-availability features covered by Oracle Ksplice

Oracle Ksplice lets you apply kernel-updates on your linux servers without reboot them.

On our demo environment we will proceed to update kernels on virtual servers named “**vdb01.oow.local**” and “**vdb02.oow.local**”.

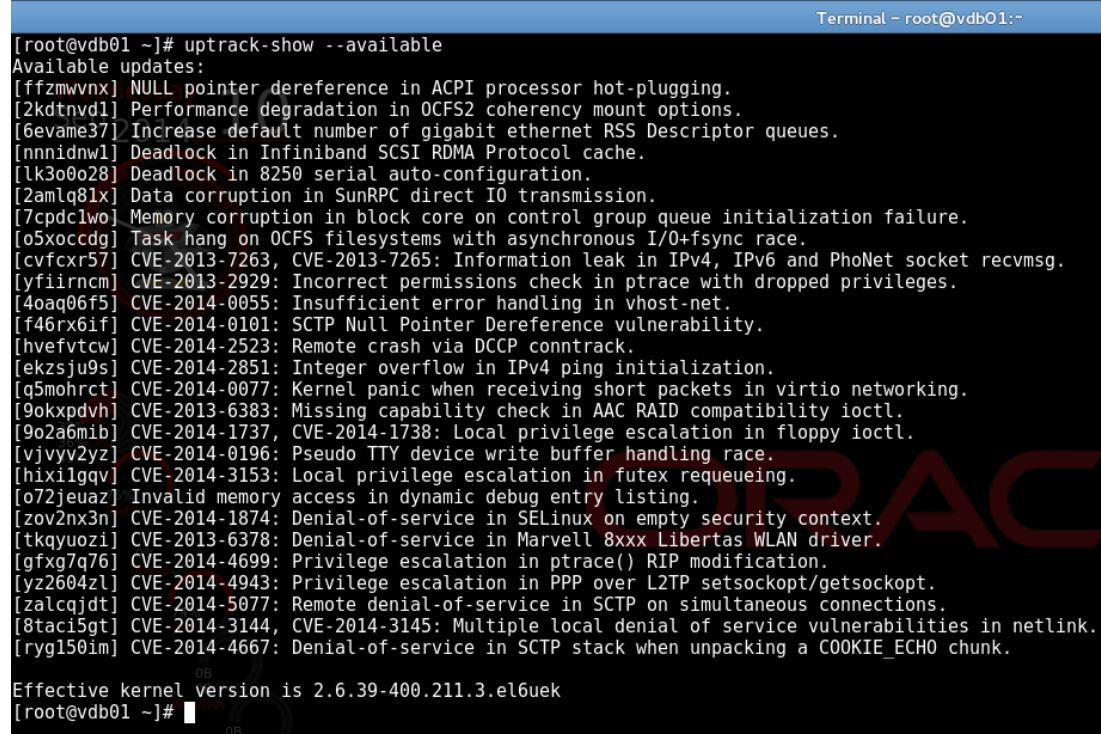
To update kernel on the servers specified above execute the following steps:

1. Connect to by ssh to the server “vdb01.oow.local”
`ssh root@192.168.56.204`
2. Show which Ksplice kernel updates are already installed by executing “**uptrack-show**”



```
[root@vdb01 ~]# uptrack-show
Installed updates:
None
Effective kernel version is 2.6.39-400.211.3.el6uek
[root@vdb01 ~]#
```

3. Show which Ksplice kernel updates are available to be installed “**uptrack-show --available**”



```
[root@vdb01 ~]# uptrack-show --available
Available updates:
[ffzwmvnx] NULL pointer dereference in ACPI processor hot-plugging.
[2kdtnvd1] Performance degradation in OCFS2 coherency mount options.
[6evame37] Increase default number of gigabit ethernet RSS Descriptor queues.
[nnnidnw1] Deadlock in Infiniband SCSI RDMA Protocol cache.
[lk3o0o28] Deadlock in 8250 serial auto-configuration.
[2am1q81x] Data corruption in SunRPC direct IO transmission.
[7cpdclwo] Memory corruption in block core on control group queue initialization failure.
[o5xoccgd] Task hang on OCFs filesystems with asynchronous I/O+fsync race.
[cvfcxr57] CVE-2013-7263, CVE-2013-7265: Information leak in IPv4, IPv6 and PhoNet socket recvmsg.
[yfiircm] CVE-2013-2929: Incorrect permissions check in ptrace with dropped privileges.
[4oaq06f5] CVE-2014-0055: Insufficient error handling in vhost-net.
[f46rx6if] CVE-2014-0101: SCTP Null Pointer Dereference vulnerability.
[hvefvtcw] CVE-2014-2523: Remote crash via DCCP conntrack.
[ekzsju9s] CVE-2014-2851: Integer overflow in IPv4 ping initialization.
[q5mohrct] CVE-2014-0077: Kernel panic when receiving short packets in virtio networking.
[90kxdvh] CVE-2013-6383: Missing capability check in AAC RAID compatibility ioctl.
[9o2a6mib] CVE-2014-1737, CVE-2014-1738: Local privilege escalation in floppy ioctl.
[vjvyv2yz] CVE-2014-0196: Pseudo TTY device write buffer handling race.
[hix1lqgv] CVE-2014-3153: Local privilege escalation in futex requeueing.
[o72jeuaz] Invalid memory access in dynamic debug entry listing.
[zov2nx3n] CVE-2014-1874: Denial-of-service in SELinux on empty security context.
[tkqyuozi] CVE-2013-6378: Denial-of-service in Marvell 8xxx Libertas WLAN driver.
[gfxq7q76] CVE-2014-4699: Privilege escalation in ptrace() RIP modification.
[yz2604zl] CVE-2014-4943: Privilege escalation in PPP over L2TP setsockopt/getsockopt.
[zalcqdjt] CVE-2014-5077: Remote denial-of-service in SCTP on simultaneous connections.
[8taci5gt] CVE-2014-3144, CVE-2014-3145: Multiple local denial of service vulnerabilities in netlink.
[ryg150im] CVE-2014-4667: Denial-of-service in SCTP stack when unpacking a COOKIE_ECHO chunk.

Effective kernel version is 2.6.39-400.211.3.el6uek
[root@vdb01 ~]#
```

4. Verify actual kernel version installed and actual kernel version in memory by executing “uname -a” and “uptrack-uname -a”.

```
[root@vdb01 ~]# uname -a
Linux vdb01.00w.local 2.6.39-400.211.3.el6uek.x86_64 #1 SMP Fri Dec 13 18:19:54 PST 2013 x86_64 x86_64 x86_64 GNU/Linux
[root@vdb01 ~]# uptrack-uname -a
Linux vdb01.00w.local 2.6.39-400.211.3.el6uek.x86_64 #1 SMP Fri Dec 13 18:19:54 PST 2013 x86_64 x86_64 x86_64 GNU/Linux
[root@vdb01 ~]# 
```

As you can see, actually, kernel installed and kernel-in-memory has the same version 2.6.39-400.211.3

5. Proceed to upgrade the kernel on your server by the command “**uptrack-upgrade -y**”.

```
[root@vdb01 ~]# uptrack-upgrade -y
The following steps will be taken:
Install [ffzmwvnx] NULL pointer dereference in ACPI processor hot-plugging.
Install [2kdtnvdl] Performance degradation in OCFS2 coherency mount options.
Install [6evame37] Increase default number of gigabit ethernet RSS Descriptor queues.
Install [nnnidnw1] Deadlock in Infiniband SCSI RDMA Protocol cache.
Install [lk3o0o28] Deadlock in serial auto-configuration.
Install [2amlq8lx] Data corruption in SunRPC direct I/O transmission.
Install [7cpdclwo] Memory corruption in block core on control group queue initialization failure.
Install [05xoccdg] Task hang on OCFS filesystems with asynchronous I/O+fsync race.
Install [cvfcxr57] CVE-2013-7263, CVE-2013-7265: Information leak in IPv4, IPv6 and PhoNet socket recvmsg.
Install [yfiiircm] CVE-2013-2929: Incorrect permissions check in ptrace with dropped privileges.
Install [4aoaq06f5] CVE-2014-0055: Insufficient error handling in vhost-net.
Install [f46rx6if] CVE-2014-0101: SCTP Null Pointer Dereference vulnerability.
Install [hvefvtcw] CVE-2014-2523: Remote crash via DCCP conntrack.
Install [ekzsju9s] CVE-2014-2851: Integer overflow in IPv4 ping initialization.
Install [q5mohrct] CVE-2014-0077: Kernel panic when receiving short packets in virtio networking.
Install [90kxpdvh] CVE-2013-6383: Missing capability check in AAC RAID compatibility ioctl.
Install [9o2a0mib] CVE-2014-1737, CVE-2014-1738: Local privilege escalation in floppy ioctl.
Install [jvvyv2yz] CVE-2014-0196: Pseudo TTY device write buffer handling race.
Install [hixilgav] CVE-2014-3153: Local privilege escalation in futex requeueing.
Install [o72jeuzal] Invalid memory access in dynamic debug entry listing.
Install [zov2nx3n] CVE-2014-1874: Denial-of-service in SELinux on empty security context.
Install [tkqyuozi] CVE-2013-6378: Denial-of-service in Marvell 8xxx Libertas WLAN driver.
Install [gfq07q76] CVE-2014-4699: Privilege escalation in ptrace() RIP modification.
Install [yz2604zl] CVE-2014-4943: Privilege escalation in PPP over L2TP setsockopt/getsockopt.
Install [zalcqjdt] CVE-2014-5077: Remote denial-of-service in SCTP on simultaneous connections.
Install [8taci5gt] CVE-2014-3144, CVE-2014-3145: Multiple local denial of service vulnerabilities in netlink.
Install [ryg150im] CVE-2014-4667: Denial-of-service in SCTP stack when unpacking a COOKIE_ECHO chunk.
Installing [ffzmwvnx] NULL pointer dereference in ACPI processor hot-plugging.
Installing [2kdtnvdl] Performance degradation in OCFS2 coherency mount options.
Installing [6evame37] Increase default number of gigabit ethernet RSS Descriptor queues.
.....
.....
Installing [yz2604zl] CVE-2014-4943: Privilege escalation in PPP over L2TP setsockopt/getsockopt.
Installing [zalcqjdt] CVE-2014-5077: Remote denial-of-service in SCTP on simultaneous connections.
Installing [8taci5gt] CVE-2014-3144, CVE-2014-3145: Multiple local denial of service vulnerabilities in netlink.
Installing [ryg150im] CVE-2014-4667: Denial-of-service in SCTP stack when unpacking a COOKIE_ECHO chunk.
Your kernel is fully up to date.
Effective kernel version is 2.6.39-400.215.7.el6uek
[root@vdb01 ~]# 
```

6. Verify different kernel version between installed and in-memory “uname -a” and “uptrack-uname -a”

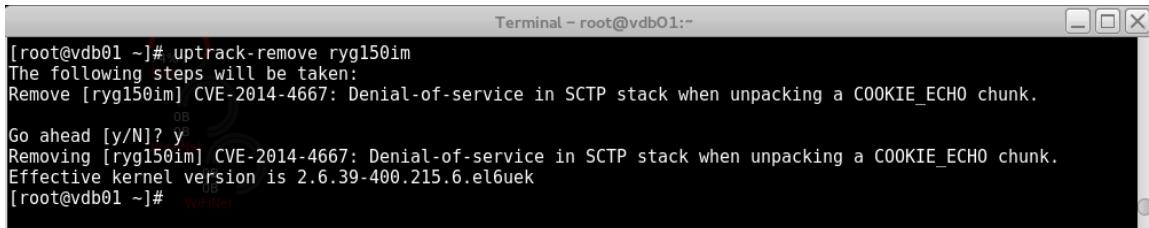
```
[root@vdb01 ~]# uname -a
Linux vdb01.00w.local 2.6.39-400.211.3.el6uek.x86_64 #1 SMP Fri Dec 13 18:19:54 PST 2013 x86_64 x86_64 x86_64 GNU/Linux
[root@vdb01 ~]# uptrack-uname -a
Linux vdb01.00w.local 2.6.39-400.215.7.el6uek.x86_64 #1 SMP Fri Aug 8 20:51:11 PDT 2014 x86_64 x86_64 x86_64 GNU/Linux
[root@vdb01 ~]# 
```

As you can see, actually:

- kernel installed is 2.6.39-400.211.3
- kernel actually in memory is 2.6.39-400.215.7

7. You can also evaluate to uninstall a single update as reported in the figure blow (confirm with Y).

Example: # **uptrack-remove ryg150im**



```
[root@vdb01 ~]# uptrack-remove ryg150im
The following steps will be taken:
Remove [ryg150im] CVE-2014-4667: Denial-of-service in SCTP stack when unpacking a COOKIE_ECHO chunk.

Go ahead [y/N]? y
Removing [ryg150im] CVE-2014-4667: Denial-of-service in SCTP stack when unpacking a COOKIE_ECHO chunk.
Effective kernel version is 2.6.39-400.215.6.el6uek
[root@vdb01 ~]#
```

As you can see now the kernel-version moved back to 2.6.39-400.215.6 (it was with a final .7)

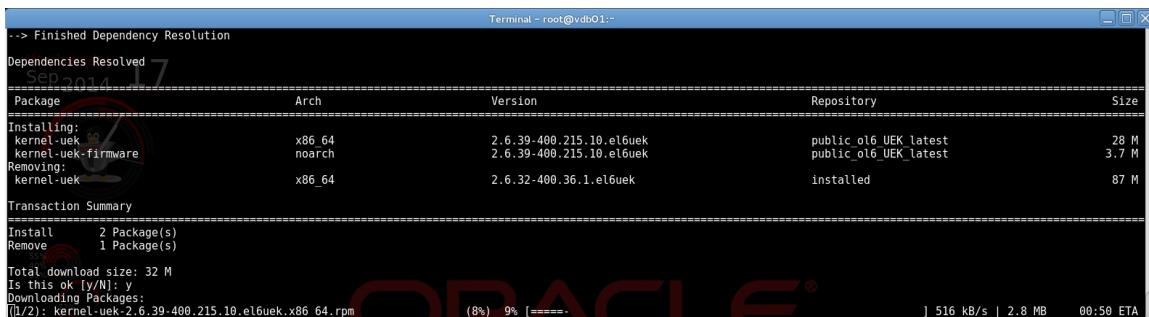
This part of the lab demonstrate how, on Oracle Linux, you can update the kernel without reboots; the feature supplied by Ksplice allows you also to install important security kernel fixes without any kind of impact on services supplied by your servers.

If you are going to reboot your server managed by Ksplice remember, also, to completely install the new kernel; this installation will allow you to have the new kernel in standard mode after the reboot (the alternative is that a Linux service named “uptrack” while booting will re-apply all kernel fixes in memory).

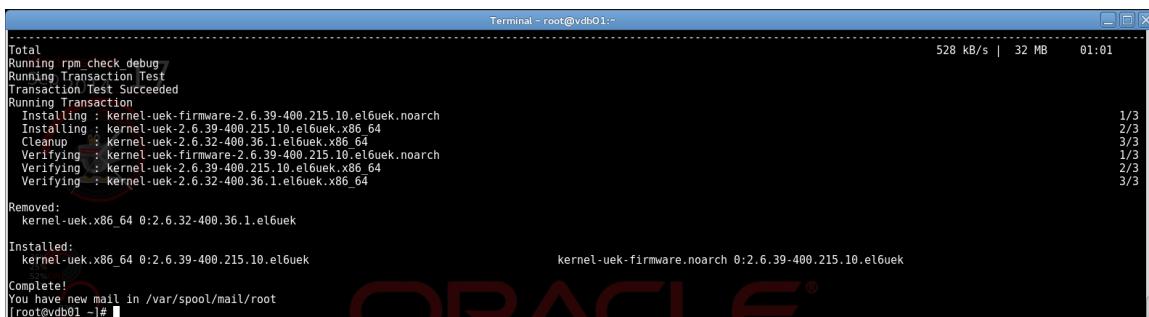
8. Following **Oracle Ksplice** best-practices we also need to physically install the newer kernel on our filesystem; this will allows, after a scheduled reboot, that the *machine restarts with the newer kernel without re-apply all the Ksplice updates*.

To update UEK Oracle Linux Kernel, execute the command:

yum update kernel-uek kernel-uek-firmware



```
--> Finished Dependency Resolution
Dependencies Resolved
Sep 2014 17
=====
Package           Arch      Version          Repository        Size
=====
Installing:
kernel-uek       x86_64   2.6.39-400.215.10.el6uek    public_ol6_uek_latest   28 M
kernel-uek-firmware x86_64   2.6.39-400.215.10.el6uek    public_ol6_uek_latest   3.7 M
Removing:
kernel-uek       x86_64   2.6.32-400.36.1.el6uek      installed            87 M
=====
Transaction Summary
=====
Install  2 Package(s)
Remove   1 Package(s)
Total download size: 32 M
Is this ok [y/N]: y
Downloading Packages:
[1/2]: kernel-uek-2.6.39-400.215.10.el6uek.x86_64.rpm (8%) 9% [=====] 516 kB/s | 2.8 MB 00:50 ETA
```



```
Terminal - root@vdb01:-
=====
Total
Running rpm_check_debug
Running transaction Test
Transaction Test succeeded
Running Transaction
  Installing : kernel-uek-firmware-2.6.39-400.215.10.el6uek.noarch
  Installing : kernel-uek-2.6.39-400.215.10.el6uek.x86_64
  Cleanup   : kernel-uek-2.6.32-400.36.1.el6uek.x86_64
  Verifying  : kernel-uek-firmware-2.6.39-400.215.10.el6uek.noarch
  Verifying  : kernel-uek-2.6.39-400.215.10.el6uek.x86_64
  Verifying  : kernel-uek-2.6.32-400.36.1.el6uek.x86_64
Removed:
  kernel-uek.x86_64 0:2.6.32-400.36.1.el6uek
Installed:
  kernel-uek.x86_64 0:2.6.39-400.215.10.el6uek
  kernel-uek-firmware.noarch 0:2.6.39-400.215.10.el6uek
=====
Complete!
You have new mail in /var/spool/mail/root
[root@vdb01 ~]#
```

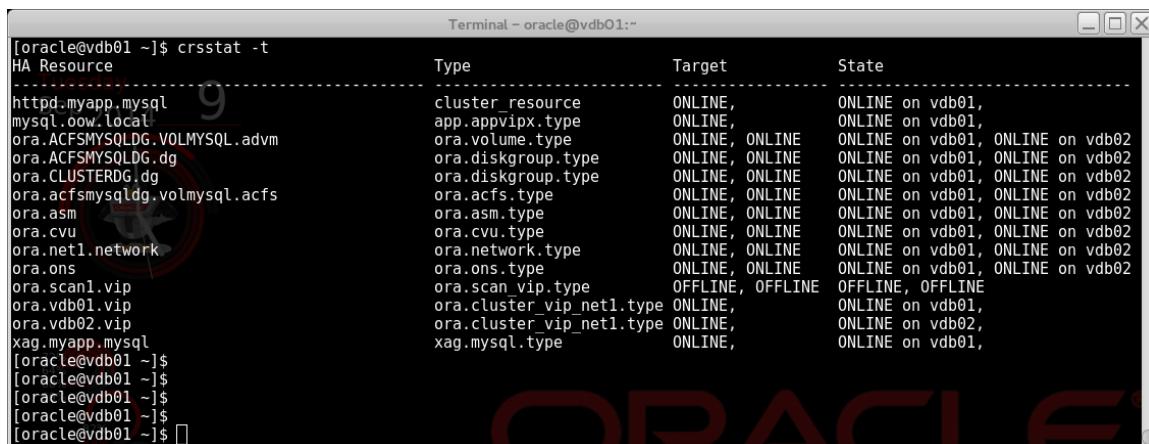
Demonstrate high-availability features covered by Oracle Clusterware

Oracle Clusterware is an enterprise clustering software included in Oracle Linux Premier Support.
With Oracle Clusterware you also obtain ACFS (**ASM Cluster file system**), a real posix compliant cluster filesystem.

ACFS allows to maintain software and application data on a cluster filesystem mounted, at the same time, on all servers that take part in Oracle Clusterware.

On our demo environment we will proceed to simulate a managed switchover of the resources (such as user-vip – virtual address and mysql database), one unmanaged failover of the resources and a resource-restart on the same node (all base activities covered by a clustering software solution).

1. Verify actual status of the resources on server “vdb01.oow.local”
With user “oracle” execute the command wrapped script “crsstat -t”:



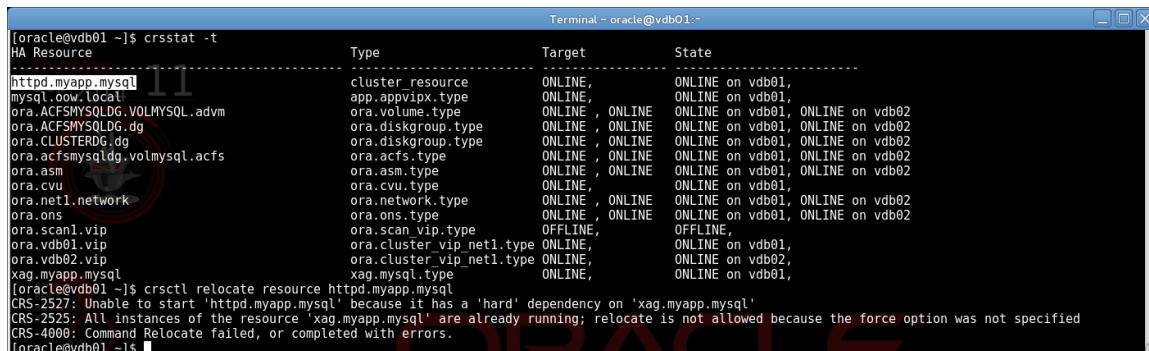
The terminal window shows the output of the crsstat -t command. It lists various HA Resources along with their Type, Target, and State. Most resources are ONLINE on vdb01, except for ora.scan1.vip which is OFFLINE, ora.cluster_vip_net1.type which is ONLINE on vdb01, ora.cluster_vip_net1.type which is ONLINE on vdb02, and xag.mysql.type which is ONLINE on vdb01.

HA Resource	Type	Target	State
httpd.myapp.mysql	cluster_resource	ONLINE,	ONLINE on vdb01,
mysql.oow.local	app.appvipx.type	ONLINE,	ONLINE on vdb01,
ora.ACFSMySQLDG.VOLMYSQL.advm	ora.volume.type	ONLINE , ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.ACFSMySQLDG.dg	ora.diskgroup.type	ONLINE , ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.CLUSTERDG.dg	ora.diskgroup.type	ONLINE, ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.acfsmysqld.volmysql.acfs	ora.acfs.type	ONLINE, ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.asm	ora.asm.type	ONLINE, ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.cvu	ora.cvu.type	ONLINE, ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.net1.network	ora.network.type	ONLINE, ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.ons	ora.ons.type	ONLINE, ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.scan1.vip	ora.scan_vip.type	OFFLINE,	OFFLINE, OFFLINE
ora.vdb01.vip	ora.cluster_vip_net1.type	ONLINE,	ONLINE on vdb01,
ora.vdb02.vip	ora.cluster_vip_net1.type	ONLINE,	ONLINE on vdb02,
xag.myapp.mysql	xag.mysql.type	ONLINE,	ONLINE on vdb01,

As shown in the figure above all resources (except “ora.scan1.vip” must be ONLINE) and our service resources are all active on node “vdb01”.

2. Try to relocate all service on the other node by executing, as user oracle:

```
# crsctl relocate resource httpd.myapp.mysql
```



The terminal window shows the crsctl relocate command being run. It attempts to relocate the httpd.myapp.mysql resource to vdb02. However, it fails because there is a hard dependency on xag.mysql.mysql, which is already running on vdb01. The message indicates that relocate is not allowed without specifying the force option.

```
[oracle@vdb01 ~]$ crsctl relocate resource httpd.myapp.mysql
CRS-2527: Unable to start 'httpd.myapp.mysql' because it has a 'hard' dependency on 'xag.mysql.mysql'
CRS-2525: All instances of the resource 'xag.mysql.mysql' are already running; relocate is not allowed
because the force option was not specified
CRS-4000: Command Relocate failed, or completed with errors.
[oracle@vdb01 ~]$
```

The message obtained is:

CRS-2527: Unable to start 'httpd.myapp.mysql' because it has a 'hard' dependency on 'xag.mysql.mysql'

CRS-2525: All instances of the resource 'xag.mysql.mysql' are already running; relocate is not allowed because the force option was not specified

CRS-4000: Command Relocate failed, or completed with errors.

This is due to the fact that Oracle Clusterware managed resources have dependencies each other; so, a “force” option is needed to complete the operation.

Service resources have the following dependencies:

Resource Name	Description	Dependency
mysql.oow.local (192.168.56.210)	User Virtual IP address	Network, ACFS filesystem
xag.myapp.mysql	MySQL database named myapp	mysql.oow.local (user-vip)
httpd.myapp.mysql	Apache server	xag.myapp.mysql (MySQL db)

3. Relocate services and verify that web-application continues working (a little seconds outage is expected)

```
[oracle@vdb01 ~]$ crsctl relocate resource httpd.myapp.mysql -f
CRS-2673: Attempting to stop 'httpd.myapp.mysql' on 'vdb01'
CRS-2677: Stop of 'httpd.myapp.mysql' on 'vdb01' succeeded
CRS-2673: Attempting to stop 'xag.myapp.mysql' on 'vdb01'
CRS-2677: Stop of 'xag.myapp.mysql' on 'vdb01' succeeded
CRS-2673: Attempting to stop 'mysql.oow.local' on 'vdb01'
CRS-2677: Stop of 'mysql.oow.local' on 'vdb01' succeeded
CRS-2672: Attempting to start 'mysql.oow.local' on 'vdb02'
CRS-2676: Start of 'mysql.oow.local' on 'vdb02' succeeded
CRS-2672: Attempting to start 'xag.myapp.mysql' on 'vdb02'
CRS-2676: Start of 'xag.myapp.mysql' on 'vdb02' succeeded
CRS-2672: Attempting to start 'httpd.myapp.mysql' on 'vdb02'
CRS-2676: Start of 'httpd.myapp.mysql' on 'vdb02' succeeded
[oracle@vdb01 ~]$ 
```

Verify application URL is reachable (you can open from your laptop browser): <http://192.168.56.210/wiki>

The screenshot shows a web browser window with the following details:

- Address Bar:** Shows the URL 192.168.56.210/wiki/index.php/Main_Page.
- Page Header:** Includes the Oracle Open World logo, navigation links like 'Main page', 'Recent changes', 'Random page', 'Help', and 'Tools' (with sub-links 'What links here', 'Related changes', 'Special pages', 'Printable version', 'Permanent link', and 'Page information').
- Page Content:**
 - The main heading is 'Main Page'.
 - A success message states: 'MediaWiki has been successfully installed.'
 - An 'Getting started' section lists links to 'Configuration settings list', 'MediaWiki FAQ', 'MediaWiki release mailing list', and 'Localise MediaWiki for your language'.
 - Footnotes at the bottom mention the page was last modified on 9 August 2014, at 10:34, and has been accessed 12 times.
 - Page footer includes links to 'Privacy policy', 'About OOW Wiki', and 'Disclaimers'.
 - A 'Powered by MediaWiki' logo is in the bottom right corner.

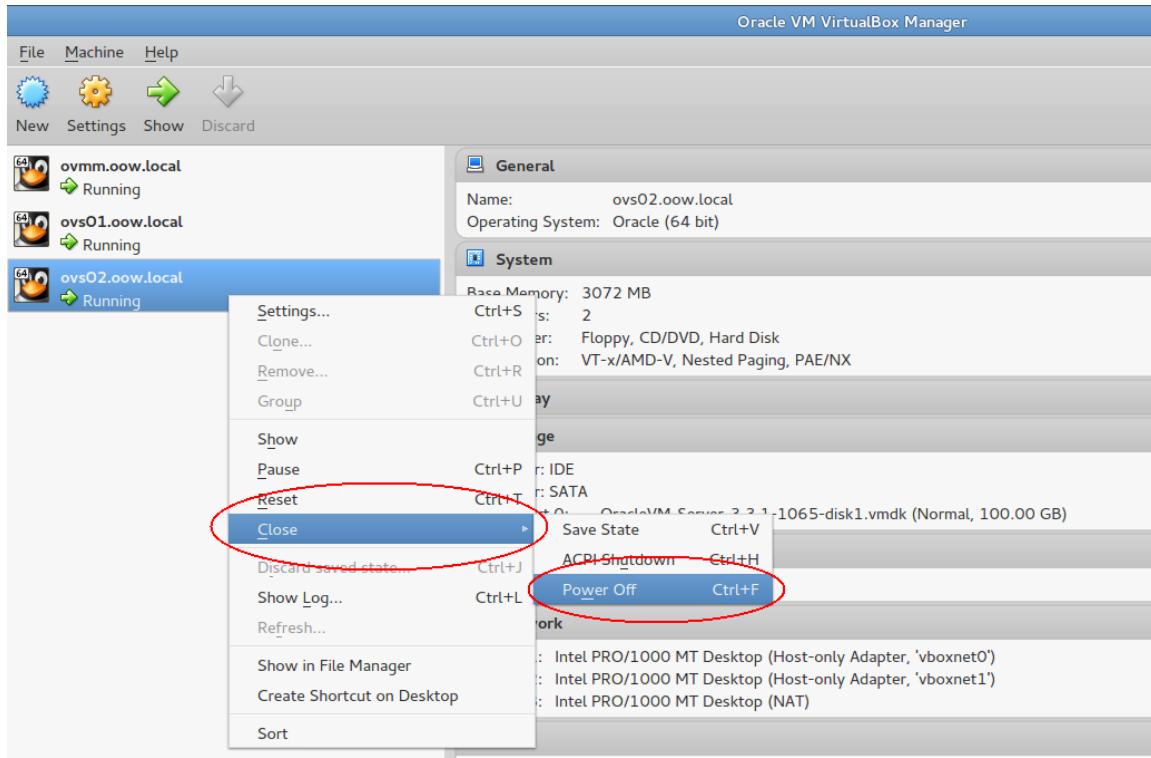
4. Verify actual resource distribution on the cluster with the command “crsstat -t”

nb: remember that "crsstat" is a wrapped script that executes "crsctl status resources -t"

```
[oracle@vdb01 ~]$ crsstat -t
[...]
HA Resource           Type          Target        State
[...]
httpd.mysql           cluster_resource ONLINE,      ONLINE on vdb02,
mysql.own.local       app.appvipx.type  ONLINE,      ONLINE on vdb02,
ora.ACFSMySQLDG.VOLMYSQL.advm ora.volume.type   ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.ACFSMySQLDG.dg     ora.diskgroup.type ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.CLUSTERDG.dg      ora.diskgroup.type ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.actmysqld.volmysql.acfs ora.acfs.type    ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.asm               ora.asm.type    ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.csv               ora.csv.type    ONLINE,      ONLINE on vdb01,
ora.net1.network      ora.network.type ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.ons               ora.ons.type    ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.scan1.vip          ora.scan.vip.type OFFLINE,     OFFLINE,
ora.vdb01.vip          ora.cluster_vip_net1.type ONLINE,      ONLINE on vdb01,
ora.vdb02.vip          ora.cluster_vip_net1.type ONLINE,      ONLINE on vdb02,
xag.myapp.mysql        xag.mysql.type  ONLINE,      ONLINE on vdb02,
[oracle@vdb01 ~]$ ]
```

As you can see service resources are now active on node "**vdb02.00w.local**".

- Shutdown (simulating a plug remove) Virtual-Box server named "ovs02.00w.local" that is the server that actually host "vdb02.00w.local" (the active node in the cluster).
To power-off the server, open "**VirtualBox console**", right-click on the server "**ovs02.00w.local**", choose "**Close**" and "**Poweroff**".



Confirm to brutally shutdown the server "**ovs02.00w.local**"



6. Verify actual resource distribution on the cluster with the command "crsstat -t" and, at the same time, with the laptop browser (Mozilla Firefox) verify that web-demo application is available.
 You'll see that, actually, only one node of the cluster is active and that all service resources are active on node "vdb01.ooow.local".

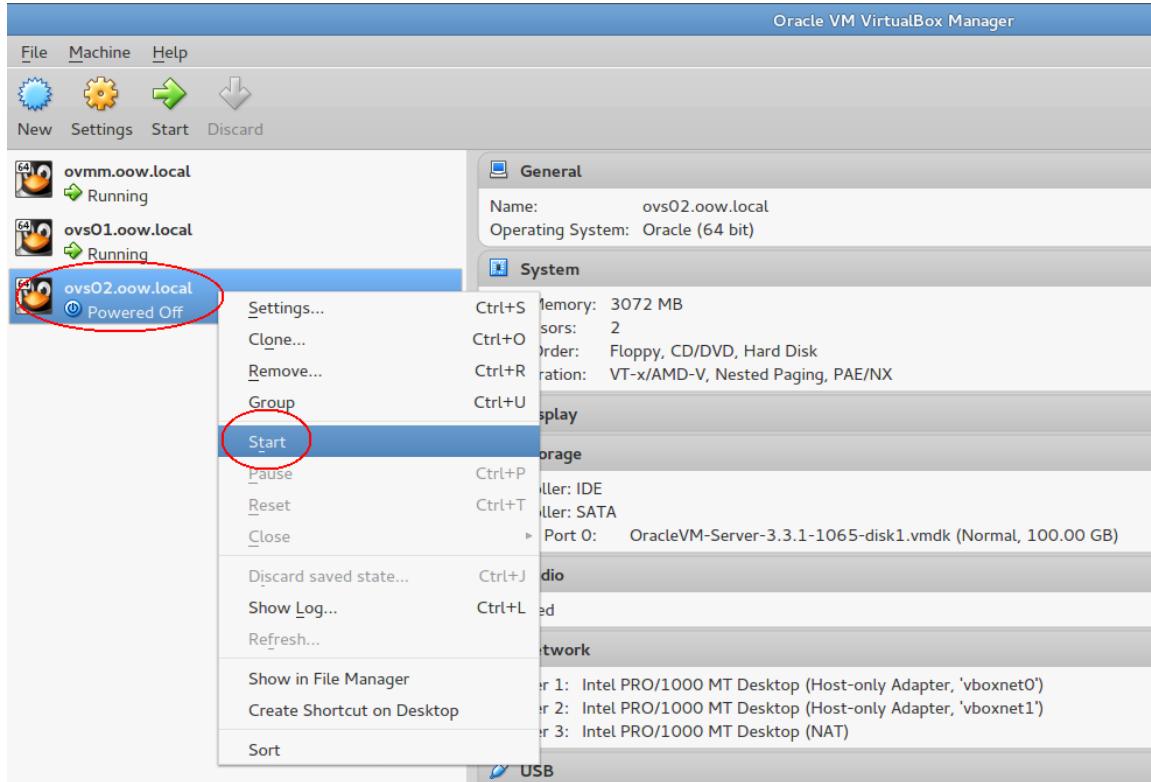
HA Resource	Type	Target	State
httpd.myapp.mysql	cluster_resource	ONLINE,	ONLINE on vdb01,
mysql.ooow.local	app.appvipx.type	ONLINE,	ONLINE on vdb01,
ora.ACFSMySQLDG.VOLMYSQLDG.advm	ora.volume.type	ONLINE,	ONLINE on vdb01,
ora.ACFSMySQLDG.dg	ora.diskgroup.type	ONLINE,	ONLINE on vdb01,
ora.CLUSTERDG.dg	ora.diskgroup.type	ONLINE,	ONLINE on vdb01,
ora.acfsmysqldg.volmysql.acfs	ora.acfs.type	ONLINE,	ONLINE on vdb01,
ora.asm	ora.asm.type	ONLINE,	ONLINE on vdb01,
ora.cvu	ora.cvu.type	ONLINE,	ONLINE on vdb01,
ora.net1.network	ora.network.type	ONLINE,	ONLINE on vdb01,
ora.ons	ora.ons.type	ONLINE,	ONLINE on vdb01,
ora.scan1.vip	ora.scan_vip.type	OFFLINE,	OFFLINE,
ora.vdb01.vip	ora.cluster_vip_net1.type	OFFLINE,	ONLINE on vdb01,
ora.vdb02.vip	ora.cluster_vip_net1.type	ONLINE,	INTERMEDIATE on vdb01,
xag.myapp.mysql	xag.mysql.type	ONLINE,	ONLINE on vdb01,

Verify application URL is reachable (you can open from your laptop browser): <http://192.168.56.210/wiki>

The screenshot shows a Mozilla Firefox browser window with the following details:

- Title Bar:** OOW Wiki
- Address Bar:** 192.168.56.210/wiki/index.php/Main_Page
- User Bar:** Create account Log in
- Page Content:**
 - Logo:** ORACLE OPEN WORLD
 - Title:** Main Page
 - Message:** MediaWiki has been successfully installed.
 - Text:** Consult the User's Guide for information on using the wiki software.
 - Section:** Getting started [edit]
 - Links:**
 - Configuration settings list
 - MediaWiki FAQ
 - MediaWiki release mailing list
 - Localise MediaWiki for your language
 - Page Footer:**
 - This page was last modified on 9 August 2014, at 10:34.
 - This page has been accessed 11 times.
 - Privacy policy About OOW Wiki Disclaimers
 - Powered by MediaWiki

7. Restart VirtualBox server named "ovs02.oow.local"
 Open **VirtualBox console**, right-click on "**ovs02.oow.local**" and the select "**Start**"



8. Connect to the "**Oracle VM Manager**" console, go to the "**Servers and VMs**" tab, expand and select "**ovmpool.oow.local**", select it, change perspective view to "**Virtual Machines**" and verify the status of Oracle VM pool and Server; everything should be as in the picture below.
 You need to see also that virtual-guests **vdb01**, **vdb02** and **lvtest** are active and running.

The screenshot shows the Oracle VM Manager web interface. At the top, the URL is https://192.168.56.200:7002/ovm/console/faces/resource/resourceView.jspx. The main navigation bar includes Health, Servers and VMs (which is selected and highlighted in blue), Repositories, Networking, Storage, Tools and Resources, and Jobs. Below this, there are two panes. The left pane shows a tree structure with 'Server Pools' expanded, showing 'ovmpool.oow.local' which contains 'ovs01.oow.local' and 'ovs02.oow.local'. It also lists 'Unassigned Servers' and 'Unassigned Virtual Machines'. The right pane shows a table titled 'Virtual Machines' with the following data:

Name	Status	Tag(s)	Event Severity	Server	Max. Memory (MB)	Memory (MB)
lvtest.oow.local	Running		Informational	ovs01.oow.local	256	256
vdb01.oow.local	Running		Informational	ovs01.oow.local	2560	2560
vdb02.oow.local	Running		Informational	ovs02.oow.local	2048	2048

9. Connect to the virtual-guest "**vdb01.oow.local**" (192.168.56.204) by ssh and verify, by clusterware commands, if the "**vdb02.oow.local**" re-joined the cluster.

```

[root@vdb01 ~]# crsstat -t
HA Resource          Type        Target      State
httpd.mysql          cluster_resource ONLINE,    ONLINE on vdb01,
mysql.own.local      app.appvipx.type  ONLINE,    ONLINE on vdb01,
ora.ACFSMySQLDG.volmySQL.advm ora.volume.type   ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.ACFSMySQLDG.dg    ora.diskgroup.type ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.CLUSTERDG.dg     ora.diskgroup.type ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.acfsmysqld.volmysql.acfs ora.acfs.type    ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.asm               ora.asm.type     ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.cvu               ora.cvu.type     ONLINE,    ONLINE on vdb01,
ora.net1.network      ora.network.type ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.ons               ora.ons.type     ONLINE,    ONLINE on vdb01, ONLINE on vdb02
ora.scan1.vip          ora.scan_vip.type OFFLINE,   OFFLINE,
ora.vdb01.vip          ora.cluster_vip_net1.type ONLINE,   ONLINE on vdb01,
ora.vdb02.vip          ora.cluster_vip_net1.type ONLINE,   ONLINE on vdb02,
xag.mysql.mysql       ora.mysql.type   ONLINE,    ONLINE on vdb01,
[root@vdb01 ~]#

```

Wait for “vdb02.own.local” join the cluster before proceede with the steps below.

10. Connect, as **root**, to the node where our managed resources are active (it should be **vdb01.own.local**) and verify which are processes of our “httpd” daemon (process number and hour/date) with the command:
ps -edaf |grep http

```

[root@vdb01 ~]# ps -edaf |grep http
root    24648  1  1 02:31 ?        00:00:00 /usr/sbin/httpd
apache  24655 24648  0 02:31 ?        00:00:00 /usr/sbin/httpd
apache  24656 24648  0 02:31 ?        00:00:00 /usr/sbin/httpd
apache  24657 24648  0 02:31 ?        00:00:00 /usr/sbin/httpd
apache  24658 24648  0 02:31 ?        00:00:00 /usr/sbin/httpd
apache  24659 24648  0 02:31 ?        00:00:00 /usr/sbin/httpd
apache  24660 24648  0 02:31 ?        00:00:00 /usr/sbin/httpd
apache  24661 24648  0 02:31 ?        00:00:00 /usr/sbin/httpd
apache  24662 24648  0 02:31 ?        00:00:00 /usr/sbin/httpd
root    24752 17220  0 02:31 pts/0   00:00:00 grep http
[root@vdb01 ~]#

```

11. Kill all httpd daemon processes, verify that no “httpd” processes are active and wait for the clusterware intervention.

Oracle Clusterware should take care in some seconds of this faulty situation.
The sequence of commands to execute, as root, is:

```

# killall httpd
# ps -edaf |grep http ==> waiting for Oracle Clusterware restart the daemon

```

Initially all httpd processes will die and, after some seconds, Oracle Clusterware will arrange for restart of the daemon; after that verify that web-demo application is available at the following URL:

<http://192.168.56.210/wiki>

The screenshot shows a terminal window titled "Terminal - root@vdb01:~". The user runs the command "killall httpd" followed by several "ps -edaf |grep http" commands. The output shows multiple httpd processes being killed, replaced by grep processes. A red oval highlights the first two grep processes, and a larger red circle highlights the entire sequence of grep processes.

```
[root@vdb01 ~]# killall httpd
[root@vdb01 ~]# ps -edaf |grep http
root 29789 17220 0 02:46 pts/0
[root@vdb01 ~]# ps -edaf |grep http
root 29850 17220 0 02:47 pts/0
[root@vdb01 ~]# ps -edaf |grep http
root 29871 17220 0 02:47 pts/0
[root@vdb01 ~]# ps -edaf |grep http
root 29935 1 5 02:47 ?
apache 29943 29935 0 02:47 ?
apache 29945 29935 0 02:47 ?
apache 29946 29935 0 02:47 ?
apache 29947 29935 0 02:47 ?
apache 29948 29935 0 02:47 ?
apache 29949 29935 0 02:47 ?
apache 29950 29935 0 02:47 ?
apache 29951 29935 0 02:47 ?
root 29953 17220 0 02:47 pts/0
[root@vdb01 ~]# 
```

12. With these step we will simulate a file system corruption with the loss of the binary "httpd"; Oracle Clusterware will be able to identify that it's not possible to restart the daemon on the current node and will proceed to a complete fail-over of the service (migrating all managed services from node "vdb01" to "vdb02").

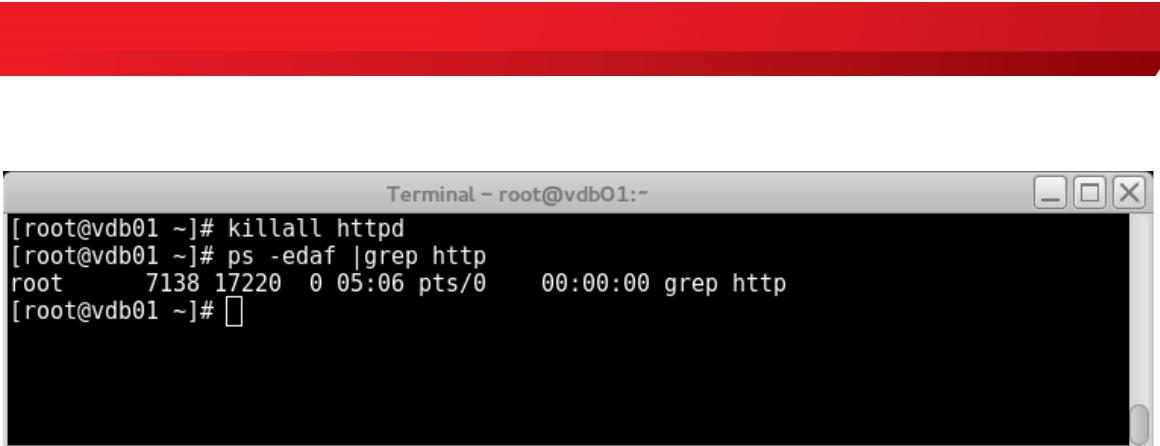
The steps to execute, as **root**, on node "**vdb01**" (node actually owning all resources) are:

Rename httpd binary on the filesystem
`# mv /usr/sbin/httpd /usr/sbin/httpd.corrupted`

The screenshot shows a terminal window titled "Terminal - root@vdb01:~". The user runs the command "file /usr/sbin/httpd" to check the file type, which is shown as an ELF 64-bit LSB shared object. They then run the command "mv /usr/sbin/httpd /usr/sbin/httpd.corrupted" to rename the file. The terminal shows the original file information and the successful renaming.

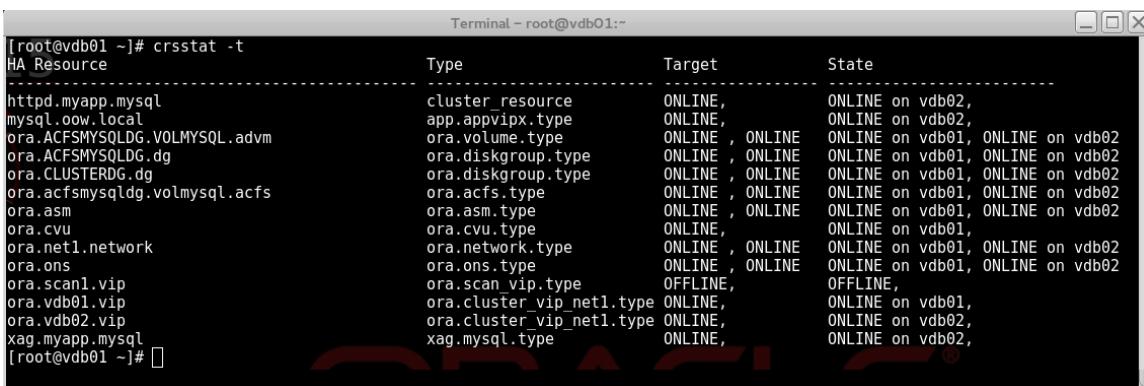
```
[root@vdb01 ~]# file /usr/sbin/httpd
/usr/sbin/httpd: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.18, stripped
[root@vdb01 ~]# mv /usr/sbin/httpd /usr/sbin/httpd.corrupted
[root@vdb01 ~]# 
```

Kill all httpd processes
`# killall httpd`



```
Terminal - root@vdb01:~  
[root@vdb01 ~]# killall httpd  
[root@vdb01 ~]# ps -edaf |grep http  
root      7138 17220  0 05:06 pts/0    00:00:00 grep http  
[root@vdb01 ~]#
```

Verify managed service status by the wrapped script (wait until everything is running on node “**vdb02**”):
crsstat -t



HA Resource	Type	Target	State
httpd.myapp.mysql	cluster_resource	ONLINE,	ONLINE on vdb02,
mysql.own.local	app.appvipx.type	ONLINE,	ONLINE on vdb02,
ora.ACFSMySQLDG.VOLMYSQL.advm	ora.volume.type	ONLINE , ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.ACFSMySQLDG.dg	ora.diskgroup.type	ONLINE , ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.CLUSTERDG.dg	ora.diskgroup.type	ONLINE , ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.acfsmysqld.volmysql.acfs	ora.acfs.type	ONLINE , ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.asm	ora.asm.type	ONLINE , ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.cvu	ora.cvu.type	ONLINE,	ONLINE on vdb01,
ora.net1.network	ora.network.type	ONLINE , ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.ons	ora.ons.type	ONLINE , ONLINE	ONLINE on vdb01, ONLINE on vdb02
ora.scan1.vip	ora.scan_vip.type	OFFLINE,	OFFLINE,
ora.vdb01.vip	ora.cluster_vip_net1.type	ONLINE,	ONLINE on vdb01,
ora.vdb02.vip	ora.cluster_vip_net1.type	ONLINE,	ONLINE on vdb02,
xag.myapp.mysql	xag.mysql.type	ONLINE,	ONLINE on vdb02,

Oracle Clusterware was not able to restart “httpd daemon” on the same node so, after three attempts, restarted all the service resources that have dependencies each-other on the other node.
The number of attempts is configured within the “failed” resource and you can check the configuration with the following command (connected as **oracle** user, the real owner of the cluster)

```
# crsctl status resource <resource_name> -p  
# crsctl status resource httpd.myapp.mysql -p
```

In the picture below we can see that the “**RESTART_ATTEMPTS**” is **3**; after this event, Oracle Clusterware proceed to migrate that resource and all dependencies on the other node.

```
Terminal - oracle@vdb01:~
```

```
[oracle@vdb01 ~]$ crsctl status resource httpd.myapp.mysql -p
NAME=httpd.myapp.mysql
TYPE=cluster_resource
ACL=owner:root:rwx,pgrp:root:r-x,other::r--,user:oracle:r-x
ACTIONS=
ACTION_SCRIPT=/mysql/cluster_scripts/apache.sh
ACTION_TIMEOUT=60
ACTIVE_PLACEMENT=0
AGENT_FILENAME=%CRS_HOME%/bin/scriptagent
AUTO_START=restore
CARDINALITY=1
CHECK_INTERVAL=60
CHECK_TIMEOUT=0
CLEAN_TIMEOUT=60
DEGREE=1
DELETE_TIMEOUT=60
DESCRIPTION=
ENABLED=1
FAILOVER_DELAY=0
FAILURE_INTERVAL=0
FAILURE_THRESHOLD=0
HOSTING_MEMBERS=vdb01 vdb02
INSTANCE_FAILOVER=1
INTERMEDIATE_TIMEOUT=0
LOAD=1
LOGGING_LEVEL=1
MODIFY_TIMEOUT=60
OFFLINE_CHECK_INTERVAL=0
PLACEMENT=restricted
RELOCATE_BY_DEPENDENCY=1
RESTART_ATTEMPTS=3
SCRIPT_TIMEOUT=60
SERVER_CATEGORY=
SERVER_POOLS=
START_CONCURRENCY=0
START_DEPENDENCIES=hard(xag.myapp.mysql)
START_TIMEOUT=0
STOP_CONCURRENCY=0
STOP_DEPENDENCIES=hard(xag.myapp.mysql)
STOP_TIMEOUT=0
UPTIME_THRESHOLD=1h
USER_WORKLOAD=no
USE_STICKINESS=0

[oracle@vdb01 ~]$
```

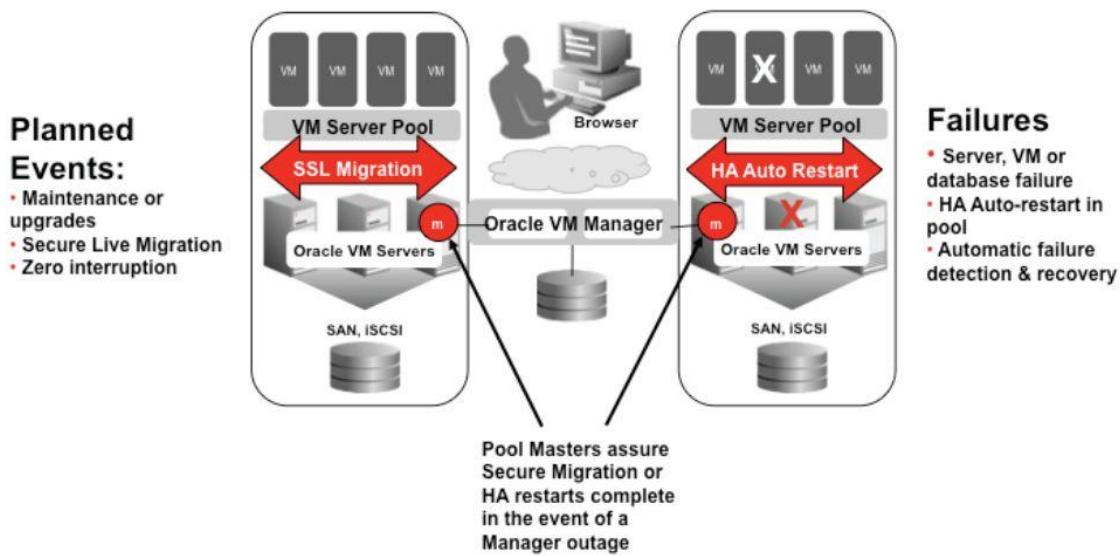
When the test is finished, please fix the simulated corruption on guest “**vdb01.oow.local**”, as root, with:
mv /usr/sbin/httpd.corrupted /usr/sbin/httpd

Demonstrate high-availability features covered by Oracle VM

Oracle VM high-availability consists of two main features:

- Oracle VM High-Availability
- Oracle VM Live-Migration

Here a picture that could better describe these features H/A features for Planned events and Failures :



Oracle VM Live-Migrate

Live migration is a process to move a running virtual machine from one Oracle VM Server to another, while applications on the existing virtual machine continue to run. Live migration ensures high availability of virtual machines. This feature is important, and useful, when the existing Oracle VM Server may be out of commission, or on a planned shutdown for maintenance purposes.

You can only migrate one virtual machine at a time. Cross-server pool live migration is not allowed. You can only migrate virtual machines from one Oracle VM Server to another within the same server pool.

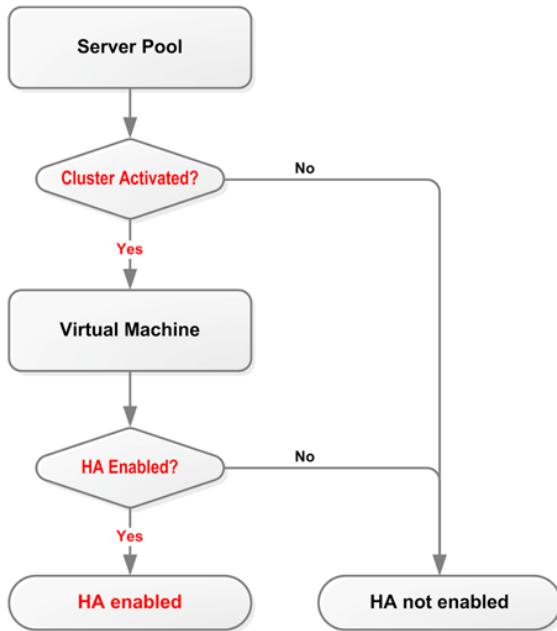
Oracle VM High-Availability

You can set up High-Availability to help ensure the uninterrupted availability of a virtual machine. If HA is configured and a Oracle VM Server is restarted or shut down, the virtual machines running on it are either restarted on, or migrated to, another Oracle VM Server.

The following prerequisites are requirement to implement HA:

- The server pool must contain multiple Oracle VM Servers.
HA cannot be implemented with a stand-alone Oracle VM Server.
- All Oracle VM Servers must be Oracle VM Server Release 3.0 or above.
- Oracle VM Pool needs to be "clustered"

The following chart will better explain requirements of Oracle VM High Availability:

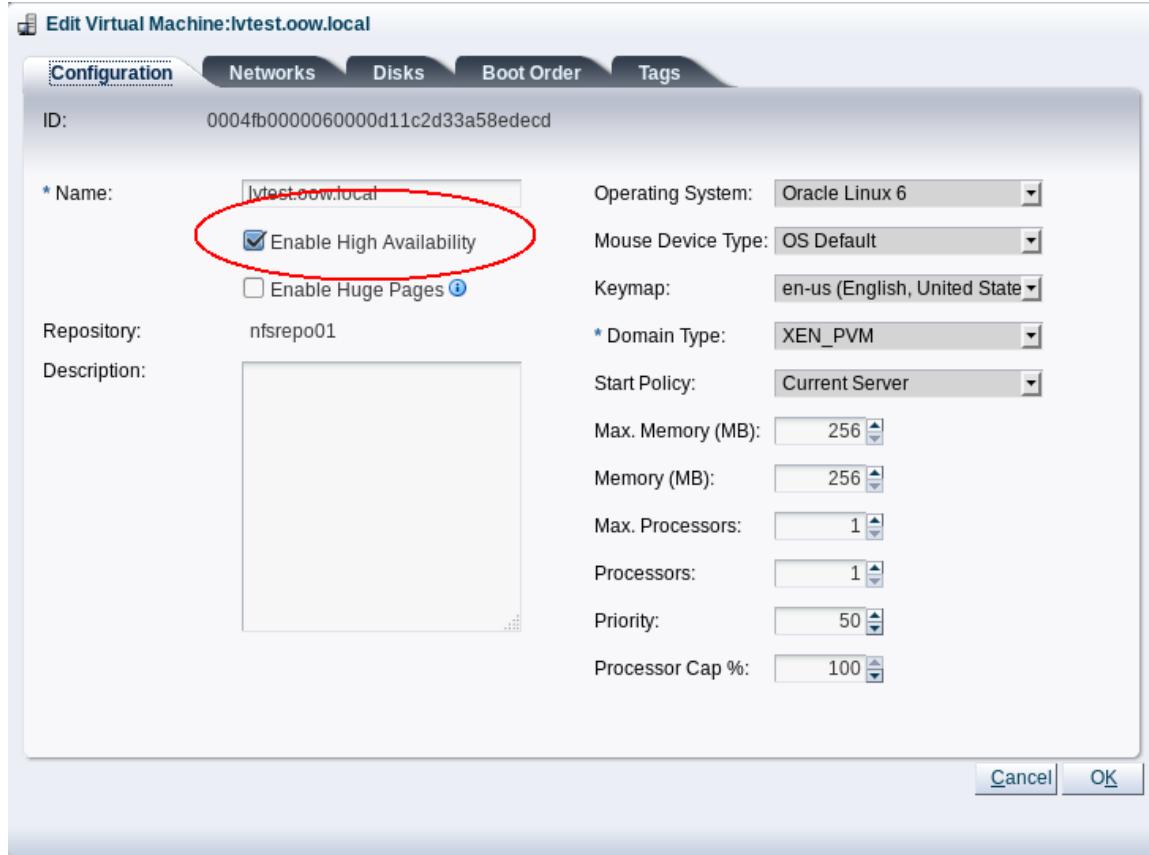


To use HA, you have to configure a clustered Oracle VM Server Pool:

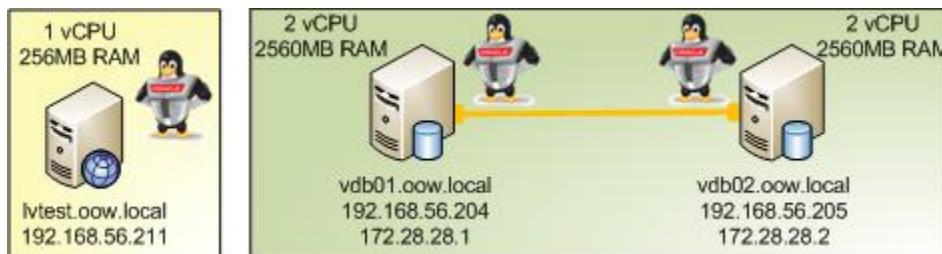
Create a Server Pool

Create Server Pool <ul style="list-style-type: none"> <input type="checkbox"/> Add Servers <input type="checkbox"/> Tags(Optional) 	<p>* Server Pool Name: <input type="text"/></p> <p>* Virtual IP Address for the Pool: <input type="text"/></p> <p>VM Console Keypad: <input type="text" value="en-us (English, United States)"/></p> <p>VM Start Policy: <input type="text" value="Best Server"/></p> <p>Secure VM Migrate: <input type="checkbox"/></p> <p>Clustered Server Pool: <input checked="" type="checkbox"/></p> <p>Timeout for Cluster: <input type="text" value="120"/> Seconds</p> <p>Storage for Server Pool:</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> Network File System <input type="radio"/> Physical Disk <p>* Storage Location: <input type="text"/> </p> <p>Description:</p>
---	---

To use HA, you have to enable High-Availability option on the Oracle VM Guest (virtual-server) :



On our architecture we have three virtual-servers (guests):



On our H/A tests, due to the reduced amount of resources available on our system,(mainly RAM), we will demonstrate both Oracle VM High-Availability and Oracle VM Live-Migrate using the guest named "**lvtest.oow.local**".

To demonstrate Oracle VM Live-Migrate features execute the following steps:

1. Connect to Oracle VM Manager and verify the status of the guest named "lvtest.oow.local".
Oracle VM Manager URL: <https://192.168.56.200:7002/ovm/console>
Username: admin
Password: Welcome1
2. Select "**Servers & VMs tab**"
3. Expand and select pool named "**ovmpool.oow.local**"
4. Select perspective "**Virtual Machine**"

5. Verify the status of the virtual-server “**lvtest**” and which Oracle VM Server owns it.

The screenshot shows the Oracle VM Home interface with the following details:

- URL:** https://192.168.56.200:7002/ovm/console/faces/resourceView.jspx
- Tab:** Servers and VMs (highlighted by a red circle)
- Perspective:** Virtual Machines (highlighted by a red circle in the toolbar)
- Server Pool:** ovmpool.oow.local (highlighted by a red circle)
- Virtual Machines Table:**

Name	Status	Event Severity	Server	Max. Memory (MB)
lvtest.oow.local	Running	Informational	ovs01.oow.local	256
vdb01.oow.local	Running	Informational	ovs01.oow.local	2560
vdb02.oow.local	Running	Informational	ovs02.oow.local	2048

In the case above the guest “**lvtest.oow.local**” is running on Oracle VM Server “**ovs01.oow.local**”.

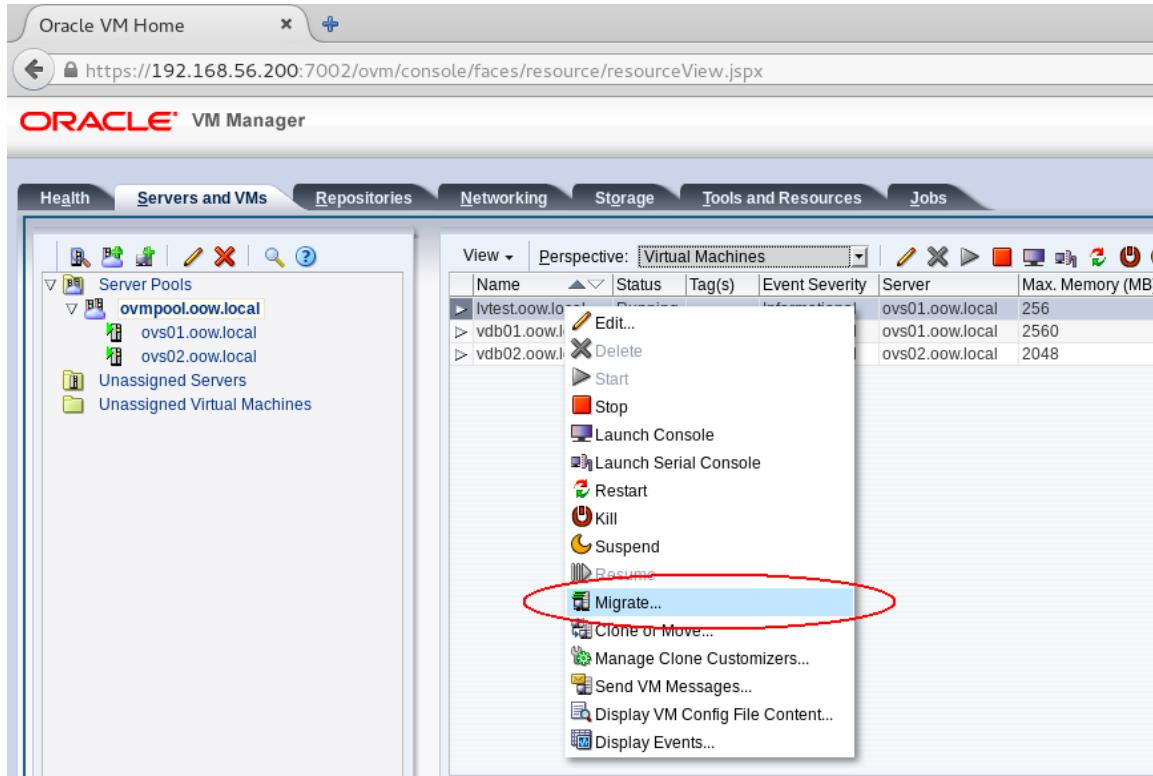
6. Connect, by ssh, to the server “**lvtest.oow.local**”
 ssh root@192.168.56.211 (password is ovsroot)

```
[scoter@area51: ~]# ssh root@192.168.56.211
root@192.168.56.211's password:
Last login: Mon Sep 15 07:58:57 2014 from 192.168.56.1
[root@lvtest ~]#
```

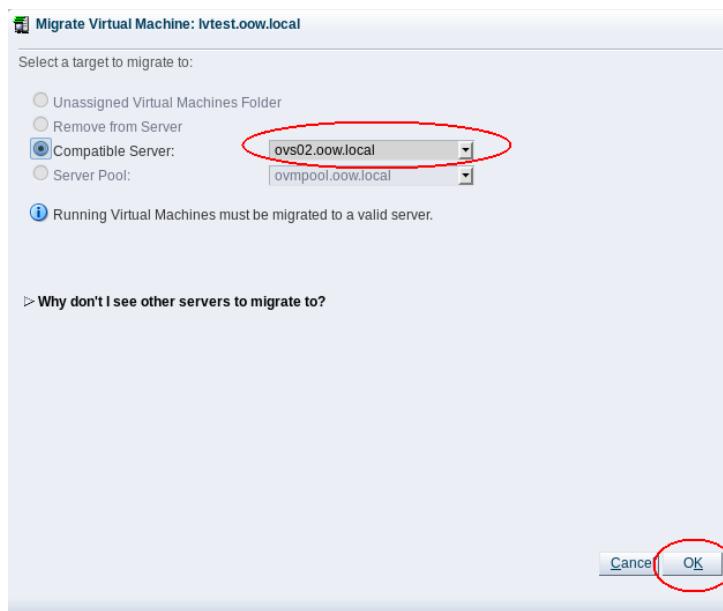
7. Launch a command that will show the date of the server updated every one second.
 # while sleep 1; do date >> /tmp/date.log; done &
 # tail -f /tmp/date.log

```
[root@lvtest ~]# while sleep 1; do date >> /tmp/date.log; done &
[1] 2186
[root@lvtest ~]# tail -f /tmp/date.log
Mon Sep 15 08:18:42 PDT 2014
Mon Sep 15 08:18:43 PDT 2014
Mon Sep 15 08:18:44 PDT 2014
Mon Sep 15 08:18:45 PDT 2014
Mon Sep 15 08:18:46 PDT 2014
Mon Sep 15 08:18:47 PDT 2014
Mon Sep 15 08:18:48 PDT 2014
Mon Sep 15 08:18:49 PDT 2014
Mon Sep 15 08:18:50 PDT 2014
Mon Sep 15 08:18:51 PDT 2014
```

8. While this script is running, execute the live-migrate of the guest by Oracle VM Manager.
 Re-open the **Oracle VM Manager console**, select guest named “**lvttest.0ow.local**”, **right-click** and select “**Migrate**”.



9. Select “compatible-server” “ovs02.0ow.local” and confirm with “OK”.



10. While migration is running, verify the output of the command executed at point (7) of this section and verify the sequentiality of the output; *in the worse case you'll lost only 1 second of the output and this is a very nice result for a demo environment built on top of VirtualBox.*

```

Mon Sep 15 08:30:53 PDT 2014
Mon Sep 15 08:30:54 PDT 2014
Mon Sep 15 08:30:55 PDT 2014
Mon Sep 15 08:30:56 PDT 2014
Mon Sep 15 08:30:57 PDT 2014
Mon Sep 15 08:30:59 PDT 2014
Mon Sep 15 08:31:00 PDT 2014
Mon Sep 15 08:31:01 PDT 2014
Mon Sep 15 08:31:02 PDT 2014
Mon Sep 15 08:31:03 PDT 2014
Mon Sep 15 08:31:04 PDT 2014
Mon Sep 15 08:31:05 PDT 2014
Mon Sep 15 08:31:06 PDT 2014
Mon Sep 15 08:31:07 PDT 2014
Mon Sep 15 08:31:08 PDT 2014
Mon Sep 15 08:31:09 PDT 2014
Mon Sep 15 08:31:10 PDT 2014
Mon Sep 15 08:31:11 PDT 2014
Mon Sep 15 08:31:12 PDT 2014
Mon Sep 15 08:31:13 PDT 2014
Mon Sep 15 08:31:13 PDT 2014
Mon Sep 15 08:31:15 PDT 2014

```

To demonstrate Oracle VM High-Availability features execute the following steps:

1. Connect to Oracle VM Manager and verify:
 - All three guests are in "Running" state
 - Which physical server owns the guest named "lvtest.oow.local"

Name	Status	Tag(s)	Event Severity	Server
lvtest.oow.local	Running		Informational	ovs02.oow.local
vdb01.oow.local	Running		Informational	ovs01.oow.local
vdb02.oow.local	Running		Informational	ovs02.oow.local

In the example above:

- Guests “**lvttest**” and “**vdb02**” are running on physical server **ovs02.oww.local**
- Guest “**vdb01**” is running on physical server **ovs01.oww.local**

2. To simulate a complete outage, verify which of the clustered servers owns the mysql/webdemo service by connecting on one of them:

`ssh root@192.168.56.204
crsstat -t`

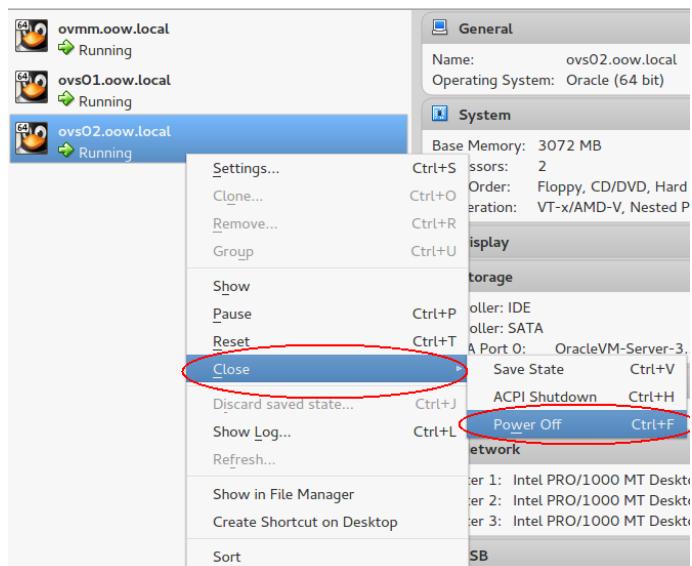
```
[root@vdb01 ~]# crsstat -t
HA Resource          Type        Target      State
httpd.myapp.mysql   cluster_resource ONLINE,    ONLINE on vdb02,
mysql.own.local     app.appvipx.type  ONLINE,    ONLINE on vdb02
ora.ACFSMySQLDG.volMySQL.advm ora.volume.type  ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.ACFSMySQLDG.dg    ora.diskgroup.type ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.CLUSTERDG.dg    ora.acfs.type    ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.acfsmysqldg.volmysql.acfs ora.asm.type    ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.asm              ora.cvu.type    ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.cvu              ora.network.type ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.net1.network     ora.ons.type    ONLINE , ONLINE  ONLINE on vdb01, ONLINE on vdb02
ora.ons              ora.scan_vip.type OFFLINE,
ora.scan1.vip        ora.cluster_vip_net1.type ONLINE,
ora.vdb01.vip        ora.cluster_vip_net1.type ONLINE,
ora.vdb02.vip        ora.cluster_vip_net1.type ONLINE,
xag.mysql.type       xag.mysql.type  ONLINE,    ONLINE on vdb02,
[root@vdb01 ~]#
```

In the case above we have that:

- **web/mysql** services are active on guest **vdb02**
- guests **vdb02** and **lvttest** are active on Oracle VM Server **ovs02.oww.local**

The target of this test is to **simulate a crash of the Oracle VM Server that owns**, at the same time, both guests **lvttest** and **vdb0(?)** that owns the **web/mysql services** (if you need to move services between cluster-nodes, you can use the steps reported in this document at the section “Demonstrate high-availability features covered by Oracle Clusterware”). This latest test will demonstrate how both components (Oracle VM and Oracle Clusterware) will together work to maintain the highest service-level possible.

3. Kill “Oracle VM Server” that owns all active web/mysql services and guest named “lvttest”. Open “VirtualBox Console”, right click on the server and choose “**Close**”, “**Poweroff**”.



Next steps to exercise on the environment will be:

- Oracle VM Manager head off the **fault** on the Oracle VM Server “**ovs02.oow.local**”

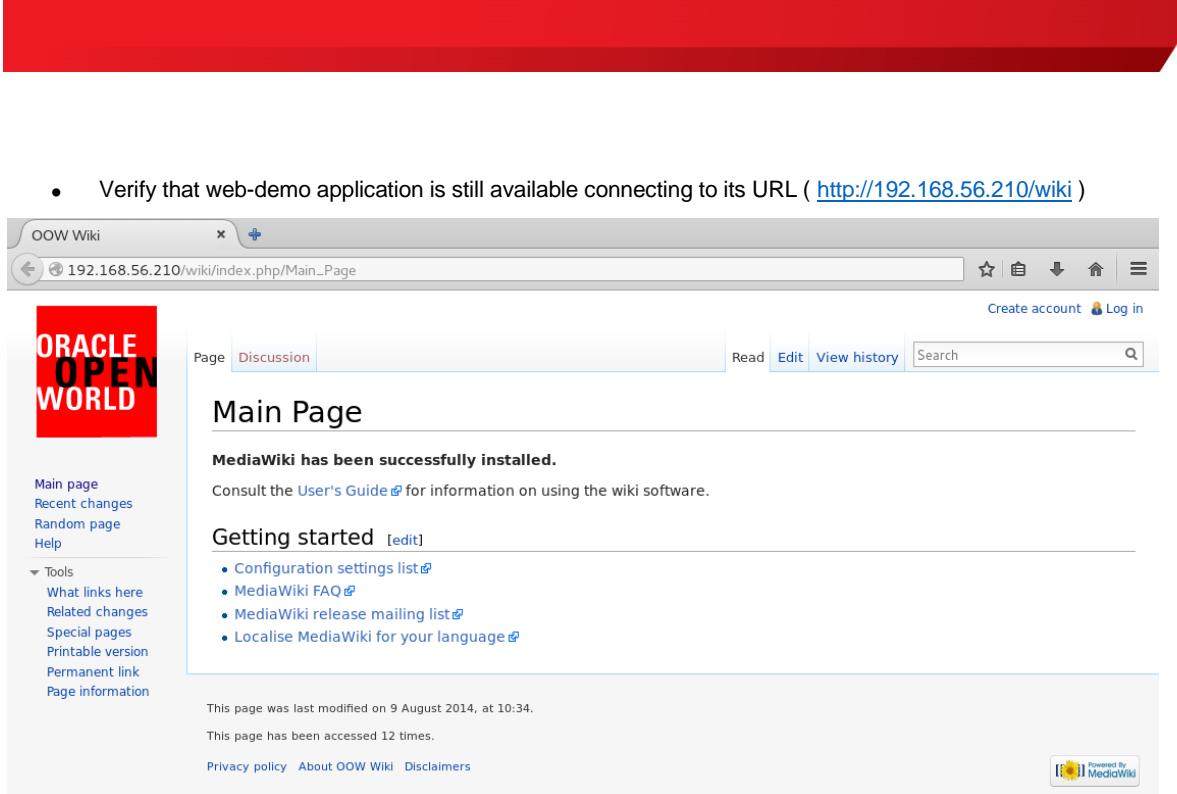
Name	Status	Tag(s)	Event Severity	Server
lvttest.oow.local	Running		Critical	ovs02.oow.local
vdb01.oow.local	Running		Informational	ovs01.oow.local
vdb02.oow.local	Running		Critical	ovs02.oow.local

- Oracle VM Pool proceed with the restarting of the guest “**lvttest**” on the alive server “**ovs01.oow.local**”

Name	Status	Tag(s)	Event Severity	Server
lvttest.oow.local	Running		Informational	ovs01.oow.local
vdb01.oow.local	Running		Informational	ovs01.oow.local
vdb02.oow.local	Stopped		Critical	ovs02.oow.local

- Oracle Clusterware (on guest “**vdb01**” proceed to relocate services on the active node of the cluster)

```
[root@vdb01 ~]# crsstat -t
HA Resource                               Type        Target          State
-----[redacted]-----[redacted]-----[redacted]
httpd.myapp.mysql                         cluster_resource ONLINE,      ONLINE on vdb01,
mysql.oow.local                           app.appvipx.type  ONLINE,      ONLINE on vdb01,
ora.ACFSMYSQLDG.VOLMYSQLDG.advm         ora.volume.type   ONLINE,      ONLINE on vdb01,
ora.ACFSMYSQLDG.dg                         ora.diskgroup.type ONLINE,      ONLINE on vdb01,
ora.CLUSTERDG.dg                          ora.diskgroup.type ONLINE,      ONLINE on vdb01,
ora.acfsmysqldg.volmysql.acfs            ora.acfs.type     ONLINE,      ONLINE on vdb01,
ora.asm                                    ora.asm.type      ONLINE,      ONLINE on vdb01,
ora.cvu                                    ora.cvu.type      ONLINE,      ONLINE on vdb01,
ora.net1.network                          ora.network.type ONLINE,      ONLINE on vdb01,
ora.ons                                    ora.ons.type      ONLINE,      ONLINE on vdb01,
ora.scan1.vip                             ora.scan.vip.type OFFLINE,    OFFLINE,
ora.vdb01.vip                            ora.cluster_vip_net1.type ONLINE,      ONLINE on vdb01,
ora.vdb02.vip                            ora.cluster_vip_net1.type ONLINE,      INTERMEDIATE on vdb01,
xag.myapp.mysql                          xag.mysql.type    ONLINE,      ONLINE on vdb01,
[root@vdb01 ~]#
```



Upshot: Oracle VM High Availability

Summarizing what we saw during this lab we can say that an infrastructure of this type has high-availability functionality built in at every level, where:

- Oracle VM is able to automatically recover from a physical-server fault restarting a guest machine on an other node
- Oracle VM allows to schedule physical server maintenance (and/or replacement) without inefficiency thanks to live-migrate
- Oracle Clusterware allows to protect MySQL Server and any further application type with Grid Infrastructure Agents or with custom scripts; it's complementary to Oracle VM but it's also able to identify a fault within the guest (ex. Process dies, process fault and/or local filesystem corruption).
- Ksplice allows to update kernel-in-memory without disruption or server maintenance

This kind of architecture, obviously, is applicable to all Oracle and non-Oracle products on top of guest virtual-machines.

Appendix A: preparation of the environment before the lab

Purpose

This section explains how to prepare the environment to run this lab. It is useful if you want to run this lab at home or office.

The first step is to find an X86 machine (server, desktop or laptop) with the required resources (4 vcpus, 16 GB of ram, and 100 GB of disk space) and install Oracle VM VirtualBox on it.

Then, there are 3 servers to install (3 VirtualBox virtual machines in fact):

- Oracle VM Server (1 of 2)
- Oracle VM Server (2 of 2)
- Oracle VM Manager

Download required binaries

This section lists the required binaries and explains how to download them.

For your X86 machine:

- 1) **Oracle VM VirtualBox 4.3.x binaries (4.3.12 during writing of this document)**
Download the version for your OS from <https://www.virtualbox.org/wiki/Downloads>
- 2) **Oracle VM VirtualBox extension Pack 4.3.x**
Download from <https://www.virtualbox.org/wiki/Downloads> (same file for all OSes)

For Oracle VM Server:

- 3) **VirtualBox template for Oracle VM Server 3.3.1 (to update – SCOTER)**
Download from <http://www.oracle.com/technetwork/server-storage/vm/template-1482544.html>
Direct link: <http://download.oracle.com/otn/vm/OracleVMServer3.2.4-b525.ova>
Filename: OracleVMServer.3.2.4-b525.ova (size 249 MB)

For Oracle VM Manager:

- 4) **VirtualBox template for Oracle VM Manager 3.3.1 (to update – SCOTER)**
Download from <http://www.oracle.com/technetwork/server-storage/vm/template-1482544.html>

For Oracle VM guest Template:

- 5) **Oracle VM VirtualBox template for Oracle Linux 6.5 x86-64**
Download for Oracle E-delivery Linux/Oracle VM platform (<https://edelivery.oracle.com/oraclevm>)
Select Product Pack “**Oracle VM Templates**” and Platform “**x86 64 bit**”, then Click Go
Select “[Oracle Linux 6 Update 5 template \(OVF\) - Paravirtualized x86_64 \(64 bit\)](#)”

Installation of Oracle VM VirtualBox

- 1) Find an x86 machine (desktop, laptop, server) matching the following prerequisites:
 - At least 16 GB of RAM
 - x86-64 CPU (Intel or AMD) - at least 4 cpus threads with Virtualization Extensions
 - OS supported by VirtualBox
 - 100 GB of disk space
- 2) Install the Oracle VM VirtualBox 4.3.x binaries on your x86 machine
- 3) Start the Oracle VM VirtualBox console

4) If not already created, create a two host only network in VirtualBox using following details:

- vboxnet0
subnet = 192.168.56.0/24
VBox Machine ip = 192.168.56.1
- vboxnet1
subnet = 172.28.28.0/24
VBox Machine ip = 172.28.28.254

5) Choose the folder you want to use to store the virtual machines files.

(go to File, Preferences, General, Default Machine Folder)

6) Install the Oracle VM VirtualBox extension Pack

(go to File, Preferences, Extensions)

Installation of Oracle VM Server (2 servers to install)

- a) In the Oracle VM VirtualBox console, import the VM from the Oracle VM Server template
 - File
 - Import Appliance
 - Select the file OracleVMServer.3.3.1-b1065.ova
 - Next
 - Change the name of the System 1 from "**Oracle VM Server 3.3.1-b1065**" to "**ovs01.oow.local**"
The name for the second server is "**ovs02.oow.local**"
 - Import
- a) Modify the settings of the virtual machine "**ovs01.oow.local**" and "**ovs02.oow.local**"
 - Configure the network
 - Network, Adapter 1, Attached to Host only Adapter, vboxnet0
 - Network, Adapter 2, Attached to Host only Adapter, vboxnet1
 - Network, Adapter 3, Attached to NAT
All Virtual-NICS need the Promiscous-Mode (advanced section) set to "**Allow ALL**"
 - Memory: **4096 MB**
 - Configure the storage:
 - Enable Host I/O cache on the SATA Controller
- b) Start the virtual machine "**ovs01.oow.local**"
- c) Configure the virtual machine (in the VM console)
 - Configure network
 - IP address : **192.168.56.201**
 - Netmask : **255.255.255.0**
 - Gateway : **192.168.56.1**
 - DNS server : **192.168.56.1**
(we will not use DNS, but we have to give an IP address here)
 - Hostname : **ovs01.oow.local**
 - Wait for the end of boot
- d) Start the virtual machine "**ovs02.oow.local**"
- e) Configure the virtual machine (in the VM console)
 - Configure network
 - IP address : **192.168.56.202**
 - Netmask : **255.255.255.0**
 - Gateway : **192.168.56.1**
 - DNS server : **192.168.56.1**
(we will not use DNS, but we have to give an IP address here)

- Hostname : **ovs02.oow.local**
 - Wait for the end of boot
- f) Open a terminal on your Unix/Linux x86 machine and connect to both VMs with ssh (you can use Putty on Microsoft Windows)
- ```
$ ssh root@192.168.56.201 (password is ovsroot)
$ ssh root@192.168.56.202 (password is ovsroot)
```

- g) Add the following lines to the /etc/hosts file

```
vm guest ip addresses
192.168.56.204 vdb01.oow.local vdb01
192.168.56.205 vdb02.oow.local vdb02
192.168.56.206 vdb01-vip.oow.local vdb01-vip
192.168.56.207 vdb02-vip.oow.local vdb02-vip
Scan-Vip Disabled
192.168.56.208 vdb-scan.oow.local vdb-scan
192.168.56.210 mysql.oow.local mysql
192.168.56.211 lvtest.oow.local lvtest

host ip addresses
192.168.56.200 ovmm.oow.local ovmm
192.168.56.201 ovs01.oow.local ovs01
192.168.56.202 ovs02.oow.local ovs02
192.168.56.202 ovmpool.oow.local ovmpool
```

### Installation of Oracle VM Manager

- a) In the Oracle VM VirtualBox console, import the VM from the Oracle VM Manager template
  - File
  - Import Appliance
  - Select the file OracleVMManager.3.3.1-b1065.ova
  - Next
  - Change the name of the System from “**Oracle VM Manager 3.3.1-b1065**” to “**ovmm.oow.local**”
  - Import
- b) Modify the settings of the virtual machine “**ovmm.oow.local**”
  - Configure the network
    - Network, Adapter 1, Attached to Host only Adapter, vboxnet0
    - Network, Adapter 2, Attached to NAT

All Virtual-NICS need the Promiscous-Mode (advanced section) set to “**Allow ALL**”
  - Memory: **4096 MB**
  - Storage: **add a new virtual-disk of 80 GB**
- c) Start the virtual machine “**ovmm.oow.local**”
- d) Configure the virtual machine (in the VM console)
  - Set root password to **ovsroot**
  - Configure network
    - IP address : **192.168.56.200**
    - Netmask : 255.255.255.0
    - Gateway : 192.168.56.1
    - DNS server : **192.168.56.1**  
(we will not use DNS, but we have to give an IP address here)
    - Hostname : ovmm.oow.local
    - Wait for the end of boot
- e) Open a terminal on your Unix/Linux x86 machine and connect to the VM with ssh (you can use Putty on Microsoft Windows)

```
$ ssh root@192.168.56.200
```

- f) Add the following lines to the file /etc/hosts

```
vm guest ip addresses
192.168.56.204 vdb01.oow.local vdb01
192.168.56.205 vdb02.oow.local vdb02
192.168.56.206 vdb01-vip.oow.local vdb01-vip
192.168.56.207 vdb02-vip.oow.local vdb02-vip
Scan-Vip Disabled
192.168.56.208 vdb-scan.oow.local vdb-scan
192.168.56.210 mysql.oow.local mysql
192.168.56.211 lvtest.oow.local lvtest

host ip addresses
192.168.56.200 ovmm.oow.local ovmm
192.168.56.201 ovs01.oow.local ovs01
192.168.56.202 ovs02.oow.local ovs02
192.168.56.202 ovmpool.oow.local ovmpool
```

- g) Create a new Volume-Group, two logical-volumes and create filesystems.

```
fdisk -l
fdisk /dev/sdc
pvcreate /dev/sdc1
vgcreate vgOVM /dev/sdc1
lvdisplay
lvcreate -L 13G -n ovspool vgOVM
lvcreate -I 100%FREE -n ovsrepo vgOVM
lvdisplay
ls -l /dev/mapper/
mkfs.ext4 /dev/mapper/vgOVM-ovspool
mkfs.ext4 /dev/mapper/vgOVM-ovsrepo
```

- h) Get logical-volume Block-ID and edit "/etc/fstab" and insert the new entries:

```
[root@ovmm ~]# blkid /dev/mapper/vgOVM-ovspool
/dev/mapper/vgOVM-ovspool: UUID="8df452d7-0b12-4a56-8b95-b30cd4bf491b" TYPE="ext4"
[root@ovmm ~]# blkid /dev/mapper/vgOVM-ovsrepo
/dev/mapper/vgOVM-ovsrepo: UUID="b854a663-04ec-45af-b021-250322866e8a" TYPE="ext4"
```

```
[root@ovmm ~]# cat /etc/fstab
#
/etc/fstab
Created by anaconda on Fri Jul 18 05:00:12 2014
#
Accessible filesystems, by reference, are maintained under '/dev/disk'
See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/VolGroup-lv_root / ext4 defaults 1 1
UUID=ea6616bd-14b7-46f6-8499-41ce4d56513a /boot ext4 defaults 1 2
UUID=81d05ca6-037b-4f53-90d0-274d067e2908 /u01 ext4 defaults 1 2
UUID=8df452d7-0b12-4a56-8b95-b30cd4bf491b /home/nfs/ovmcluster ext4 defaults 1 2
UUID=b854a663-04ec-45af-b021-250322866e8a /home/nfs/ovmdata ext4 defaults 1 2
/dev/mapper/VolGroup-lv_swap swap defaults 0 0
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

- i) Mount new filesystems and verify that they are exported by NFS protocol.

```
mount -a
```

```
cat /etc/exports
service nfs status
```

Example:

```
[root@ovmm ~]# cat /etc/exports
/home/nfs/ovmcluster *(rw,no_root_squash)
/home/nfs/ovmdata *(rw,no_root_squash)
[root@ovmm ~]# service nfs status
rpc.svcgssd is stopped
rpc.mountd (pid 1789) is running...
nfsd (pid 1806 1805 1804 1803 1802 1801 1800 1799) is running...
rpc.rquotad (pid 1785) is running...
```

## Discover Oracle VM Servers, add the file server by Oracle VM Manager, create the Server Pool

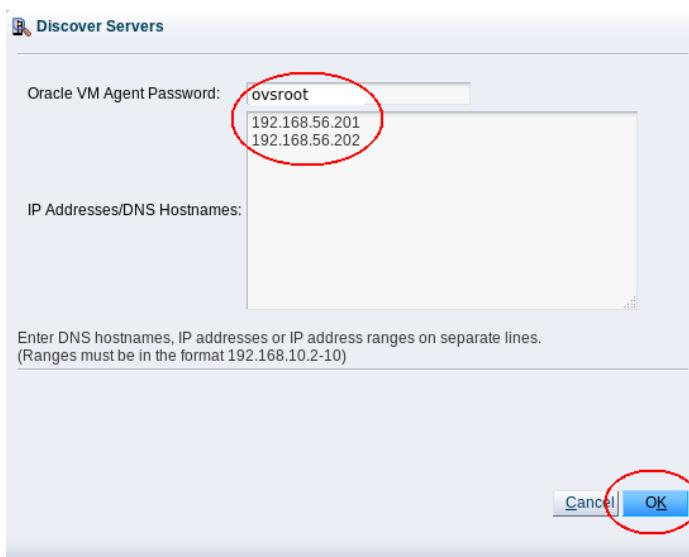
Discover Oracle VM Servers installed with the following steps:

- Click on the icon “Discover Servers...”



- Insert values:

Oracle VM Agent Password: ovsroot  
IP Addresses: 192.168.56.201 and 192.168.56.202



- c) Add the file-server that will guest all nfs-filesystems needed



- d) Insert nfs values:

Name: nfs  
Access Host IP: 192.168.56.200

**Discover a File Server**

**File Server Parameters**

- Add Admin Servers
- Select Refresh Servers
- Select File Systems

\* Storage Plug-in: Oracle Generic Network File System *Enter between 1 and 256 characters.*

\* Name: nfs

\* Access Host (IP) Address: 192.168.56.200

Uniform Exports:

Description:

**Cancel** **Next**

- e) Verify that in the “Admin” and “Refresh” server both “ovs01” and “ovs02” exist.

**Edit File Server:ovmmnfs**

**Configuration** **Admin Servers** **Refresh Servers**

Select the Server(s) that can be used for administrative access to this File Server.  
Required when using a Network File System in a clustered Server Pool.

**Available Admin Server(s)**

**Selected Admin Server(s)**

ovs01.0ow.local  
ovs02.0ow.local

**Available Refresh Server(s)**

**Selected Refresh Server(s)**

ovs01.0ow.local  
ovs02.0ow.local

- f) Refresh both NFS-Filesystem presented.

The screenshot shows the Oracle VM Manager interface with the 'Storage' tab selected. On the left, there's a tree view with 'File Servers' expanded, showing 'ovmmnfs'. In the main pane, a table lists two NFS file systems:

| Name                       | Event Severity | Refreshed | Size (GB)  |
|----------------------------|----------------|-----------|------------|
| nfs on 192.168.56.200:/... | Informational  | Yes       | 12.51 0.16 |
| nfs on 192.168.56.200:/... | Informational  | Yes       | 6.38 54.52 |

A yellow box highlights the 'Refresh File System' button at the top right of the table.

- g) Create a clustered server-pool  
Select "Servers and VMs" tab and click on the "Create Server Pool" icon.



- h) Insert the following informations ("Storage Location" choose the ~12GB NFS-Filesystem)  
In the next screen choose both servers and proceed with the pool creation.

The screenshot shows the 'Create a Server Pool' dialog box. Several fields are highlighted with red circles:

- \* Server Pool Name: ovmm.ooow.local
- \* Virtual IP Address for the Pool: 192.168.56.203
- Clustered Server Pool:
- Storage for Server Pool:  Network File System (highlighted with a red circle)

At the bottom right, there are 'Cancel' and 'Next' buttons.

- i) Create a new storage repository that will guest our vservers.

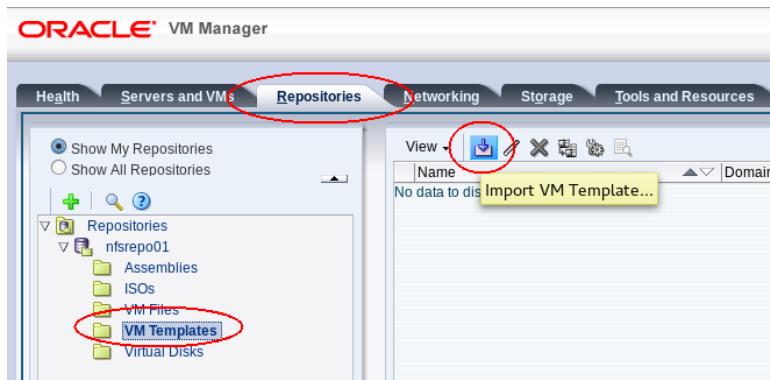


Insert the following informations, choose :  
 "Repository Location" : Network File Server  
 "Network File System": filesystem with a size of at least 80gb

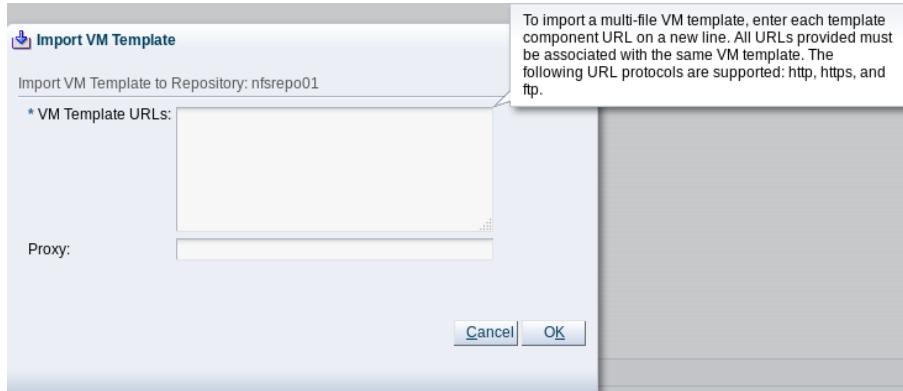
|                                                                           |                                                                                                                                                                                                                                                                                                   |
|---------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Repository Information</b>                                             | * Repository Name: <input type="text" value="repo01"/><br>* Repository Location: <input checked="" type="radio"/> Network File Server <input type="radio"/> Physical Disk<br>* Network File System: <input type="text"/><br>Share Path: <input type="text"/><br>Description: <input type="text"/> |
| <input type="button" value="Cancel"/> <input type="button" value="Next"/> |                                                                                                                                                                                                                                                                                                   |

#### Import the "Oracle VM Template" Oracle Linux 6.5 x86-64

- a) To import an Oracle VM Template proceed as shown in the following picture:



You can download this template at <http://edelivery.oracle.com>  
 To import an Oracle VM Template you have to supply an http/https/ftp URL:



## Modify the network configuration

Each “Oracle VM Server” has three network-interfaces; their role will be:

- eth0 = network dedicated to Network Management, Live Migrate, Cluster Heartbeat roles and Virtual-Machine public-Network.
  - eth1 = network dedicated to guest intracluster Network
  - eth2 = network dedicated to guest internet-access Network ( DHCP )
- a) On the “Networking” Tab, edit the “default” network and add the role “Virtual Machine”

- b) Create a new network named “Intracluster” and assign NICs “eth1” to it.

Configuration > Ports

| ID:          | 100a266cbe | * Name: | intracluster | * Network Uses: | <input type="checkbox"/> Management<br><input type="checkbox"/> Live Migrate<br><input type="checkbox"/> Cluster Heartbeat<br><input checked="" type="checkbox"/> Virtual Machine<br><input type="checkbox"/> Storage |
|--------------|------------|---------|--------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Description: |            |         |              |                 |                                                                                                                                                                                                                       |

Configuration > Ports

| Port Name          | Server          | MTU  | Addressing | IP Address | Mask | Bonding |
|--------------------|-----------------|------|------------|------------|------|---------|
| eth1 on ovs02.o... | ovs02.0ow.local | 1500 | None       |            |      | No      |
| eth1 on ovs01.o... | ovs01.0ow.local | 1500 | None       |            |      | No      |

- c) Create a new network named “Internet” and assign NICs “eth2” to it.

Configuration > Ports

| ID:          | 103c790e25 | * Name: | internet | * Network Uses: | <input type="checkbox"/> Management<br><input type="checkbox"/> Live Migrate<br><input type="checkbox"/> Cluster Heartbeat<br><input checked="" type="checkbox"/> Virtual Machine<br><input type="checkbox"/> Storage |
|--------------|------------|---------|----------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Description: |            |         |          |                 |                                                                                                                                                                                                                       |

Configuration > Ports

| Port Name          | Server          | MTU  | Addressing | IP Address | Mask          | Bonding |
|--------------------|-----------------|------|------------|------------|---------------|---------|
| eth2 on ovs01.o... | ovs01.0ow.local | 1500 | Dynamic    | 10.0.4.15  | 255.255.255.0 | No      |
| eth2 on ovs02.o... | ovs02.0ow.local | 1500 | Dynamic    | 10.0.4.15  | 255.255.255.0 | No      |

### Create three new guest(s) servers

- vdb01.0ow.local = node(1) of the Oracle Clusterware
- vdb02.0ow.local = node(2) of the Oracle Clusterware
- lvttest.0ow.local = spare-node dedicated to live-migration test(s).

To create these new three guests you can use the Oracle VM Templatized imported in the steps before.

Here the guest(s) details:

#### vdb01.0ow.local

Processors: 2  
Max Processors: 2  
Memory: 2560 MB  
Max Memory: 2560 MB  
Disk(s): default OS disk of 13GB, one more local virtual-disk of 15GB

Network: first NIC network “**192.168.56.0**”, second NIC network “**intracluster**”, third NIC network “**internet**”.

### vdb02.oow.local

Processors: 2

Max Processors: 2

Memory: 2560 MB

Max Memory: 2560 MB

Disk(s): default OS disk of 13GB, one more local virtual-disk of 15GB

Network: first NIC network “**192.168.56.0**”, second NIC network “**intracluster**”, third NIC network “**internet**”.

### lvtest.oow.local

Processors: 1

Max Processors: 1

Memory: 256 MB

Max Memory: 256 MB

Disk(s): default OS disk of 13GB

Network: first NIC network “**192.168.56.0**”, third NIC network “**internet**”.

Create additional virtual-disks that will be dedicated to Oracle Clusterware; on the “Repository” tab, expand your repository, choose “Virtual Disks” and add two new virtual-disks like in the example below:

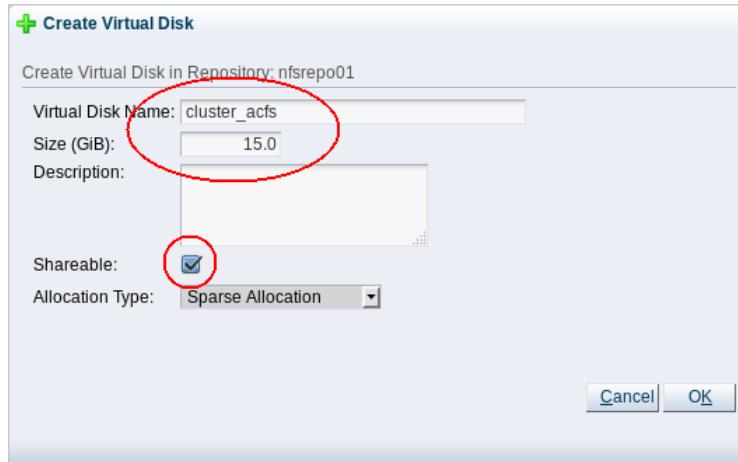
The screenshot shows the Oracle VM Manager interface with the 'Repositories' tab selected. On the left, a tree view shows 'nfsrepo01' expanded, with 'Virtual Disks' highlighted by a red oval. On the right, a table lists existing virtual disks. A yellow box highlights the 'Create Virtual Disk...' button, which is also circled in red. The table data is as follows:

| Name             | Used (GiB) | Max (GiB) | Shareable |
|------------------|------------|-----------|-----------|
| cluster_act      | 0.12       | 15.0      | Yes       |
| cluster_quorum   | 0.12       | 6.0       | Yes       |
| os_lvtest        | 6.5        | 13.0      | No        |
| os_vdb01         | 13.0       | 13.0      | No        |
| os_vdb02         | 7.39       | 13.0      | No        |
| sw_cluster_vdb01 | 14.36      | 15.0      | No        |
| sw_cluster_vdb02 | 12.55      | 15.0      | No        |

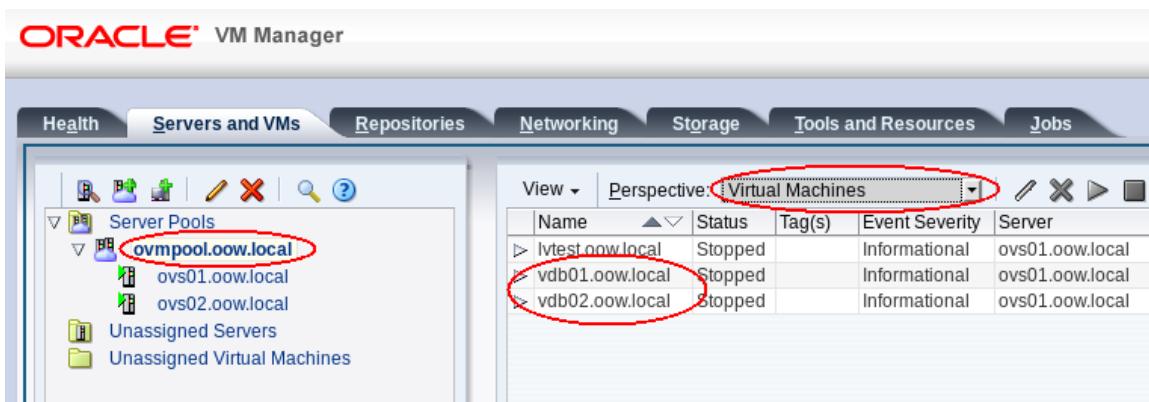
Add one **shared** disk of 6 GB named “**cluster\_quorum**” with the following details:

The dialog box is titled ‘Create Virtual Disk’ and shows the configuration for a new disk in ‘nfsrepo01’. The ‘Virtual Disk Name’ is set to ‘cluster\_quorum’, the ‘Size (GiB)’ is set to ‘6.0’, and the ‘Shareable’ checkbox is checked. The ‘Allocation Type’ is set to ‘Sparse Allocation’. The ‘OK’ button is visible at the bottom right.

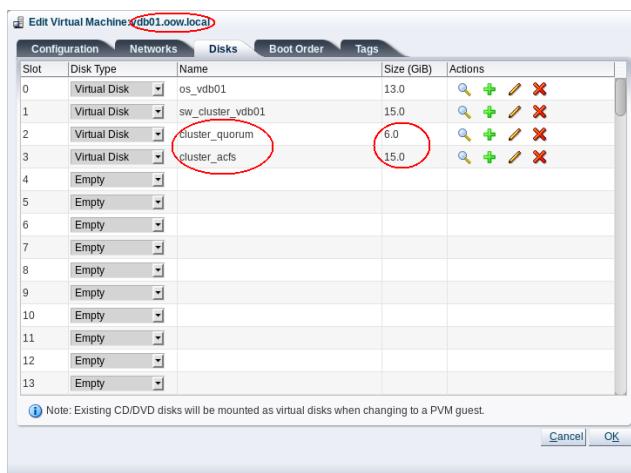
Add one **shared** disk of 15 GB named “**cluster\_acfs**” with the following details:



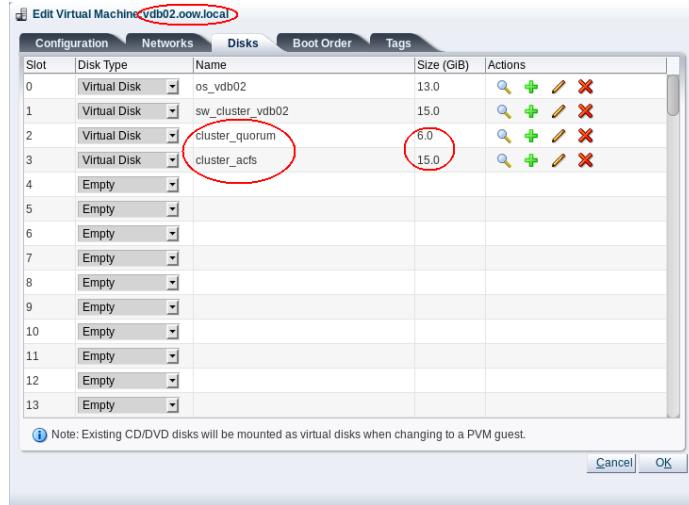
Present these two shared virtual disks to both **vdb0(x)** guests; select “**Server and VMs**” tab, expand and select your pool, select “**Virtual Machine**” perspective and edit both **vdb01/vdb02** guests:



Select and edit guest named “**vdb01.0ow.local**”:



Select and edit guest named “vdb02.oow.local”:



Start all the guests and supply the first configuration by “**Oracle VM Guest console**”.

| Name             | Status  | Tag(s) | Event Severity | Server          | Max. Memory (MB) | Memory (MB) | Max. Processors |
|------------------|---------|--------|----------------|-----------------|------------------|-------------|-----------------|
| lptest.oow.local | Running |        | Informational  | ovs01.oow.local | 256              | 256         | 1               |
| vdb01.oow.local  | Running |        | Informational  | ovs01.oow.local | 2560             | 2560        | 2               |
| vdb02.oow.local  | Stopped |        | Informational  | ovs01.oow.local | 1800             | 1800        | 2               |

Open the **console** of each guest.

| Name             | Status  | Tag(s) | Event Severity | Server          | Max. Memory (MB) | Memory (MB) | Max. Processors |
|------------------|---------|--------|----------------|-----------------|------------------|-------------|-----------------|
| lptest.oow.local | Running |        | Informational  | ovs01.oow.local | 256              | 256         | 1               |
| vdb01.oow.local  | Running |        | Informational  | ovs01.oow.local | 2560             | 2560        | 2               |
| vdb02.oow.local  | Stopped |        | Informational  | ovs01.oow.local | 1800             | 1800        | 2               |

### vdb01.oow.local details:

Hostname: **vdb01.oow.local**

Network device to configure: **eth0**

Activate interface on system boot: **yes**

Boot protocol: **static**

IP address for the interface: **192.168.56.204**

Netmask of the interface: **255.255.255.0**

Gateway IP address: **192.168.56.1**

DNS servers separated by comma: **8.8.8.8,192.168.56.1**

System Root Password: <password containing uppercase, minorcase and at least one number> ex **SimonCoter1**

```
Connected (encrypted) to: lvtest.oow.local
Ctrl Alt Ctrl-Alt-Del [OK]

Oracle VM Console

/boot: clean, 56/129024 files, 112569/524288 blocks
[OK]
Remounting root filesystem in read-write mode: EXT4-fs (xvda2): re-mounted. Opts: (null)
[OK]
Mounting local filesystems: EXT4-fs (xvda1): mounted filesystem with ordered data mode. Opts: (null)
[OK]
Enabling local filesystem quotas:
[OK]
Enabling /etc/fstab swaps: Adding 1571836k swap on /dev/xvda3. Priority:-1 extents:1 across:157183
6k SS
[OK]
Entering non-interactive startup
Welcome to Oracle Linux Server
Starting udev:
[OK]
Setting hostname lvtest.oow.local:
[OK]
Setting up Logical Volume Management: No volume groups found
[OK]
Checking filesystems
/: clean, 88225/684288 files, 1068547/2883584 blocks
/boot: recovering journal
/boot: clean, 56/129024 files, 112569/524288 blocks
[OK]
Remounting root filesystem in read-write mode:
[OK]
Mounting local filesystems:
[OK]
Enabling local filesystem quotas:
[OK]
Enabling /etc/fstab swaps:
[OK]
Entering non-interactive startup
Starting OUM template configure: network: System host name, e.g., "localhost.localdomain": vdb01.o
ow.local
network: Network device to configure, e.g., "eth0": eth0
network: Activate interface on system boot: yes or no.: yes
network: Boot protocol: dhcp or static.: static
network: IP address of the interface.: 192.168.56.204
network: Netmask of the interface.: 255.255.255.0
network: Gateway IP address.: 192.168.56.1
network: DNS servers separated by comma, e.g., "8.8.8.8,8.8.4.4": 8.8.8.8,192.168.56.1
authentication: System root password.:
```

**vdb02.oow.local** details:

Hostname: **vdb02.oow.local**

Network device to configure: **eth0**

Activate interface on system boot: **yes**

Boot protocol: **static**

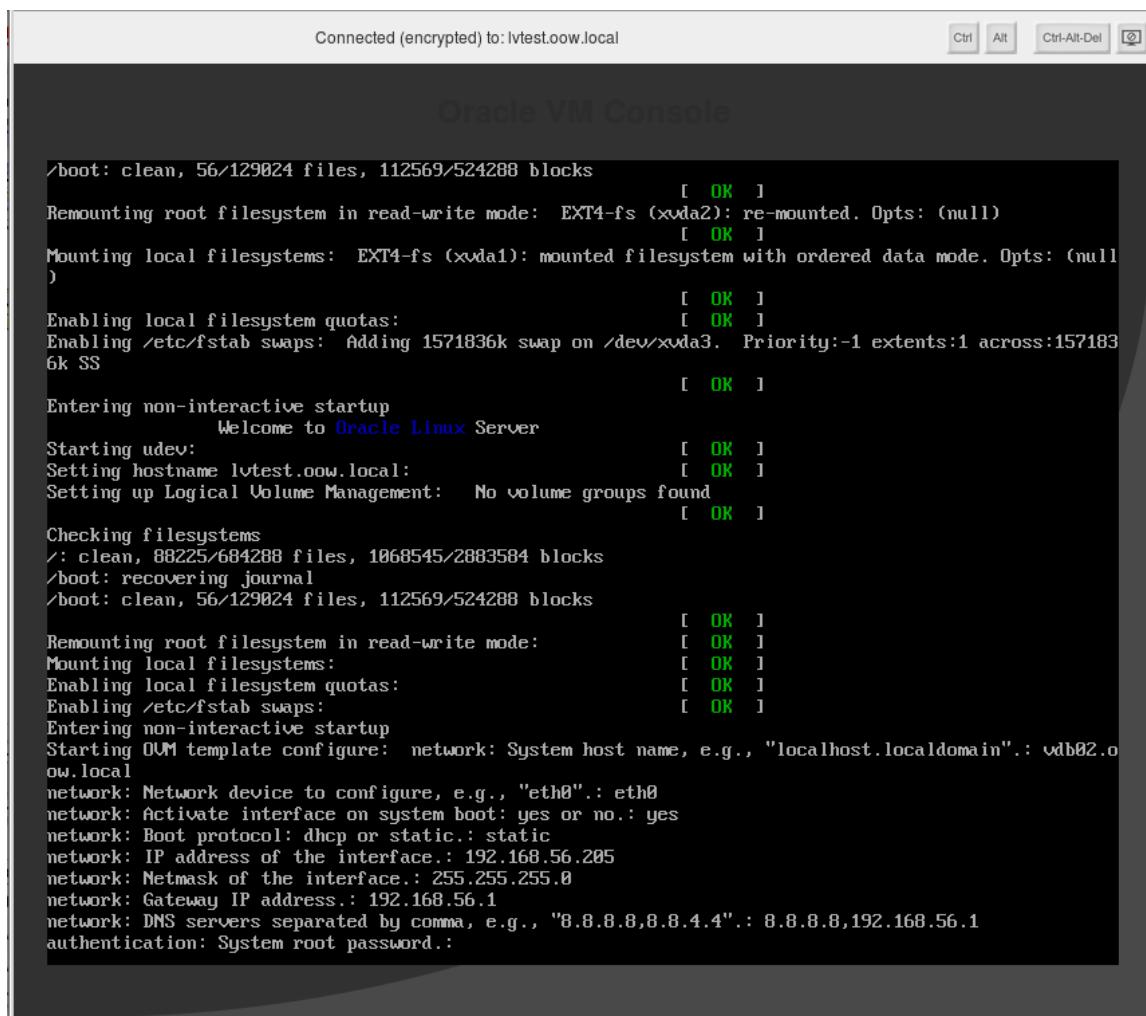
IP address for the interface: **192.168.56.205**

Netmask of the interface: **255.255.255.0**

Gateway IP address: **192.168.56.1**

DNS servers separated by comma: **8.8.8.8,192.168.56.1**

System Root Password: < password containing uppercase, minorcase and at least one number > ex **SimonCoter1**



The screenshot shows the Oracle VM Console window. At the top, it says "Connected (encrypted) to: lvtest.oow.local". Below that is the title "Oracle VM Console". The main area displays the terminal output of the Linux boot process for the vdb02.oow.local host. The output includes various system initialization steps such as mounting filesystems, setting up swap, and configuring networking (including the network card, IP address, and DNS). Each step is followed by a green "[ OK ]" status indicator. The text is in a monospaced font.

```
Connected (encrypted) to: lvtest.oow.local
Ctrl Alt Ctrl-Alt-Del

Oracle VM Console

/boot: clean, 56/129024 files, 112569/524288 blocks [OK]
Remounting root filesystem in read-write mode: EXT4-fs (xvda2): re-mounted. Opts: (null) [OK]
Mounting local filesystems: EXT4-fs (xvda1): mounted filesystem with ordered data mode. Opts: (null)
Enabling local filesystem quotas: [OK]
Enabling /etc/fstab swaps: Adding 1571836k swap on /dev/xvda3. Priority:-1 extents:1 across:157183
6k SS [OK]
Entering non-interactive startup
 Welcome to Oracle Linux Server
Starting udev: [OK]
Setting hostname lvtest.oow.local: [OK]
Setting up Logical Volume Management: No volume groups found [OK]
Checking filesystems
/: clean, 88225/684288 files, 1068545/2883584 blocks
/boot: recovering journal
/boot: clean, 56/129024 files, 112569/524288 blocks [OK]
Remounting root filesystem in read-write mode: [OK]
Mounting local filesystems: [OK]
Enabling local filesystem quotas: [OK]
Enabling /etc/fstab swaps: [OK]
Entering non-interactive startup
Starting OVM template configure: network: System host name, e.g., "localhost.localdomain": vdb02.o
ow.local
network: Network device to configure, e.g., "eth0": eth0
network: Activate interface on system boot: yes or no.: yes
network: Boot protocol: dhcp or static.: static
network: IP address of the interface.: 192.168.56.205
network: Netmask of the interface.: 255.255.255.0
network: Gateway IP address.: 192.168.56.1
network: DNS servers separated by comma, e.g., "8.8.8.8,8.8.4.4": 8.8.8.8,192.168.56.1
authentication: System root password.:
```

## lvtest.oow.local details:

Hostname:**lvtest.oow.local**

Network device to configure: **eth0**

Activate interface on system boot: **yes**

Boot protocol: **static**

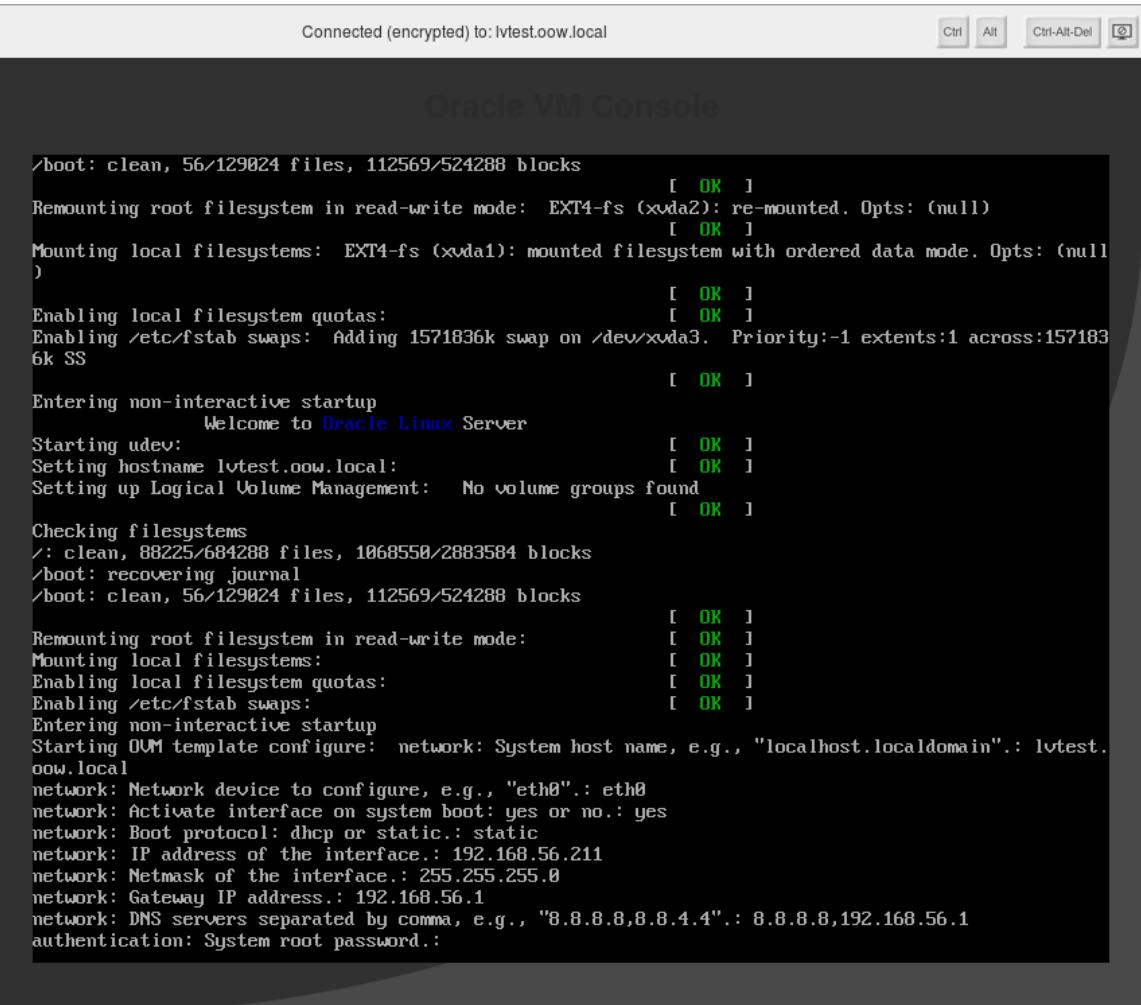
IP address for the interface: **192.168.56.211**

Netmask of the interface: **255.255.255.0**

Gateway IP address: **192.168.56.1**

DNS servers separated by comma: **8.8.8.8,192.168.56.1**

System Root Password: <password containing uppercase, minorcase and at least one number> ex **SimonCoter1**



The screenshot shows the Oracle VM Console window. At the top, it says "Connected (encrypted) to: lvtest.oow.local". Below that is the title "Oracle VM Console". The main area displays the terminal output of the Linux boot process. The output shows the system performing various tasks such as mounting filesystems, setting up swap, and configuring network settings. Most commands are followed by "[ OK ]" indicating successful execution. The output ends with the configuration of the "network" parameters.

```
Connected (encrypted) to: lvtest.oow.local
Ctrl Alt Ctrl-Alt-Del

Oracle VM Console

/boot: clean, 56/129024 files, 112569/524288 blocks
[OK]
Remounting root filesystem in read-write mode: EXT4-fs (xvda2): re-mounted. Opts: (null)
[OK]
Mounting local filesystems: EXT4-fs (xvda1): mounted filesystem with ordered data mode. Opts: (null)
[OK]
Enabling local filesystem quotas:
[OK]
Enabling /etc/fstab swaps: Adding 1571836k swap on /dev/xvda3. Priority:-1 extents:1 across:157183
6k SS
[OK]
Entering non-interactive startup
 Welcome to Oracle Linux Server
Starting udev:
[OK]
Setting hostname lvtest.oow.local:
[OK]
Setting up Logical Volume Management: No volume groups found
[OK]
Checking filesystems
/: clean, 88225/684288 files, 1068550/2883584 blocks
/boot: recovering journal
/boot: clean, 56/129024 files, 112569/524288 blocks
[OK]
Remounting root filesystem in read-write mode:
[OK]
Mounting local filesystems:
[OK]
Enabling local filesystem quotas:
[OK]
Enabling /etc/fstab swaps:
[OK]
Entering non-interactive startup
Starting OVM template configure: network: System host name, e.g., "localhost.localdomain": lvtest.
oow.local
network: Network device to configure, e.g., "eth0": eth0
network: Activate interface on system boot: yes or no.: yes
network: Boot protocol: dhcp or static.: static
network: IP address of the interface.: 192.168.56.211
network: Netmask of the interface.: 255.255.255.0
network: Gateway IP address.: 192.168.56.1
network: DNS servers separated by comma, e.g., "8.8.8.8,8.8.4.4": 8.8.8.8,192.168.56.1
authentication: System root password:
```

Connect to guests “vdb01.oow.local” and “vdb02.oow.local” and apply following changes:

### Network

#### **vdb01.oow.local**

Under path “/etc/sysconfig/network-scripts” create the following files:

```
Ifcfg-eth1
#####
NETMASK=255.255.255.0
IPADDR=172.28.28.1
BOOTPROTO=static
ONBOOT=yes
NM_CONTROLLED=no
DEVICE=eth1
IPV6INIT=no
```

```
Ifcfg-eth2
#####
DNS2=8.8.8.8
DNS1=192.168.56.1
BOOTPROTO=dhcp
ONBOOT=yes
NM_CONTROLLED=no
DEVICE=eth2
IPV6INIT=no
```

After that execute the following command:

```
nohup service network restart &
```

and verify that all ip addresses are up and running.

#### **vdb02.oow.local**

Under path “/etc/sysconfig/network-scripts” create the following files:

```
Ifcfg-eth1
#####
NETMASK=255.255.255.0
IPADDR=172.28.28.2
BOOTPROTO=static
ONBOOT=yes
NM_CONTROLLED=no
DEVICE=eth1
IPV6INIT=no
```

```
Ifcfg-eth2
#####
DNS2=8.8.8.8
DNS1=192.168.56.1
BOOTPROTO=dhcp
ONBOOT=yes
NM_CONTROLLED=no
DEVICE=eth2
IPV6INIT=no
```

After that execute the following command:

```
nohup service network restart &
```

and verify that all ip addresses are up and running.

Add the following entries in “**/etc/hosts**” of both nodes:

```
vm guest ip addresses
192.168.56.204 vdb01.oow.local vdb01
192.168.56.205 vdb02.oow.local vdb02
192.168.56.206 vdb01-vip.oow.local vdb01-vip
192.168.56.207 vdb02-vip.oow.local vdb02-vip
192.168.56.208 vdb-scan.oow.local vdb-scan
192.168.56.210 mysql.oow.local mysql
192.168.56.211 lvttest.oow.local lvttest
host ip addresses
192.168.56.200 ovmm.oow.local ovmm
192.168.56.201 ovs01.oow.local ovs01
192.168.56.202 ovs02.oow.local ovs02
192.168.56.202 ovmpool.oow.local ovmpool
```

## Packages

Execute on both nodes ( **vdb01** and **vdb02** ) the following command\*\*\*:

```
yum install oracle-rdbms-server-12cR1-preinstall oracleasm-support -y
```

\*\*\* your laptop need an internet connection to allow both guests download that packages.

## Selinux and firewall ( iptables )

On both nodes ( **vdb01** and **vdb02** ):

- Verify or edit the file “**/etc/selinux/config**” and verify that its flag is set, at least, as “**permissive**”.
- Stop and disable “**iptables**” service with the following commands:  
# service iptables stop  
# chkconfig iptables off

## NTPD configuration

Edit on both nodes (**vdb01** and **vdb02**) the file “**/etc/sysconfig/ntpd**” and add the option “**-x**”:

```
OPTIONS="-x -u ntp:ntp -p /var/run/ntp.pid"
```

Restart **ntpd** service:

```
service ntpd restart
```

## Create filesystem /u01 on both nodes ( vdb01 and vdb02 )

The filesystem will be created on the second local-disk of 15 GB size, with the following commands:

```
fdisk /dev/xvdb1 (options => n, p, 1, <return>, <return>)
mkfs.ext4 /dev/xvdb1
```

## Mount filesystem “/u01” on both nodes ( vdb01 and vdb02 )

Edit file “**/etc/fstab**” and insert the following line:

```
/dev/xvdb1 /u01 ext4 defaults 0 0
```

Mount the filesystem and verify that it's mounted:

```
mount -a
```

```
df -k
```

Output example of “df -k”:

```
[root@vdb01 ~]# df -k
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/xvda2 11362480 4207728 6593872 39% /
tmpfs 8388608 647796 7740812 8% /dev/shm
/dev/xvda1 508004 100015 386090 21% /boot
/dev/xvdb1 15480816 10597608 4096828 73% /u01
```

#### Prepare “oracleasm” disks to be used by Oracle Clusterware

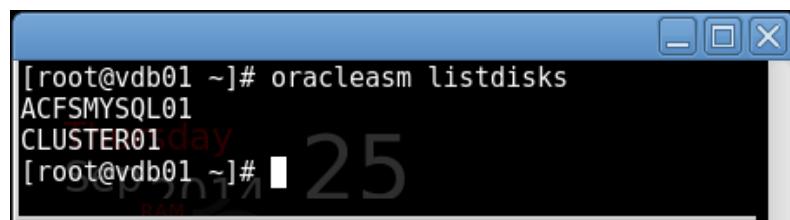
As “root” on first server (vdb01) execute:

```
fdisk /dev/xvdc1 (options n, p, 1, <return>, <return>)
fdisk /dev/xvdd1 (options n, p, 1, <return>, <return>)
oracleasm createdisk CLUSTER01 /dev/xvdc1
oracleasm createdisk ACFSMySQL01 /dev/xvdd1
```

As “root” on second server (vdb02) execute:

```
partprobe /dev/xvdc
partprobe /dev/xvdd
oracleasm scandisks
```

Check on **both nodes** the output of the command “**oracleasm listdisks**”:



#### Set password for user “oracle” on nodes “vdb01” and “vdb02”.

On both nodes, as “root” execute:

```
passwd oracle
* choose “oracle” as default password
```

#### Environment files

As “oracle” user edit the file “/home/oracle/.bash\_profile” and add the following envs:

```
#####
vdb01.oow.local
#####
User specific environment and startup programs
ORACLE_HOME=/u01/cluster/12.1.0/grid
ORACLE_BASE=/u01/app/oracle
ORACLE_SID=+ASM1
XAG_HOME=/u01/cluster/12.1.0/xag
PATH=$XAG_HOME/bin:$ORACLE_HOME/bin:$PATH:$HOME/bin

export ORACLE_BASE ORACLE_HOME ORACLE_SID PATH
#####

vdb02.oow.local
```

```
#####
User specific environment and startup programs
ORACLE_HOME=/u01/cluster/12.1.0/grid
ORACLE_BASE=/u01/app/oracle
ORACLE_SID=+ASM2
XAG_HOME=/u01/cluster/12.1.0/xag
PATH=$XAG_HOME/bin:$ORACLE_HOME/bin:$PATH:$HOME/bin

export ORACLE_BASE ORACLE_HOME ORACLE_SID PATH
```

By “**Oracle VM Manager**” reboot your guests and verify that every change ( network, filesystem, firewall, etc ) is correctly working.



| Name              | Status  | Tag(s) | Event Severity | Server          | Max. Mem | Memory (MB) | Max. Processor |
|-------------------|---------|--------|----------------|-----------------|----------|-------------|----------------|
| lvttest.oow.local | Running |        | Informational  | ovs01.oow.local | 256      | 56          | 1              |
| vdb01.oow.local   | Running |        | Informational  | ovs01.oow.local | 2560     | 2560        | 2              |
| vdb02.oow.local   | Stopped |        | Informational  | ovs01.oow.local | 1800     | 1800        | 2              |

Download from your laptop the software(s) at the following URL:

<http://www.oracle.com/technetwork/database/database-technologies/clusterware/downloads/index.html>

Download Oracle Grid Infrastructure Rel. 12.1.0.1

[Download Oracle Grid Infrastructure Standalone Agents](#)

And put the software on your virtual-guest “**vdb01.oow.local**”.

**Install Oracle Grid Infrastructure 12c on nodes “vdb01.oow.local” and “vdb02.oow.local”**

To install you need a graphical interface; to do that execute the following steps.

1. Install a vnc-server on your server “vdb01.oow.local”.

```
yum install xorg-x11-utils.x86_64 xorg-x11-server-Xorg.x86_64 xorg-x11-server-utils.x86_64 -y
yum install tigervnc-server.x86_64 -y
```

2. Start a vnc-server as user “**oracle**” and supply a vnc-password.

```
vncserver :1
```

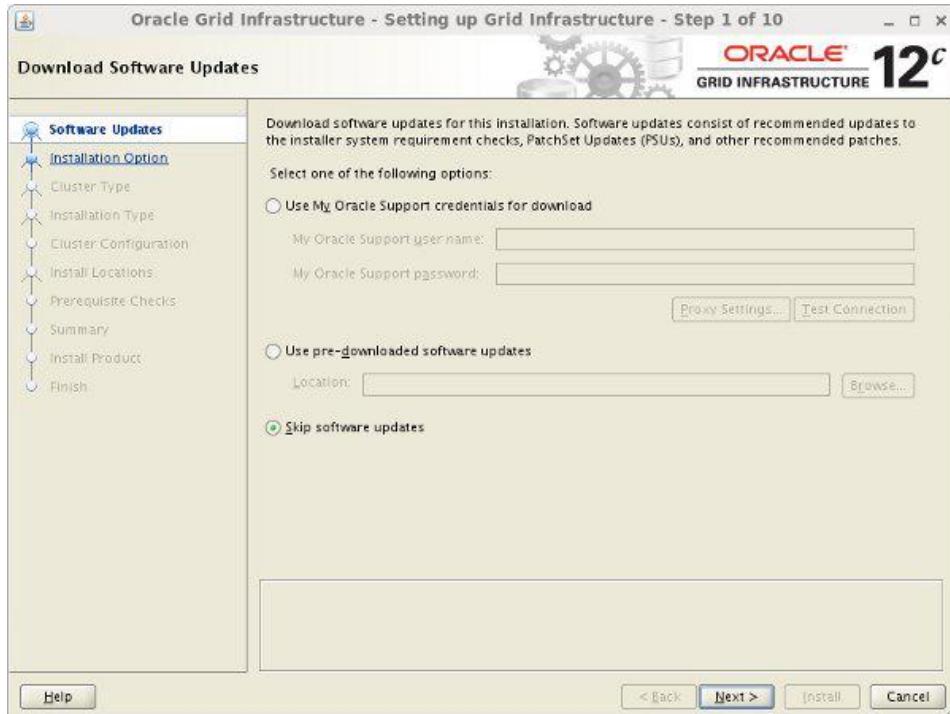
3. Connect, from your laptop, to your vnc-session

```
vncviewer 192.168.56.204:1
```

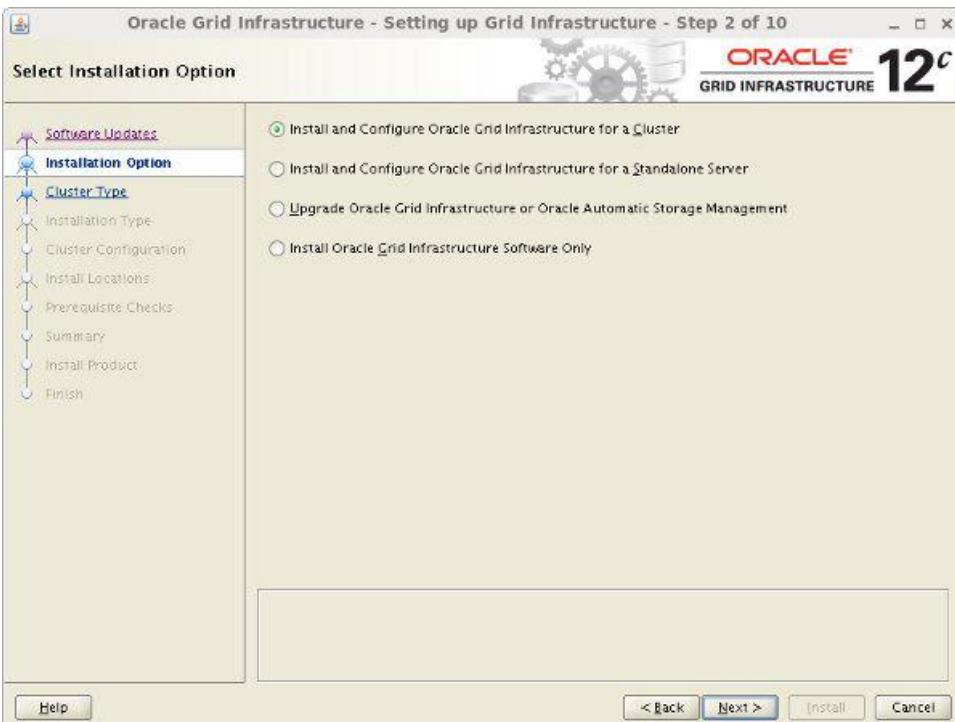
Unzip the software downloaded, change directory to the software install path and execute

```
./runInstaller
```

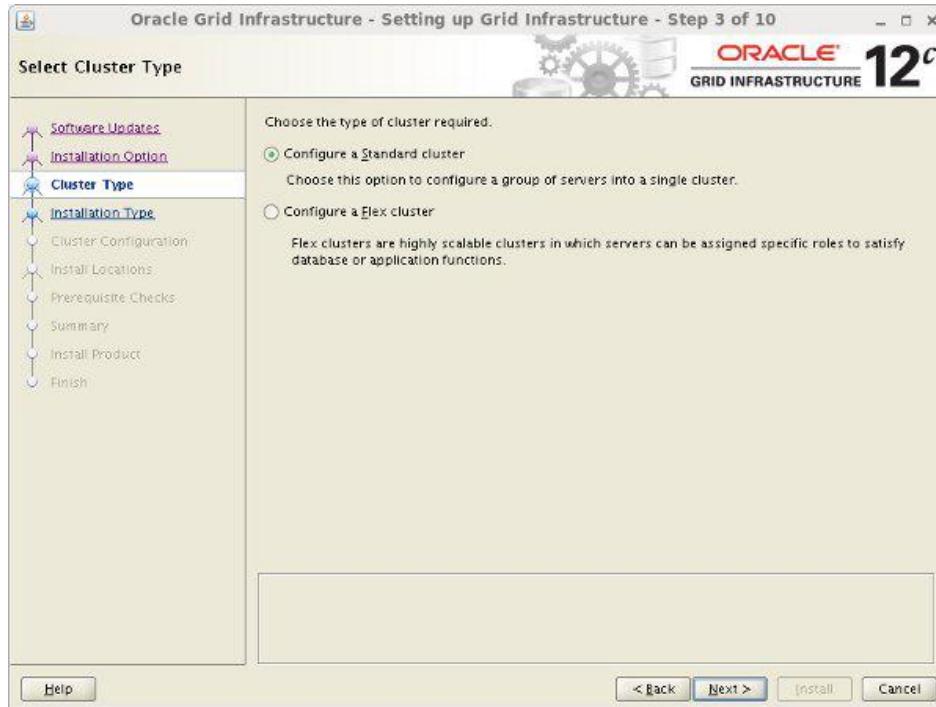
1. Select the “Skip software updates” option, then click the “Next” button.



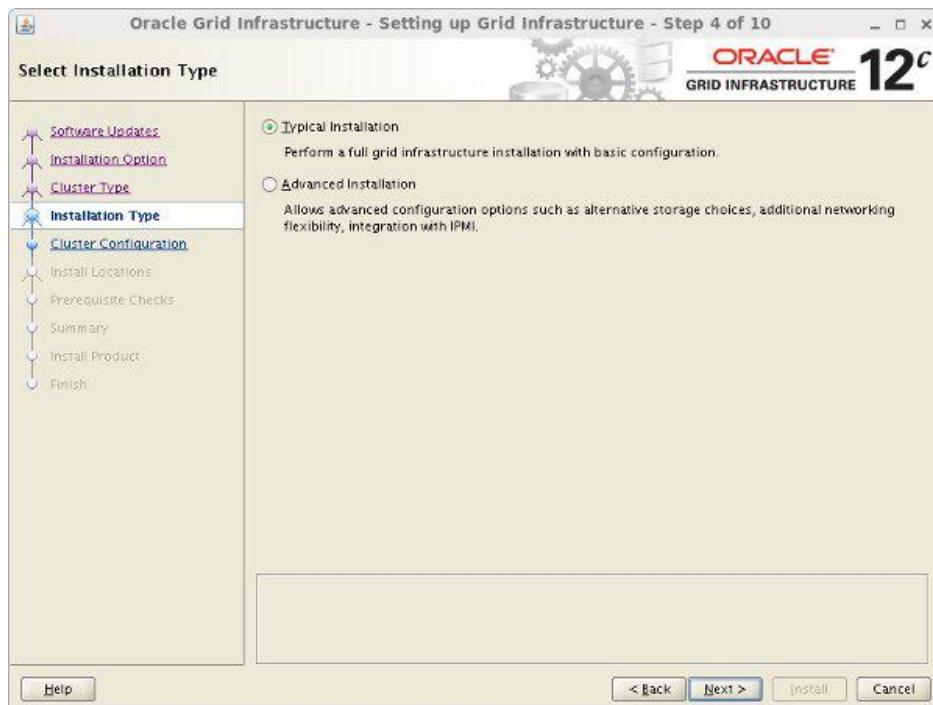
2. Select the "Install and Configure Oracle Grid Infrastructure for a Cluster".



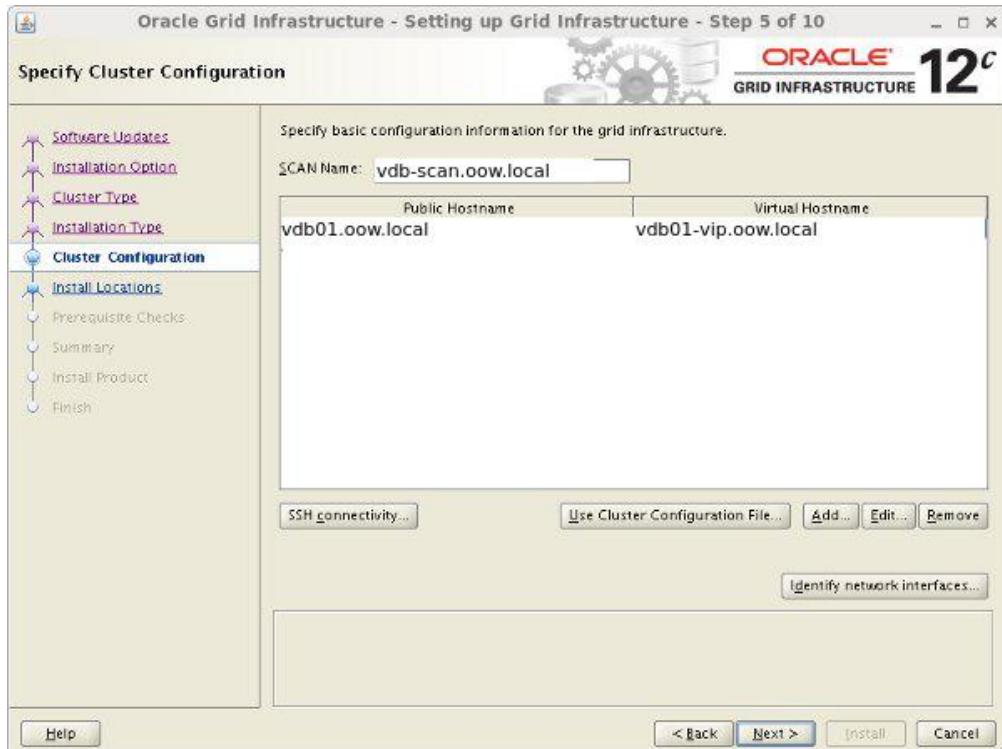
3. Accept the "Configure a Standard cluster" option by clicking the "Next" button.



4. Select the "Typical Installation" option, then click the "Next" button.



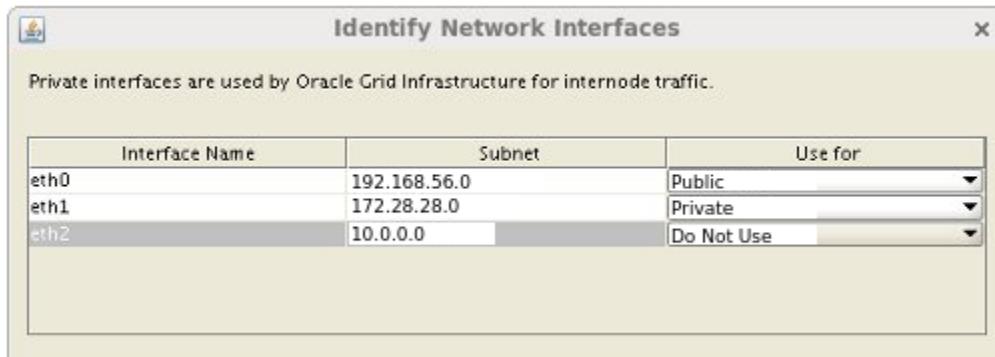
- On the "Specify Cluster Configuration" screen, enter the correct **SCAN Name** and click the "Add" button to add the second node information.



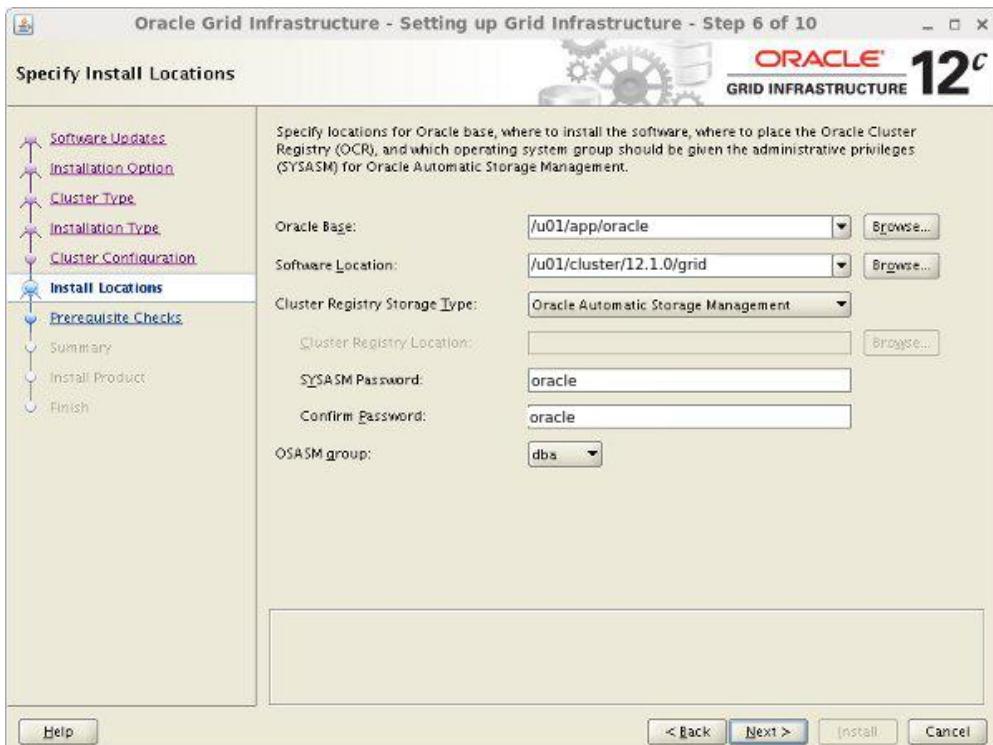
- Enter the details of the **second node** in the cluster, then click the "OK" button.



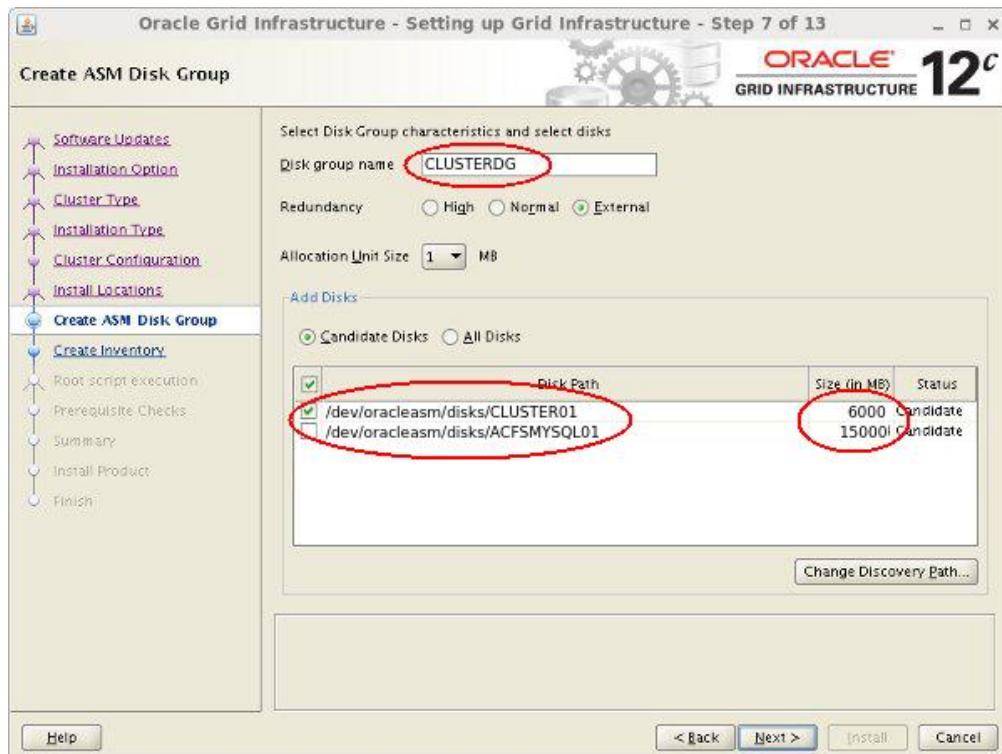
- Click the "SSH Connectivity..." button and enter the password for the "oracle" user. Click the "Setup" button to configure **SSH connectivity**, and the "Test" button to test it once it is complete.
- Click the "Identify network interfaces..." button and check the public and private networks are specified correctly. Remember to mark the **NAT interface as "Do Not Use"**. Once everything is good, click the "OK" button and the "Next" button on the previous screen.



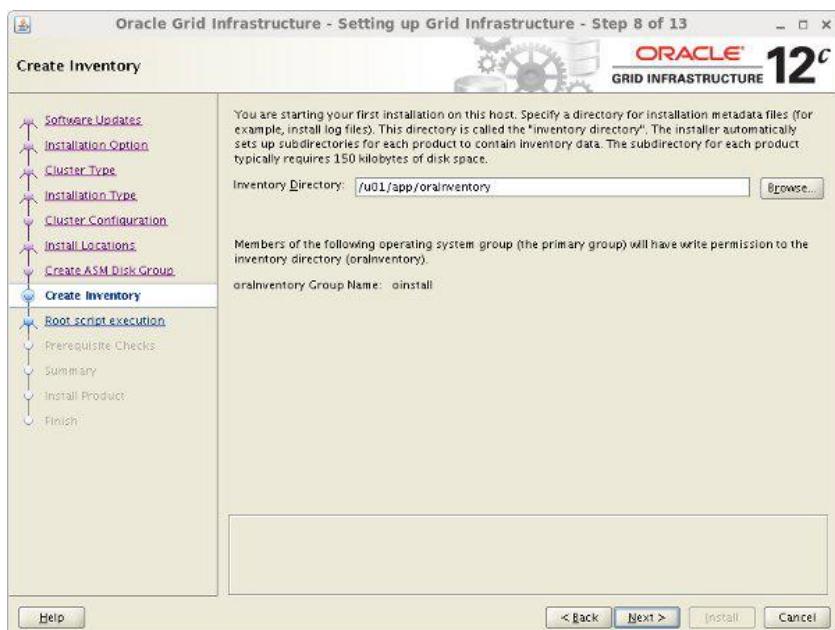
- Enter "/u01/cluster/12.1.0.1/grid" as the software location and "Automatic Storage Manager" as the cluster registry storage type. Enter the **ASM password** (oracle), select "dba" as the group and click the "Next" button.



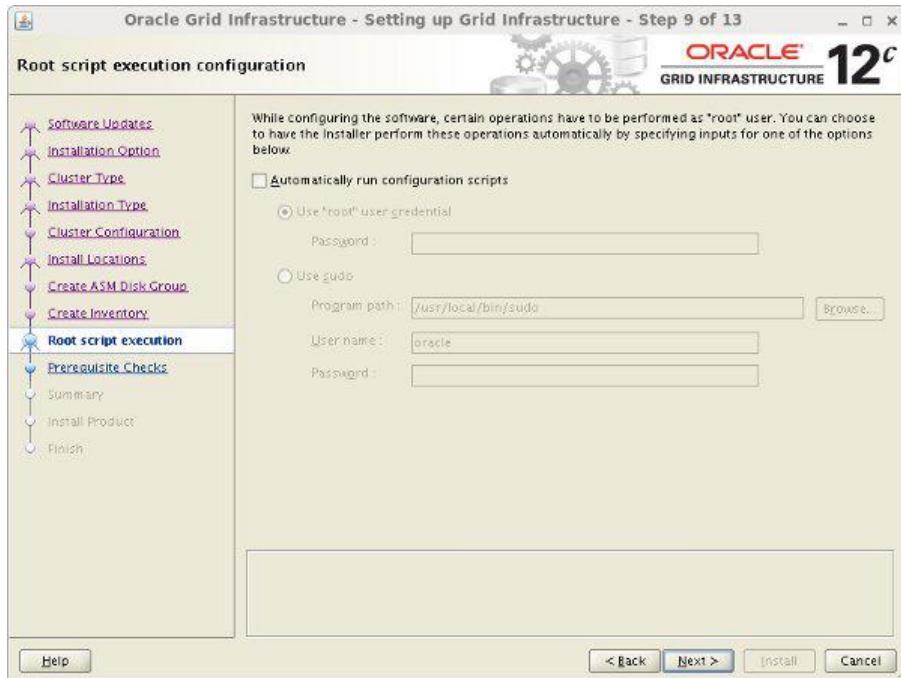
- Set the redundancy to "External", click the "Change Discovery Path" button and set the path to "/dev/oracleasm/disks/\*". Return the main screen and select the disk named "**CLUSTER01**".



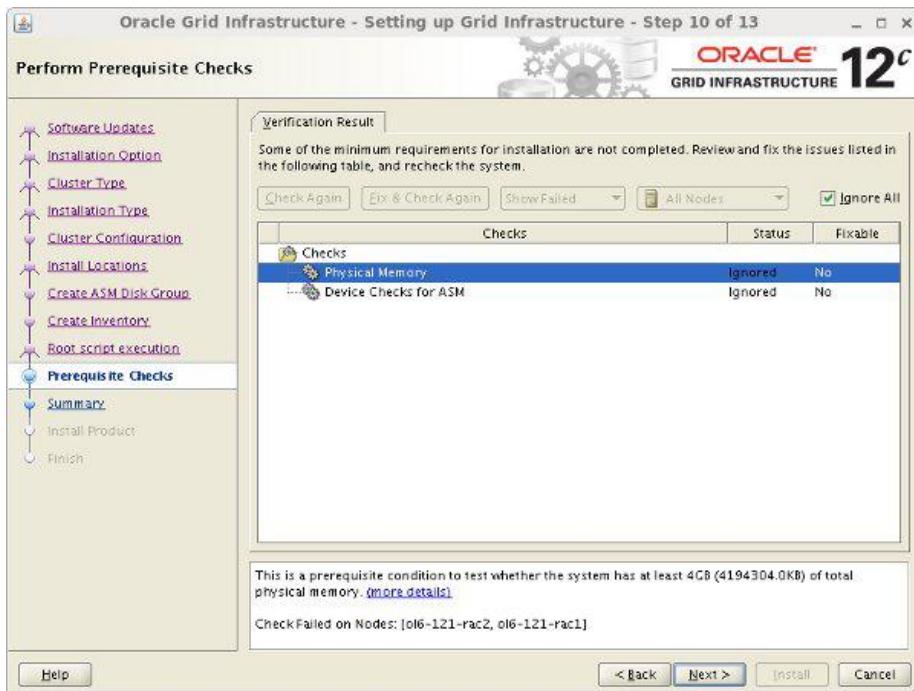
11. Accept the **default inventory** directory by clicking the "Next" button.



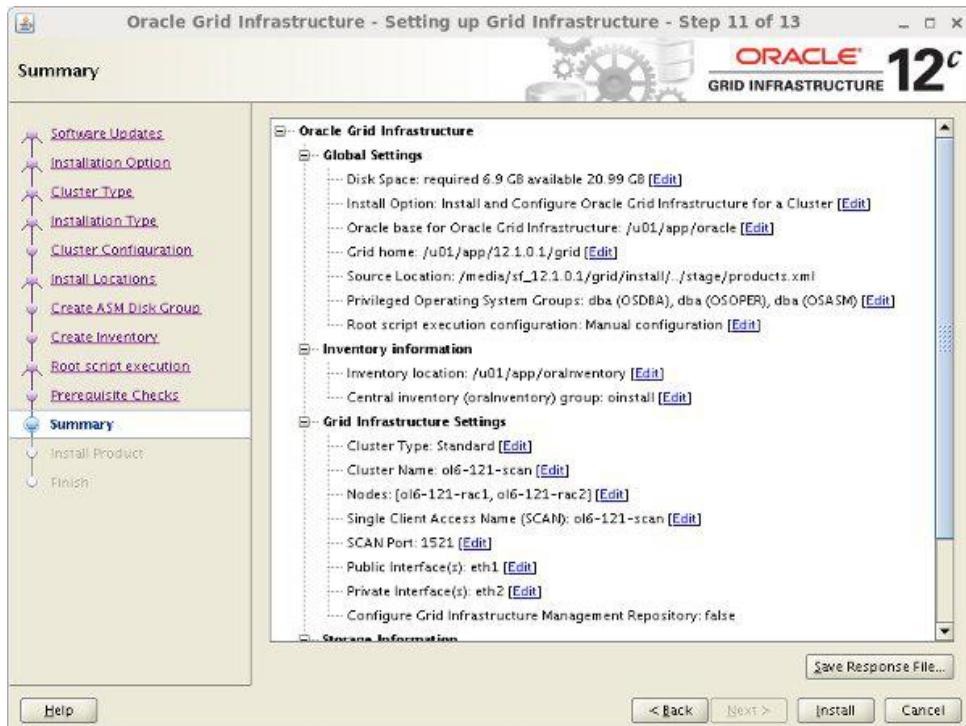
12. Select to run "**root scripts**" manually and click the "Next" button.



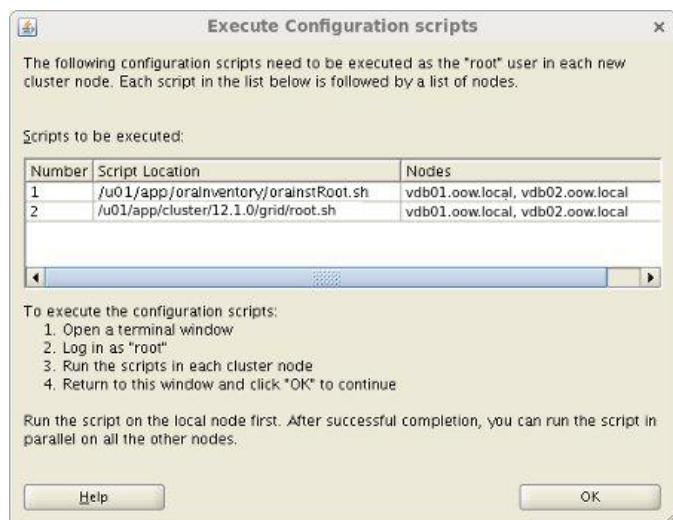
- Wait while the prerequisite checks complete. If you have any issues, either fix them or check the "**Ignore All**" checkbox and click the "**Next**" button. It is likely the "**Physical Memory**" and "**Device Checks for ASM**" and "**Swap Size**" tests will fail for this type of installation.



14. If you are happy with the **summary information**, click the "Install" button.

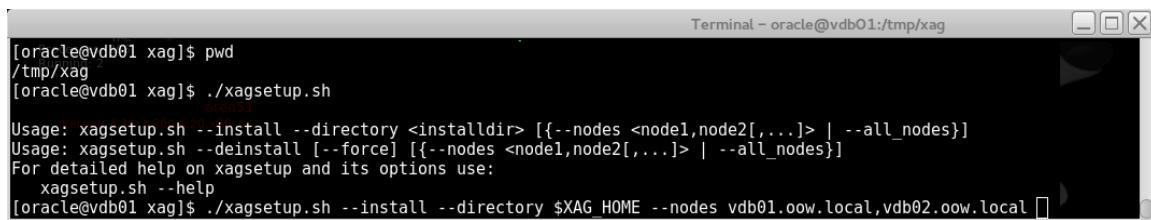


15. When prompted, run the **configuration scripts** on each node.



## Install Oracle Grid Infrastructure standalone Agents on clustered nodes

1. Connect on node “**vdb01.oow.local**” and **unzip “xagpack\_5.zip”** file under a temporary directory
2. As “**oracle**” user create the directory that will guest “Standalone Agents” on both nodes:  
**# mkdir \$XAG\_HOME**
3. Install on the cluster the “standalone Agents”:  
**# cd <xag\_unzipped\_folder>**  
**# ./xagsetup.sh --install --directory \$XAG\_HOME --nodes vdb01.oow.local, vdb02.oow.local**



A terminal window titled "Terminal - oracle@vdb01:/tmp/xag". The window shows the following command being run:

```
[oracle@vdb01 xag]$./xagsetup.sh --install --directory $XAG_HOME --nodes vdb01.oow.local, vdb02.oow.local
```

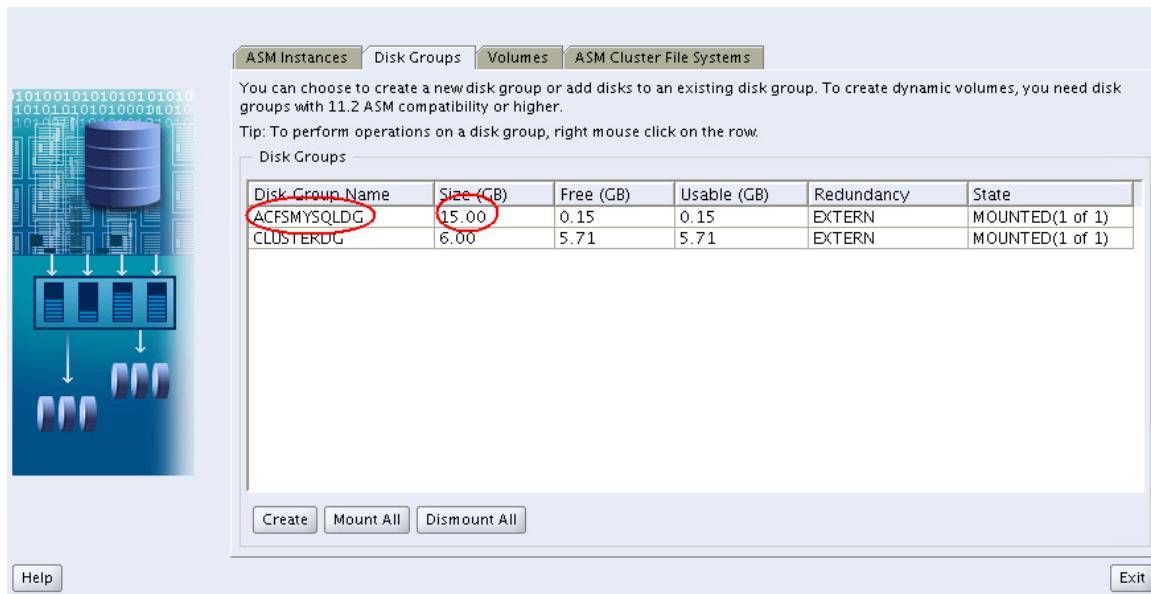
The output of the command is displayed in the terminal window.

## Download and install Oracle MySQL 5.6 Community Edition

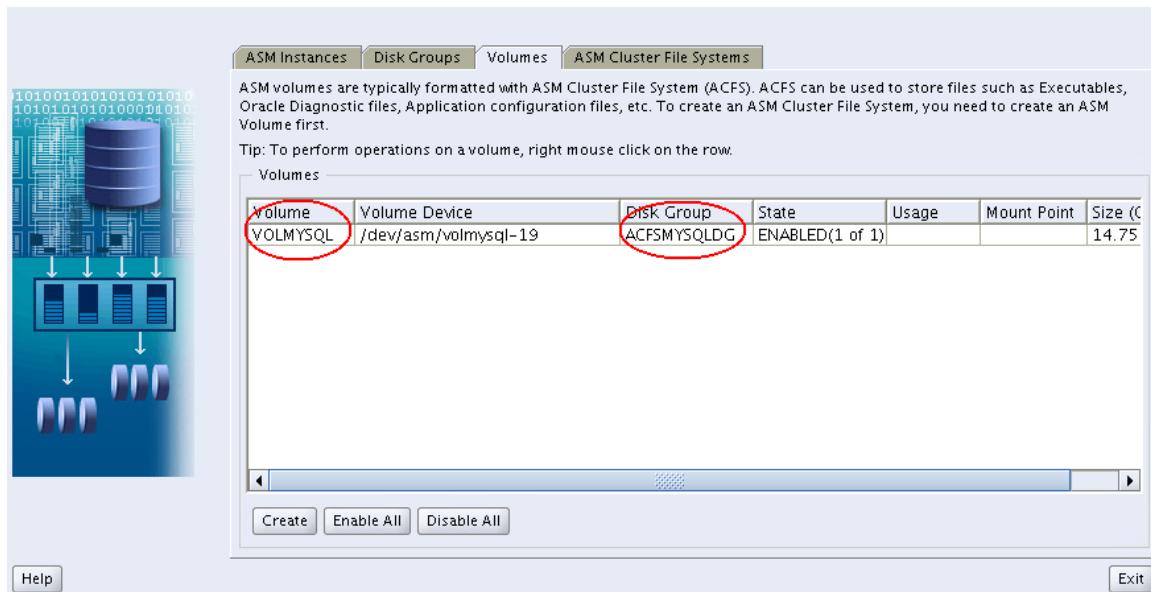
1. Download and install “**Oracle MySQL yum repository**” configuration rpm ( you need an Oracle Web Account ) at the following link:  
<http://dev.mysql.com/downloads/file.php?id=450542>
2. Install the **Yum configuration file for Oracle MySQL 5.6** on both nodes ( **vdb01** and **vdb02** ):  
**# rpm -ihv mysql-community-release-el6-5.noarch.rpm**
3. Install “**MySQL**” and “**rpm packages**” on both clustered nodes ( **vdb01** and **vdb02** ) that will be used for the web-app.  
**# yum install httpd httpd-tools php php-mysql php-gd mysql-server php-xml mysql -y**

## Create ASM Clustered Filesystem (ACFS) for application-tier

1. Connect to the vnc session opened with “oracle” account.  
**# vncviewer 192.168.56.201:1**
2. Launch command “asmca” ASM Creation Assistant.  
**# asmca**
3. Create a new diskgroup named “ACFSMYSQLDG” using the 15 GB disk named “ACFSMYSQLO1”



4. Create a new Volume of 14.75 GB named “VOLMYSQL” using the diskgroup created at step #3



5. Create a new **ACFS clustered filesystem** with the following details

Creating the ASM Cluster File System creates the on-disk structure. Use Cluster File System for Oracle Database home or datafiles. Node Local File System can be used to store Oracle Diagnostic Files, Application Files etc.

Tip: Choose an existing volume device or create a new volume by choosing Create Volume in the Volumes tab.

|               |                                     |
|---------------|-------------------------------------|
| Type of ACFS  | Cluster File System                 |
| Mount Point   | /mysql                              |
| Auto Mount    | <input checked="" type="checkbox"/> |
| Mount Options |                                     |
| User Name     | oracle                              |
| Group Name    | dba                                 |
| Description   |                                     |
| Select Volume | VOLMYSQL                            |

**OK** **Show Command** **Cancel** **Help**

**Apply custom changes to the environment for the future resource clustering**

Execute the following commands as “root”:

Node “vdb01.oow.local”

```
mv /etc/httpd /mysql
ln -s /mysql/httpd /etc/httpd
mv /etc/my.cnf /mysql
ln -s /mysql/my.cnf /etc/my.cnf
```

Node “vdb02.oow.local”

```
mv /etc/httpd /etc/httpd.old
ln -s /mysql/httpd /etc/httpd
mv /etc/my.cnf /etc/my.cnf.old
ln -s /mysql/my.cnf /etc/my.cnf
```

On node “vdb01.oow.local” create a demo MySQL database

```
mysqladmin -u root -p create myapp
```

Nb: verify that the service “mysqld” had been started, otherwise start it.

```
service mysqld status
service mysqld start (if not already started)
```

## On node “vdb01.oow.local” install a demo web-app

1. Log into MySQL monitor:  
`# mysql -u root -p`
2. Create the wiki user:  
`# CREATE USER 'wiki'@'localhost' IDENTIFIED BY 'wiki';`
3. Create MySQL database dedicated to mediawiki.  
`# CREATE DATABASE myapp;`  
`# SHOW DATABASES;`
4. Grant privileges:  
`# GRANT ALL PRIVILEGES ON myapp.* TO 'wiki'@'localhost' IDENTIFIED BY 'wiki' WITH GRANT OPTION;`  
`# FLUSH PRIVILEGES;`  
`# exit`
5. Manually start “**httpd**” daemon on node “**vdb01.oow.local**”  
`# service httpd start`
6. As “**root**” download “mediawiki” package and move it under “**/mysql**” filesystem.  
`# cd /tmp`  
`# wget http://releases.wikimedia.org/mediawiki/1.23/mediawiki-1.23.4.tar.gz`  
`# mkdir -p /mysql/html`  
`# cd /mysql/html`  
`# tar xvzf /tmp/mediawiki-1.23.4.tar.gz`  
`# mv mediawiki-1.23.4 wiki`
7. On both nodes ( **vdb01** and **vdb02** ) create the following symbolic links:  
`# ln -s /mysql/html/wiki /var/www/html/wiki`
8. Connect to the following URL and proceed with the “Wikimedia” configuration.  
  
<http://192.168.56.204/wiki>  
**nb: you have to supply all the details of the MySQL database created above.**
9. When the “**wikimedia**” setup is completed stop both “**MySQL**” and “**httpd**” services.  
`# service httpd stop`  
`# service mysqld stop`

## Create an user app-vip clustered with Oracle Grid Infrastructure 12c

This VIP ( Virtual IP ) will be used for both MySQL and WebApp.

To create the clustered VIP resource execute the following commands:

```
export ORACLE_HOME=/app/cluster/12.1.0/grid
export PATH=$ORACLE_HOME/bin:$PATH
appvipcfg create -network=1 -ip 192.168.56.210 -vipname mysql.oow.local -user=oracle
```

To start the clustered VIP execute the following command as “**oracle**” :

```
crsctl start resource mysql.oow.local
```

## Configure MySQL to be managed by Oracle Clusterware 12c

1. Create a dedicated account for Clusterware monitoring agent.

```
service mysqld start
mysql -u root -p
CREATE USER oracle@localhost IDENTIFIED BY 'oracle';
```

2. Edit “/mysql/my.cnf” file and add a section for “mysqladmin”:

```
[mysqladmin]
user=oracle
password=oracle
```

3. Stop “MySQL” daemon  
# service mysqld stop

4. Add MySQL database “myapp” as clustered resource by Oracle Clusterware 12c  
As “root”, execute the following commands:

```
. /home/oracle/.bash_profile
agctl add mysql_server myapp --mysql_home /usr --datadir /mysql --mysql_type MYSQL --
mysql_lib /mysql --vip_name mysql.oow.local
```

## Configure Apache “httpd” to be managed by Oracle Clusterware 12c

1. Create a resource-script to manage httpd-daemon, as “root”:

```
mkdir /mysql/cluster_scripts
vi /mysql/cluster_scripts/apache.sh
content of the script:

#!/bin/bash

ORA CRS_HOME=/u01/cluster/12.1.0/grid
CRS_HOME_SCRIPT=/mysql/cluster_scripts
HTTP_VIP=192.168.56.210
HTTP_PORT=80
ORACLE_BASE=/app/em
ORACLE_USER=oracle

See how we were called.
case "$1" in
 start)
 /etc/init.d/httpd start
 ;;
 stop)
 /etc/init.d/httpd stop
 ;;
 clean)
 killall -u apache
 sleep 2
 killall -u apache
 ;;
 check)
 nc -z $HTTP_VIP $HTTP_PORT
 num=`echo $?`
 exit $num
 ;;
 restart)
```

```

stop
sleep 3
start
;;
*)
echo $"Usage: $0 {start|stop|clean|restart|check}"
exit 1
esac
#####

```

- As “root” add a custom resource to Oracle Clusterware to manage “**httpd daemon**”.

Create the configuration file for the resource with the following details:

```

vi /app/cluster/12.1.0/grid/crs/public/httpd.myapp.mysql.attr
#####

NAME=httpd.myapp.mysql
TYPE=cluster_resource
ACL=owner:root:rwx,pgrp:root:r-x,other::r--,user:oracle:r-x
ACTIONS=
ACTION_SCRIPT=/mysql/cluster_scripts/apache.sh
ACTION_TIMEOUT=60
ACTIVE_PLACEMENT=0
AGENT_FILENAME=%CRS_HOME%/bin/scriptagent
AUTO_START=restore
CARDINALITY=1
CHECK_INTERVAL=60
CHECK_TIMEOUT=0
CLEAN_TIMEOUT=60
DEGREE=1
DELETE_TIMEOUT=60
DESCRIPTION=
ENABLED=1
FAILOVER_DELAY=0
FAILURE_INTERVAL=0
FAILURE_THRESHOLD=0
HOSTING_MEMBERS=vdb01 vdb02
INSTANCE_FAILOVER=1
INTERMEDIATE_TIMEOUT=0
LOAD=1
LOGGING_LEVEL=1
MODIFY_TIMEOUT=60
OFFLINE_CHECK_INTERVAL=0
PLACEMENT=restricted
RELOCATE_BY_DEPENDENCY=1
RESTART_ATTEMPTS=3
SCRIPT_TIMEOUT=60
SERVER_CATEGORY=
SERVER_POOLS=
START_CONCURRENCY=0
START_DEPENDENCIES=hard(xag.myapp.mysql)
START_TIMEOUT=0
STOP_CONCURRENCY=0
STOP_DEPENDENCIES=hard(xag.myapp.mysql)
STOP_TIMEOUT=0
UPTIME_THRESHOLD=1h
USER_WORKLOAD=no
USE_STICKINESS=0
#####

```

As "root" add the resource to the Oracle Clusterware configuration with the following command.

```
crsctl add resource httpd.myapp.mysql -type cluster_resource -file /app/cluster/12.1.0/grid/crs/public/httpd.myapp.mysql.attr
```

#### Configure Clusterware resources permission(s) to allow "oracle" account to manage them

Use the following syntax to grant correct permission(s):

```
crsctl setperm resource mysql.00w.local -u user:oracle:r-x
crsctl setperm resource xag.myapp.mysql -u user:oracle:r-x
crsctl setperm resource httpd.myapp.mysql -u user:oracle:r-x
```

#### Add a wrapped script to see clusterware resources status

The standard command to see resources status is:

```
crsctl status resource
crsctl status resource -t
```

Create the following script under path "**/usr/local/bin**" on both nodes ( **vdb01** and **vdb02** ):

```
vi /usr/local/bin/crsstat

#!/bin/bash
RSC_KEY=$1
AWK=/usr/bin/awk # if not available use /usr/bin/awk
ORA CRS_HOME=/u01/cluster/12.1.0/grid

#support -t -v -p

if ["$1" == "-t"] ; then

if ["$2" == "-v" -o "$2" == "-q" -o "$3" == "-v" -o "$3" == "-q"] ; then
$ORA CRS_HOME/bin/crsctl status resource $*
exit

fi

shift

RSC_KEY=

Table header:echo ""
$AWK \
'BEGIN {printf "%-45s %-25s %-17s %-25s\n", "HA Resource", "Type", "Target", "State";
print "%-45s %-25s %-17s %-25s\n", "-----", "-----", "-----", "-----";}'

Table body:
$ORA CRS_HOME/bin/crsctl status resource $* | sed -e 's/ //g' | $AWK \
'BEGIN { FS="[,]" ; state = 0; }
$1~/NAME/ && $2~"/$RSC_KEY/" {appname = $2; state=1;}
$1~/TYPE/ && state == 1 {apptype = $2; state=2;}
$1~/TARGET/ && state == 2 {apptarget = $2,"$3; state=3;}
$1~/STATE/ && state == 3 {appstate = $2","$3; state=4;}
state == 4 {printf "%-45s %-25s %-17s %-25s\n", appname, apptype, apptarget, appstate; state=0;}'

elif ["$2" == "-t"] ; then

if ["$3" == "-v" -o "$4" == "-q" -o "$3" == "-v" -o "$4" == "-q"] ; then
$ORA CRS_HOME/bin/crsctl status resource $*
exit
```

```

fi

shift 2

Table header:echo ""
$AWK \
'BEGIN {printf "%-45s %-25s %-17s %-25s\n", "HA Resource", "Type", "Target", "State";
 printf "%-45s %-25s %-17s %-25s\n", "-----", "-----", "-----", "-----";
 -----", "-----";}''

Table body:
$ORA_CRS_HOME/bin/crctl status resource $* | sed -e 's/ //g' | $AWK \
-BEGIN { FS="[,]" ; state = 0; }
$1~/NAME/ && $2~/$RSC_KEY/ {appname = $2; state=1;}
$1~/TYPE/ && state == 1 {apptype = $2; state=2;}
$1~/TARGET/ && state == 2 {apptarget = $2","$3; state=3;}
$1~/STATE/ && state == 3 {appstate = $2","$3; state=4;}
state == 4 {printf "%-45s %-25s %-17s %-25s\n", appname, apptype, apptarget, appstate; state=0;}''

else

$ORA_CRS_HOME/bin/crctl status resource $*

fi
#####
chown root:dba /usr/local/bin/crsstat
chmod 750 /usr/local/bin/crsstat

```

### Oracle Ksplice subscription and activation

To enable Ksplice uptrack you have to get a subscription ( with an option of 30-days trial ) at the following link:

<https://www.ksplice.com>

You can find installation and configuration instructions at the following link:

<https://www.ksplice.com/uptrack/install>

### Document Location:

The complete document of this lab can be downloaded from here:

<https://www.oracle.com/technetwork/server-storage/vm/hol-oraclevm-ha-2349885.pdf>



**ORACLE®**

CONNECT WITH US

-  [blogs.oracle.com/oracle](http://blogs.oracle.com/oracle)
-  [facebook.com/oracle](http://facebook.com/oracle)
-  [twitter.com/oracle](http://twitter.com/oracle)
-  [oracle.com](http://oracle.com)

Oracle Corporation, World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065, USA

Worldwide Inquiries  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200

**Hardware and Software. Engineered to Work Together**

Copyright © 2014, 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 and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 1114