

ORACLE®

VM



## HOL 9079: Build a complete high-available Oracle VM Architecture

Oracle VM, Oracle Linux, Ksplice, Oracle Clusterware and MySQL.

BY: SIMON COTER

THANKS TO: DOAN NGUYEN, CHRISTOPHE PAULIAT, OLIVIER CANONGE, BRUNO BOTTREAU

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	3
Preparation ( done before lab )	4
Summary of steps	5
Start the servers ( VirtualBox VMs)	5
Connect to the Oracle VM Manager 3.3.1 and become familiar with the product	7
Detailed architecture of the Lab	11
Connect to the Oracle VM Servers and verify that everything is ready to accommodate the lab	12
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	15
The architecture proposed in this lab is based on:	15
Oracle VM 	15
Oracle Linux 	15
Oracle Ksplice 	16
Oracle Clusterware 	16



Oracle MySQL Enterprise 	16
Demonstrate high-availability features covered by Oracle Ksplice	18
Demonstrate high-availability features covered by Oracle Clusterware	21
Demonstrate high-availability features covered by Oracle VM	30
Oracle VM Live-Migrate	30
Oracle VM High-Availability	30
Upshot: Oracle VM High Availability	38



## 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.

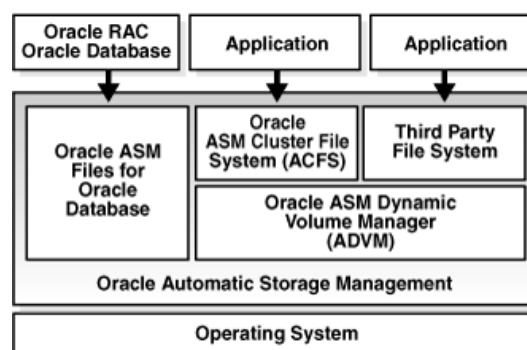
**Ksplice** is available for Oracle Linux, free of charge, for Oracle Linux customers with a Premier support subscription.

**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 ( done before lab )

To save time and fit in the one hour slot of Oracle OpenWorld labs, some actions were made before the actual lab.

Here is a quick list of these actions:

- 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 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 )

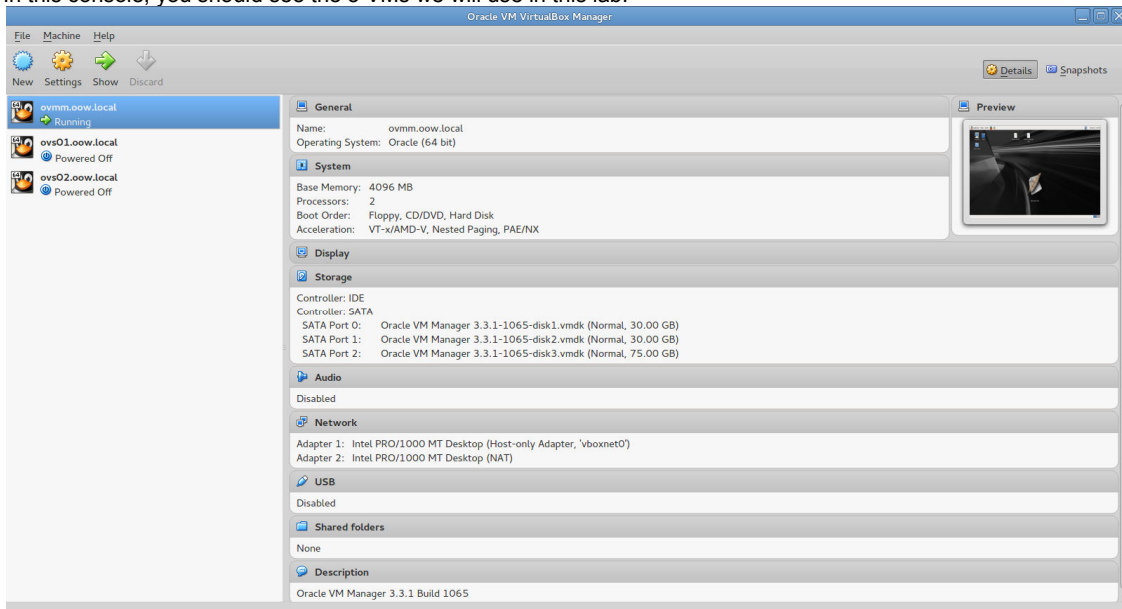
## Start the servers ( VirtualBox VMs)


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.oow.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.oow.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... [ OK ]
Starting OVM MySQL..._
```

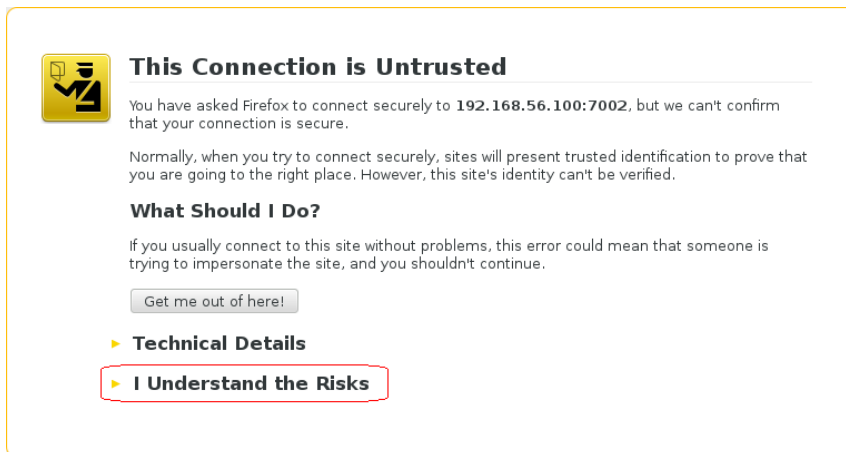
- e) Select the VM called “**ovs01.oow.local**” and click on the icon  to start it
- f) Select the VM called “**ovs02.oow.local**” and click on the icon  to start it
- g) Wait for the 3 VMs to be ready
  - o Wait for the prompt (desktop started) on “**ovmm.oow.local**” VM console
  - o When this prompt 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

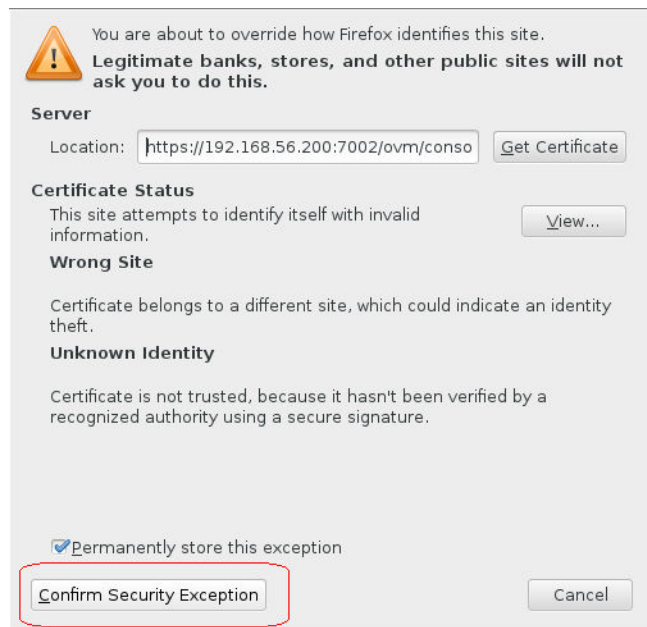
- a) On your Linux physical desktop open a Firefox browser and connect to the **Oracle VM Manager 3.3.1** console using URL <https://192.168.56.200:7002/ovm/console>

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

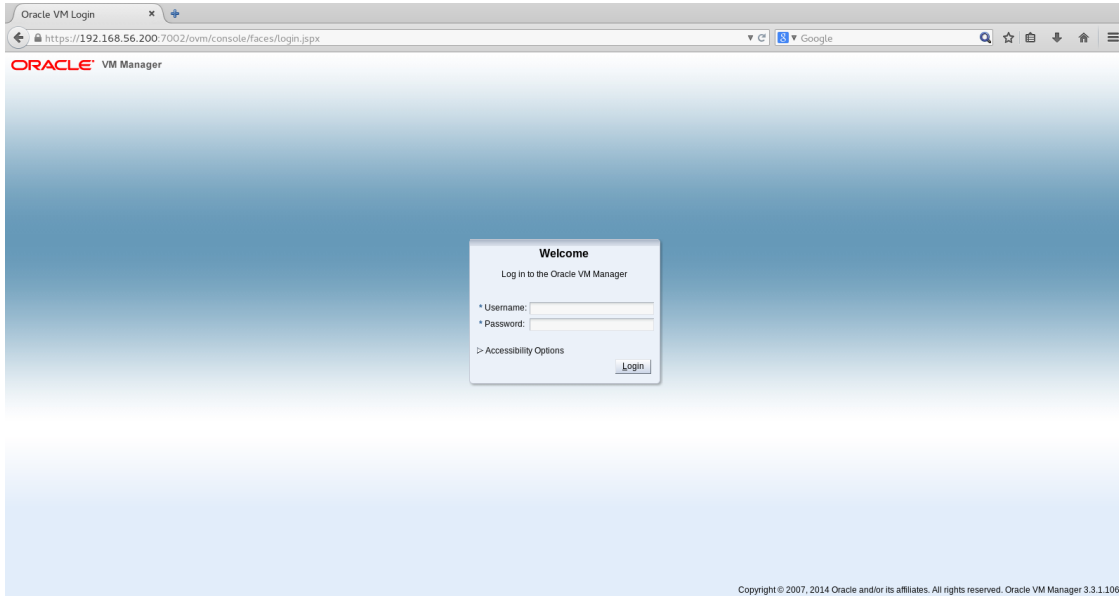
Click on **"I Understand the Risks"** and on **"Add exception"**:



Finally click on "Confirm Security Exception":



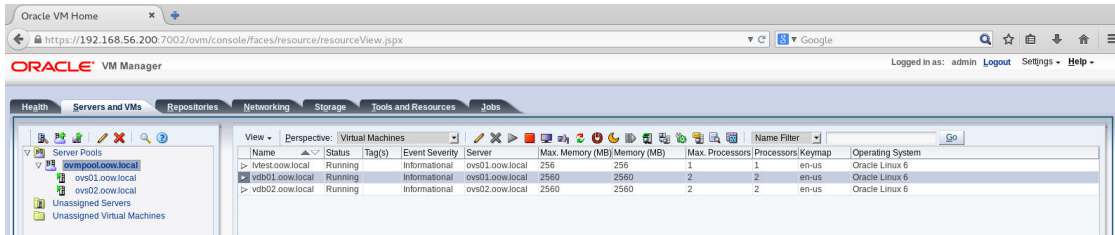
After security exception managed you should get the following login window:



- 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.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 this picture.

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

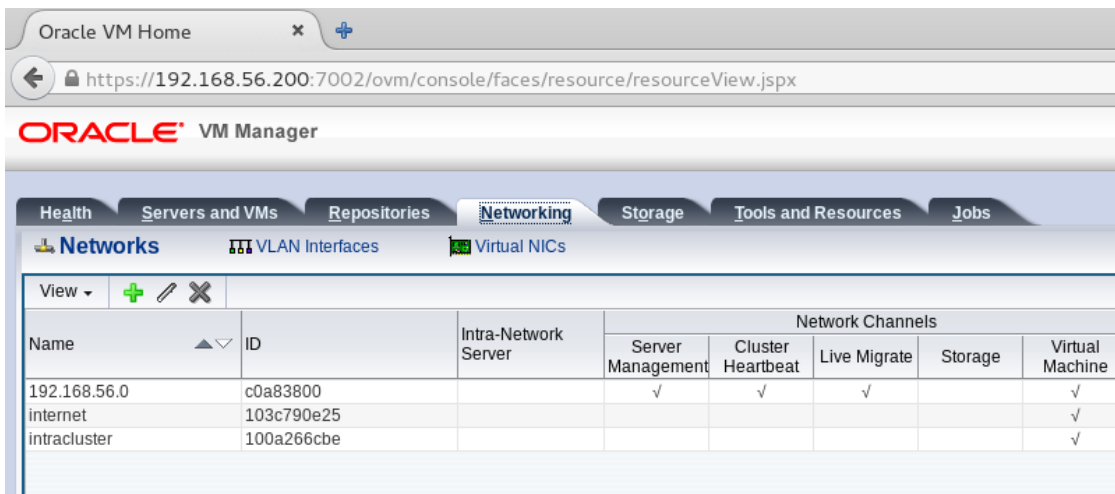


- b) Click on “**Networking**” tab and verify defined networks and their roles:

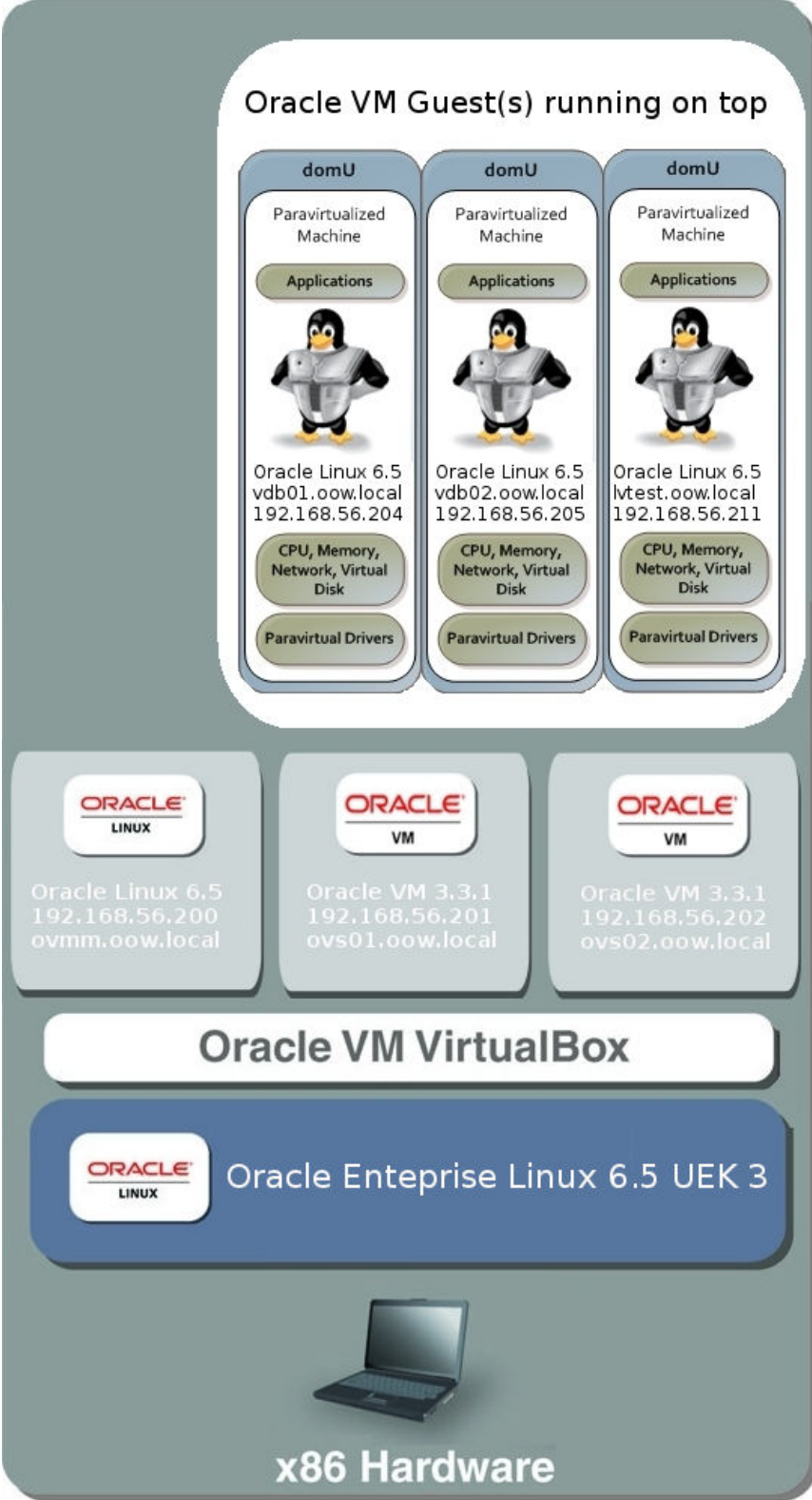
**192.168.56.0** = VirtualBox host-only Network used for default management, heartbeat and live-migrate network;also guests servers use this network to obtain access from our laptop

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

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

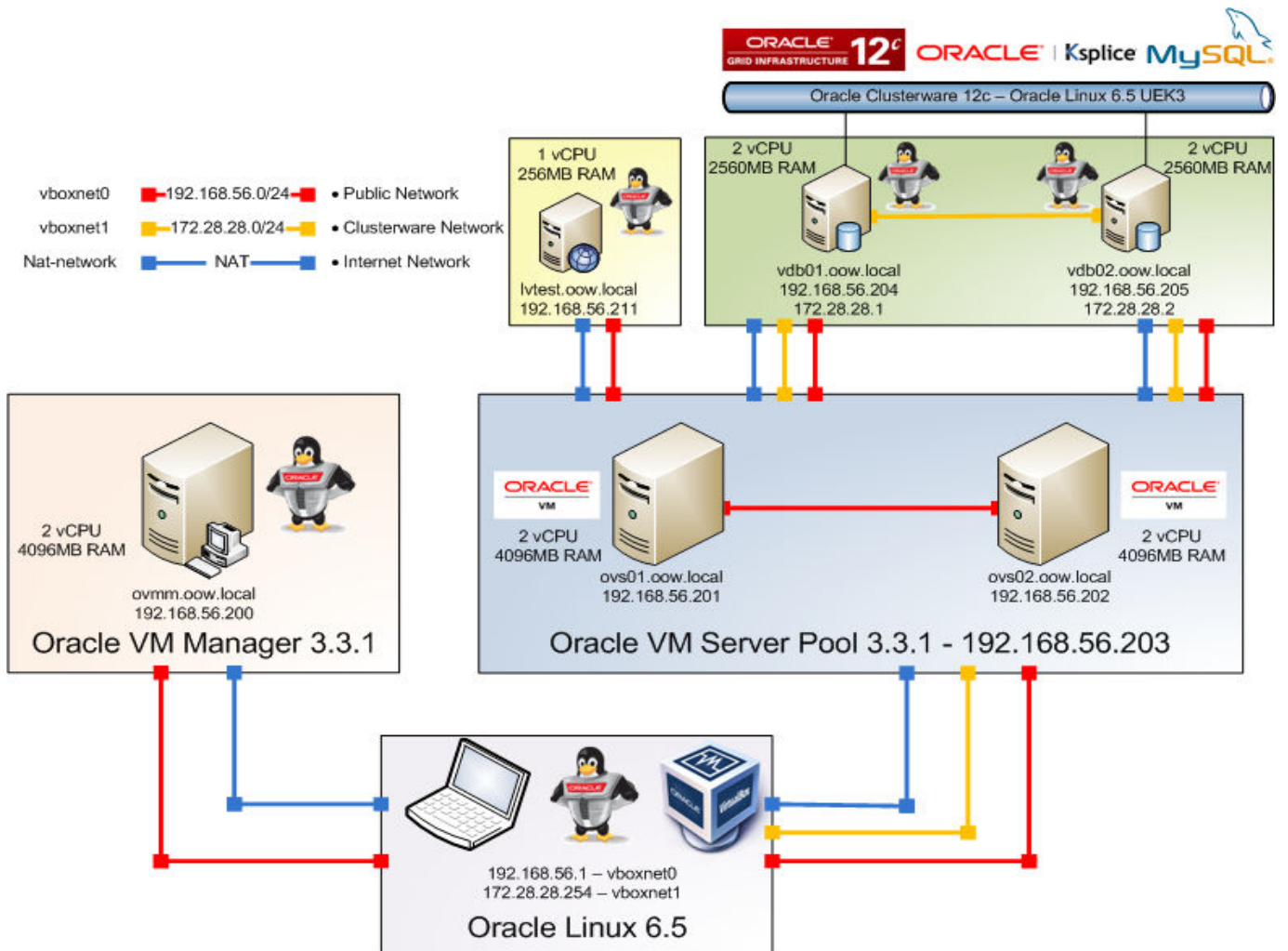


Here you can find the picture of the architecture built for this lab :





## 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](mailto:root@192.168.56.201) ( password is ovsroot )

ssh [root@192.168.56.202](mailto:root@192.168.56.202) ( password is ovsroot )

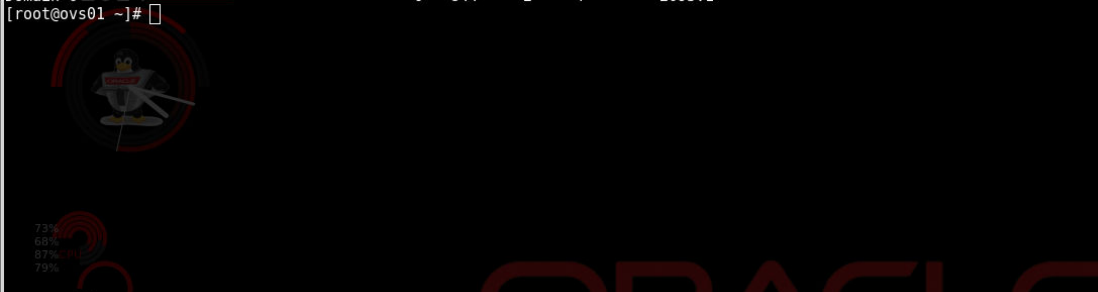
Once connected to both Oracle VM Servers, verify that Oracle VM repository is correctly mounted with the command “**df -k**”; the output should be the same as the figure below ( verify on both Oracle VM servers ):

```
Terminal - root@ovs01:~
drwxr-x--- 6 scoter users 12288 Jul 28 23:24 ProgInstalled
drwxr-xr-x 2 scoter users 4096 Sep 5 16:18 Scheduling
drwxr-xr-x 2 scoter users 4096 Jul 23 2013 Sql_scripts
drwxr-xr-x 2 scoter users 4096 Sep 9 12:34 UBI_Exalogic
drwxr-x--- 8 scoter users 4096 Sep 3 01:37 Unix_scripts
[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/sda1              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/0004fb0000030000cfe0e1ae4194da16
[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 )

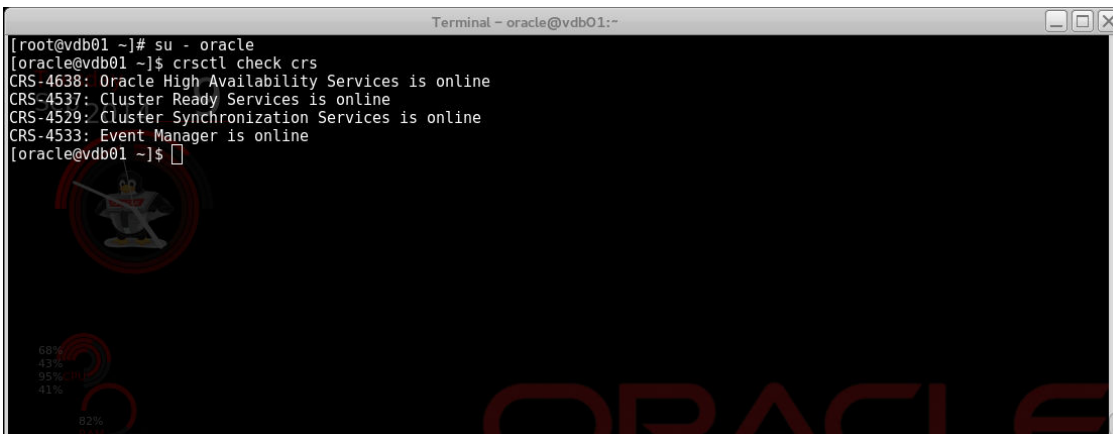
Connect by ssh to **vdb01.oow.local** / **vdb02.oow.local** and verify all Oracle Cluster managed resources are up and running:

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 )
```

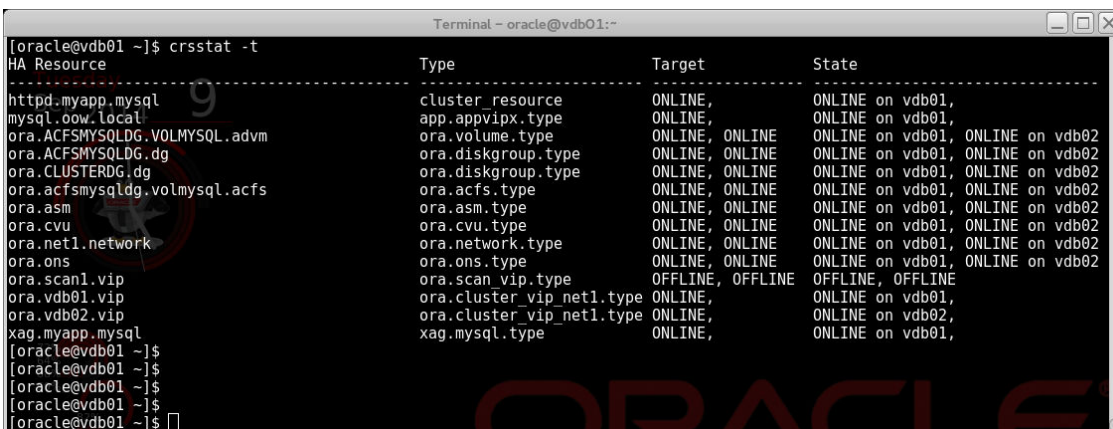
Once connected to both Oracle VM guests, verify that Oracle Clusterware is up and running and, at the same time, that all resource managed by Oracle Clusterware are correctly working (execute the commands on both guests):

- 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:



```
Terminal - oracle@vdb01:~  
[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**”.



```
Terminal - oracle@vdb01:~  
[oracle@vdb01 ~]# 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.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.acfsmysqldg.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,  
[oracle@vdb01 ~]#  
[oracle@vdb01 ~]#  
[oracle@vdb01 ~]#  
[oracle@vdb01 ~]#  
[oracle@vdb01 ~]#
```

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

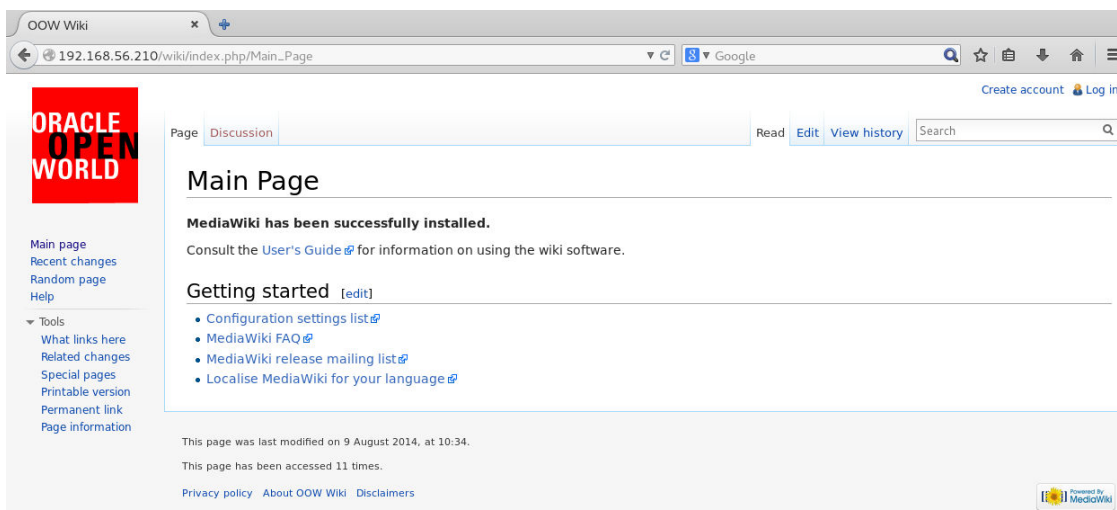
```
Terminal - oracle@vdb01:~  
[oracle@vdb01 ~]$ df -k /mysql  
Filesystem            1K-blocks    Used Available Use% Mounted on  
/dev/asm/volmysql-19 15466496 518112 14948384  4% /mysql  
[oracle@vdb01 ~]$
```

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:

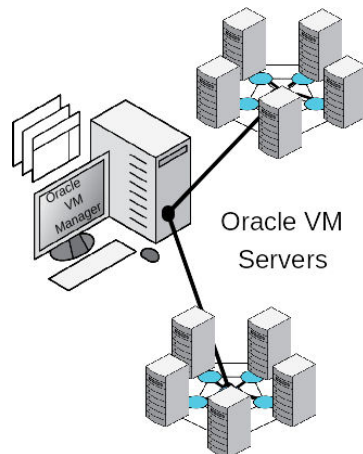
- Oracle VM 3.3.1
- Oracle Linux 6.x
- Oracle Ksplice
- Oracle Grid Infrastructure 12c ( Clusterware )
- Oracle Grid Infrastructure Agents 5.1 ( Clusterware agents to manage MySQL )
- Oracle MySQL Enterprise 5.6
- Demo web application

### Oracle VM

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

Ksplice Uptrack 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 these 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 allow to build up a complete high-available architecture that is able to grant a 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

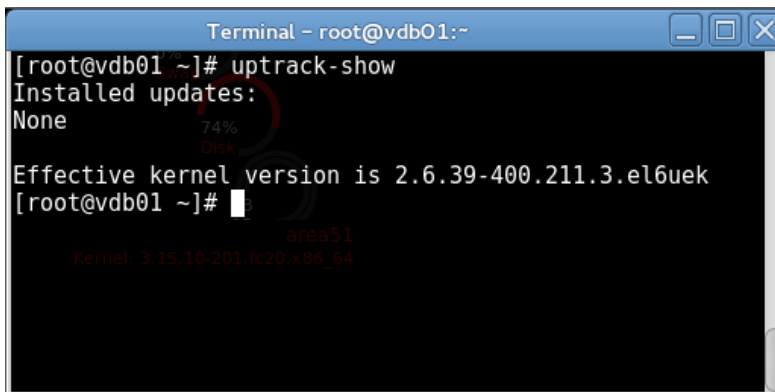


## Demonstrate high-availability features covered by Oracle Ksplice

As reported before Oracle Ksplice lets you applying 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”.

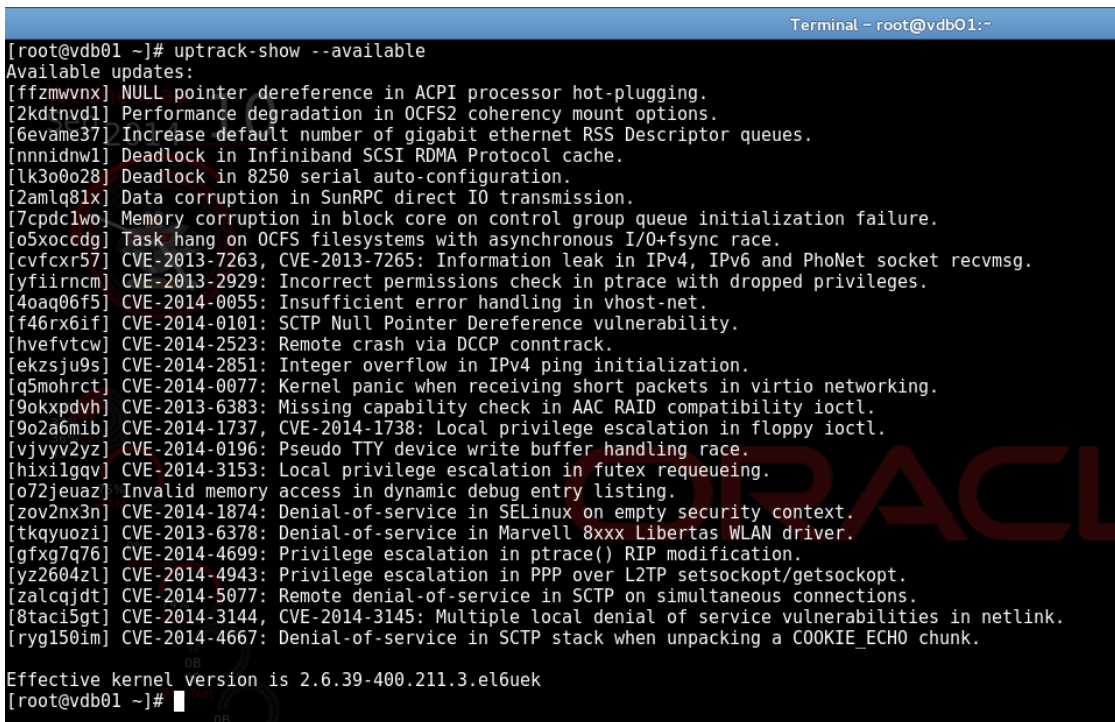
To update kernel on the servers specified above execute the following steps:

1. Connect to by ssh to the server “vdb01.oow.local”  
ssh [root@192.168.56.201](mailto:root@192.168.56.201)
2. Show which Ksplice kernel updates are already installed by executing “uptrack-show”



```
Terminal - root@vdb01:~  
[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”



```
Terminal - root@vdb01:~  
[root@vdb01 ~]# uptrack-show --available  
Available updates:  
[ffzmvwnx] NULL pointer dereference in ACPI processor hot-plugging.  
[2kdtndv1] 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.  
[lk3o028] Deadlock in 8250 serial auto-configuration.  
[2amlq81x] Data corruption in SunRPC direct IO transmission.  
[7cpdclwo] Memory corruption in block core on control group queue initialization failure.  
[o5xoccdg] Task hang on OCFS filesystems with asynchronous I/O+fsync race.  
[cvfxcxr57] CVE-2013-7263, CVE-2013-7265: Information leak in IPv4, IPv6 and PhoNet socket recvmsg.  
[yfiirncm] CVE-2013-2929: Incorrect permissions check in ptrace with dropped privileges.  
[40aq06f5] CVE-2014-0055: Insufficient error handling in vhost-net.  
[f46rx6if] CVE-2014-0101: SCTP Null Pointer Dereference vulnerability.  
[hvefvctw] CVE-2014-2523: Remote crash via DCCP conntrack.  
[ekzsj9s] CVE-2014-2851: Integer overflow in IPv4 ping initialization.  
[q5mohrct] CVE-2014-0077: Kernel panic when receiving short packets in virtio networking.  
[9okxpdvh] CVE-2013-6383: Missing capability check in AAC RAID compatibility ioctl.  
[9o2a6mib] CVE-2014-1737, CVE-2014-1738: Local privilege escalation in floppy ioctl.  
[vjyvy2yz] CVE-2014-0196: Pseudo TTY device write buffer handling race.  
[hixilgqv] 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.  
[gfgx7q76] CVE-2014-4699: Privilege escalation in ptrace() RIP modification.  
[yz2604z1] CVE-2014-4943: Privilege escalation in PPP over L2TP setsockopt/getsockopt.  
[zalcqjdt] CVE-2014-5077: Remote denial-of-service in SCTP on simultaneous connections.  
[8taci5qt] 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 ~]#
```



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

```
Terminal - root@vdb01:~
[root@vdb01 ~]# uname -a
Linux vdb01.oow.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.oow.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

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

```
Terminal - root@vdb01:~
[root@vdb01 ~]# uptrack-upgrade -y
The following steps will be taken:
Install [ff2mwvnx] NULL pointer dereference in ACPI processor hot-plugging.
Install [2kdtndv1] 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 [lk3o0z28] Deadlock in 8250 serial auto-configuration.
Install [2amlq81x] Data corruption in SunRPC direct IO transmission.
Install [7cpdc1wo] Memory corruption in block core on control group queue initialization failure.
Install [o5xoccdg] 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 [yfiirncm] CVE-2013-2929: Incorrect permissions check in ptrace with dropped privileges.
Install [4oaq06f5] 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 [ek2sj09s] CVE-2014-2851: Integer overflow in IPv4 ping initialization.
Install [q5mohrct] CVE-2014-0077: Kernel panic when receiving short packets in virtio networking.
Install [9okxpdvh] CVE-2013-6383: Missing capability check in AAC RAID compatibility ioctl.
Install [9o2a6mb1] CVE-2014-1737, CVE-2014-1738: Local privilege escalation in floppy ioctl.
Install [vjvyy2yz] CVE-2014-0196: Pseudo TTY device write buffer handling race.
Install [hixilgqv] CVE-2014-3153: Local privilege escalation in futext requeueing.
Install [o72jeusz] Invalid memory access in dynamic debug entry listing.
Install [2ov2nx3n] CVE-2014-1874: Denial-of-service in SELinux on empty security context.
Install [tkxyuoz1] CVE-2013-6378: Denial-of-service in Marvell 8xxx Libertas WLAN driver.
Install [gfg7q76] CVE-2014-4699: Privilege escalation in ptrace() RIP modification.
Install [yz2604z1] 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 [8tac15gt] 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 [ff2mwvnx] NULL pointer dereference in ACPI processor hot-plugging.
Installing [2kdtndv1] Performance degradation in OCFS2 coherency mount options.
Installing [6evame37] Increase default number of gigabit ethernet RSS Descriptor queues.
.....
Installing [yz2604z1] 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 [8tac15gt] 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 ~]#
```

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

```
Terminal - root@vdb01:~
[root@vdb01 ~]# uname -a
Linux vdb01.oow.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.oow.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

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

Example: `# uptrack-remove ryg150im`

```
Terminal - root@vdb01:~
[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 ).

- 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**

```
Terminal - root@vdb01:~
--> Finished Dependency Resolution
Dependencies Resolved
-----
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 noarch 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 528 kB/s | 32 MB 01:01
Running rpm check debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
Installing : kernel-uek-firmware-2.6.39-400.215.10.el6uek.noarch 1/3
Installing : kernel-uek-2.6.39-400.215.10.el6uek.x86_64 2/3
Cleanup : kernel-uek-2.6.32-400.36.1.el6uek.x86_64 3/3
Verifying : kernel-uek-firmware-2.6.39-400.215.10.el6uek.noarch 1/3
Verifying : kernel-uek-2.6.39-400.215.10.el6uek.x86_64 2/3
Verifying : kernel-uek-2.6.32-400.36.1.el6uek.x86_64 3/3
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

As reported above, **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**”:

```
Terminal - oracle@vdb01:~
[oracle@vdb01 ~]$ 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.ACFSMYSQLDG.VOLMYSQ 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.acfsmysqldg.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,
[oracle@vdb01 ~]$
[oracle@vdb01 ~]$
[oracle@vdb01 ~]$
[oracle@vdb01 ~]$
[oracle@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
```

```
Terminal - oracle@vdb01:~
[oracle@vdb01 ~]$ 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.ACFSMYSQLDG.VOLMYSQ 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.acfsmysqldg.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,
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,
[oracle@vdb01 ~]$ crsctl relocate resource httpd.myapp.mysql
CRS-2527: Unable to start 'httpd.myapp.mysql' because it has a 'hard' dependency on 'xag.myapp.mysql'
CRS-2525: All instances of the resource 'xag.myapp.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.myapp.mysql'
CRS-2525: All instances of the resource 'xag.myapp.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.

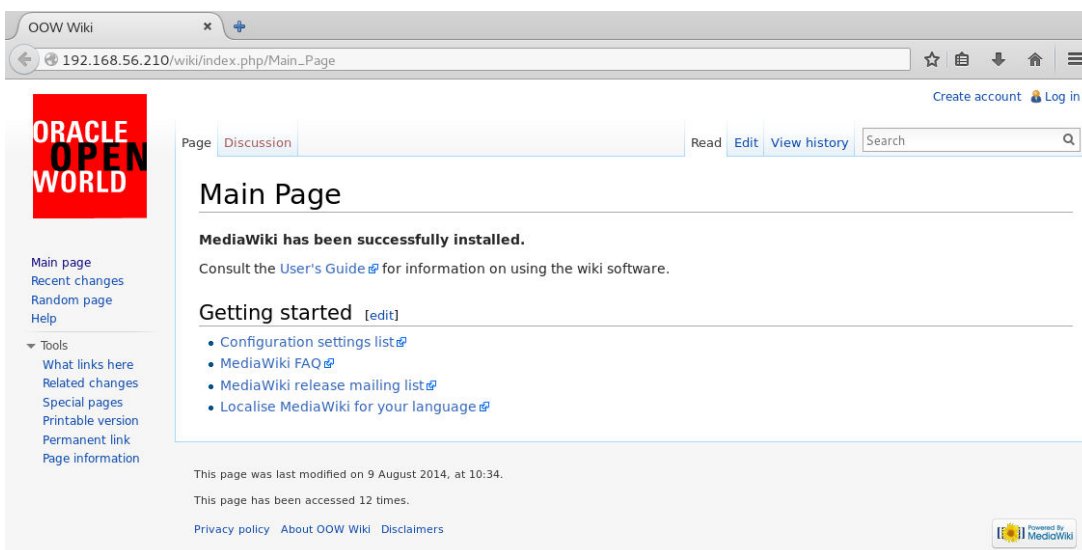
Service resources have the following dependencies:

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>



- 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”*

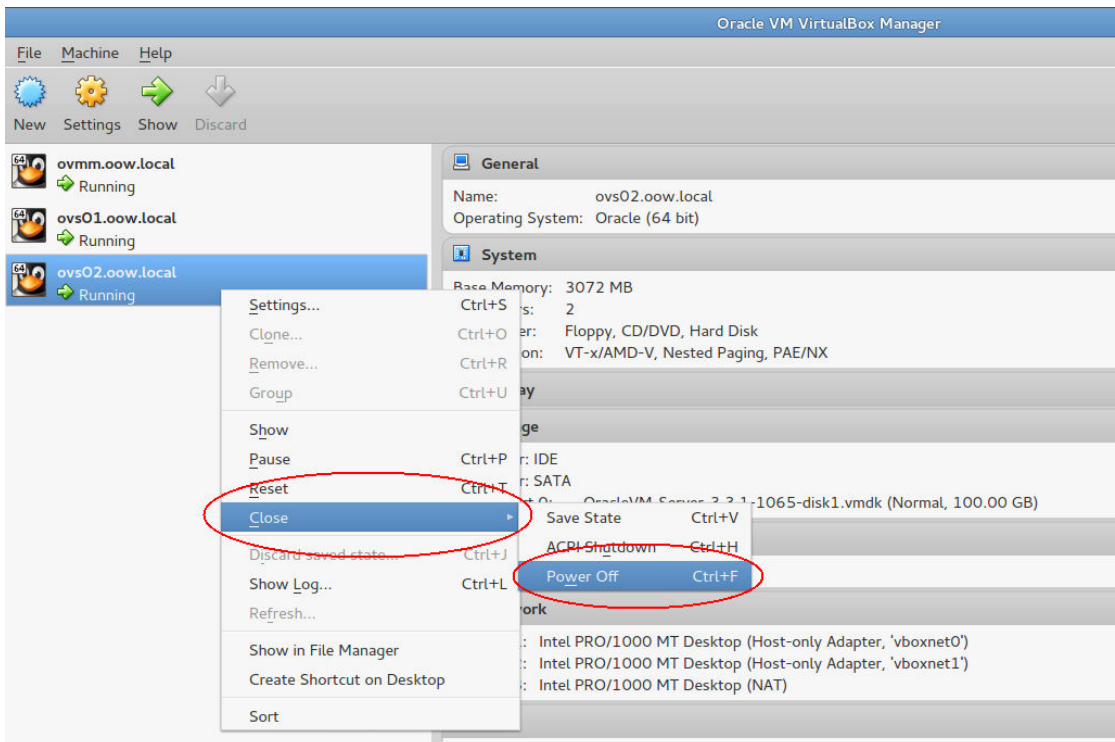
```

Terminal - oracle@vdb01:~
[oracle@vdb01 ~]$ crsstat -t
HA Resource                                     Type                Target               State
-----
http.myapp.mysql                               cluster resource    ONLINE,              ONLINE on vdb02,
mysql.oow.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.acfsmysqldg.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@vdb01 ~]$

```

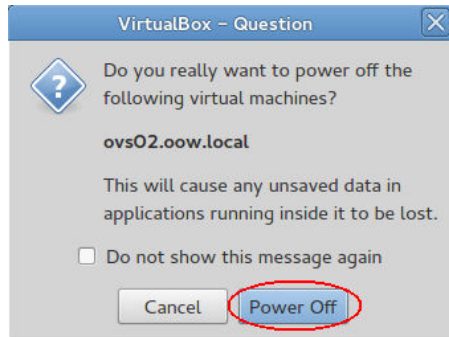
As you can see service resources are now active on node “vdb02.oow.local”.

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



Confirm to brutally shutdown the server “ovs02.oow.local”





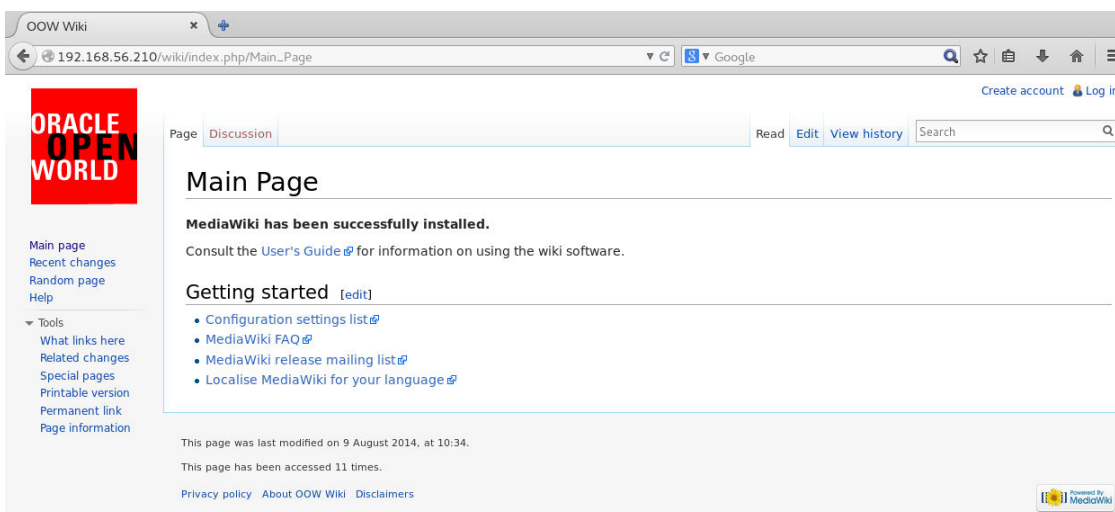
- Verify actual resource distribution on the cluster with the command “crsstat -t” and, at the same time, with the laptop browser (Mozilla Firefox) verify that web-demo application is available. You’ll see that, actually, only one node of the cluster is active and that all service resources are active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

```

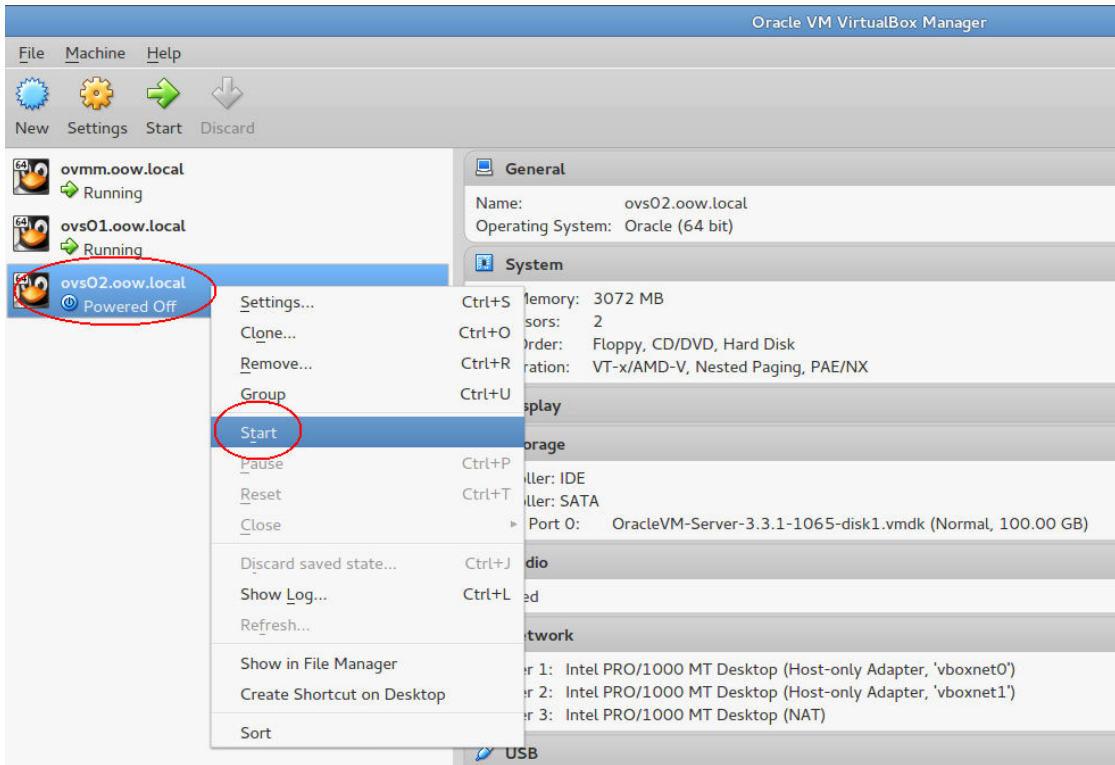
[oracle@vdb01 ~]$ crsstat -t
HA Resource                                Type                                Target                                State
-----
httpd.myapp.mysql                          cluster_resource                    ONLINE,                               ONLINE on vdb01,
mysql.oow.local                             app.appvipx.type                   ONLINE,                               ONLINE on vdb01,
ora.ACFSMySQLDG.VOLMYSQL.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.acfsmysqlldg.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,

```

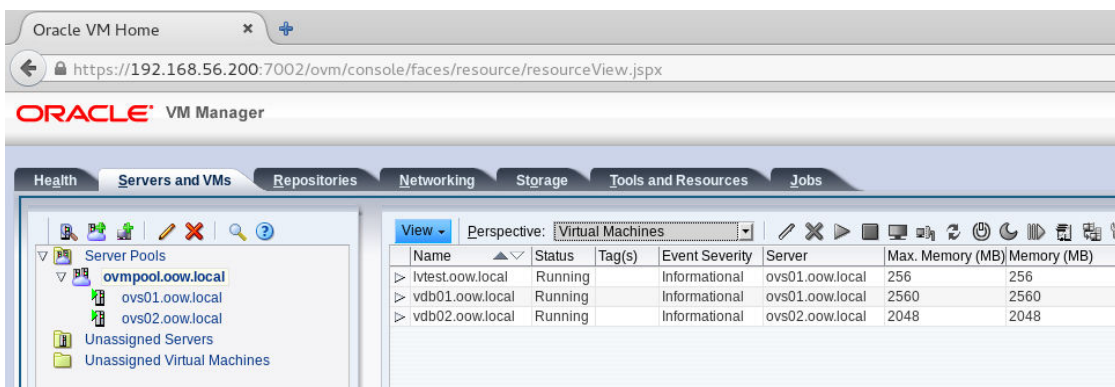
Verify application URL is reachable ( you can open from your laptop browser ): <http://192.168.56.210/wiki>



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



- 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.



- 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.

```

Terminal - root@vdb01:~
[root@vdb01 ~]# crsstat -t
HA Resource
-----
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.acfsmysql.dg.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 vdb01,
[root@vdb01 ~]#

```

Wait for "vdb02.oow.local" join the cluster before proceed with the steps below.

10. Connect, as **root**, to the node where our managed resources are active ( it should be **vdb01.oow.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>





```
Terminal - root@vdb01:~  
[root@vdb01 ~]# killall httpd  
[root@vdb01 ~]# ps -edaf |grep http  
root      29789 17220  0 02:46 pts/0        00:00:00 grep http  
[root@vdb01 ~]# ps -edaf |grep http  
root      29850 17220  0 02:47 pts/0        00:00:00 grep http  
[root@vdb01 ~]# ps -edaf |grep http  
root      29871 17220  0 02:47 pts/0        00:00:00 grep http  
[root@vdb01 ~]# ps -edaf |grep http  
root      29935      1  5 02:47 ?          00:00:00 /usr/sbin/httpd  
apache    29943 29935  0 02:47 ?          00:00:00 /usr/sbin/httpd  
apache    29945 29935  0 02:47 ?          00:00:00 /usr/sbin/httpd  
apache    29946 29935  0 02:47 ?          00:00:00 /usr/sbin/httpd  
apache    29947 29935  0 02:47 ?          00:00:00 /usr/sbin/httpd  
apache    29948 29935  0 02:47 ?          00:00:00 /usr/sbin/httpd  
apache    29949 29935  0 02:47 ?          00:00:00 /usr/sbin/httpd  
apache    29950 29935  0 02:47 ?          00:00:00 /usr/sbin/httpd  
apache    29951 29935  0 02:47 ?          00:00:00 /usr/sbin/httpd  
root      29953 17220  0 02:47 pts/0        00:00:00 grep http  
[root@vdb01 ~]#
```

- 12. With these step we will try do simulate a filesystem 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**

```
Terminal - root@vdb01:~  
[root@vdb01 ~]# file /usr/sbin/httpd  
/usr/sbin/httpd: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamical  
ly 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**

```
Terminal - root@vdb01:~  
[root@vdb01 ~]# crsstat -t  
HA Resource                                     Type          Target        State  
-----  
httpd.myapp.mysql                             cluster_resource ONLINE,      ONLINE on vdb02,  
mysql.oow.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.acfsmysqldg.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,  
[root@vdb01 ~]#
```

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**

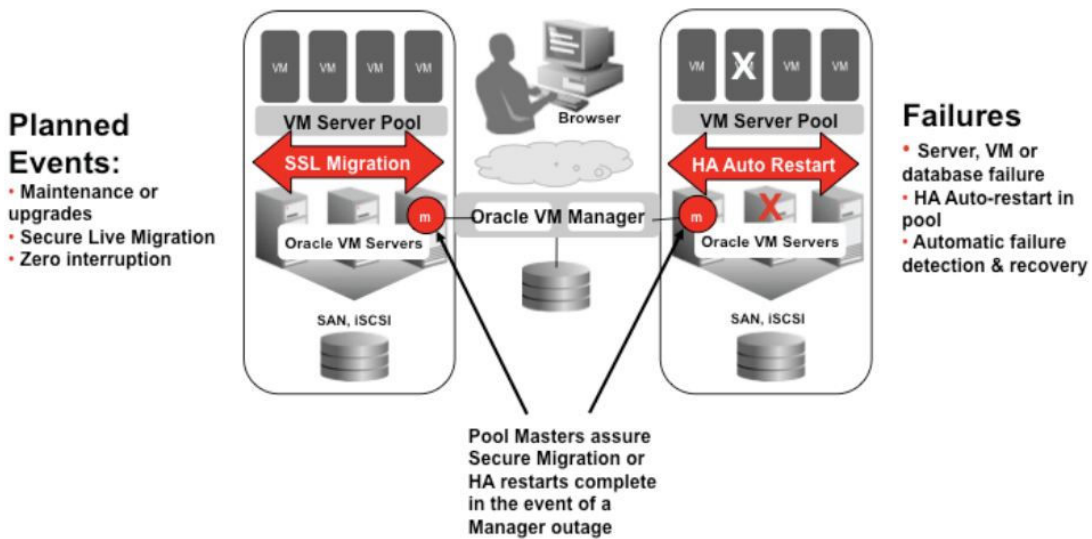


## Demonstrate high-availability features covered by Oracle VM

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.

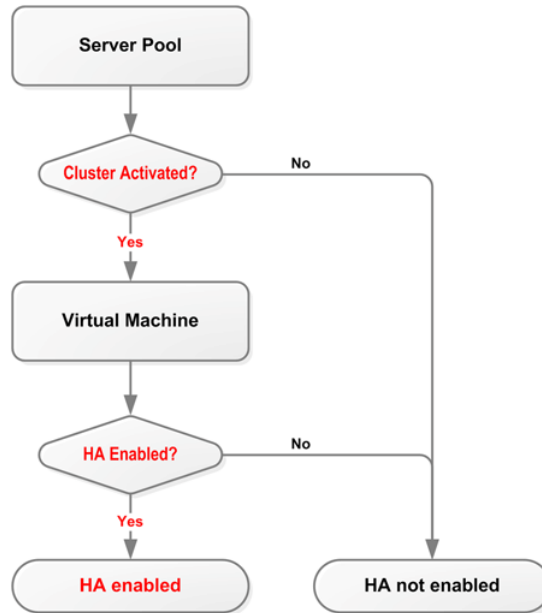
### Oracle VM High-Availability

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:

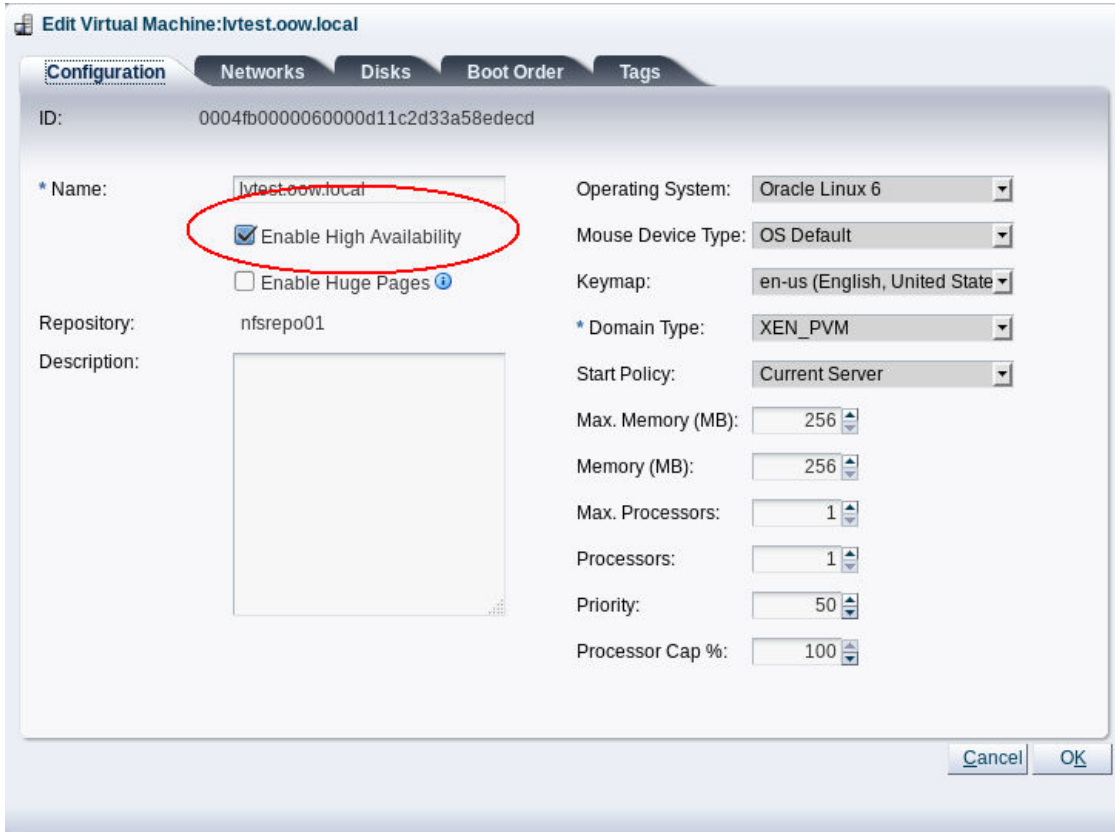
- The server pool must contain multiple Oracle VM Servers.  
HA cannot be implemented with a stand-alone Oracle VM Server.
- All Oracle VM Servers must be Oracle VM Server Release 3.0 or above.
- Oracle VM Pool needs to be “clustered”

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

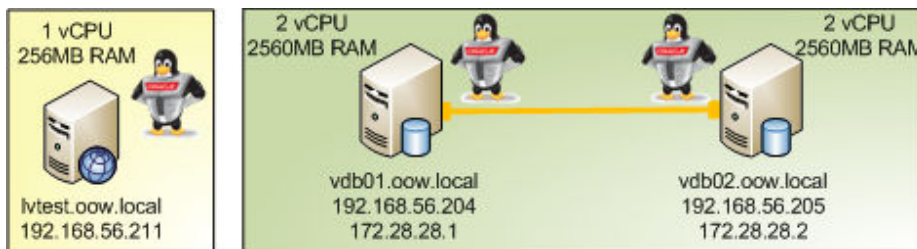


To use HA, you have to configure a clustered Oracle VM Server Pool:

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):

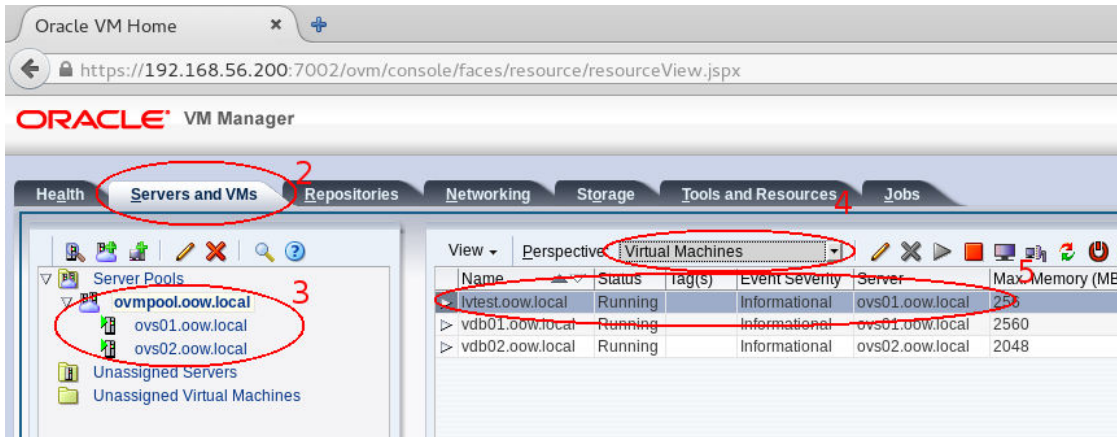


On our H/A tests, due to the reduced amount of resources available on our system,( mainly RAM ), we will demonstrate both Oracle VM High-Availability and Oracle VM Live-Migrate using the guest named "lvtest.oow.local".

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

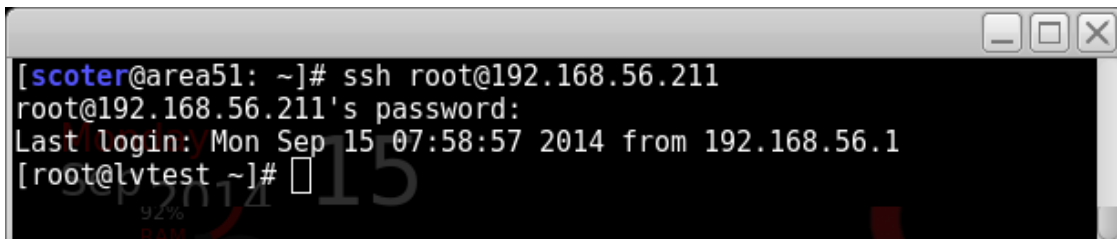
1. Connect to Oracle VM Manager and verify the status of the guest named "lvtest.oow.local".  
Oracle VM Manager URL: <https://192.168.56.200:7002/ovm/console>  
Username: admin  
Password: Welcome1
2. Select "Servers & VMs tab"
3. Expand and select pool named "ovmpool.oow.local"
4. Select perspective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

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

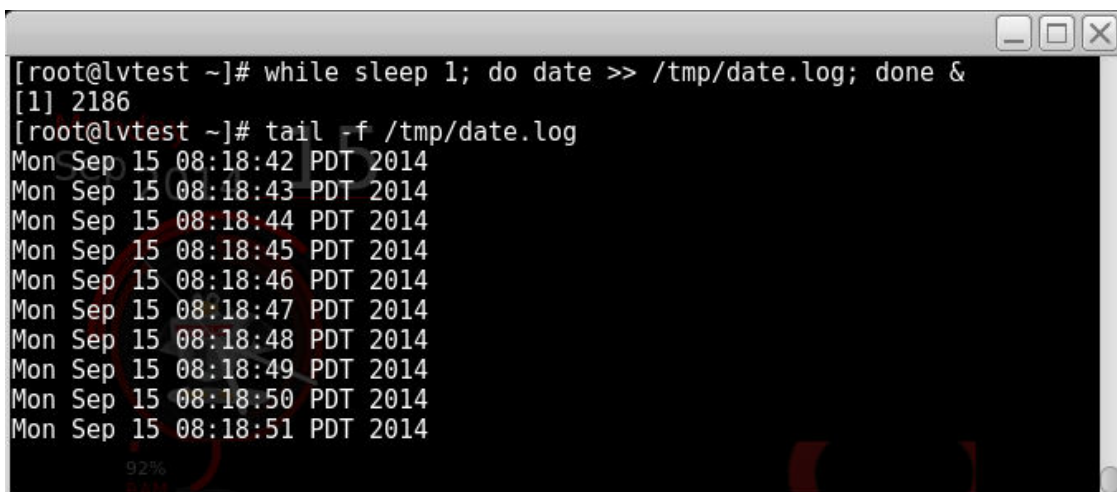


In the case above the guest “**lvtest.oow.local**” is running on Oracle VM Server “**ovs01.oow.local**”.

- Connect, by ssh, to the server “**lvtest.oow.local**”  
ssh [root@192.168.56.211](mailto:root@192.168.56.211) ( password is ovsroot )

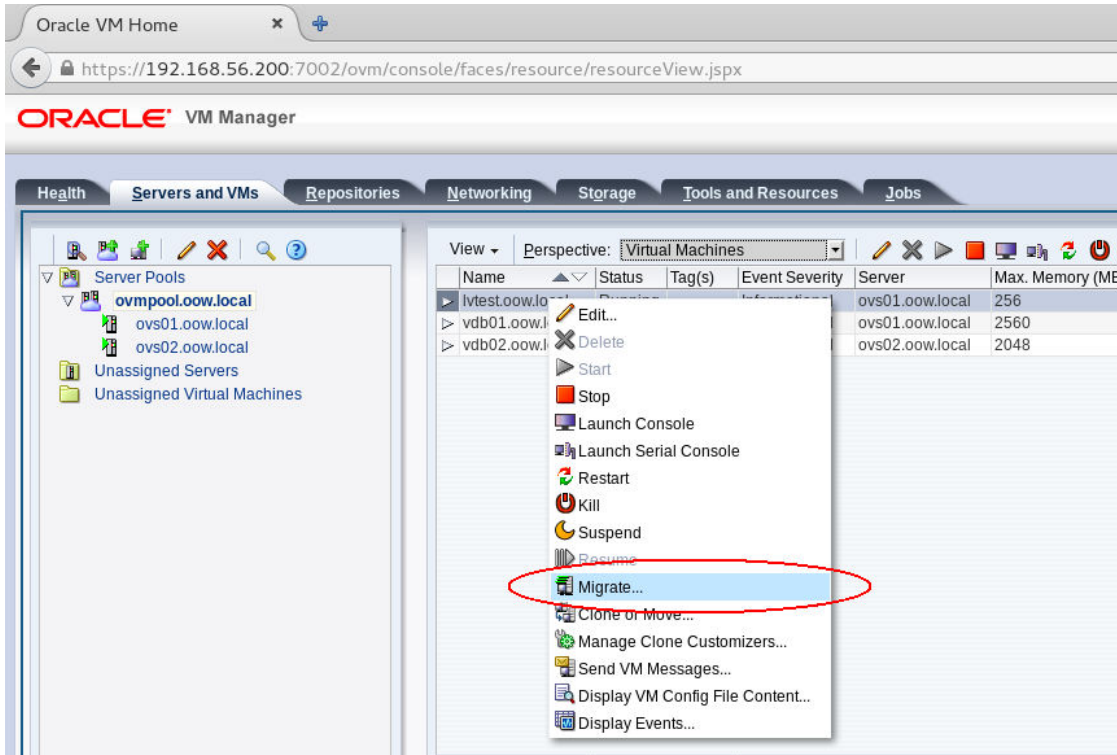


- Launch a command that will show the date of the server updated every one second.  
# **while sleep 1; do date >> /tmp/date.log; done &**  
# **tail -f /tmp/date.log**

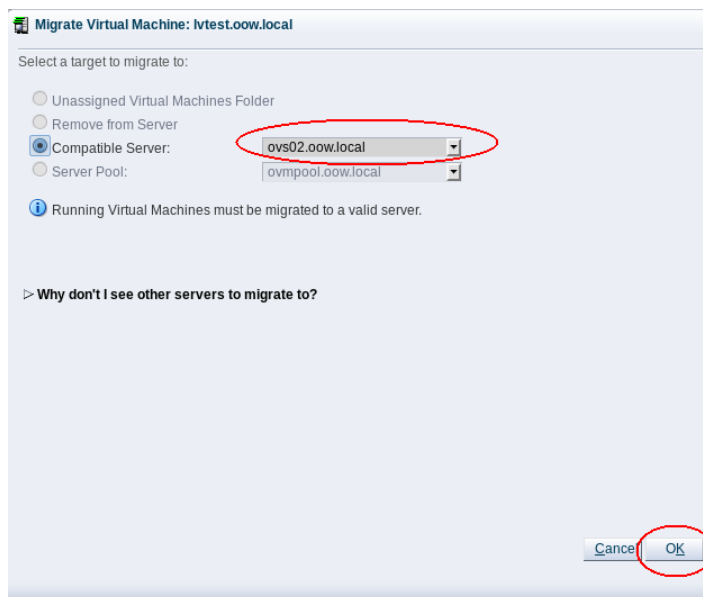




- While this script is running, execute the live-migrate of the guest by Oracle VM Manager. Re-open the **Oracle VM Manager console**, select guest named "lvtest.oow.local", **right-click** and select "**Migrate**".



- Select "compatible-server" "ovs02.oow.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.*

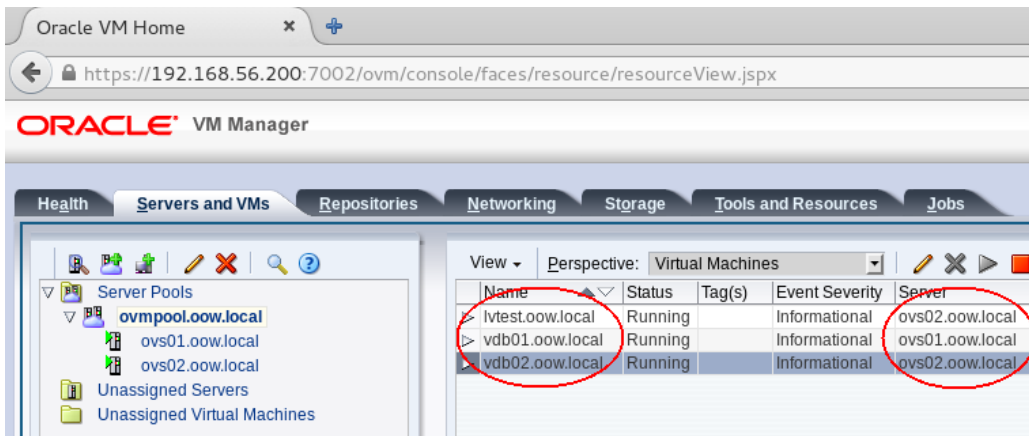
```

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 08:31:15 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.ow.local”



In the example above:

- Guests “**lvtest**” and “**vdb02**” are running on physical server **ovs02.oow.local**
- Guest “**vdb01**” is running on physical server **ovs01.oow.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
```

```

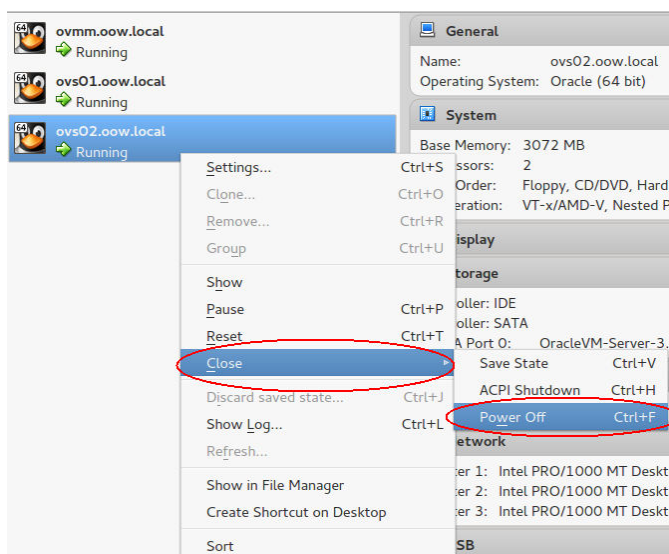
[root@vdb01 ~]# crsstat -t
HA Resource                                     Type           Target          State
-----
httpd.myapp.mysql                             cluster_resource ONLINE,         ONLINE on vdb02,
mysql.oow.local                               app.appvipx.type ONLINE,         ONLINE on 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.acfsmysql.dg.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,
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,
[root@vdb01 ~]#
  
```

In the case above we have that:

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

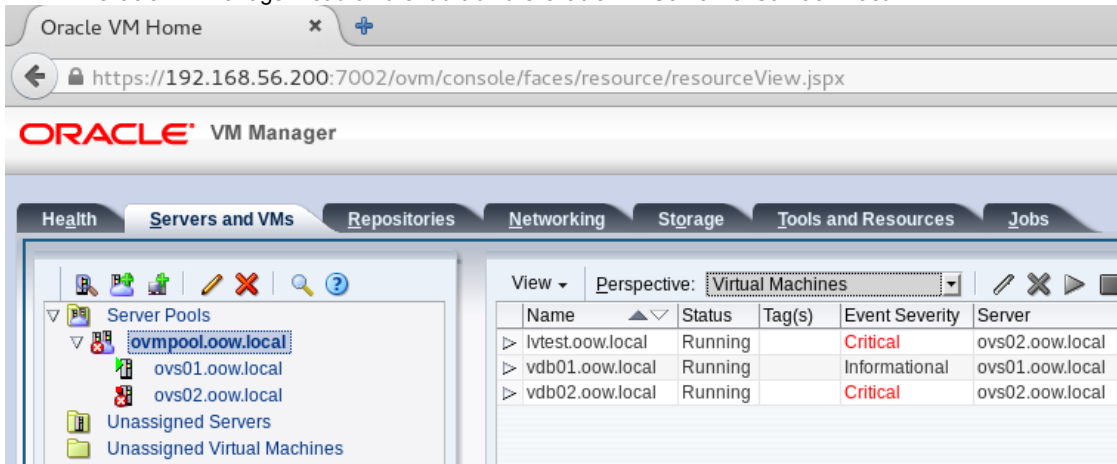
The target of this test is to **simulate a crash of the Oracle VM Server that owns**, at the same time, both guests **lvtest** and **vdb0(?)** that owns the **web/mysql** services ( if you need to move services between cluster-nodes, you can use the steps reported in this document at the section “Demonstrate 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 “lvtest”  
Open “**VirtualBox Console**”, right click on the server and choose “**Close**”, “**Poweroff**”.

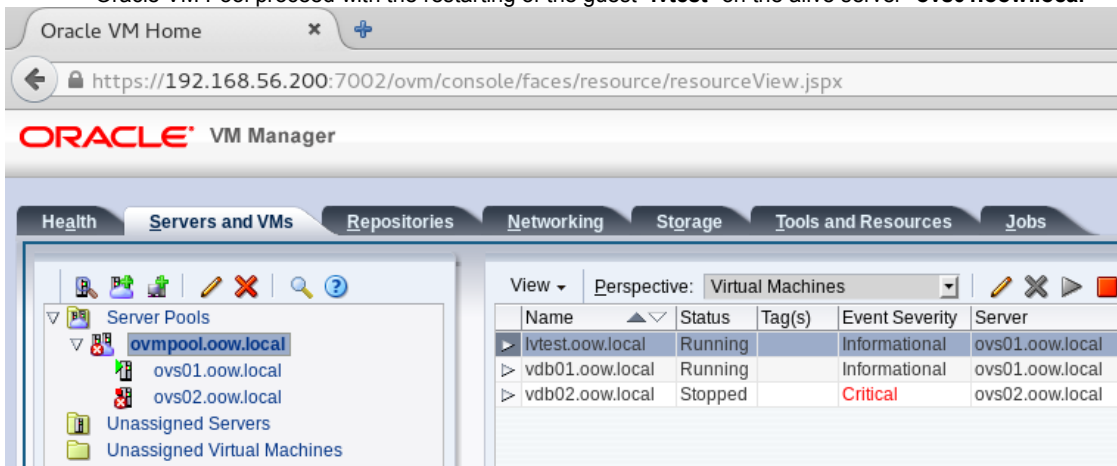


Next steps that will happen on the environment will be:

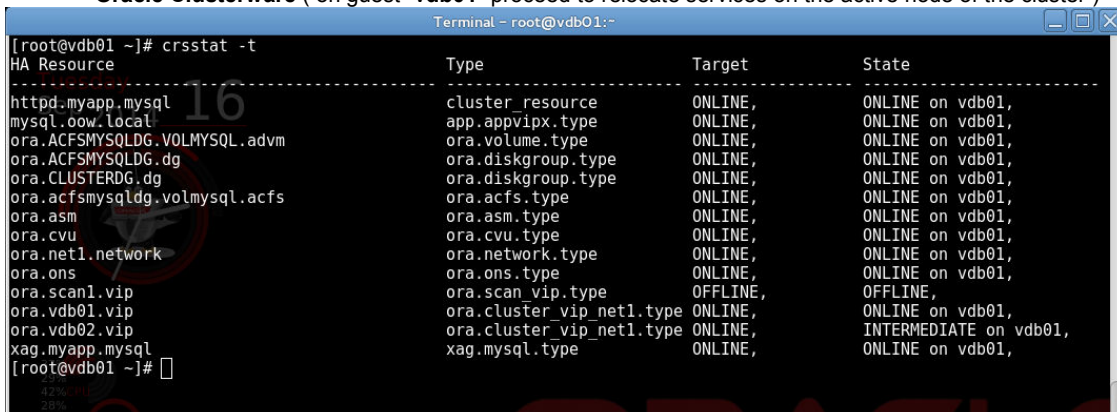
- Oracle VM Manager head off the **fault** on the Oracle VM Server “**ovs02.oow.local**”



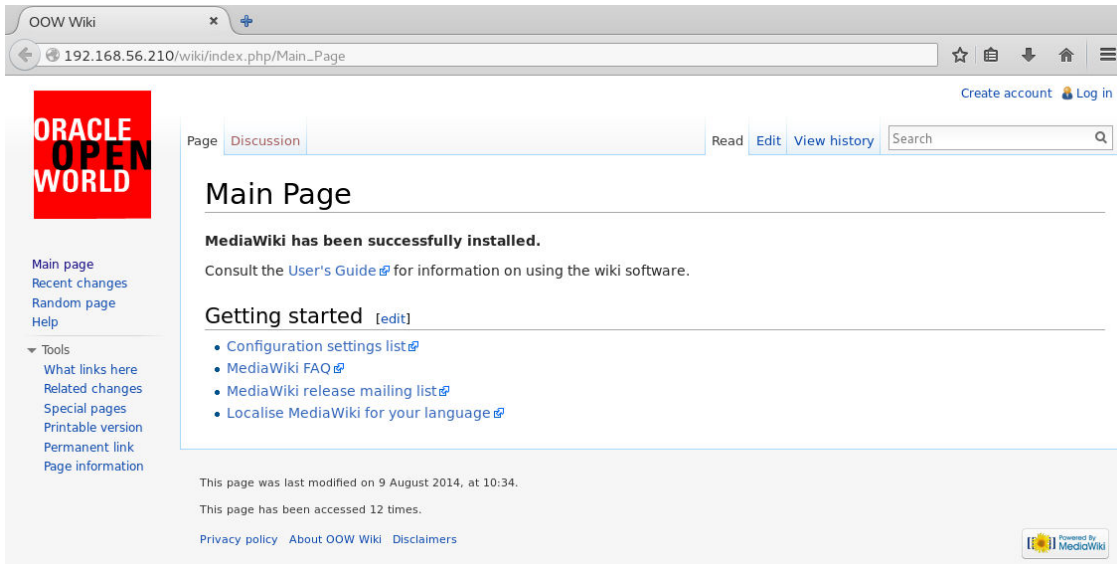
- Oracle VM Pool proceed with the restarting of the guest “**lvtest**” on the alive server “**ovs01.oow.local**”



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



- Verify that web-demo application is still available connecting to its URL ( <http://192.168.56.210/wiki> )



## 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.




**Oracle Corporation, World Headquarters**

500 Oracle Parkway  
Redwood Shores, CA 94065, USA

**Worldwide Inquiries**

Phone: +1.650.506.7000  
Fax: +1.650.506.7200

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)

**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. 0914