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

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 Clsuterware (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.

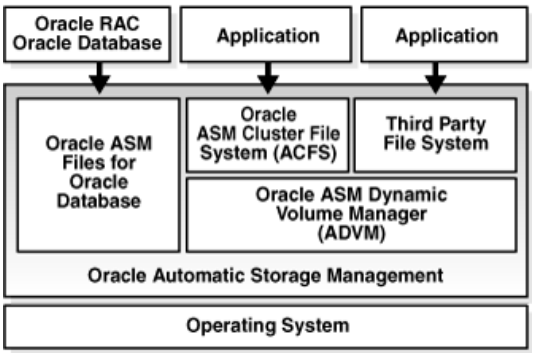
Oracle Linux, a Linux free provides the latest innovations, tools, and features that enable you to innovate, collaborate, and create solutions across traditional, cloud-based, and virtual environments.
Oracle Linux is easy to download, free to use, free to distribute, and free to update.
Oracle Linux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

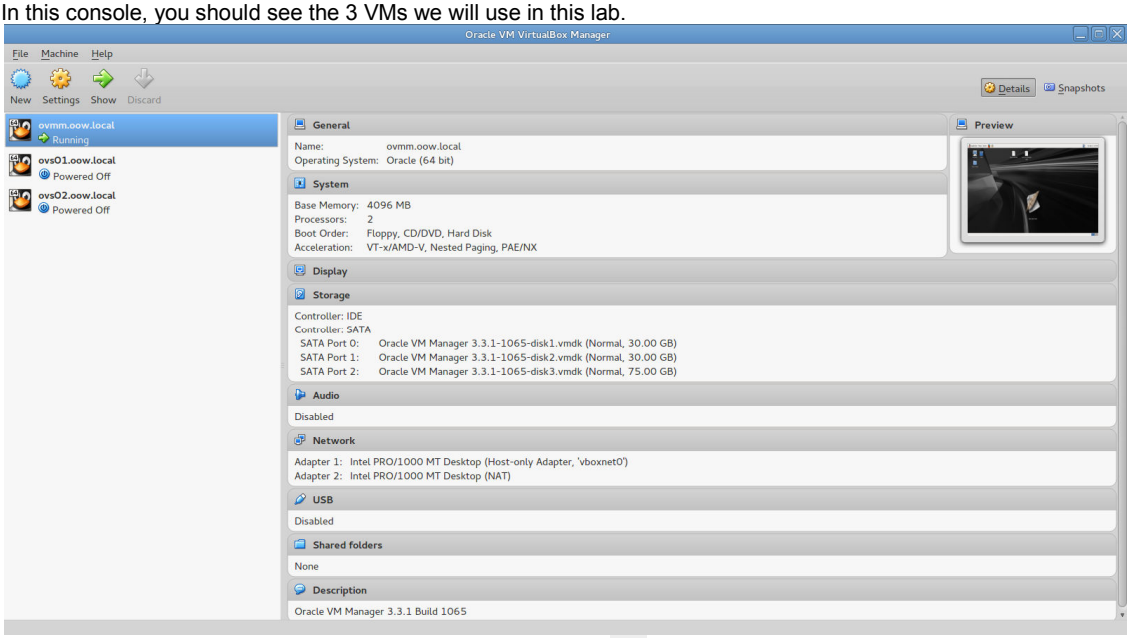
As previously explained, we will use Oracle VM VirtualBox to host the 3 servers (Oracle VM Manager and two Oracle VM Servers) on a single laptop.

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.

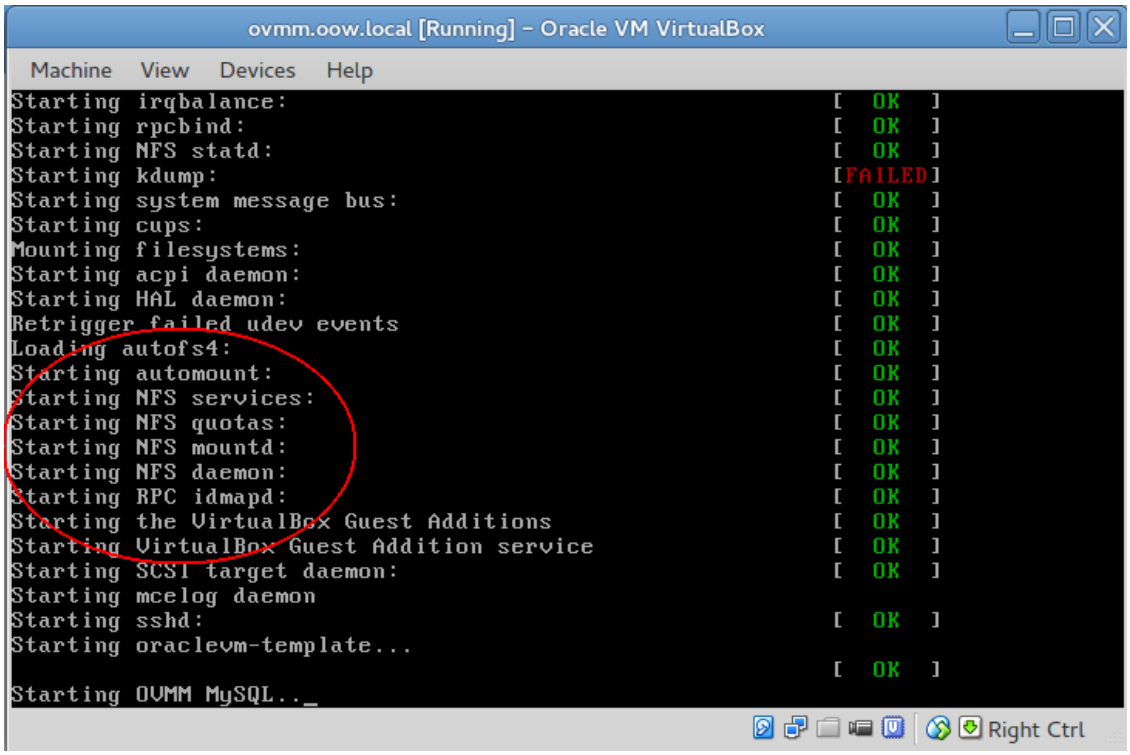
a) Start the Oracle VM VirtualBox console if not yet started by clicking on icon 

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):



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

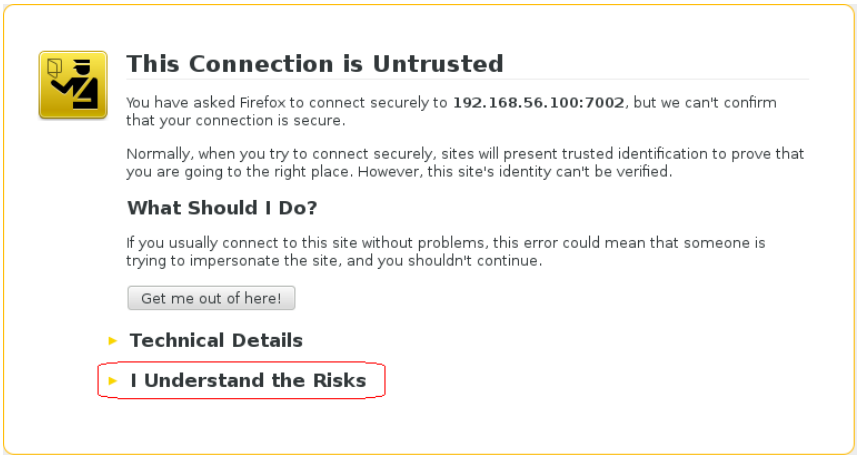
g) Wait for the 3 VMs to be ready
   - Wait for the prompt (desktop started) on "**ovmm.oow.local**" VM console
   - When this prompt is displayed, all VMs are ready (since Oracle VM Manager is the longest to start)

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

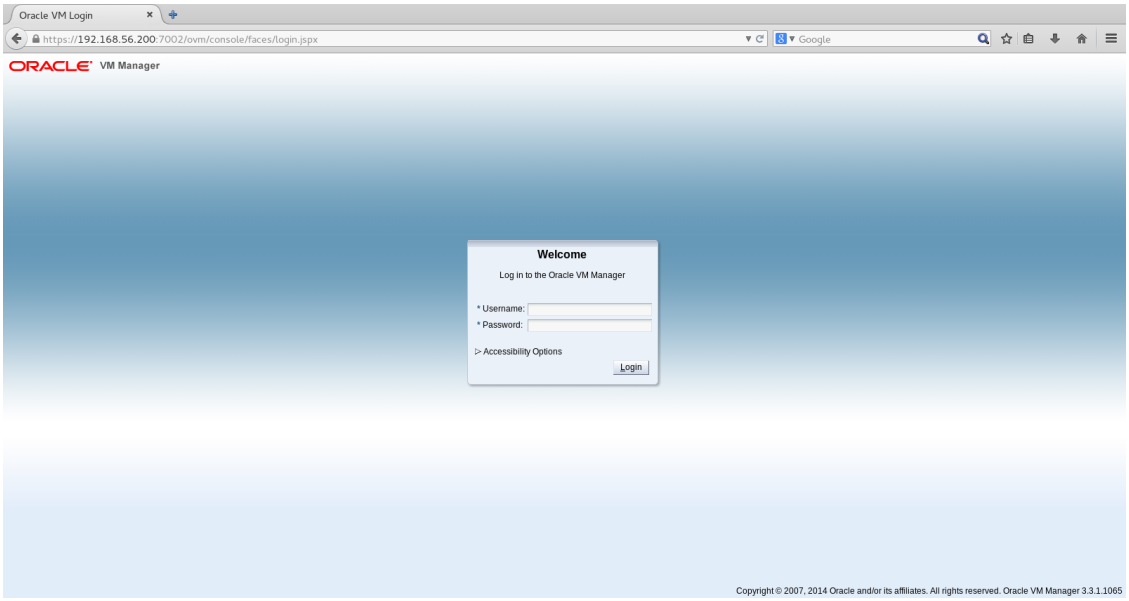
In the case that you receive some warnings proceed as described in the following screens:

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



Finally click on "Confirm Security Exception":

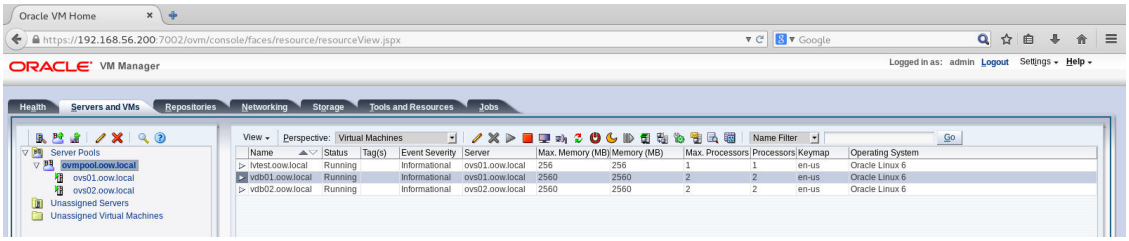
After security exception managed you should get the following login window:



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.

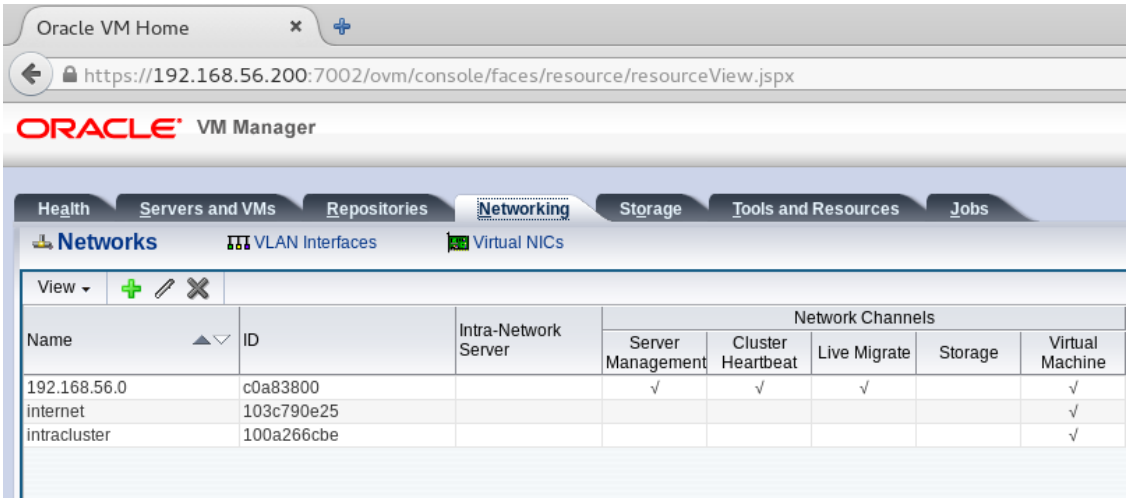


b) Click on "**Networking**" tab and verify defined networks and their roles:

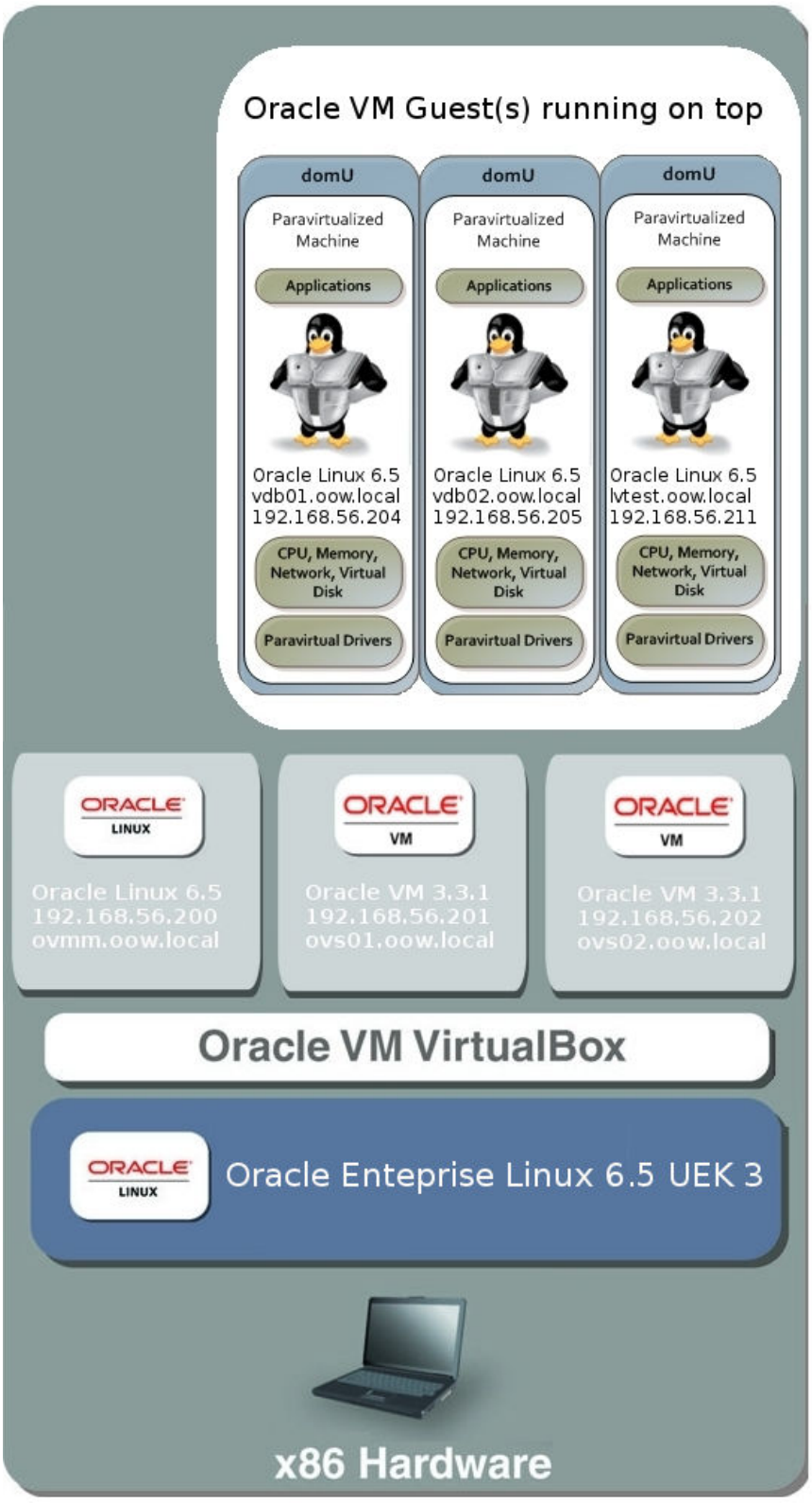
192.168.56.0 = VirtualBox host-only Network used for default management, heartbeat and live-migrate network;also guests servers use this network to obtain access from our laptop

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

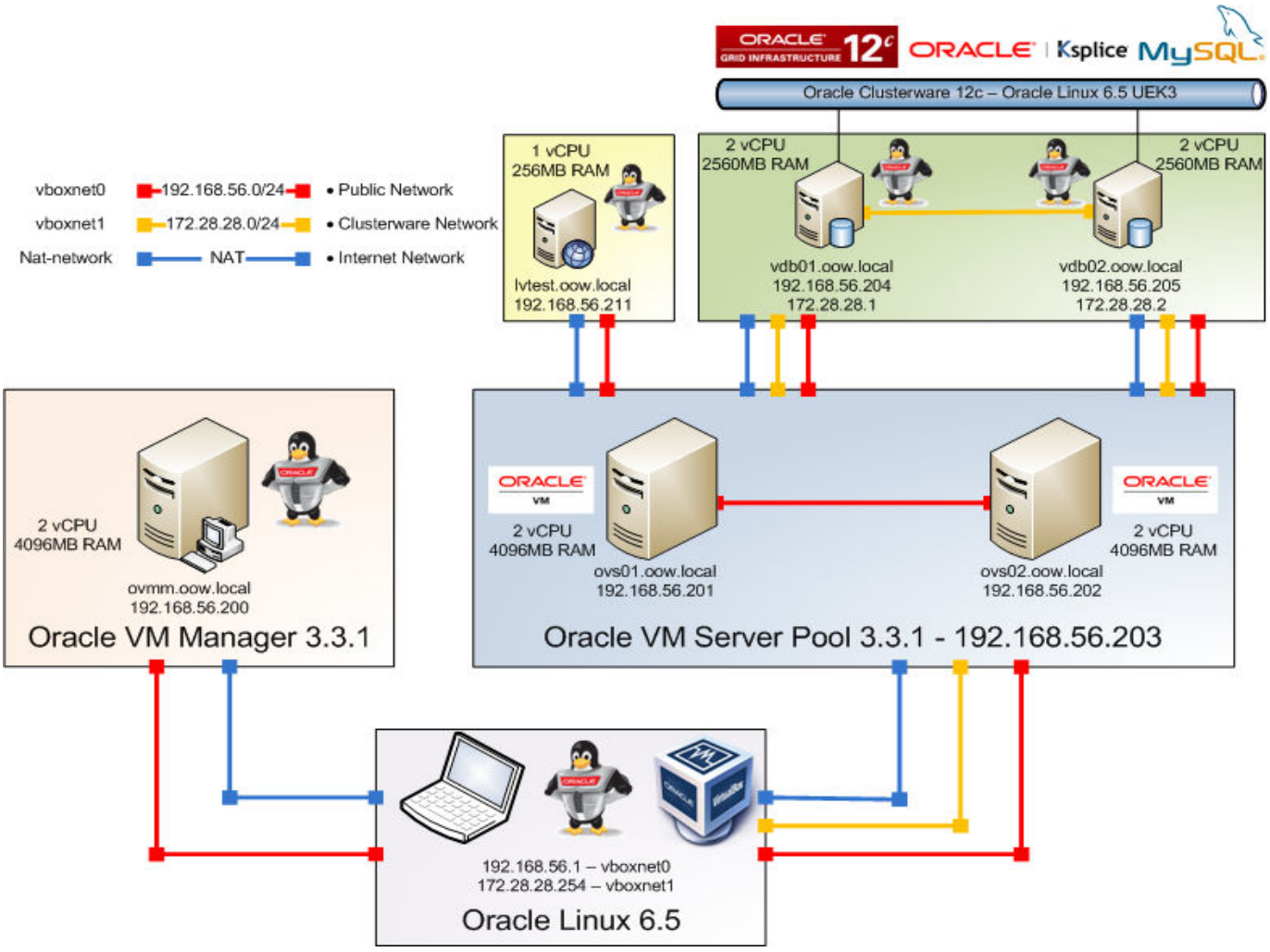
Intracluster = VirtualBox host-only Network used for intracluster on Oracle Clusterware



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


Oracle VM Guest(s) running on top
domU
Paravirtualized Machine
Applications
Oracle Linux 6.5
vdb01.oow.local
192.168.56.204
CPU, Memory, Network, Virtual Disk
Paravirtual Drivers
domU
Paravirtualized Machine
Applications
Oracle Linux 6.5
vdb02.oow.local
192.168.56.205
CPU, Memory, Network, Virtual Disk
Paravirtual Drivers
domU
Paravirtualized Machine
Applications
Oracle Linux 6.5
lvtest.oow.local
192.168.56.211
CPU, Memory, Network, Virtual Disk
Paravirtual Drivers
ORACLE LINUX
Oracle Linux 6.5
192.168.56.200
ovmm.oow.local
ORACLE VM
Oracle VM 3.3.1
192.168.56.201
ovs01.oow.local
ORACLE VM
Oracle VM 3.3.1
192.168.56.202
ovs02.oow.local
Oracle VM VirtualBox
ORACLE LINUX
Oracle Enteprise Linux 6.5 UEK 3
x86 Hardware

Detailed architecture of the Lab

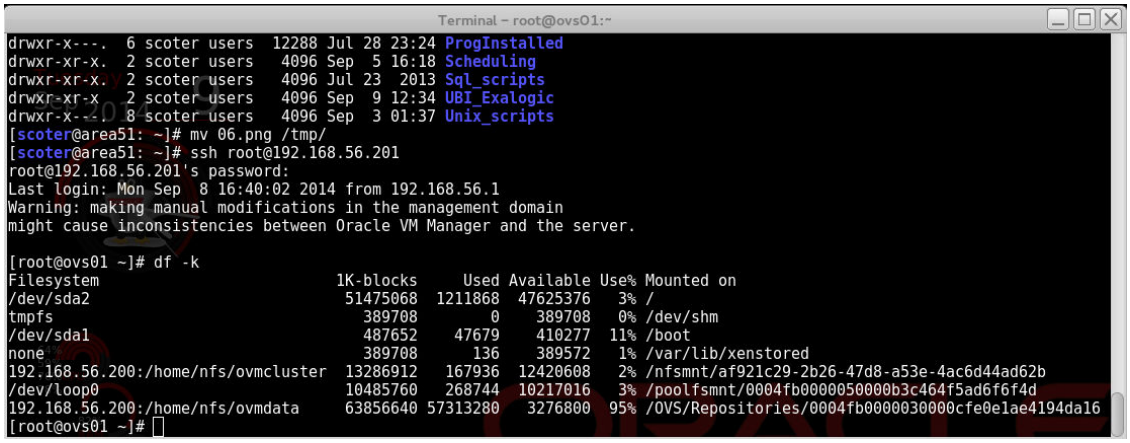


Connect to the Oracle VM Servers and verify that everything is ready to accommodate the lab

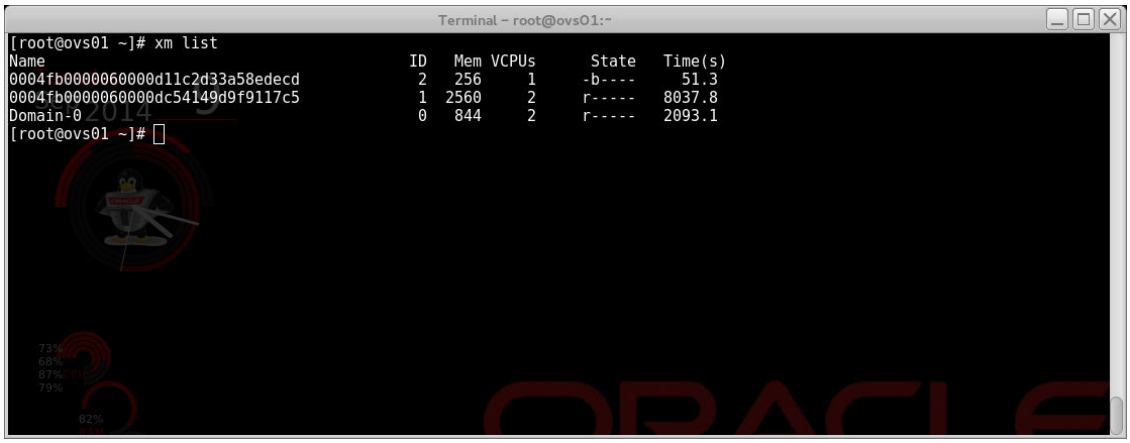
Connect by ssh to **ovs01.oow.local** / **ovs02.oow.local** and verify that Oracle VM guests are really running:
Open a terminal on your linux desktop and execute:

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):



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.



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

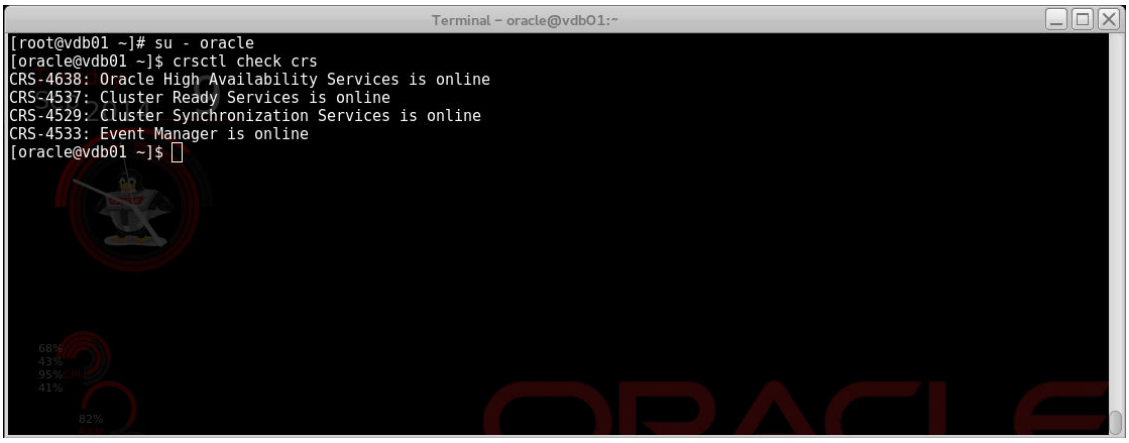
Connect by ssh to **vdb01.oow.local** / **vdb02.oow.local** and verify all Oracle Cluster managed resources are up and running:
Open a terminal on your linux desktop and execute:

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

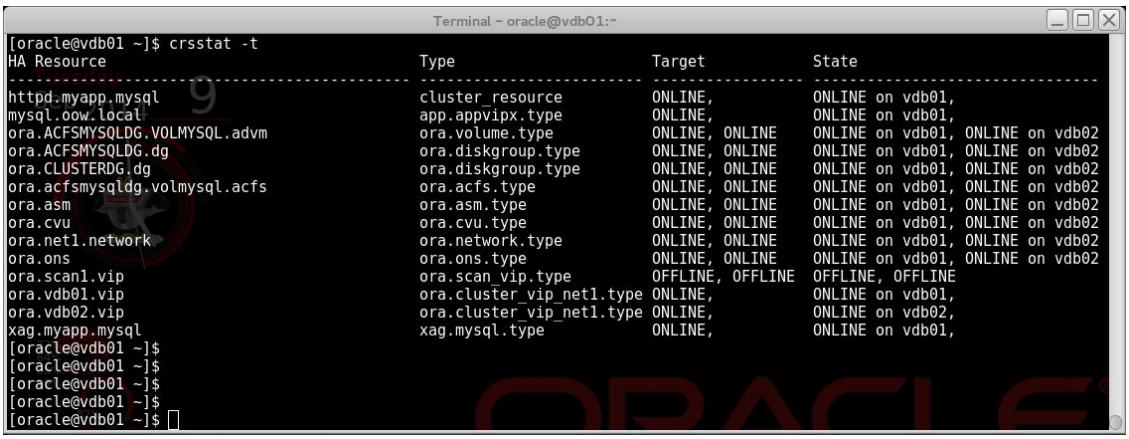
Once connected to both Oracle VM guests, verify that Oracle Clusterware is up and running and, at the same time, that all resource managed by Oracle Clusterware are correctly working (execute the commands on both guests):

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

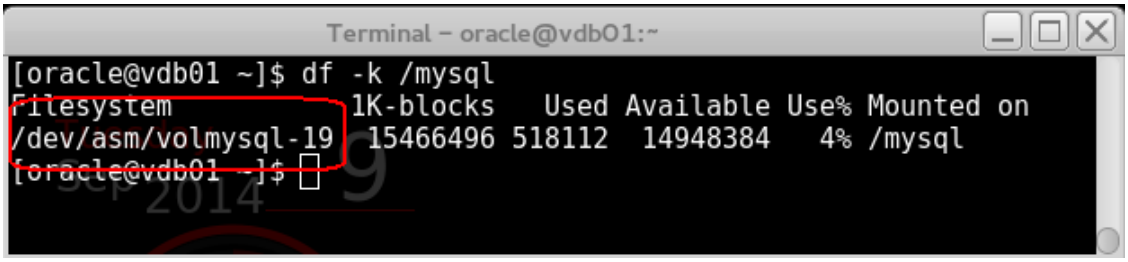
2) Verify that Oracle Clusterware services are up and running
   # crsctl check crs
   The output should be the same as shown in the figure below:



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**".



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

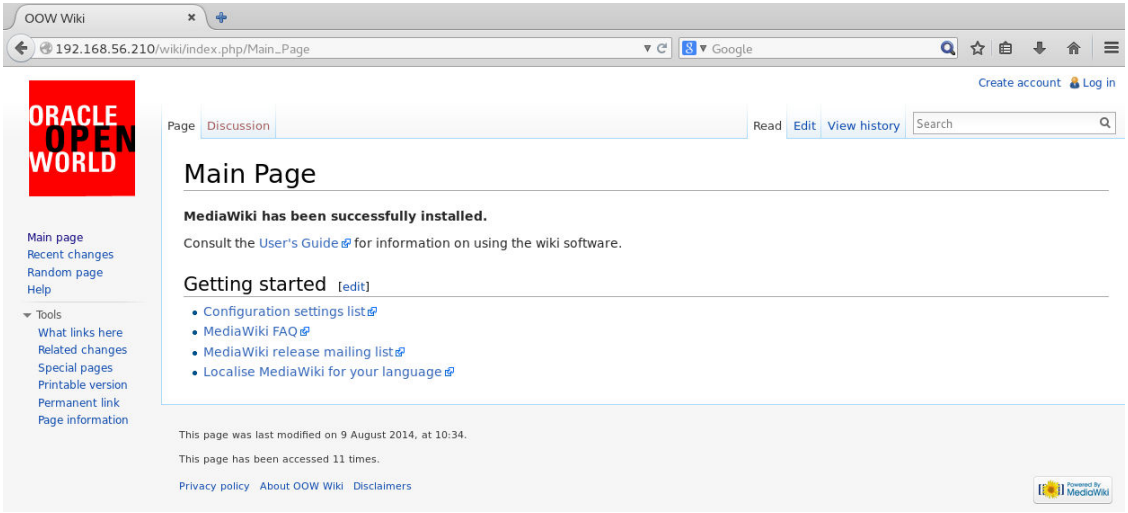


5) Verify that clustered demo web application is running

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

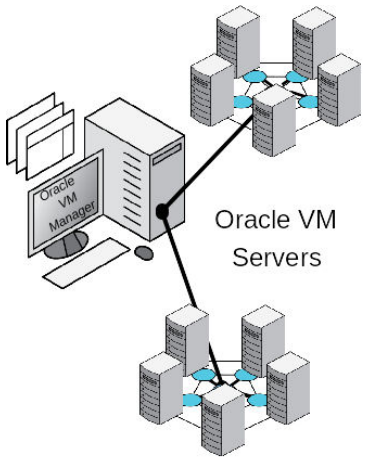
The architecture proposed in this lab is based on:

- 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

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 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

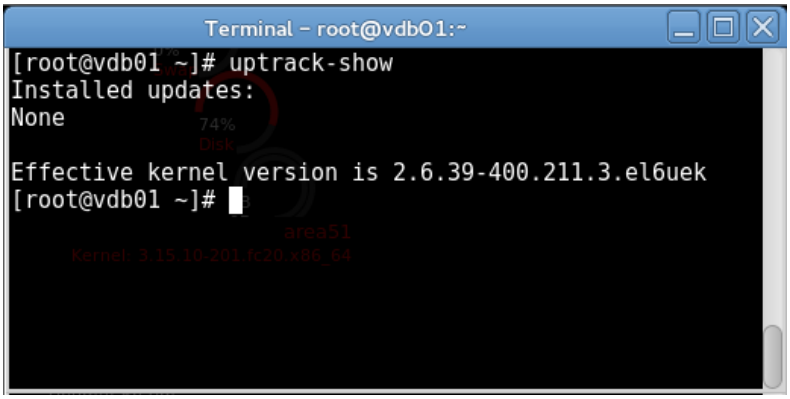
Demonstrate high-availability features covered by Oracle Ksplice

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**".

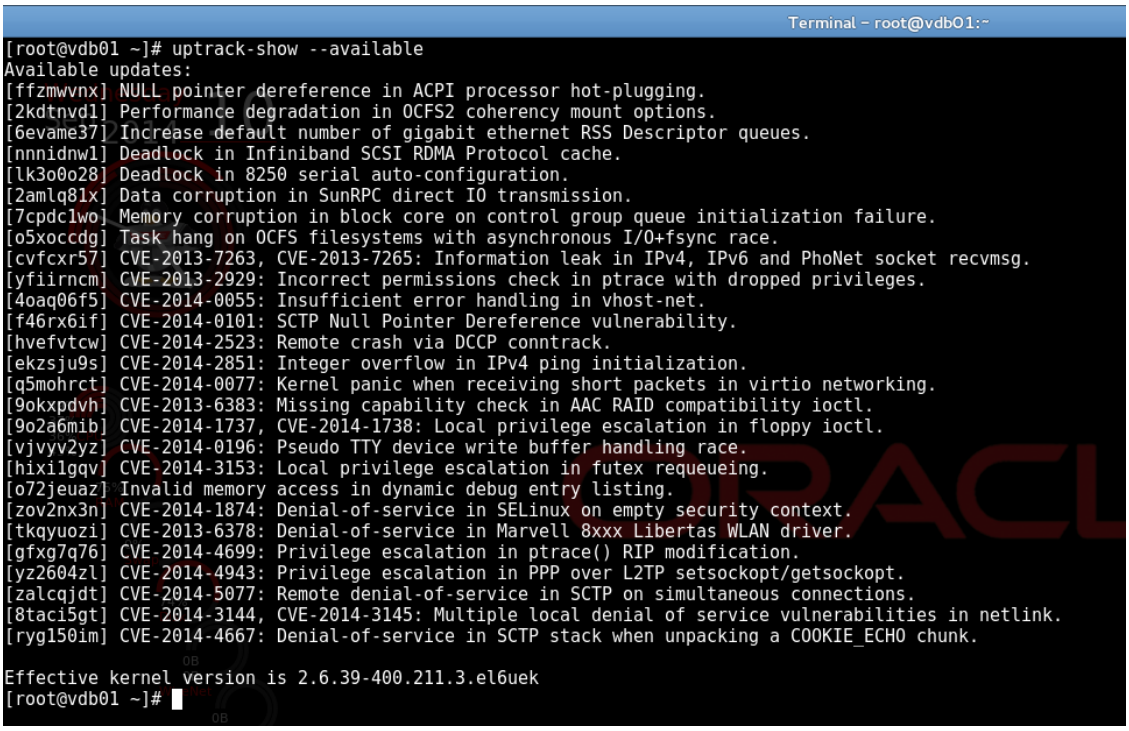
To update kernel on the servers specified above execute the following steps:

1. Connect to by ssh to the server "vdb01.oow.local"
   ssh root@192.168.56.201

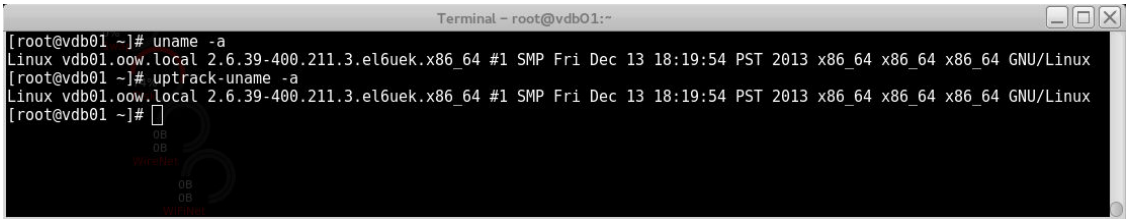
2. Show which Ksplice kernel updates are already installed by executing "**uptrack-show**"



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

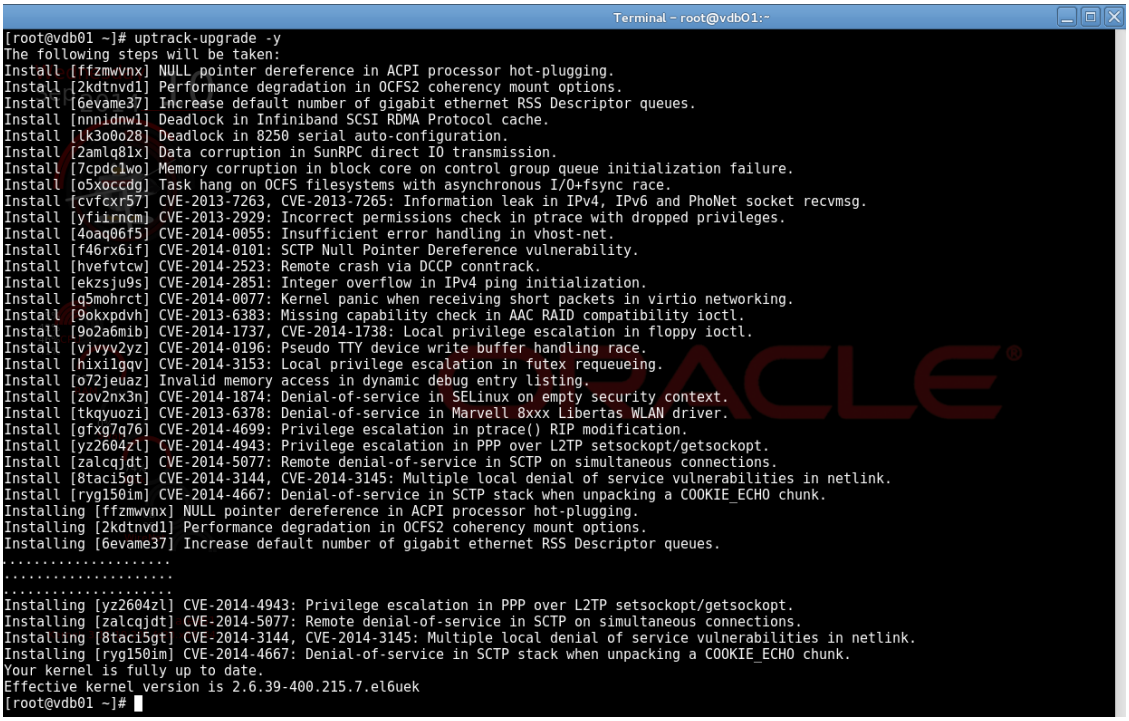


4. Verify actual kernel version installed and actual kernel version in memory by executing “uname -a” and “uptrack-uname -a”.

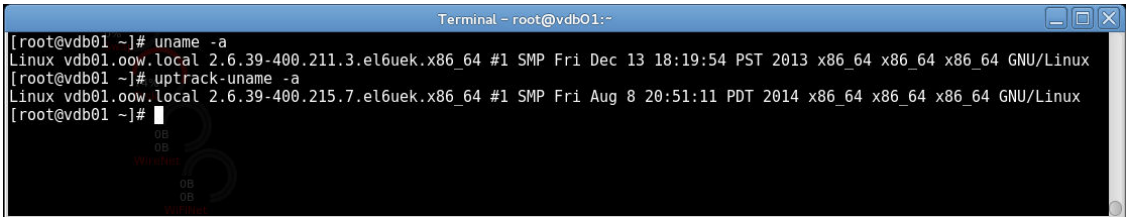


As you can see, actually, kernel installed and kernel-in-memory has the same version 2.6.39-400.211.3

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



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

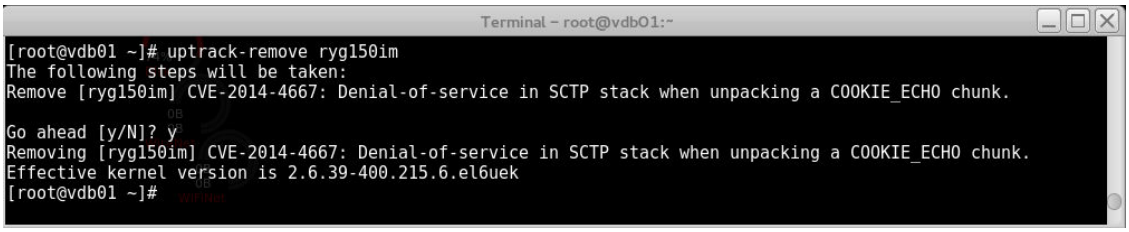


As you can see, actually:

- 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**

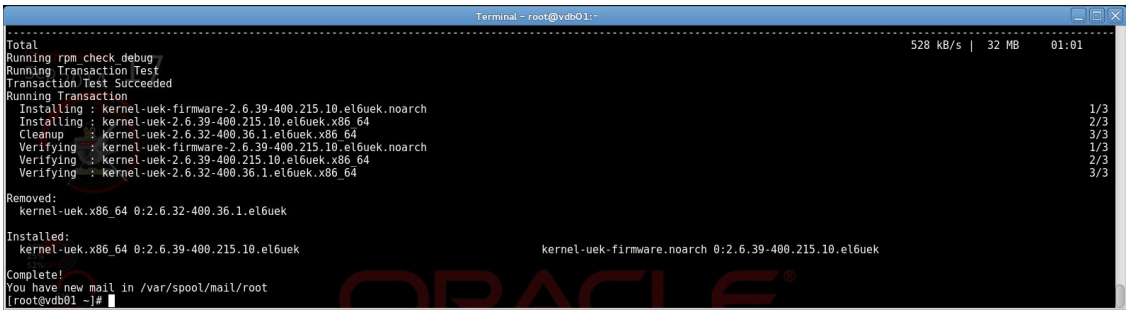


As you can see now the kernel-version moved back to 2.6.39-400.215.6 (it was with a final .7)
This part of the lab demonstrate how, on Oracle Linux, you can update the kernel without reboots; the feature supplied by Ksplice allows you also to install important security kernel fixes without any kind of 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 alows, 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:

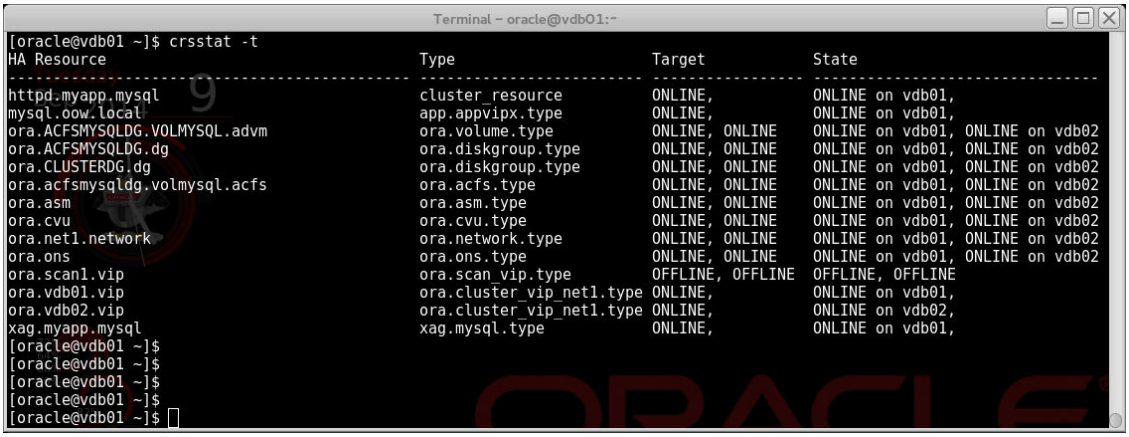
   # yum update kernel-uek kernel-uek-firmware



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).

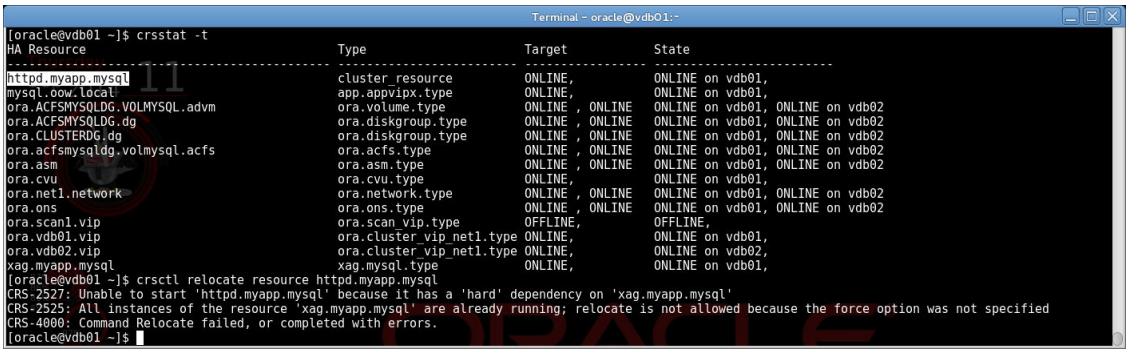
1. Verify actual status of the resources on server "vdb01.oow.local"
   With user "**oracle**" execute the command wrapped script "**crsstat -t**":



   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 message obtained is:

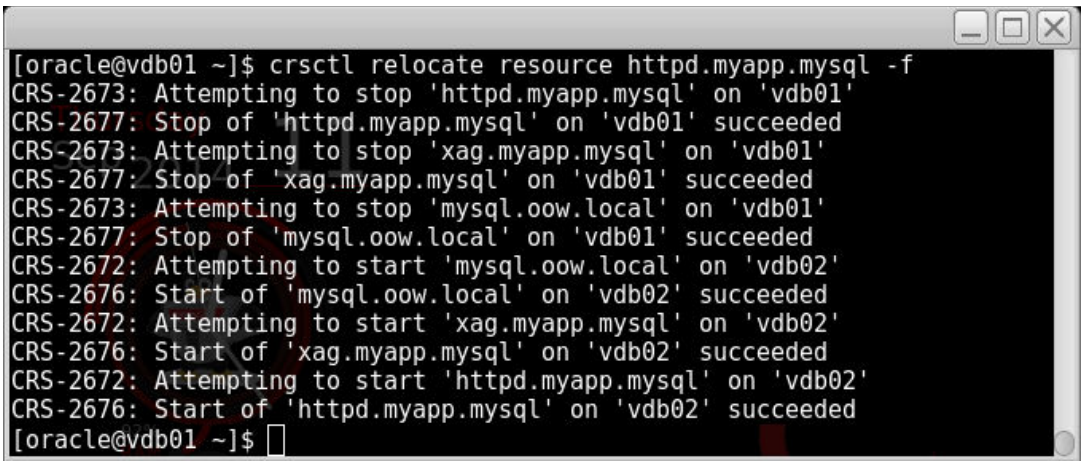
   CRS-2527: Unable to start 'httpd.myapp.mysql' because it has a 'hard' dependency on 'xag.myapp.mysql'
   CRS-2525: All instances of the resource 'xag.myapp.mysql' are already running; relocate is not allowed because the force option was not specified

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.

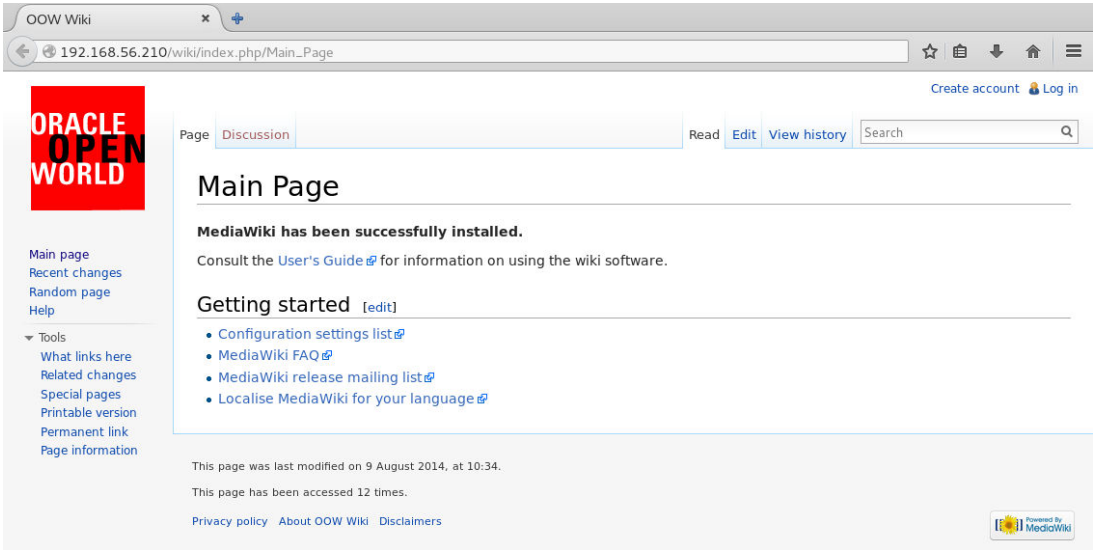
Service resources have the following dependencies:

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)

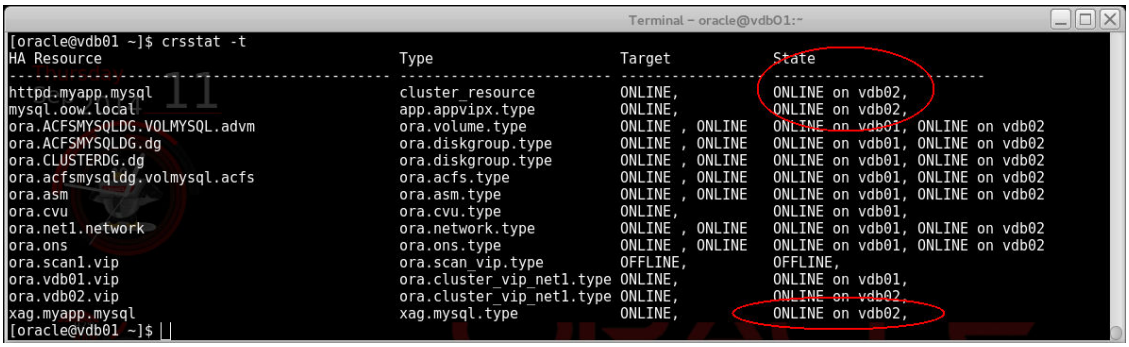


Verify application URL is reachable (you can open from your laptop browser): http://192.168.56.210/wiki



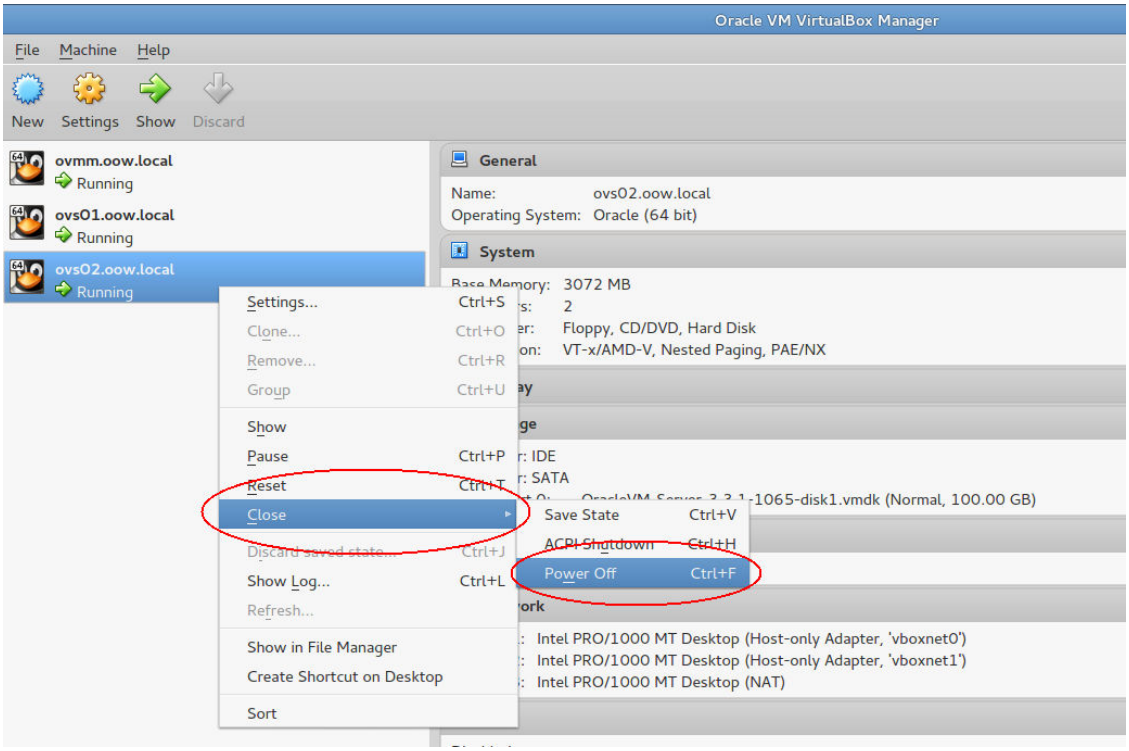
4. Verify actual resource distribution on the cluster with the command “crsstat -t”

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



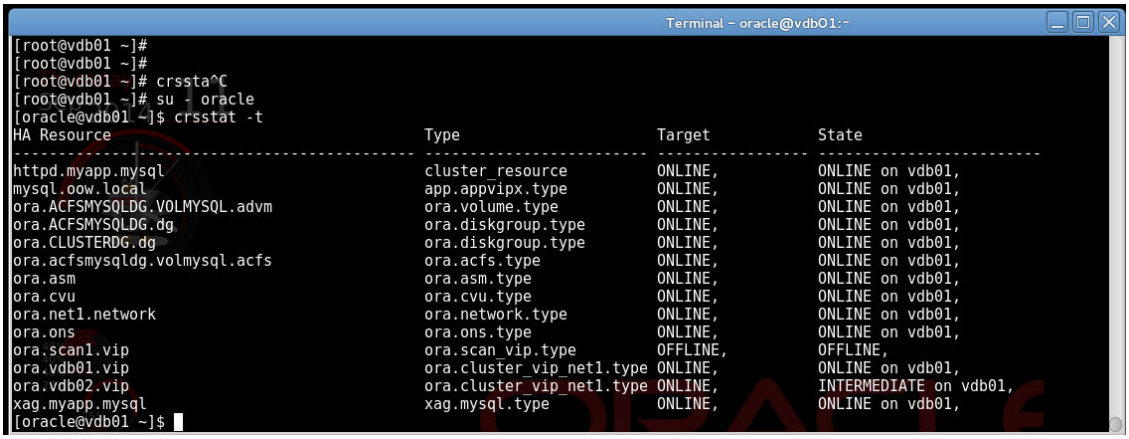
   As you can see service resources are now active on node “**vdb02.oow.local**”.

5. 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**”.

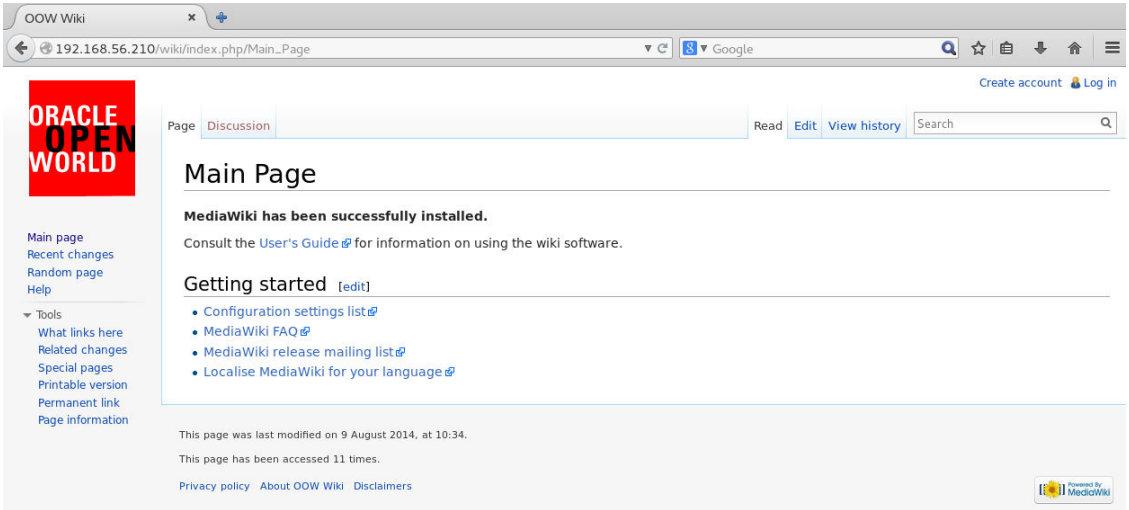


   Confirm to brutally shutdown the server “**ovs02.oow.local**”

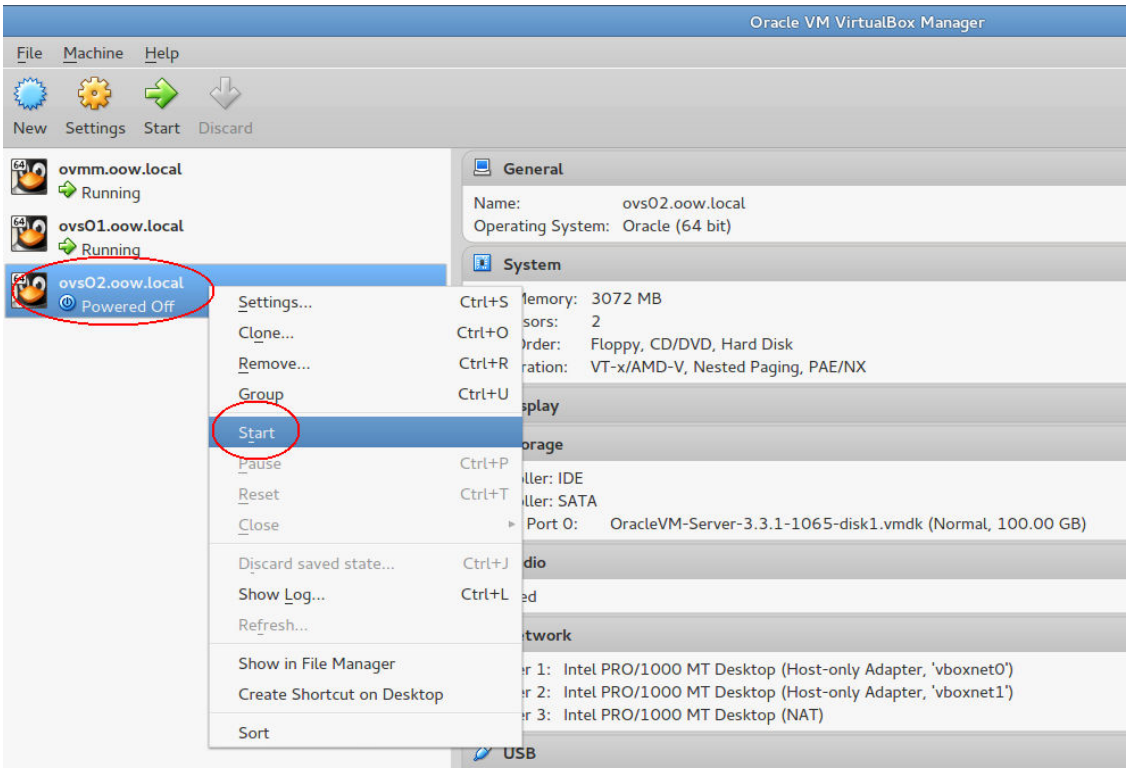
6. Verify actual resource distribution on the cluster with the command “crsstat -t” and, at the same time, with the laptop browser (Mozilla Firefox) verify that web-demo application is available.
You’ll see that, actually, only one node of the cluster is active and that all service resources are active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



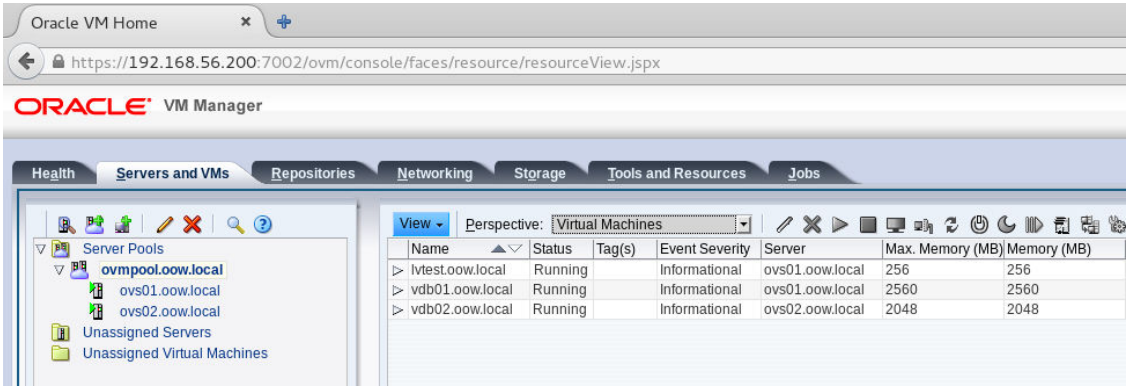
Verify application URL is reachable (you can open from your laptop browser): http://192.168.56.210/wiki



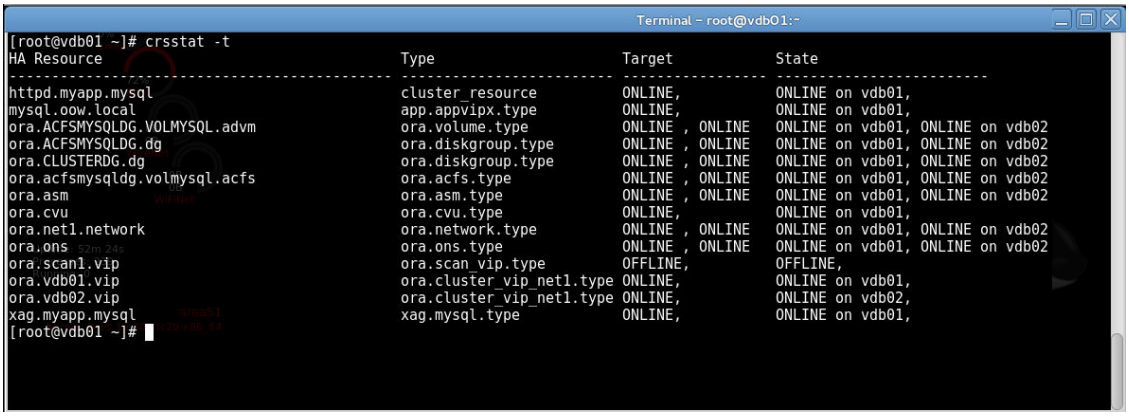
7. Restart VirtualBox server named “ovs02.oow.local”
   Open **VirtualBox console**, right-click on “**ovs02.oow.local**” and the select “**Start**”



8. Conncet 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.

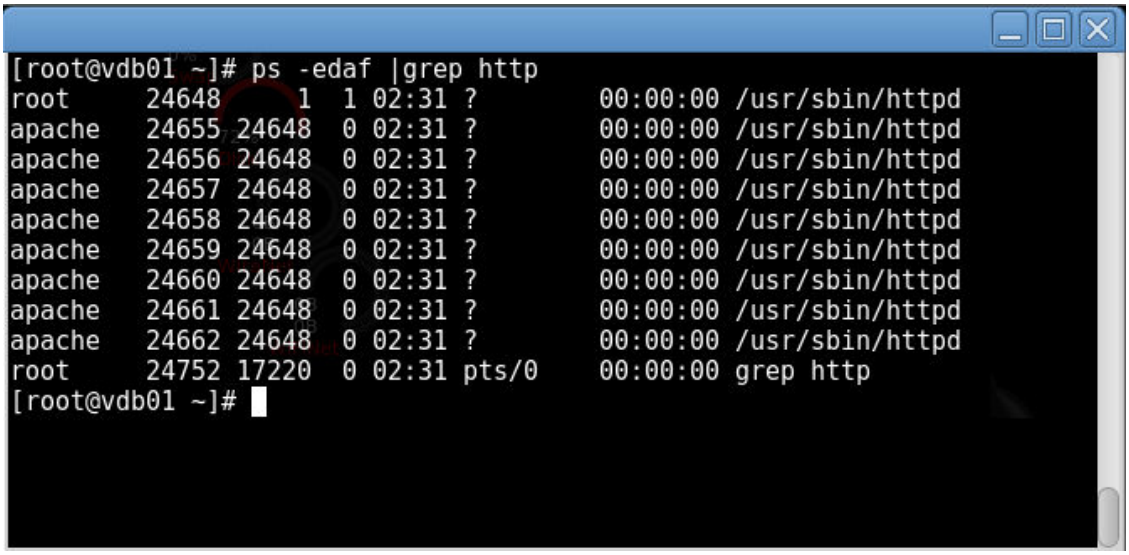


9. Connect to the virtual-guest “**vdb01.oow.local**” (192.168.56.204) by ssh and verify, by clusterware commands, if the “**vdb02.oow.local**” re-joined the cluster.



Wait for “vdb02.oow.local” join the cluster before proceeed 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

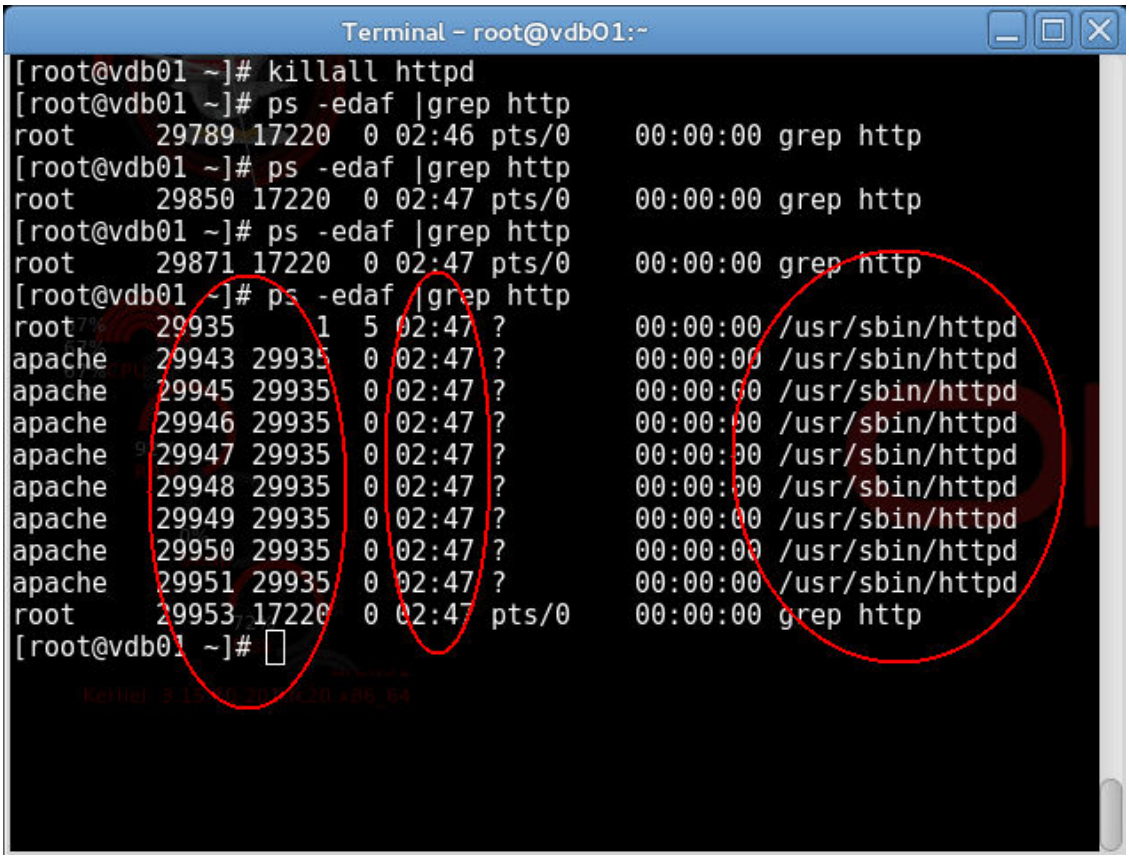


11. Kill all httpd daemon processes, verify that no “httpd” processes are active and wait for the clusterware intervention.
    Oracle Clusterware should take care in some seconds of this faulty situation.
    The sequence of commands to execute, as root, is:

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

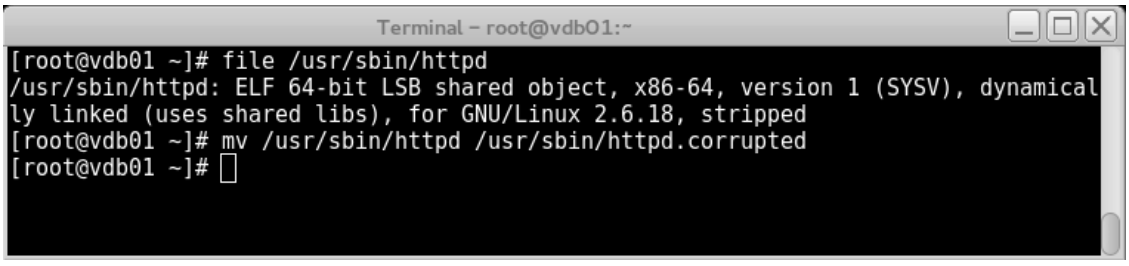
    Initially all httpd processes will die and, after some seconds, Oracle Clusterware will arrange for restart of the daemon; after that verify that web-demo application is available at the following URL:

    http://192.168.56.210/wiki

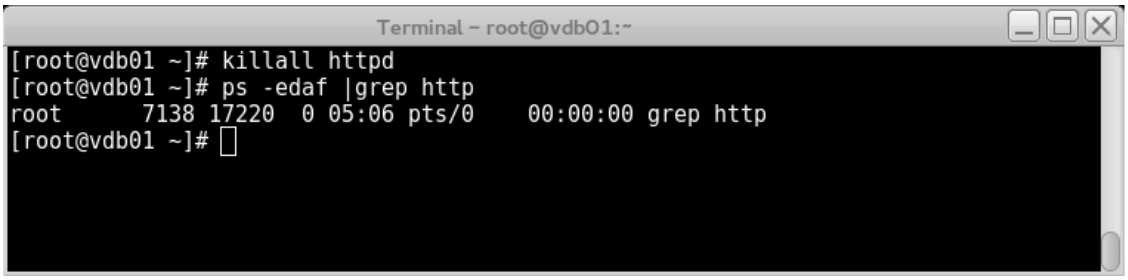


12. With these step we will try do simulate a filesystem corruption with the loss of the binary “httpd”; Oracle Clusterware will be able to identify that it’s not possible to restart the daemon on the current node and will proceed to a complete fail-over of the service (migrating all managed services from node “vdb01” to “vdb02”).
    The steps to execute, as **root**, on node “**vdb01**” (node actually owning all resources) are:

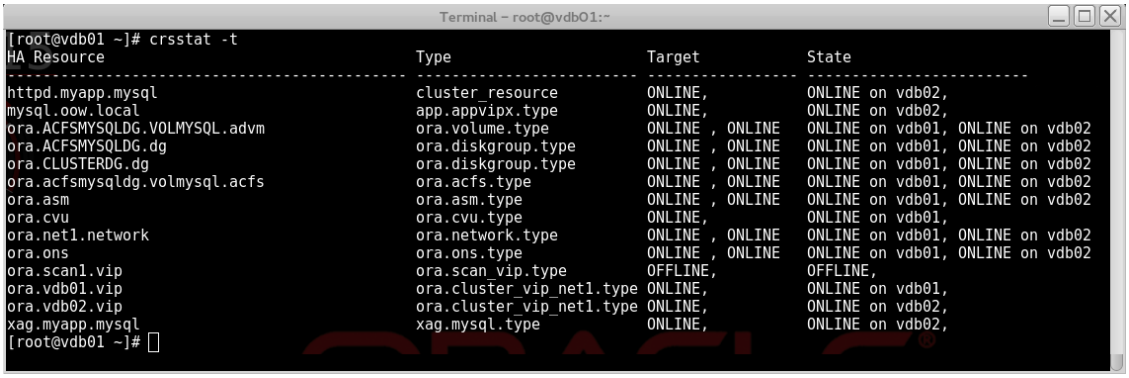
    Rename httpd binary on the filesystem
    # mv /usr/sbin/httpd /usr/sbin/httpd.corrupted



    Kill all httpd processes
    # killall httpd



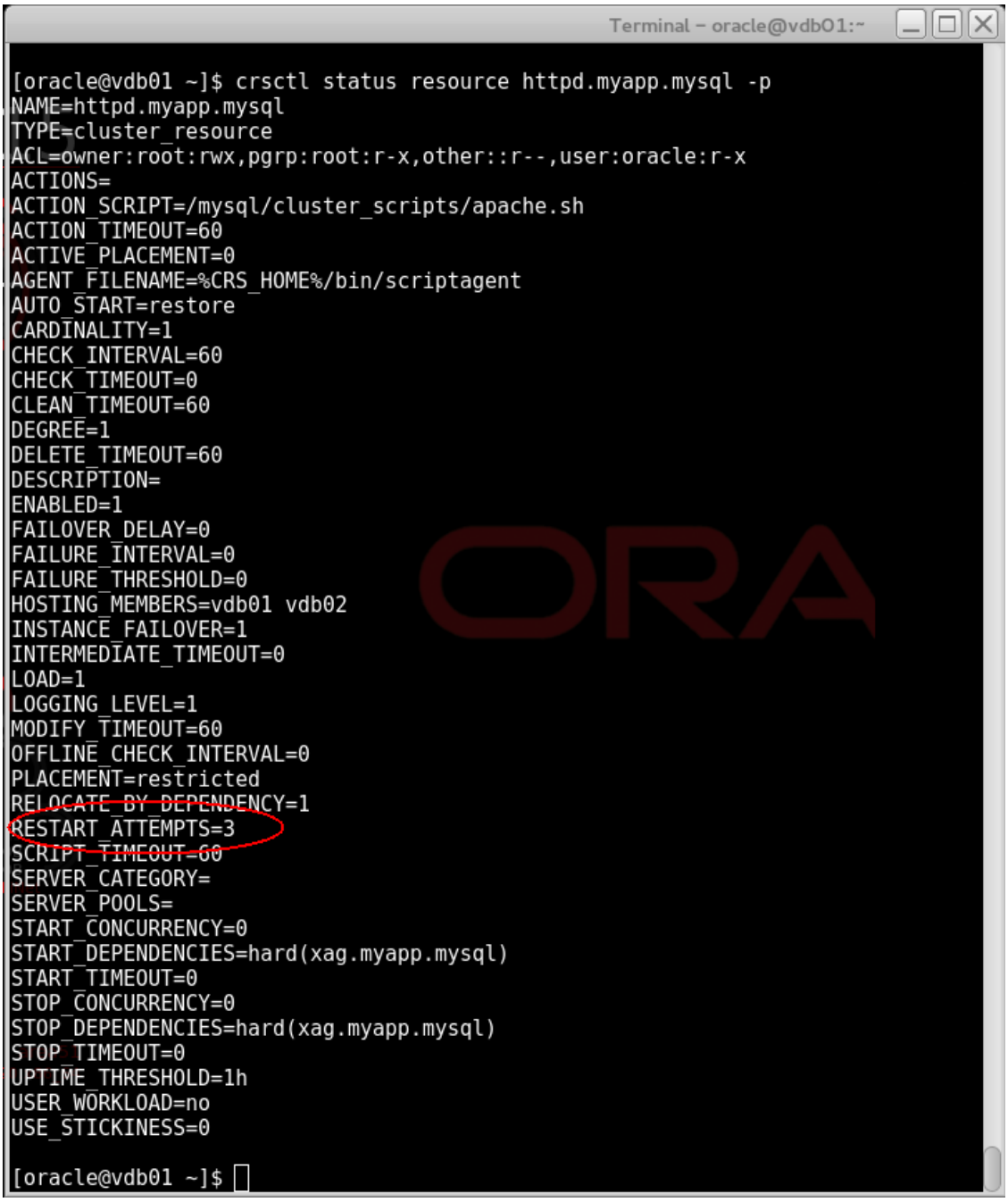
Verify managed service status by the wrapped script (wait until everything is running on node "**vdb02**"):
 # crsstat –t



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.



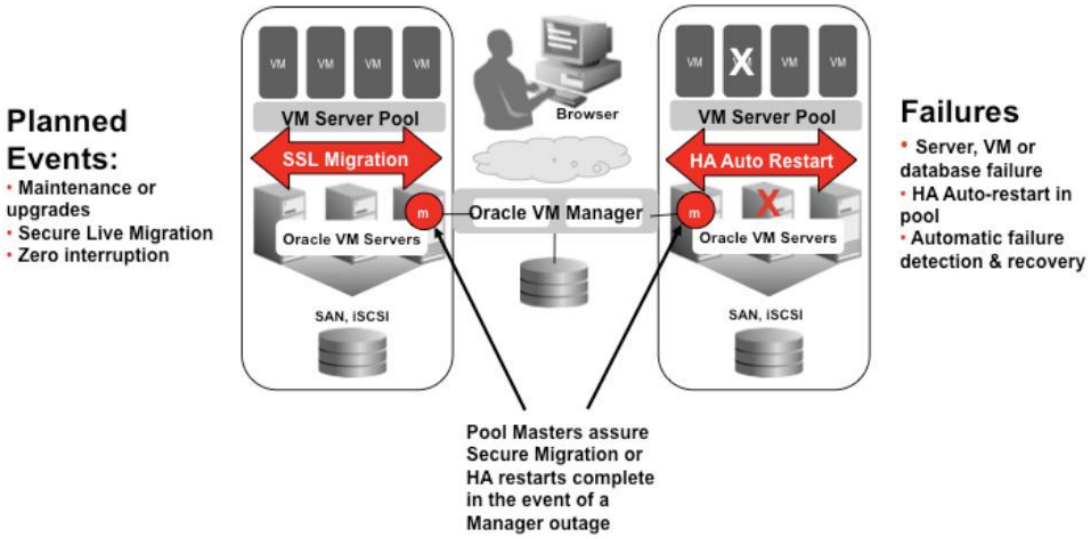
When the test is finished, please fix the simulated corruption on guest "**vdb01.oow.local**", as root, with:
 # mv /usr/sbin/httpd.corrupted /usr/sbin/httpd

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.

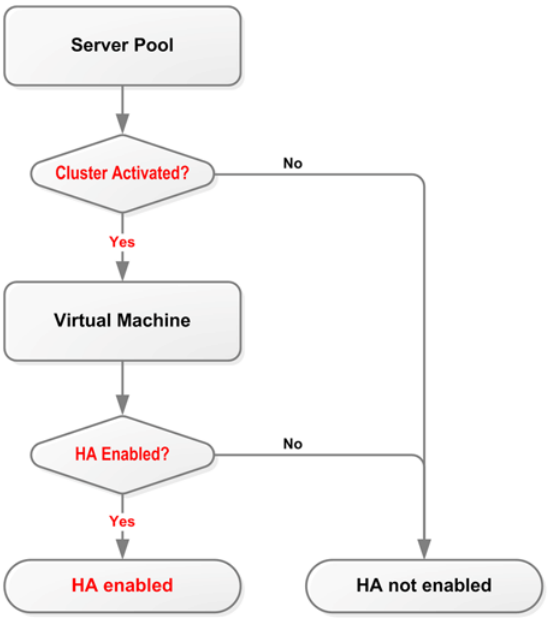
Oracle VM High-Availability

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.

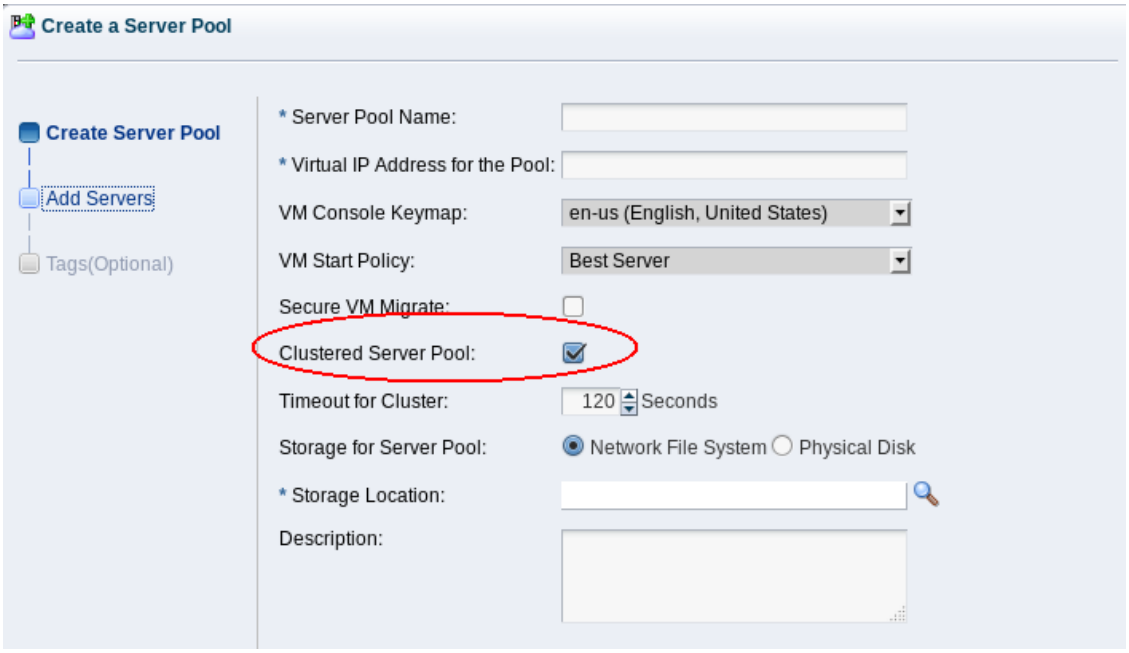
The following prerequisites are requirement to implement HA:

- 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”

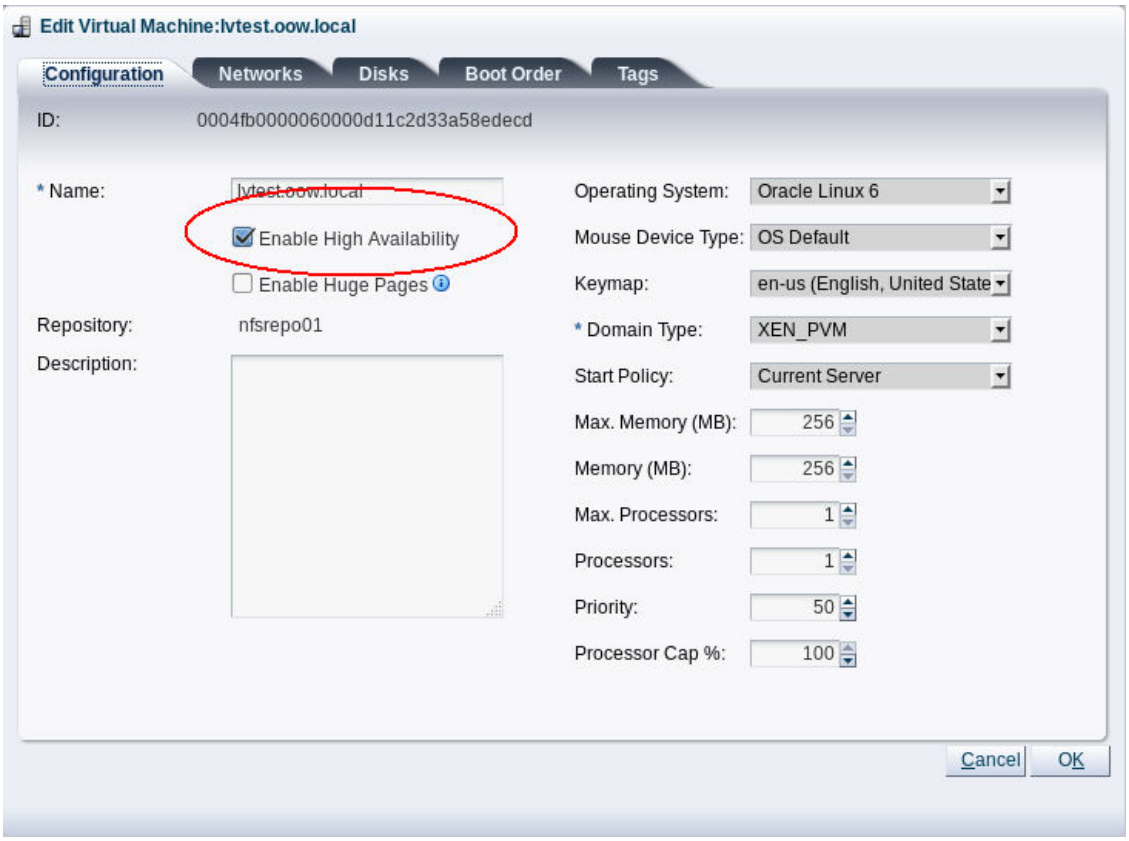
The following chart will better explain requirements of Oracle VM High Availability:



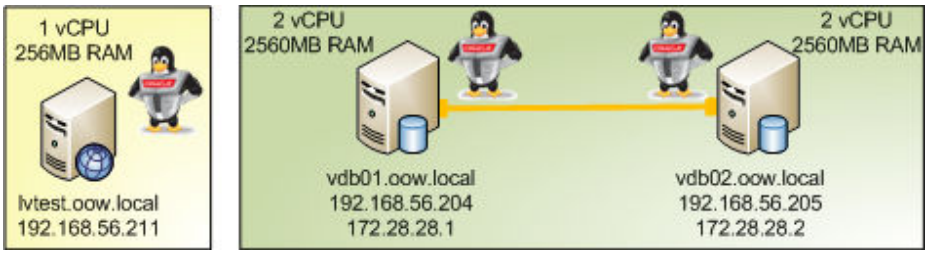
To use HA, you have to configure a clustered Oracle VM Server Pool:



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):

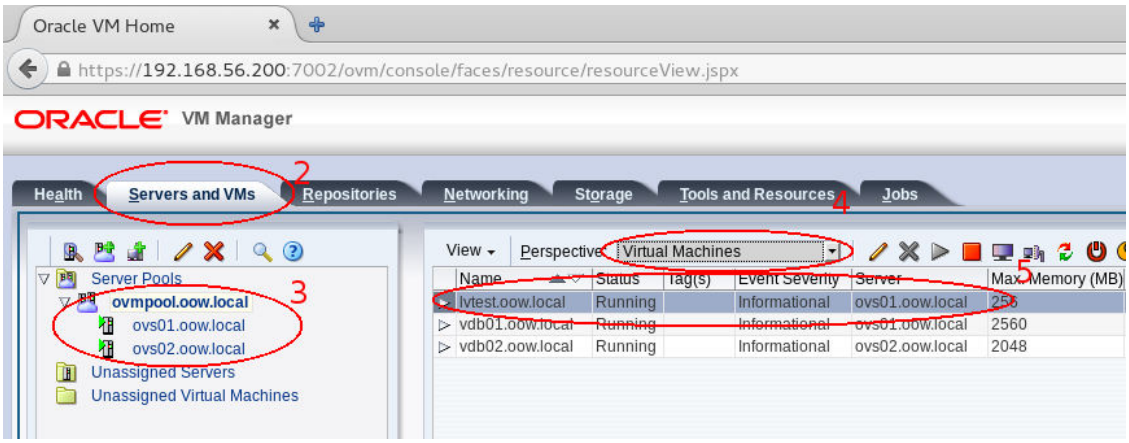


On our H/A tests, due to the reduced amount of resources available on our system,(mainly RAM), we will demonstrate both Oracle VM High-Availability and Oracle VM Live-Migrate using the guest named "**lvtest.oow.local**".

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

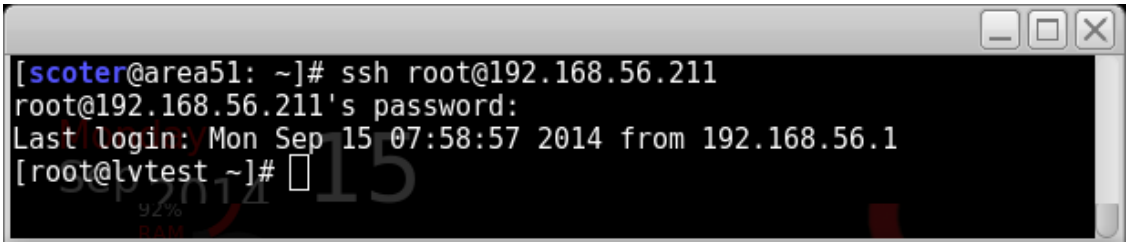
1. Connect to Oracle VM Manager and verify the status of the guest named "lvtest.oow.local".
   Oracle VM Manager URL: https://192.168.56.200:7002/ovm/console
   Username: admin
   Password: Welcome1
2. Select "**Servers & VMs tab**"
3. Expand and select pool named "**ovmpool.oow.local**"
4. Select perspective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

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

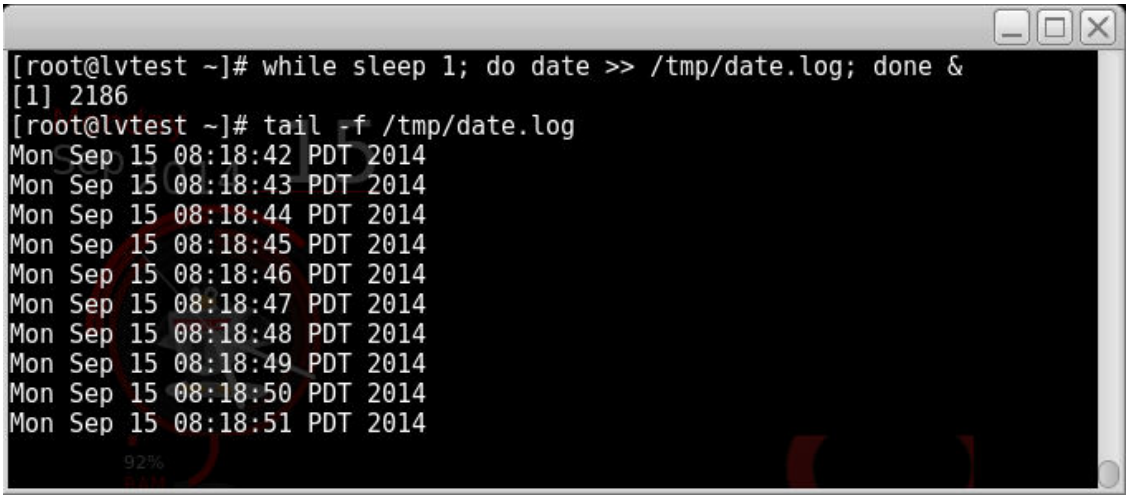


In the case above the guest "**lvtest.oow.local**" is running on Oracle VM Server "**ovs01.oow.local**".

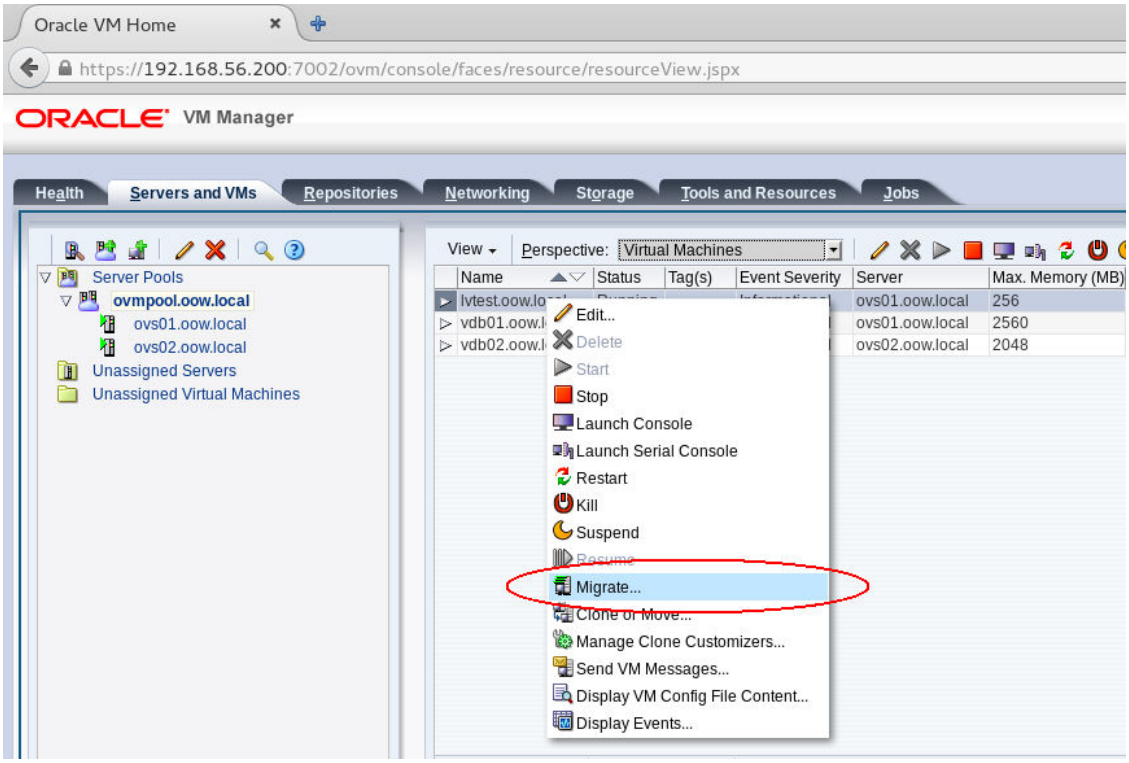
6. Connect, by ssh, to the server "**lvtest.oow.local**"
ssh root@192.168.56.211 (password is ovsroot)



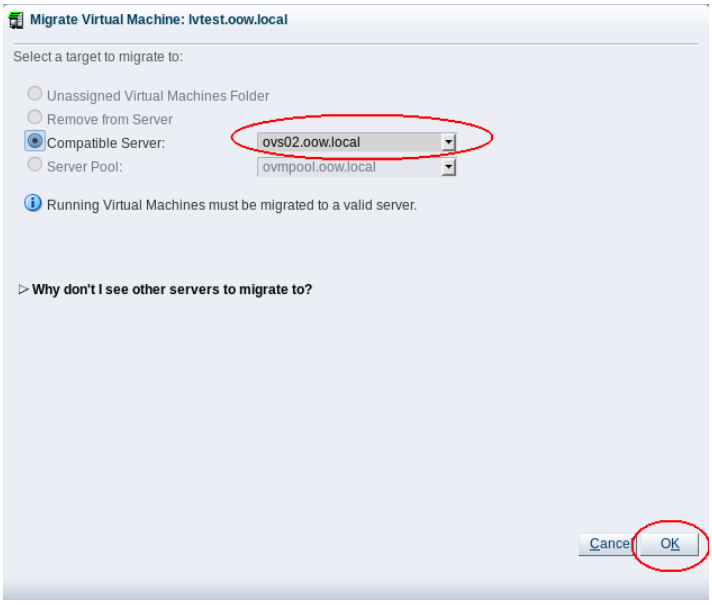
7. Launch a command that will show the date of the server updated every one second.
 # while sleep 1; do date >> /tmp/date.log; done &
 # tail -f /tmp/date.log



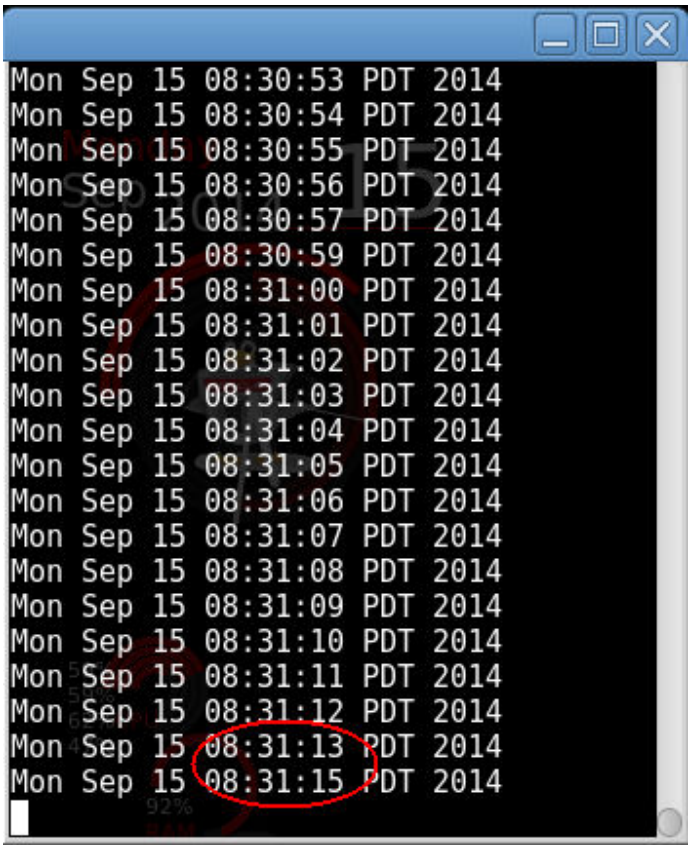
8. While this script is running, execute the live-migrate of the guest by Oracle VM Manager.
Re-open the **Oracle VM Manager console**, select guest named "**lvtest.oow.local**", **right-click** and select "**Migrate**".



9. Select "compatible-server" "ovs02.oow.ocal" and confirm with "OK".

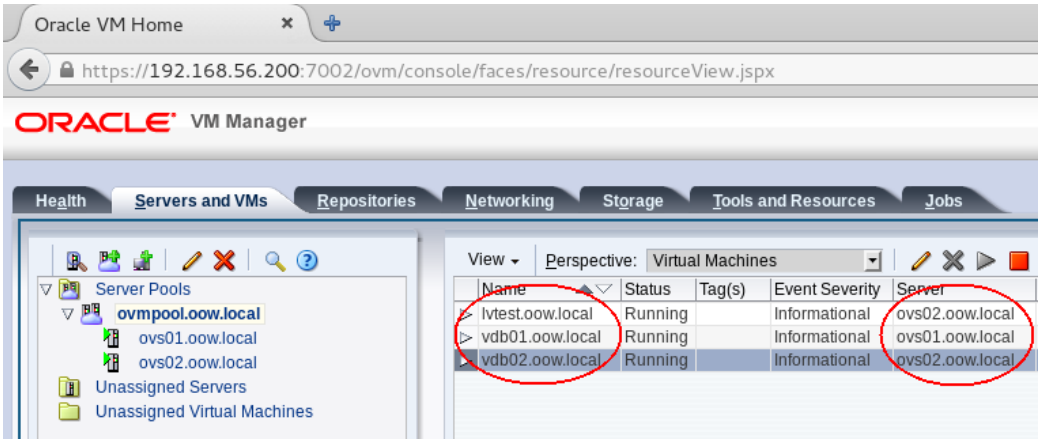


10. While migration is running, verify the output of the command executed at point (7) of this section and verify the sequentiality of the output; *in the worse case you'll lost only 1 second of the output and this is a very nice result for a demo environment built on top of VirtualBox.*



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

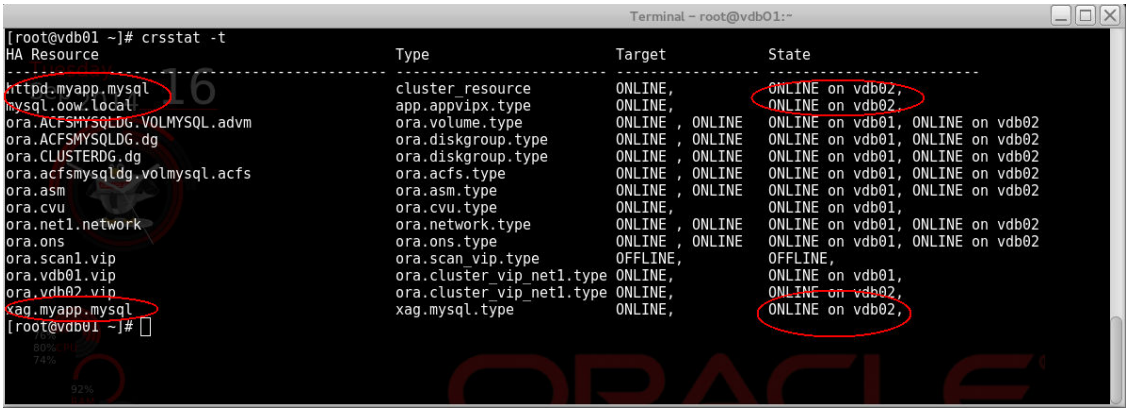
1. Connect to Oracle VM Manager and verify:
   - All three guests are in "Running" state
   - Which physical server owns the guest named "lvtest.oow.local"



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

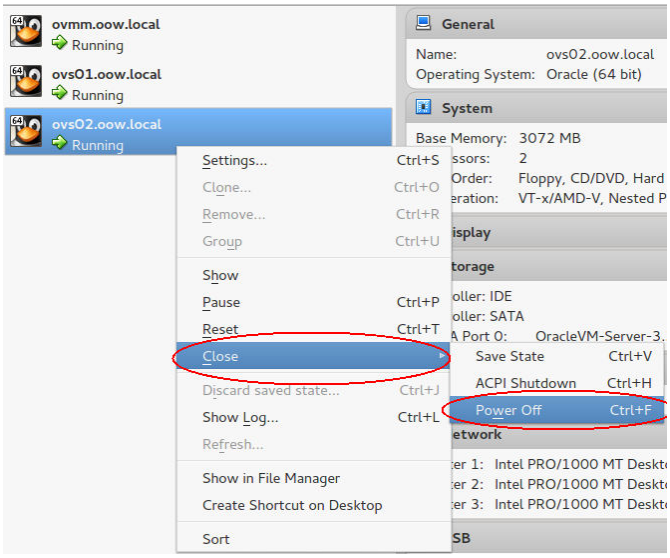


In the case above we have that:

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

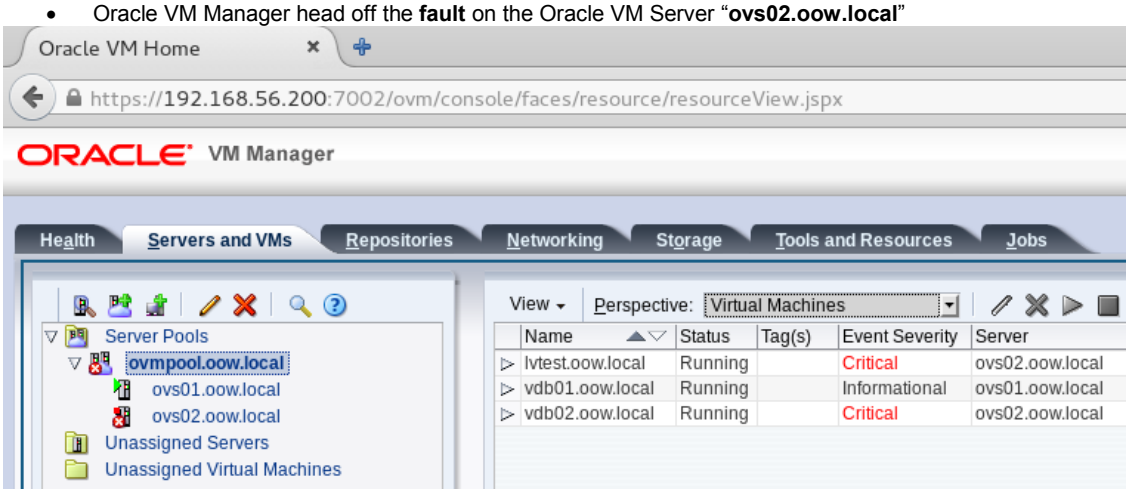
The target of this test is to **simulate a crash of the Oracle VM Server that owns**, at the sames time, both guests **lvtest** and **vdb0(?)** that owns the **web/mysql services** (if you need to move services between cluster-nodes, you can use the steps reported in this document at the section “Demonstrate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

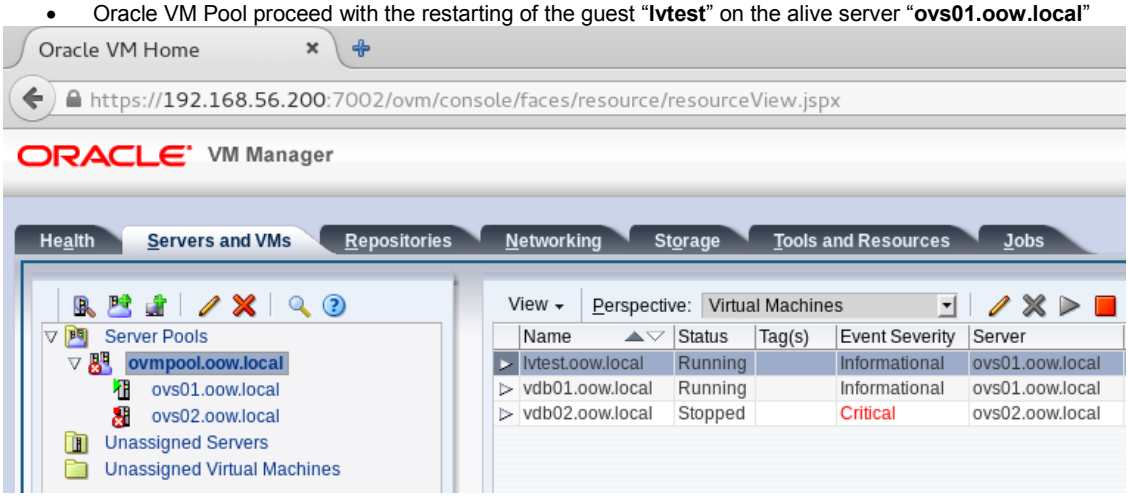


Next steps that will happens on the environment will be:

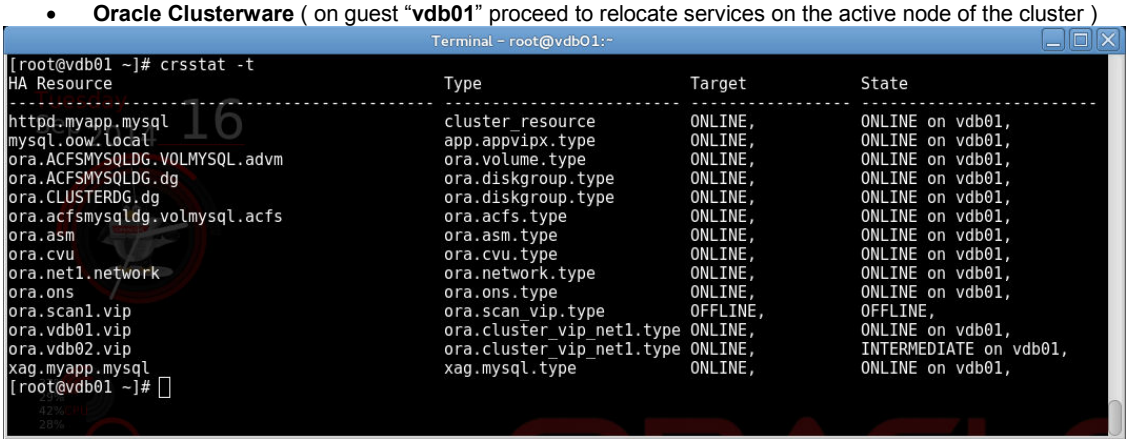
- Oracle VM Manager head off the **fault** on the Oracle VM Server "**ovs02.oow.local**"



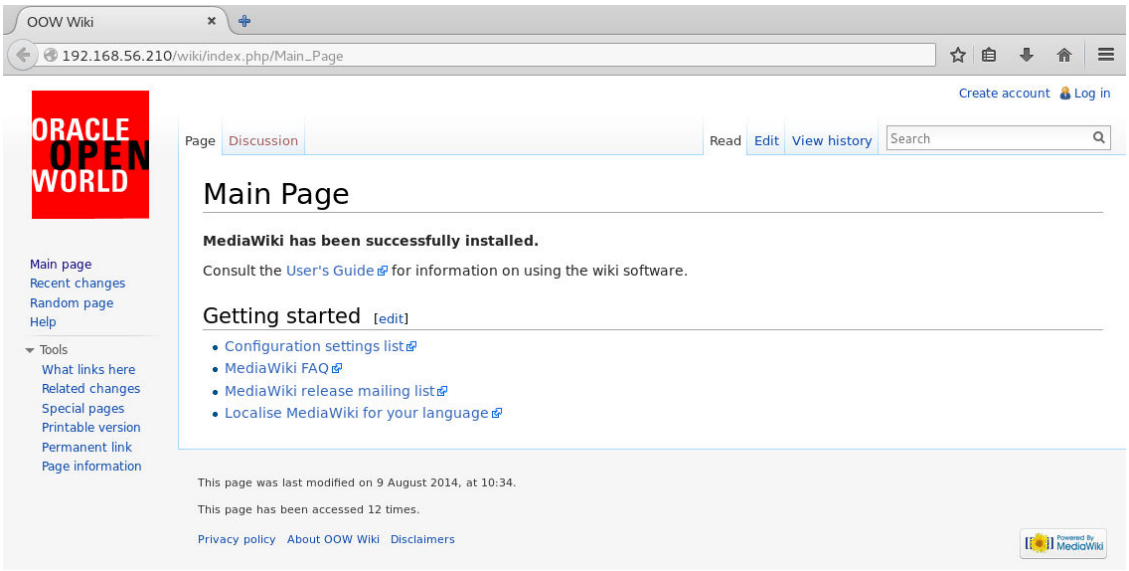
- Oracle VM Pool proceed with the restarting of the guest "**lvtest**" on the alive server "**ovs01.oow.local**"



- **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
facebook.com/oracle
twitter.com/oracle
oracle.com

Hardware and Software, Engineered to Work Together