

Create your own Template for MySQL using Oracle VM 3.3.1 Hands-On Lab

In this session, you'll learn how to create your own Oracle VM Template for your application. You will start with a template that includes Oracle Linux as the guest Operating System and create your own template with automatic deployment and configuration of MySQL 5.6.

A great lab to see how you can harness the power of Oracle VM Templates.

BY:

Olivier Canonge, Sales consultant, Oracle France

Christophe Pauliat, Sales consultant, Oracle France

Bruno Bottreau, Sales consultant, Oracle France

Simon Coter, Technical expert in core technology, Oracle Italy

Doan Nguyen, Principal Product marketing Director, Oracle USA



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
PRE-WORK TO BE DONE	Error! Bookmark not defined.
LAPTOP ENVIRONMENT	Error! Bookmark not defined.
WHAT YOU WILL LEARN	4
GLOBAL LAB PICTURE	4
START OVM SERVER AND OVM MANAGER	5
CONNECT TO THE ORACLE VM MANAGER CONSOLE	6
DISCOVER THE ORACLE VM SERVER	7
DEFAULT MANAGEMENT NETWORK	9
CREATE A SERVER POOL	9
CREATE A STORAGE REPOSITORY	11
CLONE AND START YOUR ORACLE LINUX VIRTUAL MACHINE	13
CONFIGURE ORACLE LINUX 6.5 AS AN ORACLE VM TEMPLATE	16
CLONE THIS VM AS AN ORACLE VM TEMPLATE	21
CLONE YOUR OWN ORACLE VM TEMPLATE AS A NEW VM	23
START YOUR NEW BLANK VM FOR FIRST CONFIGURATION	24
INSTALL MYSQL RPMs	25
BUILDING THE TEMPLATE INITIALIZATION SCRIPTS	26
TESTING TEMPLATES INITIALIZATION SCRIPTS	28
CONVERTING THE VM TO A TEMPLATE	30
CLONE YOUR OWN ORACLE VM MYSQL TEMPLATE AS A NEW VM	31
CONGRATULATIONS	34

Last update: September 22, 2014

Author: Olivier Canonge

Special thanks to: Gregory Verstraeten, Bruno Bottreau , Christophe Pauliat, Simon Coter, Doan Nguyen, Jean-Philippe Pinte

PREPARATION (Must be done before lab)

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

There are 2 options to perform the pre-lab set up.

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

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

OVA1: HOL9121_blankserver.ova 5.8G

OVA2: HOL9121mgr.ova 25G

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

Pre-lab Set Up:

Install Oracle Linux 6.5 (64 bits) on all the laptops.

Install Oracle VirtualBox 4.3.12r93733 + extensions on all the laptops.

Install an Oracle VM Manager 3.3.1 server in a VirtualBox virtual machine (HOL9121_ovm_mgr).

Install an Oracle VM Server 3.3.1 server in a VirtualBox virtual machine (HOL9121_ovm_srv).

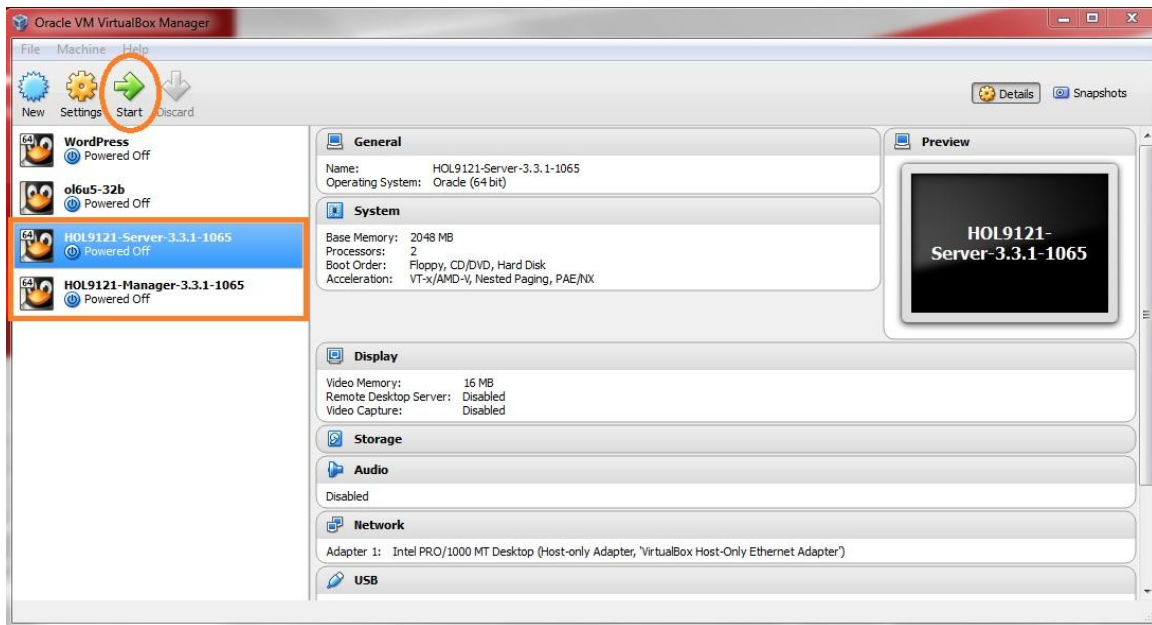
Note:

To run this lab at home or office you will need an X86 machine with 8GB of RAM and 4 cores CPU.

INITIALIZE THE ENVIRONMENT

To save time, the first thing to do is to start both VirtualBox VMs (Oracle VM Manager and Oracle VM Server) :

- Select the VM called “HOL9121-OracleVM-Manager-3.3.1-1065” and click on  to start it
- Select the VM called “HOL9121-OracleVM-Server-3.3.1-1065” and click on  to start it



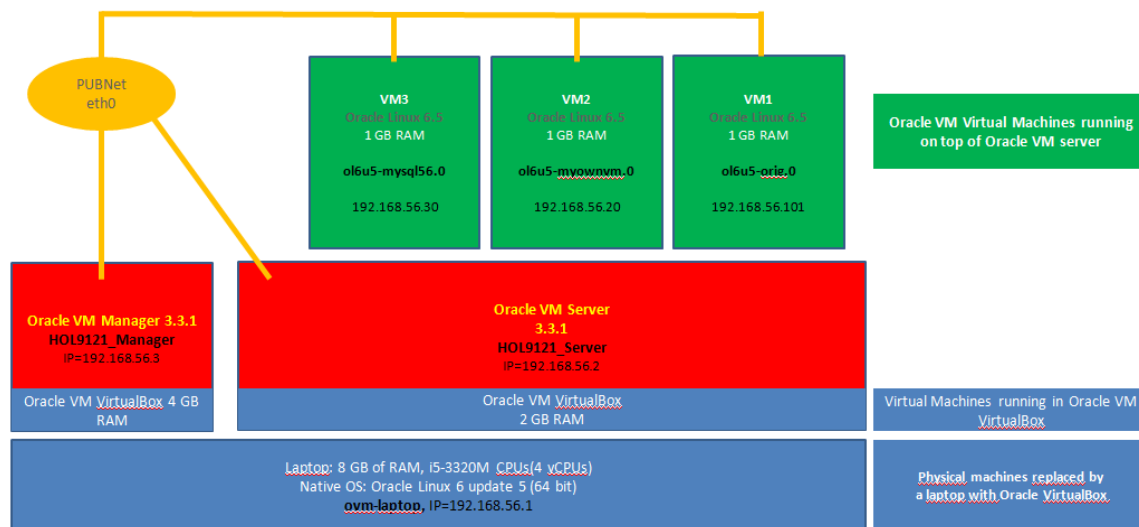
WHAT YOU WILL LEARN

In this lab, you will learn some basics on Oracle VM and execute following steps:

- 1) How to discover an Oracle VM Server
- 2) How to start an Oracle VM virtual machine from the OVM Manager
- 3) How to install Oracle VM API packages
- 4) How to create a Oracle VM template from an Oracle VM virtual machine
- 5) How to use Oracle VM messages to configure an Oracle VM virtual machine
- 6) How to use Oracle VM Command Line Interface to configure an Oracle VM virtual machine
- 7) How to create your own Oracle VM template for automatic deployment and configuration of MySQL5.6

GLOBAL LAB PICTURE

The following picture shows all the components (VirtualBox and Oracle VM virtual machines) with their names and configuration (memory, IP address, network...)



START Oracle VM SERVER AND Oracle VM MANAGER

As previously explained, we will use Oracle VirtualBox to host the 2 servers (Oracle VM Server, Oracle VM Manager) on a single laptop.

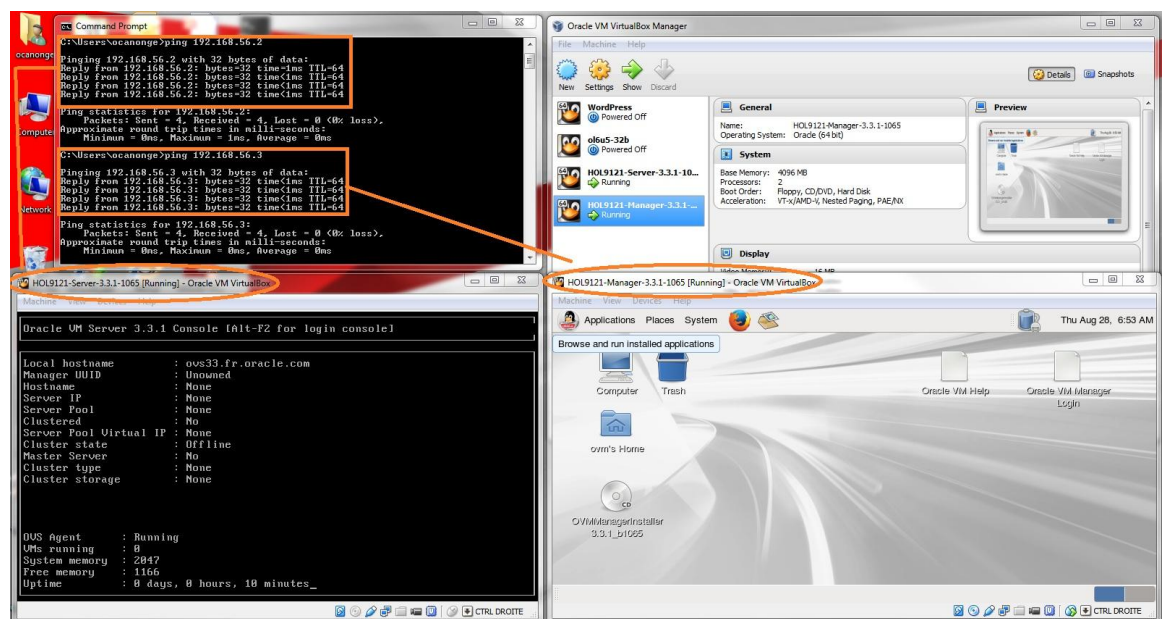
Both VMs should have been started previous section (PRE-WORK TO BE DONE), if not please start both VMs as described there

TO DO :

- Wait for both VMs to be fully started
 - Wait for the Oracle Linux display screen on the VM HOL9121_Server
 - Wait for Oracle VM server display screen on the VM HOL9121_Manager
 - Open a Terminal windows and check you are able to ping both VMs :

HOL9121_Manager : 192.168.56.3

HOL9121_Server : 192.168.56.2



Once both VMs are started and you have checked ping is OK you can :

- Minimize the main VirtualBox program window
- Minimize the Oracle VM Manager VirtualBox window
- Minimize the Oracle VM Server window

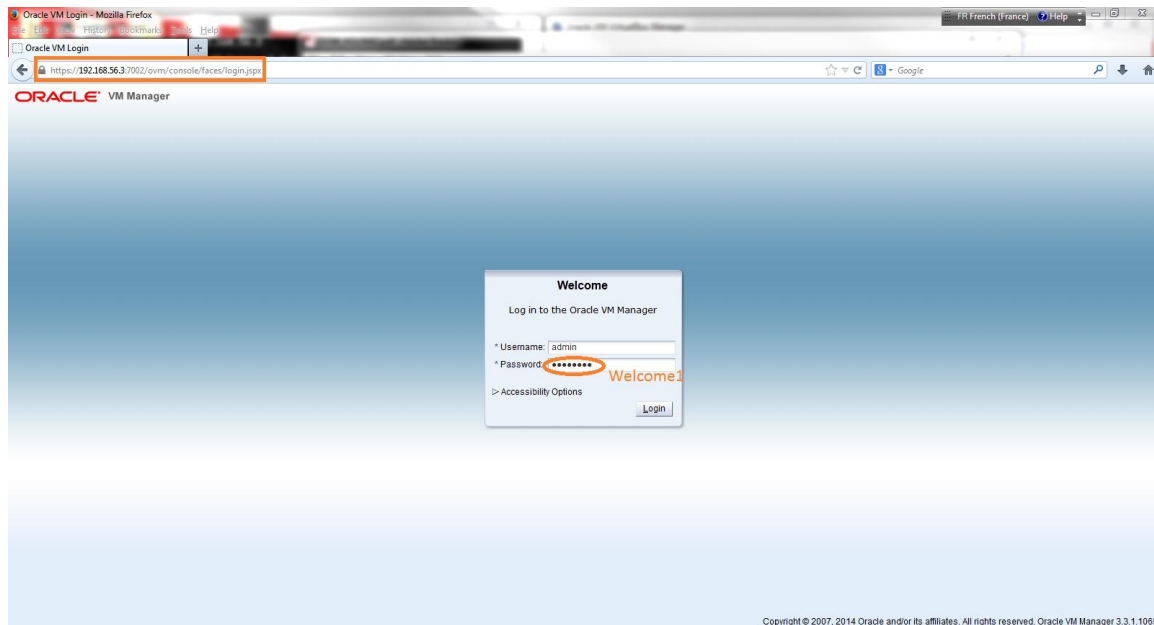
All next steps will be done from your laptop native OS.

CONNECT TO THE ORACLE VM MANAGER CONSOLE

TO DO :

- On your Linux physical desktop open a Firefox browser and connect to the Oracle VM Manager console using URL <https://192.168.56.3:7002/ovm/console>

You should get the following login page :



Log in using the following credentials:

- Login : **admin** (Oracle VM Manager Administrator)
- Password : **Welcome1**

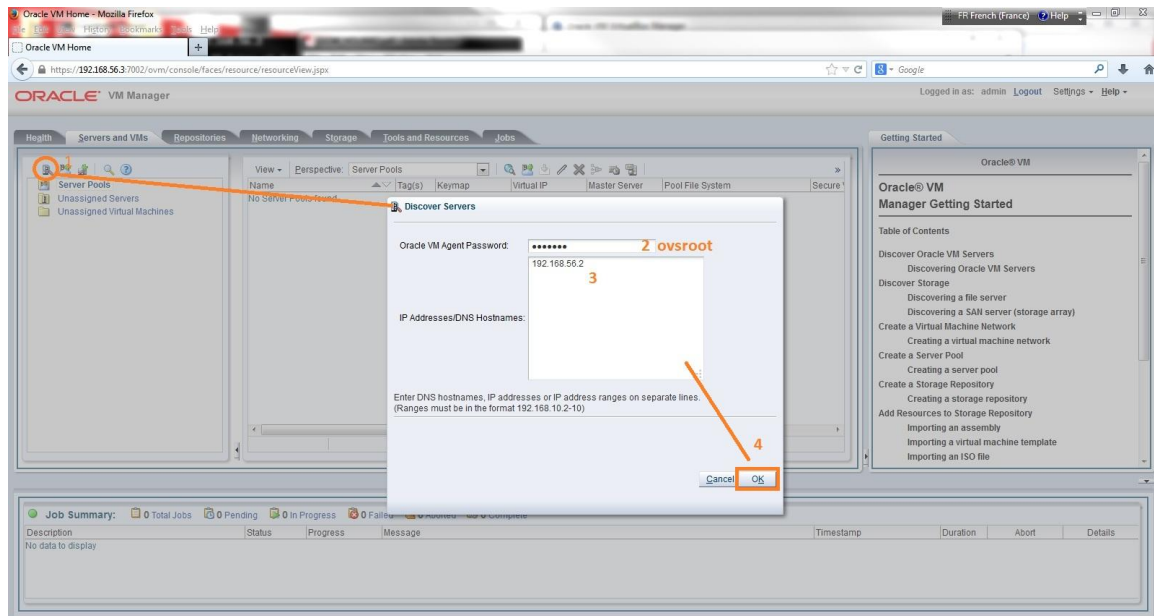
DISCOVER THE ORACLE VM SERVER

When you add Oracle VM Servers to your Oracle VM Manager environment, this process is known as discovering Oracle VM Servers. The first thing you should do to set up your virtualization environment is to discover your Oracle VM Servers.

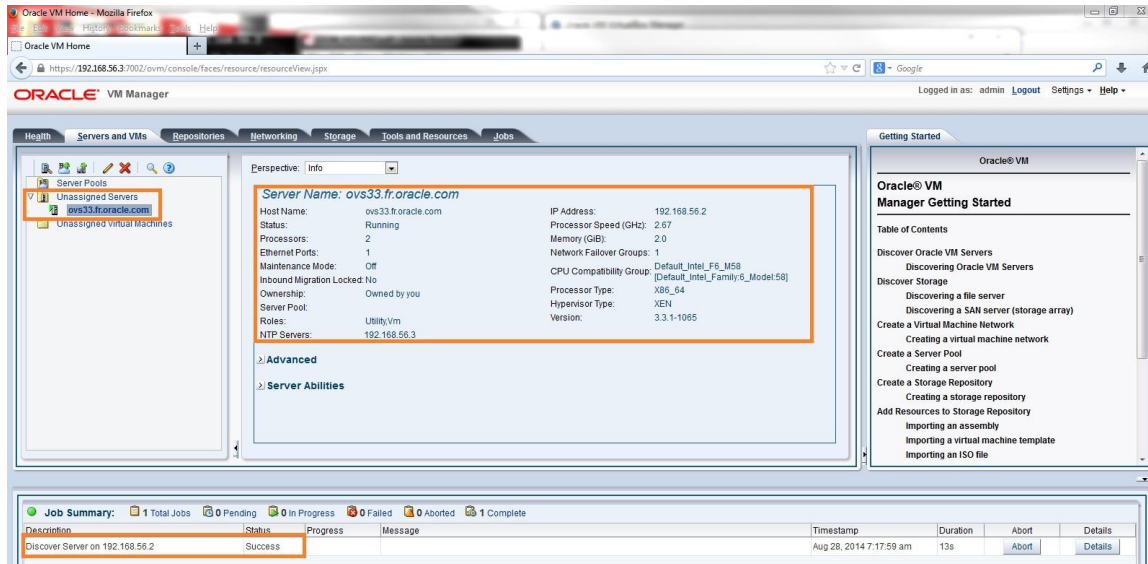
When an Oracle VM Server is discovered, it contains some basic information about itself, and about any immediate connectivity to a shared SAN, but it is considered to be in an unconfigured state. Any storage attached to the Oracle VM Server is also discovered. Depending on your hardware and networking configuration, external storage may be automatically detected during discovery of the Oracle VM Servers. In this LAB our Oracle VM Server does not have any shared storage only a local OCFS storage that has been discovered during discovery of the server.

TO DO :

- Click the Servers and VMs tab, if not already selected
- Click  (Discover Servers in the toolbar)
- In Discover Servers wizard, Enter the Oracle VM Agent password (ovsroot) and the IP address (192.168.56.2) for the Oracle VM Server(s) to be discovered. Click OK



The Oracle VM Servers are discovered and added to the **Unassigned Servers** folder in the **Servers and VMs** tab. The displayed name of a discovered Oracle VM Server is the assigned DNS name, and not the IP address :



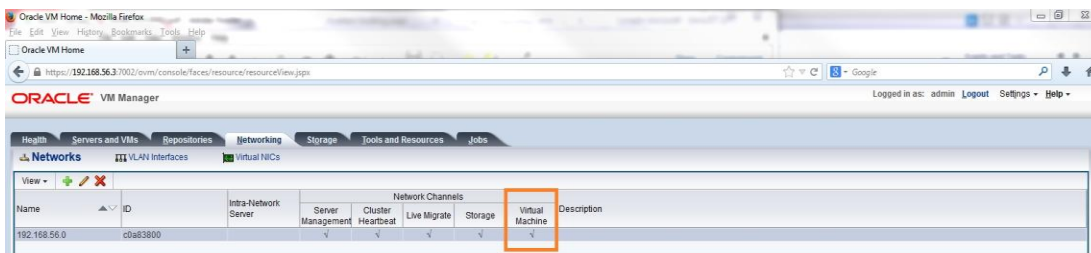
Next step is the discovery of a storage array. In our LAB we only have a local OCFS2 disk attached to the Oracle VM Server which have already been discovered during the discovery of the Oracle VM Server. So next step is to configure a virtual machine network.

DEFAULT MANAGEMENT NETWORK

Oracle VM has a number of network functions: Server Management, Live Migrate, Cluster Heartbeat, Virtual Machine, and Storage. The Server Management, Live Migrate and Cluster Heartbeat roles were automatically assigned to the management network (192.168.56.0) when you discovered the Oracle VM Server. The Virtual Machine and Storage roles are not automatically created, and you must manually create these. The Storage role is only required for iSCSI-based storage, so for the purposes of local based storage used in this HOL, it is not required. In this HOL the Virtual Machine role was already created.

TO DO :

- Click the Networking tab, then the Networks subtab
- Select existing Management Network 192.168.56.0
- Check Virtual Machine role is selected if not, add this role using 

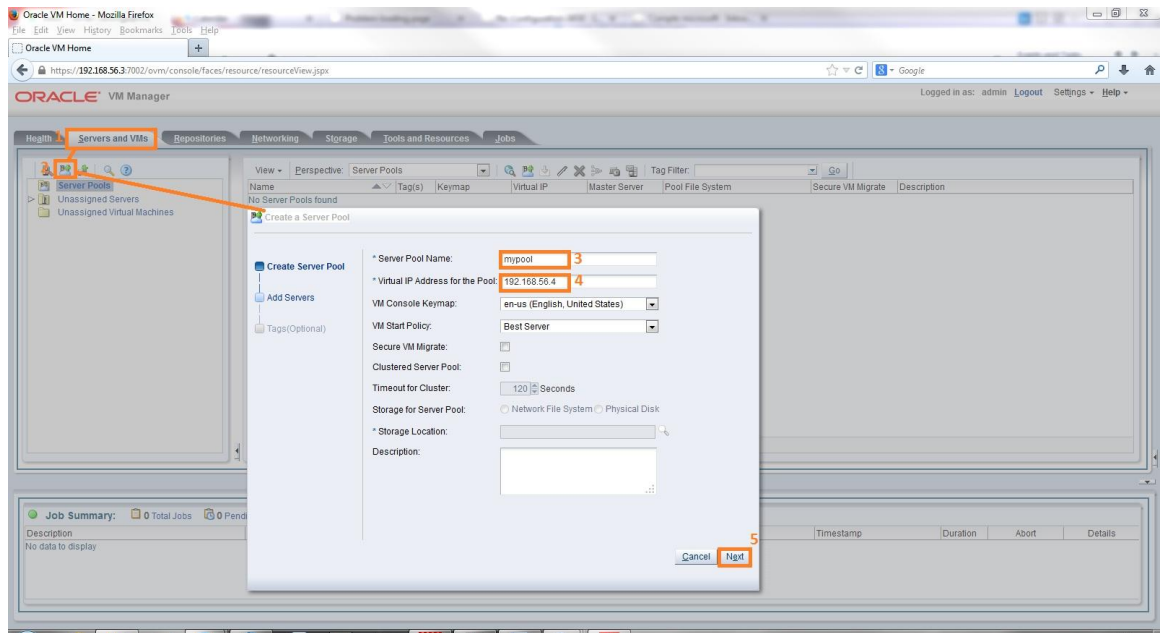


CREATE A SERVER POOL

A server pool contains a group of Oracle VM Servers, which as a group perform virtual machine management tasks, such as High Availability (HA), implementation of resource and power management policies, access to networking, storage and repositories. In this LAB, we will create a server pool with a single Oracle VM Server inside.

TO DO :

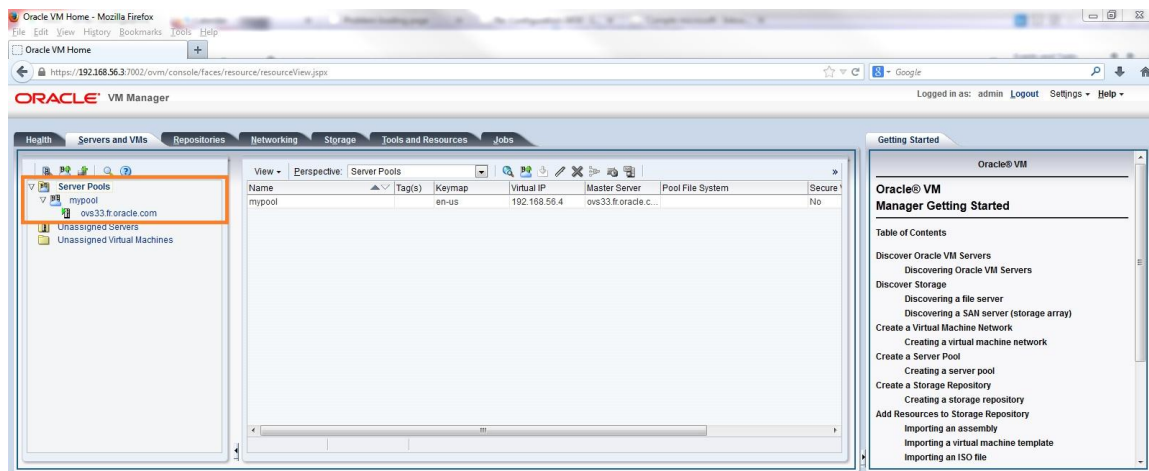
- Click the Servers and VMs tab
- Click Create Server Pool from the toolbar
- The Create a Server Pool wizard is displayed. Enter the server pool information
 - Server Pool Name : mypool
 - Virtual IP : 192.168.56.4
 - Uncheck Clustered Server Pool



- Click Next to add the Oracle VM Server ovs33.fr.oracle.com to the server pool



- Click Finish and check Oracle VM Server ovs33.fr.oracle.com is now part of your server pool mypool



You will now create a storage repository.

CREATE A STORAGE REPOSITORY

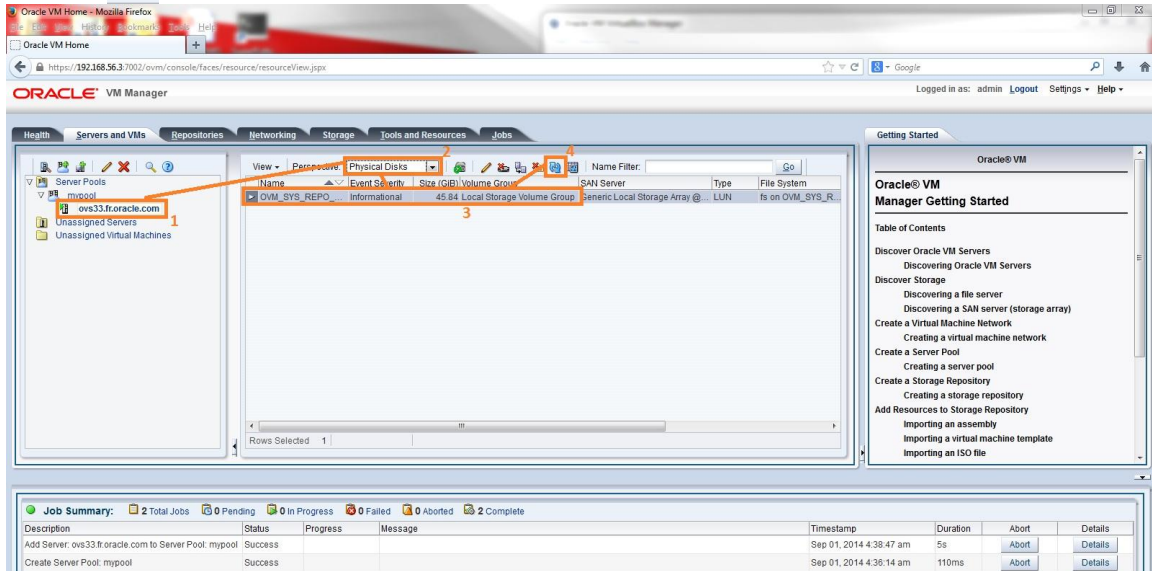
A storage repository is where Oracle VM resources may reside. Resources include virtual machines, templates for virtual machine creation, virtual machine assemblies, ISO files (DVD image files), shared virtual disks, and so on.

We will create a storage repository for Oracle VM by using a pre-existing disk. This disk already contains an existing repository with a Virtual Machine inside. Creating the repository is very simple but installing this Oracle Linux VM would take too much time during this HOL session.

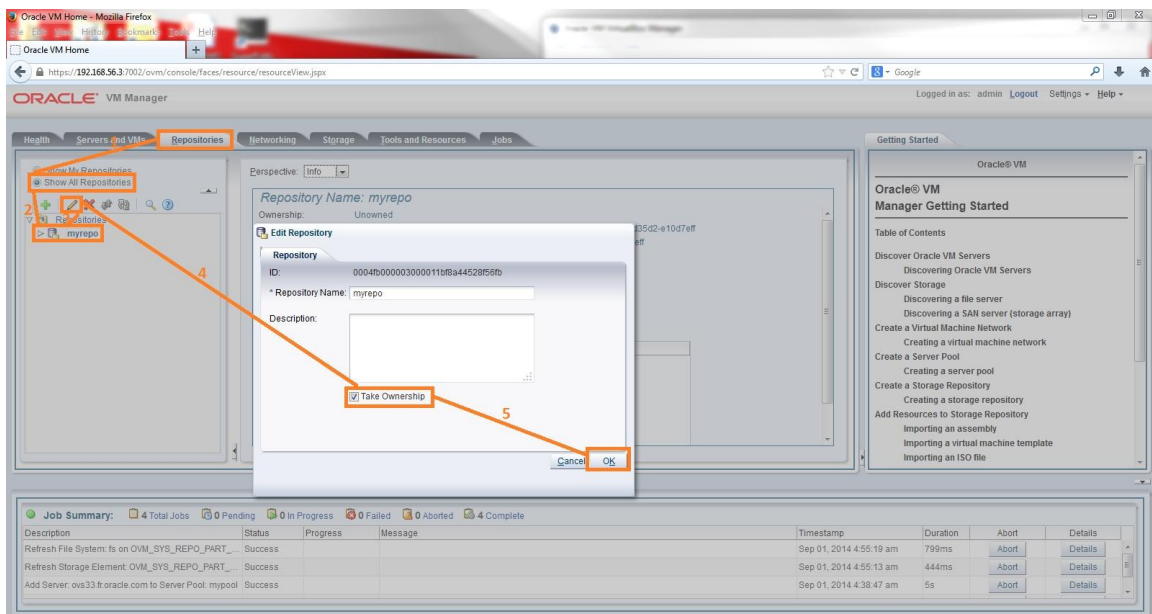
TO DO :

- In servers and VMs Tab, select ovs33.fr.oracle.com and in Perspective choose Physicals disks
- Select the existing hard disk (OVM_SYS_REPO...)

- Click on  to refresh the Physical Disk

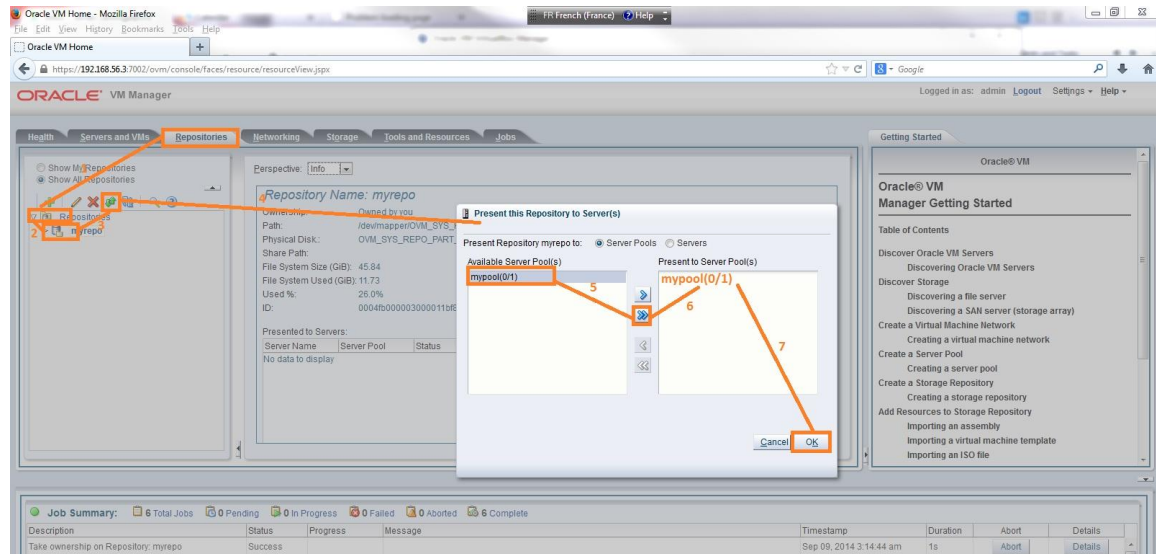


- In the Confirmation windows click OK to confirm
- In Repositories tab, click on Show All repositories, you will see the pre-existing myrepo. For now, this repo is not owned/presented to any server
- Select the myrepo and click to edit the Repository
- In the Edit Repository wizard, Set Take Ownership and press OK



- In the Repository tab, Select Repositories, Select myrepo and click on 

- In “Present this Repository to Server(s)” wizard present your server pool mypool to your server



- Click OK to complete
- The repository is now owned and presented to the Oracle VM Server ovm-srv.oow.com
- You need to refresh the repository by clicking on 

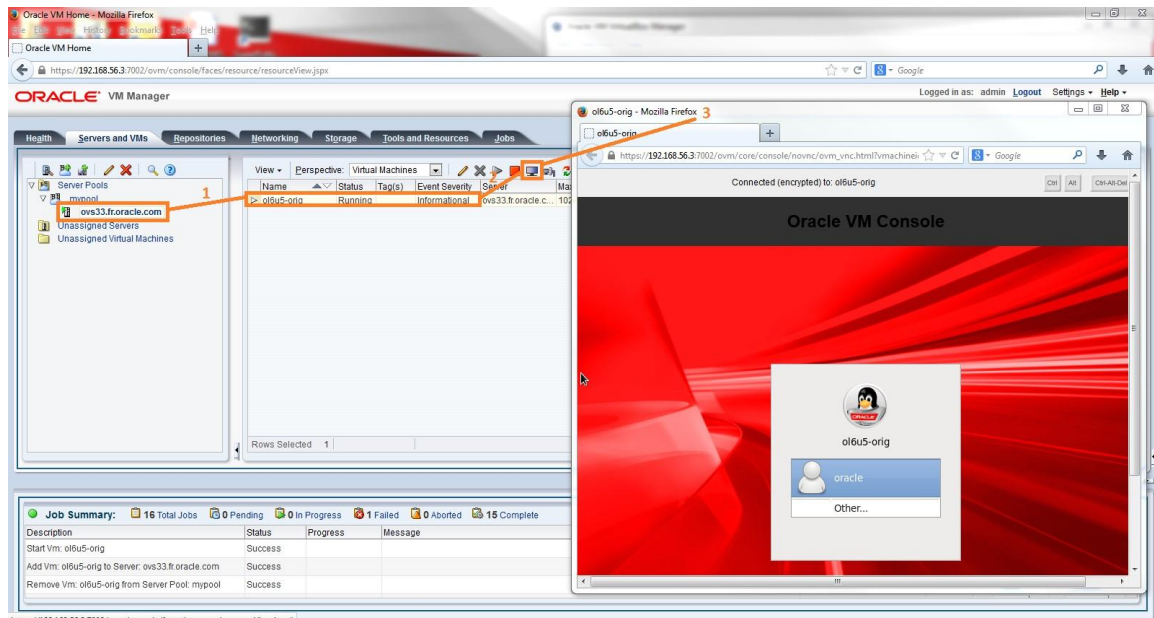
You now have a repository presented to your Oracle VM server ovs33.fr.oracle.com. This repository is having one Unassigned Virtual Machine (Oracle Linux 6.5). This Virtual Machine (ol6u5-orig) is going to be used in following steps.

CLONE AND START YOUR ORACLE LINUX VIRTUAL MACHINE

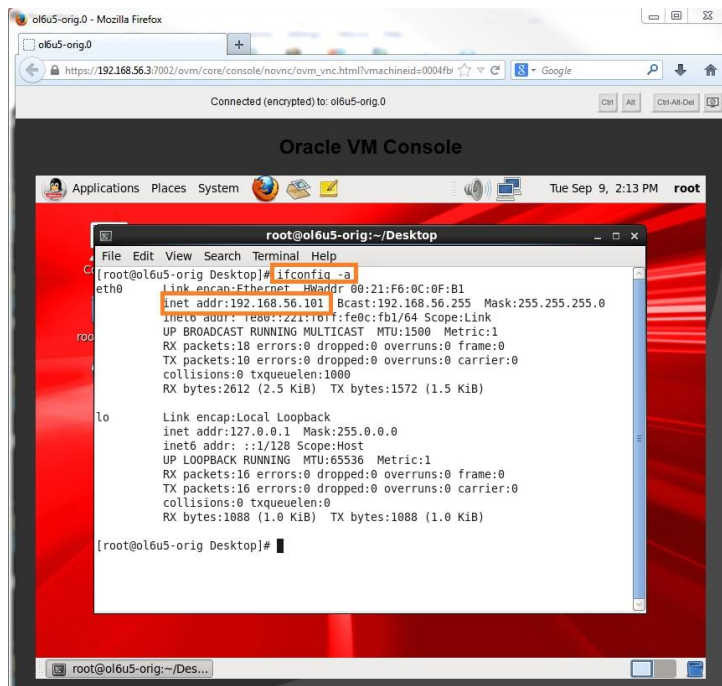
The repository have been provided to you with an Oracle Linux 6u5 virtual machine. Before starting the virtual machine you will be asked to create a clone of this virtual machine. The work for this HOL will be done on this clone avoiding modification of the original VM.

TO DO :

- In Servers and VMs Tab, Click on Unassigned Virtual Machines
- Select Virtual machine ol6u5-orig
- Click on  to clone this virtual machine,
- In “Clone or Move Virtual Machine” wizard, select Create a clone of this VM and click Next
- In “Clone or Move Virtual Machine” wizard, chose a name for your VM
 - Clone Name = ol6u5-orig



- Logging in the VM console window using credential oracle/ovsroot, open a terminal session and check the IP of the VM using `ifconfig -a`, it should be 192.168.56.101 (it might be different according your dhcp configuration, check using the console of the vm)



- Once IP has been checked, minimize the console windows
- From you host system, open a terminal a session and check you are able to ping the VM

```
1. Homelinuxterm x
[09/09/2014 14:19.31] ~
[ocanonge.OCANONGE-FR] > ping 192.168.56.101

Pinging 192.168.56.101 with 32 bytes of data:
Reply from 192.168.56.101: bytes=32 time=1ms TTL=64
Reply from 192.168.56.101: bytes=32 time=1ms TTL=64
Reply from 192.168.56.101: bytes=32 time=1ms TTL=64
Reply from 192.168.56.101: bytes=32 time=2ms TTL=64

Ping statistics for 192.168.56.101:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms

[09/09/2014 14:19.42] ~
[ocanonge.OCANONGE-FR] > 
```

CONFIGURE ORACLE LINUX 6.5 AS AN ORACLE VM TEMPLATE

Next steps will show you how to make use of Oracle VM API to configure and an Oracle Linux VM running on top of Oracle VM 3 as an Oracle VM template. Because of limited time during this session some steps have already been done and are indicated as “ALREADY DONE”.

Requirement are :

- To have a local yum repository (ALREADY DONE). This local yum repository is configured on the Oracle VM Manager virtual machine (192.168.56.3). The repository is including 3 channels : ol6_latest, ol6_addons and ol6_MySQL56. If you are interested on how it was build have a look to the ANNEXE part of this document
- Virtual Machine ol6u5-orig.0 is configured to receive updates from this local yum repository. Yum repository configuration file is in /etc/yum.repos.d. Check content of the file public-yum-ol6.repo, it looks like :

```
[root@ol6u5-orig yum.repos.d]# more public-yum-ol6.repo
[ol6_latest]
name=Oracle Linux $releasever Latest ($basearch)
baseurl=http://192.168.56.3/yum/ol6/ol6_latest/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

[ol6_addons]
name=Oracle Linux $releasever Add ons ($basearch)
baseurl=http://192.168.56.3/yum/ol6/ol6_addons/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

[ol6_MySQL56]
name=MySQL 5.6 for Oracle Linux 6 ($basearch)
baseurl=http://192.168.56.3/yum/ol6/ol6_MySQL56/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

TO DO :

- Connect to ol6u5-orig.0 using ssh (root/ovsroot) :
ssh root@192.168.56.101 (or IP address corresponding to your VM)
- Check you are able to receive some updates from your local yum repository :

```
[root@ol6u5-orig yum.repos.d]# yum repolist
Loaded plugins: refresh-packagekit, security
repo id                                repo name
status
ol6_MySQL56                            MySQL 5.6 for Oracle Linux 6 (x86_64)
91
ol6_addons                             Oracle Linux 6Server Add ons (x86_64)
153
ol6_latest                             Oracle Linux 6Server Latest (x86_64)
5,409
repolist: 5,653
[root@ol6u5-orig yum.repos.d]#
```

- Install following Oracle VM API packages (ovmd, xenstoreprovider python-simplejson ovm-template-config) :

This installs the basic necessary packages on Oracle Linux 6 to support the Oracle VM API. Xenstoreprovider is the library which communicates with the ovmapl kernel infrastructure. ovmd is a daemon that handles configuration and re-configuration events and provides a mechanism to send/receive messages between the VM and the Oracle VM Manager.

```
[root@ol6u5-orig yum.repos.d]# yum install ovmd xenstoreprovider python-simplejson \
ovm-template-config
```

```
Loaded plugins: refresh-packagekit, security
```

```
Setting up Install Process
```

```
Resolving Dependencies
```

```
--> Running transaction check
```

```
---> Package ovm-template-config.noarch 0:3.0-75.el6 will be installed
```

```
---> Package ovmd.x86_64 0:3.0-40.el6 will be installed
```

```
--> Processing Dependency: libovmapi >= 3.0 for package: ovmd-3.0-40.el6.x86_64
```

```
--> Processing Dependency: libovmapi.so() (64bit) for package: ovmd-3.0-40.el6.x86_64
```

```
---> Package python-simplejson.x86_64 0:2.0.9-3.1.el6 will be installed
```

```
---> Package xenstoreprovider.x86_64 0:3.0-11.el6 will be installed
```

```
--> Running transaction check
```

```
---> Package libovmapi.x86_64 0:3.0-6.el6 will be installed
```

```
--> Finished Dependency Resolution
```

```
Dependencies Resolved
```

Package	Arch	Version	Repository	Size
Installing:				
ovm-template-config	noarch	3.0-75.el6	ol6_addons	62 k
ovmd	x86_64	3.0-40.el6	ol6_addons	31 k
python-simplejson	x86_64	2.0.9-3.1.el6	ol6_latest	126 k
xenstoreprovider	x86_64	3.0-11.el6	ol6_addons	24 k
Installing for dependencies:				
libovmapi	x86_64	3.0-6.el6	ol6_addons	19 k

```
Transaction Summary
```

```
Install      5 Package(s)
```

```
Total download size: 262 k
```

```
Installed size: 748 k
```

```
Is this ok [y/N]:y
```

- Respond yes to any warnings, following packages should have been installed :

```
Running Transaction
Installing : libovmapi-3.0-6.el6.x86_64
1/5
Installing : ovmd-template-config-3.0-75.el6.noarch
2/5
Installing : ovmd-3.0-40.el6.x86_64
3/5
Installing : xenstoreprovider-3.0-11.el6.x86_64
4/5
Installing : python-simplejson-2.0.9-3.1.el6.x86_64
5/5
Verifying : xenstoreprovider-3.0-11.el6.x86_64
1/5
Verifying : python-simplejson-2.0.9-3.1.el6.x86_64
2/5
Verifying : ovmd-3.0-40.el6.x86_64
3/5
Verifying : ovmd-template-config-3.0-75.el6.noarch
4/5
Verifying : libovmapi-3.0-6.el6.x86_64
5/5

Installed:
  ovmd-template-config.noarch 0:3.0-75.el6      ovmd.x86_64 0:3.0-40.el6      python-
simplejson.x86_64 0:2.0.9-3.1.el6
  xenstoreprovider.x86_64 0:3.0-11.el6

Dependency Installed:
  libovmapi.x86_64 0:3.0-6.el6

Complete!
```

- In order to create an Oracle VM template that includes basic OS configuration system scripts, we are going to install all of the following packages :

ovm-template-config-authentication (authentication configuration script), ovm-template-config-datetime (datetime configuration script), ovm-template-config-firewall (firewall configuration script), ovm-template-config-network (network configuration script), ovm-template-config-selinux (selinux configuration script), ovm-template-config-ssh (ssh configuration script), ovm-template-config-system (system configuration script) and ovm-template-config-user (user configuration script).

```
[root@ol6u5-orig yum.repos.d]# yum install ovm-template*
Loaded plugins: refresh-packagekit, security
Setting up Install Process
Package ovm-template-config-3.0-75.el6.noarch already installed and latest version
Resolving Dependencies
--> Running transaction check
---> Package ovm-template-config-authentication.noarch 0:3.0-75.el6 will be installed
---> Package ovm-template-config-datetime.noarch 0:3.0-75.el6 will be installed
---> Package ovm-template-config-firewall.noarch 0:3.0-75.el6 will be installed
---> Package ovm-template-config-network.noarch 0:3.0-75.el6 will be installed
---> Package ovm-template-config-selinux.noarch 0:3.0-75.el6 will be installed
---> Package ovm-template-config-ssh.noarch 0:3.0-75.el6 will be installed
---> Package ovm-template-config-system.noarch 0:3.0-75.el6 will be installed
---> Package ovm-template-config-user.noarch 0:3.0-75.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                               Arch      Version
Repository                            Size
=====
Installing:
 ovm-template-config-authentication    noarch    3.0-75.el6
ol6_addons                            17 k
 ovm-template-config-datetime          noarch    3.0-75.el6
ol6_addons                            18 k
 ovm-template-config-firewall          noarch    3.0-75.el6
ol6_addons                            17 k
 ovm-template-config-network          noarch    3.0-75.el6
ol6_addons                            18 k
 ovm-template-config-selinux          noarch    3.0-75.el6
ol6_addons                            17 k
 ovm-template-config-ssh              noarch    3.0-75.el6
ol6_addons                            18 k
 ovm-template-config-system          noarch    3.0-75.el6
ol6_addons                            17 k
 ovm-template-config-user             noarch    3.0-75.el6
ol6_addons                            18 k

Transaction Summary
=====
Install      8 Package(s)

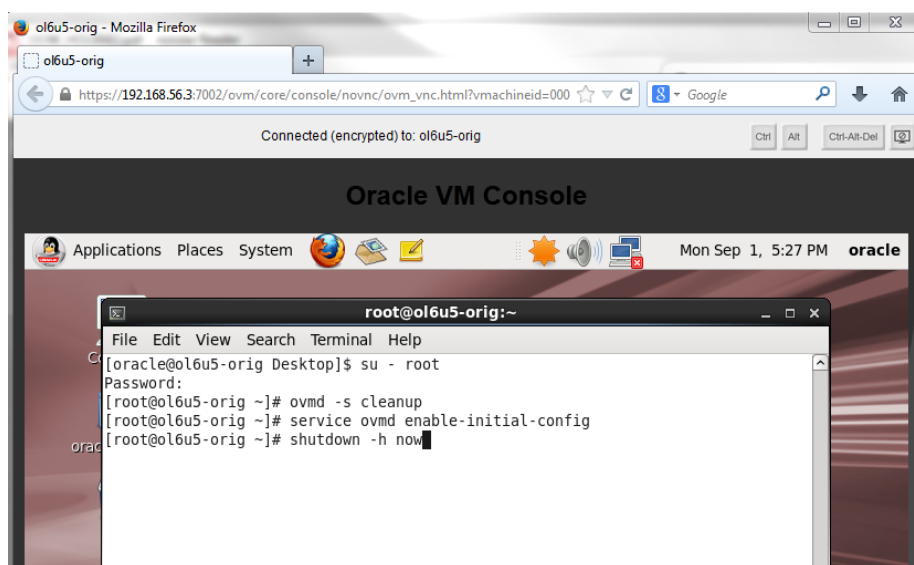
Total download size: 140 k
Installed size: 38 k
Is this ok [y/N]:y
```

- To enable ovmd do :

```
[root@ol6u5-orig yum.repos.d]# chkconfig ovmd on
[root@ol6u5-orig yum.repos.d]# /etc/init.d/ovmd start
Starting OVM guest daemon: [ OK ]
[root@ol6u5-orig yum.repos.d]#
```

- Next commands are going to blank your VM network configuration so that it should be run from the Oracle VM console of the VM (the one you minimized in Step)

```
[root@ol6u5-orig yum.repos.d]# su - root
Password : ovsroot
[root@ol6u5-orig yum.repos.d]# ovmd -s cleanup
[root@ol6u5-orig yum.repos.d]# service ovmd enable-initial-config
[root@ol6u5-orig yum.repos.d]# shutdown -h now
```

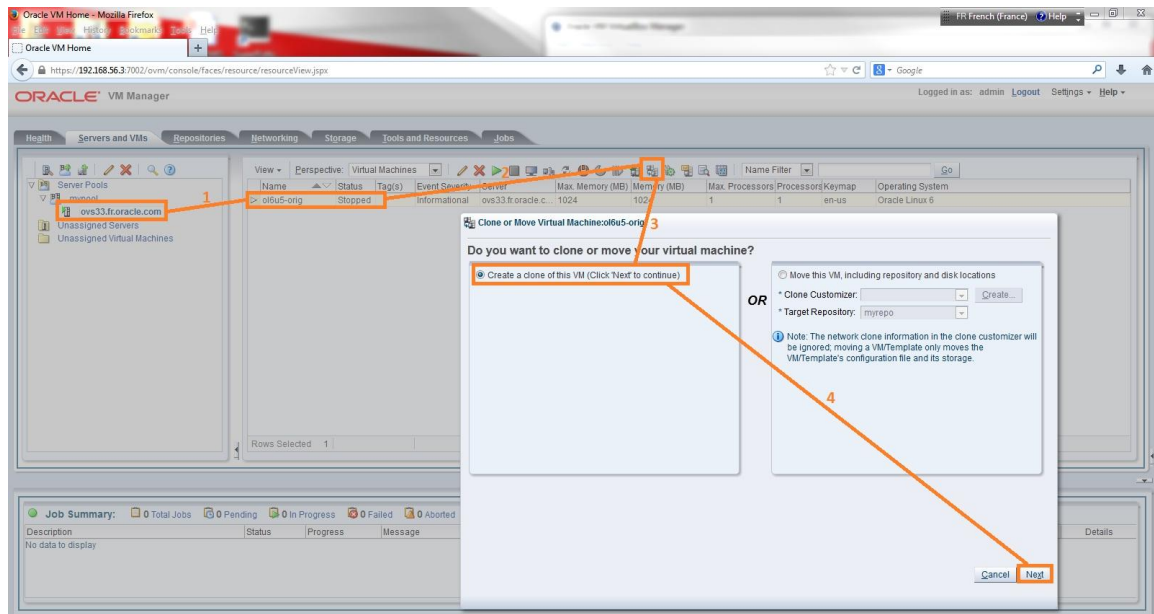


CLONE THIS VM AS AN ORACLE VM TEMPLATE

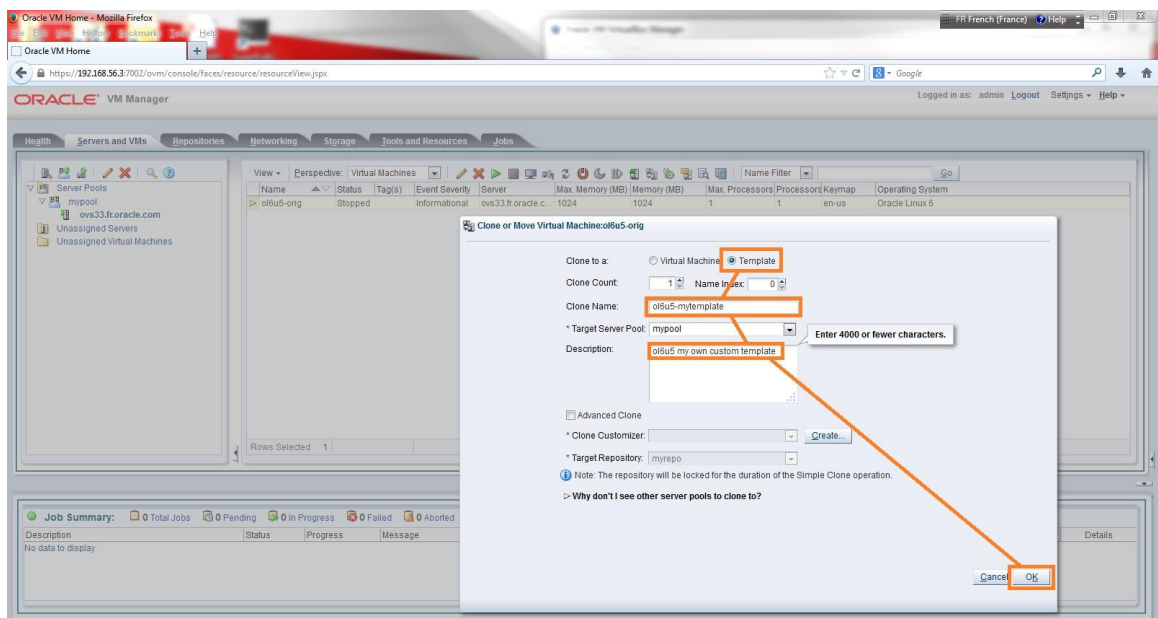
This Virtual Machine is now stopped and is ready to be cloned as an Oracle VM template.

TO DO :

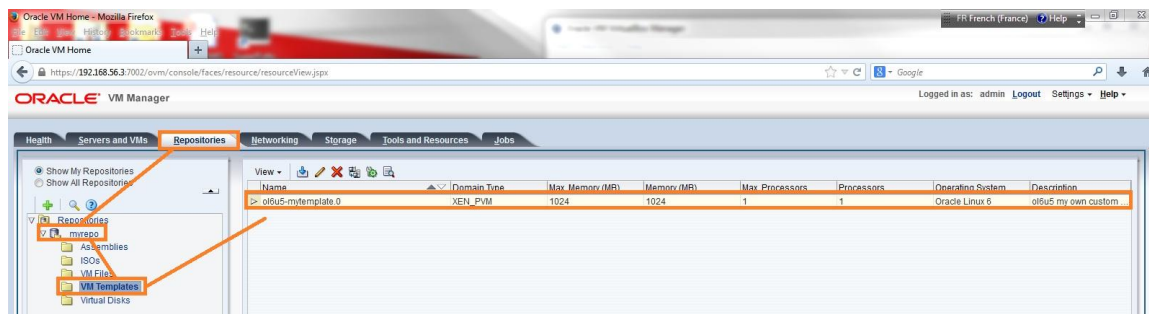
- Select Virtual Machine ol6u5-orig and click on  to clone this VM
- On "Clone or Move Virtual Machine" wizard select "Create a clone of this VM" and click Next



- On the next “Clone or Move your virtual machine” wizard select :
 - Clone to a : Template
 - Clone Name : ol6u5-mytemplate
 - Description : ol6u5 my own custom template
 - Click on OK



- In Repositories Tab, under myrepo and VM Templates check ol6u5-mytemplate.0 has been created

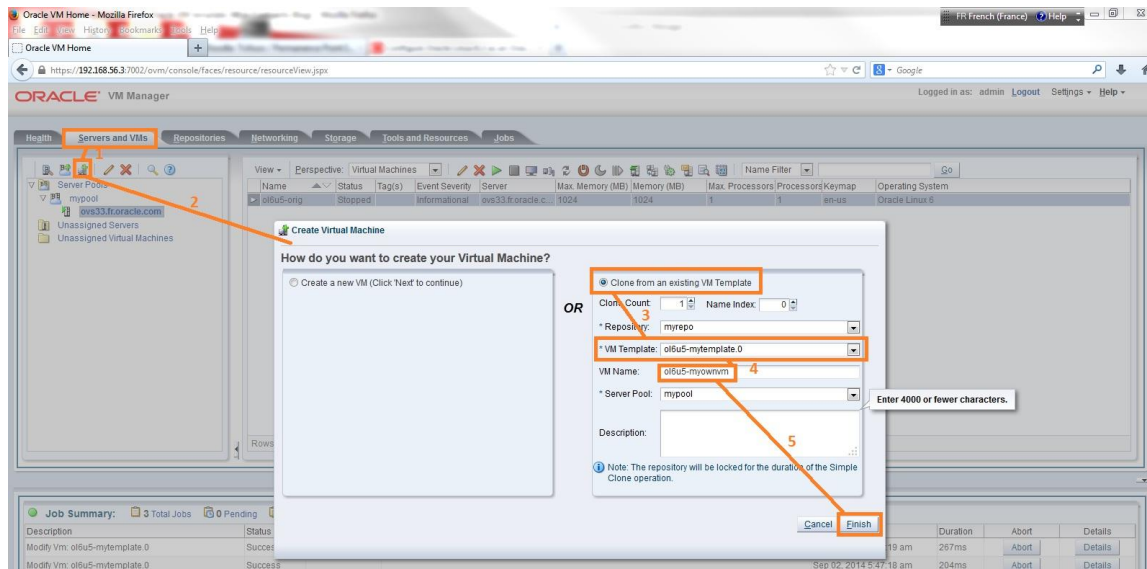


CLONE YOUR OWN ORACLE VM TEMPLATE AS A NEW VM

Now you have created your first template, we will clone a new VM from this template and check that this new VM will act as a first time boot VM and it will require configuration input through the the Virtual Machine console.

TO DO :

- In Servers and Vms Tab, click on  to create a new VM
- In “Create Virtual Machine” wizard,
 - Select Clone from an existing VM Template
 - In VM Template choose ol6u5-mytemplate.0
 - In VM Name enter the name of your vm : ol6u5-myownvm
 - Click on Finish

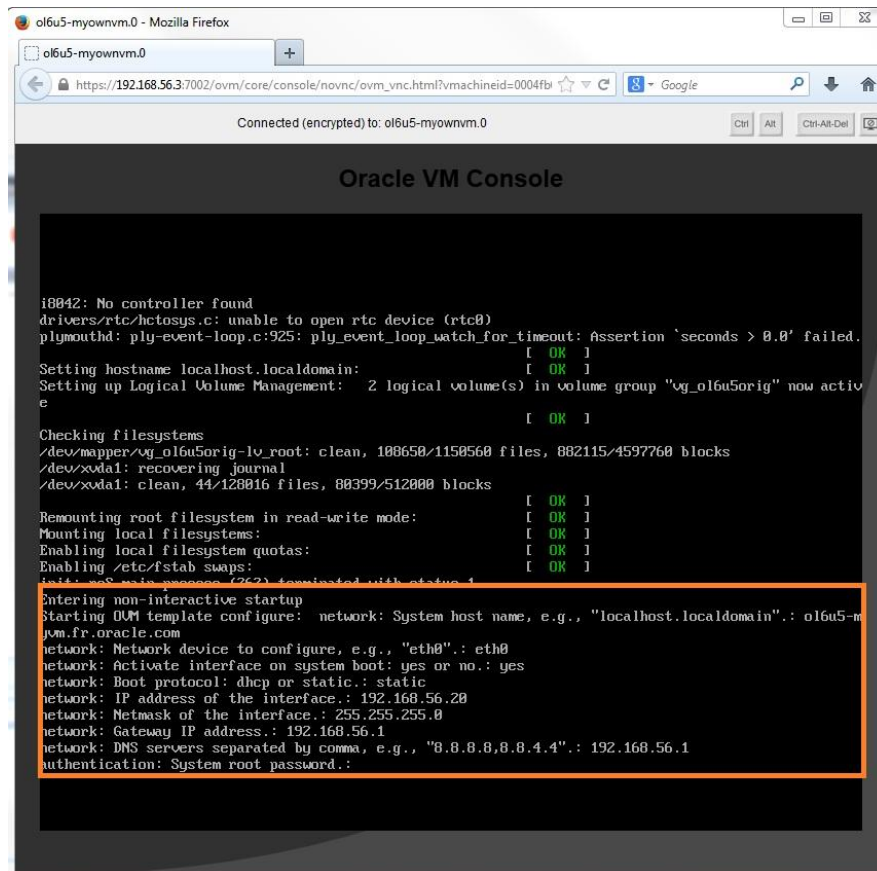


START YOUR NEW BLANK VM FOR FIRST CONFIGURATION

Your new VM is now ready to be started. This is a first boot VM with no network configuration. You will need to configure this VM using Oracle VM console.

TO DO :

- Select ol6u5-myownvm.0 and open a console by clicking on 
- Start the VM by clicking on 
- In the OVM Console windows, complete requested information (**you need to fulfill all fields, don't leave any empty**) :
 - System host name : ***ol6u5-myvm.fr.oracle.com***
 - Network device to configure : ***eth0***
 - Activate interface on system boot : ***yes***
 - Boot protocol : ***static***
 - IP address : ***192.168.56.20***
 - Netmask : ***255.255.255.0***
 - Gateway : ***192.168.56.1***
 - DNS : ***192.168.56.1***
 - System root password : ***jungle1234***



- From a terminal windows in your host system, check you are able to access the VM using ssh [root@192.168.56.20](https://192.168.56.20) using credential you defined before.

```
[ocanonge.OCANONGE-FR]> ssh root@192.168.56.20
Permanently added '192.168.56.20' (RSA) to the list of known hosts.
root@192.168.56.20's password: jungle1234
[root@ol6u5-myvm ~]#
```

You have now created your first Oracle Linux VM Template. Next steps will show you how to customize this VM and deploy an application. As an example we are going to deploy a mysql database and create a template from this VM.

INSTALL MYSQL RPMS

The installation of the software required for the template can either be done when we build the initial virtual machine or we deploy the VM. Some software (such as Enterprise Manager or Weblogic) depend on the hostname and IP address for example, in this case it makes more sense to do a silent installation at deployment time, a template initialization script would take care of this silent installation. MySQL binaries do not depend on the hostname nor the network configuration, therefore we can install them at the creation of the initial VM. However the MySQL database will

be created when the template is deployed. We install the packages for MySQL 5.6 available on our local yum repository

- ssh on your ol6u5-myvm.fr.oracle.com
- Install MySQL5.6

```
[ocanonge.OCANONGE-FR]> ssh root@192.168.56.20  
[root@ol6u5-myvm ~]# yum install mysql-community-server
```

BUILDING THE TEMPLATE INITIALIZATION SCRIPTS

The packages ovm-template-config-<something> install the template initialization scripts in /etc/template.d/script. Review the installed scripts, authentication, system, network, etc. they are Python scripts with the following structure:

- Plugin info
The name of the script and when the execution order is set there
- Python import
The declaration of the Python dependencies
- The function do_enumerate(target)
It defines and reports the input keys for the script. For each keys, the administration will send a value to deploy the template.
- The function do_configure(param)
This is the code to be executed when the template is deployed.
- The function do_cleanup(param)
This is the code to be executed when the VM is being initialized before converting it to a Template. It should be cleaned up of any specific configuration (network setting, passwords, application setting and data...).
- The main function
It calls either one of the 3 functions above depending of the argument passed to the script. This should not be modified.

Here is the structure of a generic script :

```
### BEGIN PLUGIN INFO
# name: [script name]
# [target]: [priority]
# [target]: [priority]
# description: a description that can cross multiple lines.
### END PLUGIN INFO

try:
    import json
except ImportError:
    import simplejson as json
from templateconfig.cli import main
from templateconfig.common import shell_cmd

def do_enumerate(target):
    param = []
    if target == 'configure':
        param += [{'key': 'com.oracle.linux.XXXXXX',
                    'description': 'Description for the key'},
                  {'key': 'com.oracle.linux.YYYYYY',
                    'description': 'Description for the key',
                    'choices': ['yes', 'no']}]
    return json.dumps(param)

def do_configure(param):
    param = json.loads(param)
    # your code
    return json.dumps(param)

def do_cleanup(param):
    param = json.loads(param)
    # your code
    return json.dumps(param)

if __name__ == '__main__':
    main(do_enumerate, {'configure': do_configure, 'cleanup': do_cleanup})
```

We are going to make our own script to configure and unconfigure MySQL automatically. This script must have the same structure as show above. We add several procedures and functions:

- `do_mysql_install_db()`
This procedure create the MySQL database and starts it up. It is called from the `do_configure` function.
- `do_mysql_delete_db()`
This procedure stops MySQL and delete the database. It is called from the `do_cleanup` function.
- `do_mysql_secure_installation(removeAnonymousUser, desallowRemoteRootLogin, removeTestDatabase)`
This procedure calls the “mysql_secure_installation” script. It provides script input via expect commands. This procedure is called from the `do_configure` function.
- `set_mysql_root_password(password)`

This procedure sets a password for the mysql database root user. It is called from the do_configure function.

- str2bool(v)

This is a tiny function that converts a string, yes or no, to a Boolean. It is called from the do_configure function. The purpose of this function is to make the code more readable.

For simplicity and time saving, the script was already created and is located on the Oracle VM Manager VM.

TO DO :

- Copy mysql script to your vm in “/etc/template.d/scripts”

```
[root@ol6u5-myvm]# cd /etc/template.d/scripts/  
[root@ol6u5-myvm]# scp root@192.168.56.3:/root/mysql .
```

- Set correct access rights to the file and have a look to the file

```
[root@ol6u5-myvm]# chmod 755 /etc/template.d/scripts/mysql
```

- Add this script in the Template initialization list with the command

```
[root@ol6u5-myvm]# ovm-chkconfig --add mysql
```

- Review the “configure” and “cleanup” order with the command

```
[root@ol6u5-myvm]# ovm-chkconfig -list  
name configure unconfigure reconfigure cleanup suspend resume migrate shutdown  
authentication on:90 off off off off off off off  
datetime on:50 off off on:50 off off off off  
firewall on:41 off off off off off off off  
mysql on:70 off off on:30 off off off off  
network on:50 off off on:50 off off off off  
selinux on:30 off off off off off off off  
ssh on:70 off off on:30 off off off off  
system on:60 off off on:60 off off off off  
user on:60 off off on:40 off off off off
```

TESTING TEMPLATES INITIALIZATION SCRIPTS

You have 3 options to choose from to test your Templates initialization scripts, there are documented in the Oracle whitepaper Oracle VM Templates Automated Virtual Machine Provisioning in the section “Appendix: Testing of Template Configuration Scripts”.

TO DO :

- First, in order to validate our function “do_enumerate”, we enumerate the keys that the script is expecting with the following command :

```
[root@ol6u5-mylv scripts]# ovm-template-config --human-readable --enumerate -  
-script mysql configure  
  
[('70',  
  'mysql',  
  [{u'choices': [u'yes', u'no'],  
    u'description': u'Remove anonymous user from MySQL? : yes or no.',  
    u'key': u'com.oracle.linux.mysql.remove-anonymous-user'},  
   {u'choices': [u'yes', u'no'],  
    u'description': u'Disallow root login remotely? : yes or no.',  
    u'key': u'com.oracle.linux.mysql.desallow-remote-root-login'},  
   {u'choices': [u'yes', u'no'],  
    u'description': u'Remove test database and access to it? : yes or no.',  
    u'key': u'com.oracle.linux.mysql.remove-test-database'},  
   {u'description': u'root password for MySQL',  
    u'key': u'com.oracle.linux.mysql.mysql-root-password',  
    u'password': True,  
    u'required': True}]]]
```

Then we create a file in json format that contain the key/value pair to be used as input for our script as follow. We name this file mysql.json.

- Go in /var/tmp of ol6u5-mylv and copy or create a simple mysql.json, content of the file :

```
[root@ol6u5-mylv /var/tmp]# scp root@192.168.56.3:/root/mysql.json .  
[root@ol6u5-mylv /var/tmp]# cat /var/tmp/mysql.json  
{  
  "com.oracle.linux.mysql.remove-anonymous-user": "yes",  
  "com.oracle.linux.mysql.desallow-remote-root-login": "yes",  
  "com.oracle.linux.mysql.remove-test-database": "no",  
  "com.oracle.linux.mysql.mysql-root-password": "ovsroot"  
}
```

- We can then execute the cleanup and configure part of our script with the following commands:

```
[root@ol6u5-mylv ]# /etc/template.d/scripts/mysql cleanup \  
< /var/tmp/mysql.json  
{  
  "com.oracle.linux.mysql.remove-anonymous-user": "yes",  
  "com.oracle.linux.mysql.desallow-remote-root-login": "yes",  
  "com.oracle.linux.mysql.remove-test-database": "no",  
  "com.oracle.linux.mysql.mysql-root-password": "ovsroot"  
}  
  
[root@ol6u5-mylv ]# /etc/template.d/scripts/mysql configure \  
< /var/tmp/mysql.json  
{  
  "com.oracle.linux.mysql.remove-anonymous-user": "yes",  
  "com.oracle.linux.mysql.desallow-remote-root-login": "yes",  
  "com.oracle.linux.mysql.remove-test-database": "no",  
  "com.oracle.linux.mysql.mysql-root-password": "ovsroot"  
}
```

CONVERTING THE VM TO A TEMPLATE

TO DO :

- To enable ovmd do :

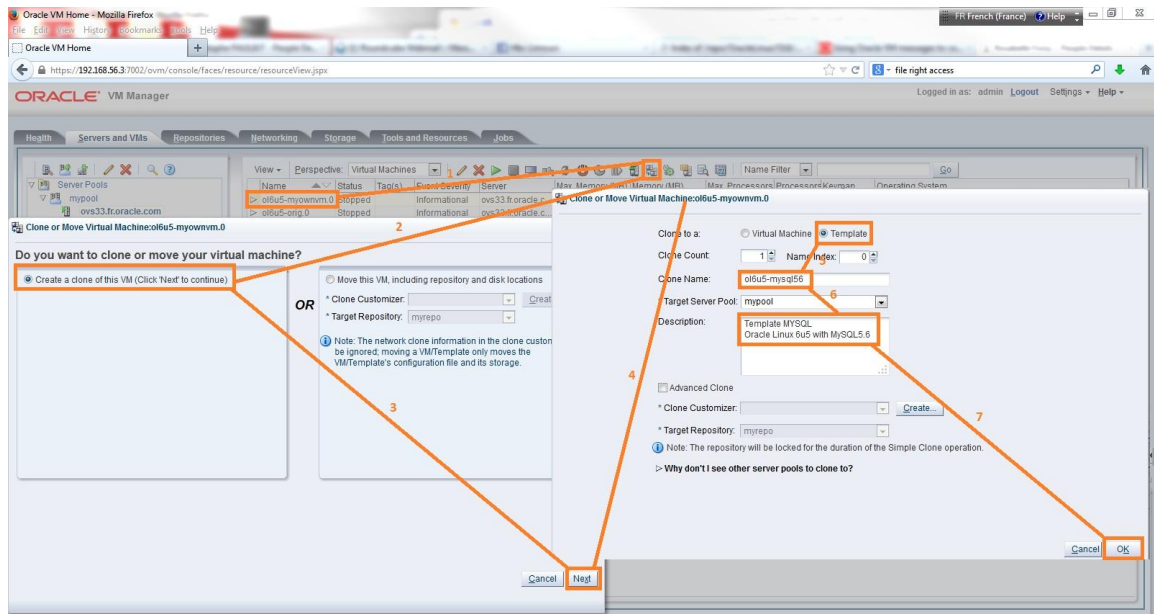
```
[root@ol6u5-orig yum.repos.d]# chkconfig ovmd on
[root@ol6u5-orig yum.repos.d]# /etc/init.d/ovmd start
Starting OVM guest daemon: [ OK ]
[root@ol6u5-orig yum.repos.d]#
```

- **Next commands are going to blank your VM network configuration so that it should be run from the Oracle VM console of the VM (the one you minimized in Step)**

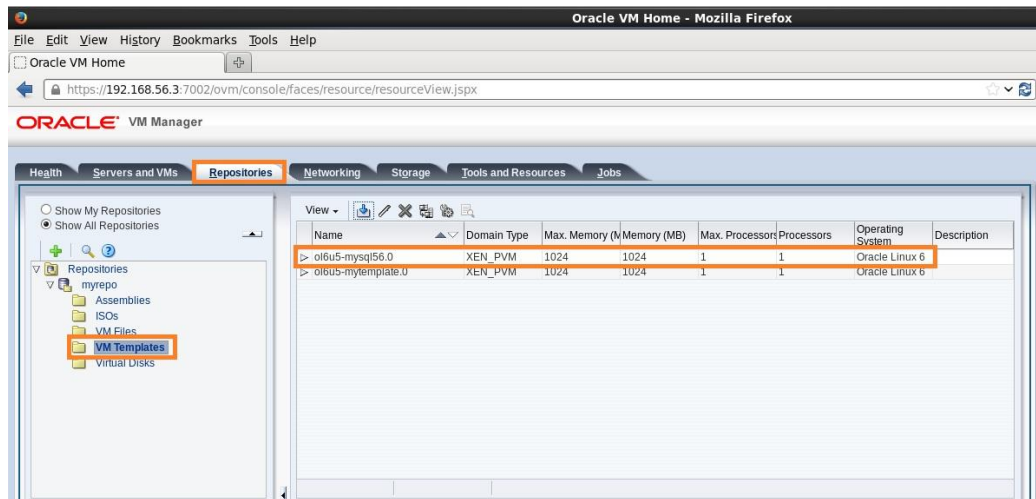
```
[root@ol6u5-orig yum.repos.d]# su - root
Password : jungle1234
[root@ol6u5-orig yum.repos.d]# ovmd -s cleanup
[root@ol6u5-orig yum.repos.d]# service ovmd enable-initial-config
[root@ol6u5-orig yum.repos.d]# shutdown -h now
```

The Virtual Machine ol6u5-myownvm.0 is now stopped and is ready to be cloned as an Oracle VM template with mysql database inside.

- Select Virtual Machine ol6u5-myownvm and click on  to clone this VM
- On “Clone or Move Virtual Machine” wizard select “Create a clone of this VM” and click Next
- On “Clone or Move Virtual Machine” wizard set :
 - Template
 - Clone Name : ol6u5-mysql56
 - Description field is optional



- Click OK
- In Repositories Tab, under myrepo and VM Templates check o16u5-mytemplate.0 has been created



CLONE YOUR OWN ORACLE VM MYSQL TEMPLATE AS A NEW VM

To deploy the Oracle VM Template, we have 2 options. Either we use the Oracle VM Manager Web interface (as we did before) or we use a command line script. Alternatively, we can deploy the VM and send the key/value messages using a shell script executed from the Oracle VM Manager. We chose to write this script in expect to call Oracle VM CLI commands.

TO DO :

- Open a terminal and connect on the Oracle VM Manager using ssh :

```
[root@hostlaptop]# ssh root@192.168.56.3 (password : ovsroot)
[root@ovm33]# cd /root
[root@ovm33]# rpm -ivh tc1-8.5.7-6.el6.x86_64.rpm
[root@ovm33]# rpm -ivh expect-5.44.1.15-5.el6_4.x86_64.rpm
[root@ovm33]# chmod 755 create_MySQL56_PVM.cli
```

Set the variables defined at the beginning of the script with the values you want to set for MySQL VM deployment. A script create_MySQL56_PVM.cli has been created for you on the OVM Manager in /root directory :

- Check content of create_MySQL56_PVM.cli script :

```
##### Variables to set #####
### OVM Manager
set ovmUser admin
set adminServer localhost
set ovmPassword Welcome1
### VM
set vmName ol6u5-mysql
set vmTemplateName ol6u5-mysql56.0
set serverPoolName mypool
### OS
set hostname mysql.fr.oracle.com
set host {"192.168.56.30 mysql.fr.oracle.com mysql"}
set device eth0
set onboot yes
set bootproto static
set ipaddr 192.168.56.30
set netmask 255.255.255.0
set gateway 192.168.56.1
set dnsserver 192.168.56.1
set rootpassword jungle1234
### MySQL
set removeanonymoususer yes
set desallowremoterootlogin yes
set removetestdatabase yes
set mysqlrootpassword ovsroot
```

The Timeout value has to be adapted to the time it takes to clone a VM. If the template is on a OCFS2 repository, the Timeout value can be set to 10, if it is NFS, it should be around 180.

- Run the script `create_MySQL_PVM.cli` and check output in the Job Summary windows of the Oracle VM manager GUI

```
[root@ovm33]# cd /root
[root@ovm33]# ./create_MySQL56_PVM.cli
```

The screenshot displays the Oracle VM Manager GUI with the Job Summary window open. The terminal window on the left shows the execution of the script `create_MySQL56_PVM.cli`, which clones a VM named `ol6u5-mysql56.0`. The Job Summary window on the right shows a list of jobs, with the 'Clone Vm: ol6u5-mysql56.0 in Server Pool: mypool' job highlighted in orange.

Description	Status	Progress	Message	Timestamp	Duration	Abort
Sent VM message to Vm: ol6u5-mysql	Success			Sep 10, 2014 4:18:57 am	236ms	Abort
Sent VM message to Vm: ol6u5-mysql	Success			Sep 10, 2014 4:18:56 am	211ms	Abort
Sent VM message to Vm: ol6u5-mysql	Success			Sep 10, 2014 4:18:55 am	276ms	Abort
Sent VM message to Vm: ol6u5-mysql	Success			Sep 10, 2014 4:18:54 am	241ms	Abort
Sent VM message to Vm: ol6u5-mysql	Success			Sep 10, 2014 4:18:54 am	334ms	Abort
Sent VM message to Vm: ol6u5-mysql	Success			Sep 10, 2014 4:18:53 am	317ms	Abort
Sent VM message to Vm: ol6u5-mysql	Success			Sep 10, 2014 4:18:52 am	246ms	Abort
Start Vm: ol6u5-mysql	Success			Sep 10, 2014 4:18:42 am	32ms	Abort
Modify Vm: ol6u5-mysql	Success			Sep 10, 2014 4:18:41 am	125ms	Abort
Clone Vm: ol6u5-mysql56.0 in Server Pool: mypool	Success			Sep 10, 2014 4:18:34 am	3s	Abort

Once the VM is deployed, you can ssh to the network address you provided. The hostname is set, a MySQL database is running. Validate by running the following command:

```
[root@hostlaptop]# ssh root@192.168.56.30      (Password is jungle1234)
[root@mysql ~]# mysql -h localhost -u root -p
Enter password: ovsroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.6.20 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0,00 sec)

mysql>
```

CONGRATULATIONS

You are now at the end of this Hands-On LAB session.

You have learned how to create your own Oracle VM Template and how to customize this template with your own application inside.

Oracle VM Template is a key feature of the Oracle VM virtualization solution to enable rapid deployment of enterprise applications. Oracle already provides over 100 ready-to-use Oracle VM Templates for Oracle applications.



REFERENCES:

- Oracle VM Documentation
<http://www.oracle.com/technetwork/server-storage/vm/documentation/index.html>
- Wim Coekaerts Blog
<https://blogs.oracle.com/wim/>
- Create a local yum repository
<http://www.oracle.com/technetwork/articles/servers-storage-admin/yum-repo-setup-1659167.html>
- Full documentation of this lab
<https://www.oracle.com/technetwork/server-storage/vm/hol-oraclevm-templates-2349884.pdf>