Last Modified: 15-Apr-2017

Oracle VM templates for Oracle Database – Single Instance and Real Application Clusters 12c Release 1 (12.1.0.2.170418) Oracle Linux X86 64bit (OneCommand)

Table of Contents:

Overview:

This document introduces the <u>Oracle VM Templates for Oracle Database</u>. It explains the concepts of the templates and how to use the templates to deploy **Single Instance**, **Oracle Restart** (formerly known as: Single Instance/High Availability [SIHA]), or a fully functional N-node **Oracle Real Application Clusters** (RAC).

Templates are provided in 64-bit mode only:
- Oracle Linux 6 Update 9 64bit   **OR** Oracle Linux 7 Update 3 64bit

For the latest version of this document and more information on the benefits of Oracle VM and Templates, see OTN at: <u>http://www.oracle.com/technetwork/server-storage/vm/database-templates-12c-11gr2-1972804.html</u> and <u>Note# 1185244.1</u> on My Oracle Support.

NOTE: Oracle VM 3 users should use the deploycluster tool and documentation on the above URL.

Shared Disks Implementation Choices:

The shared disks holding the Oracle RAC database files may be configured as '**Physical'** or '**Virtual'** disks. Recently Oracle RAC Support policies allow Production deployments to use Virtual disks with some cautionary provisions (See Whitepaper below for details). Historically this document was split into two, however, going forward it is merged and users should read it based on if they use **Physical** or **Virtual** disks.

For more details carefully review the updated Oracle RAC on Oracle VM environments whitepaper at: <u>http://www.oracle.com/technetwork/database/clustering/oracle-rac-in-oracle-vm-environment-131948.pdf</u>

In **Single Instance deployments** above guidelines do not apply. Further details see Whitepaper above.

Naming Convention:

This document describes the **x86-64** (64bit) DB/RAC OVM Templates (12.1.0.2.170418). Template file & directory names are as follows:

OVM_OL6U9_X86_64_12102DBSE_PVHVM: Oracle Linux 6 U9, Oracle Database 12c **Standard Edition 2**
OVM_OL7U3_X86_64_12102DBRAC_PVHVM: Oracle Linux 7 U3, Oracle Database 12c **Enterprise Edition**

Screenshots refer to templates as: OVM_EL5U4_X86_**11GR2**RAC_PVM, these are generic screenshots, substitute your version number, e.g. 11202, 11203, 11204, 12101, 12102 based on release used.

NOTE: The Oracle OneCommand DB/RAC build scripts inside the template (under /u01/racovm directory) are only supported inside the Oracle VM DB/RAC Templates. Do not modify or run them in other environments.

Minimum Hardware Requirements:

The physical resources required for Single Instance and Oracle RAC [1]deployments are as follows:

Minimum Storage Requirements

Filesystem space consumption depends on SPARSE FILES[2] support for the repository (/OVS) as well as cloning method [3]in OVM3 and deployment mode[4]. Below are some approximations on space requirements:

	No Sparse file support	Sparse file support (approx)
Under /OVS (repository):	**64bit**	**64bit**
Template Storage	60GB	20GB
Each node of running VM	60GB	22GB
Temporary unzip Template[5]	**~6GB + ~6GB**	**~6GB + ~6GB**

Additional local/shared disk to hold database, minimum **5GB.**

Minimum CPU Requirements
Each VM requires a minimum of 1 physical core, recommended to run with 2 or more physical cores for the hosting server.

Minimum Memory Requirements
Default template memory is set to 3GB. Please follow the Oracle Database documentation for minimum memory requirements. At the time of writing minimum: **Single Instance:** 1GB RAM, **RAC:** 2GB RAM RAC deployment in test configuration may run with ~2GB of RAM; Clusterware only ~1GB of RAM.

Minimum Network Requirements
Network Adapters:
- **For RAC:** Each guest VM requires 2 network devices, which are 'virtualized' from physical cards in the hosting server, therefore a minimum of 2 physical NICS are required for the hosting server. In test configuration it is possible to use the templates with only 1 NIC, by assigning all guests' NICs to the single bridge (e.g. xenbr0) OR creating a second bridge (e.g. xenbr1) on a fake Dom0 NIC (eth1), also known as 'Host-Only networking' which would disable the live-migration option.
- **Single Instance:** One NIC is needed for the public network. Second NIC is optional.

IP addresses:
- **For RAC:** 5 public and 2 private (for a 2 node cluster). If a DNS Server is used, 2 additional IP addresses should be used to provide additional IPs for the SCAN name.
- **Single Instance:** 1 public IP

Certification & Support Information
See Note# 464754.1 on My Oracle Support for information on Certified Software on Oracle VM.

---

[1] When running Oracle RAC in a production environment, none of the Oracle RAC VM guest machines can run on the same Oracle VM Server.

[2]Oracle VM Server 3 as well as Oracle VM Server 2.2 (with OCFS2 v1.4.4) have sparse file support. Ext3/ext4 filesystem have sparse file support. NFS depends if server filesystem supports sparse files or not. Space consumed by sparse files varries by filesystem's minimum allocation unit often referred to as *cluster size*.

[3] If OVM3 "thin-clone" is used, space consumed will **greatly** reduce since VMs are "reflinked" (OCFS2 v1.8 feature) to the master template file and sections in the file are copied only when modified.

[4] Post deployment it is possible to remove unused Oracle homes in Single Instance or Clusterware only deployments

[5] Template stored in a tar.gz file inside a ZIP file hence size is listed twice

WHAT'S NEW:

- Applied all current recommended patches on top of **12.1.0.2.170418** (Oracle Linux 7 enabled)
- Both Oracle Database Enterprise Edition as well as Oracle Database Standard Edition 2 are released
- Upgraded OS to **Oracle Linux 6 Update 9** OR **Oracle Linux 7 Update 3** with OVMAPI support
- Default kernel boot upgraded to Unbreakable Enterprise Kernel (**UEKR4**) 4.1.12-61

Previous What's New:

- Automation support for 12c features:
  - Oracle **Flex Cluster** and/or **Flex ASM** (including add/remove N-number of Hub/Leaf nodes)
  - Creation/deletion of **Container Database** with optional N-number of **Pluggable Databases**
  - **DB Express** configuration for database
  - Dedicated **ASM Network Interface** (1st NIC: Public, 2nd NIC: Private, 3rd NIC: ASM)
  - Database can be created inside an ACFS Filesystem
- Automation support for **Single Instance & Oracle Restart** (Formerly known as Single Instance/HA)
- Automation support for databases on **local & shared filesystems**
- Automation support for **Admin Managed** or **Policy Managed** database creation
  - Including basic Server Pool creation (for Policy Managed)
- Allow multiple DNS IPs (see netconfig.txt)
- Add **OSRevert** utility: making switching rpm footprints easier 'basic' & 'db-minimum' predefined
- **Deploycluster** for OVM3 environments fully supports Single Instance mode
- Support for Yum@Deploy, add or update rpms at deploy time! (see netconfig.txt)
- Mutli diskgroup support for ASM (see FAQ & params.ini)
- Support for multiplexed Controlfile & Redo logfiles
- Templates are btrfs-ready (simple command to convert ext4 to btrfs filesystem, see FAQ)

Templates Description:

The **Oracle VM Templates for Oracle Database** allow building a **Single Instance**, **Oracle Restart** (Single Instance/HA), or **RAC clusters** of any number of nodes. The Oracle RAC 12c Release 1 (12.1.0.2.170418) software includes Oracle Clusterware and ASM (12.1.0.2.170418), Oracle Database 12c Release 1 (12.1.0.2.170418) and Oracle JVM (12.1.0.4.170418). In a Single Instance deployment the RAC feature is not linked into the binary and the Grid Infrastructure home may be removed for space savings.

See FAQ section for steps on how to add or remove patches before and/or after template deployment.

Note: The Templates include the latest OS & Oracle/RDBMS patches at time of the release. It is strongly recommended to update to latest patches (at time of deployment or placement into production usage) using standard OS/Oracle patching practices. OS updates may be obtained from Unbreakable Linux Network (ULN) and RDBMS patches from Critical Patch Updates (CPU) page on OTN. Templates released since 2014 offer Yum@Deploy, see FAQ for details.

The entire install is automated, with the user only providing the required input parameters (node names, IP addresses, etc.) when guests are booted. Oracle VM2 users can perform fully automated builds directly from Dom0, including automated network setup to all VMs. Oracle VM3 users should use the more powerful **Deploycluster** tool (downloadable from OTN with separate documentation) for fully

automated deployments of either **Single Instance**, **Oracle Restart** (Single Instance/HA) or **Oracle RAC**.

- The OS image contains a minimal install of Oracle Linux. Only basic RPM packages are installed with an option to reduce the footprint further using a tool called **OSRevert** (see FAQ).
- The default root password is 'ovsroot', oracle password is 'oracle' and sys/system password is 'oracle', all can be changed during the build. It is strongly advised to modify these passwords post-install. See section deploying the templates securely for further details.
- The install is performed using root access. With ssh passwordless login temporarily enabled for root user. At the completion of the install (on all nodes), ssh for root will be disabled.
- Optionally, the entire install could be done as the Oracle user, provided that 'sudo' access is configured on the current node only.
- During the installation process an Oracle Single Instance or RAC database is created. It is possible to skip this database creation and do a clusterware only install, see params.ini for various options.
- The OS and Oracle image disks inside the template are sparse files (non allocated space is not counted against filesystem), meaning the size displayed via 'ls -l' is not the actual space consumed on-disk. To see actual space consumed use 'ls -ls'. If /OVS filesystem has no sparse-file support then both sizes will be the same (all space is allocated upfront).
- Each template is delivered as 2 ZIP files (~6GB in size total), inside each ZIP is a **tar.gz** (tar gzip file), with very similar sizes, hence you will need ~12GB of temporary storage to fully expand each template. Oracle VM2: Once the template is imported into /OVS/seed_pool, these zip/tar.gz files may be deleted.
- OVM2 users planning to use guests running ext4 as boot device (currently all the OL5/OL6 based templates) require [Oracle VM 2.2.2](#) or above.

Feedback:

Feel free to post feedback at the Oracle VM or Oracle RAC Forums on the Oracle Technology Network:

Oracle VM:
https://community.oracle.com/community/server_%26_storage_systems/virtualization/oracle_vm_server_for_x86

Oracle RAC:
https://community.oracle.com/community/database/high_availability/rac__asm_%26_clusterware_installation

Or contact Oracle Support.

NOTE: The Oracle OneCommand DB/RAC build scripts inside the template (under /u01/racovm directory) are only supported inside the Oracle VM DB/RAC Templates. Do not modify or run them in other environments.

Deployment Overview

As mentioned above, by default, for an Oracle RAC deployment, each VM requires 2 (or more) network adapters (NICs) as well as 5 (or more) common shared disks (using fewer disks is supported). The steps to create such VMs are different for OVM2 and OVM3. The rest of this document explains the OVM2 methods.

A VM for Single Instance deployment does not need any shared disks or second NIC, simply clone a VM from the base template and deploy using deploycluster. You may also follow normal Oracle VM documentation to create the VMs for Single Instance or RAC deployment.

Deployment in an Oracle VM 3 environment

Oracle VM 3 users should use the [deploycluster tool](#) along with its documentation available on OTN. The tool provides for fully automated Single Instance, Oracle Restart or Oracle RAC deployments without ever logging into any of the VMs. See the template's README or Deploycluster documentation on how to import this template.

Main steps for a Single Instance deployment, using Oracle VM Manager 2

A Single Instance deployment only needs a single VM, no shared disks or second interface. As such, follow only steps **1, 3, 4 (only create 1 VM), 7 and 8** in the RAC section below.

Main steps for an Oracle RAC deployment, using Oracle VM Manager 2

These steps are using Oracle VM Manager; however all of them could be done from the command line, using Oracle VM Command Line Interface (CLI).

1) Load the template into the /OVS/seed_pool directory, on DOM-0:

```
# cd /tmp
# unzip -q /tmp/p24339999_10_Linux-x86-64_1of3.zip &     [ For SE2 use bug 24339997 ]
# unzip -q /tmp/p24339999_10_Linux-x86-64_2of3.zip &     [ For SE2 use bug 24339997 ]
# unzip -q /tmp/p24339999_10_Linux-x86-64_3of3.zip &     [ For SE2 use bug 24339997 ]
# wait
# cd /OVS/seed_pool
# tar xzf /tmp/OVM_OL7U3_X86_64_12102DBRAC_PVHVM-1of2.tar.gz &
# Note below 'cat' command should be all on one line
# cat /tmp/OVM_OL7U3_X86_64_12102DBRAC_PVHVM-2of2-partA.tar.gz
/tmp/OVM_OL7U3_X86_64_12102DBRAC_PVHVM-2of2-partB.tar.gz | tar xz &
# wait
```

 (When the above commands complete, the ZIP & TAR.GZ files may be deleted from /tmp)

This will create the following:
/OVS/seed_pool/OVM_OL7U3_X86_64_12102DBRAC_PVHVM
|- System.img (OS image file)
|- Oracle12102DBRAC_x86_64-xvdb.img (database software image file)
|- vm.cfg (VM configuration file)
|- README.txt
|- These PDF documents
|- utils (files to help with automated network setup and build

2.1) Create 5 shared Physical disks to hold the Database

Shared disks must be configured on DOM-0 of both OVM Servers, these should be persistently named physical devices (multi-pathed or not) accessible from both nodes, e.g.:

```
lrwxrwxrwx 1 root root 4 Dec 28 01:28 /dev/racdevc1 -> sdc1
lrwxrwxrwx 1 root root 4 Dec 28 01:28 /dev/racdevd2 -> sdd2
lrwxrwxrwx 1 root root 4 Dec 28 01:28 /dev/racdeve2 -> sde2
lrwxrwxrwx 1 root root 4 Dec 28 01:28 /dev/racdevf4 -> sdf4
lrwxrwxrwx 1 root root 4 Dec 28 01:28 /dev/racdevg1 -> sdg1
```

The following symlinked devices were created with udev rules similar to this one below, on Dom0, in /etc/udev/rules.d/55-oracle-devices.rules file:

```
KERNEL=="sd*", BUS=="scsi", PROGRAM=="/sbin/scsi_id -g -u -s %p",
RESULT=="360a98000686f61506434386f65663577", SYMLINK+="racdevc%n", OWNER="root",
GROUP="root", MODE="0640"
```

Using SYMLINK+= (instead of NAME= in udev rules) keeps the original kernel device names, e.g. /dev/sd*. Using udev rules ensures persistent device naming across reboots. The OneCommand RAC install requires 5 such devices, minimum size of 1024MB each (See APPENDIX C: Using More (or less) Disks). These will hold your database, size them appropriately

2.2) Create 5 shared Virtual disks to hold the Database

Using Oracle VM manager (or any other method) in our example they are named: ASM1, ASM2, ASM3, ASM4, ASM5, you may choose any name you wish. As this is a test environment you can use files visible in the Dom-0 environment to represent the shared disks.

- Select Resources->Shared Virtual Disks
- Click the **Create** button:
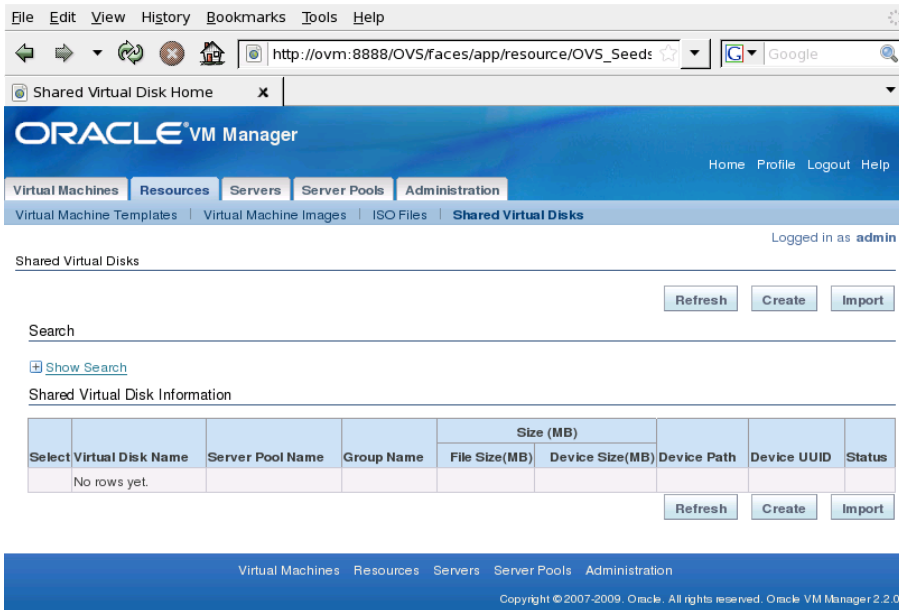
- Complete the fields on the form Click **Next** and **Confirm** on the following page.
- Repeat this step to create a total of 5 shared disks (Minimum size for each disk is 1024 MB)
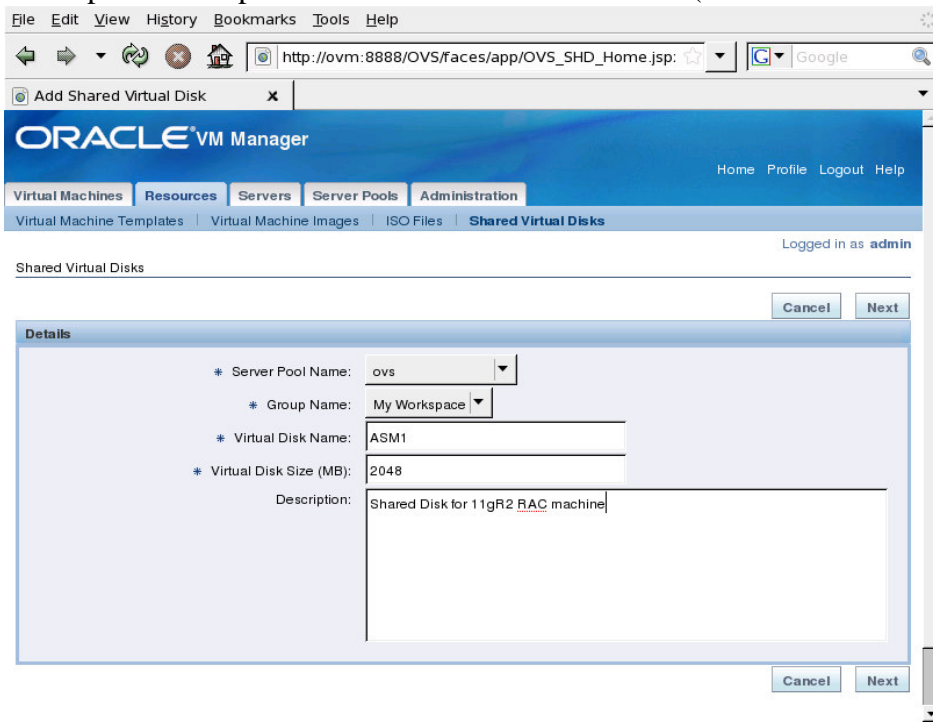
- When done creating all devices will have an **Active** status:
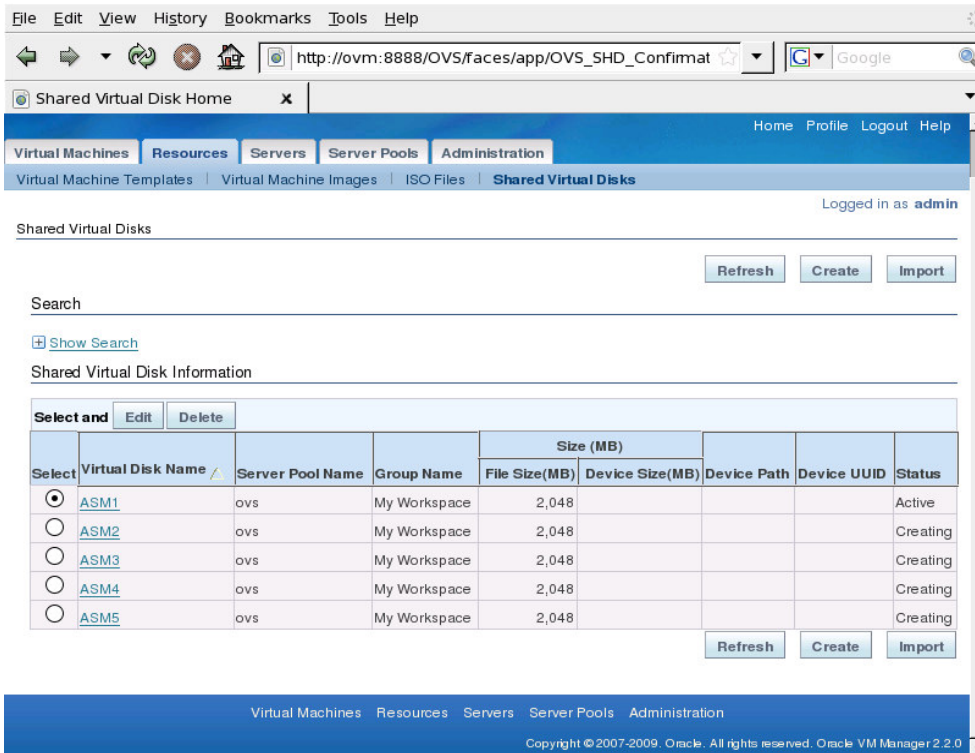
3) Import the template (from step 1) using Oracle VM manager 2

Make sure all unzipping/untarring operations completed before proceeding with these steps.

- Click Resources -> Virtual Machine Templates
- Click **Import** button
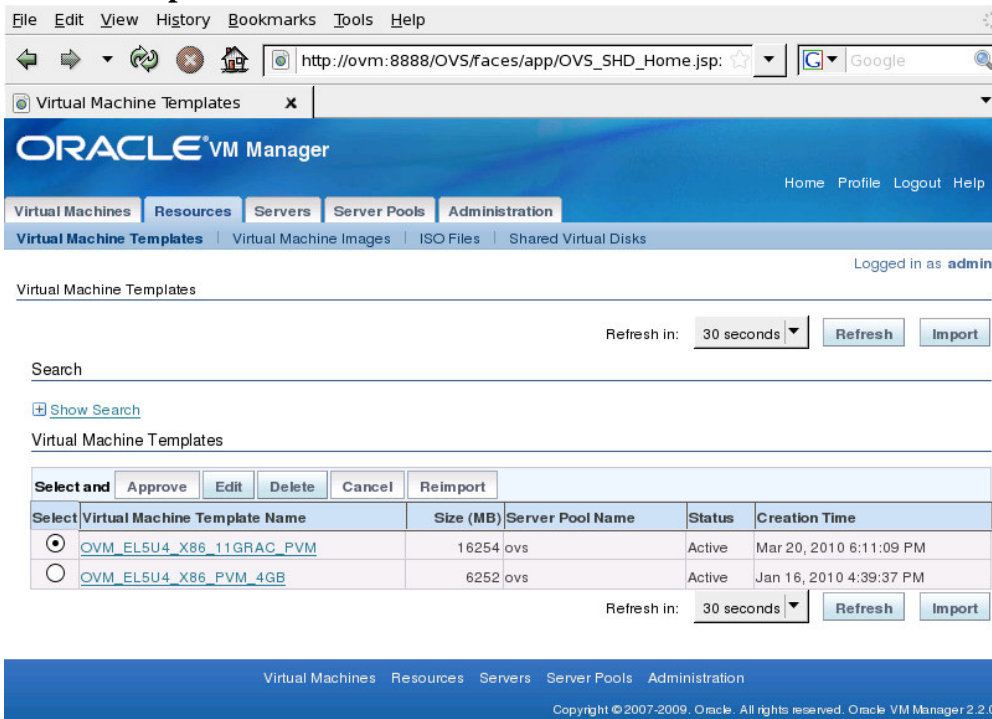
This Oracle VM manager already knows about 2 existing templates. We are going to add a 3rd template.

- Select "Select from Server Pool (Discover and register)", click **Next**
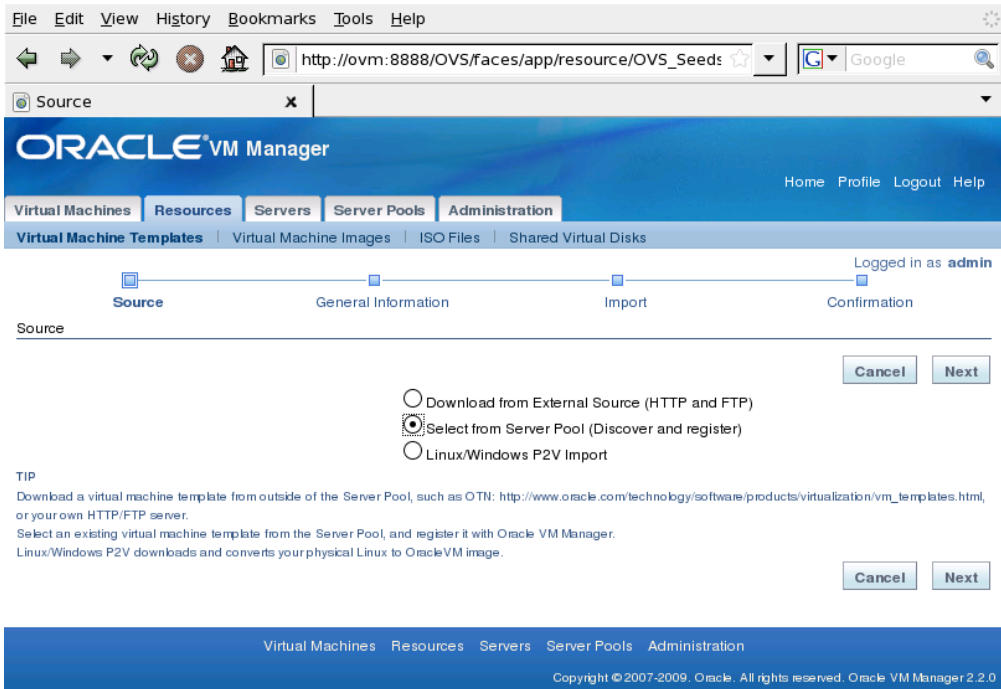
- Pull-down the correct "Server Pool Name"
- Choose the newly untarred template from pull-down "Virtual Machine Template Name".
- Select "Oracle Enterprise Linux 5" or "Oracle Enterprise Linux 5 64-bit" for "Operating System".
- Optionally enter Description or change username/passwords
- Click **Next** and **Confirm** on the following page.
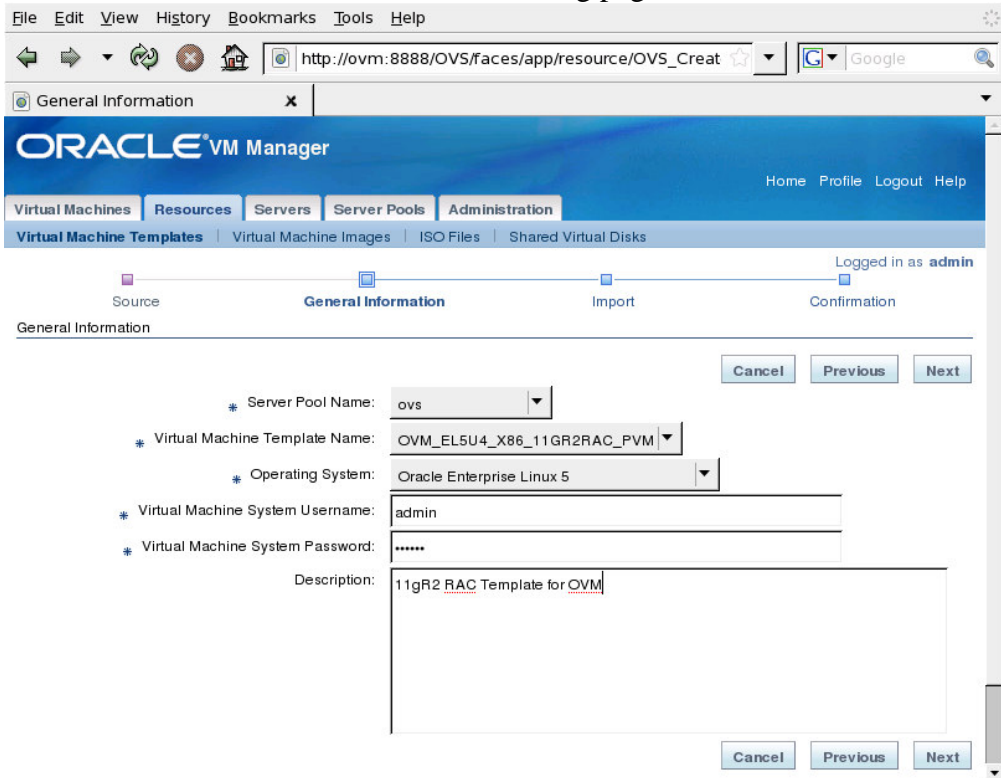
- Template name might be different in screenshot based on release used
- Wait for VM template to change from "Importing" to "Pending" then proceed
- Select Pending VM template from radio box, and Click **Approve** button
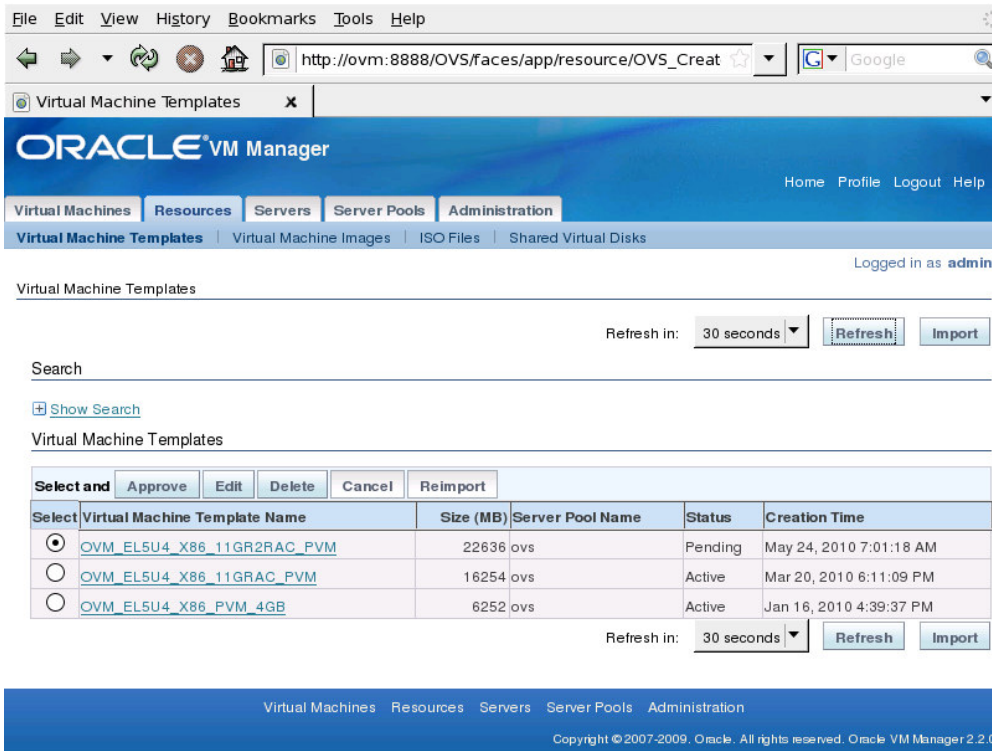
- ▪ Template name might be different in screenshot based on release used

- ▪ Click **Approve**
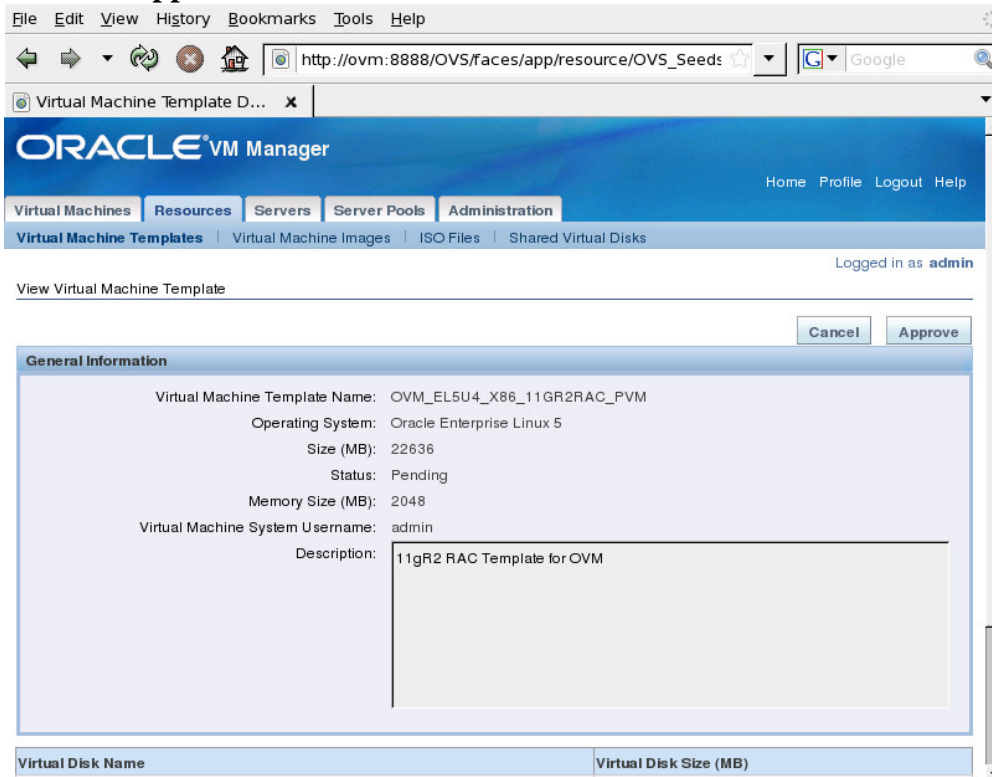
- ▪ Template name might be different in screenshot based on release used

The VM Template is now imported and ready to be deployed as many times as needed.

4) Create Virtual Machines using the imported template

In our example we name the VMs racnode1 & racnode2, you may choose any name you wish. For Single Instance or Oracle Restart (SIHA) deployment a single VM is needed.

IMPORTANT – The only non-default item you will need is to change the second NIC to xenbr1 (the first NIIC on xenbr0 will automatically be configured, see following page)

- Select Virtual machines
- Click on the **Create Virtual Machine** button
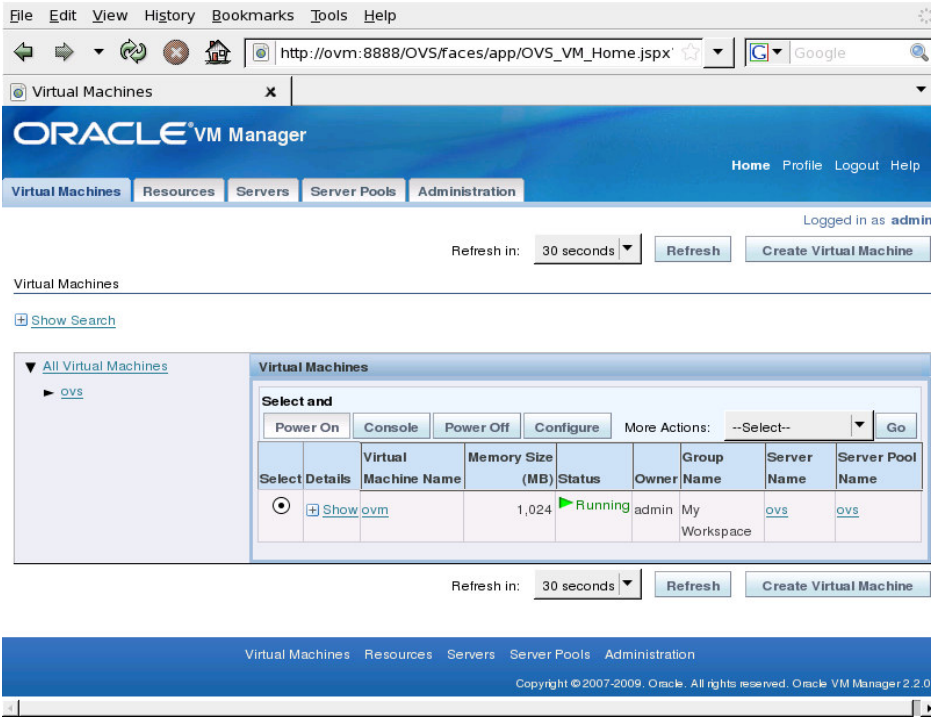
- Select the "Create virtual machine based on… template" radio button
- Click **Next**, and select the correct Server Pool on the following page.
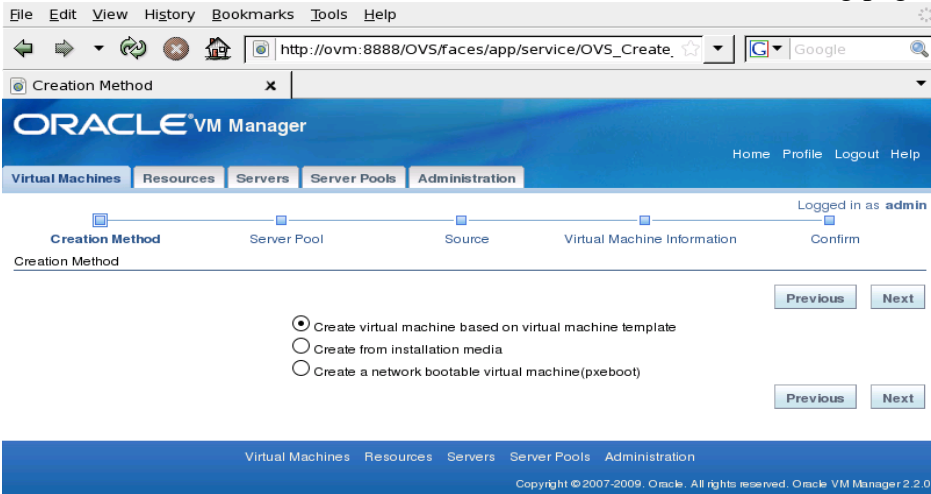
- Select the newly imported template (32bit or 64bit)
- Click **Next**
- Note: Template name might be different in screenshot based on release used
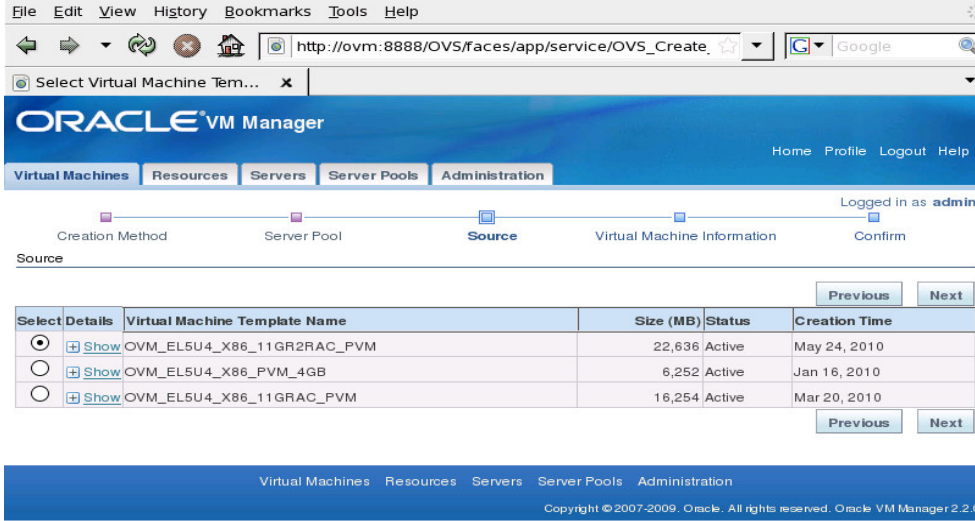
- **IMPORTANT:** Assign the second network adapter to **xenbr1**
- Enter the Virtual Machine Name: racnode1 (or racnode2 for second node)
- Enter the password twice – this will be the VNC password for the console
- Click **Next** and **Confirm** on the following page.
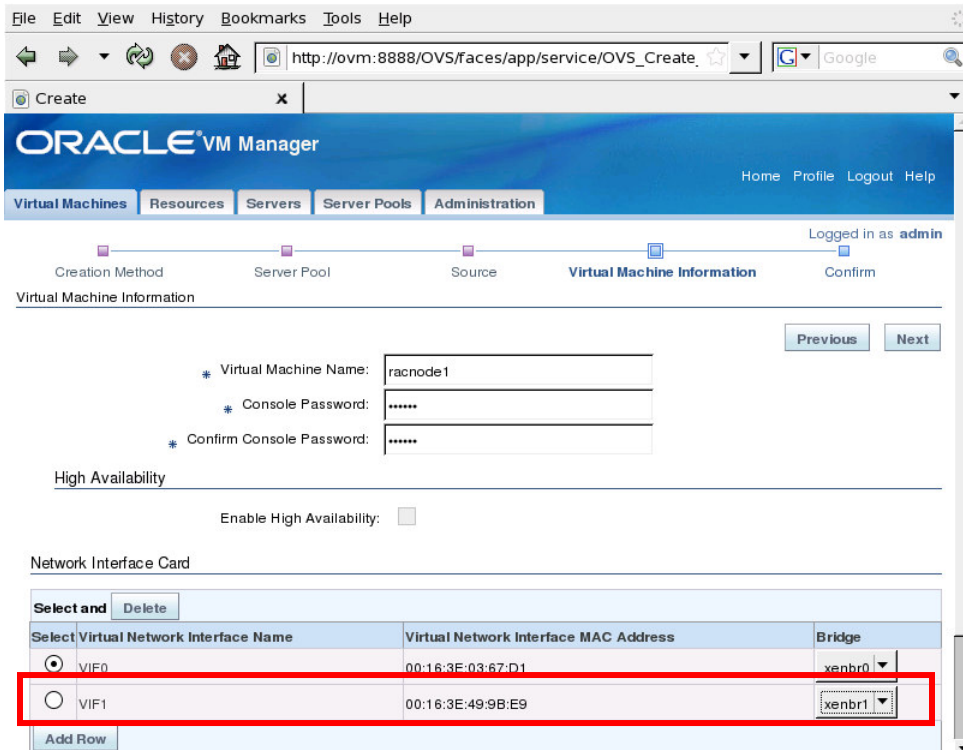
The VM will be built – This will take some time. Repeat this step for the second node.

* Do not power up the VMs yet

5.1) Add the shared Physical disks to both VMs.

Edit the vm.cfg for both VM's. These can be found on the Oracle VM Server in /OVS/running_pool/*racnode1/vm.cfg and /OVS/running_pool/*racnode2/vm.cfg, add:

```
'phy:/dev/racdevc1,xvdc,w!',
'phy:/dev/racdevd2,xvdd,w!',
'phy:/dev/racdeve2,xvde,w!',
'phy:/dev/racdevf4,xvdf,w!',
'phy:/dev/racdevg1,xvdg,w!',
```

to the 'disk' section, see below for sample vm.cfg.

The OneCommand for Oracle VM build engine, Clusterware, Oracle RAC or ASM do not require that disks have identical names or ordering on all nodes, however, it might be easier for humans to deal with identical names and ordering. Therefore it is recommended to add disks in same order with same disk names to all VMs.

5.2) Add the shared Virtual disks to both VMs.

- Select the Virtual machines tab
- Select the radio button for the first guest & Click the **Configure** button
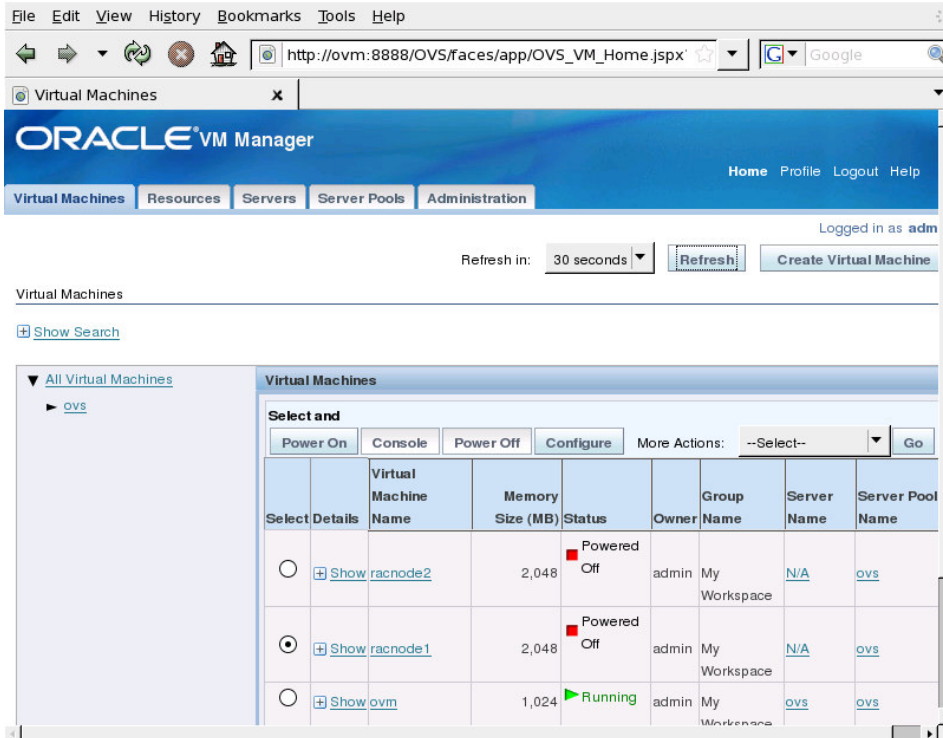
- Select the Storage tab
- Click the **Attach/Detach Shared Virtual Disk** button
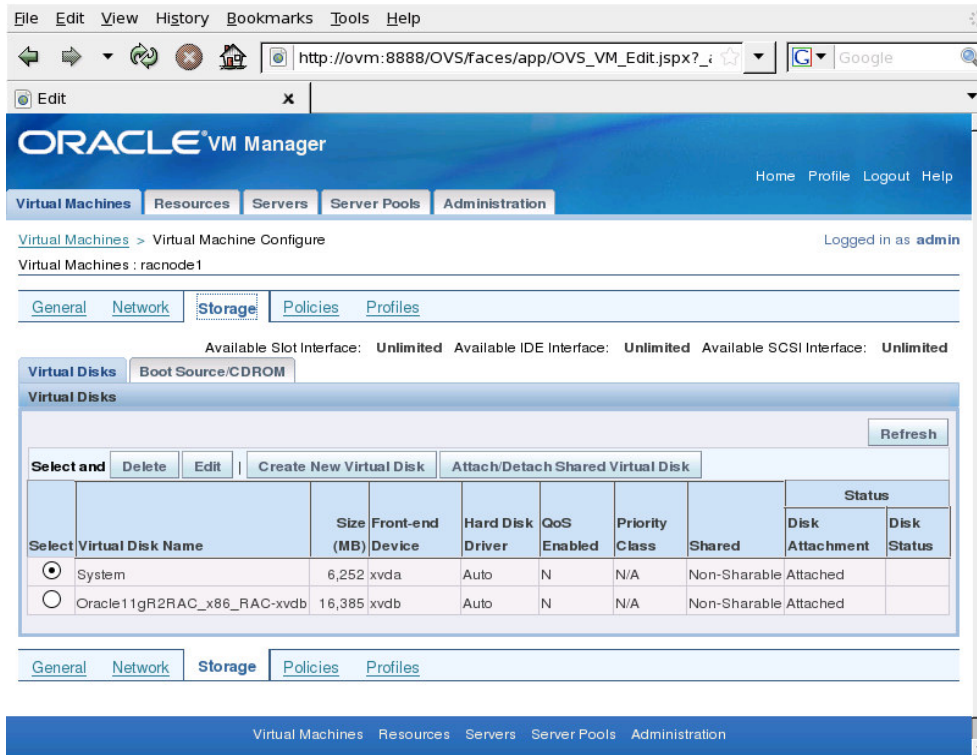
The OneCommand for Oracle VM build engine, Oracle RAC or ASM do not require that disks have identical names or ordering on all nodes, however, it might be easier for humans to deal with identical names and ordering. Therefore it is recommended to add disks in same order with same disk names to all VMs, e.g.; ASM1, ASM2, ASM3, ASM4, ASM5.

▪ Move at least 5 shared disks to the guest VM by clicking the appropriate **Move** buttons
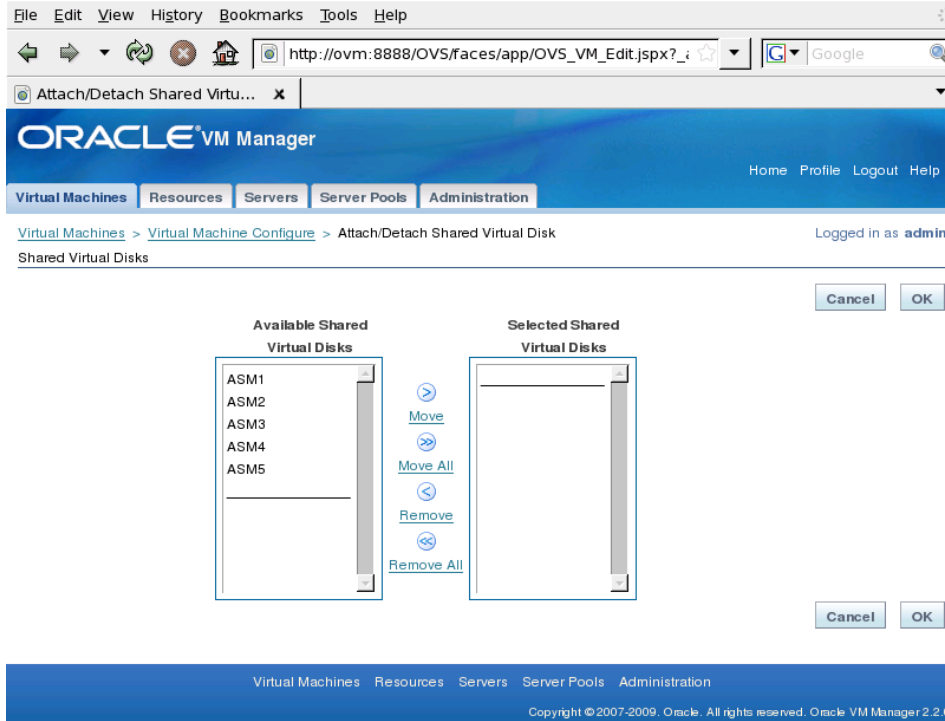
▪ Click the **OK** button
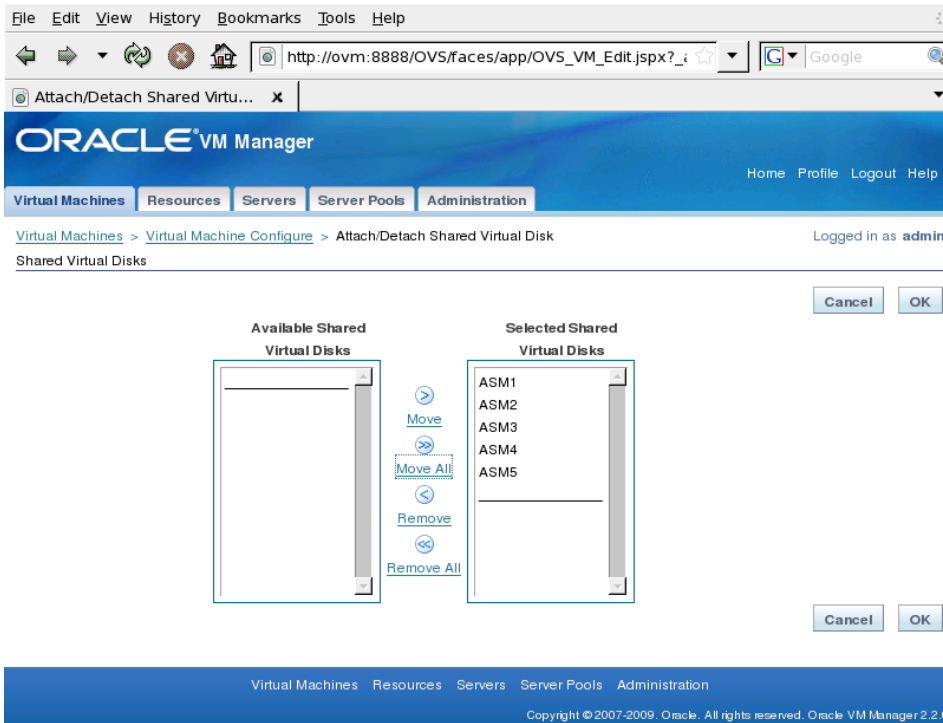
- Notice disk names: xvdc, xvdd, xvde, xvdf, xvdg. The automated Oracle RAC install process inside the templates depends on these exact device names. It is possible to use any disk names, see Appendix C: Using More Disks.
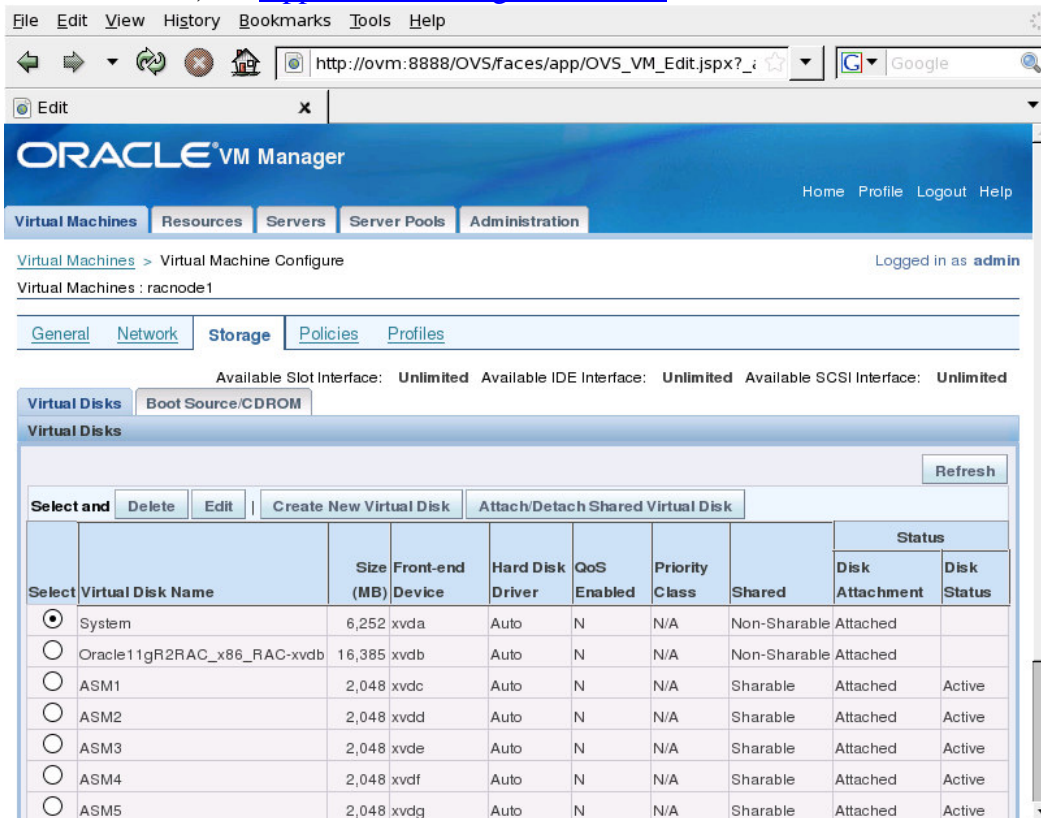
- Repeat the attach steps for the second virtual machine in the cluster

6.1) When complete, inspect the VM's vm.cfg configuration files – Physical disks

These can be found on the Oracle VM Server in /OVS/running_pool/*racnode1/vm.cfg and
/OVS/running_pool/*racnode2/vm.cfg. You should see the 5 shared disks which will be presented to the
guests as /dev/xvdc /dev/xvdd /dev/xvde /dev/xvdf /dev/xvdg. They can be in any order on both nodes as
described in the previous section. You should also be able to spot the 2 NICs, xenbr0 and xenbr1. This is a
sample vm.cfg file:

```
bootloader = '/usr/bin/pygrub'
disk = ['file:/OVS/running_pool/3_racnode1/System.img,xvda,w',
'file:/OVS/running_pool/3_racnode1/Oracle12102RAC_x86_64-xvdb.img,xvdb,w',
'phy:/dev/racdevc1,xvdc,w!',
'phy:/dev/racdevd2,xvdd,w!',
'phy:/dev/racdeve2,xvde,w!',
'phy:/dev/racdevf4,xvdf,w!',
'phy:/dev/racdevg1,xvdg,w!',
]
memory = '2048'
name = '3_racnode1'
on_crash = 'restart'
on_reboot = 'restart'
uuid = 'd98efba4-7445-8459-9326-adae28249200'
vcpus = 2
vfb = ['type=vnc,vncunused=1,vnclisten=0.0.0.0,vncpasswd=oracle']
vif = ['bridge=xenbr0,mac=00:16:3E:5C:AE:24,type=netfront',
'bridge=xenbr1,mac=00:16:3E:24:D3:34,type=netfront',
]
vif_other_config = []
```

If anything needs to change, use Oracle VM Manager to do so. Editing the file manually may put the repository out of sync.

6.2) Inspect the VM's vm.cfg configuration files (Optional) – Virtual disks

These can be found on the Oracle VM Server in /OVS/running_pool/*racnode1/vm.cfg and
/OVS/running_pool/*racnode2/vm.cfg. You should see the 5 shared disks which will be presented to the
guests as /dev/xvdc /dev/xvdd /dev/xvde /dev/xvdf /dev/xvdg. They can be in any order on both nodes as
described in the previous section. You should also be able to spot the 2 NICs, xenbr0 and xenbr1. This is a
sample vm.cfg file:

```
bootloader = '/usr/bin/pygrub'
disk = ['file:/OVS/running_pool/3_racnode1/System.img,xvda,w',
'file:/OVS/running_pool/3_racnode1/Oracle12102RAC_x86_64-xvdb.img,xvdb,w',
'file:/OVS/sharedDisk/ASM1.img,xvdc,w!',
'file:/OVS/sharedDisk/ASM2.img,xvdd,w!',
'file:/OVS/sharedDisk/ASM3.img,xvde,w!',
'file:/OVS/sharedDisk/ASM4.img,xvdf,w!',
'file:/OVS/sharedDisk/ASM5.img,xvdg,w!',
]
memory = '2048'
name = '3_racnode1'
on_crash = 'restart'
on_reboot = 'restart'
uuid = 'd98efba4-7445-8459-9326-adae28249200'
vcpus = 2
vfb = ['type=vnc,vncunused=1,vnclisten=0.0.0.0,vncpasswd=oracle']
vif = ['bridge=xenbr0,mac=00:16:3E:5C:AE:24,type=netfront',
```

```
'bridge=xenbr1,mac=00:16:3E:24:D3:34,type=netfront',
]
vif_other_config = []
```

If anything needs to change, use Oracle VM Manager to do so. Editing the file manually may put the repository out of sync

7) Power ON both virtual machines

- Select the Virtual Machines tab
- Select the radio button for the first node
- Click the **Power On** button, repeat for second node (Single Instance just start the single VM)
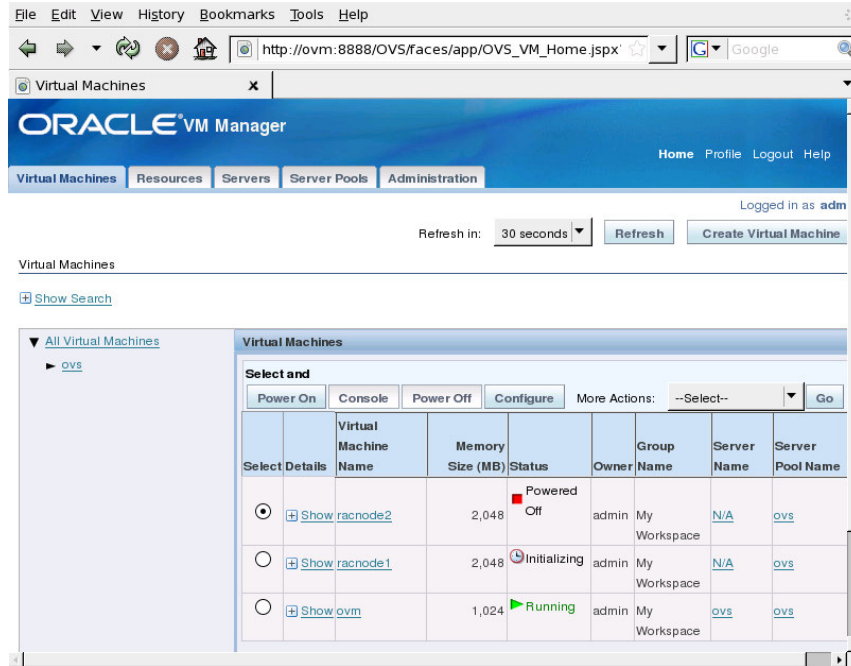
- If the VNC plugin has been configured for Oracle VM manager you can connect to the console by selecting the radio button next to the node and clicking the **Console** button. Open both consoles.
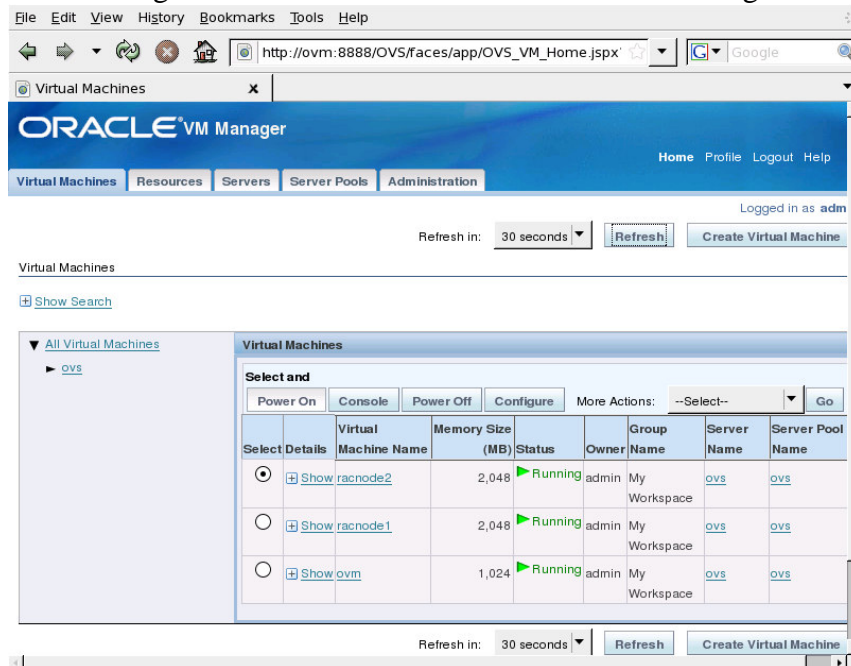
Alternatively, use vncviewer from any PC; First get the ports from Dom0:

Dom0 # **xm list**
```
Name                        ID    Mem   VCPUs    State   Time(s)
3_racnode1                   1    2048     2     -b----    72.4
4_racnode2                   2    2048     2     r-----    68.1
```
Dom0 # **xm list -l** <VM-full name or ID> **| grep location**

Use the port that appears on line as "location 0.0.0.0:**5901**" e.g.:
$ vncviewer <ovm_server_host>:5901   (substitute your OVM Server node name)

A third alternative, for advanced users is described in APPENDIX D: Using Text Mode Console (Oracle VM2)

8) Complete the First Boot interview:

First Boot is the time when a newly instantiated Guest VM is started for the first time.
At this point in time the guest machines are identical – they have no personality e.g. no hostname, no IP address etc. First boot adds this personality.

The **deploycluster** tool available on OTN for Oracle VM 3 environments can easily send the complete network and build information to the VMs to quickly deploy a fully functional Single Instance or RAC environment. Oracle VM 2 users may use the traditional 2-node interview shown below, simply type **yes** at the only required key/question during first boot console interview:

'com.oracle.racovm.netconfig.interview-on-console'
'Do you want to access the 2-node RAC interview on console ? [YES|NO]

Preparing for a Single Instance build

Oracle VM 2 users deploying Single Instance, should type **OUT** (at the above question) and setup a simple file called **netconfig.ini** based on the sample/provided netconfig-sample-si.ini to include only Public Name/IP of one VM, Public Adapter/Mask and optionally DNSIP, DOMAINNAME and PUBGW (Public Gateway). Set either CLONE_SINGLEINSTANCE=yes (for Single Instance) or CLONE_SINGLEINSTANCE_HA=yes (for Oracle Restart, aka SIHA), if both are omitted then a RAC deplyoment is assumed. Once that file is ready, simply type:

> # /u01/racovm/netconfig.sh -n1

Then skip to next section for running **buildsingle.sh.**

Preparing for an Oracle RAC build

The following is displayed on the console of both guests. You must identify which node is going to be the master; the interactive interview will run on that node.

```
  == Oracle DB/RAC [Version] OneCommand for Oracle VM - template configuration (x86_64) ==

    NODE SELECTION
    ==============
    Is this the first node in the cluster  (YES/NO): [    ]
```
Figure 1

On **Node1** Enter: **YES**   (On **Node2** Enter: **NO)**

The configuration utility then waits until the **NO** response has been given on the second node.

Node1

```
   == Oracle DB/RAC [Version] OneCommand for Oracle VM - template configuration (x86_64) ==

     NODE SELECTION
     ==============
     Is this the first node in the cluster  (YES/NO): [YES]


     IMPORTANT
     =========
     Please answer this question on the other node in the cluster
     WAITING
........
```
Figure 2

Go to the Node2 console screen

Node2

```
  == Oracle DB/RAC [Version] OneCommand for Oracle VM - template configuration (x86_64) ==

     NODE SELECTION
     ==============
     Is this the first node in the cluster  (YES/NO): [NO ]
WAITING for NODE1 interview phase to be complete!
...
```
Figure 3

Enter: **NO**

You may enter the responses in any order. This is all the interaction required on the second node. All subsequent configuration steps now take place on the first node.

After completing the node identification steps you must now provide information for first boot to complete. A simple data entry screen is displayed on Node1:

Node1

```
NODE DETAILS            NODE 1                            NODE 2
 ===========  ------------------------------   -------------------------------
 Public Name:[racnode1                     ] [racnode2                      ]
 Public IP  :[10.10.10.200   ]              ❶  [10.10.10.201    ]           ❷
 Private Name[racnode1-priv                ] [racnode2-priv                 ]
 Private IP :[192.168.1.200  ]              [192.168.1.201   ]
 VIP Nodename[racnode1-vip                 ] [racnode2-vip                  ]
 VIP IP     :[10.10.10.202   ]              [10.10.10.203    ]
 GLOBAL DETAIL     ❸
 ============
 Domain Name :[us.oracle.com               ] DNS Server IP :[              ]
 Public Network Adapter Name :[eth0         ]  ❹
 Public Subnet mask           :[255.255.255.0 ]
 Default Gateway IP           :[10.10.10.200  ]
 Private Network Adapter Name:[eth1          ]
 Private Subnet mask          :[255.255.255.0 ]

 CLUSTER DETAILS
 ===============                     ❺    ❻
```

```
Enter the Cluster Name :[rac-cluster    ]
Enter the SCAN name for this cluster  :[rac-cluster       ][10.10.10.204    ]
   Enter YES to accept the above values, NO to  reenter: [YES]  ❼

   DO YOU WANT TO CONFIGURE THIS CLUSTER YES/NO : >YES <
```
Figure 4

Enter the relevant values, basic user input validation is performed, if you make a mistake – at the reenter prompt enter 'NO' to cycle through the interview process.

Step 1 & 2: Node1&2 details, use your own machines names / IP addresses
   Basic IP address and name validation is performed. Node names cannot contain dots (use domain name for that). All Public and VIP IPs must be on same subnet; similarly the Private IPs should be on the same subnet, no duplicate IPs or node names allowed.

Step 3: Optionally enter Domain name and DNS Server IP. If no domain name is entered: "localdomain" is used. DNS Server IP is optional. If a DNS Server is specified you must configure the Grid Infrastructure SCAN name/IP Addresses within the DNS Server

Step 4: NIC & general network details, use your own network details
   Adapter names must exist and be different from one another, typically they will be eth0 and eth1. Network masks must be in dot notation format, e.g. 255.255.252.0

Step 5: Unique Clustername - 1 to 15 alphanumeric or hyphens – no spaces

Step 6: The SCAN Name. If a DNS Server IP address was NOT entered previously then an IP Address for the SCAN will be requested – this should be in the same subnet as the public/VIP network

Step 7: Opportunity to reenter the configuration data and a final chance to exit without saving

When you select YES at the final confirmation prompt you will see this on both VMs as the network fully configures itself as per interview:

Node1 and Node2

```
    Generating Network Configuration files...
    Modifying /etc/hosts
    Modifying /etc/resolv.conf
    Modifying /etc/sysconfig/network
    Modifying /etc/sysconfig/network-scripts/ifcfg-eth0
    Modifying /etc/sysconfig/network-scripts/ifcfg-eth1
Shutting down interface eth0:                              [  OK  ]
Shutting down interface eth1:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:                                [  OK  ]
Bringing up interface eth1:                                [  OK  ]
Shutting down kernel logger:                               [  OK  ]
Shutting down system logger:                               [  OK  ]
Starting system logger:                                    [  OK  ]
Starting kernel logger:                                    [  OK  ]
Shutting down Avahi daemon:                                [  OK  ]
Starting Avahi daemon...                                   [  OK  ]
Starting nscd:                                             [  OK  ]
Reloading nscd:                                            [  OK  ]
```

```
Template configuration disabled.

INFO (node:racnode1): Network configuration completed...

Oracle Linux Server release 5.6
Kernel 2.6.18-238.0.0.0.1.el5xen on an x86_64

racnode1 login: root
Password:
```
Figure 5

If for some reason you entered incorrect data (no network between the two nodes) you may redo just this interview phase by logging in as root to both machines and invoking: **/u01/racovm/netconfig.sh.** <u>Do not run this after the installation of RAC software is complete.</u>

Install Single Instance or Grid Infrastructure with Oracle RAC

You have now completed the node preparation phase and the nodes are ready to configure the software.

Standard Single Instance Build:

If you followed the above Single Instance instructions there should be either CLONE_SINGLEINSTANCE=yes (for Single Instance) or CLONE_SINGLEINSTANCE_HA=yes (for Oracle Restart, aka SIHA) in the netconfig.ini file. By default, a standard Single Instance install will create a Single Instance database called 'ORCL' (sid ORCL) all database passwords will be 'oracle'. To proceed with the default build (for custom build see APPENDIX A: Build Options), login as root (default root password is ovsroot), and run the commands:

```
[root@single ~]# cd /u01/racovm
[root@single racovm]# ./buildsingle.sh
Are you sure you want to install Single Instance?
Do not run if software is already installed and/or running..  [yes|no]? yes
```

If you answer anything other than "yes" the installation will not continue.

Standard RAC Build:

By default, a standard RAC install will create an Oracle RAC database called 'ORCL' (sid ORCL1 & ORCL2) all database passwords will be 'oracle'. To proceed with the default build (for custom build see APPENDIX A: Build Options), on node1 only, login as root (default root password is ovsroot), and run the commands:

```
[root@racnode1 ~]# cd /u01/racovm
[root@racnode1 racovm]# ./buildcluster.sh
Are you sure you want to install Oracle RAC?
Do not run if software is already installed and/or running..  [yes|no]? yes
```

If you answer anything other than "yes" the installation will not continue. <u>You do not need to run anything on the second node.</u>

Monitoring the Build:

There is an install log in the current directory named: buildcluster1.log or buildsingle1.log (each run will get the lowest free numbered file). By default the progress log is located at /tmp/progress-racovm.out.

Typically the full install will take ~30-45 minutes (~7-20min for Single Instance) although this is dependant on the hardware used to host the VMs and the build options. You can watch the progress of the install by viewing progress logfile on node1:

```
[root@racnode1 racovm]# cat /tmp/progress-racovm.out
2010-06-17 14:12:52:[buildcluster:Start:racnode1] Create 11gR2 RAC Cluster
2010-06-17 14:12:52:[setsshroot:Start:racnode1] SSH Setup for the root user...
2010-06-17 14:12:52:[setsshroot:Done :racnode1] SSH Setup for the root user completed successfully
2010-06-17 14:12:52:[setsshroot:Time :racnode1] Completed successfully in 0 seconds (0h:00m:00s)
2010-06-17 14:12:52:[copykit:Start:racnode1] Copy kit files to remote nodes
2010-06-17 14:12:52:[copykit:Done :racnode1] Copy kit files to remote nodes
2010-06-17 14:12:52:[copykit:Time :racnode1] Completed successfully in 0 seconds (0h:00m:00s)
2010-06-17 14:12:52:[usrsgrpslocal:Start:racnode1] Verify Oracle users & groups (create/modify mode)..
2010-06-17 14:12:53:[usrsgrpslocal:Done :racnode1] Verify Oracle users & groups (create/modify mode)..
2010-06-17 14:12:53:[usrsgrpslocal:Time :racnode1] Completed successfully in 1 seconds (0h:00m:01s)
2010-06-17 14:12:54:[usrsgrps:Time :racnode1] Completed successfully in 2 seconds (0h:00m:02s)
2010-06-17 14:12:54:[printparams:Time :racnode1] Completed successfully in 0 seconds (0h:00m:00s)
```

The install log printed to the screen will have all commands executed in verbose mode, so you can see as the various tools, like clone.pl, netca, dbca, emca are executed along with their output.

Default Installation Specifications:
SID: ORCL1 & ORCL2
DB name: ORCL
Grid Infrastructure Home: /u01/app/11.2.0/grid **or** /u01/app/12.1.0/grid
Oracle RAC Home: /u01/app/oracle/product/11.2.0/dbhome_1 **or** /u01/app/oracle/product/12.1.0/dbhome_1
ORACLE_BASE: /u01/app/oracle
Central Inventory: /u01/app/oraInventory
Root OS Password: ovsroot

APPENDIX A: Build Options

Before invoking **/u01/racovm/buildcluster.sh** (or buildsingle.sh for Single Instance) you may edit **/u01/racovm/params.ini** to modify some build options (bottom part of the file). The top part of params.ini should be modified by advanced users or if instructed to by Oracle Support. If using **deploycluster** tool (OVM3 only), a custom params.ini may be passed using the **-P** flag, it will be sent to all VMs and the buildcluster (or buildsingle) will then use that instead of the shipped params.ini inside the VM. Small samples of the options that may be modified taken directly from params.ini:

```
#
# Build Database? The BUILD_RAC_DATABASE will build a RAC database and
# BUILD_SI_DATABASE a single instance database (also in a RAC environment)
# Default: yes
BUILD_RAC_DATABASE=yes
#BUILD_SI_DATABASE=yes


# The Database Name
# Default: ORCL
DBNAME=ORCL


#
# The Instance name, may be different than database name. Limited in length of
```

```
# 1 to 8 for a RAC DB & 1 to 12 for Single Instance DB of alphanumeric characters.
# Ignored for Policy Managed DB.
# Default: ORCL
SIDNAME=ORCL


# Configures a Single Instance environment, including a database as
# specified in BUILD_SI_DATABASE. In this mode, no Clusterware or ASM will be
# configured, hence all related parameters (e.g. ALLDISKS) are not relevant.
# The database must reside on a filesystem.
# This parameter may be placed in netconfig.ini for simpler deployment.
# Default: no
#CLONE_SINGLEINSTANCE=no


# Configures a Single Instance/HA environment, aka Oracle Restart, including
# a database as specified in BUILD_SI_DATABASE. The database may reside in
# ASM (if RACASMGROUPNAME is defined), or on a filesystem.
# This parameter may be placed in netconfig.ini for simpler deployment.
# Default: no
#CLONE_SINGLEINSTANCE_HA=no


# Local Listener port number (default 1521)
# Default: 1521
LISTENERPORT=1521


# Allows color coding of log messages, errors (red), warning (yellow),
# info (green). By default no colors are used.
# Default: NO
CLONE_LOGWITH_COLORS=no
```

If you do not wish to store the passwords for the root or Oracle user in the configuration file, remove or comment them, and they will be prompted for at the start of the build.

APPENDIX B: Adding More Nodes (Oracle VM2)

Oracle VM 3 users should use the new deploycluster tool along with its documentation available on OTN.

The tool provides for fully automated Single Instance and RAC deployments without ever logging into any of the VMs. Below steps are for Oracle VM 2 users.

It is possible to add nodes to the 2-node cluster (after above build is done) using the kit as follows:

1. Follow step 4 in this document to create a 3rd or 4th VM, using Oracle VM Manager or Oracle VM Command Line Interface (CLI).

2. Follow step 5 and attach the same shared storage as the existing cluster nodes have.

   Boot the additional VM(s) and connect to console via VNC or text mode console (See APPENDIX D: Using Text Mode Console (Oracle VM2))

3. At the node selection prompt type "OUT":

   ```
   == Oracle DB/RAC [Version] OneCommand for Oracle VM – template configuration (x86_64) =

       NODE SELECTION
       ==============
       Is this the first node in the cluster  (YES/NO): [OUT]
   ```

4. There will be no interview. The VM will be called "localhost" and will have no network configured.

   From one of the existing VMs copy & paste (available only via text mode console, see APPENDIX D: Using Text Mode Console (Oracle VM2)) the contents of the file **/u01/racovm/netconfig.ini** as **/u01/racovm/netconfig.ini** in the new VM (overwriting the current file). If you couldn't get the copy & paste to work you can use a normal text editor to type that information in the new VM.

5. Edit **/u01/ racovm/netconfig.ini** on the new VM and add the information about the new VMs, e.g.:

   ```
   NODE3=racnode3
   NODE3IP=192.168.1.205
   NODE3PRIV=racnode3-priv
   NODE3PRIVIP=10.10.10.205
   NODE3VIP=racnode3-vip
   NODE3VIPIP=192.168.1.206
   ```

6. Invoke the network configuring script in silent mode as root on the new VM. Be sure that you are connected via the console (either VNC or text mode) as this will establish/restart the network:

   # /u01/racovm/netconfig.sh -n3

   Flag **-n3** means 3rd node, **-n4** would be 4th node, etc. Do not run this procedure on a node with Oracle RAC software already configured.   This will configure the network in the VM to the values of NODE**3*** listed in netconfig.ini. You may add as many nodes as you wish in this fashion. If you

enter information about other nodes in netconfig.ini (e.g. NODE4* or NODE5*) the generated /etc/hosts file will contain all their details. Note that silent mode network configuration allows longer node or domain names compared to the text mode interview.

The add node procedure will automatically propagate this more complete /etc/hosts to the existing nodes, so that all nodes know about the others. If the hosts are registered in DNS, the /etc/hosts file may be skipped by setting resolver order in /etc/nsswitch.conf.

7. Now that network is up on the new node, copy the more up-to date **/u01/racovm/netconfig.ini** from the new node (it should contain all newly added nodes as well as existing nodes) to any existing cluster member (e.g. racnode1) where you plan to run the addnode(s) procedure from.

   # scp /u01/racovm/netconfig.ini racnode1:/u01/racovm

8. Finally, run the addnode(s) procedure from the cluster member (e.g. racnode1) you copied the updated netconfig.ini to:

   # cd /u01/racovm
   # ./racovm.sh -S addnodes -N racnode3   2>&1 | tee addnode3.log

The "2>&1 | tee " means save stdout and stderr to addnode3.log, this is useful since errors are printed to stderr, so using only "tee" or ">" will only capture stdout without any possible errors. If you do not need logging you may omit all of that.

It is possible to add several nodes in one command invocation, just separate them with a comma. By default, new database instance(s) will not be created on the new node(s), if that is required, add the step "addinstances" to the above command, or run it separately at a later time. Here is the sample command to add 2 nodes with their instances:

 # ./racovm.sh -S addnodes,addinstances -N racnode3,racnode4   2>&1 | tee addnode-inst-node3-4.log

Or, to just add instances on a new node (assuming it was already added using addnodes as described above):

 # ./racovm.sh -S addinstances -N racnode3   2>&1 | tee addinstances-node3.log

APPENDIX C: Using More (or less) Disks

Advanced users may want to use differently named, less than or more than 5 disks for database/ASM.

> **NOTE:** Templates released in 2014 and after support ASMLib disk names in the ALLDISKS and all other disk variables (see params.ini), ALLDISKS_ASMLIB retained for backwards compatibility.

Before the Build:

There is an option to use differently named or more than 5 disks before the build. If you attached more than 5 shared devices to the VMs, before you run **buildcluster.sh** or **buildsingle.sh**, edit **/u01/racovm/params.ini** and modify **ALLDISKS** to contain the list of all devices or partitions that should be used by the database/ASM. The **params.ini** file describes the rules for setting this parameter. Remember that whatever disks/partitions are specified should be discoverable by **RACASMDISKSTRING** in the **params.ini** file who's default is "/dev/xvd[c-g]1". Do not set the discovery string to a too wide value, e.g. /dev/*, since the udev rule is written based on this string, and so might affect devices you did not intend on affecting, like the OS boot device!

As an example – if you add a 6th device: /dev/xvdh.

> **ALLDISKS**="/dev/xvdc /dev/xvdd /dev/xvde /dev/xvdf /dev/xvdg /dev/xvdh"

The **RACASMDISKSTRING** should be:

> **RACASMDISKSTRING**="/dev/xvd[c-h]1"   (Discovers xvdc1, xvdd1, through xvdh1)
>
> **WRONG: RACASMDISKSTRING**="/dev/xvd?1" (Since it also discovers xvda1, xvdd1 which
>   are the /u01 and /boot mount points by default)

The discovery string may have multiple entries separated by space.

By default, the build procedure will automatically partition all the disks with 1 partition each. An already partitioned disk, e.g. /dev/xvdc3 will be used as-is. It will also write a new UDEV rule, default rule file is: /etc/udev/rules.d/99-oracle.rules.

NOTE: If you want to use less than the default and recommended 5 disks (for example 3), set ASM_MIN_DISKS=3 in params.ini.

After the build:

After the build, simply add more disks to the VMs, partition them correctly and add them to ASM as you normally would. If you follow this route remember to modify the UDEV rules file on all nodes to give the new disk the correct permissions. Default rules file is: /etc/udev/rules.d/99-oracle.rules.

APPENDIX D: Using Text Mode Console (Oracle VM2)

Dom0 access is not supported in Oracle VM 3, hence these steps are for Oracle VM 2 users only. Oracle VM 3 users should connect to VM's VNC/GUI console from the Manager UI. Using the **deploycluster** tool avoids the need for console access altogether.

Advanced users may wish to connect to console during boot of the VM, useful to allow copy/paste when no network is setup. From Dom0, boot the VMs using the following command; it will append the arguments in "extra" to the kernel boot line, this works **only for PVM guests**:

Dom0# **cd /OVS/running_pool/<name-of-VM>**
Dom0# **xm create -c vm.cfg extra="console=hvc0"**

This will boot the VM and connect the console to your terminal.
Or, if you prefer to do it manually or if the guest is running in PVHVM use these steps:

Dom0# **cd /OVS/running_pool/<name-of-VM>**
Dom0# **xm create -c vm.cfg**

Then on the grub menu (first screen), select the kernel to boot using the up/down arrow keys, then type **e** (for "edit"), you will see a screen similar to this:

```
    pyGRUB   version 0.6
    root (hd0,0)
    kernel /vmlinuz-2.6.18-238.0.0.0.1.el5xen ro root=LABEL=/ numa=off
    initrd /initrd-2.6.18-238.0.0.0.1.el5xen.img
```

Scroll to the (**kernel** boot line), type **e** (for "edit") and add a space followed by **console=hvc0** (for PVM guests), or **console=ttyS0** (for PVHVM guests) to the end of that line. Press **enter** then **b** or **ctrl-x** (to boot). The console is now connected to your terminal.

Note: In grub edit mode BACKSPACE key may not work, use CTRL-H instead.

Following either of the above methods will allow copy and paste into/from the VM's console; this is only needed when network is not yet setup.

NOTE: Oracle Linux 7 template users should also add "**plymouth.enable=0"** to the kernel boot line above to allow proper echoing of commands if the character mode console is used. This disables the plymouthd daemon during boot and will allow the characters entered by the first boot interview to be echoed to the console.

APPENDIX E: Building Clusters with more than 2 Nodes (Oracle VM2)

Oracle VM 3 users should use the new deploycluster tool along with its documentation available on OTN. The tool provides for fully automated RAC deployments without ever logging into any of the VMs.

In order to build larger clusters, the network needs to be configured on all nodes. There are two methods of setting up the network on multiple nodes. Assume we want to build a 4 node cluster here are the two methods to setup the network and build the Oracle RAC cluster:

Partial Interview:

1. Instantiate the 4 VM's including adding the shared storage to all 4 VMs

2. Boot all 4 VMs. They will all pause at the 'Is this the first node in the cluster [ ]' question

   On Node1 answer YES   On Node2 answer NO
   One the other node in the cluster enter the word OUT

3. On node1 complete the network config screen for the first 2 nodes in the cluster as a normal 2-node cluster, and let them continue to configure the network. The hostnames on nodes1 and node2 will be correct. On the other nodes it will show localhost – and network will not be configured.

4. On Node1 login as root (default password ovsroot) and edit the /u01/racovm/netconfig.ini file. You will see that for each node 6 lines have been written. Copy the 6 lines and paste for each new node, modify the node number as well as network information, e.g.:
   ```
   NODE3=racnode3
   NODE3IP=192.168.1.205
   NODE3PRIV=racnode3-priv
   NODE3PRIVIP=10.10.10.205
   NODE3VIP=racnode3-vip
   NODE3VIPIP=192.168.1.206
   ```
   Add another such block for node4, all variables will start with NODE4, e.g. NODE4IP=…

5. Update the /etc/hosts file on node1 & node2 (since their network is already configured):

   On node1:
   # **cd /u01/racovm**
   # **./netconfig.sh -n1 -H**   (-H means to only update /etc/hosts based on content of netconfig.ini)
   # **scp netconfig.ini racnode2:/u01/racovm**
   # **ssh racnode2 /u01/racovm/netconfig.sh -n2 -H**   (will update /etc/hosts on node2)

6. **CHOOSE ONE METHOD 6.1 or 6.2:** Create an exact copy of the updated file in the /u01/racovm directory on node3 & node4 and use it to configure their network. Choose one of these two methods:

   6.1 Copy & paste the entire netconfig.ini file using the console in text mode, see APPENDIX D: Using Text Mode Console (Oracle VM2) for further information on accessing the console in text mode. Once the file is on node3 & node4 in /u01/racovm directory, run:

   On node3:   On node4:
   # **cd /u01/racovm**   # **cd /u01/racovm**
   # **./netconfig.sh -n3**   # **./netconfig.sh -n4**

The network is now setup on node3 & node4, go to step 7 to build the cluster.

6.2 On node1 stamp the network information (netconfig.ini) on to the still empty shared disk (which is shared amongst all VMs), as follows:

 # cd /u01/racovm
 # ./netconfig.sh -W   (Writes the network information from netconfig.ini to shared storage)
```
INFO (node:racnode1): Network configuration file netconfig.ini written to /dev/xvdc
Run: netconfig.sh -R -c /dev/xvdc (assuming identical device name on remote
nodes) to read the file on the remote nodes
```

By default /dev/xvdc is used but this can be overridden using netconfig.sh's -c flag.
On node3 and node4 read and apply the network configuration as follows:

On node3:	On node4:
# cd /u01/racovm	**# cd /u01/racovm**
# ./netconfig.sh -R -n3	**# ./netconfig.sh -R -n4**

If you wish, you may break the above command into two steps, first reading the netconfig.ini file, which allows you to inspect it for correctness, then apply the network configuration, as follows:

 # ./netconfig.sh -R   (Reads the network information from shared storage to netconfig.ini)
 # ./netconfig.sh -n3   (use -n4 on node4)

The network is now setup on node3 & node4. Proceed to build the cluster.

Network setup from Dom0/grub:

There is an easier way to configure all nodes at once, directly from Dom 0 as follows:

a) Instantiate the 4 VM's including adding the shared storage to all 4 VMs, do not boot them yet.

b) Prepare a netconfig.ini file with network information for all 4 nodes using any text editor.

c) From Dom0 stamp the network information (netconfig.ini) on to the still empty shared disk (which is shared amongst all VMs). Stamp the disk that will be visible to the VM as /dev/xvdc (using the -c flag, it is possible to use a any other shared disk). You can find netconfig.sh (and sample netconfig.ini) on Dom0 in the "utils" directory under /OVS/running_pool/<vmname>/**utils.**

3.1 If the shared disk is a file based disk it would need to be stamped via the loopback device, as follows **(Look in vm.cfg for the device mapped to xvdc)**:

Dom0 # **losetup -vf /OVS/sharedDisk/shared-diskname.img**
```
Loop device is /dev/loop3
```

Dom0 # **./netconfig.sh -W -c /dev/loop3**   (you may ignore the feedback from the command)

```
 INFO (node:dom0): Network configuration file netconfig.ini written to /dev/loop3
Run: netconfig.sh -R -c /dev/loop3 (assuming identical device name on remote
nodes) to read the file on the remote nodes
```

Dom0 # **losetup -d /dev/loop3**   **(remember to delete the loop device)**

3.2 If the shared disk is a physical disk on Dom0 that is passed to the VMs as 'phy:/dev/racdevc1' for example, then it should be stamped directly as follows:

Dom0 # **./netconfig.sh -W -c /dev/racdevc1**   (you may ignore the feedback from the command)
```
 INFO (node:dom0): Network configuration file netconfig.ini written to
/dev/racdevc1
 Run: netconfig.sh -R -c /dev/racdevc1 (assuming identical device name on remote
nodes) to read the file on the remote nodes
```

d) Boot the VMs using the following command, which will append the arguments in "extra" to the kernel boot line ("extra" method work for PVM VMs only). These arguments will be processed by the template first-boot module, which will bypass the normal interview and directly read the network information from the shared disk and setup the network on that VM (based on the node number passed in the **-n** flag):

Dom0 # **cd /OVS/running_pool/<name-of-VM1>**
Dom0 # **xm create -c vm.cfg extra="console=hvc0 template-reconfig-args='-R -n1'"**

For VM's 2-4 just change the number following the -n flag, e.g. -n2 for VM number 2.

Or, if you prefer to do it manually on the grub menu, boot each VM and at the grub menu (first screen), quickly type **e** (for "edit"), you will see this screen:
```
pyGRUB  version 0.6
root (hd0,0)
kernel /vmlinuz-2.6.18-238.0.0.0.1.el5xen ro root=LABEL=/ console=hvc0
template-reconfig-args='-R -n1'
initrd /initrd-2.6.18-238.0.0.0.1.el5xen.img
```

Scroll to second line (**kernel** boot line), type **e** (for "edit") and add a space followed by `console=hvc0 template-reconfig-args='-R -n1'` to the end of that line. Press **enter** then **b** (to boot). The VM will boot and read its network configuration from the shared disk then configure itself as node1.

Follow the same procedure for node2, node3 and node4, use -n2, -n3 and -n4 respectively.
Note: Use **single** quotes (') to enclose the arguments '-R -n1'; do not use double quotes. The grub example above wraps the kernel line due to page layout, in real life it would all appear on one line.

If you attached the same shared disks to all 4 VMs and stamped the right disk (/dev/xvdc) with the netconfig.ini file, then all 4 VMs have their network fully configured.

Fully automated & scripted cluster creation:

It is possible to add the '**-b**' flag to any one of the above VMs (e.g**.** template-reconfig-args='-R -n1 **-b'**), which will cause the buildcluster.sh to be called when network is setup on all VMs. If you didn't supply the '**-b**' flag then proceed to build the cluster in the section below.

Build the cluster:

From any node (as root) run: **# /u01/racovm/buildcluster.sh**

This will build a 4-node cluster and a 4 instance Oracle RAC database.

APPENDIX F: Frequently Asked Questions / Troubleshooting

1) Why is the 2-node interview unavailable due to disk or NIC mis-configuration?

During the 2-node console based interview:

```
== Oracle DB/RAC [Version] OneCommand for Oracle VM - template configuration (x86_64) ==
  NODE SELECTION
  ==============
  Is this the first node in the cluster  (YES/NO): [YES]
```

You see this error:

```
ERROR – SHARED DISKS INCORRECTLY CONFIGURED
For the Oracle RAC OneCommand on Oracle VM install to work,
a minimum of 5 shared disks must be configured with the following
names: /dev/xvdc /dev/xvdd /dev/xvde /dev/xvdf /dev/xvdg.

Power OFF the VM & configure shared disks correctly using Oracle VM Manager,
OVM CLI, or manually editing the vm.cfg configuration file.
NOTE: It is also possible to edit params.ini and reduce the setting of ASM_MIN_DISKS

Refer to included PDF file for more details.
```

This means that the shared disks were not attached correctly to that VM. Make sure the vm.cfg has the shared disks listed on both VMs and the shared disks are presented to the VMs as /dev/xvdc,d,e,f,g. The automated install relies on these exact names in order to work correctly. Advanced users should refer to Appendix C if they wish to deviate from this naming convention.

Or this error:

```
ERROR - NETWORK CARDS INCORRECTLY CONFIGURED
For the Oracle RAC OneCommand on Oracle VM install to work,
a minimum of 2 NICS must be configured as eth0 & eth1 (xenbr0 & xenbr1).
This should appear in the vm.cfg config file

You may add the NICs dynamically now via Oracle VM Manager,
OVM CLI, or Power OFF the VM and manually edit the vm.cfg configuration file.

Refer to included PDF file for more details.

Press ANY key to recheck  (Last check: <date>)
```

This means that the network cards were incorrectly configured. The VM requires two NICs, named eth0 connected to xenbr0 and eth1 connected to xebbr1. You may hotplug a NIC and hit enter to recheck, or you may power off and add the NICs.

2) Is it possible to clean the environment and start all over (with different settings)?

YES! If you experiment with the build process by editing **/u01/racovm/params.ini** and incorrectly entered data causes the build to fail or you stop the build before it finishes, you may clean the

environment in preparation for another build. Since this operation also clears **the shared disks** (if executed on first (Hub) node) extreme caution should be taken when running clean. If cleanup only on local node is needed, exchange "clean" with "cleanlocal":

/u01/racovm/racovm.sh -S clean

Note that by default, as a safeguard to prevent accidental removal of the environment, the above command will **fail** if there are any running processes on a VM. To bypass that, change CLONE_CLEAN_CONFIRM_WHEN_RUNNING to **no** in params.ini, then issue "./racovm.sh -S copykit", (will copy kit files+params.ini to all nodes) now you can clean all VMs despite running processes. So a typical cleanup on a clustered system may look like this:

/u01/racovm/racovm.sh -S setsshroot,copykit,clean

To further cleanup the OS from any network/hostname, **see FAQ#6 in the [Deploycluster documentation](#)**, which involves running, in summary:
- o `/u01/racovm/racovm.sh –S setsshroot,copykit,clean`
- o `ovmd –s cleanup  (on all VMs)`
    - o `OL5/6: service dbracovm-template.init enable-all (on all VMs)`
    - o `OL7: /usr/bin/dbracovm-template.init enable-all  (on all VMs)`
- o `Poweroff all VMs`
- o `Run deploycluster.py to redeploy & configure the VM(s)`

3) **Why is an error returned when a remove node operation is attempted?**

Similar safeguards are built for other operations. For example, this removenodes request:

./racovm.sh -S removenodes –N racnode2

Will fail if processes are still running on the node(s) to be removed with the following messages:

```
WARNING (node:racnode2): Above Oracle processes are currently running!!!!

ERROR (node:racnode1): It is not possible to remove nodes from the cluster with
existing software running on them, shutdown the above processes and re-run
'removenodes'. If you are absolutely certain it is safe to clean the nodes before
removing them from this cluster, set CLONE_CLEAN_ON_REMNODES=yes and re-run the
'removenodes' operation.
```

Either shutdown these processes on racnode2, or change CLONE_CLEAN_ON_REMNODES to yes in params.ini (or set in the environment of the shell), then re-issue the command.

4) **Why is an error returned when an attempt to stamp a disk using netconfig.sh -W?**

If you try to stamp a disk with the network information, and the disk is busy or held by any process, an error similar to this one will be displayed:

[root@racnode1 racovm] # ./netconfig.sh -W
```
ERROR (node:racnode1): Device (/dev/xvdc) used to transfer network configuration to
remote node(s) is incorrect or busy (blockdev) (see output below):
BLKRRPART: Device or resource busy
```

This is a safeguard measure not to overwrite possible user data. The stamping of shared storage should be used only when there is no user data on the disks. To transfer the netconfig.ini to the new node, try one of these options:

- Copy/paste the netconfig.ini using console in text mode, see APPENDIX D: Using Text Mode Console (Oracle VM2)) for further information on accessing the console in text mode.
- If there is an extra non-used disk shared by the added VMs and the current VM, you may stamp to that disk, simply add the '-c /dev/xvdx' assuming /dev/xvdx is that spare disk. Commands will be: ./netconfig.sh –c /dev/xvdx -W to write, and -R to read on the new node.

5) Is it possible to control resolver order and the values in /etc/hosts file?

YES! If the hosts are registered in DNS, the /etc/hosts file may be skipped by setting resolver order in /etc/nsswitch.conf. Change this line "hosts: files dns" to "hosts: dns files" thus making DNS be the first option for resolving names, then if "nscd" (by default it may be installed but not turned ON in the Oracle DB/RAC OVM templates) is used issue:

 # service nscd reload

In addition, it is possible to instruct netconfig.sh to **append** or completely **skip** writing into /etc/hosts, see netconfig.txt for details on NETCONFIG_APPEND_ETCHOSTS & NETCONFIG_SKIP_WRITING_ETCHOSTS.

6) Is it possible to relocate /tmp to a separate disk (or add an extra disk before deployment)?

YES! It is a best practice to relocate /tmp to a separate disk, so that an accidental fillup of /tmp doesn't also fillup the root (/) filesystem. This is not done in the templates to reduce footprint and allow easier /tmp setting at deployment time. It is extremely easy to do that using the *modifyjeos* tool (included in the *ovm-modify-jeos* rpm from http://edelivery.oracle.com/linux). When guest is shutdown issue:

 # modifyjeos -f System.img -P tmp.img 5000 /tmp
```
Starting VM image reconfiguration...
Processing tmp.img product image...
Mounting VM image...
Mounting product image tmp.img...
Persisting mount point /tmp ...
Unmount VM image...
Successfully created JeOS image files...
```

The above command creates a 5GB tmp.img disk and will add the following line to /etc/fstab inside the guest, so that next time that guest boots it will use /tmp as a dedicated mount point on the newly created tmp.img disk.

```
LABEL=tmp      /tmp         ext3    defaults        1 2
```

7) Is it possible to connect to a specific instance in RAC?

YES! Since 11.2 NETCA no longer writes individual instance's connect data. If for some reason you need to connect to a specific instance in RAC, use EZConnect as follows:
```
[oracle@racnode1 ~]$ sqlplus system/oracle@racnode1:1521/ORCL/ORCL1
SQL*Plus: Release 11.2.0.3.0 Production on Thu Aug 26 01:34:35 2011
Copyright (c) 1982, 2011, Oracle.  All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 – Production
```

```
With the Partitioning, Real Application Clusters, Automatic Storage
Management, OLAP, Data Mining and Real Application Testing options
```

Note the node number followed by port, database and instance names.

8) Is it possible to create/delete and operate on multiple Databases?

YES! By default, OneCommand (**racovm.sh**) operates on the database defined in DBNAME &
SIDNAME in params.ini, if you want to create/operate on other databases, you may change DBNAME
& SIDNAME in params.ini or comment them in the file, and specify on the command line, e.g.:

[root@racnode1 racovm] # **DBNAME=MYORCL ./racovm.sh -sS createdb**

The above command will create a new database called MYORCL (SIDNAME will default to
MYORCL, e.g. MYORCL1, MYORCL2, etc.)

9) Is it possible to create a Policy Managed Database?

By default the templates create an Aministrator Managed Database. **Starting with the DB/RAC
Template released in 2013, Policy Managed database automation is supported via
DBCA_DATABASE_POLICY in params.ini**.

For reference, it takes two commands to manually convert an Admin Managed to Policy Managed, as
follows:
 [oracle@racnode1 bin]$ **/u01/app/11.2.0/grid/bin/srvctl config db -d orcl**
Database unique name: ORCL
Database name: ORCL
Oracle home: /u01/app/oracle/product/11.2.0/dbhome_1
Oracle user: oracle
  <snip>
Management policy: AUTOMATIC
Server pools: ORCL
Database instances: ORCL1,ORCL2
 <snip>
Type: RAC
Database is administrator managed

[oracle@racnode1 bin]$ **/u01/app/11.2.0/grid/bin/srvctl add srvpool -g prod01 -l 2 -u 2**
[oracle@racnode1 bin]$ **/u01/app/11.2.0/grid/bin/srvctl status srvpool**
Server pool name: Free
Active servers count: 0
Server pool name: Generic
Active servers count: 2
Server pool name: prod01
Active servers count: 0
[oracle@racnode1 bin]$ **/u01/app/11.2.0/grid/bin/srvctl modify db -d orcl -g prod01**
[oracle@racnode1 bin]$ **/u01/app/11.2.0/grid/bin/srvctl config db -d orcl**
Database unique name: ORCL
Database name: ORCL
Oracle home: /u01/app/oracle/product/11.2.0/dbhome_1

Oracle user: oracle
 <snip>
Management policy: AUTOMATIC
Server pools: prod01
Database instances:
 <snip>
Type: RAC
Database is policy managed

10) **It it possible to "mix and match" the OS disk from one DB/RAC OVM template to another?**

YES! For example, if we assume that:

Oracle 11.2.0.4.4 was released with OS disk of Oracle Linux 5 Update 11
Oracle 12.1.0.2.1 was released with OS disk of Oracle Linux 6 Update 6

If there is a need to have 11.2.0.4.4 run with OL6U6 then in:
- Oracle VM 3: **Unzip and untar then import each disk individually** then create a VM from the imported disks. If this change is needed after the software is installed, then follow a 'removenodes' followed by 'addnodes' procedure to add the new node(s) with the modified OS version. It is not recommended to run with mixed OS versions on a cluster for prolonged periods of time; ensure that all cluster nodes are upgraded to the same OS level as described in My Oracle Support Note# 220970.1.
- Oracle VM2: Extract only the OS disk ("System.img") from the 12.1.0.2.1 released template (the OS disk is inside the smaller zip file) and unzip it into the same folder as the Oracle disk of the 11.2.0.4.4 release.

11) **Can oneoff patches be added or removed before and/or after the template is fully built?**
YES! Patches can be added or removed at any time as follows:

After the cluster is fully built: As any patch is applied or rolled back; follow the README supplied with the oneoff patch, e.g. shutdown any databases, services, etc. apply or rollback the patch using **opatch** and restart services, etc.

After the template's first boot, but before the cluster is built: The templates are delivered as 'software-only' installs, without any central inventory or configuration. In order to apply or rollback patches in this configuration follow these steps, only one node is needed:
- Make sure network is fully configured on a single node (to be patched) using the Deploycluster tool or by one of these manual methods. At the boot prompt 'Is this the first node in the cluster' type OUT, then edit **/u01/racovm/netconfig.ini** and run **/u01/racovm/netconfig.sh –n1** as needed to configure the network (described in prior sections of this document). Alternatively, setup 2 VMs using normal screen interview and only use one of the VMs.
- Attach both the Grid home (skip this step for Single Instance) and the RAC home to the central inventory by running:

[root@racnode1 racovm] **# /u01/racovm/racovm.sh -sS giattachlocal,racattachlocal**

- Switch to the oracle user and apply or rollback the patches as they normally would be using **opatch**. Follow the README supplied with the oneoff patch skipping any instructions to shutdown databases or services, since none are yet defined.
- Once all the patches are applied or rolled back correctly, clean up the Oracle disk using:

  [root@racnode1 racovm] **# /u01/racovm/racovm.sh -sS cleanlocal**

- Shutdown the VM, and use only the Oracle disk, the OS disk was modified (network configured, etc.) and can therefor be discarded. Combine the Oracle disk with an unmodified OS disk, and that becomes the new template 'golden image', which is used to create as many VMs as needed.

> Tip: To apply multiple fixes using opatch use the 'napply' option, to rollback multiple fixes use the 'nrollback' option.

To remove ALL patches: If for some reason you would like to remove ALL oneoff patches and revert back to base release of the template. Then apply your own selection of patches on top of the base release, run these commands which will remove ALL patches from the given Oracle Home (For example, will revert an 11.2.0.4.4 to 11.2.0.4.0). This is of course not recommended; unless you will apply newer patches that better fit your needs keeping in mind security, performance and other aspects.

```
$ opatch lsinventory > invlist
$ # Following will create a long comma separated list
$ patches="$(grep –P 'Patch\s+\d+' invlist |awk '{ printf $2"," }')"
$ patches="${patches%,*}"  # Remove last comma
$ # remove the -local flag to remove the patches from ALL nodes
$ # add -silent to skip confirmation prompts
$ opatch nrollback -id $patches -local
```

If running this on a configured Grid Home, before this block of commands run the following as **root** user to unlock the home and allow the Oracle user to write there:

<Grid_HOME>/crs/install/rootcrs.pl –unlock

After the block of commands run the following to activate the stack, as the **root** user:

<Grid_HOME>/crs/install/rootcrs.pl -patch

12) **Is it possible to change the user passwords on all nodes?**

As long as passwordless-ssh is configured one can use **doall.sh** to run commands on all nodes, e.g:

 # /u01/racovm/doall.sh "echo 'hArDpasSWd' | passwd --stdin root"
 # history -c; clear

Will change root password on all nodes; run history command clears the command from history buffer. Note, above command will display the password for a brief time in "ps" output. You can run this command from any other user account; or as root to other users by changing 'root' to the desired user. If

passwordless-ssh is not configured you may add "-sp" as first argument to **doall.sh** this will attempt to setup passwordless-ssh with the password stored in params.ini or if missing, will prompt for a password. If that does not work, you can always ssh to each node, and run *passwd* to change the password manually.

13) **Do the templates support RAC One Node deployment?**

YES! Simply follow the steps in Note# 1232802.1 on My Oracle Support paying attention to the troubleshooting section at the bottom of the article. Assuming a two node cluster with a database instance running on racnode1; following these commands will create a service ("SERV") and convert that database to be of RAC One Node type:

[oracle@ racnode1 bin]$ **srvctl add service -d ORCL -s SERV -r ORCL1**
[oracle@ racnode1 bin]$ **srvctl convert database -d ORCL -c RACONENODE -i ORCL**
[oracle@ racnode1 bin]$ **srvctl config database -d ORCL**
Database unique name: ORCL
Database name: ORCL
Oracle home: /u01/app/oracle/product/11.2.0/dbhome_1
Oracle user: oracle
<snip>
Services: SERV
Type: RACOneNode
Online relocation timeout: 30
Instance name prefix: ORCL
Candidate servers: racnode1
Database is administrator managed

Status of instances:

[oracle@ racnode1 bin]$ **srvctl status database -d orcl**
Instance ORCL1 is running on node racnode1
Online relocation: INACTIVE

To manually initiate instance relocation:

[oracle@ racnode1 bin]$ **srvctl relocate database -d orcl -n racnode2**

See Note# 1232802.1 and official documentation for additional details.

14) **Is it possible to automatically reduce the OS footprint?**

YES! The DB/RAC OVM Templates released in 2013 include a utility called **OSRevert** which can easily help transition the OS from one saved rpm Target to another.

```
# DESCRIPTION:
#   OSRevert allows Saving (-S flag) a certain set of RPMs (Target)
#   so that at a future point in time, the system could be
#   Reverted (-R flag) to that target.
#   By default the Revert operation only removes RPMs that are present
#   in the running system and not in the Target. However, if yum
#   repository access is configured properly, then automated
#   addition (-a flag) of RPMs can also be performed.
```

It it possible to include NETCONFIG_OSREVERT_TARGET=db-minimum, in netconfig.ini so that during network setup, before a buildcluster or buildsingle is initiated, the OS will revert to the 'db-minimum' saved target, thus making the OS footprint smaller and more secure. This operation requires no network access since rpms are being removed. At at later time, if network access is configured, it is possible to revert the OS back to the 'basic' saved target, using the following command:

/usr/bin/OSRevert.sh -T basic -R –a

Templates released in 2014 and beyong also include an OS Revert Target called 'os-minimum' which is bare-minimum OS. Reverting to it will not allow installing and running the Database or Clusterware software. Use 'db-minimum' or above to run the Database or Clusterware.

15) **Are the DB/RAC Templates available as Standard Edition (SE)?**

In October 2015 a DB/RAC Template with Oracle Database 12.1.0.2.5 Standard Edition 2 (SE2) was released in addition to a separate 12.1.0.2.5 Enterprise Edition (EE) template.

Keep in mind that regardless of the DB/RAC Templates, transitioning from one edition to another (SE to EE or vice versa) requires a re-install.

As such, please select the correct edition of the DB/RAC Templates to deploy as appropriate for your licnese and testing needs. Although once deployed, the environment can be cloned, configured and repurposed as Single Instance or Oracle RAC many times (see FAQs below), still, in all this life-cycle, the edition remains as the original selection.

16) **What deployment modes do the DB/RAC Templates support?**

There are 3 main deployment models:
- **Single Instance** (CLONE_SINGLEINSTANCE=yes in netconfig.ini or params.ini)
- **Single Instance/HA (Oracle Restart)** (CLONE_SINGLEINSTANCE_HA=yes in netconfig.ini or params.ini)
- **Oracle Real Application Clusters (RAC)**

The SIHA (Oracle Restart) & Oracle RAC allow for a **Clusterware Only** deployment, where by no Database is created, only the Clusterware is setup using the Grid Home and configured on the selected nodes. Simply set CLONE_CLUSTERWARE_ONLY=yes in params.ini, no database will be created and in fact the DB Home will not be used at all (see following question).

17) **Which Oracle Home may be removed, based on deployment mode?**

The DB/RAC Templates include both Grid Home (/u01/app/12.1.0/grid) and DB Home (/u01/app/oracle/product/12.1.0/dbhome_1) [In 11gR2 "12.1" changes to "11.2" in the path].

In **Single Instance** deployment (CLONE_SINGLEINSTANCE=yes) it is allowed to completely remove the Grid Home, before or after deployment.

In **Clusterware Only** deployment (CLONE_CLUSTERWARE_ONLY=yes), it is allowed to completely remove the DB Home, before or after deployment since a database is not created.

The **RAC & Single Instance/HA** (in non-Clusterware only) deployments require both Oracle homes to exist.

18) Is it possible to switch from one mode to another, e.g. from Single Instance to RAC or vice versa?

YES! The exact same templates are used for all modes of operation. To switch from one to another, simply follow the cleanup steps (FAQ#2), be sure to run the cleanup on the already configured mode, and only after the cleanup completed modify to the new mode. Also, make sure the pre-requisites for the new mode are met, e.g. RAC needs a second NIC, and shared storage, then re-run the buildcluster.sh or buildsingle.sh.

It is also possible to switch from one mode to another without a cleanup, by following the standard documentation on such conversions, for example, DBCA is able to convert a Single Instance database to RAC, assuming the environment is setup right.

19) Is it possible to place the Database on a (shared) filesystem?

YES! For **Single Instance** deployment, the database **must** reside on a filesystem since ASM is not configured. For **Single Instance/HA** or **RAC** deployments when ASM is configured (by default), it is required to set DATABASE_ON_FS=YES in order to create the database on a filesystem. The filesystem location is specified in FS_DATAFILE_LOCATION. Oracle RAC 12c supports placement of the database inside an ACFS filesystem and the template automation supports that as well. To clean this location, during a 'clean' operation, set CLONE_CLEAN_FS_LOCATIONS=yes in params.ini, otherwise the database files will be left around even after a clean operation (this is a designed safeguard to prevent accidental database removal).

20) Is it required to configure ASM?

Although it is **highly** recommended to configure ASM and place the database files inside it, it is not required. In the DB/RAC Template automation, if the RACASMGROUPNAME is set to empty string, ASM will not be configured. It is then required to place the OCR & Voting disks on a supported shared filesystem in the case of RAC deployment. Manually setup NFS or OCFS2 then set CLONE_OCR_DISKS & CLONE_VOTING_DISKS to 3 files each as well as CLONE_OCRVOTE_IN_ASM =no (see params.ini for details).
Note that **Single Instance/HA** requires no OCR or Voting disks, and **Single Instance** deployment never configures an ASM instance.
Carefully review the mount options for the OCR/Voting filesystem as well as the filesystem that will hold the database files, as per current product documentation.

21) Is ASMLib supported in the DB/RAC OVM Templates?

YES! Since initial release of the templates ASMLib is a valid path to the ASM disks, however, the creation of the ASMLib disks is not automated. Follow normal ASMLib documentation to install the necessary two rpms (newer templates have the kernel driver under /u01/Extras), configure the ASMLib driver using "/etc/init.d/oracleasm configure". Create some disks, e.g.

/etc/init.d/oracleasm createdisk xvdf /dev/xvdf1
/etc/init.d/oracleasm listdisks
XVDF

Specify the newly created disk(s) in params.ini using:

ALLDISKS_ASMLIB="ORCL:XVDF"

Both ORCL:name, or just "name" of disk are honored. Add to the discovery string, e.g.:

RACASMDISKSTRING="/dev/xvd[c-e]1 ORCL:XV*"

Here a glob starting with "ORCL:" means ASMLib disks. Finally, make sure any reference to that disk (/dev/xvdf) in ALLDISKS is removed.

Now a standard **buildcluster.sh** or **buildsingle.sh** (In SIHA mode) will use these ASMLib disks

NOTE: Templates released in 2014 and after support ASMLib disk names in the ALLDISKS and all other disk variables (see params.ini and FAQ#24), ALLDISKS_ASMLIB retained for backwards compatibility.

22) Can the DB/RAC OVM Templates be used in HVM mode?

YES! Since UEK/UEK2/UEK3 kernels all support switching from HVM to PVM seemlessly, simply use Oracle VM Manager and change the virtualization mode (Domain type), then commands like 'lspci' and 'dmidecode' will function as they do in HVM mode. Refer to UEK release notes for details on HVM mode and refer to latest certification details (e.g. Note# 464754.1) on My Oracle Support for support information relating to RAC & HVM mode.

23) Can a Container Database be created?

YES! In the DB/RAC Templates released in 2013 and beyond, set DBCA_CONTAINER_DB =yes in params.ini and run 'createdb' step to create a database: racovm.sh -S createdb (see params.ini for additional details)

24) Can additional ASM Diskgroups be created?

YES! In templates released before 2014 that was a manual task, however, in templates released in 2014 and beyond, that task is fully automated and integrated with Deploycluster. The initial ASM diskgroup is still created using the ALLDISKS disk variable (it may also include ASMLib disks), all disks specified there will create the initial ASM diskgroup.

The name of a Recovery diskgroup may be specified in the variable RACASMGROUPNAME_RECO. If this diskgroup is specified then database creation will automatically use it as the Recovery diskgroup for the database.

The **redundancy**, **attributes** and **disks** are specified in **RACASM_RECO_REDUNDANCY**, **RACASM_RECO_ATTRIBUTES** and **RACASM_RECO_DISKS** respectively.

Valid values for redundancy are EXTERNAL, NORMAL (default) or HIGH.

Diskgroup **attribute** syntax must use single quotes around attribute names and values, separated by an equal sign, e.g. 'au_size'='2M', separate each such pair by comma for additional attributes (entire string is enclosed in double quotes). For example:

RACASM_RECO_ATTRIBUTES="'compatible.asm'='12.1.0.0.0', 'compatible.rdbms'='12.1.0.0.0'"

Disk names may be physical disks or ASM Lib disk names.

Additional extra ASM Diskgroups may be specified in **RACASMGROUPNAME_EXTRA**, separate each name by a space.

The **redundancy**, **attributes** and **disks** are specified in **RACASM_x_REDUNDANCY**, **RACASM_x_ATTRIBUTES** and **RACASM_x_DISKS** respectively (**x** represents the number of the diskgroup starting with 1). For example:
RACASMGROUPNAME_EXTRA='TEST MYDEV' will attempt to create 2 extra diskgroups, provided that RACASM_1_* and RACASM_2_* are set correctly (at least RACASM_*_DISKS must be specified, the rest have defaults).

The following variables allow various components to use the created diskgroups:
  ACFS_DISKGROUP=""   # Diskgroup for ACFS filesystem
  DBCA_DISKGROUP=""   # Diskgroup for Database
  MGMTDB_DISKGROUP=""   # Diskgroup for Management DB (12c only)

It is possible to set CLONE_DISKGROUP_CREATE_BG=yes and thus allow for these diskgroups to be created in the background while the rest of the installation continues onwards.

See params.ini for additional details.

25) Can the Database Controlfiles and REDO logs be multiplexed?

YES! Follow the generic Oracle documentation in the [Database Administrator's Guide on setting db_create_online_log_dest_n](#) and set the following in params.ini:

DBCA_INITORA_PARAMETERS="db_create_online_log_dest_1='+EXTRA',db_create_online_log_dest_2='+DATA'"

Will create multiplexed Redo logs and Controlfiles into EXTRA & DATA diskgroups assuming these diskgroups are created properly (see FAQ#24 on Diskgroup creation)

26) Can REDO logs size be set?

YES! All templates released in 2014 and beyond allow setting a specific size for the REDO log via **REDOLOG_FILESIZE_MB** in params.ini.

27) What does Yum@Deploy mean? How to add / update RPMs during or after deploy time?

YES!   In addition to the OSRevert functionality described in FAQ#14, templates released in 2014 and beyond offer a feature called Yum@Deploy. This option allows to automatically run YUM (Install and/or Update) during firstboot for initial deployment, as well as subsequent deployments and day to day usage. They are also immediately available to nodes being added using the 'addnodes' step.

NETCONFIG_YUM_INSTALL : List of rpms to install, can contain wildcards, e.g. x11*

NETCONFIG_YUM_UPDATE   : YES/NO, to enable yum update to bring up existing rpms to latest
                         revision
NETCONFIG_YUM_PROXY   : Proxy used for all Yum operations, e.g. http://...
NETCONFIG_YUM_OPTIONS : Various options passed directly to Yum (no validation)
                         e.g. --nodeps --skip-broken, etc

If NETCONFIG_YUM_INSTALL is set, all listed rpms will be installed, following that, if
NETCONFIG_YUM_UPDATE is set to YES or TRUE, a full 'yum update' will run to bring up all
installed rpms to latest version. These steps may be run directly at any time using the -S flag to run an
individual step using the following step names:

  yuminstall[local]*- add extra RPMs before configuring the Oracle software (all or local node)
  yumupdate[local]*- update all RPMs before configuring the Oracle software (all or local node)

28) How are process and kernel limits set in the templates?

By default the templates come preintalled with **oracle-validated** (OL5) or **oracle-rdbms-server-*-
preinstall** (OL6 & OL7). All the **kernel** and **user limits** are set by these rpms. If any changes are
needed, edit these rpms respective configuration files (/etc/sysconfig/<rpm-name>/<rpm-name>.param).
Then re-run their 'verify' script (see documentation with these rpms) to apply the changes.

If these rpms are removed or not installed for any reason, the templates have some default/fallback
values, and also allow overriding these by setting in params.ini: CLONE_<soft|hard>_<limitname>
where limitname can be: nofile, nproc, stack, core, memlock.
If the **oracle-validated** / **preinstall** rpms are installed, then above CLONE_* settings are ignored, please
use the rpm mechanism to adjust these limits.

Oracle Clusterware limits may be specified in: CRS_LIMIT_<LIMITNAME>, these are applied
during installation to <Grid_Home>/crs/install/s_crsconfig_<nodename>_env.txt.
Valid limit names (in upper case): NPROC, STACK, OPENFILE, MEMLOCK, CORE

See params.ini for additional details.

29) How to convert to BTRFS filesystem? Is it supported?

BTRFS is supported for production usage since March 2012. It has a very **rich set of features**!
Converting to btrfs is a quick and easy process, requiring only a few minutes and one command actually
converts the filesystem.

Here is the link to the official documentation:
https://docs.oracle.com/cd/E37670_01/E37355/html/ol_use_case7_btrfs.html

Below is a real-world example of converting the template's ext4 /u01 to btrfs as per above
documentation, which is also captured in the templates in this file: /u01/Extras/README_btrfs.txt:

[root@testc105 /]# **df -Th /u01**
Filesystem   Type Size Used Avail Use% Mounted on
/dev/xvdb1   ext4 45G 14G 29G 32% /u01

Unmount and run fsck:

[root@testc105 /]# **umount /u01**
[root@testc105 /]# **fsck.ext4 -f /dev/xvdb1**
e2fsck 1.43-WIP (20-Jun-2013)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/u01: 48144/2949120 files (0.2% non-contiguous), 3718263/11796224 blocks

Convert /dev/xvdb1 to btrfs:

[root@testc105 /]# **btrfs-convert /dev/xvdb1**
creating btrfs metadata.
creating ext2fs image file.
cleaning up system chunk.
conversion complete.

Mount the newly converted filesystem:

[root@testc105 /]# **mount /dev/xvdb1 /u01**
[root@testc105 /]# **df -Th /u01**
Filesystem   Type   Size Used Avail Use% Mounted on
/dev/xvdb1   btrfs 45G   15G   21G 41% /u01

It is also possible to mount the ORIGINAL filesystem in READONLY mode because the conversion
process automatically saves it as a subvolume called 'ext2_saved':

[root@testc105 /]# **btrfs subvolume list /u01**
ID 256 gen 11 top level 5 path ext2_saved

First mount the snapshot on a temporary mount point:

[root@testc105 /]# **mkdir /u01_tmp**
[root@testc105 /]# **mount -t btrfs -o subvol=ext2_saved /dev/xvdb1 /u01_tmp**
[root@testc105 /]# **df -Th /u01_tmp**
Filesystem   Type   Size Used Avail Use% Mounted on
/dev/xvdb1   btrfs 45G   15G   21G 41% /u01_tmp

This is the snapshot:

[root@testc105 /]# **ls -lsh /u01_tmp/**
total 15G
15G -r--------. 1 root root 45G Dec 31 1969 image

Now mount the original filesystem in read-only mode:

[root@testc105 /]# **mkdir /u01_orig**
[root@testc105 /]# **mount -t ext4 -o loop,ro /u01_orig/image /u01_orig**

[root@testc105 /]# **df -Th /u01_orig**
Filesystem   Type Size Used Avail Use% Mounted on
/u01_tmp/image ext4   45G   14G   29G 32% /u01_orig

To reclaim the space that the snapshot holds in the subvolume, delete it as follows:
[root@testc105 /]# **btrfs subvolume delete /u01/ext2_saved**
Delete subvolume '/u01/ext2_saved'

See the BTRFS documentation for more details.

30) Can the OCR/Voting disks be placed in a separate ASM Diskgroup?

YES. See FAQ#24 for additional information on creating multiple diskgroups during initial deployment.

Note that ALLDISKS should hold all disks only for the primary diskgroup (RACASMGROUPNAME) (defaults to 'DATA'). So, one could set the default diskgroup to be called OCRVOTE or 'SYSTEM', then create as many additional diskgroups as needed, using RACASMGROUPNAME_EXTRA, see FAQ#24 and params.ini for details and examples.