How to install Oracle 12c RAC on a VirtualBox

Roee Ebenstein

This guide is not a dumb “step by step”, but only nearly one. Some steps are skipped (for example – the “user configuration” after the first login to the newly installed Linux machines).
I assume the user is familiar with Linux, and a bit with Oracle, therefore not every step is explained.
The important stuff is explained in a “step by step” with screenshots.

Note: you cannot skip parts of the tutorial, you have to follow the text. The process is likely to fail if you won’t follow all that is written.
If I wrote something, it is because without doing it - the process would either fail, or would not behave as expected.

You will need:

Oracle VirtualBox, latest version, installed on your machine (I use version 4.3.30)
Oracle Unbreakable Linux installation DVD (version 7.1)
Oracle installation disks for 12c - Grid Infrastructure and Database

Your machine should have at least 12GB of RAM and at least 8 cores (more than 4).
Your processor has to support virtualization.
You will need about 185GB of free hard drive space.

You are expected to have basic understanding of linux (ssh, shell commands, yum, etc.) and Oracle DB (you don’t have to be a DBA, but you need to know how to use sqlplus).

You are expected to know how to operate vi, or to migrate whatever I write for vi to a different text editor. Although I usually write how to do stuff in vi, I do expect you to be able to use it without additional help.

Machine Creation

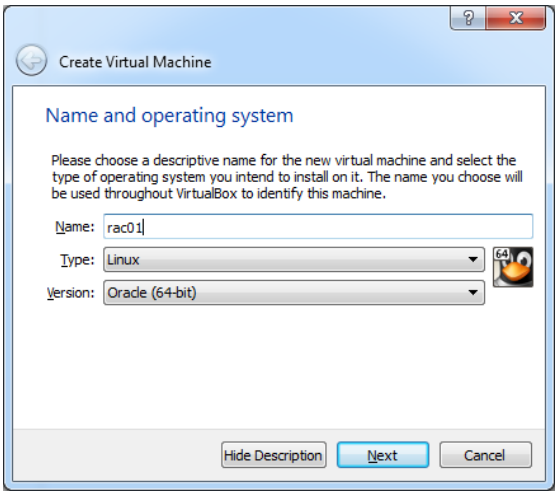
High level

We will create two machines, each has to have at least 4GB of RAM (we'll use 4.5GB), hard drive of 65GB (dynamic growing, so it won't use this area), at least 2 processors for each (we will use 4), and 3 network adapters each – public, private, and NAT.

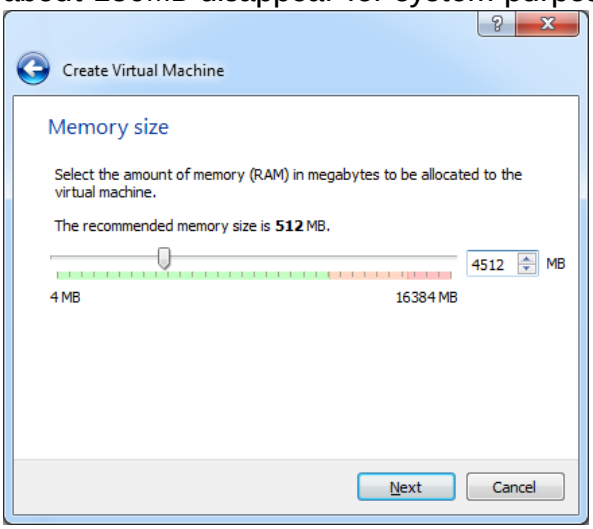
VirtualBox machines creation

Create two machines – one named rac01, and the other – rac02.
Notice that at some of the images the machine is named "rac1" – this is because I already created "rac01", and didn't want to rename it – yet you will use "rac01" and "rac02" – on all the demo fields – I show the correct values for you to fill.

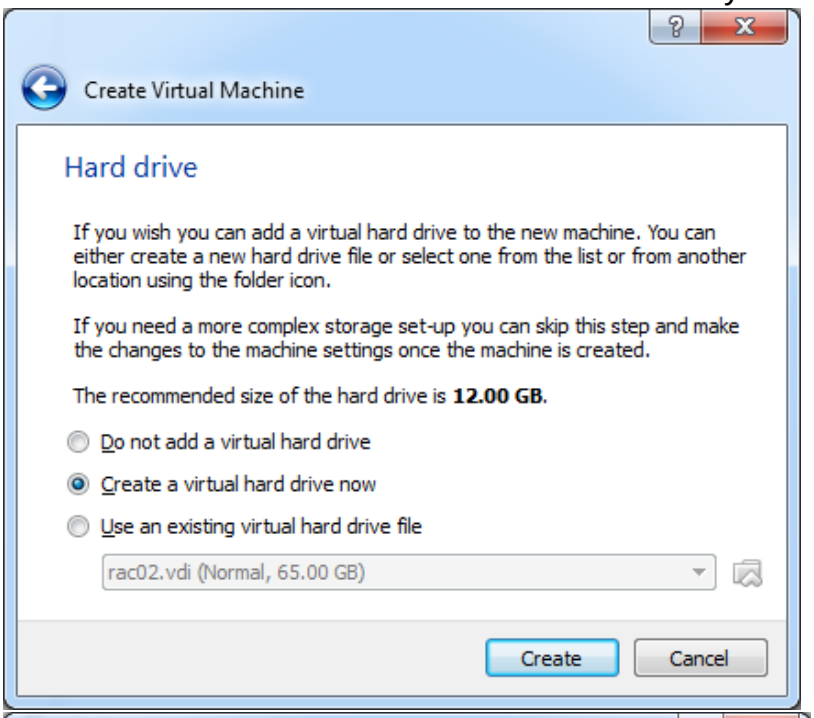
In VirtualBox, hit the "new" button, and create "rac01"

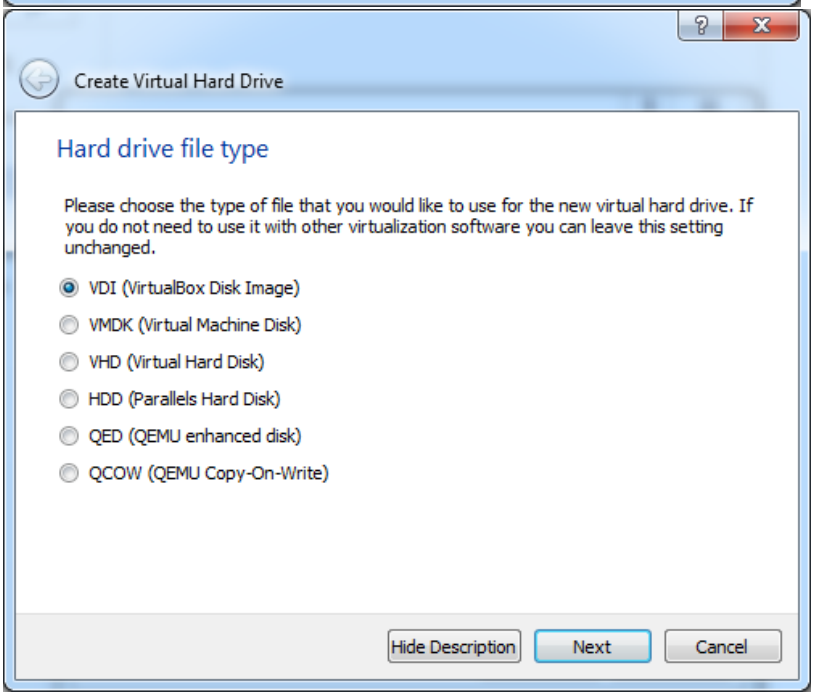


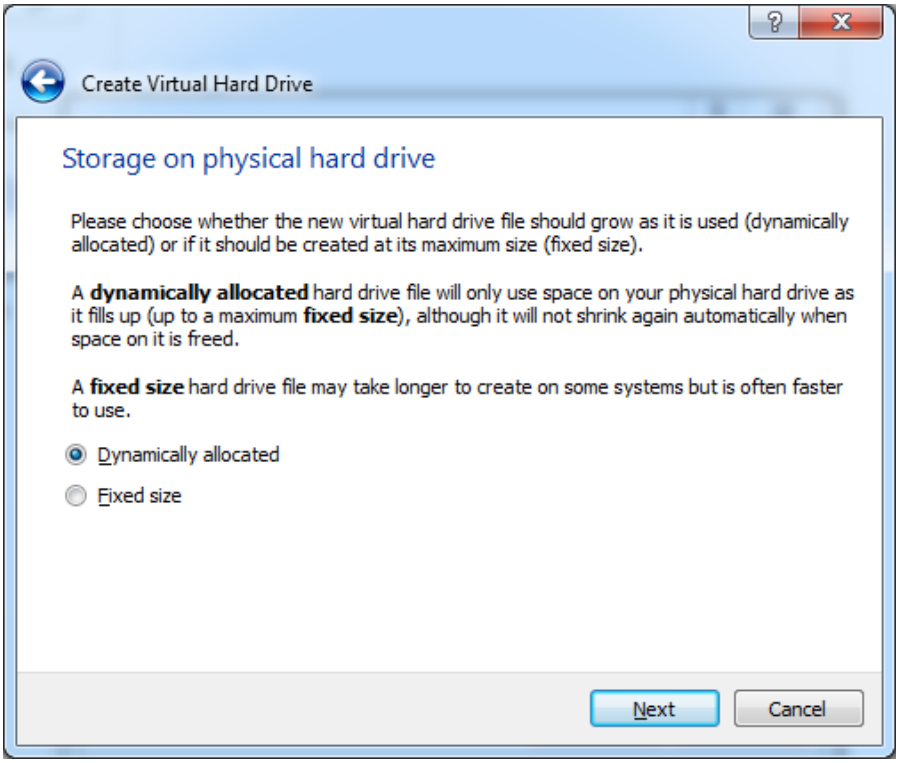
Assign 4512MB for memory (4096 is the minimum by Oracle, while after installation about 150MB disappear for system purposes – therefore the exaggeration …):



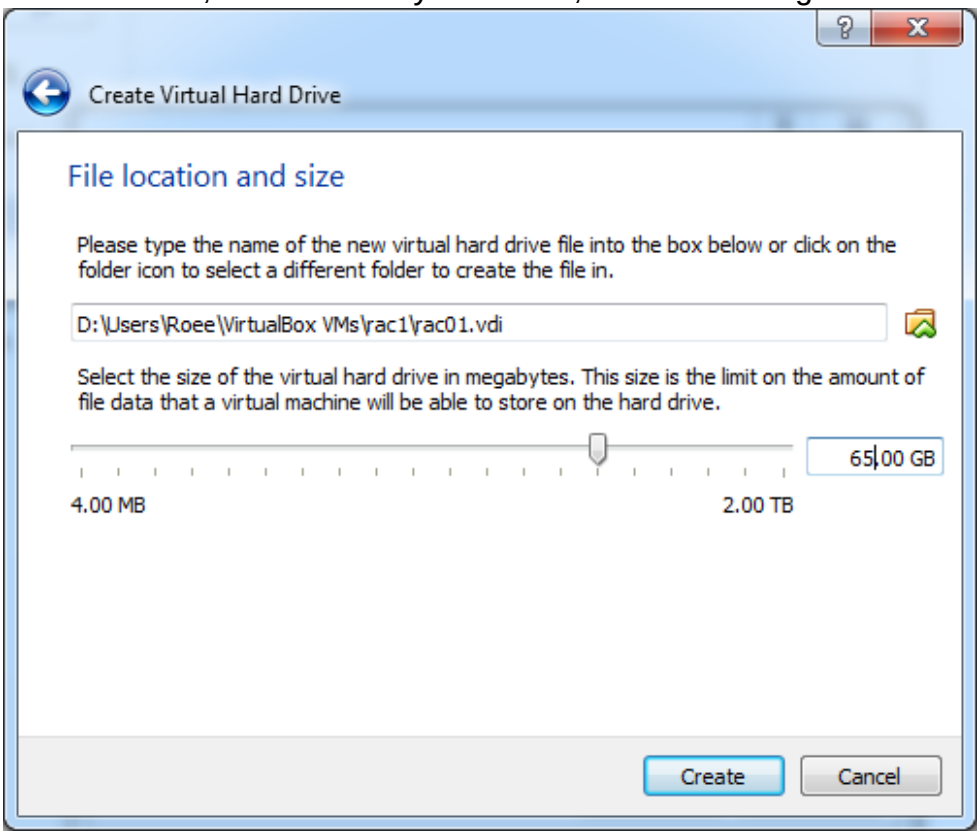
Create a Virtual disk of type VDI, increasing dynamically (which means – the virtual machine hard drive will be slow, we don't care about it -- the size of the VDI on the host will be as the inhabitant data within the VM – you should expect about 25-35GB):



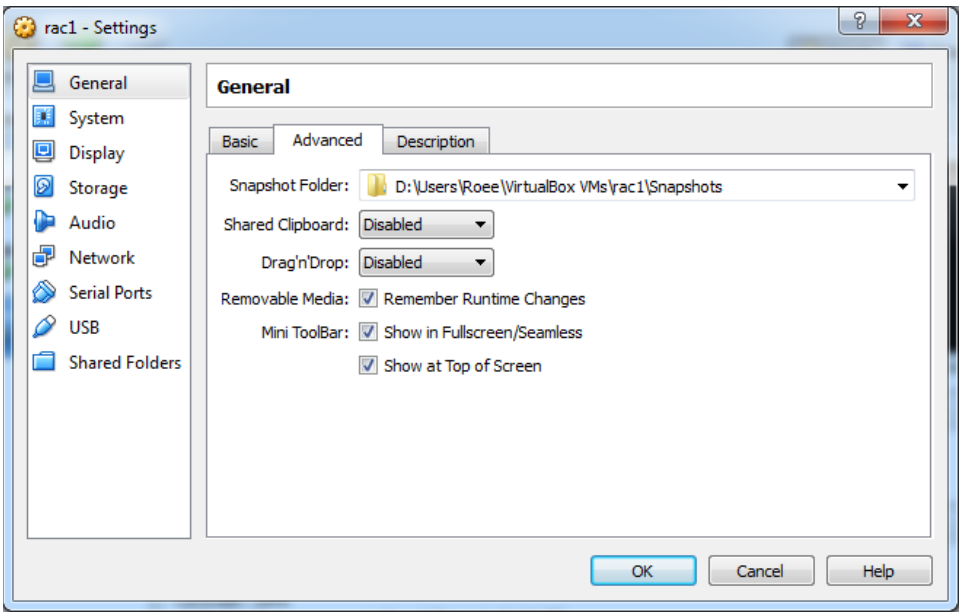




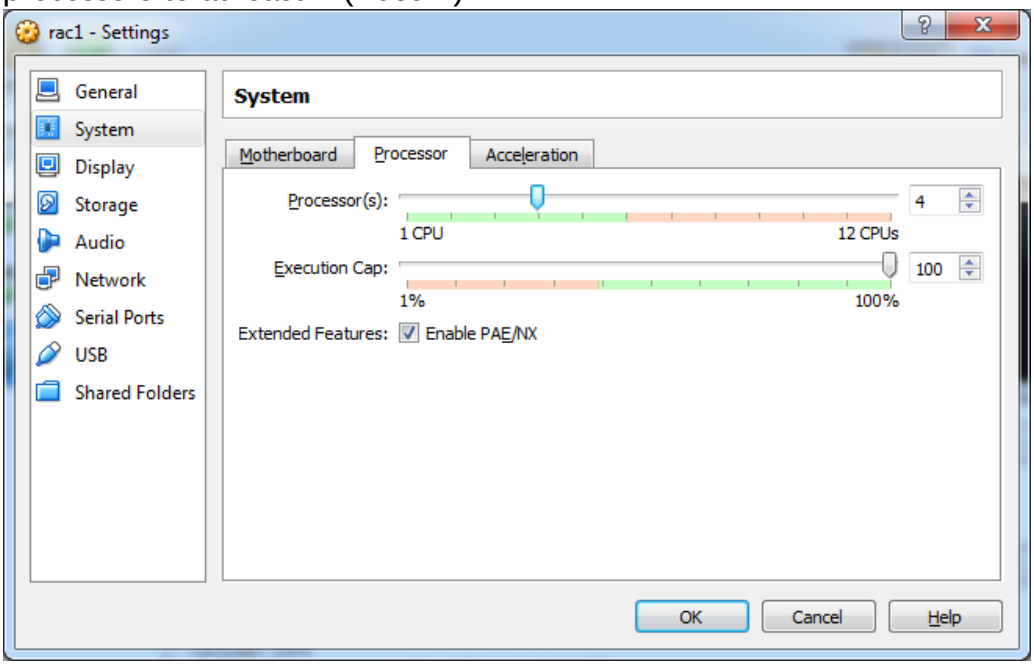
Create the file, notice where you store it, and do not forget to allocate 65GB for it:



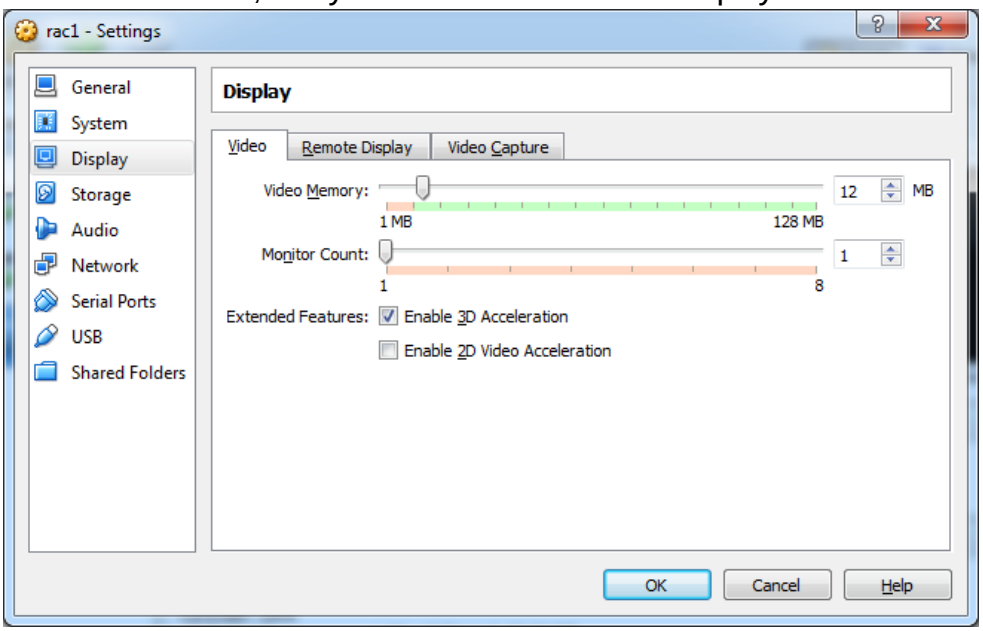
After the machine was created, right click on it, and hit “settings”.
In the “Advanced” tab (of the defaulted menu, “General”) checkmark “Show at Top of Screen”:



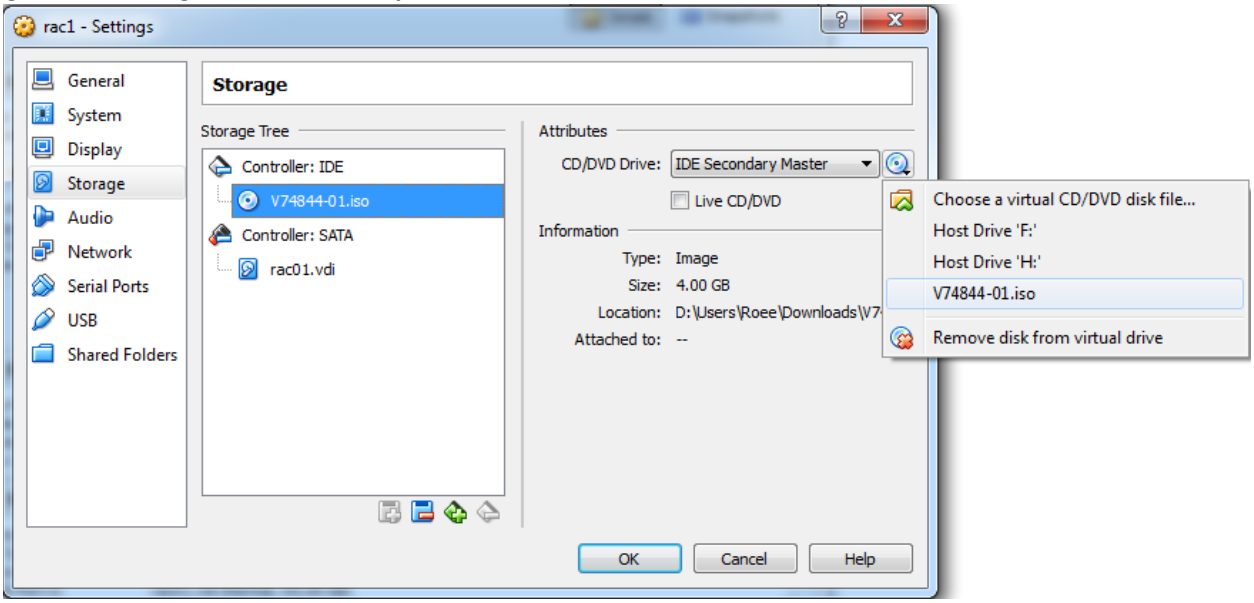
On the “System” menu, under “Processor” increase the number of the chosen processors to at least 2 (I use 4):



You don't have to, but you can activate under "Display" the 3D acceleration:

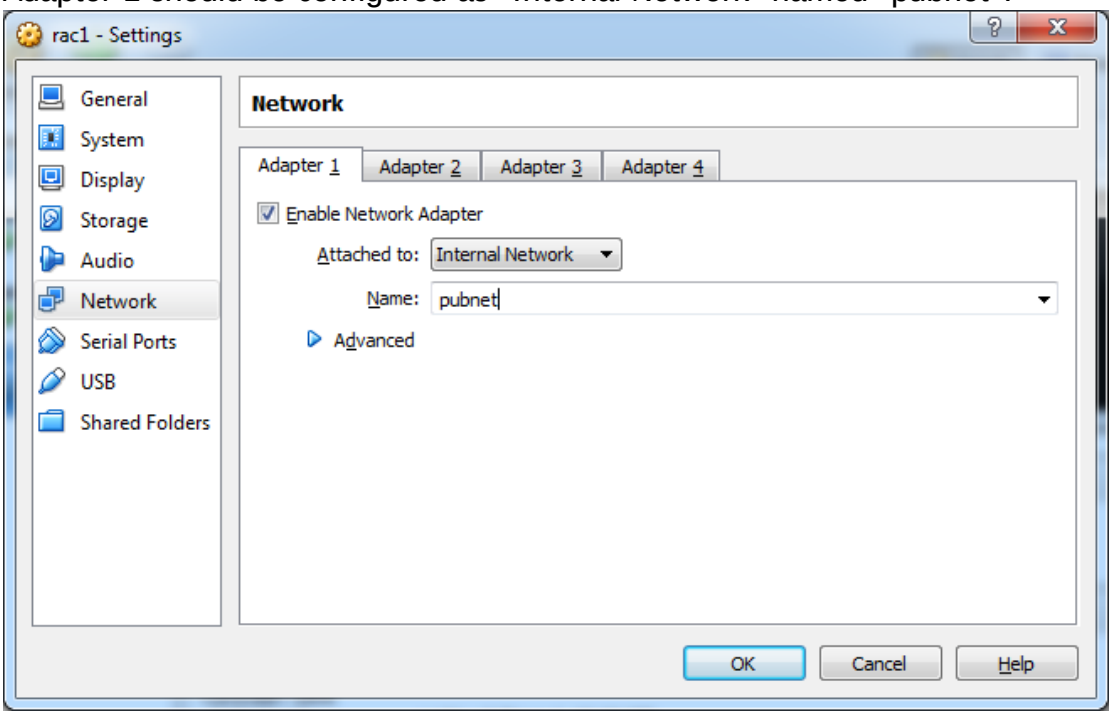


On the "Storage" configuration, select the CD-Rom icon, and on its right hit the icon for selecting an ISO. Select the ISO that contains the Oracle Unbreakable Linux (7/7.1) installation – the name of the ISO should be V74844-01.iso (though by the time you'll get it – it might be different):

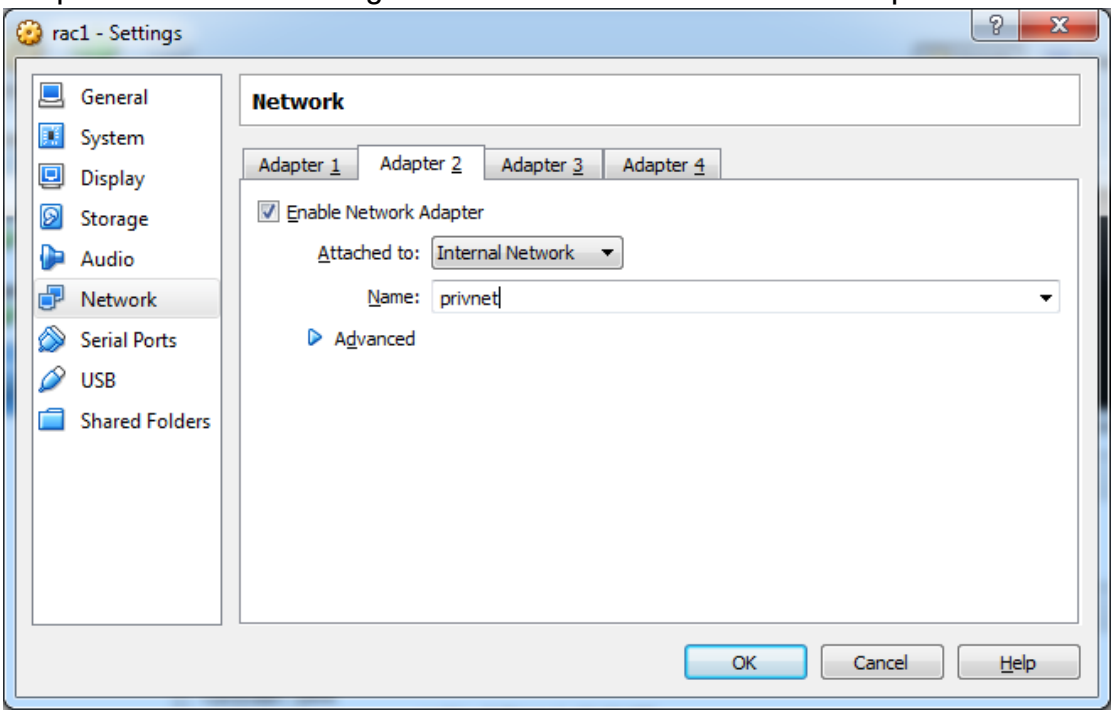


This is very important:
Under Network, create 3 devices – the naming of the networks is not important as long as it is consistent among all machines:

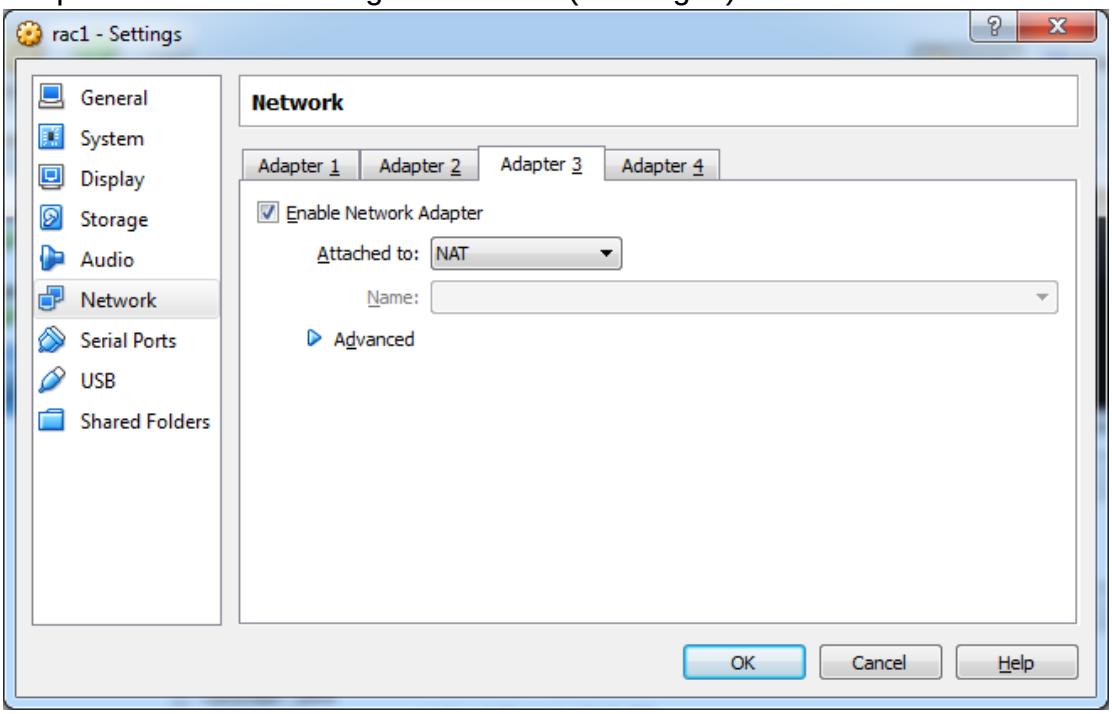
Adapter 1 should be configured as “Internal Network” named “pubnet”:



Adapter 2 should be configured as “Internal Network” named “privnet”:



Adapter 3 should be configured as NAT (or bridged):



Hit OK.

Do the same process again for rac02 (We are building a 2 nodes RAC). Don't forget to type "rac02" wherever I used "rac01" here.

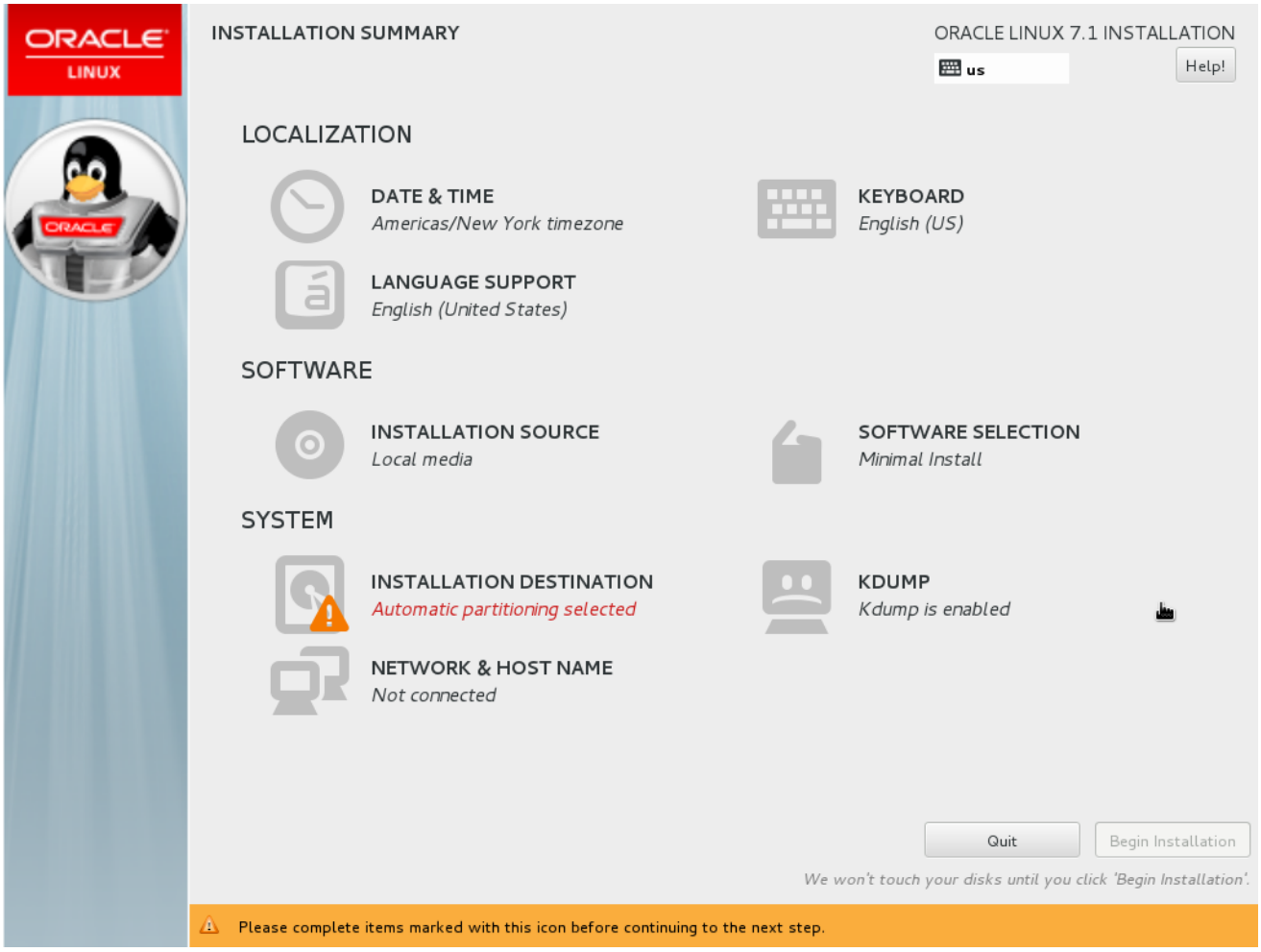
OS Installation

The following process can be done in parallel for both machines. Make sure you are not getting confusing when filling fields.

After starting the machine, hit enter within the VM to start the installer.
Notice, and remember, which key will release the control from the VM and give it back to the OS. Without it you'd might get "stuck" in the virtual machine, and won't be able to go back to the hosting operating system. On my Windows machine it is the right Ctrl, and on my Mac it is the right command. Notice closely which key it is!

The first screen allows you to select the language you would like the machine to use. Select "English (United States)".

Next the main installation menu is shown.
After each configuration we will go back to it by pressing "Done" at the top left of the screen.



On the Network & Host name settings:

On each machine, write the name of the machine in the bottom of the screen "rac01.ractest" for example.

For rac01 we will use the public ip address 192.168.56.71, while for rac02 - 192.168.56.72.

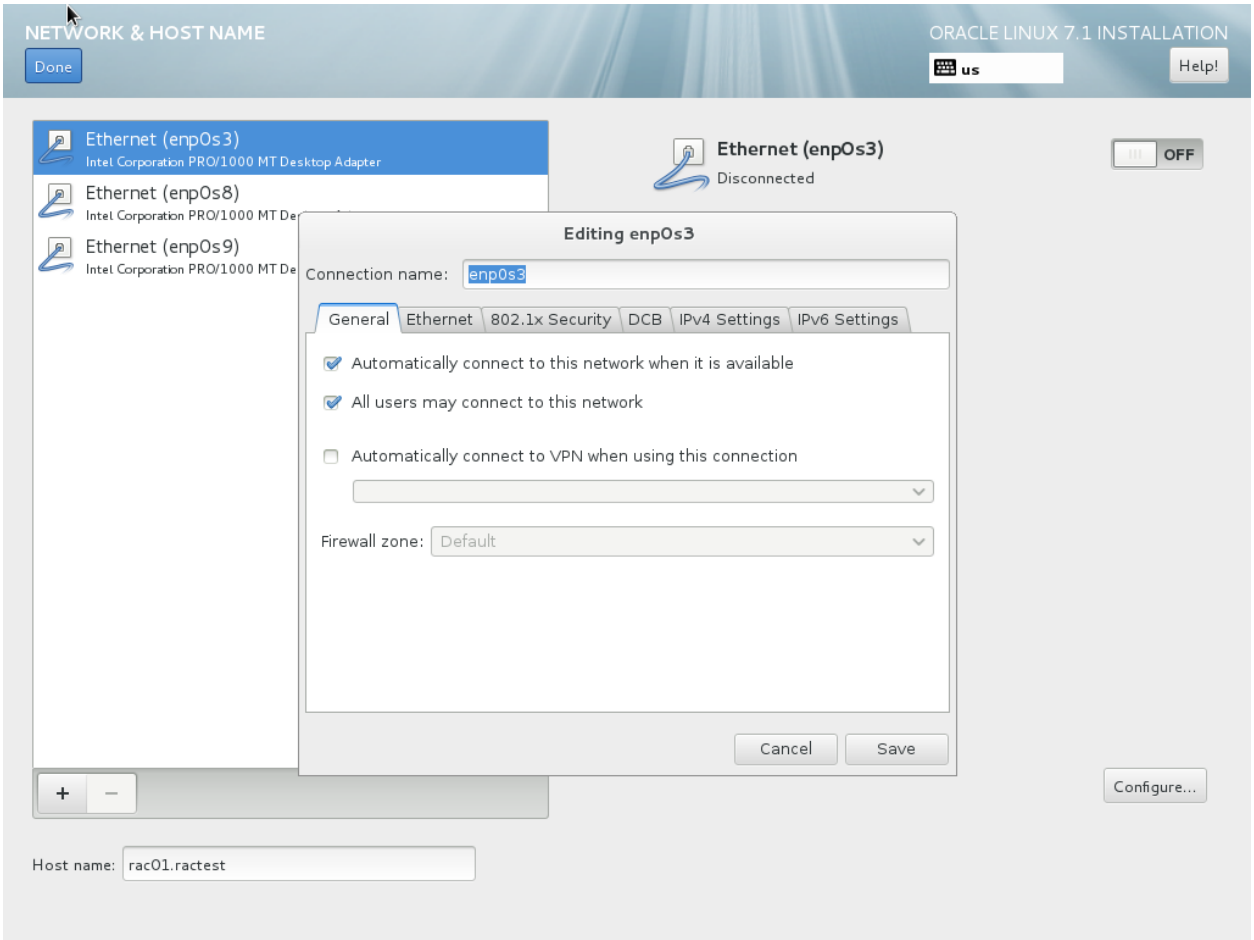
For rac01 we will use the private ip address 192.168.10.1, while for rac02 - 192.168.10.2.

The NAT device will be assigned an IP automatically (and because of it cannot be used for the RAC easily).

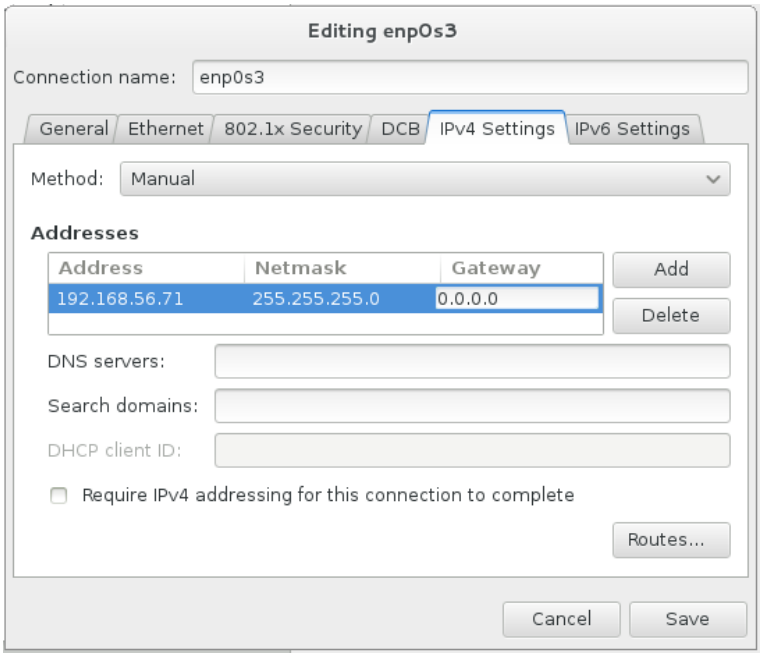
If you're trying to use more nodes –increase the number for each additional node.

To change the configuration of a specific device – stand on the device and choose configure.

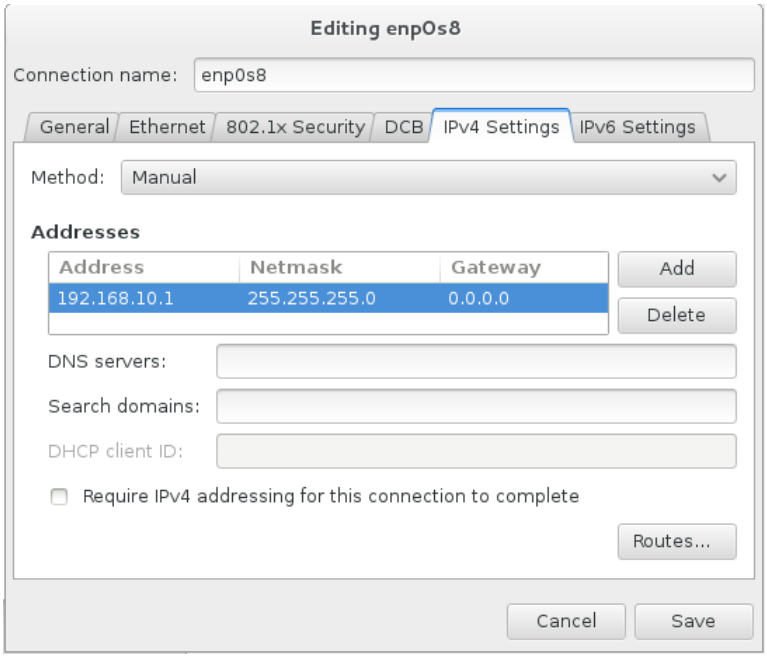
Don't forget to set each network device to auto-connect (as shown in the following illustration) – it has to be done for each of the devices in addition for setting the IP address:



For the first device, the public one, set the IP as shown (192.168.56.71 , 255.255.255.0 , 0.0.0.0).
Make sure the device is set to auto-connect in the “General” tab.



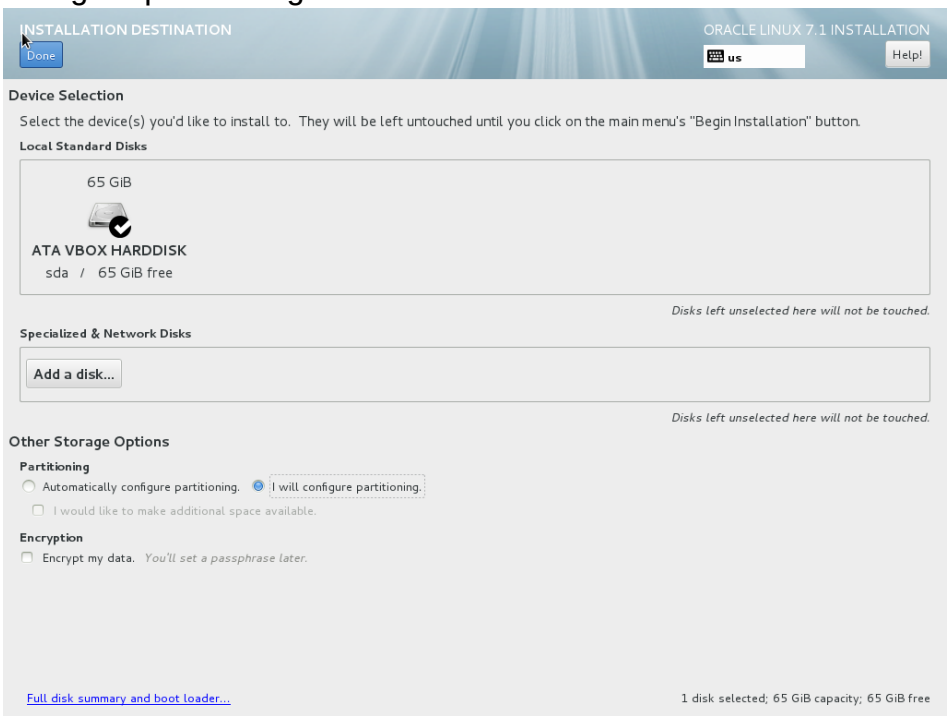
For the second device, the private one, set the IP as shown (192.168.10.1 , 255.255.255.0 , 0.0.0.0).
Make sure the device is set to auto-connect in the “General” tab.



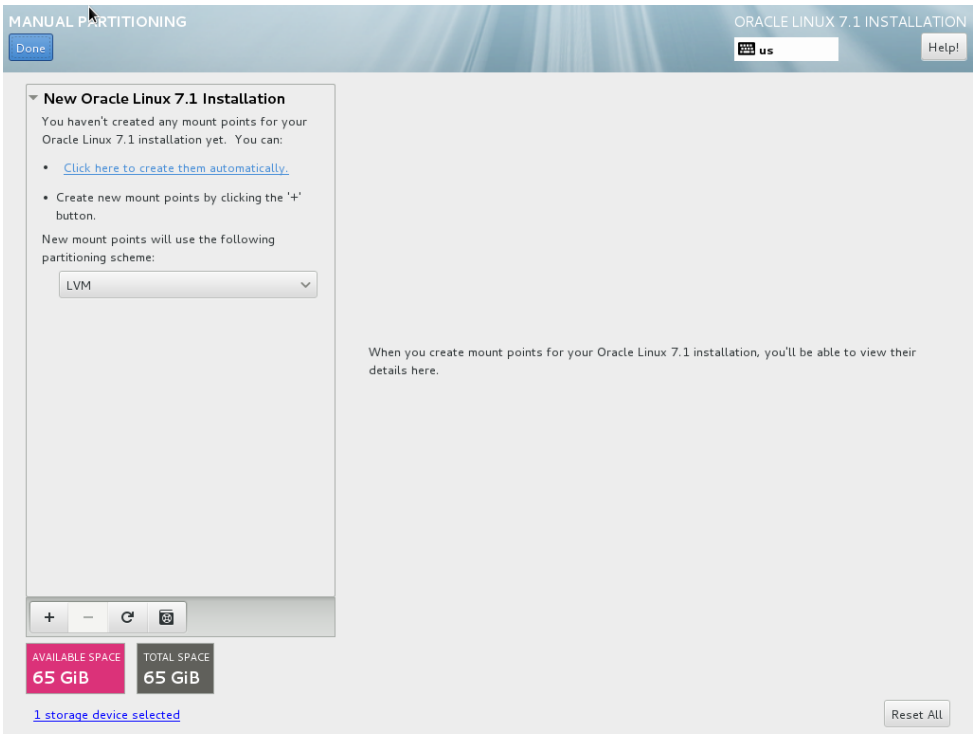
For the third device, on the general field, set it to auto-connect (it will use DHCP, so not change is required for the IPv4 settings).

On the "Installation Destination":

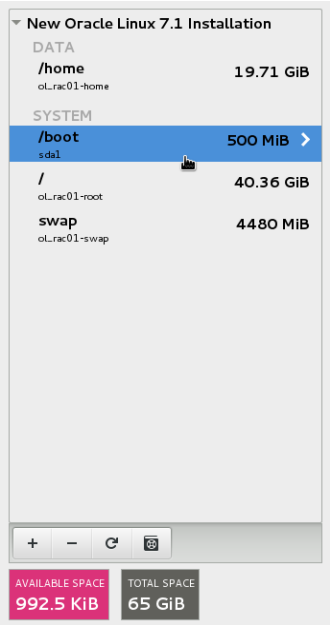
Make sure the only existing hard drive has a checkmark on it, and hit the "I will configure partitioning":



Hit the "Done". You will get to the partition configuration screen.
Hit the "Click here to create them automatically" (in a moment we'll modify it):

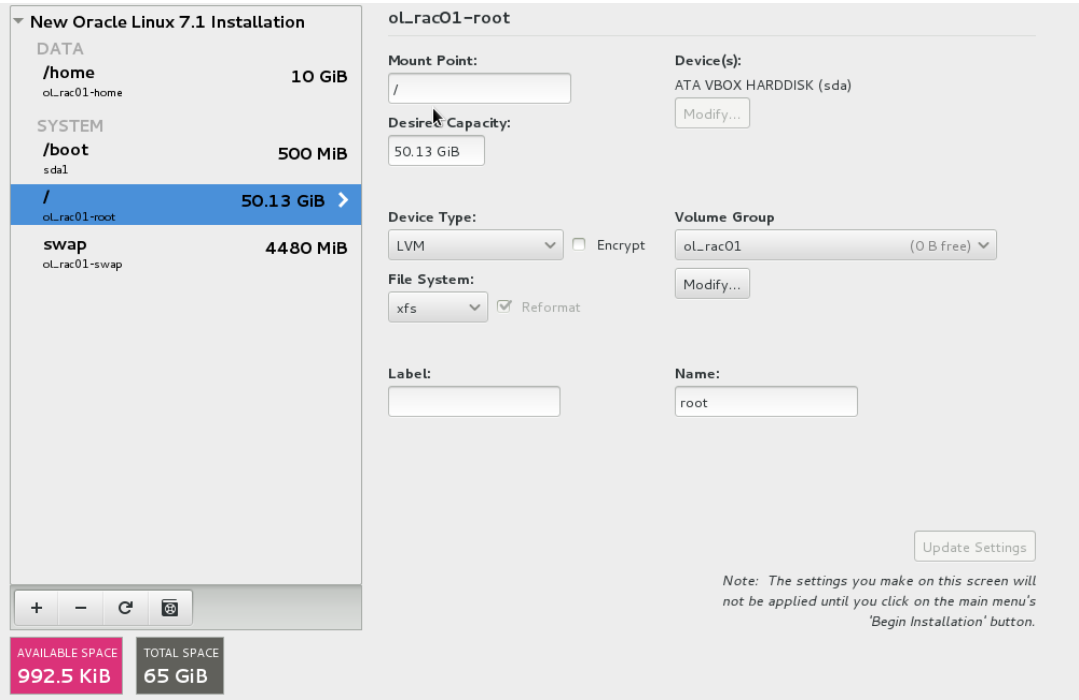


This is the default configuration you should see:



If the swap is less than 4G – make sure you increase it to at least 4GB(recommended to be the RAM size allocated to the machine, no more than 16GB is required for swap).

Now, select /home, decrease it to 10GiB, select / and increase it by the 10 Gib taken from /home (make sure you hit “update settings” after each change).
The final results should look like that:



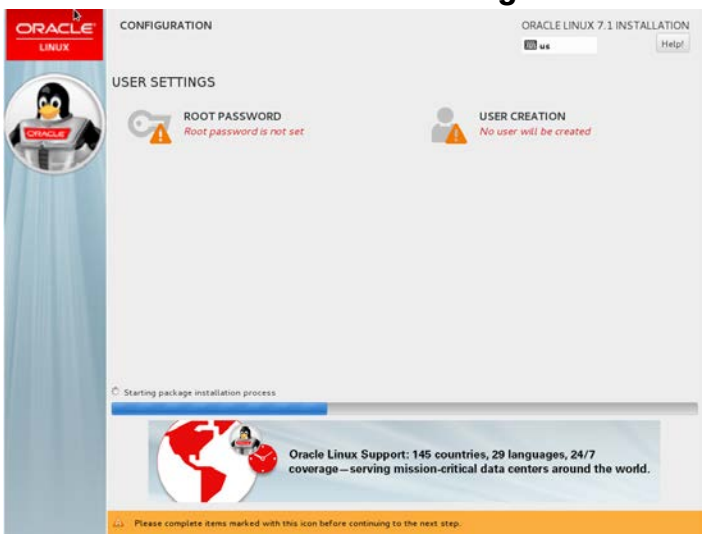
When hitting “done” you’ll have to confirm the changes by hitting “Accept Changes”.

Next, Software Selection:
Select “Server with GUI”,
With the following packages: “Java Platform”, “KDE”, “Load Balancer”, “Performance Tools”, “Compatibility Libraries”, and “Development Tools” – not all are necessary for oracle, but it will be useful.

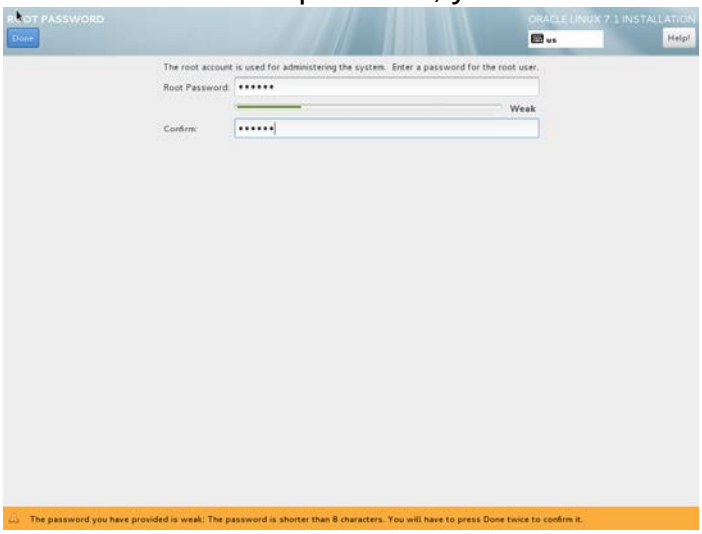
Now, Disable KDUMP (I don’t think a screen shot is necessary for it).

You’d might have to wait for a few seconds until the “Begin Installation” button appears.
Once it appears – hit it.

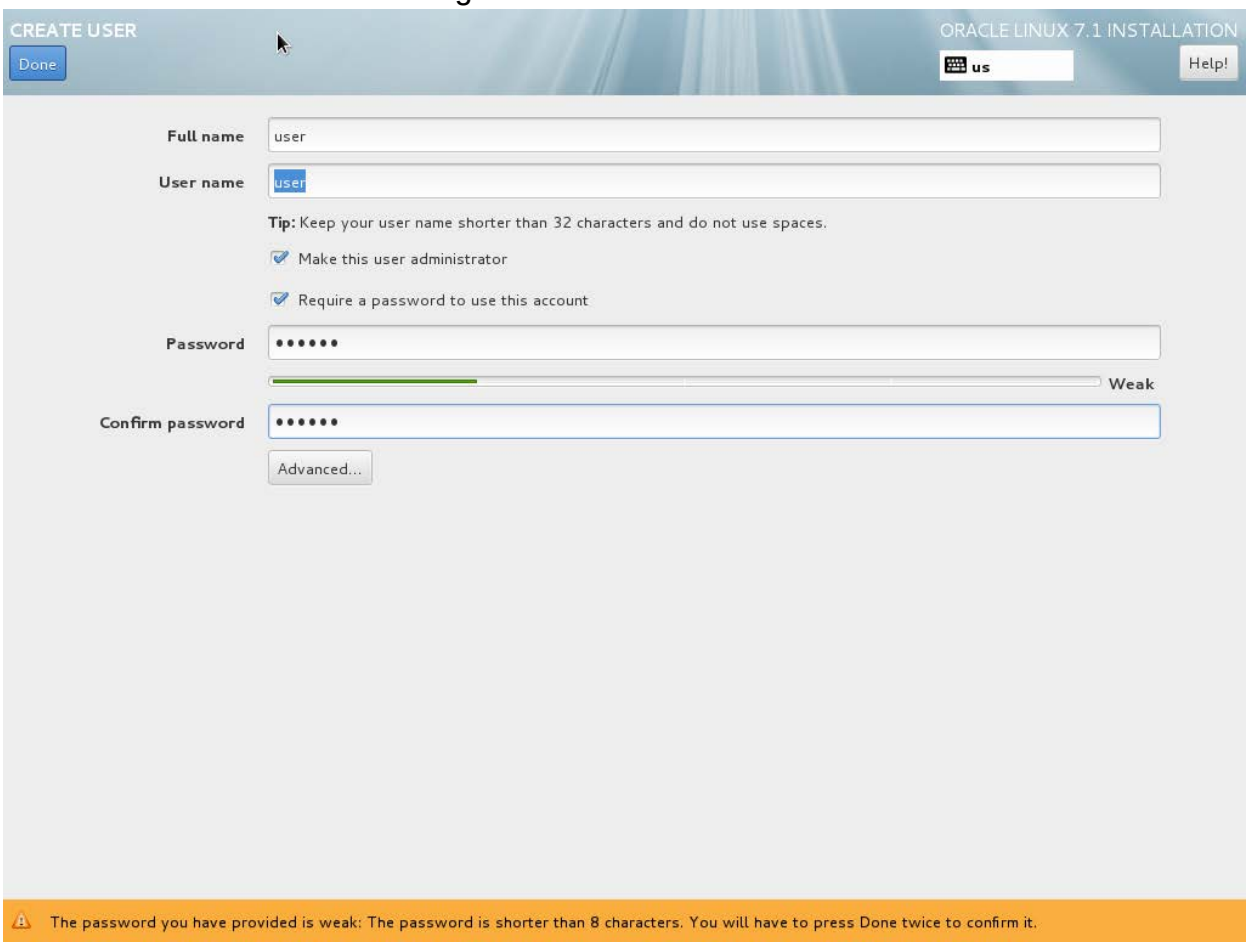
While the installation is running we need to set the users and their passwords:



Select ROOT password, and type `oracle` as the password in both fields.
Because it is a weak password, you’ll have to hit the “Done” twice.



Now hit “User Creation”.
We will create a user named `user` with a password `oracle` who’s configured to be an administrator.
You’ll have to hit “done” twice again.



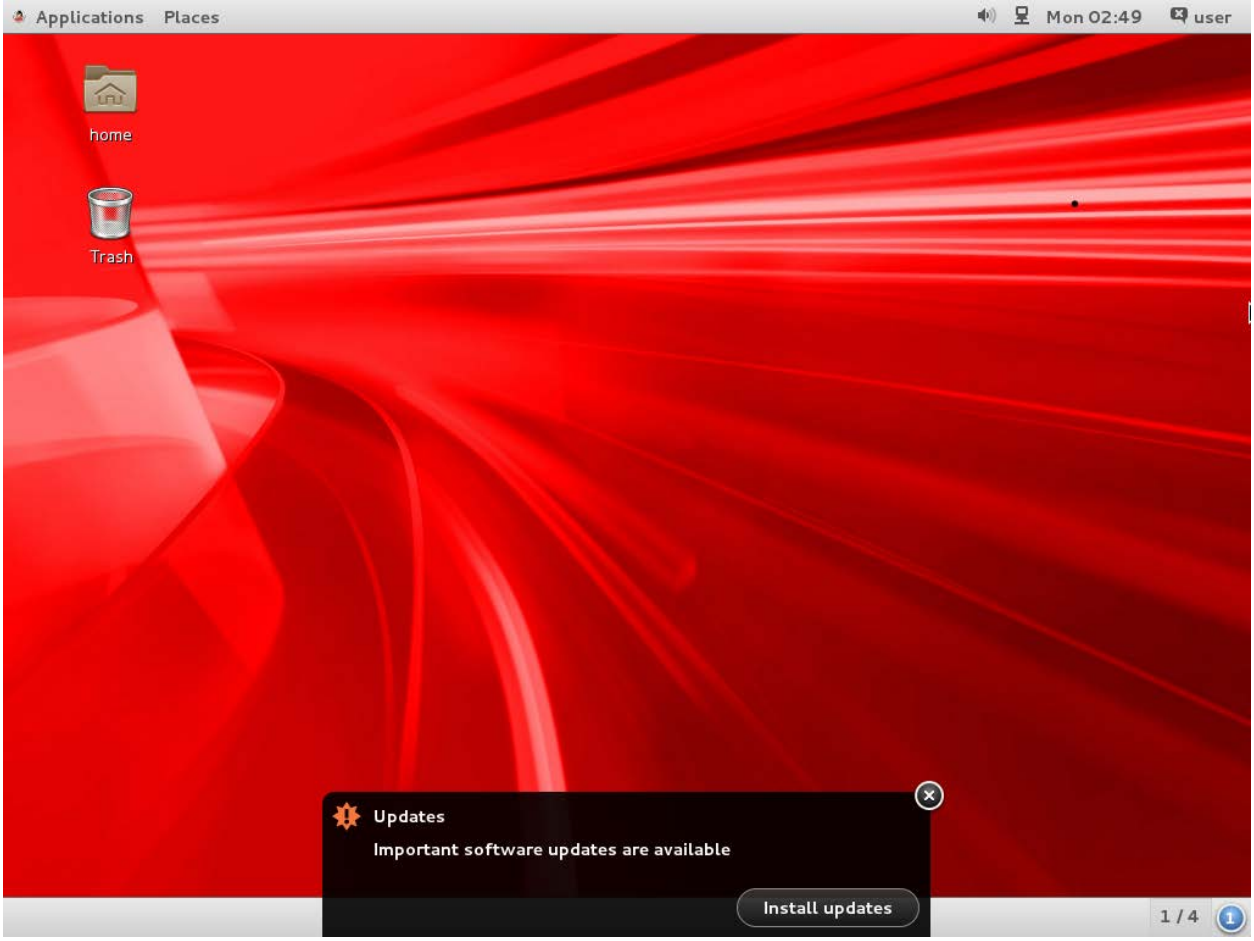
After the installation is complete, reboot the machine, accept the license, and skip / decline the registration to the Oracle Network (ULN).

Congratulations! You now have two machine that are running Oracle Unbreakable Linux, and have the network configuration for RAC.

Software and OS updates and configuration

Software update

After you're done installing the OS, and you login for the first time – this is the screen that will appear (your password is `oracle`):

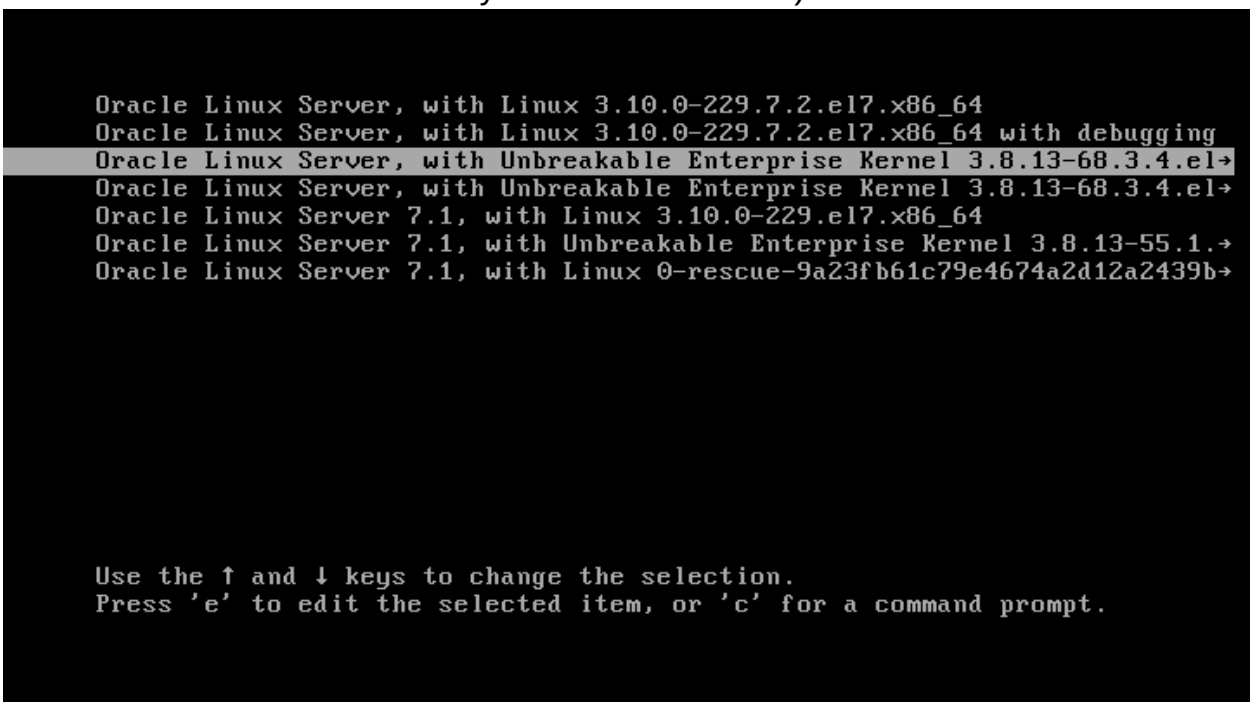


A few minutes after you login, the "Install Updates" popup will appear. Hit it and follow the instruction for updating everything (or type `yum update` in a terminal as root).

Notice that it will update the Kernel, a process that we will revert later because ASMLib drivers are not usually available for the latest kernel version.

Restart the machine, while making sure you start into a kernel version that has an ASMLib driver. In my case it is 3.8.13-68.3.4 (version 3.10* that appears would not be

able to use an ASM drivers the day this text was written):

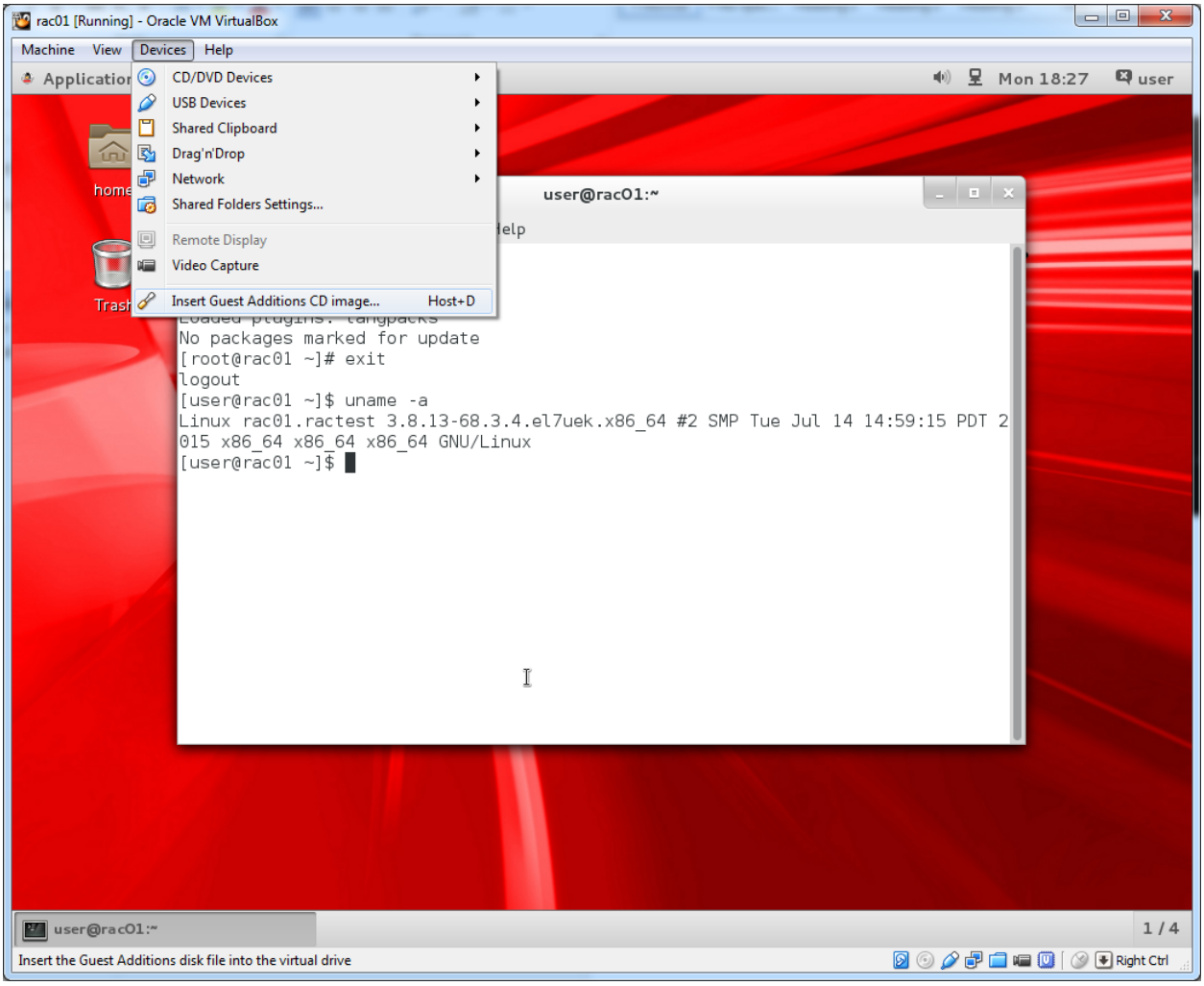


Later we will configure it to be the default (since it will change through the process, in the meanwhile on every reboot select it manually).

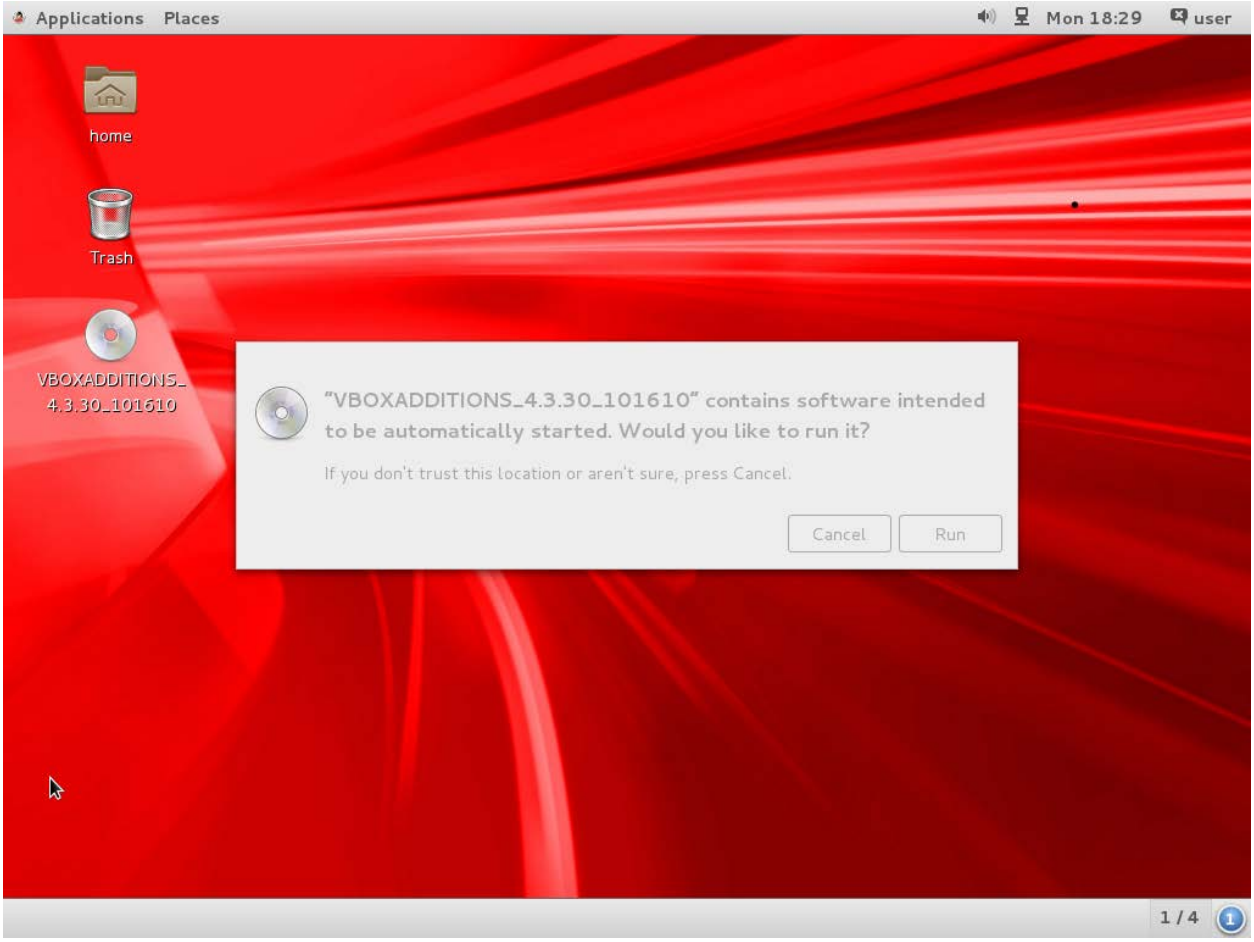
Make sure you're consistent with it, otherwise things will fail with weird error messages that do not reflect the problem.

Guest additions

Now install guest additions, by choosing “insert Guest Additions CD image…” from the “Devices” menu on both machines.



Hit the run within the VM afterwards (it will automatically detect the inserted “CD”):



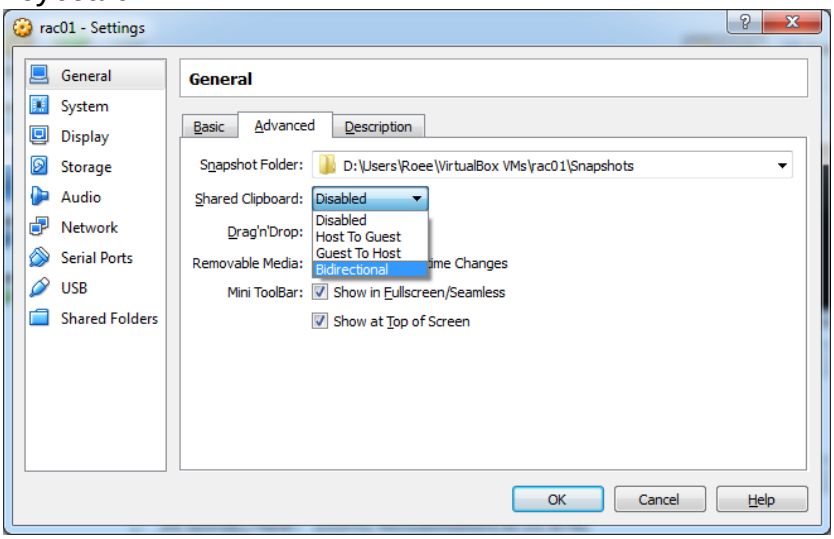
Insert your password (`oracle`) when asked to, hit “authenticate”, and wait for the install to finish.

When the installation is completed, hit “enter” to exit it and manually **shut down** the machine (from the top right of the screen, under the menu that pops when you hit the word `user` (your username).
You’d probably notice that now your mouse can move in and out of the VM without being locked in (and it is unnecessary to use the host key to release the keyboard and mouse from the virtual machine).

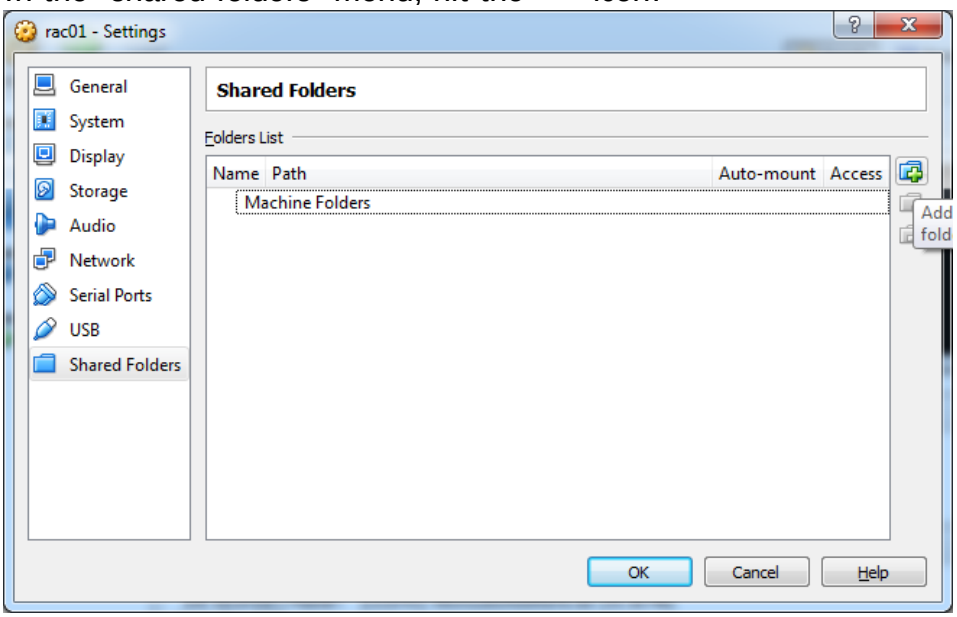
NOTE: if you login to your system, and you get a message the VM guest additions is not running, you probably use the wrong kernel (run `uname -a` to make sure you use the right kernel version). If necessary – reboot and choose the right kernel manually.

Bidirectional clipboard and shared folders:

In the VM machine, right click on each machine and choose “settings”.
In the “General” menu, and the “Advanced” tab –choose “bidirectional” for shared keyboard:

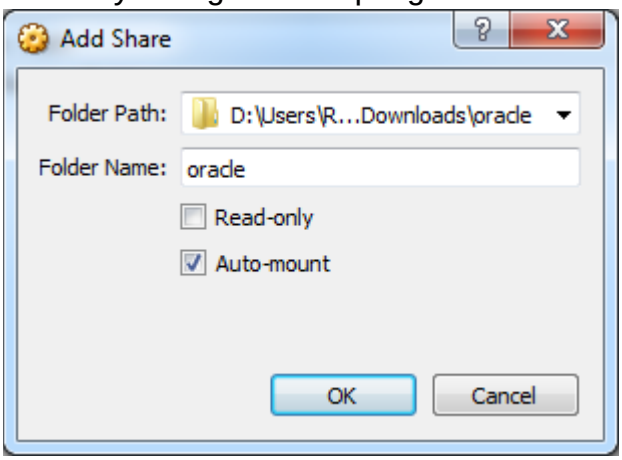


In the “shared folders” menu, hit the “+” icon:



Browse into the directory where you saved the oracle installer files and mark “auto-mount”. Name the folder `oracle`, if you’ll give a different name – you will have to

manually change the scripts given later:



Start your machines by hitting the “start” (don’t forget to choose the right kernel version) and login to `user`.

Start the terminal (under “Applications” and “Favorite” menus at the top of the screen). We will have to switch to root first, add oracle to the “vboxsf” group, and re-login (the groups permission within this environment apply only at login, therefore you have to logout and login).
Execute the following commands to allow access to the shared directory on both machines (note – you can copy from this document, and paste in the terminal by using “ctrl+shift+v” if the shared clipboard was set correctly, and the guest additions run):

```
su -
```

type the password “oracle” when requested

```
usermod -a -G vboxsf user
exit
exit
```

Log out from the user, and relogin to linux,

If you haven’t unzipped the installation disks, we’ll do so now.
On one of the machines restart the terminal, and execute (if the files have not been unzipeed yet):

```
cd /media/sf_oracle
unzip linuxamd64_12102_database_1of2.zip
unzip linuxamd64_12102_database_2of2.zip
unzip linuxamd64_12102_grid_1of2.zip
unzip linuxamd64_12102_grid_2of2.zip
```

Network settings

Hostname and hosts configuration

We need to be able to ping among all the machines freely, for that – we build a `hosts` file that contains all the machine names and IP addresses.
Some addresses should have been configure in a DNS server (scan for example).
We will be lazy and won't configure a DNS server. The installation will yell about it, we will instruct the installation to ignore it.

From a terminal, on both machines:
`su -`
type your password, `oracle`.
`vi /etc/hosts`
Hit "shift+g" (go to end of file) simultaneously, type "o" (add text after), and paste (ctrl+shift+v) the following few lines (if you changed the names of the machines, the ip addresses, or the domain, make sure to adjust the file content):

```
#public:
192.168.56.71   rac01.ractest   rac01
192.168.56.72   rac02.ractest   rac02

#private:
192.168.10.1    rac01-priv.ractest      rac01-priv
192.168.10.2    rac02-priv.ractest      rac02-priv

#virtual:
192.168.56.81   rac01-vip.ractest       rac01-vip
192.168.56.82   rac02-vip.ractest       rac02-vip

#Scan:
192.168.56.91   rac-scan.ractest        rac-scan
192.168.56.92   rac-scan.ractest        rac-scan
192.168.56.93   rac-scan.ractest        rac-scan
```

if completed successfully – hit "esc" (escape button, leave text edit mode) and type ":wq" (write and quit) and afterward the enter key, if not hit "esc" and type ":q!" (quit without writing), and redo it.
Make sure you can ping <u>between all machines to all machines</u>.

On each machine run:
`vi /etc/hostname`
and make sure the name of the machine is as expected. If it is not modify it (use "dd" to delete the wrong line, "i" to change to text edit mode, "esc" to leave edit mode, and ":wq" + the enter key afterwards to save and quit).
The content should be the machine name, with the domain name(`rac01.ractest` for example).

Password less SSH

Whatever is written here should be done for both, the `oracle` user and `root`.
I will demonstrate it here for the `oracle` user (which is required for oracle to be installed). The root should be done afterwards, the same way, and although it is not required, it will save you a lot of headache managing the machine.

If you don't want to follow these instructions – you can use whichever guide you'd like to for setting passwordless ssh between linux machines. The important thing is that you have to be able to ssh passwordless from each machine to each machine, using the `oracle` user, without entering a password – including to the same machine.

Log in to the `oracle` user and open a terminal. Within execute (on all machines):

```
ssh-keygen -t rsa
```

hit "enter" multiple times, until you get back to the terminal (allow the defaults to apply). If you're asked to overwrite an existing file – answer "yes".

On the first machine run:

```
cd ~
vi .ssh/authorized_keys
```

Hit the "i", and paste (ctrl+shift+v) the results of the following command on all machines to the first machine(you need to hit the "i" only once, that will put you in text edit mode, now paste one after another the results of these commands from all machines):

```
cd ~
cat .ssh/id_rsa.pub
```

The output of this command should look similar to (**do not use this example!** Each machine get a different key for ssh):


```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDImzUyhtTnFi8nYOwDbaWwZUVycL1VH9nH
ElPom3XhEm48KF+PRq/dO1hV9BA/oCJnUK7IVBlWyUFo2JuoeisuyB1jVIg4I3K0
XJZQP5JVqdy04czb6GdzkGoSwL828j4QrzFej9lOOw5np7tzowFRl0F9wTQyTXTX
h9zEnb3dA7RwT4JpmoDRHg24PeDI0B6rbt2TwPD1LVDp4xrDH2yLlm4Td43sbakv
cfhPxEDkxadIxG5fWSj92VgSAZLI8Z/gIAzsTIwNl5xhZ6pIjQxe7B9Z2FUkNUou
zJU/C0Skt3lvokC3dOTdhH50WLxxOIVZIRq645qViLe8gThX7ewH
oracle@rac02.ractest
```


After all the machine's ssh keys have been copied to the first one, open another terminal (and log in the relevant user – `oracle` or `root` by running `su -`) and get the local ssh key. On the first machine within the second terminal run:

```
cd ~
cat .ssh/id_rsa.pub
```

and copy the output (ctrl+shift+c).

Switch to the other terminal, where all the other keys have been pasted to. Paste the local key (ctrl+shift+v), hit “esc” and “:w” to save the document without exiting from it.

Mark all the content of the file, and copy it (ctrl+shift+c) – you can now exit this specific terminal by typing “:q” and the enter key (if you cannot exit – it means you didn’t save the changes – validate the file content and type “:wq” and enter).
On all other machines, within a terminal logged in with the relevant user, run:

```
cd ~
vi .ssh/authorized_keys
```

and paste the content copied (type “i” to edit the text and “ctrl+shift+v” to paste, “esc” and “:wq” + the enter key).

Now – on all machines run:

```
cd ~
chmod -R 700 .ssh
chmod 640 .ssh/authorized_keys
```

Now validate you can ssh from each machine to each machine, including itself, by executing `ssh` *`machine_name`*. For example:

```
ssh rac01
ssh rac01.ractest
ssh rac02
ssh rac02.ractest
```

If you can’t – you probably forgot to paste the content of .ssh/id_rsa.pub to the authorized_keys file on the machine that can’t connect (or didn’t change the file permission with the chmod).

Repeat this process for the root user (login to root from each terminal using `su -`)

Disable selinux and firewall

On all machine, in a terminal execute:

```
vi /etc/selinux/config
```

Find the line with the text:

```
SELINUX=enforcing
```

Delete it (by hitting “dd”)

instead of it, put the following line (by hitting “o” and pasting “ctrl+shift+v”):

```
SELINUX=disabled
```

Save the edit and exit by hitting “esc” and “:wq”+enter.

MAKE SURE THE FILE CONTENT IS CORRECT, THE MACHINE WILL NOT RESTART IF THERE IS ANY SPELLING MISTAKE HERE.

On all machines, as root, execute:

```
systemctl disable firewalld
systemctl stop firewalld
```

Restart both machines to disable selinux either from the user interface, or by running

```
init 6
```

in a terminal with `root` permissions.

Oracle Software pre-requisits

In a terminal login as root, and install all required software:

```
su -
```

type your password, oracle .

```
yum install binutils  compat-libcap1 gcc gcc-c++ glibc.i686
glibc glibc-devel.i686 glibc-devel ksh libaio.i686 libaio
libaio-devel.i686 libaio-devel libgcc.i686 libgcc libstdc++.i686
libstdc++ libstdc++-devel.i686 libstdc++-devel libXi.i686 libXi
libXtst.i686 libXtst make sysstat
```

Hit “y” and enter to approve the installation of all of the prerequisite software.

The following installation will create the user `oracle`.
It can be run, out of the box, only on Oracle Unbreakable Linux. If you decided to use a different distribution, which is not recommended – go to Oracle web site and find the instruction on how to configure Oracle’s repository for your distribution.

```
yum install oracle-rdbms-server-12cR1-preinstall
```

Approve the installation by typing “y”+enter.

Now, on both machines, we will configure additional required settings for oracle (still from within the root user):

```
vi /etc/security/limits.conf
```

Add the following lines at the end of the file (“shifg+g”, “o”, “ctrl+shift+v” after copying both lines, and “:wq”+enter to save the file):

```
oracle soft nproc 2047
oracle hard nproc 16384
```

Continue by executing:

```
passwd oracle
```

type the password `oracle` twice.

```
usermod -a -G vboxsf oracle
groupadd asm
groupadd asmop
usermod -a -G asm oracle
usermod -a -G asmop oracle
mkdir /u01
chown oracle:oinstall /u01
cd /media/sf_oracle/grid/rpm/
CVUQDISK_GRP=oinstall; export CVUQDISK_GRP
rpm -iv cvuqdisk-1.0.9-1.rpm
```

you’d might have to change the above rpm file name to fit the version that is on the oracle installation disk you have. I recommend typing “rpm -iv cvuqdisk” and hitting the tab to autocomplete it automatically

```
su - oracle
vi .bashrc
```

Hit “shift+g” simultaneously, type “o”, and paste (ctrl+shift+v) the following command:
```
umask 022
```
if you completed it successfully – hit “esc” (escape) and type “:wq”+enter, if not hit “esc” and type “:q!”+enter, and redo it.
```
exit
```
The exit will return the control to the `root` user.

The terminal should be in the user root on all machines.
```
vi /etc/security/limits.conf
```
type “shift+g” simultaneously, type “o”, and paste the following two lines:
```
oracle            soft    nofile          4096
oracle            hard    nofile          65536
```
if you completed it successfully – hit “esc” (escape) and type “:wq”+enter, if not hit “esc” and type “:q!”+enter, and redo it.
Now we will set shared memory for the machine:
```
vi /etc/fstab
```
type “shift+g” simultaneously, type “o”, and paste the following two lines:
```
shmfs /dev/shm tmpfs size=12g 0 0
```
if you completed it successfully – hit “esc” (escape) and type “:wq”+enter, if not hit “esc” and type “:q!”+enter, and redo it.

Restart the machine, and this time login as the user “oracle” and not the user `user`.
You can restart the machine from the user `root` by issuing the command:
```
init 6
```
Make sure you use the right Kernel version – her, my kernel version changed – therefore I used the commands given in the “Setting the system to boot to the right kernel version” subsection to make it automatically run the right kernel.

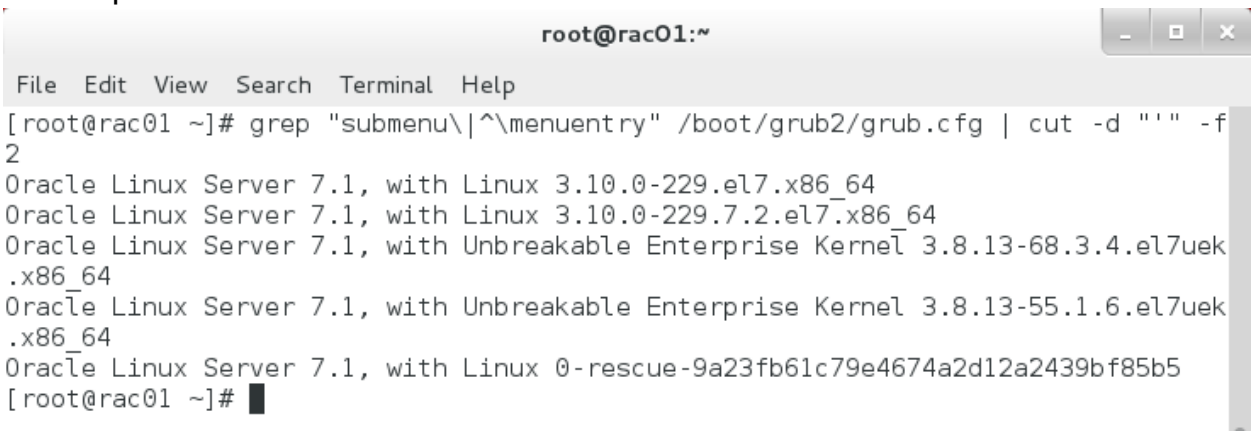
After the restart, you can check the configuration applied by running:
```
df –h
```
and looking for the 12g shm drive,
other changes would be noticeable from the oracle installation (it will warn about it)

Setting the system to boot to the right kernel version

From a terminal logged in as root, execute the following command (you would really want to copy paste it…):

```
grep "submenu\|^\menuentry" /boot/grub2/grub.cfg | cut -d "'" -f2
```

The output of this command should be similar to that:



Mark with the mouse the line of the kernel version you want the system to boot to (make sure you're choosing the right kernel version), and attach it by pasting it between the commas of the following command:

```
grub2-set-default ""
```

The result should be similar to:

```
grub2-set-default "Oracle Linux Server 7.1, with Unbreakable Enterprise Kernel 3.8.13-68.3.4.el7uek.x86_64"
```

Restart the machine and validate it boots to the right kernel.

You can always run

```
uname –a
```

to validate the currently executing kernel version

ASM disk configuration

This section is divided to two: setting the shared drive for the ASM in VirtualBox, and setting the oracleasm within the virtual machine.

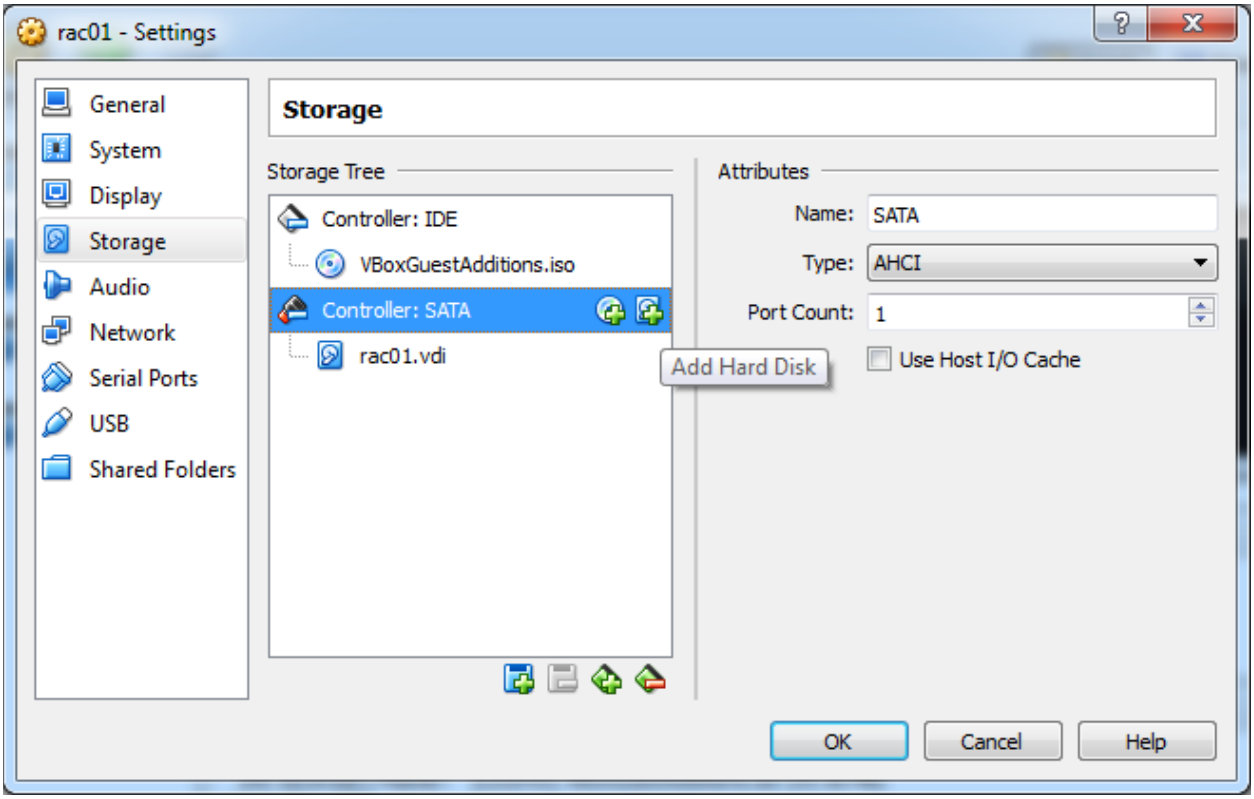
Setting the shared drive

First, turn off all the machines by issuing

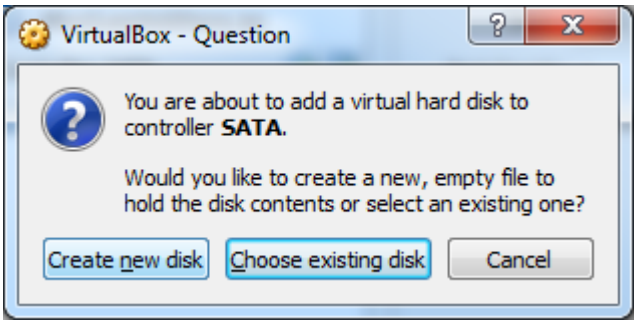
```
init 0
```

as root (or through the menu).

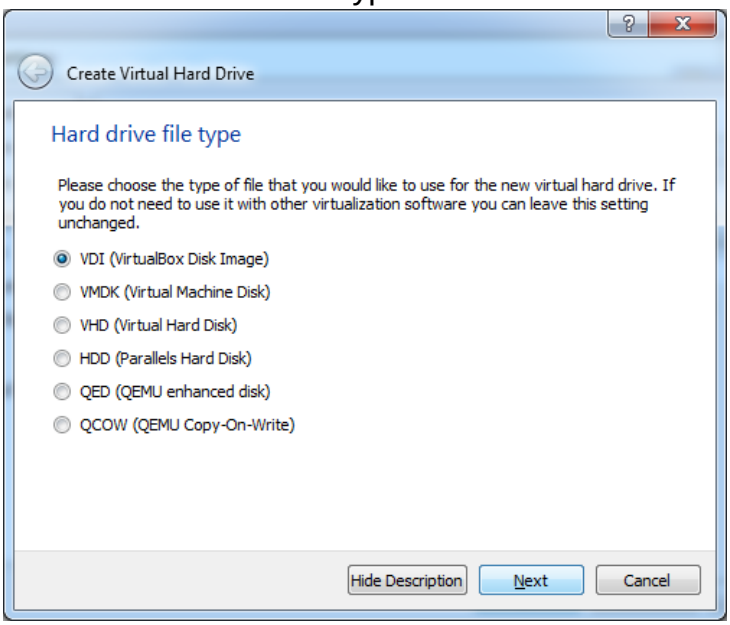
Within the VirtualBox application, right click on one of the machines, and select "Settings..", browse into the "Storage" menu, Select the "Controller: Sata", and hit the "+" on the hard disk icon:



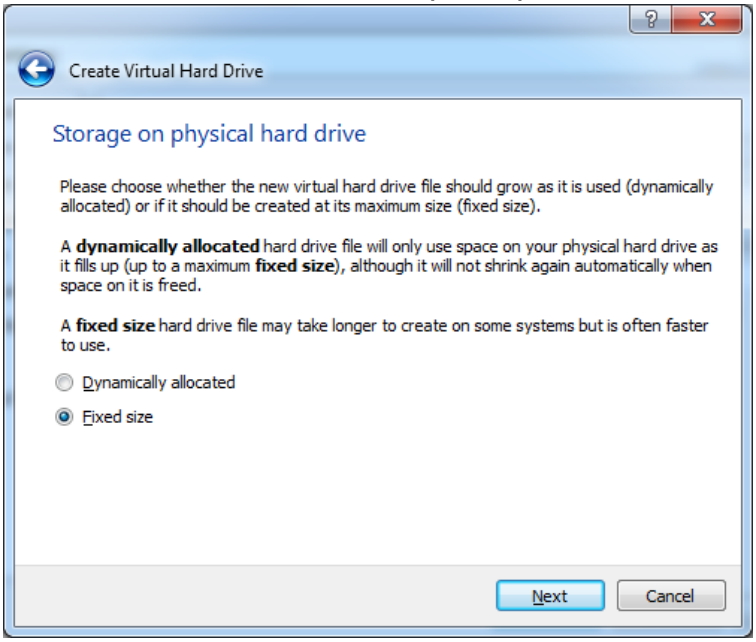
Choose "Create new disk":



Select “VDI” for the file type and hit next:

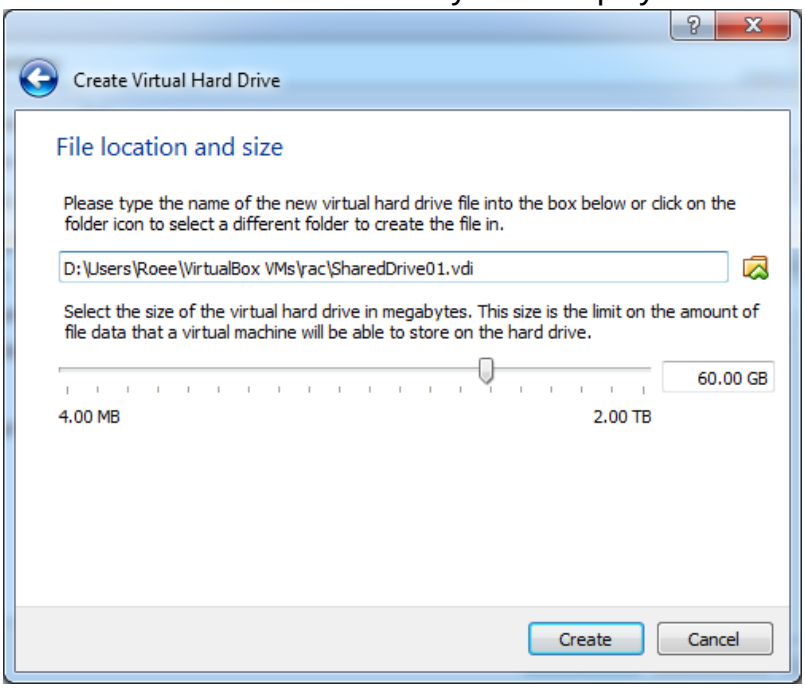


Now, we will use “Fixed size” (since performance is important here), and hit next:

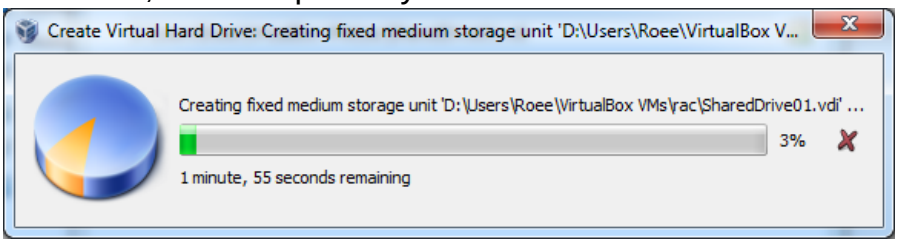


Choose a file size of 60GB (**less than 55GB won’t work!** It is a minimum requirement of oracle).
Make sure your placing the file where you want it to be (notice it is a shared disk among multiple machines, therefore – you’d might want to put it in its on directory, and

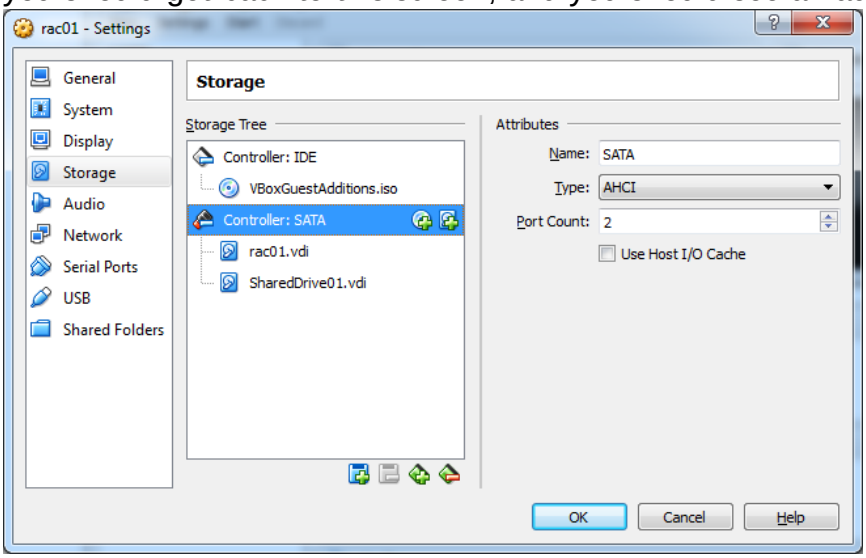
name it with a suffix of “01” so you could play with adding a “02” disk later):



Hit Create, and wait patiently as the line is filled:

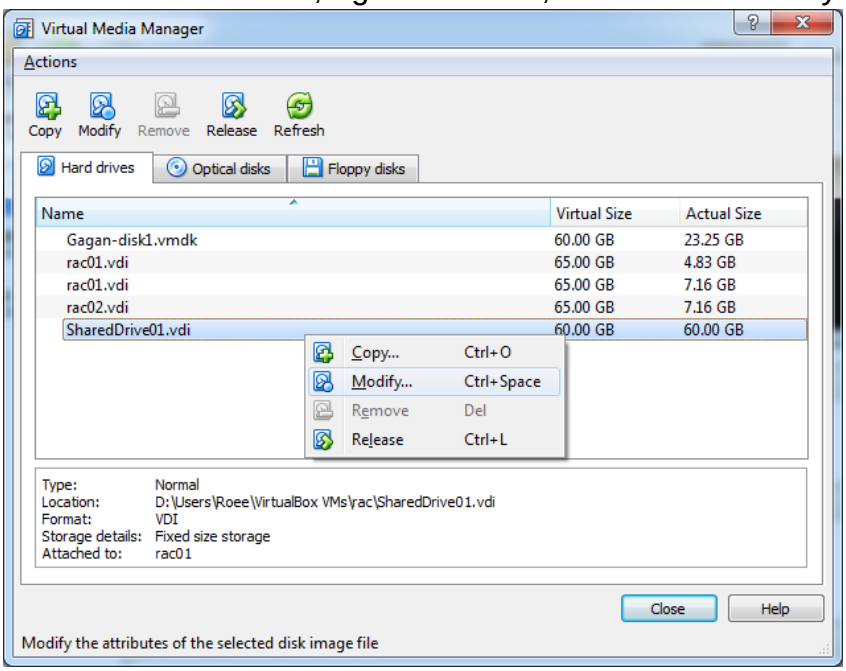


When the creation is done,
you should get back to this screen, and you should see an additional drive:

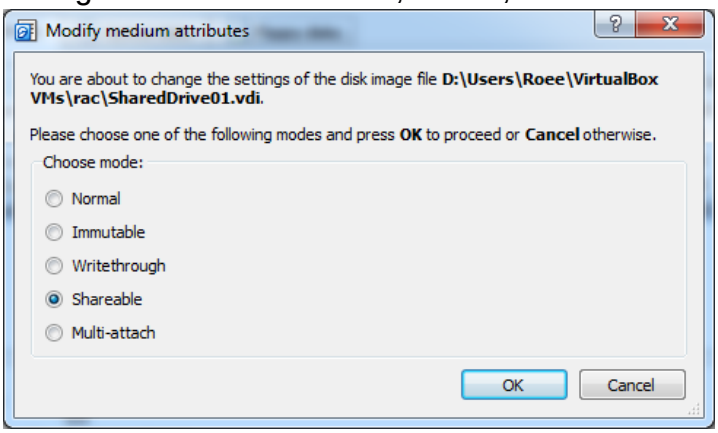


Sometimes, as a result of a bug, another grayed screen appears in front of it. Just hit the enter key, and it will take you back to the virtual box main screen.

Now on the VirtualBox main screen, choose the “file” menu at the top left, and select “Virtual Media Manager…”
Select the shared disk, right click on it, and choose “Modify…”:

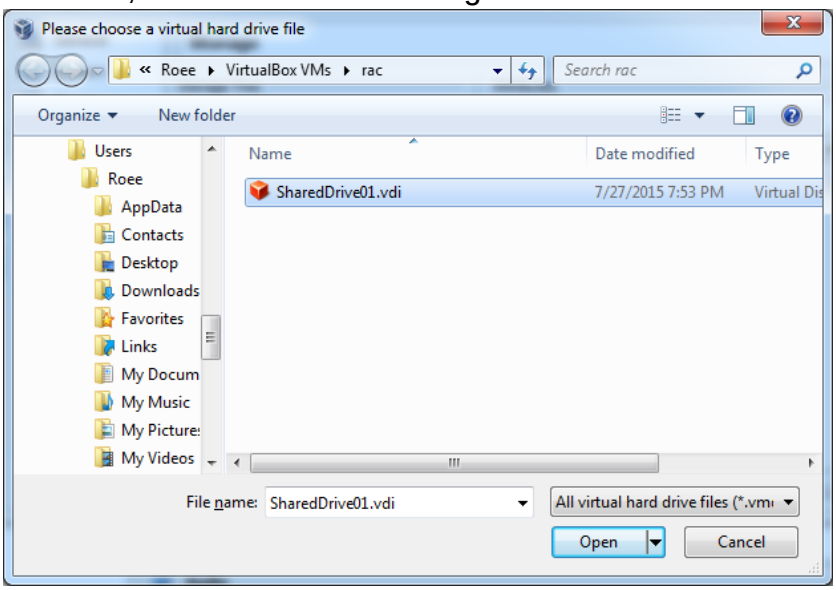


Change it to be “Shareable”, hit OK, and close the virtual media manager:

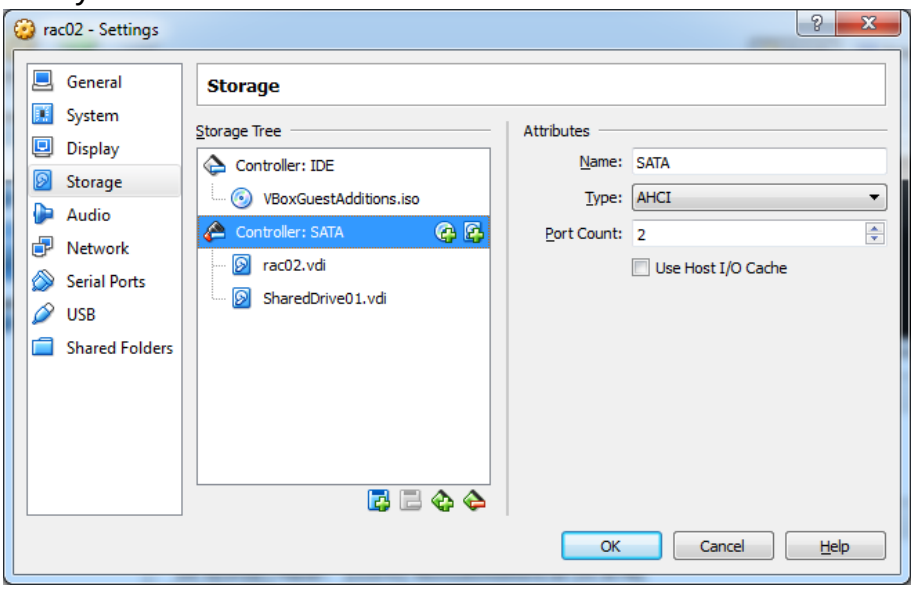


Now on each of the rest of the machines, go to the “Settings…” by right clicking on the machine, and in the storage menu add a hard drive as before, but instead of creating a

new disk, hit the “Choose existing disk” and browse to the shared drive you created:



Now you’ll see the same vdi file in all machines:



Hit OK, and start the machines.
As always – make sure the right kernel version is initiated.

Configuring the drive and ASMLib

On one of the machines, in a terminal logged in as root (su -) execute:

`fdisk /dev/sdb`

hit “n”, and enters until you get back to the command prompt.
Hit “w” to write the changes and exit fdisk.

On all machines, from a terminal running as root, run:

`yum install oracleasm-support`

Approve the installation by typing y

`/etc/init.d/oracleasm configure`

The user should be set to `oracle`

The group should be set to `asm`

On the last prompt hit y (the default), for the driver to scan for drives automatically.

```
This will configure the on-boot properties of the Oracle ASM library
driver.  The following questions will determine whether the driver is
loaded on boot and what permissions it will have.  The current values
will be shown in brackets ('[]').  Hitting <ENTER> without typing an
answer will keep that current value.  Ctrl-C will abort.

Default user to own the driver interface []: oracle
Default group to own the driver interface []: asm
Scan for Oracle ASM disks on boot (y/n) [y]:
Writing Oracle ASM library driver configuration: done
Initializing the Oracle ASMLib driver:                     [  OK  ]
Scanning the system for Oracle ASMLib disks:               [  OK  ]
```

On one of the machines, from a terminal connected as `root` run:

`oracleasm createdisk DISK1 /dev/sdb1`

Make sure no error return – if you did anything as described by here in this document – you won’t have any errors.

```
[root@rac01 ~]# oracleasm createdisk DISK1 /dev/sdb1
Writing disk header: done
Instantiating disk: done
```

On all other machines, run from a terminal connected as `root`:

`oracleasm scandisks`

It should state as a response that `DISK1` was found:

```
[root@rac02 ~]# oracleasm scandisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
Instantiating disk "DISK1"
```

Any output different than what shown here would require you to investigate and fix it.
YOU MUST HAVE THIS SET UP CORRECTLY FOR CONTINTUING.
All nodes have to see DISK1 through the ASM.

Setting udev for the device

On one of the nodes, connect to a terminal as root and execute:

```
/usr/lib/udev/scsi_id -g -u -d /dev/sdb1
```

Note that `/dev/sdb1` might have to change if you didn't follow by instructions as is.
Copy the output – and replace it in all the code given below where appropriate.
The output should look similar to:
`1ATA_VBOX_HARDDISK_VBbaffce72-62c14e41`
Therefore everywhere you see that appears – replace it with the output you've got.

On all machines, as root, execute:

```
vi /etc/udev/rules.d/99-oracle-asmdevices.rules
```

And insert into the file the following line:

```
KERNEL=="sd?1", SUBSYSTEM=="block",
PROGRAM=="/usr/lib/udev/scsi_id -g -u -d /dev/$parent",
RESULT=="1ATA_VBOX_HARDDISK_VBbaffce72-62c14e41",
SYMLINK+="DISK1", OWNER="oracle", GROUP="asm", MODE="0660"
```

Notice the "RESULT" section should be changed to the output you've received from `scsi_id`.

In some cases the `scsi_id` would return different id on different nodes – make sure on each node you use the local scsi id. I never seen different results, but I read it is possible…

Restart the machines.
Execute:

```
ls -l /dev/sdb1
```

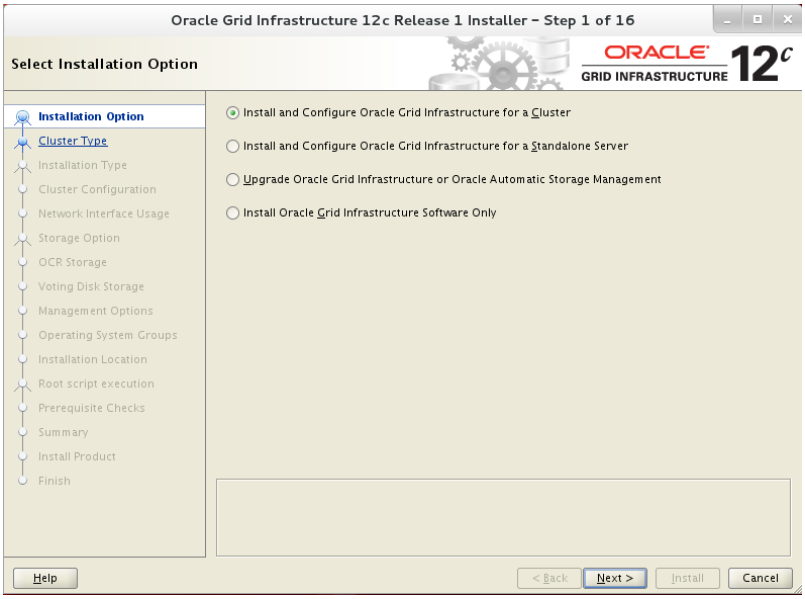
if the owner of the device is oracle – you've done it correctly, otherwise – fix the problem (read the instructions given here again, and validate all the steps were completed successfully).

Grid Infrastructure installation

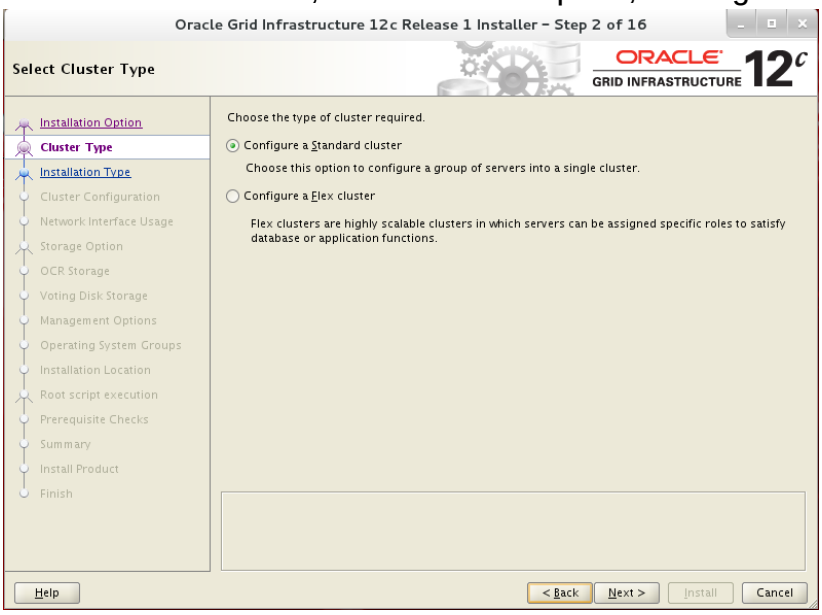
On the first node (`rac01` in my configuration), in a terminal logged in as `oracle`, browse to the grid infrastructure directory, and run the runInstaller file:

```
cd /media/sf_oracle/grid/
./runInstaller
```

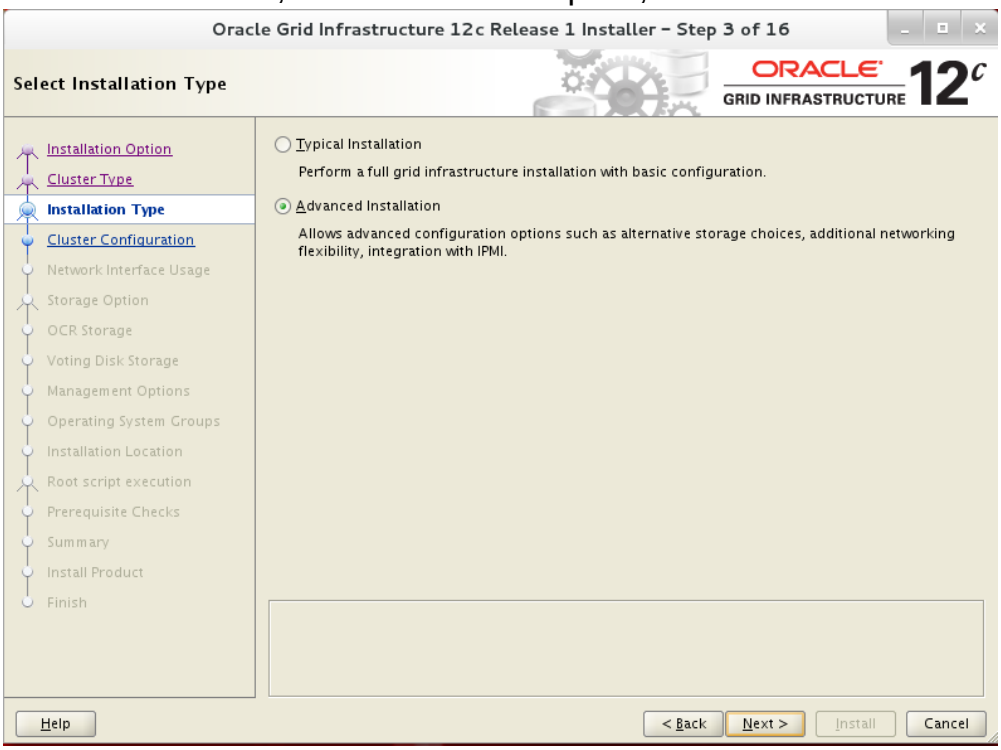
On the first screen, select the first option, "Install and Configure Oracle Grid Infrastructure for a Cluster":



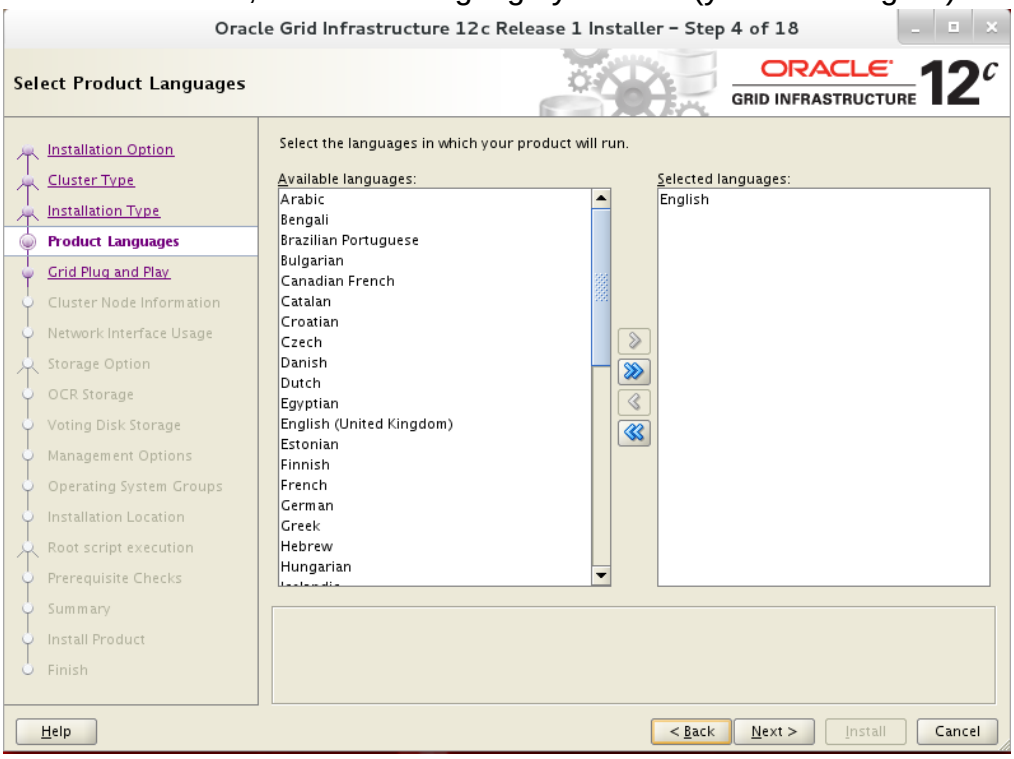
On the second screen, select the first option, "Configure a Standard cluster":



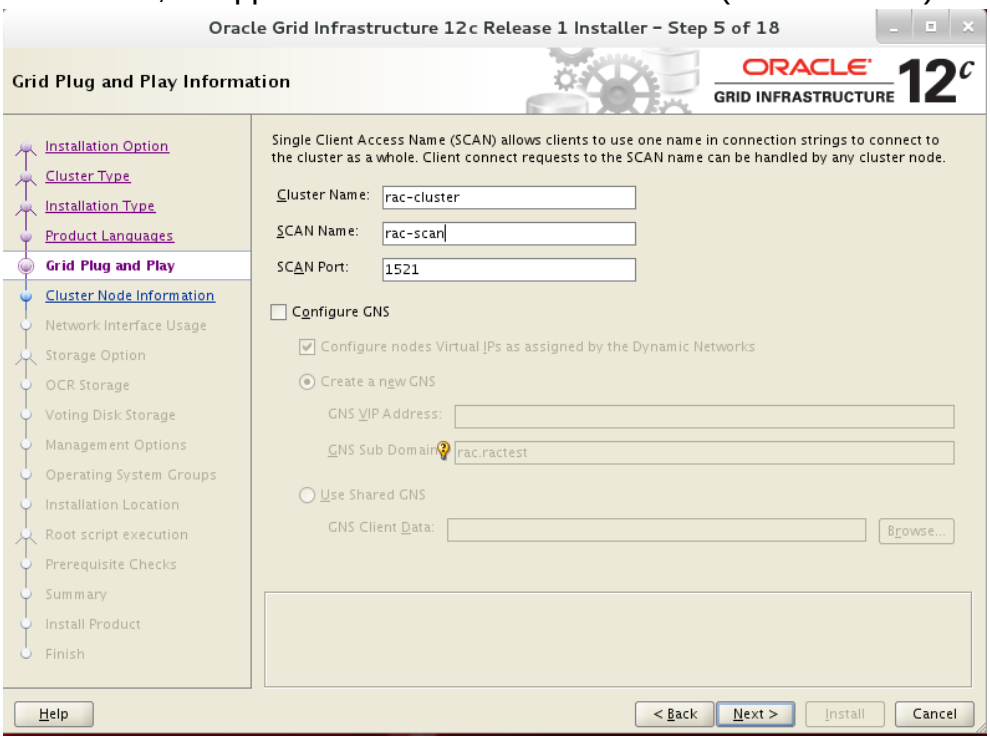
On the third screen, select the second option, “Advanced Installation”:



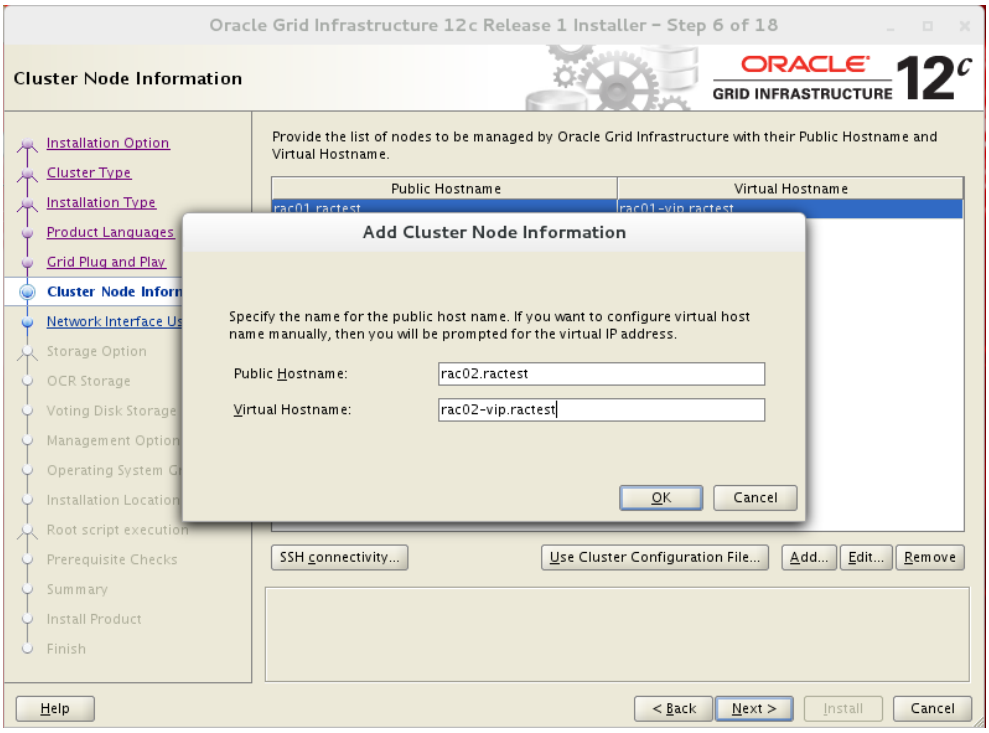
On the 4th screen, select the language you want (you want English):



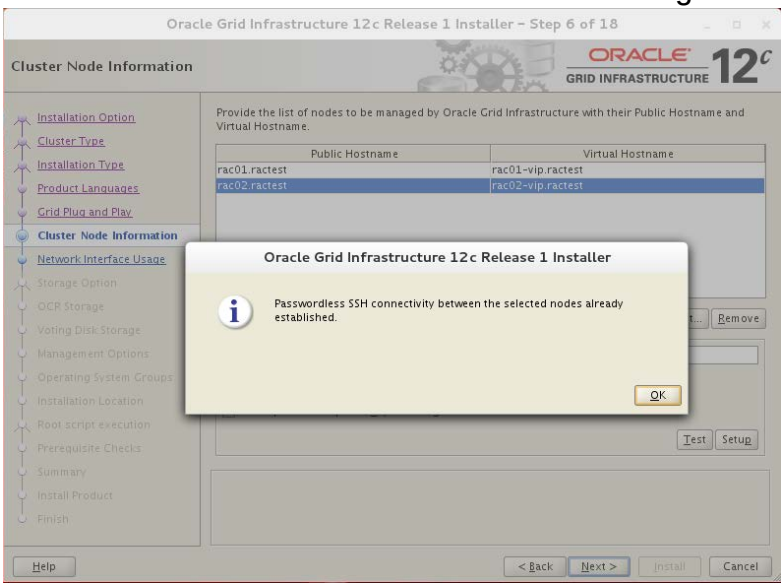
On the 5th screen, deselect the “Configure GNS”, and rename the “SCAN Name:” to `rac-scan`, as appears in the end of the hosts file (`\etc\hosts`):



On the 6th screen, hit the “add” for each machine you want the cluster to include, and fill in the information of this machine – the public name, and the vip, as appears in the hosts file, with its domain (for example: `rac02.ractest` and `rac02-vip.ractest`):

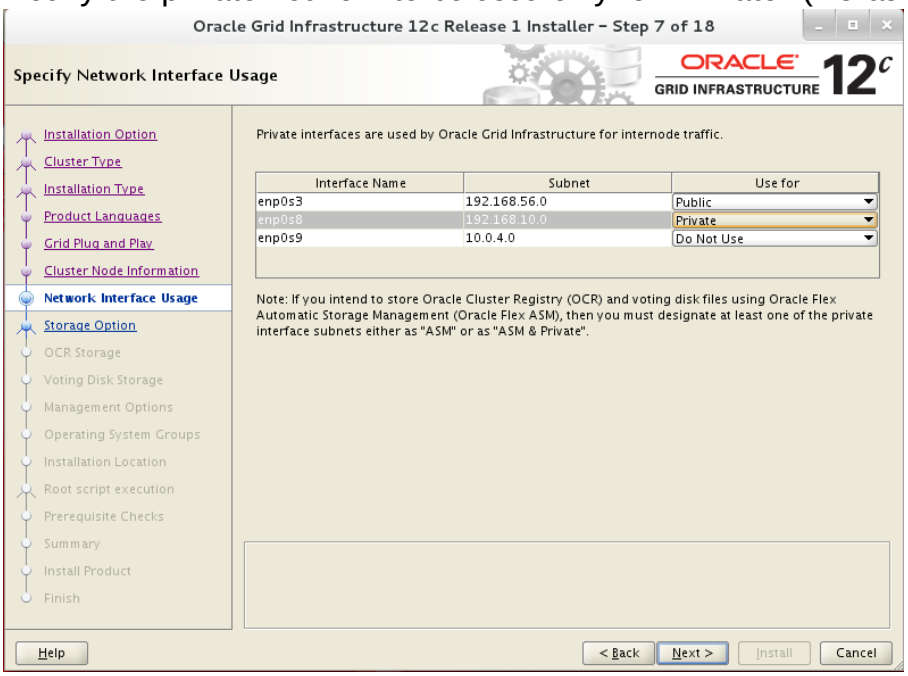


Wait for a few seconds (if you won't the following process would fail), and hit the "SSH Connectivity…" button, a tab will appear at the bottom part of the screen.
Select "Test" and wait for a confirmation message:

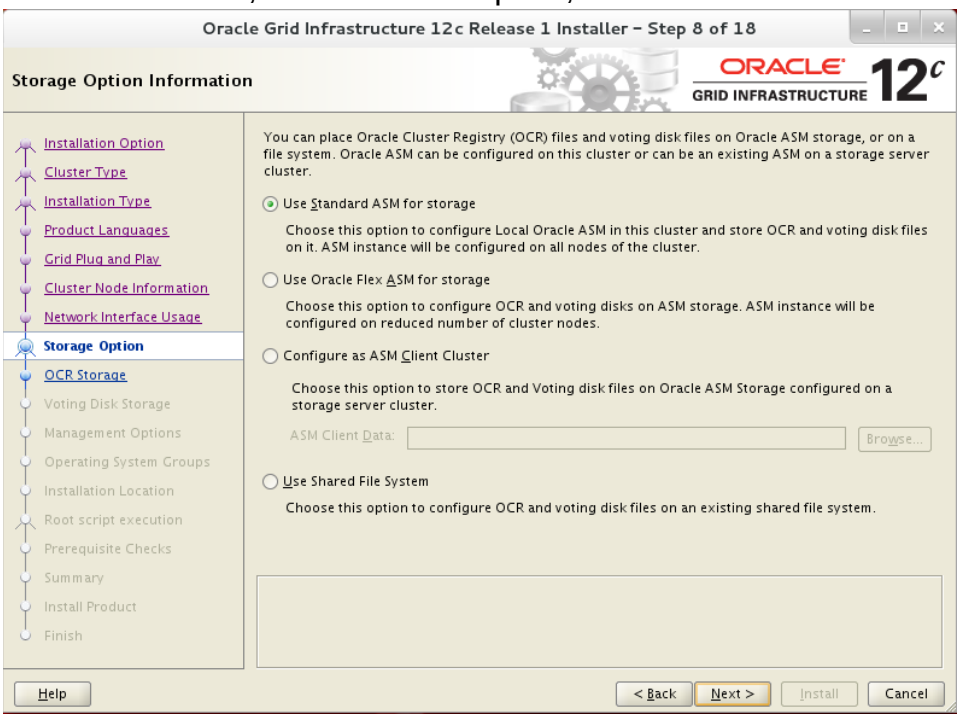


If the test fails – you did not configure the passwordless ssh correctly. Exit the installer, and fix it (you have to be able to ssh from each machine to all others machines, including the same one, without a password) from the terminal which will run the installer. DO NOT ASK ORACLE INSTALLATION TO CONFIGURE IT.

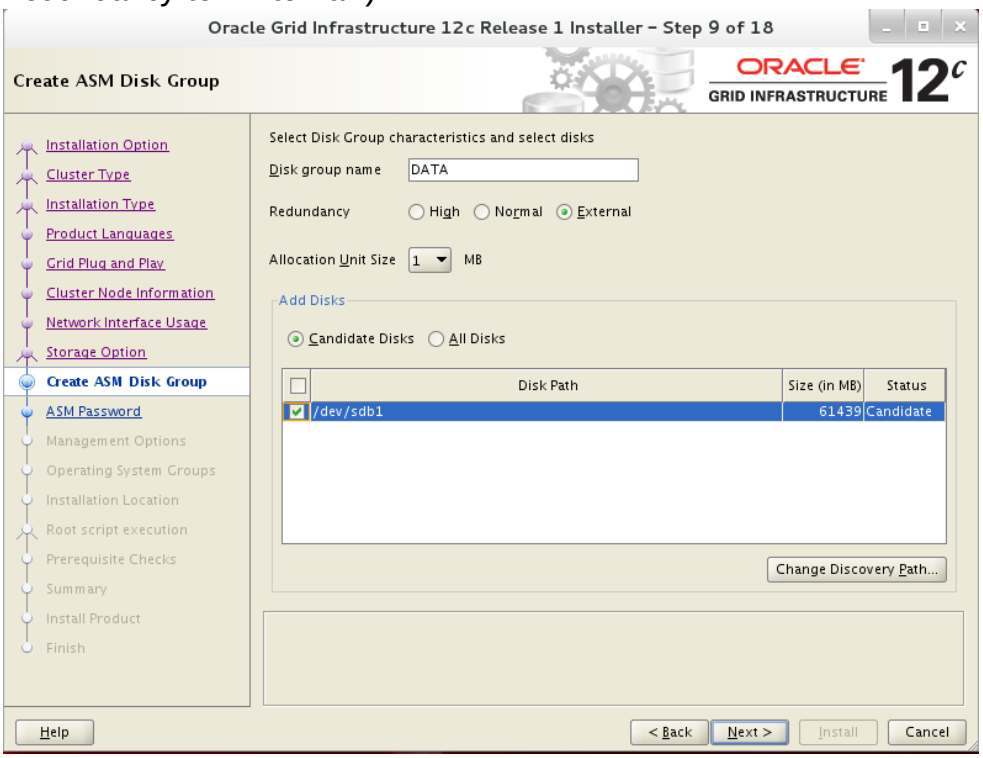
On the 7th screen, make sure the 192.168.56.0 subnet is assigned to the public network, and assign the 192.168.10.0 subnet only to the private network.
Modify the private network to be used only for "Private" (NO asm..):



On the 8th screen, select the first option, “Use Standard ASM for Storage”:

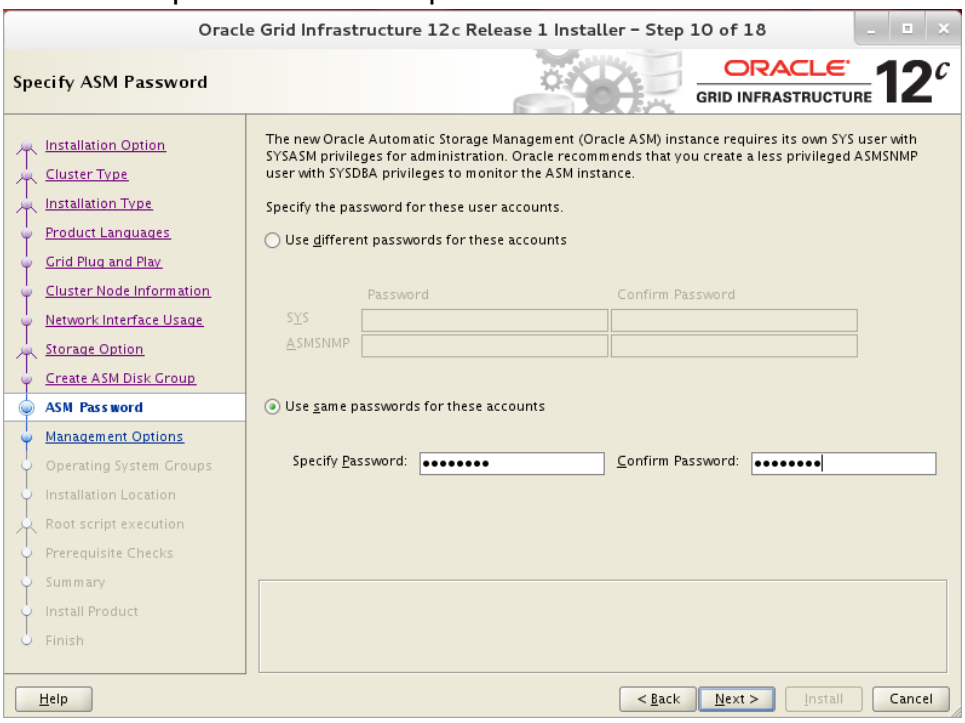


On the 9th screen, select the only disk that appears there (while changing its Redundancy to “External”):

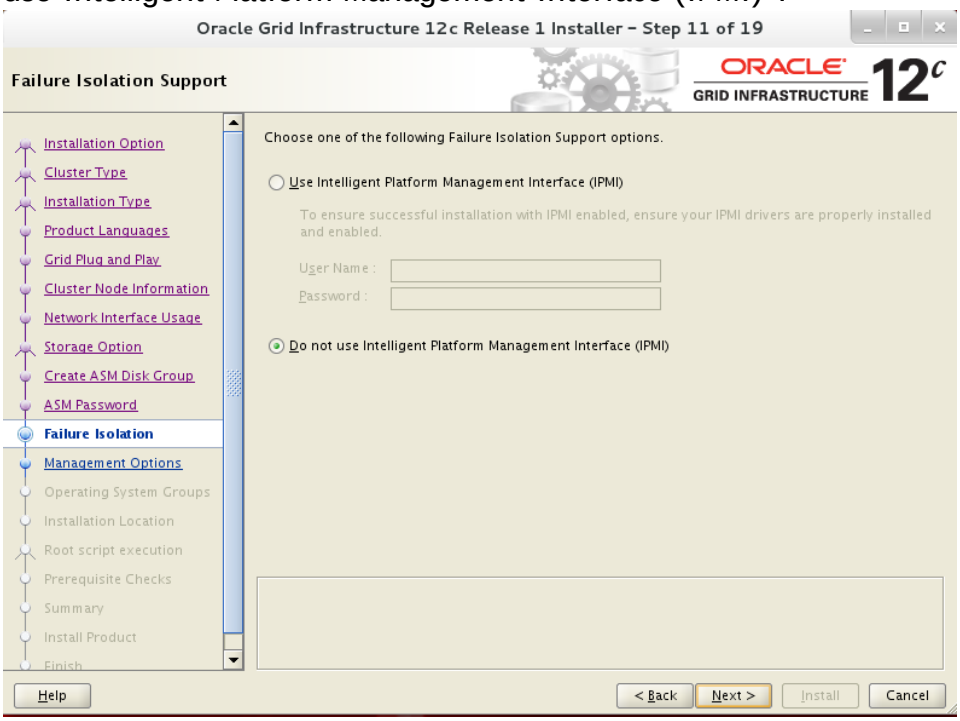


If you didn’t complete the udev configuration successfully the disk will not appear.

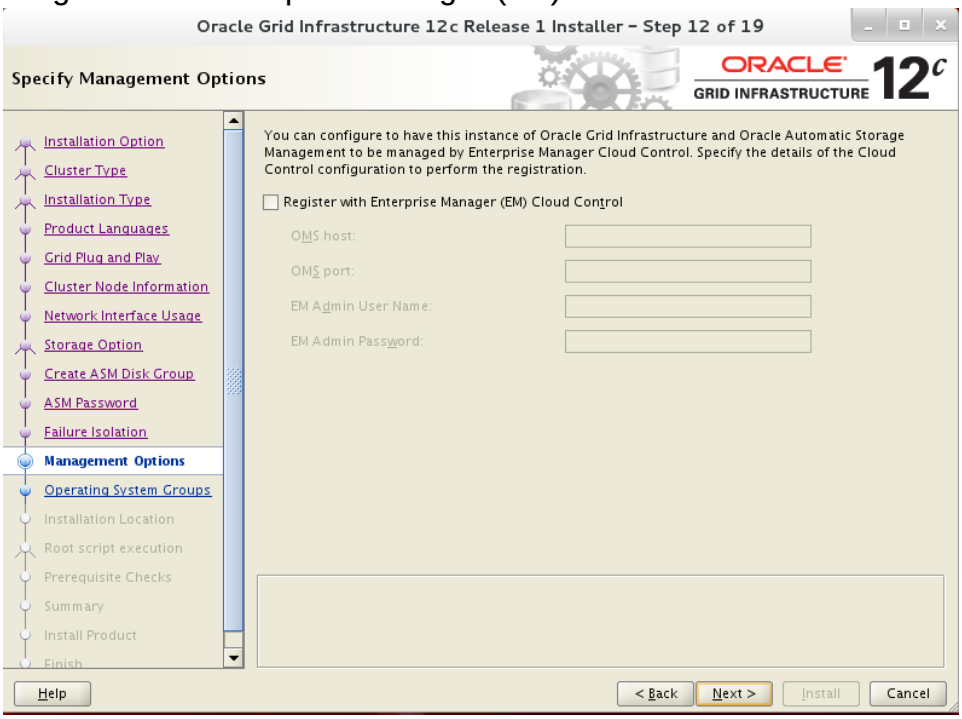
On the 10th screen select the second option, “use same passwords for these accounts”, and set the password to `Asm1pass` :



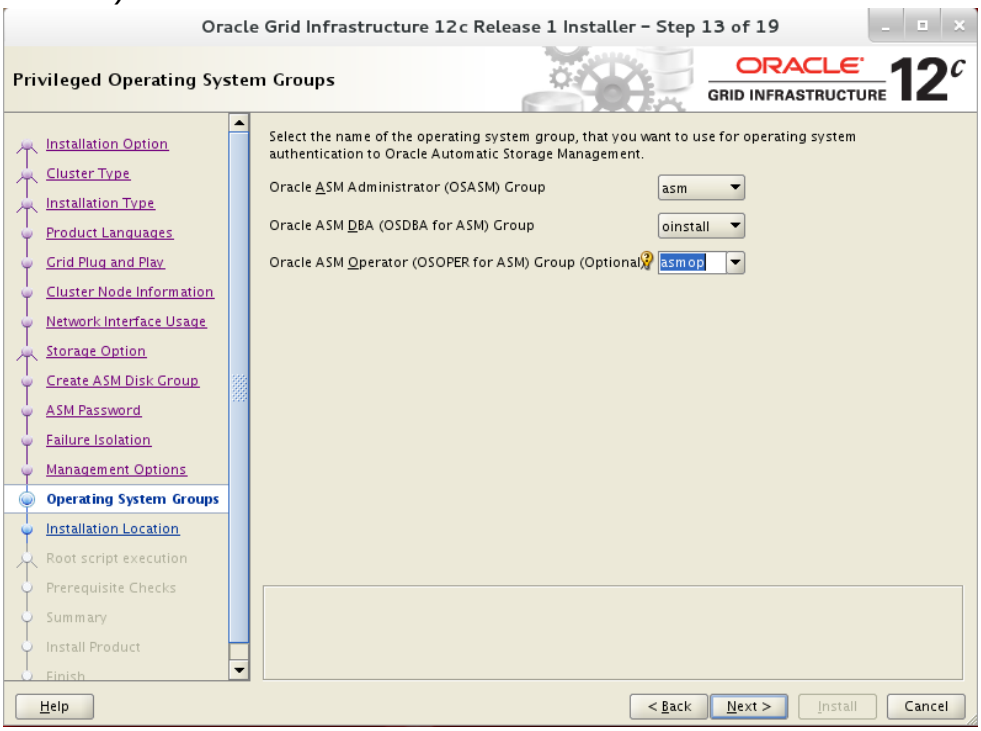
On the 11th screen, don’t change anything – leave the second option selected, “Do not use Intelligent Platform Management Interface (IPMI)”:



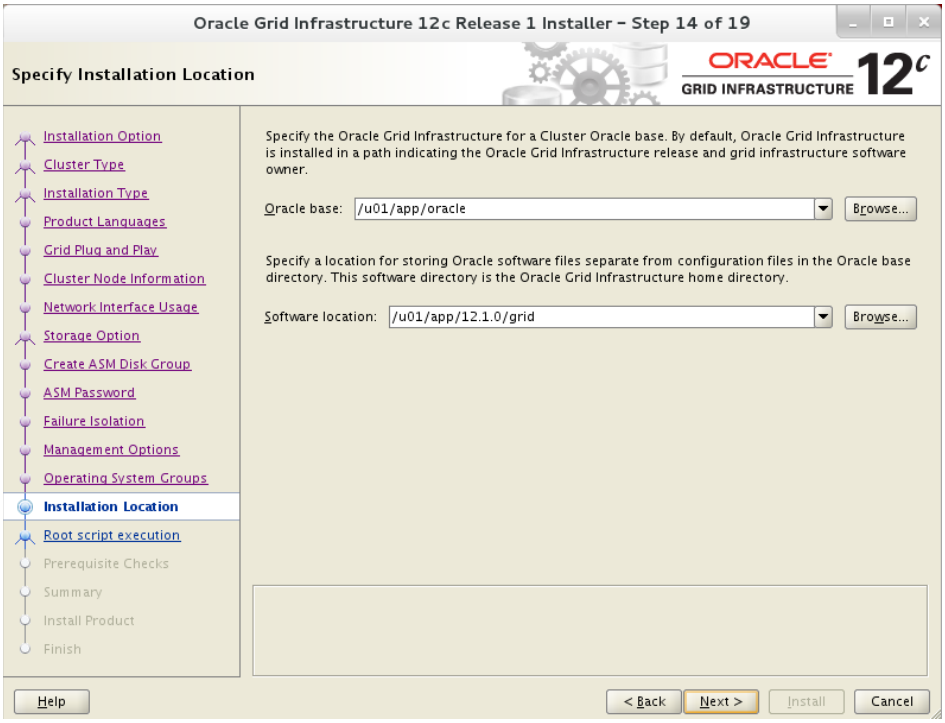
On the 12th screen, do not change anything – leave the checkbox unmarked for "Register with Enterprise Manager (EM) Cloud Control":



On the 13th screen, assign the group `asm` to the first option, OSASM, the group `oinstall` to the second one (OSDBA for ASM), and `asmop` for the last one (OSOPER for ASM):

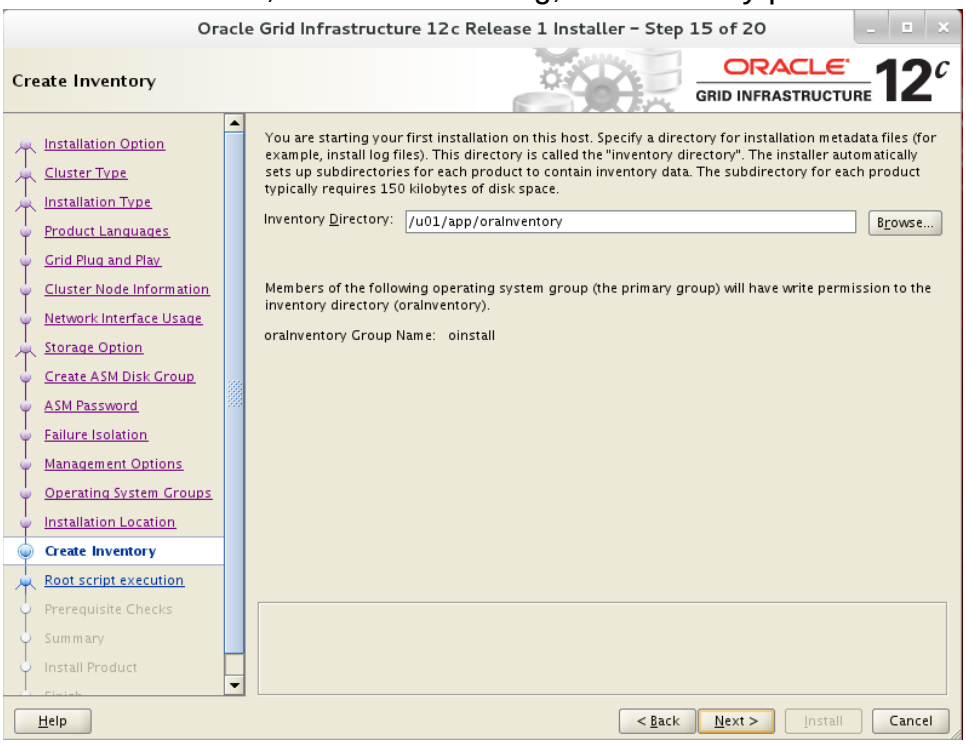


On the 14th screen, make sure the paths begin with `/u01`:

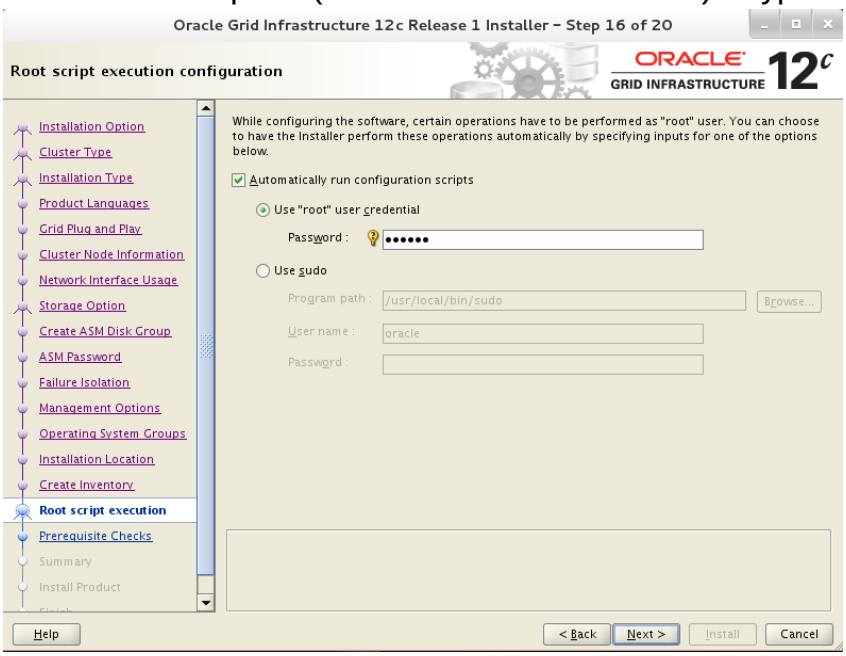


If it doesn't, make sure you didn't skip creating the directory and allowing the `oracle` user full permissions on it.

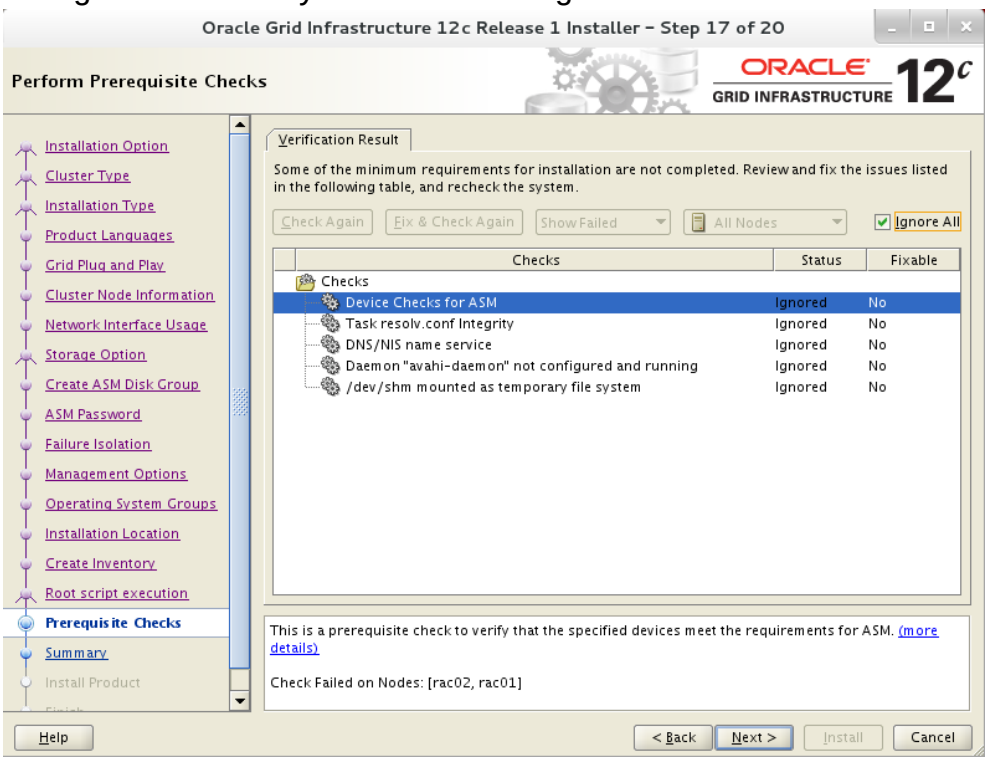
On the 15th screen, don't touch a thing, the inventory path should begin with `/u01`:



On the 16th screen, mark the checkmark ("Automatically run configuration scripts"), and select the first option ("Use root user credentials") – type the root password, `oracle`:

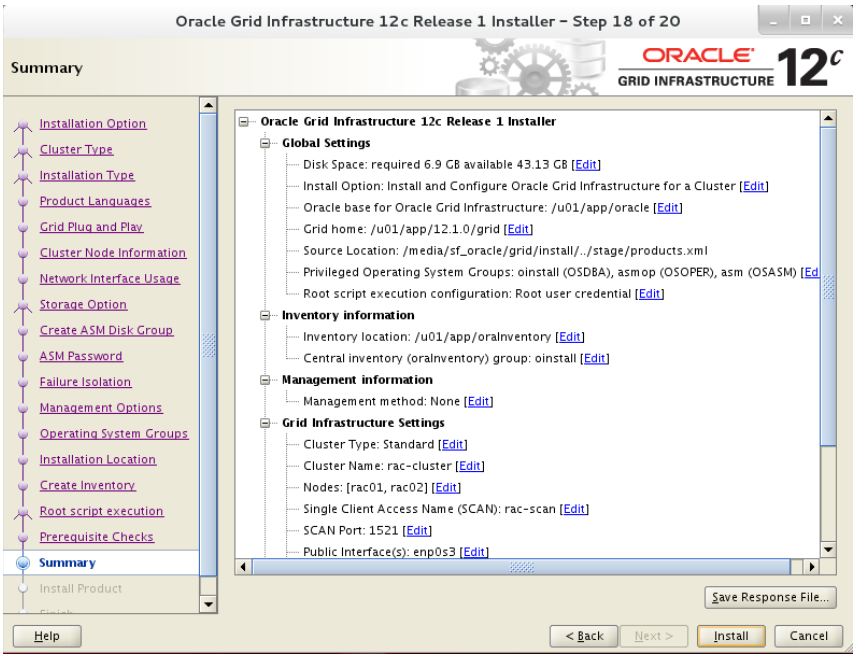


On the 17th screen, you'll see an output of prerequisites analysis.
If you see more than the warning and errors shown here – you've done something wrong. Fix it manually before continuing:



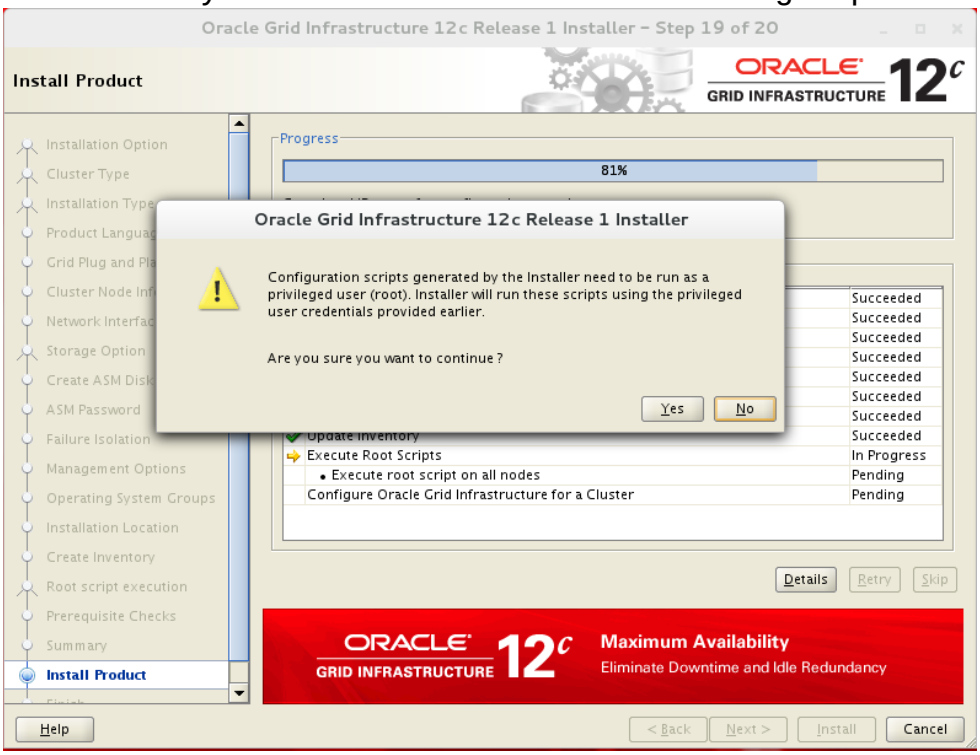
Mark the ignore all, and hit "next" (confirm you want to continue by hitting "yes").

Choose Install:

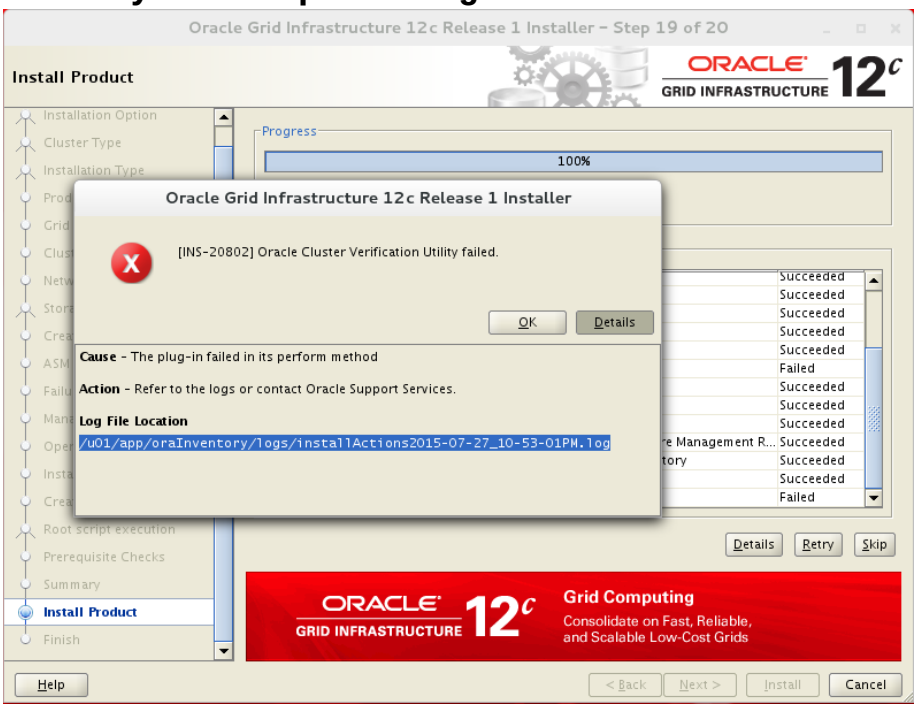


And wait for the installation to finish.

In the middle you'll have to hit "Yes" to confirm running scripts as root:

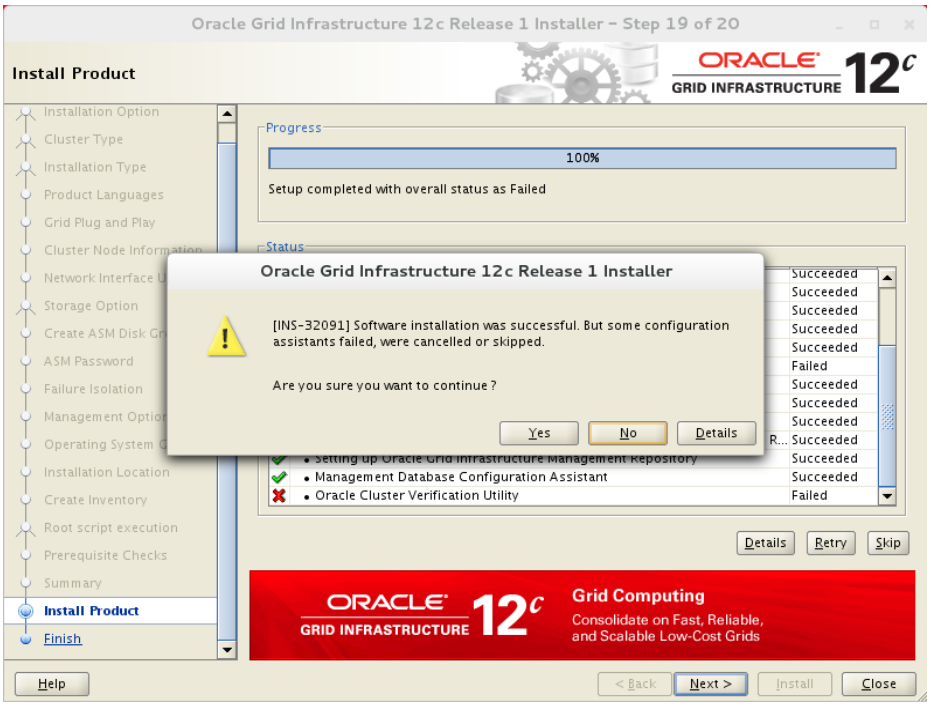


NOTE – you are expected to get an error:



This error appears because we do not use a DNS for the SCAN addresses. YOU CAN IGNOR IT.

After hitting the OK on the error screen, hit “Next”.
You’d be asked to confirm it, hit “Yes”:



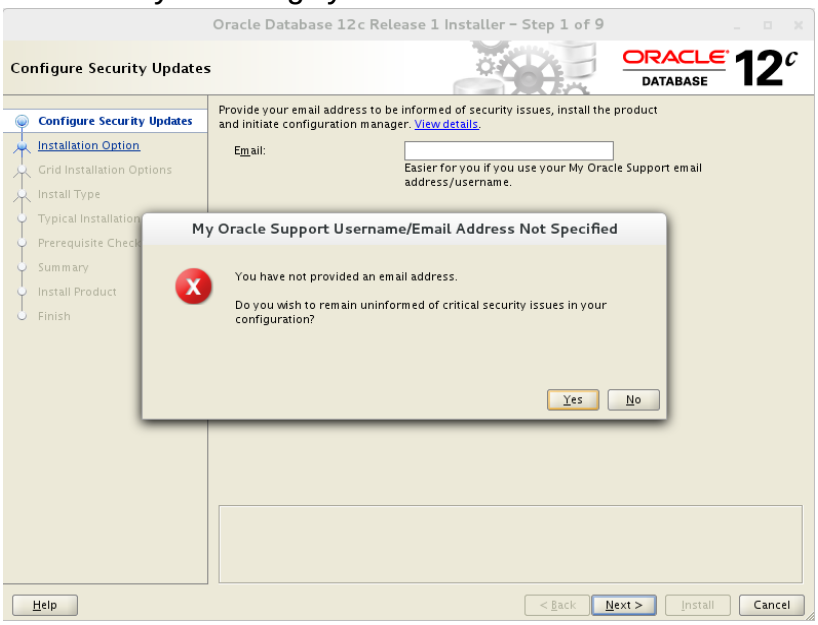
Close the installer – You now have a happy and healthy CRS on your virtual machines.

Database Installation

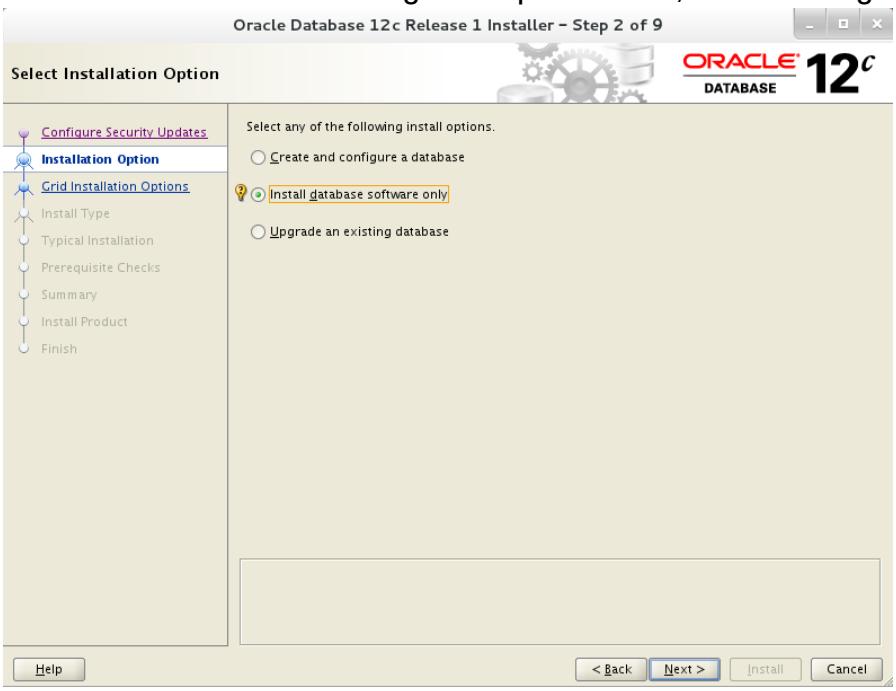
Within a terminal that runs as oracle, execute:

```
cd /media/sf_oracle/database/
./runInstaller
```

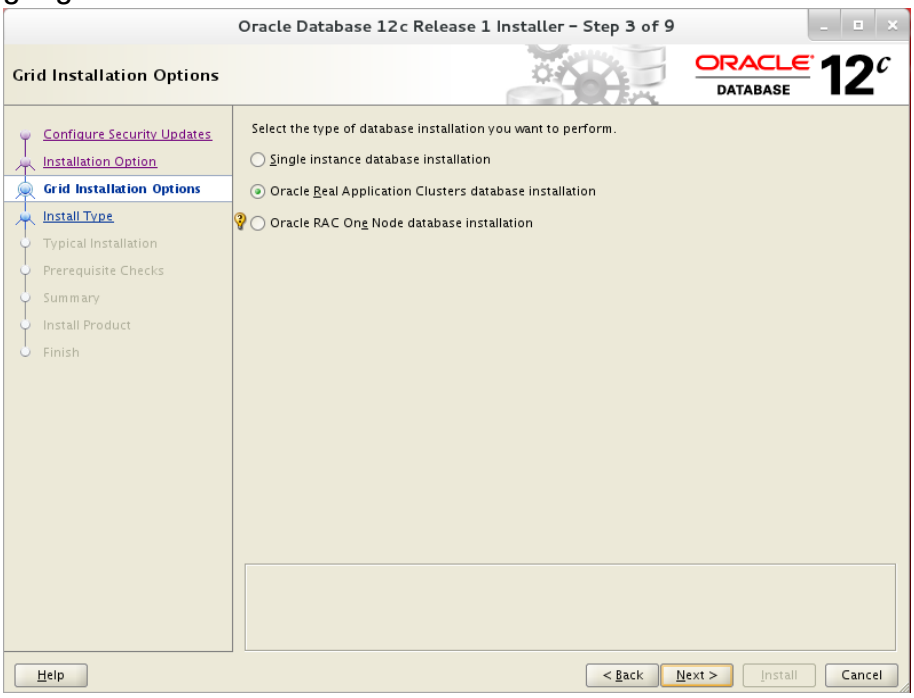
On the first screen, unmark the checkbox for the email, hit next, and confirm this selection by choosing “yes”:



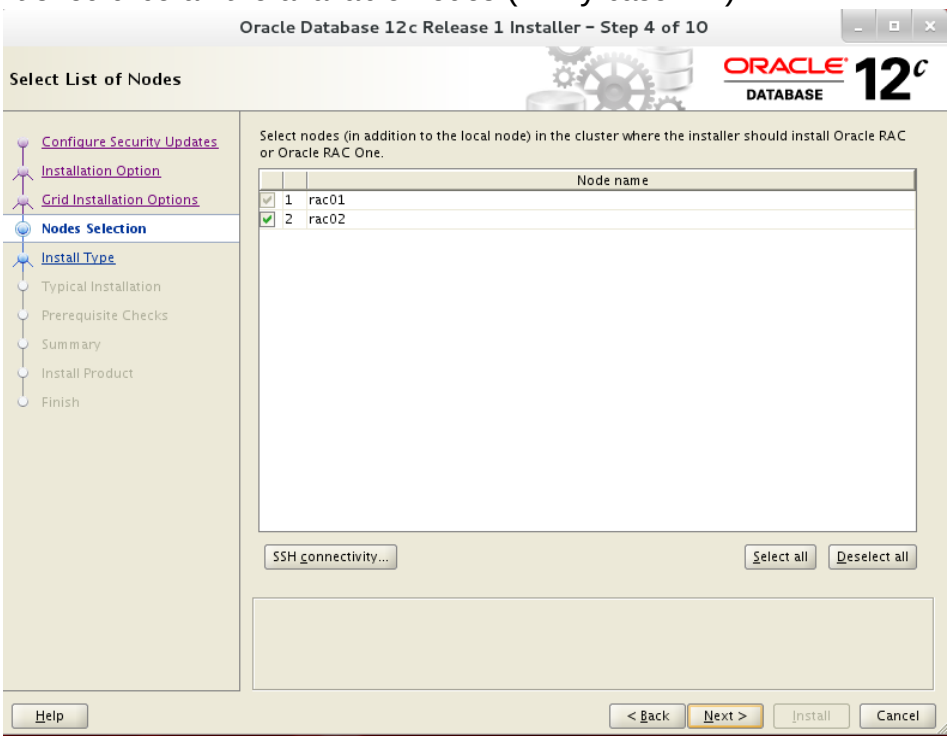
On the second screen, choose the second option “Install database software only”. This is for us not to use the general prebuilt DB, but to configure it ourselves:



On the 3rd screen, make sure the second option is selected – “Oracle Real Application Clusters database installation”.
If it is disabled – something went wrong with the previous configurations and should be googled and fixed.

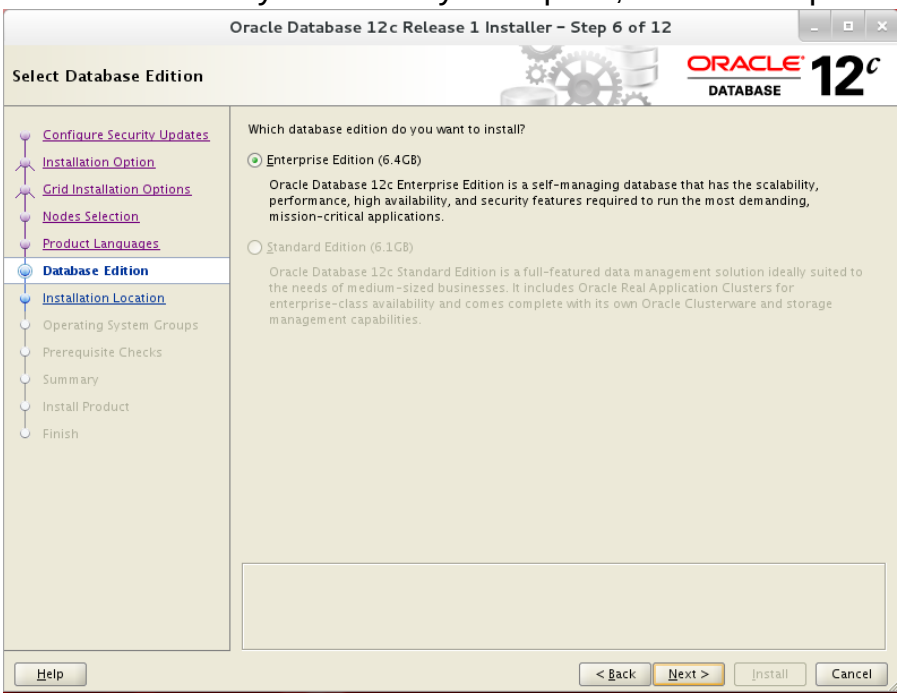


On the 4th screen, select the nodes you want the software to be installed on.
It should be all the available nodes (in my case – 2):

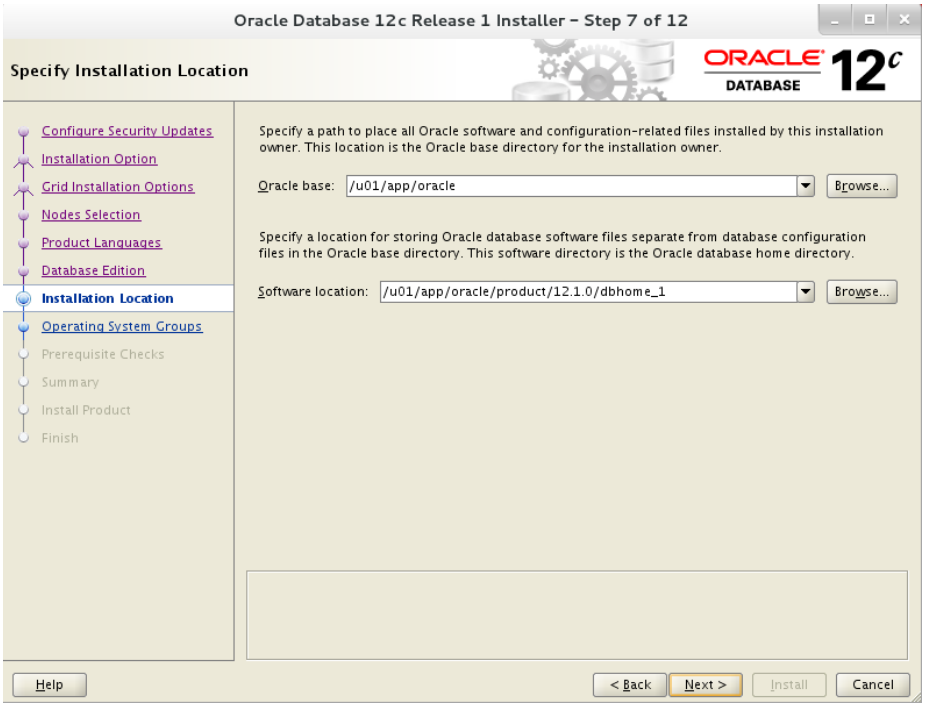


On the 5th screen, Product language selection (not shown here), select English (which was hopefully already selected for you) and hit next.

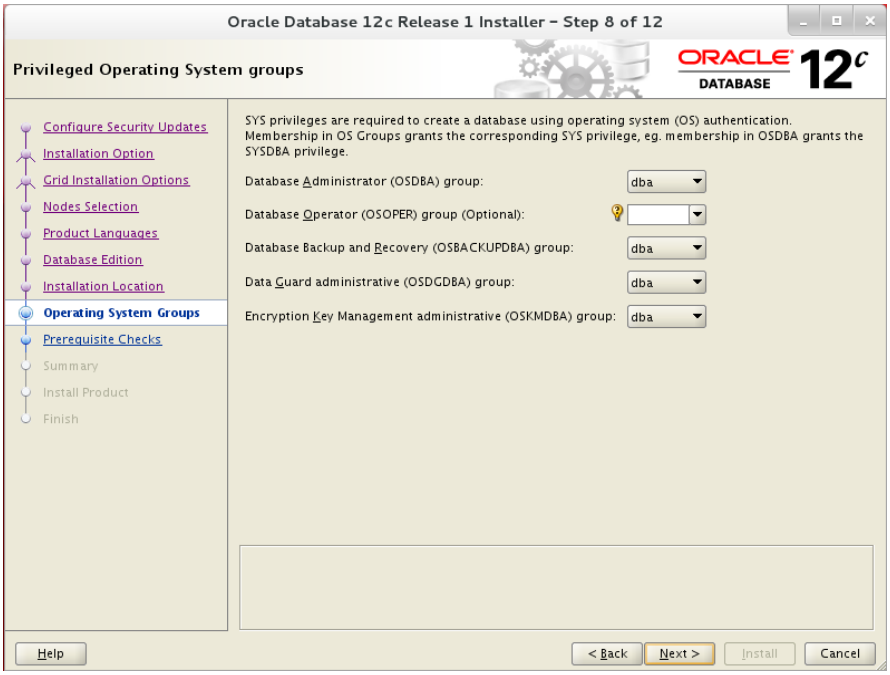
On the 6th screen you have only one option, Oracle Enterprise Server, so hit next:



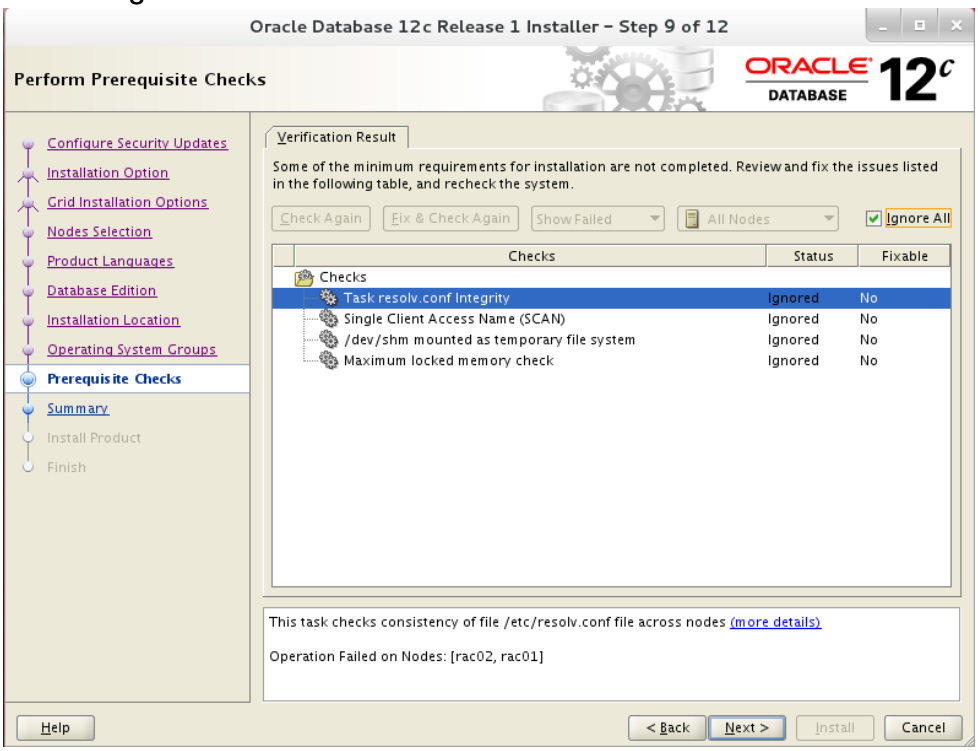
On the 7th screen, make sure all the locations begin with `/u01` as done for the grid. If it is not already filled, there is a problem with the directory existence or permissions – fix it and continue:



On the 8th screen, make sure dba is the group for all the required options:

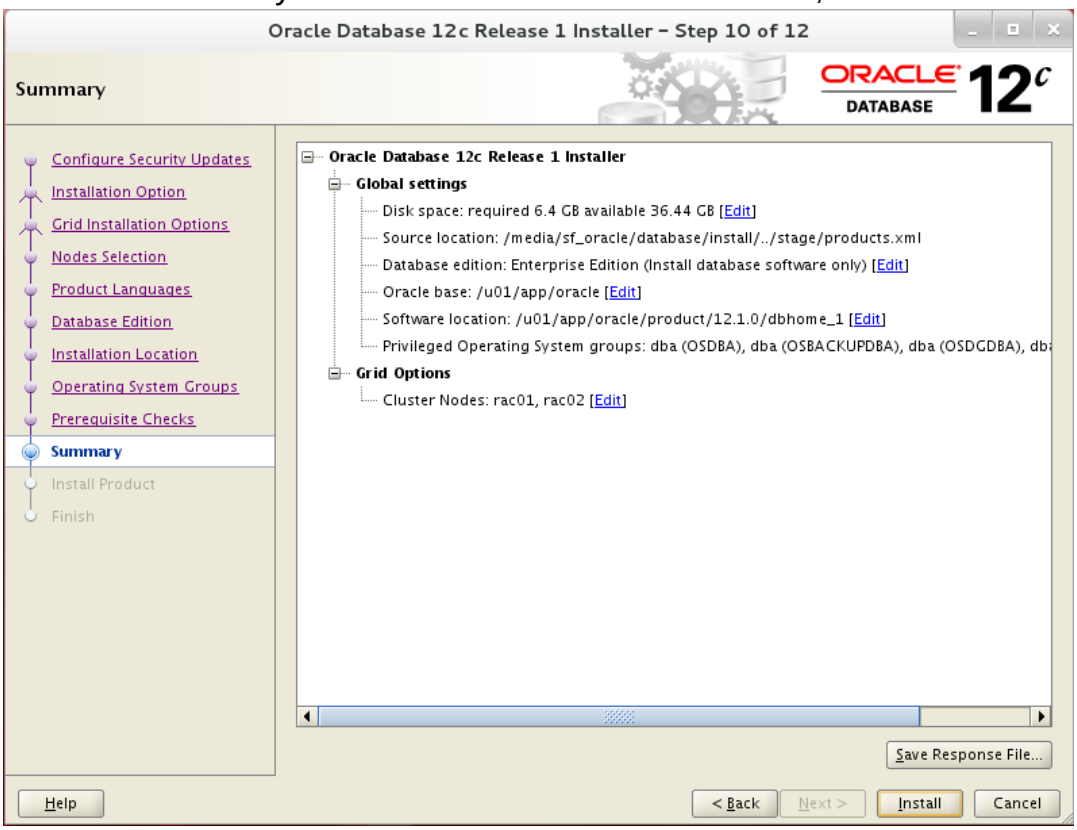


On the 9th screen you'll get a report of the prerequisites checks.
Make sure you have exactly what shown here. If you have anything else in addition – you've done something wrong and it should be fixed.
Hit the "ignore all" as before:

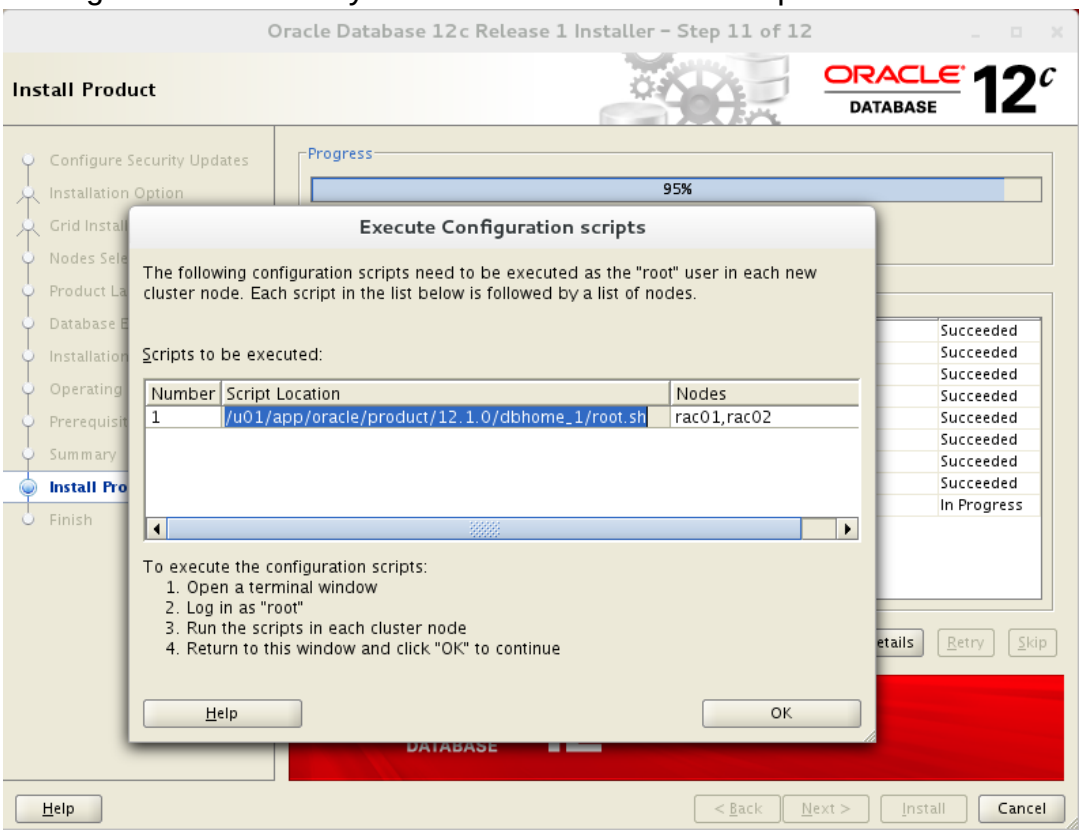


You will have to confirm it, as before, after hitting the "Next".

On the 10th screen you confirm the installation. Hit Install, and the installation will start:



Through the installation you'll be asked to run root scripts:



Don't hit OK before running the scripts.

On both machines, login as `root` in a terminal and execute the following script file:
`/u01/app/oracle/product/12.1.0/dbhome_1/root.sh`

The script requires you to hit "enter" once while it runs.

After running the script on both machines, select "OK", and close the installer.

Congratulations, you have the database software installed.

DBCA

After installing the software, it is time for creating the database instance.

Configure the ORACLE_HOME for the database. With my setting it should be:

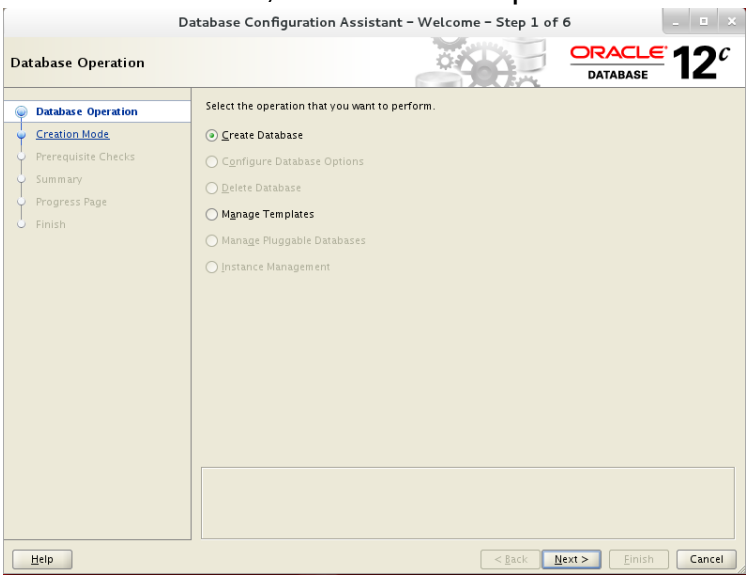
```
export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
export PATH=$ORACLE_HOME/bin:$PATH
```

Start dbca:

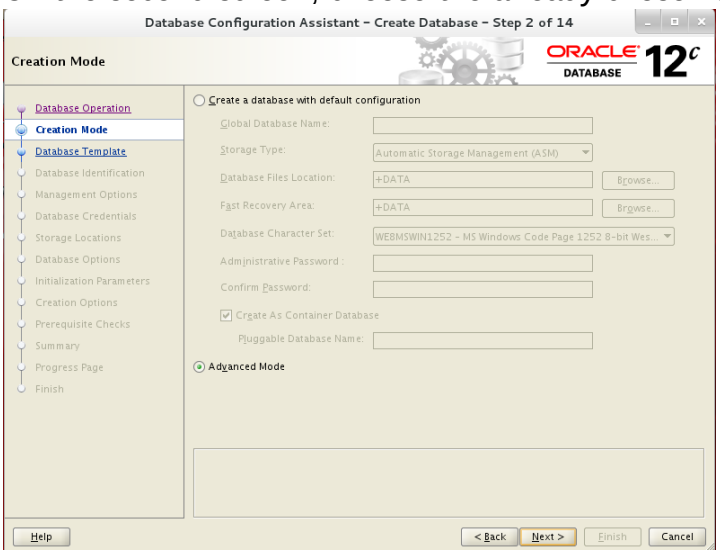
```
dbca
```

On the following screens – you can feel free to change the configuration I use by the features you want to, or don't want to, play with.

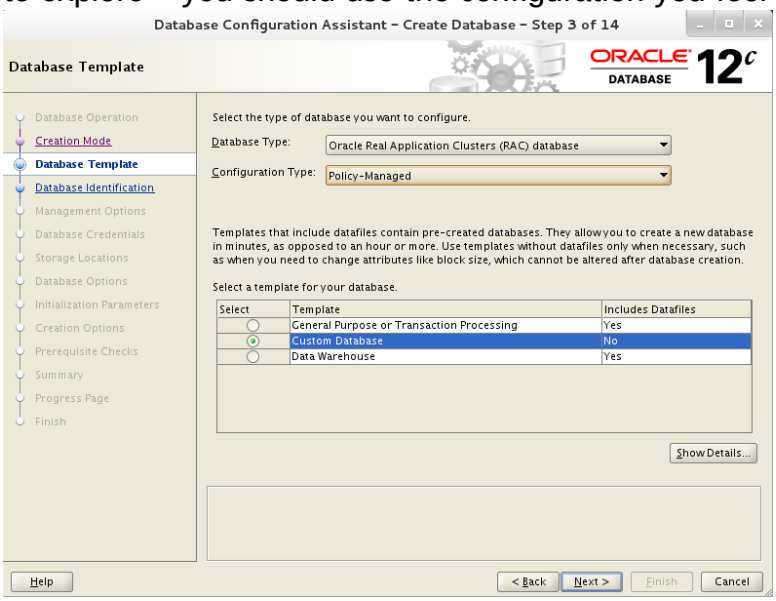
On the first screen, select the first option "Create Database":



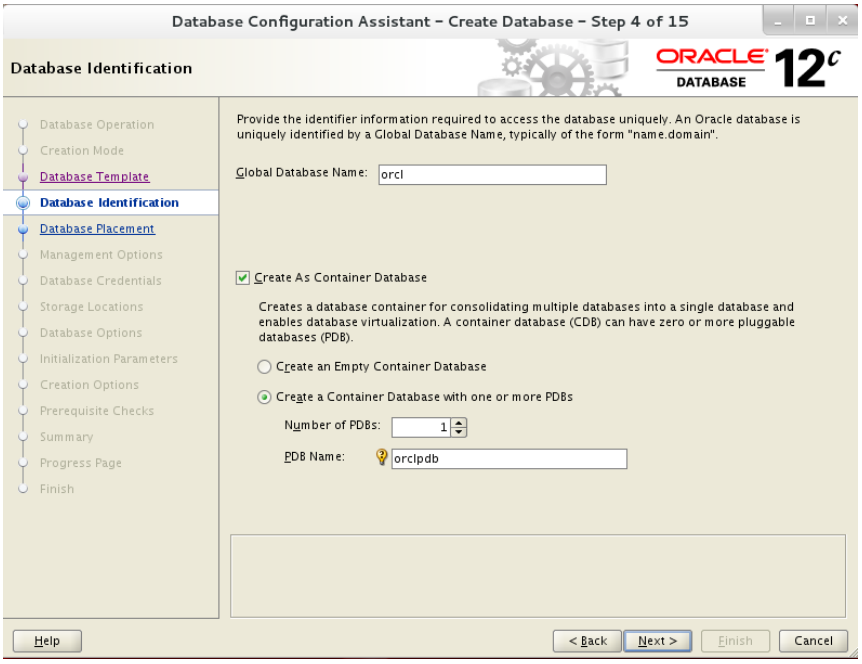
On the second screen, choose the already chosen second option – "Advanced Mode":



On the third screen – choose the second option – “Custom Database”.
Notice the configuration type – I use “Policy-Managed” since it is a new feature I want to explore – you should use the configuration you feel more comfortable using:

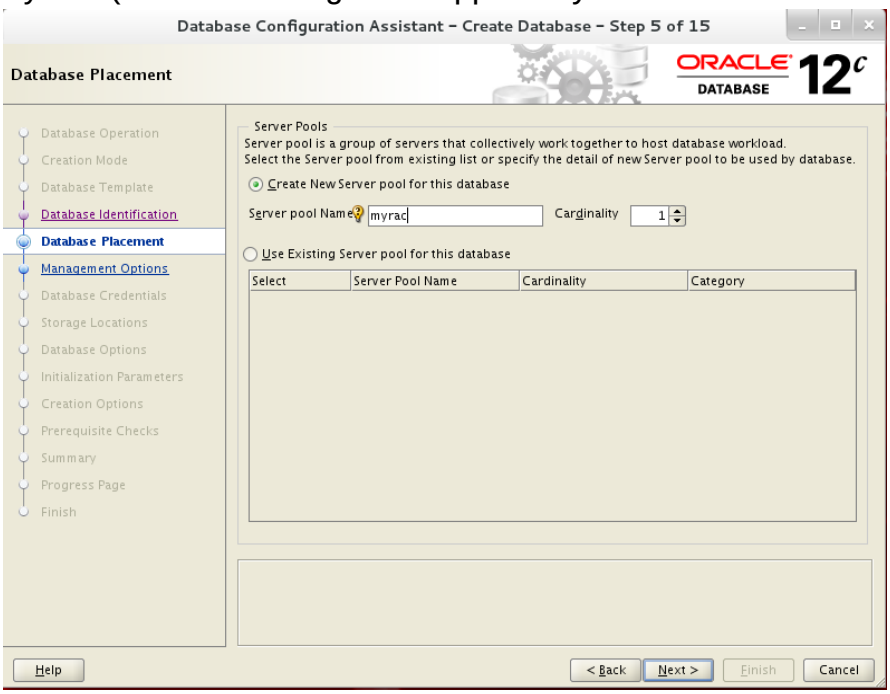


On the next screen, screen 4, choose the Global database name.
I chose `orcl`.
In addition, you can chose if you want to configure it as a PDB (pluggable DB, a new feature) – Note it is a complex configuration that might give you a headache later.
Since I’m exploring – I configured it and named it `orclpdb`, you don’t have to:



the name of each instance will be `orcl_#`, where # is a number of the instance.

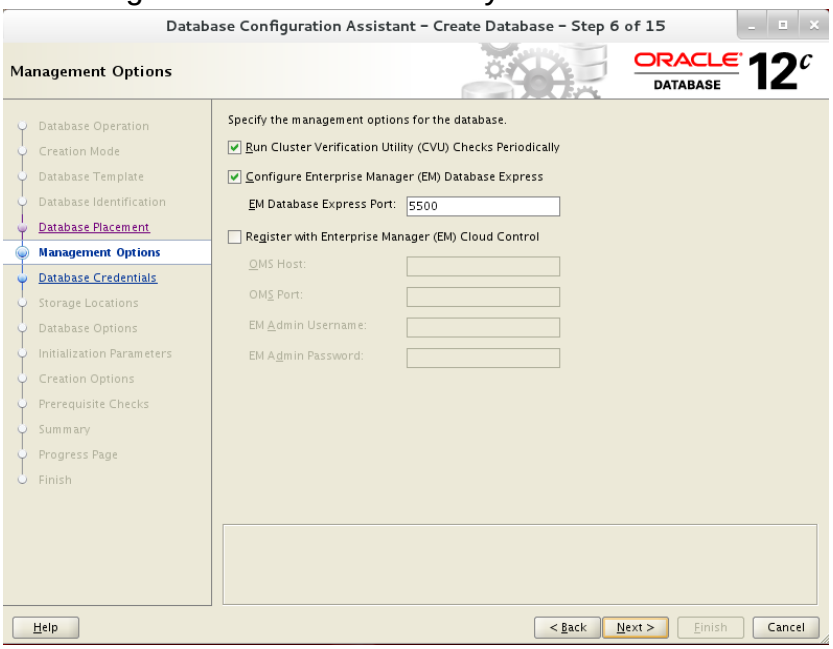
Next, we configure a server pool (the screen explains well what it is..) I called mine `myrac` (This screen might not appear if you didn't choose to create a Container DB):



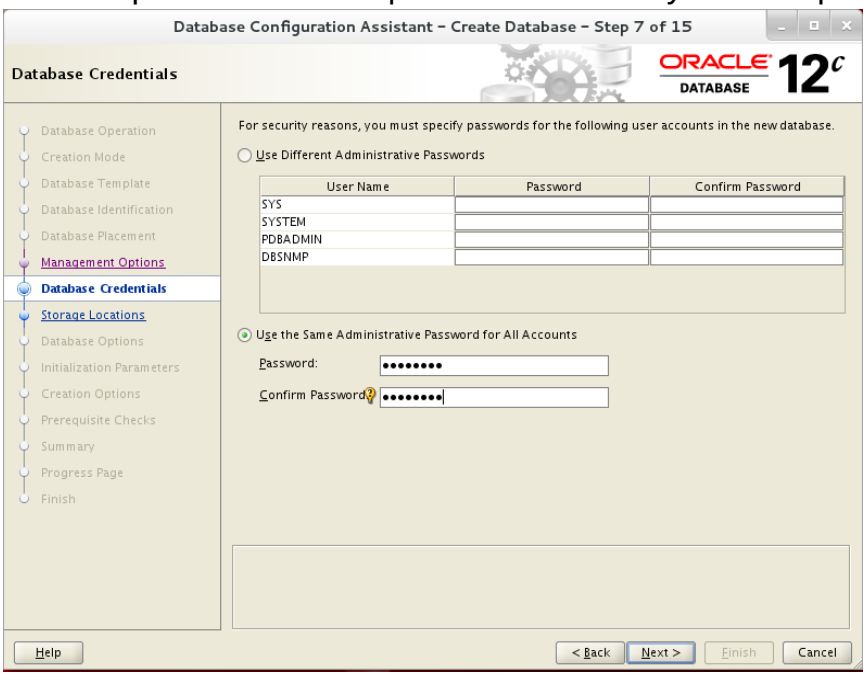
If you want the RAC to run on both nodes immediately after the installation is complete, change the cardinality to 2.

Anyway, since it is a new feature, later I will show how to change it.

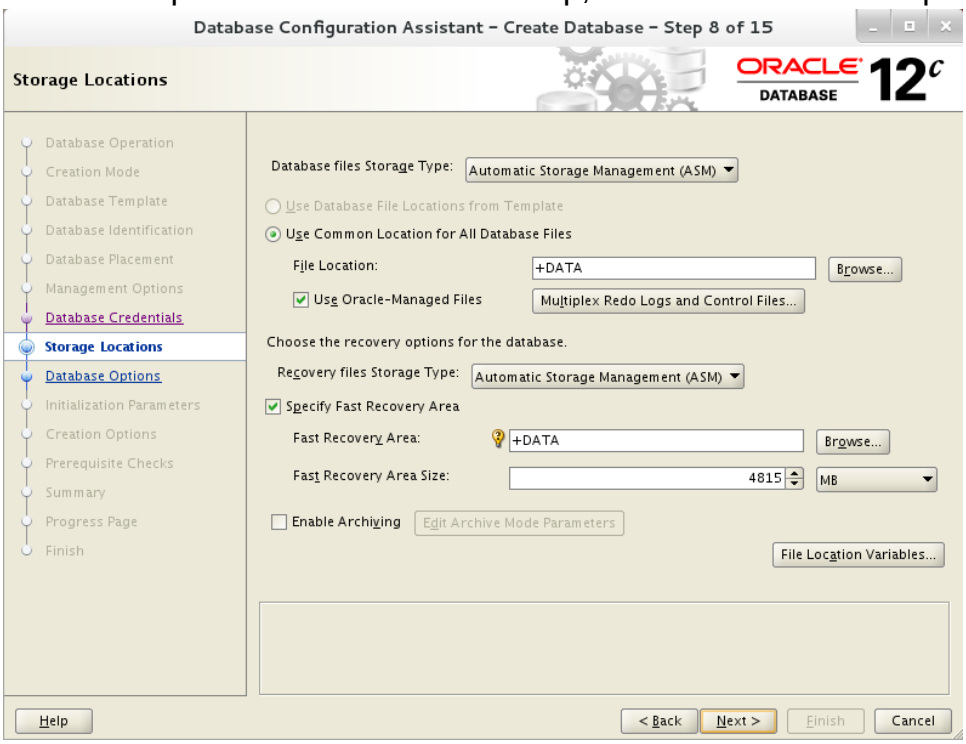
Next, on the 6th screen, we decide if we want cluster verifications checks to run periodically, to connect to an EM server, or to configure an EM express – don't change the configuration that was set for you:



Next, select “Use the Same Administrative Password for All Accounts”.
I set the password to `Ora1pass` for consistency with the previous chosen one:

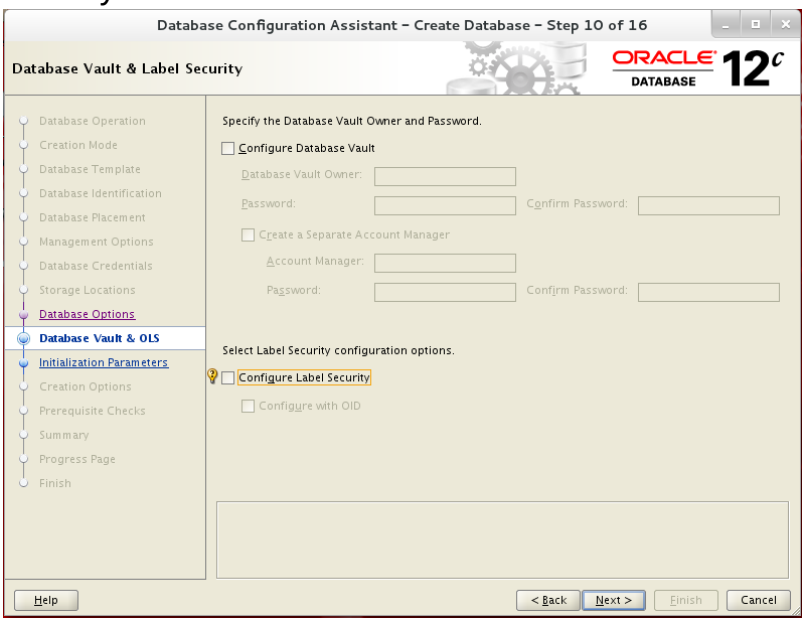


On the 8th screen, leave the configurations to use ASM as is.
The only thing you’d might want to change is to not use “Fast Recovery Area” by unmarking the checkmark before it.
I’m in an exploration mood for this setup, therefore – I left this option on:

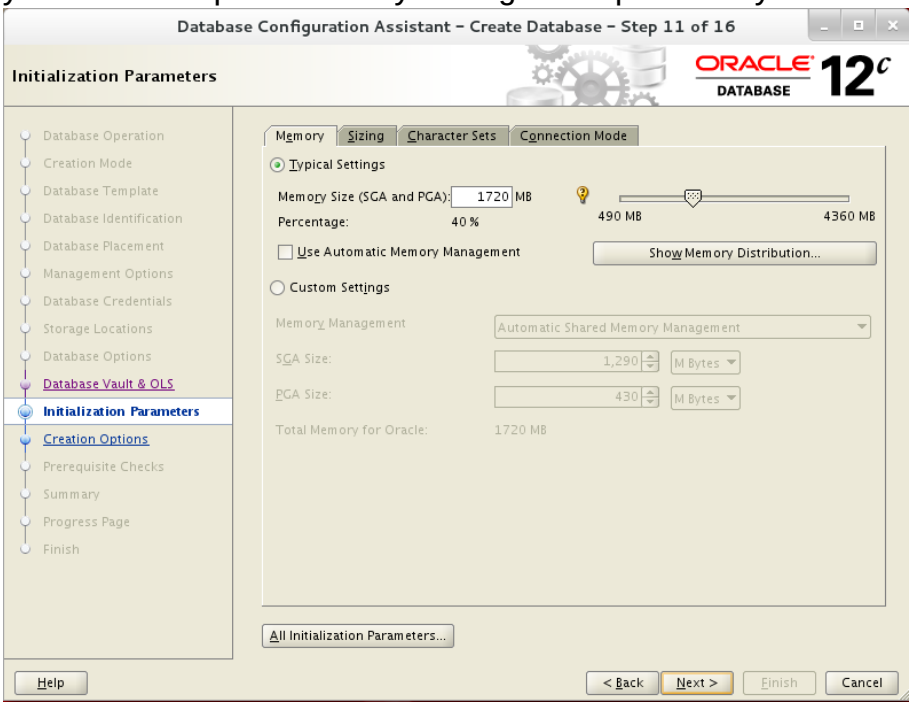


On the next screen, you have no option to choose, therefore I didn't add a screenshot. You are installing all the features in this database. Hit next…

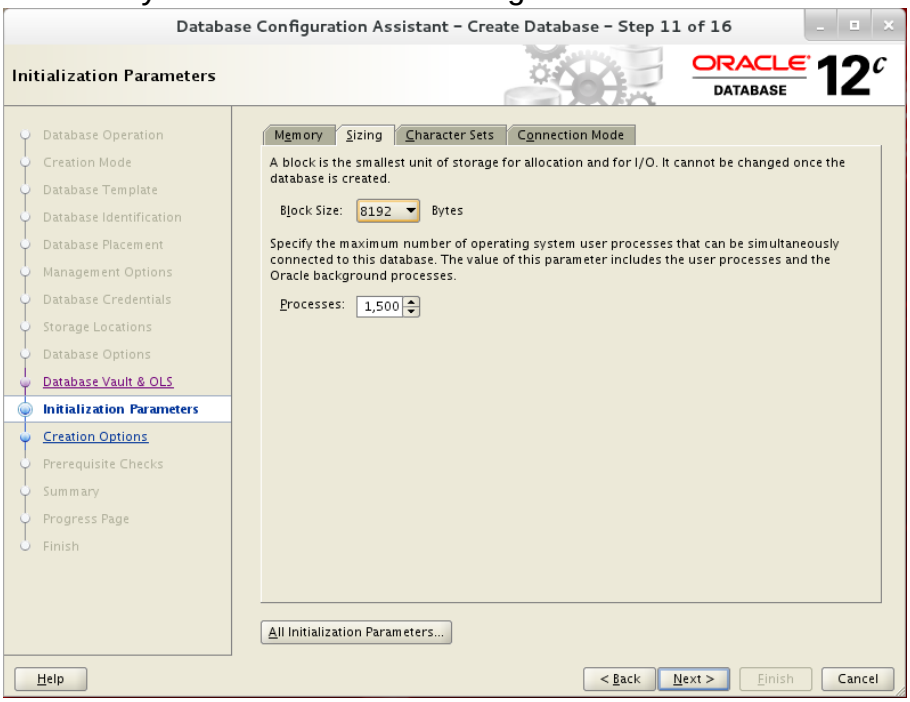
On the 10th screen, for your on good, unmark the two checkmarks.
Unless you know what you're doing, or want to explore it specifically, Vault or Label Security are both are a headache…:



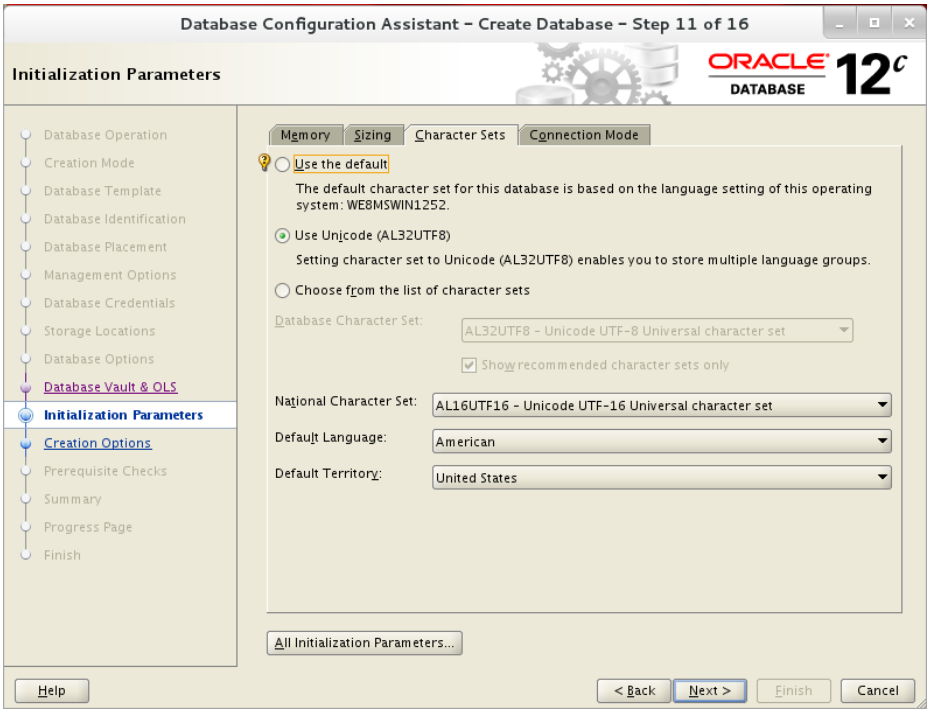
On the 11th screen you set the database parameters.
I don't recommend changing the memory distribution on installation, but later, when you want to explore memory management particularly:



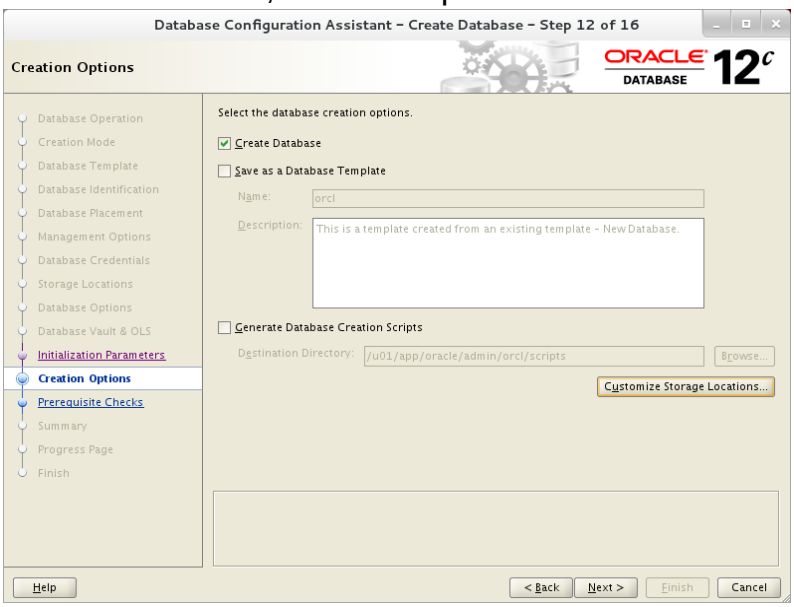
On the sizing menu, I usually increase the number of processes to 1500, just because 300 is very low... You can also change the DB default block size:



Change the character set to the second option, AL32UTF8:

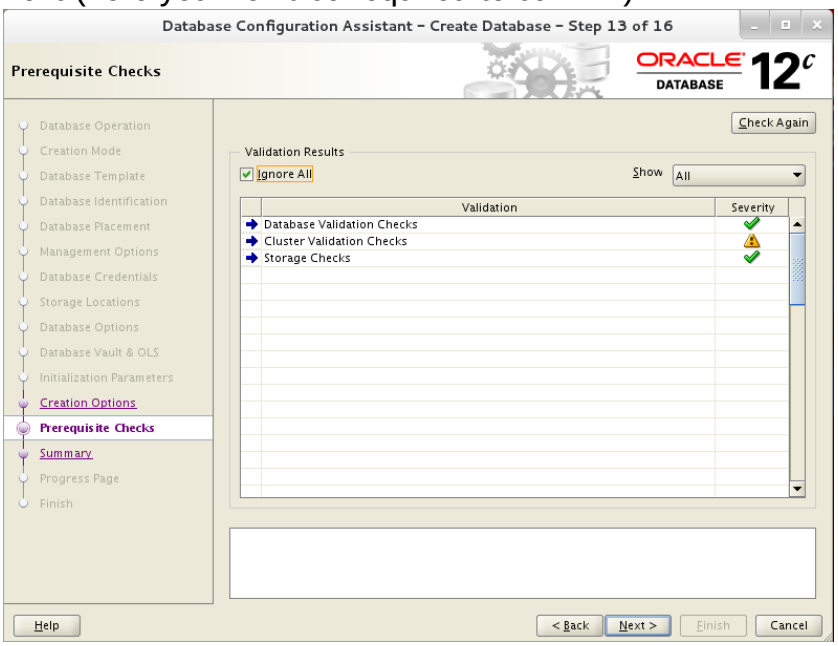


On the 12th screen, leave the option of “Create Database” selected:

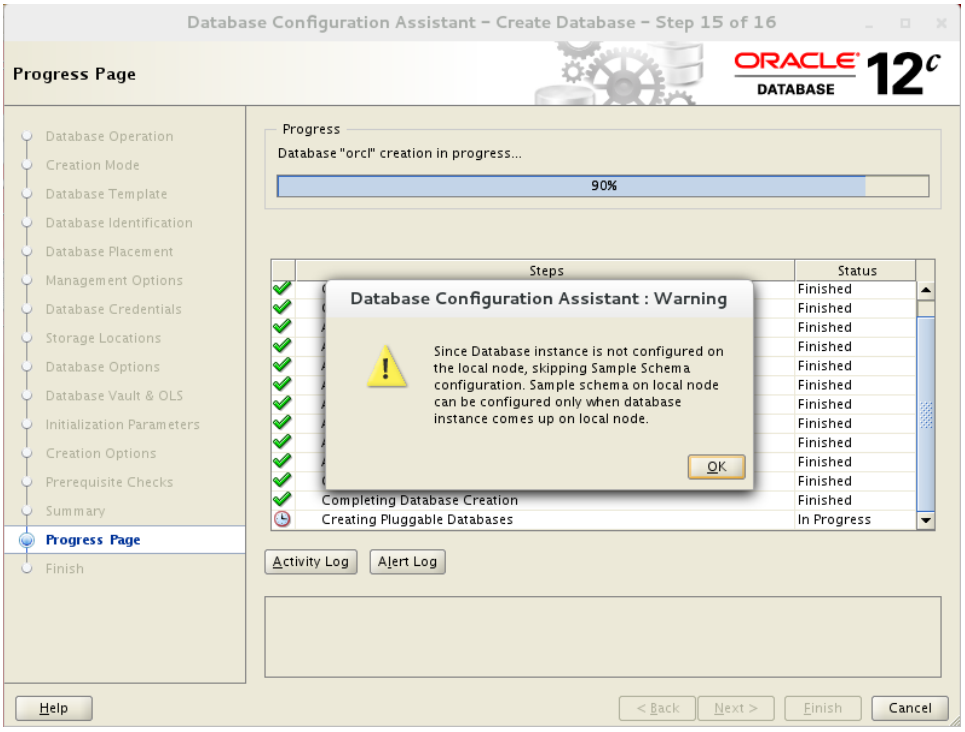


Although usually I tend to “Customize Storage Locations…” and increase the file sizes to reasonable sizes for reducing installation time (less extents allocations that requires increasing the datafile sizes), this time we will just Hit “Next” and wait…

The prerequisites check should find the same issues as before, select the “Ignore All” (on this installation Oracle moved this option to the other side of the screen…) and hit next (here you won’t be required to confirm):

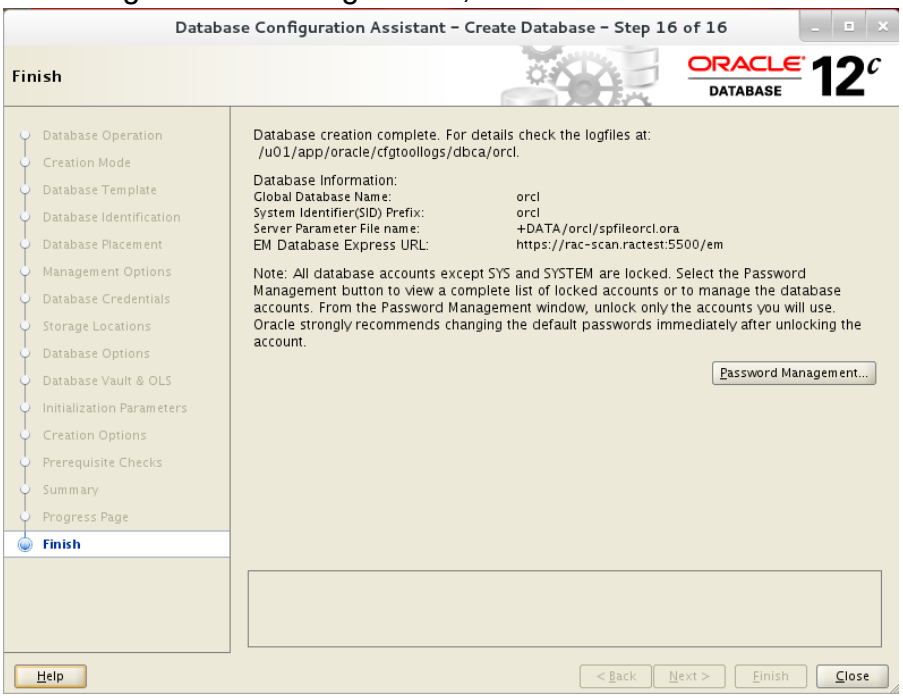


On the next screen – select “finish”, and DBCA will start…
Wait patiently until you get the following screen with the following message:



Hit OK (this screen might not appear if the node that runs the installer has an Oracle instance running – since I used cardinality of 1, only one instance was created, and the clusterware chose to run it on the other node).

You will get the following screen, hit the “close”:



Congrats!
Oracle DB is installed on the RAC!

In my case, the database was configured to run only on one node.
It started on node number two.
When turning the node that runs the instance off, the instance automatically starts on an available node.
As mentioned before, the reason for that is the cardinality given in the DBCA.

For changing the cardinality, within a terminal, run
`srvctl config srvpool`
This will output the server pools configured, make sure yours is `myrac` and run:
`srvctl modify srvpool -g myrac -u 2`

In a few minutes you'll have a two node RAC.

Run
`srvctl status database -d orcl`
to see which instance runs on which node.

Post installation

Although these are not perfect configuration scripts, it will allow you to connect and manage the database instances.
Notice that with the new Oracle configurations – the instance id is not constant, therefore the script is a bit more complex than the hard coded old ones.

If you configured a Pluggable Database (PDB), in a terminal, logged in as Oracle, on both machines, run:

```
vi /u01/app/oracle/product/12.1.0/dbhome_1/network/admin/tnsnames.ora
```

And append into the file the following few lines:

```
ORCLPDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = rac-scan)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orclpdb)
    )
  )
```

Afterwards, even if you did not configure a PDB, execute on both nodes:

```
vi .bashrc
```

And at the end of the file add the following few lines

```
#(c) ALL RIGHTS RESERVED TO ROEE EBENSTEIN
pmon=ora_pmon_
inst="$(ps -e | grep $pmon)"
ar=($inst)
arlen=${#ar[@]}
if [ "$arlen" -gt 2 ]
then
        inst=${ar[3]:${#pmon}}
        export ORACLE_SID=$inst
else
        export ORACLE_SID=NOTSET
fi
export GRID_HOME=/u01/app/12.1.0/grid
export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
export PATH=$ORACLE_HOME/bin:$GRID_HOME/bin:$PATH
```

If you configured a PDB you'd might also want to add this line to the `.bashrc` file (read documentation about what it means...):

```
export TWO_TASK=orclpdb
```

Save and exit.
Starting at the next logon the environment will be set for you to connect to the DB

immediately if the instance have already started. If the instance have not started you will have to set the `ORACLE_SID` environment variable manually.
You can check if the `ORACLE_SID` was configured by executing:

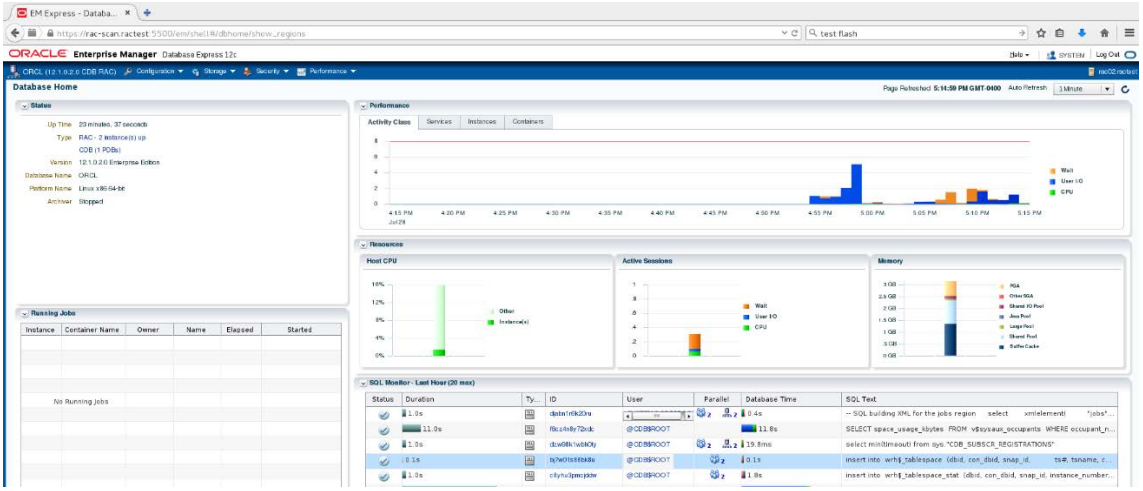
```
echo $ORACLE_SID
```

Notice that if you'd like to use the ASM, you'll have to run:

```
export ORACLE_HOME=/u01/app/12.1.0/grid
export PATH=$ORACLE_HOME/bin:$PATH
export ORACLE_SID=+ASM1
```

While modifying the instance name to the right one (you can learn from the above script how to figure out your ASM instance name ☺).

In addition, you would probably like to install flash through your browser, so you could browse to the configured EM at:
https://rac-scan.ractest:5500/em



I recommend strongly installing a third machine, which could even be a windows one, and connect it only to the public network. This simulate a workstation client. Since the RAC machines might get busy and not as responsive as expected (after all, it's a VM), the user experience on these machines would usually be bad.

ENJOY,
Roee Ebenstein
evenro at yahoo dot com , for any questions or comments regarding this document ☺