DIY - Build your Own Cloud

Bhavin Hingu
SME – Oracle (Database / Clustering / Cloud Control)
www.OracleArchitect.com

Hello friends, I am sharing my experience of building robust fully functional personal Cloud System at home that is fully accessible and manageable remotely. After successful completion of design/architecture and implementation, I am able to spin up Linux guest VMs from the gold image and access them remotely as needed without any issues. Further, I am using the Automation to create the guest VMs and install the Oracle software as part of the single build. Fun stuff...!!!

Technical knowledge and hands on experience in below area is desired:

VM software (for e.g., Oracle VM VirtualBox)
Network (basic to mid-level)
Storage – NAS (basic knowledge)
Linux administration (mid to advanced level)
Programming/scripting knowledge.

The Goal:

My goal was to build robust and fully functional Cloud at home which can be accessible remotely any time I want. I needed to create multiple guest VMs to install and run the Oracle Enterprise software for practice and learning purposes. I also needed to back them up periodically. Since the guest VMs are VDI files on VM host, backing them up, restoring them or migrating them to different VM host is much easier as compared to the set up that I had earlier where I was using the physical boxes in my lab. I also thought of an option to lease Virtual Private Cloud from AWS but I chose BWS (Bhavin Web Services. .hehe) over AWS just because I can proudly say I use my own homegrown Cloud system ☺.

Design towards the Goal:

VM Host Consideration (Machine where Hypervisor gets installed):

Because I required to create about 12 to 14 guest VMs to run multiple Oracle software installs (like Grid Infrastructure, Oracle RAC Databases, GoldenGate, Oracle EBS, Oracle Utilities, Oracle OEM Cloud Control etc. etc.),
I decided to look for a machine that has at least 32 Cores and 128GB RAM. After doing some research, I decided to buy the Dell Precision T7610 (refurbished) that has 32 cores 128GB RAM with 250GB SSD and 2TB spindle storage HD and most importantly fit in my budget. ☺

Storage Consideration:

To store the guest VMs, I decided to use the 2TB of local storage that came with the machine itself. But I also used my existing Western Digital's MyCloud 24TB NAS Storage

for backing up of my Cloud VMs and other Cloud related files. I created the sharable NFS volume on NAS of about 1TB in size and mounted it on the VM host.

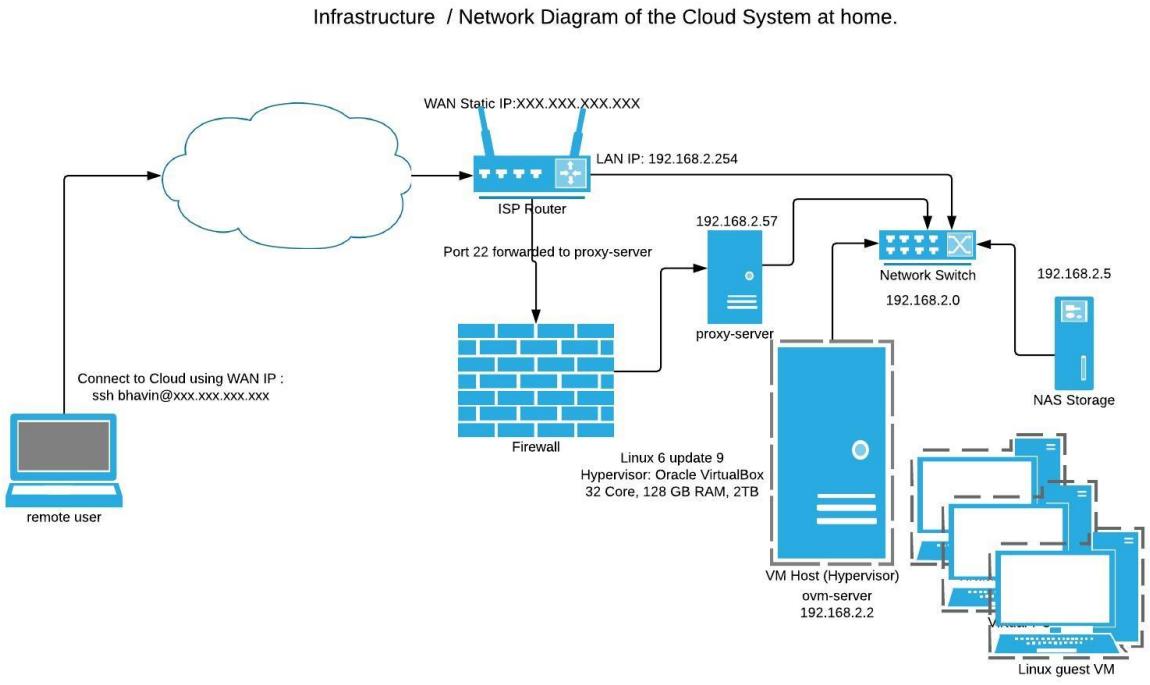
Network Consideration:

Here I required to setup the LAN for NAS storage, VM Host and the Proxy server so that they all can connect to the ISP router. So, I needed the Network Switch for the connectivity,

Proxy Server Consideration:

To access the Cloud system remotely, I simply used the DMZ setup in the ISP supplied router to forward certain ports like ssh/VNC etc. to a proxy server which is nothing but a Linux machine that is connected to the same LAN as VM Hosts, guest VMs and other IOT at home. I wanted to have a server that act as a proxy machine for all the remote connection requests from outside network primarily for the security reason. The need of proxy machine in this setup is not mandatory. One could also configure ISP router to forward the required ports directly to any of the guest VMs or VM Host itself for that matter and would still work.

After considering all of the above points, I came up with the below design / Architecture of my Cloud



Hardware Used:

Network:

Network Switch – 1 (D-Link 24-Port Rackmountable Gigabit Switch)
Network Cables - 5

Cost: roughly $200

Storage:

2TB of Local storage on the Hypervisor / VM Host.
24TB NAS from WD's MyCloud PR4100 for Backup
Cost: $1500

VM Server:

Dell Precision T7610 Workstation (32 Core/ 128GB RAM/ 250GB SSD / 2TB spindle HD - refurbished

Cost: $2300

Proxy Server:

Dell 8GB RAM /250GB HD
Cost: $400

Total Cost to build the Cloud Infrastructure: $4500

Software Used:

- Oracle Linux 6 update 9 (downloaded on DVD from otn.oracle.com)
- Oracle VM VirtualBox 5.2
- VNC Viewer (for Xwindows / X Desktop)

Technical Steps / implementation of the Design:

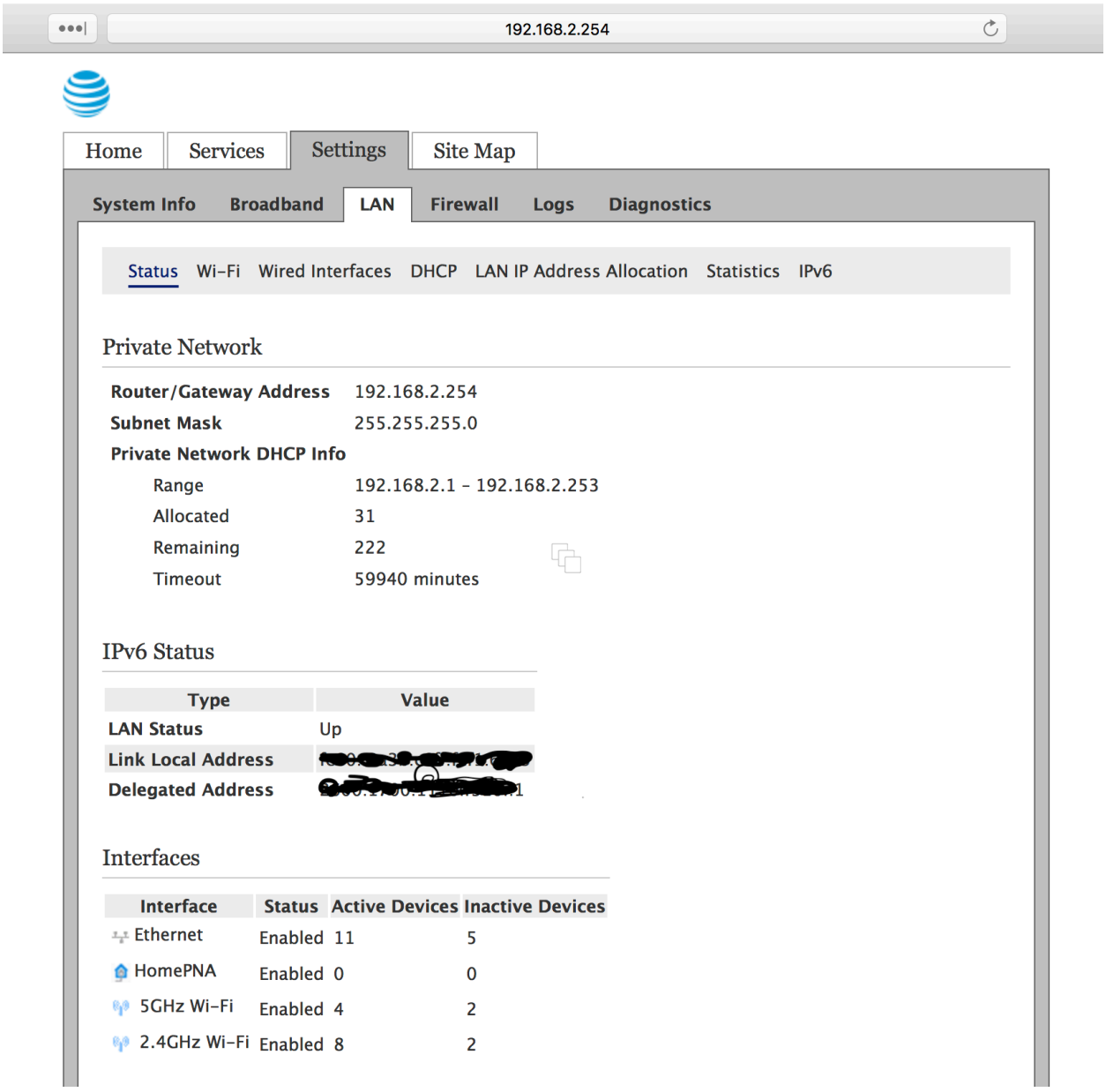
Here are the steps that I had to complete successfully in order to build the Cloud Infrastructure as per the design.

- Update the LAN setting in the ISP Router.
- Install Linux on the VM Host and Proxy Server
- Connect the devices to the network
- Stage the required software/downloads.
- Install Oracle VM VirtualBox Hypervisor on the VM Host.
- Create the Guest VMs.
- Start the Guest VMs

- Setup the Backup Server for these VMs.
- Accessing the Cloud System remotely.

Change the ISP router's default setting:

I decided to start with changing the ISP's default network setting in the router to my own custom setting as shown in the screenshot below. All the subsequent devices that would connect will have a network of 192.168.2.0. I could have used the default router setting and would still work fine.



Install Linux:

Install Linux OS on the machine that is going to be the VM host machine

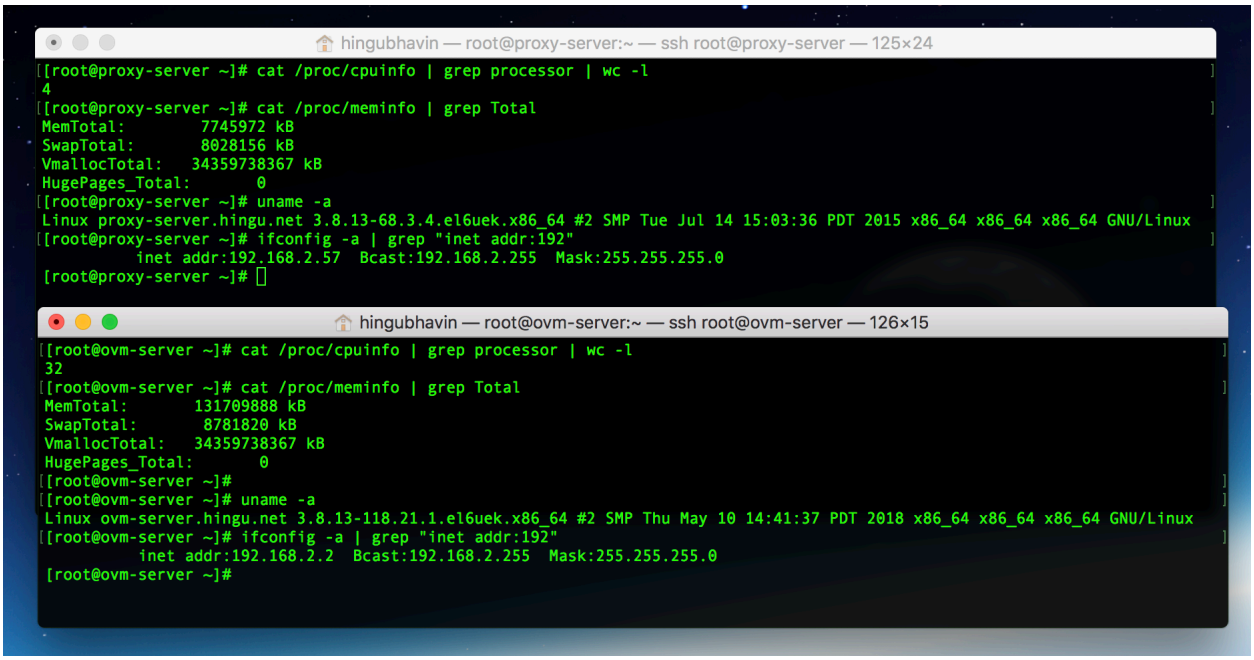
Host Name: ovm-server.hingu.net

eth0: 192.168.2.2/255.255.255.0/192.168.2.254 Gateway

Install Linux OS on Proxy Server:

Hostname: proxy-server.hingu.net

eth0: 192.168.2.57/255.255.255.0/192.168.2.254 Gateway.



Connect the devices to the LAN Network via Switch:

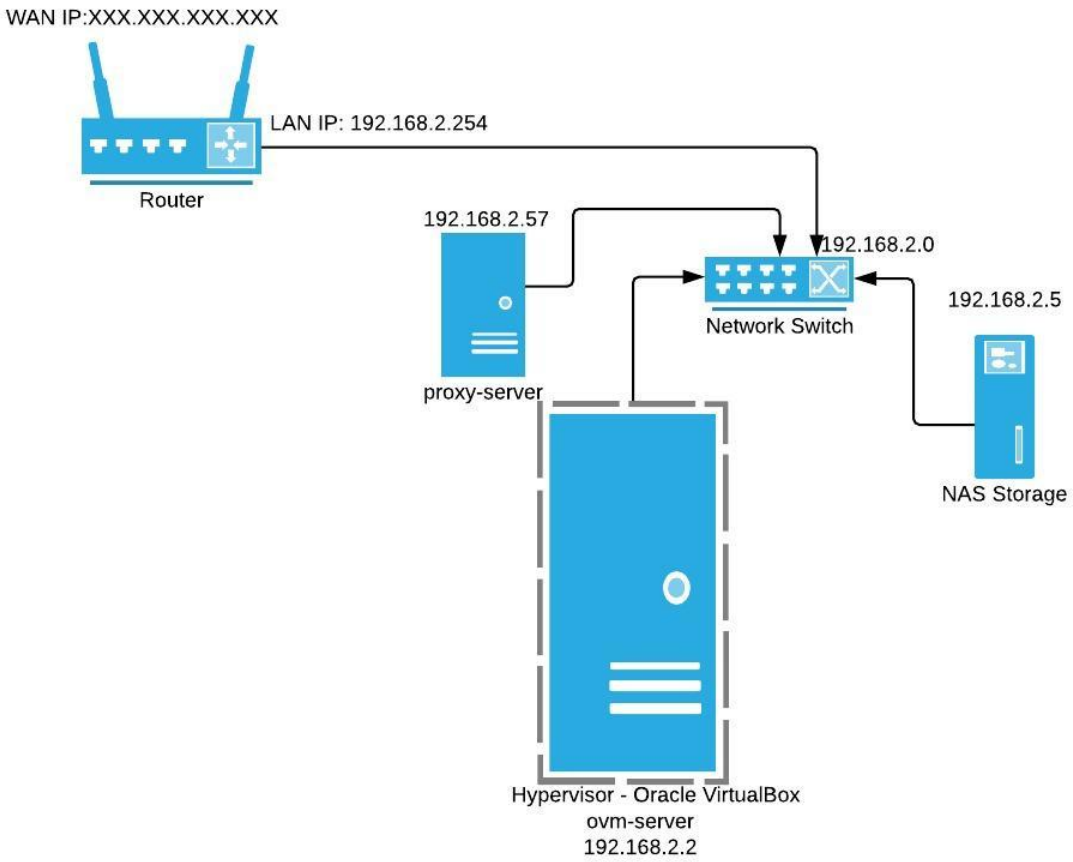
At this point I was ready to connect the below devices to the LAN Network Switch.

ISP router

Linux Machine – Proxy Server (proxy-server.hingu.net)

Linux Machine – VM Host (ovm-server.hingu.net)

WD's MyCloud NAS Storage – (MyCloudPR4100 - Pre-configured by WD).



Preparing the Storage for the Guest VMs:

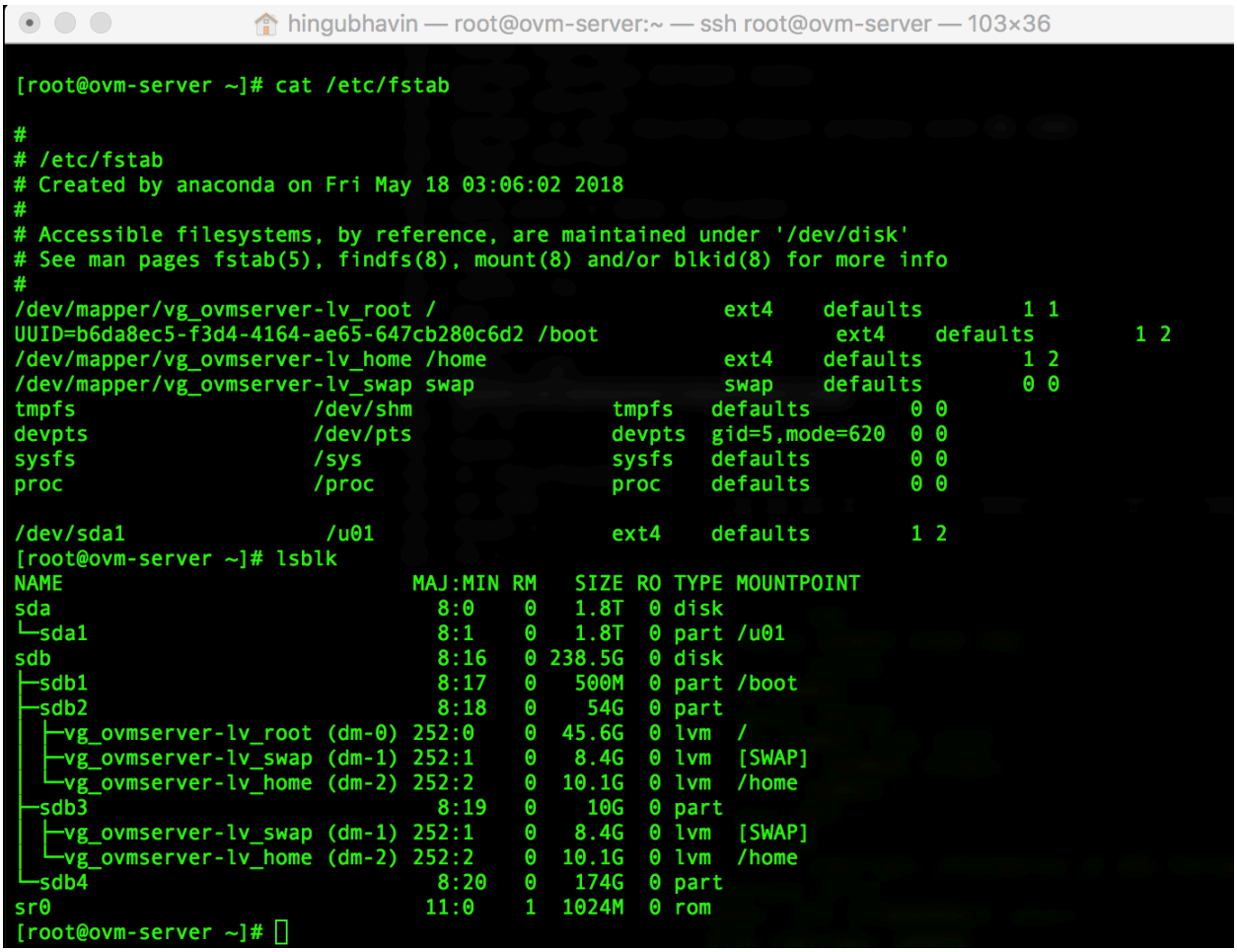
I wanted to utilize the 2TB of secondary storage hard drive that came with the machine to store all the guest VMs. For that, I had to first formatted it with ext4 since it was ntfs originally.

```
parted /dev/sda mklabel msdos
parted -a opt /dev/sda mkpart primary ext4 0% 100%
mkfs.ext4 -L data /dev/sda1

mkdir /u01
mount -o defaults /dev/sda1 /u01
```

put the entry in the /etc/fstab so that it gets auto mounted during startup. Created the directory under this mount to store the guest VMs.

```
mkdir /u01/guestVMs
```

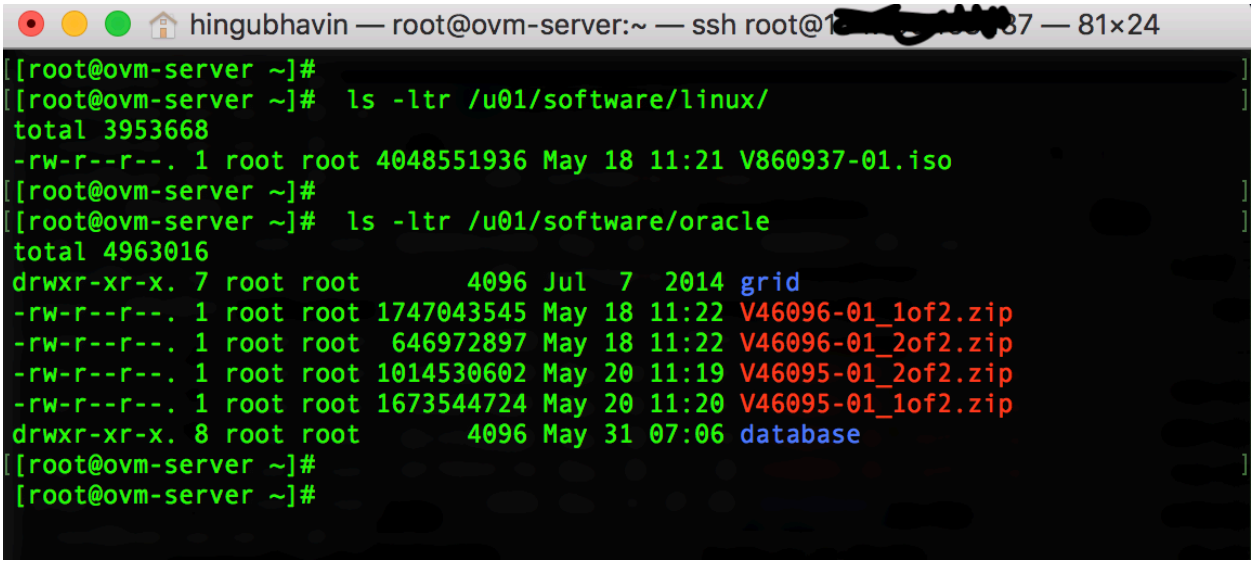


Stage the required downloads:

Created the staging area to store the ISO images of the Linux that I used in creating the guest VMs. I also downloaded the Oracle software like Grid Infrastructure, Oracle Database, etc.

```
mkdir -p /u01/software/linux
mkdir -p /u01/software/oracle
```

Downloaded the listed software from oracle's site: http://www.oracle.com/technetwork/indexes/downloads/index.html



Install Oracle VM VirtualBox on the VM Host (ovm-server):

Below is the Oracle VM VirtualBox document Link that I followed to successfully Install and configure the Oracle VM VirtualBox on the VM host (ovm-server).

http://www.oracle.com/technetwork/server-storage/virtualbox/documentation/index.html

```
yum install VirtualBox-5.2
yum list VirtualBox-5.2
```

Create the guest VMs

I created the first guest VM manually using GUI very first time which I then provisioned it for as per the requirement of Oracle Install (applied all the kernel settings/updates/RPMs/user group creation etc etc).
Then, I used this gold build to create all the subsequent VMs through the cloning. I used the /u01 mount that I created earlier for the storage of VM specific files (VDI) files.

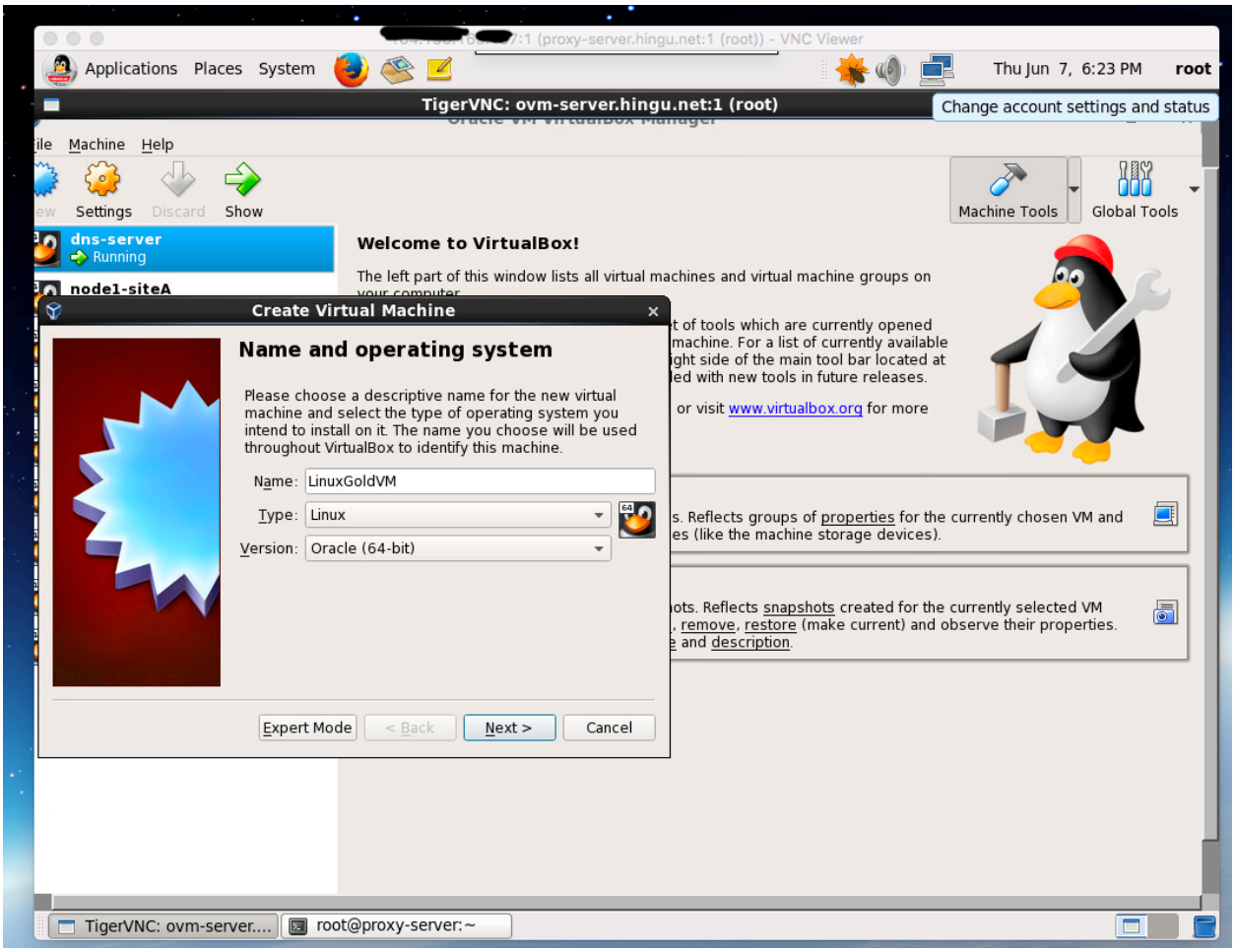
```
mkdir /u01/guestVMs/node
```

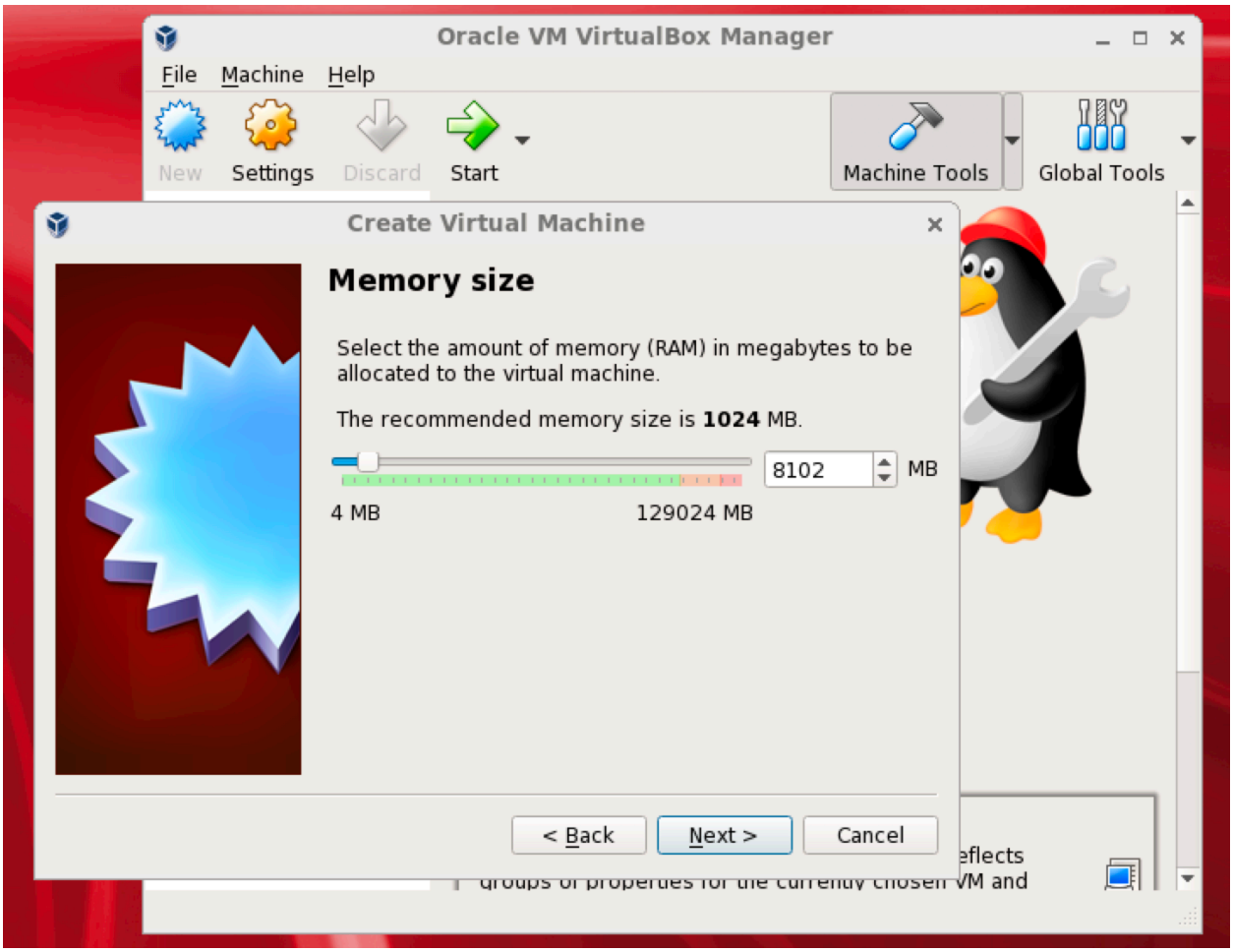
Here is the sample example of the process that I followed to create the guest VM manually.

Using X terminal, I started the VirtualBox as root and followed the instructions on the screen to create the guest VM. The instructions on screens are very straight forward. It asks for inputs like type of OS, Virtual Hard Disk, Memory, CPU, type of Network etc. As I wanted to access the VMs from any of the computer within the network, I decided to use the “Bridged Adaptor” instead of Host-Based / NAT Adaptor

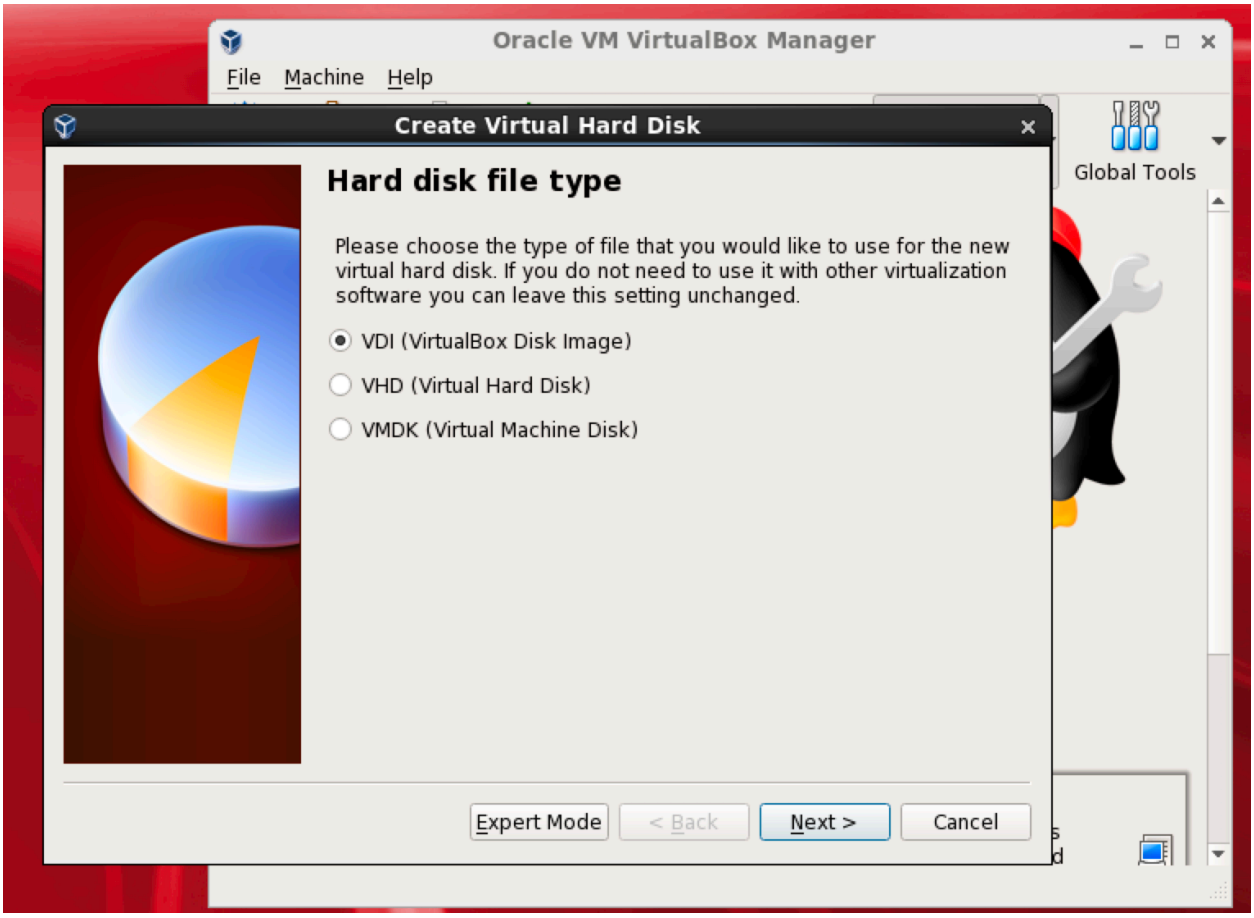
Start the VirtualBox as root by executing below command:

```
VirtualBox &
```

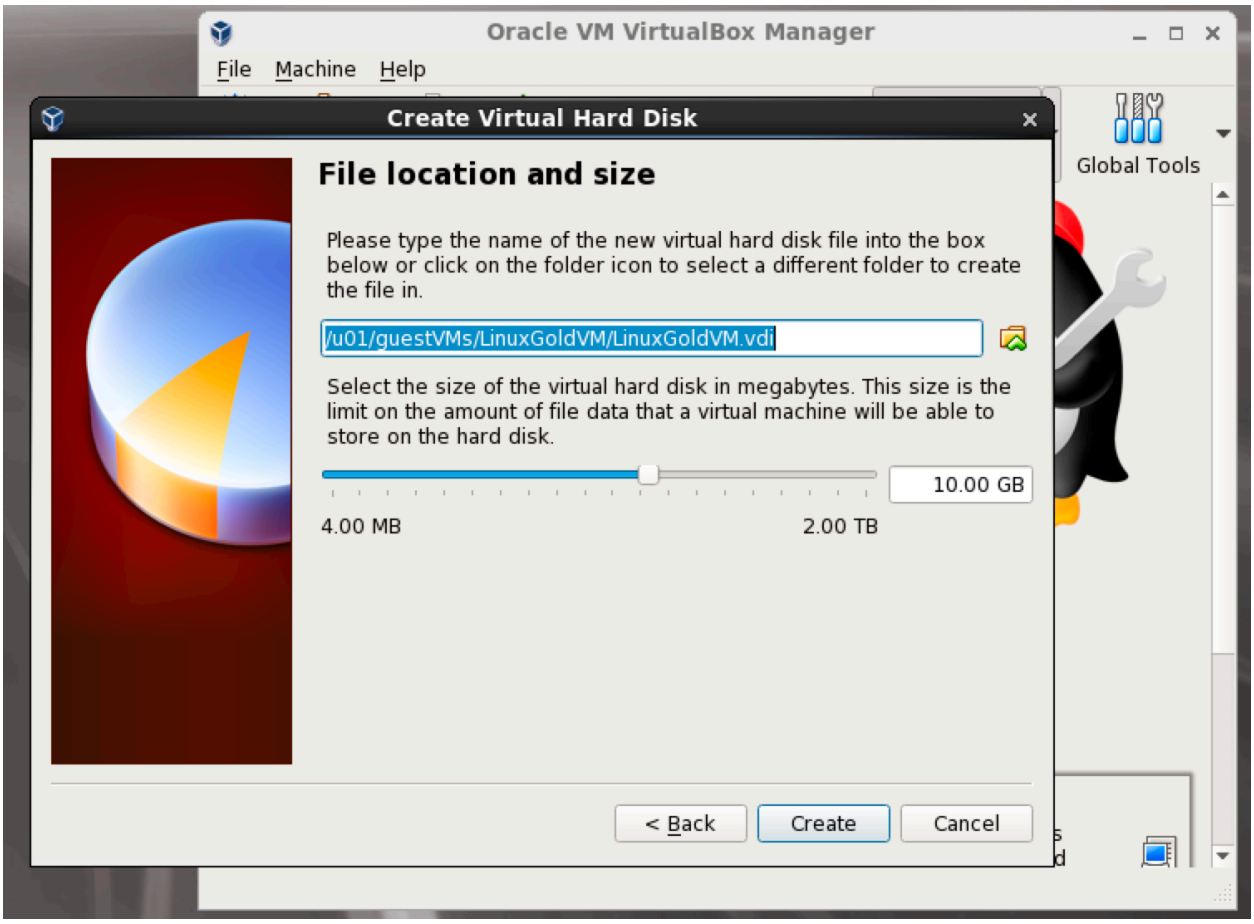



Oracle VM VirtualBox Manager
File
Machine
Help
New
Settings
Discard
Start
Machine Tools
Global Tools
Create Virtual Machine
Memory size
Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.
The recommended memory size is 1024 MB.
8102
MB
4 MB
129024 MB
< Back
Next >
Cancel


Oracle VM VirtualBox Manager
File
Machine
Help
New
Settings
Discard
Start
Machine Tools
Global Tools
Create Virtual Machine
Hard disk
If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.
If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.
The recommended size of the hard disk is 10.00 GB.
Do not add a virtual hard disk
Create a virtual hard disk now
Use an existing virtual hard disk file
asmdisk_crsdata_01.vdi (Shareable, 20.00 GB)
< Back
Create
Cancel

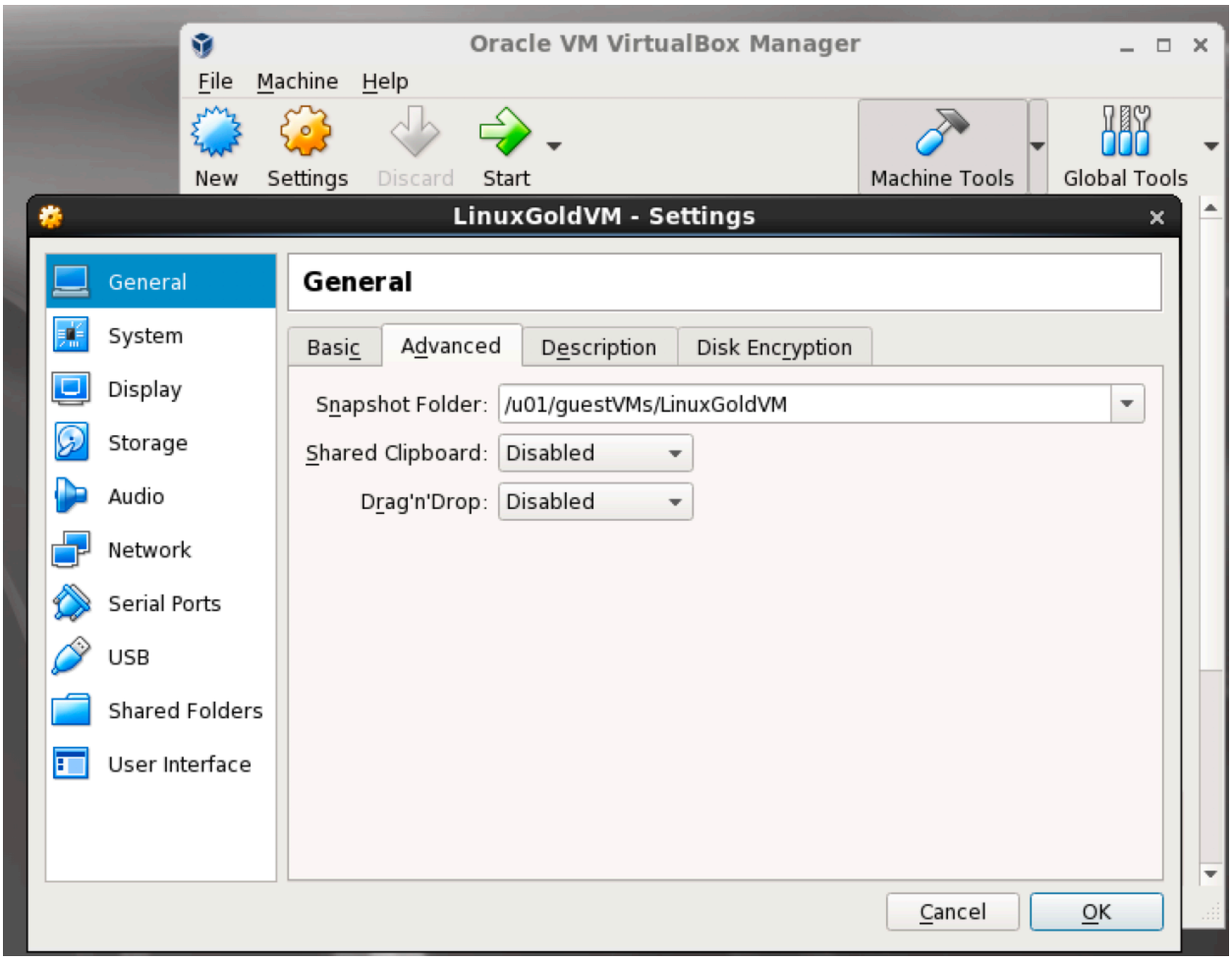

Oracle VM VirtualBox Manager
File
Machine
Help
Global Tools
Create Virtual Hard Disk
Hard disk file type
Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.
VDI (VirtualBox Disk Image)
VHD (Virtual Hard Disk)
VMDK (Virtual Machine Disk)
Expert Mode
< Back
Next >
Cancel

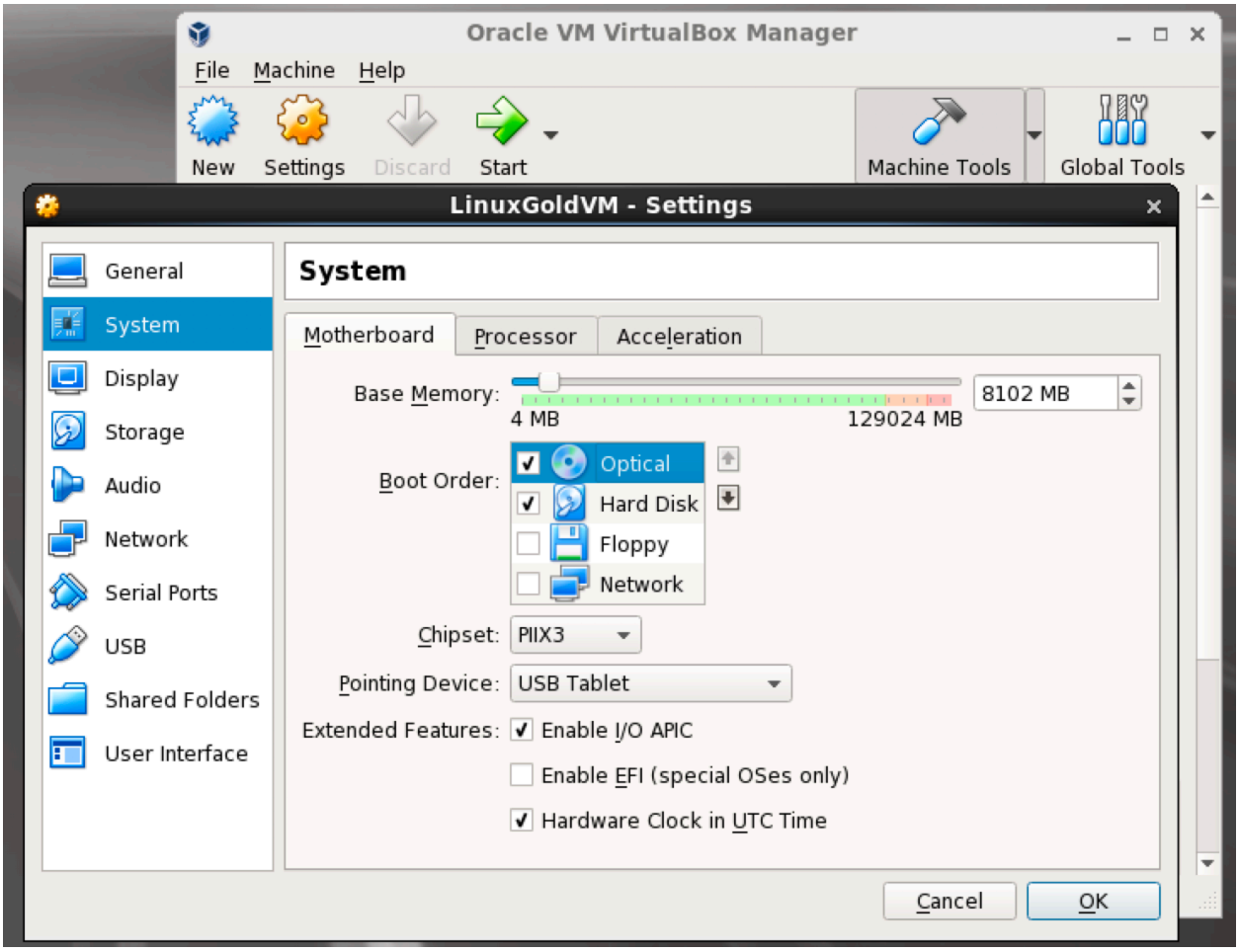
Here I make sure to change the path of the VM file to the /u01 mount that I created specifically to store the guest VMs.

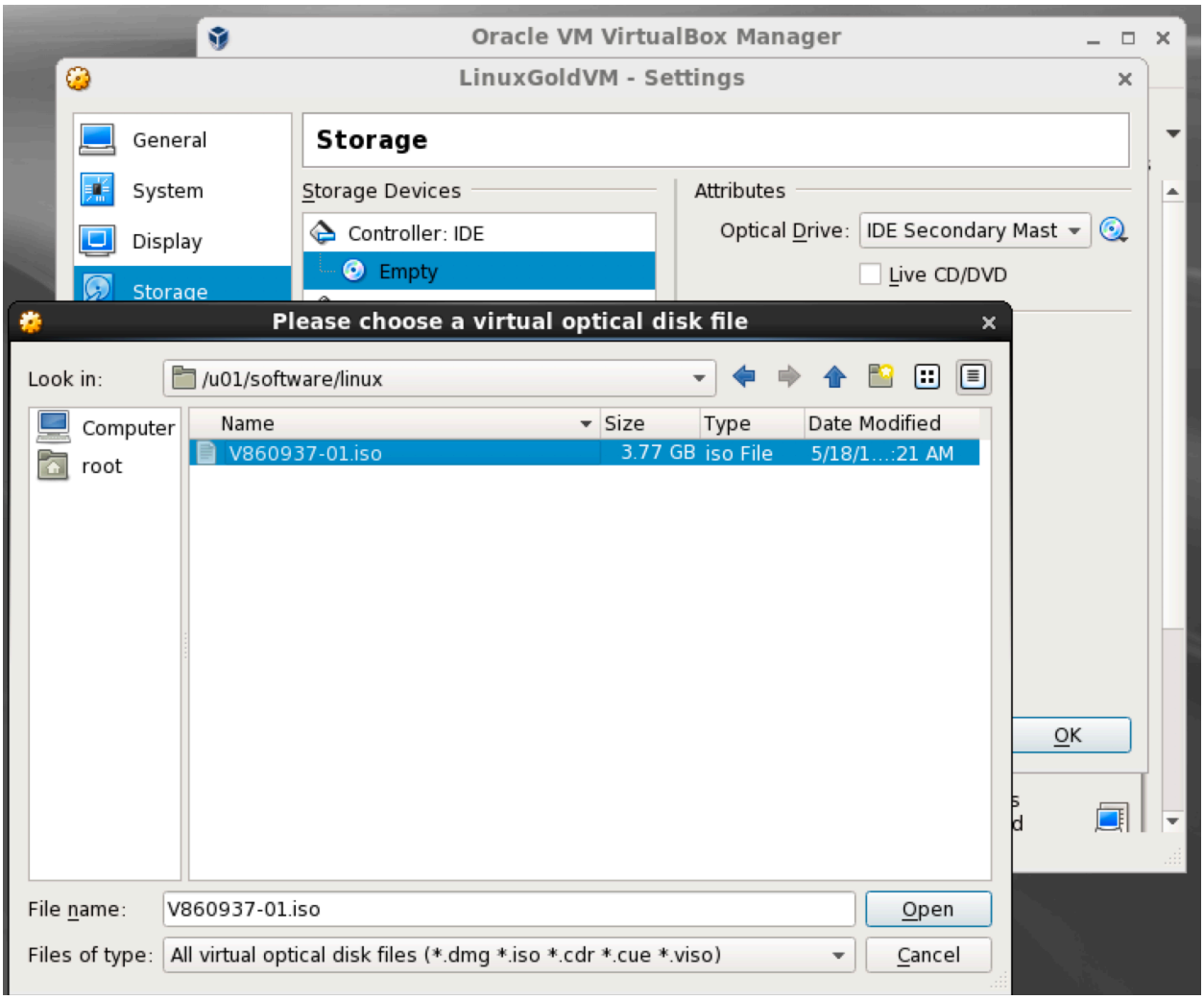


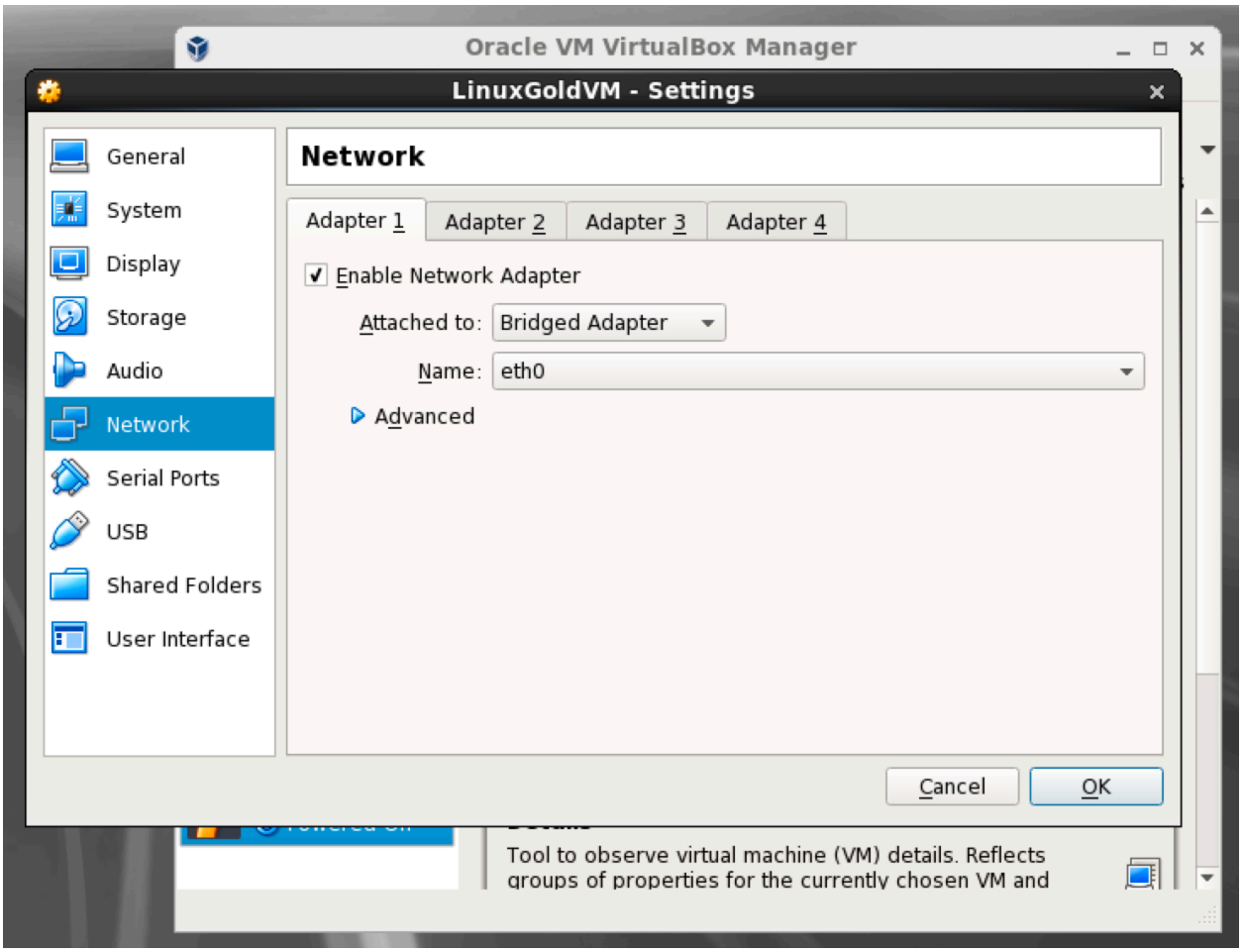
Once the guest VM template is created, click on “Settings” to change the System / Storage and Network values as shown in the next series of screenshots.


Oracle VM VirtualBox Manager
File
Machine
Help
New
Settings
Discard
Start
Machine Tools
Global Tools
dns-server
Running
node1-siteA
Powered Off
node2-siteA
Powered Off
node3-siteA
Powered Off
node1-siteB
Powered Off
node2-siteB
Powered Off
node1-siteC
Powered Off
node2-siteC
Powered Off
LinuxGoldVM
Powered Off
Welcome to VirtualBox!
The left part of this window lists all virtual machines and virtual machine groups on your computer.
The right part of this window represents a set of tools which are currently opened (or can be opened) for the currently chosen machine. For a list of currently available tools check the corresponding menu at the right side of the main tool bar located at the top of the window. This list will be extended with new tools in future releases.
You can press the F1 key to get instant help, or visit www.virtualbox.org for more information and latest news.
Details
Tool to observe virtual machine (VM) details. Reflects groups of properties for the currently chosen VM and

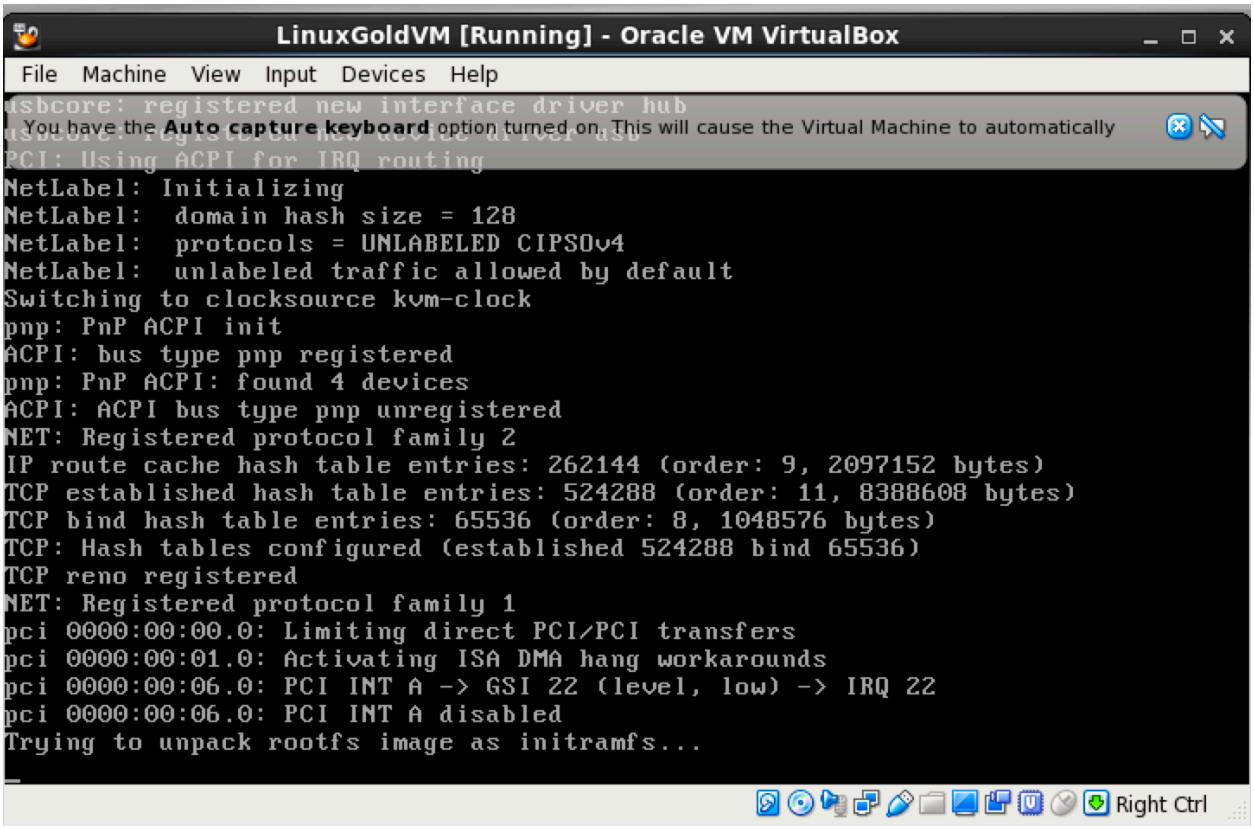

Oracle VM VirtualBox Manager
File
Machine
Help
New
Settings
Discard
Start
Machine Tools
Global Tools
LinuxGoldVM - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
General
Basic
Advanced
Description
Disk Encryption
Snapshot Folder:
/u01/guestVMs/LinuxGoldVM
Shared Clipboard:
Disabled
Drag'n'Drop:
Disabled
Cancel
OK

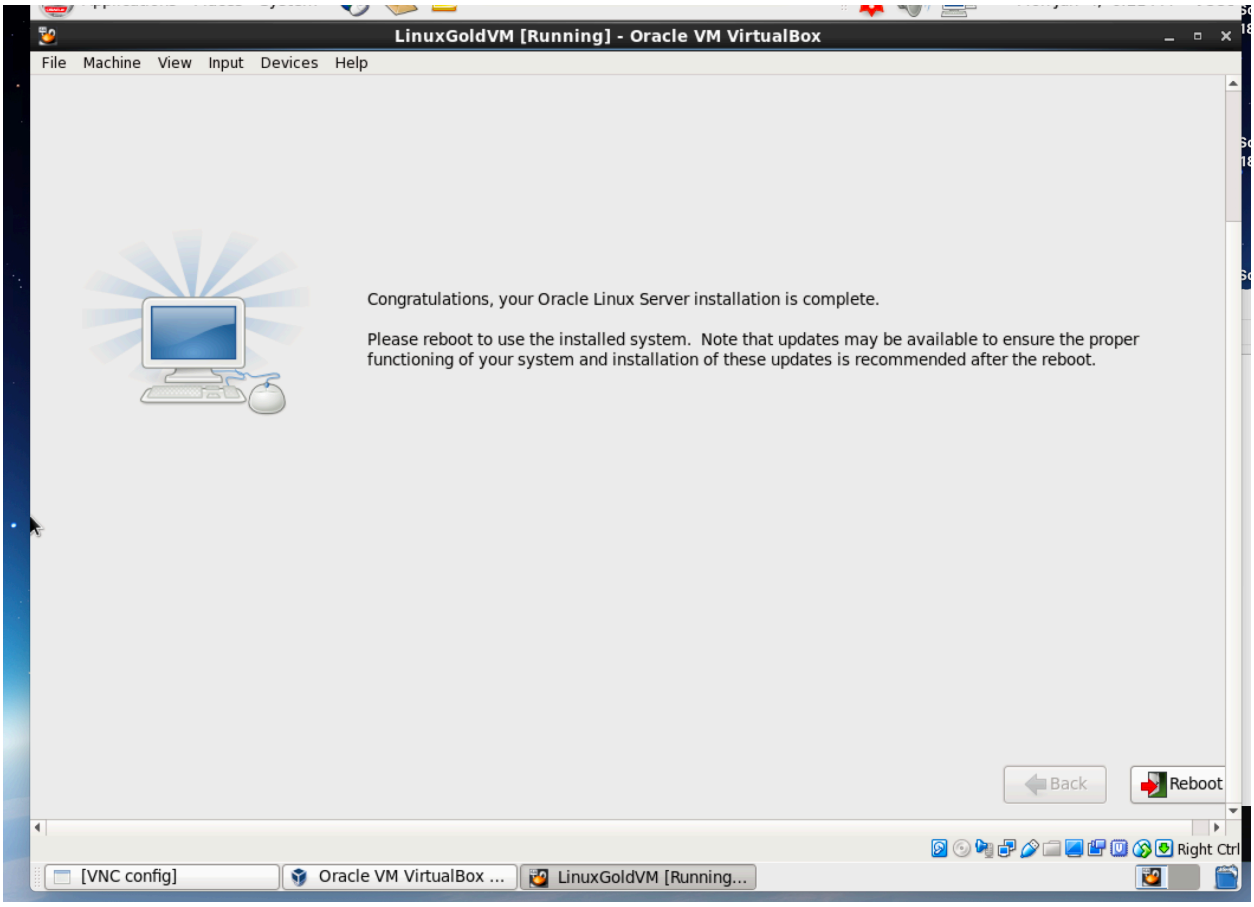

Oracle VM VirtualBox Manager
File
Machine
Help
New
Settings
Discard
Start
Machine Tools
Global Tools
LinuxGoldVM - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
System
Motherboard
Processor
Acceleration
Base Memory:
4 MB
129024 MB
8102 MB
Boot Order:
Optical
Hard Disk
Floppy
Network
Chipset:
PIIX3
Pointing Device:
USB Tablet
Extended Features:
Enable I/O APIC
Enable EFI (special OSes only)
Hardware Clock in UTC Time
Cancel
OK


Oracle VM VirtualBox Manager
LinuxGoldVM - Settings
General
System
Display
Storage
Storage
Storage Devices
Controller: IDE
Empty
Attributes
Optical Drive:
IDE Secondary Mast
Live CD/DVD
Please choose a virtual optical disk file
Look in:
/u01/software/linux
Computer
root
Name
Size
Type
Date Modified
V860937-01.iso
3.77 GB
iso File
5/18/1...:21 AM
File name:
V860937-01.iso
Open
Files of type:
All virtual optical disk files (*.dmg *.iso *.cdr *.cue *.viso)
Cancel
OK



At this Point, I was ready to Click the “Start” button on the VirtualBox main window to install the Oracle Linux OS. I simply followed the standard process of installing Linux OS depending on what software its going to host.


LinuxGoldVM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
You have the Auto capture keyboard option turned on. This will cause the Virtual Machine to automatically
usbcore: registered new interface driver hub
PCI: Using ACPI for IRQ routing
NetLabel: Initializing
NetLabel: domain hash size = 128
NetLabel: protocols = UNLABELED CIPSOv4
NetLabel: unlabeled traffic allowed by default
Switching to clocksource kvm-clock
pnp: PnP ACPI init
ACPI: bus type pnp registered
pnp: PnP ACPI: found 4 devices
ACPI: ACPI bus type pnp unregistered
NET: Registered protocol family 2
IP route cache hash table entries: 262144 (order: 9, 2097152 bytes)
TCP established hash table entries: 524288 (order: 11, 8388608 bytes)
TCP bind hash table entries: 65536 (order: 8, 1048576 bytes)
TCP: Hash tables configured (established 524288 bind 65536)
TCP reno registered
NET: Registered protocol family 1
pci 0000:00:00.0: Limiting direct PCI/PCI transfers
pci 0000:00:01.0: Activating ISA DMA hang workarounds
pci 0000:00:06.0: PCI INT A -> GSI 22 (level, low) -> IRQ 22
pci 0000:00:06.0: PCI INT A disabled
Trying to unpack rootfs image as initramfs...
Right Ctrl



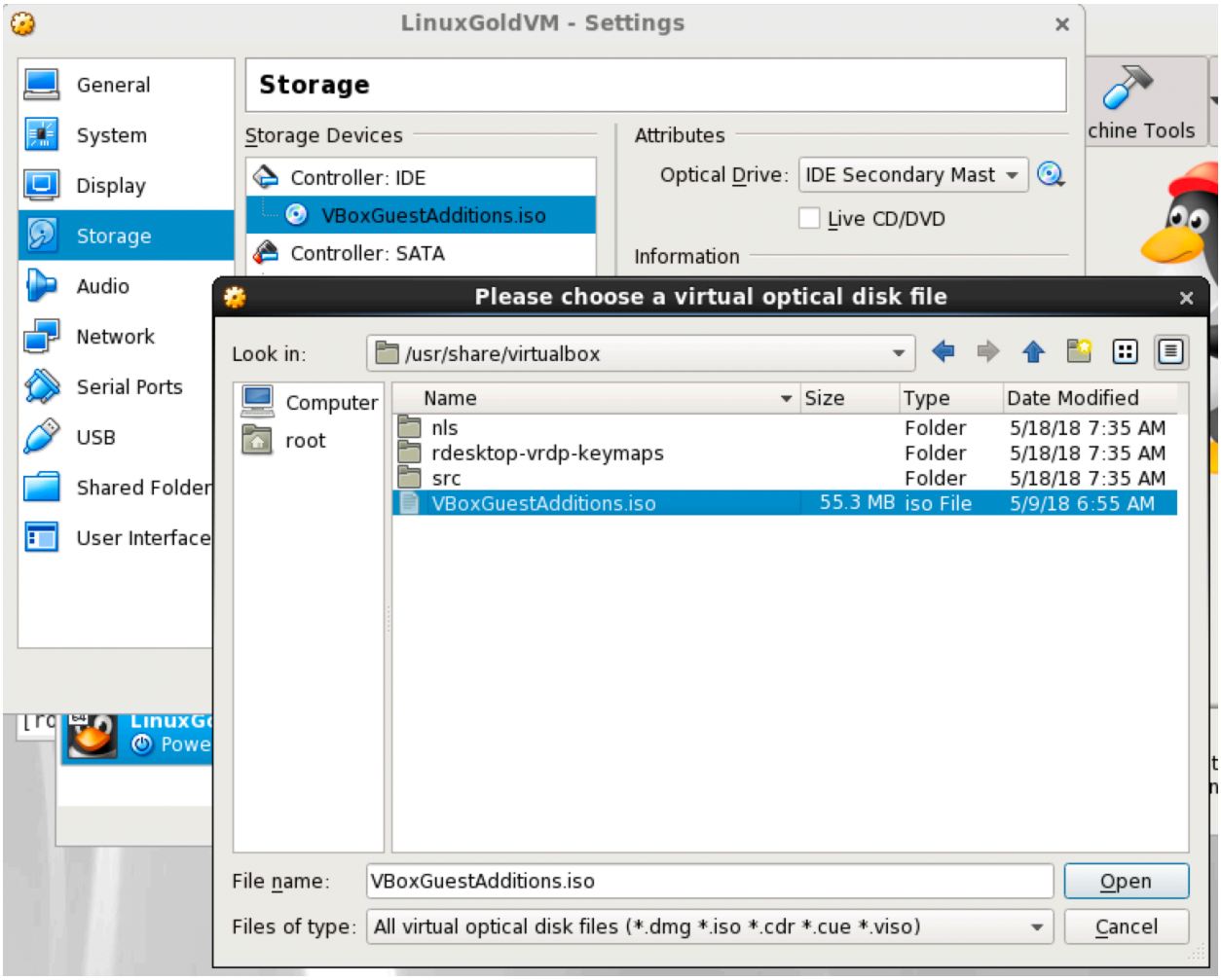
Install the guestAdditions in the guest VM:

Power off the VM:

```
VBoxManage controlvm 'LinuxGoldVM' poweroff
```

Mount the VBoxGuestAdditions.iso :

This file is located under /usr/share/virtualbox on the VM Host (ovm-server)



Power On the VM and mount the Virtual Disk under /media:

```
nohup /usr/bin/VBoxHeadless -startvm "LinuxGoldVM" &
mount -t iso9660  /dev/sr0 /media
```

Install the prerequisite RPMs:

```
yum install kernel-uek-devel
yum install kernel-uek-devel-`uname -r`
yum install kernel-headers
yum install kernel-devel
```

Execute the VBoxLinuxAdditions.run file:

```
cd /media
```

```
./VBoxLinuxAdditions.run
```

```
[root@LinuxGoldVm media]# ./VBoxLinuxAdditions.run
Verifying archive integrity... All good.
Uncompressing VirtualBox 5.2.12 Guest Additions for Linux........
VirtualBox Guest Additions installer
Removing installed version 5.2.12 of VirtualBox Guest Additions...
You may need to restart your guest system to finish removing the guest drivers.
Copying additional installer modules ...
Installing additional modules ...
VirtualBox Guest Additions: Building the VirtualBox Guest Additions kernel modul
es.
VirtualBox Guest Additions: Starting.
[root@LinuxGoldVm media]#
```

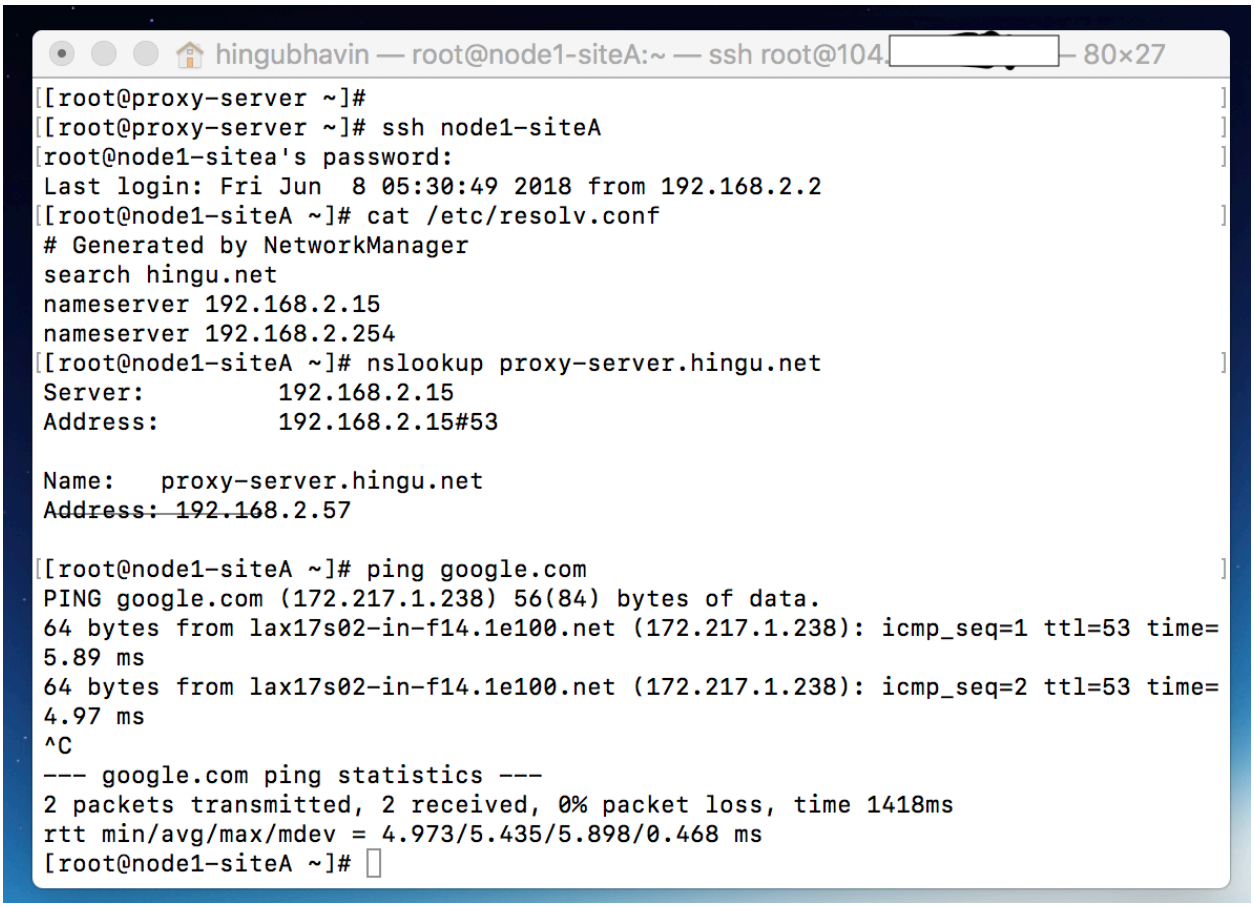
Reboot the VM after the install of guestAdditions:

```
VBoxManage controlvm 'LinuxGoldVM' poweroff
nohup /usr/bin/VBoxHeadless -startvm "LinuxGoldVM" &
```

```
[root@ovm-server ~]# VBoxManage controlvm 'LinuxGoldVM' poweroff
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
[root@ovm-server ~]#
[2]+  Done                    nohup /usr/bin/VBoxHeadless -startvm "LinuxGoldVM"
[root@ovm-server ~]# nohup /usr/bin/VBoxHeadless -startvm "LinuxGoldVM" &
[2] 32618
[root@ovm-server ~]# nohup: ignoring input and appending output to `nohup.out'

[root@ovm-server ~]#
[root@ovm-server ~]# VBoxManage showvminfo 'LinuxGoldVM' | grep State
State:           running (since 2018-06-05T14:26:18.252000000)
[root@ovm-server ~]#
```

The guest VM gets internet connection via ISP Router's default gateway (192.168.2.254) defined in the /etc/resolv.conf file. This file also contains my private DNS to resolve other VMs within the LAN "hingu.net" domain. I define the DNS1 and DNS2 in the ifcfg-eth0 file (NIC for public network) in each VM from where the Network Manager in Linux picks the Value of these DNSs and update the /etc/resolve.net during the startup of network services.



Clone the Gold image:

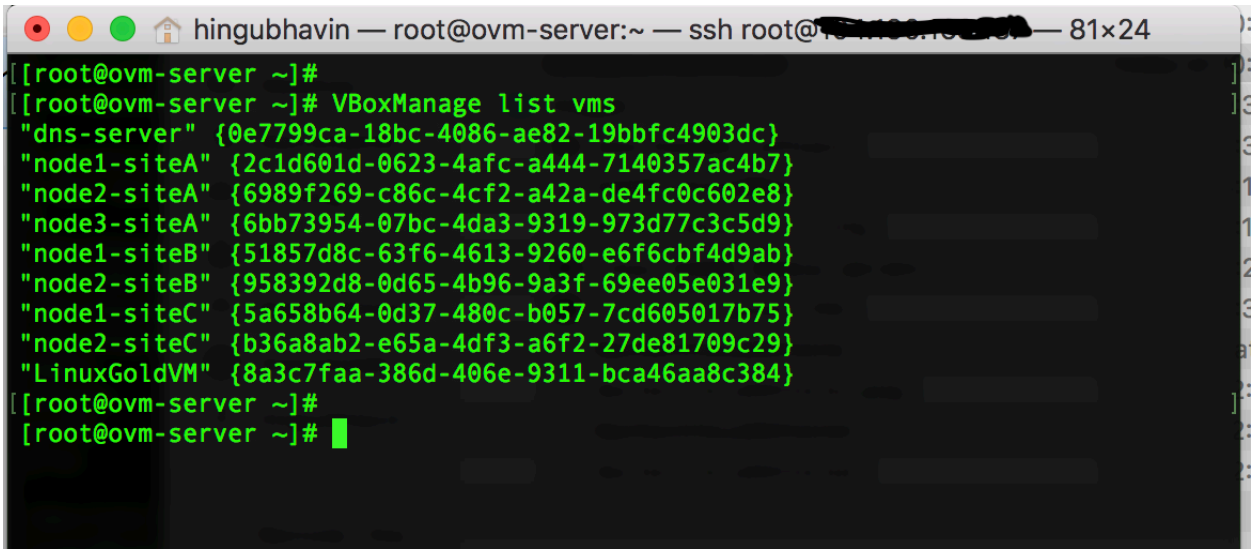
After fully provisioning the first VM, I created the subsequent VMs by cloning the node1-siteA VM as shown below.

```
VBoxManage clonevm node1-siteA --name node2-siteA --basefolder
/u01/guestVMs --register
VBoxManage clonevm node1-siteA --name node3-siteA --basefolder
/u01/guestVMs --register
VBoxManage clonevm node1-siteA --name node1-siteB --basefolder
/u01/guestVMs --register
VBoxManage clonevm node1-siteA --name node2-siteB --basefolder
/u01/guestVMs --register
VBoxManage clonevm node1-siteA --name node1-siteC --basefolder
/u01/guestVMs --register
VBoxManage clonevm node1-siteA --name node2-siteC --basefolder
/u01/guestVMs --register
```

I had to make the below network changes in the cloned VM in order for it to join the LAN network:

- Update the ifcfg-eth0 file with (a) new HWADDR of NIC (ifconfig -a | grep eth0 gives you the HWADDR), (b) remove the UUID, (c) change the IPADDR to one that is new and not allocated to any devices.
- Update the network file with the new hostname.
- Update the /etc/udev/rules.d/70-persistent-net.rules and to assign right HWADDR for eth0 and remove rest of the HWADDR lines from there.
- Reboot the VM.

Here is the List of the VMs that I have created in my Cloud system.



To get the detailed information of any VMs in the Cloud, I use

```
VBoxManage showvminfo 'node1-siteA'
```

Start the guest VMs:

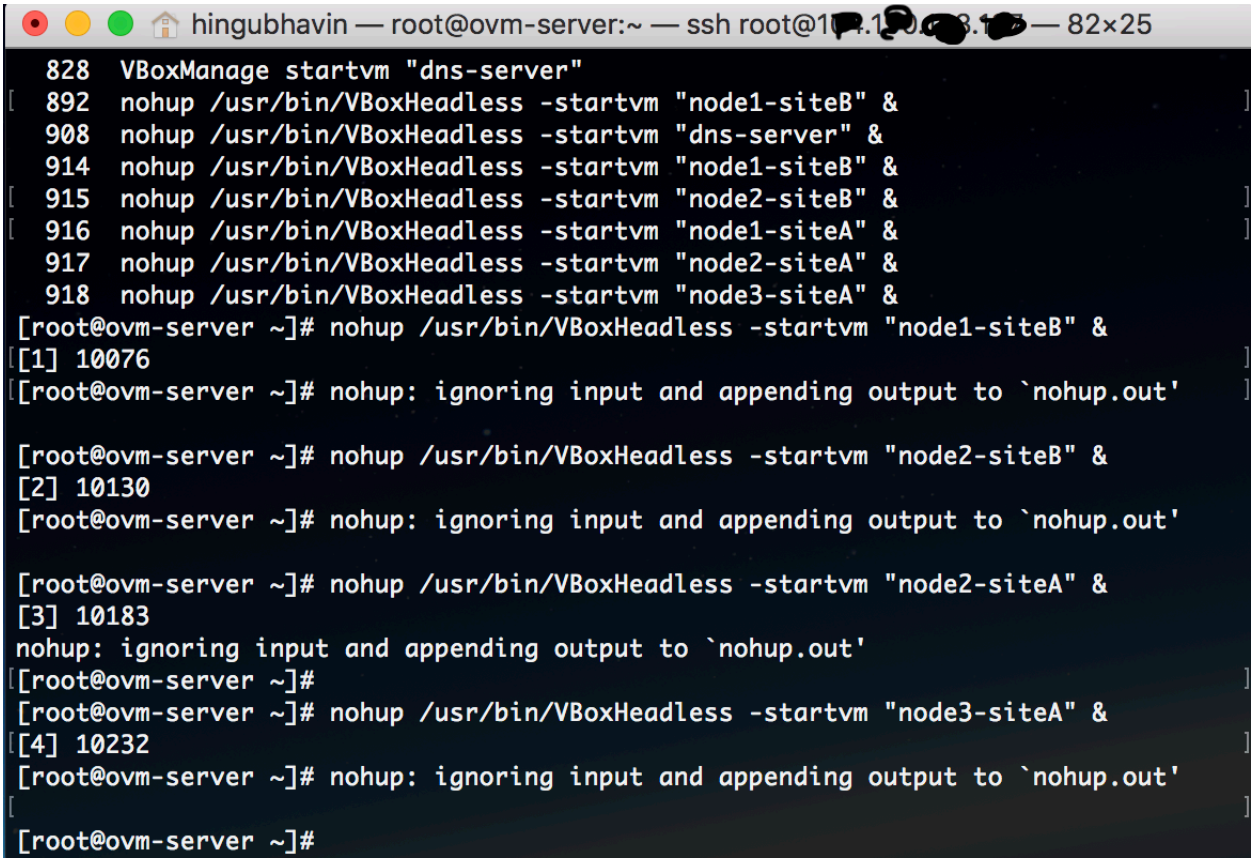
After Installing Oracle software on these VMs, I decided to check the load on the VM host (ovm-server) with all the VMs up and running. Also, at this point, I was ready to configure the DMZ in the ISP router and test the remote
accessibility to the Cloud System at home.

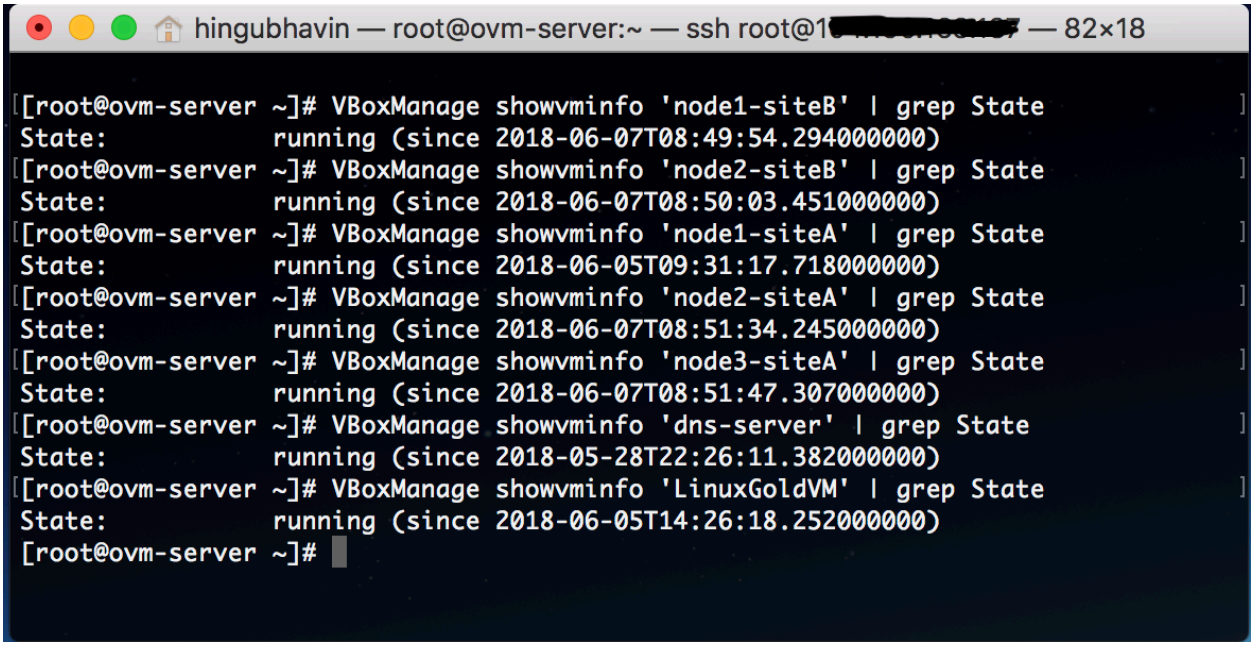
```
nohup /usr/bin/VBoxHeadless -startvm "dns-server" &
nohup /usr/bin/VBoxHeadless -startvm "node1-siteB" &
nohup /usr/bin/VBoxHeadless -startvm "node2-siteB" &
nohup /usr/bin/VBoxHeadless -startvm "node1-siteA" &
```

```
nohup /usr/bin/VBoxHeadless -startvm "node2-siteA" &
nohup /usr/bin/VBoxHeadless -startvm "node3-siteA" &
nohup /usr/bin/VBoxHeadless -startvm "LinuxGoldVM" &
```

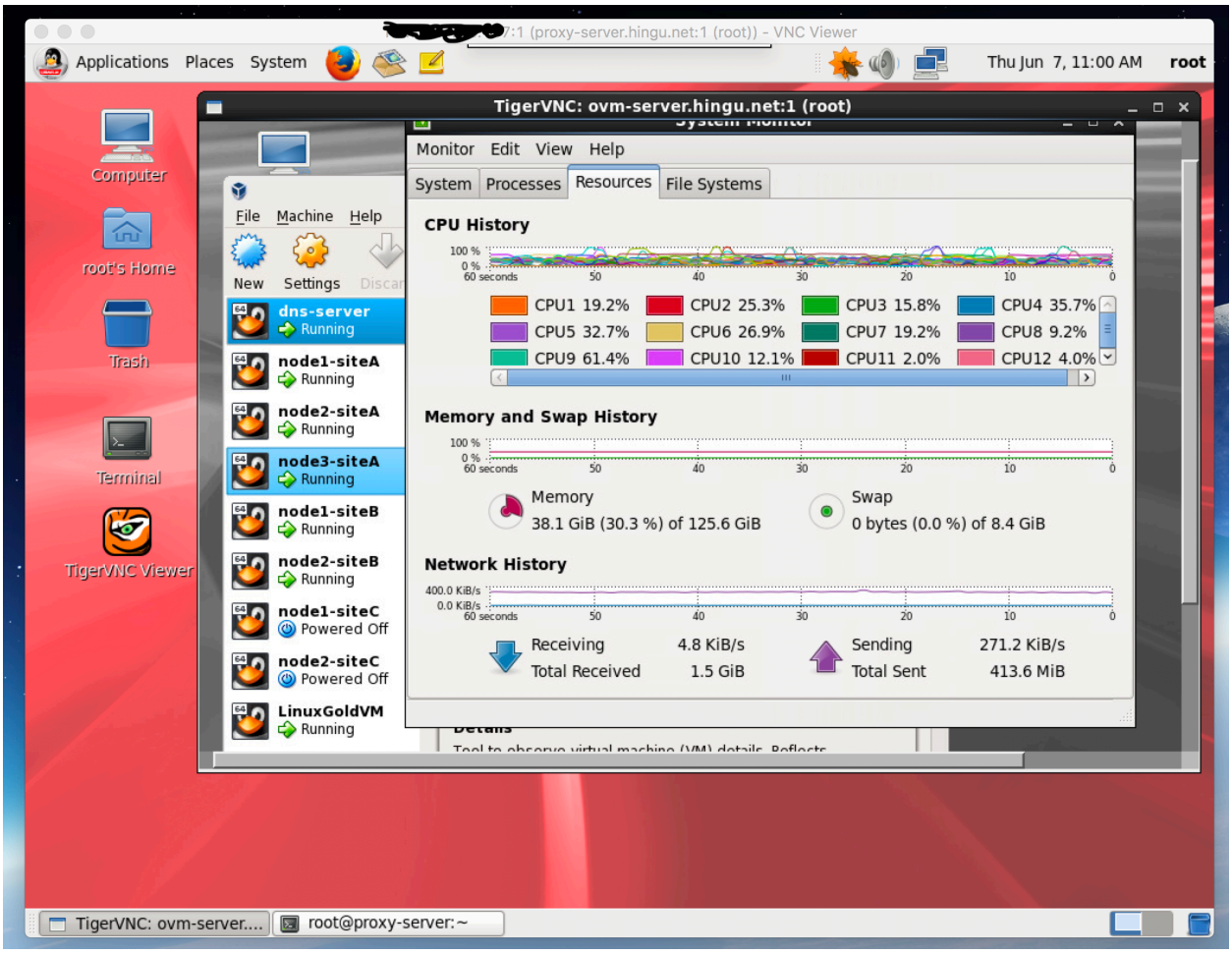
Get the status of the VMs

```
VBoxManage showvminfo 'dns-server' | grep State
VBoxManage showvminfo 'node1-siteA' | grep State
VBoxManage showvminfo 'node2-siteA' | grep State
VBoxManage showvminfo 'node3-siteA' | grep State
VBoxManage showvminfo 'node1-siteB' | grep State
VBoxManage showvminfo 'node2-siteB' | grep State
VBoxManage showvminfo 'LinuxGoldVM' | grep State
```



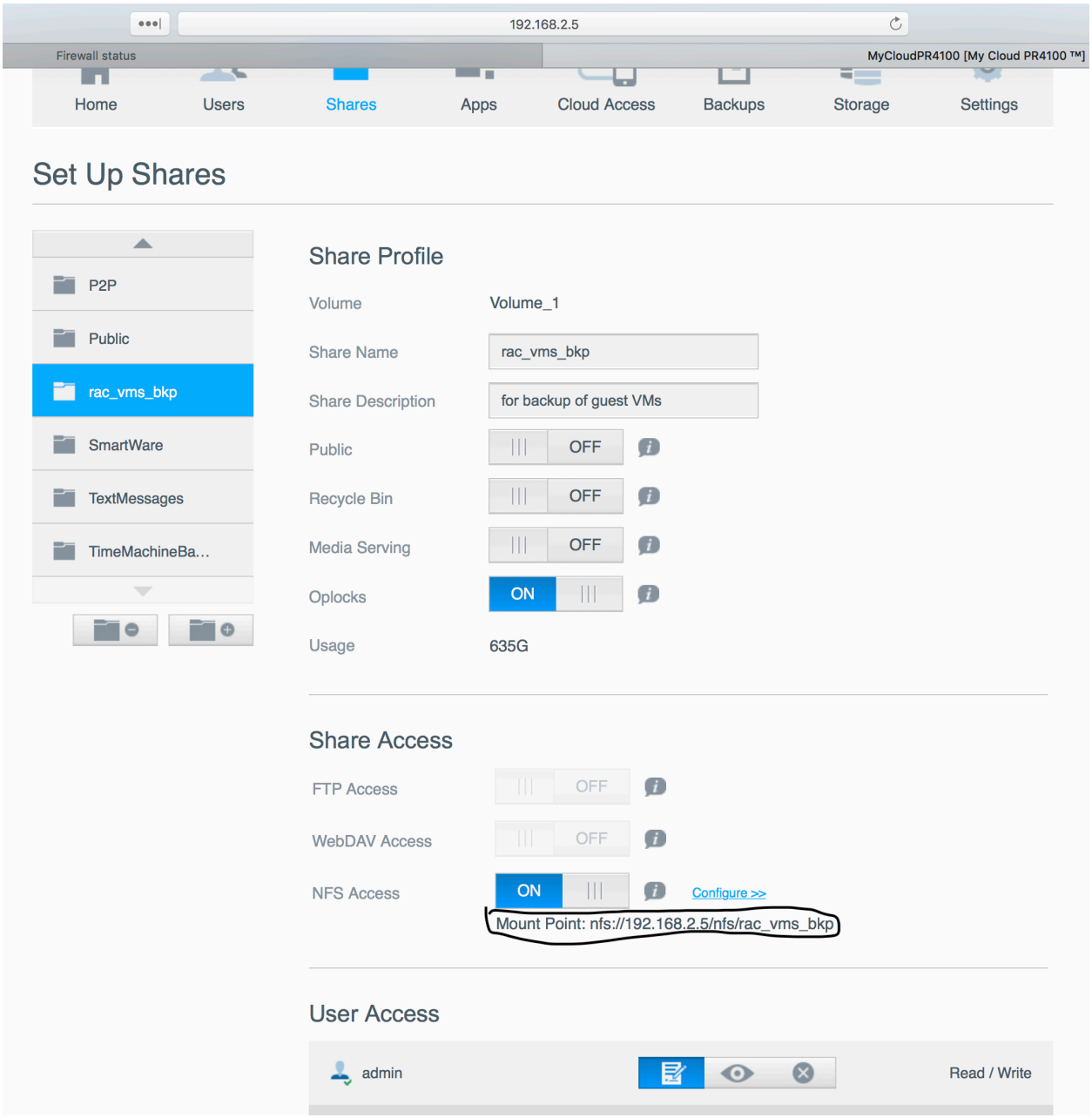


VM host – Resource Utilization Status:



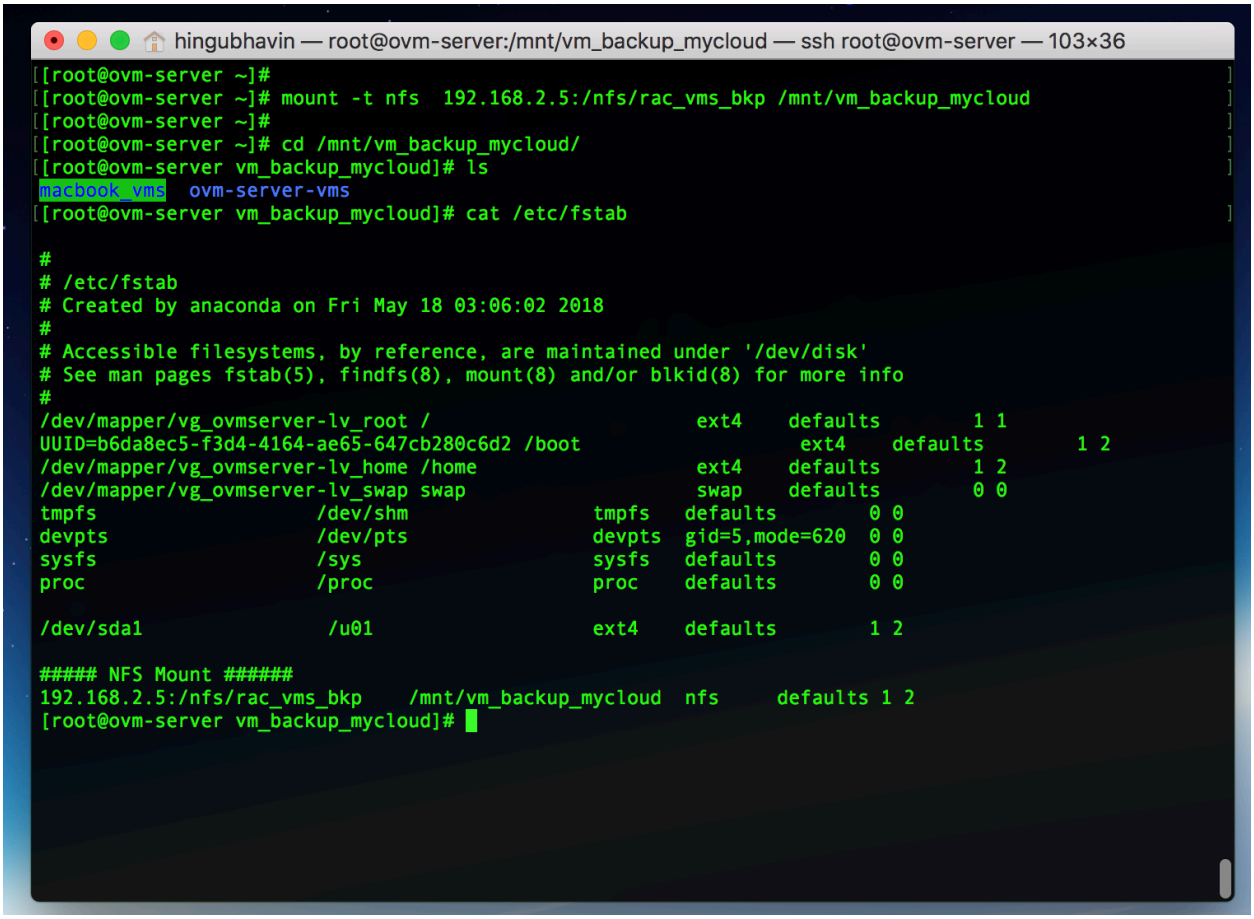
Setting up the Backup Server for Guest VMs:

Backing up the VMs is simply copying their VDI files to the backup media. Here, I took an advantage of existing NAS storage (MyCloud PR4100 from WD) to treat it like a backup server for the guest VMs. After carving out a 1TB LUN on the 24TB NAS presenting it to the VM host (ovm-server), it became ready for me to start backing up the VMs.



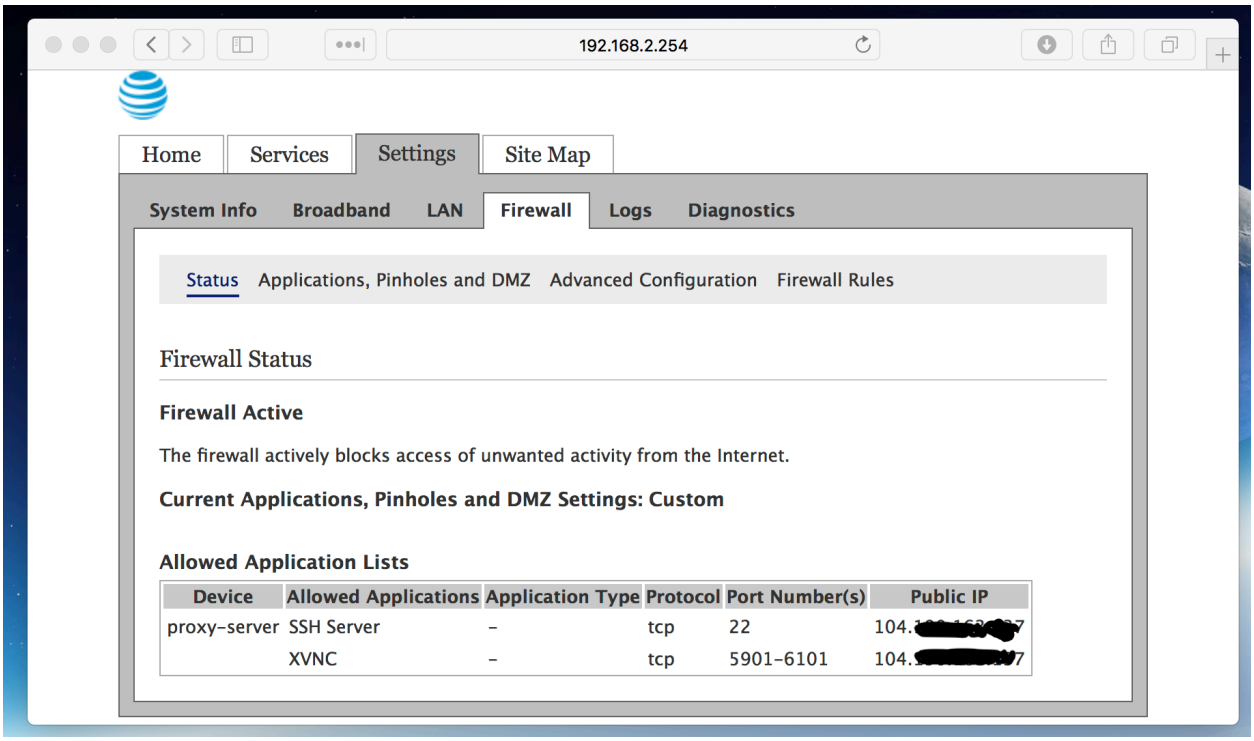

```
mount -t nfs  192.168.2.5:/nfs/rac_vms_bkp /mnt/vm_backup_mycloud
```

added this to the /etc/fstab for auto mount during the system startup.

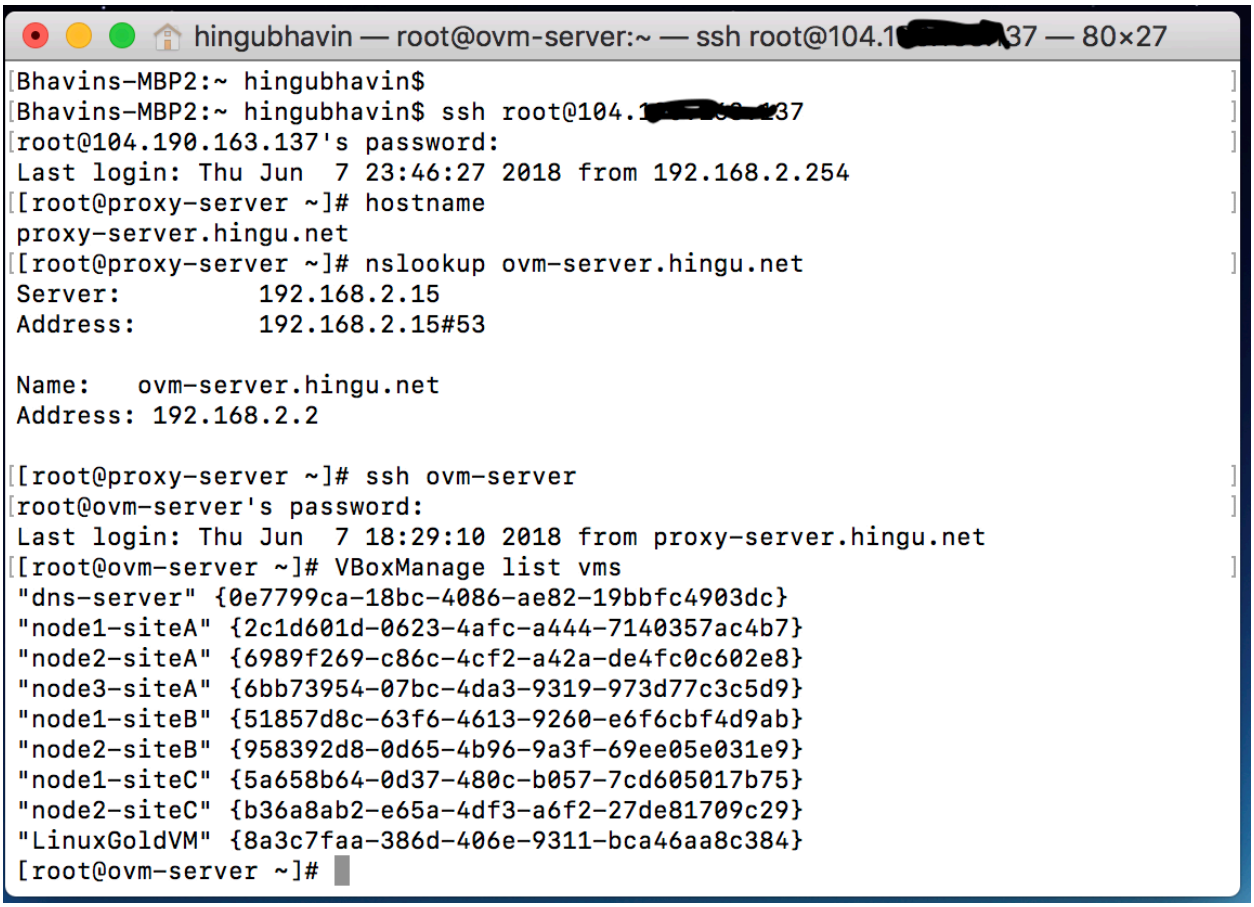


Accessing the Cloud Remotely:

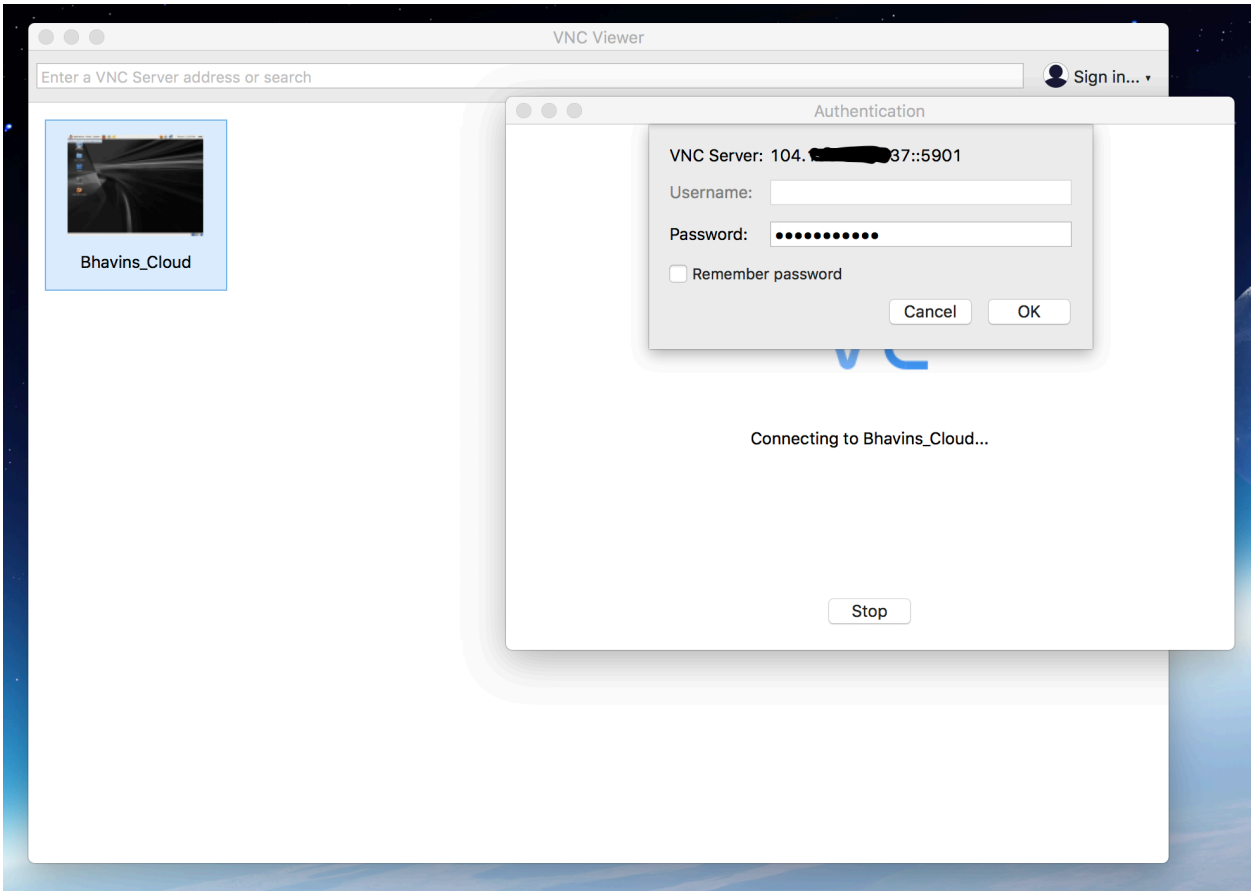
At this point, I was ready to modify the ISP (AT&T) router to setup the DMZ to forward the inbound connection request to the proxy-server only for certain ports like, 22 (ssh) and ports for VNC servers (5901 – 6101).
Here is what my router setting looks like after making the specified changes.



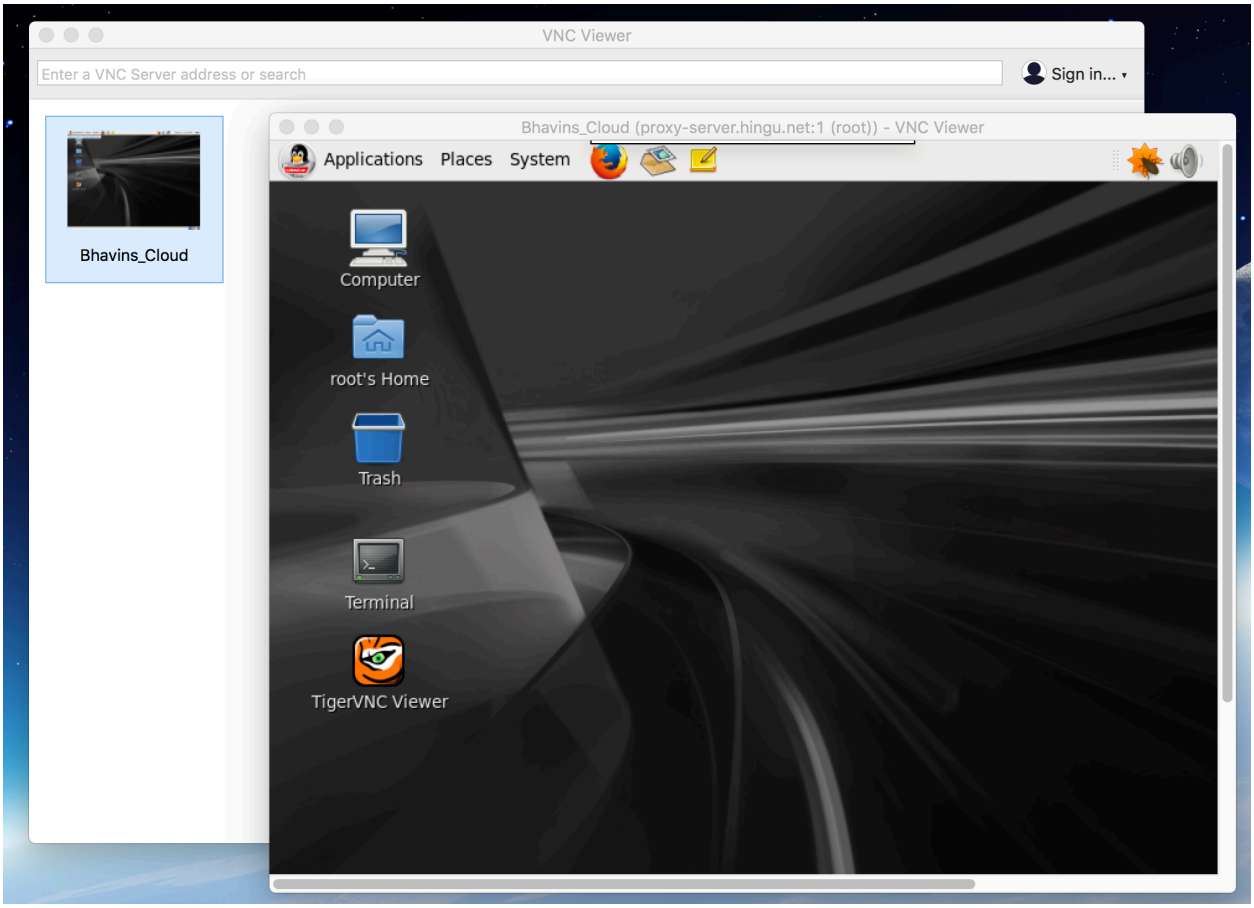
Time to test the connectivity using Public Address of the Route:

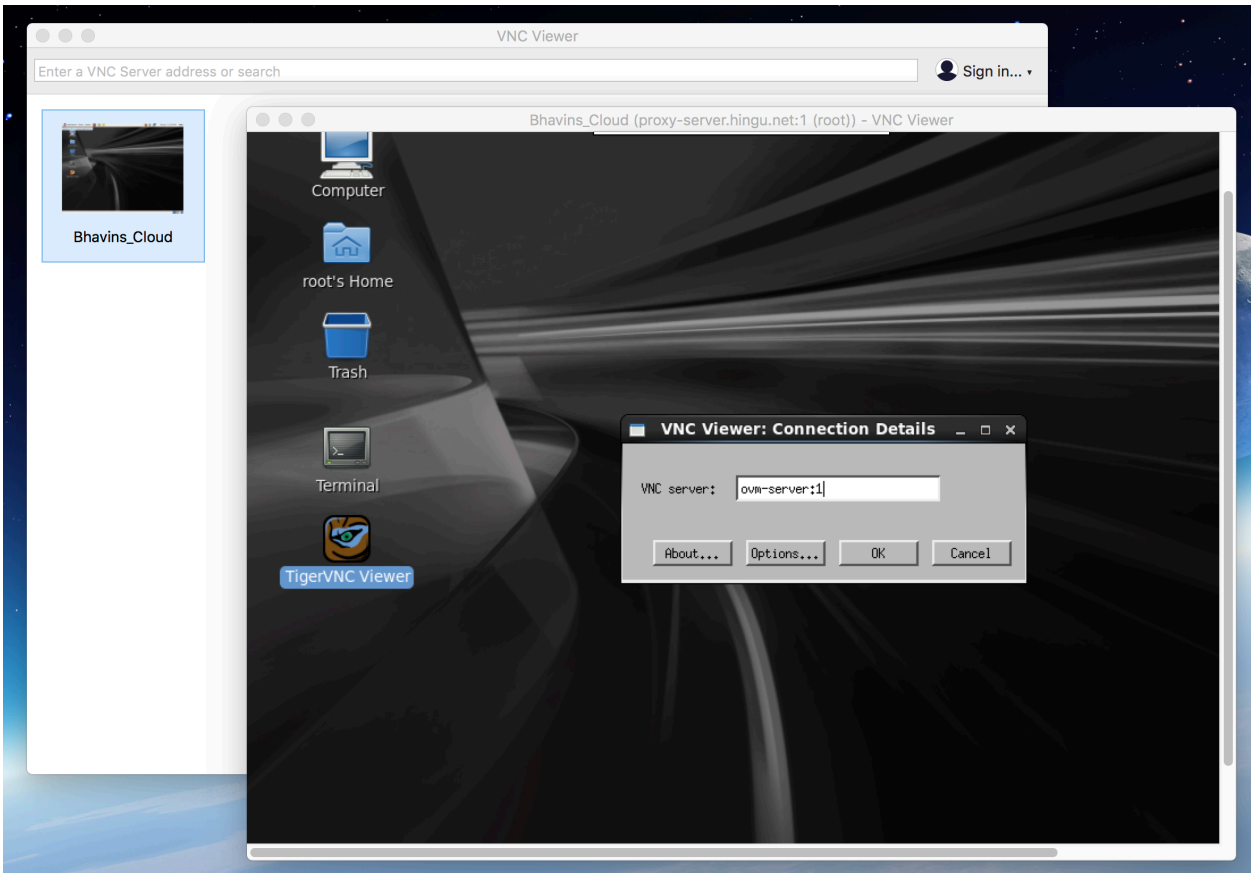

hingubhavin — root@ovm-server:~ — ssh root@104.1[redacted]37 — 80×27
Bhavins-MBP2:~ hingubhavin$
Bhavins-MBP2:~ hingubhavin$ ssh root@104.1[redacted]37
root@104.190.163.137's password:
Last login: Thu Jun 7 23:46:27 2018 from 192.168.2.254
[root@proxy-server ~]# hostname
proxy-server.hingu.net
[root@proxy-server ~]# nslookup ovm-server.hingu.net
Server: 192.168.2.15
Address: 192.168.2.15#53
Name: ovm-server.hingu.net
Address: 192.168.2.2
[root@proxy-server ~]# ssh ovm-server
root@ovm-server's password:
Last login: Thu Jun 7 18:29:10 2018 from proxy-server.hingu.net
[root@ovm-server ~]# VBoxManage list vms
"dns-server" {0e7799ca-18bc-4086-ae82-19bbfc4903dc}
"node1-siteA" {2c1d601d-0623-4afc-a444-7140357ac4b7}
"node2-siteA" {6989f269-c86c-4cf2-a42a-de4fc0c602e8}
"node3-siteA" {6bb73954-07bc-4da3-9319-973d77c3c5d9}
"node1-siteB" {51857d8c-63f6-4613-9260-e6f6cbf4d9ab}
"node2-siteB" {958392d8-0d65-4b96-9a3f-69ee05e031e9}
"node1-siteC" {5a658b64-0d37-480c-b057-7cd605017b75}
"node2-siteC" {b36a8ab2-e65a-4df3-a6f2-27de81709c29}
"LinuxGoldVM" {8a3c7faa-386d-406e-9311-bca46aa8c384}
[root@ovm-server ~]#

Using VNC:



Connected to the proxy-server on X-Desktop. I could use TigerVNC on proxy-server to connect to any of the Linux guest VMs or ovm-server in I need to run any X application. Generally, I prefer install all my Oracle software using silent mode through my automation scripts but in case if I ever want to use X-windows, VNCViewer could be very handy.


VNC Viewer
Enter a VNC Server address or search
Sign in...
Bhavins_Cloud
Bhavins_Cloud (proxy-server.hingu.net:1 (root)) - VNC Viewer
Applications
Places
System
Computer
root's Home
Trash
Terminal
TigerVNC Viewer


VNC Viewer
Enter a VNC Server address or search
Sign in...
Bhavins_Cloud
Bhavins_Cloud (proxy-server.hingu.net:1 (root)) - VNC Viewer
Computer
root's Home
Trash
Terminal
TigerVNC Viewer
VNC Viewer: Connection Details
VNC server:
ovm-server:1
About...
Options...
OK
Cancel


Bhavins_Cloud (proxy-server.hingu.net:1 (root)) - VNC Viewer
Applications
Places
System
TigerVNC: ovm-server.hingu.net:1 (root)
Browse and run installed applications
Oracle VM VirtualBox Manager
File
Machine
Help
New
Settings
Discard
Show
Machine Tools
dns-server
Running
node1-siteA
Powered Off
node2-siteA
Powered Off
node3-siteA
Powered Off
node1-siteB
Powered Off
node2-siteB
Powered Off
node1-siteC
Powered Off
node2-siteC
Powered Off
LinuxGoldVM
Powered Off
Welcome to VirtualBox!
The left part of this window lists all virtual machines and virtual machine groups on your computer.
The right part of this window represents a set of tools which are currently opened (or can be opened) for the currently chosen machine. For a list of currently available tools check the corresponding menu at the right side of the main tool bar located at the top of the window. This list will be extended with new tools in future releases.
You can press the F1 key to get instant help, or visit www.virtualbox.org for more information and latest news.
Details
Tool to observe virtual machine (VM) details.
Computer
root's Home
Trash
Terminal
TigerVNC Viewer
Oracle VM VirtualBox
Bhavins_Cloud

Summary:

So, in the Amazon's terminology. I can say that I have created my own VPC at home from ground up. The storage subsystem that I have used here is what Amazon referred to as S3 (2TB local mount /u01 for the database storage and VDI files for VMs). The guest VMs are my Amazon EC2 instances and the dns server that I setup to resolve "hingu.net" private domain in my LAN is my Private DNS. The NFS mount that I created using the 1TB LUN carved out of NAS storage and is shared across all the VMs is referred to as Amazon EFS.