

KVM Installation on CentOS 6.x.

KVM is a **kernel-based Virtual Machine** which grows quickly in maturity and popularity in the Linux server market. Red Hat officially dropped **Xen** in favor of **KVM** since **RHEL 6**. With **KVM** being officially supported by Red Hat, installing **KVM** on RedHat-based systems should be a breeze.

Table of Contents

- Preliminary Check - Check Hardware Virtualization Support.
- Disable SELinux.
- Install KVM, QEMU and RPMs/packages.
 - Install KVM, QEMU and user-space tools.
 - Start libvirtd daemon, and set it to auto-start.
 - Check if KVM has successfully been installed.
 - Also we can do a `group install` (Optional)
- Configure Linux Bridge for VM Networking.
 - Install `bridge-utils`.
 - Disable Network Manager.
 - Create `bridge`.
- Install VirtManager.
 - To install VirtManager.
 - Launch VirtManager Remotely.
 - Create `wrapper` for `virt-manager`.
- Installing vnc-server.
 - Initial Installation.
 - Adding VNC user and setting `vncpasswd`.
 - Logging in from Remote machine.
 - Screenshot.
- Troubleshooting KVM and VirtManager setup.
- Upgrade CPU/RAM in KVM.
 - Listing `virtual` servers.
 - Getting Information about VM.
 - Edit Hardware for each VM.
 - Checking VM information.
 - Interface Changes.
- Adding HDD to Virtual Machine.
 - First Create a `image` using `qemu-img` command.
 - Add `image` to VM as a `virtio` HDD.
 - Method 1 - Adding `image` from `virt-manager` UI.

- Method 2 - Adding `image` using `virsh` command.
 - Added `image` will appear as `/dev/vda` in the VM, format it using the `mke2fs` command.
 - Add to `/etc/fstab`, so that the device can be mounted on boot up.
- Useful Links.

KVM is a **kernel-based Virtual Machine** which grows quickly in maturity and popularity in the Linux server market. Red Hat officially dropped **Xen** in favor of KVM since RHEL 6. With KVM being officially supported by Red Hat, installing KVM on RedHat-based systems should be a breeze.

Preliminary Check - Check Hardware Virtualization Support.

We can check the `cpuinfo` details to check if our h/w can do VM.

```
$ egrep -i 'vmx|svm' --color=always /proc/cpuinfo
```

If CPU flags contain “`vmx`” or “`svm`”, it means hardware virtualization support is available. IF NOT THEN DO NOT PROCEED.

Disable SELinux.

Before installing KVM, be aware that there are several SELinux booleans that can affect the behavior of KVM and libvirt. We will set `disable` in SELinux

```
#To disable SELinux on CentOS:
$ sudo -e /etc/selinux/config
```

```
SELINUX=disabled
```

Now reboot the server.

Install KVM, QEMU and RPMs/packages.

Install KVM and `virtinst` (a tool to create VMs).

Install KVM, QEMU and user-space tools.

```
$ sudo yum install kvm libvirt python-virtinst qemu-kvm
```

Start libvirtd daemon, and set it to auto-start.

```
$ sudo service libvirtd start
$ sudo chkconfig libvirtd on
```

Check if KVM has successfully been installed.

```
$ sudo virsh -c qemu:///system list
```

```
Id      Name                               State
-----
```

Also we can do a group install (Optional)

```
$ sudo yum groupinstall "Virtualisation Tools" "Virtualization Platform"
$ sudo yum install python-virtinst
```

Configure Linux Bridge for VM Networking.

Installing KVM alone does not allow VMs to communicate with each other or access external networks. We will create a “bridged networking” via Linux bridge.

Install bridge-utils.

Install a package needed to create and manage bridge devices:

```
$ sudo yum install bridge-utils
```

Disable Network Manager.

Disable Network Manager service if it’s enabled, and switch to default net manager as follows.

```
$ sudo service NetworkManager stop
$ sudo chkconfig NetworkManager off
$ sudo chkconfig network on
$ sudo service network start
```

Create bridge.

To create a bridge we need to configure an active network interface like eth0. Here we have a static IP assignment.

Note : copy file ifcfg-eth0 as ifcfg-br0, and edit them.

```
sudo vim /etc/sysconfig/network-scripts/ifcfg-eth0
```

Add the below lines, we have added a BRIDGE=br0.

```
DEVICE="eth0"
NM_CONTROLLED="no"
ONBOOT="yes"
TYPE="Ethernet"
BRIDGE=br0
```

Next we create a bridge br0

```
sudo vim /etc/sysconfig/network-scripts/ifcfg-br0
```

Add below lines. Need to look out for - DEVICE, BOOTPROTO, TYPE

```
DEVICE="br0"  
BOOTPROTO="static"  
NM_CONTROLLED="no"  
ONBOOT="yes"  
TYPE="Bridge"  
NETMASK=255.255.255.192  
IPADDR=10.10.18.36  
GATEWAY=10.10.18.1
```

Now lets restart network

```
$ sudo service network restart
```

This will restart the Network. IMPORTANT : make sure we have the bridge configured correctly, else we will loose connectivity if we are on ssh.

```
[ahmed@zahmed-server ~]$ ifconfig  
br0      Link encap:Ethernet  HWaddr A0:D3:C1:FA:25:C8  
         inet addr:10.10.18.36  Bcast:10.130.18.63  Mask:255.255.255.192  
         inet6 addr: fe80::a2d3:c1ff:fefa:25c8/64 Scope:Link  
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
         RX packets:2056649 errors:0 dropped:0 overruns:0 frame:0  
         TX packets:2027594 errors:0 dropped:0 overruns:0 carrier:0  
         collisions:0 txqueuelen:0  
         RX bytes:294572910 (280.9 MiB)  TX bytes:115946202 (110.5 MiB)  
  
eth0     Link encap:Ethernet  HWaddr A0:D3:C1:FA:25:C8  
         inet6 addr: fe80::a2d3:c1ff:fefa:25c8/64 Scope:Link  
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
         RX packets:2148683 errors:0 dropped:0 overruns:0 frame:0  
         TX packets:2030336 errors:0 dropped:0 overruns:0 carrier:0  
         collisions:0 txqueuelen:1000  
         RX bytes:338047294 (322.3 MiB)  TX bytes:136116027 (129.8 MiB)  
         Interrupt:32  
  
lo       Link encap:Local Loopback  
         inet addr:127.0.0.1  Mask:255.0.0.0  
         inet6 addr: ::1/128 Scope:Host  
         UP LOOPBACK RUNNING  MTU:16436  Metric:1  
         RX packets:6546 errors:0 dropped:0 overruns:0 frame:0  
         TX packets:6546 errors:0 dropped:0 overruns:0 carrier:0  
         collisions:0 txqueuelen:0  
         RX bytes:393148 (383.9 KiB)  TX bytes:393148 (383.9 KiB)  
  
virbr0   Link encap:Ethernet  HWaddr 52:54:00:C7:5E:B6  
         inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0  
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
         collisions:0 txqueuelen:0  
         RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Install VirtManager.

The final step is to install a desktop UI called VirtManager for managing virtual machines (VMs) through libvirt.

To install VirtManager.

```
$ sudo yum install virt-manager libvirt qemu-system-x86 openss
```

Launch VirtManager Remotely.

```
$ sudo yum install xauth
$ sudo vim /etc/ssh/sshd_config
```

Make sure we have the below line uncommented in `sshd_config`

```
X11Forwarding yes
```

Create wrapper for virt-manager.

Create a following executable wrapper script for virt-manager.

```
$ sudo -e /usr/bin/vm
```

Add below lines to the file `vm`

```
#!/bin/bash
xauth list | while read line; do
    sudo -i xauth add $line
done
sudo -i virt-manager
```

Let give it `exe` permissions.

```
$ sudo chmod +x /usr/bin/vm
```

Installing vnc-server.

Initial Installation.

Installing base libraries for `vncserver` to work.

```
sudo su
yum groupinstall Desktop
```

Further install

```
yum install gnome-core xfce4
yum install tigervnc-server
```

Now make the service on after every reboot

```
chkconfig vncserver on
```

Adding VNC user and setting vncpasswd.

Assuming we already have a user on the server `ahmed`

```
[root@ahmed-server ~]# su - ahmed
[ahmed@ahmed-server ~]$ vncpasswd
Password:*****
Verify:*****
[ahmed@ahmed-server ~]$
```

Now lets configure `vncserver`

```
sudo vim /etc/sysconfig/vncservers
```

Uncomment the line and add as below

```
VNCSERVERS="2:ahmed"
VNCSERVERARGS[2]="-geometry 1024x768"
```

Now restart `vncserver`.

```
[root@ahmed-server ~]# service vncserver restart
Shutting down VNC server: [ OK ]
Starting VNC server: 2:ahmed xauth: creating new authority file /home/ahmed/.Xauthority
```

```
New 'ahmed-server:2 (ahmed)' desktop is ahmed-server:2
```

```
Creating default startup script /home/ahmed/.vnc/xstartup
Starting applications specified in /home/ahmed/.vnc/xstartup
Log file is /home/ahmed/.vnc/ahmed-server:2.log
```

```
[ OK ]
```

```
[root@server1 ~]#
```

Logging in from Remote machine.

Install `vnc-viewer` and enter the ip address of `ahmed-server` followed `:2`, which is where the `vncserver` is listing.

```
10.10.18.36:2
```

And the password will be the `vncpasswd` set during the configuration above.

Screenshot.

Troubleshooting KVM and VirtManager setup.

- If you see the following error when attempting to launch VirtManager remotely, make sure that you use the wrapper script (`vm`) to launch it, as described above.

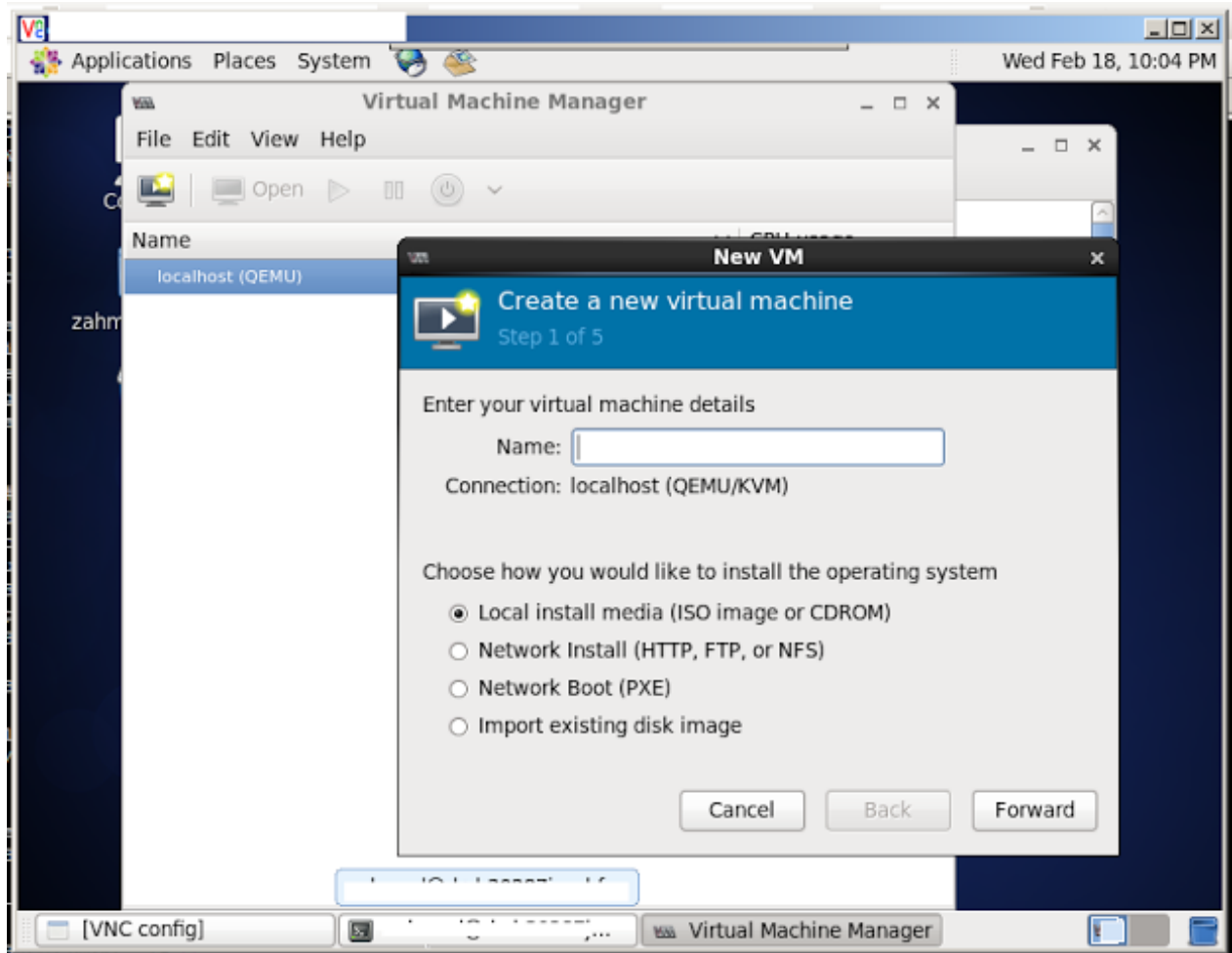


Figure 1: Screen Shot

```
X11 connection rejected because of wrong authentication.
Traceback (most recent call last):
  File "/usr/share/virt-manager/virt-manager.py", line 383, in
    main()
  File "/usr/share/virt-manager/virt-manager.py", line 286, in main
    raise gtk_error
RuntimeError: could not open display
```

b. If you see the following D-Bus error:

```
D-Bus library appears to be incorrectly set up; failed to read machine
uuid: UUID file '/var/lib/dbus/machine-id'
```

Then run the command below and reboot the host machine.

```
$ sudo sh -c 'dbus-uuidgen > /var/lib/dbus/machine-id'
```

c. If you have font issue while running VirtManager, install the following font, and relaunch it.

```
$ sudo yum install dejavu-lgc-sans-fonts
```

Upgrade CPU/RAM in KVM.

You can follow the following steps to increase memory size of your KVM virtual machine.

1. Update the configuration using command `sudo virsh edit <vm-name>`
2. reboot VM server.

Listing virtual servers.

```
[ahmed@ahmed-server ~]$ sudo virsh list
Id      Name                               State
-----
 8      VM-1                               running
12      VM-2                               running
13      VM-3                               running
15      VM-4                               running
```

Getting Information about VM.

```
[ahmed@ahmed-server ~]$ sudo virsh dominfo VM-1
Id:      8
Name:    VM-1
UUID:    588ff640-25be-9b18-5eb3-f93c471848e6
OS Type: hvm
State:   running
CPU(s):  4
CPU time: 503.3s
Max memory: 8388608 KiB
Used memory: 8388608 KiB
Persistent: yes
Autostart: disable
Managed save: no
Security model: none
Security DOI: 0
```


Edit Hardware for each VM.

```
[ahmed@ahmed-server ~]$ sudo virsh edit VM-1
Domain VM-1 XML configuration edited.
```

Here is now the XML looks like.

Before

```
<domain type='kvm'>
  <name>VM-1</name>
  <uuid>588ff640-25be-9b18-5eb3-f93c471848e6</uuid>
  <memory unit='KiB'>8388608</memory>
  <currentMemory unit='KiB'>8388608</currentMemory>
  <vcpu placement='static'>4</vcpu>
  <os>
    <type arch='x86_64' machine='rhel6.5.0'>hvm</type>
    <boot dev='hd' />
  </os>
  <features>
    <acpi />
    <apic />
    <pae />
  </features>
  ...
  ...
</domain>
```

After

```
<domain type='kvm'>
  <name>VM-1</name>
  <uuid>588ff640-25be-9b18-5eb3-f93c471848e6</uuid>
  <memory unit='KiB'>33554432</memory>
  <currentMemory unit='KiB'>33554432</currentMemory>
  <vcpu placement='static'>8</vcpu>
  <os>
    <type arch='x86_64' machine='rhel6.5.0'>hvm</type>
    <boot dev='hd' />
  </os>
  <features>
    <acpi />
    <apic />
    <pae />
  </features>
  ...
  ...
</domain>
```

Checking VM information.

```
[ahmed@ahmed-server ~]$ sudo virsh dominfo VM-1
Id:          -
Name:        VM-1
```

```
UUID:          588ff640-25be-9b18-5eb3-f93c471848e6
OS Type:       hvm
State:         shut off
CPU(s):        8
Max memory:    33554432 KiB
Used memory:   33554432 KiB
Persistent:    yes
Autostart:     disable
Managed save: no
Security model: none
Security DOI:  0
```

Once this is done restart the server. And we are done.

Interface Changes.

By default VMs take 10M as their transmission speed.

`virtio` - virtual driver which takes the speed of the bridge (which is 1G). `e1000` - making driver use the 1G speed (this again depends on the Bridge).

Currently have set the speed `virtio`.

Below are details on how to change it. All command needs to be run on the HOST machine (36) User Command below to change configuration on KVM.

```
sudo virsh edit <vm_name>
sudo virsh edit VM-1
```

This will bring up the XML which has the VM configuration which can be changed. Changes with take effect on reboot.

Change Interface Adding `<model type='virtio' />` below. By Default it takes 10M driver.

```
<interface type='bridge'>
  <mac address='52:54:00:60:08:62' />
  <source bridge='br0' />
  <model type='virtio' />
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0' />
</interface>
```

More Details here : <https://help.ubuntu.com/community/KVM/Networking#virtio>

Adding HDD to Virtual Machine.

Steps to create `virtio` HDD.

1. First Create a `image` using `qemu-img` command.
2. Add `image` to VM as a `virtio` HDD.
3. This will appear as `/dev/vda` in the VM, format it using the `mke2fs` command.
4. Add to `/etc/fstab`, so that the device can be mounted on boot up.

Details below.

First Create a image using qemu-img command.

Below is the command to create image.

```
[ahmed@ahmed-server ~]# qemu-img create VM-1-NEW-HDD.img 100G
```

This will create a HDD with 100GB disk.

Add image to VM as a virtio HDD.

Method 1 - Adding image from virt-manager UI.

Steps to Adding from the UI.

1. Select a VM, right-click -> open.
2. Next on the screen select view -> details.
3. In the new window select Add Hardware -> Storage -> select managed or other existing storage.
 - Select Device Type : virtio
 - Select Storage Format : raw

After Adding the HDD we can see it as below.

More Info Here : <http://unix.stackexchange.com/questions/92967/how-to-add-extra-disks-on-kvm-based-vm>

Method 2 - Adding image using virsh command.

To edit the VM configuration use below command.

```
[ahmed@ahmed-server ~]# virsh edit VM-1
```

Format

```
virsh edit <Virtual_Machine_NAME>
```

To add the image to the server add the below xml tag.

```
<disk type='file' device='disk'>
  <driver name='qemu' type='raw' cache='none' />
  <source file='/virtual_machines/images/VM-1-ADD.img' />
  <target dev='vda' bus='virtio' />
  <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0' />
</disk>
```

Now reboot the VM, after restart you will see the new device.

```
[ahmed@ahmed-server ~]# fdisk -l
```

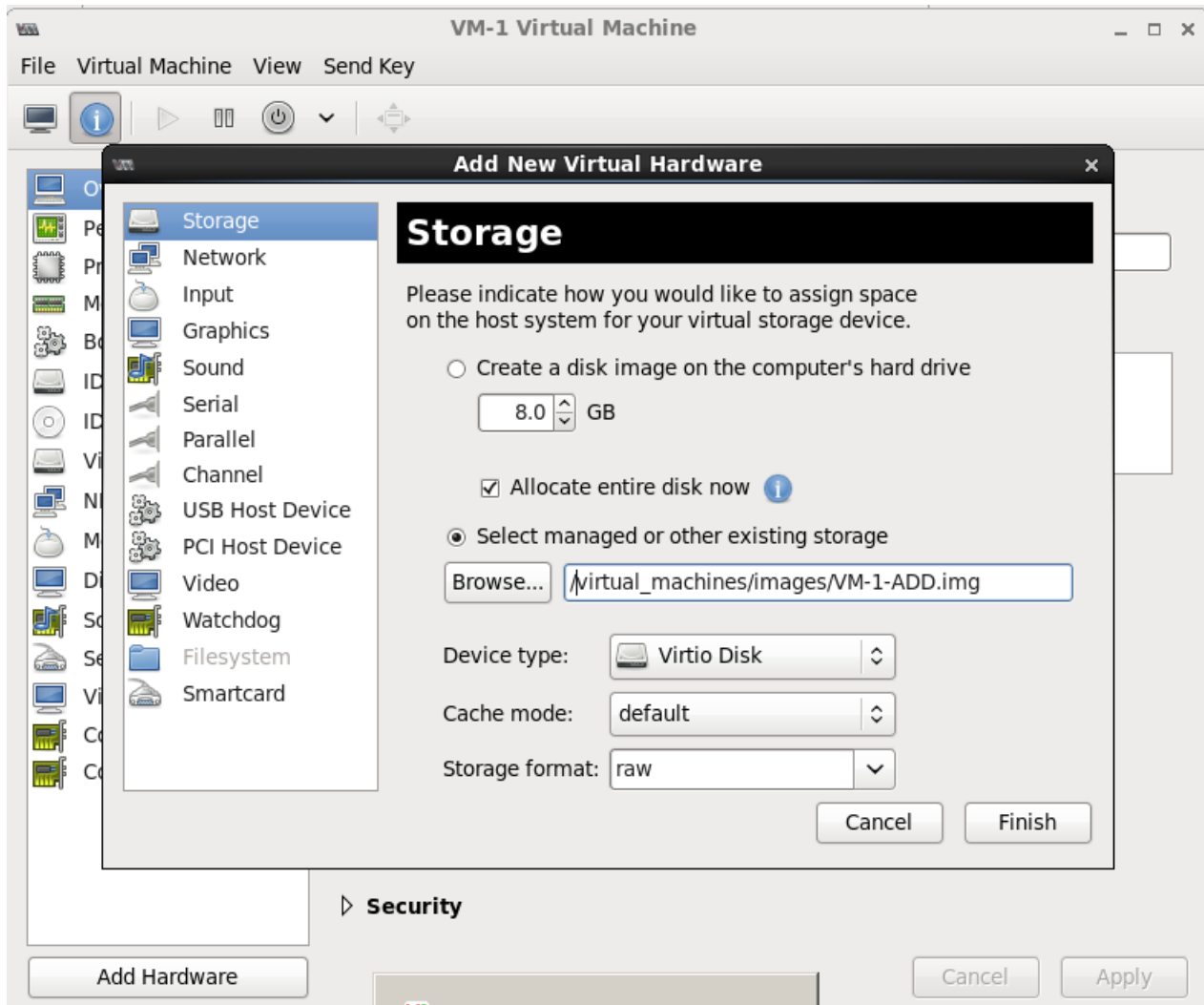


Figure 2: 'Adding New HDD'

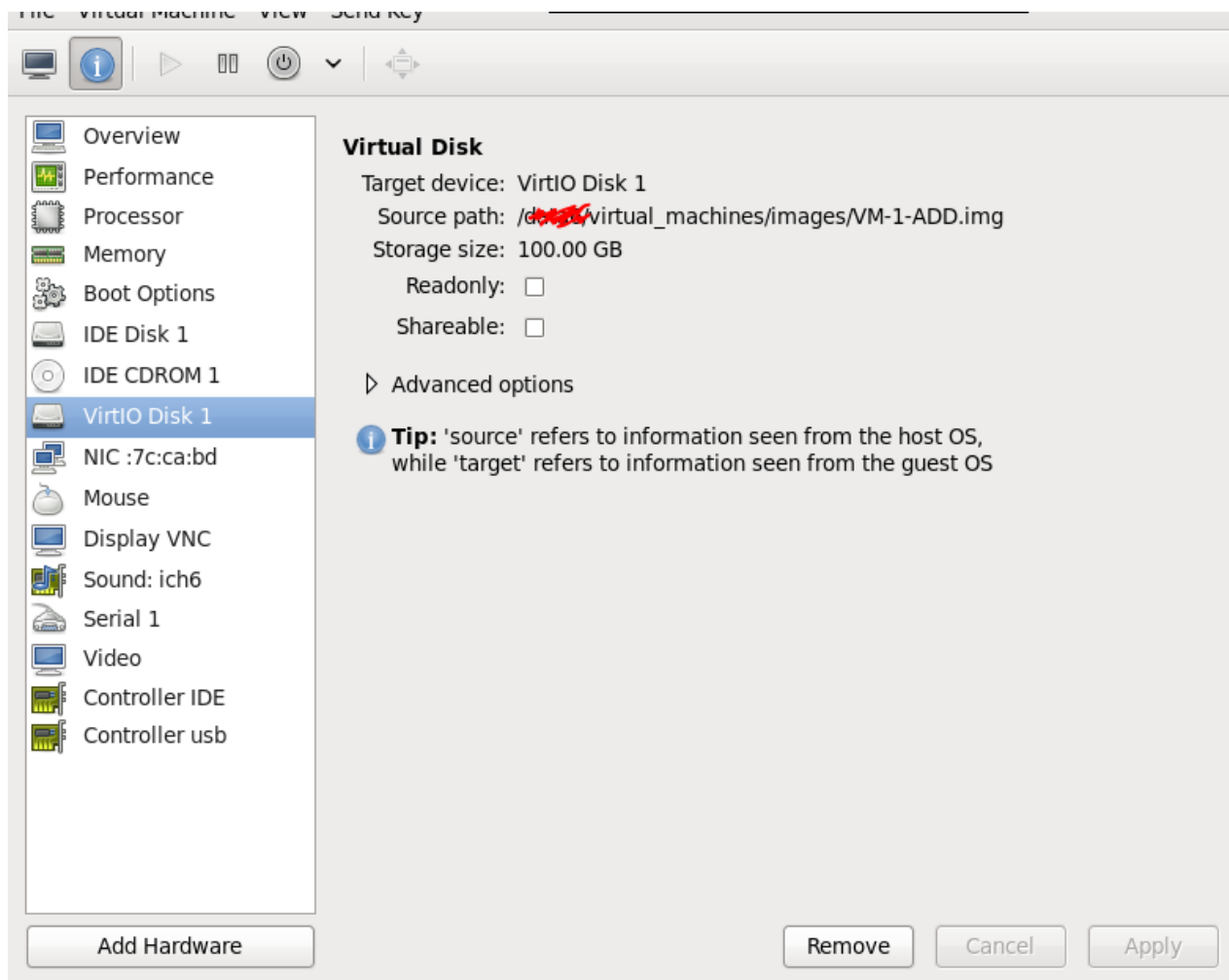


Figure 3: 'After Adding HDD'

```
Disk /dev/vda: 107.4 GB, 107374182400 bytes
16 heads, 63 sectors/track, 208050 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000
```

Figure 4: 'fdisk output'

Added image will appear as /dev/vda in the VM, format it using the mke2fs command.

Before we mount the device we need to format the device.

```
[ahmed@ahmed-server ~]# mke2fs -j /dev/vda
```

This will format the device.

Add to /etc/fstab, so that the device can be mounted on boot up.

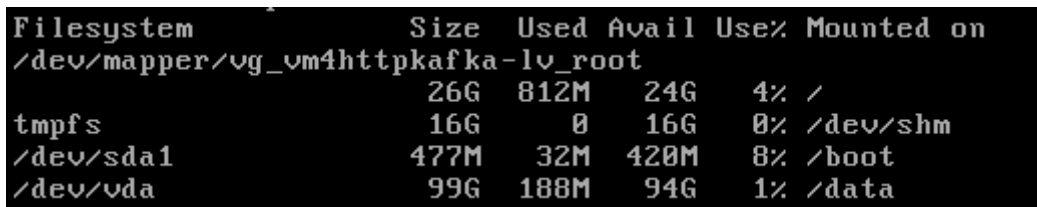
```
# /etc/fstab
#
# Column Details here : http://man7.org/linux/man-pages/man5/fstab.5.html
# -----
/dev/vda          /data          ext3          defaults      0 0
```

Now we can check by mounting.

```
[ahmed@ahmed-server ~]# mount -a
```

Check by running below command.

```
[ahmed@ahmed-server ~]# df -h
```



Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/vg_vm4httpkafka-lv_root	26G	812M	24G	4%	/
tmpfs	16G	0	16G	0%	/dev/shm
/dev/sda1	477M	32M	420M	8%	/boot
/dev/vda	99G	188M	94G	1%	/data

Figure 5: 'DF OUTPUT'

Useful Links.

<http://xmodulo.com/install-configure-kvm-centos.html>
<http://www.cyberciti.biz/faq/kvm-virtualization-in-redhat-centos-scientific-linux-6/>
<https://www.howtoforge.com/vnc-server-installation-centos-6.5>
<http://wiki.centos.org/HowTos/KVM>
<https://www.howtoforge.com/virtualization-with-kvm-on-a-centos-6.4-server-p4>

Redhat Documentation

<http://unix.stackexchange.com/questions/92967/how-to-add-extra-disks-on-kvm-based-vm>
http://www.techotopia.com/index.php/Adding_a_New_Disk_Drive_to_a_CentOS_6_System
<http://blog.zwiegnat.com/linux-server/add-new-hard-drive-to-centos-linux/>
<http://man7.org/linux/man-pages/man5/fstab.5.html>