



Inspiring Excellence

**CSE-491 Cloud Computing  
Assignment- 2**

**Everything is Virtual?**

**Submitted By:**

**Mehedi Hasan**

**Id: 17301046**

**Submitted To,**

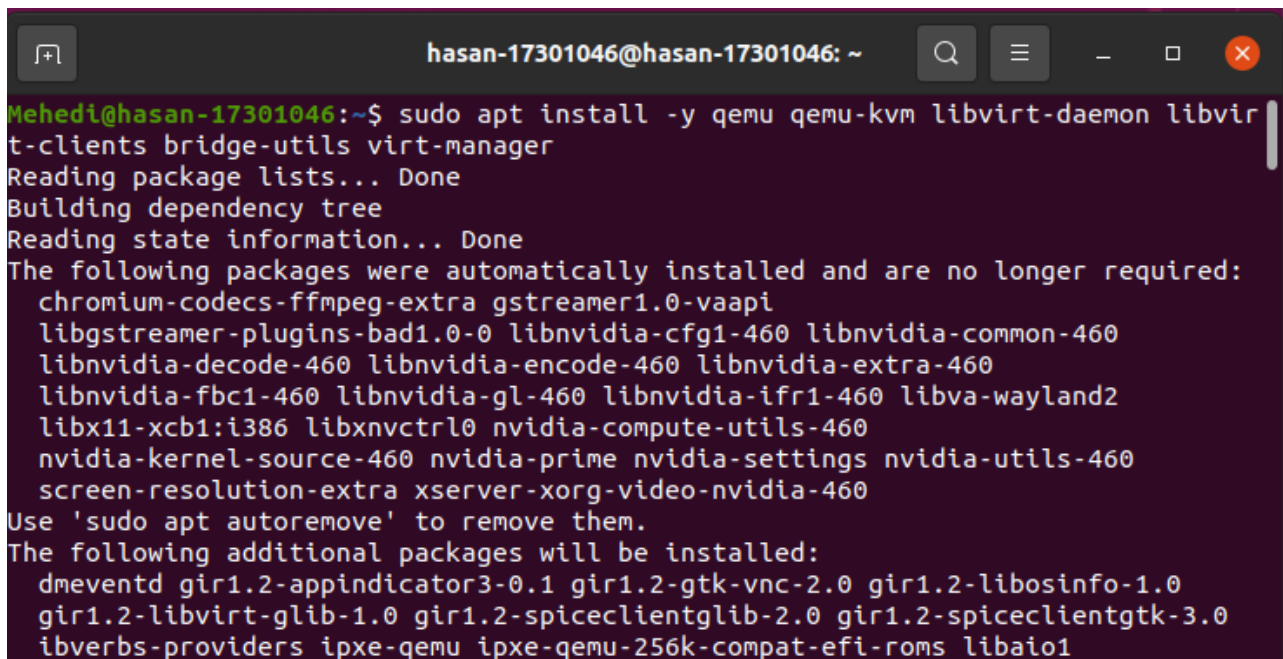
**Jannatun Noor Mukta**

# Assignment-2

## Part-1

**Install-Kvm:** KVM is kernel Based Virtual Machine. Which help us to install multiple os and server in One main Host machine. Suppose I have One Host machine which is ubuntu. But for some work purpose I need to install other OS. Instead of installing a new os using another drive and another partition I can easily install it in Virtual Machine with a very limited space, CPU and ram. I can do multiple work at a same time.

```
sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils-virt-manager
```

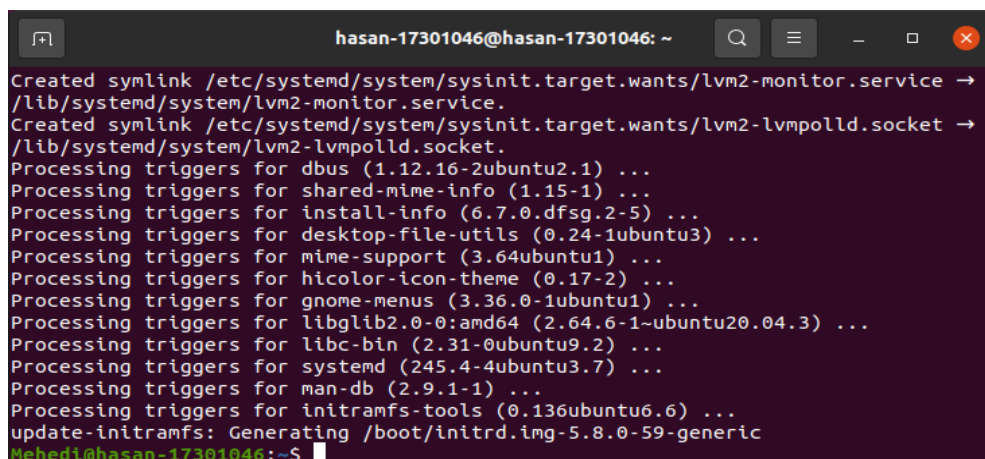
A terminal window with a dark background and light text. The window title is 'hasan-17301046@hasan-17301046: ~'. The user 'Mehedi' is at the prompt. They have run the command 'sudo apt install -y qemu qemu-kvm libvirt-daemon libvirt-clients bridge-utils virt-manager'. The terminal output shows the package lists being read, the dependency tree being built, and state information being read. It lists several packages that were automatically installed and are no longer required, such as 'chromium-codecs-ffmpeg-extra', 'gststreamer1.0-vaapi', and 'libnvidia-common-460'. It also lists additional packages that will be installed, including 'dmeventd', 'gir1.2-appindicator3-0.1', and 'gir1.2-gtk-vnc-2.0'.

```
Mehedi@hasan-17301046:~$ sudo apt install -y qemu qemu-kvm libvirt-daemon libvirt-clients bridge-utils virt-manager
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libnvidia-cfg1-460 libnvidia-common-460
  libnvidia-decode-460 libnvidia-encode-460 libnvidia-extra-460
  libnvidia-fbc1-460 libnvidia-gl-460 libnvidia-ifr1-460 libva-wayland2
  libx11-xcb1:i386 libxnvctrl0 nvidia-compute-utils-460
  nvidia-kernel-source-460 nvidia-prime nvidia-settings nvidia-utils-460
  screen-resolution-extra xserver-xorg-video-nvidia-460
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  dmeventd gir1.2-appindicator3-0.1 gir1.2-gtk-vnc-2.0 gir1.2-libosinfo-1.0
  gir1.2-libvirt-glib-1.0 gir1.2-spiceclientglib-2.0 gir1.2-spiceclientgtk-3.0
  ibverbs-providers ipxe-qemu ipxe-qemu-256k-compat-efi-roms libaio1
```

In order to Run Install Kvm we need to install few more packages also. Here qemu is type 2 hypervisor which help us to use virtual hardware.

Here libvirt is a collection of software that will help us to use virtual machine more convenient way. Such as storage and network management and so on.

Virt-manager is basically a gui of kvm. When everything install will be done we need a gui to install and virtualization. This virt-manager will show us the GUI.

A terminal window with a dark background and light text. The window title is 'hasan-17301046@hasan-17301046: ~'. The user 'Mehedi' is at the prompt. The terminal output shows the creation of symlinks for 'lvm2-monitor.service' and 'lvm2-lvmpolld.socket'. It then lists several triggers being processed for various packages, including 'dbus', 'shared-mime-info', 'install-info', 'desktop-file-utils', 'mime-support', 'hicolor-icon-theme', 'gnome-menus', 'libglib2.0-0:amd64', 'libc-bin', 'systemd', 'man-db', and 'initramfs-tools'. Finally, it shows the generation of the 'initrd.img' file.

```
Created symlink /etc/systemd/system/sysinit.target.wants/lvm2-monitor.service → /lib/systemd/system/lvm2-monitor.service.
Created symlink /etc/systemd/system/sysinit.target.wants/lvm2-lvmpolld.socket → /lib/systemd/system/lvm2-lvmpolld.socket.
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for shared-mime-info (1.15-1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for libglib2.0-0:amd64 (2.64.6-1-ubuntu20.04.3) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
Processing triggers for systemd (245.4-4ubuntu3.7) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for initramfs-tools (0.136ubuntu6.6) ...
update-initramfs: Generating /boot/initrd.img-5.8.0-59-generic
Mehedi@hasan-17301046:~$
```

KVM installation Done.

So, now we need to check our libvirt is activated or not. In order to check it we can run a command

```
hasan-17301046@hasan-17301046: ~
Mehedi@hasan-17301046:~$ sudo systemctl status libvirtd
[sudo] password for hasan-17301046:
● libvirtd.service - Virtualization daemon
   Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2021-07-02 10:51:58 +06; 25min ago
     TriggeredBy: ● libvirtd.socket
                  ● libvirtd-admin.socket
                  ● libvirtd-ro.socket
        Docs: man:libvirtd(8)
              https://libvirt.org
    Main PID: 12768 (libvirtd)
      Tasks: 19 (limit: 32768)
     Memory: 16.1M
    CGroup: /system.slice/libvirtd.service
            └─12768 /usr/sbin/libvirtd
              └─12951 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default
                └─12952 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default
```

sudo systemctl status libvirtd this will show us out libvirt successfully installed in our device.

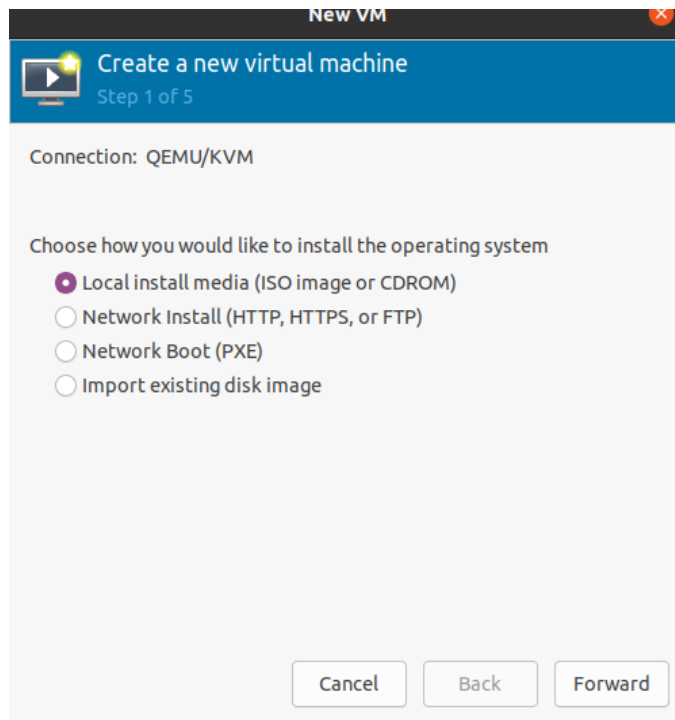
Now We need to Add a user to Libvirt and Kvm to use the virt-manager.

```
hasan-17301046@hasan-17301046: ~
জুলাই 02 10:51:59 hasan-17301046 dnsmasq-dhcp[12951]: DHCP, IP range 192.168.122.0/24
জুলাই 02 10:51:59 hasan-17301046 dnsmasq-dhcp[12951]: DHCP, sockets bound exclusively
Mehedi@hasan-17301046:~$ sudo systemctl is-active libvirtd
[sudo] password for hasan-17301046:
active
Mehedi@hasan-17301046:~$ sudo adduser 'mehedi@hasan-17301046' libvirt
adduser: The user `mehedi@hasan-17301046' does not exist.
Mehedi@hasan-17301046:~$ sudo adduser 'mehedi' libvirt
adduser: The user `mehedi' does not exist.
Mehedi@hasan-17301046:~$ sudo adduser 'hasan-17301046' libvirt
The user `hasan-17301046' is already a member of `libvirt'.
Mehedi@hasan-17301046:~$ sudo adduser '[hasan-17301046]' kvm
adduser: The user `[hasan-17301046]' does not exist.
Mehedi@hasan-17301046:~$ sudo adduser 'hasan-17301046' kvm
Adding user `hasan-17301046' to group `kvm' ...
Adding user hasan-17301046 to group kvm
```

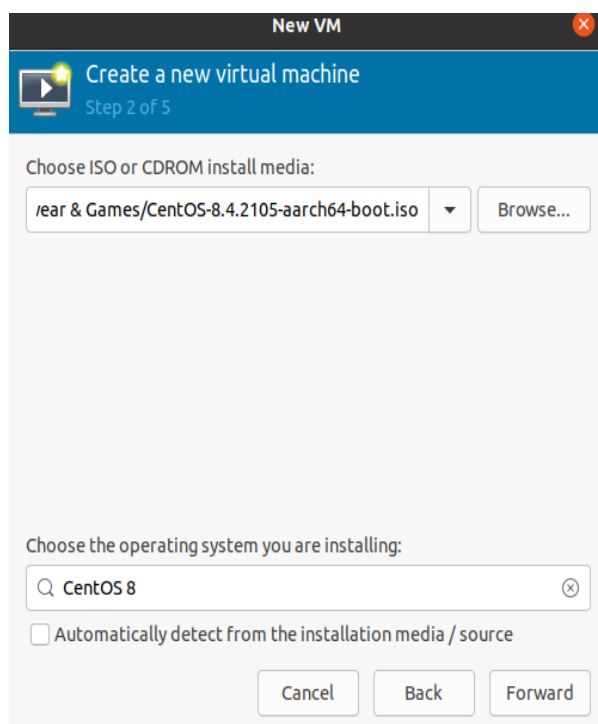
Now our Kvm is fully installed. If we command virt-manager It will show our kvm gui.

## Part-2:

Now we need to install an OS in our KVM using GUI method.



First we need to choose the ISO image that we wanted to install in our machine.



Iso Choosing done. Now we need to set the memory and Cpu

New VM

Create a new virtual machine  
Step 3 of 5

Choose Memory and CPU settings:

Memory: 2048 — +  
Up to 15896 MiB available on the host

CPUs: 2 — +  
Up to 4 available

Cancel Back Forward

We choose 2GB of Ram and 2 Cpu cores to run this machine.

New VM

Create a new virtual machine  
Step 4 of 5

☒ Enable storage for this virtual machine

☒ Create a disk image for the virtual machine

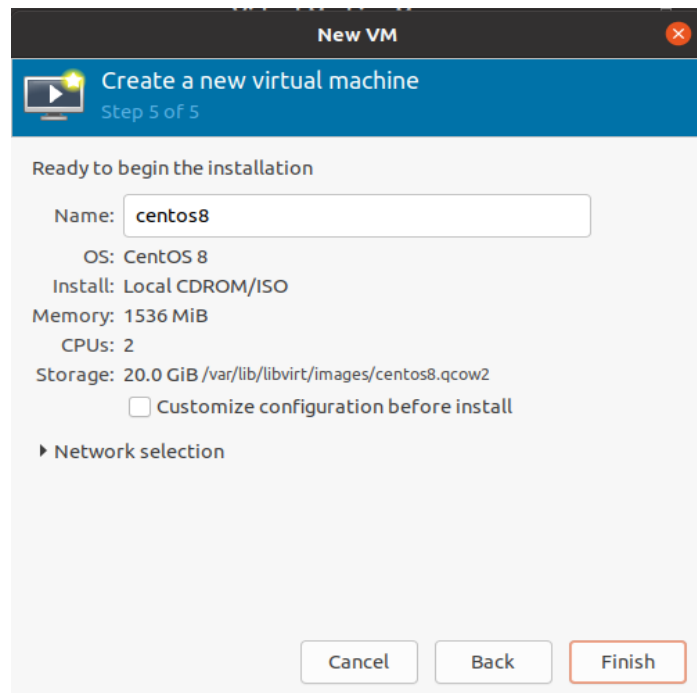
20.0 — + GiB  
86.3 GiB available in the default location

☐ Select or create custom storage

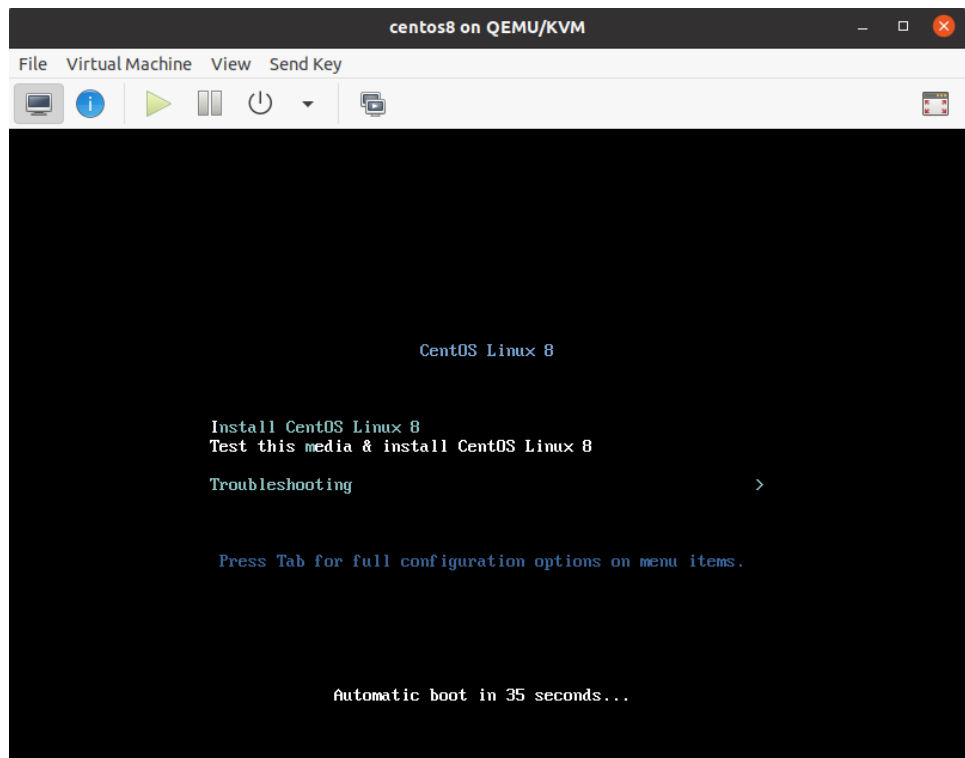
Manage...

Cancel Back Forward

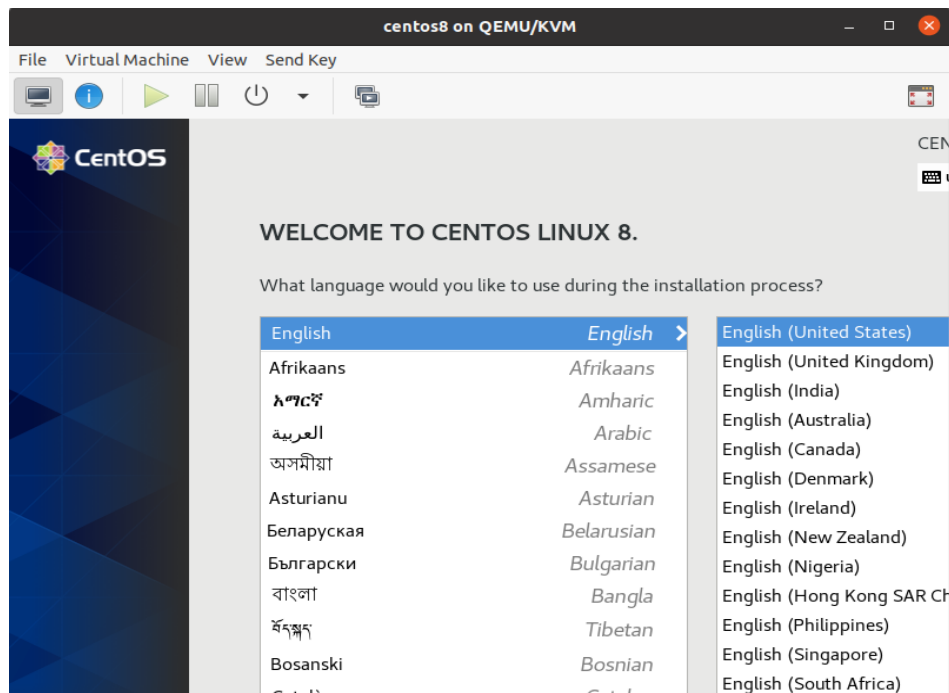
Storage we choose 20Gb for this Machine. We are going to install CentOS We can increase our storage up to our allocating space in host OS. As my Storage is 100GB so I can choose up to 86.3Gb for Virtualization.



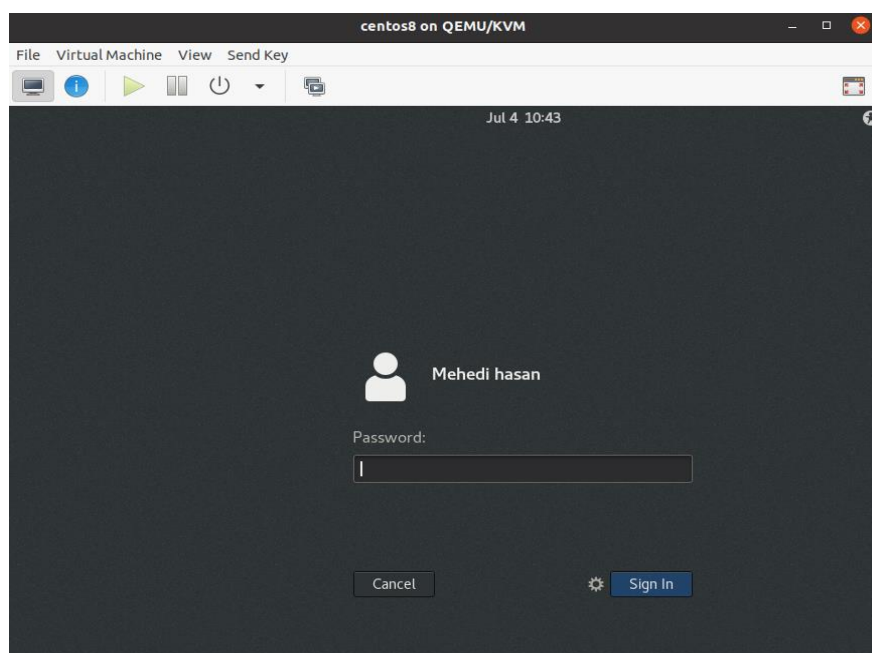
Now we need to give a name to Our OS. After that it will start Installation part.



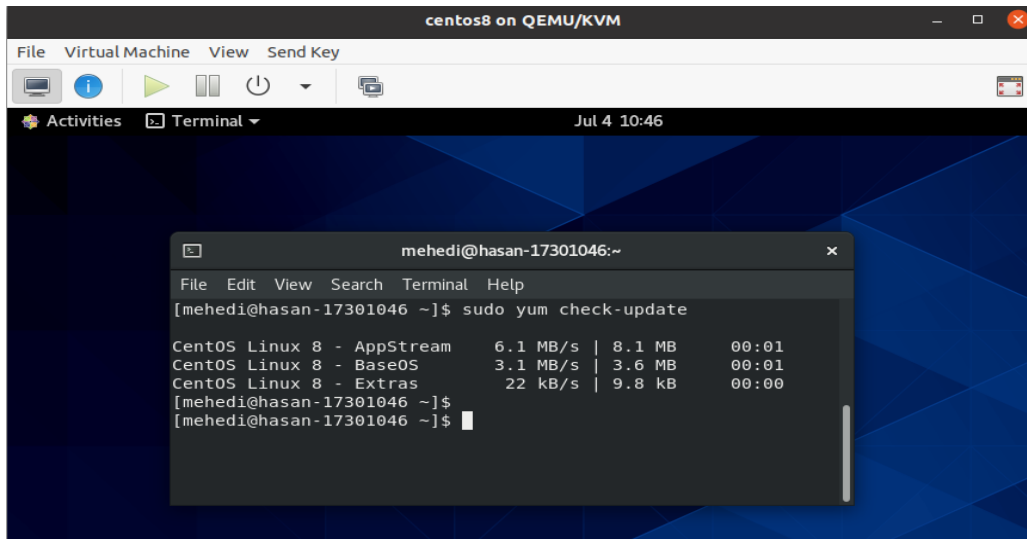
Now It's Exactly look like the way we setup Our OS in bios Method.



Our Installation is Done. Now we need to open it and update our os using Command.



Gui Installation Method Completed.



### Task-3 :

KVM Based Vm using “Virt-install”

1. Here virt-install --name Server-01 is our OS name. Previously we wrote it CentOS. Now we named it as a Server-01
2. we need to choose which type of OS it is. Like Linux, Debian, Windows etc
3. Os variant is which version of Os we are going to use it.
4. Allocating ram for this virtual machine
5. choose the disk size, bus and format
6. Graphics driver
7. hypervisor and iso choose
8. finally boot.

```
hasan-17301046@hasan-17301046: ~  
Mehedi@hasan-17301046:~$ virt-install --name server-01 \  
> --os-type linux \  
> --os-variant fedora-unknown \  
> --ram 2048 \  
> --disk /kvm/disk/server-01.img,device=disk,bus=virtio,size=20,format=qcow2 \  
> --graphics vnc,listen=0.0.0.0 \  
> --noautoconsole \  
> --hvm \  
> --cdrom /kvm/iso/Fedora-Workstation-Live-x86_64-34-1.2.iso \  
> --boot cdrom,hd  
  
Starting install...  
Allocating 'server-01.img' | 20 GB 00:06  
Domain installation still in progress. You can reconnect to  
the console to complete the installation process.  
Mehedi@hasan-17301046:~$
```



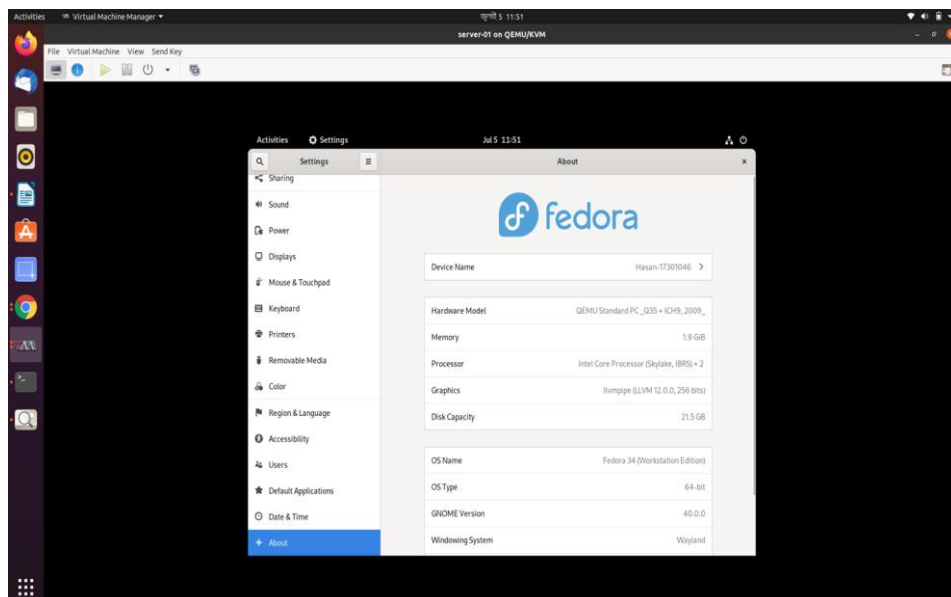
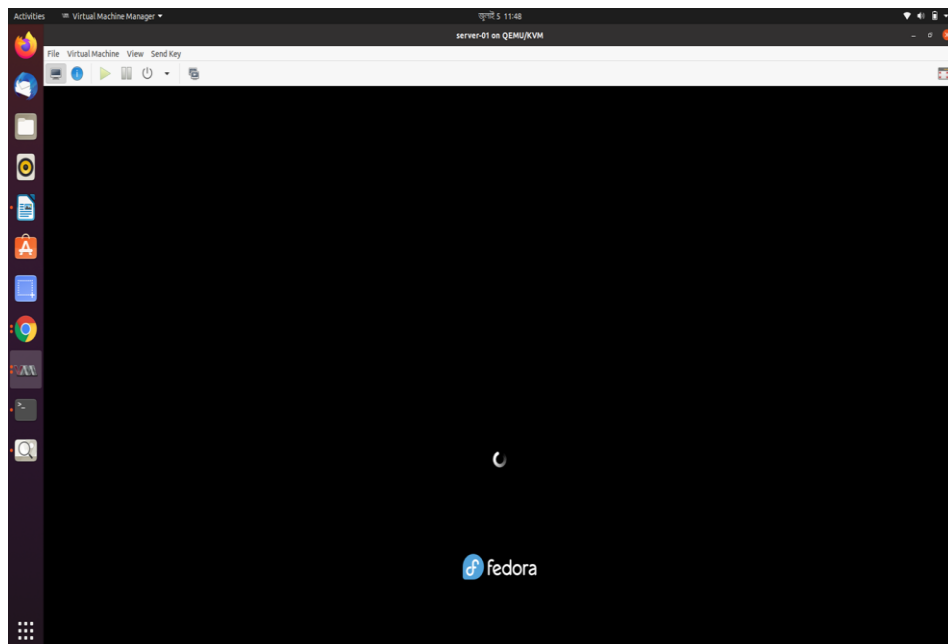
If all the process is right, we are going to see this message that console complete the installation process.

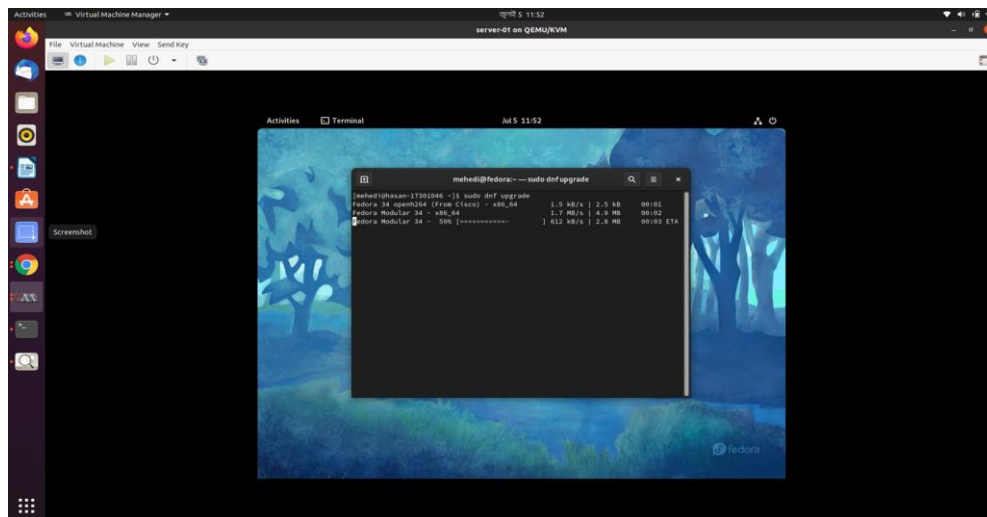
```
Mehedi@hasan-17301046:~$ virsh list --all
Id    Name         State
-----
1     server-01    running
-     centos8      shut off

Mehedi@hasan-17301046:~$
```

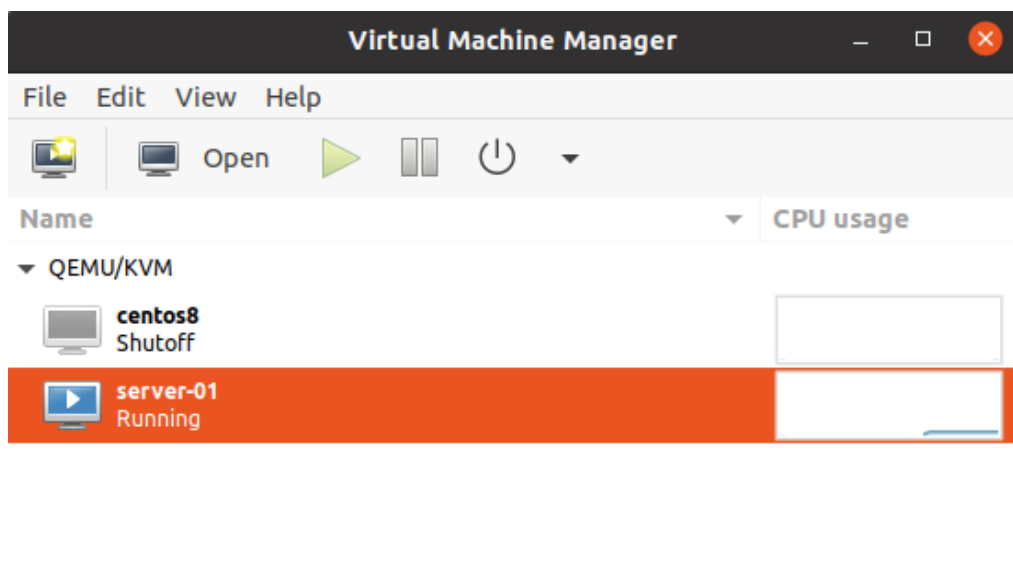
We can check it by `virsh list`. How many OS now installed and running in our Machine. Now If we are running virt manager we can see server-01 is there.

Now its time to installed our Fedora Os.





Fedora Successfully installed in Our Machine Using Cli Method.



**Part-5:** Make a shared folder between host os and Guest os.

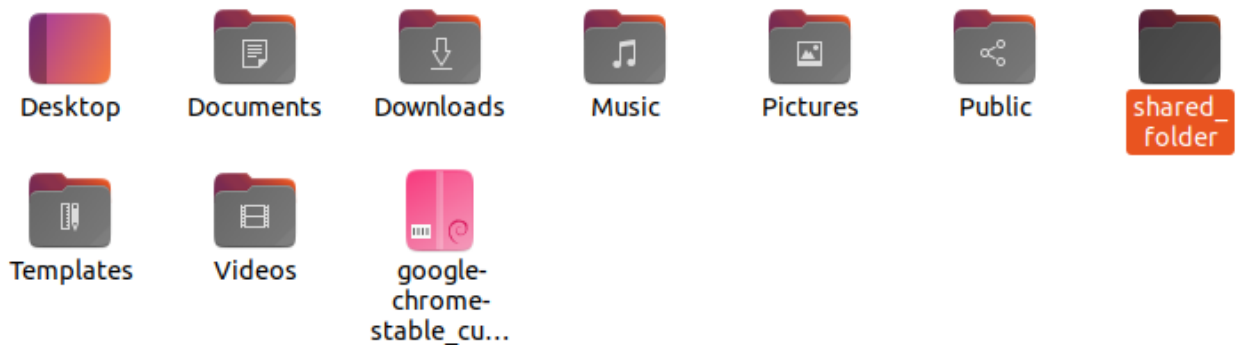
To Execute this operation, we are going to use Sambe. Sudo apt install Samba. It will install samba packages in our host machine.

```
mehedi@Hasan-17301046:~$ sudo apt install samba
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libnvidia-cfg1-460 libnvidia-common-460
  libnvidia-decode-460 libnvidia-encode-460 libnvidia-extra-460
samba-ad-dc.service is a disabled or a static unit, not starting it.
Processing triggers for ufw (0.36-6) ...
Processing triggers for systemd (245.4-4ubuntu3.7) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
```

After that we need to find out the location of samba's packages. Because we need to add a config file.

```
mehedi@Hasan-17301046:~$ sudo gedit /etc/samba/smb.conf
```

Now need to add couple of things in our samba conf file in order to give the access to sharing file. We also create a folder name **Shared\_folder** in our home directory. And add that directory and permission in smb.conf file.



```
243 [guest]
244 path = /home/mehedi/shared_folder
245 read only = no
246 guest ok = yes
247 guest only = yes
248
249
```

Now we need to give that folder permission so we can use it in our guest machine.

```
mehedi@Hasan-17301046: ~
mehedi@Hasan-17301046:~$ chmod 777 shared_folder/
mehedi@Hasan-17301046:~$ ls -l
total 81648
-rwxr-xr-x 2 mehedi mehedi 4096 জুলাই 7 19:03 Desktop
-rwxr-xr-x 2 mehedi mehedi 4096 জুলাই 8 12:20 Documents
-rwxr-xr-x 3 mehedi mehedi 4096 জুলাই 8 11:37 Downloads
-rw-rw-r-- 1 mehedi mehedi 83565736 জুলাই 17 08:09 google-chrome-stable_curre
t_amd64.deb
-rwxr-xr-x 2 mehedi mehedi 4096 জুলাই 7 19:03 Music
-rwxr-xr-x 2 mehedi mehedi 4096 জুলাই 8 12:32 Pictures
-rwxr-xr-x 2 mehedi mehedi 4096 জুলাই 7 19:03 Public
-rwxrwxrwx 2 mehedi mehedi 4096 জুলাই 8 12:32 shared_folder
-rwxr-xr-x 2 mehedi mehedi 4096 জুলাই 7 19:03 Templates
-rwxr-xr-x 2 mehedi mehedi 4096 জুলাই 7 19:03 Videos
mehedi@Hasan-17301046:~$
```

Its time to restart the samba to perform accordingly.

```
mehedi@Hasan-17301046: /etc/samba
mehedi@Hasan-17301046:/etc/samba$ sudo service smb restart
mehedi@Hasan-17301046:/etc/samba$
```

Finally we need ip of host machine to connect it in our Guest machine.

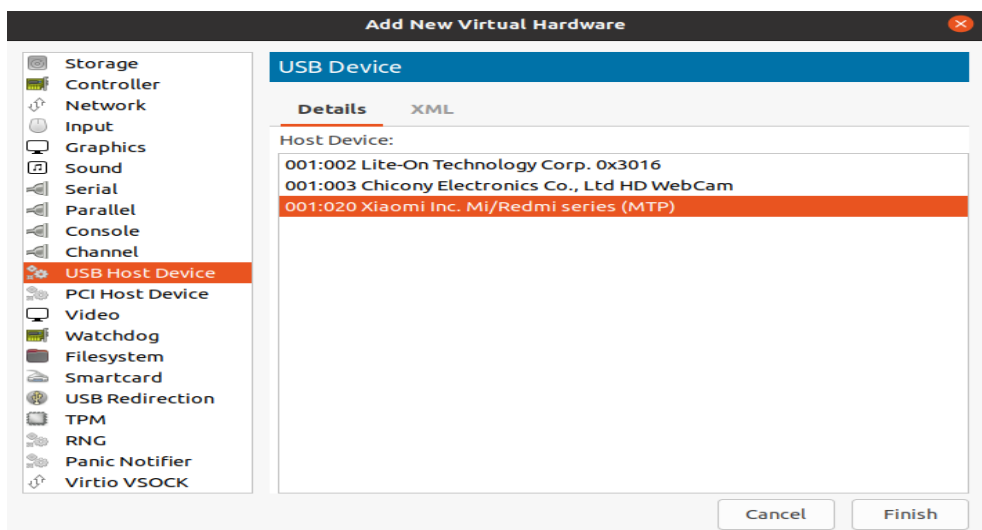
```
mehedi@Hasan-17301046: /etc/samba
mehedi@Hasan-17301046:/etc/samba$ sudo service smb restart
mehedi@Hasan-17301046:/etc/samba$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp3s0f1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 98:29:a6:47:ae:79 brd ff:ff:ff:ff:ff:ff
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 98:22:ef:77:dc:c9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.31.15/24 brd 192.168.31.255 scope global dynamic noprefixroute wlp2s0
        valid_lft 26260sec preferred_lft 26260sec
```

Ip addr will give us the ip of host machine. Now we need to add this ip in our guest machine using smb://192.168.31.15 if we add this address in our guest machine/connect it will show us the **guest** folder of Shared\_folder. And if we want, we can add the file, read the file. Both part of change can be accepted.



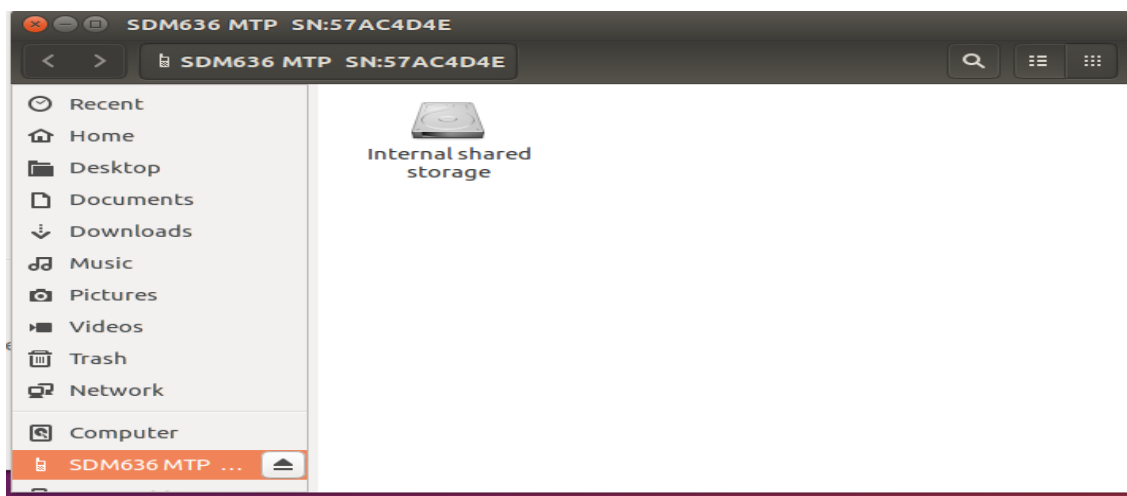
## Part-6

Connect Mobile in My virtual Machine. First we need to connect our device to the Main Host. Then we need to open our virtual machine and add a new hardware name USB Host Device. And choose our device name.

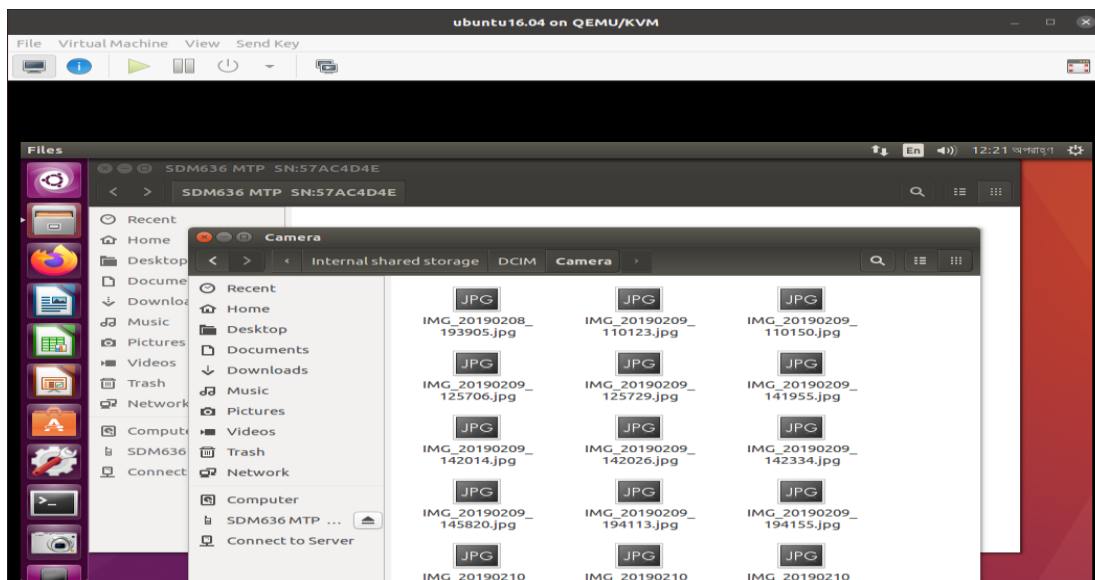


Then we need to install a packages name mtpfs Which is a Media Trasnfer Protocol extension for Ubuntu

```
mehedi@hasan-17301046: ~  
mehedi@hasan-17301046:~$ sudo apt-get install mtpfs  
[sudo] password for mehedi:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
mtpfs is already the newest version (1.1-5).  
The following package was automatically installed and is no longer required:  
  snapd-login-service  
Use 'sudo apt autoremove' to remove it.  
0 upgraded, 0 newly installed, 0 to remove and 11 not upgraded.  
mehedi@hasan-17301046:~$
```



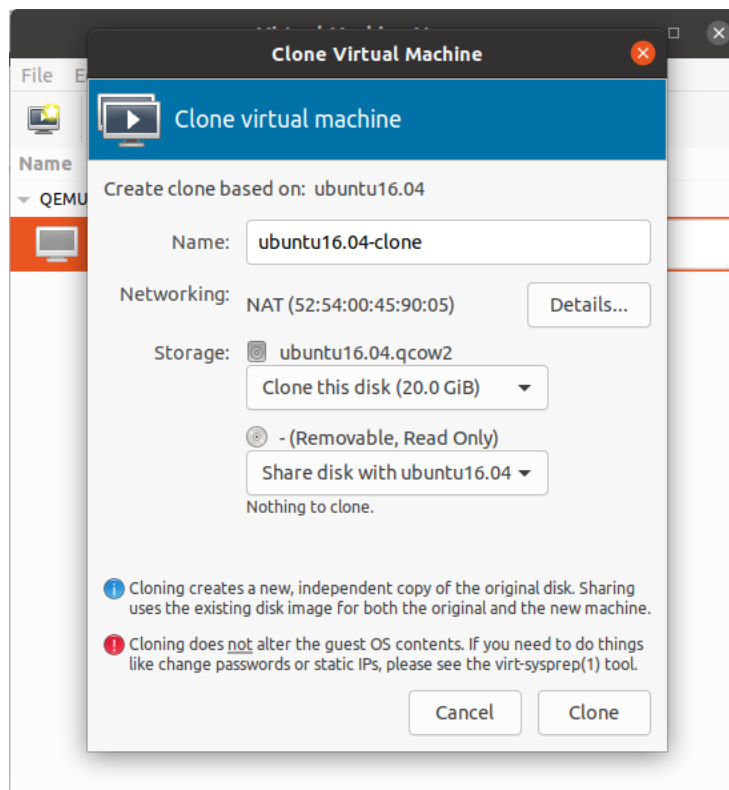
after that it will show something called Internal Shared Storage.



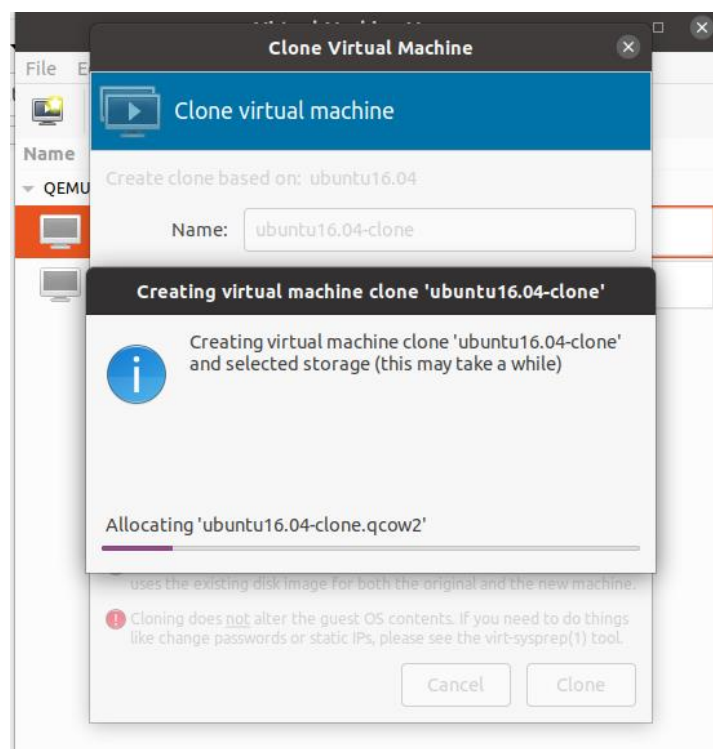
Finally we are able to connect our mobile into VM-Os.

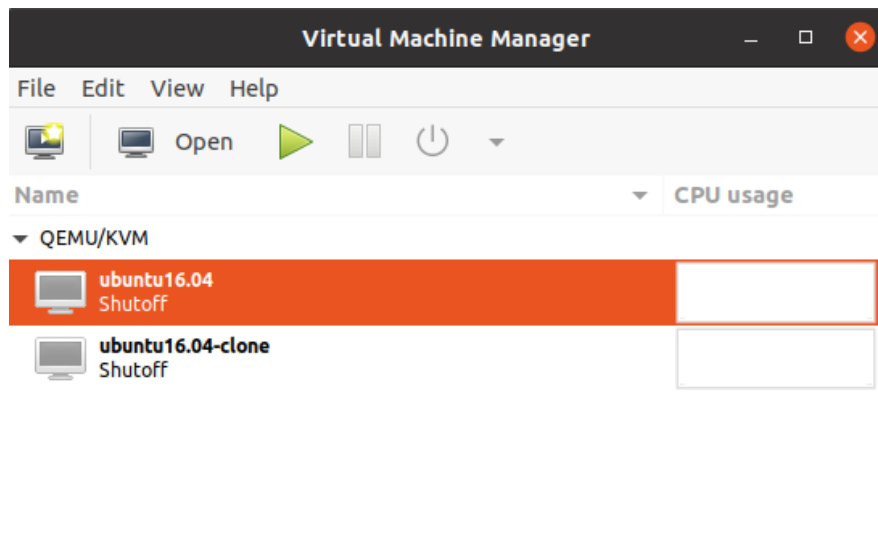
## Part-7: Clone a VM Using GUI and KVM Command.

First we need to open our KVM machine. It will show our available device. First we need to choose a device then click right button of our mouse and it will give a option called Clone. If we open it, we will get a option like this way

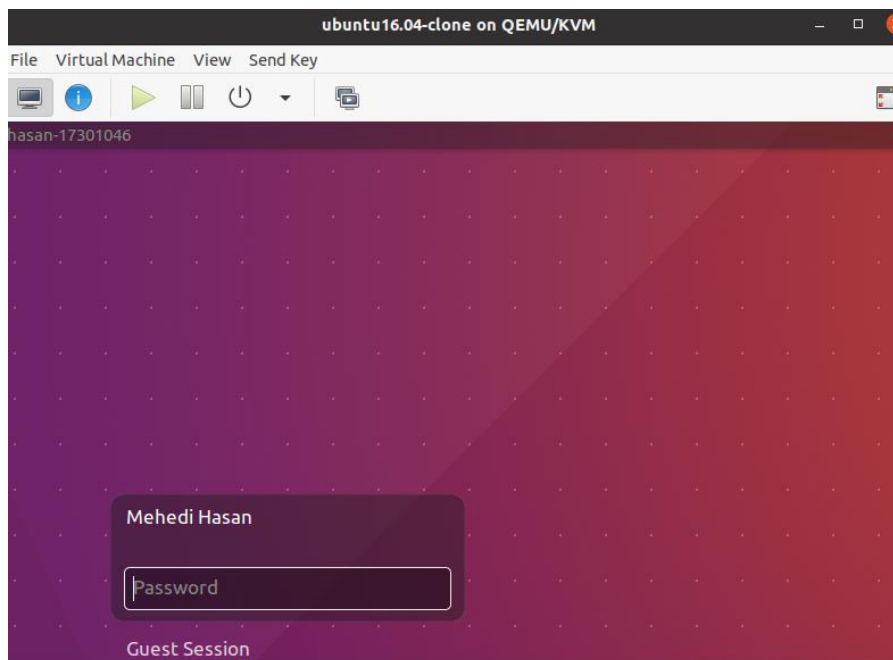


After that we just need to click on Clone. It will take some time and clone the Machine.





The Device is successfully cloned. Now if we can we can run our machine. And it will work fine.



We can Do the same thing using Kvm Command. We just need to write some of the command. First I am going to check how many device are in my kvm list. So for that I will run a command called Virsh list - -all. It will show all my Machine Devices and their State.

```
mehedi@Hasan-17301046: ~  
mehedi@Hasan-17301046:~$ virsh list --all  
Id      Name                State  
-----  
-       ubuntu16.04         shut off  
-       ubuntu16.04-clone   shut off  
mehedi@Hasan-17301046:~$
```

After That I will execute a command call virt-clone which will clone my existing machine. Need to choose the name and path and new name of my clone device.

```
mehedi@Hasan-17301046: ~  
mehedi@Hasan-17301046:~$ virsh list --all  
Id      Name                                State  
-----  
-       ubuntu16.04                        shut off  
-       ubuntu16.04-clone                  shut off  
  
mehedi@Hasan-17301046:~$ virt-clone --original ubuntu16.04 --name ubuntu-clone -  
f /var/lib/libvirt/images/ubuntu-clone.qcow2  
Allocating 'ubuntu-clone.qcow2' 3% [-          ] 58 MB/s | 783 MB 05:41 ETA
```

It will take some times to complete the clone and after that it will give a msg that clone successfully created.

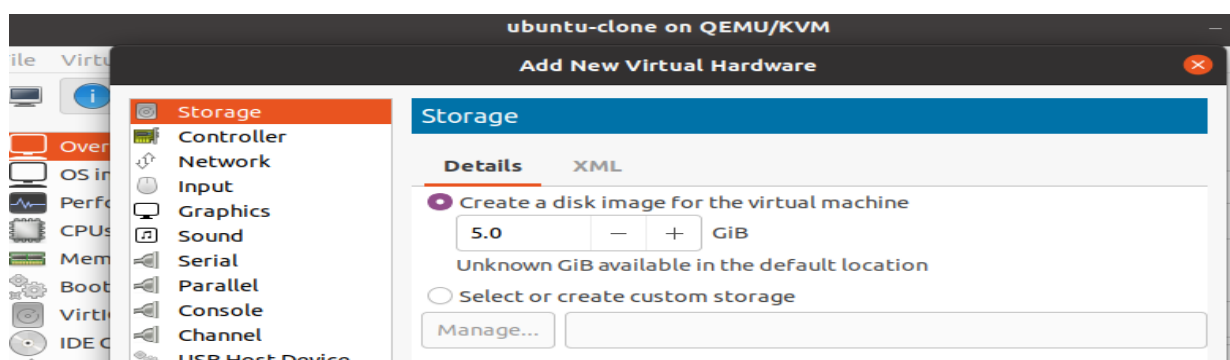
```
mehedi@Hasan-17301046: ~  
mehedi@Hasan-17301046:~$ virt-clone --original ubuntu16.04 --name ubuntu-clone -  
f /var/lib/libvirt/images/ubuntu-clone.qcow2  
Allocating 'ubuntu-clone.qcow2' | 20 GB 02:20  
  
Clone 'ubuntu-clone' created successfully.  
mehedi@Hasan-17301046:~$ virt list --all  
Command 'virt' not found, did you mean:  
  command 'vira' from snap vira (1.0.2)  
  command 'vit' from deb vit (2.0.0-1)  
  command 'dirt' from deb dput-ng (1.29)  
  
See 'snap info <snapname>' for additional versions.  
  
mehedi@Hasan-17301046:~$ virsh list --all  
Id      Name                                State  
-----  
-       ubuntu-clone                       shut off  
-       ubuntu16.04                       shut off  
-       ubuntu16.04-clone                 shut off  
  
mehedi@Hasan-17301046:~$
```

Here I can see my new clone device which I named it ubuntu-clone.

## Part-8: Adding Hard disk to My new Clones VM using Gui

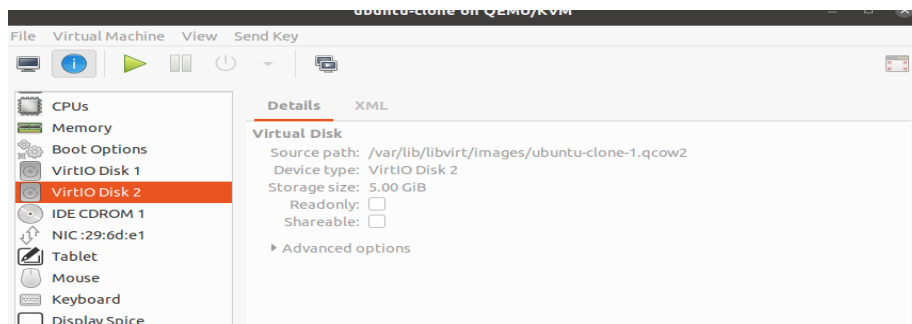
First need to open my Virtual Machine and choose the machine that I want to add an external disk.

Its helpful when the internal storage is full. I need to choose my machine and click on console > Add Hardware. Here we see a option Storage. Need to select my allocating space for my Machine.





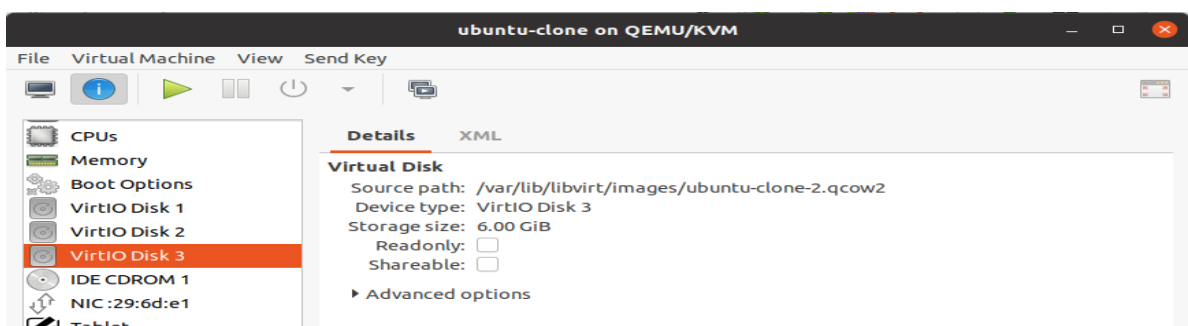
After Selecting is done it will create a separate disk for my Machine.



Now its time to check my Guest Cloned Machine, that it was successfully detect my external disk.

```
mehedi@hasan-17301046: ~  
mehedi@hasan-17301046:~$ lsblk  
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT  
vdb         252:16  0     5G  0 disk  
sr0         11:0    1 1024M  0 rom  
vda         252:0   0    23G  0 disk  
└─vda1      252:1   0    20G  0 part /  
mehedi@hasan-17301046:~$
```

Successfully Added One storage in my Virtual Machine. In this way I can Create another one and can added to my Cloned Virtual Machine.



## Part-9: Adding Storage Using CLI

First we need to choose the path of our images folder. Then we need to create a new disk for our machine. Which will work as a external storage. So we run qemu-img create. It will format the disk and make a partition.

```
mehedi@Hasan-17301046: /var/lib/libvirt/images  
mehedi@Hasan-17301046:~$ cd /var/lib/libvirt/images  
mehedi@Hasan-17301046:/var/lib/libvirt/images$ sudo qemu-img create -f raw ubuntu-clone-disk5-img 2G  
[sudo] password for mehedi:  
Formatting 'ubuntu-clone-disk5-img', fmt=raw size=2147483648
```

Now we need to attach our formatted storage in our machine. Virsh attach-disk will add our external disk to our target machine. VDD which is virtual disk drive.

```
mehedi@Hasan-17301046:/var/lib/libvirt/images$ virsh attach-disk ubuntu-clone --source /var/lib/libvirt/images/ubuntu-clone-disk5.img --target vdd --persistent
Disk attached successfully
```

In the same way we can create a new storage for our machine.

```
mehedi@Hasan-17301046:/var/lib/libvirt/images$ sudo qemu-img create -f raw ubuntu-clone-disk5.img 1.5G
Formatting 'ubuntu-clone-disk5.img', fmt=raw size=1610612736
```

Now its time to attached the disk in our virtual machine.

### Alternative way- Resize Virtual disk

First, we need to check the virtual block devices in a virtual server. To execute this operation we need to run virsh domblklist. It will show us all the available blocks for adding new storage.

```
mehedi@Hasan-17301046:~$ sudo virsh domblklist ubuntu-clone
Target    Source
-----
vda       /var/lib/libvirt/images/ubuntu-clone.qcow2
vdb       /var/lib/libvirt/images/ubuntu-clone-3.qcow2
hda       -
```

After That we need to check the info of image. Qemu-img info /Path will show us the full info of that machine.

```
mehedi@Hasan-17301046:~$ sudo qemu-img info /var/lib/libvirt/images/ubuntu-clone.qcow2
image: /var/lib/libvirt/images/ubuntu-clone.qcow2
file format: qcow2
virtual size: 23 GiB (24696061952 bytes)
disk size: 5.29 GiB
cluster_size: 65536
Format specific information:
  compat: 1.1
  lazy refcounts: true
  refcount bits: 16
  corrupt: false
mehedi@Hasan-17301046:~$
```

In my Storage its showing 23Gb. Now we are going to add 2GB more in my machine. So we execute qemu-img resize /path which will resize my desired storage in the .qcow2 file

```
mehedi@Hasan-17301046:~$ sudo qemu-img resize /var/lib/libvirt/images/ubuntu-clone.qcow2 +2G
Image resized.
mehedi@Hasan-17301046:~$
```

After that If we want, we can again check my info of storage. Again, We are executing qemu-img info. Before it was 23 GB now it will be 25GB

```
mehedi@Hasan-17301046:~$ sudo qemu-img info /var/lib/libvirt/images/ubuntu-clone.qcow2
image: /var/lib/libvirt/images/ubuntu-clone.qcow2
file format: qcow2
virtual size: 25 GiB (26843545600 bytes)
disk size: 5.29 GiB
cluster_size: 65536
```

To use this storage, we need to open my Guest cloned Machine. Which is virsh start

```
mehedi@Hasan-17301046:~$ sudo virsh start ubuntu-clone
Domain ubuntu-clone started
```

Then need to block the size that will be added in my cloned Machine.

```
mehedi@Hasan-17301046:~$ sudo virsh blockresize ubuntu-clone /var/lib/libvirt/images/ubuntu-clone.qcow2 25G
Block device '/var/lib/libvirt/images/ubuntu-clone.qcow2' is resized
mehedi@Hasan-17301046:~$
```

To Confirm my disk size we will run fdisk -l /path

```
mehedi@Hasan-17301046:~$ sudo fdisk -l /var/lib/libvirt/images/ubuntu-clone.qcow2
Disk /var/lib/libvirt/images/ubuntu-clone.qcow2: 5.3 GiB, 5679284224 bytes, 11092352 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
mehedi@Hasan-17301046:~$
```

Now if we want to check we can run lsblk in my guest machine. I can see it was added.

```
mehedi@hasan-17301046: ~
mehedi@hasan-17301046:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vdb         252:16  0     5G  0 disk
sr0         11:0    1 1024M  0 rom
vda         252:0   0    25G  0 disk
└─vda1      252:1   0    20G  0 part /
mehedi@hasan-17301046:~$
```

In this way We can add another storage in my VM machine.

```
mehedi@Hasan-17301046:~$ sudo qemu-img resize /var/lib/libvirt/images/ubuntu-clone.qcow2 +3G
Image resized.
```

This time we are going to add extra 3GB of storage.

```

mehedi@Hasan-17301046:~$ sudo virsh start ubuntu-clone
Domain ubuntu-clone started

mehedi@Hasan-17301046:~$ sudo virsh blockresize ubuntu-clone /var/lib/libvirt/images/ubuntu-clone.qcow2 +3G
Block device '/var/lib/libvirt/images/ubuntu-clone.qcow2' is resized
mehedi@Hasan-17301046:~$

```

After blocking it we can see it is resized. In this way we can add as much storage as we want (up to available storage)

#### Part-10: Migrating VM to another host.

```

mehedi@Hasan-17301046: ~
mehedi@Hasan-17301046:~$ sudo ls -l /var/lib/libvirt/images
[sudo] password for mehedi:
total 10589900
-rw----- 1 root root 21478375424 জুলাই 7 20:08 ubuntu16.04.qcow2
-rw----- 1 root root 5295308800 জুলাই 7 20:24 ubuntu-clone.qcow2
mehedi@Hasan-17301046:~$

```

Sudo ls will give us the output of available device file that we can migrate to another host.

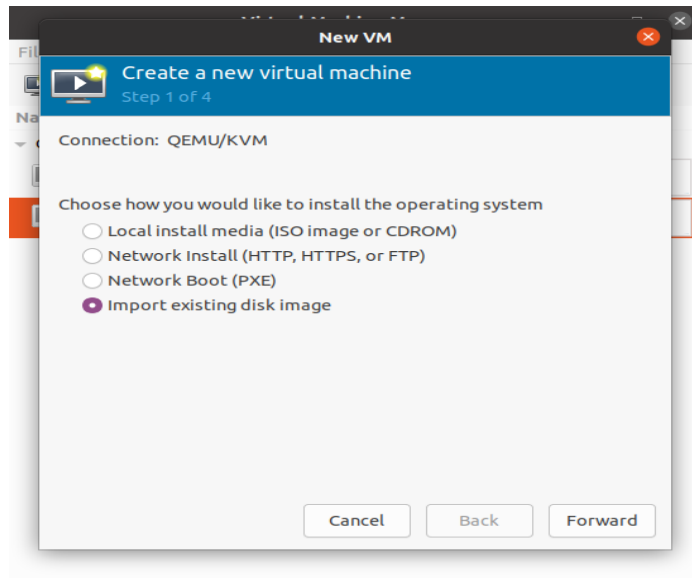


```

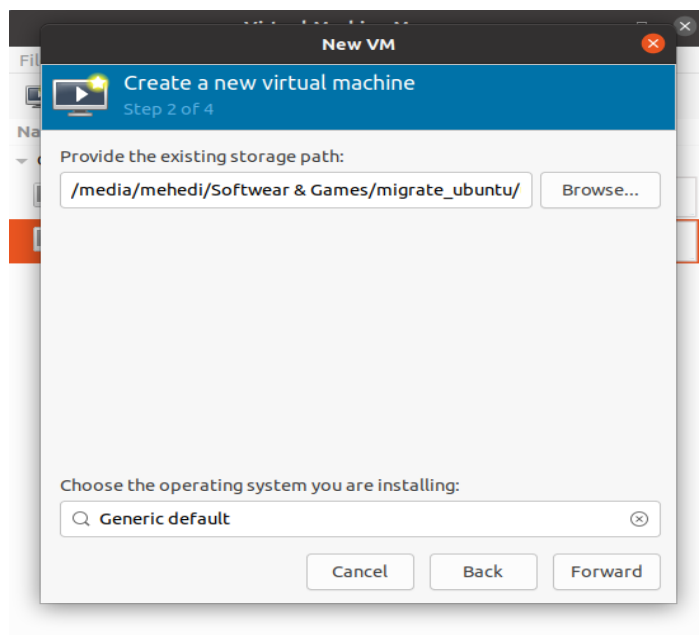
mehedi@Hasan-17301046: /media/mehedi/Softwear & Game...
mehedi@Hasan-17301046:/media/mehedi/Softwear & Games/migrate_ubuntu$ sudo cp /va
r/lib/libvirt/images/ubuntu16.04.qcow2 .
mehedi@Hasan-17301046:/media/mehedi/Softwear & Games/migrate_ubuntu$

```

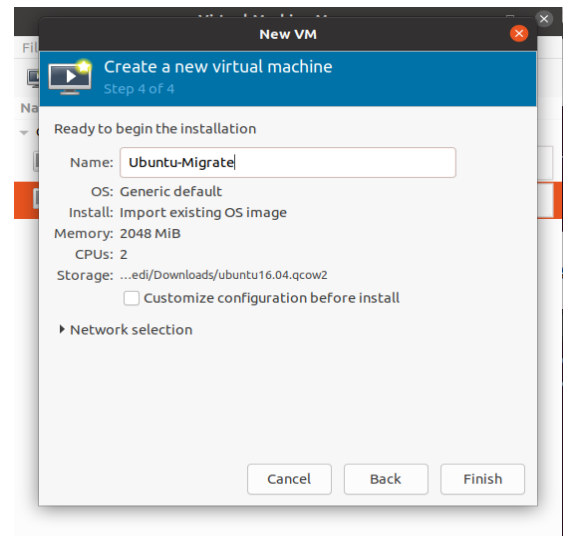
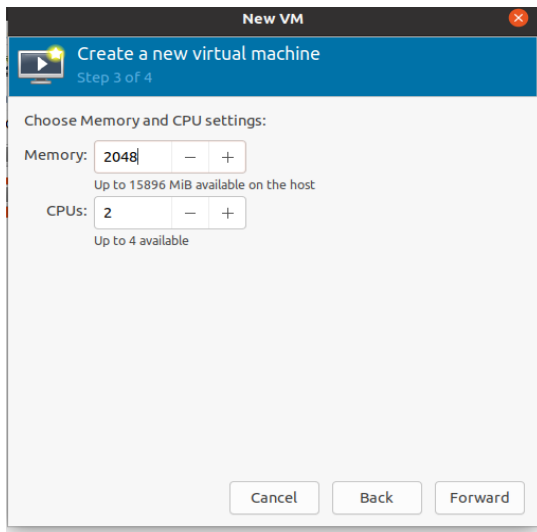
Sudo cp will copy the .qcow2 file to our guest location. we have chosen my external hdd to copy the file. Here we are getting a .qcow2 file after the command. And we can use this file to any other machine to host. Just need to install the KVM and add this file. No need to install the OS again.



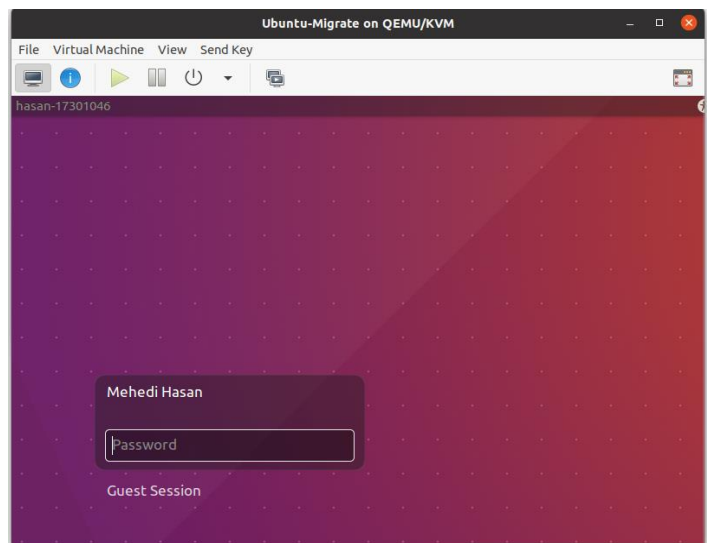
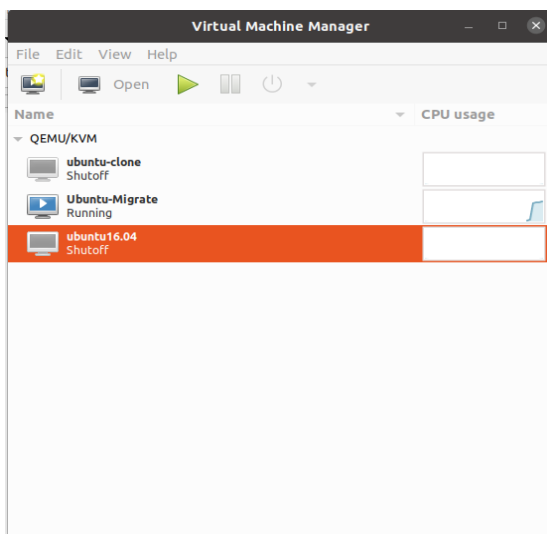
To Migrate this file, we need to import the disk image file from kvm. After that need to choose the path where it was located.



And if we click the forward button it will show us choose the storage size, ram and cpu for this machine.



Now our Migration is Done. Now we can see the new machine in our Virtual Machine.



So, Finally Migration is done. And now we can use my friend Migrated Machine in computer. And my friend can use my machine to their computer. Even we can also use our own migrated machine in our any alternative Device Also.