



Inspiring Excellence

CSE484(Cloud Computing)

Assignment 2: Everything is Virtual?

Name: Habibun Nabi Hemel

Id: 22241042

Semester: Spring24

Submitted to: Jannatun Noor

Pre Knowledge

The screenshot shows a Reddit post in the [r/qemu_kvm](#) subreddit. The post is titled "Running VMs with KVM as hypervisor in MacOS". It asks if it's possible to run KVM in macOS using MacPorts and if it has the same performance advantages as KVM in Linux. The post has 1 upvote and 2 comments. The first comment, by [Drwankingstein](#), says the short answer is no because OS X doesn't use the Linux kernel. The second comment, by [Eldiabolo18](#), says QEMU supports this feature and is fast. Both comments have 2 upvotes.

KVM is not run directly on Windows and macOS. So, we have an option of nested virtualization. However some hardware/PCs do not support nested virtualization so we need to change the BIOS which is different from one hardware. such as the Asus laptop will have one type of BIOS some other laptops will have others.

Worst case: The laptop will not support nested virtualization not possible such as its almost not possible for the Mac m1 user to run KVM in nest virtualization.

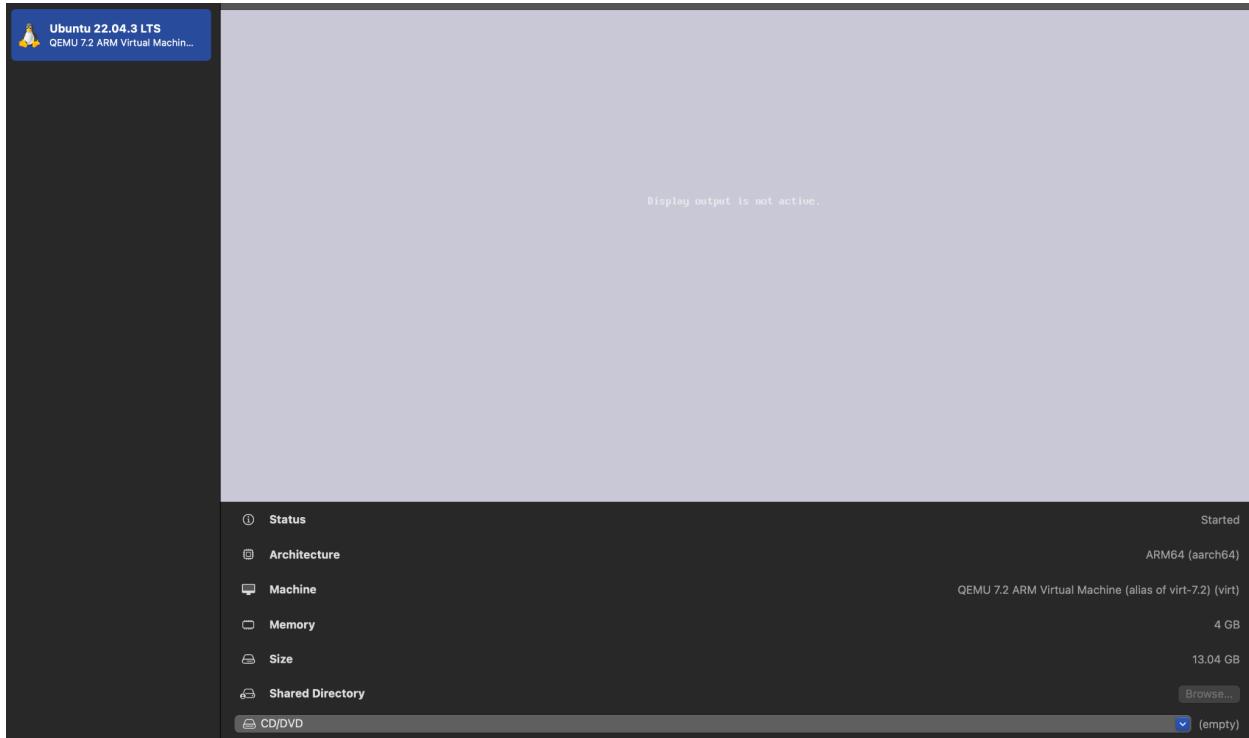
Let me check for the Macbook M1 chip that its ARM(OS) and hardware system supports the nested virtualization or not.

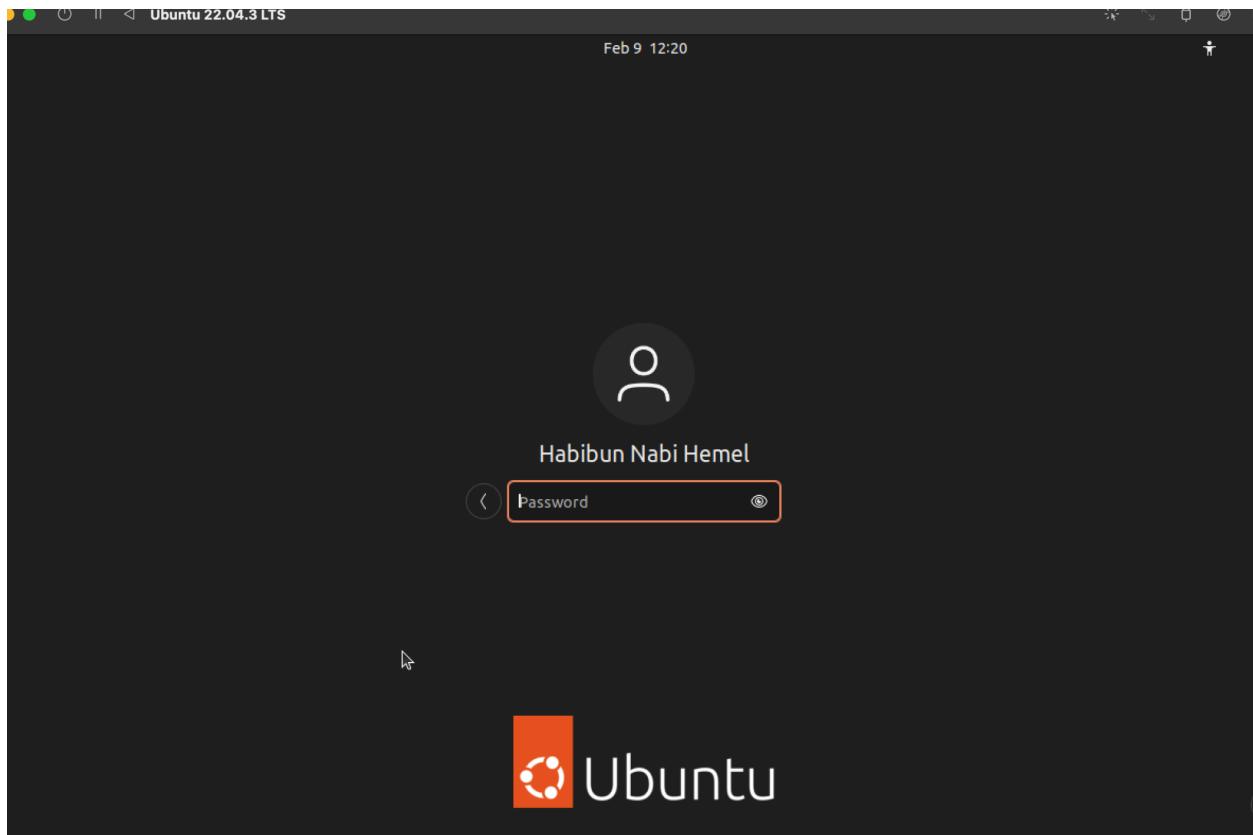
UTM is an opensource hypervisor for Mac users (similar to KVM)

Windows. Linux. Meet **Apple Silicon**.

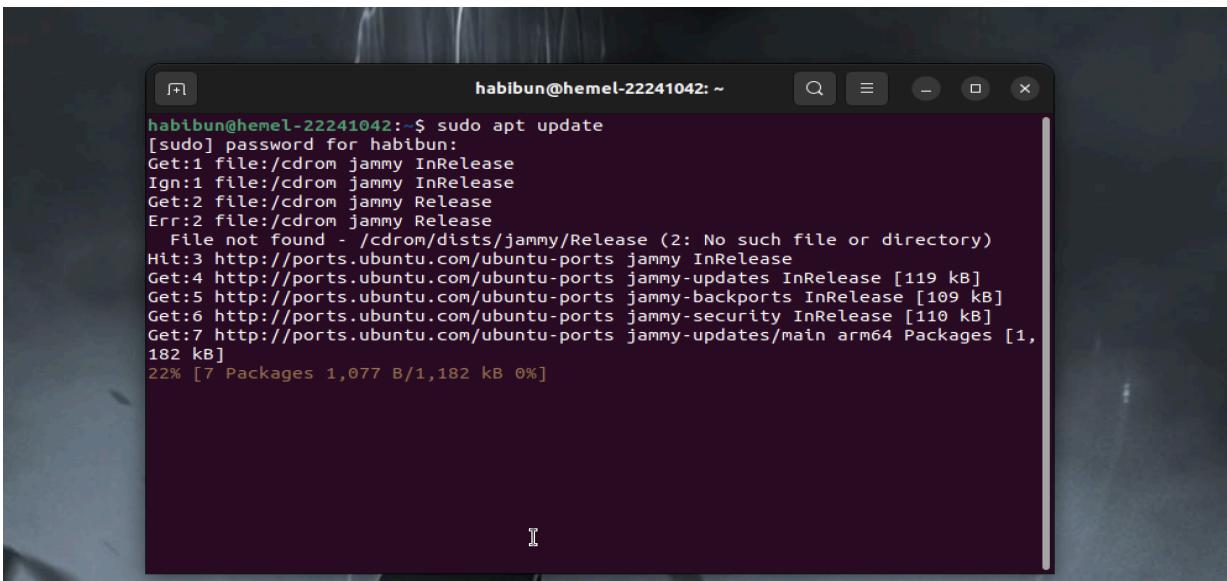
UTM employs Apple's Hypervisor virtualization framework to run ARM64 operating systems on Apple Silicon at near native speeds. On Intel Macs, x86/x64 operating system can be virtualized. In addition, lower performance emulation is available to run x86/x64 on Apple Silicon as well as ARM64 on Intel. For developers and enthusiasts, there are dozens of other emulated processors as well including: ARM32, MIPS, PPC, and RISC-V. Your Mac can now truly run anything.

Now i will download the UTM Software and create a Virtual Machine on it





Now I will check the mac m1 chip supports the nested virtual machine or not
Before Installing KVM directly we need to do some checks which are pre-requisites. Firstly, ensure the current (hosted OS) is up to date. Launch the terminal and update your local package index to get off the ground. Run the command: **sudo apt update**



```
habibun@hemel-22241042:~$ sudo apt update
[sudo] password for habibun:
Get:1 file:/cdrom jammy InRelease
Ign:1 file:/cdrom jammy InRelease
Get:2 file:/cdrom jammy Release
Err:2 file:/cdrom jammy Release
  File not found - /cdrom/dists/jammy/Release (2: No such file or directory)
Hit:3 http://ports.ubuntu.com/ubuntu-ports jammy InRelease
Get:4 http://ports.ubuntu.com/ubuntu-ports jammy-updates InRelease [119 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports jammy-backports InRelease [109 kB]
Get:6 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease [110 kB]
Get:7 http://ports.ubuntu.com/ubuntu-ports jammy-updates/main arm64 Packages [1,182 kB]
22% [7 Packages 1,077 B/1,182 kB 0%]
```

```
habibun@hemel-22241042:~$ sudo apt upgrade
[sudo] password for habibun:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
  linux-headers-5.15.0-94 linux-headers-5.15.0-94-generic
  linux-image-5.15.0-94-generic linux-modules-5.15.0-94-generic
  linux-modules-extra-5.15.0-94-generic
The following packages have been kept back:
  open-vm-tools
The following packages will be upgraded:
  libssl3 linux-generic linux-headers-generic linux-image-generic openssl
5 upgraded, 5 newly installed, 0 to remove and 1 not upgraded.
5 standard LTS security updates
Need to get 0 B/122 MB of archives.
After this operation, 617 MB of additional disk space will be used.
Do you want to continue? [Y/n] ■
```

Before you proceed any further, you need to check if your CPU supports KVM virtualization. To doing that [egrep -c '\(vmx|svm\)' /proc/cpuinfo](#)

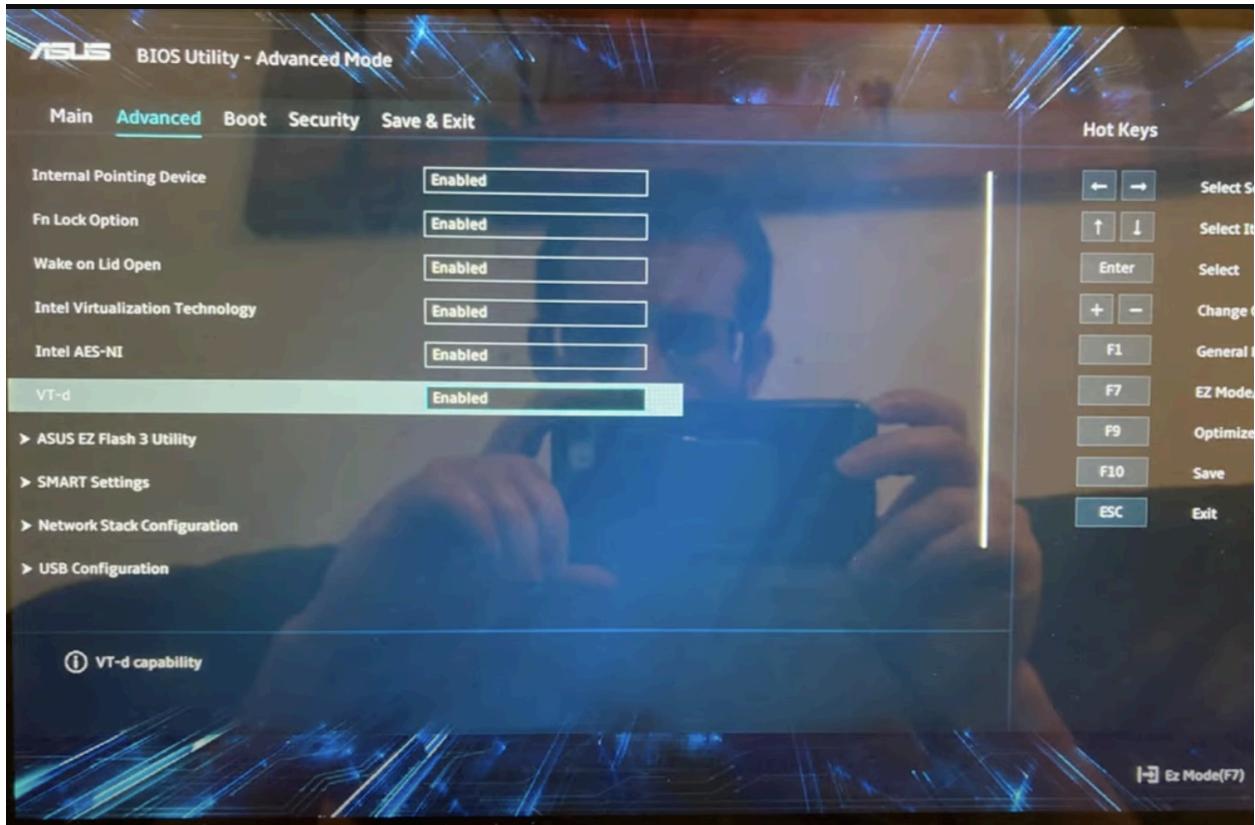
Run the following command in the terminal to check if KVM is compatible. If the command returns a non-zero value, it means that the CPU supports virtualization.

```
habibun@hemel-22241042:~$ egrep -c '(vmx|svm)' /proc/cpuinfo
0
habibun@hemel-22241042:~$
```

OPPS!! So for my case its show 0 which means i need to do some extra work to make it supported.

So my machine is not supporting that so I have the last option that is doing some change in the BIOS setting .lets do it This process can be differ from one to another like asus laptop users will do some change according to their manul ,dell users will do in another way ,mac users will do it another way

ASUS users can do it in this way:



But for mac users, it is not straightforward forward and its dangerous to do change in mac's BIOS:

WARNING.

Opening BIOS In Mac

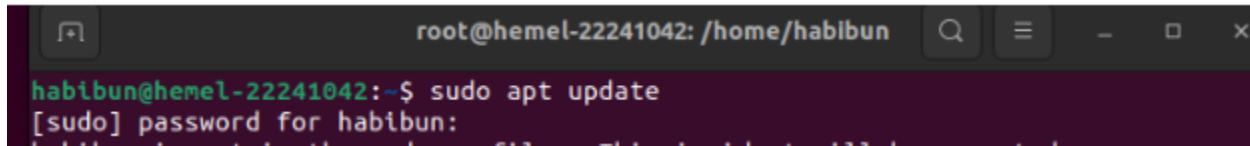
Technically, Macbooks don't come with BIOS, but they have a similar boot firmware called Open Firmware or Extensible Firmware Interface. They serve the same purpose as BIOS. You should be extra careful while you are making changes to this interface. One mistake can damage your Macbook permanently.

Let's try another Device

TASK1. Install KVM

1) Update Ubuntu 22.04

To begin, open the terminal and refresh your local package index using the following command. So, first, run the update command: `sudo apt update`

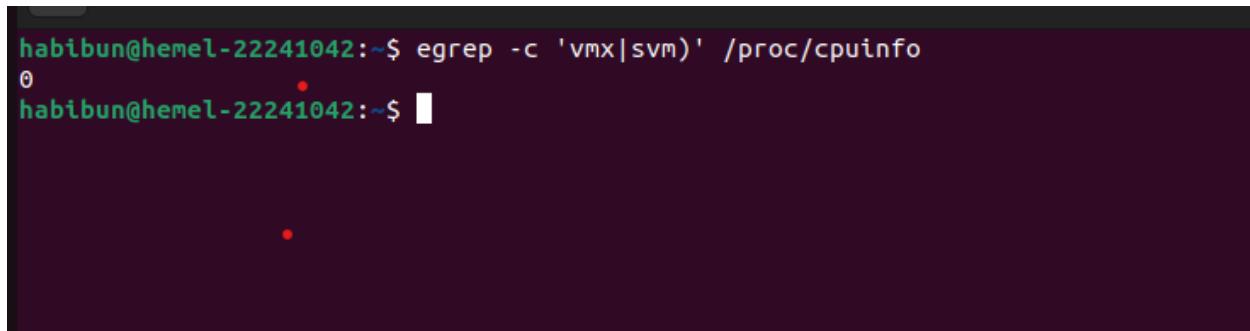


```
root@hemel-22241042: /home/hibibun
habibun@hemel-22241042:~$ sudo apt update
[sudo] password for habibun:
```

2) Check if Virtualization is enabled

Before you proceed, it's essential to verify whether your CPU supports KVM virtualization. To do so, execute the following command. If the output value is greater than 0, it indicates that virtualization is enabled. Otherwise, if the output is 0 or less, virtualization is disabled, and you'll need to enable it.

Then run : `egrep -c '(vmx|svm)' /proc/cpuinfo`

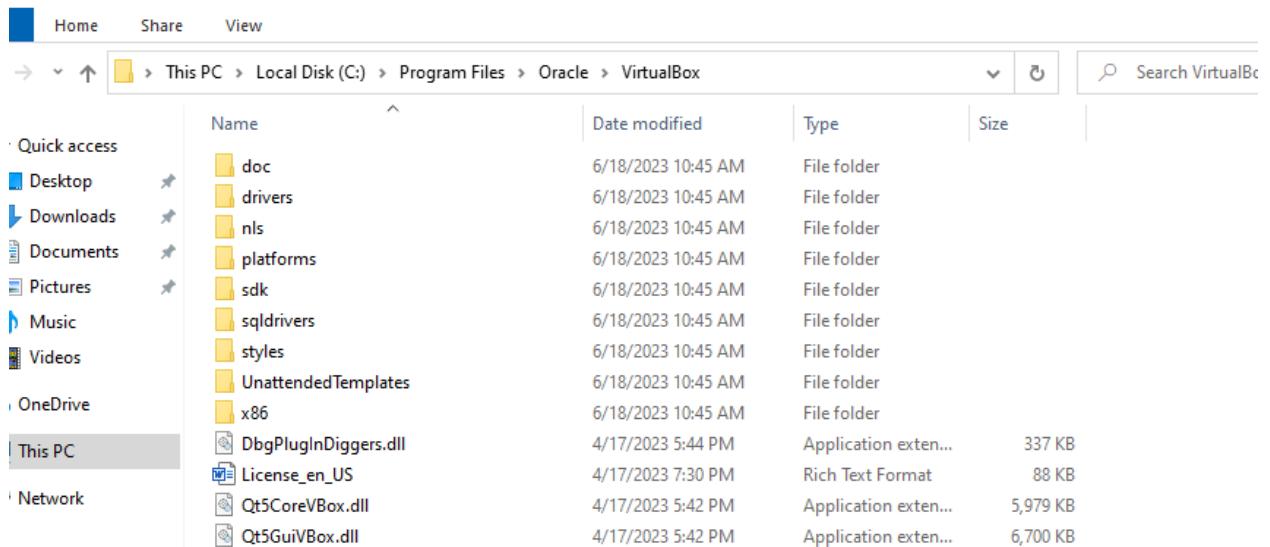


```
habibun@hemel-22241042:~$ egrep -c '(vmx|svm)' /proc/cpuinfo
0
habibun@hemel-22241042:~$
```

Based on the provided output, you can conclude that virtualization is enabled as the result displayed is greater than 0. However, if virtualization is not enabled, ensure to activate the virtualization feature in your system's BIOS settings.

The extra steps :

1. Currently I am using Oracle VM box. Which comes with nested virtualization support. To enable this first we have to go to the installation folder of Oracle VM. which is in my case [C:\Program Files\Oracle\VirtualBox](#).



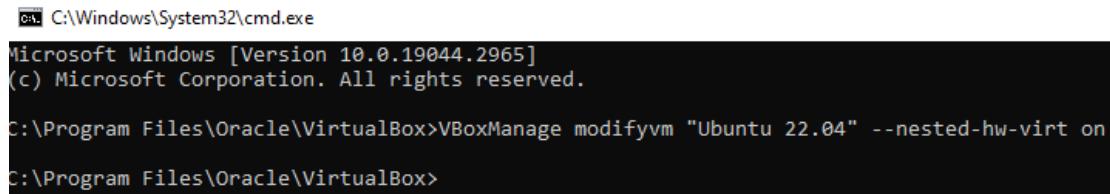
A screenshot of the Windows File Explorer interface. The left sidebar shows standard folder icons for Quick access, Desktop, Downloads, Documents, Pictures, Music, Videos, OneDrive, This PC (which is selected), and Network. The main pane displays a list of files and folders within the 'VirtualBox' directory of 'This PC'. The list includes: doc, drivers, nls, platforms, sdk, sqldrivers, styles, UnattendedTemplates, x86, DbgPlugInDiggers.dll, License_en_US, Qt5CoreVBox.dll, and Qt5GuiVBox.dll. The columns are Name, Date modified, Type, and Size.

	Name	Date modified	Type	Size
· Quick access				
Desktop	doc	6/18/2023 10:45 AM	File folder	
Downloads	drivers	6/18/2023 10:45 AM	File folder	
Documents	nls	6/18/2023 10:45 AM	File folder	
Pictures	platforms	6/18/2023 10:45 AM	File folder	
Music	sdk	6/18/2023 10:45 AM	File folder	
Videos	sqldrivers	6/18/2023 10:45 AM	File folder	
OneDrive	styles	6/18/2023 10:45 AM	File folder	
This PC	UnattendedTemplates	6/18/2023 10:45 AM	File folder	
Network	x86	6/18/2023 10:45 AM	File folder	
	DbgPlugInDiggers.dll	4/17/2023 5:44 PM	Application exten...	337 KB
	License_en_US	4/17/2023 7:30 PM	Rich Text Format	88 KB
	Qt5CoreVBox.dll	4/17/2023 5:42 PM	Application exten...	5,979 KB
	Qt5GuiVBox.dll	4/17/2023 5:42 PM	Application exten...	6,700 KB

2. Then open cmd and paste the command bellow (As my processor is intel so this command is only for Intel processors)

VBoxManage modifyvm "Ubuntu 22.04" --nested-hw-virt on

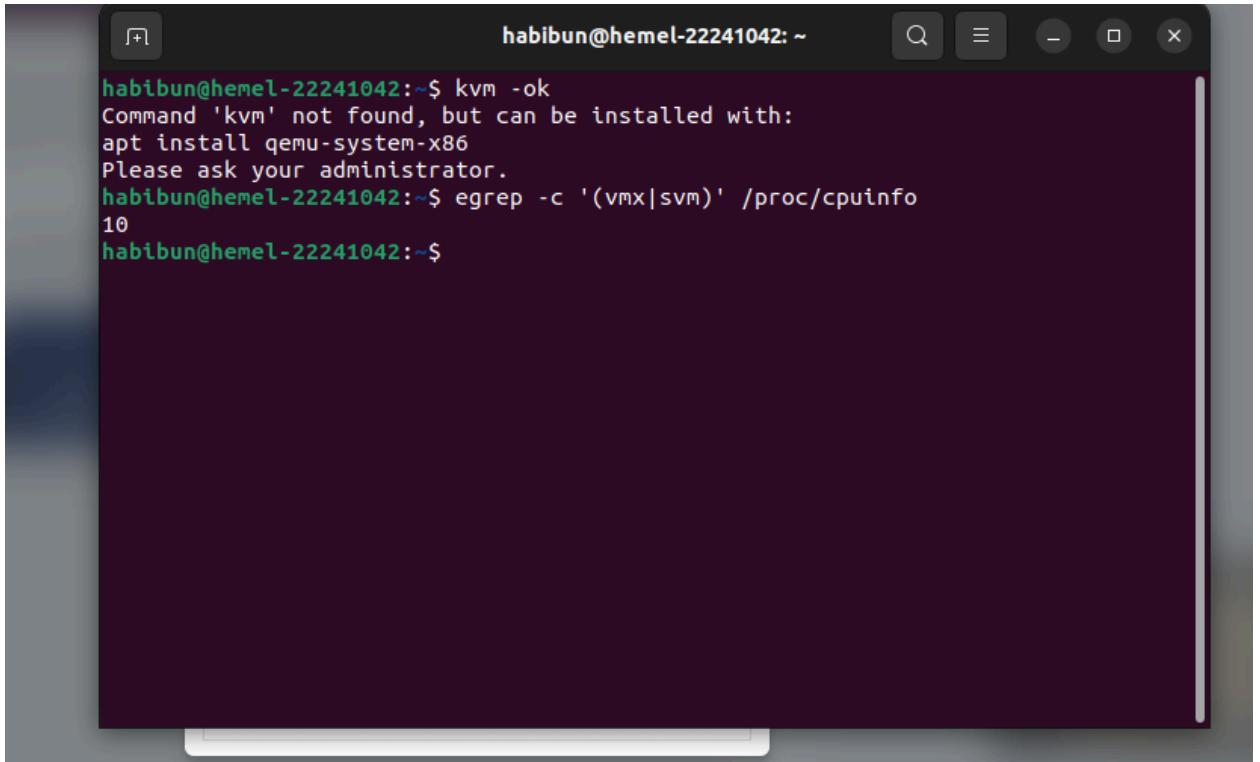
Here ubuntu 22.04 is my machine name, you have change it based on your own machine name.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\Oracle\VirtualBox>VBoxManage modifyvm "Ubuntu 22.04" --nested-hw-virt on
C:\Program Files\Oracle\VirtualBox>
```

3. After this we will check the virtualization support or not? Then use [kvm-ok](#) command to verify if KVM virtualization is enabled or not . Here it say kvm not found. That means you need to enable KVM first. So, need to give the command: [sudo apt install cpu-checker](#) .and give the command: [egrep -c '\(vmx|svm\)' /proc/cpuinfo](#).

A screenshot of a terminal window titled "habibun@hemel-22241042: ~". The window contains the following text:

```
habibun@hemel-22241042:~$ kvm -ok
Command 'kvm' not found, but can be installed with:
apt install qemu-system-x86
Please ask your administrator.
habibun@hemel-22241042:~$ egrep -c '(vmx|svm)' /proc/cpuinfo
10
habibun@hemel-22241042:~$
```

As it is showing a value which is greater than 0 so this virtual machine has the capability to run kvm. If the command returns a value of 0, your processor is not capable of running KVM. On the other hand, any other number means you can proceed with the installation.

In addition, you can verify if KVM virtualization is enabled by running the following command:

sudo kvm-ok

```
habibun@hemel-22241042:~$ sudo kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
habibun@hemel-22241042:~$ ^[[200~sudo adduser 'username' libvirt~
sudo: command not found
habibun@hemel-22241042:~$ sudo adduser 'username' libvirt
adduser: The user `username' does not exist.
habibun@hemel-22241042:~$ sudo adduser 'habibun' libvirt
adduser: The user `habibun' does not exist.
habibun@hemel-22241042:~$ sudo adduser $USER libvirt
Adding user `habibun' to group `libvirt' ...
Adding user habibun to group libvirt
Done.
habibun@hemel-22241042:~$
```

Give a User :

```
habibun@hemel-22241042:~$ sudo adduser $USER kvm
Adding user `habibun' to group `kvm' ...
Adding user habibun to group kvm
Done.
habibun@hemel-22241042:~$
```

Verify the Installation:

```
habibun@hemel-22241042:~$ virsh list --all
error: failed to connect to the hypervisor
error: Failed to connect socket to '/var/run/libvirt/libvirt-sock': Permission d
enied

habibun@hemel-22241042:~$ sudo virsh list --all
 Id  Name   State
-----
habibun@hemel-22241042:~$
```

Installing kvm

Command: sudo apt install -y qemu-kvm virt-manager libvirt-daemon-system virtinst libvirt-clients bridge-utils

```
habibun@hemel-22241042:~$ sudo apt install -y qemu-kvm virt-manager libvirt-daemon-system virtinst libvirt-clients bridge-utils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'qemu-system-x86' instead of 'qemu-kvm'
bridge-utils is already the newest version (1.7-1ubuntu3).
virt-manager is already the newest version (1:4.0.0-1).
virtinst is already the newest version (1:4.0.0-1).
libvirt-clients is already the newest version (8.0.0-1ubuntu7.8).
libvirt-daemon-system is already the newest version (8.0.0-1ubuntu7.8).
qemu-system-x86 is already the newest version (1:6.2+dfsg-2ubuntu6.16).
0 to upgrade, 0 to newly install, 0 to remove and 21 not to upgrade.
habibun@hemel-22241042:~$
```

With all the packages installed, enable and start the Libvirt daemon.

After that we need to enable and start Libvirt Daemon .

Two command are :

- **sudo systemctl enable --now libvirtd**
- **sudo systemctl start libvirtd**

```
habibun@hemel-22241042:~$ sudo systemctl enable --now libvirtd
habibun@hemel-22241042:~$
```

Then

Check demon is active or not. Confirm that the virtualization daemon is running as shown. Command :

- **sudo systemctl status libvirtd**

```
habibun@hemel-22241042:~$ sudo systemctl status libvirtd
● libvirtd.service - Virtualization daemon
   Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; vendor pres>
   Active: active (running) since Sat 2024-02-10 12:38:35 +06; 33min ago
     TriggeredBy: ● libvirtd-admin.socket
                   ● libvirtd-ro.socket
                   ● libvirtd.socket
   Docs: man:libvirtd(8)
         https://libvirt.org
 Main PID: 5324 (libvirtd)
   Tasks: 21 (limit: 32768)
  Memory: 9.9M
    CPU: 316ms
   CGroup: /system.slice/libvirtd.service
           ├─5324 /usr/sbin/libvirtd
           ├─5452 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/defa>
           └─5453 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/defa>
```

Then ran the **sudo apt install virt-manager** command to install virtual manager

```
habibun@hemel-22241042:~$ sudo apt install virt-manager
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
virt-manager is already the newest version (1:4.0.0-1).
0 to upgrade, 0 to newly install, 0 to remove and 21 not to upgrade.
habibun@hemel-22241042:~$
```

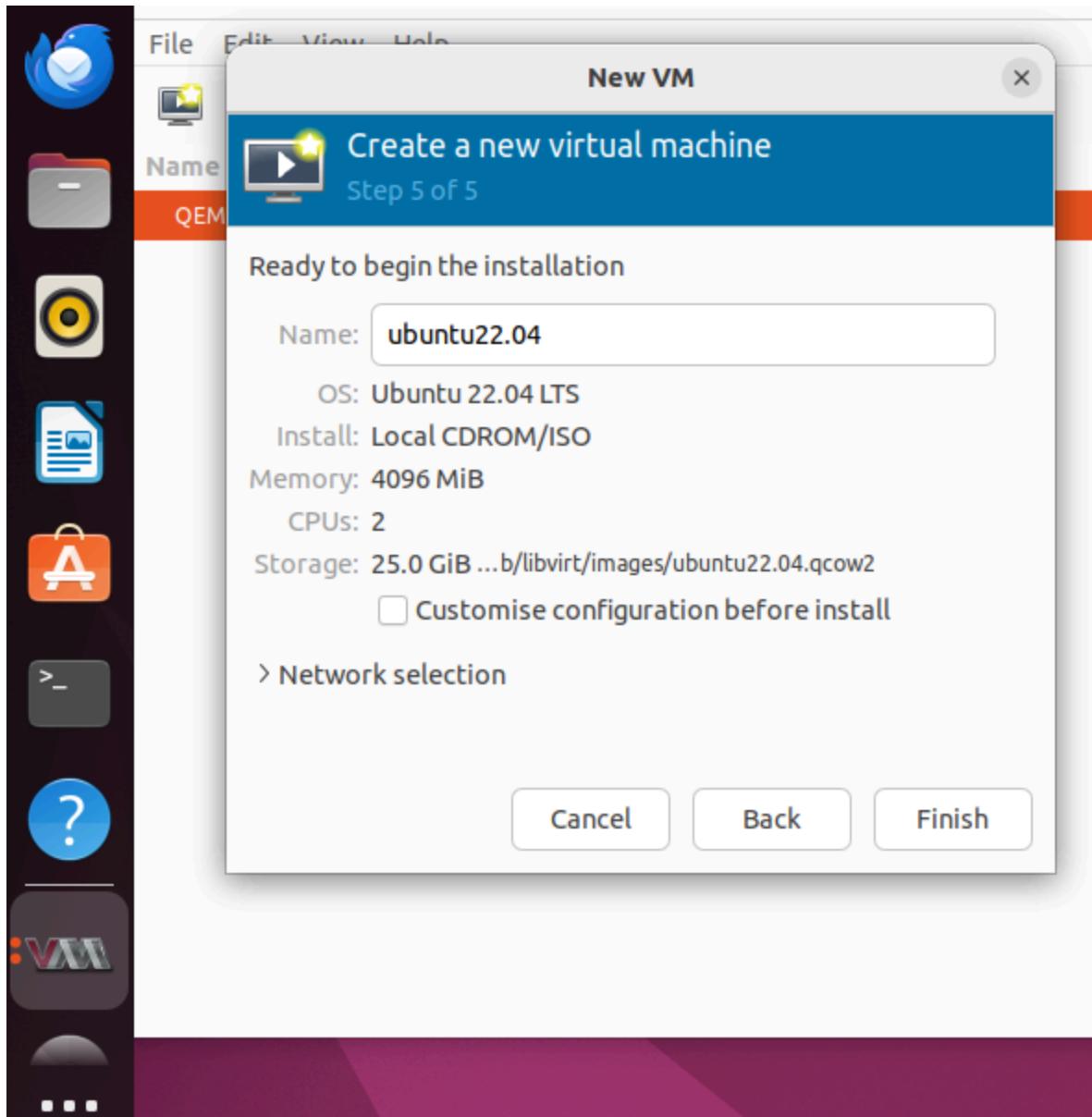
More clean view:

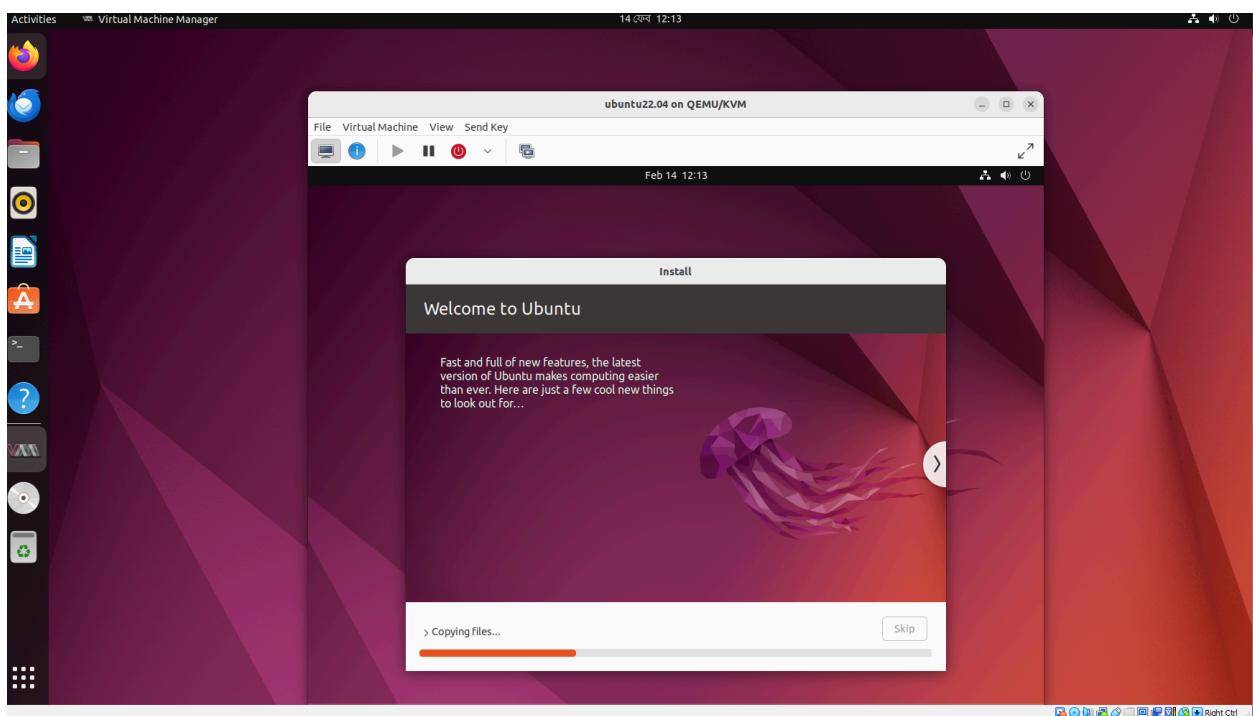
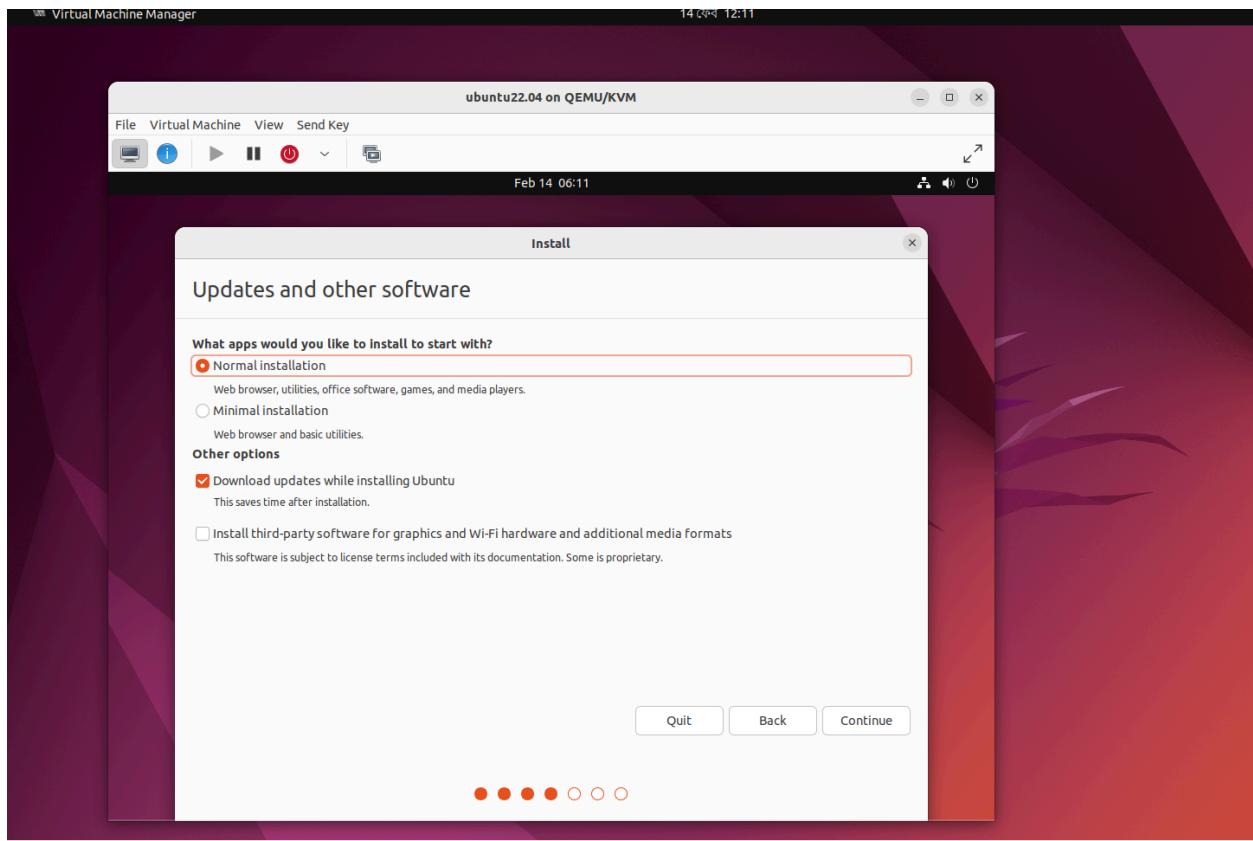
```
habibun@hemel-22241042:~$ sudo systemctl enable --now libvirtd
habibun@hemel-22241042:~$ sudo systemctl start libvirtd
habibun@hemel-22241042:~$ sudo systemctl status libvirtd
● libvirtd.service - Virtualization daemon
   Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; vendor pres>
   Active: active (running) since Wed 2024-02-14 10:56:48 +06; 27min ago
     TriggeredBy: ● libvirtd-admin.socket
                   ● libvirtd-ro.socket
                   ● libvirtd.socket
   Docs: man:libvirtd(8)
         https://libvirt.org
 Main PID: 725 (libvirtd)
    Tasks: 21 (limit: 32768)
   Memory: 20.1M
      CPU: 392ms
     CGroup: /system.slice/libvirtd.service
             └─725 /usr/sbin/libvirtd
                 ├─995 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/defau>
                 ├─996 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/defau>

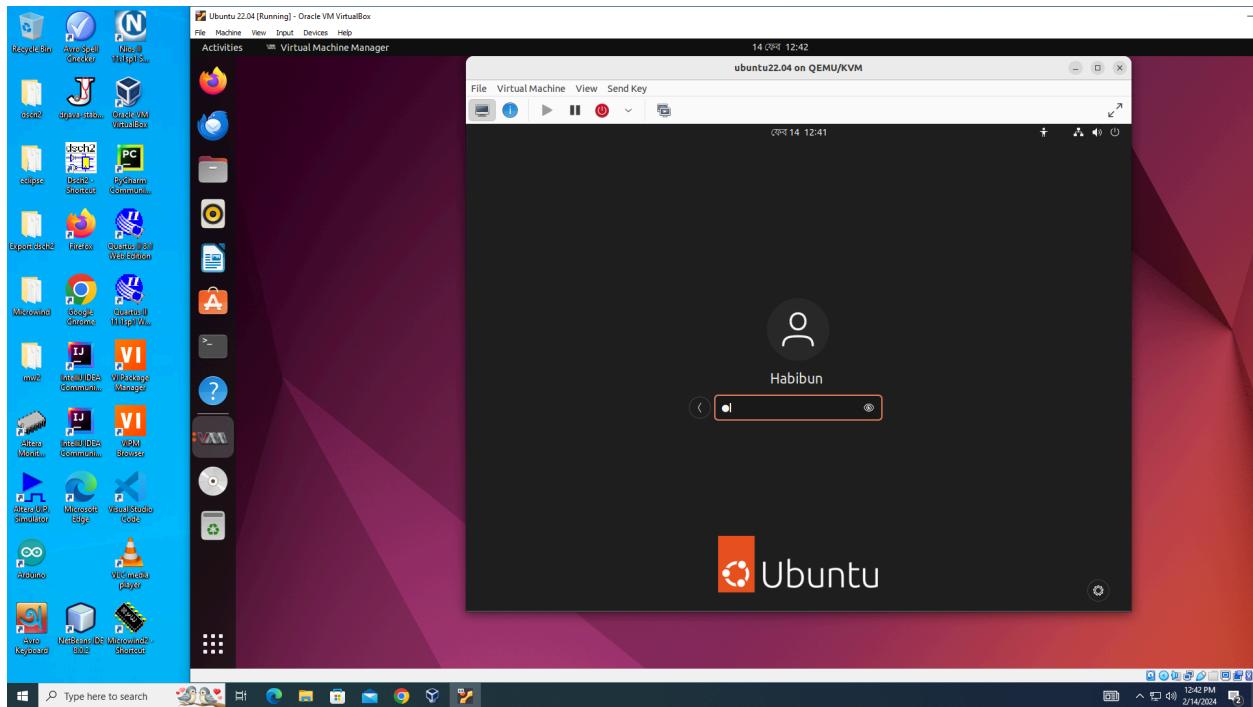
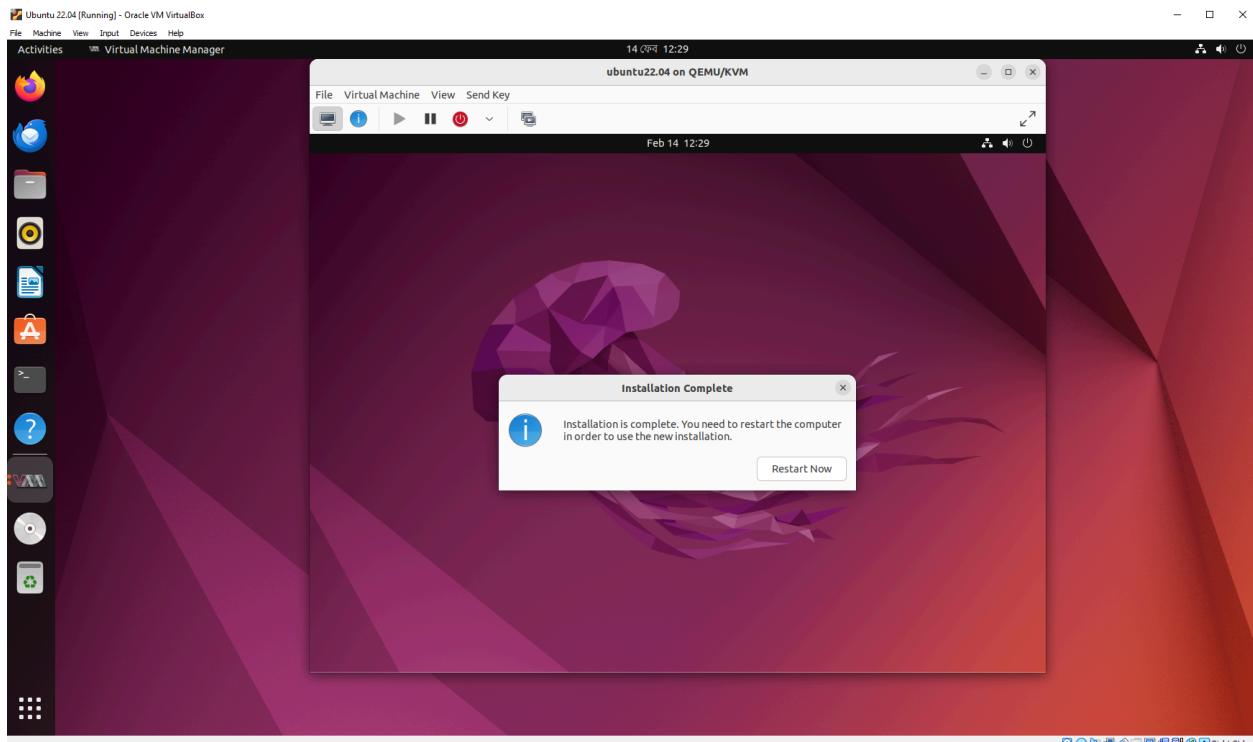
Feb 14 10:56:48 hemel-22241042 systemd[1]: Started Virtualization daemon.
Feb 14 10:56:49 hemel-22241042 dnsmasq[995]: started, version 2.86 cachesize 150
Feb 14 10:56:49 hemel-22241042 dnsmasq[995]: compile time options: IPv6 GNU-getopt
Feb 14 10:56:49 hemel-22241042 dnsmasq-dhcp[995]: DHCP, IP range 192.168.122.2 >
```

TASK2. Create a VM using VMM (virtual machine manager) i.e. using GUI.

To create a virtual machine we have to choose an ISO file for me it's ubuntu 22.04. Then i have followed some easy steps in GUI







TASK3. Create a kvm-based VM using “virt-install” cli.

Step1: To set up a KVM-based virtual machine, you need to construct a command with various parameters tailored to your specific requirements. Here is a sample command that you can adapt to suit your needs.

```
habibun@hemel-22241042:~$ sudo virt-install \
--name commandLineVM \
--memory 2048 \
--vcpus 2 \
--disk size 4 \
--os-variant ubuntu-22.04.3 \
--cdrom /home/habibun/iso file for nested vm/ubuntu-22.04.3-desktop-amd64.iso\
```

Command: `sudo virt-install \
--name commandLineVM \
--memory 2048 \
--vcpus 2 \
--disk size 4\`

```
--os-variant ubuntu-22.04.3\  
--cdrom /home/habibun/iso file for nested  
vm/ubuntu-22.04.3-desktop-amd64.iso\
```

Initially, I designated the name of my new virtual machine as "commandLineVM" utilizing the --name parameter. Following that, I allocated 2048 MB of memory using --memory 2048. Subsequently, I assigned 2 cores to the CPU with --vcpus 2. Lastly, I specified the path to the image file using --cdrom /home/habibun/iso file for nested vm/ubuntu-22.04.3-desktop-amd64.iso lastly, I have selected the size of the disk using --disk size 4 Lastly, I have entered the name of the OS using --os-variant ubuntu-22.04.3.

Step 2) Install KVM, Qemu, virt-manager & libvirtd daemon

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

```
habibun@hemel-22241042:~$ sudo apt install -y qemu qemu-kvm libvirt-daemon libvi  
rt-clients bridge-utils virt-manager  
[sudo] password for habibun:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Note, selecting 'qemu-system-x86' instead of 'qemu-kvm'  
bridge-utils is already the newest version (1.7-1ubuntu3).  
virt-manager is already the newest version (1:4.0.0-1).  
libvirt-clients is already the newest version (8.0.0-1ubuntu7.8).  
libvirt-daemon is already the newest version (8.0.0-1ubuntu7.8).  
libvirt daemon set to manually installed
```

Once installed, we must confirm if the required kvm modules are loaded. Therefore, run the command: **lsmod | grep kvm**

```
habibun@hemel-22241042:~$ lsmod | grep kvm  
kvm_intel          487424  0  
kvm                1409024  1 kvm_intel  
irqbypass         12288   1 kvm  
habibun@hemel-22241042:~$ █
```

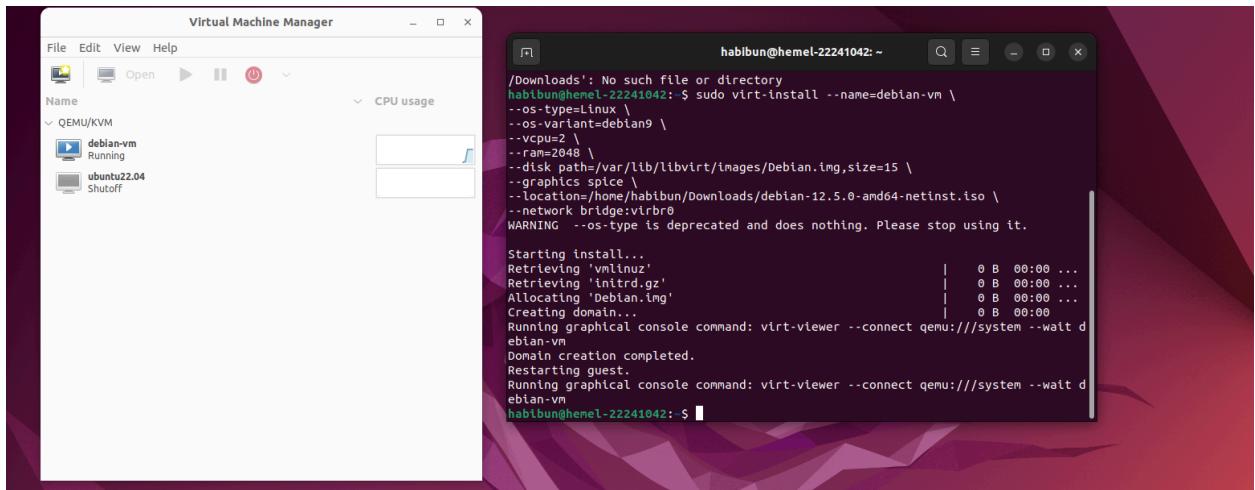
Additionally, confirm that the libvирtd daemon is running as follows.

Command: **sudo systemctl status libvирtd.service**

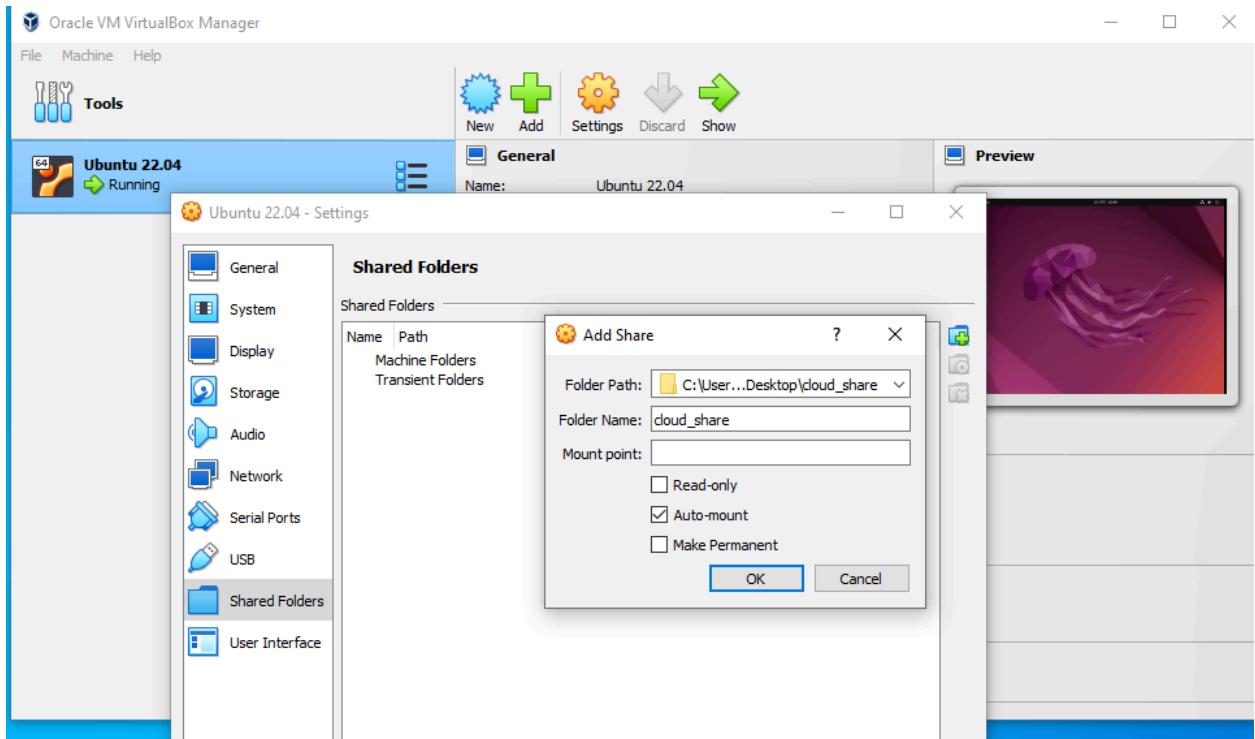
```
habibun@hemel-22241042:~$ sudo systemctl status libvирtd.service
● libvирtd.service - Virtualization daemon
   Loaded: loaded (/lib/systemd/system/libvирtd.service; enabled; vendor pres>
   Active: active (running) since Thu 2024-02-15 11:36:50 +06; 7min ago
 TriggeredBy: ● libvирtd-admin.socket
               ● libvирtd-ro.socket
               ● libvирtd.socket
   Docs: man:libvирtd(8)
         https://libvirt.org
 Main PID: 740 (libvирtd)
   Tasks: 21 (limit: 32768)
  Memory: 43.8M
    CPU: 577ms
   CGroup: /system.slice/libvирtd.service
           └─ 740 /usr/sbin/libvирtd
                 ├─1085 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/defa>
                 ├─1086 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/defa>
```

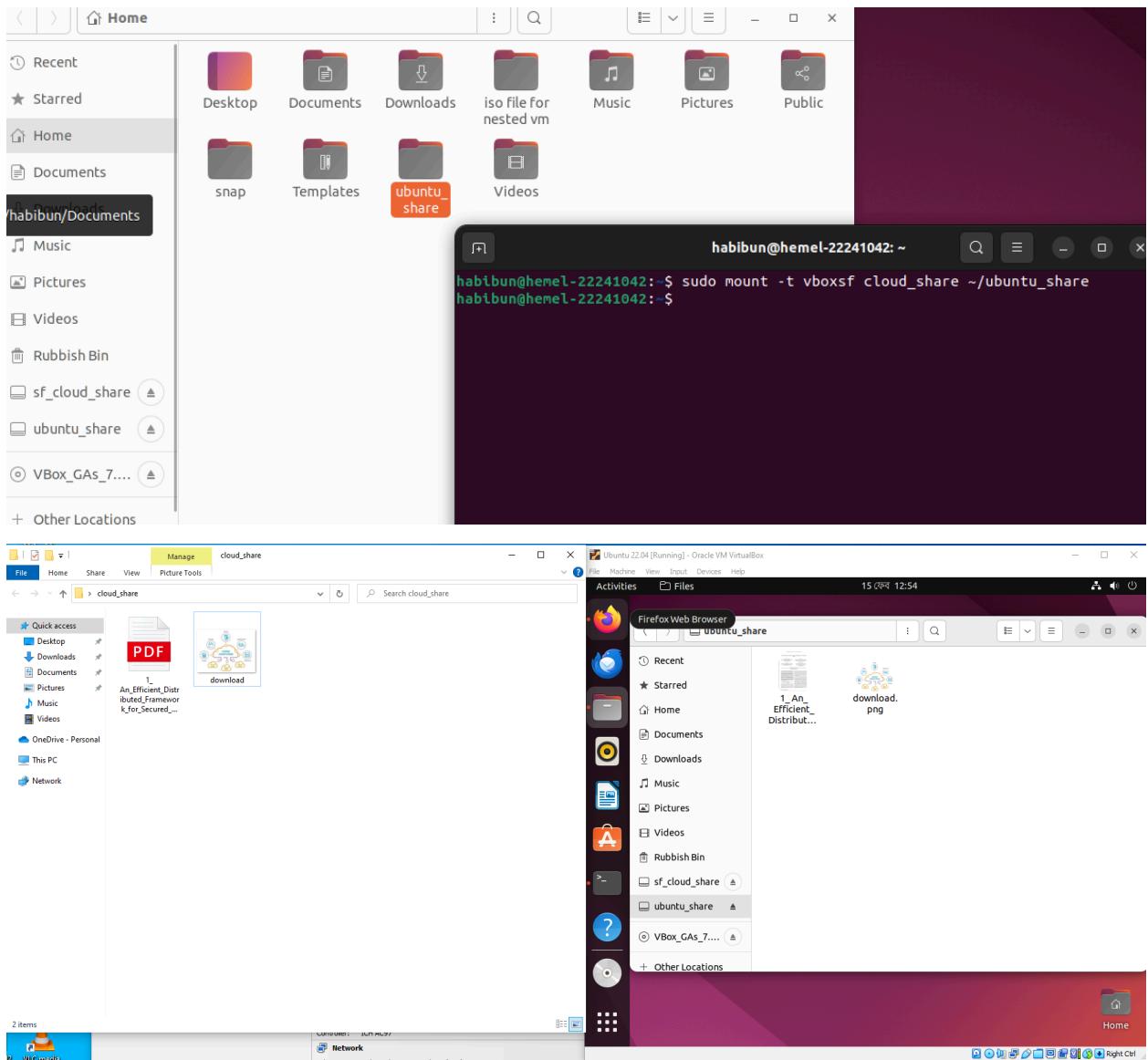
```
habibun@hemel-22241042:~$ sudo virt-install --name=debian-vm \
--os-type=Linux \
--os-variant=debian9 \
--vcpu=2 \
--ram=2048 \
--disk path=/var/lib/libvirt/images/Debian.img,size=15 \
--graphics spice \
--location=/home/habibun/Downloads/debian-12.5.0-amd64-netinst.iso \
--network bridge:virbr0
WARNING  --os-type is deprecated and does nothing. Please stop using it.

Starting install...
Retrieving 'vmlinuz'                                |  0 B  00:00 ...
Retrieving 'initrd.gz'                             |  0 B  00:00 ...
Allocating 'Debian.img'                            |  0 B  00:00 ...
Creating domain...                                 |  0 B  00:00
Running graphical console command: virt-viewer --connect qemu:///system --wait debian-vm
```

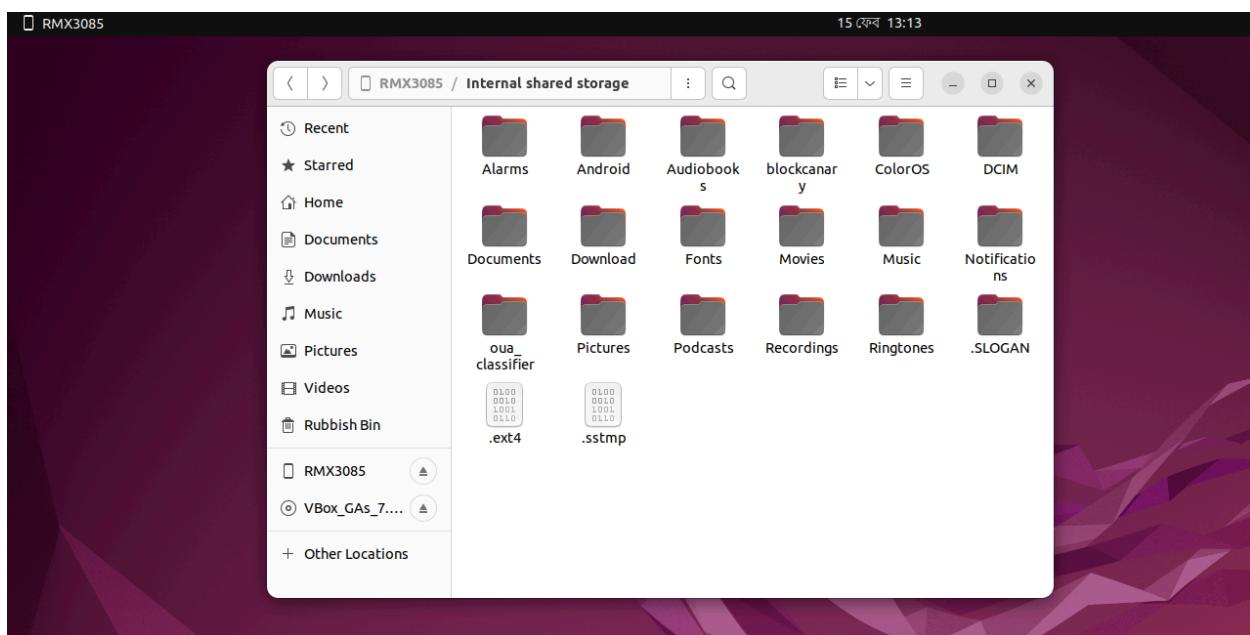
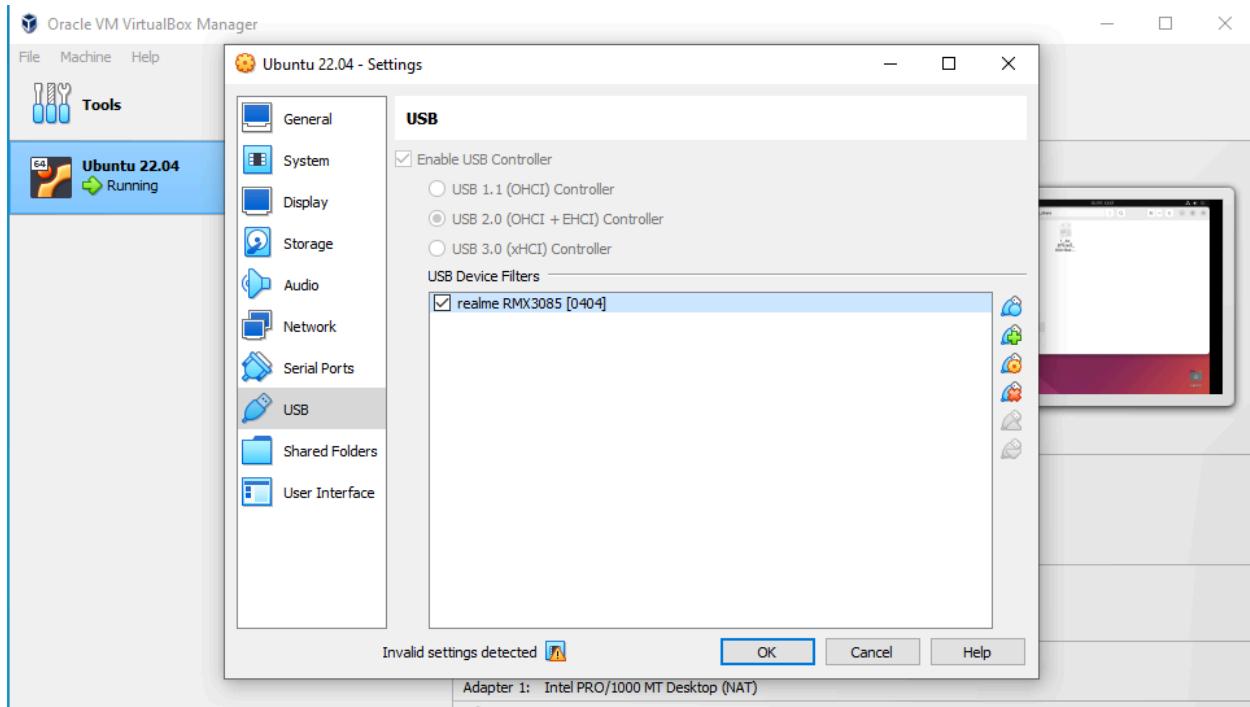


TASK 5. Make a shared folder between host os and guest os using cli. If any changes happen in Guest OS then the same change should be reflected in the Host-OS. This must be applicable for vice versa.



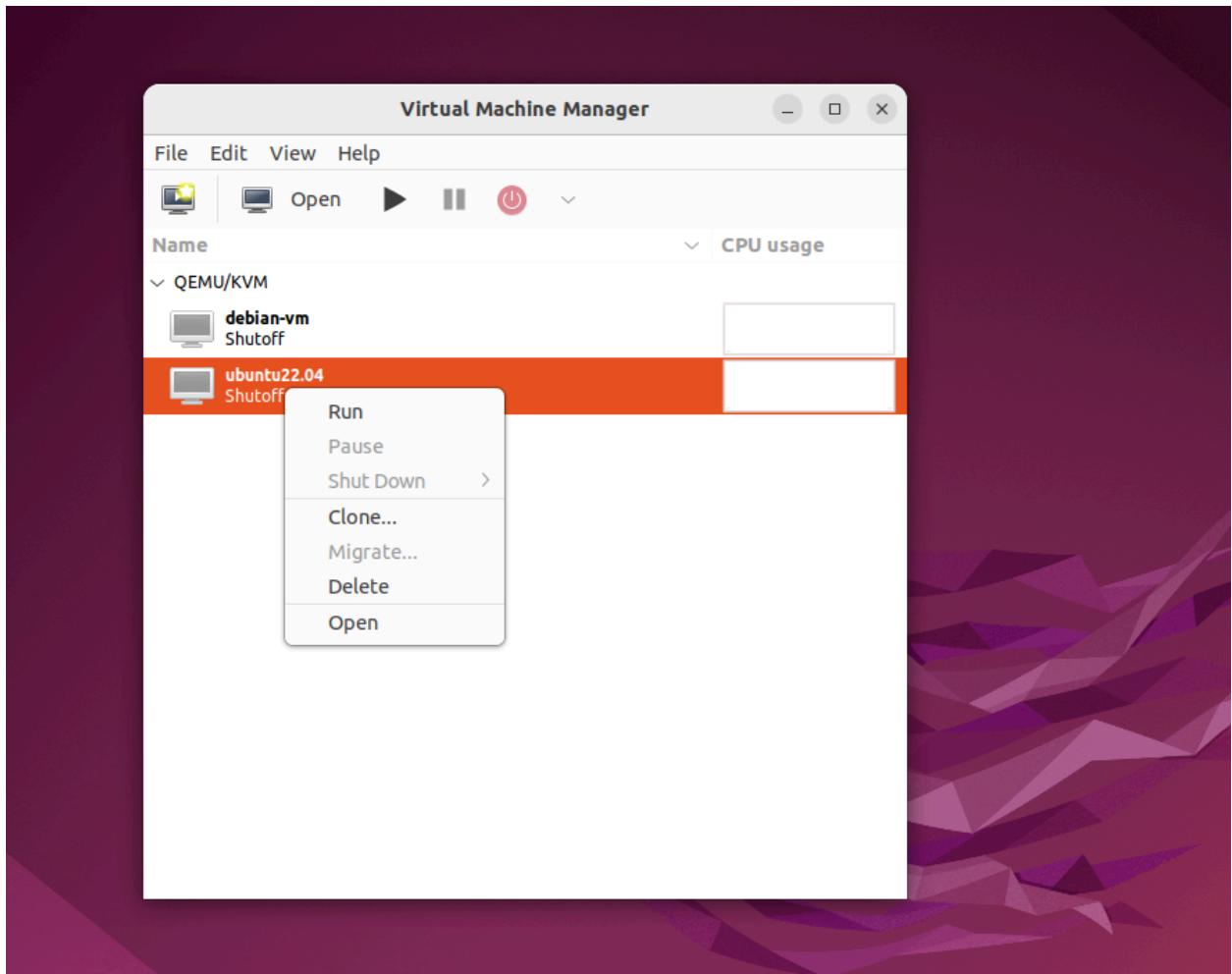


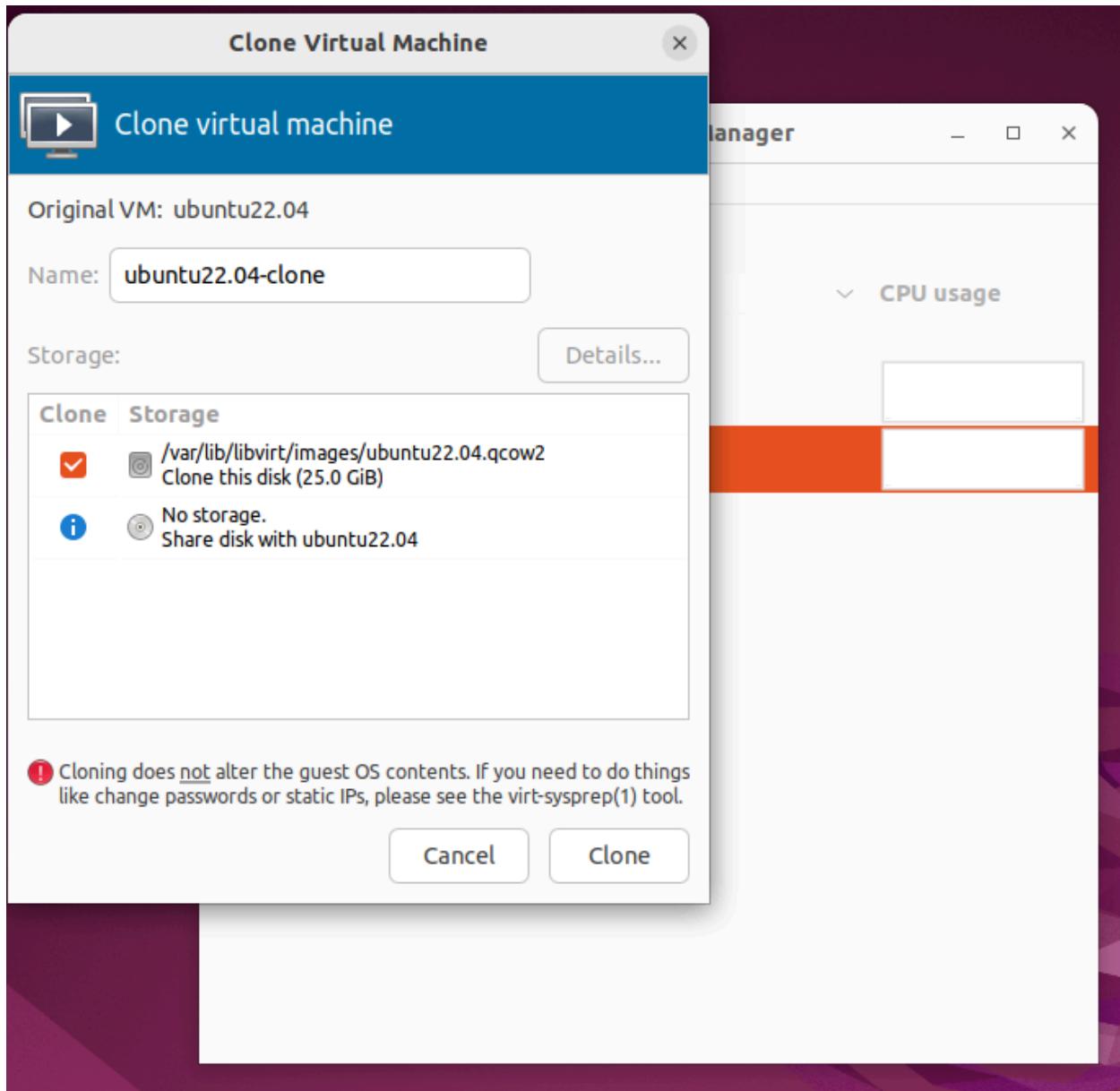
TASK6. Connect the guest OS with a phone and access the phone's files.

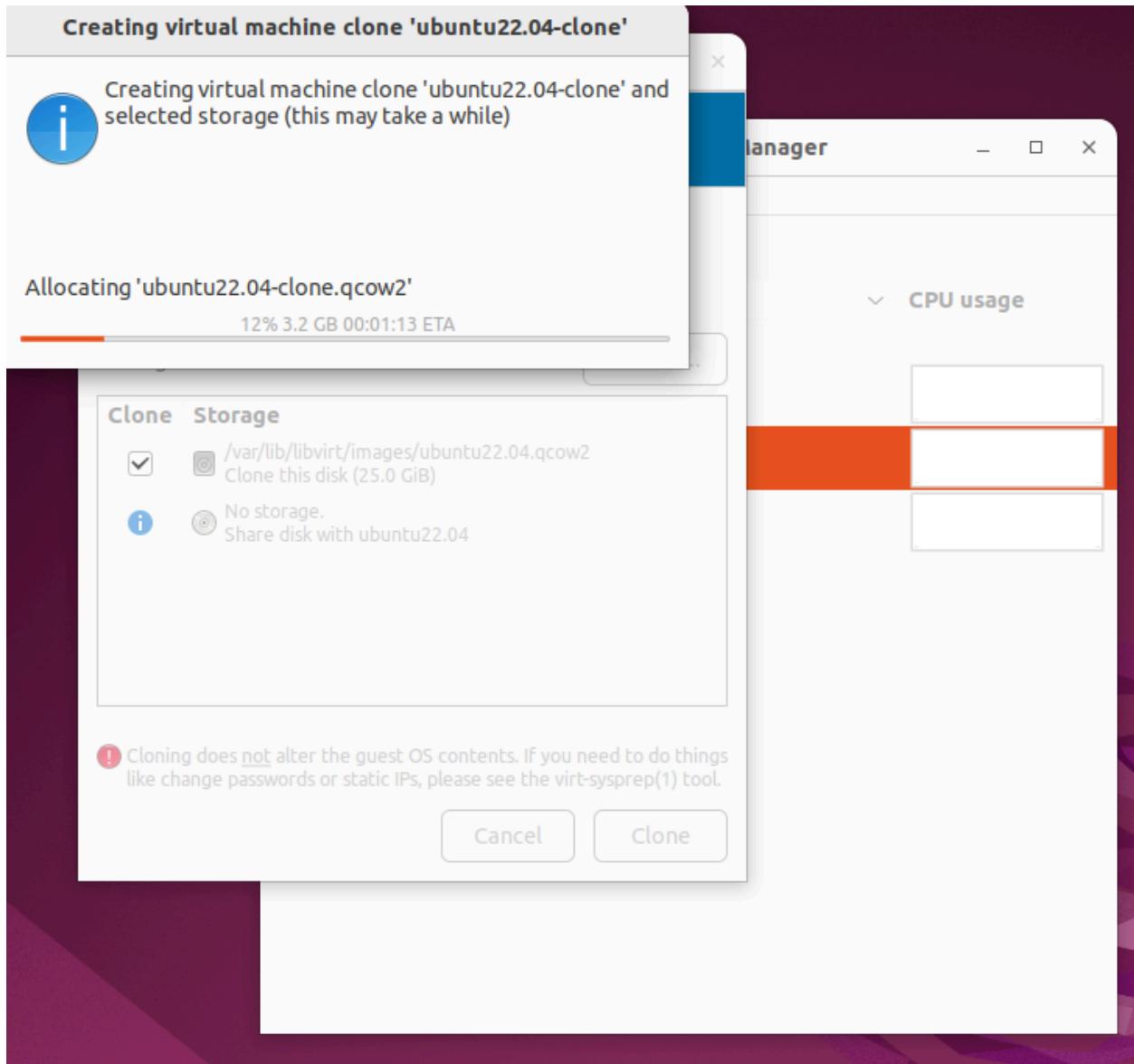


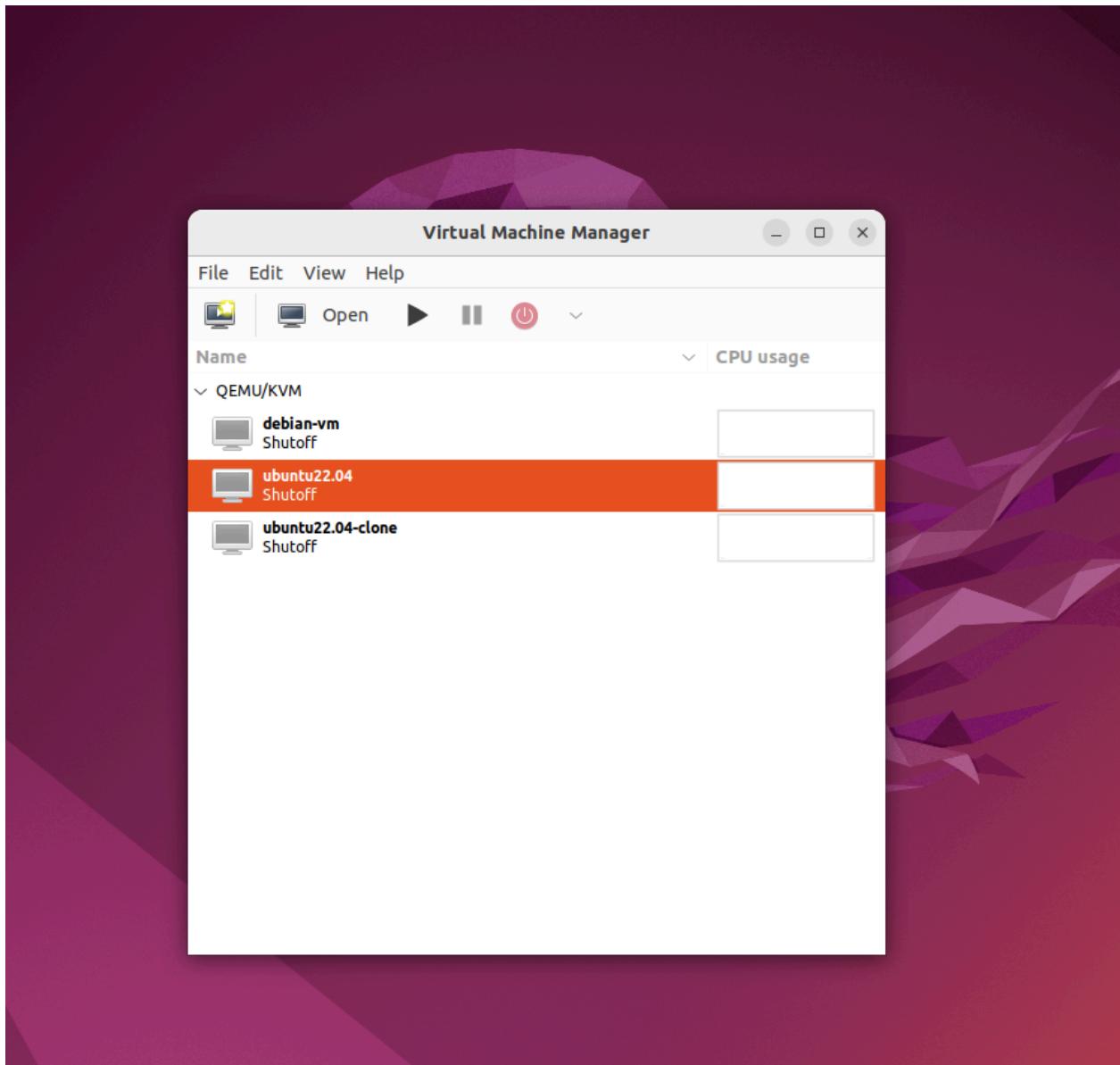
TASK 7. cloning a VM using GUI(PART1)

Using GUI cloning is very easy ,all the following steps are given below which is not so hard that some one needs any description to do that .









Command: [Virsh list –all](#)(this command will help to know many vm is created)

```
habibun@hemel-22241042:~$ virsh list --all
  Id  Name           State
  --  -- 
 -  debian-vm       shut off
 -  ubuntu22.04     shut off
 -  ubuntu22.04-clone  shut off

habibun@hemel-22241042:~$
```

TASK 7.cloning a VM using command (PART2)

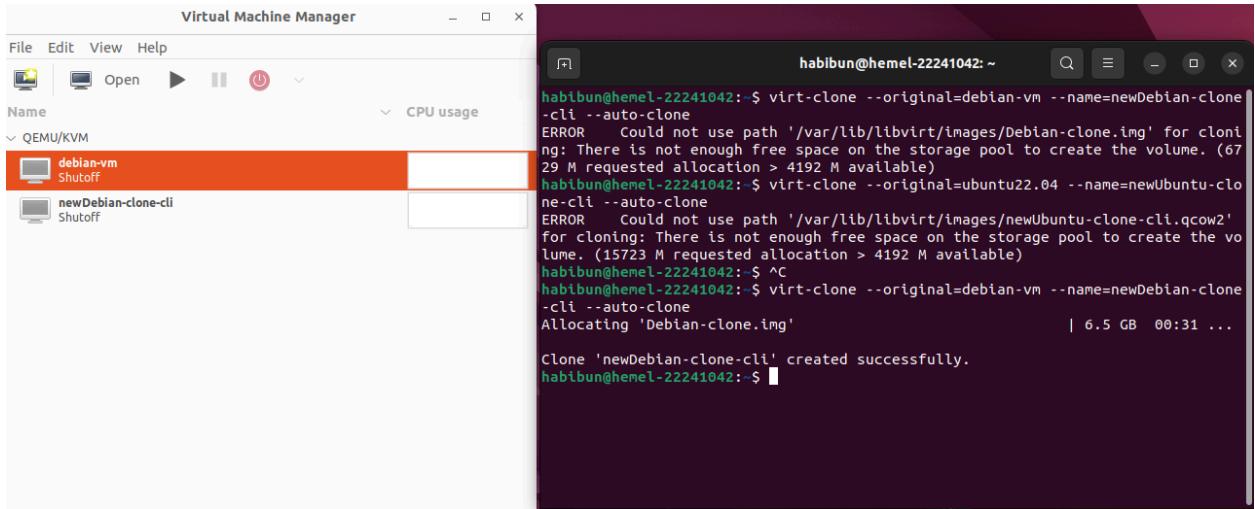
Following command will be:

```
virt-clone --original=debian-vm --name=newDebian-clone-cli --auto-clone
```

There debina-vm is the main vm name and newDebian-clone-cli is the cloned one.

```
habibun@hemel-22241042:~$ virt-clone --original=debian-vm --name=newDebian-clone
-cli --auto-clone
ERROR    Could not use path '/var/lib/libvirt/images/Debian-clone.img' for cloning: There is not enough free space on the storage pool to create the volume. (67
29 M requested allocation > 4192 M available)
habibun@hemel-22241042:~$ virt-clone --original=ubuntu22.04 --name=newUbuntu-clone-cli --auto-clone
ERROR    Could not use path '/var/lib/libvirt/images/newUbuntu-clone-cli.qcow2' for cloning: There is not enough free space on the storage pool to create the volume. (15723 M requested allocation > 4192 M available)
habibun@hemel-22241042:~$ █
```

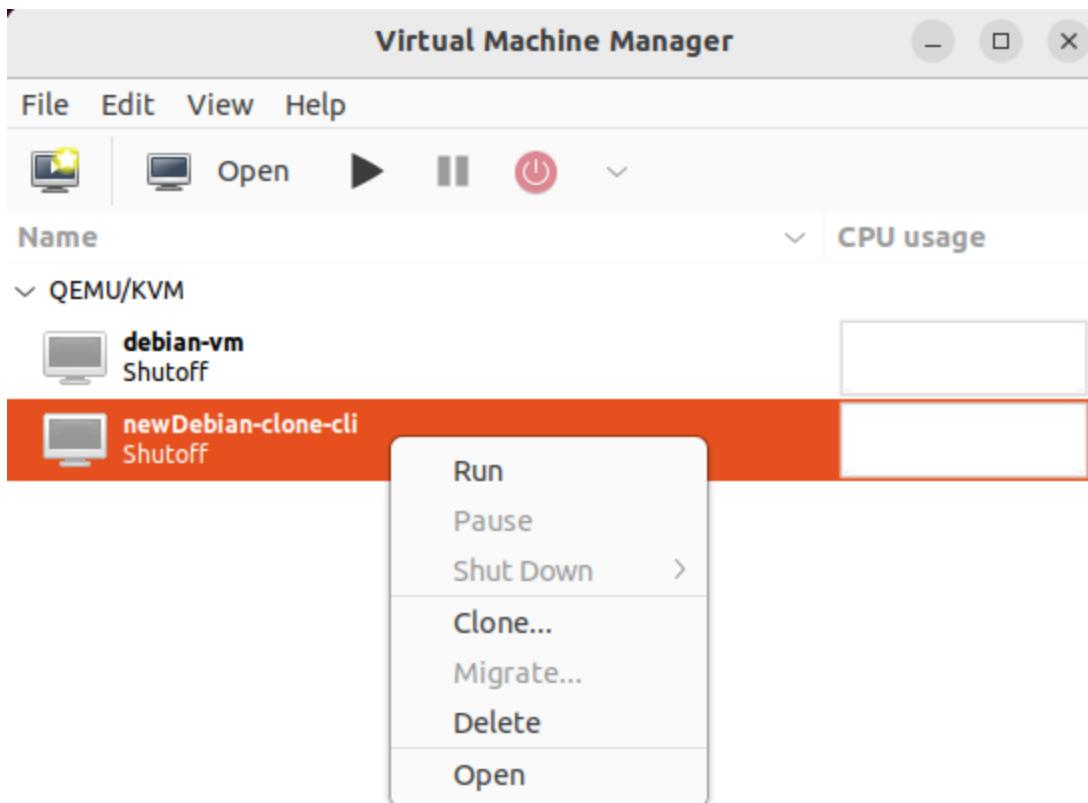
```
habibun@hemel-22241042:~$ virt-clone --original=debian-vm --name=newDebian-clone
-cli --auto-clone
Allocating 'Debian-clone.i  5% [-] 68 MB/s | 836 MB 03:34 ETA
```

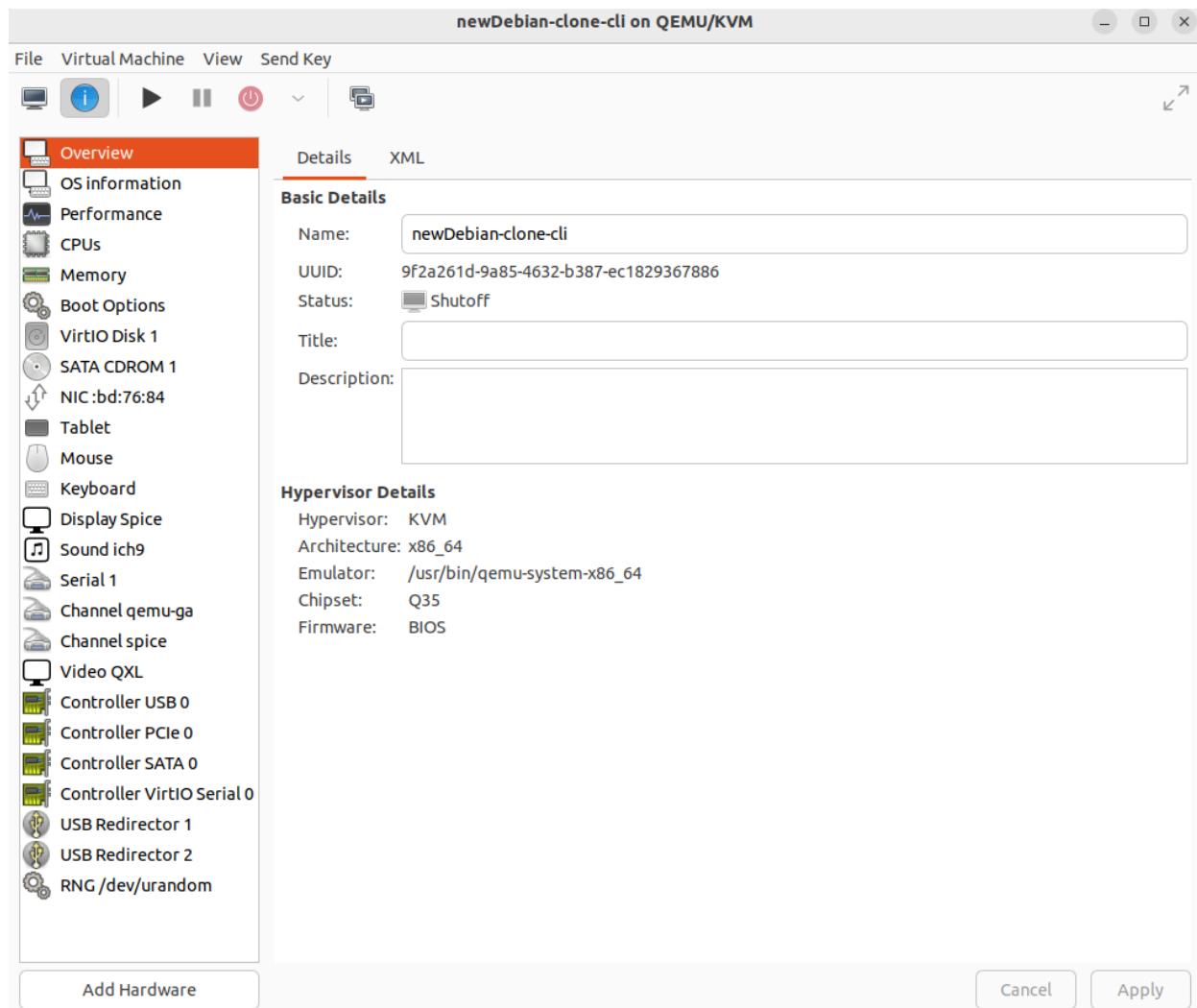


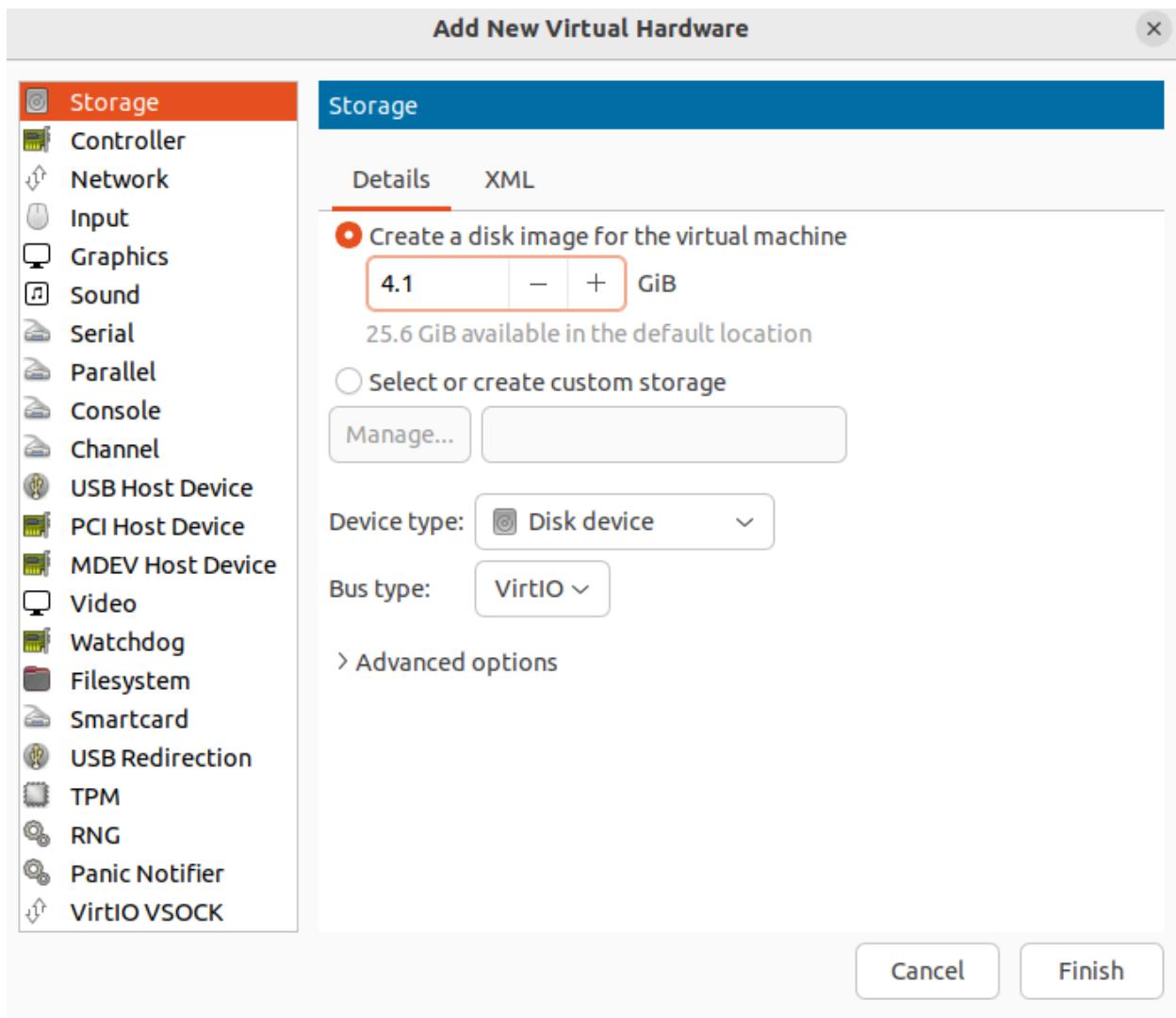
```
habibun@hemel-22241042:~$ virsh list --all
 Id  Name          State
 -+  debian-vm    shut off
 -+  newDebian-clone-cli  shut off

habibun@hemel-22241042:~$
```

TASK 8. Add two hard disks in a new cloned virtual machine using GUI.







newDebian-clone-cli on QEMU/KVM

File Virtual Machine View Send Key

Overview OS information Performance CPUs Memory Boot Options VirtIO Disk 1 VirtIO Disk 2 SATA CDROM 1 NIC :bd:76:84 Tablet Mouse Keyboard Display Spice Sound ich9 Serial 1 Channel qemu-ga Channel spice

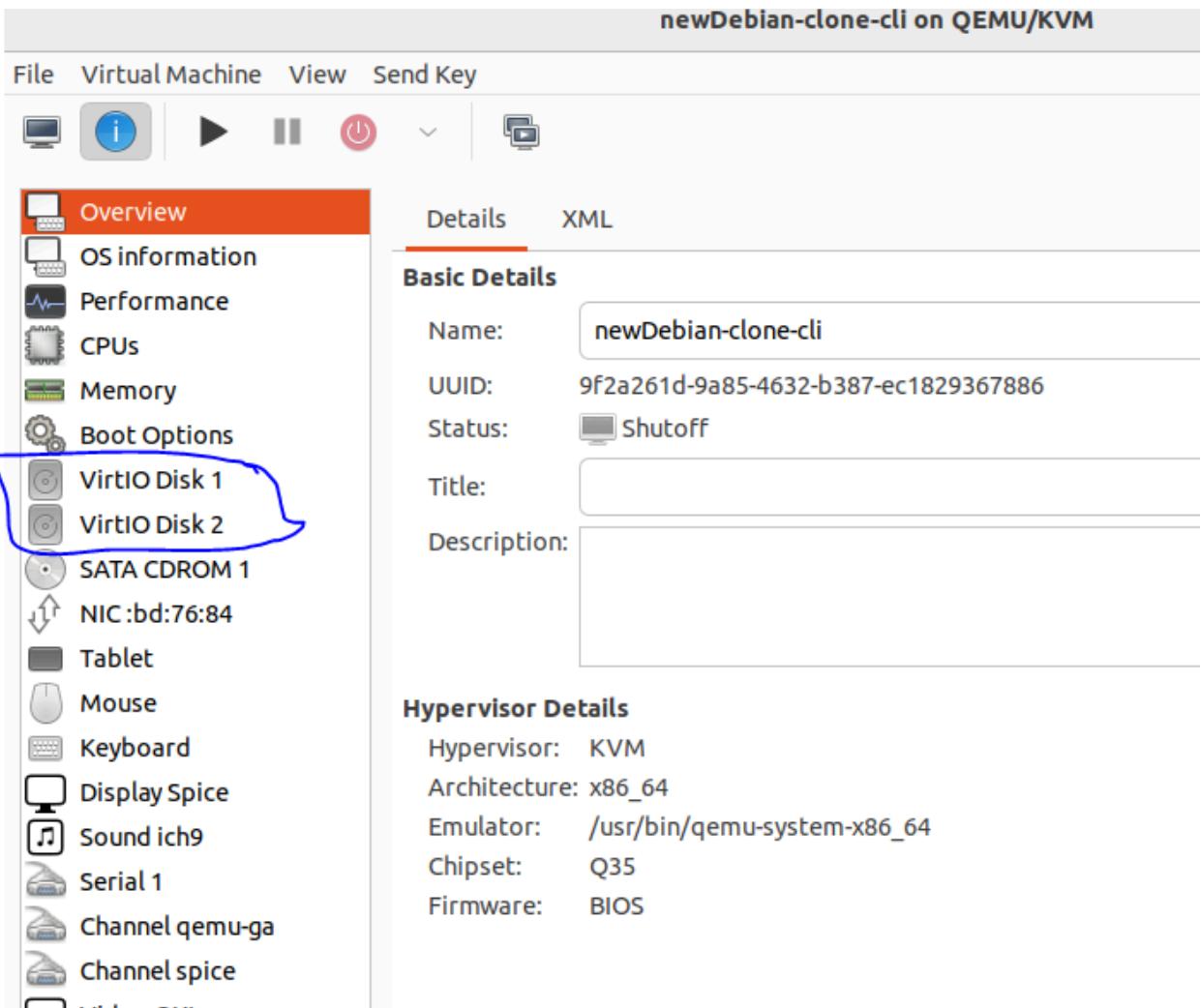
Details XML

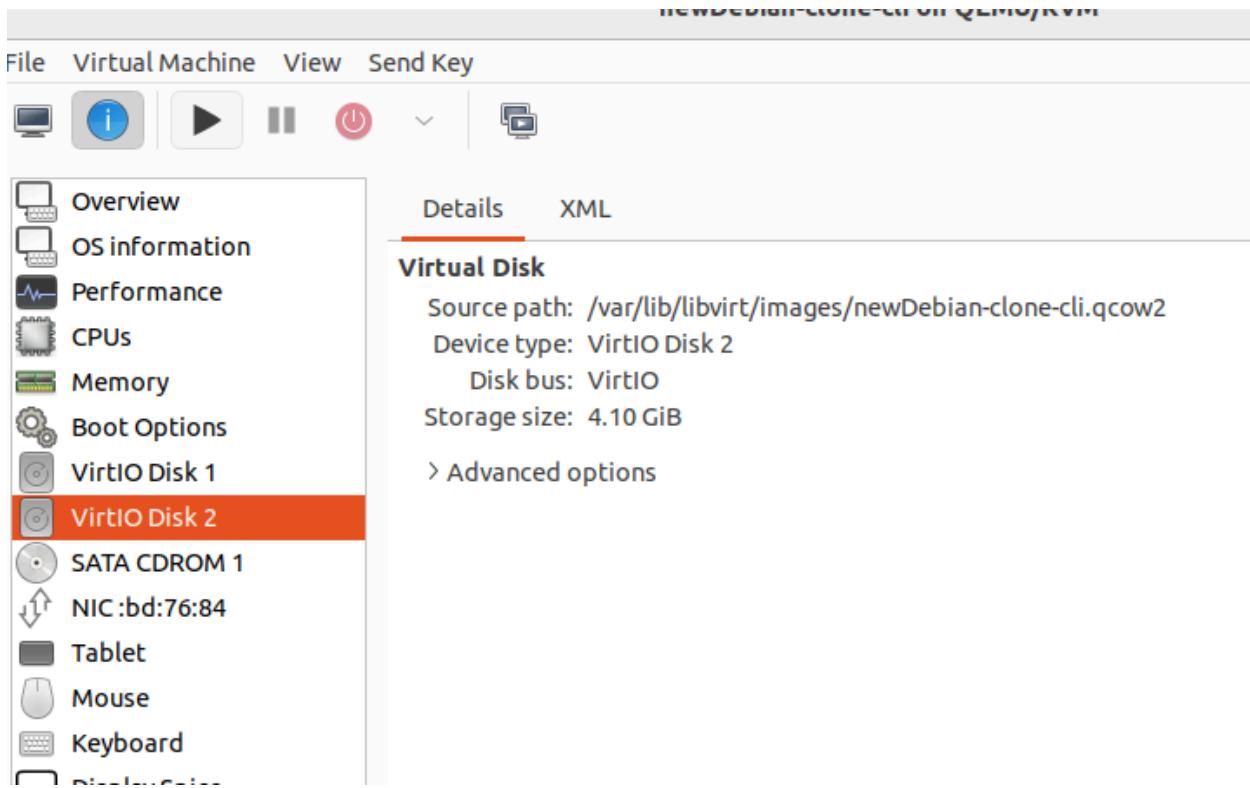
Basic Details

Name: newDebian-clone-cli
UUID: 9f2a261d-9a85-4632-b387-ec1829367886
Status: Shutoff
Title:
Description:

Hypervisor Details

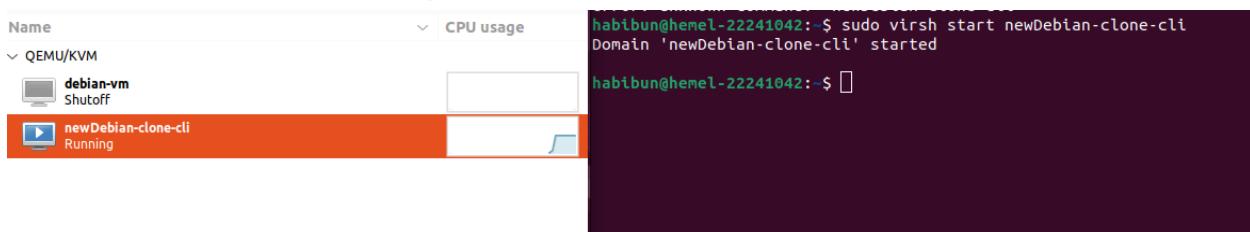
Hypervisor: KVM
Architecture: x86_64
Emulator: /usr/bin/qemu-system-x86_64
Chipset: Q35
Firmware: BIOS





TASK9. Add two hard disks in a new cloned virtual machine using a kvm-based command.

To add a extra harddisk to a cloned virtual machine at first we need to start the Virtual machine with the help of : **Sudo virsh start newDebian-clone-cli**



Now I have to create a image file. For that the command is need to use:

```
sudo qemu-img create -f qcow2 /var/lib/libvirt/images/VirtIODisk2.img 2G
```

where I have select storage **type qcow2** & allocat **2GB**. Also, I give **name** and selected the **location**.

```
habibun@hemel-22241042:~$ sudo qemu-img create -f qcow2 /var/lib/libvirt/images/  
VirtIODisk2.img 2G  
Formatting '/var/lib/libvirt/images/VirtIODisk2.img', fmt=qcow2 cluster_size=655  
36 extended_l2=off compression_type=zlib size=2147483648 lazy_refcounts=off refc  
ount_bits=16  
habibun@hemel-22241042:~$
```

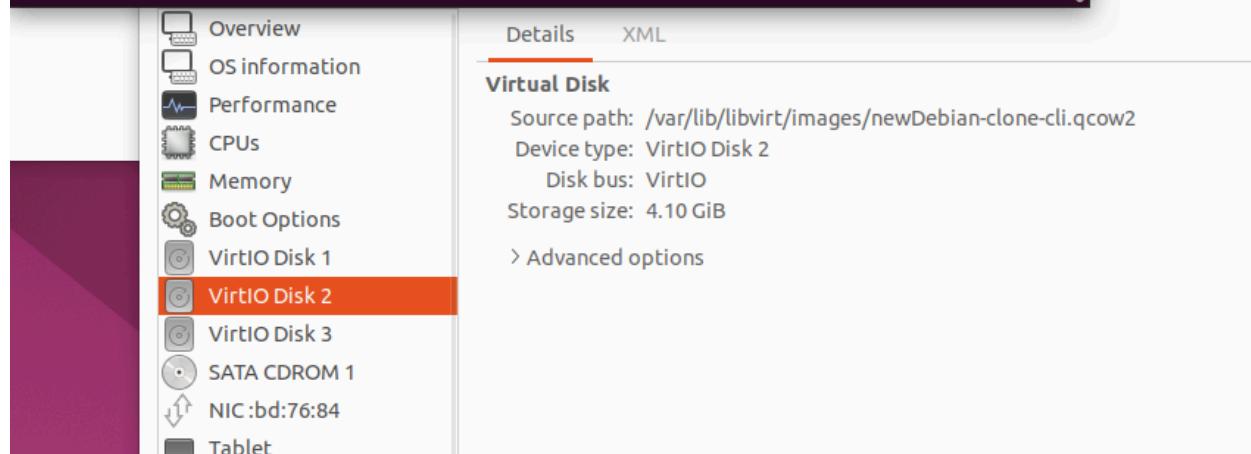
Then attach the img file with my virtual machine. To that we run **sudo virsh attach-disk newDebian-clone-cli /var/lib/libvirt/images/VirtIODisk2.img vdb --cache none** command.

By the same tecnic , we can add more hard disk. Form now one we just have to change the **name** and **vdc**. The commands are:

- **sudo qemu-img create -f qcow2 /var/lib/libvirt/images/VirtIODisk3.img 2G**
- **sudo virsh attach-disk newDebian-clone-cli /var/lib/libvirt/images/VirtIODisk3.img vdc --cache none**

```
habibun@hemel-22241042:~$ sudo qemu-img create -f qcow2 /var/lib/libvirt/images/VirtIODisk3.img 2G
Formatting '/var/lib/libvirt/images/VirtIODisk3.img', fmt=qcow2 cluster_size=655
36 extended_l2=off compression_type=zlib size=2147483648 lazy_refcounts=off refc
ount_bits=16
habibun@hemel-22241042:~$ sudo virsh attach-disk newDebian-clone-cli /var/lib/li
bvirt/images/VirtIODisk3.img vdb --cache none
error: Failed to attach disk
error: XML error: target 'vdb' duplicated for disk sources '/var/lib/libvirt/ima
ges/newDebian-clone-cli.qcow2' and '/var/lib/libvirt/images/VirtIODisk3.img'
habibun@hemel-22241042:~$ sudo virsh attach-disk newDebian-clone-cli /var/lib/li
bvirt/images/VirtIODisk3.img vdc --cache none
Disk attached successfully

habibun@hemel-22241042:~$
```



TASK10. How to migrate a VM to another host?
Show step-by-step commands and output about migrating any of your VMs. Migrate your VM to your friend's host machine and vice versa.

Firstly give the command to check how much virtual machine are there . command: [virsh list--all](#)

```
habibun@hemel-22241042:~$ virsh list --all
 Id  Name          State
 -----
 2   newDebian-clone-cli  running
 -   debian-vm       shut off

habibun@hemel-22241042:~$
```

i am trying to take the configuration file of my VM to the home directory first. Command:

[sudo virsh dumpxml debian-vm > /home/habibun/debian-vm.xml](#)

```
habibun@hemel-22241042:~$ sudo virsh dumpxml debian-vm > /home/habibun/debian-vm
.xml
habibun@hemel-22241042:~$
```

After that copy the image file of my virtual machine from the img directory to home directory.command: [cd/var/lib/libvirt/images/](#)

```
habibun@hemel-22241042:~$ cd /var/lib/libvirt/images/
habibun@hemel-22241042:/var/lib/libvirt/images$
```



newDebian-clone-cli.qcow2

```
habibun@hemel-22241042:/var/lib/libvirt/images$ sudo ls --all
.
..
newDebian-clone-cli.qcow2  VirtIODisk3.img
NewDisk.img                VirtIODisk.img
Debian-clone.img  'VirtIODisk$.img'
Debian.img      VirtIODisk2.img
habibun@hemel-22241042:/var/lib/libvirt/images$ sudo cp newDebian-clone-cli.qcow2 /home/habibun
habibun@hemel-22241042:/var/lib/libvirt/images$
```

now for coping the file command: **[sudo cp newDebian-clone-cli.qcow2](#)**
[/home/habibun](#)



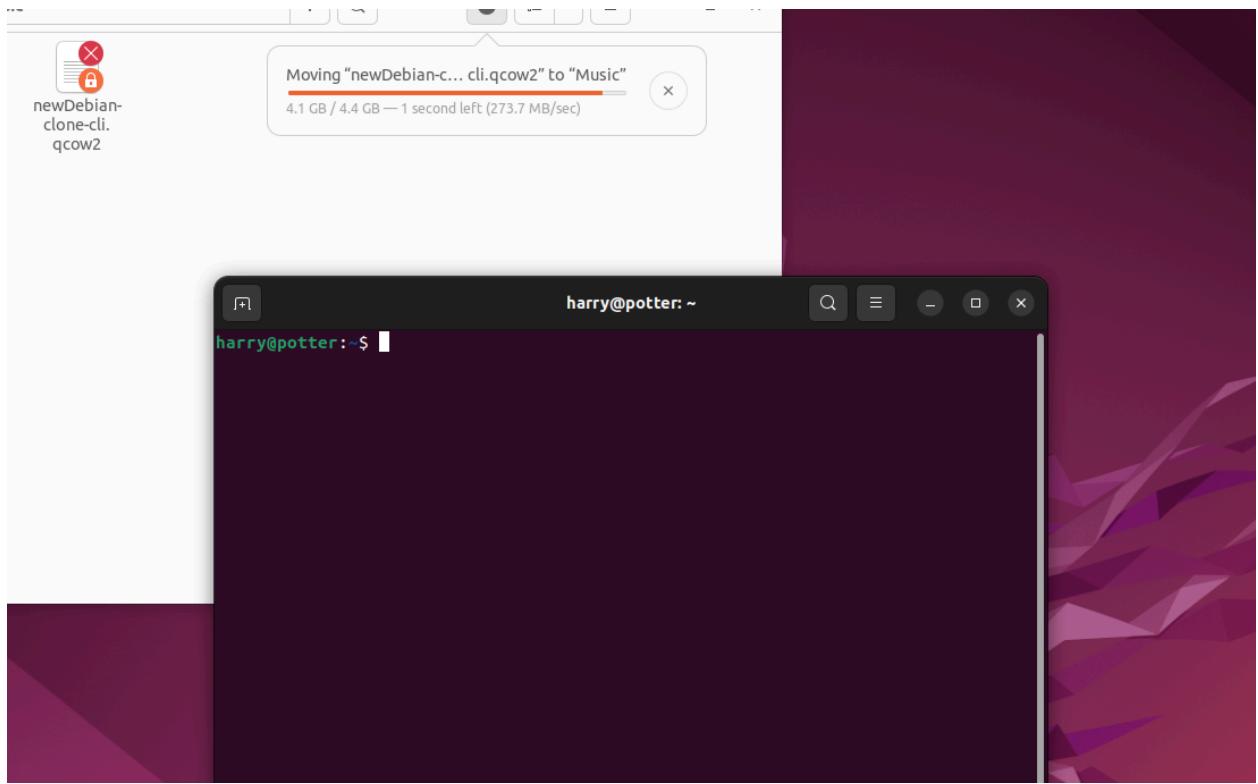
ewDebian-clone-cli.qcow2

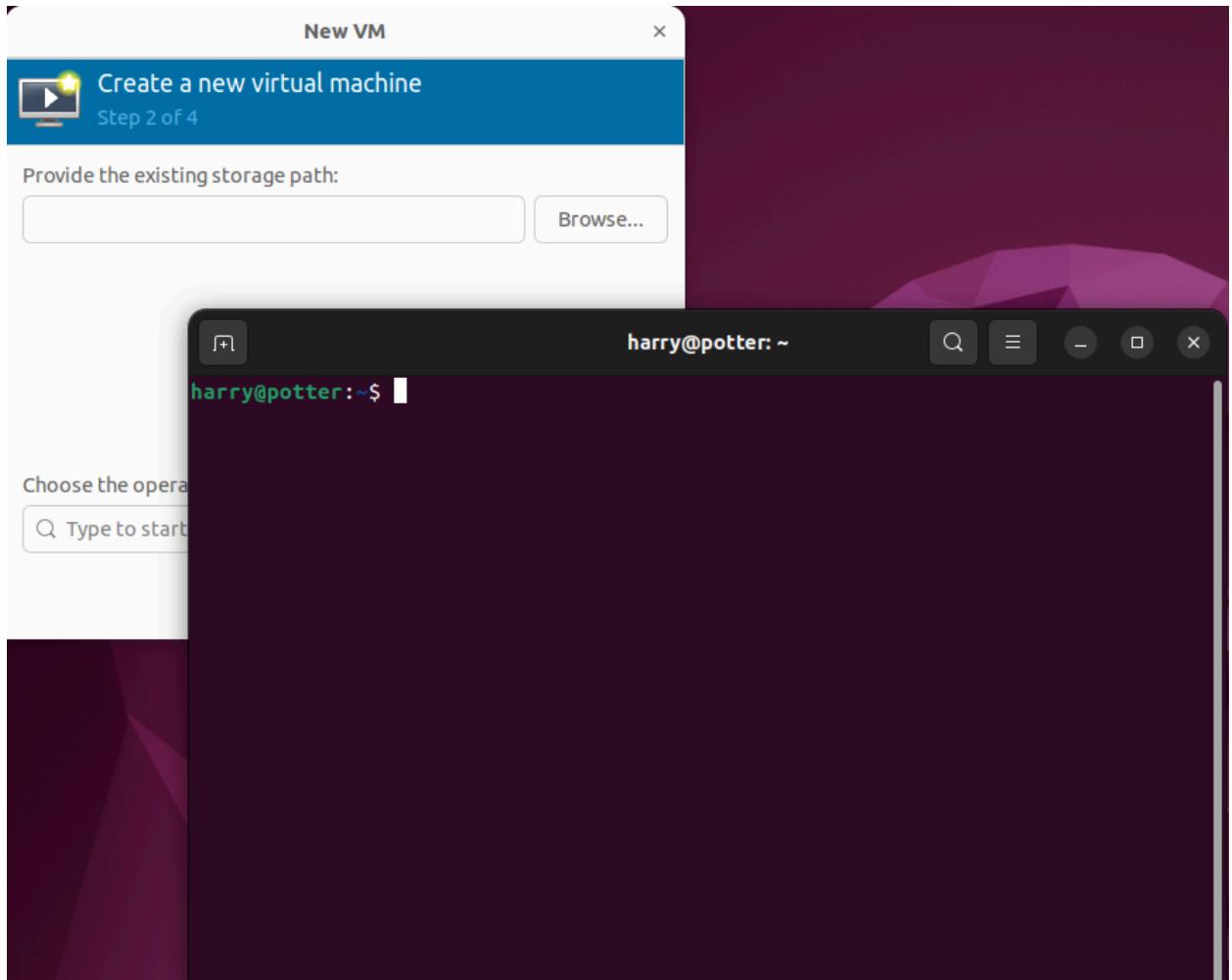
```
E : /home/habibun
habibun@hemel-22241042:/var/lib/libvirt/images$ ^C
habibun@hemel-22241042:/var/lib/libvirt/images$ sudo chown habibun:habibun /home/habibun/newDebian-clone-cli.qcow2
habibun@hemel-22241042:/var/lib/libvirt/images$
```

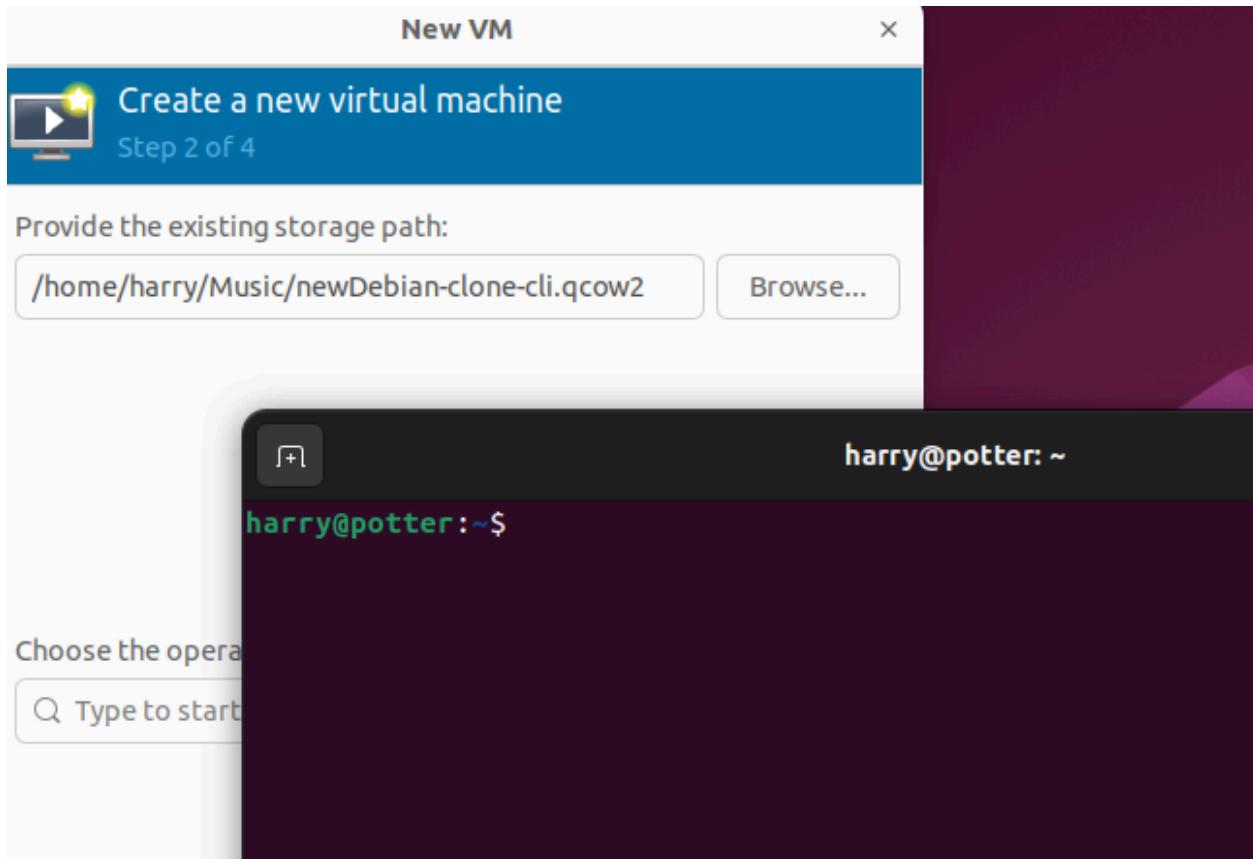
To ensure that a user has permission to send a qcow2 file, you'll need to use the "sudo chown" command. This command allows the adjustment of permissions from the root to the user, as specified in the command.

[sudo chown habibun:habibun /home/habibun/newDebian-clone-cli.qcow2](#)

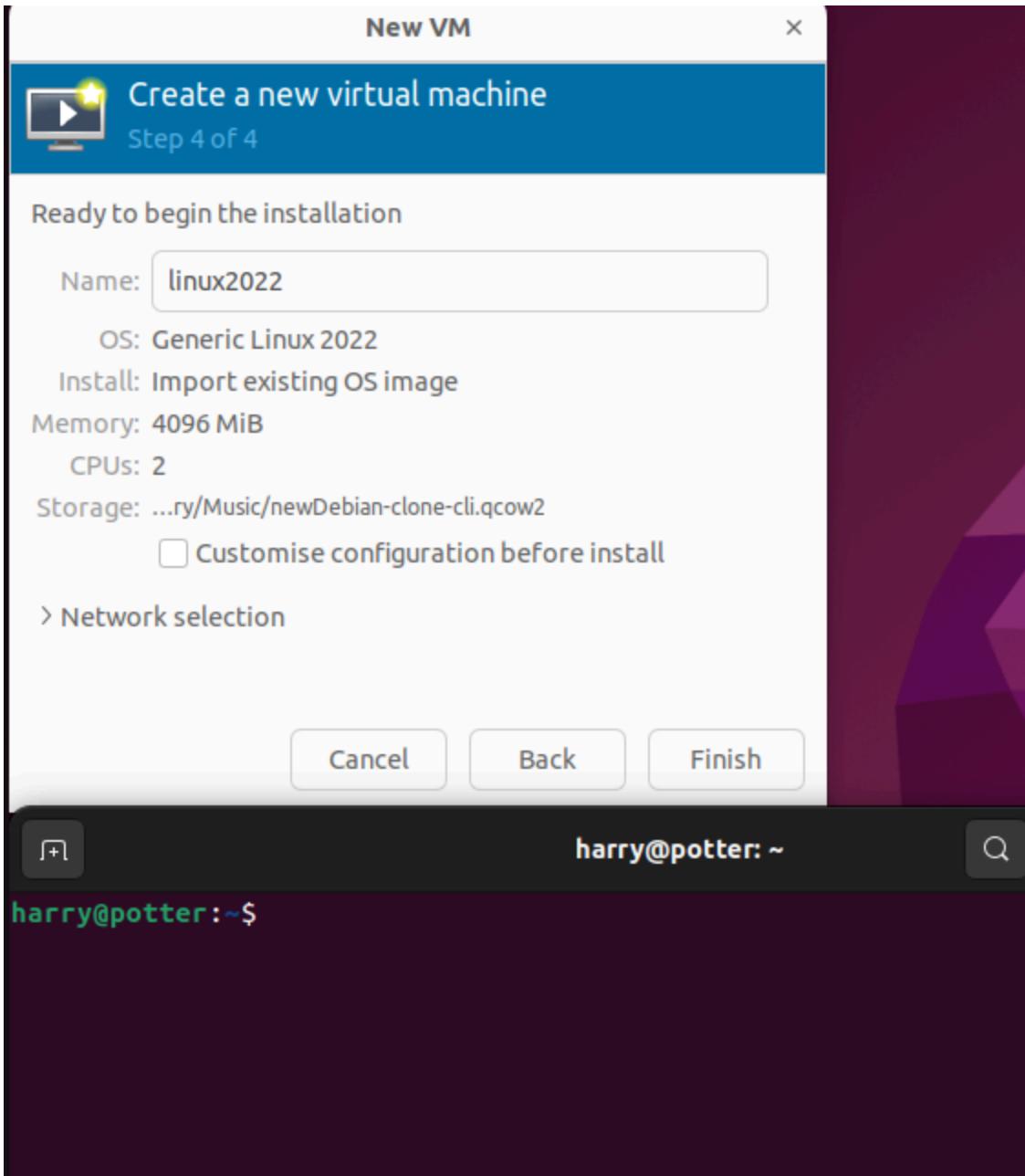
Now, the task is to send the cqow2 file which is a heavy file (4.07gb) and the next part will be taking care of my brother's laptop.

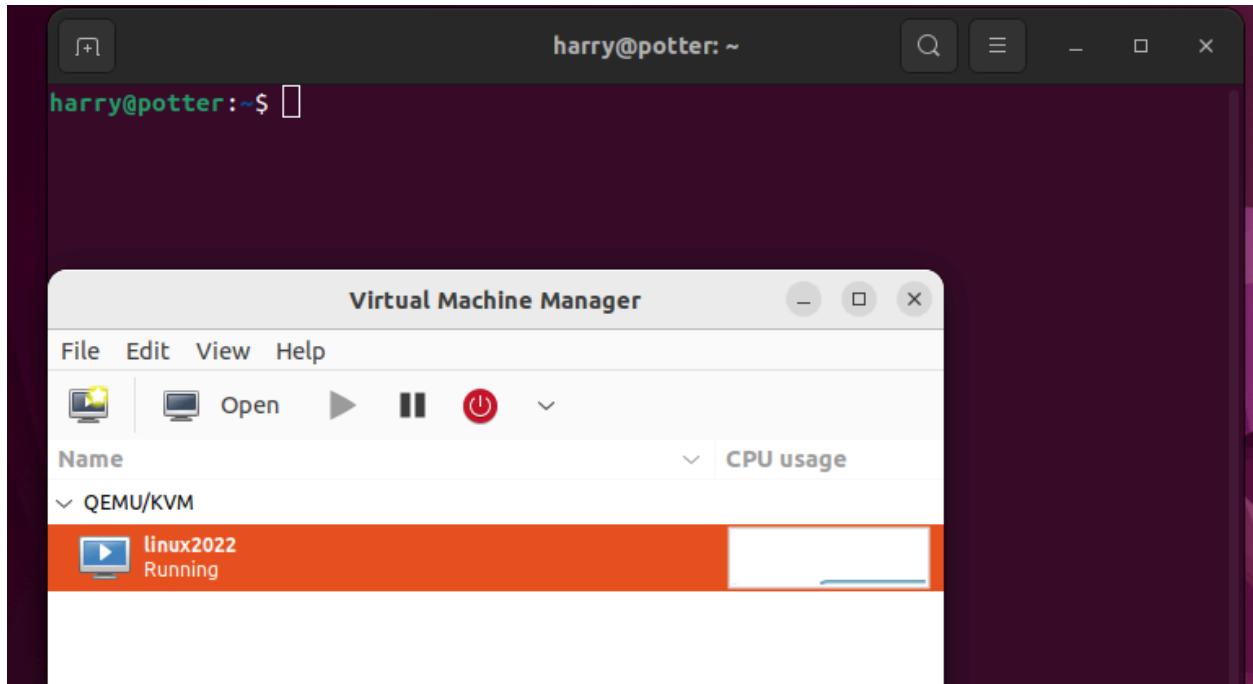






As the VMM did not show the debian9 version so we choose Generic Linux using the migrating the shared OS img .





Finally the migrated VM runs in my friends hardware .

THE END