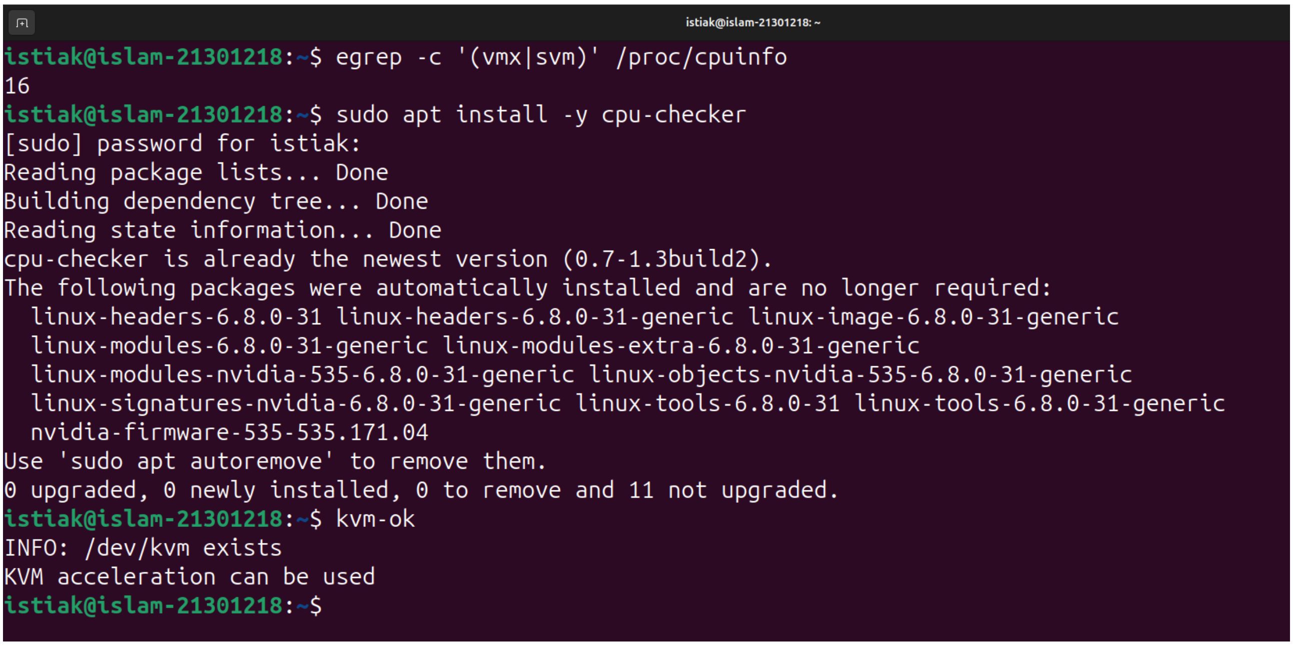
# Availability of KVM on my machine

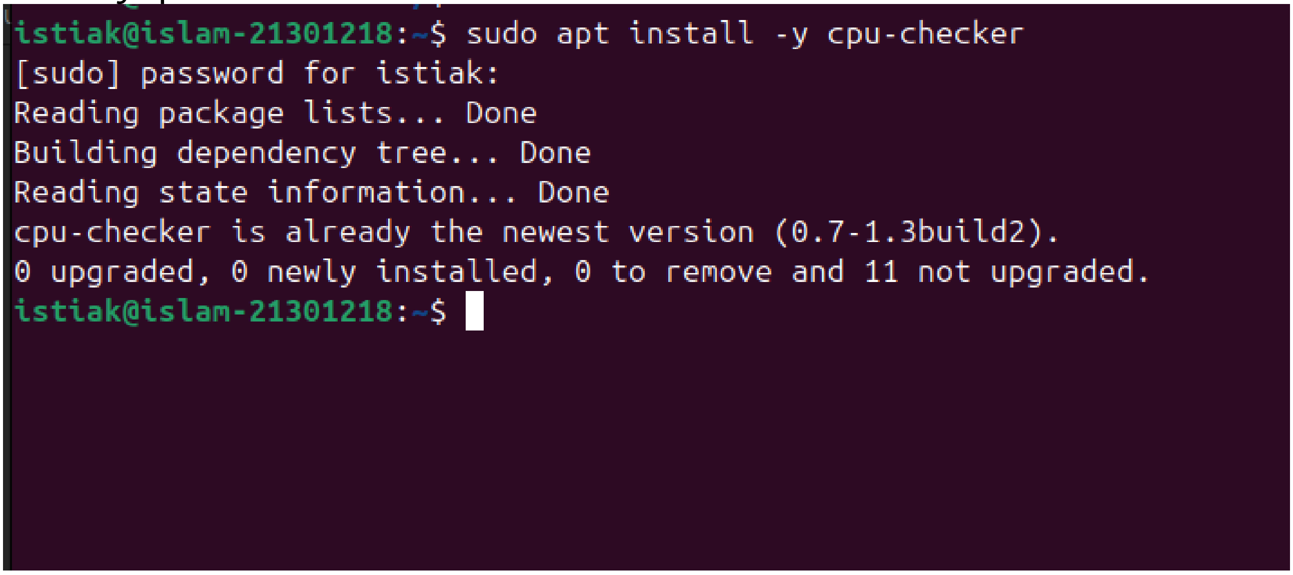
I am using Ubuntu 24.04. First I need to update and upgrade my system by using `sudo apt-get update` then `sudo apt-get upgrade`.
Then we need to check if virtualization enable or not. For this we need to check our CPU supports KVM or not. To obtain this we need to run `egrep -c '(vmx|svm)' /proc/cpuinfo` After run this command if the output is grater than 0 it’s mean our virtualization is enabled.  If it 0 then we need to enable it from BIOS setting.



istiak@islam-21301218: ~

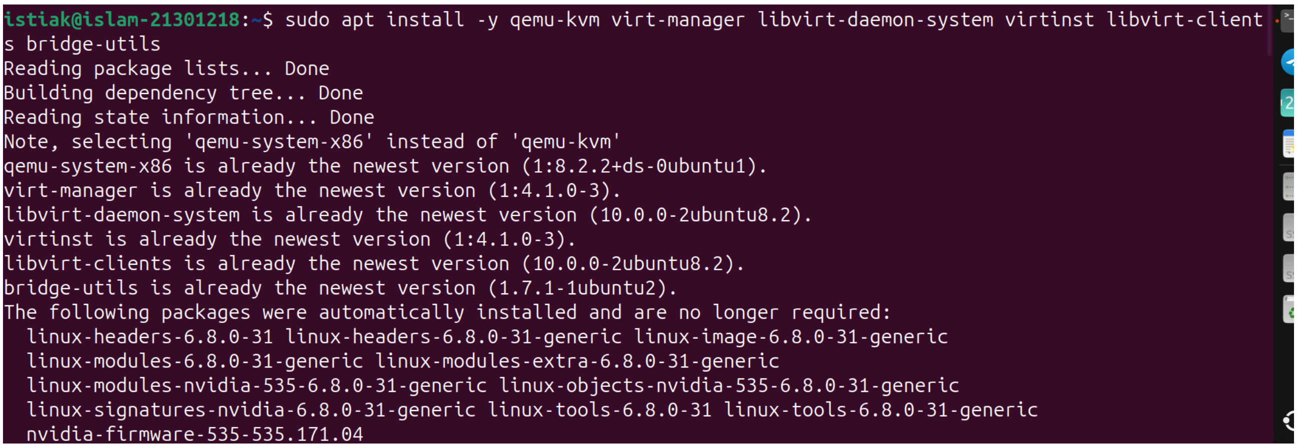
```
istiak@islam-21301218:~$ egrep -c '(vmx|svm)' /proc/cpuinfo
16
istiak@islam-21301218:~$ sudo apt install -y cpu-checker
[sudo] password for istiak:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
cpu-checker is already the newest version (0.7-1.3build2).
The following packages were automatically installed and are no longer required:
  linux-headers-6.8.0-31 linux-headers-6.8.0-31-generic linux-image-6.8.0-31-generic
  linux-modules-6.8.0-31-generic linux-modules-extra-6.8.0-31-generic
  linux-modules-nvidia-535-6.8.0-31-generic linux-objects-nvidia-535-6.8.0-31-generic
  linux-signatures-nvidia-6.8.0-31-generic linux-tools-6.8.0-31 linux-tools-6.8.0-31-generic
  nvidia-firmware-535-535.171.04
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 11 not upgraded.
istiak@islam-21301218:~$ kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
istiak@islam-21301218:~$
```

Moreover, we can verify by `kvm-ok` command. If not ok we will run `sudo apt install -y cpu-checker`. Now it must show KVM is exists.



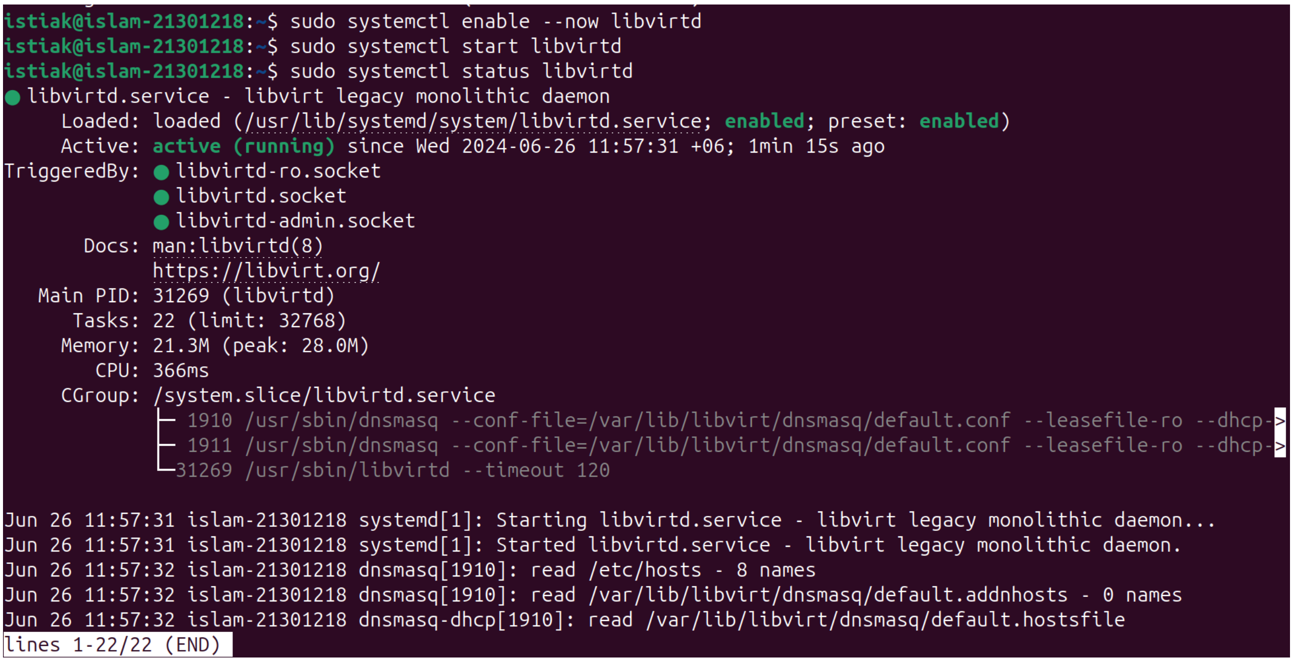
```
istiak@islam-21301218:~$ sudo apt install -y cpu-checker
[sudo] password for istiak:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
cpu-checker is already the newest version (0.7-1.3build2).
0 upgraded, 0 newly installed, 0 to remove and 11 not upgraded.
istiak@islam-21301218:~$
```

Now we will install KVM on my Ubuntu 24.04. To obtain this we need to run this command `*sudo apt install -y qemu-kvm virt-manager libvirt-daemon-system virtinst libvirt-clients bridge-utils*`



```
istiak@islam-21301218:~$ sudo apt install -y qemu-kvm virt-manager libvirt-daemon-system virtinst libvirt-client
s bridge-utils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'qemu-system-x86' instead of 'qemu-kvm'
qemu-system-x86 is already the newest version (1:8.2.2+ds-0ubuntu1).
virt-manager is already the newest version (1:4.1.0-3).
libvirt-daemon-system is already the newest version (10.0.0-2ubuntu8.2).
virtinst is already the newest version (1:4.1.0-3).
libvirt-clients is already the newest version (10.0.0-2ubuntu8.2).
bridge-utils is already the newest version (1.7.1-1ubuntu2).
The following packages were automatically installed and are no longer required:
  linux-headers-6.8.0-31 linux-headers-6.8.0-31-generic linux-image-6.8.0-31-generic
  linux-modules-6.8.0-31-generic linux-modules-extra-6.8.0-31-generic
  linux-modules-nvidia-535-6.8.0-31-generic linux-objects-nvidia-535-6.8.0-31-generic
  linux-signatures-nvidia-6.8.0-31-generic linux-tools-6.8.0-31 linux-tools-6.8.0-31-generic
  nvidia-firmware-535-535.171.04
```

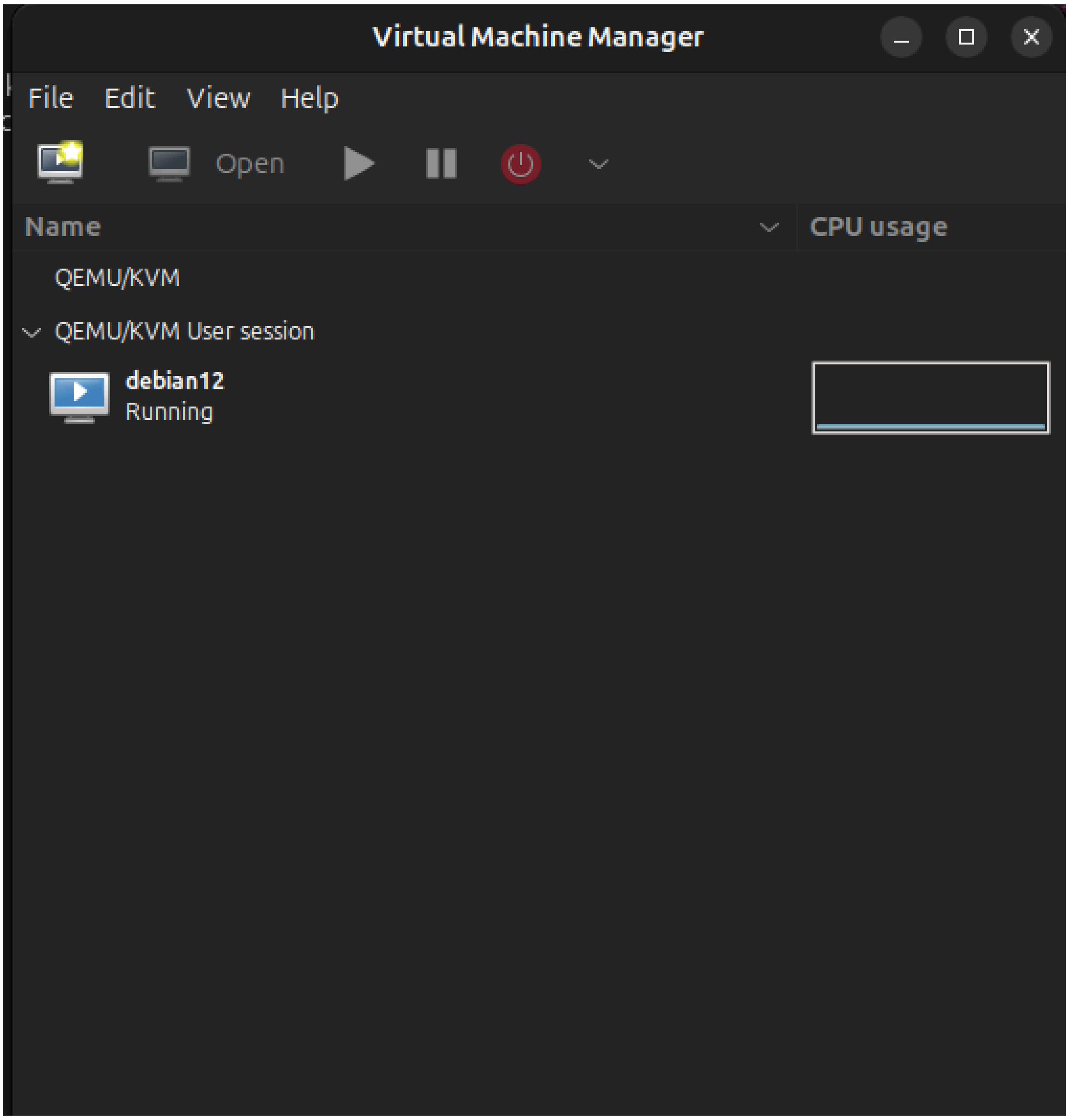
After that we need to enable and start the libvirt deamon. For this we need to run this command respectively
`sudo systemctl enable --now libvirtd`
`sudo systemctl start libvirtd`
we can see the and confirm the virtualization deamon is running by,
`sudo systemctl status libvirtd`



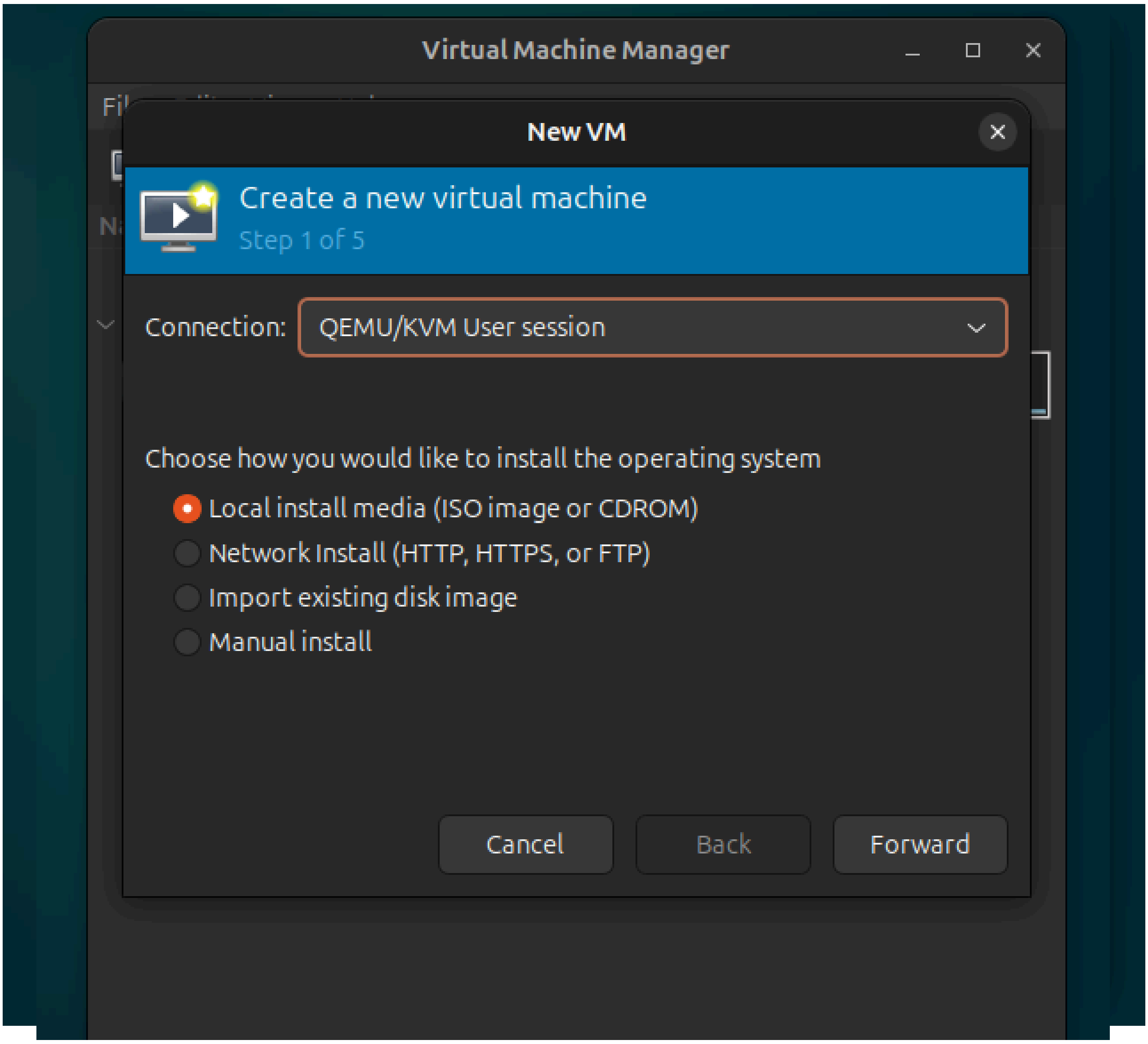
```
istiak@islam-21301218:~$ sudo systemctl enable --now libvirtd
istiak@islam-21301218:~$ sudo systemctl start libvirtd
istiak@islam-21301218:~$ sudo systemctl status libvirtd
● libvirtd.service - libvirt legacy monolithic daemon
     Loaded: loaded (/usr/lib/systemd/system/libvirtd.service; enabled; preset: enabled)
     Active: active (running) since Wed 2024-06-26 11:57:31 +06; 1min 15s ago
TriggeredBy: ● libvirtd-ro.socket
             ● libvirtd.socket
             ● libvirtd-admin.socket
       Docs: man:libvirtd(8)
             https://libvirt.org/
   Main PID: 31269 (libvirtd)
      Tasks: 22 (limit: 32768)
     Memory: 21.3M (peak: 28.0M)
        CPU: 366ms
     CGroup: /system.slice/libvirtd.service
             ├─ 1910 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro --dhcp->
             ├─ 1911 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default.conf --leasefile-ro --dhcp->
             └─31269 /usr/sbin/libvirtd --timeout 120

Jun 26 11:57:31 islam-21301218 systemd[1]: Starting libvirtd.service - libvirt legacy monolithic daemon...
Jun 26 11:57:31 islam-21301218 systemd[1]: Started libvirtd.service - libvirt legacy monolithic daemon.
Jun 26 11:57:32 islam-21301218 dnsmasq[1910]: read /etc/hosts - 8 names
Jun 26 11:57:32 islam-21301218 dnsmasq[1910]: read /var/lib/libvirt/dnsmasq/default.addnhosts - 0 names
Jun 26 11:57:32 islam-21301218 dnsmasq-dhcp[1910]: read /var/lib/libvirt/dnsmasq/default.hostsfile
lines 1-22/22 (END)
```

So far we have installed Virtual Machine Manager.

# create a new virtual machine GUI



First select the ‘create new virtual machine’



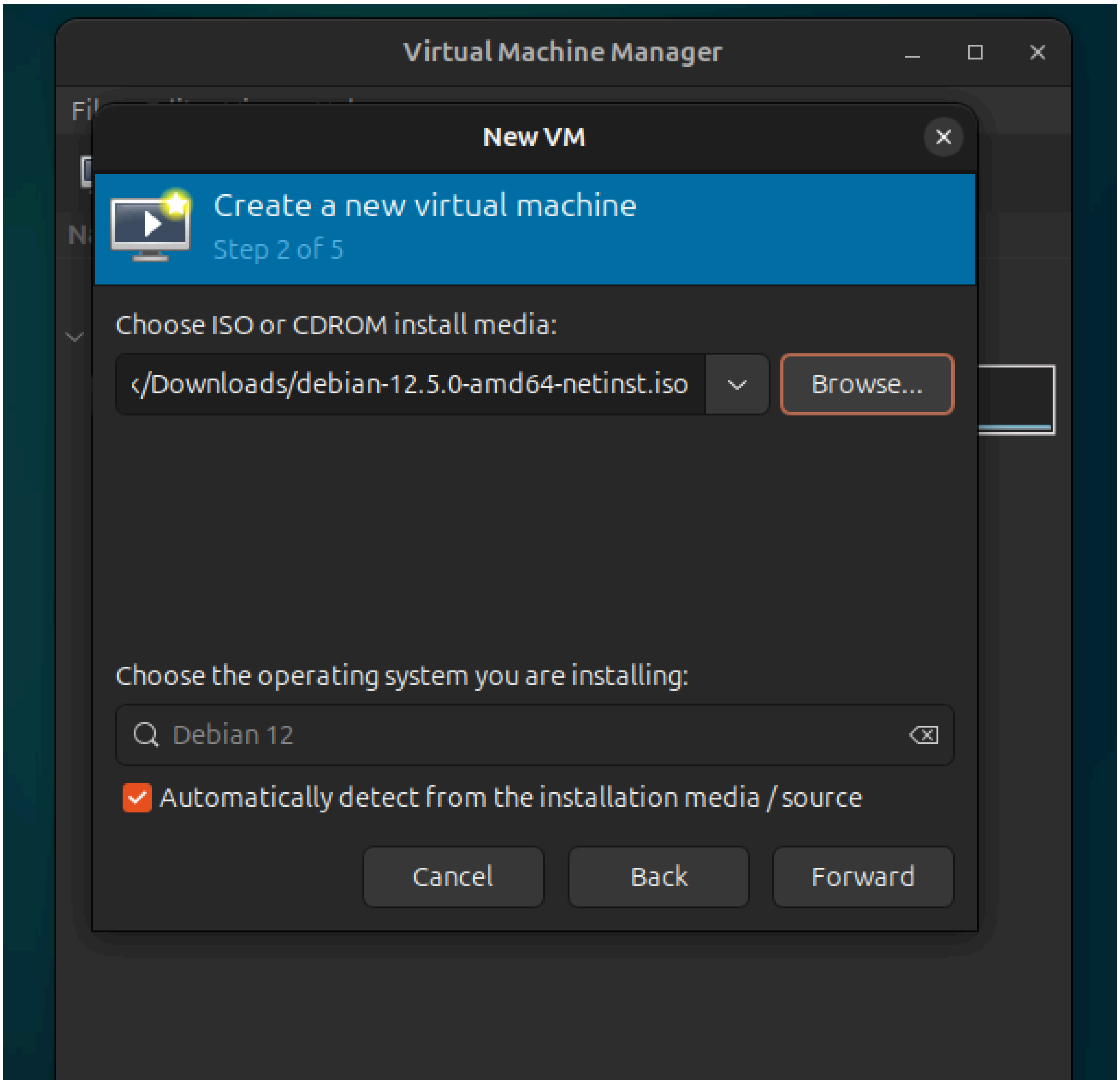
Create a new virtual machine
Step 1 of 5

Connection: QEMU/KVM User session

Choose how you would like to install the operating system

- ● Local install media (ISO image or CDROM)
- ○ Network Install (HTTP, HTTPS, or FTP)
- ○ Import existing disk image
- ○ Manual install

select **forward** with this credential.



Choose ISO or CDROM install media:

‹/Downloads/debian-12.5.0-amd64-netinst.iso

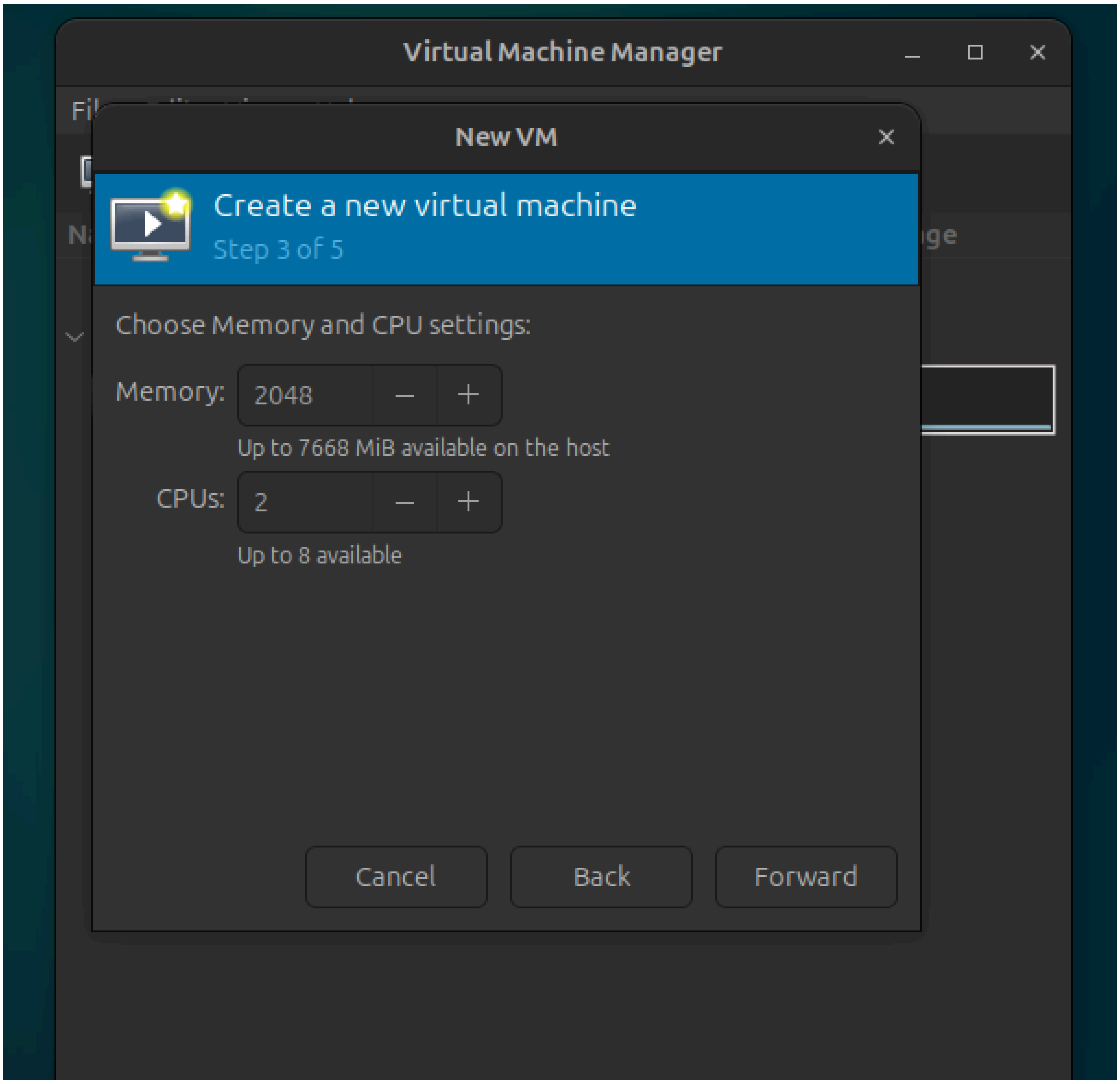
Choose the operating system you are installing:

Debian 12

☑ Automatically detect from the installation media / source

Browse the desire iso file. Here I have selected debian 12.5 then press **Forward**.

Select desire memory and cpu core. Then select Forward.



Choose Memory and CPU settings:

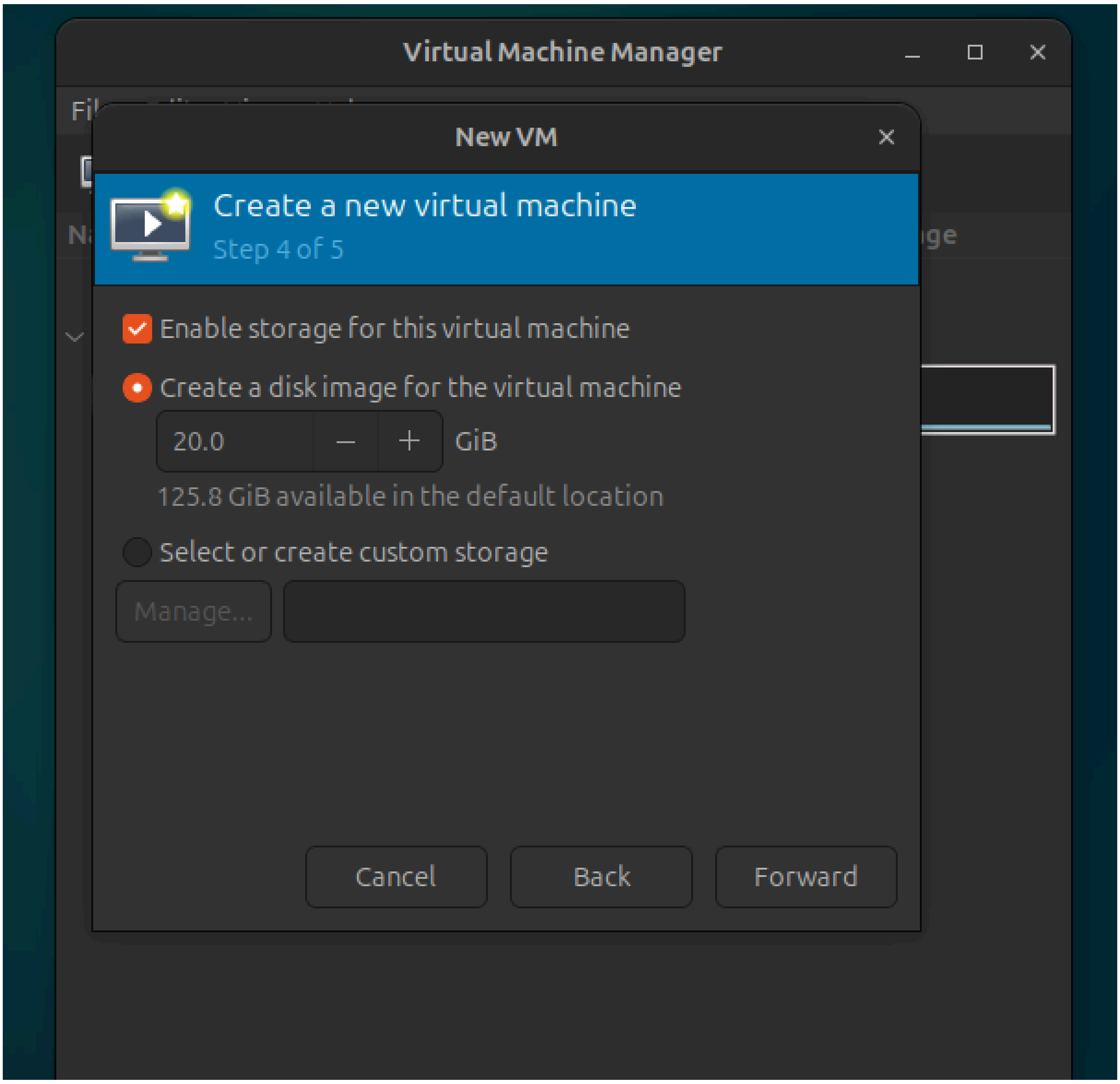
Memory: 2048

Up to 7668 MiB available on the host

CPUs: 2

Up to 8 available

Now Provide disk image for virtual machine. 20 Gib or less is enough for beginner. Then press Forward.



Virtual Machine Manager

New VM

Create a new virtual machine

Step 4 of 5

☑ Enable storage for this virtual machine

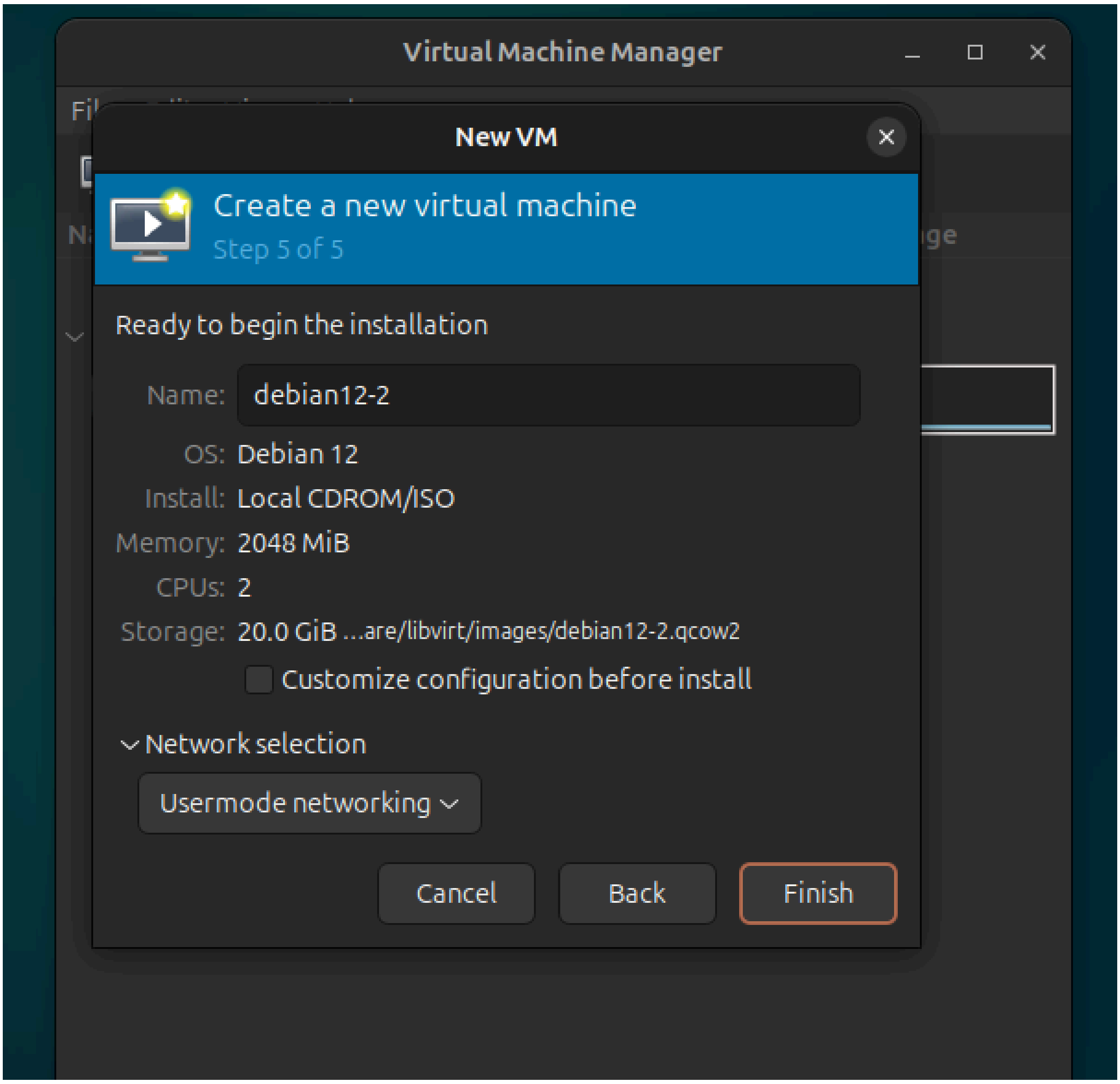
● Create a disk image for the virtual machine

20.0 GiB

125.8 GiB available in the default location

○ Select or create custom storage

Here we can rename our virtual machine. I keep it as it is. As I installing debian 12 so it showing debian 12-2 as name. Pressing Finish it will create a new virtual machine.



Ready to begin the installation

Name: debian12-2

OS: Debian 12

Install: Local CDROM/ISO

Memory: 2048 MiB

CPUs: 2

Storage: 20.0 GiB ...are/libvirt/images/debian12-2.qcow2

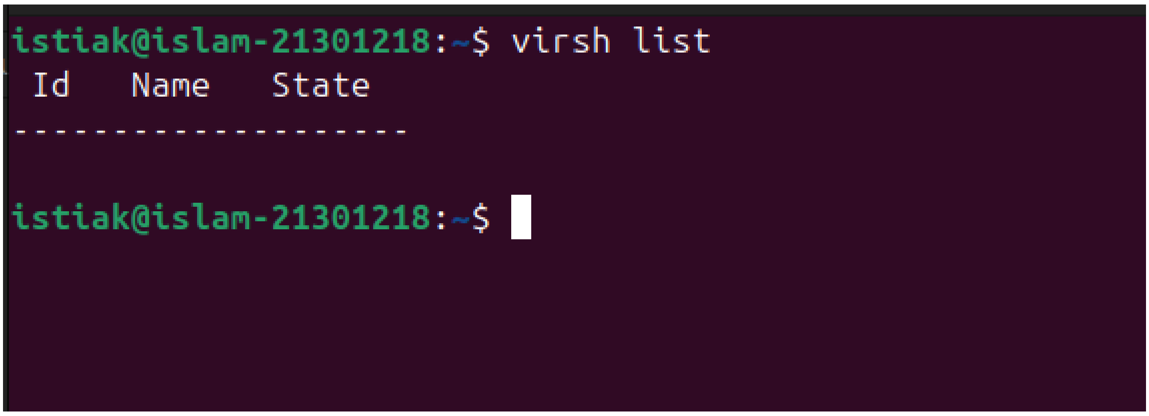
□ Customize configuration before install

Network selection

Usermode networking

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

To see the virtual machine from the cli we can run `virsh list` Primarily it show an empty list of virtual machine as we don’t install yet any.



```
istiak@islam-21301218:~$ virsh list
 Id   Name   State
--------------------

istiak@islam-21301218:~$ 
```

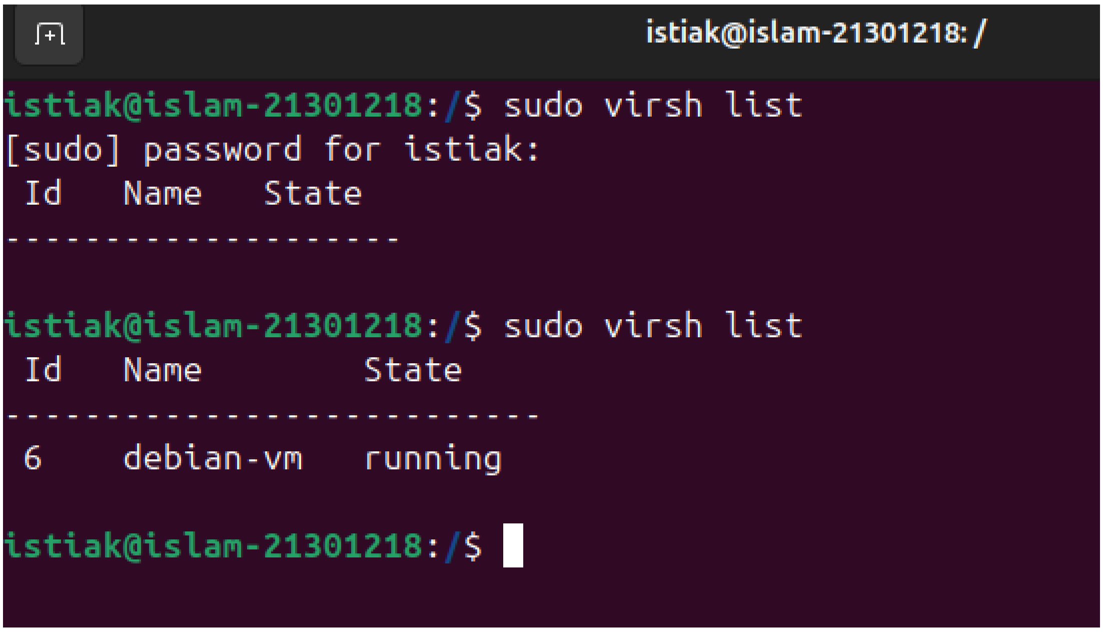
Now in our local computer install virt manager using this command `sudo apt install virt-manager`

I used this code for install kvm-based VM using CLI:

```
virt-install \
--name debian-vm \
--ram 2048 \
--vcpus 2 \
--disk path=/var/lib/libvirt/images/debian12-vm.img,size=20 \
--os-variant debian12\
--network bridge=virbr0,model=virtio \
--graphics none \
--console pty,target_type=serial \
--cdrom ~/Downloads/debian-12.5.0-amd64-netinst.iso
```

after that lets check from the CLI wither 'debian-vm' create or not. For that we need to run this command,

*`sudo virsh list`*



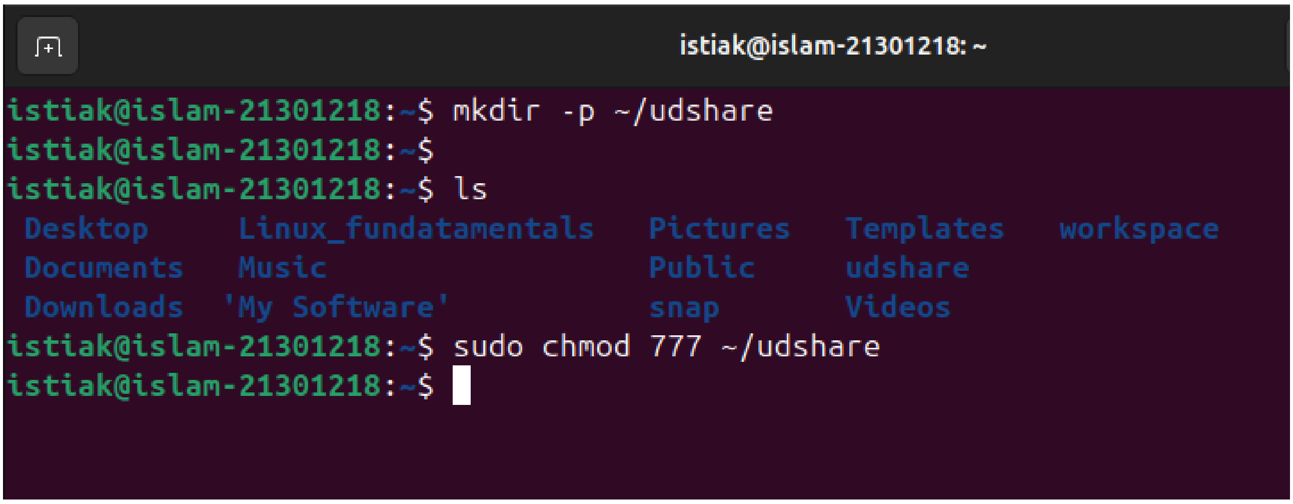
```
istiak@islam-21301218:/$ sudo virsh list
[sudo] password for istiak:
 Id   Name   State
--------------------

istiak@islam-21301218:/$ sudo virsh list
 Id   Name        State
---------------------------
 6    debian-vm   running

istiak@islam-21301218:/$ 
```

we can see our new vm has been created.

## Share Folder within Host and Guest

First need to create a folder for share and give permission to the shared folder to read write This folder should be the parent folder in the host os. So run this command:
`mkdir -p ~/udshare` # For creating parent folder

`sudo chmod 777 ~/udshare` #For giving permission

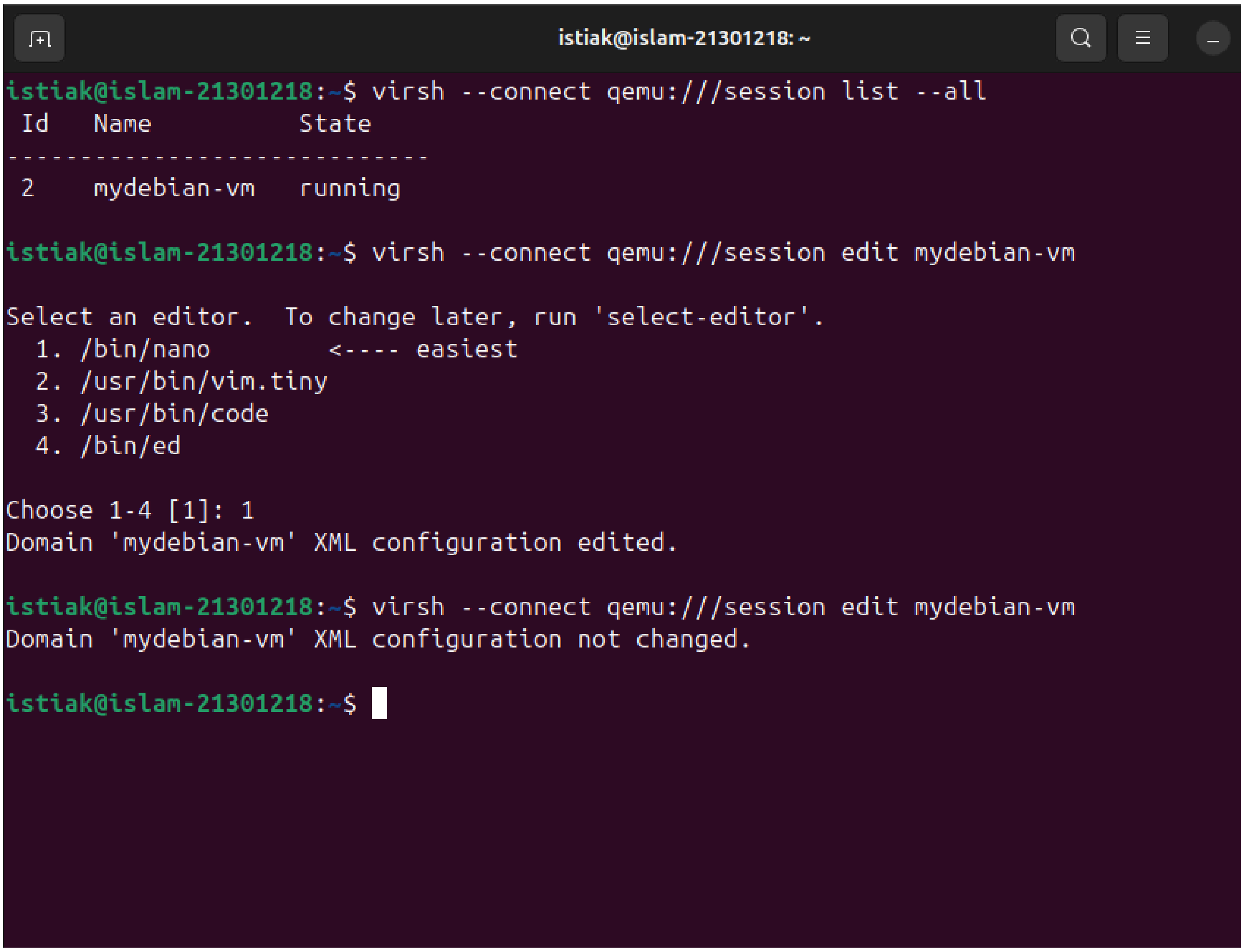


```
istiak@islam-21301218:~$ mkdir -p ~/udshare
istiak@islam-21301218:~$
istiak@islam-21301218:~$ ls
 Desktop      Linux_fundatamentals   Pictures   Templates   workspace
 Documents    Music                  Public     udshare
 Downloads   'My Software'           snap       Videos
istiak@islam-21301218:~$ sudo chmod 777 ~/udshare
istiak@islam-21301218:~$ 
```

Now we need to configure the file system for sharing folder.

First we need to edit the file with:

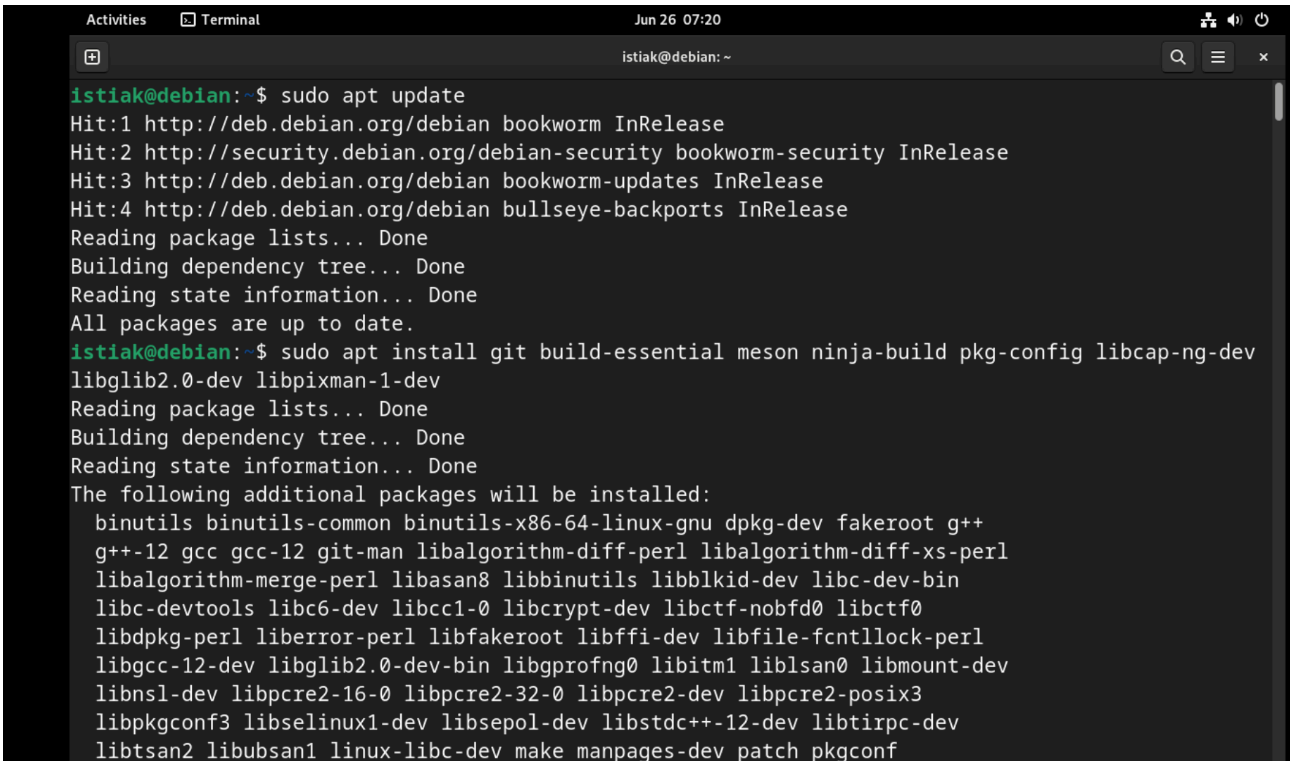
```
<filesystem
  type='mount' accessmode='passthrough'>
  <source dir='/home/istiak/udshare'/>
  <target dir='udshare'/>
  <readonly/>
</filesystem>
```



```
istiak@islam-21301218:~$ virsh --connect qemu:///session list --all
 Id   Name          State
-----------------------------
 2    mydebian-vm   running

istiak@islam-21301218:~$ virsh --connect qemu:///session edit mydebian-vm

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.tiny
  3. /usr/bin/code
  4. /bin/ed

Choose 1-4 [1]: 1
Domain 'mydebian-vm' XML configuration edited.

istiak@islam-21301218:~$ virsh --connect qemu:///session edit mydebian-vm
Domain 'mydebian-vm' XML configuration not changed.

istiak@islam-21301218:~$ 
```

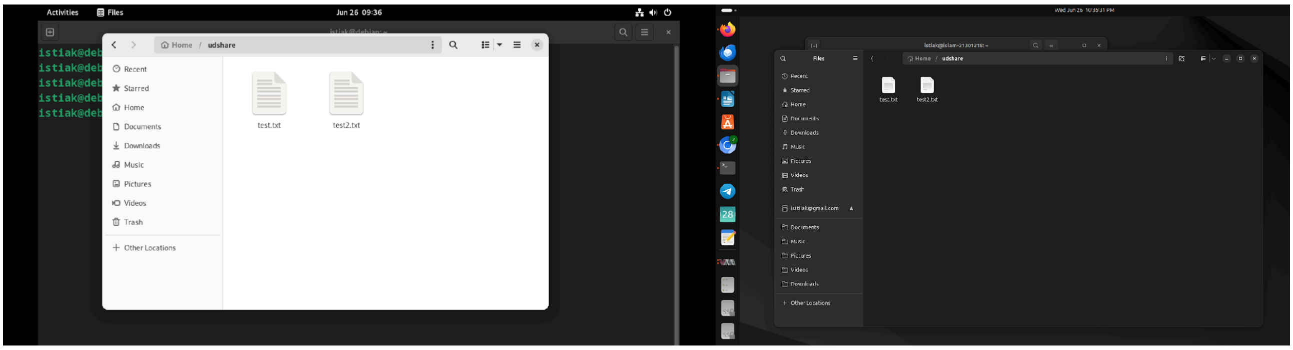
Now we need to install VirtioFSD for sharing file both Guest and Host. To obtain this we need to run install this in our guest os mean, mydebian-vm.

`sudo apt install git build-essential meson ninja-build pkg-config libcap-ng-dev libglib2.0-dev libpixman-1-dev`



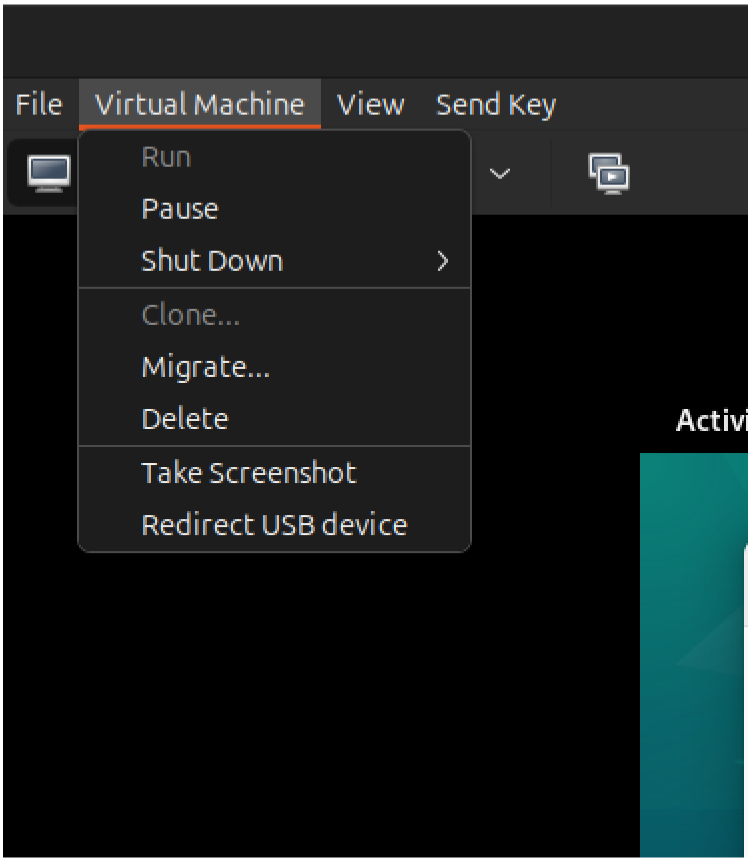
```
istiak@debian:~$ sudo apt update
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 http://security.debian.org/debian-security bookworm-security InRelease
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease
Hit:4 http://deb.debian.org/debian bullseye-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
istiak@debian:~$ sudo apt install git build-essential meson ninja-build pkg-config libcap-ng-dev
libglib2.0-dev libpixman-1-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu dpkg-dev fakeroot g++
  g++-12 gcc gcc-12 git-man libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan8 libbinutils libblkid-dev libc-dev-bin
  libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0
  libdpkg-perl liberror-perl libfakeroot libffi-dev libfile-fcntllock-perl
  libgcc-12-dev libglib2.0-dev-bin libgprofng0 libitm1 liblsan0 libmount-dev
  libnsl-dev libpcre2-16-0 libpcre2-32-0 libpcre2-dev libpcre2-posix3
  libpkgconf3 libselinux1-dev libsepol-dev libstdc++-12-dev libtirpc-dev
  libtsan2 libubsan1 linux-libc-dev make manpages-dev patch pkgconf
```

Here we can see the share folder within guest os and Host os.

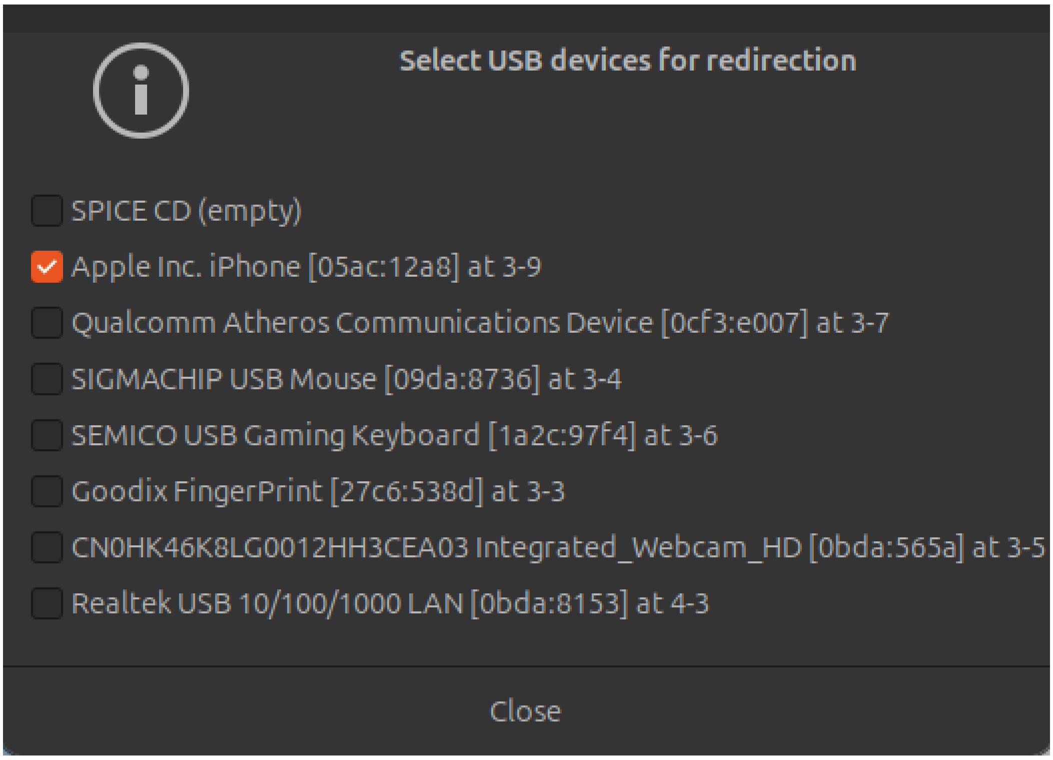


## Connecting my phone to Guest OS

To do this first we need to connect our phone to the laptop. Then go to the guest os. Then select the USB redirection.

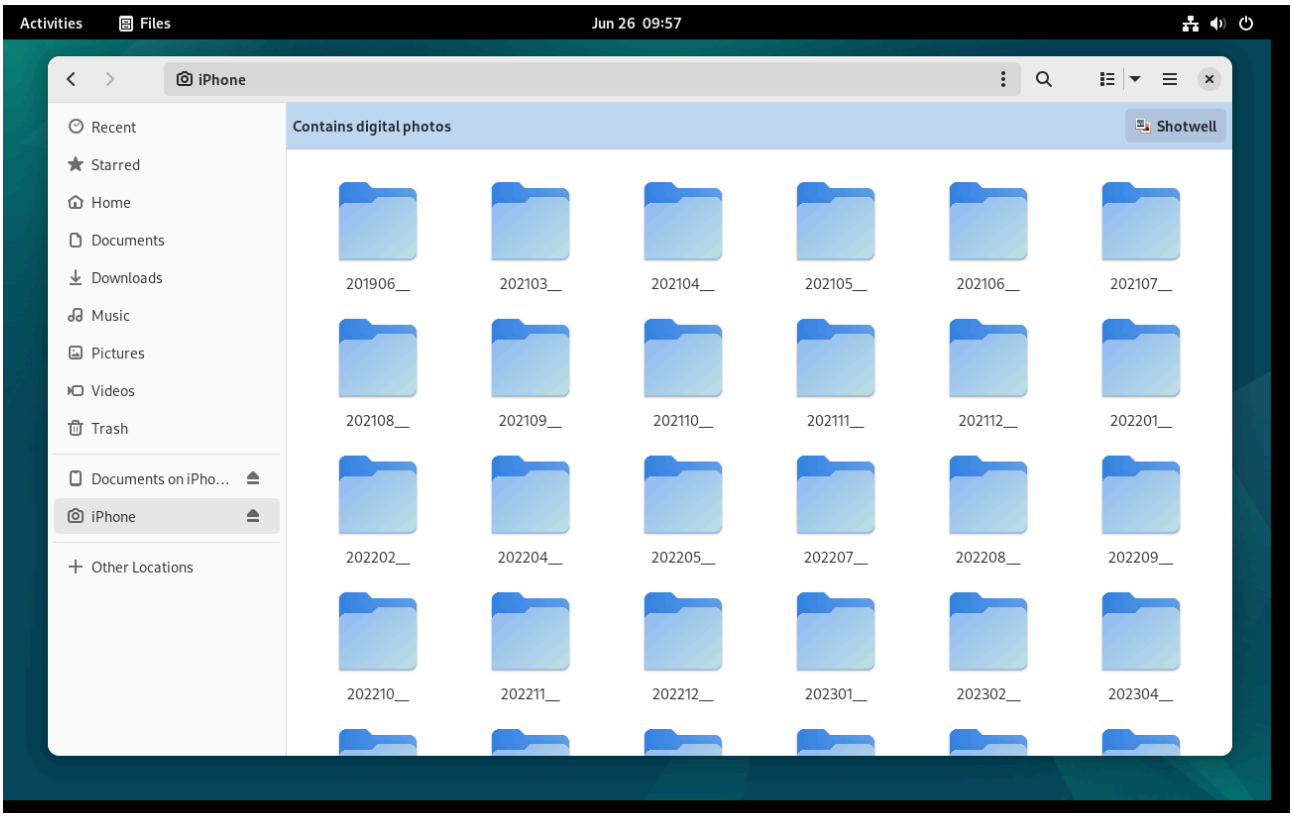


After selecting the Redirect USB device. We need select the device.
In my case, I have select Apple Inc as my phone is an Apple device. Then close it and go to the file manager.


Select USB devices for redirection

- ☐ SPICE CD (empty)
- ☑ Apple Inc. iPhone [05ac:12a8] at 3-9
- ☐ Qualcomm Atheros Communications Device [0cf3:e007] at 3-7
- ☐ SIGMACHIP USB Mouse [09da:8736] at 3-4
- ☐ SEMICO USB Gaming Keyboard [1a2c:97f4] at 3-6
- ☐ Goodix FingerPrint [27c6:538d] at 3-3
- ☐ CN0HK46K8LG0012HH3CEA03 Integrated_Webcam_HD [0bda:565a] at 3-5
- ☐ Realtek USB 10/100/1000 LAN [0bda:8153] at 4-3

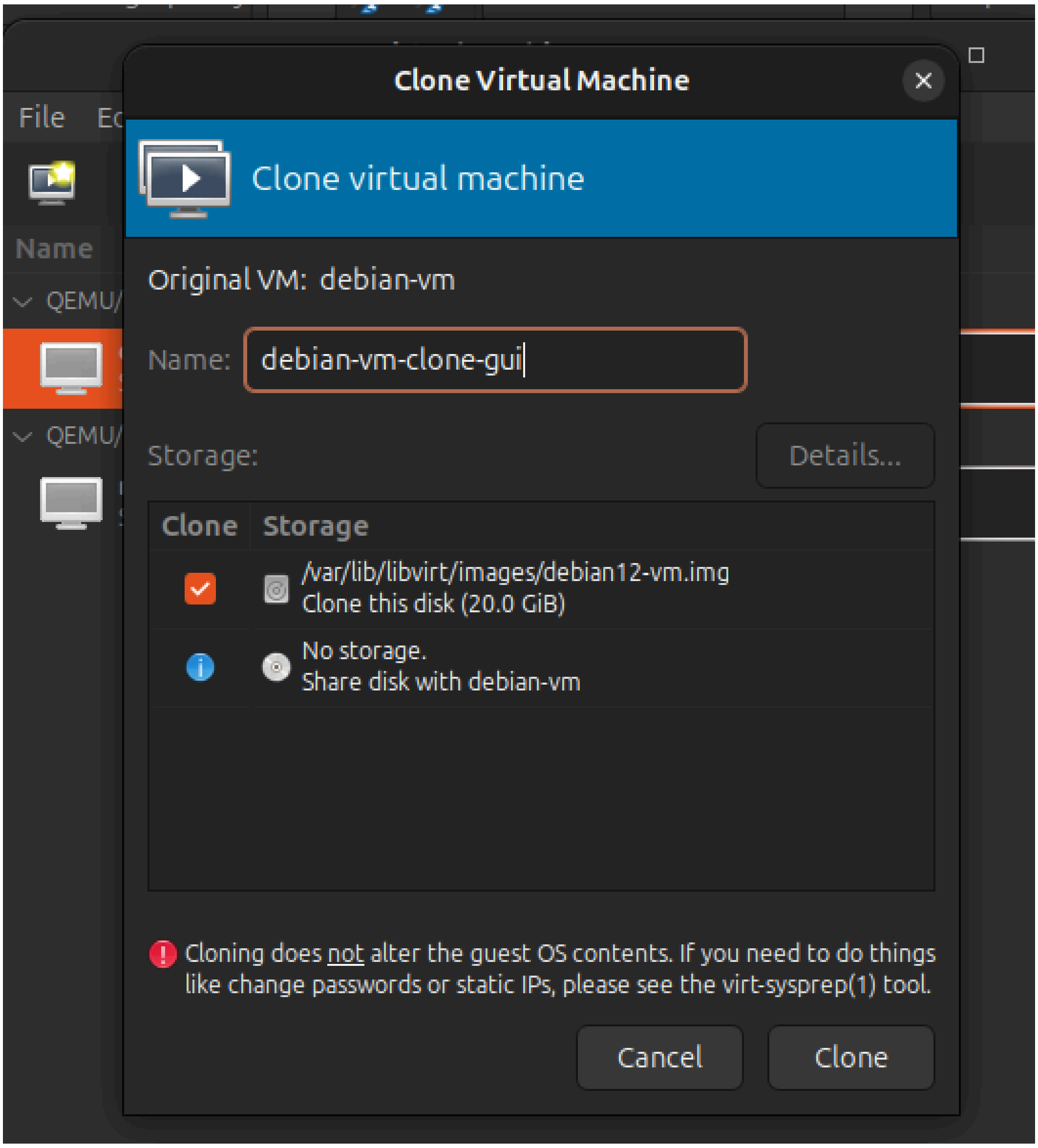
After closing, we open the file and see the our phone in the guest os.



Here are all the file of my phone. From here we can transfer file as we need.

## **Cloning our VM using GUI**

For cloning first we need to shut down our existing vm.



Original VM: debian-vm

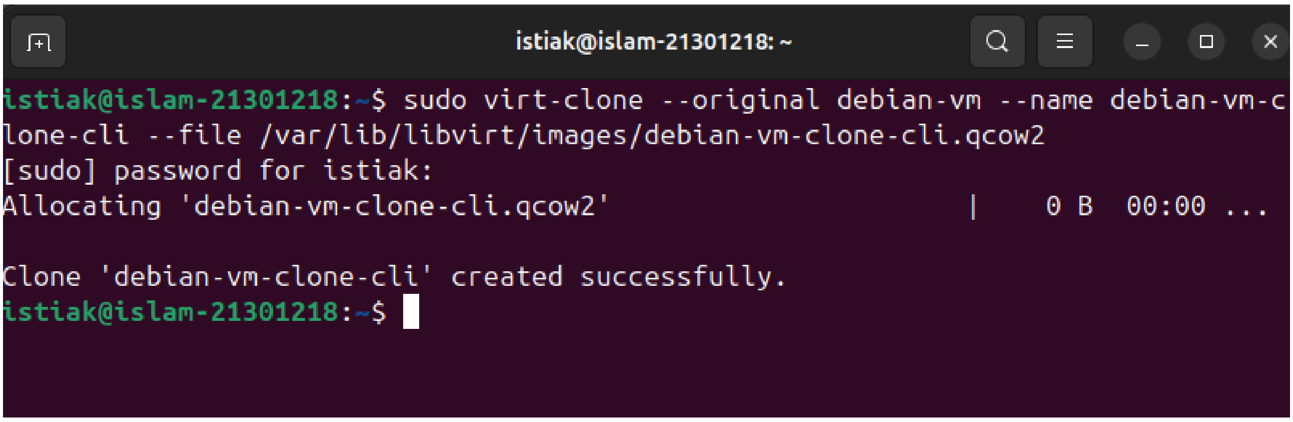
Name: debian-vm-clone-gui

Storage:

| Clone | Storage |
|---|---|
| ☑ | /var/lib/libvirt/images/debian12-vm.img<br>Clone this disk (20.0 GiB) |
| ⓘ | No storage.<br>Share disk with debian-vm |

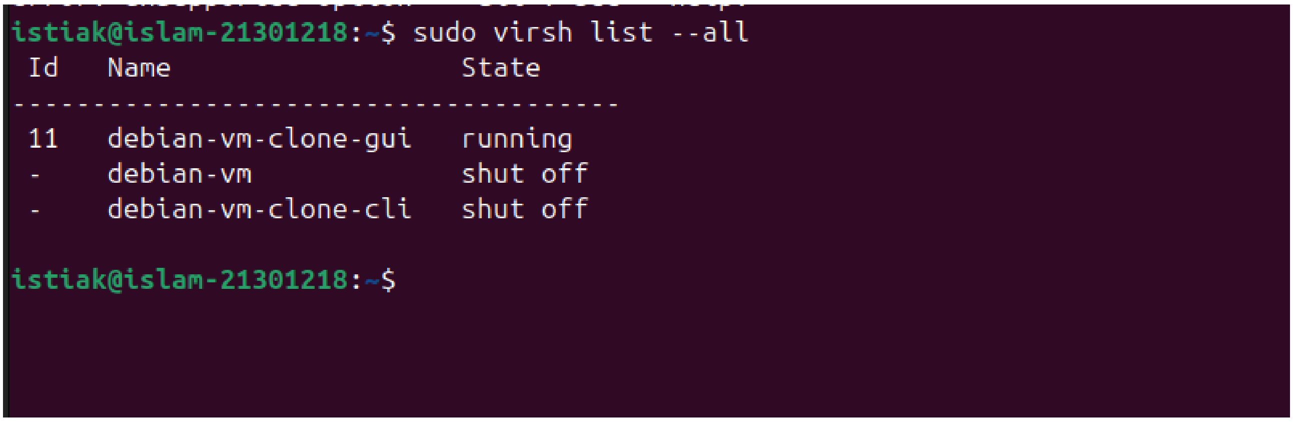
Cloning does not alter the guest OS contents. If you need to do things like change passwords or static IPs, please see the virt-sysprep(1) tool.

With the cli:



```
istiak@islam-21301218:~$ sudo virt-clone --original debian-vm --name debian-vm-c
lone-cli --file /var/lib/libvirt/images/debian-vm-clone-cli.qcow2
[sudo] password for istiak:
Allocating 'debian-vm-clone-cli.qcow2'                        |    0 B  00:00 ...

Clone 'debian-vm-clone-cli' created successfully.
istiak@islam-21301218:~$ 
```

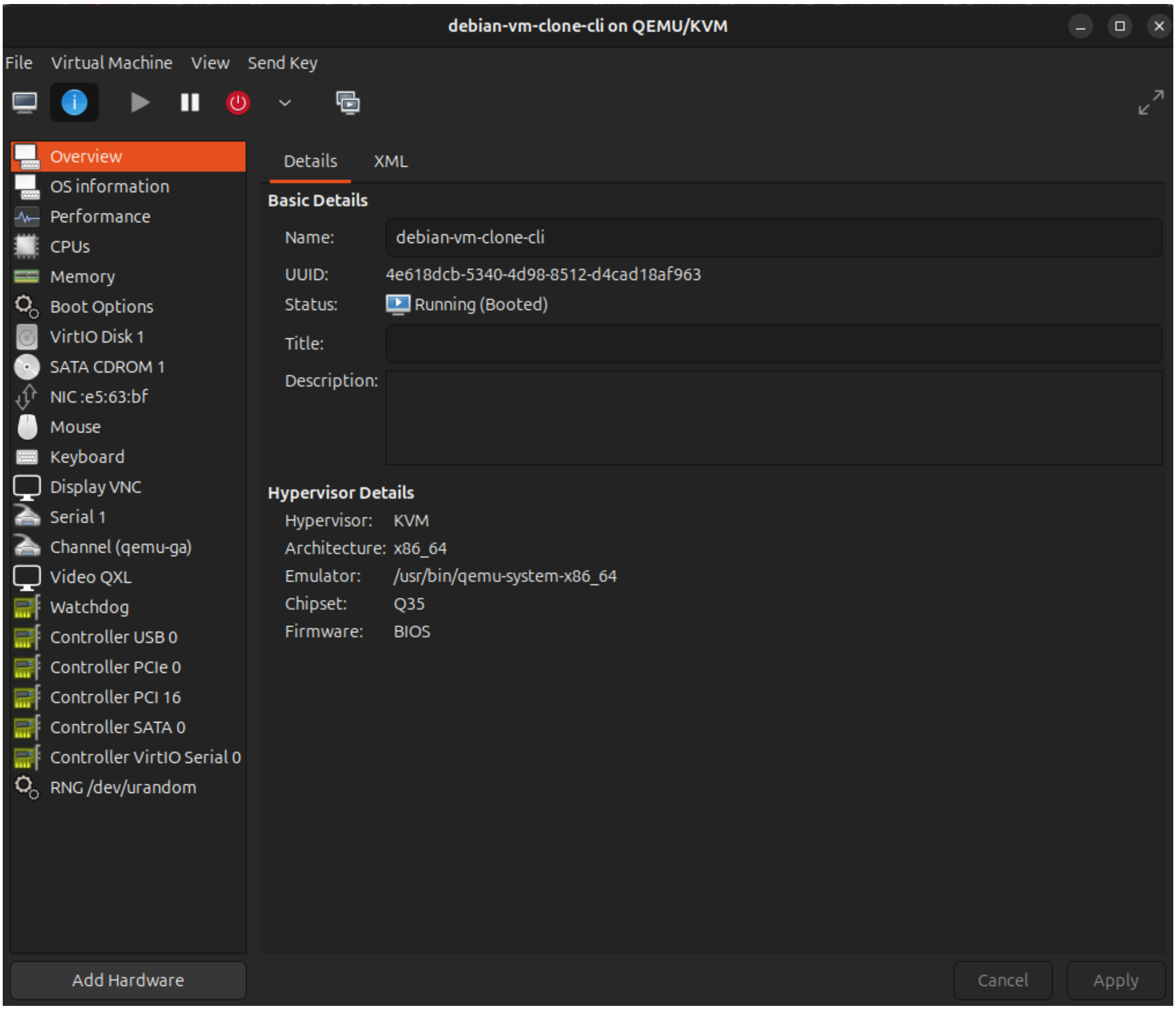
Here are the all vm and cloned vm:



```
istiak@islam-21301218:~$ sudo virsh list --all
 Id   Name                  State
---------------------------------------
 11   debian-vm-clone-gui   running
 -    debian-vm             shut off
 -    debian-vm-clone-cli   shut off

istiak@islam-21301218:~$
```

# Adding Two hard disk on newly cloned VM

First we need to run one cloned VM. Then press the ‘i’ icon for overview of vm.



Basic Details

Name: debian-vm-clone-cli

UUID: 4e618dcb-5340-4d98-8512-d4cad18af963

Status: Running (Booted)

Title:

Description:

Hypervisor Details

Hypervisor: KVM

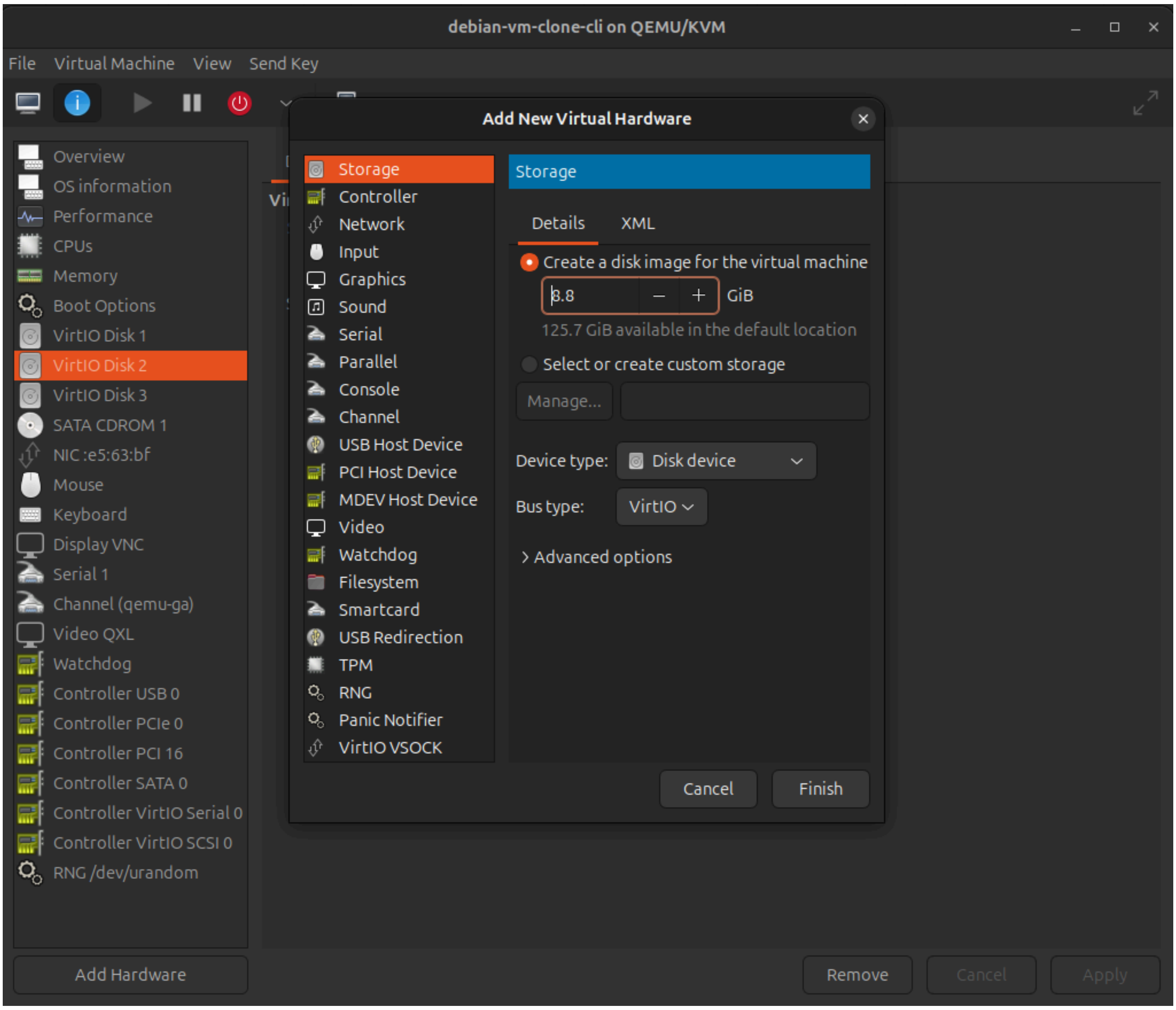
Architecture: x86_64

Emulator: /usr/bin/qemu-system-x86_64

Chipset: Q35

Firmware: BIOS

Now For creating virtual disk press the ‘Add Hardware’ then select desire space and click on finish.



● Create a disk image for the virtual machine

8.8 GiB

125.7 GiB available in the default location

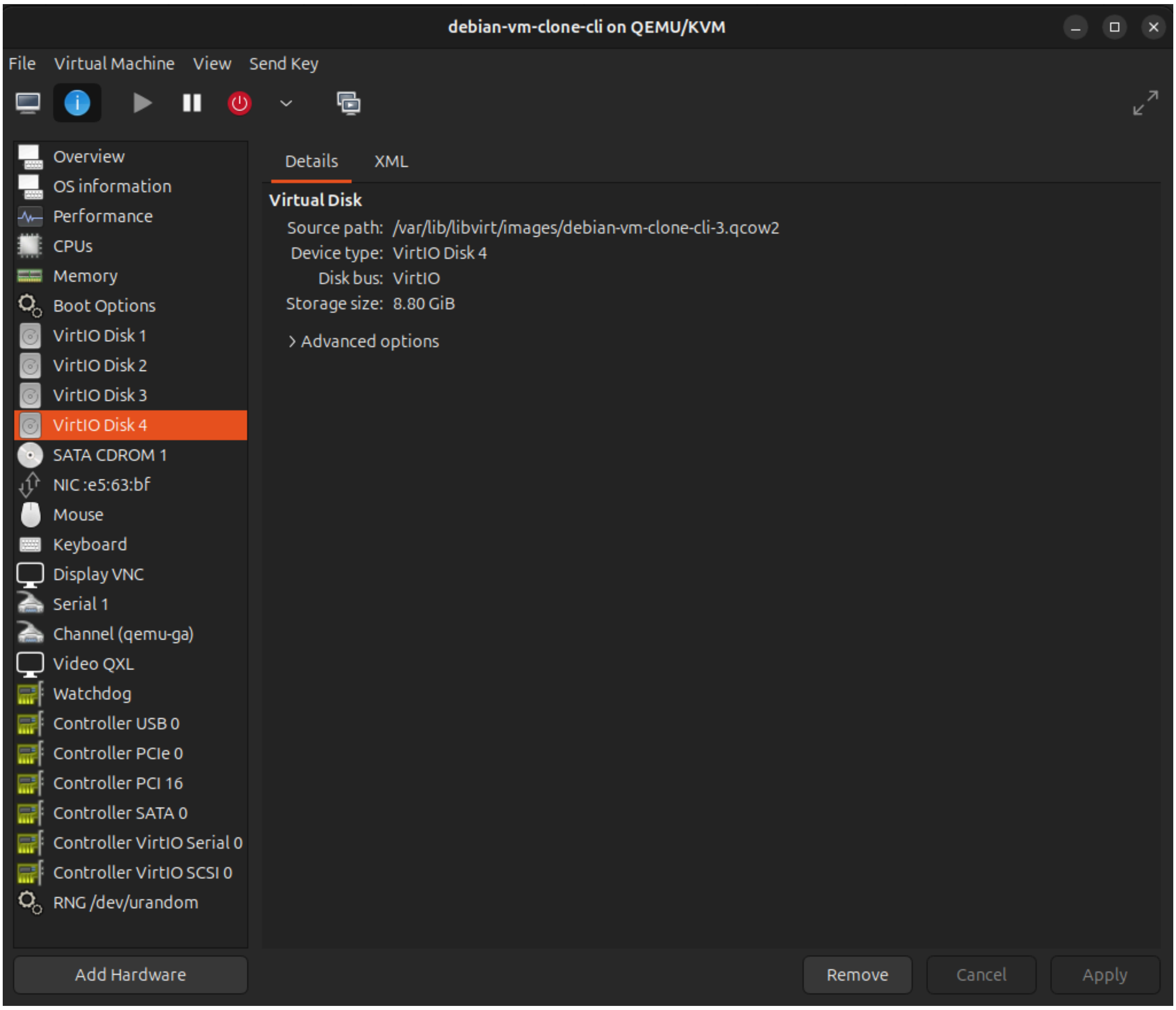
○ Select or create custom storage

Device type: Disk device

Bus type: VirtIO

I have added total 4 virtual disk

Here are the all four (previous one) virtual disk.



Virtual Disk

Source path: /var/lib/libvirt/images/debian-vm-clone-cli-3.qcow2
Device type: VirtIO Disk 4
Disk bus: VirtIO
Storage size: 8.80 GiB

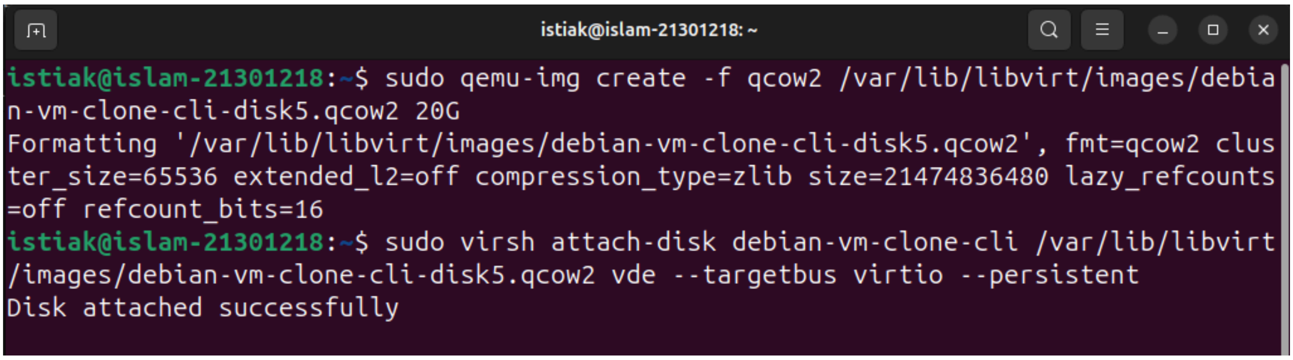
› Advanced options

Now we can create virtual disk from the CLI. For this we need to run this command. It will create a ‘VirtIO Disk 5’ with 20GB.

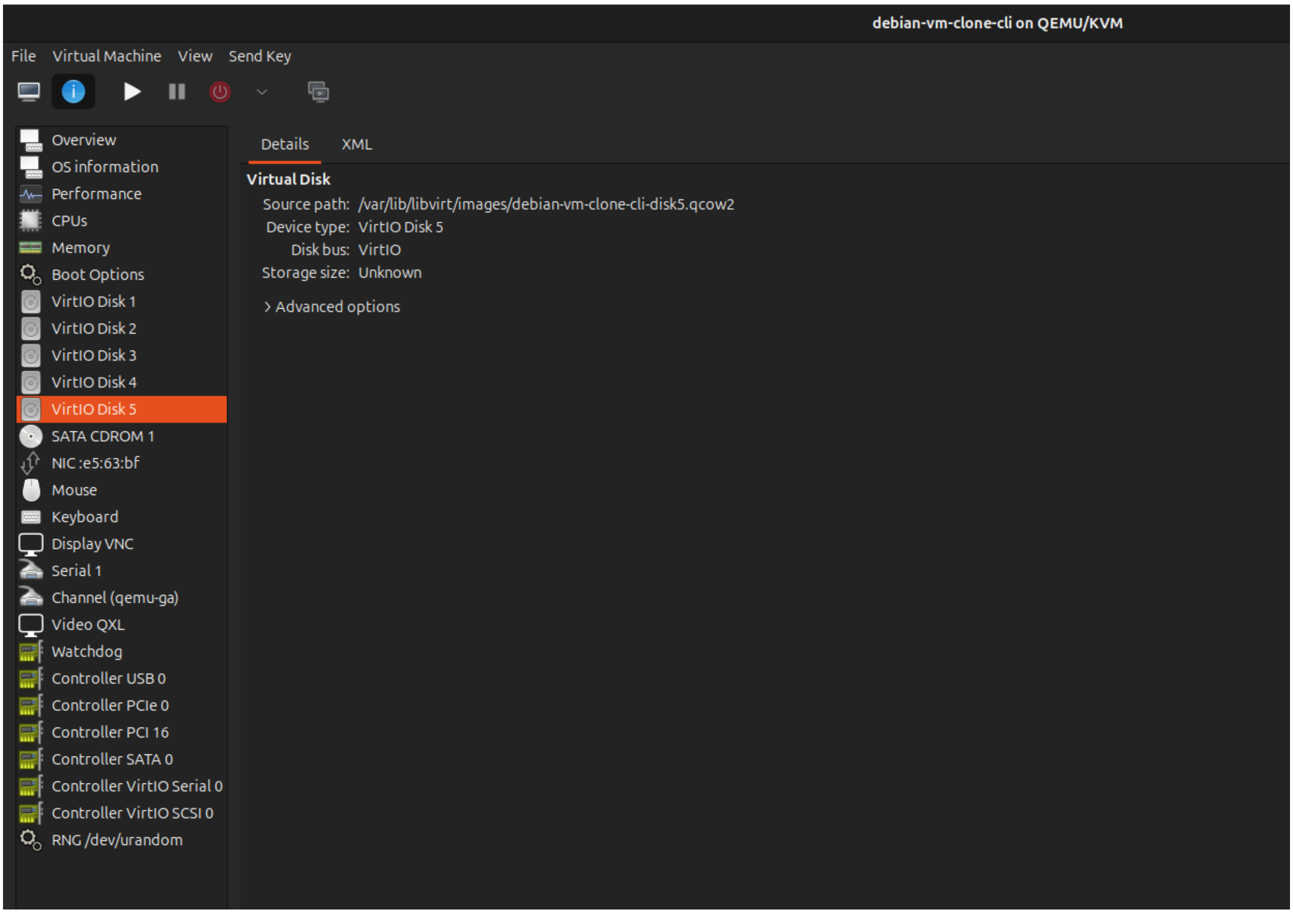
Create the disk:
`sudo qemu-img create -f qcow2
/var/lib/libvirt/images/debian-vm-clone-cli-disk5.qcow2 20G`

Attach the disk to my ‘debian-vm-clone-cli’ VM:

`sudo virsh attach-disk debian-vm-clone-cli
/var/lib/libvirt/images/debian-vm-clone-cli-disk5.qcow2 vde --targetbus virtio
--persistent`



```
istiak@islam-21301218:~$ sudo qemu-img create -f qcow2 /var/lib/libvirt/images/debia
n-vm-clone-cli-disk5.qcow2 20G
Formatting '/var/lib/libvirt/images/debian-vm-clone-cli-disk5.qcow2', fmt=qcow2 clus
ter_size=65536 extended_l2=off compression_type=zlib size=21474836480 lazy_refcounts
=off refcount_bits=16
istiak@islam-21301218:~$ sudo virsh attach-disk debian-vm-clone-cli /var/lib/libvirt
/images/debian-vm-clone-cli-disk5.qcow2 vde --targetbus virtio --persistent
Disk attached successfully
```

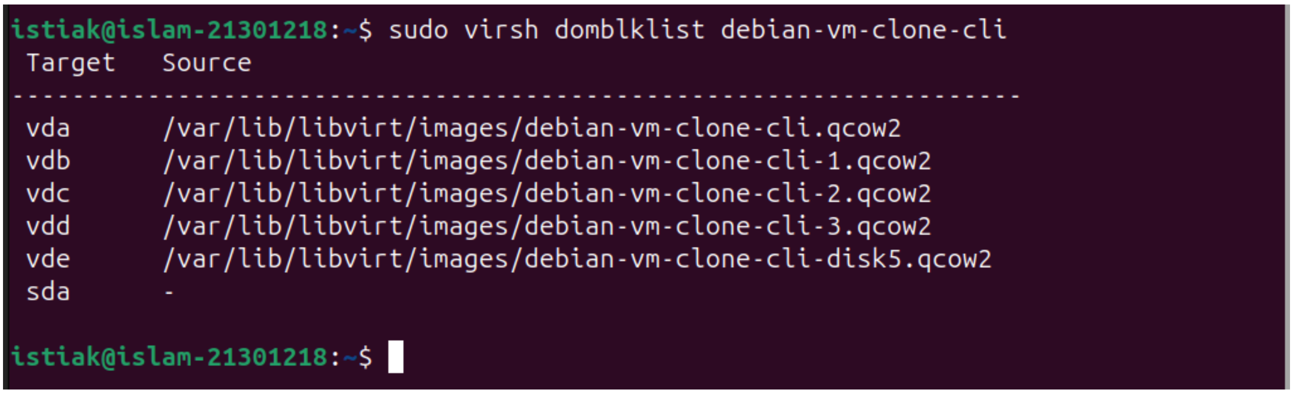


Virtual Disk

Source path: /var/lib/libvirt/images/debian-vm-clone-cli-disk5.qcow2
Device type: VirtIO Disk 5
Disk bus: VirtIO
Storage size: Unknown

Also we can see our available disk by running this code:

`sudo virsh domblklist debian-vm`

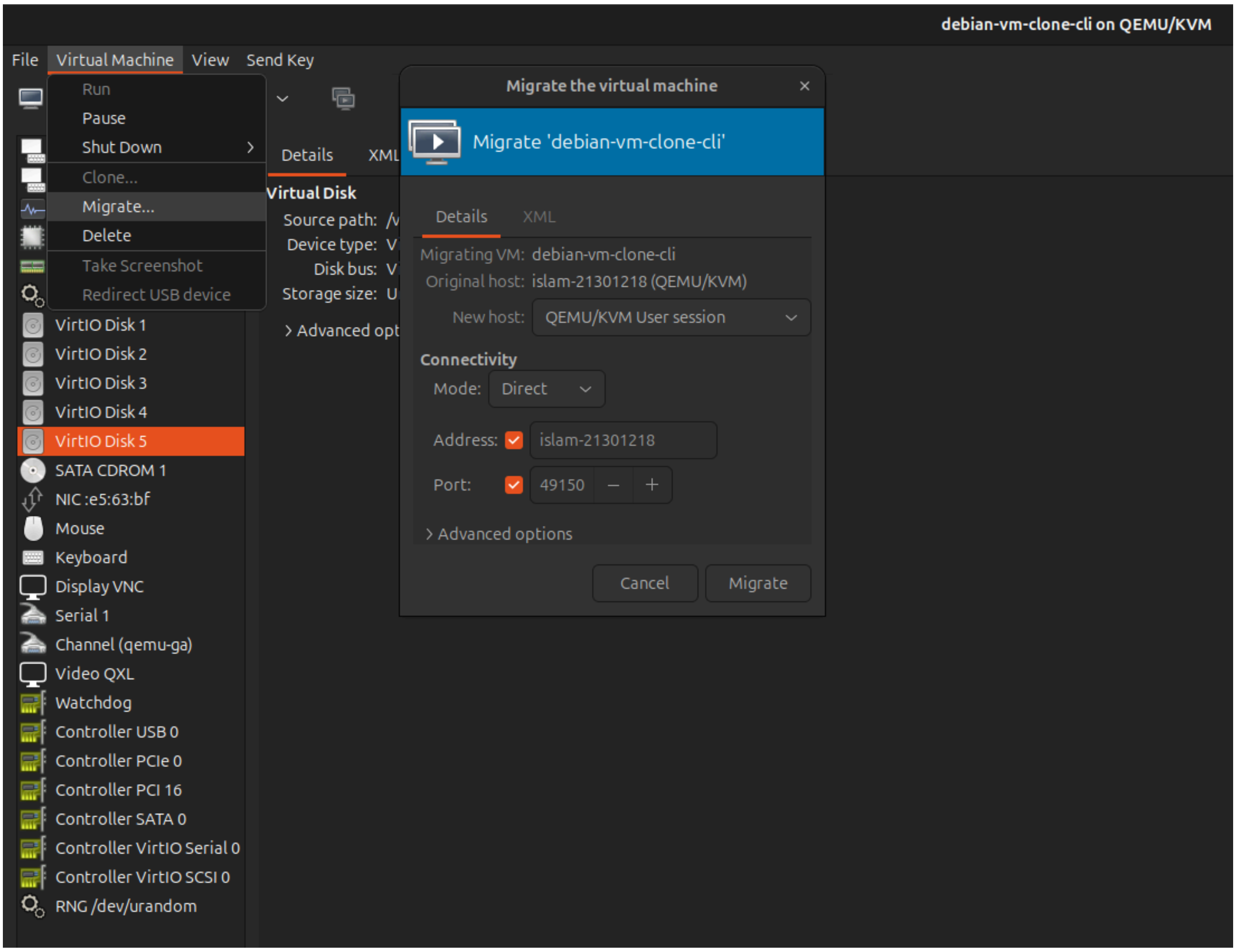


```
istiak@islam-21301218:~$ sudo virsh domblklist debian-vm-clone-cli
 Target   Source
-------------------------------------------------------------------
 vda      /var/lib/libvirt/images/debian-vm-clone-cli.qcow2
 vdb      /var/lib/libvirt/images/debian-vm-clone-cli-1.qcow2
 vdc      /var/lib/libvirt/images/debian-vm-clone-cli-2.qcow2
 vdd      /var/lib/libvirt/images/debian-vm-clone-cli-3.qcow2
 vde      /var/lib/libvirt/images/debian-vm-clone-cli-disk5.qcow2
 sda      -

istiak@islam-21301218:~$ 
```

Here we can see previously we have created up to 4 disk using GUI. And the last one was created from CLI.
We can add more drive using the same code just we need to change the name of the ‘disk’

## Migrate a VM to another host

Using the GUI,



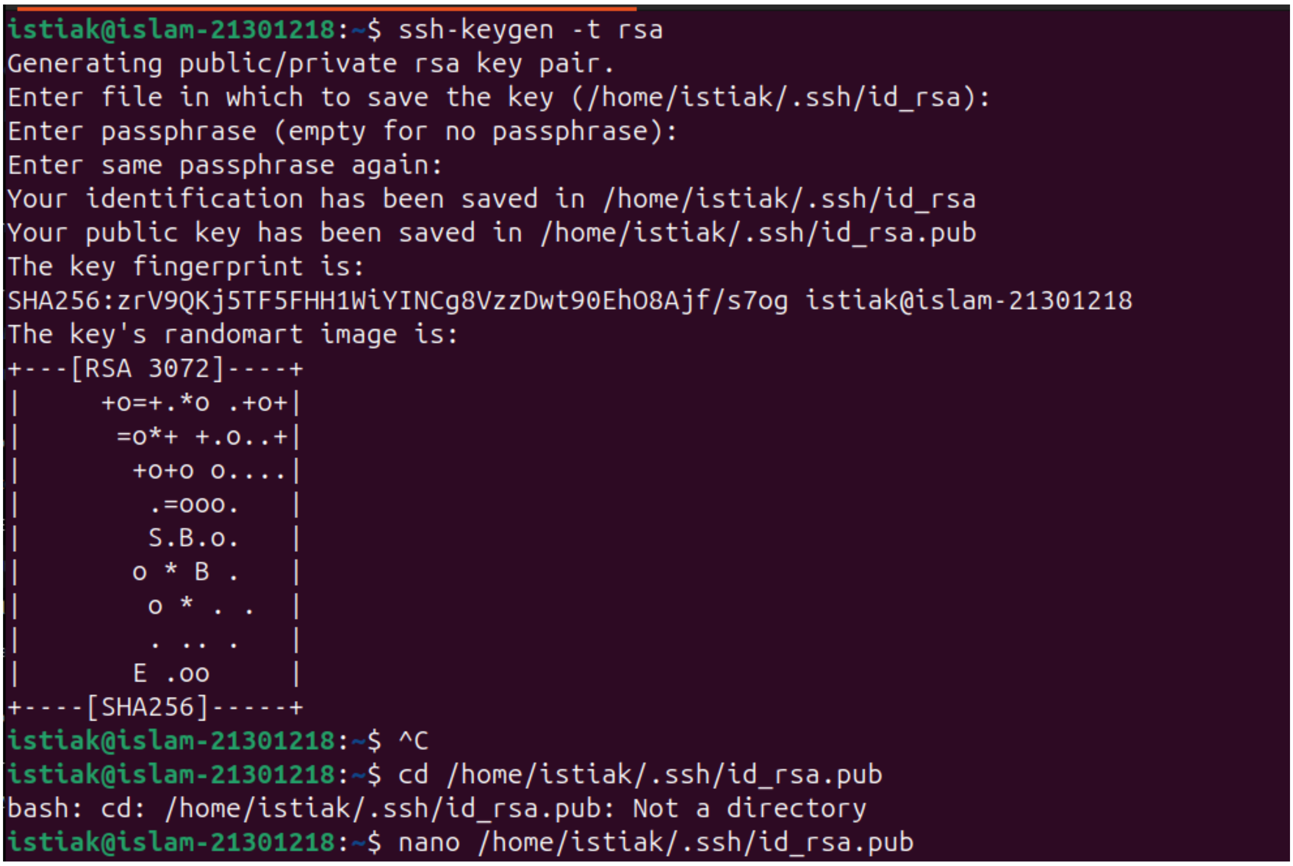
Migrating VM: debian-vm-clone-cli
Original host: islam-21301218 (QEMU/KVM)
New host: QEMU/KVM User session

Connectivity
Mode: Direct
Address: ☑ islam-21301218
Port: ☑ 49150

First we need to generate SSH key for both system. To do that run this code,

`ssh-keygen -t rsa`

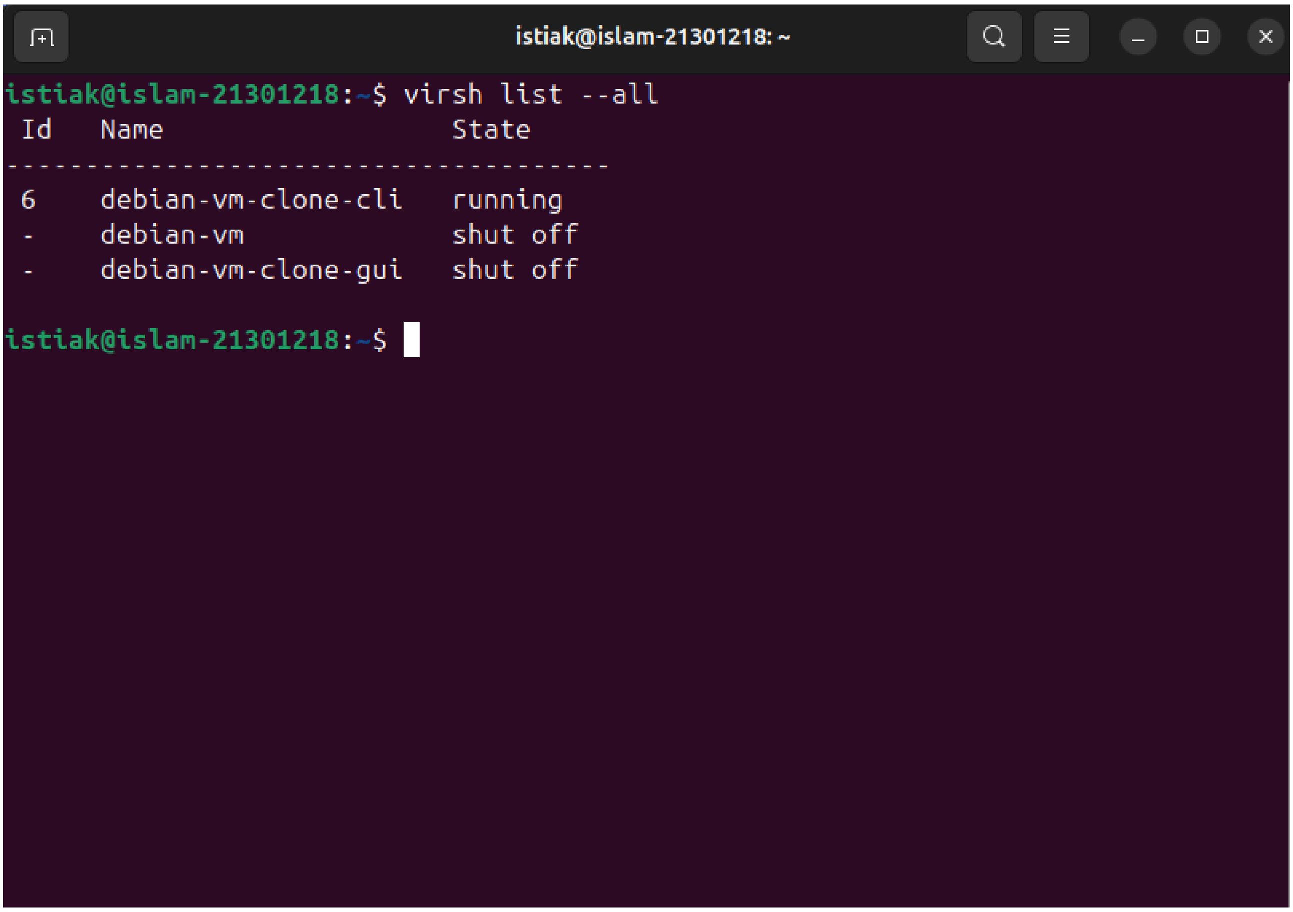
The public key is stored in /home/istiak/.ssh/id_rsa.pub directory. We need to copy the code using nano.



```
istiak@islam-21301218:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/istiak/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/istiak/.ssh/id_rsa
Your public key has been saved in /home/istiak/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:zrV9QKj5TF5FHH1WiYINCg8VzzDwt90EhO8Ajf/s7og istiak@islam-21301218
The key's randomart image is:
+---[RSA 3072]----+
|     +o=+.*o .+o+|
|      =o*+ +.o..+|
|       +o+o o....|
|        .=ooo.   |
|        S.B.o.   |
|       o * B .   |
|        o * . .  |
|        . .. .   |
|       E .oo     |
+----[SHA256]-----+
istiak@islam-21301218:~$ ^C
istiak@islam-21301218:~$ cd /home/istiak/.ssh/id_rsa.pub
bash: cd: /home/istiak/.ssh/id_rsa.pub: Not a directory
istiak@islam-21301218:~$ nano /home/istiak/.ssh/id_rsa.pub
```

For finding the destination host we need to run this code. For my case I am using my friends host machine. 'ahmad@habibullah-21301236' is the target host.

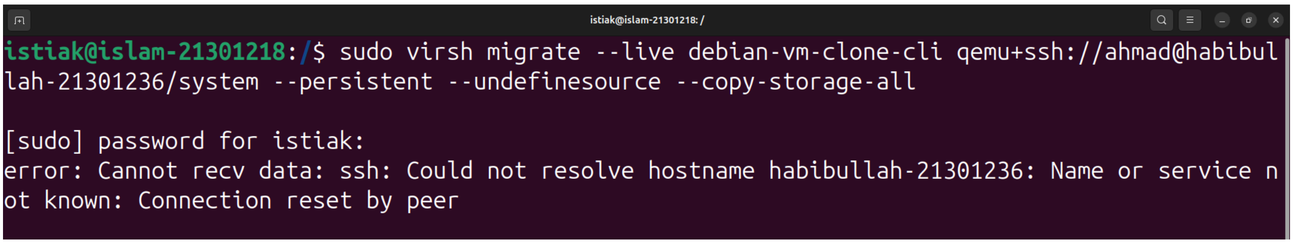
check all the VM's in my machine:



```
istiak@islam-21301218:~$ virsh list --all
 Id   Name                   State
-------------------------------------
 6    debian-vm-clone-cli    running
 -    debian-vm              shut off
 -    debian-vm-clone-gui    shut off

istiak@islam-21301218:~$ 
```

we can migrate via virsh migrate command.
`sudo virsh migrate --live debian-vm-clone-cli
qemu+ssh://ahmad@habibullah-21301236/system --persistent --undefinesource
–copy-storage-all`

Actually we need to transfer the ‘debian-vm.qcow’ file to another host.
Currently my machine is not connecting



```
istiak@islam-21301218:/$ sudo virsh migrate --live debian-vm-clone-cli qemu+ssh://ahmad@habibul
lah-21301236/system --persistent --undefinesource --copy-storage-all

[sudo] password for istiak:
error: Cannot recv data: ssh: Could not resolve hostname habibullah-21301236: Name or service n
ot known: Connection reset by peer
```