Assignment 02

Abir Ahammed Bhuiyan
ID: 20101197
Dept. of Computer
Science and Engineering

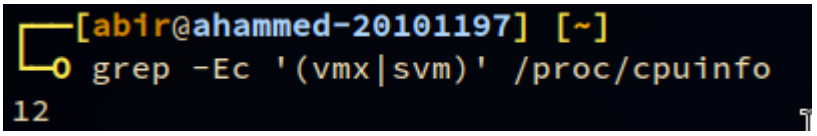
Section : 01
Subject : CSE484
Semester : Spring-2024
Date : 15.02.2024

1. Install KVM

Check virtualization is enabled or not

```
grep -Ec '(vmx|svm)' /proc/cpuinfo
```

If the output of this command is more than 0 that means virtualization is enabled in our computer.



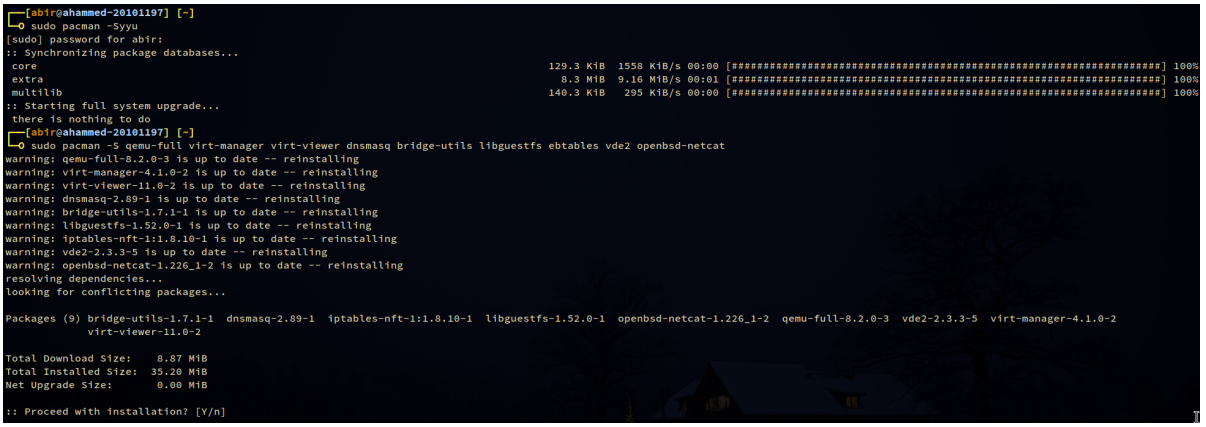
Install the Required KVM Packages

Updating the existing packages in the `Arch Linux` system.

```
sudo pacman -Syyu
```

By running the below command will install the required packages to run `KVM` in our system.

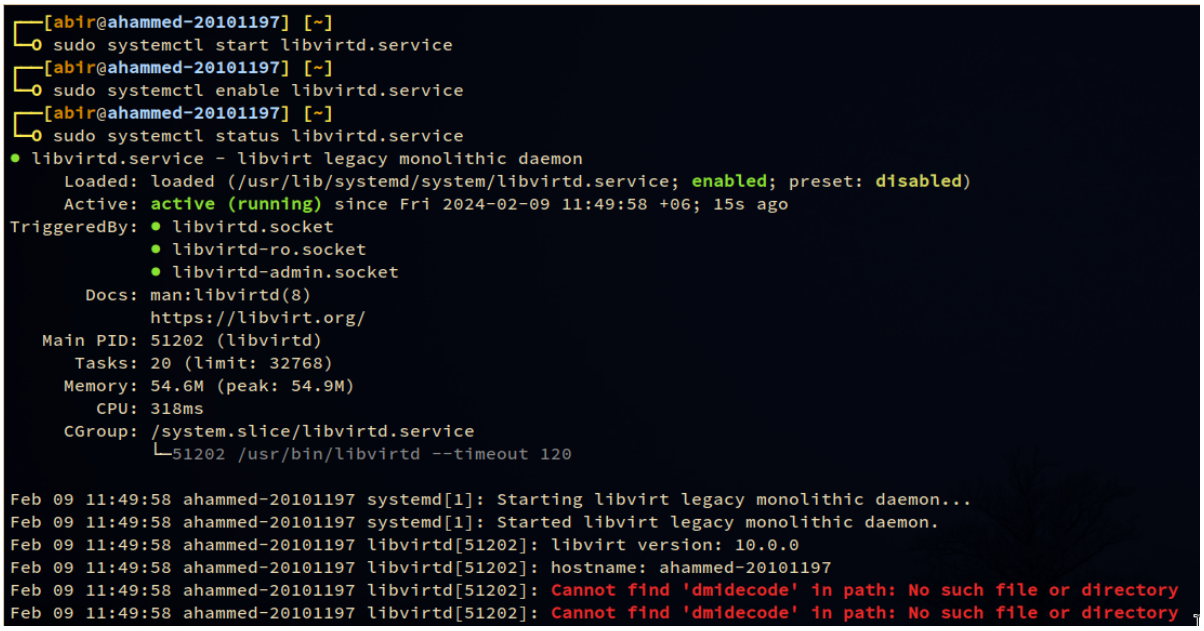
```
sudo pacman -S qemu-full virt-manager virt-viewer dnsmasq bridge-utils libguestfs
ebtables vde2 openbsd-netcat
```



Configure the libvirtd Service

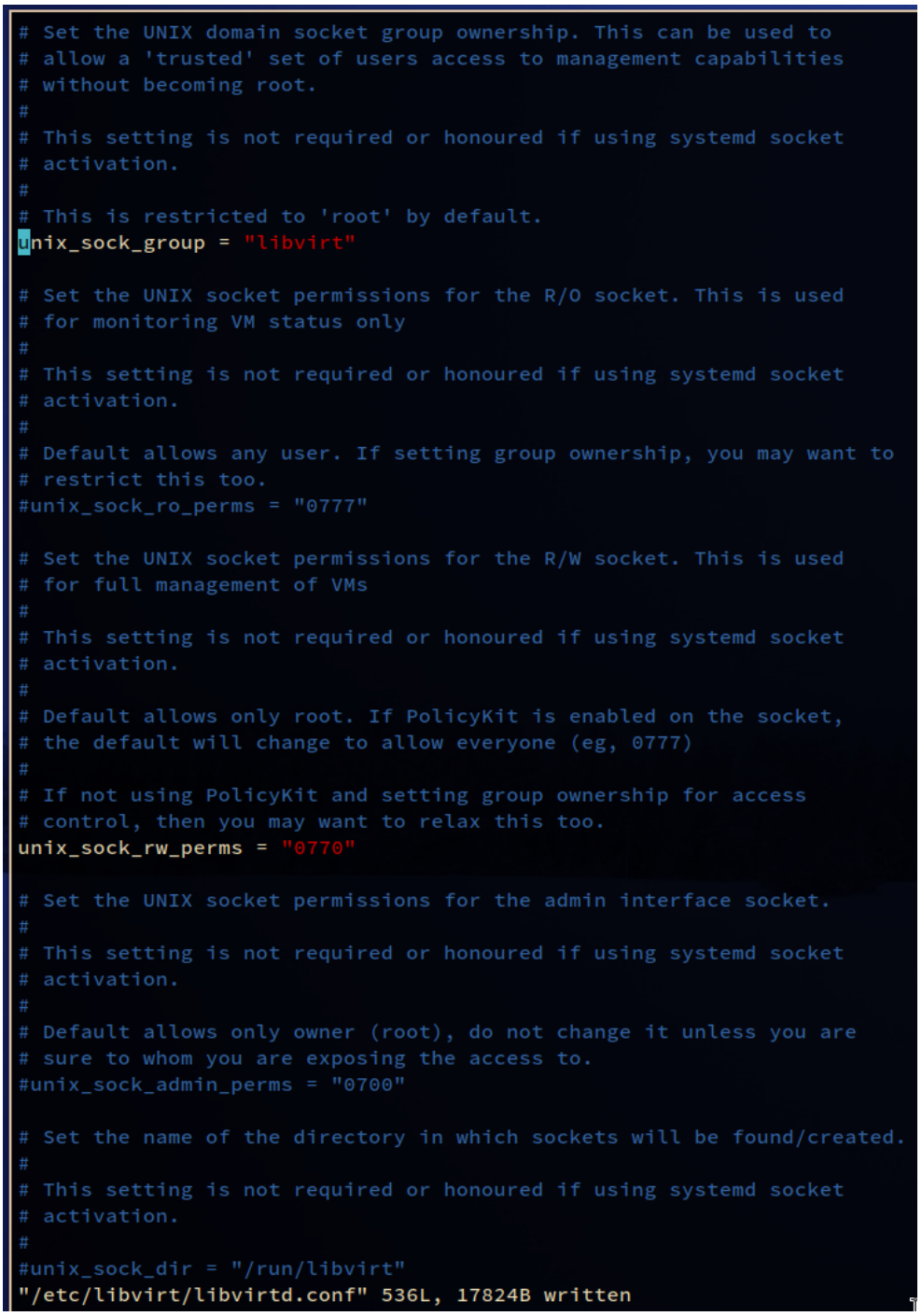
Start and enable `libvirtd` service with the below commands and checking status

```
sudo systemctl start libvirtd.service
sudo systemctl enable libvirtd.service
sudo systemctl status libvirtd.service
```



We need to make some changes to the `libvirtd` configuration file located at `/etc/libvirt/libvirtd.conf`. After opening the file we have to uncomment (i.e. removing `#` characters) from matching lines below and save the file.

```
unix_sock_group = "libvirt"
unix_sock_rw_perms = "0770"
```



Next we have to add our current user to the libvirt group.

```
sudo usermod -aG libvirt $USER
```

Restart the `libvirtd` service.

```
sudo systemctl restart libvirtd.service
```

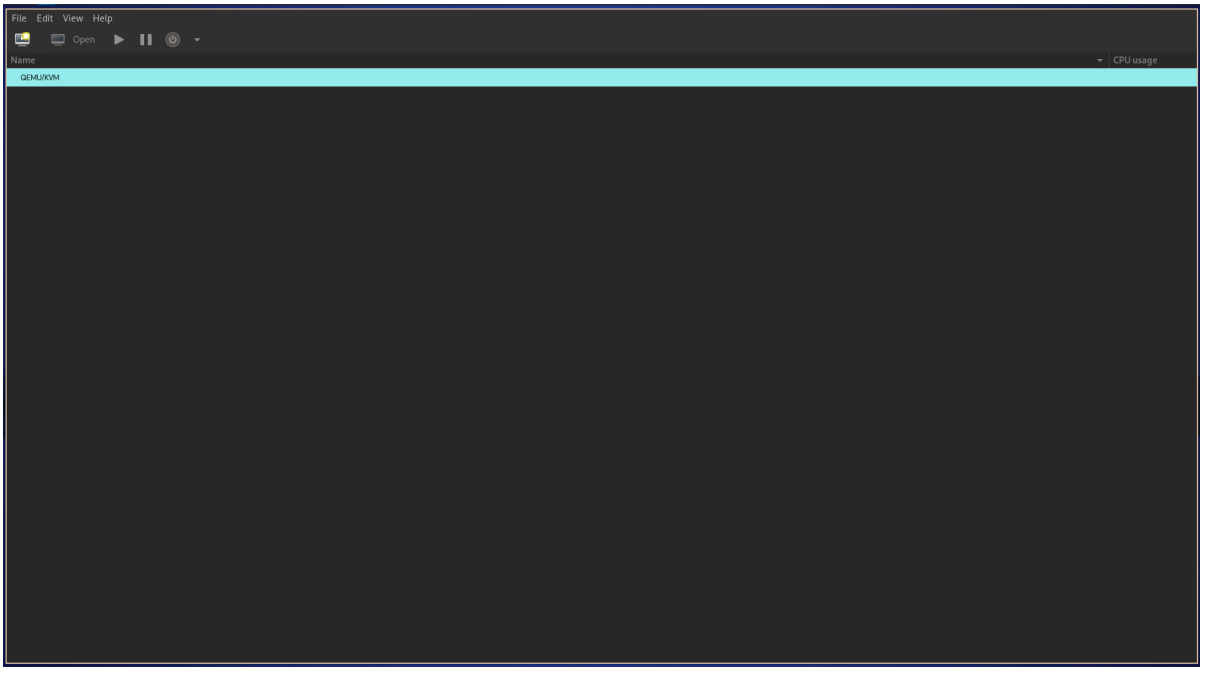
```
[abir@ahammed-20101197] [~]
$ sudo usermod -aG libvirt $USER
[abir@ahammed-20101197] [~]
$ sudo systemctl restart libvirtd.service
[abir@ahammed-20101197] [~]
$ sudo systemctl status libvirtd.service
● libvirtd.service - libvirt legacy monolithic daemon
     Loaded: loaded (/usr/lib/systemd/system/libvirtd.service; enabled; preset: disabled)
     Active: active (running) since Fri 2024-02-09 11:57:38 +06; 10s ago
TriggeredBy: ● libvirtd.socket
             ● libvirtd-ro.socket
             ● libvirtd-admin.socket
       Docs: man:libvirtd(8)
             https://libvirt.org/
   Main PID: 61546 (libvirtd)
      Tasks: 20 (limit: 32768)
     Memory: 48.7M (peak: 49.1M)
        CPU: 221ms
     CGroup: /system.slice/libvirtd.service
             └─61546 /usr/bin/libvirtd --timeout 120

Feb 09 11:57:38 ahammed-20101197 systemd[1]: Starting libvirt legacy monolithic daemon...
Feb 09 11:57:38 ahammed-20101197 systemd[1]: Started libvirt legacy monolithic daemon.
Feb 09 11:57:38 ahammed-20101197 libvirtd[61546]: libvirt version: 10.0.0
Feb 09 11:57:38 ahammed-20101197 libvirtd[61546]: hostname: ahammed-20101197
Feb 09 11:57:38 ahammed-20101197 libvirtd[61546]: Cannot find 'dmidecode' in path: No such file or directory
Feb 09 11:57:38 ahammed-20101197 libvirtd[61546]: Cannot find 'dmidecode' in path: No such file or directory
[abir@ahammed-20101197] [~]
$
```

That is how the `KVM` has been installed in our system.

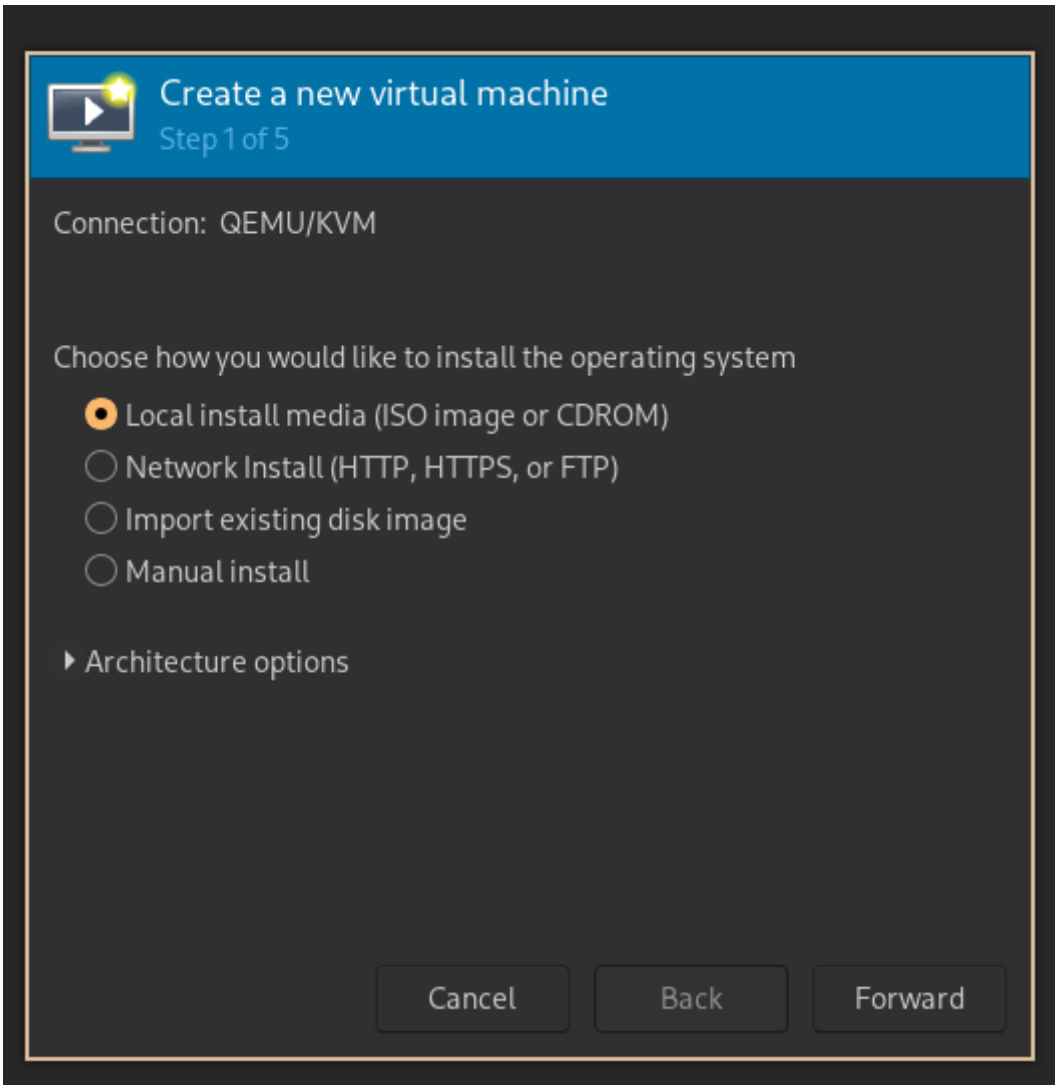
2. Create a VM using VMM (virtual machine manager) i.e. using GUI.

open `virt-manager`.

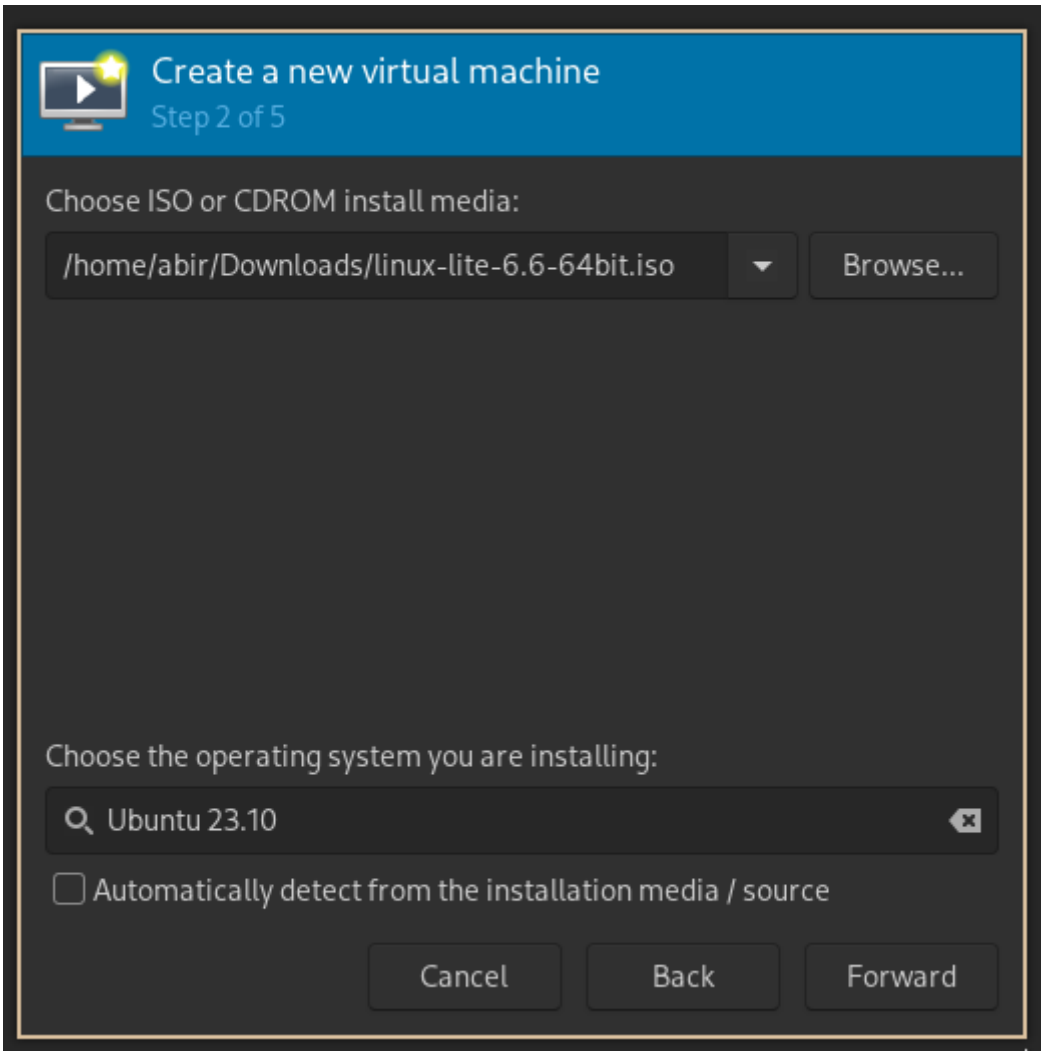


Click on `create a new virtual machine` button on the upper left corner. It will pop up a new window.

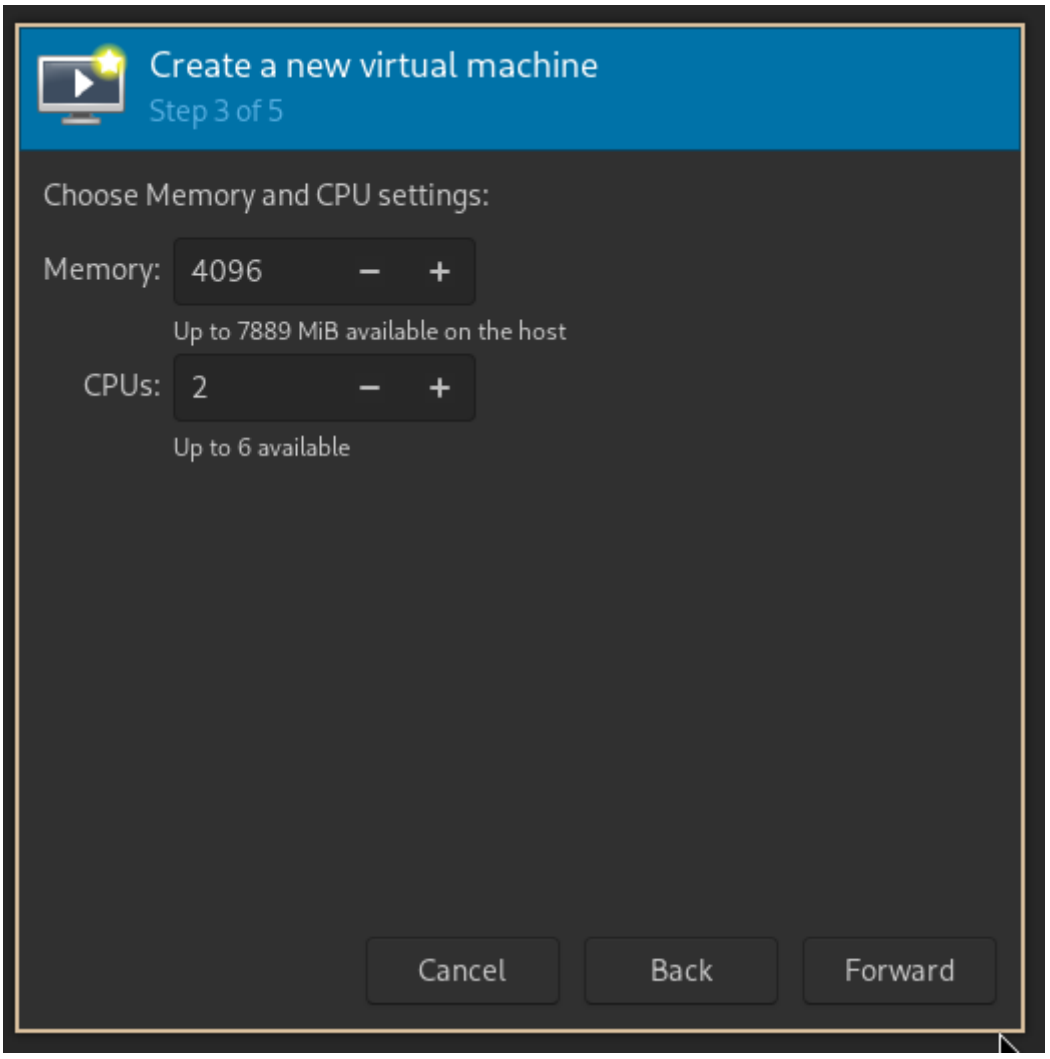
Click on the `Local install media (ISO image or CDROM)` and click forward.



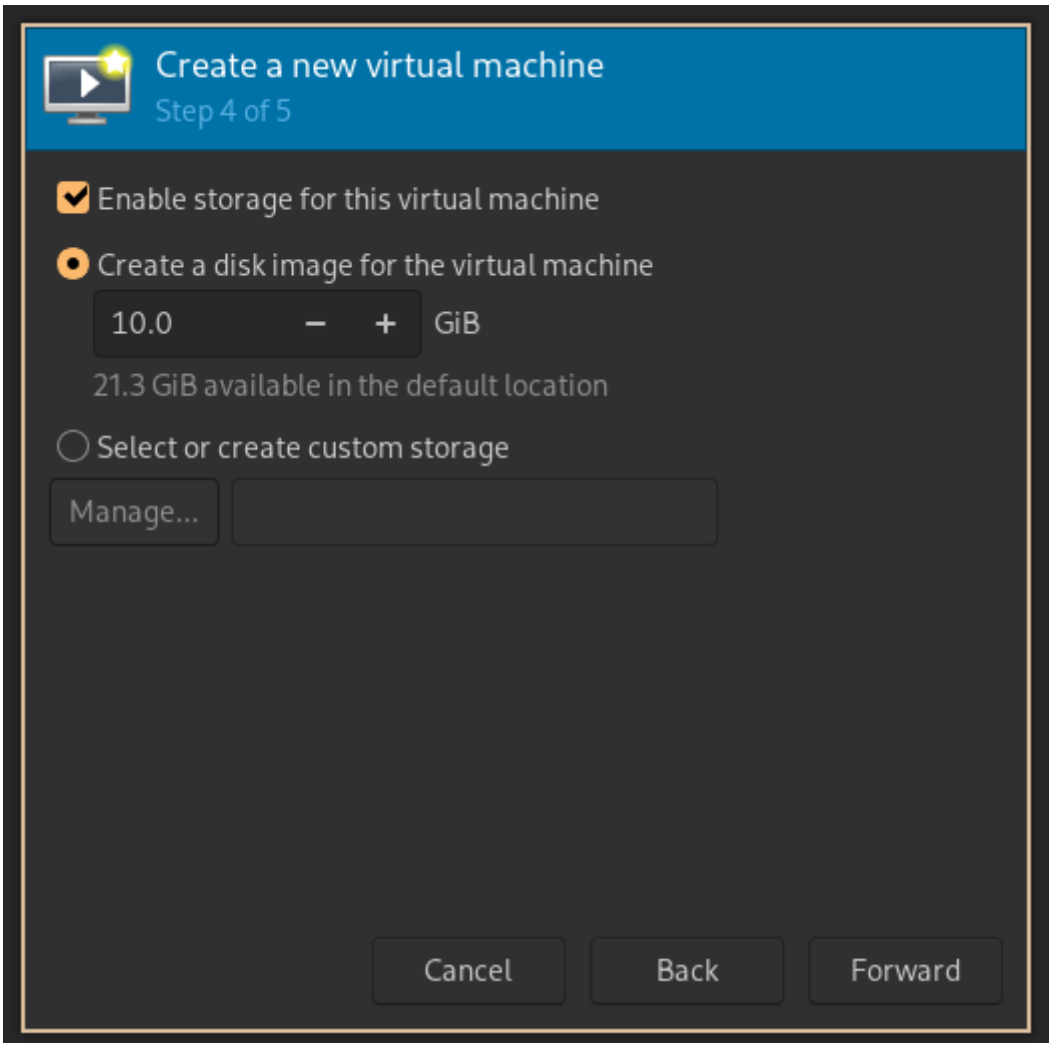
Here we are tyring to install `linux-lite` that is based on `Ubuntu 23.10` hence we have selected the options like below. Click forward.



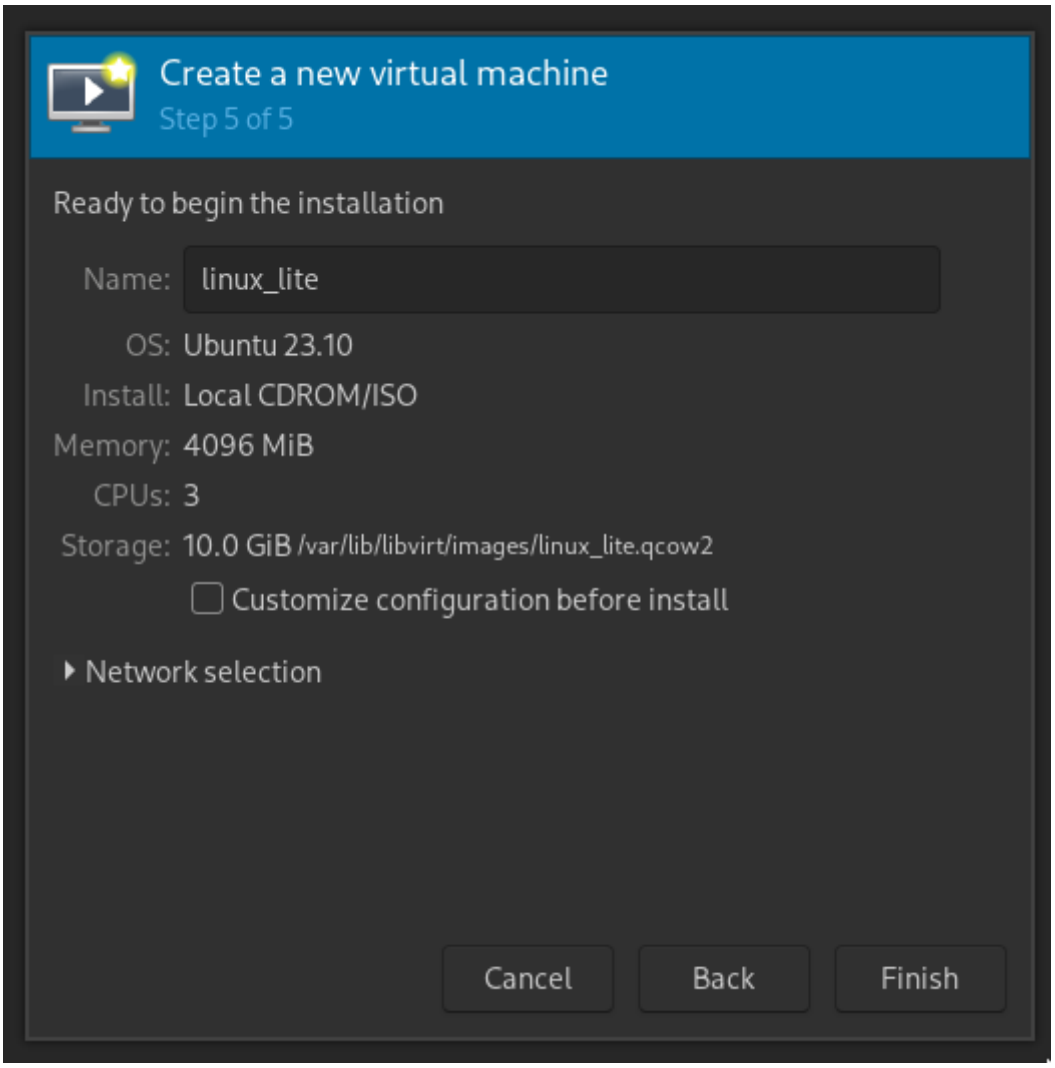
Here we have to allocated memory and CPU. For our case we have allocated 4GB of RAM along with 2 CPU cores. Click forward.



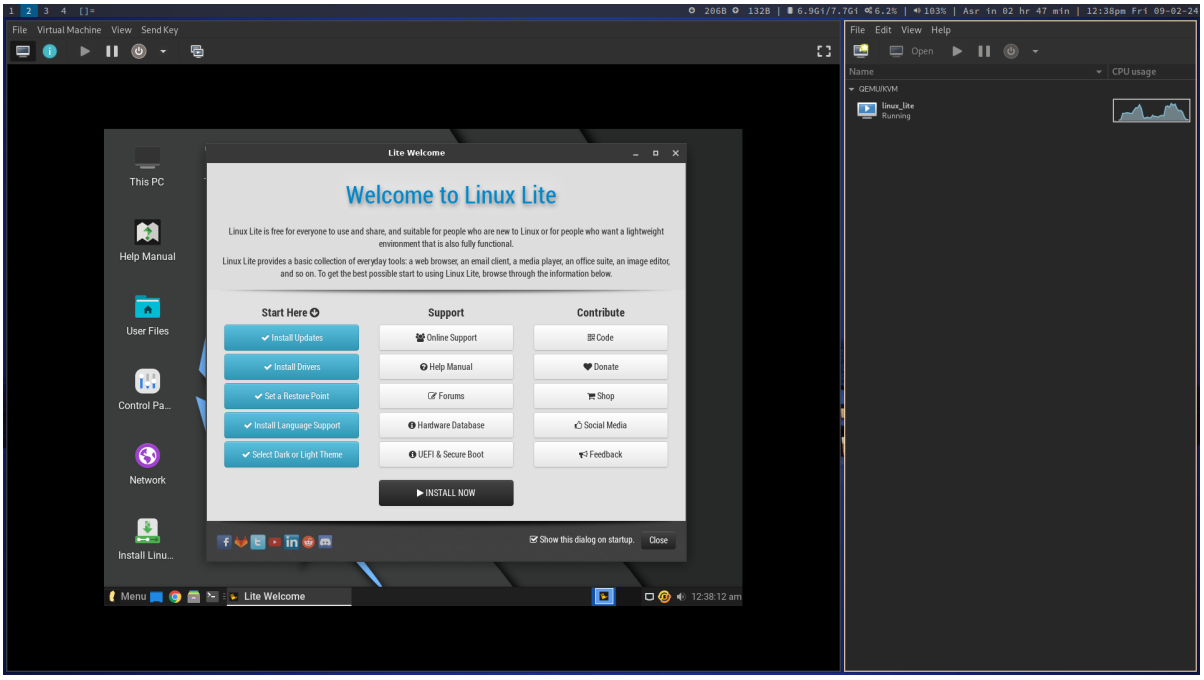
Here we have to specify storage size for the VHD (Virtual Hard Disk). We have allocated 10GB of space for the guest OS.



At last we have to give a name (in our case `linux_lite`) and click on finish.



Wow, we have successfully created a VM and it is runnig `Linux Lite`.

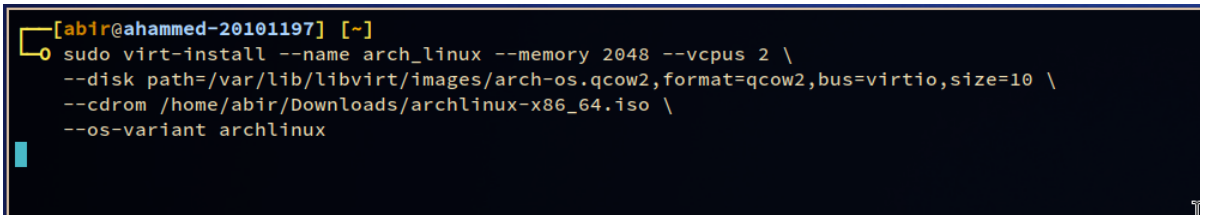


After running the VM just install the OS.

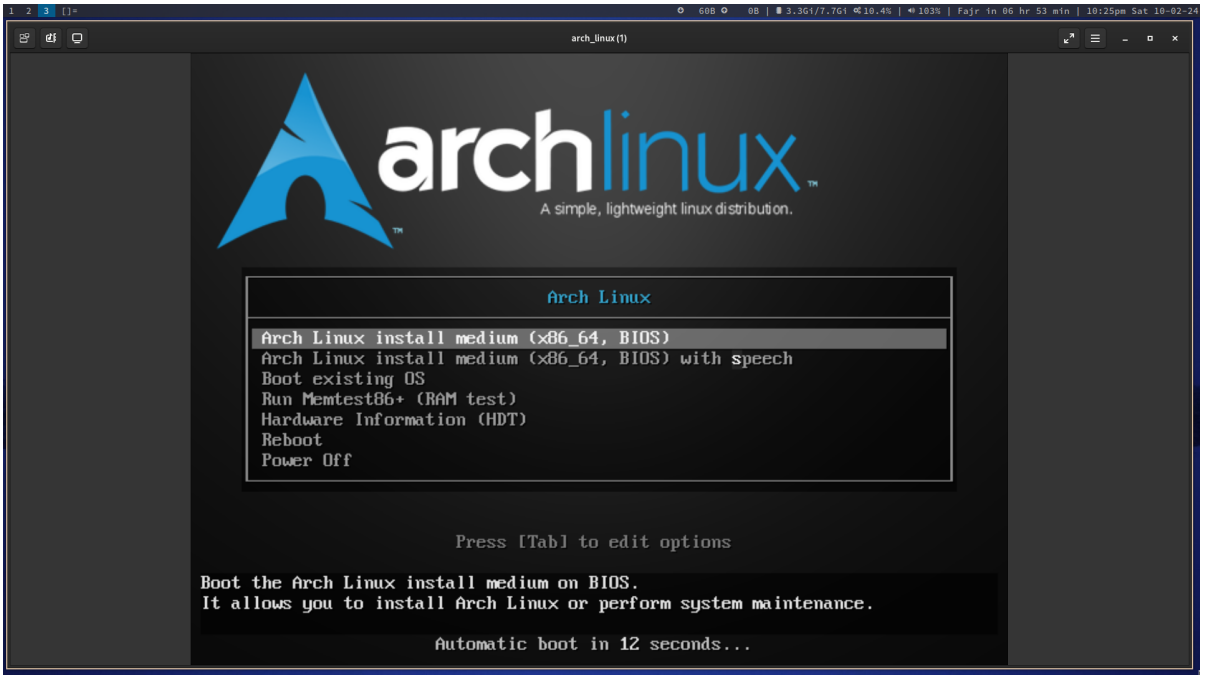
3. Create a kvm-based VM using `virt-install` cli.

For this we will create a VM for `Arch Linux`. First, we have to fire `virt-install` command like below.

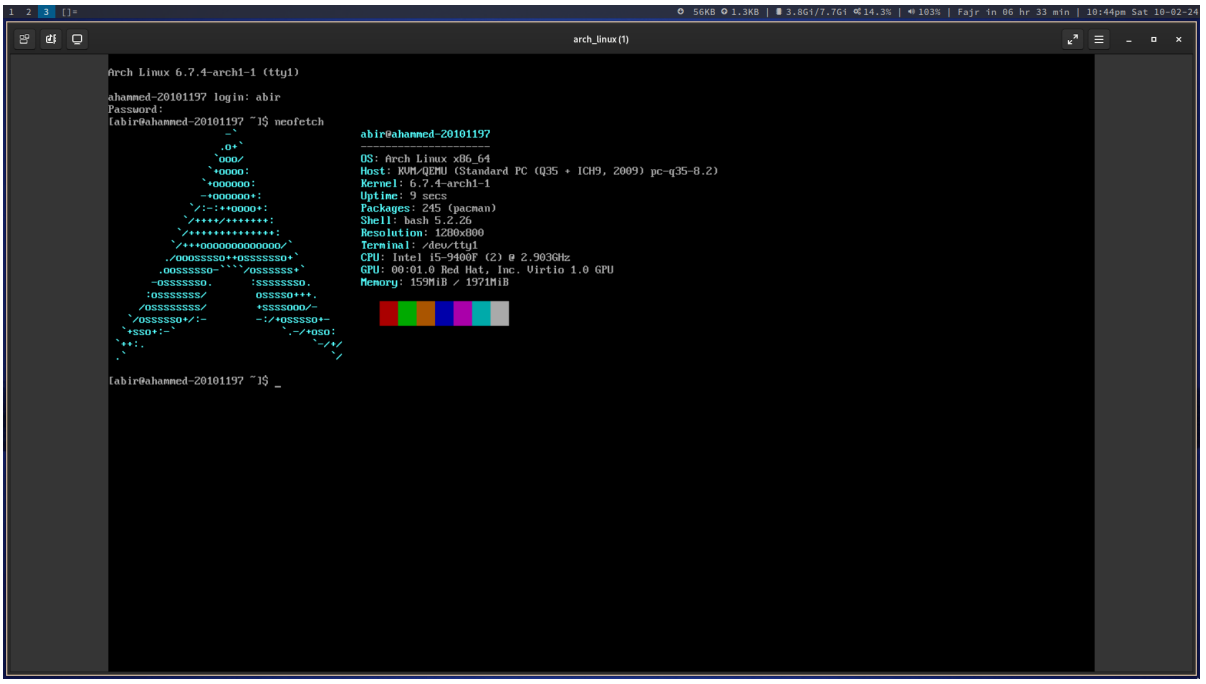
```
sudo virt-install --name arch_linux --memory 2048 --vcpus 2 \
    --disk path=/var/lib/libvirt/images/arch-
os.qcow2,format=qcow2,bus=virtio,size=10 \
    --cdrom /home/abir/Downloads/archlinux-x86_64.iso \
    --os-variant archlinux
```



After firing that command we will see a new window like below.

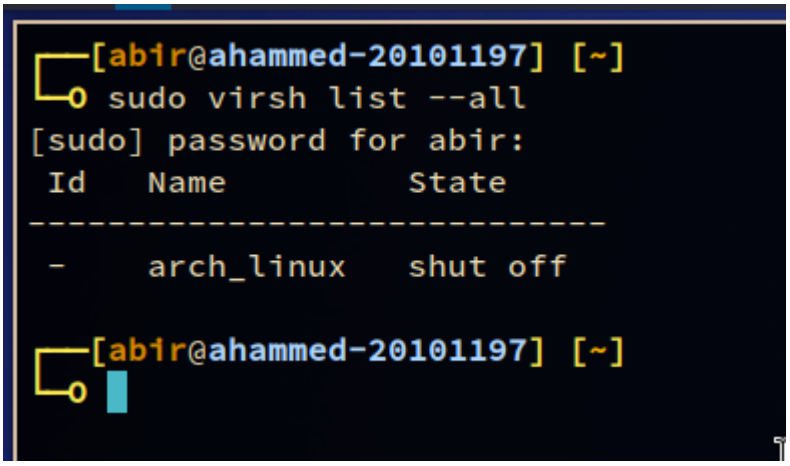


Install `Arch Linux` using `archinstall` command.



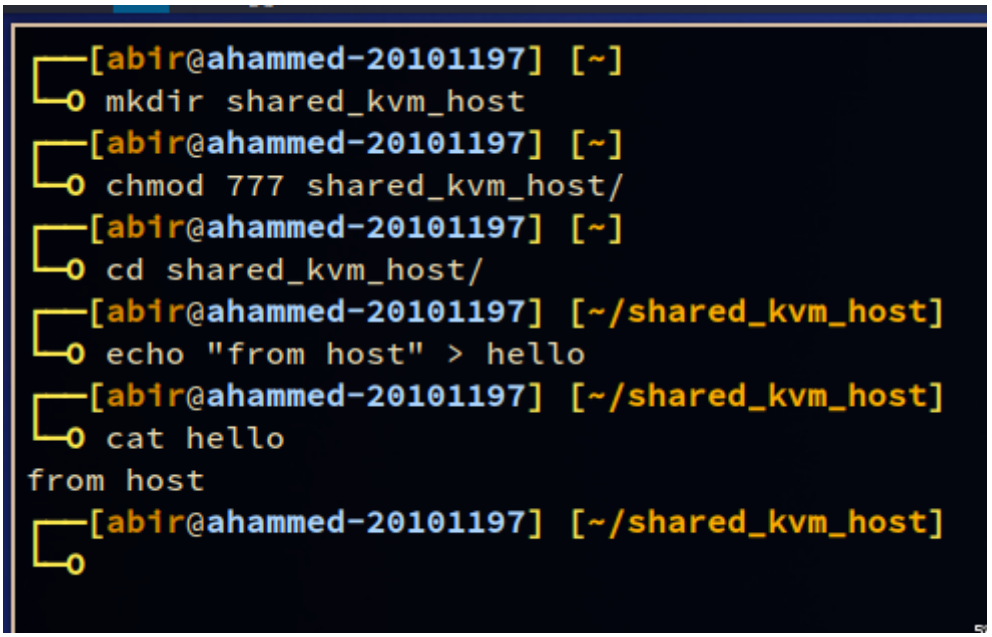
Using the below command we can see that our desired VM has been created successfully.

```
sudo virsh list --all
```



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.

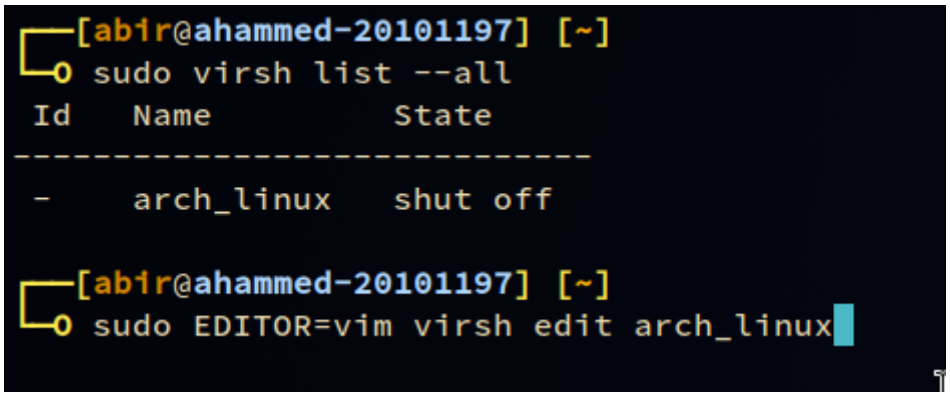
To achieve this we have to first make a directory and change its access permissions like below. Also, create a file to test our procedure.



The below command is essential for creating a shared folder. The command will be used to open our VM's `config` file.

```
sudo EDITOR=vim virsh edit arch_linux
```

For editing we will be using `vim` text editor so we are using `EDITOR=vim`. You can use any editor you like.



After opening the `config` file we have to insert the below lines to activate shared folder feature.

```
<memoryBacking>
    <source type='memfd'/>
    <access mode='shared'/>
</memoryBacking>
```

```
<filesystem type='mount' accessmode='passthrough'>
    <driver type='virtiofs'/>
    <source dir='/home/abir/shared_kvm_host'/>
    <target dir='shared_kvm_host'/>
</filesystem>
```

After saving the `config` file just start the VM like below. `virt-viewer` will open our VM in GUI mode.

```
┌──[abir@ahammed-20101197] [~]
└─$ sudo virsh list --all
 Id   Name         State
-----------------------------
 -    arch_linux   shut off

┌──[abir@ahammed-20101197] [~]
└─$ sudo virsh start arch_linux
Domain 'arch_linux' started

┌──[abir@ahammed-20101197] [~]
└─$ sudo virt-viewer arch_linux
```

Then we have to create a folder inside `guest os` and mount our shared folder using mount command like below.

```
mkdir shared_kvm
```

```
sudo mount -v -t virtiofs shared_kvm_host shared_kvm/
```

```
[abir@ahammed-20101197 ~]$ mkdir shared_kvm
[abir@ahammed-20101197 ~]$ sudo mount -v -t virtiofs shared_kvm_host shared_kvm/
[sudo] password for abir:
mount: shared_kvm_host mounted on /home/abir/shared_kvm.
[abir@ahammed-20101197 ~]$
```

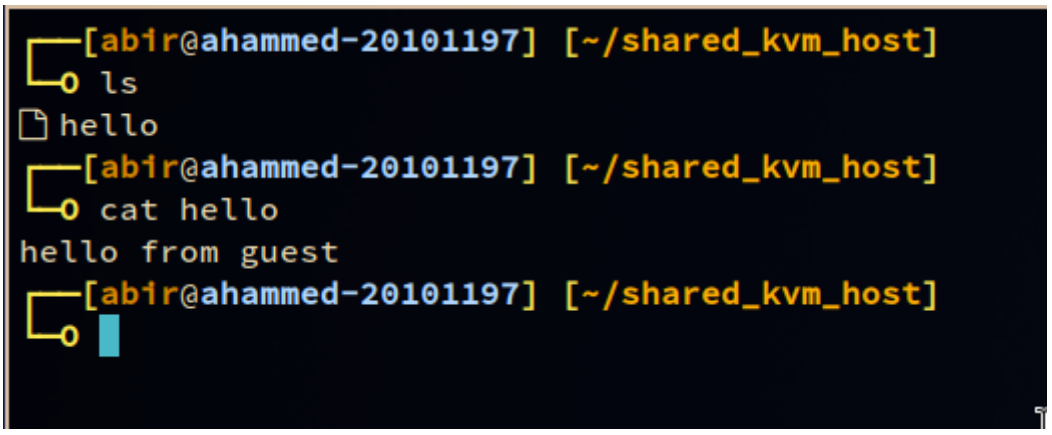
Using the `cat` on the saved file in the shared folder we can verfiy that our shared folder is actually working.

```
[abir@ahammed-20101197 ~]$ cd shared_kvm/
[abir@ahammed-20101197 shared_kvm]$ ls
hello
[abir@ahammed-20101197 shared_kvm]$ cat hello
from host
[abir@ahammed-20101197 shared_kvm]$ _
```

Now change the text file to see if the change can be observed from the host also.

```
[abir@ahammed-20101197 shared_kvm]$ ls -al
total 12
drwxrwxrwx 2 abir abir 4096 Feb 11 00:09 .
drwx------ 6 abir abir 4096 Feb 11 00:17 ..
-rw-r--r-- 1 abir abir   10 Feb 11 00:09 hello
[abir@ahammed-20101197 shared_kvm]$ cat hello
from host
[abir@ahammed-20101197 shared_kvm]$ echo "hello from guest" > hello
[abir@ahammed-20101197 shared_kvm]$ cat hello
hello from guest
[abir@ahammed-20101197 shared_kvm]$
```

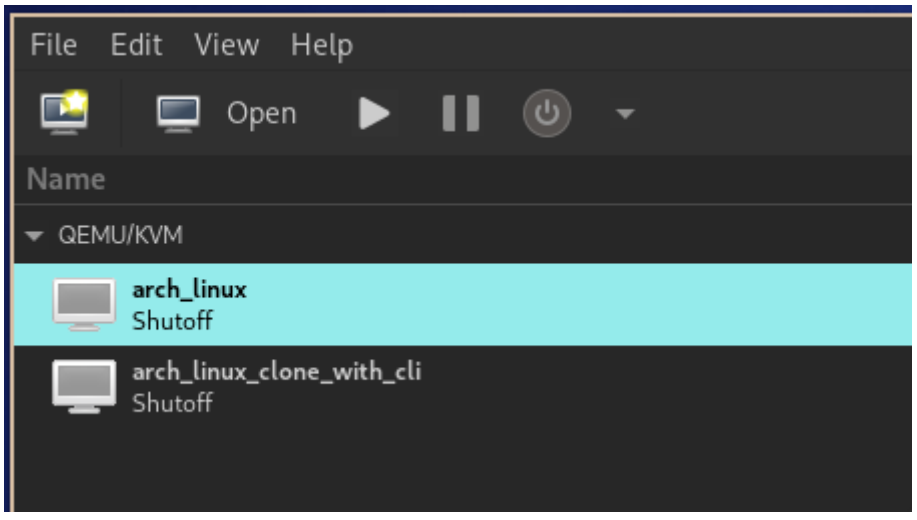
The changes from the `guest os` can also be seen from the `host os`.



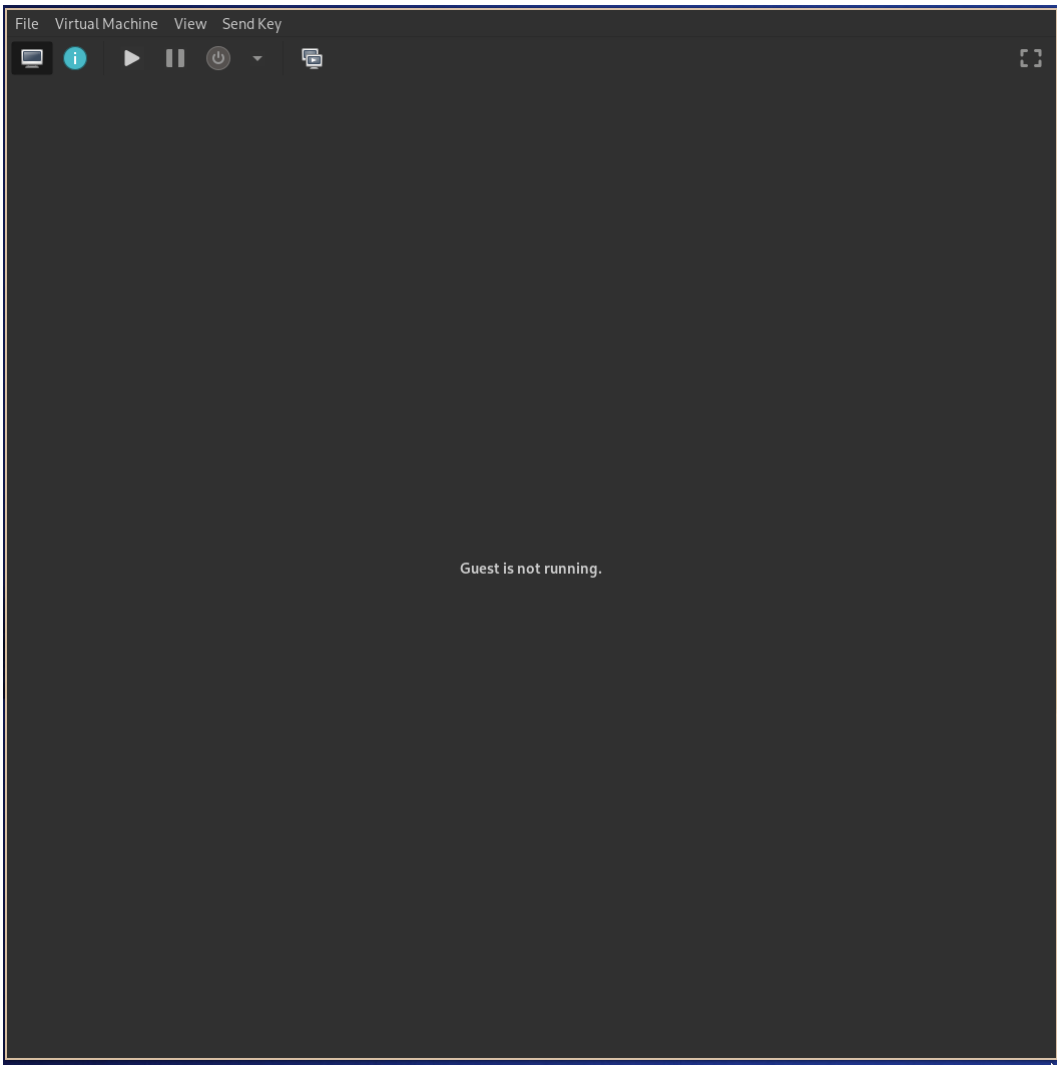
6. Connect the guest OS with a phone and access the phone's files.

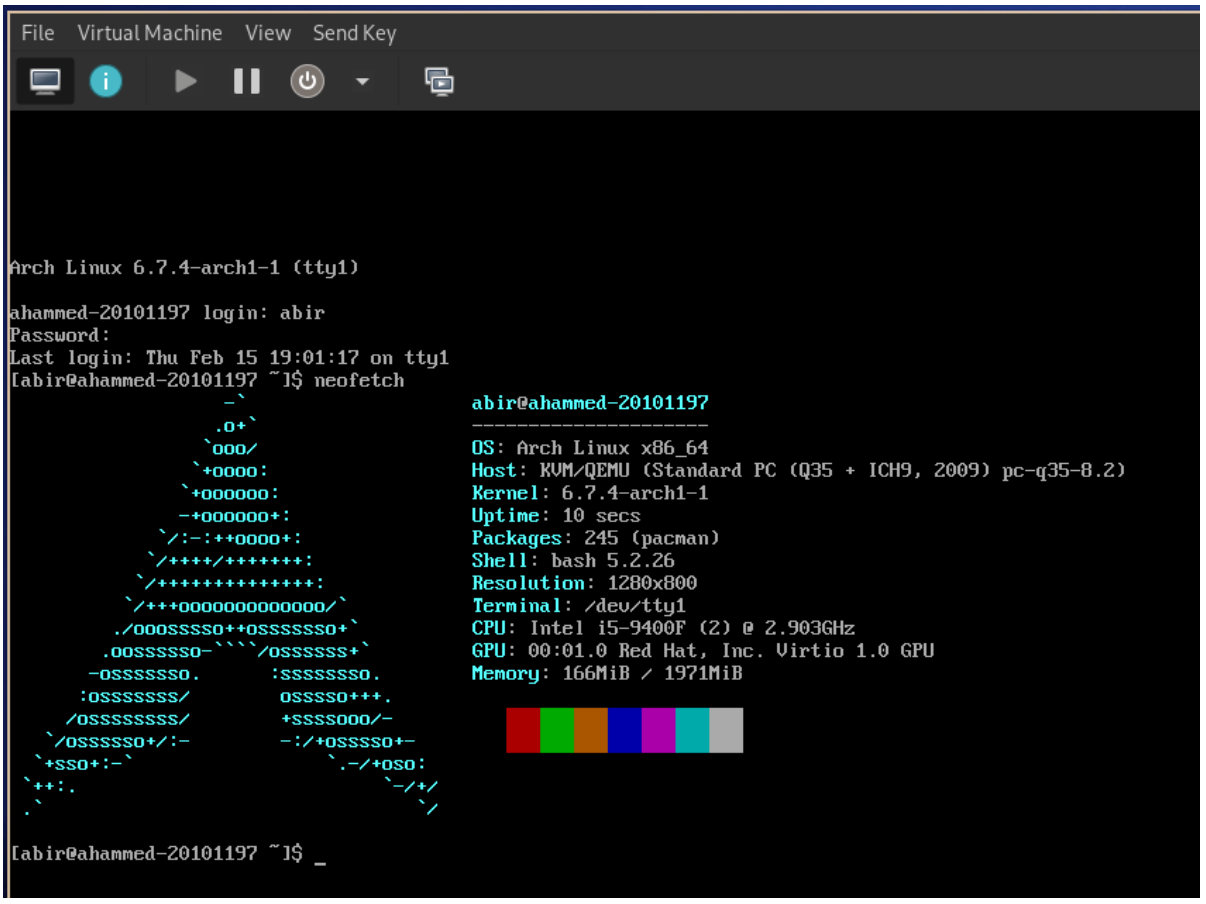
First, connect your phone with the PC using a usb cable. Make sure on the phone file transfer is enabled.

Then in the PC open `virt-manager`.



Start the VM.



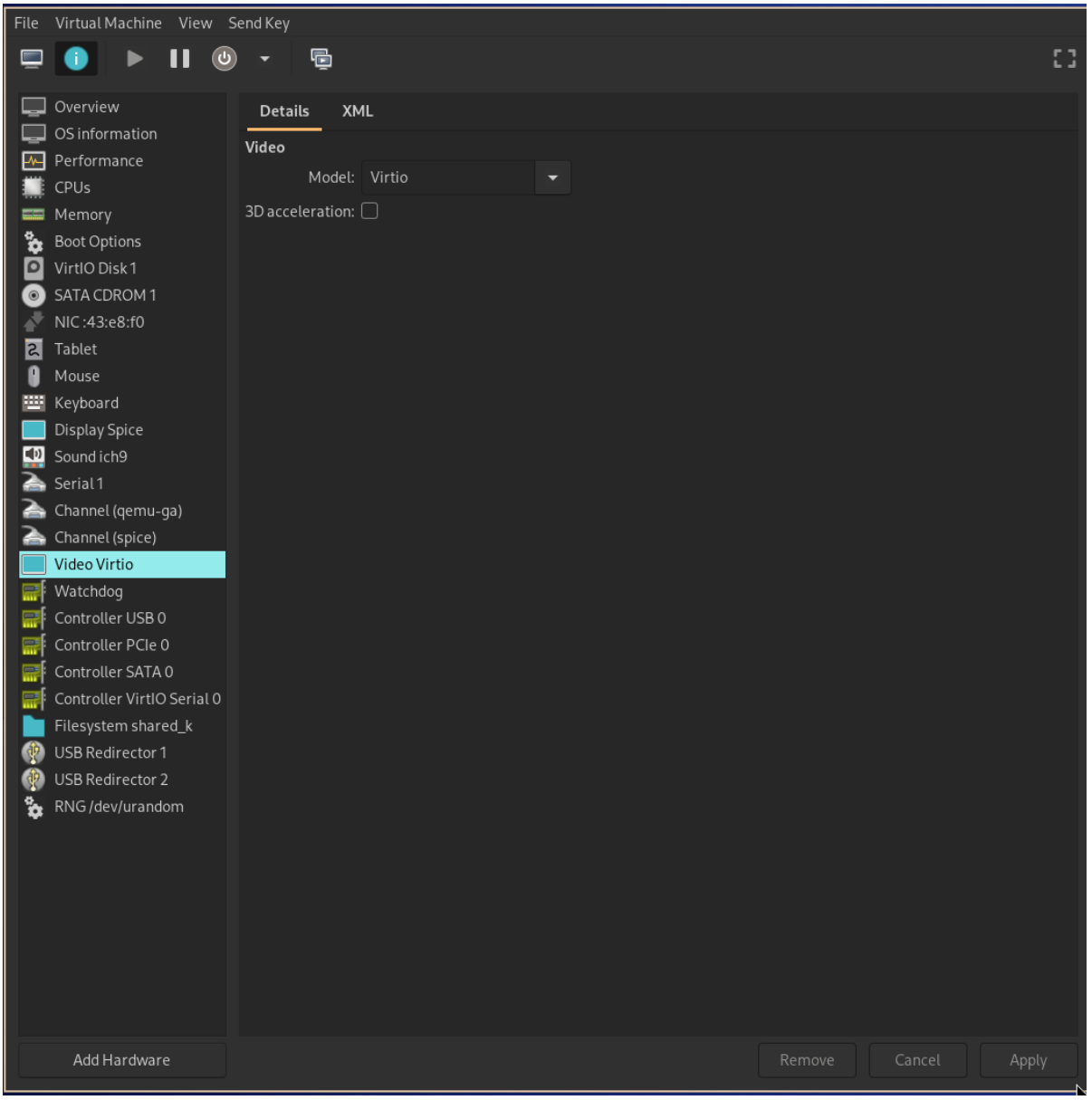


We have to install a package called `android-file-transfer` for this purpose.

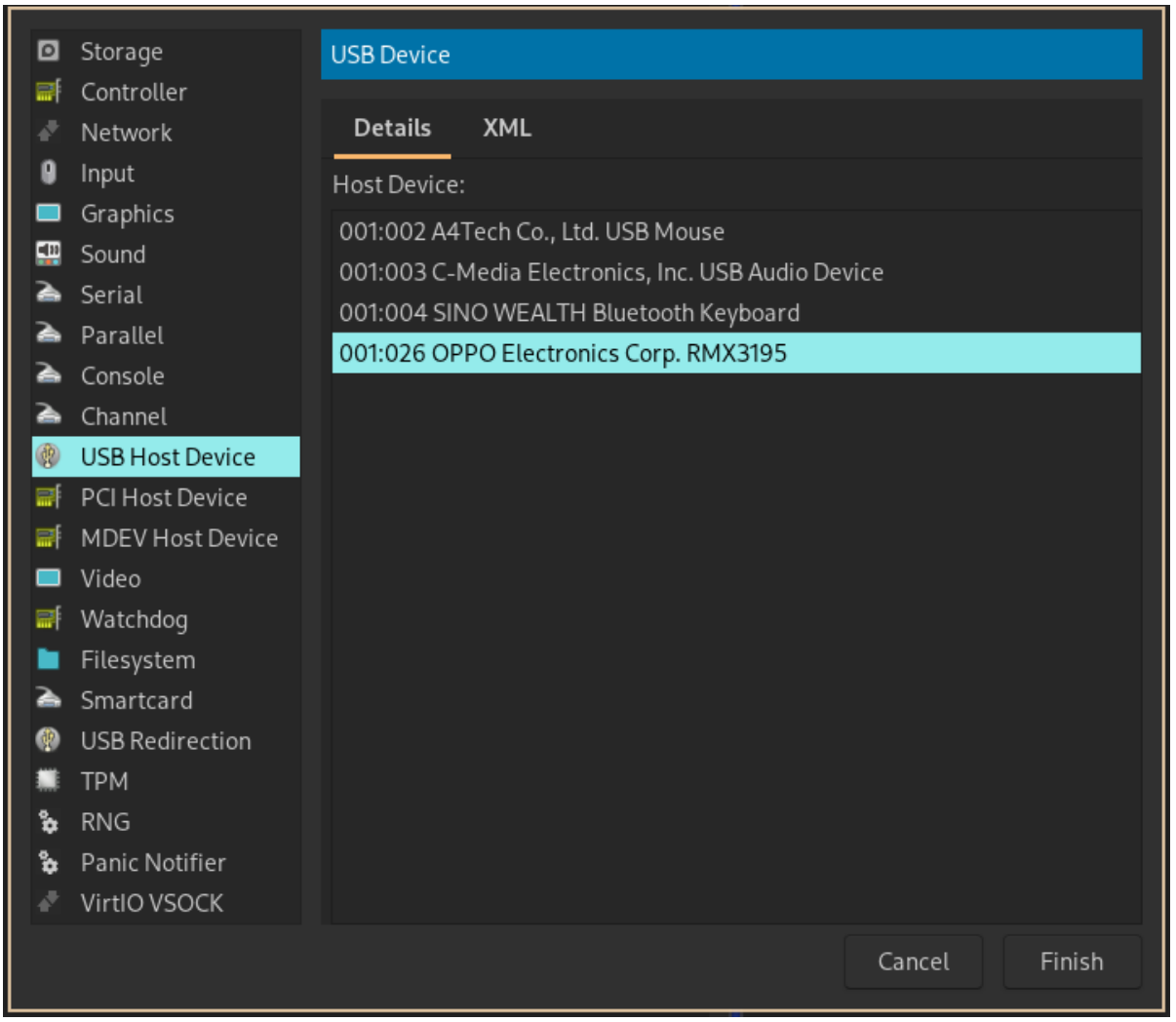
```
sudo pacman -S android-file-transfer
```

After installing the package click on the `info` icon it will open a new window like below.

From there select `Add Hardware`.



You will see a new window, from there select `USB Host Device` and select your phones model and click on `Finish`.

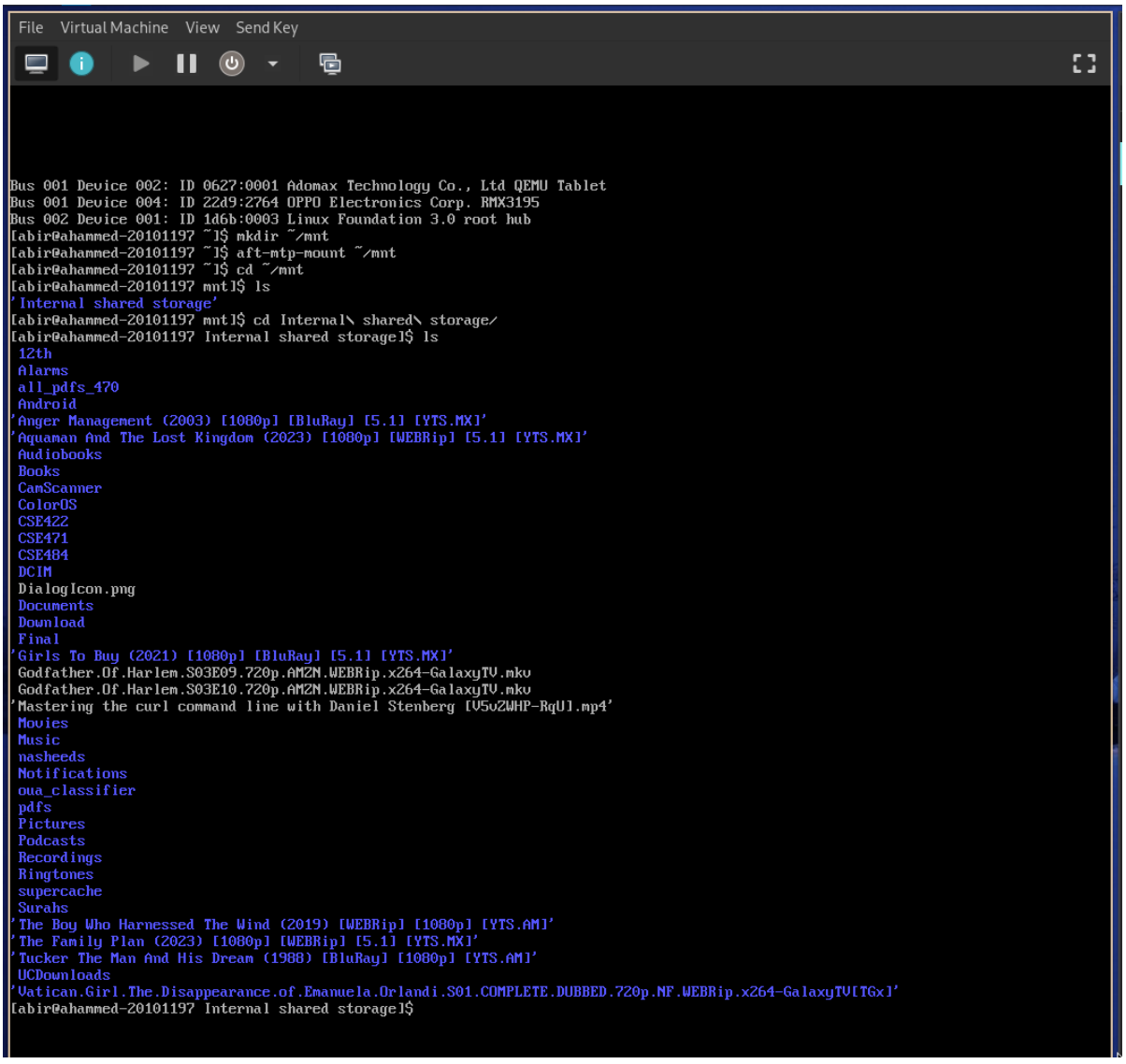


Now get back to the VM and check if the usb device is connected or not using `lsusb` command.

Then create a new directory (mount point) and mount your phone.

```
mkdir ~/mnt
aft-mtp-mount ~/mnt
```

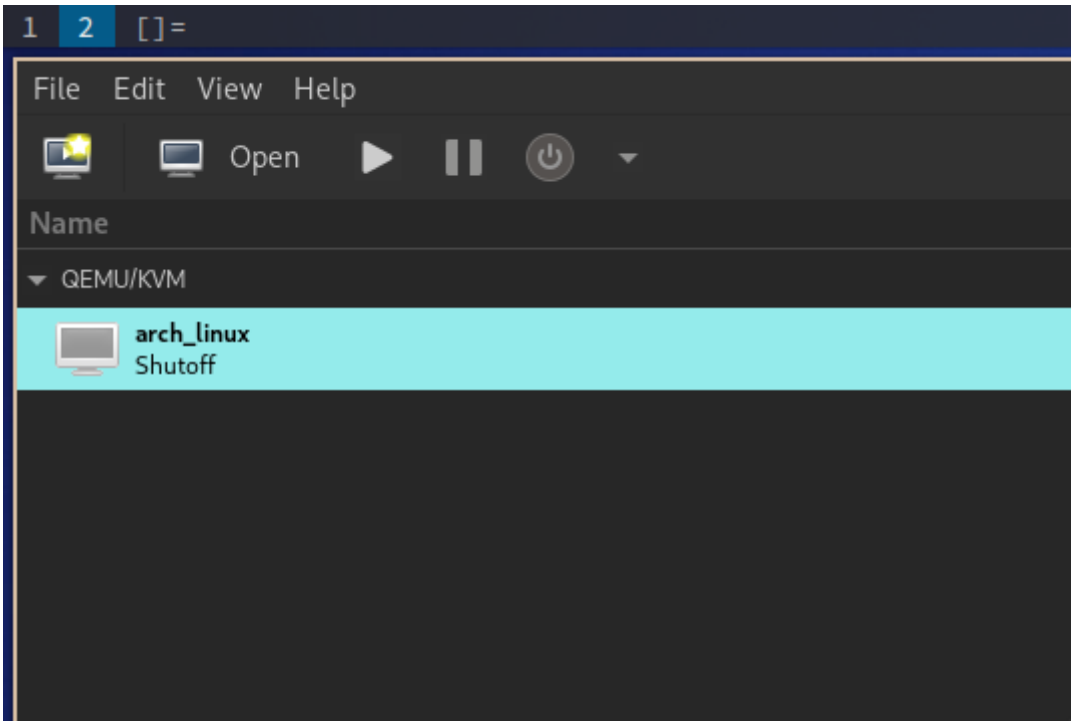
Then check inside the `~/mnt` directory and you will see your phone's content being shown.



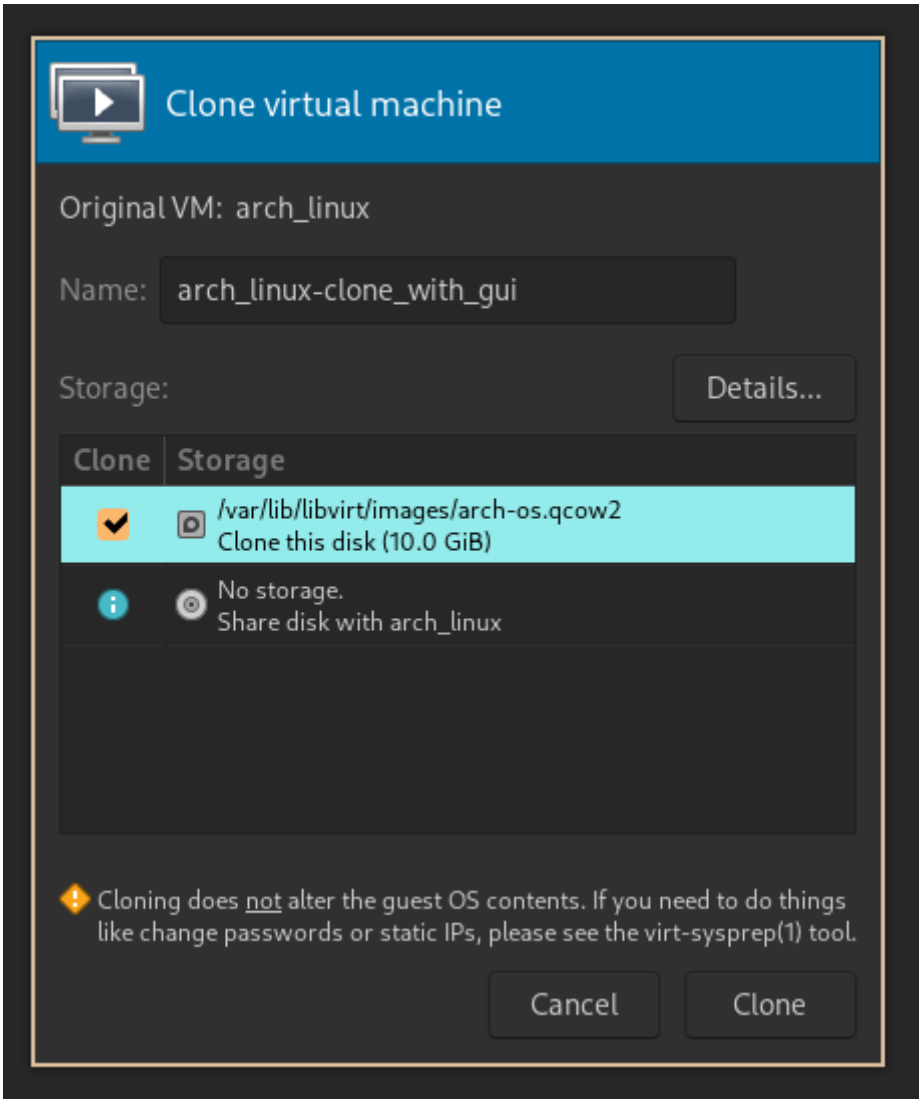
7. Clone a VM using GUI and using a kvm-based command.

Using GUI

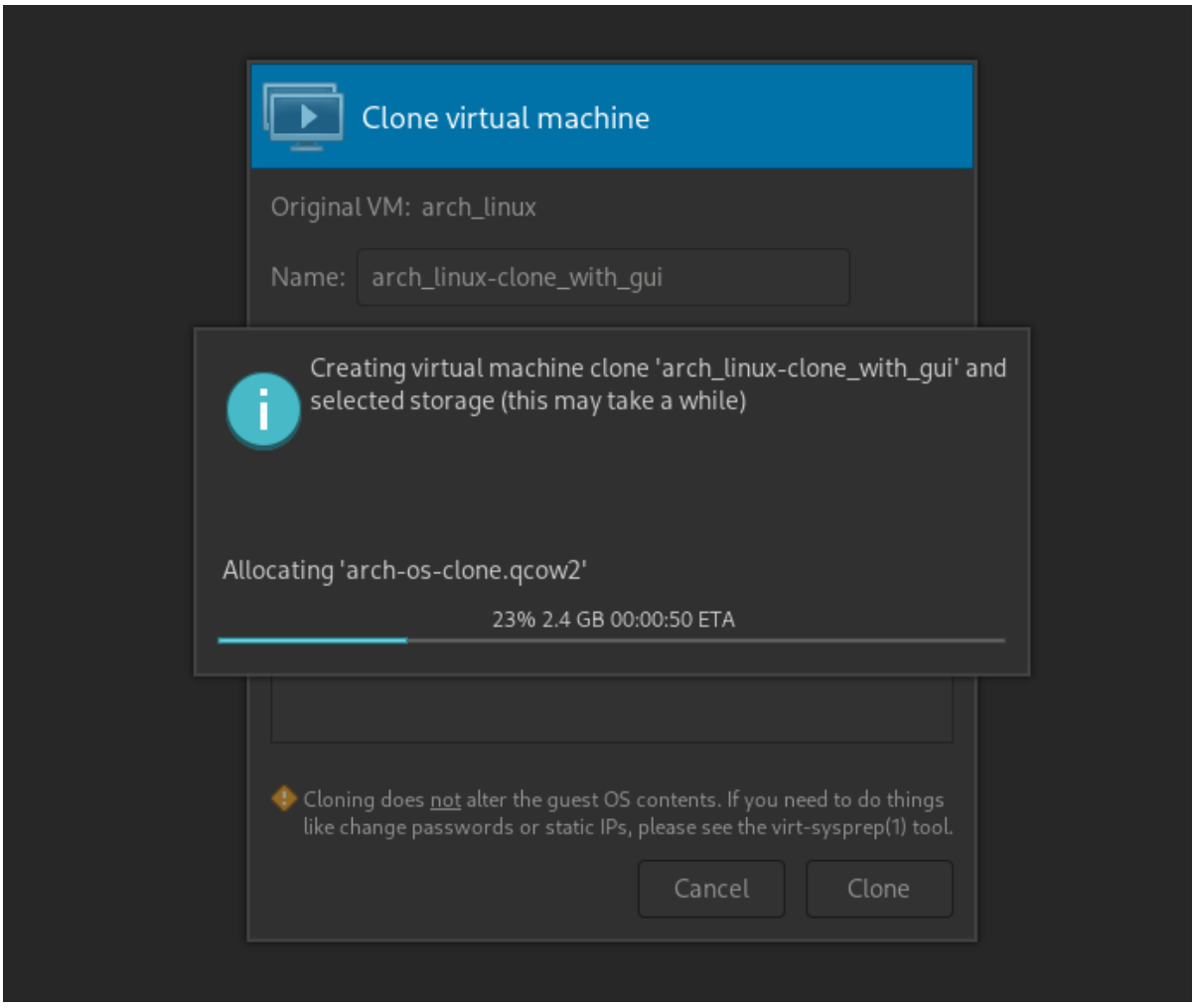
First, open `virt-manager`.



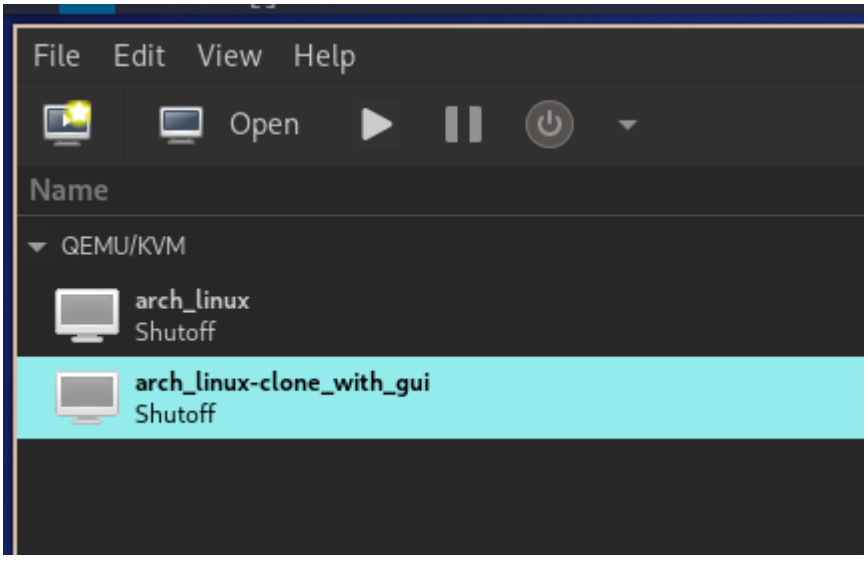
Right click on our selected os will open a menu. Just select `clone` from there.



Then just click on `clone` and we are done cloning.



In the `virt-manager` we can see our cloned VM.

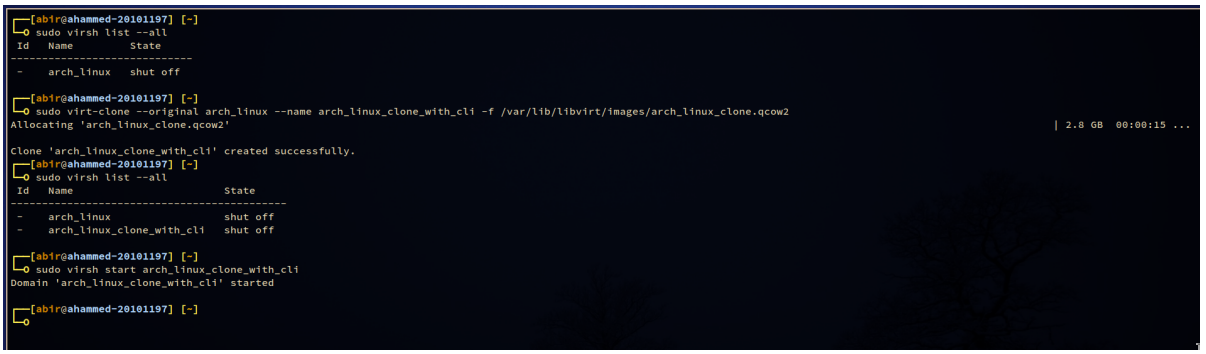


Using CLI

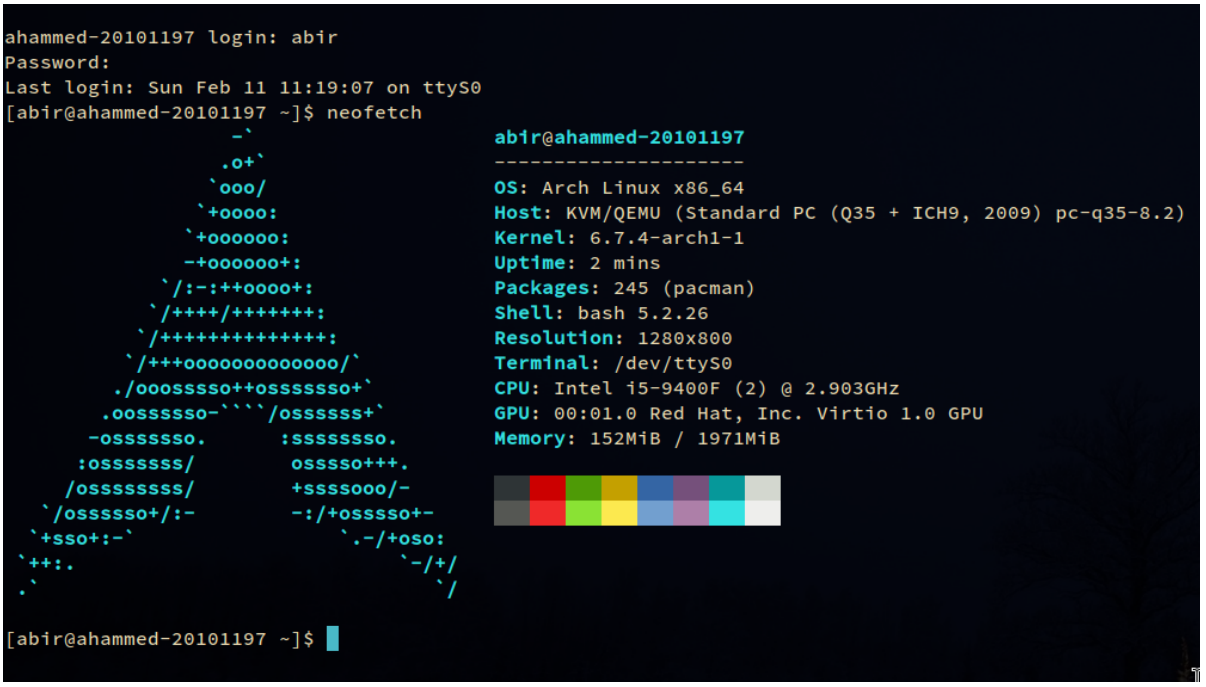
We can also clone a VM by using a command like below.

```
sudo virt-clone --original arch_linux --name arch_linux_clone_with_cli -f
/var/lib/libvirt/images/arch_linux_clone.qcow2
```

Using the `virsh` command we can verify our work.

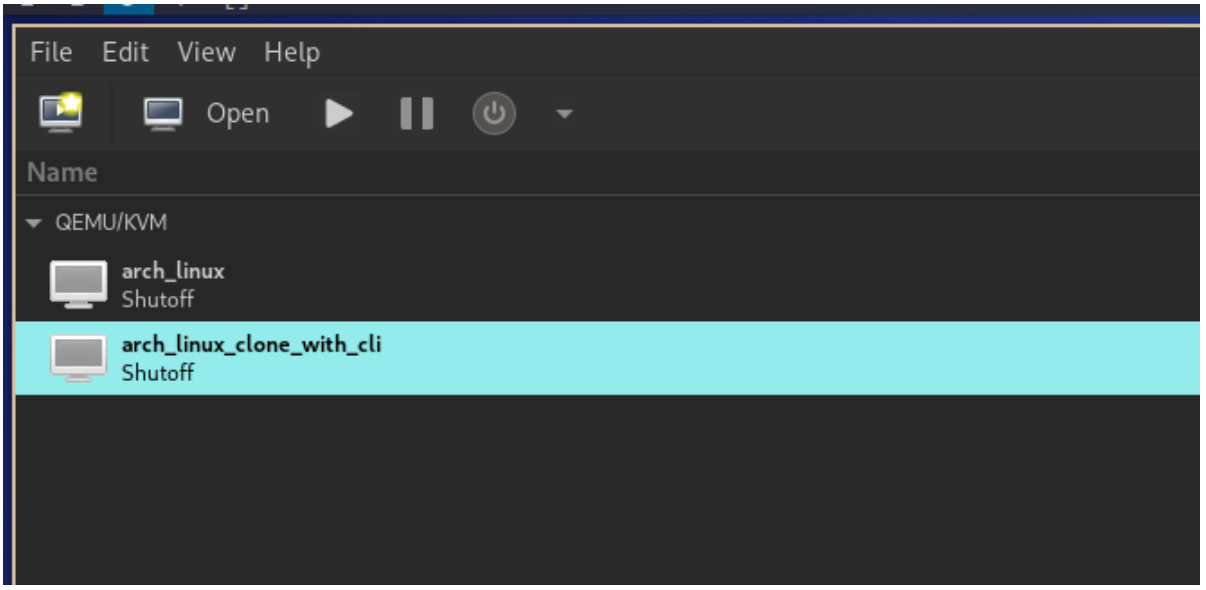


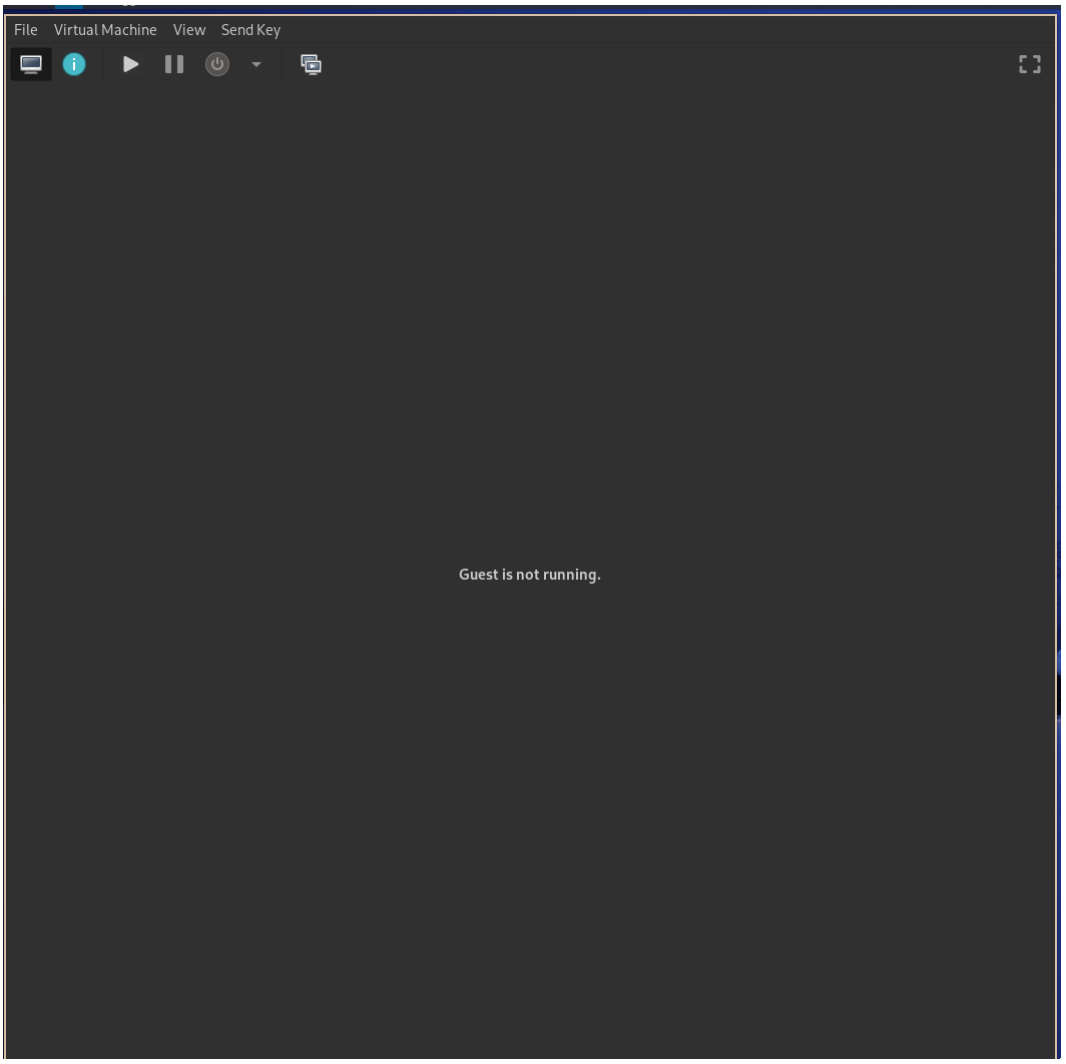
Our cloned VM is running fine.



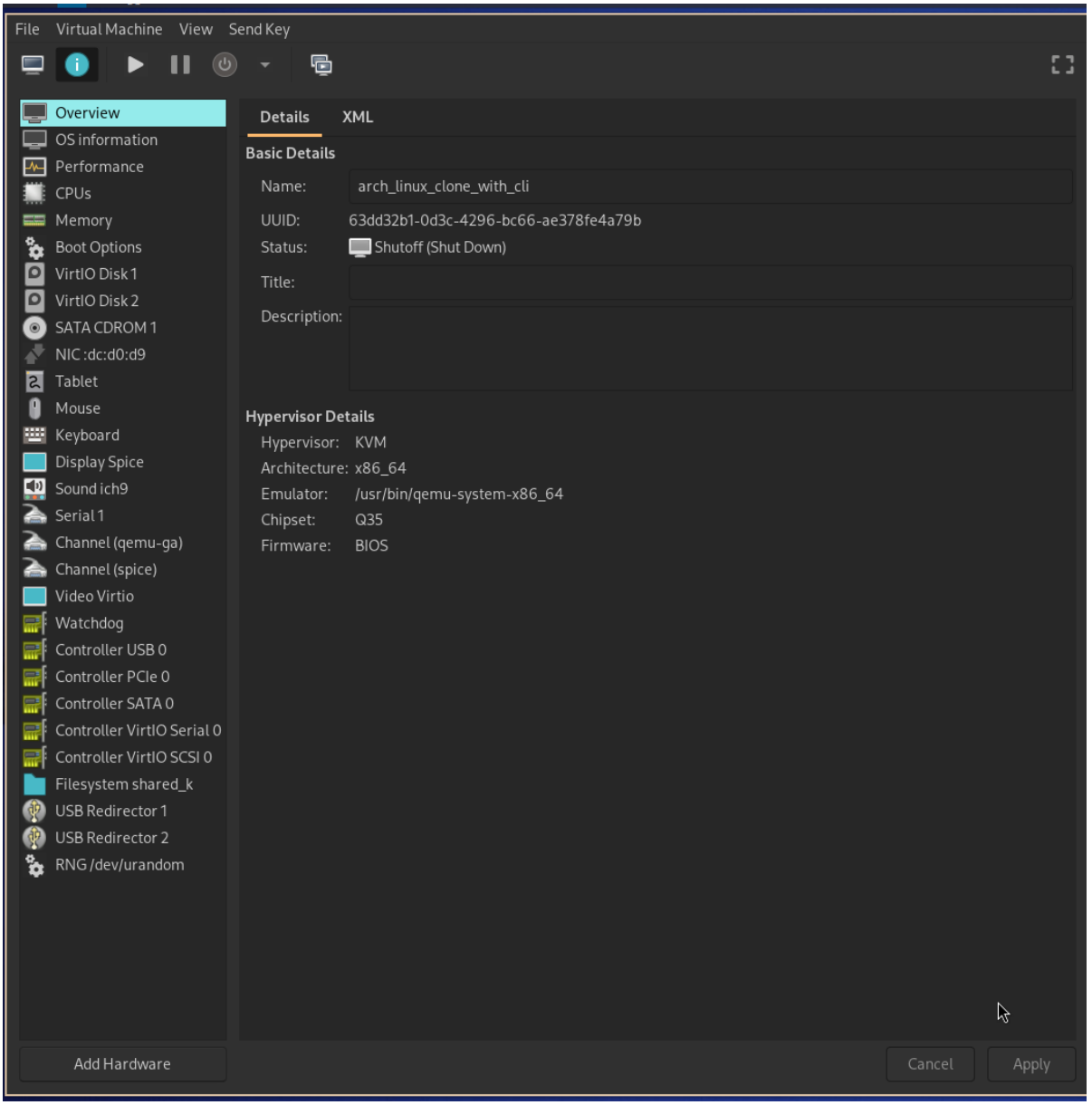
8. Add two hard disks in a new cloned virtual machine using GUI.

Open `virt-manager` select your VM but don't start it.

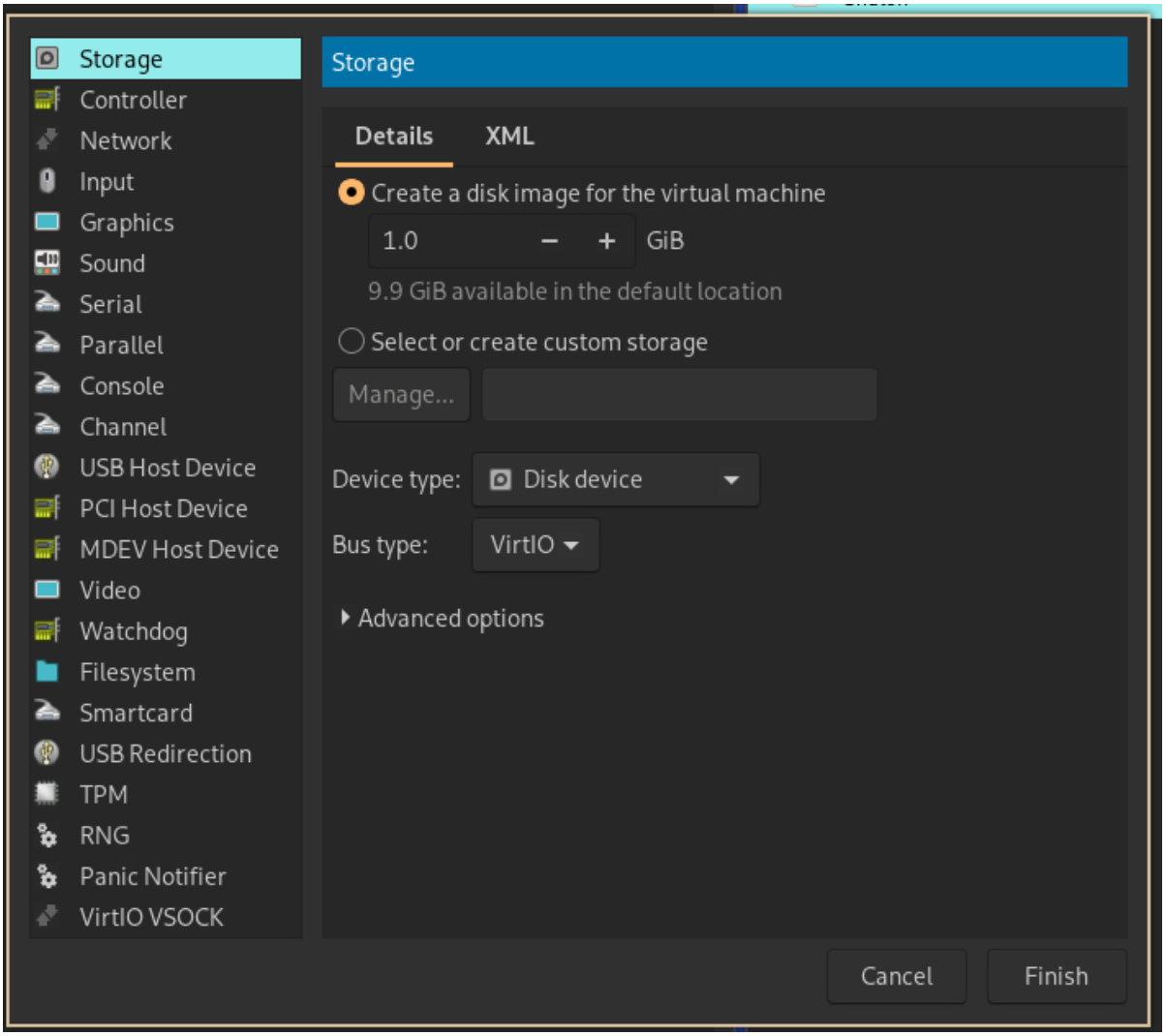




Click on the `info` icon and from there select `Add Hardware`.

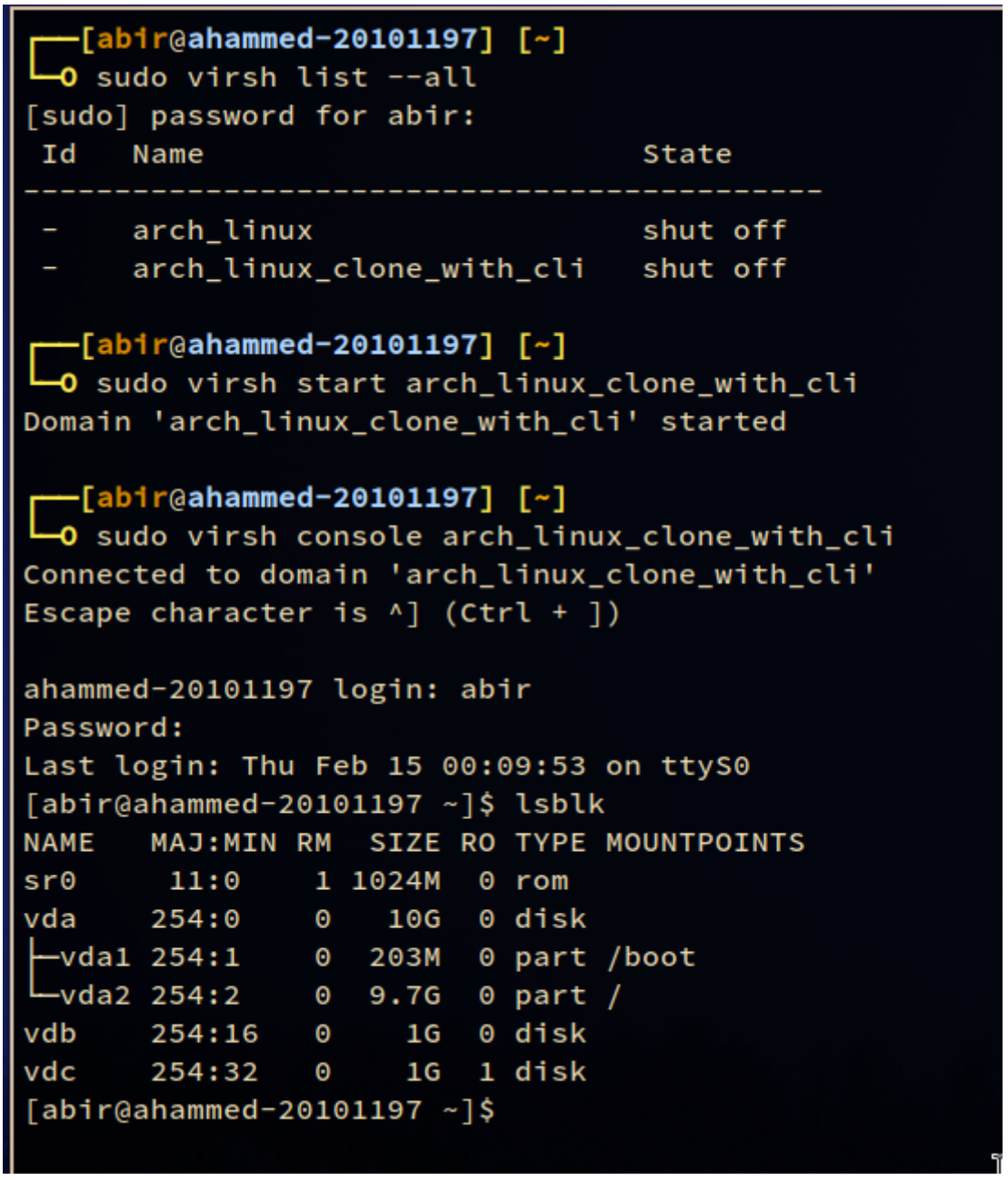


Now select `Storage` and give size to your storage and makesure `Device type` is set to `Disk device` and `Bus type` is set to `VirtIO`.



This way create two storage. Start the VM using CLI or GUI and login.

We can use `lsblk` to justify if our storage devices are connected or not. From the below screen shot we can see two new storages named `/dev/vdb` and `/dev/vdc` have been recognized.

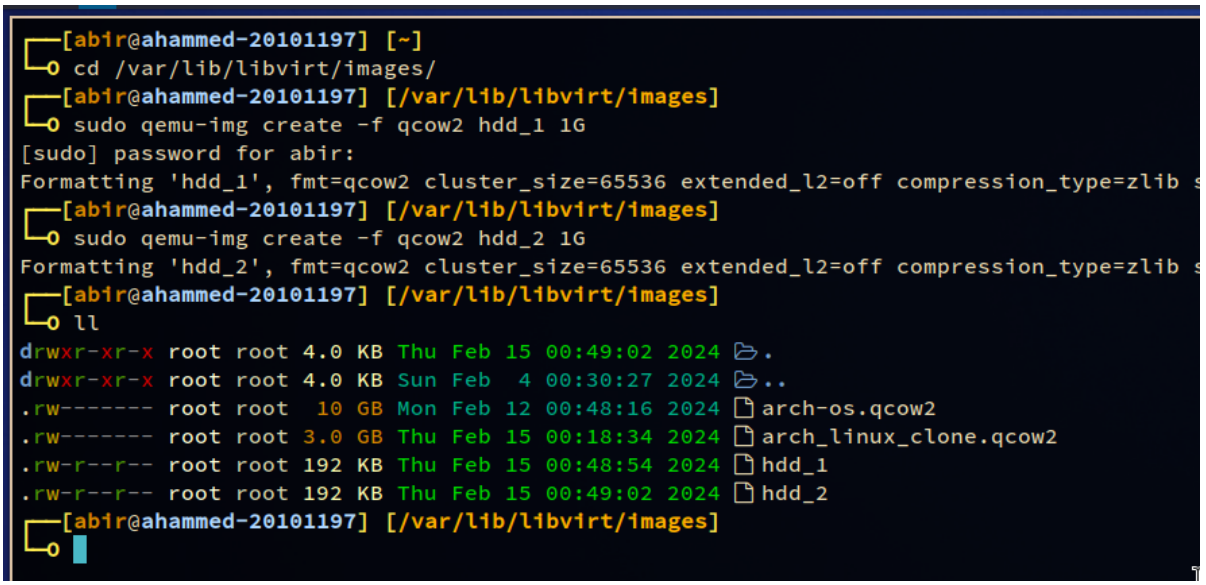


9. Add two hard disks in a new cloned virtual machine using a kvm-based command.

First, go to the `/var/lib/libvirt/images/` because we will create our hard disks there and fire the below command twice to create two hard disks.

```
sudo qemu-img create -f qcow2 hdd_1 1G
```

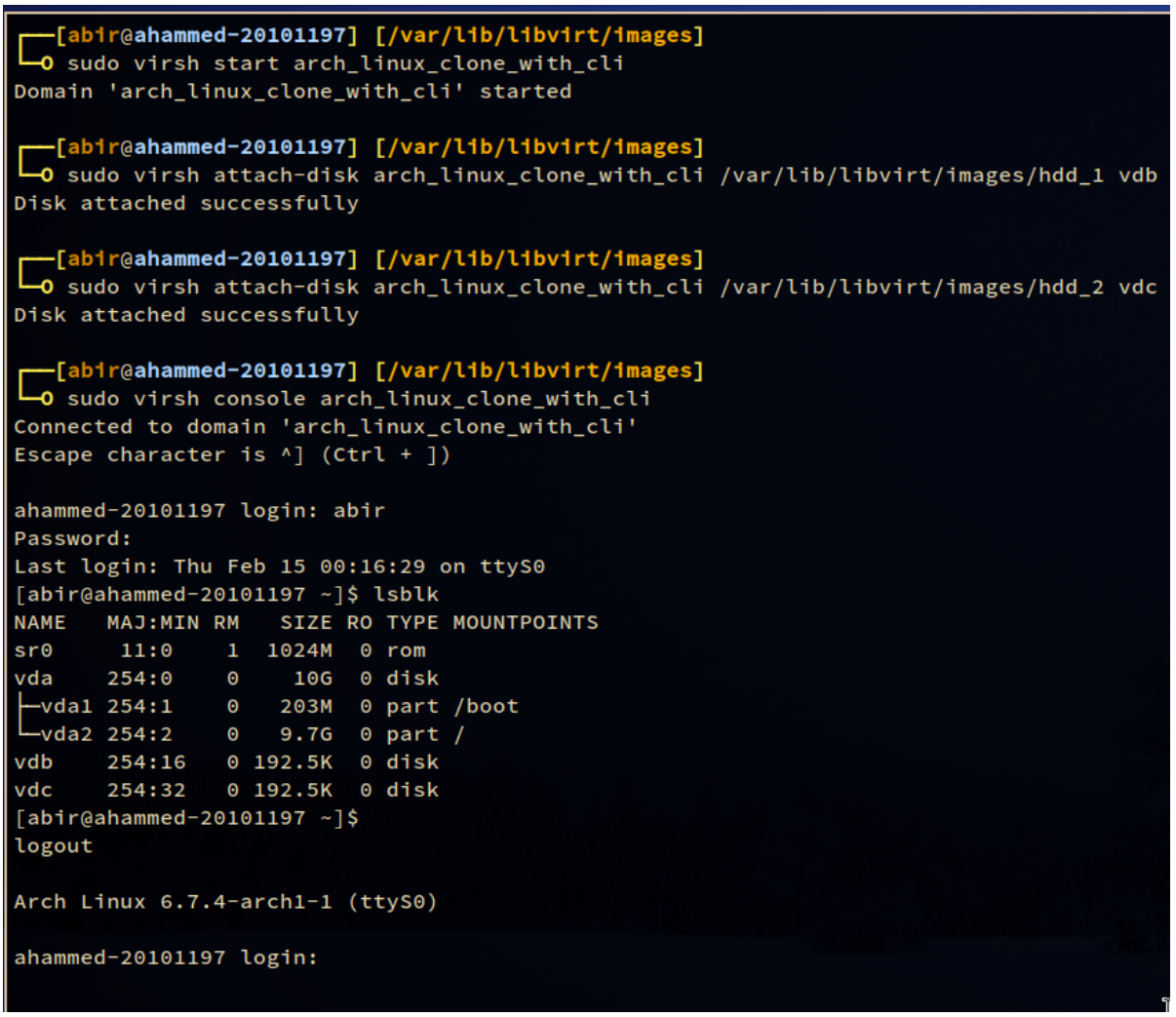
From the screen shot we can see that two new hard disks (hdd_1 and hdd_2) have been created.



Now we have to attach those hard disks using the below command.

```
sudo virsh attach-disk arch_linux_clone_with_cli /var/lib/libvirt/images/hdd_1
vdb
sudo virsh attach-disk arch_linux_clone_with_cli /var/lib/libvirt/images/hdd_2
vdc
```

Now start the VM and using `lsblk` command we can ensure that two new hard disks have been attached.



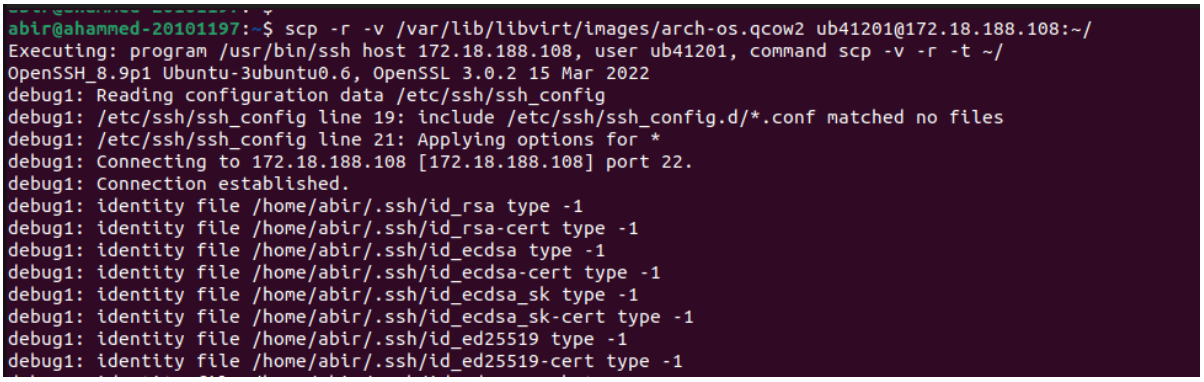
10. How to migrate a VM to another host? Show step-by-step commands and output about migrating any of your VMs. Migrate your VM to your friend's host machine and vice versa. Must show the output after successful migration.

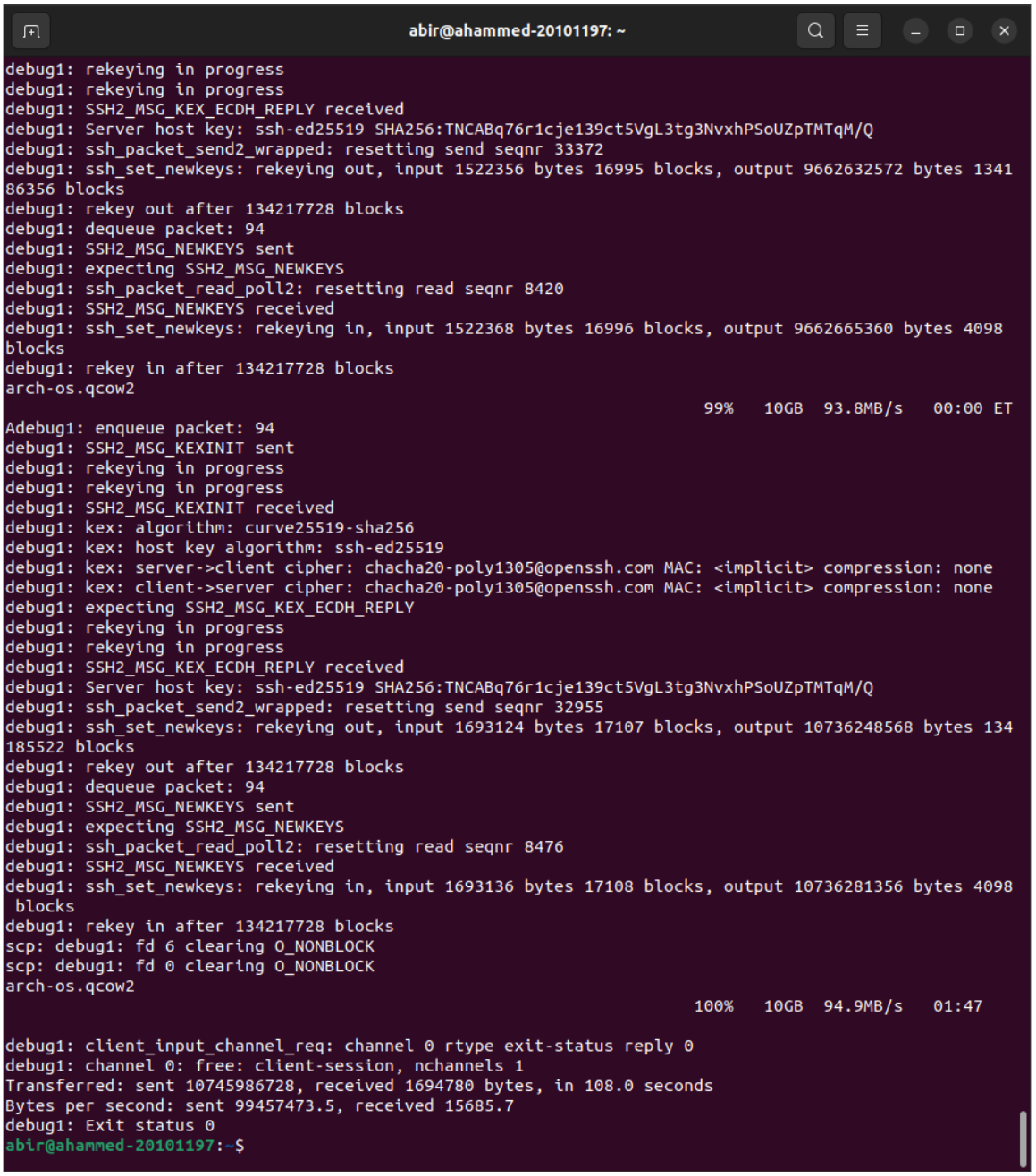
In our machine:

First we have copy the virtual disk image (i.e. `*.qcow2`) file from our computer to our friend's computer. To copy, we can use either pendrive or we can copy over secure shell. Here, we are copying `*.qcow2` file over secure shell more preciesly we are using the `scp` command.

```
scp -r -v /var/lib/libvirt/images/arch-os.qcow2 ub41201@172.18.188.108:~/
```

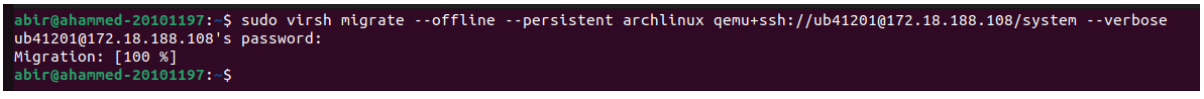
We are copying `arch-os.qcow2` from our machine to the machine located at `172.18.188.108` also the user name is `ub41201`.





The transaction was successful. After that we can use `virsh migrate` command to directly migrate the config file from our machine to the destination machine.

```
sudo virsh migrate --offline --persistent --verbose archlinux
qemu+ssh://ub41201@172.18.188.108/system
```



In friend's machine:

After the successful migration, we just have to change our copied image's directory to `/var/lib/libvirt/images/`.

```
sudo cp arch-os.qcow2 /var/lib/libvirt/images/
```

```
ub41201@tahmid-20101555:~$ sudo cp arch-os.qcow2 /var/lib/libvirt/images/
[sudo] password for ub41201:
```

Then, we can use our regular `virsh` command to start our newly migrated VM.

```
ub41201@tahmid-20101555:~$ sudo virsh list --all
 Id   Name        State
---------------------------
 -    archlinux   shut off

ub41201@tahmid-20101555:~$ sudo virsh start archlinux
Domain 'archlinux' started

ub41201@tahmid-20101555:~$ sudo virsh console archlinux
Connected to domain 'archlinux'
Escape character is ^] (Ctrl + ])

ahammed-20101197 login: abir
Password:
Last login: Thu Feb 15 12:59:57 on ttyS0
[abir@ahammed-20101197 ~]$ neofetch
                    -`                    abir@ahammed-20101197
                   .o+`                   --------------------
                  `ooo/                   OS: Arch Linux x86_64
                 `+oooo:                  Host: KVM/QEMU (Standard PC (Q35 + ICH9, 2009) pc-q35-6.2)
                `+oooooo:                 Kernel: 6.7.4-arch1-1
                -+oooooo+:                Uptime: 17 secs
              `/:-:++oooo+:               Packages: 245 (pacman)
             `/++++/+++++++:              Shell: bash 5.2.26
            `/++++++++++++++:             Resolution: 1024x768
           `/+++ooooooooooooo/`           Terminal: /dev/ttyS0
          ./ooosssso++osssssso+`          CPU: 11th Gen Intel i5-11400 (3) @ 2.592GHz
        .oossssso-````/ossssss+`          GPU: 00:01.0 Red Hat, Inc. Virtio 1.0 GPU
       -ossssssso.      :ssssssso.        Memory: 154MiB / 2014MiB
      :osssssss/        osssso+++.
     /ossssssss/        +ssssooo/-
   `/ossssso+/:-        -:/+osssso+-
  `+sso+:-`                 `.-/+oso:
 `++:.                           `-/+/
 .`                                 `/

[abir@ahammed-20101197 ~]$
```