BRAC UNIVERSITY

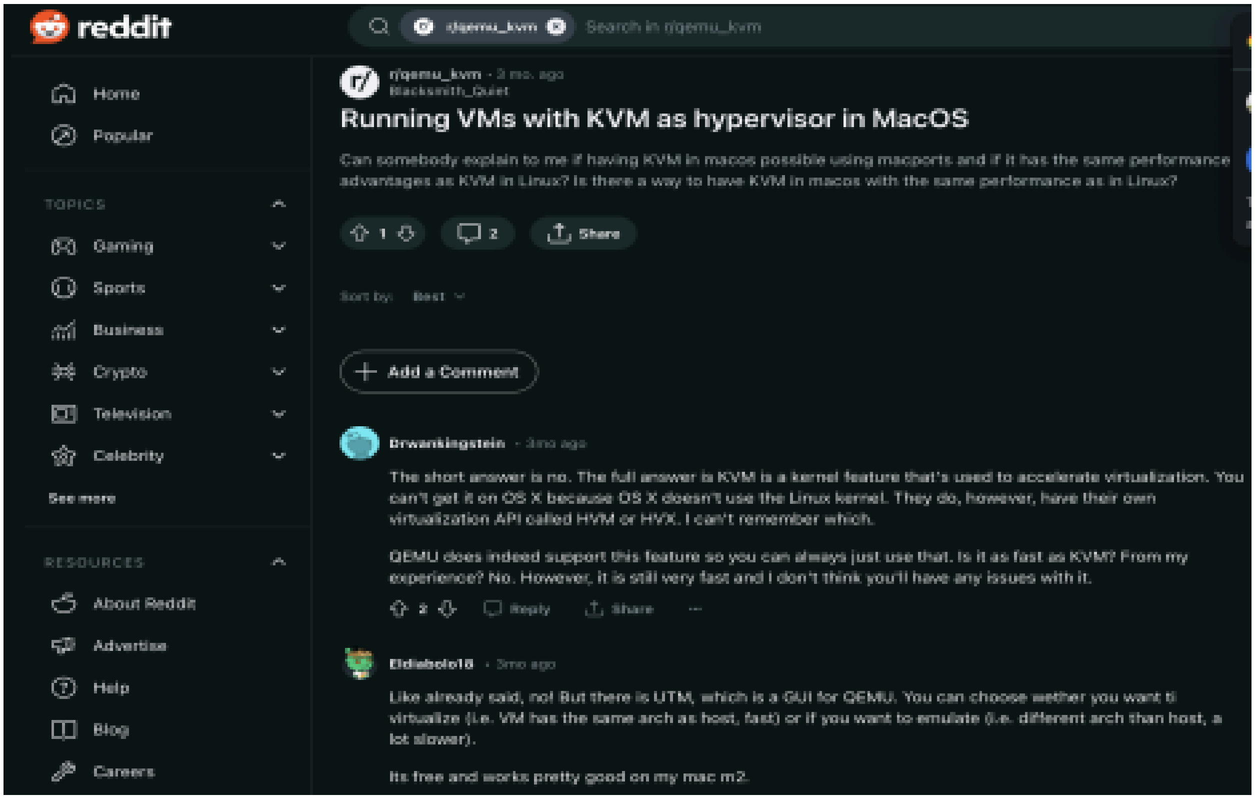
Inspiring Excellence

CSE484(Cloud Computing)

Assignment 2: Everything is Virtual?

Name: Habibun Nabi Hemel
Id: 22241042
Semester: Sping24
Submitted to: Jannatun Noor

Pre Knowledge



r/qemu_kvm · 3 mo. ago
Blacksmith_Quiet

Running VMs with KVM as hypervisor in MacOS

Can somebody explain to me if having KVM in macos possible using macports and if it has the same performance advantages as KVM in Linux? Is there a way to have KVM in macos with the same performance as in Linux?

Drwankingstein · 3mo ago

The short answer is no. The full answer is KVM is a kernel feature that's used to accelerate virtualization. You can't get it on OS X because OS X doesn't use the Linux kernel. They do, however, have their own virtualization API called HVM or HVX. I can't remember which.

QEMU does indeed support this feature so you can always just use that. Is it as fast as KVM? From my experience? No. However, it is still very fast and I don't think you'll have any issues with it.

Eldiabolo18 · 3mo ago

Like already said, no! But there is UTM, which is a GUI for QEMU. You can choose wether you want ti virtualize (i.e. VM has the same arch as host, fast) or if you want to emulate (i.e. different arch than host, a lot slower).

Its free and works pretty good on my mac m2.

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.

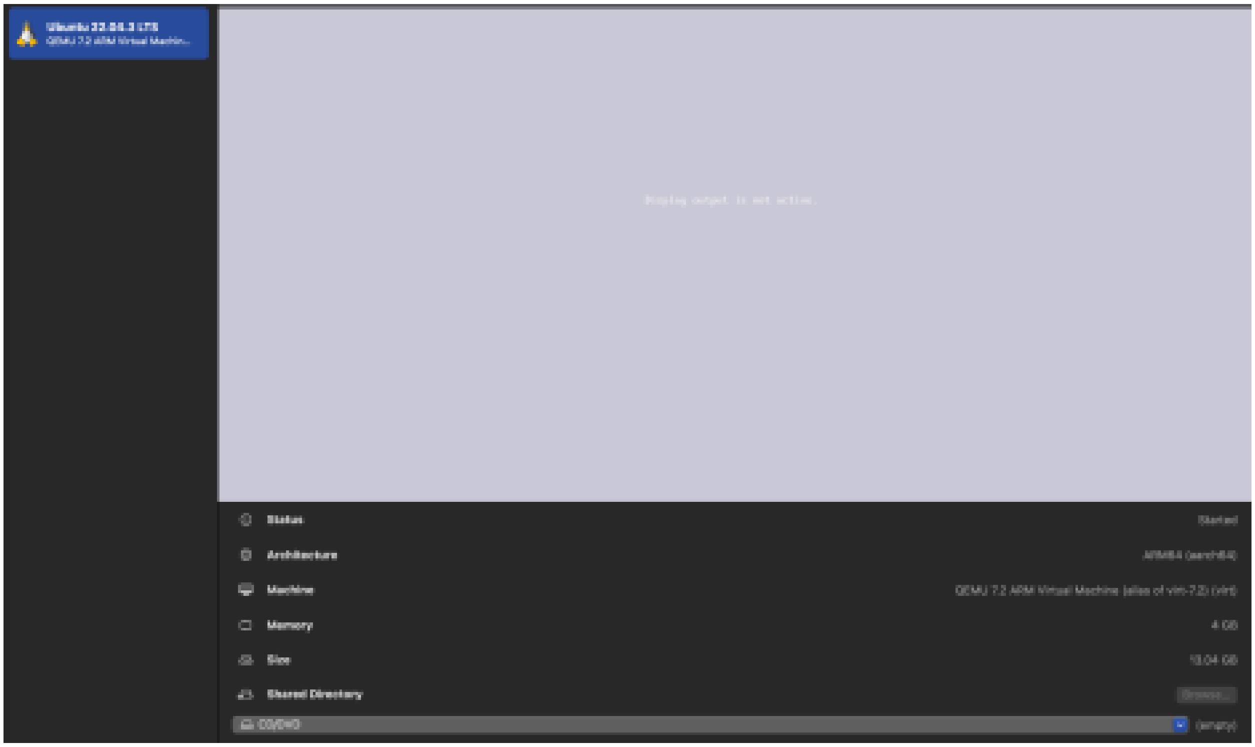
Let me check for the Macbook M1 chip that its ARM(OS) and hardware system supports the nested virtualization or not.

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



Ubuntu 22.04.3 LTS
QEMU 7.2 ARM Virtual Machin...

Display output is not active.

Status Started

Architecture ARM64 (aarch64)

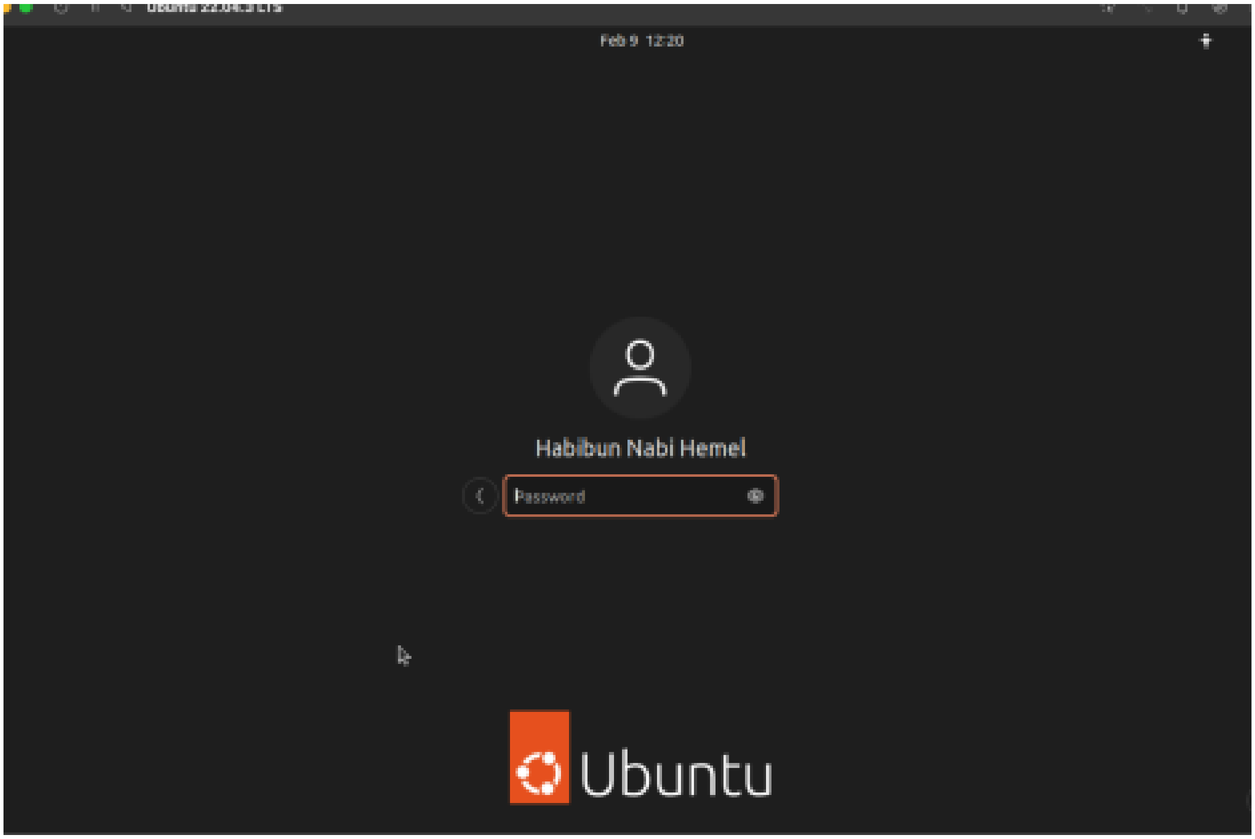
Machine QEMU 7.2 ARM Virtual Machine (alias of virt-7.2) (virt)

Memory 4 GB

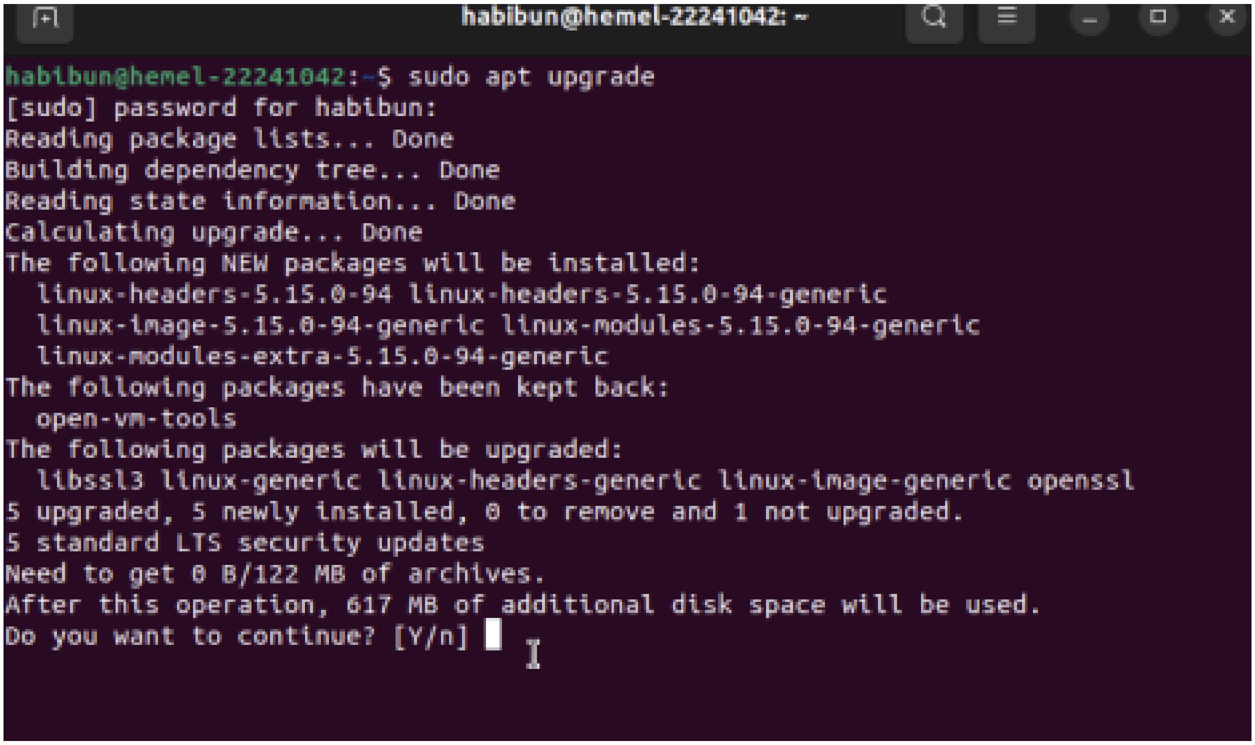
Size 10.04 GB

Shared Directory Browse...

CD/DVD (empty)



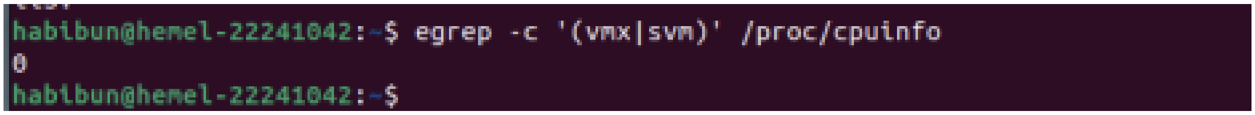
Now I will check the mac m1 chip supports the nested virtual machine or not Before Installing KVM directly we need to do some checks which are pre-requisites. Firstly, ensure the current (hosted OS) is up to date. Launch the terminal and update your local package index to get off the ground. Run the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



```
habibun@hemel-22241042:~$ sudo apt 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.

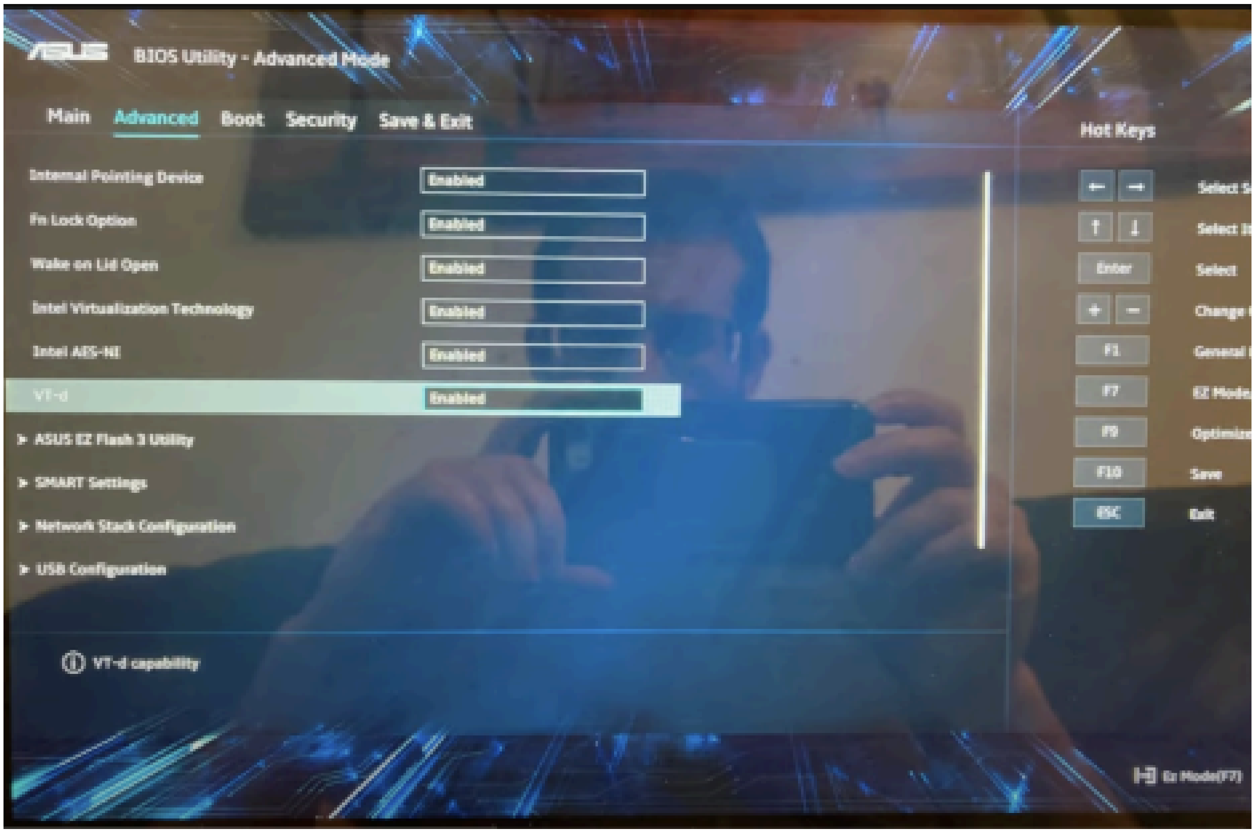


```
habibun@hemel-22241042:~$ egrep -c '(vmx|svm)' /proc/cpuinfo
0
habibun@hemel-22241042:~$
```

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

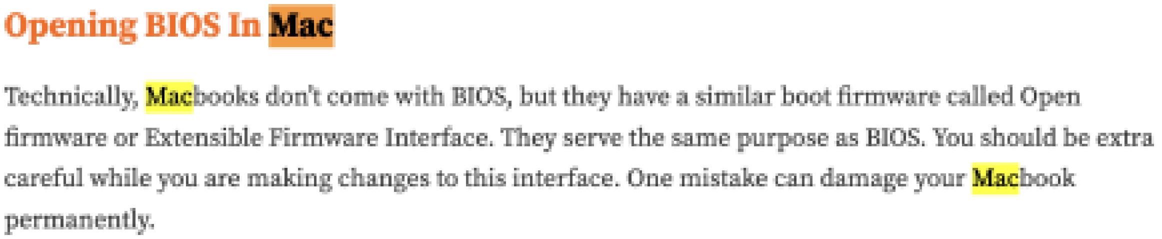
So my machine is not supporting that so I have the last option that is doing some change in the BIOS setting .lets do it This process can be differ from one to another like asus laptop users will do some change according to their manul ,dell users will do in another way ,mac users will do it another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

ASUS users can do it in this way:



Internal Pointing Device: Enabled
Fn Lock Option: Enabled
Wake on Lid Open: Enabled
Intel Virtualization Technology: Enabled
Intel AES-NI: Enabled
VT-d: Enabled
ASUS EZ Flash 3 Utility
SMART Settings
Network Stack Configuration
USB Configuration
VT-d capability

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



Opening BIOS In Mac

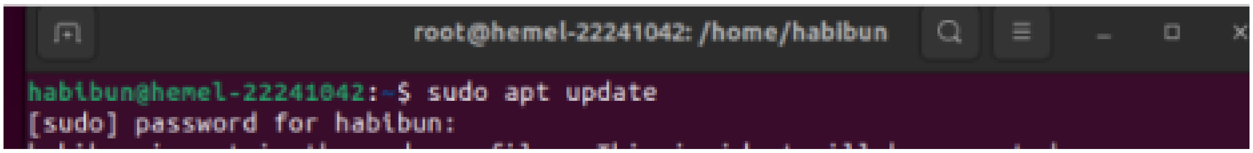
Technically, Macbooks don’t come with BIOS, but they have a similar boot firmware called Open firmware or Extensible Firmware Interface. They serve the same purpose as BIOS. You should be extra careful while you are making changes to this interface. One mistake can damage your Macbook permanently.

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**

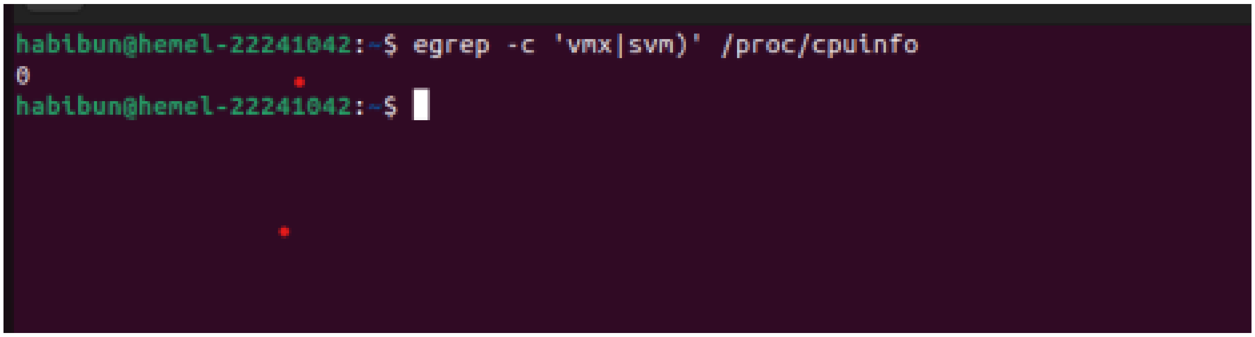


```
habibun@hemel-22241042:~$ sudo apt update
[sudo] password for habibun:
```

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**

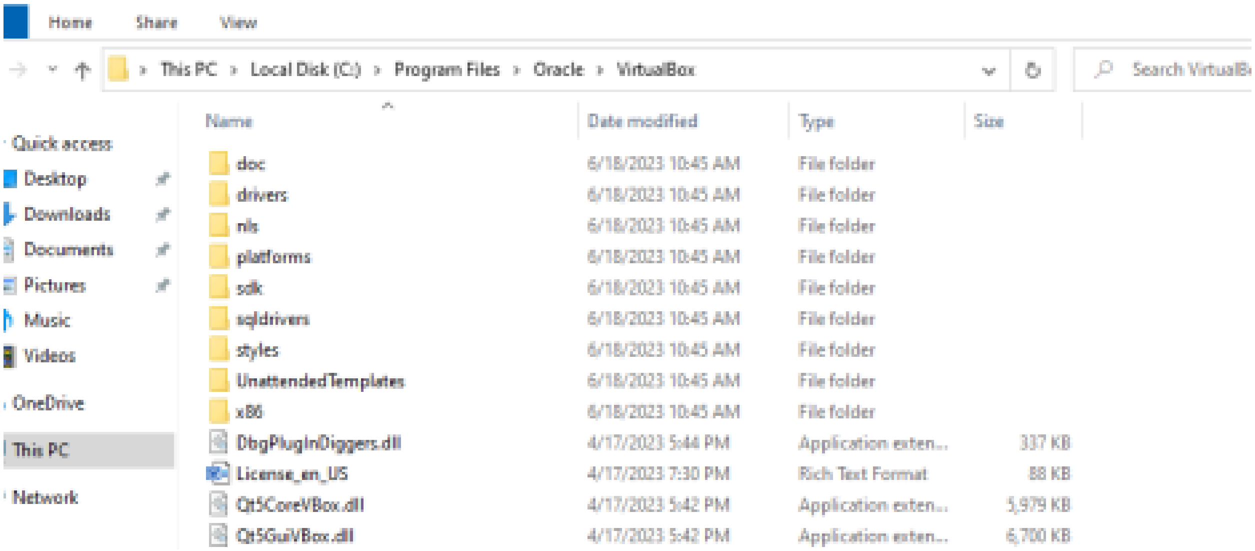


```
habibun@hemel-22241042:~$ egrep -c 'vmx|svm)' /proc/cpuinfo
0
habibun@hemel-22241042:~$ 
```

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.**

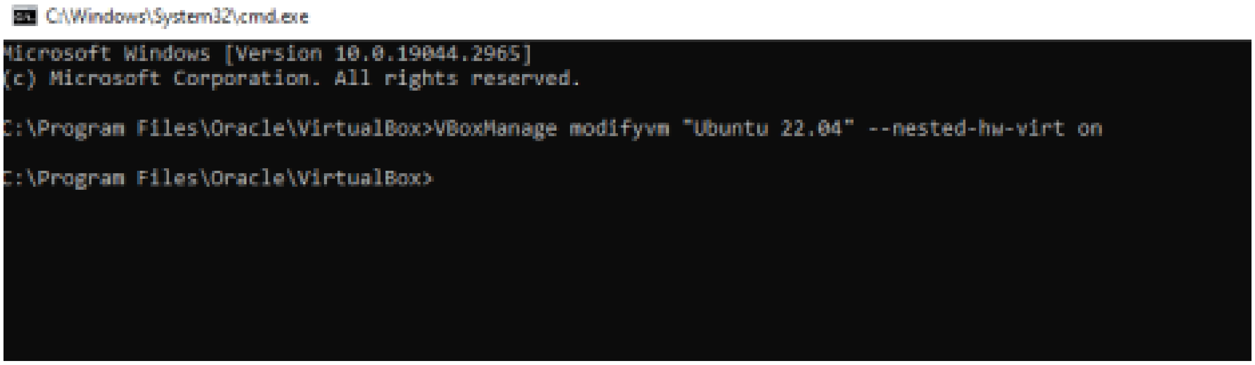


Name	Date modified	Type	Size
doc	6/18/2023 10:45 AM	File folder	
drivers	6/18/2023 10:45 AM	File folder	
nls	6/18/2023 10:45 AM	File folder	
platforms	6/18/2023 10:45 AM	File folder	
sdk	6/18/2023 10:45 AM	File folder	
sqldrivers	6/18/2023 10:45 AM	File folder	
styles	6/18/2023 10:45 AM	File folder	
UnattendedTemplates	6/18/2023 10:45 AM	File folder	
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.

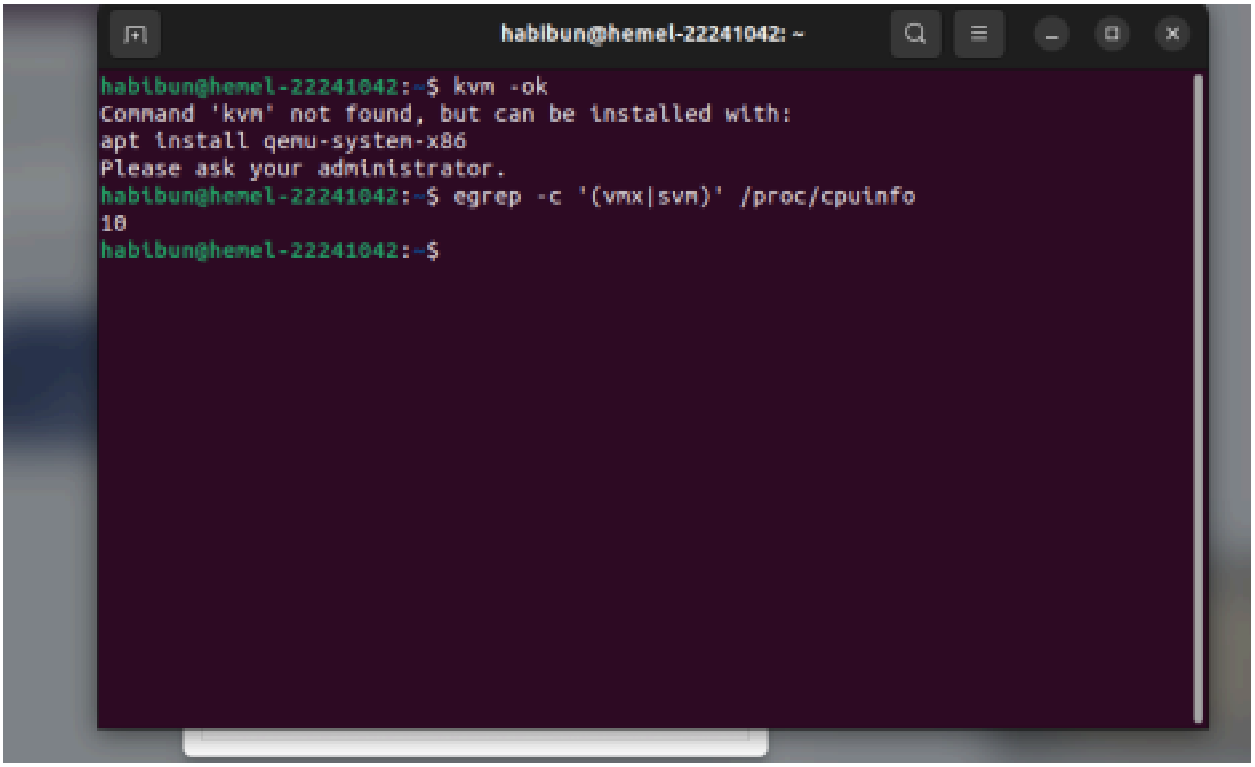


```
Microsoft Windows [Version 10.0.19044.2965]
(c) Microsoft Corporation. All rights reserved.

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.**

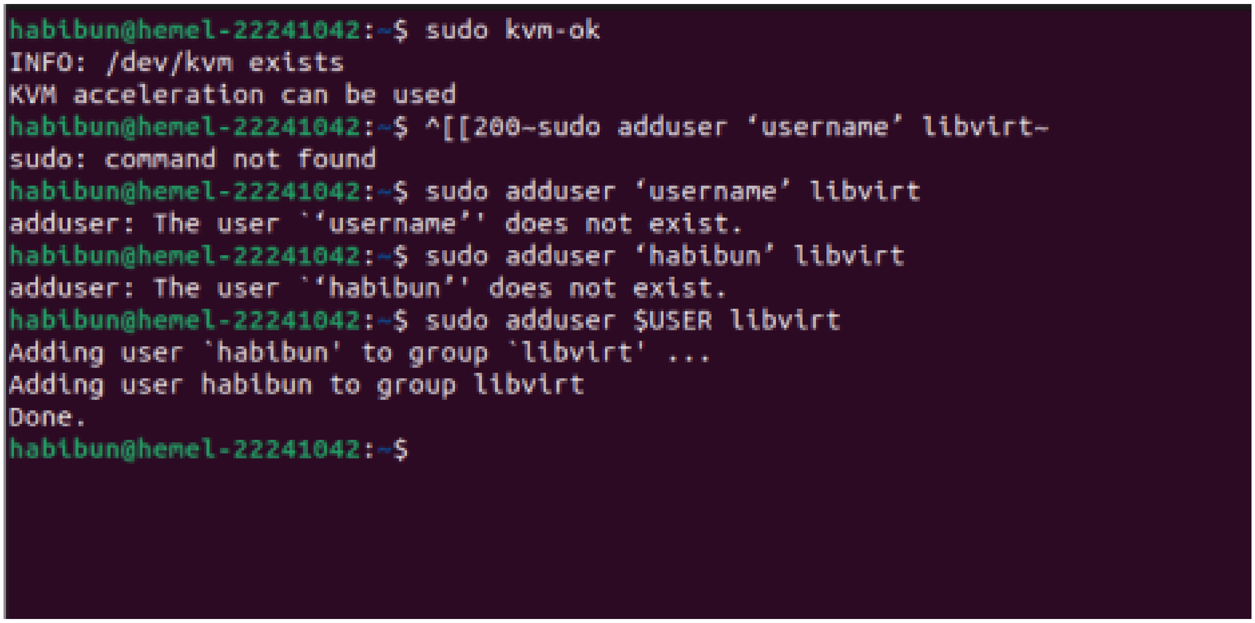


```
habibun@hemel-22241042:~$ kvm -ok
Command 'kvm' not found, but can be installed with:
apt install qemu-system-x86
Please ask your administrator.
habibun@hemel-22241042:~$ egrep -c '(vmx|svm)' /proc/cpuinfo
10
habibun@hemel-22241042:~$
```

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:

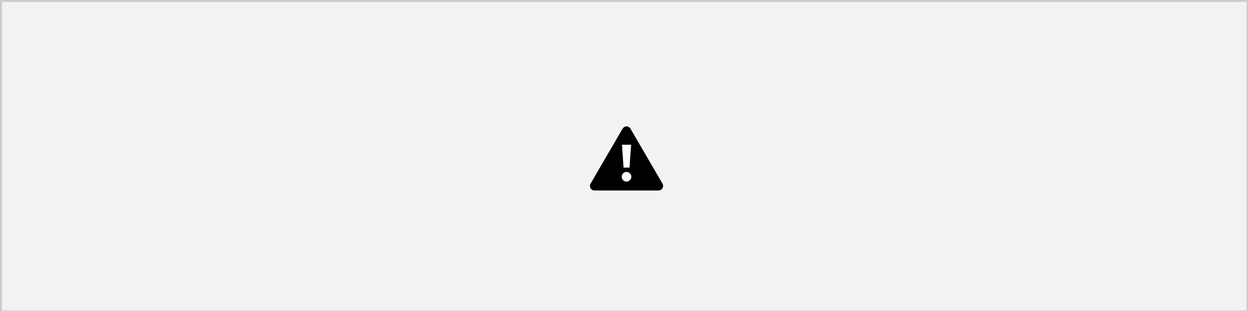
sudo kvm-ok



```
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 :

Verify the Installation:



Installing kvm

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

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

After that we need to enable and start Libvirt Demon .

Two command are :

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

Then

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

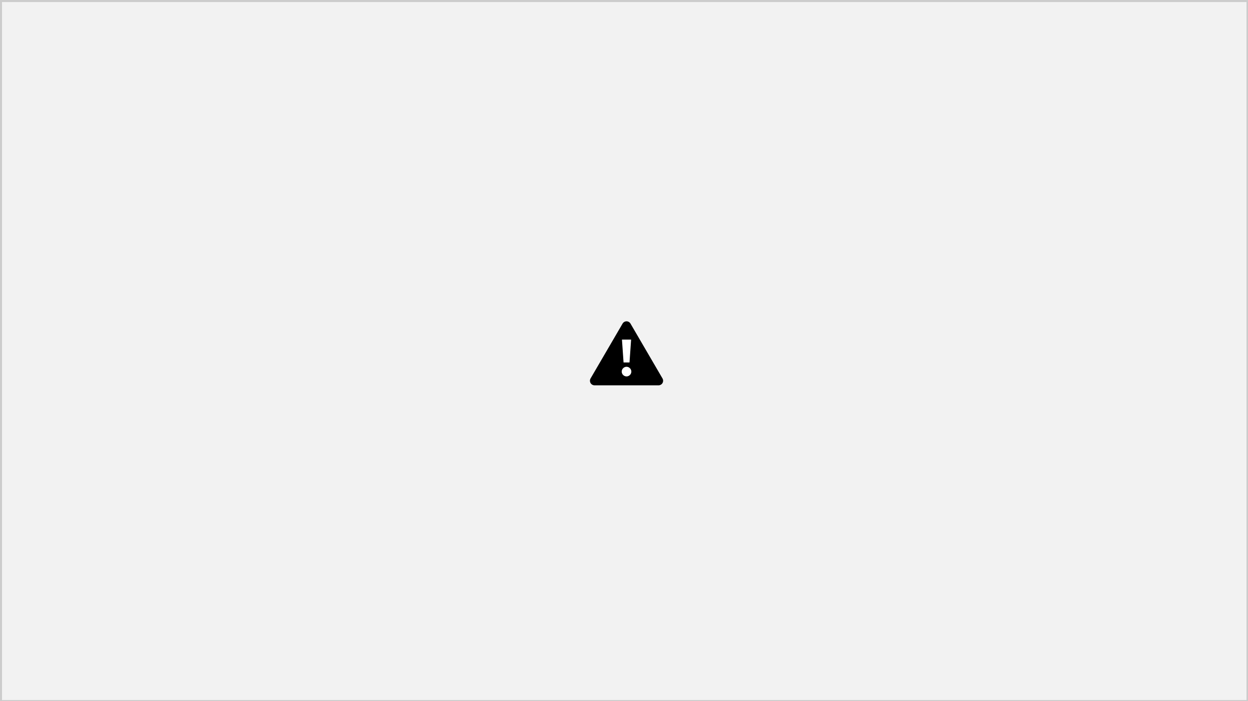
- **sudo systemctl status libvirtd**

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

More clean view:

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.

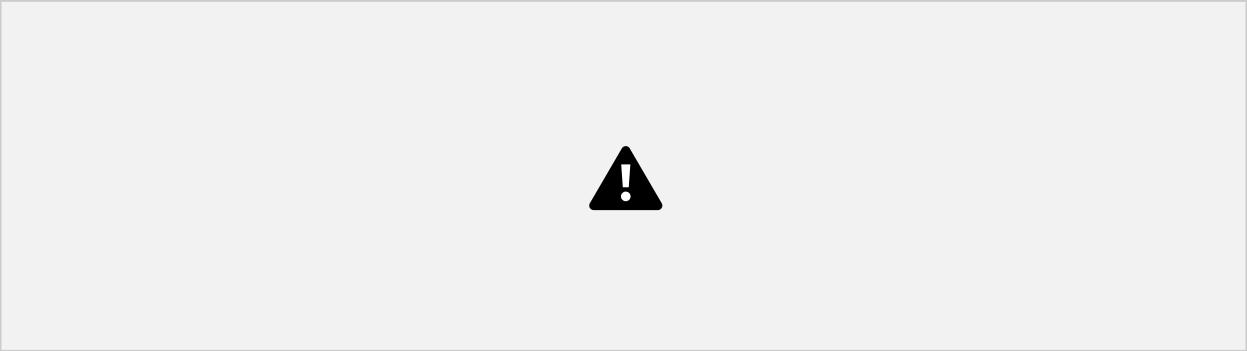
Command:

```
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\
```

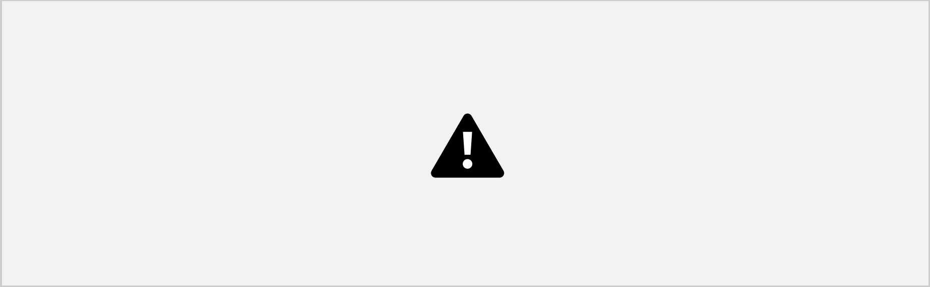
Initially, I designated the name of my new virtual machine as "commandLineVM" utilizing the --name parameter. Following that, I allocated 2048 MB of memory using --memory 2048. Subsequently, I assigned 2 cores to the CPU with --vcpus 2. Lastly, I specified the path to the image file using --cdrom /home/habibun/iso 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

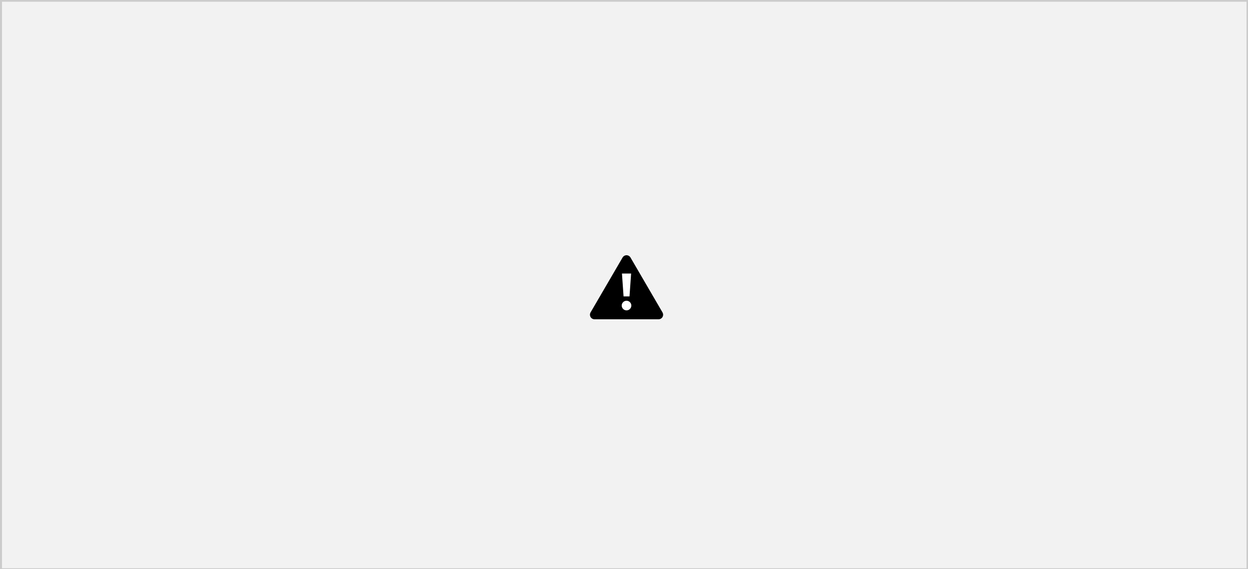
```
sudo apt install -y qemu qemu-kvm libvirt-daemon libvirt-clients bridge-utils virt-manager
```



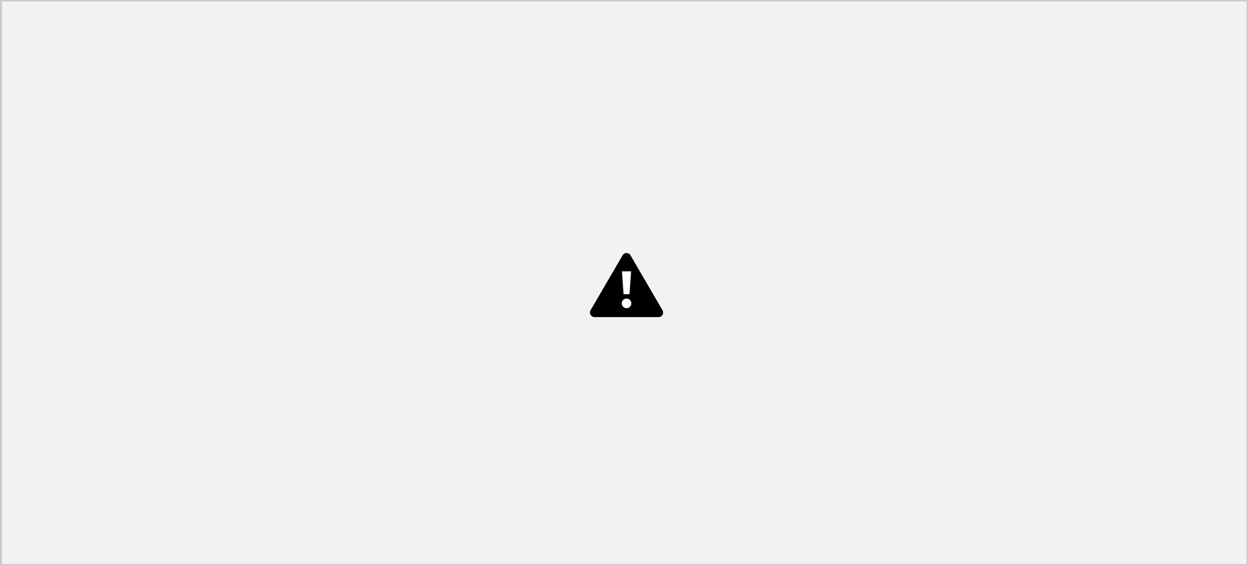
Once installed, we must confirm if the required kvm modules are loaded. Therefore, run the command: **lsmod | grep kvm**



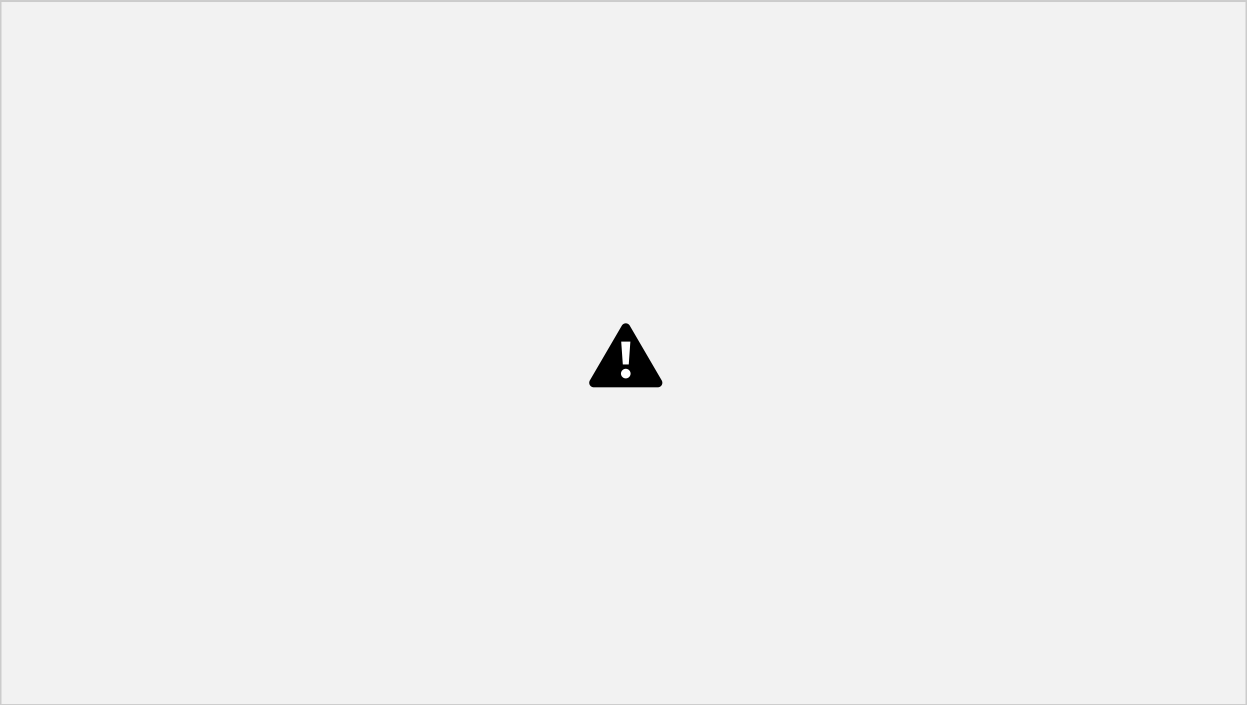
Additionally, confirm that the libvirtd daemon is running as follows.
Command: **sudo systemctl status libvirtd.service**

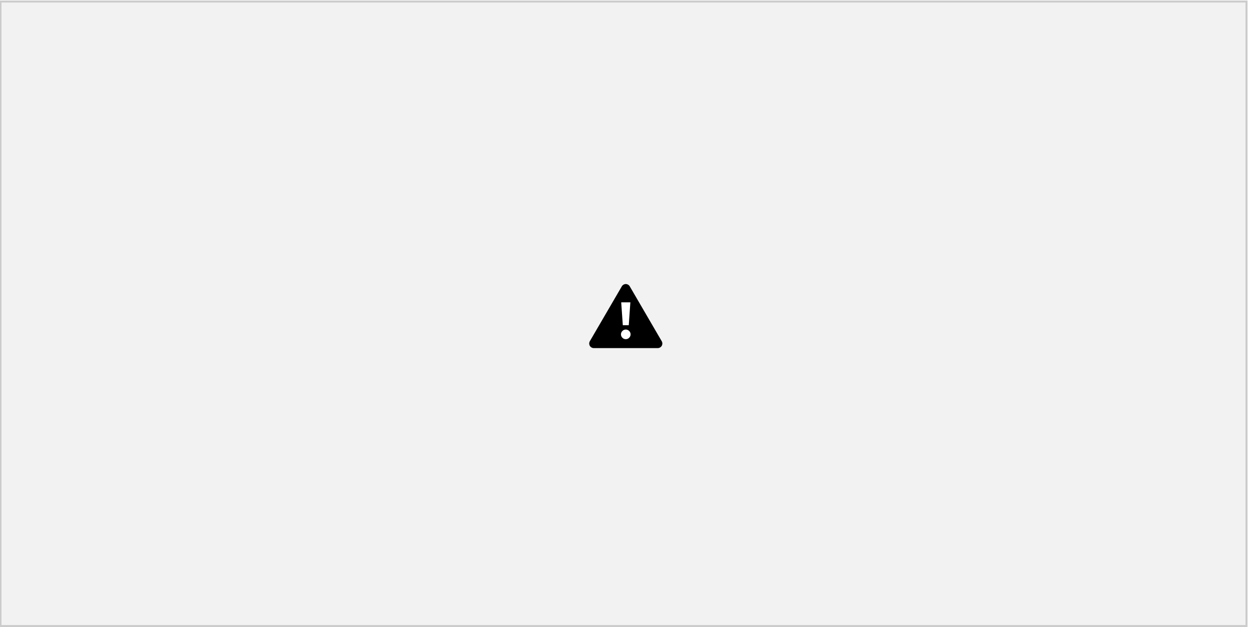


TASK 5. Make a shared folder between host os and guest os using cli. If any changes happen in Guest OS then the same change should be reflected in the Host-OS. This must be applicable for vice versa.



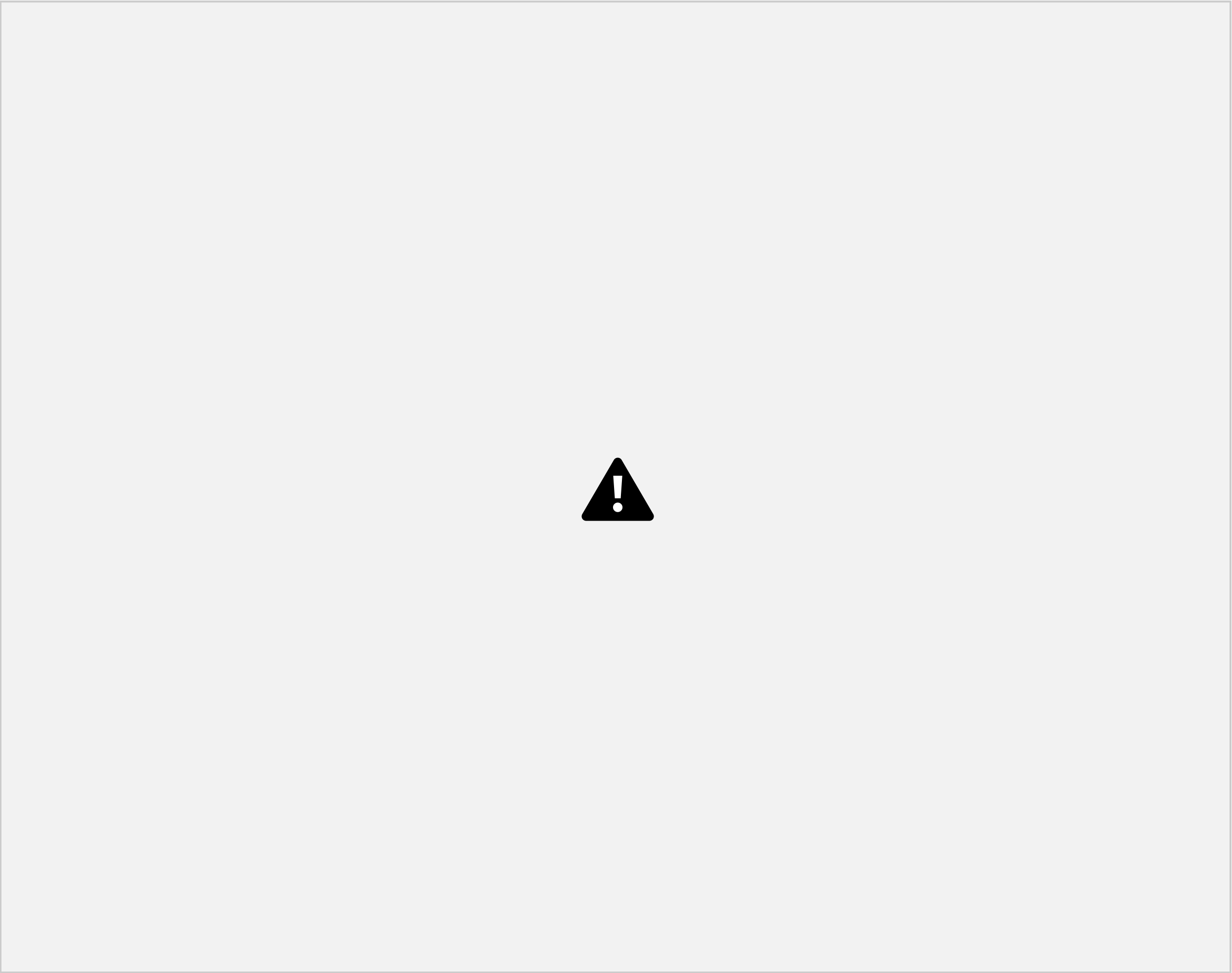
TASK6. Connect the guest OS with a phone and access the phone’s files.

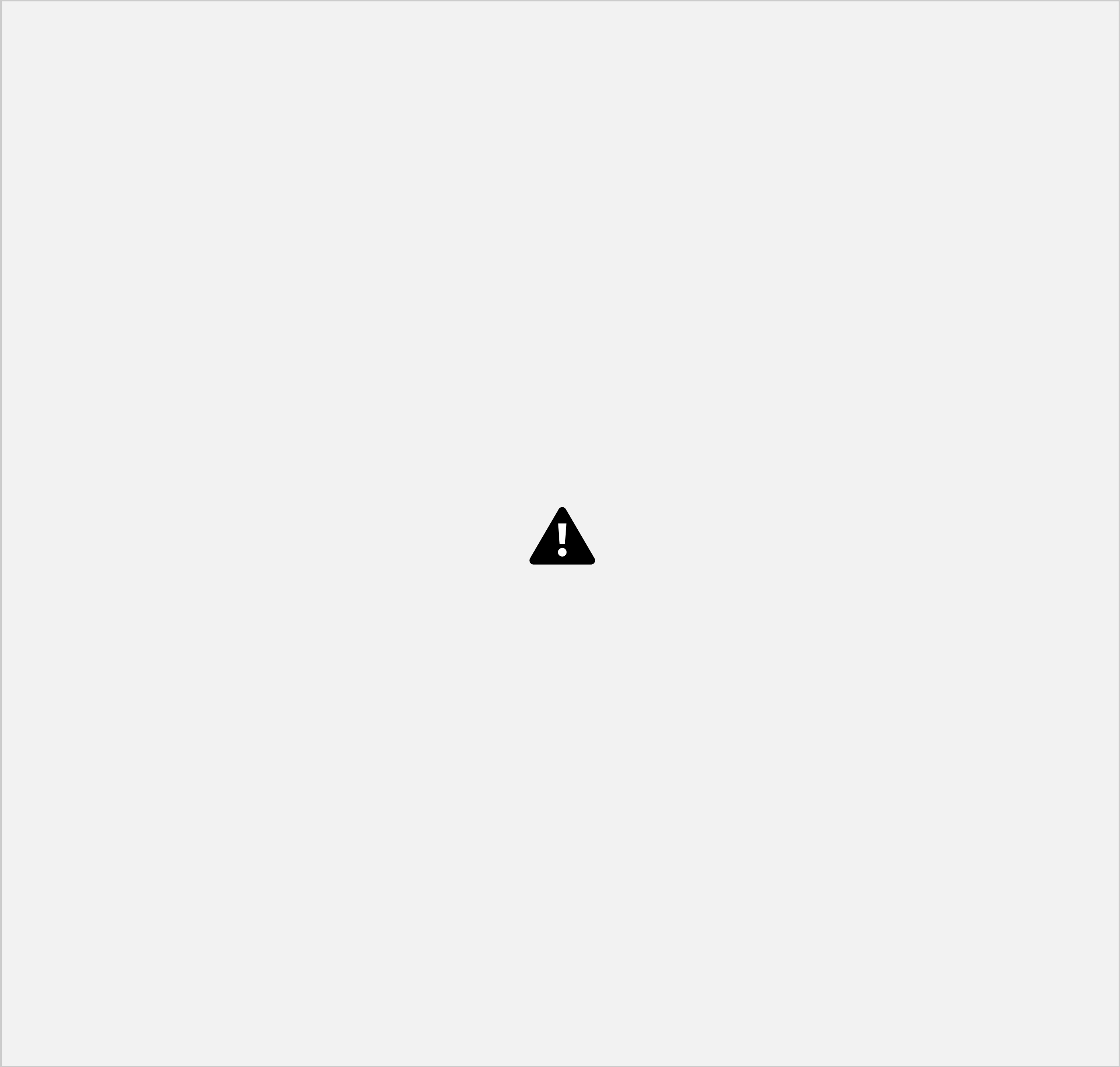




TASK 7. cloning a VM using GUI(PART1)

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)

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.

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

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

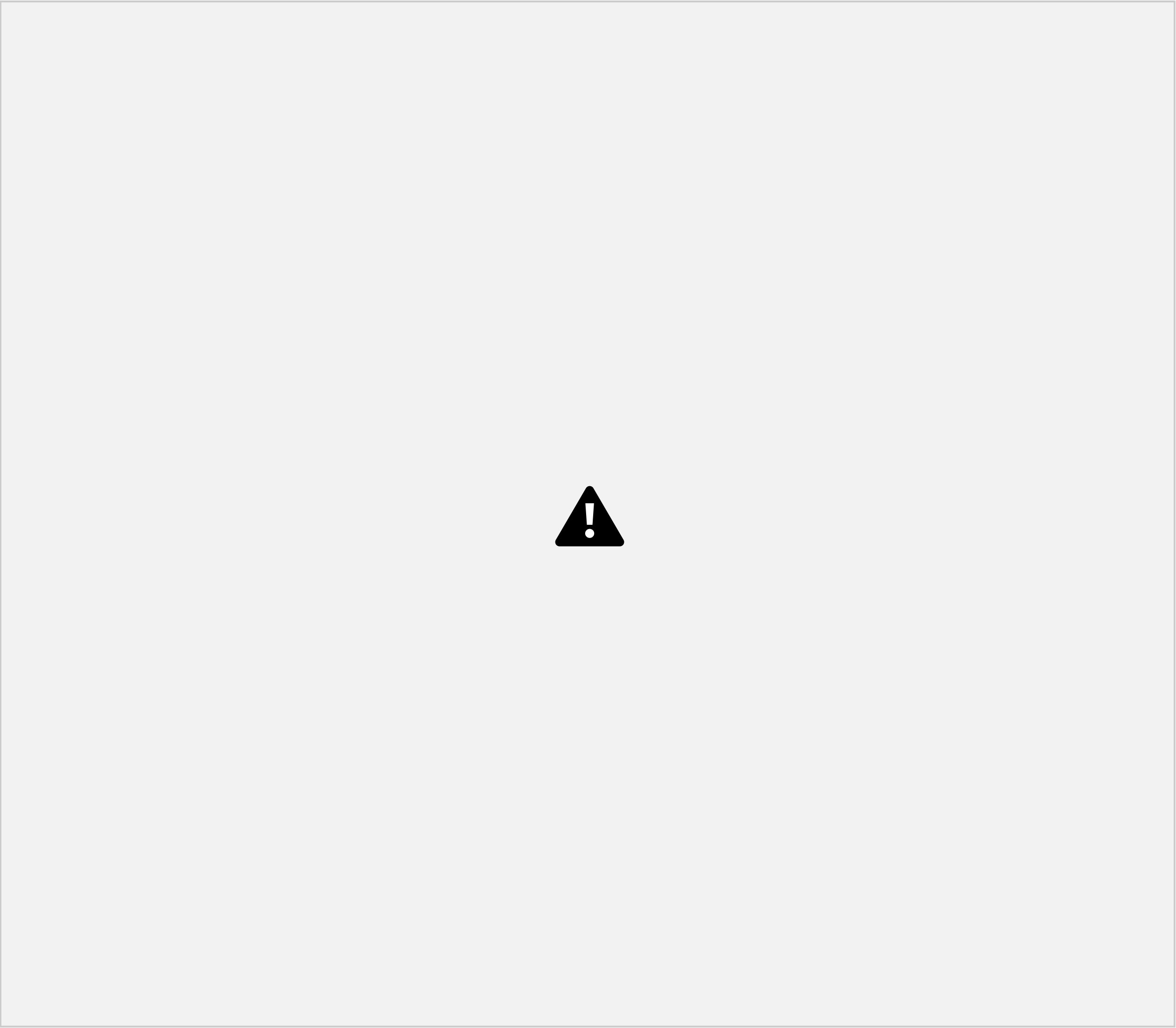
Too add a extra harddisk to a cloned virtual machine at first we need to start the Virtual machine with the help of : **Sudo virsh start newDebian-clone-cli**

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**.

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**



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**

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**

After that copy the image flie of my virtual machine from the img directory to home directory.command: **cd/var/lin/libvirt/images/**

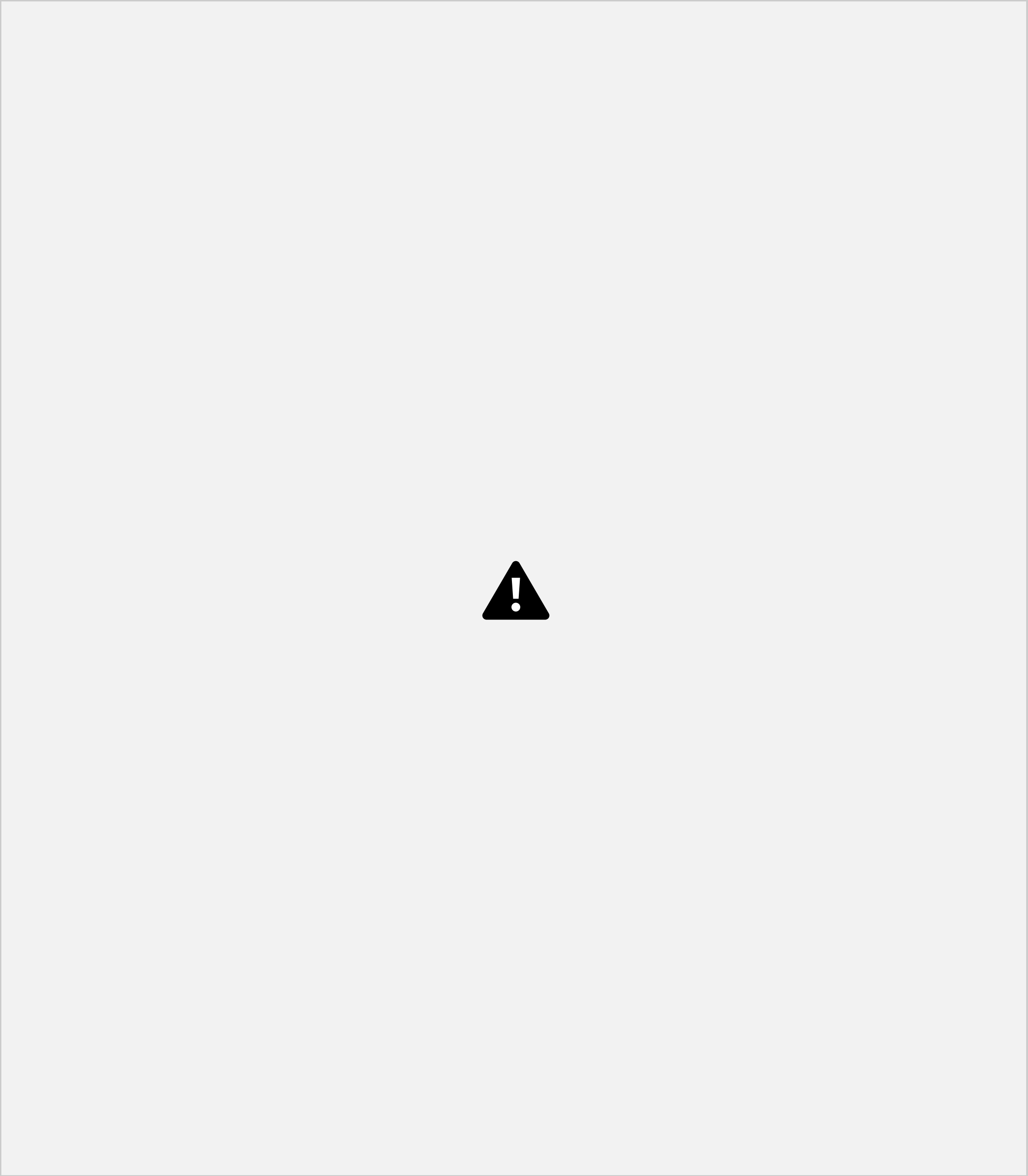
now for coping the file command**: sudo cp newDebian-clone-cli.qcow2 /home/habibun**

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 vesion so we choose Generic Linux using the migrating the shared OS img .



Finally the migrated VM runs in my friends hardware .

THE END