

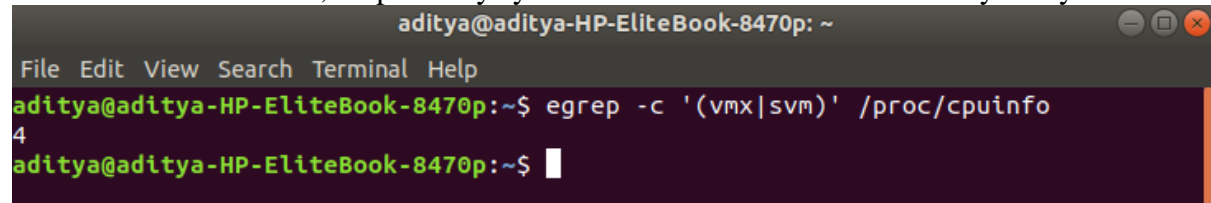
Aim: Installation and Configuration of virtualization using KVM.

Steps to install Virtualization using KVM

To create a virtual machine first we need to ensure that virtualization is enabled on our system. It is mandatory to create virtual machines. There are multiple ways to check if virtualization is enabled,

```
$ egrep -c '(vmx|svm)' /proc/cpuinfo
```

The above command /proc/cpuinfo gives information about the processor. The output of the command will be a number. The output number 1 or more than that represents that virtualization is enabled, output 0 says you need to enable virtualization on your system.

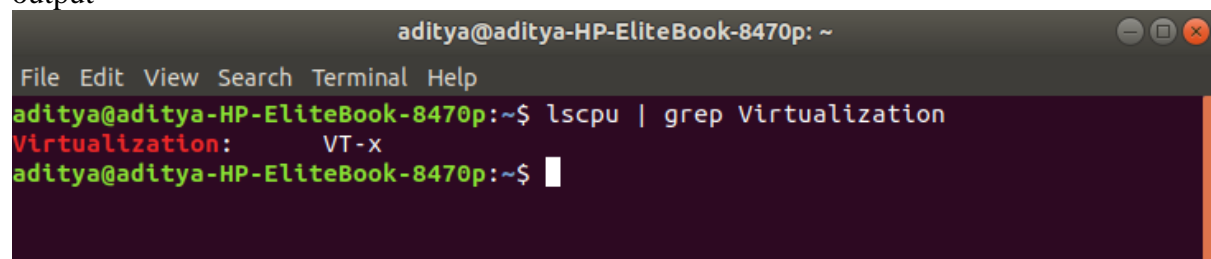


```
aditya@aditya-HP-EliteBook-8470p: ~  
File Edit View Search Terminal Help  
aditya@aditya-HP-EliteBook-8470p:~$ egrep -c '(vmx|svm)' /proc/cpuinfo  
4  
aditya@aditya-HP-EliteBook-8470p:~$
```

Virtualization Inability

```
$ lscpu | grep Virtualization
```

This command is used to check which type of virtualization your processor supports. If the system contains a CPU with Intel VT support, the above command will provide the following output



```
aditya@aditya-HP-EliteBook-8470p: ~  
File Edit View Search Terminal Help  
aditya@aditya-HP-EliteBook-8470p:~$ lscpu | grep Virtualization  
Virtualization:      VT-x  
aditya@aditya-HP-EliteBook-8470p:~$
```

Virtualization Type

Installing KVM on Ubuntu

```
$ sudo apt-get install virt-manager
```

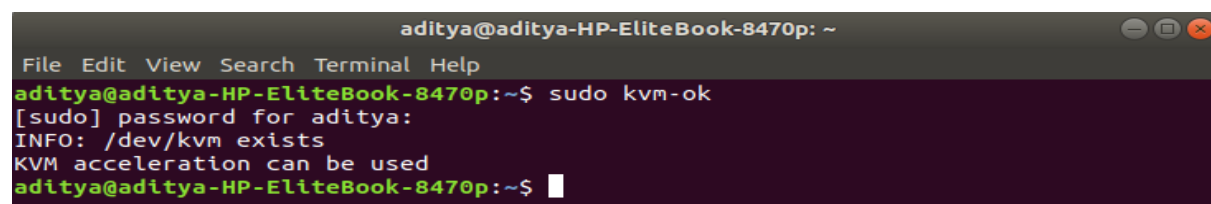
Now we know our system is capable of creating and running virtual machines, it's time to install tools which will create our virtual machines. To install KVM and other KVM dependencies such as virt-manager, bridge-utils enter command:

```
$ sudo apt -y install bridge-utils cpu-checker libvirt-clients libvirt-daemon qemu qemu-kvm
```

1. bridge-utils: The bridge-utils package contains a utility needed to create and manage bridge devices. This is useful in setting up networks for a hosted virtual machine (VM).
2. cpu-checker: Outputs the specifications of CPU
3. libvirt-clients: a toolkit to manage virtualization platforms/clients and hypervisors
4. qemu: A program that can run the operating system of the machine on different machines
5. qemu-kvm: Runs process using KVM module

All dependencies are installed now run command its time to check if KVM ins installed successfully:

```
$ sudo kvm-ok
```



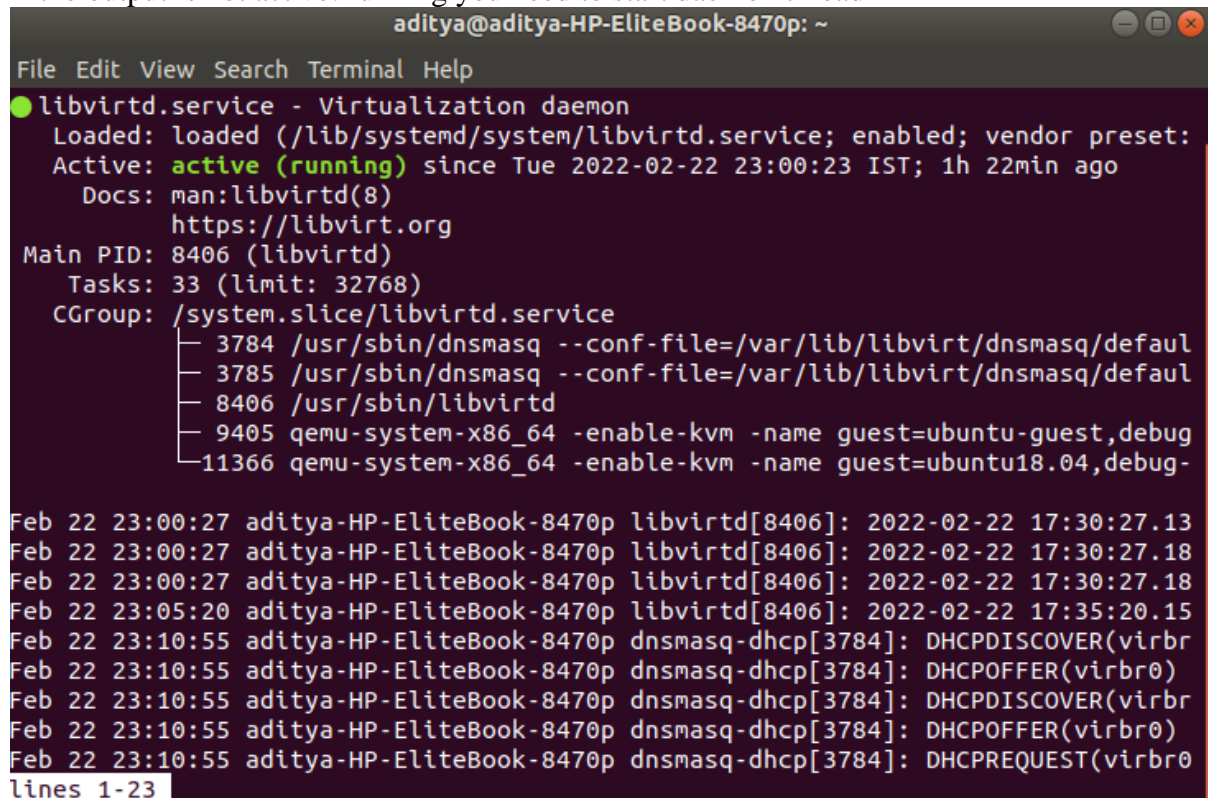
```
aditya@aditya-HP-EliteBook-8470p: ~  
File Edit View Search Terminal Help  
aditya@aditya-HP-EliteBook-8470p:~$ sudo kvm-ok  
[sudo] password for aditya:  
INFO: /dev/kvm exists  
KVM acceleration can be used  
aditya@aditya-HP-EliteBook-8470p:~$
```

Check if KVM is Installed properly

Also, we need to confirm if the virtualization daemon – libvirtd-daemon – is running, to do so enter the command.

```
$ sudo systemctl status libvirtd
```

If the output is not active: running you need to start daemon thread



```
aditya@aditya-HP-EliteBook-8470p: ~  
File Edit View Search Terminal Help  
● libvirtd.service - Virtualization daemon  
   Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; vendor preset:  
   Active: active (running) since Tue 2022-02-22 23:00:23 IST; 1h 22min ago  
     Docs: man:libvirtd(8)  
           https://libvirt.org  
   Main PID: 8406 (libvirtd)  
     Tasks: 33 (limit: 32768)  
    CGroup: /system.slice/libvirtd.service  
            └─ 3784 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default  
            └─ 3785 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/default  
            └─ 8406 /usr/sbin/libvirtd  
            └─ 9405 qemu-system-x86_64 -enable-kvm -name guest=ubuntu-guest,debug  
            └─ 11366 qemu-system-x86_64 -enable-kvm -name guest=ubuntu18.04,debug-  
  
Feb 22 23:00:27 aditya-HP-EliteBook-8470p libvirtd[8406]: 2022-02-22 17:30:27.13  
Feb 22 23:00:27 aditya-HP-EliteBook-8470p libvirtd[8406]: 2022-02-22 17:30:27.18  
Feb 22 23:00:27 aditya-HP-EliteBook-8470p libvirtd[8406]: 2022-02-22 17:30:27.18  
Feb 22 23:05:20 aditya-HP-EliteBook-8470p libvirtd[8406]: 2022-02-22 17:35:20.15  
Feb 22 23:10:55 aditya-HP-EliteBook-8470p dnsmasq-dhcp[3784]: DHCPDISCOVER(virbr  
Feb 22 23:10:55 aditya-HP-EliteBook-8470p dnsmasq-dhcp[3784]: DHCPOFFER(virbr0)  
Feb 22 23:10:55 aditya-HP-EliteBook-8470p dnsmasq-dhcp[3784]: DHCPDISCOVER(virbr  
Feb 22 23:10:55 aditya-HP-EliteBook-8470p dnsmasq-dhcp[3784]: DHCPOFFER(virbr0)  
Feb 22 23:10:55 aditya-HP-EliteBook-8470p dnsmasq-dhcp[3784]: DHCPREQUEST(virbr0  
lines 1-23
```

If the daemon thread is not running enter the following command to start the thread,

```
$ sudo systemctl enable --now libvirtd
```

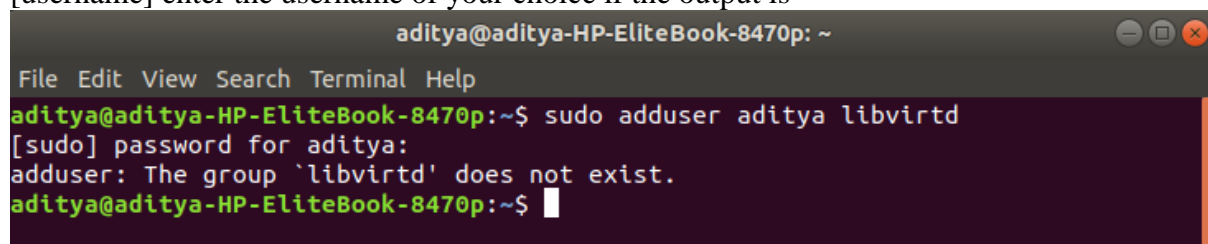
Adding a user to KVM

In this part, we are going to create a user for KVM. To prevent root user from using KVM and root user is only available when root user is a part of libvirt/libvirtd group.

To add a new user to KVM, use the following command,

```
sudo adduser [username] libvirtd
```

[username] enter the username of your choice if the output is



```
aditya@aditya-HP-EliteBook-8470p: ~  
File Edit View Search Terminal Help  
aditya@aditya-HP-EliteBook-8470p:~$ sudo adduser aditya libvirtd  
[sudo] password for aditya:  
adduser: The group `libvirtd' does not exist.  
aditya@aditya-HP-EliteBook-8470p:~$
```

Adding User to KVM

If your KVM is already a member of the non-root user and serves the same purpose as libvirtd group then you don't need to add yourself to the group.

Creating Virtual Machine

There are two ways to create a virtual machine

1. Using the command line
 2. Using the graphical interface
- **Create a Virtual Machine via Command Line**

virt-install is a command which is used to create virtual machines in Linux, following is the command which creates a VM.

```
sudo virt-install --name=ubuntu-guest --os-variant=ubuntu20.04 --vcpu=2 --ram=2048 --graphics none --location=[local path to iso file] --network bridge:vibr0
```

The above command creates a Ubuntu virtual machine with version 20.04 and the name ubuntu-guest.

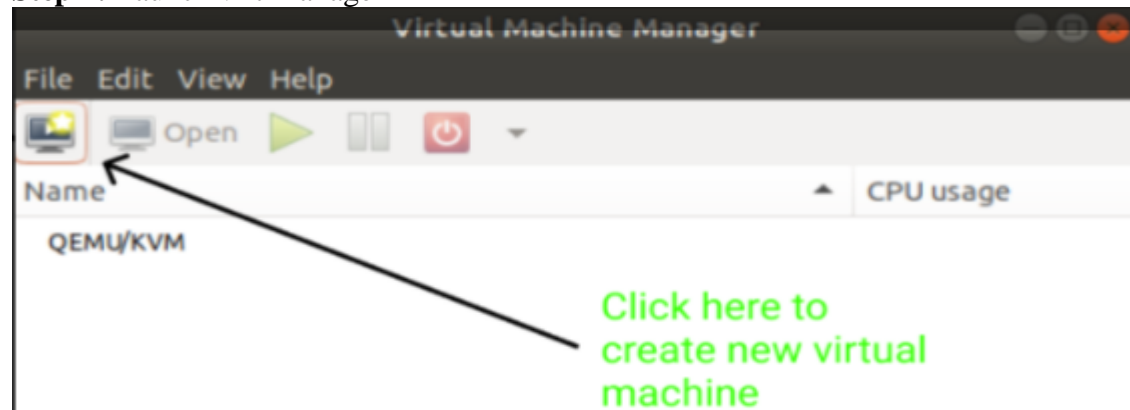
1. Name: specify the name of the virtual machine being created
2. vcpu: Number of virtual CPUs to configure for the guest.
3. Ram: Memory to allocate for guest instance in megabytes. According to your machine, you can specify the given memory of the VM.
4. Graphics spice: If no graphics option is specified, “virt-install” will default to ‘–graphics vnc’ if the display environment variable is set, otherwise ‘–graphics none’ is used.
5. Location: location of iso file on which virtual machine will be built. It can be the path to an ISO image, or to a CDROM device. It can also be a URL from which to fetch/access a minimal boot ISO image.
6. Network-bridge: Connect the guest to the host network, Connect to a bridge device in the host called “BRIDGE”.

Create a Virtual machine using a graphical interface

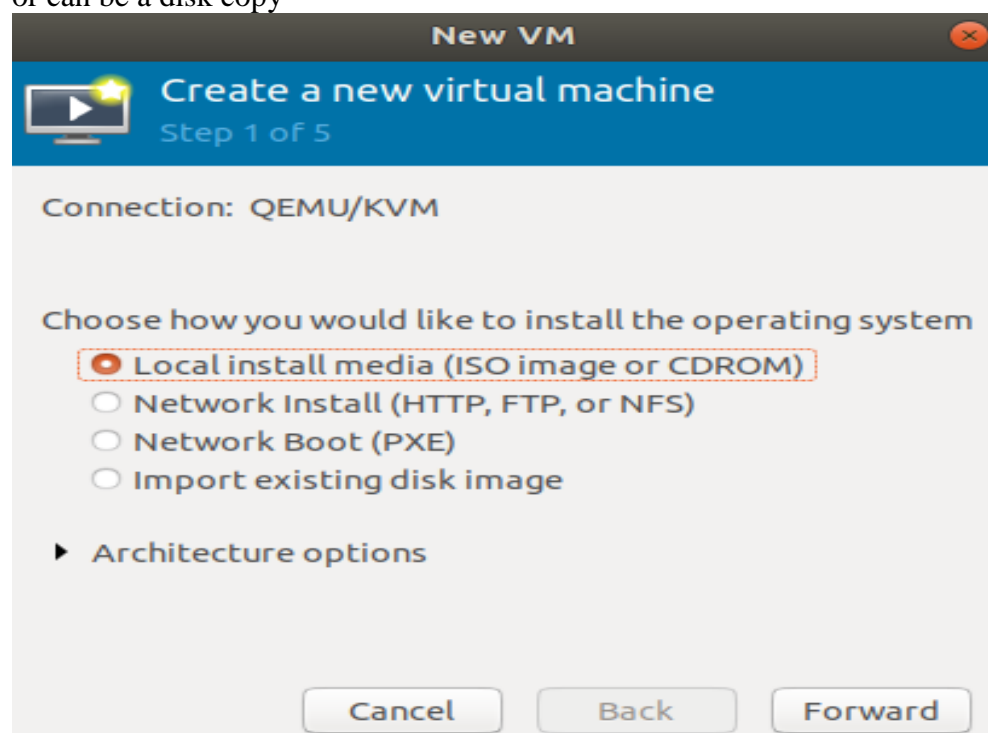
If you are not much familiar with the command line don't worry there's another way to create a virtual machine using a tool called virt-manager you can easily create virtual machines.

Steps to create VM using virt-manager,

Step 1: Launch virt-manager



Step 2: Choose installation media it can be an iso file of OS, you can install from the network or can be a disk copy



Step 3: Enter the path/URL of your file click next

New VM

Create a new virtual machine
Step 2 of 5

Locate your install media

☐ Use CDROM or DVD

No media detected (/dev/sr0) ▾

☒ Use ISO image:

/home/aditya/Downloads/ubuntu-18.04.6-desk ▾ Browse...

☒ Automatically detect operating system based on install media

OS type: Linux
Version: Ubuntu 18.04 LTS

Cancel Back Forward

Step 4: Enter memory and CPU requirements

New VM

Create a new virtual machine
Step 3 of 5

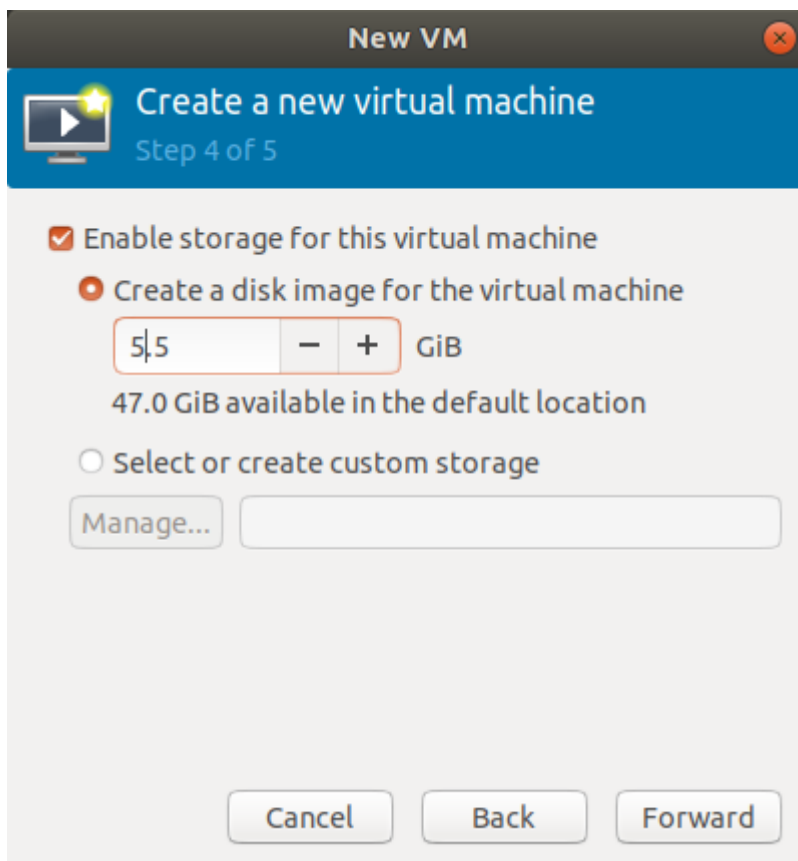
Choose Memory and CPU settings

Memory (RAM): 1024 - +
Up to 3801 MiB available on the host

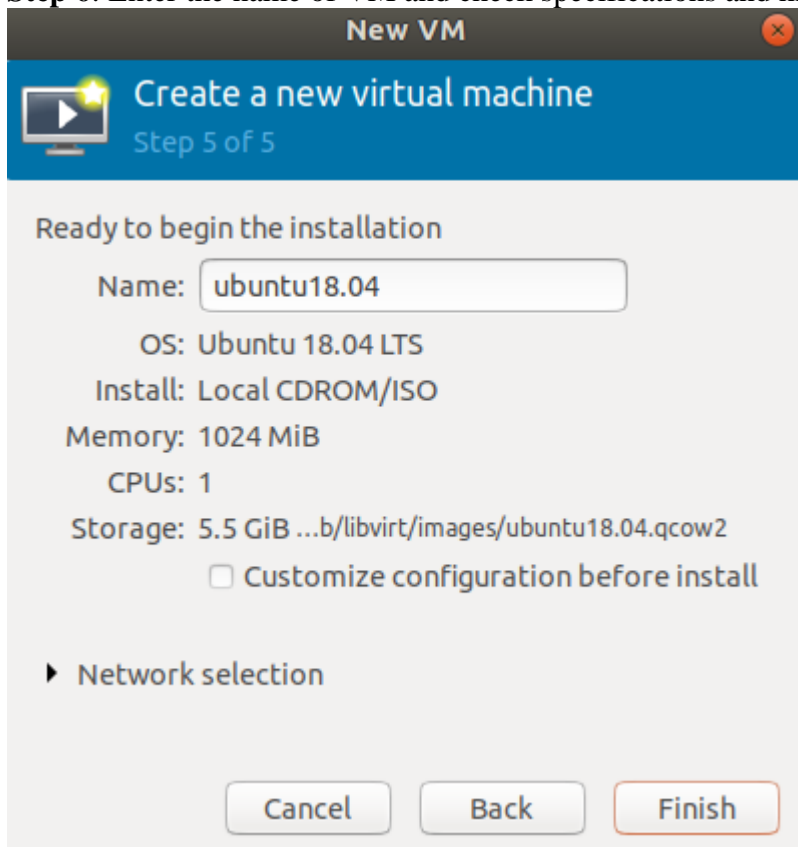
CPUs: 1 - +
Up to 4 available

Cancel Back Forward

Step 5: Enter required disk space



Step 6: Enter the name of VM and check specifications and hit the finish



Now the installations should be completed and the virtual machine should start running.

Conclusion: Hence we have successfully installed and Configured virtualization using KVM.