Aim -
To study and implement Hosted Virtualization using KVM.

Theory -
Kernel-based Virtual Machine (KVM) is an open source virtualization technology built into Linux®. Specifically, KVM lets you turn Linux into a hypervisor that allows a host machine to run multiple, isolated virtual environments called guests or virtual machines (VMs).
KVM is part of Linux. If you've got Linux 2.6.20 or newer, you've got KVM. KVM was first announced in 2006 and merged into the mainline Linux kernel version a year later. Because KVM is part of existing Linux code, it immediately benefits from every new Linux feature, fix, and advancement without additional engineering.

How does KVM work?
KVM converts Linux into a type-1 (bare-metal) hypervisor. All hypervisors need some operating system-level components—such as a memory manager, process scheduler, input/output (I/O) stack, device drivers, security manager, a network stack, and more—to run VMs. KVM has all these components because it's part of the Linux kernel. Every VM is implemented as a regular Linux process, scheduled by the standard Linux scheduler, with dedicated virtual hardware like a network card, graphics adapter, CPU(s), memory, and disks.

KVM features
KVM is part of Linux. Linux is part of KVM. Everything Linux has, KVM has too. But there are specific features that make KVM an enterprise's preferred hypervisor.

- Security
  KVM uses a combination of security-enhanced Linux (SELinux) and secure virtualization (sVirt) for enhanced VM security and isolation. SELinux establishes security boundaries around VMs. sVirt extends SELinux's capabilities, allowing Mandatory Access Control (MAC) security to be applied to guest VMs and preventing manual labeling errors.

- Storage
  KVM is able to use any storage supported by Linux, including some local disks and network-attached storage (NAS). Multipath I/O may be used to improve storage and provide redundancy. KVM also supports shared file systems so VM images may be shared by multiple hosts. Disk images support thin provisioning, allocating storage on demand rather than all up front.

- Hardware support
  KVM can use a wide variety of certified Linux-supported hardware platforms. Because hardware vendors regularly contribute to kernel development, the latest hardware features are often rapidly adopted in the Linux kernel.

- Memory management
  KVM inherits the memory management features of Linux, including non-uniform memory access and kernel same-page merging. The memory of a VM can be swapped, backed by large volumes for better performance, and shared or backed by a disk file.

- Live migration
  KVM supports live migration, which is the ability to move a running VM between physical hosts with no service interruption. The VM remains powered on, network connections remain active, and applications continue to run while the VM is relocated. KVM also saves a VM's current state so it can be stored and resumed later.

- Performance and scalability
  KVM inherits the performance of Linux, scaling to match demand load if the number of guest machines and requests increases. KVM allows the most demanding application workloads to be virtualized and is the basis for many enterprise virtualization setups, such as data centers and private clouds (via OpenStack®).

- Scheduling and resource control
  In the KVM model, a VM is a Linux process, scheduled and managed by the kernel. The Linux scheduler allows fine-grained control of the resources allocated to a Linux process and guarantees a quality of service for a particular process. In KVM, this includes the completely fair scheduler, control groups, network name spaces, and real-time extensions.

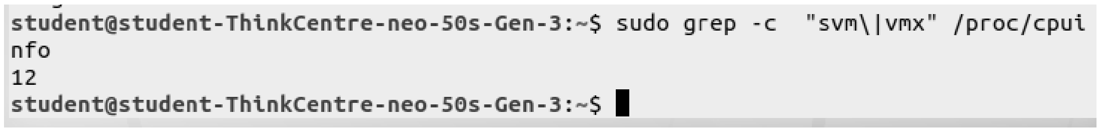
- Lower latency and higher prioritization
  The Linux kernel features real-time extensions that allow VM-based apps to run at lower latency with better prioritization (compared to bare metal). The kernel also divides processes that require long computing times into smaller components, which are then scheduled and processed accordingly.

<u>Procedure -</u>

The Steps to Create and run Virtual machines in KVM are as follows-

1. Check whether the CPU has hardware virtualization support.

KVM only works if your CPU has hardware virtualization support – either Intel VT-x or AMD-V. To determine whether your CPU includes these features, run the following command:
sudo grep -c "svm\|vmx" /proc/cpuinfo

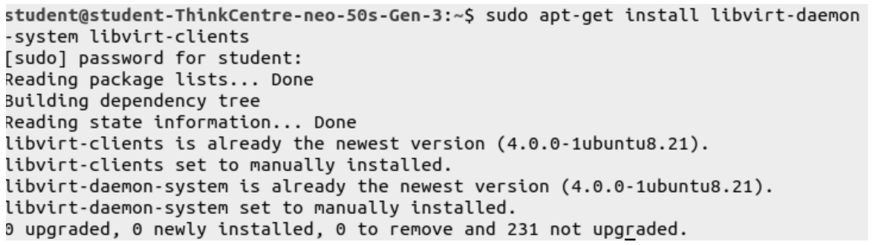


```
student@student-ThinkCentre-neo-50s-Gen-3:~$ sudo grep -c  "svm\|vmx" /proc/cpui
nfo
12
student@student-ThinkCentre-neo-50s-Gen-3:~$ 
```

A 0 indicates that your CPU doesn't support hardware virtualization, while a 1 or more indicates
that it does.

2. Install KVM and supporting packages.

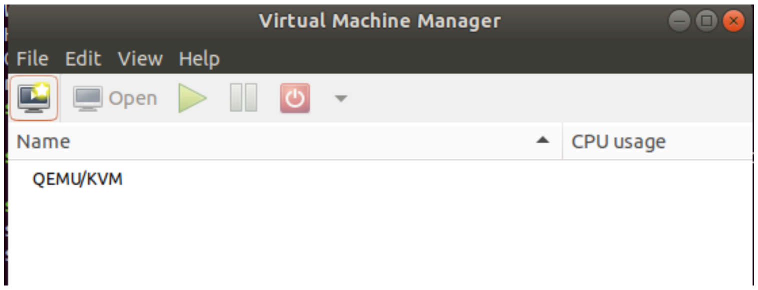
Virt-Manager is a graphical application for managing your virtual machines.you can use the kvm command directly, but libvirt and Virt-Manager simplify the process.
sudo apt-get install qemu-kvm libvirt-daemon bridge-utils virt-manager



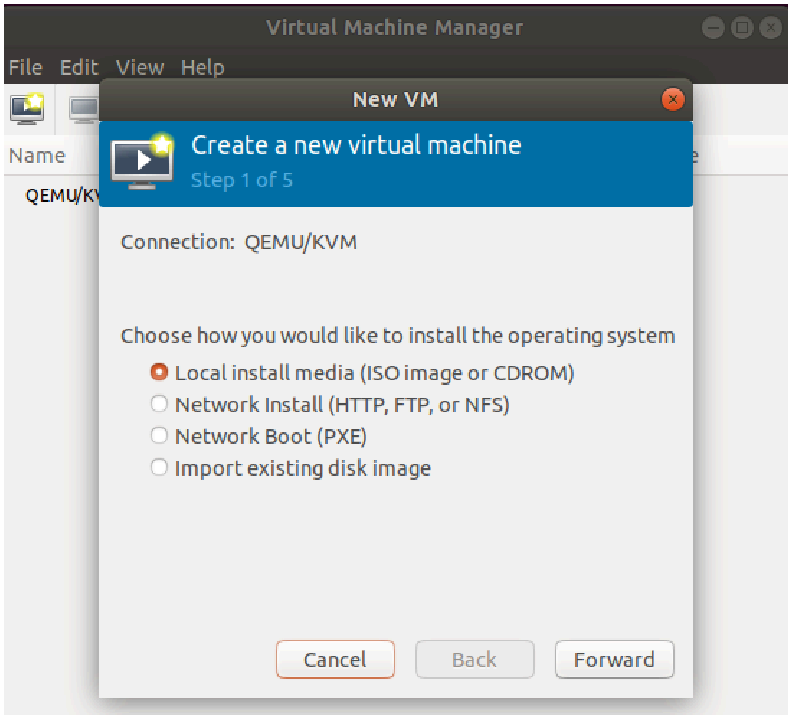
```
student@student-ThinkCentre-neo-50s-Gen-3:~$ sudo apt-get install libvirt-daemon
-system libvirt-clients
[sudo] password for student:
Reading package lists... Done
Building dependency tree
Reading state information... Done
libvirt-clients is already the newest version (4.0.0-1ubuntu8.21).
libvirt-clients set to manually installed.
libvirt-daemon-system is already the newest version (4.0.0-1ubuntu8.21).
libvirt-daemon-system set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 231 not upgraded.
```

3. Start virtual manager with -
   sudo virt-manager

4. Click on the computer icon on the upper-left corner of the window.



5. In the dialogue box that opens, select the option to install the VM using the ISO image. Then click on Forward.

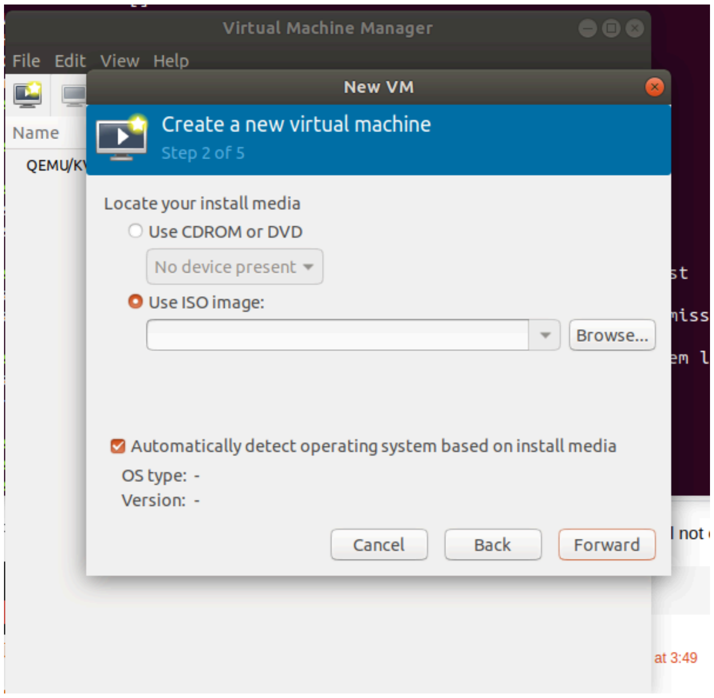


Connection: QEMU/KVM

Choose how you would like to install the operating system

- [✓ Local install media (ISO image or CDROM)]
- Network Install (HTTP, FTP, or NFS)
- Network Boot (PXE)
- Import existing disk image

6. Click on Browse Local and navigate to the path where you stored the ISO file you wish to install.

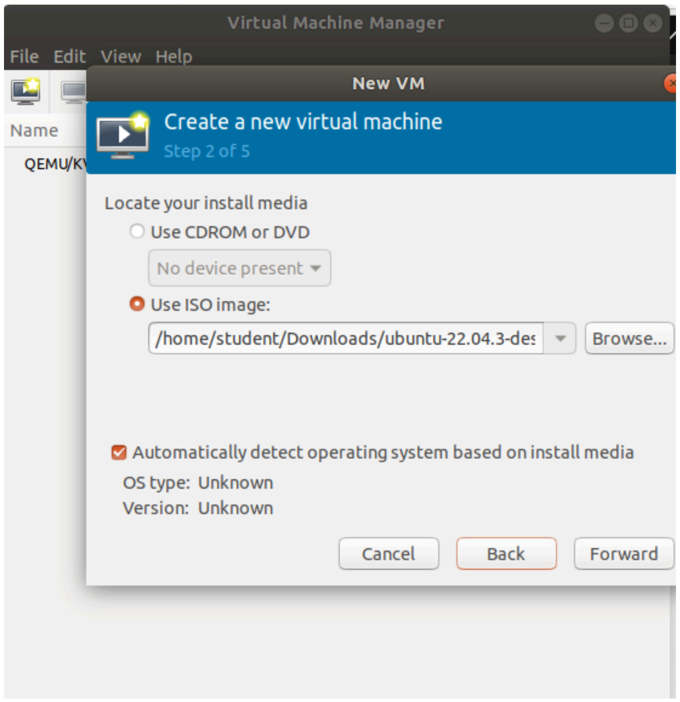


Locate your install media

- Use CDROM or DVD
  No device present
- [✓ Use ISO image:]

[✓] Automatically detect operating system based on install media

OS type: -
Version: -



Create a new virtual machine
Step 2 of 5

Locate your install media

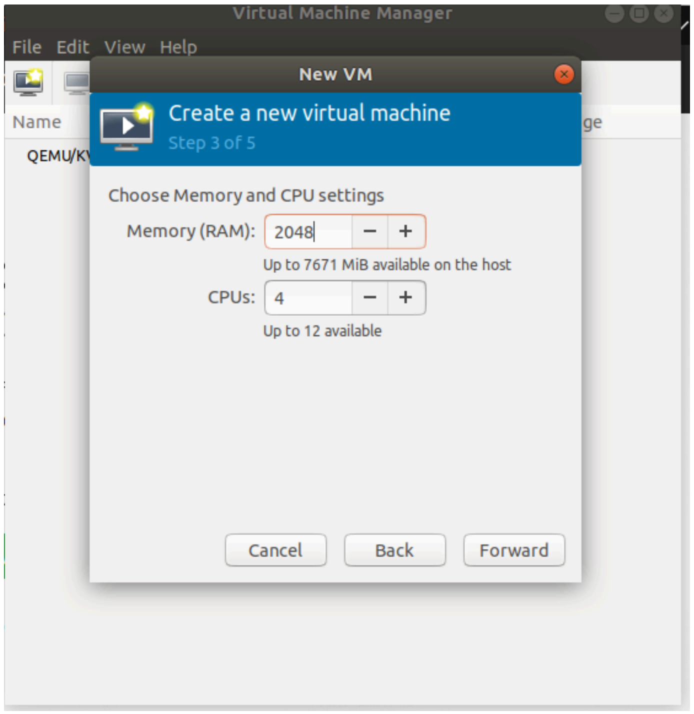
- Use CDROM or DVD: No device present
- [✓ Use ISO image:] /home/student/Downloads/ubuntu-22.04.3-des

Browse...

[✓] Automatically detect operating system based on install media

OS type: Unknown
Version: Unknown

7. Enter the amount of RAM and the number of CPUs you wish to allocate to the virtual machine and proceed to the next step.

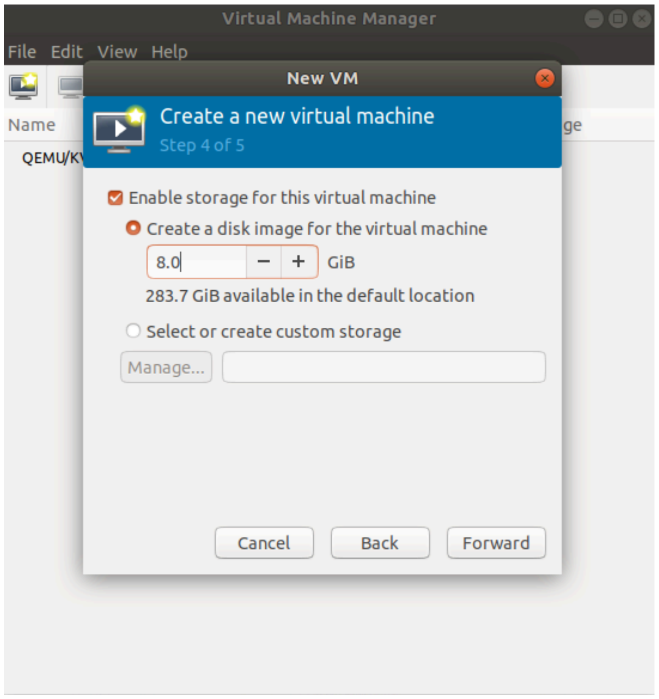


Create a new virtual machine
Step 3 of 5

Choose Memory and CPU settings

Memory (RAM): 2048
Up to 7671 MiB available on the host

CPUs: 4
Up to 12 available



Enable storage for this virtual machine [✓]

Create a disk image for the virtual machine [✓]

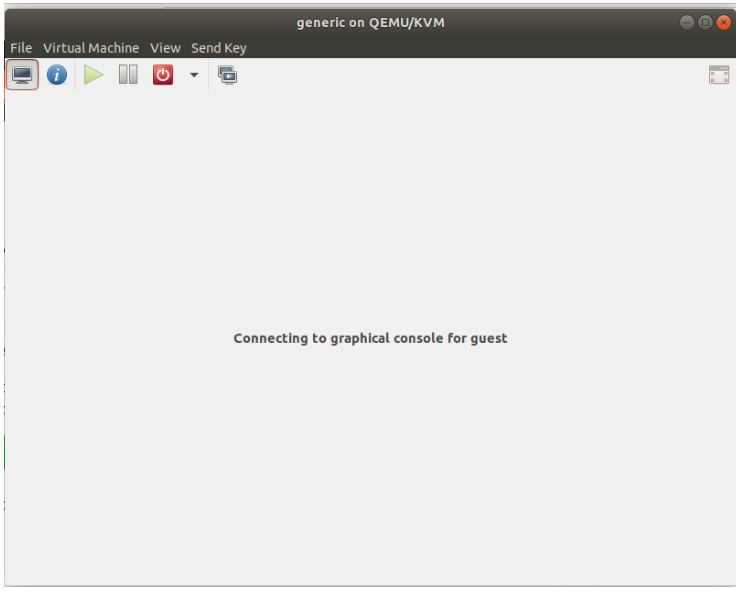
8.0 GiB

283.7 GiB available in the default location

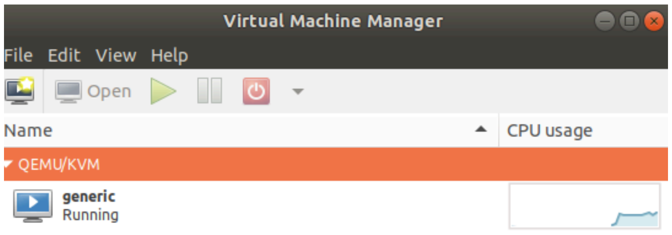
Select or create custom storage

Manage...

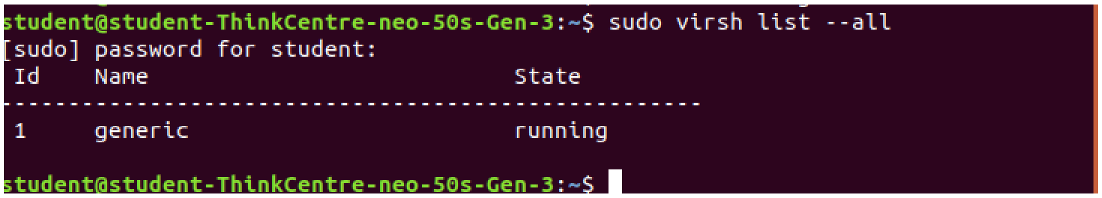
8. Now wait for the VM to start and you can see it in the VM Manager window.



Connecting to graphical console for guest



9. To check the status of the VMs on the terminal run -
   sudo virsh list –all



```
student@student-ThinkCentre-neo-50s-Gen-3:~$ sudo virsh list --all
[sudo] password for student:
 Id    Name                           State
----------------------------------------------------
 1     generic                        running

student@student-ThinkCentre-neo-50s-Gen-3:~$
```

Conclusion -

Thus we have successfully learned about KVM and implemented Hosted Virtualization using KVM.