BSCS602

Pratik Patil

Roll No:58

KERALEEYA SAMAJAM(REGD.) DOMBIVLI’S
MODEL COLLEGE
EMPOWERED AUTONOMOUS

PRACTICAL 6

Date:

Aim: Installation and configuration of virtualization using KVM.

Objectives: From this experiment, the student will be able to,
- Understand the concept of virtualization.
- Understand KVM architecture and its configuration.

Outcomes: The learner will be able to
- Analyze user models and develop user centric interfaces.
- To analyze local and global impact of computing on individuals, organizations and society.
- To engage in life-long learning development and higher studies.
- To understand, identify,analyze and design the problem, implement and validate the solutions including both hardware and software.

Hardware / Software Required: Ubuntu operating system, open-source software KVM, Internet.

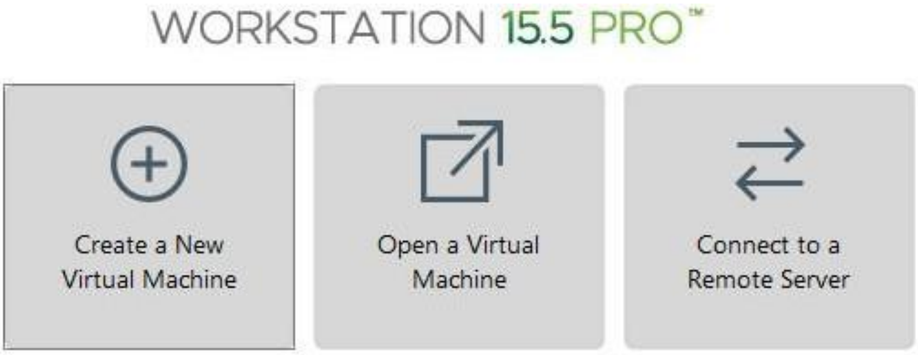
Theory:

Virtualization is software that separates physical infrastructures to create various dedicated resources.It is the fundamental technology that powers cloud computing. The technology behind virtualization is known as a virtual machine monitor(VMM) or virtual manager, which separates compute environments from the actual physical infrastructure.

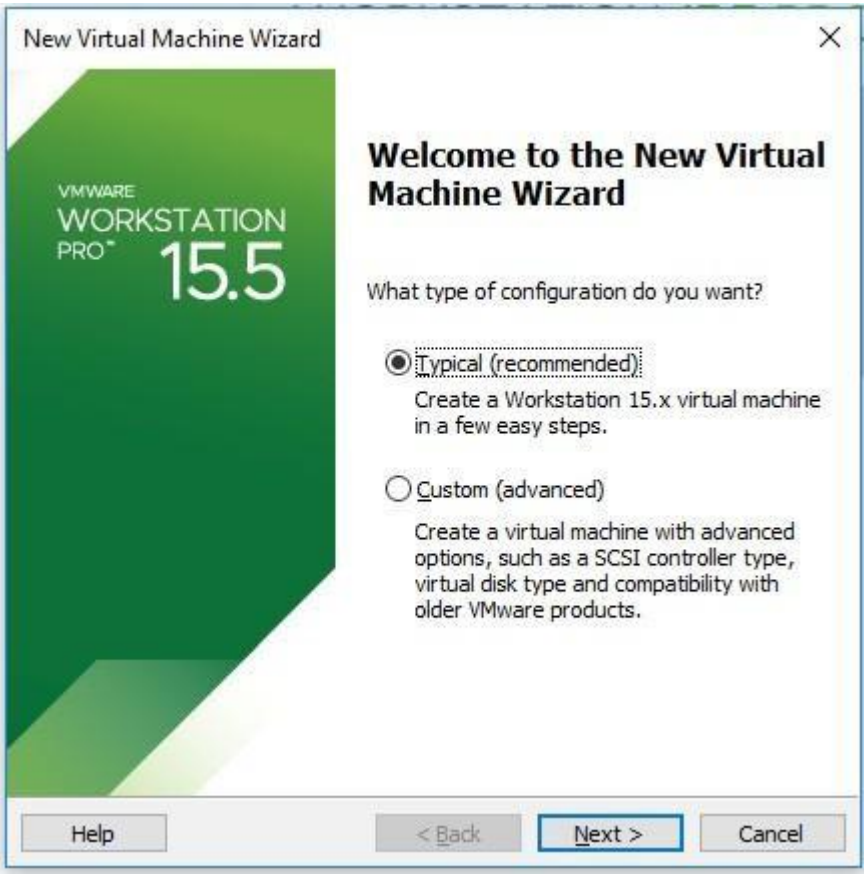
Virtualizations make servers,workstations,storage and other systems independent of the physical hardware layer. This is done by installing a Hypervisor on top of the hardware layer, where the systems are then installed.

There are three areas of IT where virtualization is making headboards , network virtualization,storage virtualization and server virtualization:
- Network virtualization is a method of combining the available resources in a network by splitting up the available bandwidth into channels , each of which is independent from the others, and each of which can be assigned(or reassigned) to a particular server or device in real time. The idea is that virtualization disguises the true complexity of the network by separating it into manageable parts , much like your partitioned hard drive makes it easier to manage your files.

Step 1: Open VMware to create a new virtual machine and select the option “Create New Virtual Machine”.



Step 2: Select the “Typical(recommended)” option and click on Next.


New Virtual Machine Wizard

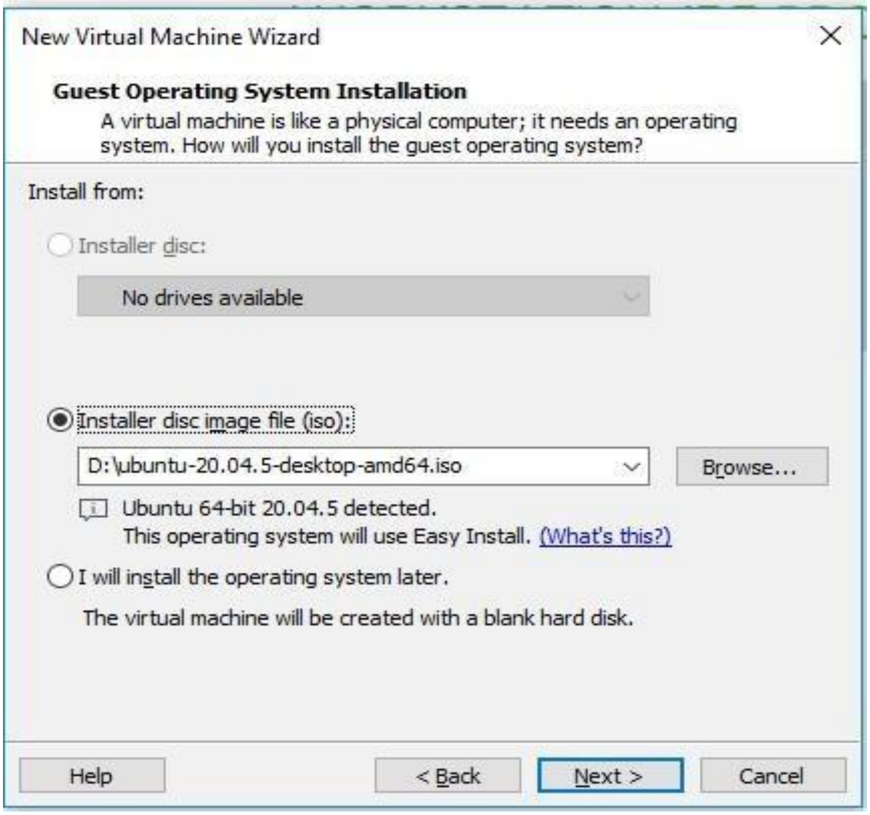
Welcome to the New Virtual Machine Wizard

What type of configuration do you want?

● Typical (recommended)
Create a Workstation 15.x virtual machine in a few easy steps.

○ Custom (advanced)
Create a virtual machine with advanced options, such as a SCSI controller type, virtual disk type and compatibility with older VMware products.

Step 3: The following window will appear click on Next and proceed.



New Virtual Machine Wizard

Guest Operating System Installation

A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?

Install from:

○ Installer disc:

No drives available

● Installer disc image file (iso):

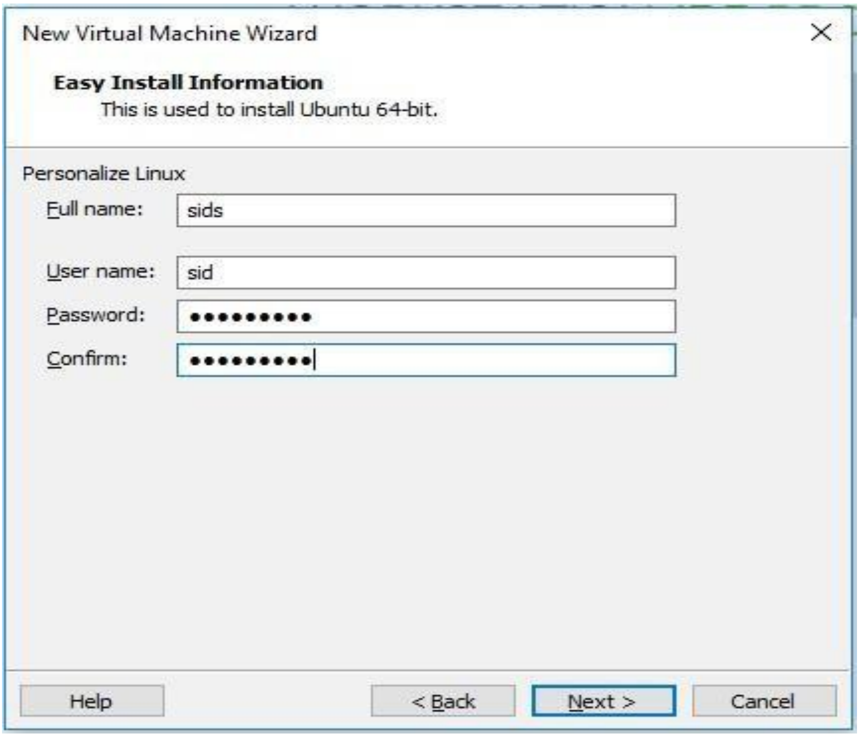
D:\ubuntu-20.04.5-desktop-amd64.iso

Ubuntu 64-bit 20.04.5 detected.
This operating system will use Easy Install. (What's this?)

○ I will install the operating system later.

The virtual machine will be created with a blank hard disk.

Step 4: Fill the following fields such as the Full name,Username and Password . Click on Next.



New Virtual Machine Wizard

Easy Install Information

This is used to install Ubuntu 64-bit.

Personalize Linux

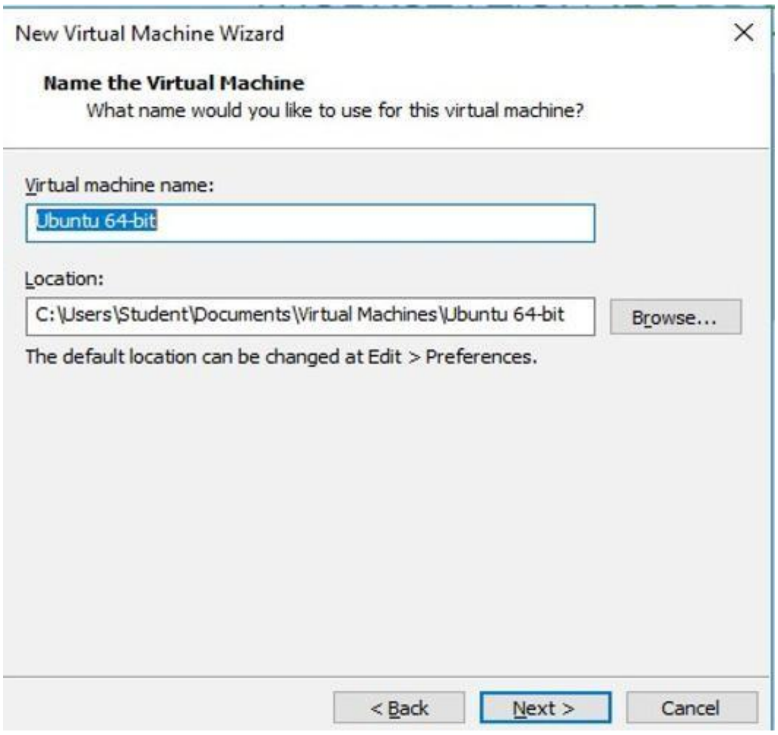
Full name: sids

User name: sid

Password: ••••••••••

Confirm: ••••••••••

Step 5: The following fields show the Name and Location where the Virtual Machine Is Stored



New Virtual Machine Wizard

Name the Virtual Machine

What name would you like to use for this virtual machine?

Virtual machine name:

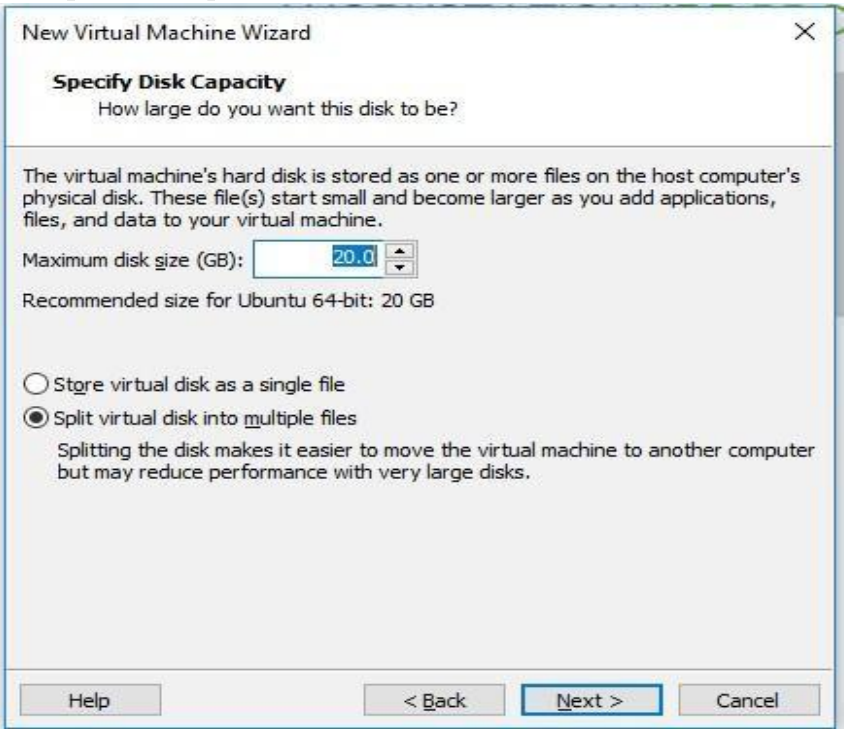
Ubuntu 64-bit

Location:

C:\Users\Student\Documents\Virtual Machines\Ubuntu 64-bit

The default location can be changed at Edit > Preferences.

Step 6: Keep the maximum disk size as 20.0 GB and select the “Split virtual disk into multiple files” option.



New Virtual Machine Wizard

Specify Disk Capacity

How large do you want this disk to be?

The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.

Maximum disk size (GB): 20.0

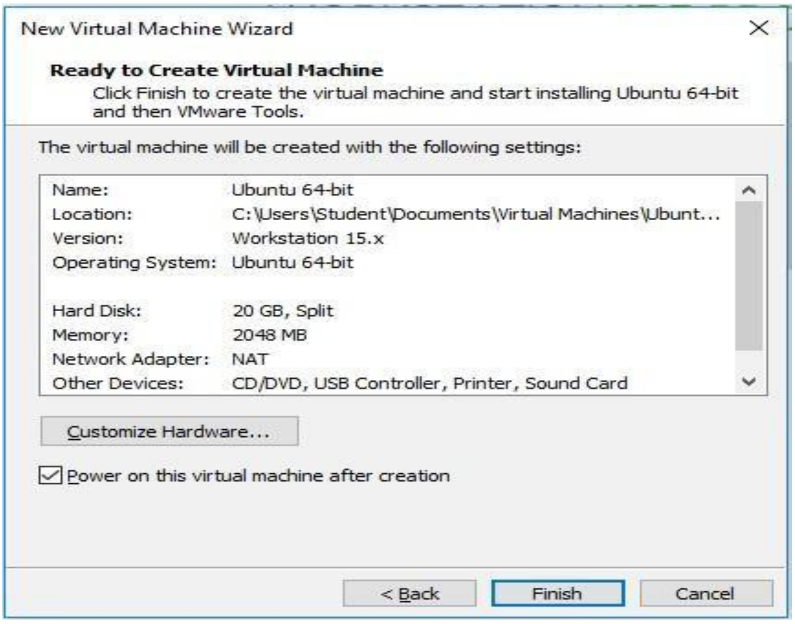
Recommended size for Ubuntu 64-bit: 20 GB

○ Store virtual disk as a single file

● Split virtual disk into multiple files

Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

Step 7: Click on the option of “Customize Hardware”.



New Virtual Machine Wizard

Ready to Create Virtual Machine

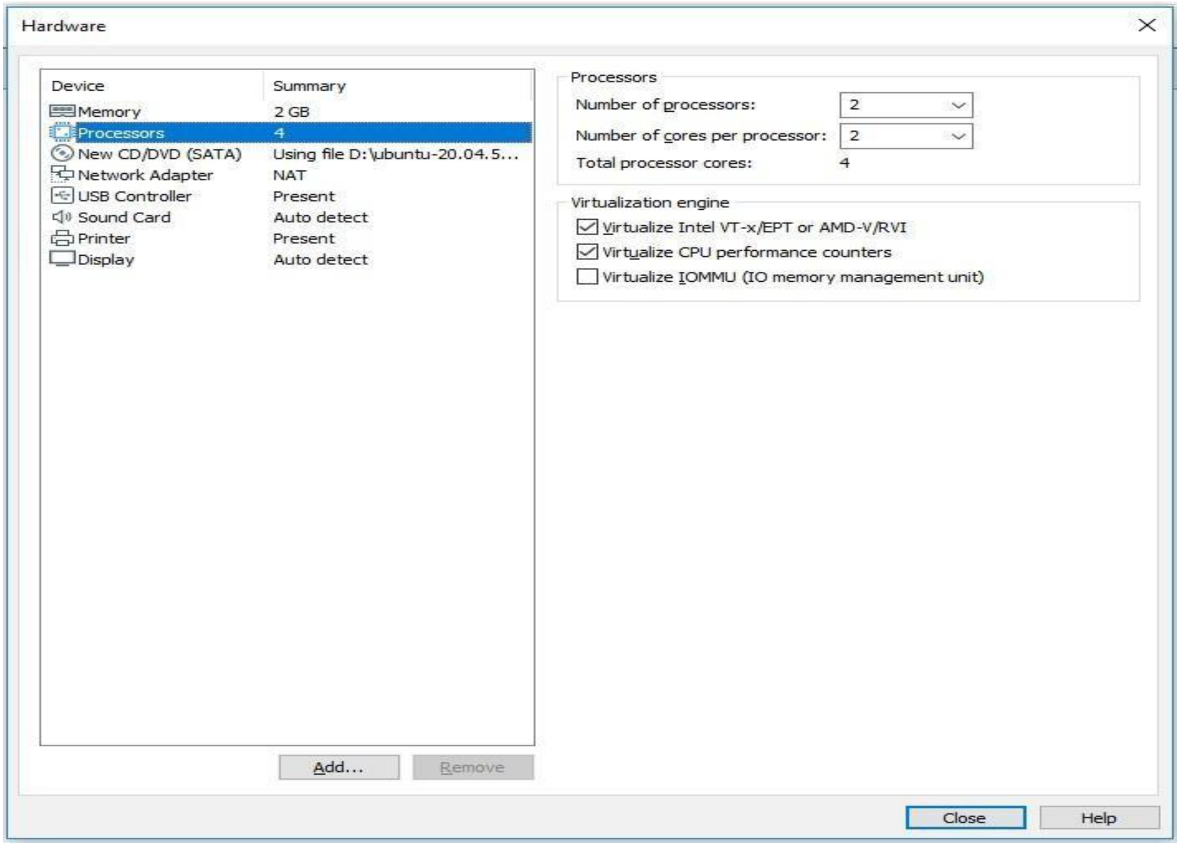
Click Finish to create the virtual machine and start installing Ubuntu 64-bit and then VMware Tools.

The virtual machine will be created with the following settings:

Name:	Ubuntu 64-bit
Location:	C:\Users\Student\Documents\Virtual Machines\Ubunt...
Version:	Workstation 15.x
Operating System:	Ubuntu 64-bit
Hard Disk:	20 GB, Split
Memory:	2048 MB
Network Adapter:	NAT
Other Devices:	CD/DVD, USB Controller, Printer, Sound Card

☑ Power on this virtual machine after creation

Step 8: Click on processors and keep the value of Number of processors and Number of core processors as 2. And make sure that the below two checkboxes are ticked . Click on close and then on finish.



Hardware

Device	Summary
Memory	2 GB
Processors	4
New CD/DVD (SATA)	Using file D:\ubuntu-20.04.5...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Processors

Number of processors: 2

Number of cores per processor: 2

Total processor cores: 4

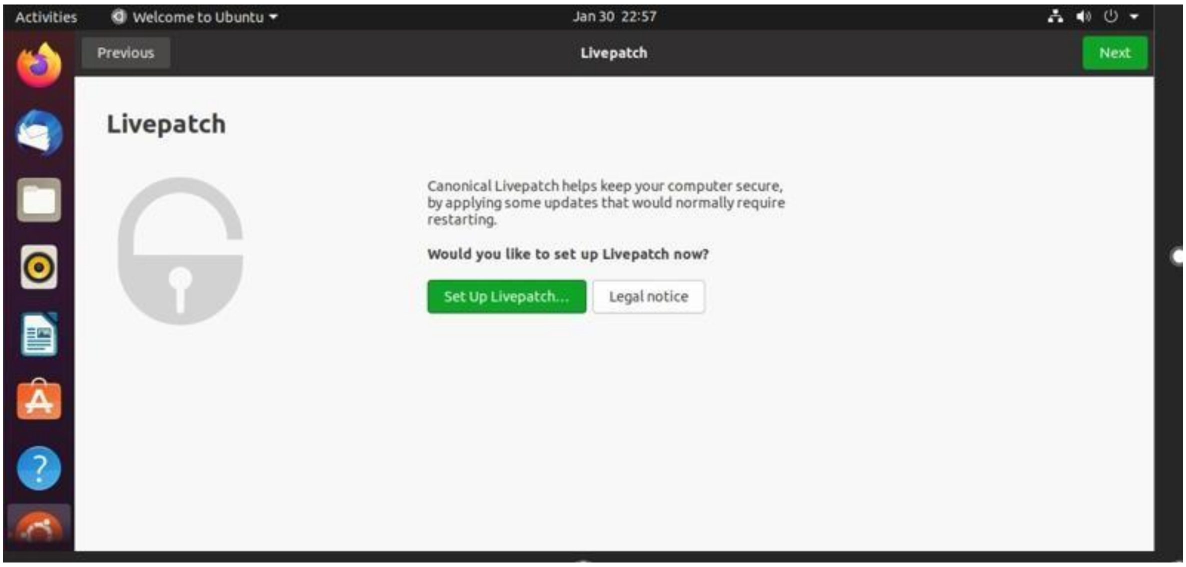
Virtualization engine

☑ Virtualize Intel VT-x/EPT or AMD-V/RVI

☑ Virtualize CPU performance counters

☐ Virtualize IOMMU (IO memory management unit)

Step 9: The following window will appear once the virtual machine starts, go to the command prompt section



Livepatch

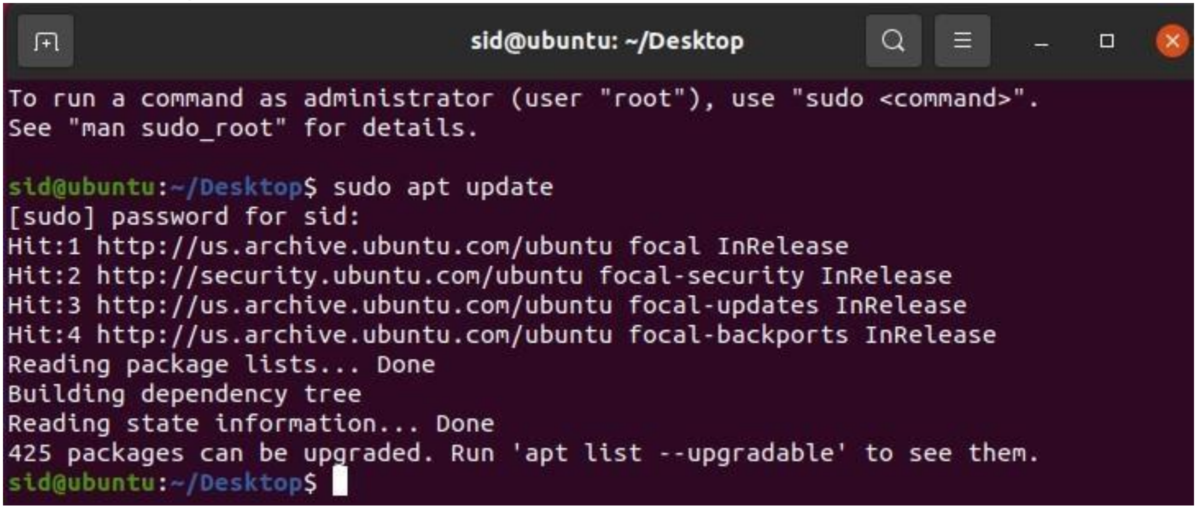
Canonical Livepatch helps keep your computer secure, by applying some updates that would normally require restarting.

Would you like to set up Livepatch now?

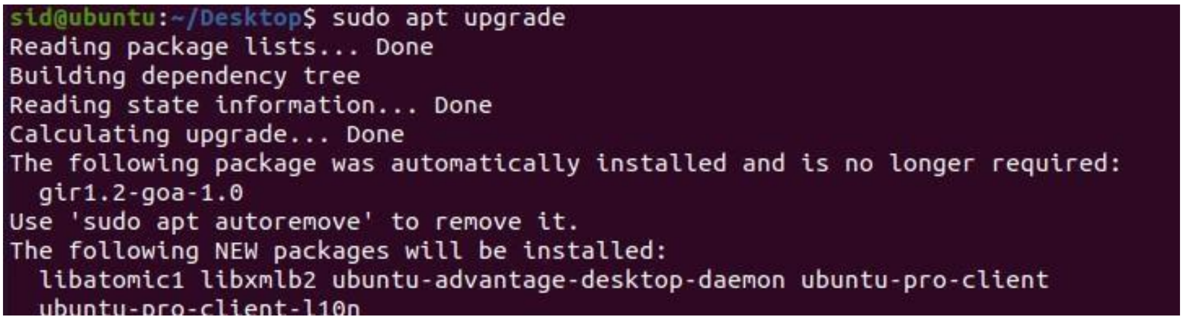
Step 10 : Update Ubuntu by executing following commands

● Sudo apt update

● Sudo apt upgrade

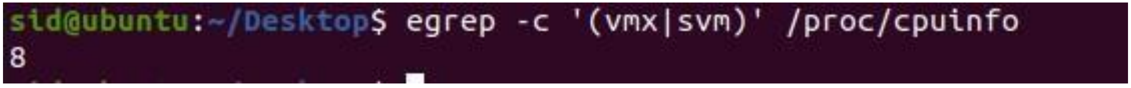


```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

sid@ubuntu:~/Desktop$ sudo apt update
[sudo] password for sid:
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
425 packages can be upgraded. Run 'apt list --upgradable' to see them.
sid@ubuntu:~/Desktop$ 
```



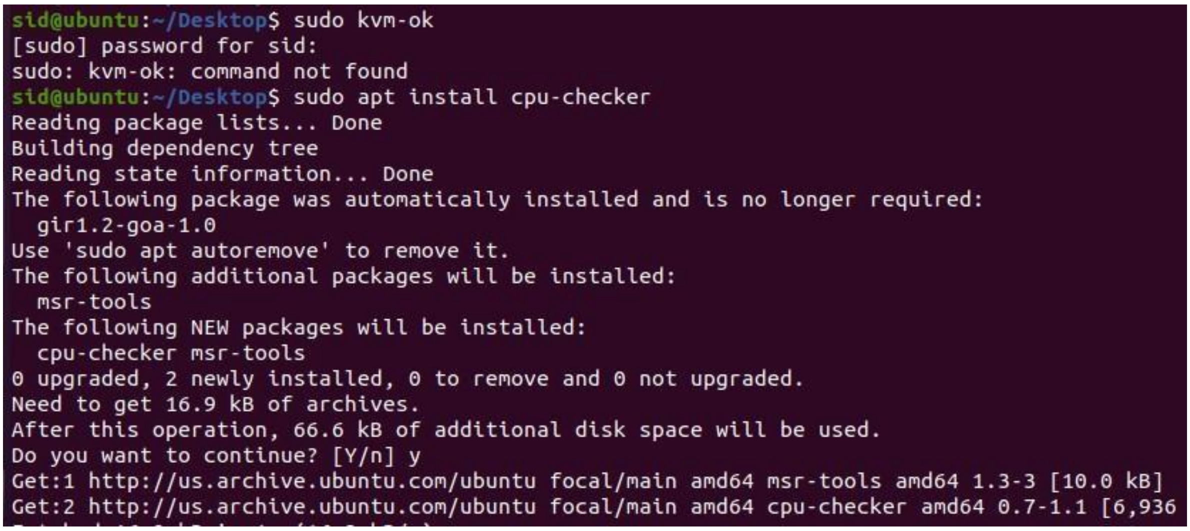
```
sid@ubuntu:~/Desktop$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  gir1.2-goa-1.0
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  libatomic1 libxmlb2 ubuntu-advantage-desktop-daemon ubuntu-pro-client
  ubuntu-pro-client-l10n
```

Step 11: Check the virtualization support for Ubuntu by executing the following command. This command returns a numeric value ,if it returns 0 it means that our processor is not capable of running KVM. Any other value indicates its KVM.

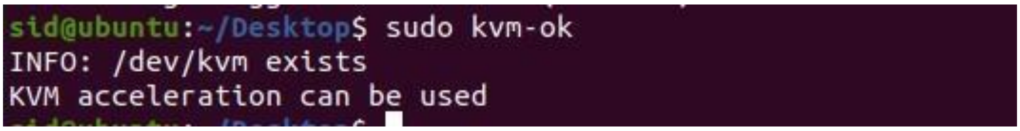


```
sid@ubuntu:~/Desktop$ egrep -c '(vmx|svm)' /proc/cpuinfo
8
```

Step 12: Check if the system is accelerated by using the command “sudo kvm-ok”. If kvm-ok returns an error then install cpu-checker to resolve the issue .Once the installation is done rerun the command “sudo kvm-ok”.

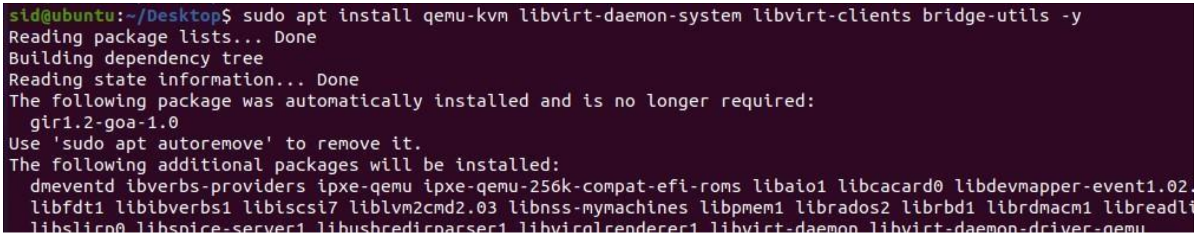


```
sid@ubuntu:~/Desktop$ sudo kvm-ok
[sudo] password for sid:
sudo: kvm-ok: command not found
sid@ubuntu:~/Desktop$ sudo apt install cpu-checker
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  gir1.2-goa-1.0
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  msr-tools
The following NEW packages will be installed:
  cpu-checker msr-tools
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 16.9 kB of archives.
After this operation, 66.6 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu focal/main amd64 msr-tools amd64 1.3-3 [10.0 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal/main amd64 cpu-checker amd64 0.7-1.1 [6,936
```



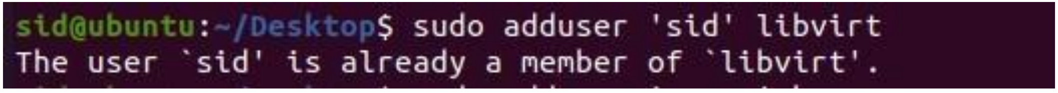
```
sid@ubuntu:~/Desktop$ sudo kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
```

Step 13: Install KVM package by executing the following command.



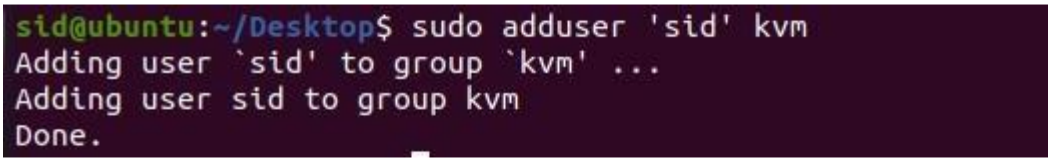
```
sid@ubuntu:~/Desktop$ sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  gir1.2-goa-1.0
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  dmeventd ibverbs-providers ipxe-qemu ipxe-qemu-256k-compat-efi-roms libaio1 libcacard0 libdevmapper-event1.02.
  libfdt1 libibverbs1 libiscsi7 liblvm2cmd2.03 libnss-mymachines libpmem1 librados2 librbd1 librdmacm1 libreadli
  libslirp0 libspice-server1 libusbredirparser1 libvirglrenderer1 libvirt-daemon libvirt-daemon-driver-qemu
```

Step 14: Only members of libvirt and KVM usergroups can run virtual machine . If you want a specific user to run a virtual machine then we have to add that user group with the following command.



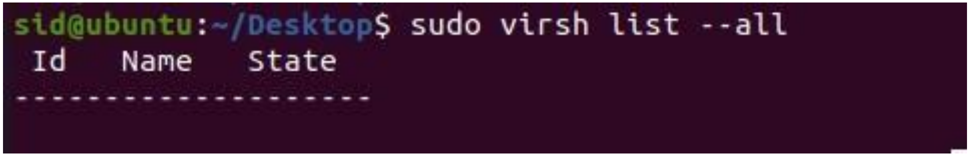
```
sid@ubuntu:~/Desktop$ sudo adduser 'sid' libvirt
The user `sid' is already a member of `libvirt'.
```

Step 15: Do the same thing with the KVM group.



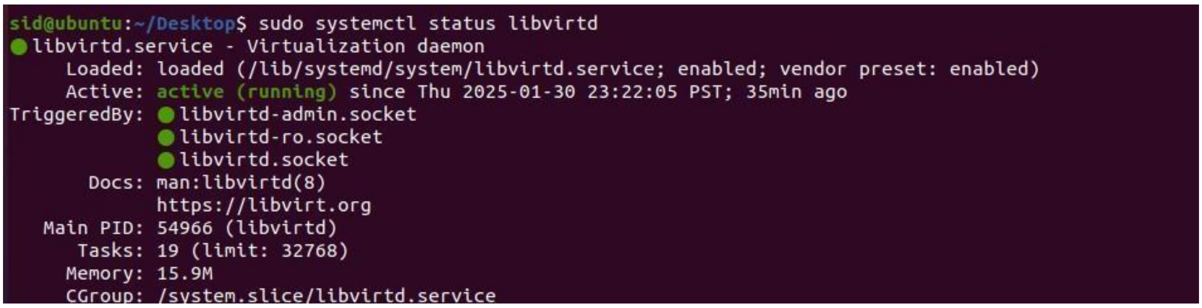
```
sid@ubuntu:~/Desktop$ sudo adduser 'sid' kvm
Adding user `sid' to group `kvm' ...
Adding user sid to group kvm
Done.
```

Step 16: Verify the installation. Confirm the KVM installation with virsh command . The command will list all active and inactive virtual machines on the system.



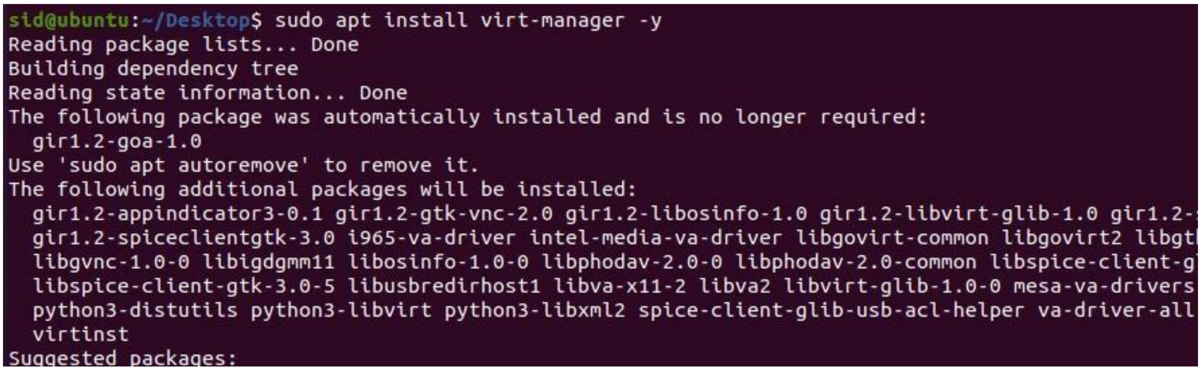
```
sid@ubuntu:~/Desktop$ sudo virsh list --all
 Id   Name   State
--------------------
```

Step 17: Use systemctl command to check the status of libvirtd, the daemon that provides the backend services for libvirt virtualization management system. If the virtualization is not active then activate it with following command
“sudo systemctl enable - - now libvirtd”

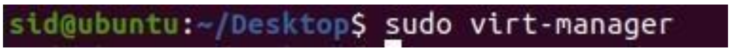


```
sid@ubuntu:~/Desktop$ sudo systemctl status libvirtd
● libvirtd.service - Virtualization daemon
     Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2025-01-30 23:22:05 PST; 35min ago
TriggeredBy: ● libvirtd-admin.socket
             ● libvirtd-ro.socket
             ● libvirtd.socket
       Docs: man:libvirtd(8)
             https://libvirt.org
   Main PID: 54966 (libvirtd)
      Tasks: 19 (limit: 32768)
     Memory: 15.9M
     CGroup: /system.slice/libvirtd.service
```

Step 18: Create a virtual machine on Ubuntu. For creating a virtual machine install virt-manager, it is a tool for creating and managing virtual machines.

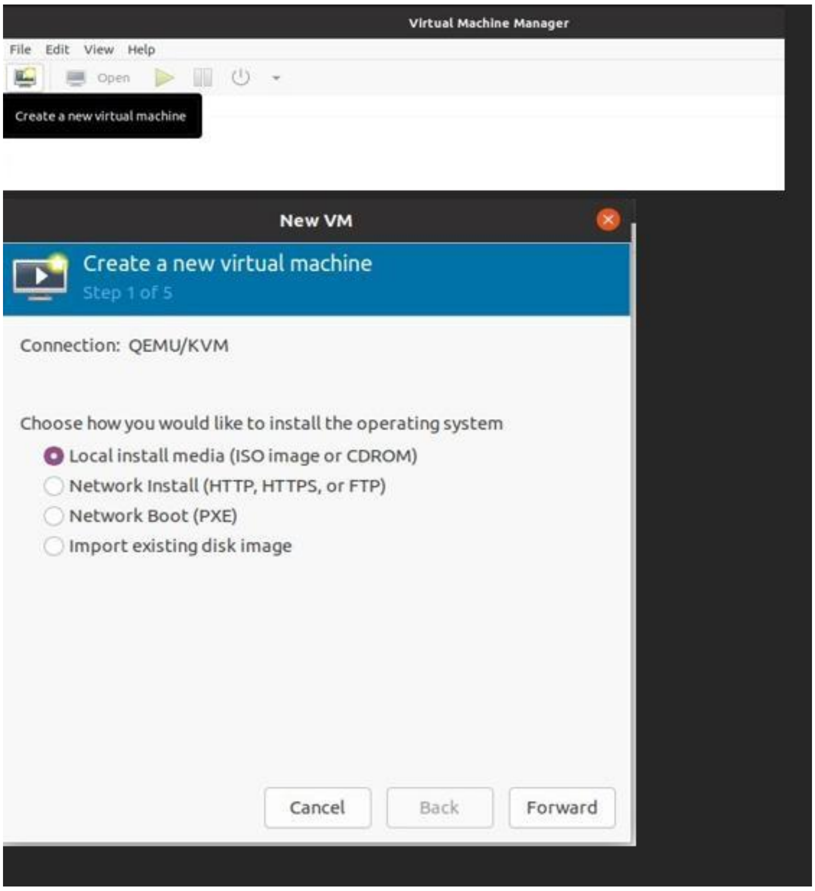


```
sid@ubuntu:~/Desktop$ sudo apt install virt-manager -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  gir1.2-goa-1.0
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  gir1.2-appindicator3-0.1 gir1.2-gtk-vnc-2.0 gir1.2-libosinfo-1.0 gir1.2-libvirt-glib-1.0 gir1.2-
  gir1.2-spiceclientgtk-3.0 i965-va-driver intel-media-va-driver libgovirt-common libgovirt2 libgt
  libgvnc-1.0-0 libigdgmm11 libosinfo-1.0-0 libphodav-2.0-0 libphodav-2.0-common libspice-client-g
  libspice-client-gtk-3.0-5 libusbredirhost1 libva-x11-2 libva2 libvirt-glib-1.0-0 mesa-va-drivers
  python3-distutils python3-libvirt python3-libxml2 spice-client-glib-usb-acl-helper va-driver-all
  virtinst
Suggested packages:
```



```
sid@ubuntu:~/Desktop$ sudo virt-manager
```

Step 19: Copy the Ubuntu ISO file to the desktop of the virtual machine and click on the Following option.



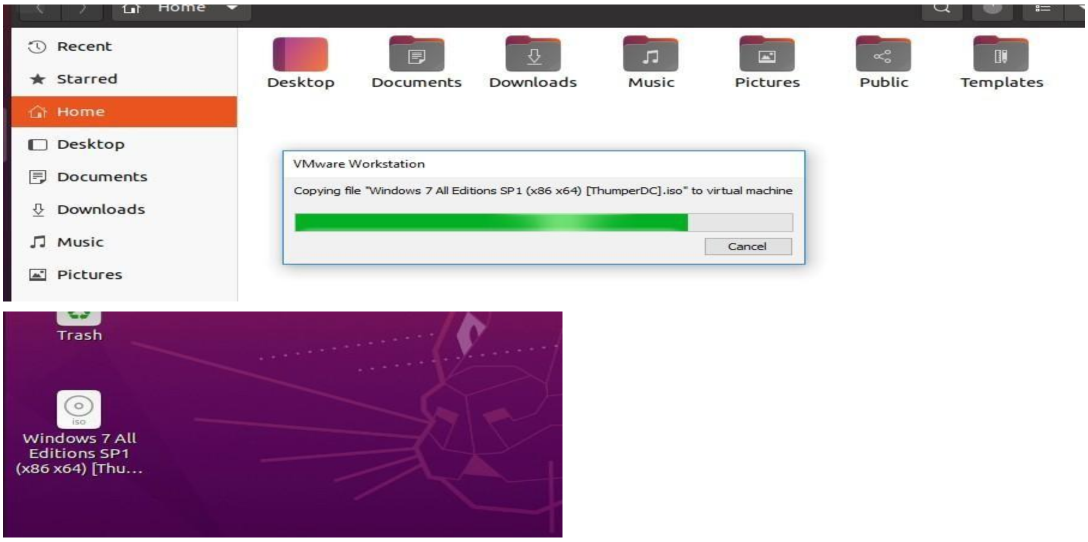
Create a new virtual machine
Step 1 of 5

Connection: QEMU/KVM

Choose how you would like to install the operating system

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

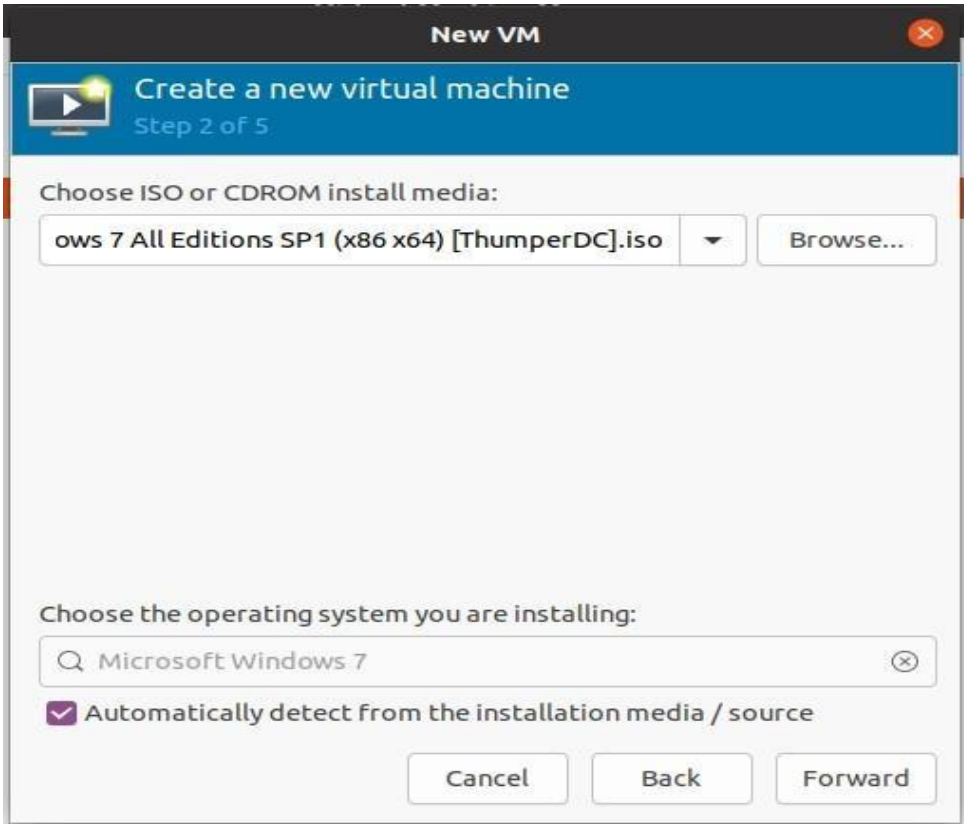
Step 20: Here the file is being copied.



VMware Workstation

Copying file "Windows 7 All Editions SP1 (x86 x64) [ThumperDC].iso" to virtual machine

Step 21: Browse the ISO file and tick the checkbox as follows. Click on forward.



Choose ISO or CDROM install media:

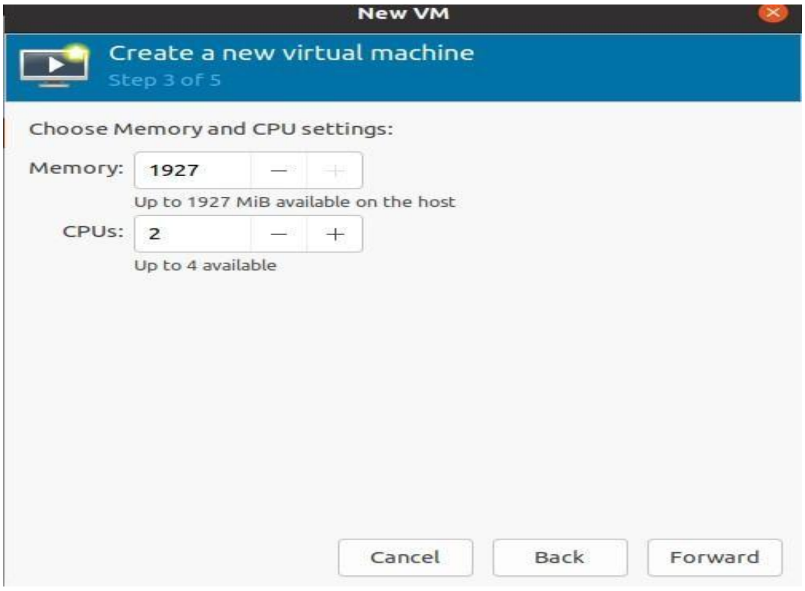
ows 7 All Editions SP1 (x86 x64) [ThumperDC].iso

Choose the operating system you are installing:

Microsoft Windows 7

☑ Automatically detect from the installation media / source

Step 22: Select the cpu and memory as required. Click on forward.



Choose Memory and CPU settings:

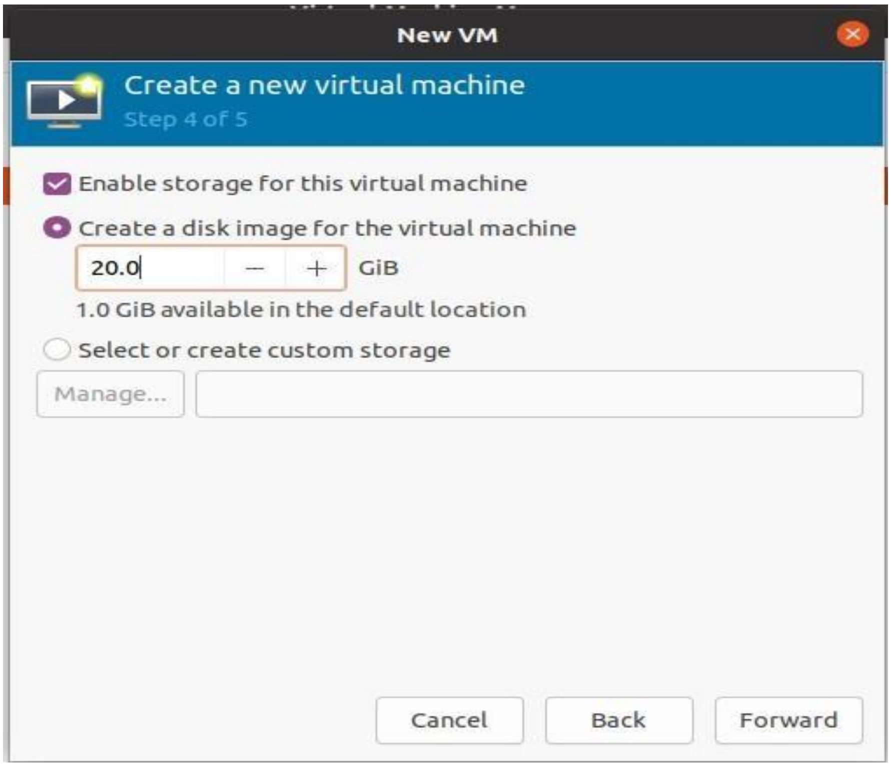
Memory: 1927

Up to 1927 MiB available on the host

CPUs: 2

Up to 4 available

Step 23: Give the disk size as required in that field. Click on forward.



Create a new virtual machine

Step 4 of 5

☑ Enable storage for this virtual machine

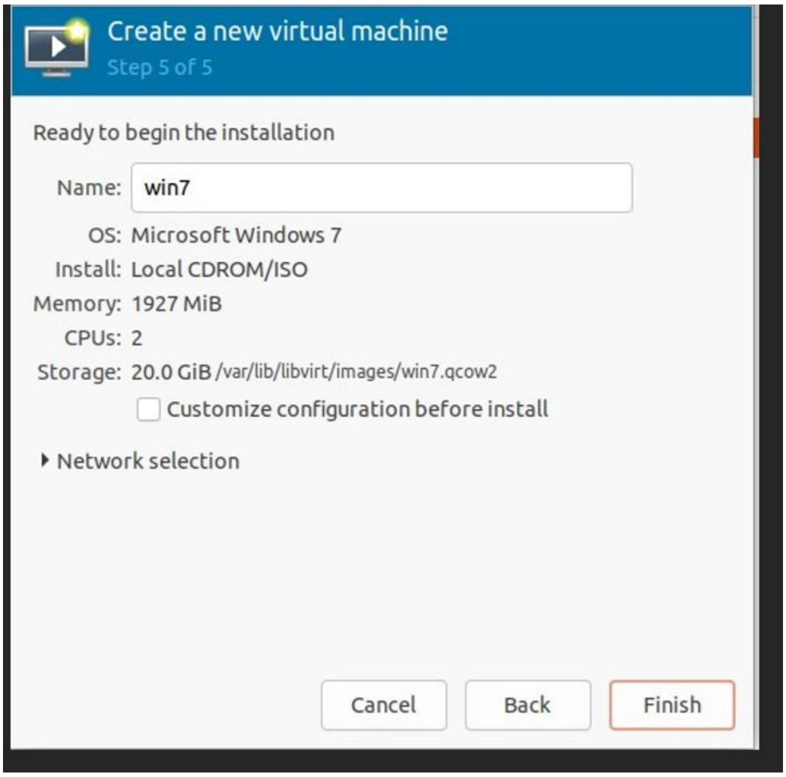
● Create a disk image for the virtual machine

20.0 GiB

1.0 GiB available in the default location

○ Select or create custom storage

Step 24: The following window will appear which is the final stage of the installation



Create a new virtual machine

Step 5 of 5

Ready to begin the installation

Name: win7

OS: Microsoft Windows 7

Install: Local CDROM/ISO

Memory: 1927 MiB

CPUs: 2

Storage: 20.0 GiB /var/lib/libvirt/images/win7.qcow2

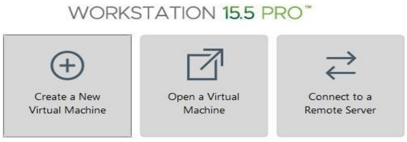
□ Customize configuration before install

Date:-

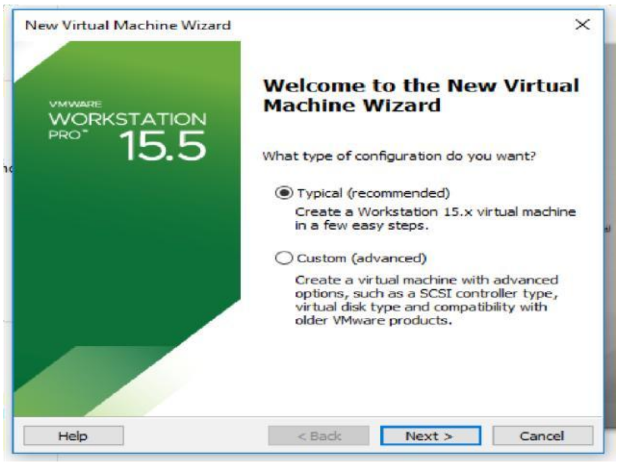
Practical 7

Aim : Study and implementation of Infrastructure as a Service.

Step 1: Open VMware to create a new virtual machine and select the option “Create a New Virtual Machine”.



Step 2: Select the “Typical(recommended)” option and click on Next.



New Virtual Machine Wizard

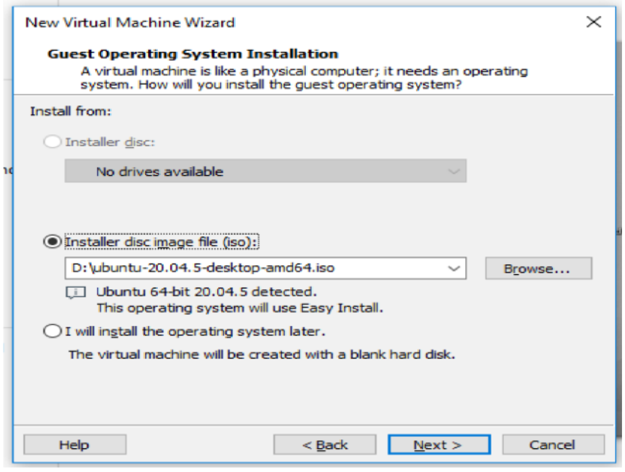
Welcome to the New Virtual Machine Wizard

What type of configuration do you want?

● Typical (recommended)
Create a Workstation 15.x virtual machine in a few easy steps.

○ Custom (advanced)
Create a virtual machine with advanced options, such as a SCSI controller type, virtual disk type and compatibility with older VMware products.

Step 3: The following window will appear, click on Next and proceed.



New Virtual Machine Wizard

Guest Operating System Installation
A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?

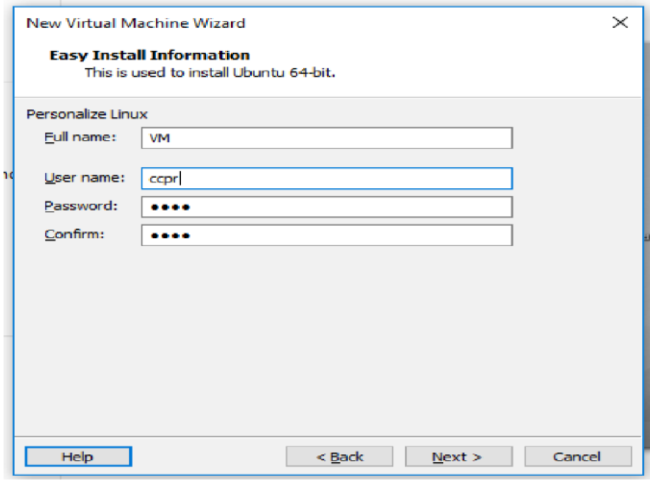
Install from:

○ Installer disc:
No drives available

● Installer disc image file (iso):
D:\ubuntu-20.04.5-desktop-amd64.iso
Ubuntu 64-bit 20.04.5 detected.
This operating system will use Easy Install.

○ I will install the operating system later.
The virtual machine will be created with a blank hard disk.

Step 4: Fill the following fields such as the Full name,Username and Password . Click on Next. (Password-ccpr)


New Virtual Machine Wizard

Easy Install Information
This is used to install Ubuntu 64-bit.

Personalize Linux

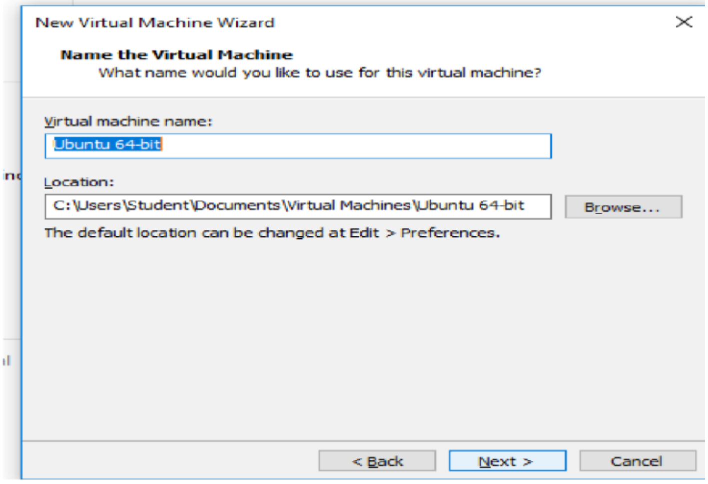
Full name: VM

User name: ccpr

Password: ●●●●

Confirm: ●●●●

Step 5: The following fields show the Name, Location where the Virtual Machine is stored.


New Virtual Machine Wizard

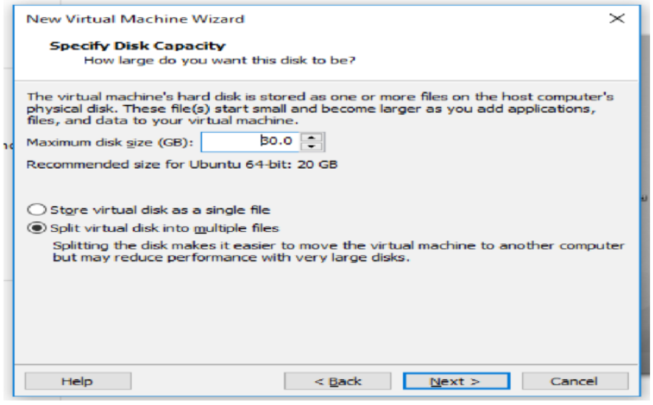
Name the Virtual Machine
What name would you like to use for this virtual machine?

Virtual machine name: Ubuntu 64-bit

Location: C:\Users\Student\Documents\Virtual Machines\Ubuntu 64-bit

The default location can be changed at Edit > Preferences.

Step 6: Keep the maximum disk size as 30.0 GB and select the "Split virtual disk into multiple files" option. Click on Next.


New Virtual Machine Wizard

Specify Disk Capacity
How large do you want this disk to be?

The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.

Maximum disk size (GB): 30.0

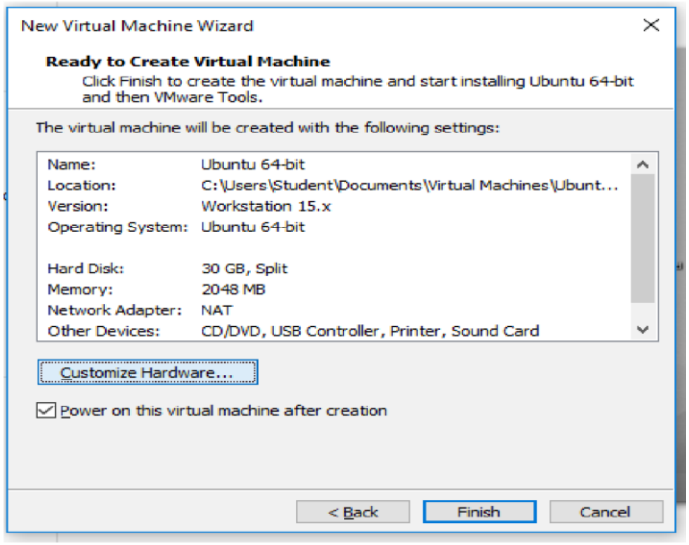
Recommended size for Ubuntu 64-bit: 20 GB

○ Store virtual disk as a single file

● Split virtual disk into multiple files

Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

Step 7: Click on the option of “Customize Hardware”.



New Virtual Machine Wizard

Ready to Create Virtual Machine

Click Finish to create the virtual machine and start installing Ubuntu 64-bit and then VMware Tools.

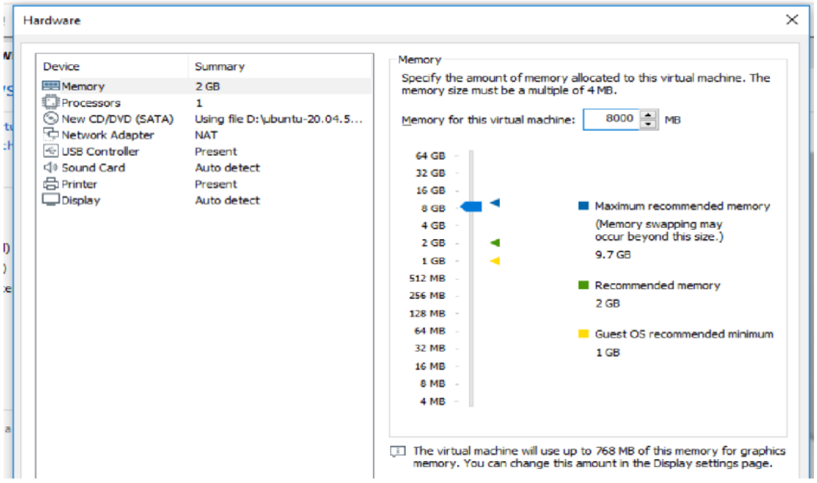
The virtual machine will be created with the following settings:

Name:	Ubuntu 64-bit
Location:	C:\Users\Student\Documents\Virtual Machines\Ubunt...
Version:	Workstation 15.x
Operating System:	Ubuntu 64-bit
Hard Disk:	30 GB, Split
Memory:	2048 MB
Network Adapter:	NAT
Other Devices:	CD/DVD, USB Controller, Printer, Sound Card

Customize Hardware...

☑ Power on this virtual machine after creation

Step 8: Click on Memory and set the memory as 8000 MB and then click on processors and keep the value of Number of processors and Number of core processors as 2. And make sure that the below two checkboxes are ticked . Click on close and then on finish.



Hardware

Device	Summary
Memory	2 GB
Processors	1
New CD/DVD (SATA)	Using file D:\ubuntu-20.04.5...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

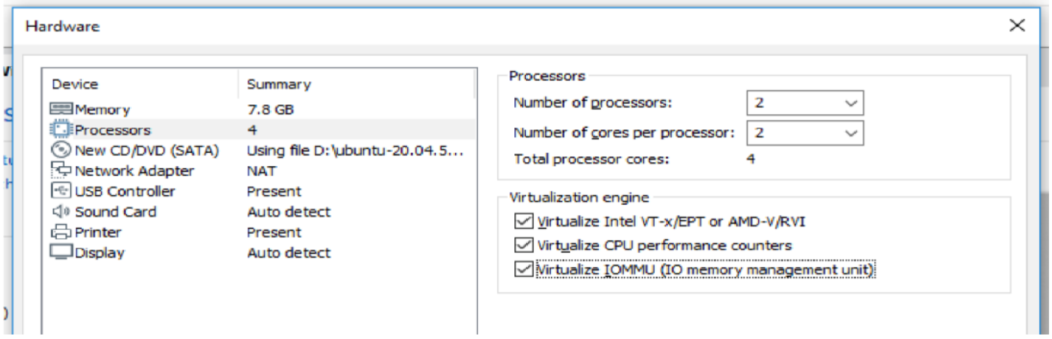
Memory for this virtual machine: 8000 MB

■ Maximum recommended memory (Memory swapping may occur beyond this size.) 9.7 GB

■ Recommended memory 2 GB

■ Guest OS recommended minimum 1 GB

The virtual machine will use up to 768 MB of this memory for graphics memory. You can change this amount in the Display settings page.



Hardware

Device	Summary
Memory	7.8 GB
Processors	4
New CD/DVD (SATA)	Using file D:\ubuntu-20.04.5...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Processors

Number of processors: 2

Number of cores per processor: 2

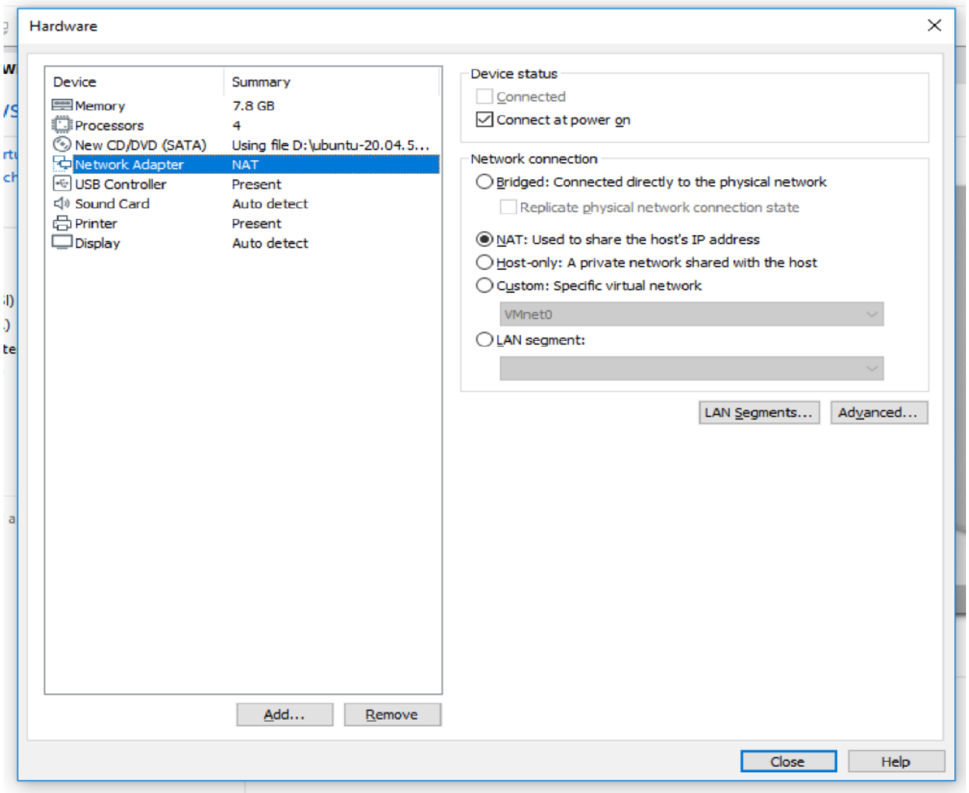
Total processor cores: 4

Virtualization engine

☑ Virtualize Intel VT-x/EPT or AMD-V/RVI

☑ Virtualize CPU performance counters

☑ Virtualize IOMMU (IO memory management unit)



Hardware

Device	Summary
Memory	7.8 GB
Processors	4
New CD/DVD (SATA)	Using file D:\ubuntu-20.04.5...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Device status

□ Connected

☑ Connect at power on

Network connection

○ Bridged: Connected directly to the physical network

□ Replicate physical network connection state

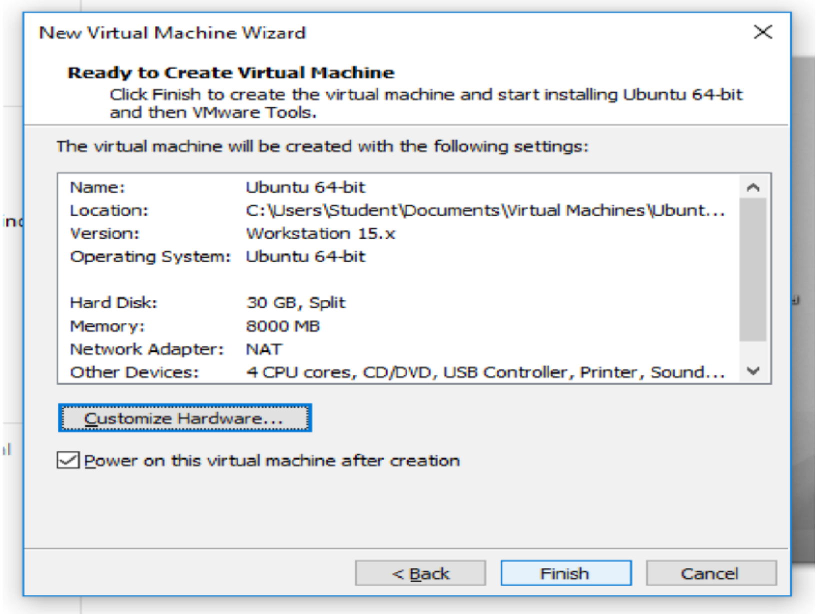
● NAT: Used to share the host's IP address

○ Host-only: A private network shared with the host

○ Custom: Specific virtual network

VMnet0

○ LAN segment:



New Virtual Machine Wizard

Ready to Create Virtual Machine

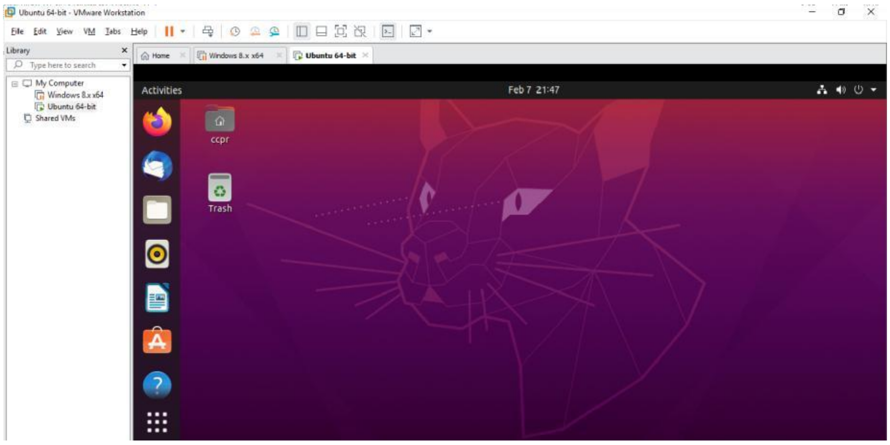
Click Finish to create the virtual machine and start installing Ubuntu 64-bit and then VMware Tools.

The virtual machine will be created with the following settings:

Name:	Ubuntu 64-bit
Location:	C:\Users\Student\Documents\Virtual Machines\Ubunt...
Version:	Workstation 15.x
Operating System:	Ubuntu 64-bit
Hard Disk:	30 GB, Split
Memory:	8000 MB
Network Adapter:	NAT
Other Devices:	4 CPU cores, CD/DVD, USB Controller, Printer, Sound...

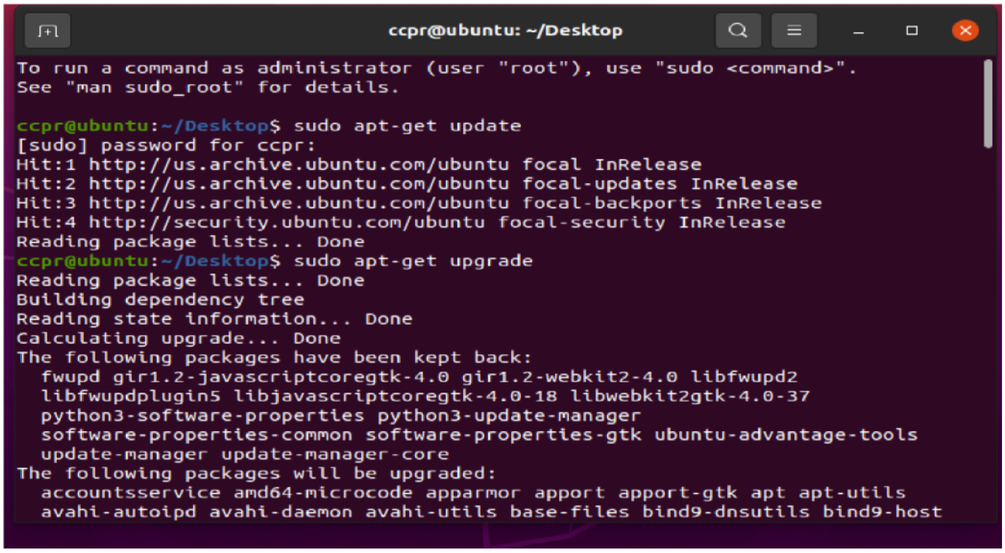
☑ Power on this virtual machine after creation

Step 9: Now the following window will appear then open command prompt.

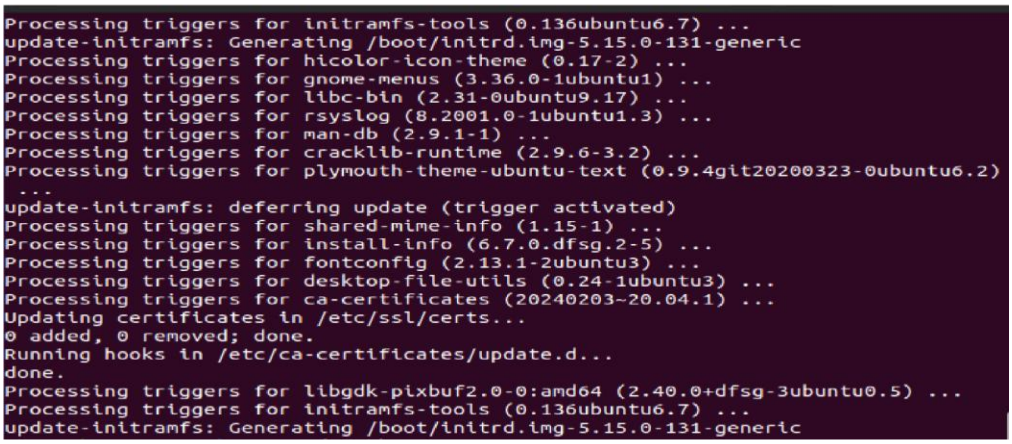


Step 10: Update Ubuntu by executing following commands

- Sudo apt update
- Sudo apt upgrade

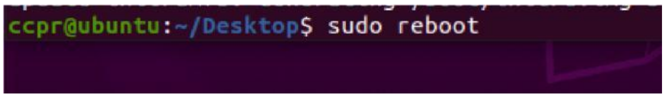


```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ccpr@ubuntu:~/Desktop$ sudo apt-get update
[sudo] password for ccpr:
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
ccpr@ubuntu:~/Desktop$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  fwupd gir1.2-javascriptcoregtk-4.0 gir1.2-webkit2-4.0 libfwupd2
  libfwupdplugin5 libjavascriptcoregtk-4.0-18 libwebkit2gtk-4.0-37
  python3-software-properties python3-update-manager
  software-properties-common software-properties-gtk ubuntu-advantage-tools
  update-manager update-manager-core
The following packages will be upgraded:
  accountsservice amd64-microcode apparmor apport apport-gtk apt apt-utils
  avahi-autoipd avahi-daemon avahi-utils base-files bind9-dnsutils bind9-host
```



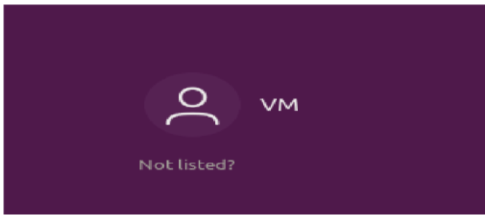
```
Processing triggers for initramfs-tools (0.136ubuntu6.7) ...
update-initramfs: Generating /boot/initrd.img-5.15.0-131-generic
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.17) ...
Processing triggers for rsyslog (8.2001.0-1ubuntu1.3) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for cracklib-runtime (2.9.6-3.2) ...
Processing triggers for plymouth-theme-ubuntu-text (0.9.4git20200323-0ubuntu6.2)
 ...
update-initramfs: deferring update (trigger activated)
Processing triggers for shared-mime-info (1.15-1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
Processing triggers for fontconfig (2.13.1-2ubuntu3) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
Processing triggers for ca-certificates (20240203~20.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Processing triggers for libgdk-pixbuf2.0-0:amd64 (2.40.0+dfsg-3ubuntu0.5) ...
Processing triggers for initramfs-tools (0.136ubuntu6.7) ...
update-initramfs: Generating /boot/initrd.img-5.15.0-131-generic
```

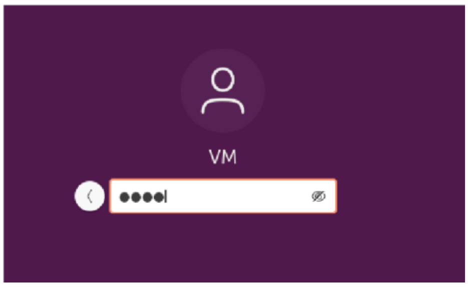
Step 11: Now reboot the system by the ‘sudo reboot’ or ‘init 6’ command.



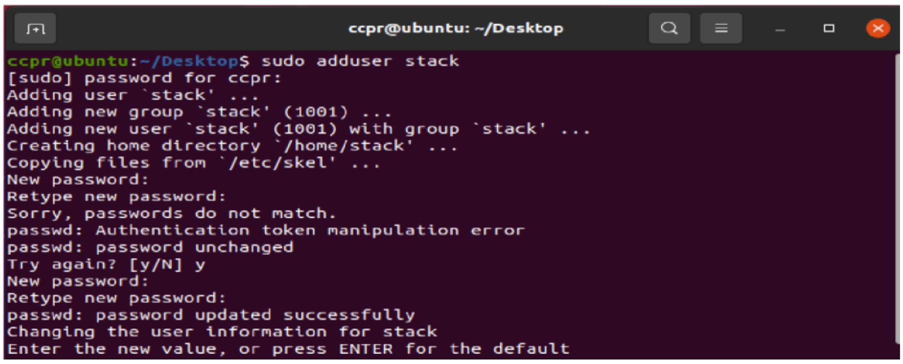
```
ccpr@ubuntu:~/Desktop$ sudo reboot
```

Step 12: Now select your created machine and enter the respective password.



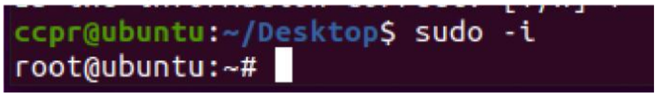


Step 13: Create new user called stack



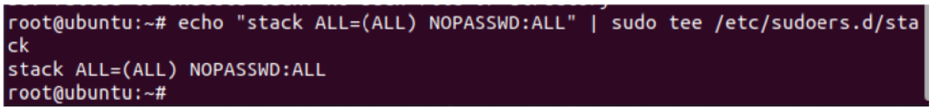
```
ccpr@ubuntu:~/Desktop$ sudo adduser stack
[sudo] password for ccpr:
Adding user `stack' ...
Adding new group `stack' (1001) ...
Adding new user `stack' (1001) with group `stack' ...
Creating home directory `/home/stack' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for stack
Enter the new value, or press ENTER for the default
```

Step 14: Log in as root.



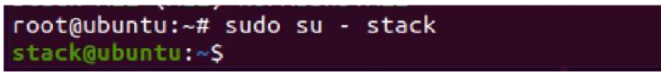
```
ccpr@ubuntu:~/Desktop$ sudo -i
root@ubuntu:~#
```

Step 15: Assign the sudo privileges to stack user as follows (#echo “stack ALL=(ALL) NOPASSWD:ALL” | sudo tee /etc/sudoers.d/stack)



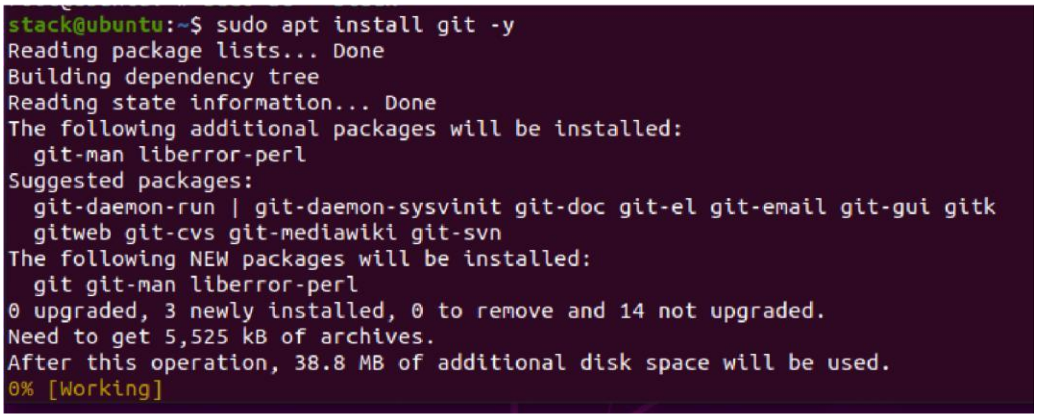
```
root@ubuntu:~# echo "stack ALL=(ALL) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/sta
ck
stack ALL=(ALL) NOPASSWD:ALL
root@ubuntu:~#
```

Step 16: Switch to the stack user, by following command “sudo su - stack”.



```
root@ubuntu:~# sudo su - stack
stack@ubuntu:~$
```

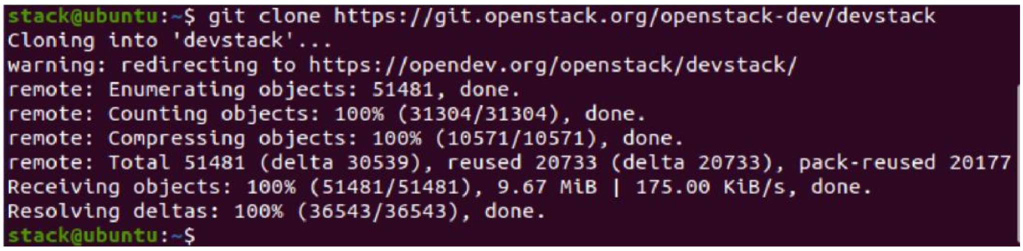
Step 17: Install the Git using the command “sudo apt install git -y”.



```
stack@ubuntu:~$ sudo apt install git -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 14 not upgraded.
Need to get 5,525 kB of archives.
After this operation, 38.8 MB of additional disk space will be used.
0% [Working]
```

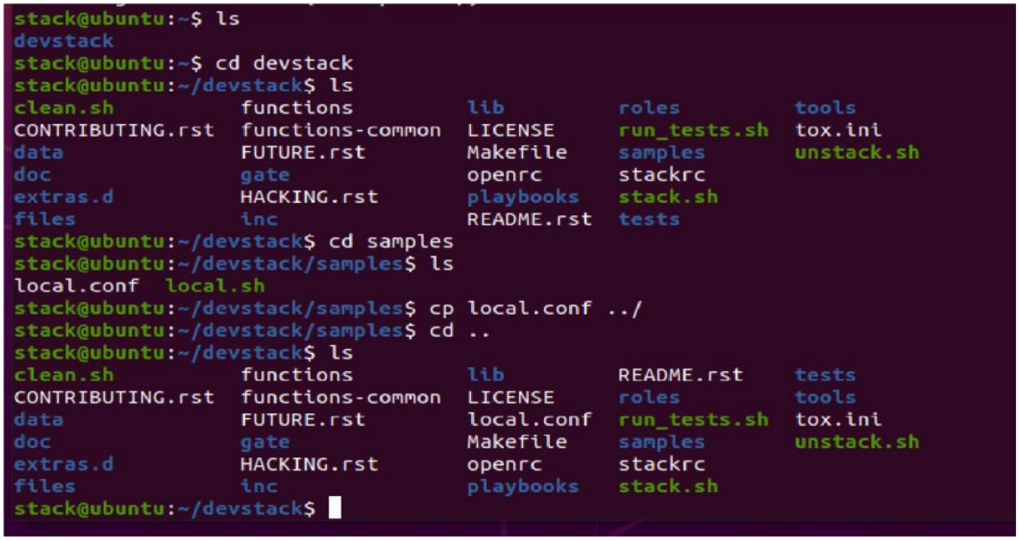
Step 18: Download OpenStack

Once Git is installed, download the DevStack from github by following command “git clone https://git.openstack.org/openstack-dev/devstack”.



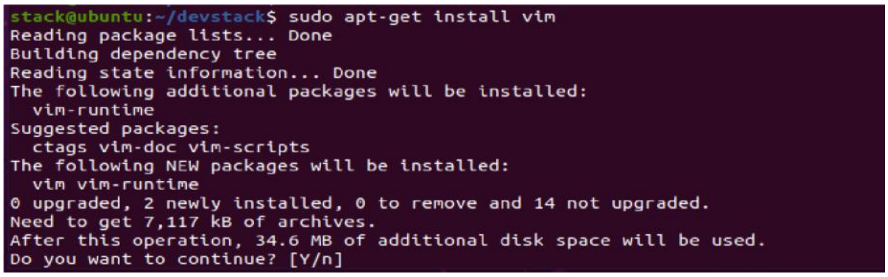
```
stack@ubuntu:~$ git clone https://git.openstack.org/openstack-dev/devstack
Cloning into 'devstack'...
warning: redirecting to https://opendev.org/openstack/devstack/
remote: Enumerating objects: 51481, done.
remote: Counting objects: 100% (31304/31304), done.
remote: Compressing objects: 100% (10571/10571), done.
remote: Total 51481 (delta 30539), reused 20733 (delta 20733), pack-reused 20177
Receiving objects: 100% (51481/51481), 9.67 MiB | 175.00 KiB/s, done.
Resolving deltas: 100% (36543/36543), done.
stack@ubuntu:~$
```

Step 19: Go to DevStack directory and look for local.conf file.

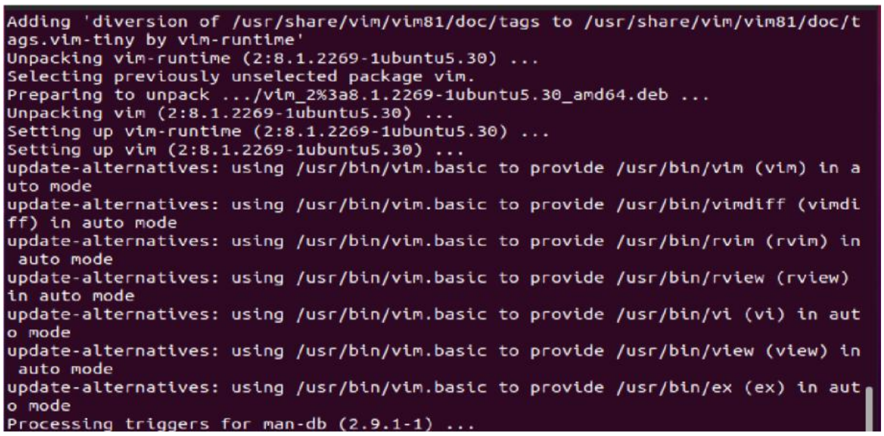


```
stack@ubuntu:~$ ls
devstack
stack@ubuntu:~$ cd devstack
stack@ubuntu:~/devstack$ ls
clean.sh          functions         lib         roles         tools
CONTRIBUTING.rst  functions-common  LICENSE     run_tests.sh  tox.ini
data              FUTURE.rst        Makefile    samples       unstack.sh
doc               gate              openrc      stackrc
extras.d          HACKING.rst       playbooks   stack.sh
files             inc               README.rst  tests
stack@ubuntu:~/devstack$ cd samples
stack@ubuntu:~/devstack/samples$ ls
local.conf  local.sh
stack@ubuntu:~/devstack/samples$ cp local.conf ../
stack@ubuntu:~/devstack/samples$ cd ..
stack@ubuntu:~/devstack$ ls
clean.sh          functions         lib         README.rst    tests
CONTRIBUTING.rst  functions-common  LICENSE     roles         tools
data              FUTURE.rst        local.conf  run_tests.sh  tox.ini
doc               gate              Makefile    samples       unstack.sh
extras.d          HACKING.rst       openrc      stackrc
files             inc               playbooks   stack.sh
stack@ubuntu:~/devstack$ █
```

Step 20: Install vim using the command “sudo apt-get install vim”.

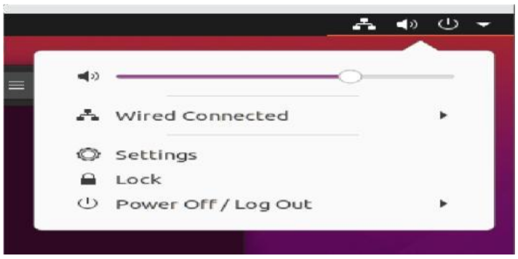


```
stack@ubuntu:~/devstack$ sudo apt-get install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  vim-runtime
Suggested packages:
  ctags vim-doc vim-scripts
The following NEW packages will be installed:
  vim vim-runtime
0 upgraded, 2 newly installed, 0 to remove and 14 not upgraded.
Need to get 7,117 kB of archives.
After this operation, 34.6 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

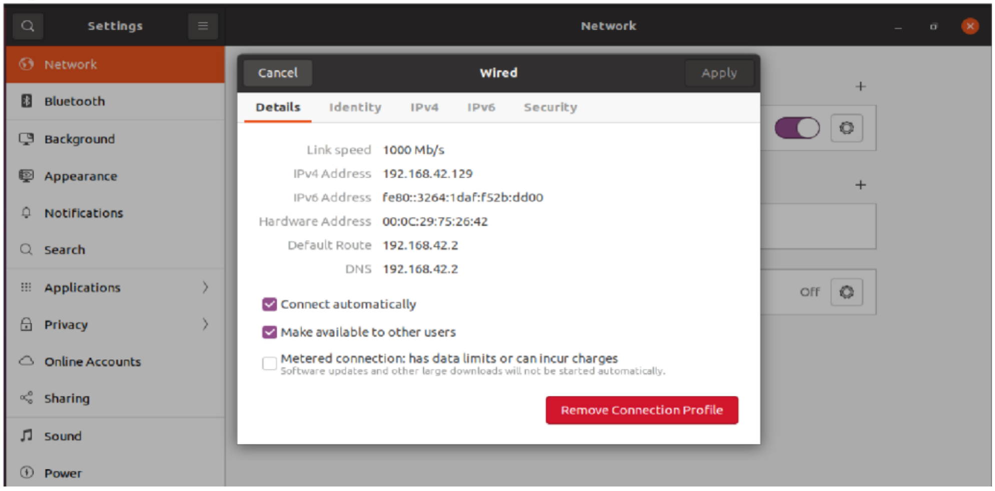


```
Adding 'diversion of /usr/share/vim/vim81/doc/tags to /usr/share/vim/vim81/doc/t
ags.vim-tiny by vim-runtime'
Unpacking vim-runtime (2:8.1.2269-1ubuntu5.30) ...
Selecting previously unselected package vim.
Preparing to unpack .../vim_2%3a8.1.2269-1ubuntu5.30_amd64.deb ...
Unpacking vim (2:8.1.2269-1ubuntu5.30) ...
Setting up vim-runtime (2:8.1.2269-1ubuntu5.30) ...
Setting up vim (2:8.1.2269-1ubuntu5.30) ...
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/vim (vim) in a
uto mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/vimdiff (vimdi
ff) in auto mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/rvim (rvim) in
 auto mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/rview (rview)
in auto mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/vi (vi) in aut
o mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/view (view) in
 auto mode
update-alternatives: using /usr/bin/vim.basic to provide /usr/bin/ex (ex) in aut
o mode
Processing triggers for man-db (2.9.1-1) ...
```

Step 21: For fetching IP address go to ‘Wired Connected’ option.



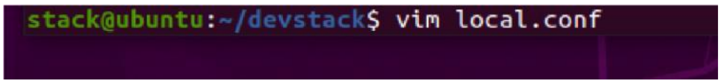
Step 22: Now note down the IPv4 Address. Here “192.168.42.129”



Link speed 1000 Mb/s
IPv4 Address 192.168.42.129
IPv6 Address fe80::3264:1daf:f52b:dd00
Hardware Address 00:0C:29:75:26:42
Default Route 192.168.42.2
DNS 192.168.42.2

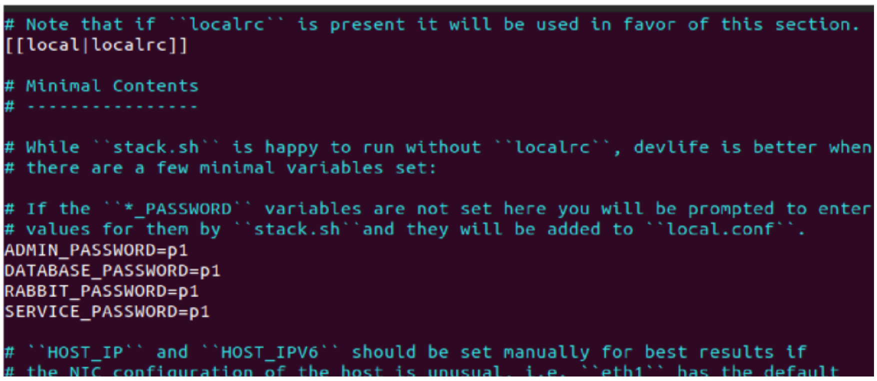
☑ Connect automatically
☑ Make available to other users
□ Metered connection: has data limits or can incur charges
Software updates and other large downloads will not be started automatically.

Step 23: Now open the local.conf file using command “vim local.conf”.



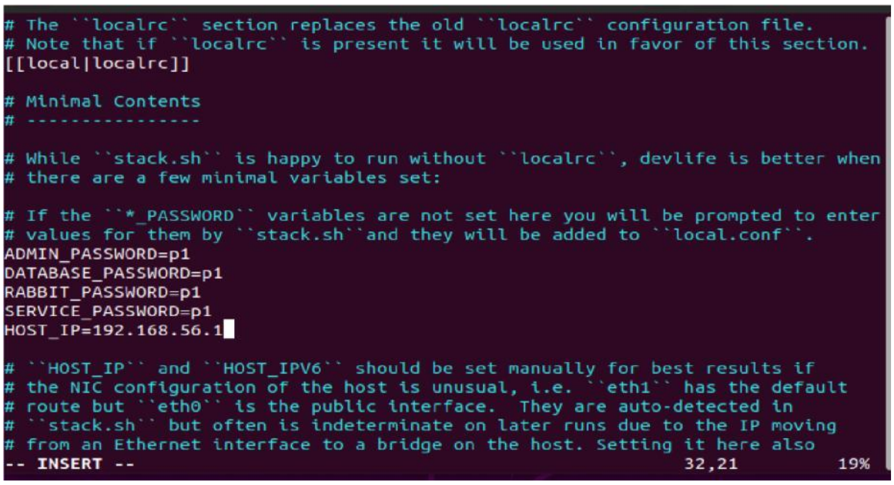
```
stack@ubuntu:~/devstack$ vim local.conf
```

Step 24: Once the file is opened go to insert mode and then after the device_password field write the system IP address that we copied from settings above “HOST_IP=192.168.42.129”



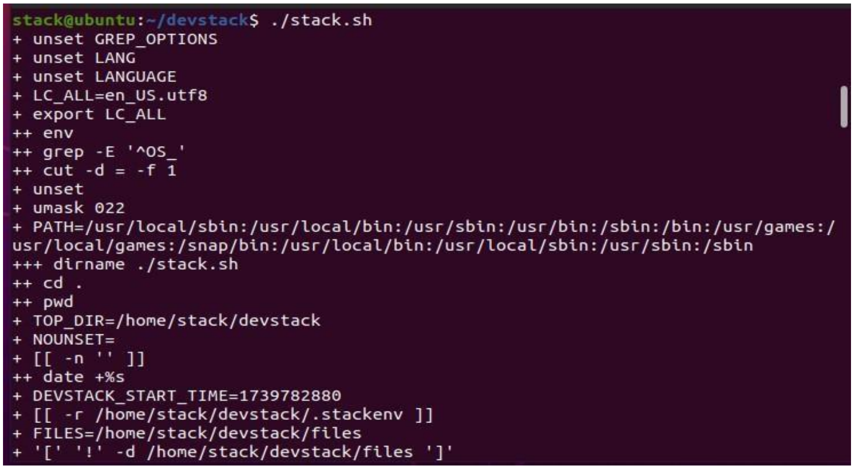
```
# Note that if ``localrc`` is present it will be used in favor of this section.
[[local|localrc]]

# Minimal Contents
# ----------------

# While ``stack.sh`` is happy to run without ``localrc``, devlife is better when
# there are a few minimal variables set:

# If the ``*_PASSWORD`` variables are not set here you will be prompted to enter
# values for them by ``stack.sh``and they will be added to ``local.conf``.
ADMIN_PASSWORD=p1
DATABASE_PASSWORD=p1
RABBIT_PASSWORD=p1
SERVICE_PASSWORD=p1

# ``HOST_IP`` and ``HOST_IPV6`` should be set manually for best results if
# the NIC configuration of the host is unusual, i.e. ``eth1`` has the default
```

HOSTIP=192.168.42.129



```
# The ``localrc`` section replaces the old ``localrc`` configuration file.
# Note that if ``localrc`` is present it will be used in favor of this section.
[[local|localrc]]

# Minimal Contents
# ----------------

# While ``stack.sh`` is happy to run without ``localrc``, devlife is better when
# there are a few minimal variables set:

# If the ``*_PASSWORD`` variables are not set here you will be prompted to enter
# values for them by ``stack.sh``and they will be added to ``local.conf``.
ADMIN_PASSWORD=p1
DATABASE_PASSWORD=p1
RABBIT_PASSWORD=p1
SERVICE_PASSWORD=p1
HOST_IP=192.168.56.1

# ``HOST_IP`` and ``HOST_IPV6`` should be set manually for best results if
# the NIC configuration of the host is unusual, i.e. ``eth1`` has the default
# route but ``eth0`` is the public interface.  They are auto-detected in
# ``stack.sh`` but often is indeterminate on later runs due to the IP moving
# from an Ethernet interface to a bridge on the host. Setting it here also
-- INSERT --                                                  32,21         19%
```

Step 26: To install and run the OpenStack, execute the following commands “./stack.sh”



```
stack@ubuntu:~/devstack$ ./stack.sh
+ unset GREP_OPTIONS
+ unset LANG
+ unset LANGUAGE
+ LC_ALL=en_US.utf8
+ export LC_ALL
++ env
++ grep -E '^OS_'
++ cut -d = -f 1
+ unset
+ umask 022
+ PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/
usr/local/games:/snap/bin:/usr/local/bin:/usr/local/sbin:/usr/sbin:/sbin
+++ dirname ./stack.sh
++ cd .
++ pwd
+ TOP_DIR=/home/stack/devstack
+ NOUNSET=
+ [[ -n '' ]]
++ date +%s
+ DEVSTACK_START_TIME=1739782880
+ [[ -r /home/stack/devstack/.stackenv ]]
+ FILES=/home/stack/devstack/files
+ '[' '!' -d /home/stack/devstack/files ']'
```