CLOUD INFRASTRUCTURE AND SERVICES

PRACTICAL FILE

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

(Information Technology)

Submitted By:

Harkirat Kaur

Jasleen kaur

Shivay Bhandari

Raghav Malhotra

Submitted to:

Dr. Kamaljit kaur

Department of Information Technology,

Guru Nanak Dev Engineering College

Ludhiana-141106

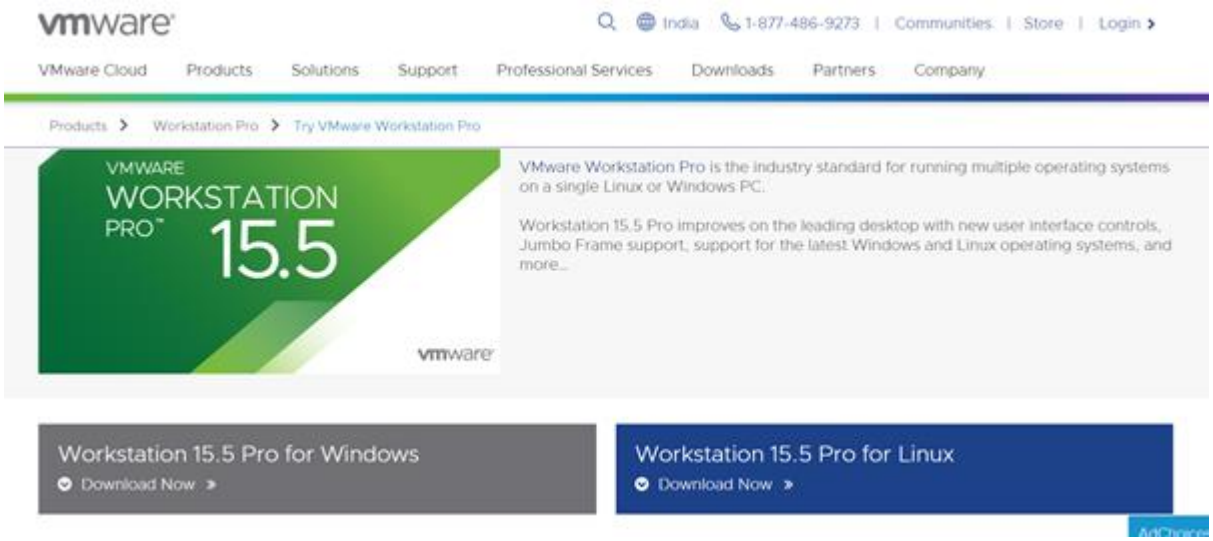
INDEX

Experiment Number	Topic	Remarks
1	**Install VirtualBox/VMware with different flavours of Linux or windows OS on top of Linux/Windows**	
2	**Introduction to openstack and its components.**	
3	**Installation of Open-Stack using Micro-Stack**	
4	**Creating and launching basic virtual machine.**	
5	**Creating and Managing Images and Templates**	
6	**Creating and Managing networks .**	
7	**Creating and Managing Users**	
8	**Managing security groups and policies**	

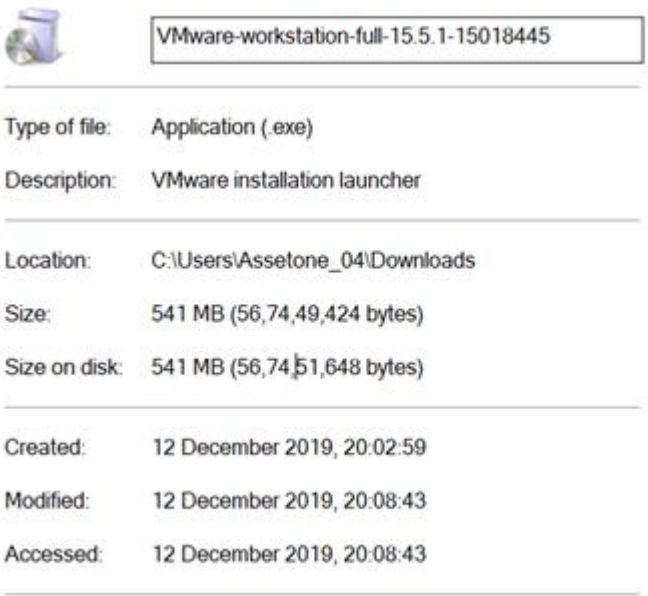
Practical 1:

Install VirtualBox/VMware with different flavours of Linux or windows OS on top of Linux/Windows

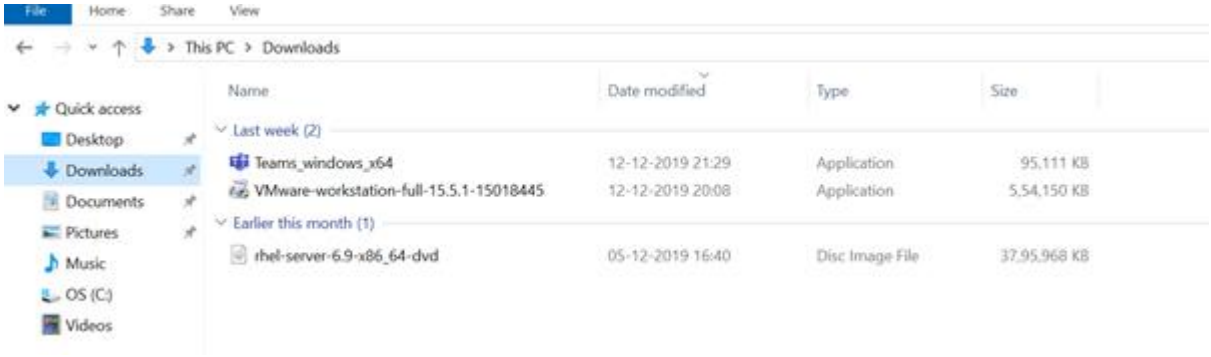
1. Installing VMware Workstation from given below link. There are two options for downloading one is Windows and other for Linux. My Base Operating System is Windows8, So I choose for VMware for Windows. If Your Base OS is Linux go and choose VMware for Linux Link. **https://www.vmware.com/in/products/workstation-pro/workstation-pro-evaluation.html**



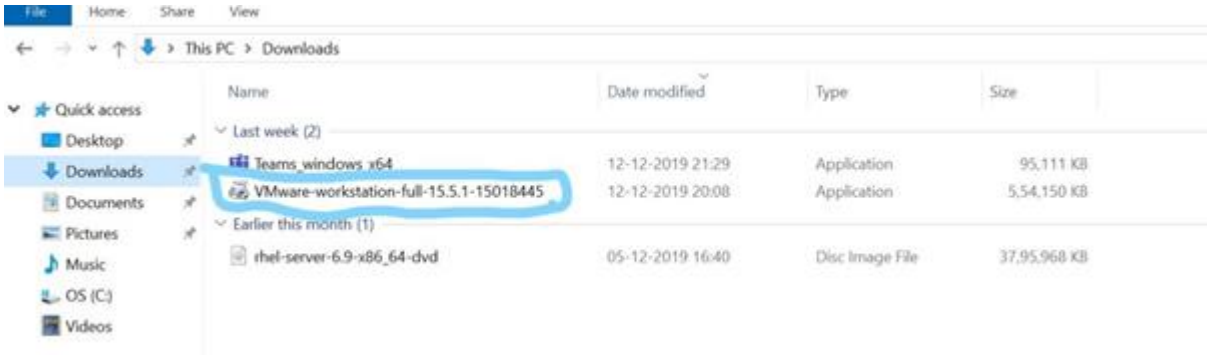
2.Check your VMware Properties.



3. Go to Download Folder.



4. Click the VMware downloaded File and Install it.

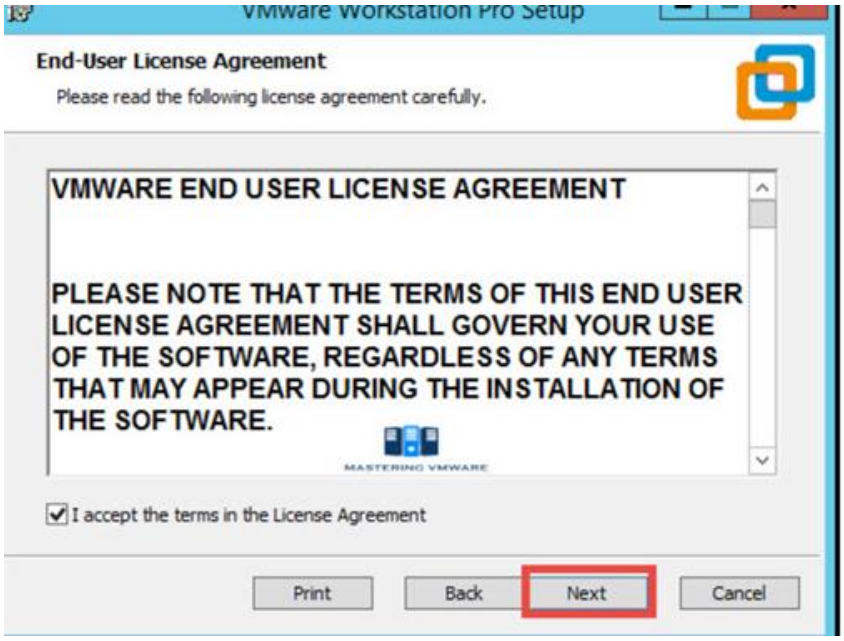


5.Click on VMware Software and click and choose "**Pin to Taskbar".**

6.Click on VMware Software and Click on Next to the Installation wizard.

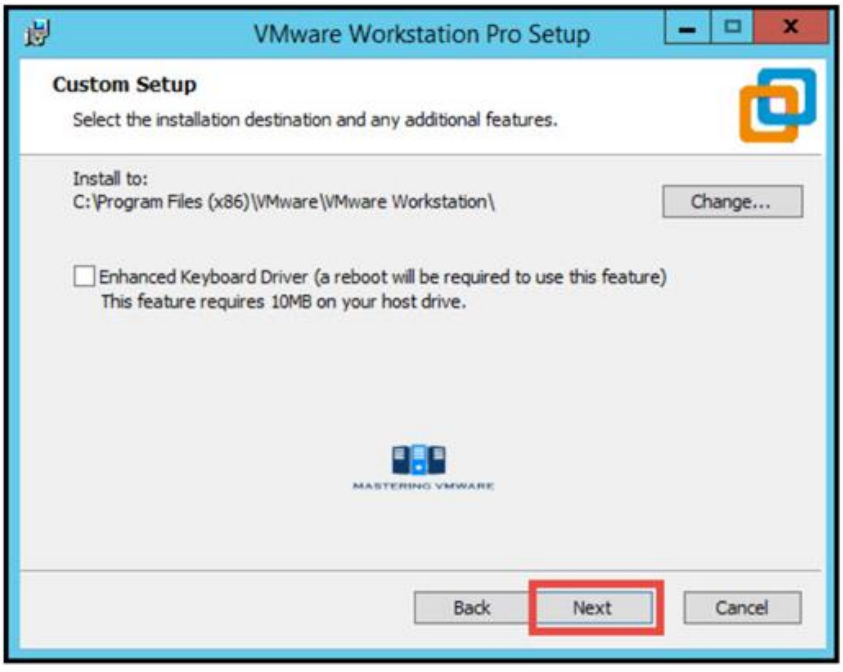
7. Read and Accept the VMware End User license agreement.

Click Next to Continue.

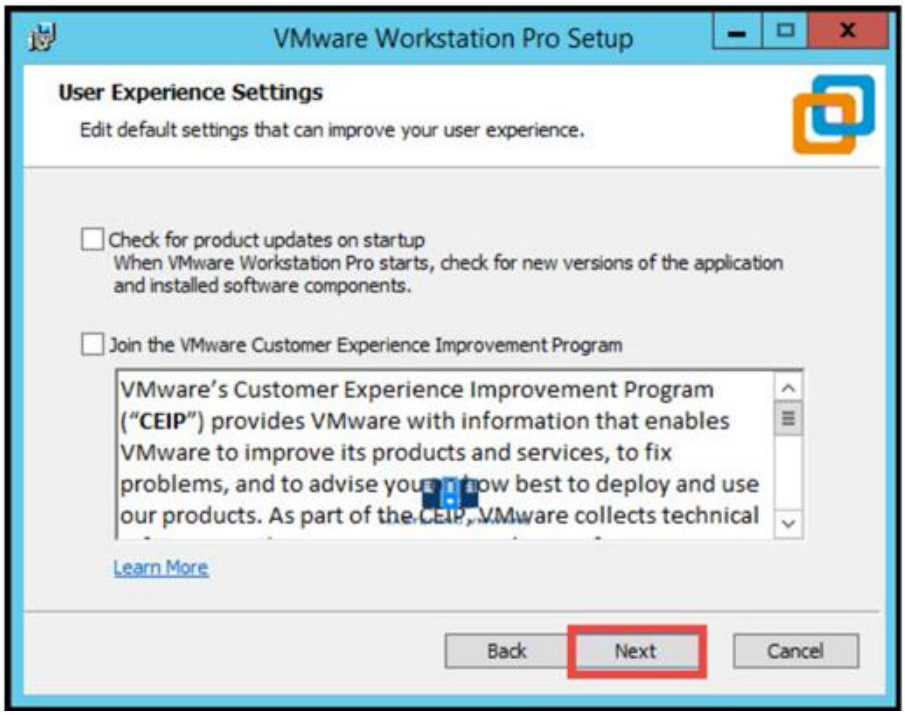


8. Specify the Installation directory. You can also enable Enhance keyboard driver here.

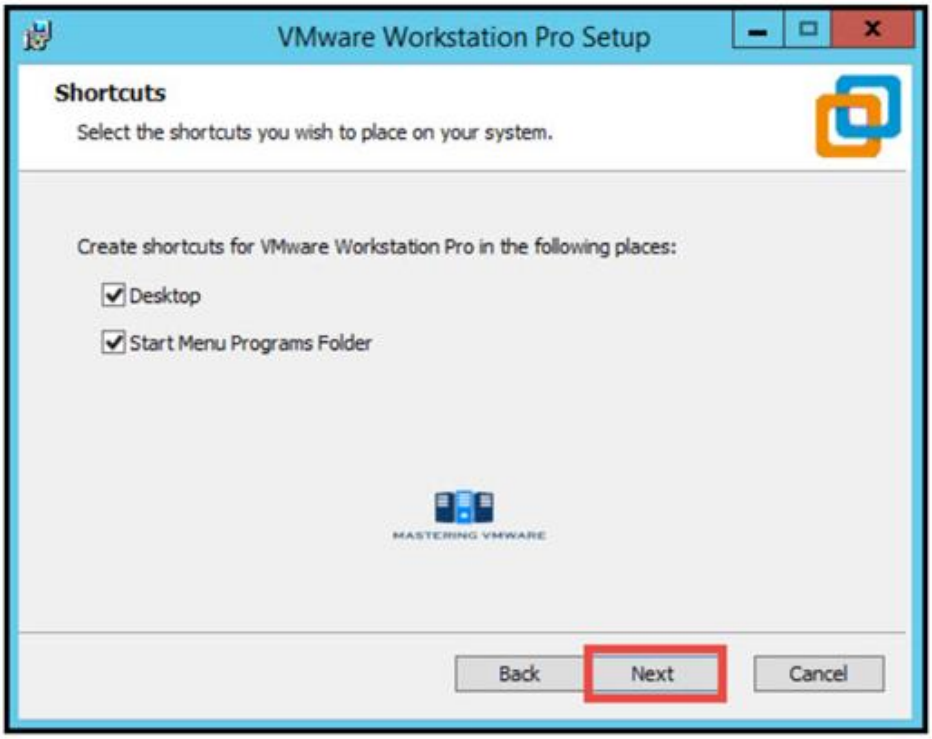
Click Next to continue.



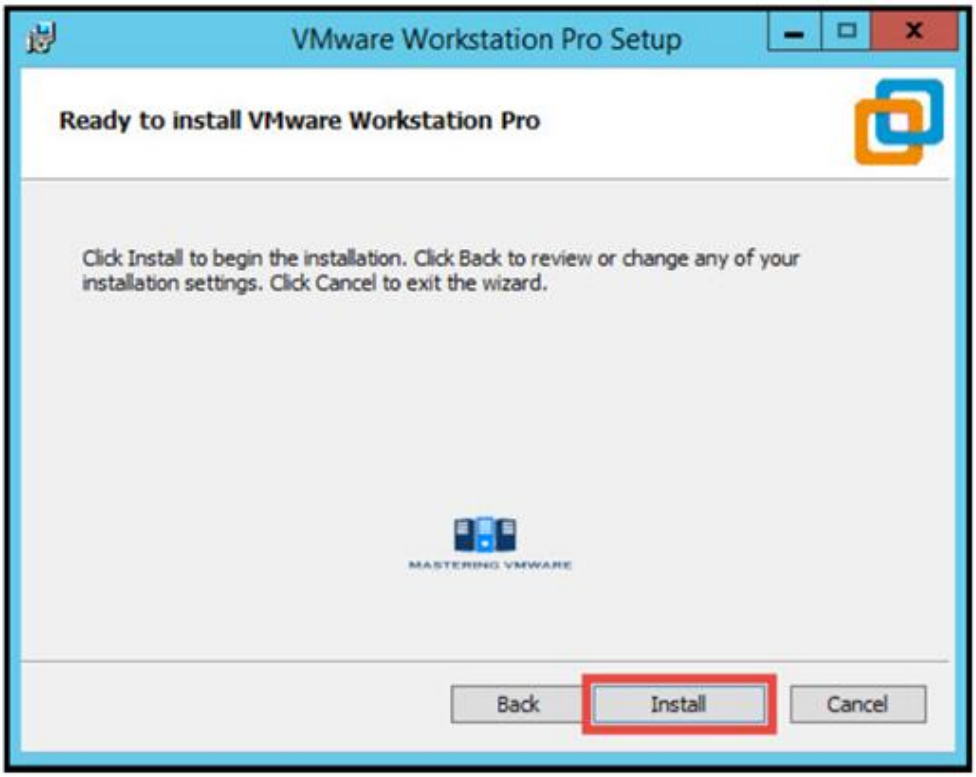
9. You can enable product startup and join the VMware Customer experience Improvement program here. Click Next to Continue.



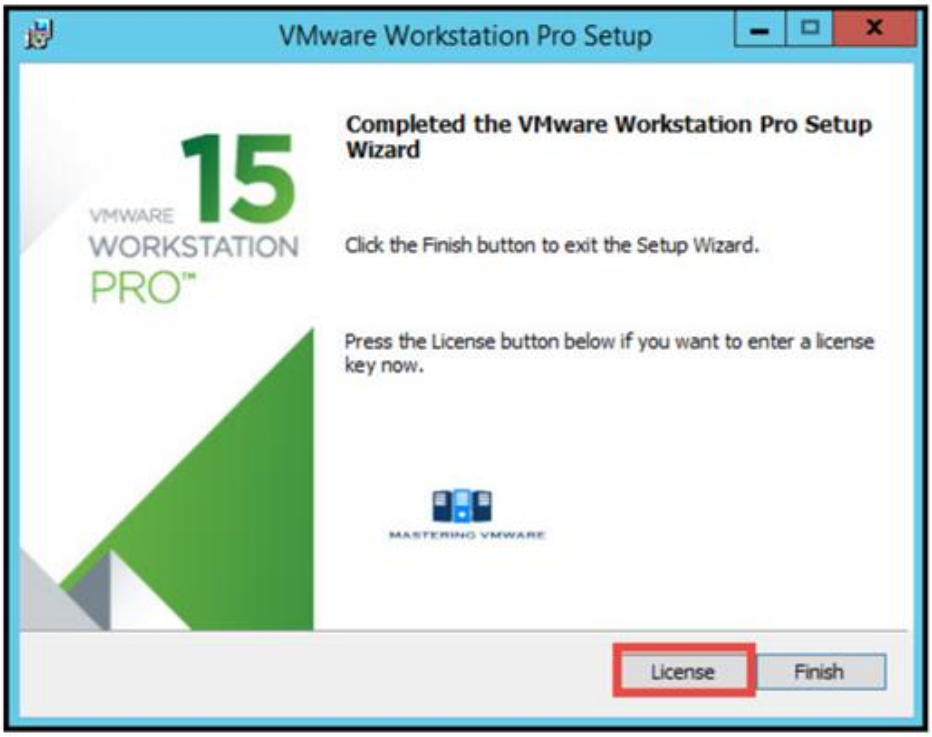
10. Select the shortcuts you want to create for easy access to VMware Workstation.Click Next to Continue.



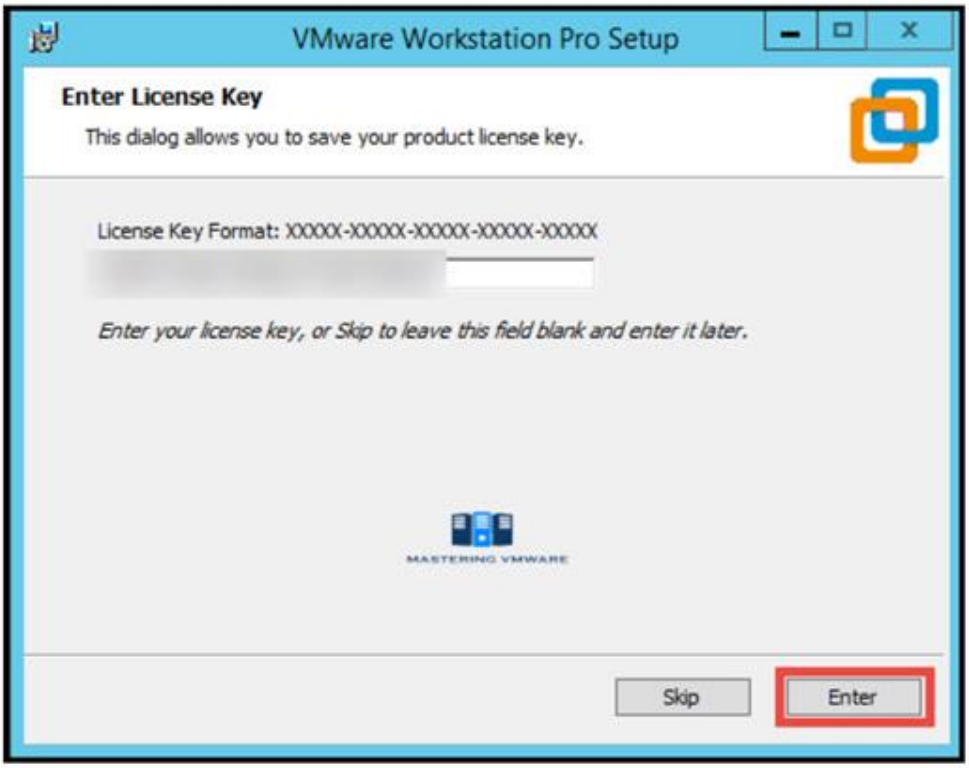
11. Click Install button to start the installation.



12. Installation will take just few seconds to complete.If you have license-key then click on License to enter the license or you can also click Finish to exit the Installer.

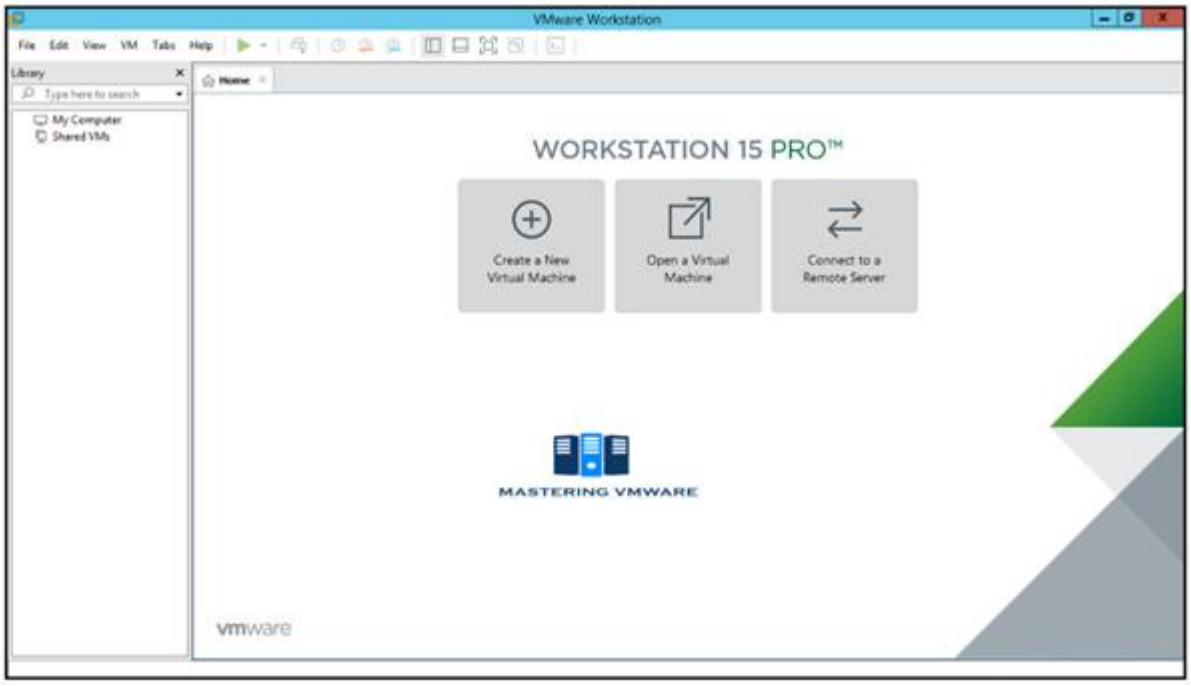


13. Provide the License Key for VMware Workstation Pro.Press Enter to continue.



14. Click Finish to exit the wizard.

15. That’s it we have successfully installed VMware Workstation Pro.Now you can start the VMware Workstation Pro by clicking on the shortcut on Desktop.Below is the Home screen of the VMware Workstation pro which you will see every time when you start Workstation.



VMware successfully setup and installed.

Practical 2:

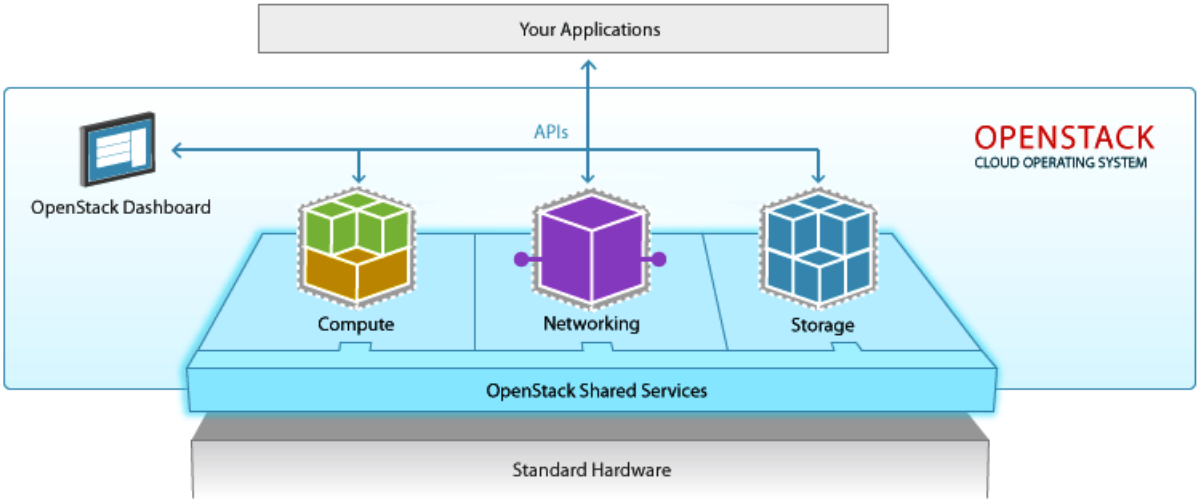
Introduction to openstack and its components.

OpenStack is an open source cloud software which consists of a series of allied projects controlling large pools of computing, storage, and network resources in a data center while managing through a dashboard. With OpenStack users can create virtual machines and other instances that do do different things in the cloud environment. It is a platform that makes horizontal scaling easy, i.e. tasks that run at the same time can easily be available to different number of users instantly by just manipulating with the instances.

According to the the National Institute of Standards and Technology (NIST), the cloud can come in three different service models:

- Cloud Software as a Service (SaaS),
- Cloud Platform as a Service (PaaS),
- Cloud Infrastructure as a Service (IaaS))

OpenStack allows users to quickly create new VM or instance upon which other cloud components can run, thus providing infrastructure. That puts OpenStack in the Cloud Infrastructure as a Service category. That infrastructure runs a “platform”, so the users can develop and deliver applications to the end users.



OpenStack helps your business run faster and delivers cost-effective infrastructure to manage data analytics, transactions, and business applications.

OpenStack Components

OpenStack consists of multiple components with a modular architecture and various code names. Let’s have a brief look at the components of OpenStack.

Compute (Nova)

OpenStack Compute is a cloud computing fabric controller, which manages pools of computer resources and work with virtualization technologies, bare metals, and high-performance computing configurations. Nova's architecture provides flexibility to design the cloud with no proprietary software or hardware requirements and also delivers the ability to integrate the legacy systems and third-party products.

Nova can be deployed using hypervisor technologies such as KVM, VMware, LXC, XenServer, etc. It is used to manage numerous virtual machines and other instances that handle various computing tasks.

Image Service (Glance)

OpenStack image service offers discovering, registering, and restoring virtual machine images. Glance has client-server architecture and delivers a user REST API, which allows querying of virtual machine image metadata and also retrieval of the actual image. While deploying new virtual machine instances, Glance uses the stored images as templates.

OpenStack Glance supports Raw, VirtualBox (VDI), VMWare (VMDK, OVF), Hyper-V (VHD), and Qemu/KVM (qcow2) virtual machine images.

Object Storage (Swift)

OpenStack Swift creates redundant, scalable data storage to store petabytes of accessible data. The stored data can be leveraged, retrieved and updated. It has a distributed architecture, providing greater redundancy, scalability, and performance, with no central point of control.

Swift is a profoundly available, shared, eventually consistent object store. It helps organizations to store lots of data safely, cheaply and efficiently. Swift ensures data replication and distribution over various devices, which makes it ideal for cost-effective, scale-out storage.

Dashboard (Horizon)

Horizon is the authorized implementation of OpenStack's Dashboard, which is the only graphical interface to automate cloud-based resources. To service providers and other commercial vendors, it supports with third party services such as monitoring, billing, and other management tools. Developers can automate tools to manage OpenStack resources using EC2 compatibility API or the native OpenStack API.

Identity Service (Keystone)

Keystone provides a central list of users, mapped against all the OpenStack services, which they can access. It integrates with existing backend services such as LDAP while acting as a common authentication system across the cloud computing system.

Keystone supports various forms of authentication like standard username & password credentials, AWS-style (Amazon Web Services) logins and token-based systems. Additionally, the catalog provides an endpoint registry with a queryable list of the services deployed in an OpenStack cloud.

Networking (Neutron)

Neutron provides networking capability like managing networks and IP addresses for OpenStack. It ensures that the network is not a limiting factor in a cloud deployment and offers users with self-service ability over network configurations. OpenStack networking allows

users to create their own networks and connect devices and servers to one or more networks. Developers can use SDN technology to support great levels of multi-tenancy and massive scale.

Neutron also offers an extension framework, which supports deploying and managing of other network services such as virtual private networks (VPN), firewalls, load balancing, and intrusion detection system (IDS)

Block Storage (Cinder)

OpenStack Cinder delivers determined block-level storage devices for application with OpenStack compute instances. A cloud user can manage their storage needs by integrating block storage volumes with Dashboard and Nova.

Cinder can use storage platforms such as Linux server, EMC (ScaleIO, VMAX, and VNX), Ceph, Coraid, CloudByte, IBM, Hitachi data systems, SAN volume controller, etc. It is appropriate for expandable file systems and database storage.

Telemetry (Ceilometer)

Ceilometer delivers a single point of contact for billing systems obtaining all of the measurements to authorize customer billing across all OpenStack core components. By monitoring notifications from existing services, developers can collect the data and may configure the type of data to meet their operating requirements.

Orchestration (Heat)

Heat is a service to orchestrate multiple composite cloud applications through both the CloudFormation-compatible Query API and OpenStack-native REST API, using the AWS CloudFormation template format.

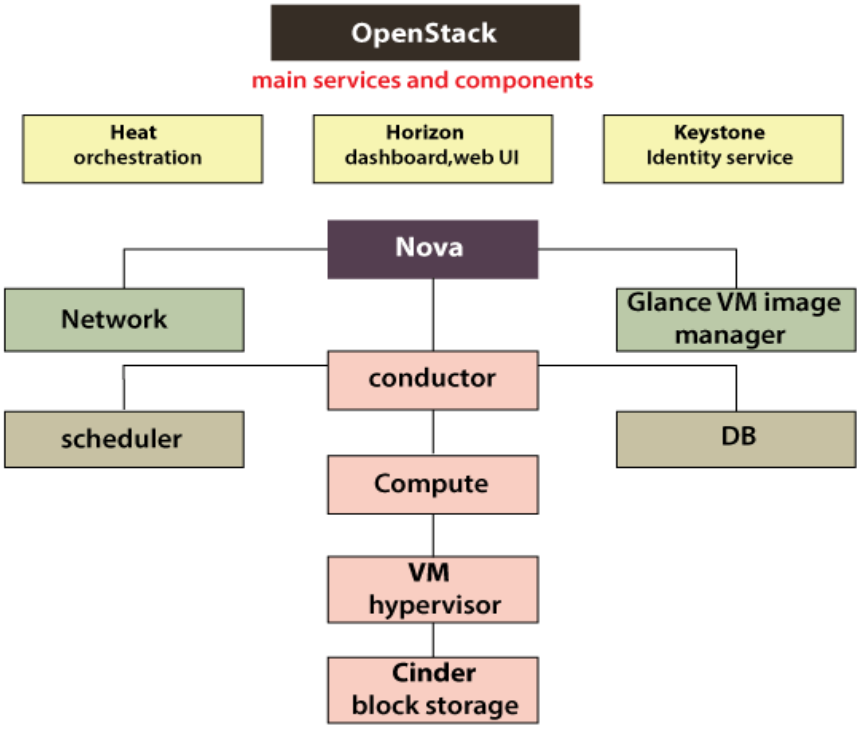


Fig: Openstack Architecture

Practical 3:

Installation of Open-Stack using Micro-Stack

What is OpenStack?

OpenStack is a collection of open-source projects designed to work together to form the basis of a cloud. OpenStack can be used for both private and public clouds.

What is Micro Stack?

Micro Stack provides a single or multi-node OpenStack deployment which can run directly on your workstation. Although made for developers to prototype and test, it is also suitable for edge, IOT, and appliances. Micro Stack is an OpenStack in a snap which means that all OpenStack services and supporting libraries are packaged together in a single package which can be easily installed, upgraded or removed. MicroStack includes all key OpenStack components: Keystone, Nova, Neutron, Glance and Cinder.

Step 1 - Install Micro-Stack

Install MicroStack from the beta channel:

```
$ sudo snap install microstack --devmode --beta
```

Step 2 – Initialise Micro-Stack

MicroStack needs to be initialised, so that netwoks and databases get configured . To do this , run:

```
$ sudo microstack init --auto --control
```

This may take about 20 minutes or more to complete, depending on your hardware specs.

```
microstack.init - INFO - Complete. Marked microstack as initialized!
```

Step 3 – Interact with Open-Stack Web UI

To interact with your cloud via web UI visit http://10.20.20.1/. The password for the admin user can be obtained in this way: sudo snap get microstack config.

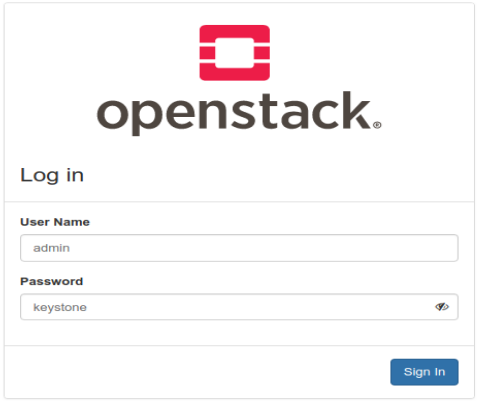
Credentials.Keystone-password

```
$ ssh -i /home/donaldsebleung/snap/microstack/common/.ssh/id_microstack cirros@10.20.20.118
```

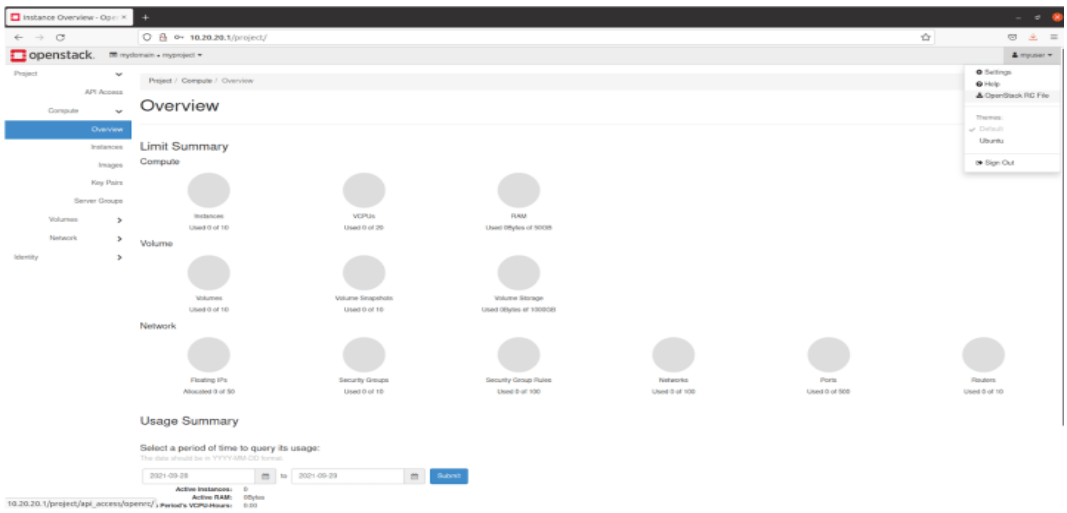
Sample output:

nkpeTcuG3As5MwkkwPdoSgYiBo8FbtvR

Type the credentials and press the “Sign In” button:



If everything goes fine you should see the landing page:



You can now start playing with your OpenStack installation(i.e., create additional users, launch instances, etc.).

CLI

You can also interact with your OpenStack cloud via the CLI by using the microstack.openstack command.

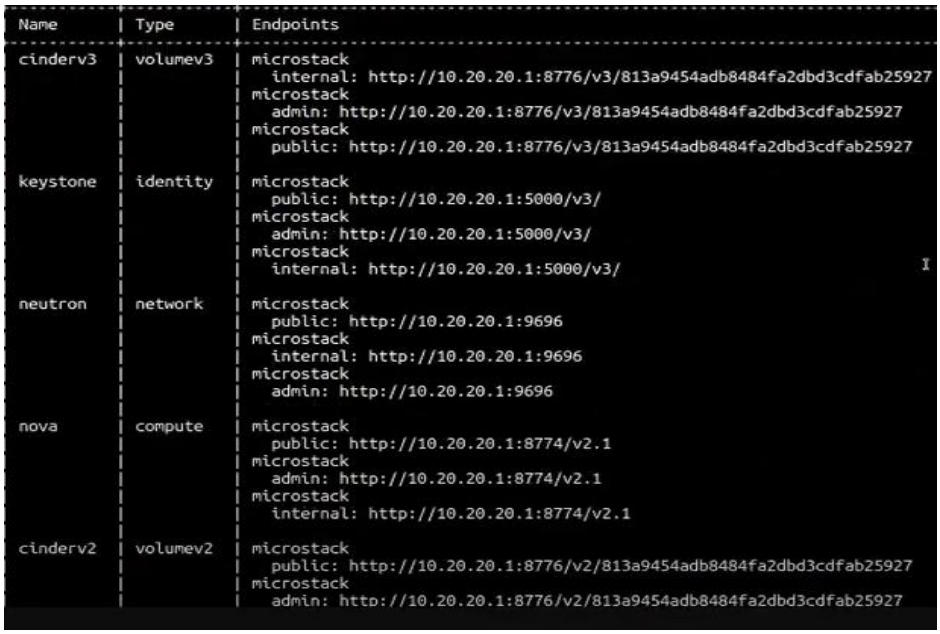
The syntax is identical to the client delievered by the python- openstackclient package.

For example, to list avaialable OpenStack endpoints

Run: microstack.openstackcatalog list

You can run microstack.openstack –help to get a list of available subcommands and their required syntax.

Step 4 – Launch and access a VM



Test launch

To launch your first OpenStack instance (VM) called “test” based on the CirrOS image, run the following:

microstack launch cirros --name test

The resulting output provides the information you need to SSH to the instance:

```
`ssh -i /home/ubuntu/snap/microstack/common/.ssh/id_microstack cirros@10.20.20.123`
```

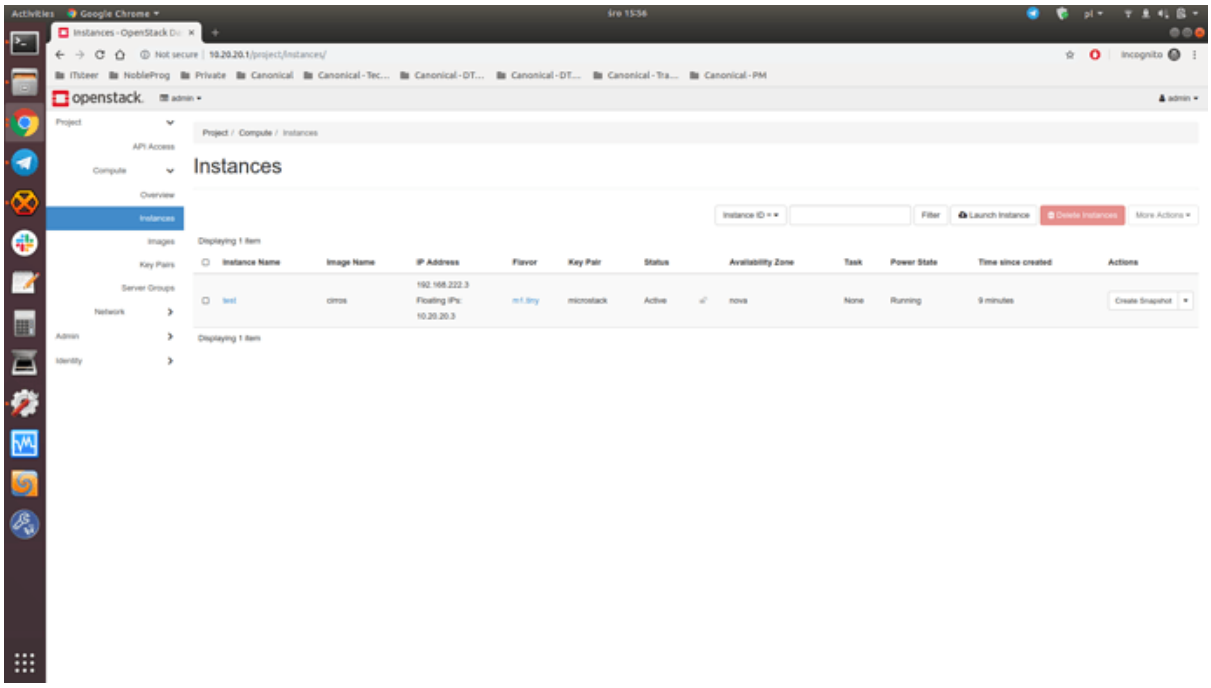
Note that the IP address of the instance may be different in your environment. In order to connect to the instance run the command from the output:

ssh -i /home/ubuntu/snap/microstack/common/.ssh/id_microstack cirros@10.20.20.123
Now that you are connected to the instance you can use normal Linux commands. Note that the CirrOS image provides a minimalist operating system! For example:

```
$ uptime
 14:51:42 up 4 min,  1 users,  load average: 0.00, 0.00, 0.00
```

To disconnect from the instance, type exit (or Ctrl-d).

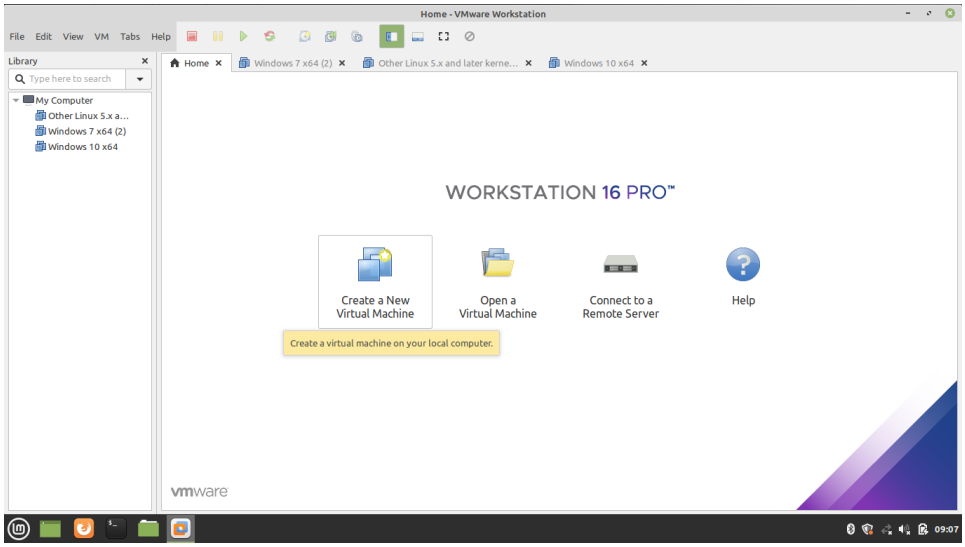
You can also view the instance from the web UI. Go to http://10.20.20.1/ and click on the “Instances” tab on the left:



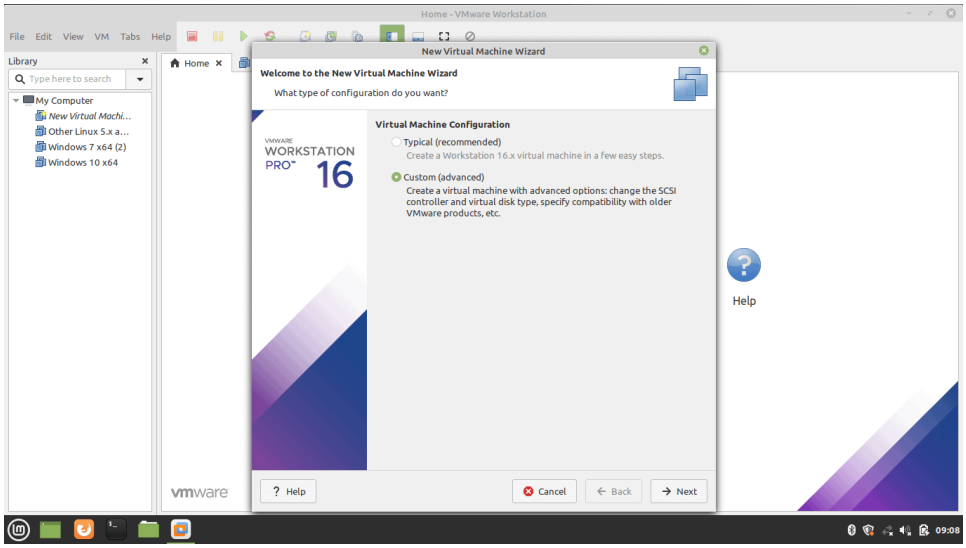
Practical 4:

Creating and launching basic virtual machine.

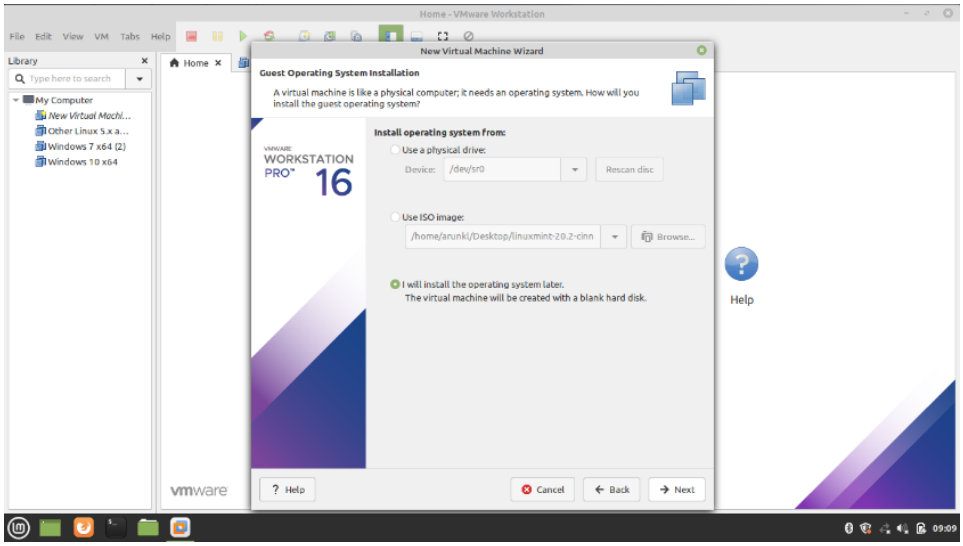
Step 1 – Start VMware Workstation
Open the app after installation. Create a new Virtual Machine.



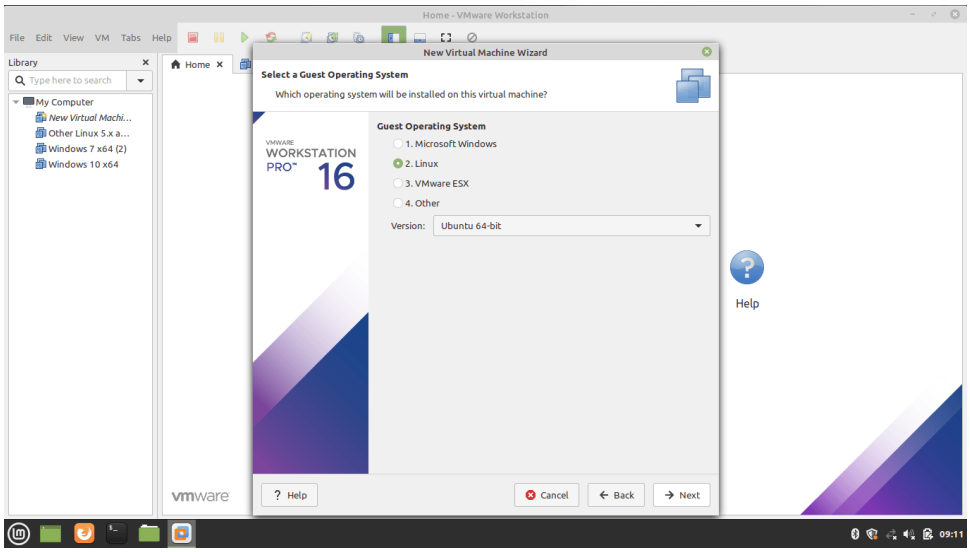
Step 2 – Select Custom Configuration Wizard
You can choose either Typical or Custom Wizard. We recommend selecting Custom if you want to install with all the configurations. If you are okay with default configurations then go ahead with Typical configurations.



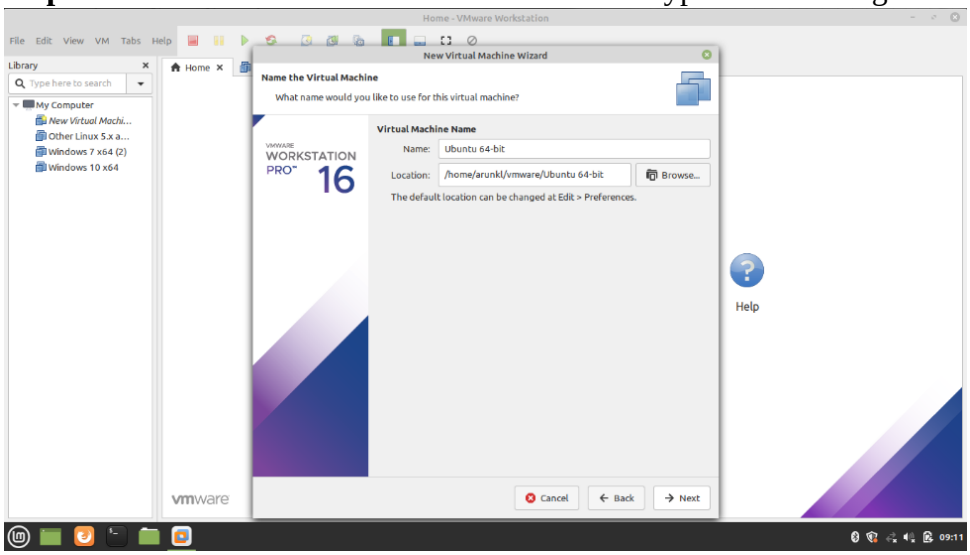
Step 3 – Select the Operating System Media
Select ‘I will install the operating system later’ for an interactive installation.



Step 4 – Select Guest Operating System.In our case we are selecting Linux operating system.

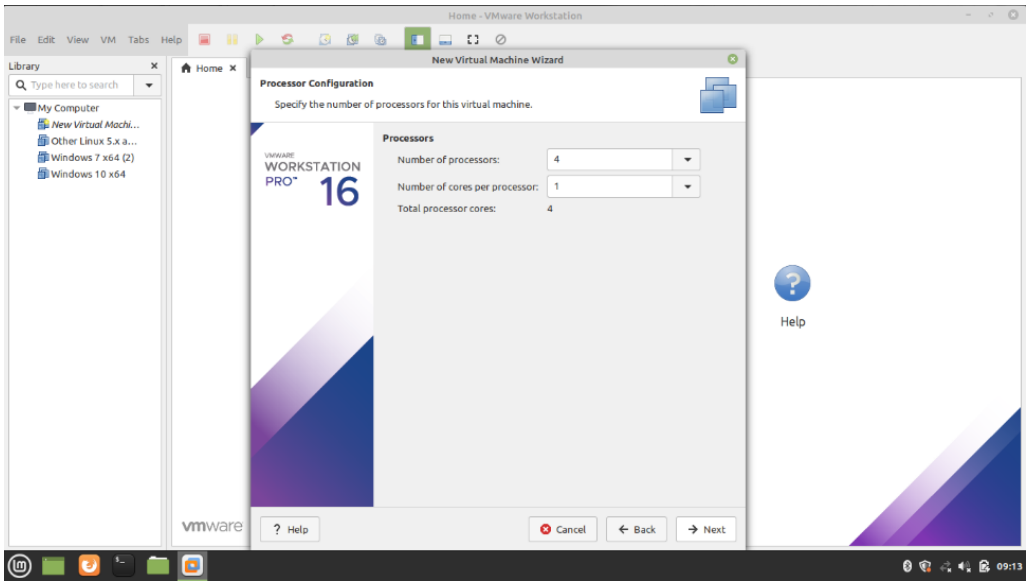


Step 5 – Name the Virtual Machine and location .Type a name and give the location details.



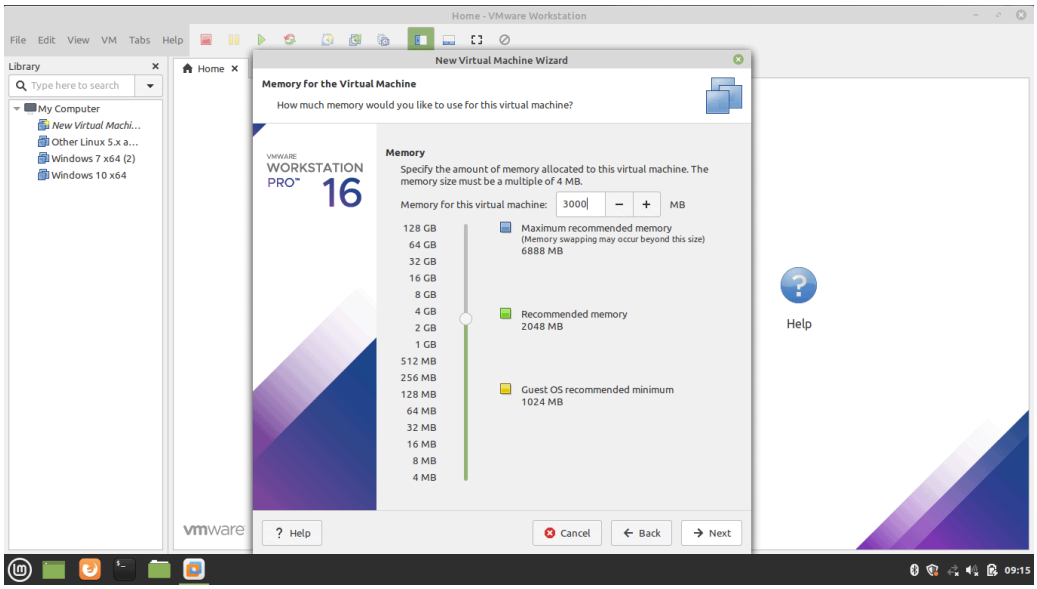
Step 6 – Allocate the Processors

Assign the processors, Calculate the processor required to run the host machine. Assign the leftover resources to the virtual machine.



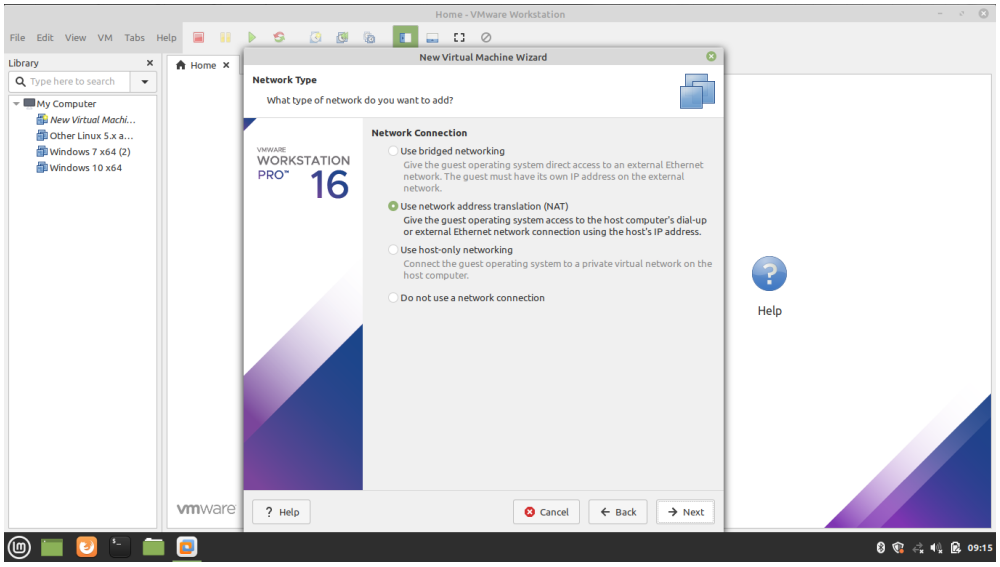
Step 7 – Allocate the Memory for Virtual Machine

Memory allocation calculation is the same as the processor allocation. Leave sufficient memory for the host system and allocate the remaining memory for the virtual machine.

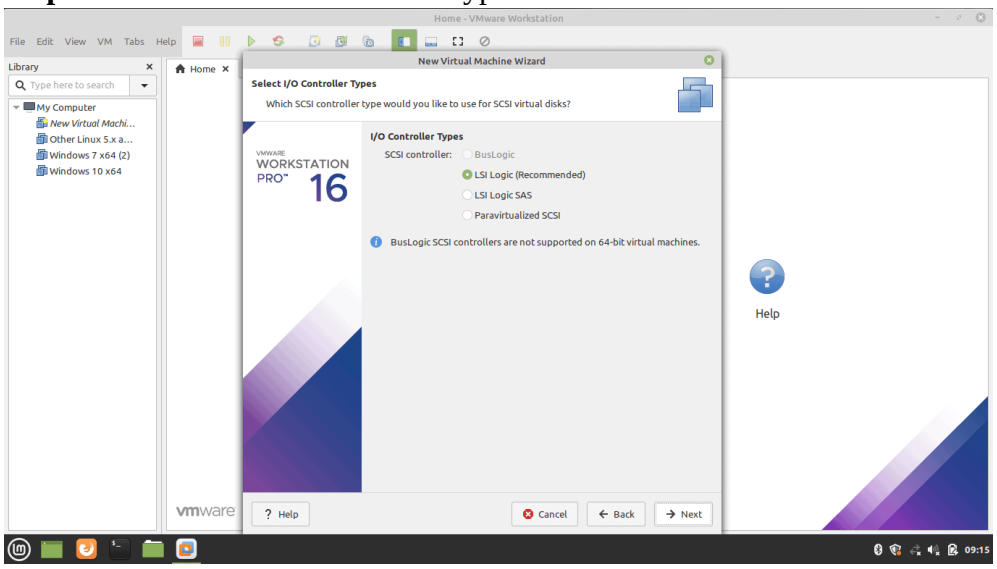


Step 8 – Choose the Network Configuration

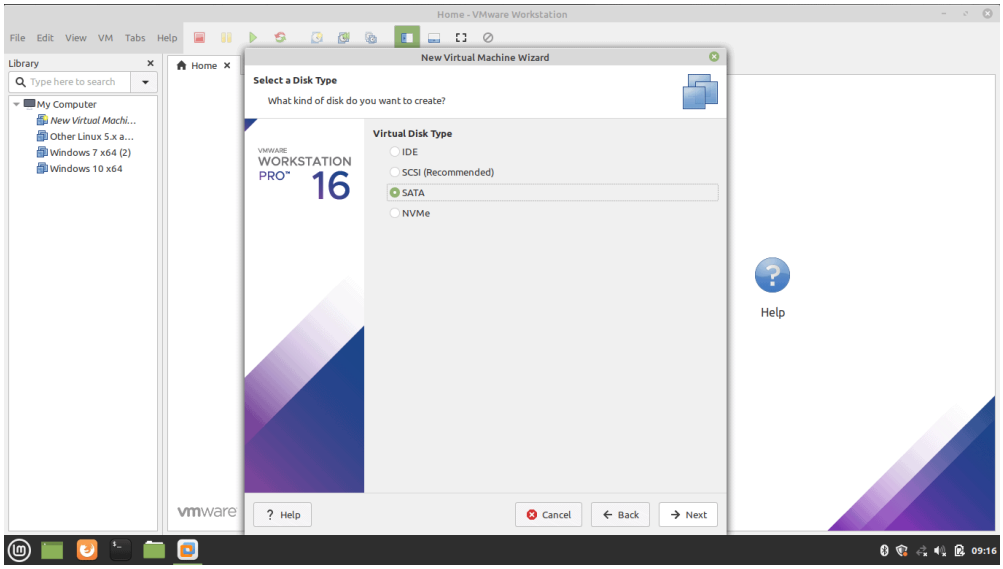
Select any one of the network configurations as per your requirement.



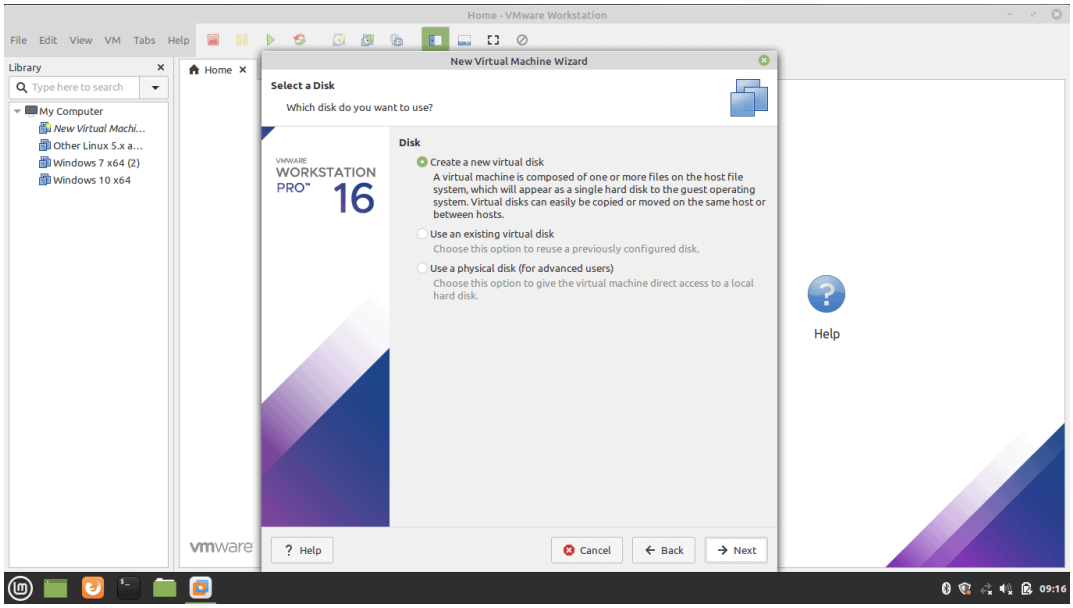
Step 9 – Select the I/O Controller Type



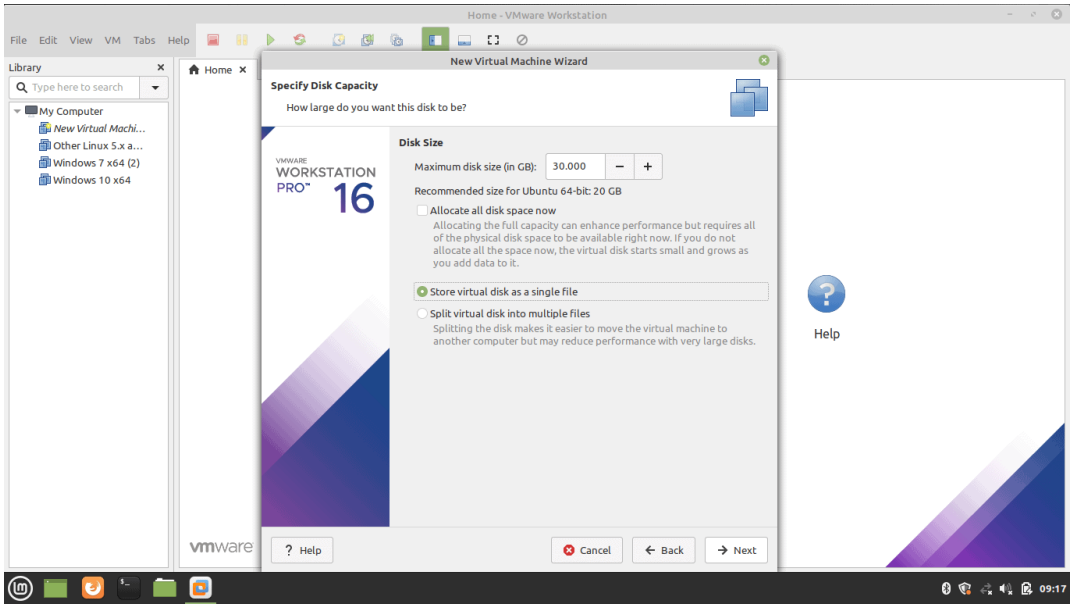
Step 10 – Select Disk Type



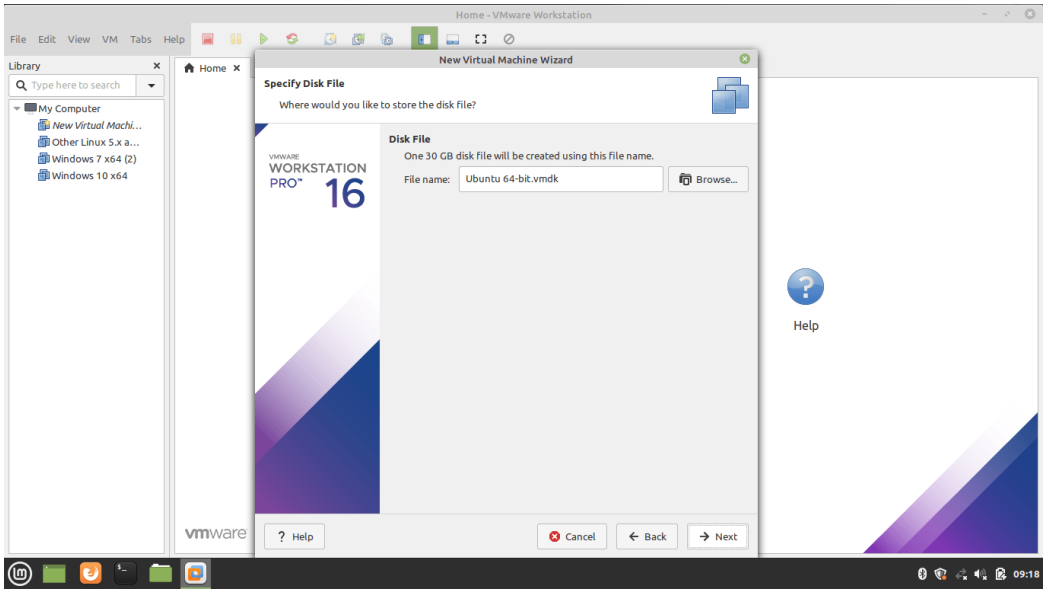
Step 11 – Select Virtual Disk
Select the Virtual Disk if you have or create one.



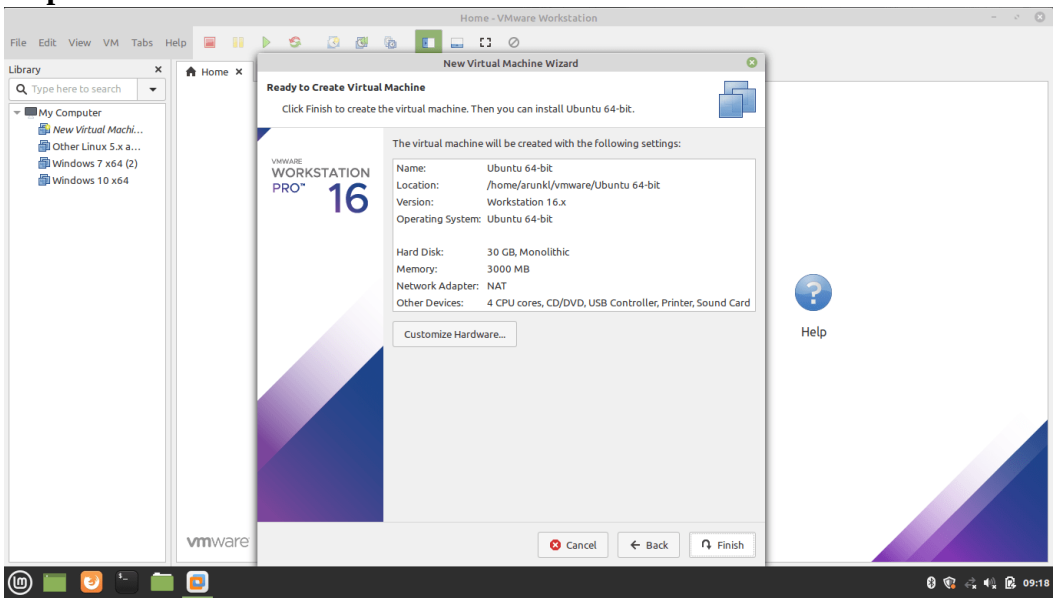
Step 12 – Select Disk Capacity
Select the disk size. Selecting a single disk will increase the performance. However, selecting a split disk will help in the disk transfer scenario.



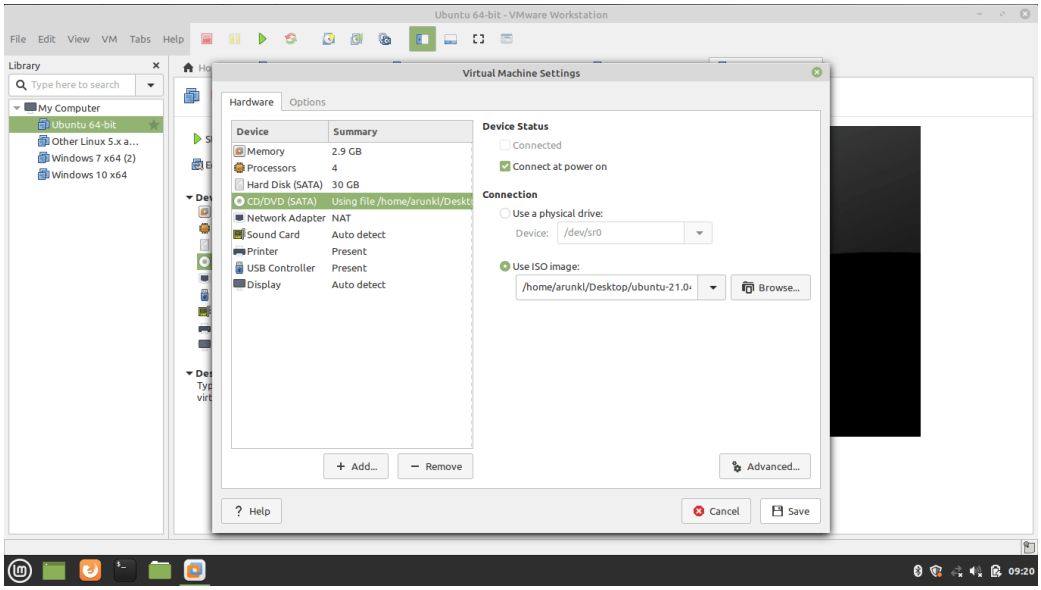
Step 13 – Select Virtual Disk File
Select the Virtual Disk File which is usually the iso file of the operating system.



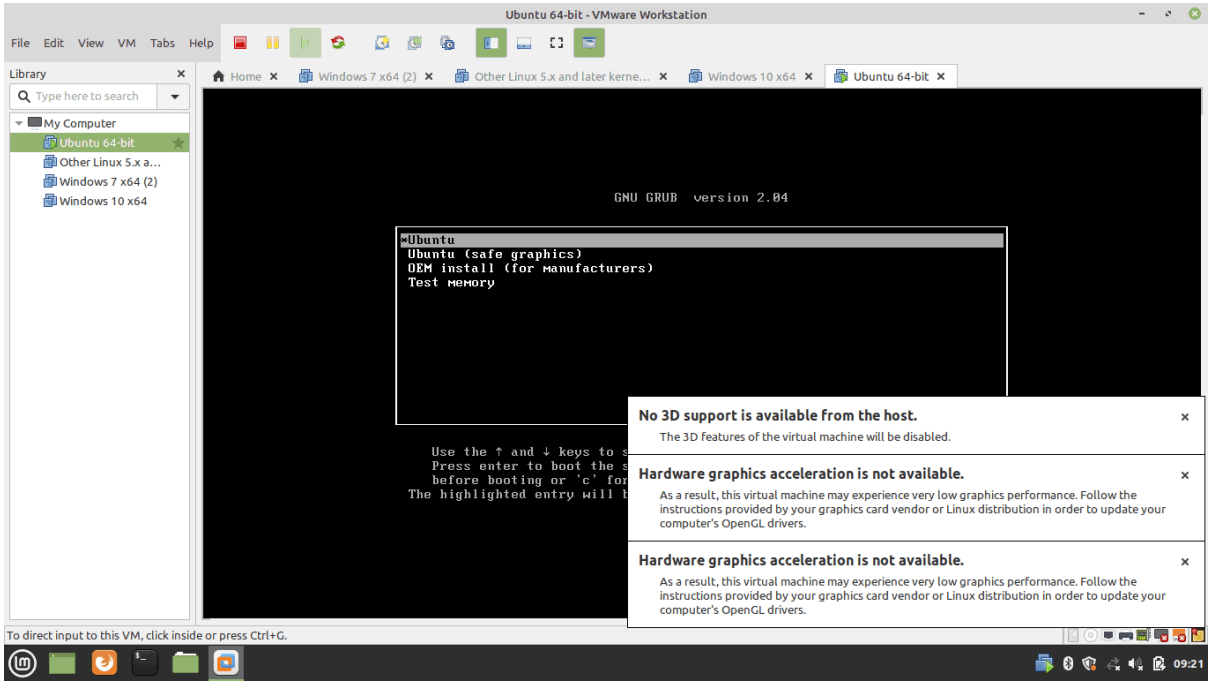
Step 14 – Create Virtual Machine.Now click on click virtual machine button



Step 15 – Supply Ubuntu ISO Image to Virtual Machine.Download Ubuntu image. Edit the CD/DVD settings and import the downloaded Ubuntu image.



Step 16 – Power On the Virtual Machine.
Press the Play button to power on the Virtual Machine.



Practical 5:

Creating and Managing Images and Templates

The cloud operator assigns roles to users. Roles determine who can upload and manage images. The operator might restrict image upload and management to only cloud administrators or operators.

You can upload images through the openstack image create command or the image service API. You can use the openstack client for the image management. It provides mechanisms to list and delete images, set and delete image metadata, and create images of a running instance or snapshot and backup types.

After you upload an image, you cannot change it.

Manage images

1. **List or get details for images(glance)**
   To get a list of images and to get further details about a single image, use openstack image list and openstack image show commands.

```
$ openstack image list
+--------------------------------------+---------------------------------+--------+
| ID                                   | Name                            | Status |
+--------------------------------------+---------------------------------+--------+
| dfc1dfb0-d7bf-4fff-8994-319dd6f703d7 | cirros-0.3.5-x86_64-uec         | active |
| a3867e29-c7a1-44b0-9e7f-10db587cad20 | cirros-0.3.5-x86_64-uec-kernel  | active |
| 4b916fba-6775-4092-92df-f41df7246a6b | cirros-0.3.5-x86_64-uec-ramdisk | active |
| d07831df-edc3-4817-9881-89141f9134c3 | myCirrosImage                   | active |
+--------------------------------------+---------------------------------+--------+
```

   When viewing a list of images, you can also use grep to filter the list as follows:

```
$ openstack image list | grep 'cirros'
| dfc1dfb0-d7bf-4fff-8994-319dd6f703d7 | cirros-0.3.5-x86_64-uec         | active |
| a3867e29-c7a1-44b0-9e7f-10db587cad20 | cirros-0.3.5-x86_64-uec-kernel  | active |
| 4b916fba-6775-4092-92df-f41df7246a6b | cirros-0.3.5-x86_64-uec-ramdisk | active |
```

2. Create or update an image(glance)
   To create an image, use openstack image create

```
$ openstack image create imageName
```

   To update an image by name or ID, use openstack image set:

```
$ openstack image set imageName
```

   The following list explains the optional arguments that you can use with the create and set commands to modify image properties. For more information, refer to the Openstack Image command reference.

The following example shows the command that you would use to upload a CentOs 6.3 image qcow2 format and configure it for public access:

```
$ openstack image create --disk-format qcow2 --container-format bare \
  --public --file ./centos63.qcow2 centos63-image
```

The following example shows how to update an existing image with a properties that describe the disk bus, the CD-ROM bus, and the VIF model:

```
$ openstack image set \
    --property hw_disk_bus=scsi \
    --property hw_cdrom_bus=ide \
    --property hw_vif_model=e1000 \
    f16-x86_64-openstack-sda
```

3. Create an image from ISO image
   You can upload ISO images to the Image service(glance). You can subsequently boot an ISO image using Compute.
   ISO image using Compute.
   In the Image service, run the following command:

```
$ openstack image create ISO_IMAGE --file IMAGE.iso \
  --disk-format iso --container-format bare
```

Operationally, to confirm the upload in Image service, run:

```
$ openstack image list
```

Practical 6:

Creating and Managing networks.

Create Networks:

1. List the extensions of the system

```
$ openstack extension list -c Alias -c Name --network
+---------------------------------------+---------------------------+
| Name                                  | Alias                     |
+---------------------------------------+---------------------------+
| Default Subnetpools                   | default-subnetpools       |
| Network IP Availability               | network-ip-availability   |
| Auto Allocated Topology Services      | auto-allocated-topology   |
| Neutron L3 Configurable external gateway | ext-gw-mode               |
| Address scope                         | address-scope             |
| Neutron Extra Route                   | extraroute                |
+---------------------------------------+---------------------------+
```

2. Create a network

```
$ openstack network create net1
Created a new network:
+---------------------------+--------------------------------------+
| Field                     | Value                                |
+---------------------------+--------------------------------------+
| admin_state_up            | UP                                   |
| availability_zone_hints   |                                      |
| availability_zones        |                                      |
| created_at                | 2016-12-21T08:32:54Z                 |
| description               |                                      |
| headers                   |                                      |
| id                        | 180620e3-9eae-4ba7-9739-c5847966e1f0 |
| ipv4_address_scope        | None                                 |
| ipv6_address_scope        | None                                 |
| mtu                       | 1450                                 |
| name                      | net1                                 |
| port_security_enabled     | True                                 |
| project_id                | c961a8f6d3654657885226378ade8220     |
| provider:network_type     | vxlan                                |
| provider:physical_network | None                                 |
| provider:segmentation_id  | 14                                   |
| revision_number           | 3                                    |
| router:external           | Internal                             |
| shared                    | False                                |
| status                    | ACTIVE                               |
| subnets                   |                                      |
| tags                      | []                                   |
| updated_at                | 2016-12-21T08:32:54Z                 |
+---------------------------+--------------------------------------+
```

3. Create a network with specified provider network type

```
$ openstack network create net2 --provider-network-type vxlan
Created a new network:
+---------------------------+--------------------------------------+
| Field                     | Value                                |
+---------------------------+--------------------------------------+
| admin_state_up            | UP                                   |
| availability_zone_hints   |                                      |
| availability_zones        |                                      |
| created_at                | 2016-12-21T08:33:34Z                 |
| description               |                                      |
| headers                   |                                      |
| id                        | c0a563d5-ef7d-46b3-b30d-6b9d4138b6cf |
| ipv4_address_scope        | None                                 |
| ipv6_address_scope        | None                                 |
| mtu                       | 1450                                 |
| name                      | net2                                 |
| port_security_enabled     | True                                 |
| project_id                | c961a8f6d3654657885226378ade8220     |
| provider:network_type     | vxlan                                |
| provider:physical_network | None                                 |
| provider:segmentation_id  | 87                                   |
| revision_number           | 3                                    |
| router:external           | Internal                             |
| shared                    | False                                |
| status                    | ACTIVE                               |
| subnets                   |                                      |
| tags                      | []                                   |
| updated_at                | 2016-12-21T08:33:34Z                 |
+---------------------------+--------------------------------------+
```

Create Subnets:

1. Create a subnet

```
$ openstack subnet create subnet1 --network net1
  --subnet-range 192.0.2.0/24
+-------------------+--------------------------------------+
| Field             | Value                                |
+-------------------+--------------------------------------+
| allocation_pools  | 192.0.2.2-192.0.2.254                |
| cidr              | 192.0.2.0/24                         |
| created_at        | 2016-12-22T18:47:52Z                 |
| description       |                                      |
| dns_nameservers   |                                      |
| enable_dhcp       | True                                 |
| gateway_ip        | 192.0.2.1                            |
| headers           |                                      |
| host_routes       |                                      |
| id                | a394689c-f547-4834-9778-3e0bb22130dc |
| ip_version        | 4                                    |
| ipv6_address_mode | None                                 |
| ipv6_ra_mode      | None                                 |
| name              | subnet1                              |
| network_id        | 180620e3-9eae-4ba7-9739-c5847966e1f0 |
| project_id        | e17431afc0524e0690484889a04b7fa0     |
| revision_number   | 2                                    |
| service_types     |                                      |
| subnetpool_id     | None                                 |
| updated_at        | 2016-12-22T18:47:52Z                 |
+-------------------+--------------------------------------+
```

Create routers:

1. Create a router

```
$ openstack router create router1
+-------------------------+--------------------------------------+
| Field                   | Value                                |
+-------------------------+--------------------------------------+
| admin_state_up          | UP                                   |
| availability_zone_hints |                                      |
| availability_zones      |                                      |
| created_at              | 2016-12-22T18:48:57Z                 |
| description             |                                      |
| distributed             | True                                 |
| external_gateway_info   | null                                 |
| flavor_id               | None                                 |
| ha                      | False                                |
| headers                 |                                      |
| id                      | e25a24ee-3458-45c7-b16e-edf49092aab7 |
| name                    | router1                              |
| project_id              | e17431afc0524e0690484889a04b7fa0     |
| revision_number         | 1                                    |
| routes                  |                                      |
| status                  | ACTIVE                               |
| updated_at              | 2016-12-22T18:48:57Z                 |
+-------------------------+--------------------------------------+
```

2. Link router of external provider network

```
$ openstack router set ROUTER --external-gateway NETWORK
```

3. Link the router to subnet

```
$ openstack router add subnet ROUTER SUBNET
```

Create Ports:

1. Create a port with specified IP address

```
$ openstack port create --network net1 --fixed-ip subnet=subnet1,ip-address=192.0.2.40 port1
+-----------------------+--------------------------------------+
| Field                 | Value                                |
+-----------------------+--------------------------------------+
| admin_state_up        | UP                                   |
| allowed_address_pairs |                                      |
| binding_host_id       |                                      |
| binding_profile       |                                      |
| binding_vif_details   |                                      |
| binding_vif_type      | unbound                              |
| binding_vnic_type     | normal                               |
| created_at            | 2016-12-22T18:54:43Z                 |
| description           |                                      |
| device_id             |                                      |
| device_owner          |                                      |
| extra_dhcp_opts       |                                      |
| fixed_ips             | ip_address='192.0.2.40', subnet_id='a |
|                       | 394689c-f547-4834-9778-3e0bb22130dc' |
| headers               |                                      |
| id                    | 031ddba8-3e3f-4c3c-ae26-7776905eb24f |
| mac_address           | fa:16:3e:df:3d:c7                    |
| name                  | port1                                |
| network_id            | 180620e3-9eae-4ba7-9739-c5847966e1f0 |
| port_security_enabled | True                                 |
| project_id            | e17431afc0524e0690484889a04b7fa0     |
| revision_number       | 5                                    |
| security_groups       | 84abb9eb-dc59-40c1-802c-4e173c345b6a |
| status                | DOWN                                 |
| updated_at            | 2016-12-22T18:54:44Z                 |
+-----------------------+--------------------------------------+
```

2. Create a port without specified IP address

```
$ openstack port create port2 --network net1
+-----------------------+--------------------------------------+
| Field                 | Value                                |
+-----------------------+--------------------------------------+
| admin_state_up        | UP                                   |
| allowed_address_pairs |                                      |
| binding_host_id       |                                      |
| binding_profile       |                                      |
| binding_vif_details   |                                      |
| binding_vif_type      | unbound                              |
| binding_vnic_type     | normal                               |
| created_at            | 2016-12-22T18:56:06Z                 |
| description           |                                      |
| device_id             |                                      |
| device_owner          |                                      |
| extra_dhcp_opts       |                                      |
| fixed_ips             | ip_address='192.0.2.10', subnet_id='a |
|                       | 394689c-f547-4834-9778-3e0bb22130dc'  |
| headers               |                                      |
| id                    | eac47fcd-07ac-42dd-9993-5b36ac1f201b |
| mac_address           | fa:16:3e:96:ae:6e                    |
| name                  | port2                                |
| network_id            | 180620e3-9eae-4ba7-9739-c5847966e1f0 |
| port_security_enabled | True                                 |
| project_id            | e17431afc0524e0690484889a04b7fa0     |
| revision_number       | 5                                    |
| security_groups       | 84abb9eb-dc59-40c1-802c-4e173c345b6a |
| status                | DOWN                                 |
| updated_at            | 2016-12-22T18:56:06Z                 |
+-----------------------+--------------------------------------+
```

3. Query a port with fixed specified IP address

```
$ neutron port-list --fixed-ips ip_address=192.0.2.2 \
  ip_address=192.0.2.40
+---------------+------+-------------------+-------------------------------------------+
| id            | name | mac_address       | fixed_ips                                 |
+---------------+------+-------------------+-------------------------------------------+
| baf13412-26... |      | fa:16:3e:f6:ec:c7 | {"subnet_id"... ..."ip_address": "192.0.2.2"}   |
| f7a08fe4-e7... |      | fa:16:3e:97:e0:fc | {"subnet_id"... ..."ip_address": "192.0.2.40"}  |
+---------------+------+-------------------+-------------------------------------------+
```

Practical 7:

Creating and Managing Users

List Users

1. List all users:

```
$ openstack user list
+----------------------------------+----------+
| ID                               | Name     |
+----------------------------------+----------+
| 352b37f5c89144d4ad0534139266d51f | admin    |
| 86c0de739bcb4802b8dc786921355813 | demo     |
| 32ec34aae8ea432e8af560a1cec0e881 | glance   |
| 7047fcb7908e420cb36e13bbd72c972c | nova     |
+----------------------------------+----------+
```

2. Create a user:
    a. To create a user, you must specify a name. Optionally, you can specify a project ID, password, and email address. It is recommended that you include the project ID and password because the user cannot log in to the dashboard without this information.

```
$ openstack user create --project new-project --password PASSWORD new-user
+------------+----------------------------------+
| Field      | Value                            |
+------------+----------------------------------+
| email      | None                             |
| enabled    | True                             |
| id         | 6322872d9c7e445dbbb49c1f9ca28adc |
| name       | new-user                         |
| project_id | 0b0b995694234521bf93c792ed44247f |
| username   | new-user                         |
+------------+----------------------------------+
```

3. Update a user:
    a. To temporarily disable a user account:

```
$ openstack user set USER_NAME --disable
```

    b. To enable a disabled user account:

```
$ openstack user set USER_NAME --enable
```

c. To change the name and description for a user account:

```
$ openstack user set USER_NAME --name user-new --email new-user@example.com
User has been updated.
```

4. Delete a user:
   a. Delete a specified user account:

```
$ openstack user delete USER_NAME
```

5. Assigning role to user:

```
$ openstack role add --user USER_NAME --project TENANT_ID ROLE_NAME
```

Practical 8:

Managing security groups and policies

1. List and View Current Security Groups:

```
export OS_USERNAME=demo00
export OS_TENANT_NAME=tenant01
```

```
$ openstack security group list
+--------------------------------------+---------+-------------+
| Id                                   | Name    | Description |
+--------------------------------------+---------+-------------+
| 73580272-d8fa-4927-bd55-c85e43bc4877 | default | default     |
| 6777138a-deb7-4f10-8236-6400e7aff5b0 | open    | all ports   |
+--------------------------------------+---------+-------------+
```

```
$ openstack security group rule list GROUPNAME
```

2. Create a security group:
    a. Ensure your system variables are set for the user and project for which you are creating security group rules.
    b. Add the new security group, as follows:

```
$ openstack security group create GroupName --description Description
```

    c. Add a new group rule, as follows:

```
$ openstack security group rule create SEC_GROUP_NAME \
    --protocol PROTOCOL --dst-port FROM_PORT:TO_PORT --remote-ip CIDR
```

    d. View all rules for the new security group, as follows:

```
$ openstack security group rule list global_http
+--------------------------------------+-------------+-----------+-----------------+-----------------------+
| ID                                   | IP Protocol | IP Range  | Port Range      | Remote Security Group |
+--------------------------------------+-------------+-----------+-----------------+-----------------------+
| 353d0611-3f67-4848-8222-a92adbdb5d3a | tcp         | 0.0.0.0/0 | 80:80           | None                  |
| 63536865-e5b6-4df1-bac5-ca6d97d8f54d | tcp         | 0.0.0.0/0 | 443:443         | None                  |
+--------------------------------------+-------------+-----------+-----------------+-----------------------+
```

3. Delete a security group:
    a. Ensure your system variables are set for the user and project for which you are deleting a security group.To enable a disabled user account
    b. Delete the new security group, as follows:

```
$ openstack security group delete GROUPNAME
```

4. Create security group rules for a cluster of instances:
    a. Make sure to set the system variables for the user and project for which you are creating a security group rule.
    b. Add a source group, as follows:

```
$ openstack security group rule create secGroupName \
    --remote-group source-group --protocol ip-protocol \
    --dst-port from-port:to-port
```

5. Create and manage security group rules
    a. To list the rules for a security group, run the following command:

```
$ openstack security group rule list SECURITY_GROUP_NAME
```

    b. To allow SSH access to the instances, choose one of the following options:
        i. Allow access from all IP address, specified as IP subnet

```
$ openstack security group rule create SECURITY_GROUP_NAME \
      --protocol tcp --dst-port 22:22 --remote-ip 0.0.0.0/0
```

        ii. Allow access only from IP address from other security groups to access the specified port:

```
$ openstack security group rule create SECURITY_GROUP_NAME \
      --protocol tcp --dst-port 22:22 --remote-group SOURCE_GROUP_NAME
```

    c. To allow pinging of the instances, choose one of the following options:
        i. Allow pinging from all IP addresses, specified as IP subnet

```
$ openstack security group rule create --protocol icmp \
  SECURITY_GROUP_NAME
```

        ii. Allow only members of other security groups to ping instances

```
$ openstack security group rule create --protocol icmp \
  --remote-group SOURCE_GROUP_NAME SECURITY_GROUP
```

    d. To allow access through a UDP port, such as allowing to a DNS server that runs on a VM, choose one of the following options:
        i. Allow UDP access from IP address specidief as IP subnet

```
$ openstack security group rule create --protocol udp \
  --dst-port 53:53 SECURITY_GROUP
```

ii. Allow only IP addresses from other security groups to access the specified port

```
$ openstack security group rule create --protocol udp \
  --dst-port 53:53 --remote-group SOURCE_GROUP_NAME SECURITY_GROUP
```

6. Delete a security group rule
    a. To delete a security group rule, specify the ID of the rule

```
$ openstack security group rule delete RULE_ID
```