[2CEIT603: CLOUD COMPUTING]

Practical: 11

AIM- Study and Explore Architecture and working of Open Stack.

Submitted By:Adeshara Brijesh

Enrollment number: 21012021001

Department of Computer Engineering/ Information Technology

1. INTRODUCTION OF OPEN STACK:

- **What is OpenStack?**

OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed and provisioned through APIs with common authentication mechanisms. OpenStack is a cloud OS that is used to control the large pools of computing, storage, and networking resources within a data center. OpenStack is an open-source and free software platform. This is essentially used and implemented as an IaaS for cloud computing.

A dashboard is also available, giving administrators control while empowering their users to provision resources through a web interface.

Beyond standard infrastructure-as-a-service functionality, additional components provide orchestration, fault management and service management amongst other services to ensure high availability of user applications.

We can call the OpenStack a software platform that uses pooled virtual resources to create and manage private and public cloud. OpenStack offers many cloud-related services (such as networking, storage, image services, identity, etc.) by default. This can be handled by users through a web-based dashboard, a RESTful API, or command-line tools. OpenStack manages a lot of virtual machines; this permits the usage of physical resources to be reduced.

- **Basic Principles of OpenStack**

Open Source: Under the Apache 2.0 license, OpenStack is coded and published. Apache allows the community to use it for free.

Open Design: For the forthcoming update, the development group holds a Design Summit every 6 months.

Open Development: The developers maintain a source code repository that is freely accessible through projects like the Ubuntu Linux distribution via entig100s.

Open Community: OpenStack allows open and transparent documentation for the community.

2. PURPOSE OF OPEN STACK:

- **OpenStack's Primary Purpose:**

Build and manage private and public clouds: Organizations can create their own cloud infrastructure (private cloud) or offer cloud services to others (public cloud) using OpenStack.

Enable Infrastructure-as-a-Service (IaaS): OpenStack provides the core functionalities for IaaS, allowing users to provision and manage VMs, storage, and networking on-demand.

Increase resource utilization: By pooling resources, OpenStack allows for better utilization of hardware, reducing waste and optimizing costs.

Offer greater flexibility and control: Organizations have more control over their cloud environment compared to using public cloud providers.

3. COMPONENTS OF OPENSTACK:

- **Major Components of OpenStack are given below:**

Compute (Nova): Compute is a controller that is used to manage resources in virtualized environments. It handles several virtual machines and other instances that perform computing tasks.

Object Storage (Swift): To store and retrieve arbitrary data in the cloud, object storage is used. In Swift, it is possible to store the files, objects, backups, images, videos, virtual machines, and other unstructured data. Developers may use a special identifier for referring the file and objects in place of the path, which directly points to a file and allows the OpenStack to manage where to store the files.

Block Storage (Cinder): This works in the traditional way of attaching and detaching an external hard drive to the OS for its local use. Cinder manages to add, remove, create new disk space in the server. This component provides the virtual storage for the virtual machines in the system.

Networking (Neutron): This component is used for networking in OpenStack. Neutron manages all the network-related queries, such as IP address management, routers, subnets, firewalls, VPNs, etc. It confirms that all the other components are well connected with the OpenStack.

Dashboard (Horizon): This is the first component that the user sees in the OpenStack. Horizon is the web UI (user interface) component used to access the other back-end services. Through individual API (Application programming interface), developers can access the OpenStack's components, but through the dashboard, system administrators can look at what is going on in the cloud and manage it as per their need.

Identity Service (Keystone): It is the central repository of all the users and their permissions for the OpenStack services they use. This component is used to manage identity services like authorization, authentication, AWS Styles (Amazon Web Services) logins, token-based systems, and checking the other credentials (username & password).

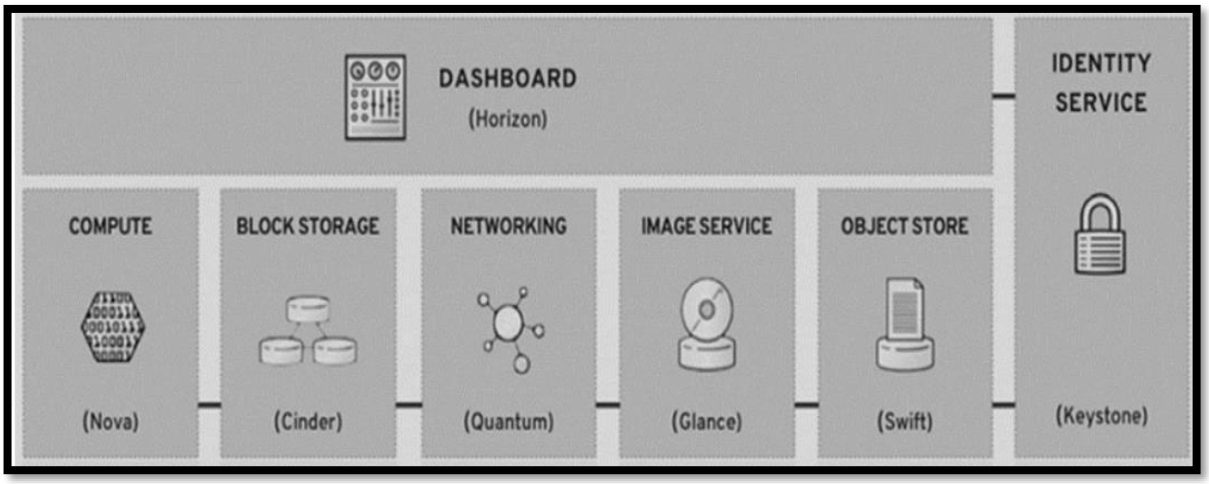
Image Service (Glance): The glance component is used to provide the image services to OpenStack. Here, image service means the images or virtual copies of hard disks. When we plan to deploy a new virtual machine instance, then glance allows us to use these images as templates. Glance allows virtual box (VDI), VMware (VMDK, OVF), Raw, Hyper-V (VHD) and KVM (qcow2) virtual images.

Telemetry (Ceilometer): It is used to meter the usage and report it to OpenStack's individual users. So basically, Telementry provides billing services to OpenStack's individual users.

Orchestration (Heat): It allows the developers to store the cloud application's necessities as a file so that all-important resources are available in handy. This component organizes many complex applications of the cloud through the templates, via both the local OpenStack REST API and Query API.

Shared File System (Manila): It offers storage of the file to a virtual machine. This component gives an infrastructure for managing and provisioning file shares.

Elastic Map-reduce (Sahara): The Sahara component offers a simple method to the users to preplanned Hadoop clusters by referring to the multiple options such as the Hadoop version, cluster topology and hardware details of nodes and some more.



[Components Of OpenStack]

4. ARCHITECTURE OF OPENSTACK:

OpenStack contains a modular architecture along with several code names for the components.

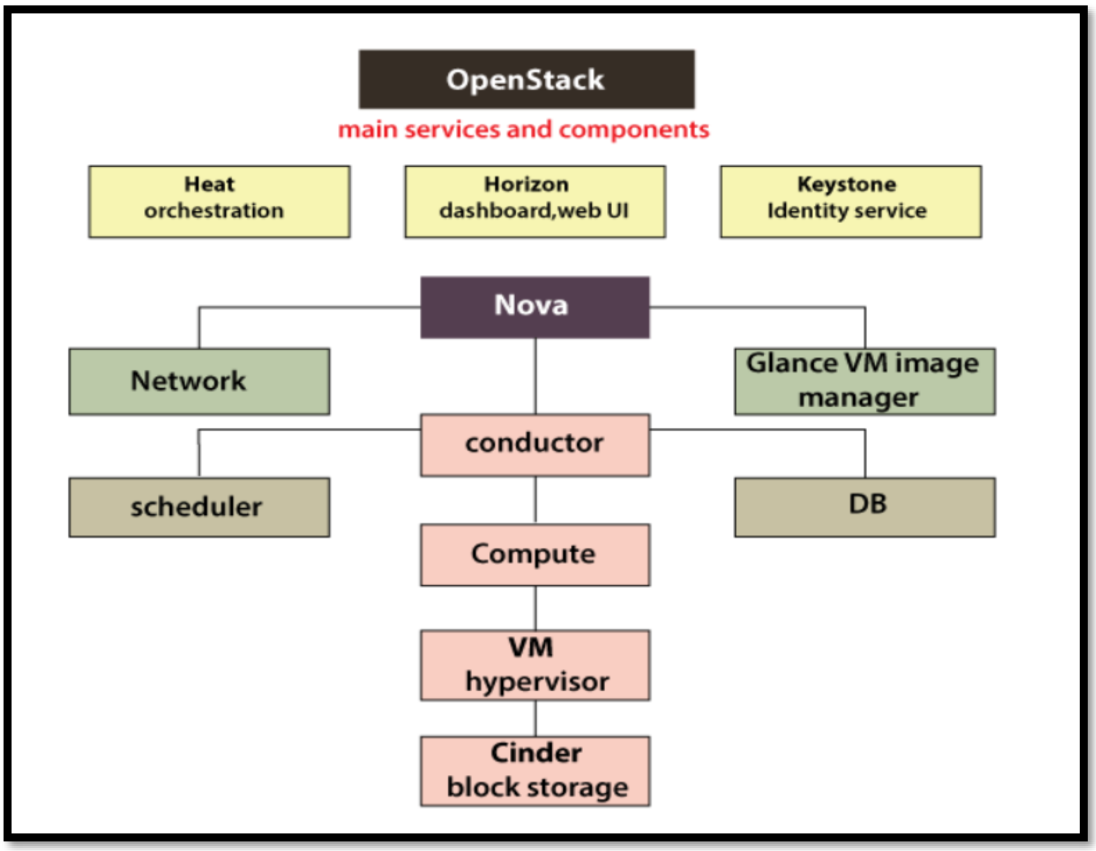
- **Nova (Compute)**

Nova is a project of OpenStack that facilitates a way for provisioning compute instances. Nova supports building bare-metal servers, virtual machines. It has narrow support for various system containers.

It executes as a daemon set on the existing Linux server's top for providing that service.
This component is specified in Python. It uses several external libraries of Python such as SQL toolkit and object-relational mapper (SQLAlchemy), AMQP messaging framework (Kombu), and concurrent networking libraries (Eventlet). Nova is created to be scalable horizontally.

We procure many servers and install configured services identically, instead of switching to any large server.

Because of its boundless integration into organization-level infrastructure, particularly Nova performance, and general performance of monitoring OpenStack, scaling facility has become a progressively important issue.



- **Neutron (Networking)**

Neutron can be defined as a project of OpenStack. It gives "network connectivity as a service" facility between various interface devices (such as vNICs) that are handled by some other types of OpenStack services (such as Nova). It operates the Networking API of OpenStack.

It handles every networking facet for VNI (Virtual Networking Infrastructure) and various authorization layer factors of PNI (Physical Networking Infrastructure) in an OpenStack platform.

OpenStack networking allows projects to build advanced topologies of the virtual network. It can include some of the services like VPN (Virtual Private Network) and a firewall.

Neutron permits dedicated static DHCP or IP addresses. It permits Floating IP addresses to enable the traffic to be rerouted.

- **Cinder (Block Storage)**

Cinder is a service of OpenStack block storage that is used to provide volumes to Nova VMs, containers, ironic bare-metal hosts, and more. A few objectives of cinder are as follows:

- **Open-standard:** It is any reference implementation for the community-driven APIs.
- **Recoverable:** Failures must be not complex to rectify, debug, and diagnose.
- **Fault-Tolerant:** Separated processes ignore cascading failures.
- **Highly available:** Can scale to serious workloads.

- **Component-based architecture:** Include new behaviours quickly.

- **Keystone (Identity)**

Keystone is a service of OpenStack that offers shared multi-tenant authorization, service discovery, and API client authentication by implementing Identity API of OpenStack. Commonly, it is an authentication system around the cloud OS.

Keystone could integrate with various directory services such as LDAP. It also supports standard password and username credentials, Amazon Web Services (AWS) style, and token-based systems logins.

The catalog of keystone service permits API clients for navigating and discovering various cloud services dynamically.

- **Metadata Definitions**

Image hosts a metadefs catalog. It facilitates an OpenStack community along with a path to determine several metadata valid values and key names that could be used for OpenStack resources.

- **Swift (Object Storage)**

Swift is an eventually consistent and distributed blob/object-store. The object store project of OpenStack is called Swift and it provides software for cloud storage so that we can retrieve and store a large amount of data along with a general API.

It is created for scale and upgraded for concurrency, availability, and durability across the whole data set. Object storage is ideal to store unstructured data that could grow without any limitations.

Rackspace, in 2009 August, started the expansion of the forerunner to the OpenStack Object Storage same as a complete substitution for the product of Cloud Files. The starting development team includes nine developers. Currently, an object storage enterprise (SwiftStack) is the prominent developer for OpenStack Swift with serious contributions from IBM, HP, NTT, Red Hat, Intel, and many more.

- **Horizon (Dashboard)**

Horizon is a canonical implementation of Dashboard of OpenStack which offers the web-based UI to various OpenStack services such as Keystone, Swift, Nova, etc. Dashboard shifts with a few central dashboards like a "Settings Dashboard", a "System Dashboard", and a "User Dashboard".

It envelopes Core Support. The horizon application ships using the API abstraction set for many projects of Core OpenStack to facilitate a stable and consistent collection of reusable techniques for developers.

With these abstractions, the developers working on OpenStack Horizon do not require to be familiar intimately with the entire OpenStack project's APIs.

5. PROS AND CONS OF OPENSTACK:

- **Pros of OpenStack:**

  Open Source: OpenStack's open-source nature allows for customization and flexibility, enabling users to modify the platform according to their specific needs without any limitations or restrictions on usage.

  Scalability: OpenStack offers seamless scalability, allowing enterprises to easily spin up or down servers on-demand, thus adapting to fluctuating workloads and business demands effectively.

  Security: With robust security features, OpenStack ensures data integrity and protection, alleviating concerns about data loss during cloud migration and providing responsive security professionals to maintain a secure environment.

  Automation: OpenStack's automation capabilities streamline cloud management tasks, making operations more efficient. Its built-in tools and APIs enable easy automation of processes like VM provisioning, enhancing overall productivity.

  Development Support: The open-source model of OpenStack encourages global collaboration and innovation, with contributions from a diverse community of developers and support from leading IT companies, resulting in rapid development and improvement of the platform.

  Easy Access and Management: OpenStack provides multiple access points, including command-line tools, a graphical user interface (GUI) dashboard, and APIs, making it easy for users and administrators to access and manage various aspects of the platform.

  Services: OpenStack offers a comprehensive suite of services, each serving different purposes within the cloud environment, such as computing, networking, storage, and dashboard interfaces, catering to diverse needs and requirements.

  Strong Community: OpenStack boasts a vibrant community of experts, developers, and users who collaborate to enhance and improve the platform continually, ensuring its relevance and longevity in the rapidly evolving cloud computing landscape.

  Compatibility: OpenStack is compatible with various public cloud systems like AWS, providing interoperability and flexibility for organizations to integrate and leverage multiple cloud environments seamlessly.

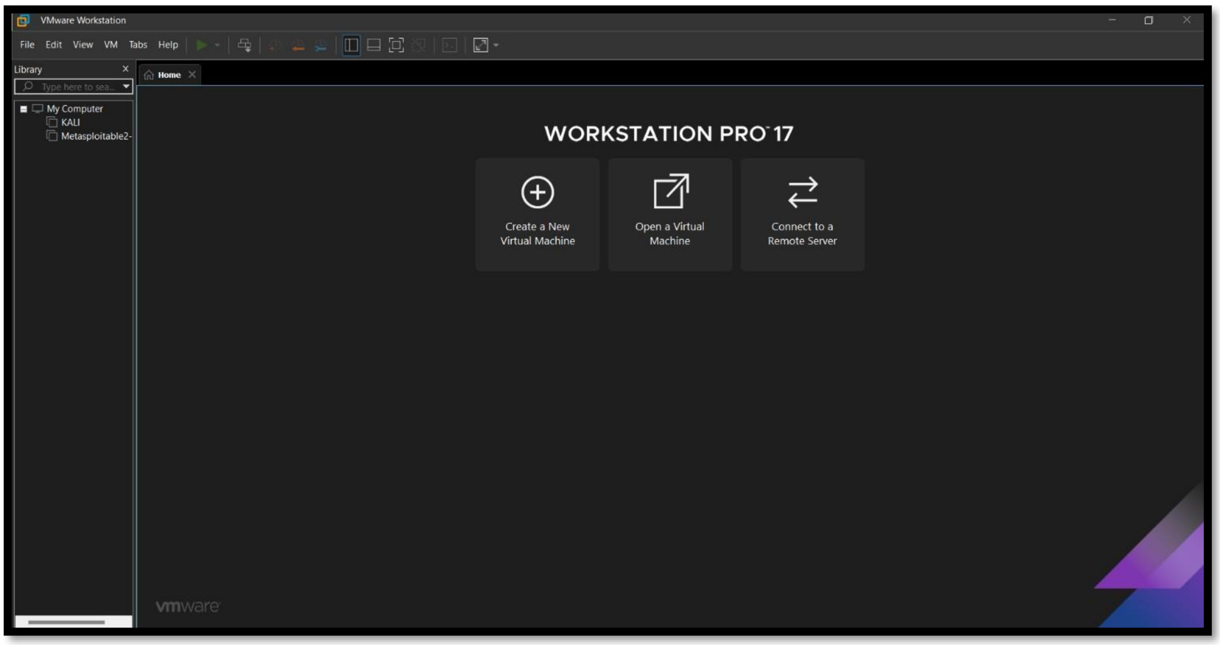
- **Cons of OpenStack:**

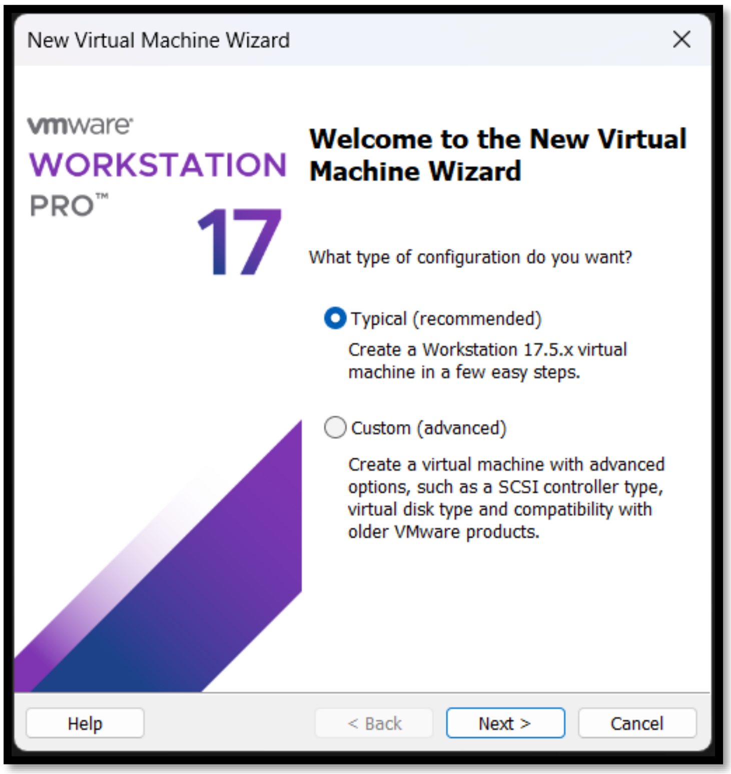
- OpenStack is not very robust when orchestration is considered.
- Even today, the APIs provided and supported by OpenStack are not compatible with many of the hybrid cloud providers, thus integrating solutions becomes difficult.
- Like all cloud service providers OpenStack services also come with the risk of security breaches.

6. STEPS TO IMPLEMENT:

- **Step 1: Install Ubuntu on VMware**

First, open VMware, then click "Creat a New Virtual Machine" to create a virtual machine.





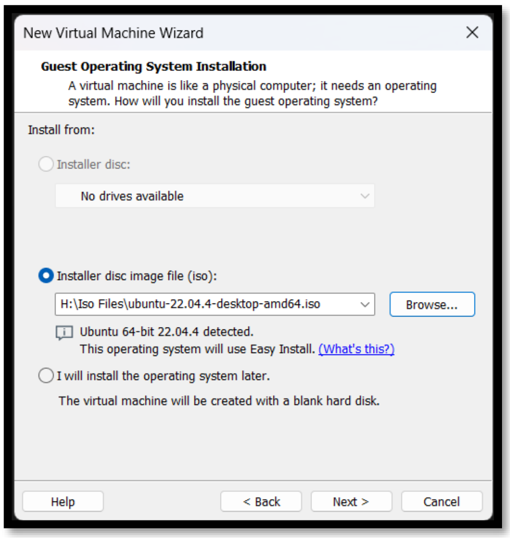
Welcome to the New Virtual Machine Wizard

What type of configuration do you want?

[✓ Typical (recommended)]
Create a Workstation 17.5.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.

Select the .iso of the "Ubuntu" Operating System which you can download on their site and click "Next".



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):]

H:\Iso Files\ubuntu-22.04.4-desktop-amd64.iso

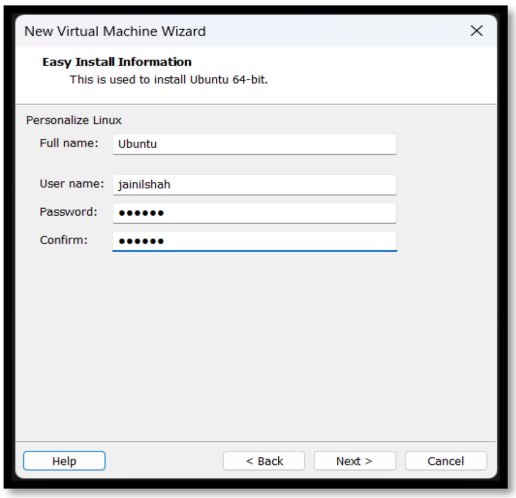
Ubuntu 64-bit 22.04.4 detected.

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.

Enter your details like Username and password, then click "Next":



New Virtual Machine Wizard

Easy Install Information

This is used to install Ubuntu 64-bit.

Personalize Linux

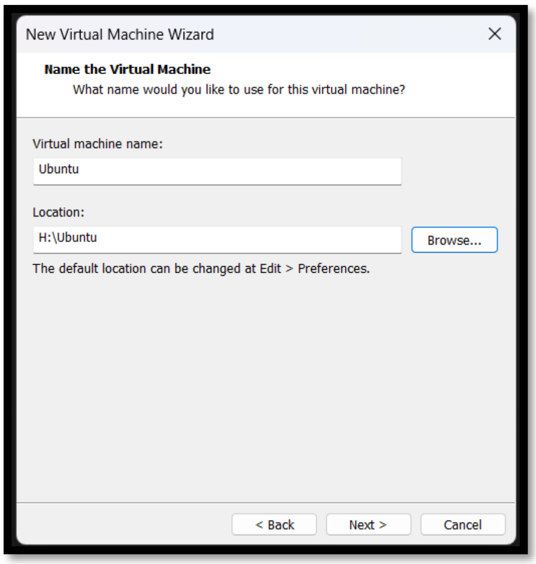
Full name: Ubuntu

User name: jainilshah

Password: ••••••

Confirm: ••••••

Name your OS and select the path where you want to install the Virtual OS:



New Virtual Machine Wizard

Name the Virtual Machine

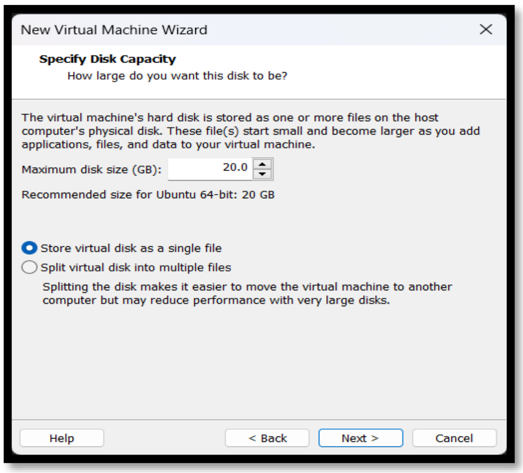
What name would you like to use for this virtual machine?

Virtual machine name: Ubuntu

Location: H:\Ubuntu

The default location can be changed at Edit > Preferences.

Specify disk capacity for the OS, default is 20GB:



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

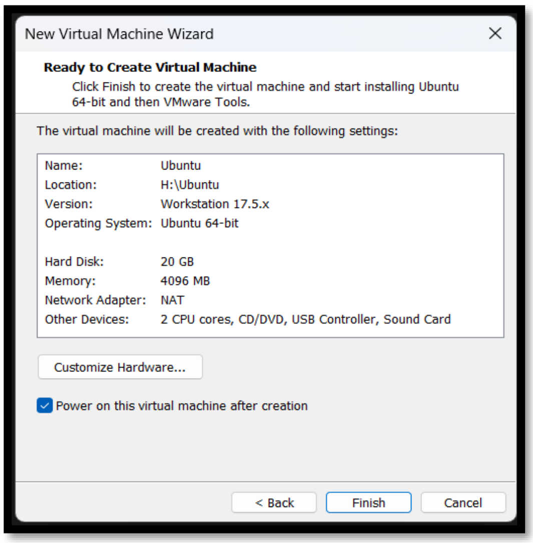
The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.

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.

After that click “Finish” to proceed the installation of Ubuntu:



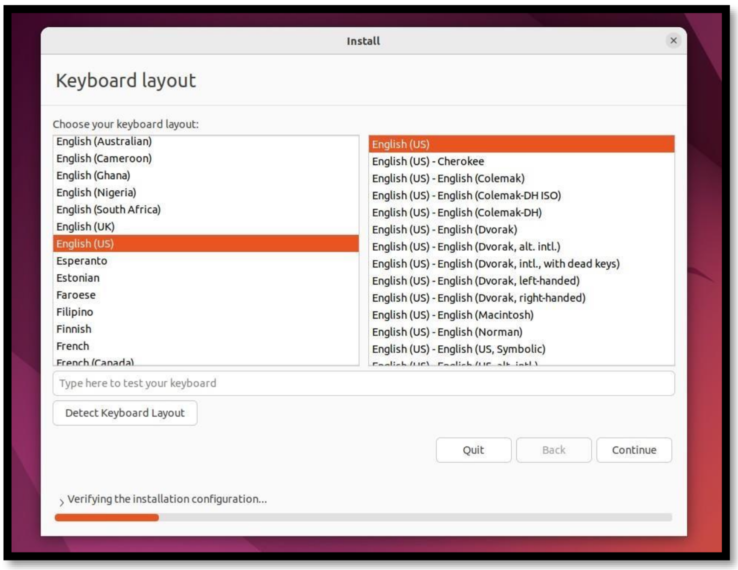
Ready to Create Virtual Machine
Click Finish to create the virtual machine and start installing Ubuntu 64-bit and then VMware Tools.

The virtual machine will be created with the following settings:

Name:	Ubuntu
Location:	H:\Ubuntu
Version:	Workstation 17.5.x
Operating System:	Ubuntu 64-bit
Hard Disk:	20 GB
Memory:	4096 MB
Network Adapter:	NAT
Other Devices:	2 CPU cores, CD/DVD, USB Controller, Sound Card

[✓ Power on this virtual machine after creation]

Select your preferred language and click “Continue”:



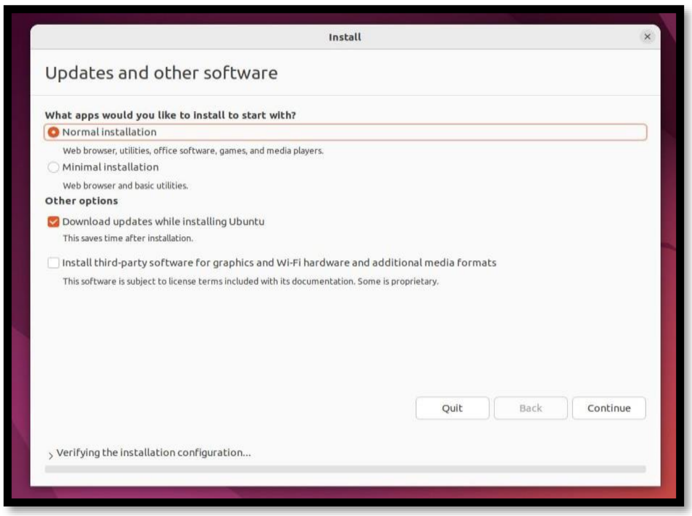
Choose your keyboard layout:

English (Australian)	[✓ English (US)]
English (Cameroon)	English (US) - Cherokee
English (Ghana)	English (US) - English (Colemak)
English (Nigeria)	English (US) - English (Colemak-DH ISO)
English (South Africa)	English (US) - English (Colemak-DH)
English (UK)	English (US) - English (Dvorak)
[✓ English (US)]	English (US) - English (Dvorak, alt. intl.)
Esperanto	English (US) - English (Dvorak, intl., with dead keys)
Estonian	English (US) - English (Dvorak, left-handed)
Faroese	English (US) - English (Dvorak, right-handed)
Filipino	English (US) - English (Macintosh)
Finnish	English (US) - English (Norman)
French	English (US) - English (US, Symbolic)
French (Canada)	

Type here to test your keyboard

Verifying the installation configuration...

In the "Updates and other software" section, check "Normal installation" and continue.


Install

Updates and other software

What apps would you like to install to start with?

[✓ Normal installation]
Web browser, utilities, office software, games, and media players.

Minimal installation
Web browser and basic utilities.

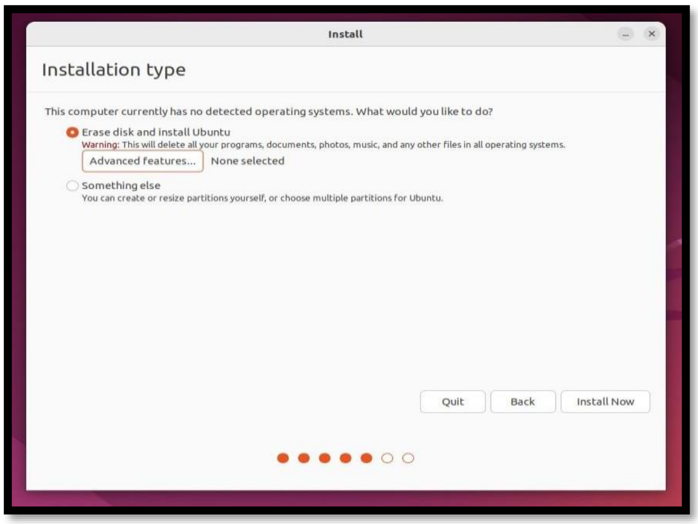
Other options

[✓ Download updates while installing Ubuntu]
This saves time after installation.

Install third-party software for graphics and Wi-Fi hardware and additional media formats
This software is subject to license terms included with its documentation. Some is proprietary.

Verifying the installation configuration...

In "Installation type", check "Erase disk and install Ubuntu".


Install

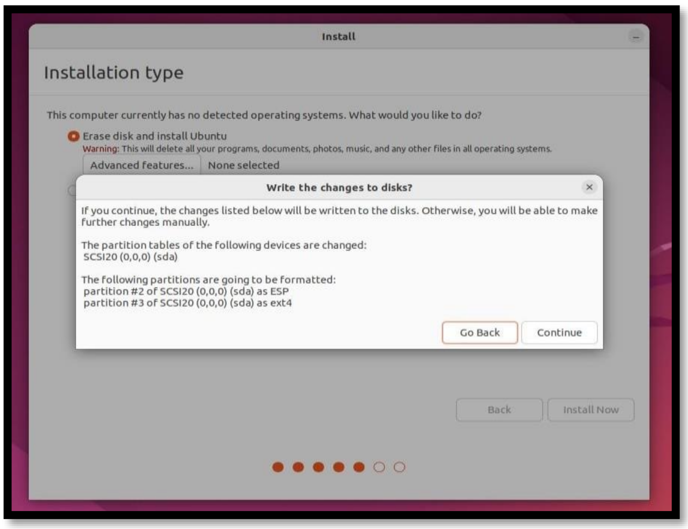
Installation type

This computer currently has no detected operating systems. What would you like to do?

[✓ Erase disk and install Ubuntu]
Warning: This will delete all your programs, documents, photos, music, and any other files in all operating systems.
Advanced features... None selected

Something else
You can create or resize partitions yourself, or choose multiple partitions for Ubuntu.

Click "Continue".


Install

Installation type

This computer currently has no detected operating systems. What would you like to do?

[✓ Erase disk and install Ubuntu]
Warning: This will delete all your programs, documents, photos, music, and any other files in all operating systems.
Advanced features... None selected

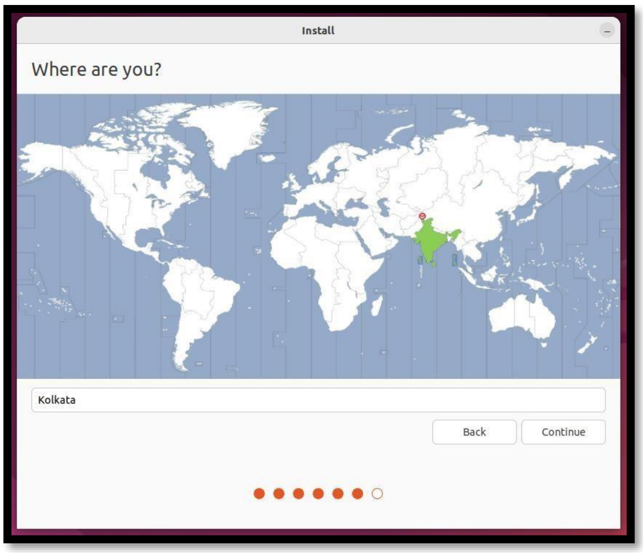
Write the changes to disks?

If you continue, the changes listed below will be written to the disks. Otherwise, you will be able to make further changes manually.

The partition tables of the following devices are changed:
SCSI20 (0,0,0) (sda)

The following partitions are going to be formatted:
partition #2 of SCSI20 (0,0,0) (sda) as ESP
partition #3 of SCSI20 (0,0,0) (sda) as ext4

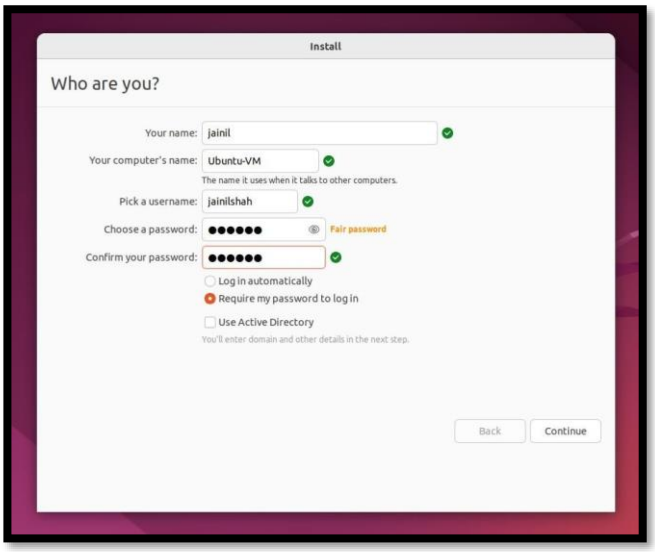
Choose your current location.



Where are you?

Kolkata

Now, set up your profile.



Who are you?

Your name: jainil

Your computer's name: Ubuntu-VM

The name it uses when it talks to other computers.

Pick a username: jainilshah

Choose a password: ●●●●●● Fair password

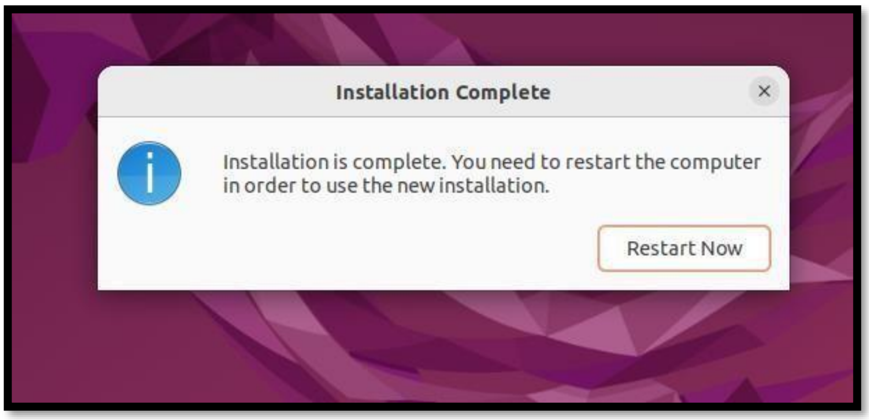
Confirm your password: ●●●●●●

Log in automatically [✓ Require my password to log in]

Use Active Directory

You'll enter domain and other details in the next step.

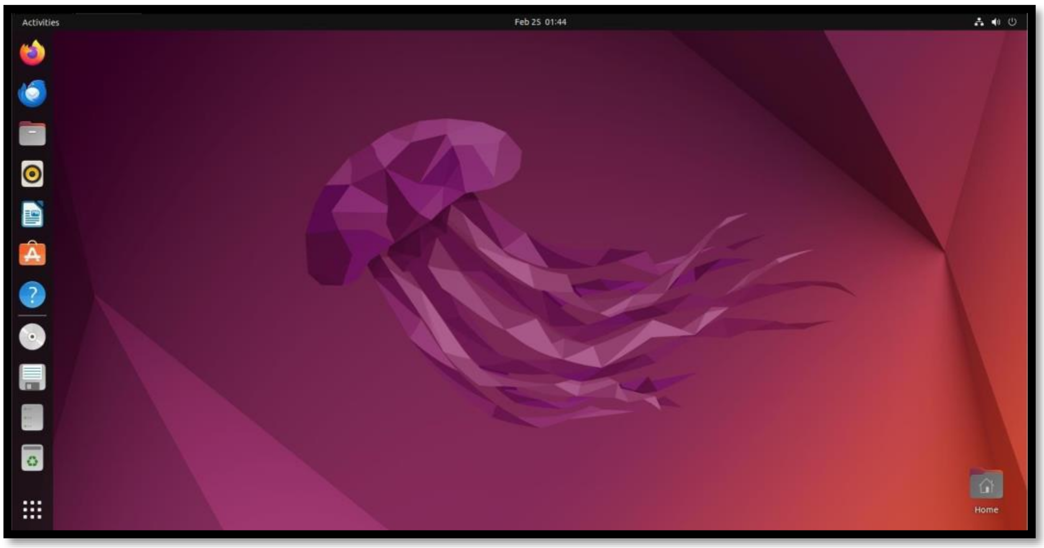
You'll see Ubuntu getting installed. After the installation, restart it.



Installation Complete

Installation is complete. You need to restart the computer in order to use the new installation.

After logging in, you'll see the Ubuntu desktop.

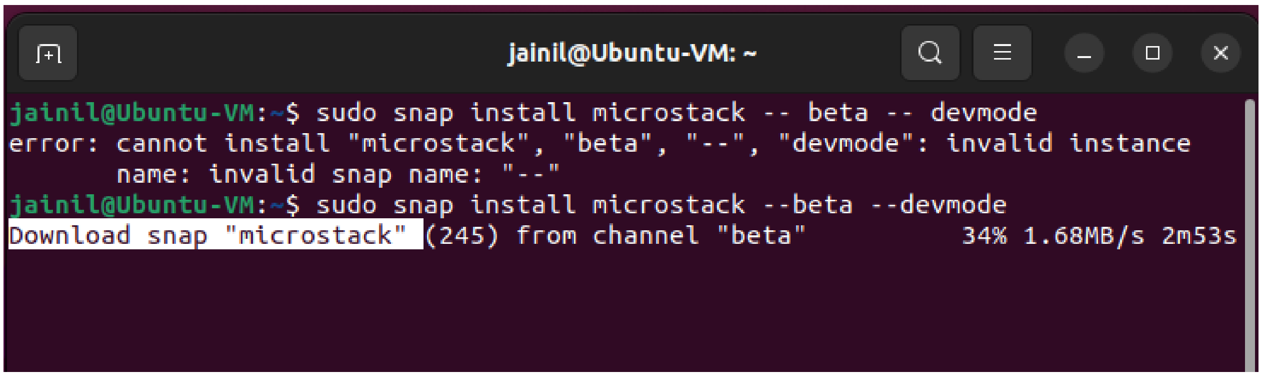


We have successfully installed Ubuntu in VMware. It's ready to use for your future development projects.

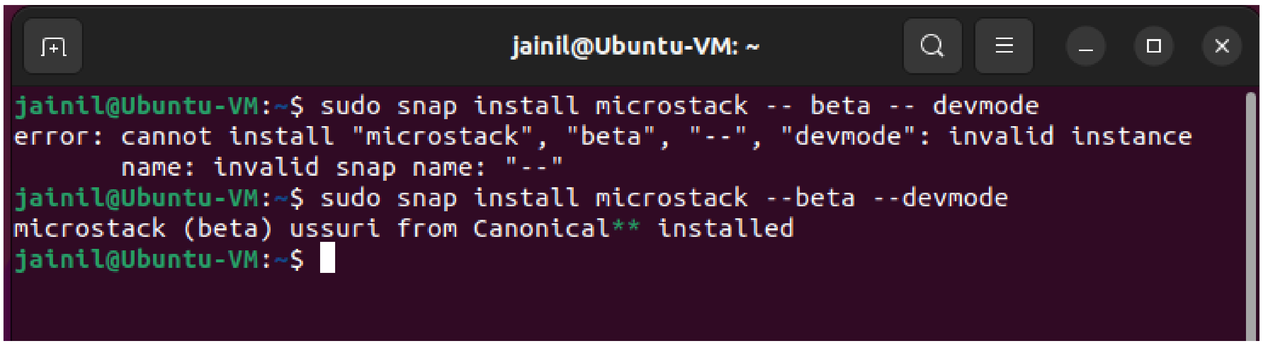
Step 2:

Step 1: Install OpenStack(microstack) on Ubuntu :

Syntax: sudo snap install microstack --beta –devmode



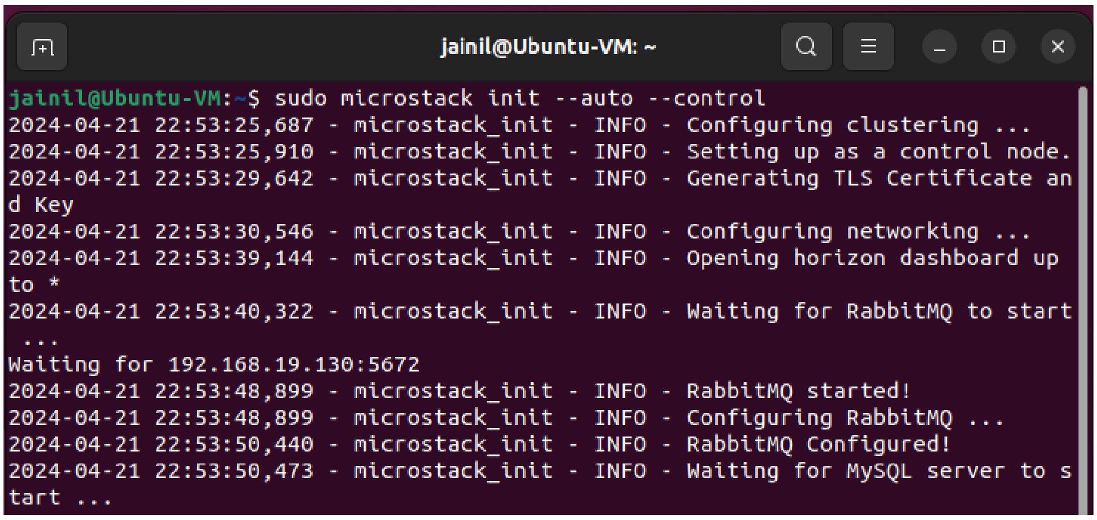
```
jainil@Ubuntu-VM:~$ sudo snap install microstack -- beta -- devmode
error: cannot install "microstack", "beta", "--", "devmode": invalid instance
       name: invalid snap name: "--"
jainil@Ubuntu-VM:~$ sudo snap install microstack --beta --devmode
Download snap "microstack" (245) from channel "beta"          34% 1.68MB/s 2m53s
```



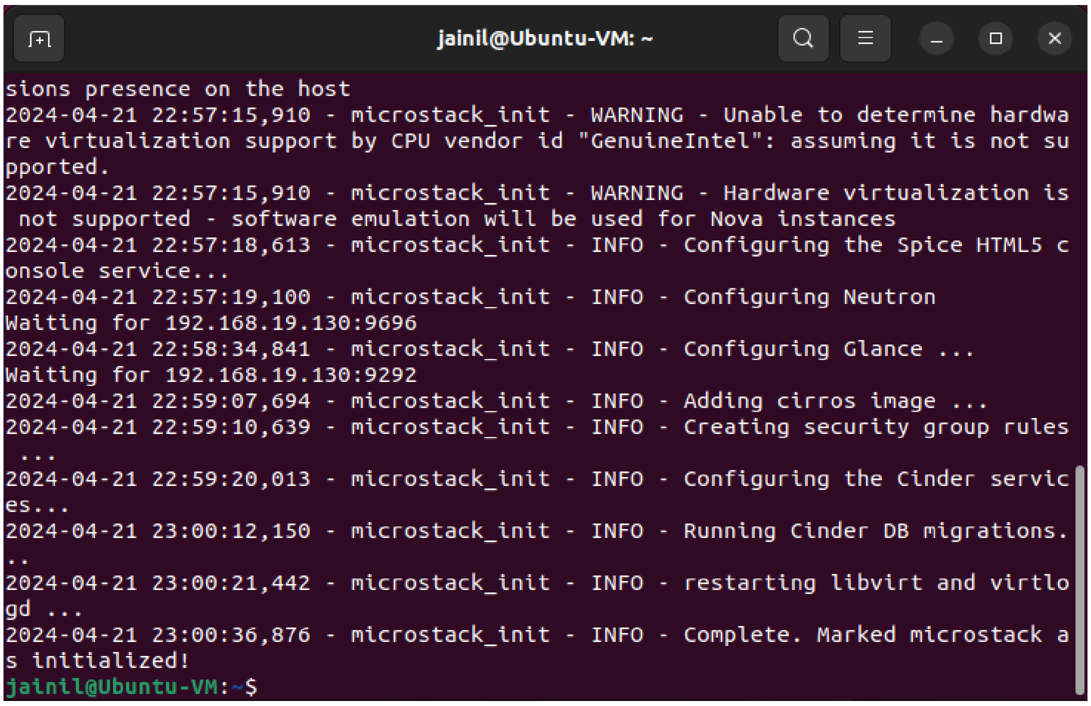
```
jainil@Ubuntu-VM:~$ sudo snap install microstack -- beta -- devmode
error: cannot install "microstack", "beta", "--", "devmode": invalid instance
       name: invalid snap name: "--"
jainil@Ubuntu-VM:~$ sudo snap install microstack --beta --devmode
microstack (beta) ussuri from Canonical** installed
jainil@Ubuntu-VM:~$
```

Step 2: Initialize the MicroStack OpenStack cloud on a system :

Syntax: sudo microstack init --auto --control



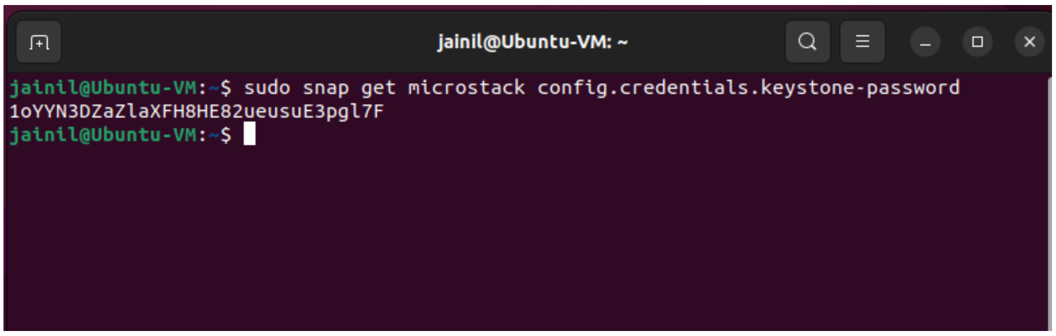
```
jainil@Ubuntu-VM:~$ sudo microstack init --auto --control
2024-04-21 22:53:25,687 - microstack_init - INFO - Configuring clustering ...
2024-04-21 22:53:25,910 - microstack_init - INFO - Setting up as a control node.
2024-04-21 22:53:29,642 - microstack_init - INFO - Generating TLS Certificate an
d Key
2024-04-21 22:53:30,546 - microstack_init - INFO - Configuring networking ...
2024-04-21 22:53:39,144 - microstack_init - INFO - Opening horizon dashboard up
to *
2024-04-21 22:53:40,322 - microstack_init - INFO - Waiting for RabbitMQ to start
 ...
Waiting for 192.168.19.130:5672
2024-04-21 22:53:48,899 - microstack_init - INFO - RabbitMQ started!
2024-04-21 22:53:48,899 - microstack_init - INFO - Configuring RabbitMQ ...
2024-04-21 22:53:50,440 - microstack_init - INFO - RabbitMQ Configured!
2024-04-21 22:53:50,473 - microstack_init - INFO - Waiting for MySQL server to s
tart ...
```



```
sions presence on the host
2024-04-21 22:57:15,910 - microstack_init - WARNING - Unable to determine hardwa
re virtualization support by CPU vendor id "GenuineIntel": assuming it is not su
pported.
2024-04-21 22:57:15,910 - microstack_init - WARNING - Hardware virtualization is
 not supported - software emulation will be used for Nova instances
2024-04-21 22:57:18,613 - microstack_init - INFO - Configuring the Spice HTML5 c
onsole service...
2024-04-21 22:57:19,100 - microstack_init - INFO - Configuring Neutron
Waiting for 192.168.19.130:9696
2024-04-21 22:58:34,841 - microstack_init - INFO - Configuring Glance ...
Waiting for 192.168.19.130:9292
2024-04-21 22:59:07,694 - microstack_init - INFO - Adding cirros image ...
2024-04-21 22:59:10,639 - microstack_init - INFO - Creating security group rules
 ...
2024-04-21 22:59:20,013 - microstack_init - INFO - Configuring the Cinder servic
es...
2024-04-21 23:00:12,150 - microstack_init - INFO - Running Cinder DB migrations.
..
2024-04-21 23:00:21,442 - microstack_init - INFO - restarting libvirt and virtlo
gd ...
2024-04-21 23:00:36,876 - microstack_init - INFO - Complete. Marked microstack a
s initialized!
jainil@Ubuntu-VM:~$
```

Step 3: Retrieve the Keystone password from the MicroStack configuration:

Syntax: sudo snap get microstack config.credentials.keystone-password



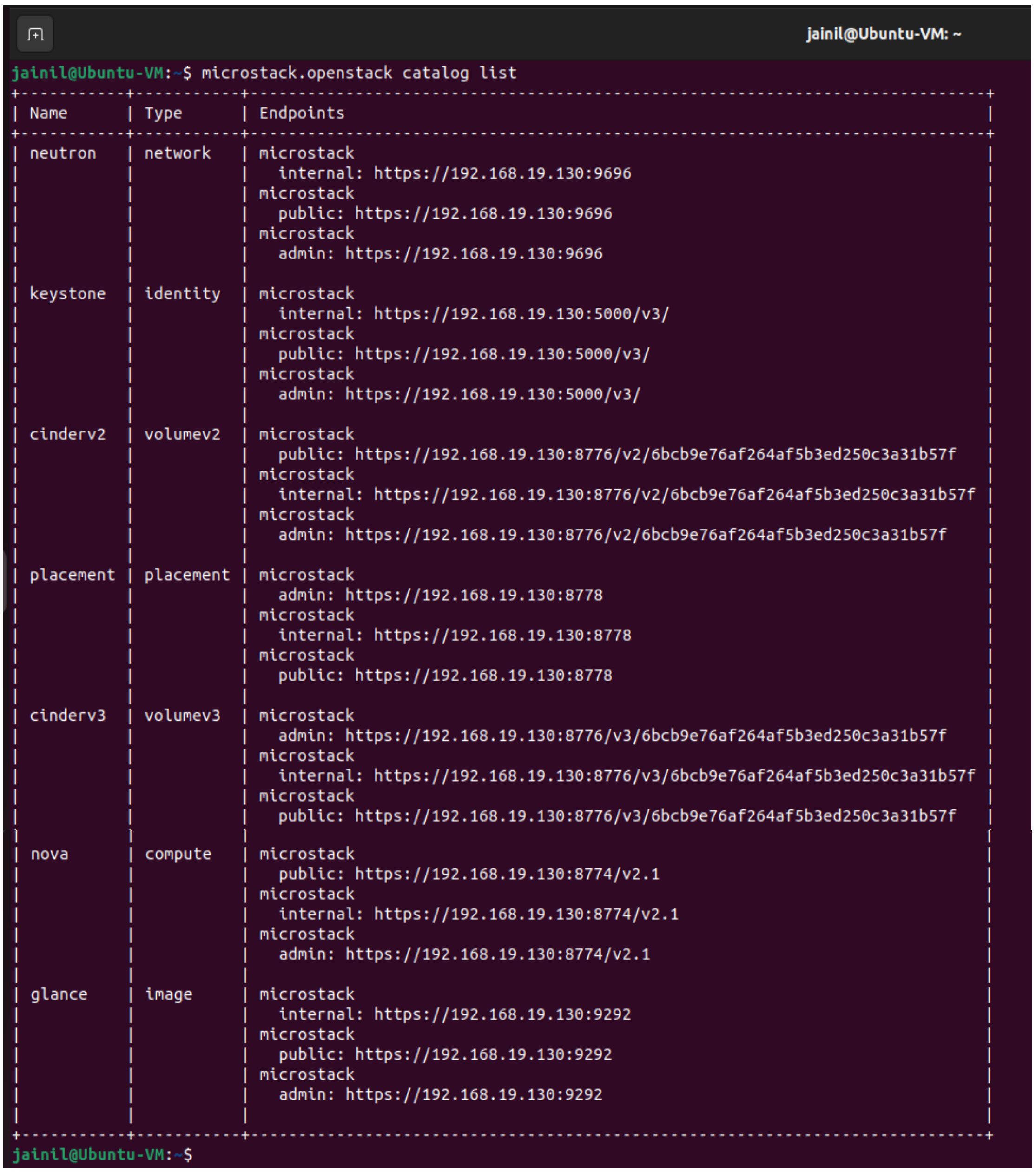
```
jainil@Ubuntu-VM:~$ sudo snap get microstack config.credentials.keystone-password
1oYYN3DZaZlaXFH8HE82ueusuE3pgl7F
jainil@Ubuntu-VM:~$
```

User: admin

Password: 1oYYN3DZaZlaXFH8HE82ueusuE3pgl7F

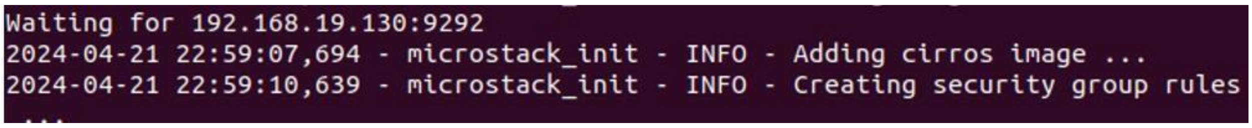
Step 4: List the available services in the OpenStack service catalog provided by MicroStack:

Syntax: microstack.openstack catalog list

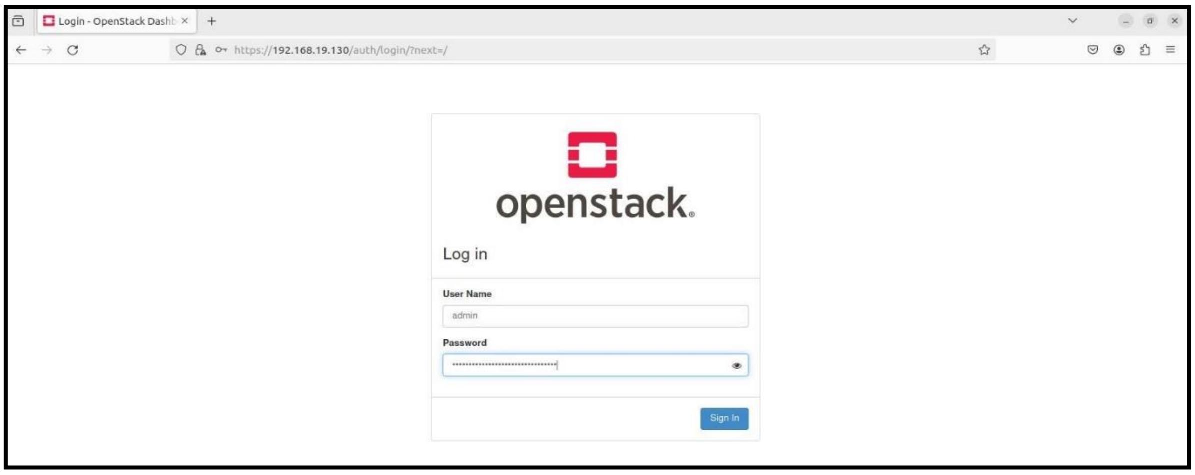


```
jainil@Ubuntu-VM: ~
jainil@Ubuntu-VM:~$ microstack.openstack catalog list
+-----------+-----------+--------------------------------------------------------------------------------+
| Name      | Type      | Endpoints                                                                      |
+-----------+-----------+--------------------------------------------------------------------------------+
| neutron   | network   | microstack                                                                     |
|           |           |   internal: https://192.168.19.130:9696                                        |
|           |           | microstack                                                                     |
|           |           |   public: https://192.168.19.130:9696                                          |
|           |           | microstack                                                                     |
|           |           |   admin: https://192.168.19.130:9696                                           |
|           |           |                                                                                |
| keystone  | identity  | microstack                                                                     |
|           |           |   internal: https://192.168.19.130:5000/v3/                                    |
|           |           | microstack                                                                     |
|           |           |   public: https://192.168.19.130:5000/v3/                                      |
|           |           | microstack                                                                     |
|           |           |   admin: https://192.168.19.130:5000/v3/                                       |
|           |           |                                                                                |
| cinderv2  | volumev2  | microstack                                                                     |
|           |           |   public: https://192.168.19.130:8776/v2/6bcb9e76af264af5b3ed250c3a31b57f      |
|           |           | microstack                                                                     |
|           |           |   internal: https://192.168.19.130:8776/v2/6bcb9e76af264af5b3ed250c3a31b57f    |
|           |           | microstack                                                                     |
|           |           |   admin: https://192.168.19.130:8776/v2/6bcb9e76af264af5b3ed250c3a31b57f       |
|           |           |                                                                                |
| placement | placement | microstack                                                                     |
|           |           |   admin: https://192.168.19.130:8778                                           |
|           |           | microstack                                                                     |
|           |           |   internal: https://192.168.19.130:8778                                        |
|           |           | microstack                                                                     |
|           |           |   public: https://192.168.19.130:8778                                          |
|           |           |                                                                                |
| cinderv3  | volumev3  | microstack                                                                     |
|           |           |   admin: https://192.168.19.130:8776/v3/6bcb9e76af264af5b3ed250c3a31b57f       |
|           |           | microstack                                                                     |
|           |           |   internal: https://192.168.19.130:8776/v3/6bcb9e76af264af5b3ed250c3a31b57f    |
|           |           | microstack                                                                     |
|           |           |   public: https://192.168.19.130:8776/v3/6bcb9e76af264af5b3ed250c3a31b57f      |
|           |           |                                                                                |
| nova      | compute   | microstack                                                                     |
|           |           |   public: https://192.168.19.130:8774/v2.1                                     |
|           |           | microstack                                                                     |
|           |           |   internal: https://192.168.19.130:8774/v2.1                                   |
|           |           | microstack                                                                     |
|           |           |   admin: https://192.168.19.130:8774/v2.1                                      |
|           |           |                                                                                |
| glance    | image     | microstack                                                                     |
|           |           |   internal: https://192.168.19.130:9292                                        |
|           |           | microstack                                                                     |
|           |           |   public: https://192.168.19.130:9292                                          |
|           |           | microstack                                                                     |
|           |           |   admin: https://192.168.19.130:9292                                           |
|           |           |                                                                                |
+-----------+-----------+--------------------------------------------------------------------------------+
jainil@Ubuntu-VM:~$
```

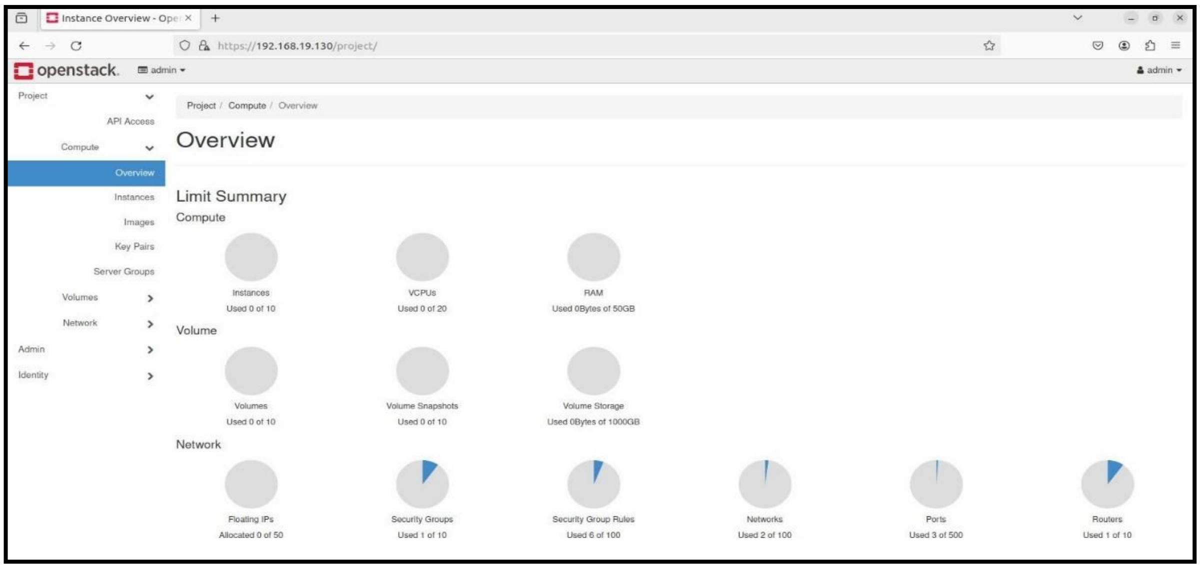
Step 5: Now open Firefox in Ubuntu and type the IP address given earlier :



```
Waiting for 192.168.19.130:9292
2024-04-21 22:59:07,694 - microstack_init - INFO - Adding cirros image ...
2024-04-21 22:59:10,639 - microstack_init - INFO - Creating security group rules
...
```



OpenStack Dashboard:



Overview

Limit Summary

Compute

Instances Used 0 of 10
VCPUs Used 0 of 20
RAM Used 0Bytes of 50GB

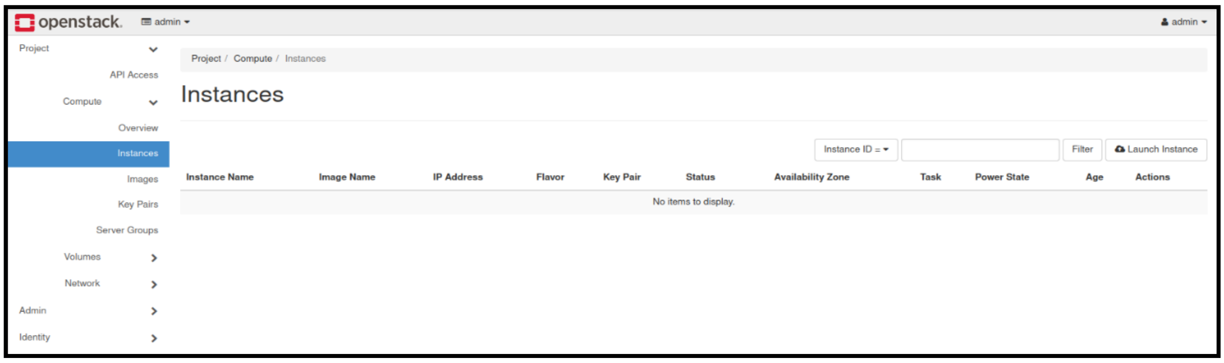
Volume

Volumes Used 0 of 10
Volume Snapshots Used 0 of 10
Volume Storage Used 0Bytes of 1000GB

Network

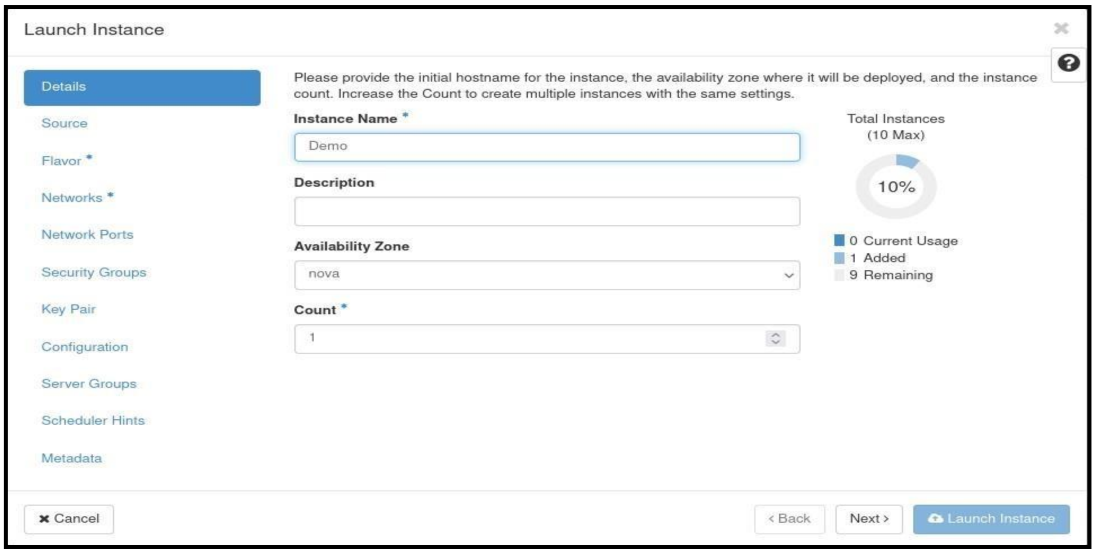
Floating IPs Allocated 0 of 50
Security Groups Used 1 of 10
Security Group Rules Used 6 of 100
Networks Used 2 of 100
Ports Used 3 of 500
Routers Used 1 of 10

Checking for instances:



Instances

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
No items to display.										



Launch Instance

Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

Instance Name * Demo

Description

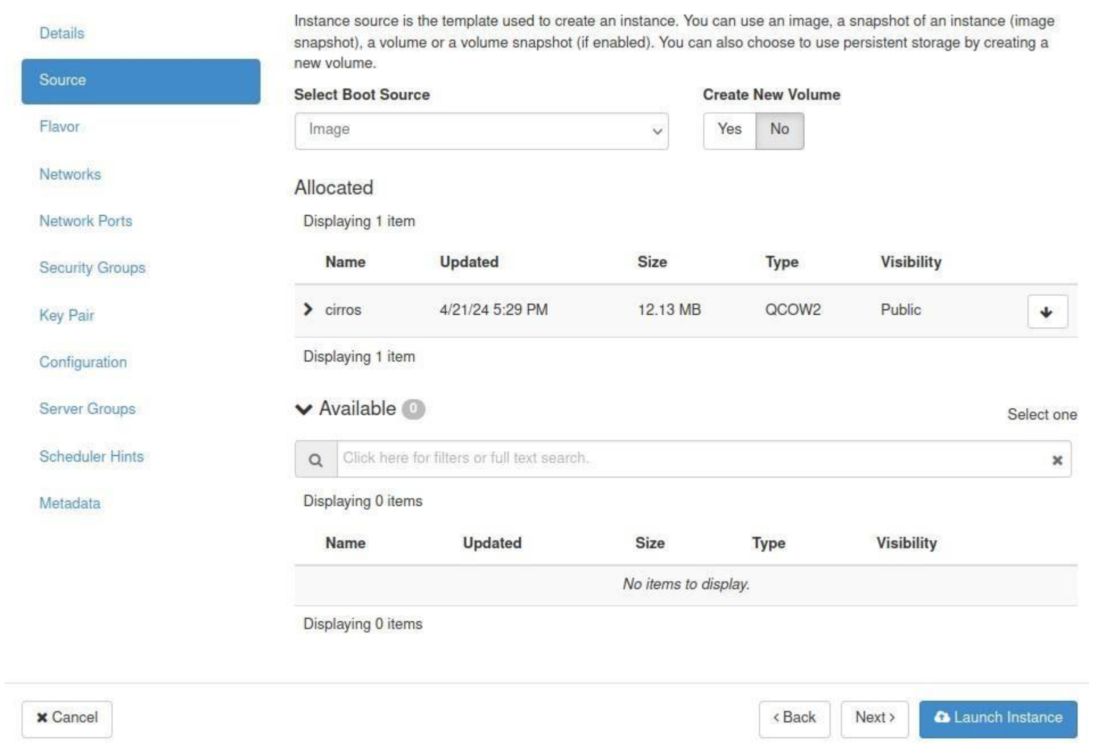
Availability Zone nova

Count * 1

Total Instances (10 Max)

10%

- 0 Current Usage
- 1 Added
- 9 Remaining



Instance source is the template used to create an instance. You can use an image, a snapshot of an instance (image snapshot), a volume or a volume snapshot (if enabled). You can also choose to use persistent storage by creating a new volume.

Select Boot Source: Image

Create New Volume: Yes [✓ No]

Allocated

Displaying 1 item

Name	Updated	Size	Type	Visibility
cirros	4/21/24 5:29 PM	12.13 MB	QCOW2	Public

Displaying 1 item

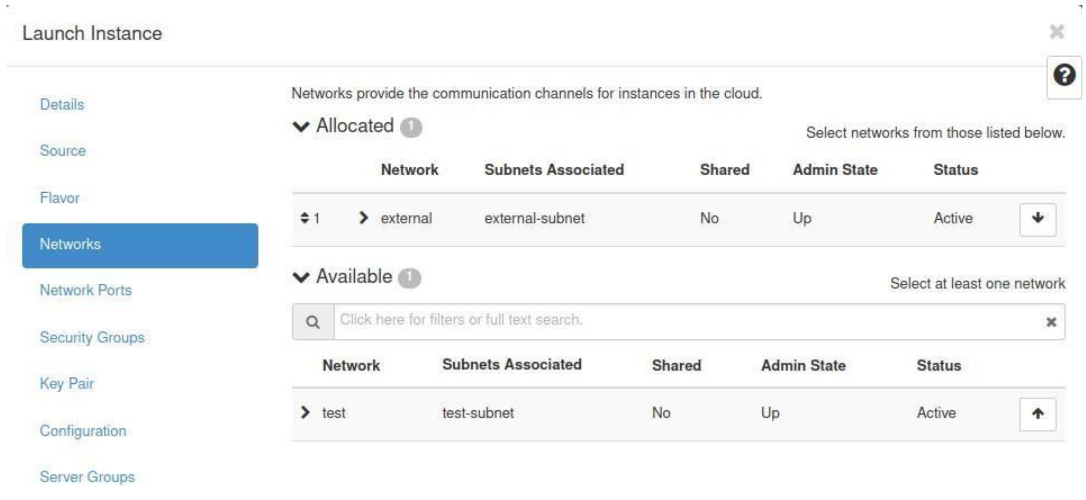
Available 0 — Select one

Click here for filters or full text search.

Displaying 0 items

Name	Updated	Size	Type	Visibility
No items to display.				

Displaying 0 items



Launch Instance

Networks provide the communication channels for instances in the cloud.

Allocated 1 — Select networks from those listed below.

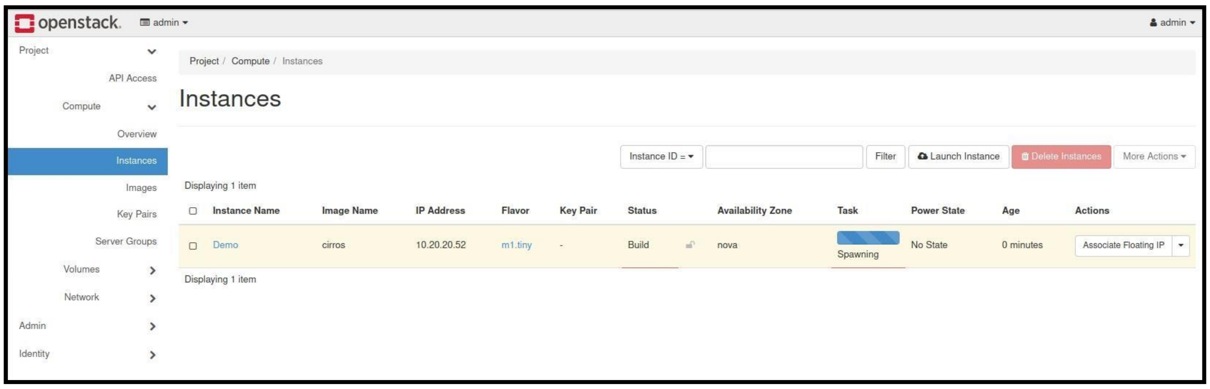
	Network	Subnets Associated	Shared	Admin State	Status
1	external	external-subnet	No	Up	Active

Available 1 — Select at least one network

Click here for filters or full text search.

Network	Subnets Associated	Shared	Admin State	Status
test	test-subnet	No	Up	Active

Instances is Spawning:


openstack. admin ▾ admin ▾

Project / Compute / Instances

Project ⌄ · API Access · Compute ⌄ · Overview · Instances · Images · Key Pairs · Server Groups · Volumes › · Network › · Admin › · Identity ›

Instances

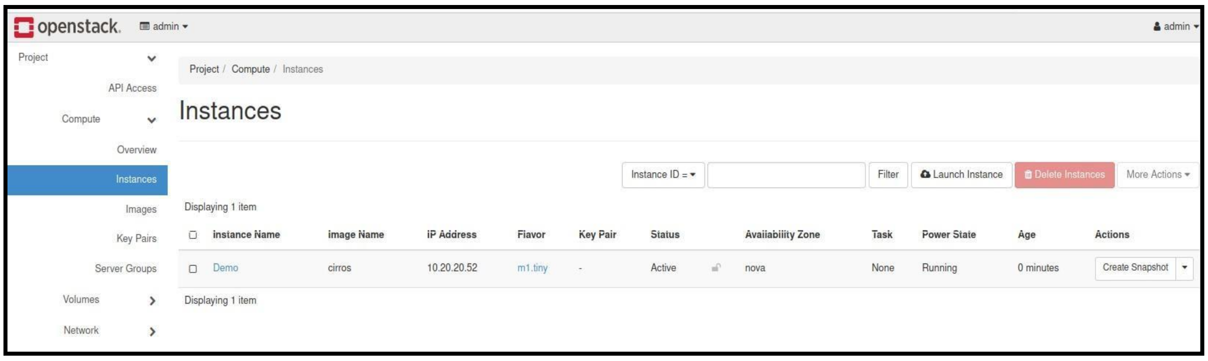
Instance ID = ▾ Filter | Launch Instance | Delete Instances | More Actions ▾

Displaying 1 item

	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
☐	Demo	cirros	10.20.20.52	m1.tiny	-	Build	nova	Spawning	No State	0 minutes	Associate Floating IP ▾

Displaying 1 item

Instances is ready:


openstack. admin ▾ admin ▾

Project / Compute / Instances

Project ⌄ · API Access · Compute ⌄ · Overview · Instances · Images · Key Pairs · Server Groups · Volumes › · Network ›

Instances

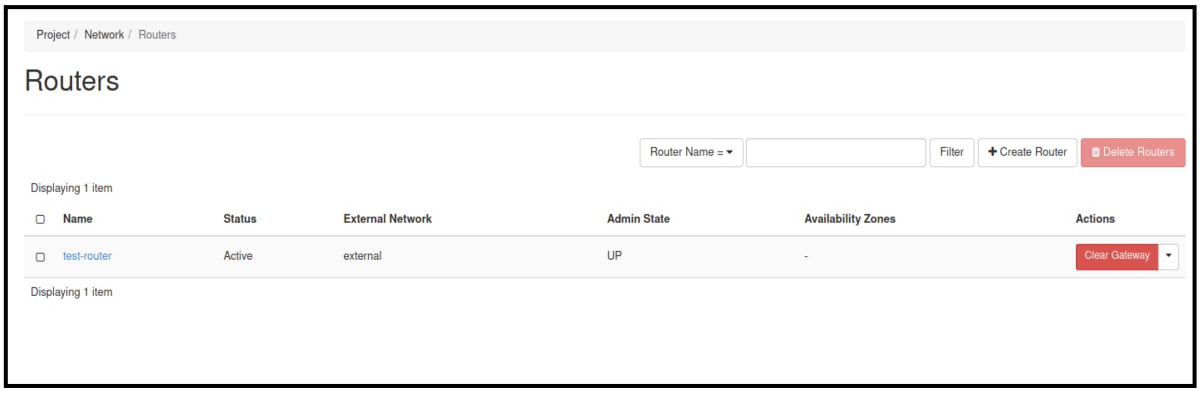
Instance ID = ▾ Filter | Launch Instance | Delete Instances | More Actions ▾

Displaying 1 item

	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
☐	Demo	cirros	10.20.20.52	m1.tiny	-	Active	nova	None	Running	0 minutes	Create Snapshot ▾

Displaying 1 item

Set Routers Gateway:


Project / Network / Routers

Routers

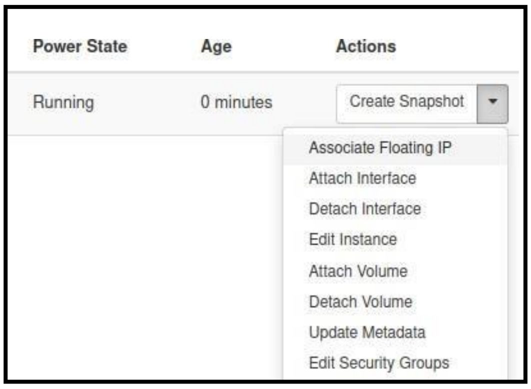
Router Name = ▾ Filter | + Create Router | Delete Routers

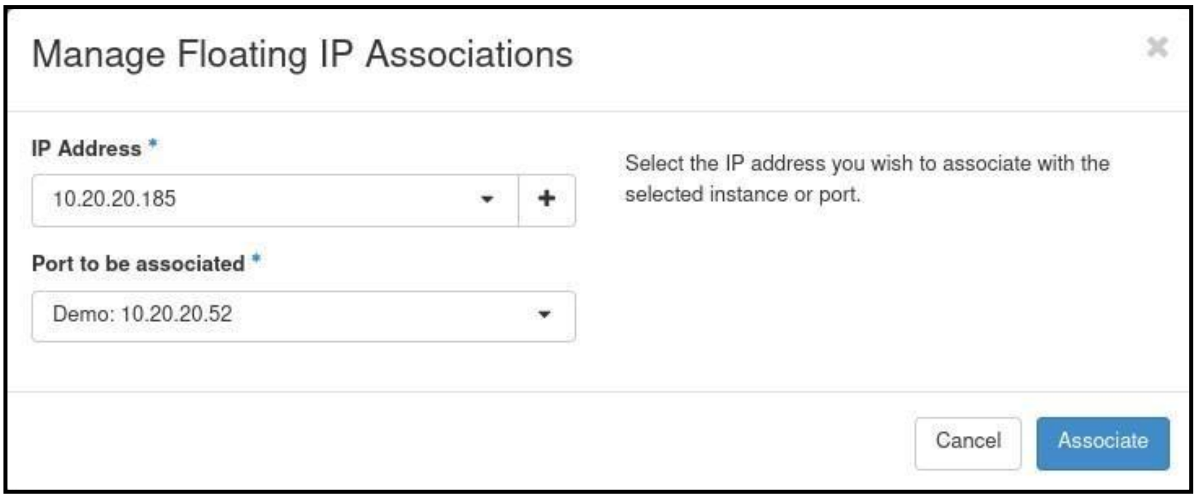
Displaying 1 item

	Name	Status	External Network	Admin State	Availability Zones	Actions
☐	test-router	Active	external	UP	-	Clear Gateway ▾

Displaying 1 item

Add the floating IP address :


Power State	Age	Actions
Running	0 minutes	Create Snapshot ▾


Manage Floating IP Associations

IP Address *

10.20.20.185

Port to be associated *

Demo: 10.20.20.52

Select the IP address you wish to associate with the selected instance or port.

7. COMPARSION WITH RELEVENT TECHNOLOGY AND SOFTWARE:

S.No.	OpenStack	AWS
1.	OpenStack is categorized as Cloud Management Platforms and Infrastructure as a Service (IaaS).	AWS Lambda is categorized as a Cloud Platform as a Service (PaaS).
2.	Glance handles the images.	AMI (Amazon Machine Image) handles the images.
3.	LBaaS of OpenStack handles the load balance traffic.	The ELB (Elastic Load Balancer) automatically distributes the incoming traffic from the services to the EC2 instances.
4.	Each virtual instance will automatically be allocated an IP address. It is handled by DHCP.	AWS allocates a private IP address to every new instance using DHCP.
5.	Identity authentication services are handled by Keystone.	Identity authentication services are handled by IAM Identity and Access management.
6.	Swift handles object storage.	Object storage is managed by S3 (simple storage service) bucket
7.	A cinder component manages block storage.	Block storage is managed by EBS (Elastic Block Storage)
8.	OpenStack provides MYSQL and PostgreSQL for the relational databases.	Users of AWS use an instance of MySQL or Oracle 11g.
9.	OpenStack uses MongoDB, Cassandra, or Couchbase for a non-relational database.	For a non-relational database, AWS uses EMR (Elastic Map Reduce).
10.	For networking, OpenStack uses Neutron.	For networking, AWS uses VPC (Virtual Private Cloud).

8. Conclusion:

OpenStack is one of the best environments in organizations for cloud computing. OpenStack's ease of linear scalability and open-source architecture have attracted many clients and enthusiasts of technology to come forward and contribute to development. This has only made OpenStack stronger over the years. OpenStack can be called for cloud computing, with all the benefits and endless modular functionality, as it proves to be an affordable option for the longer term.