**[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](https://www.javatpoint.com/restful-web-services) 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](https://www.javatpoint.com/linux-tutorial) distribution via entig100s.

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

1. **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.

## **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).

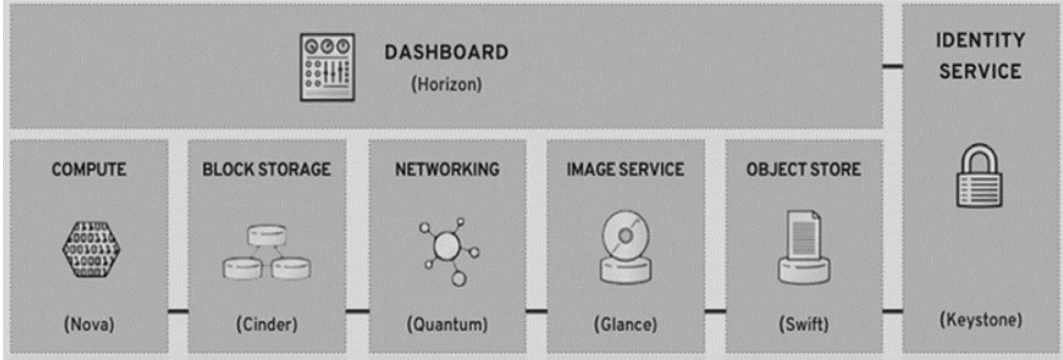
**Image Service (Glance):** The glance component is used to provide the image services to OpenStack. Here, image service means the images or virtual copies of hard disks. When we plan to deploy a new virtual machine instance, then glance allows us to use these images as templates. Glance allows virtual 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]**

1. **ARCHITECTURE OF OPENSTACK:**

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

### Nova (Compute)

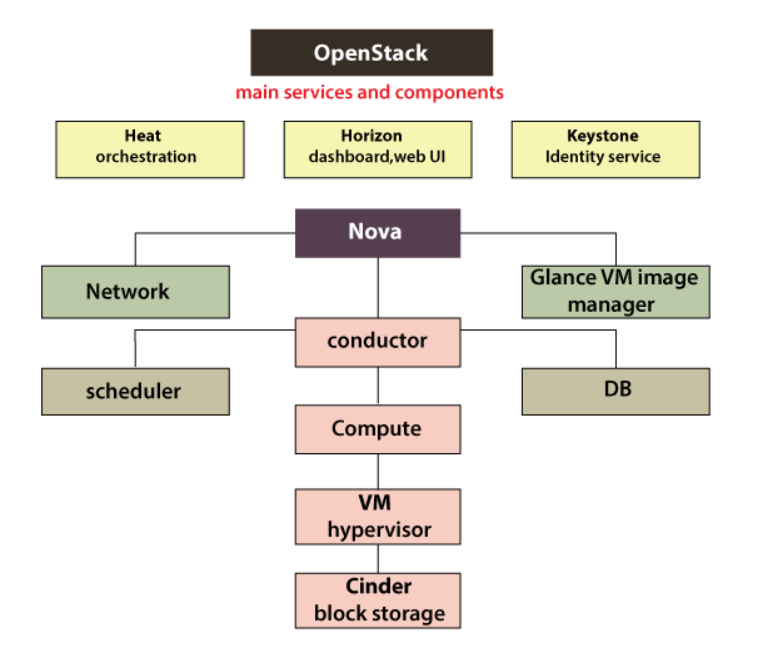
Nova is a project of OpenStack that facilitates a way for provisioning compute instances. Nova supports building bare-metal servers, virtual machines. It has narrow support for various system containers.

It executes as a daemon set on the existing [Linux](https://www.javatpoint.com/linux-tutorial) server's top for providing that service.

This component is specified in Python. It uses several external libraries of Python such as [SQL](https://www.javatpoint.com/sql-tutorial) 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)](https://www.javatpoint.com/vpn-full-form) and a firewall.

Neutron permits dedicated static DHCP or [IP](https://www.javatpoint.com/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.

1. **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.

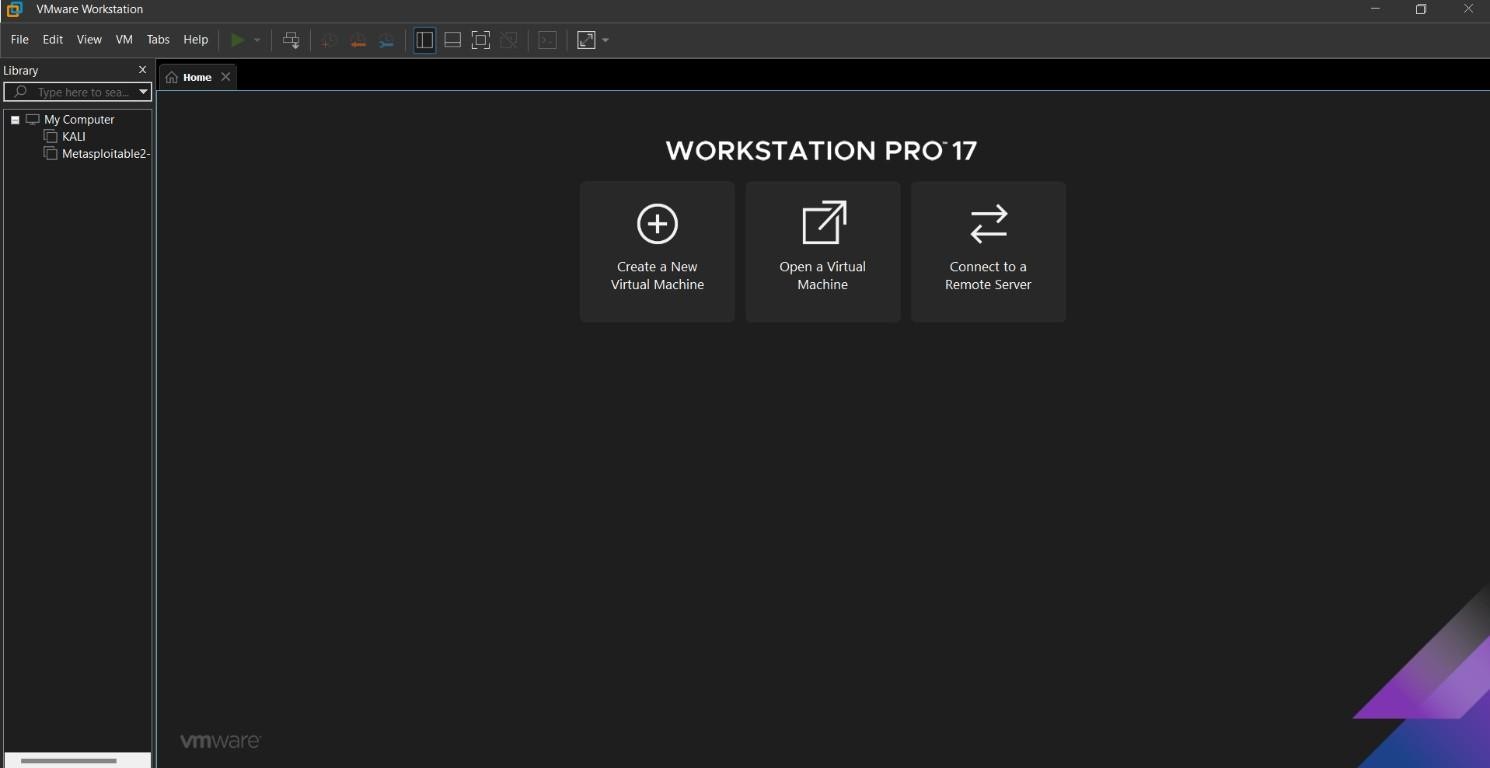
**Strong Community:** OpenStack boasts a vibrant community of experts, developers, and users who collaborate to enhance and improve the platform continually, ensuring its relevance and longevity in the rapidly evolving cloud computing landscape.

**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.

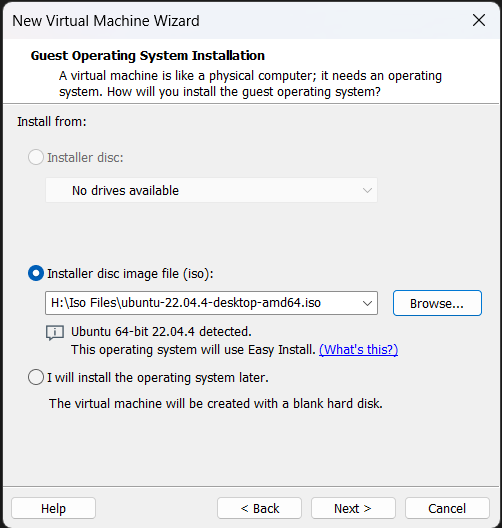
1. **STEPS TO IMPLEMENT:**

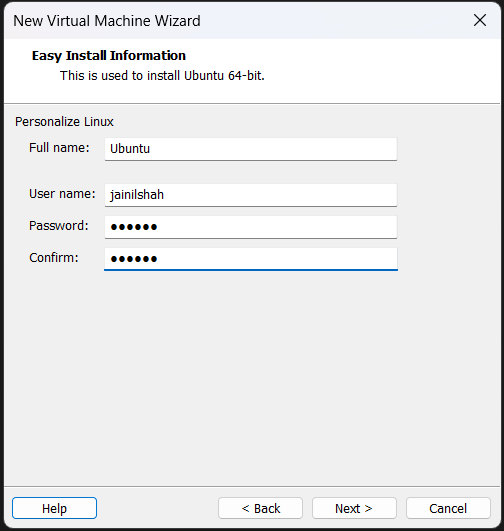
* **Step 1: Install Ubuntu on VMware**

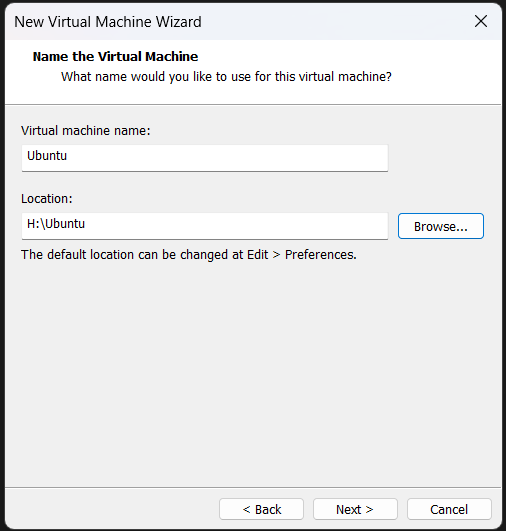
First, open VMware, then click "Creat a New Virtual Machine" to create a virtual machine.

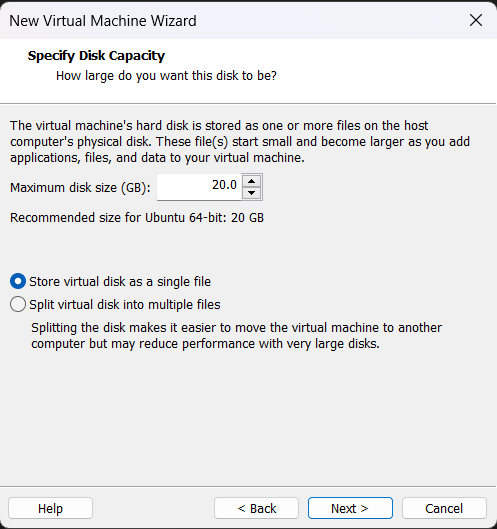


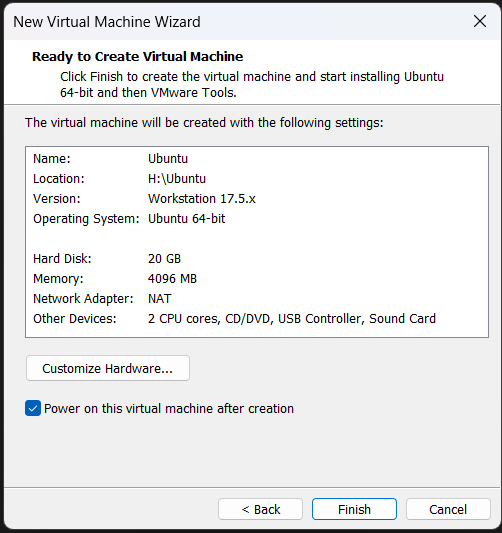
Select the .iso of the “[Ubuntu](https://ubuntu.com/download/desktop)” Operating System which you can download on their site and click “Next”.

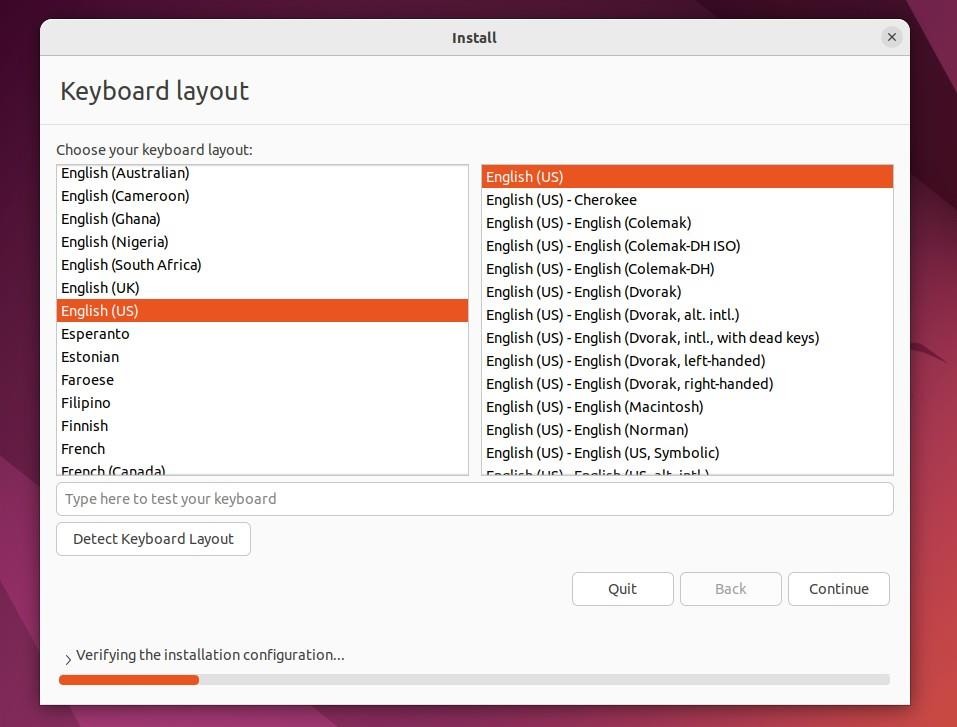


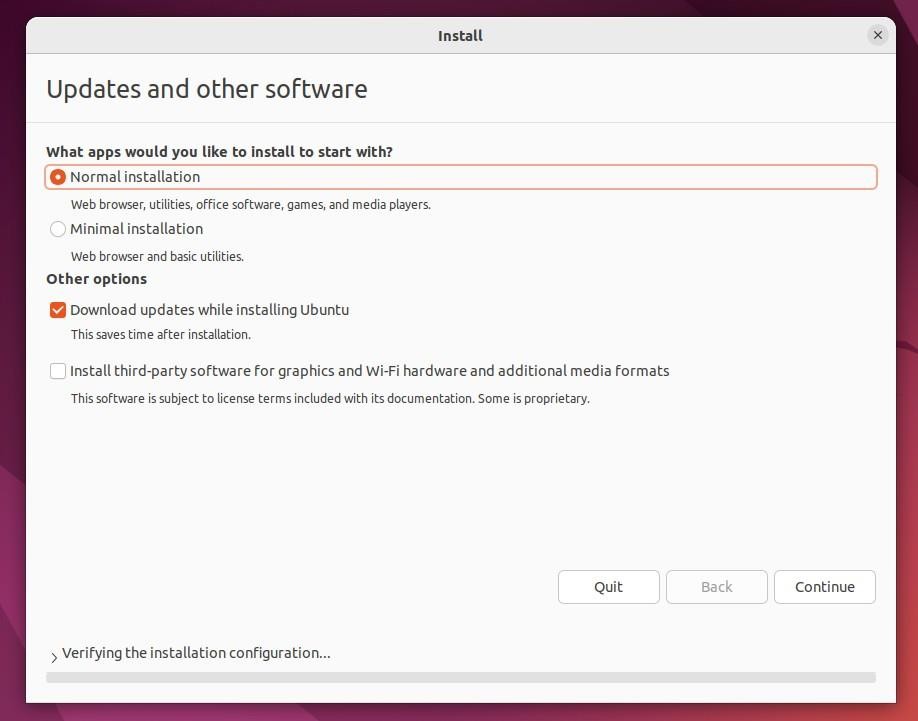
Enter your details like Username and password, then click “Next”:

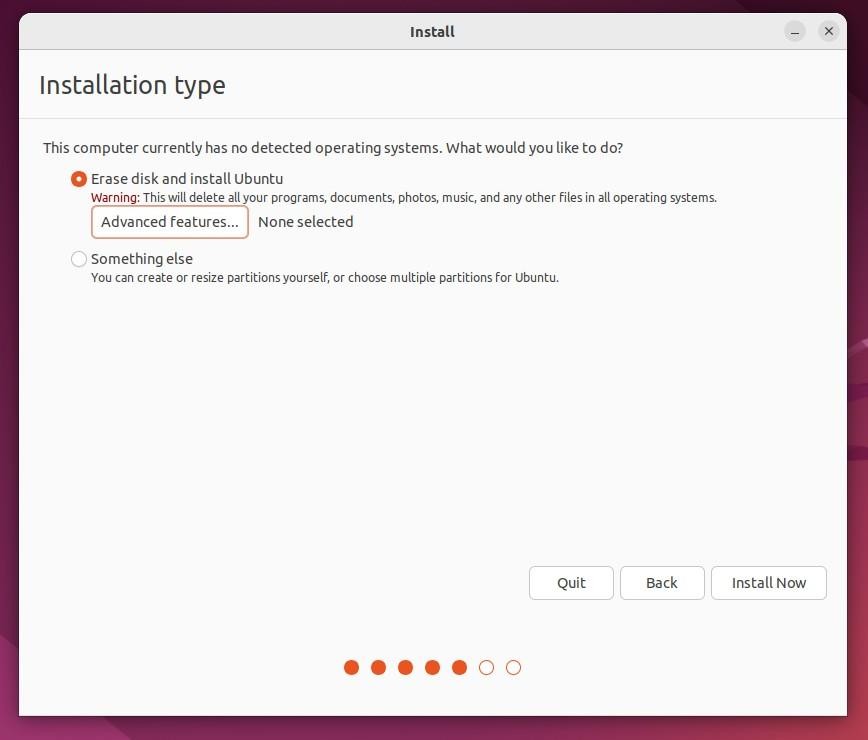
Name your OS and select the path where you want to install the Virtual OS:

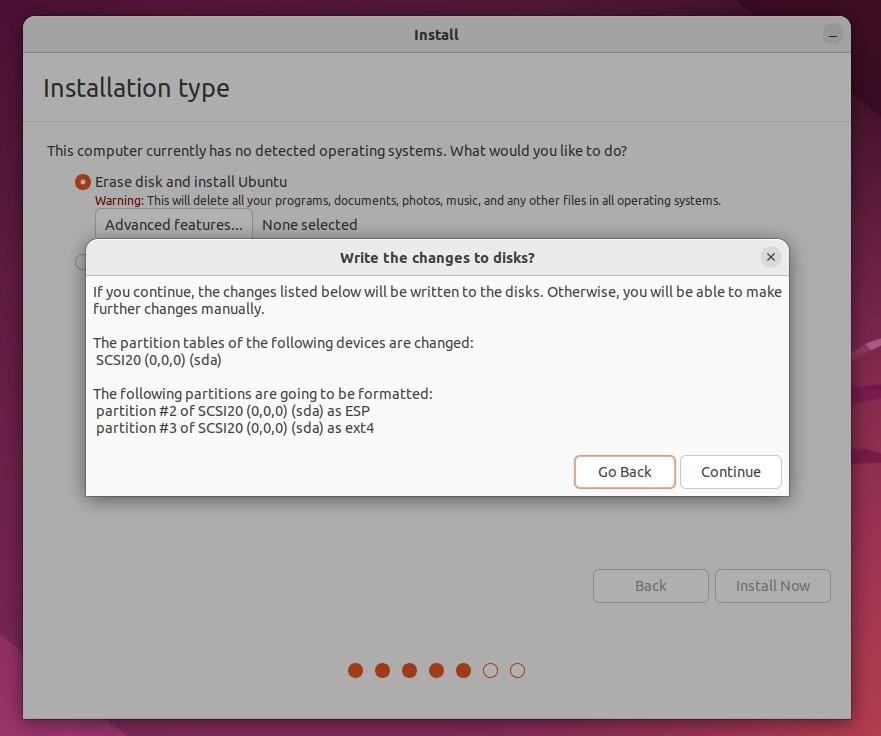
Specify disk capacity for the OS, default is 20GB:

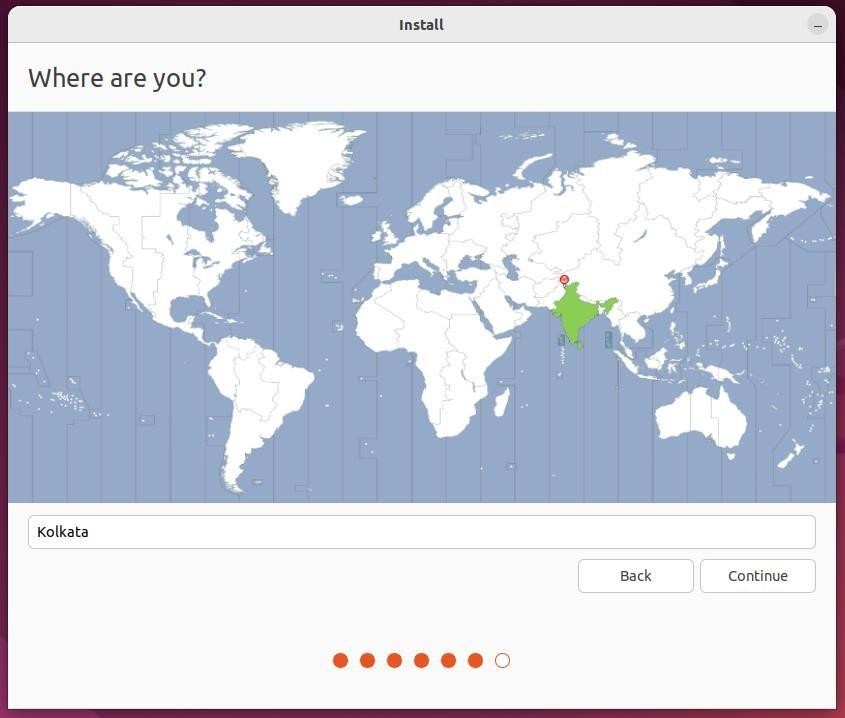
After that click “Finish” to proceed the installation of Ubuntu:

Select your preferred language and click “Continue”:

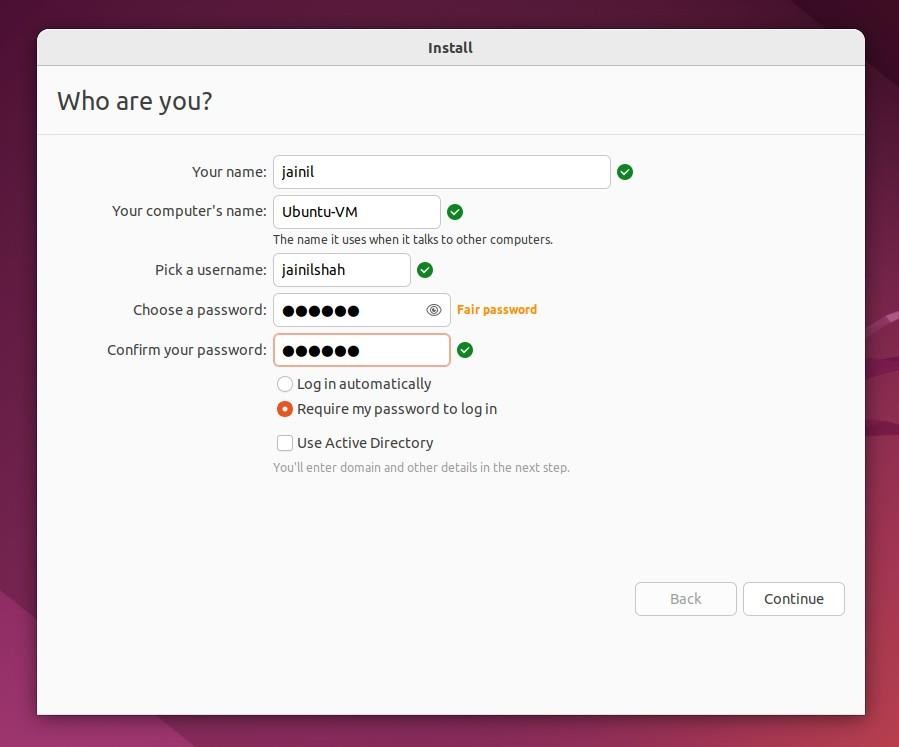
In the "Updates and other software" section, check "Normal installation" and continue.

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

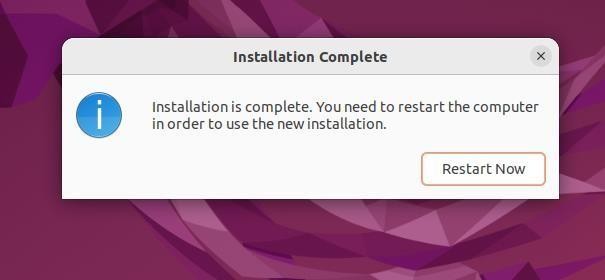
Click "Continue".

Choose your current location.

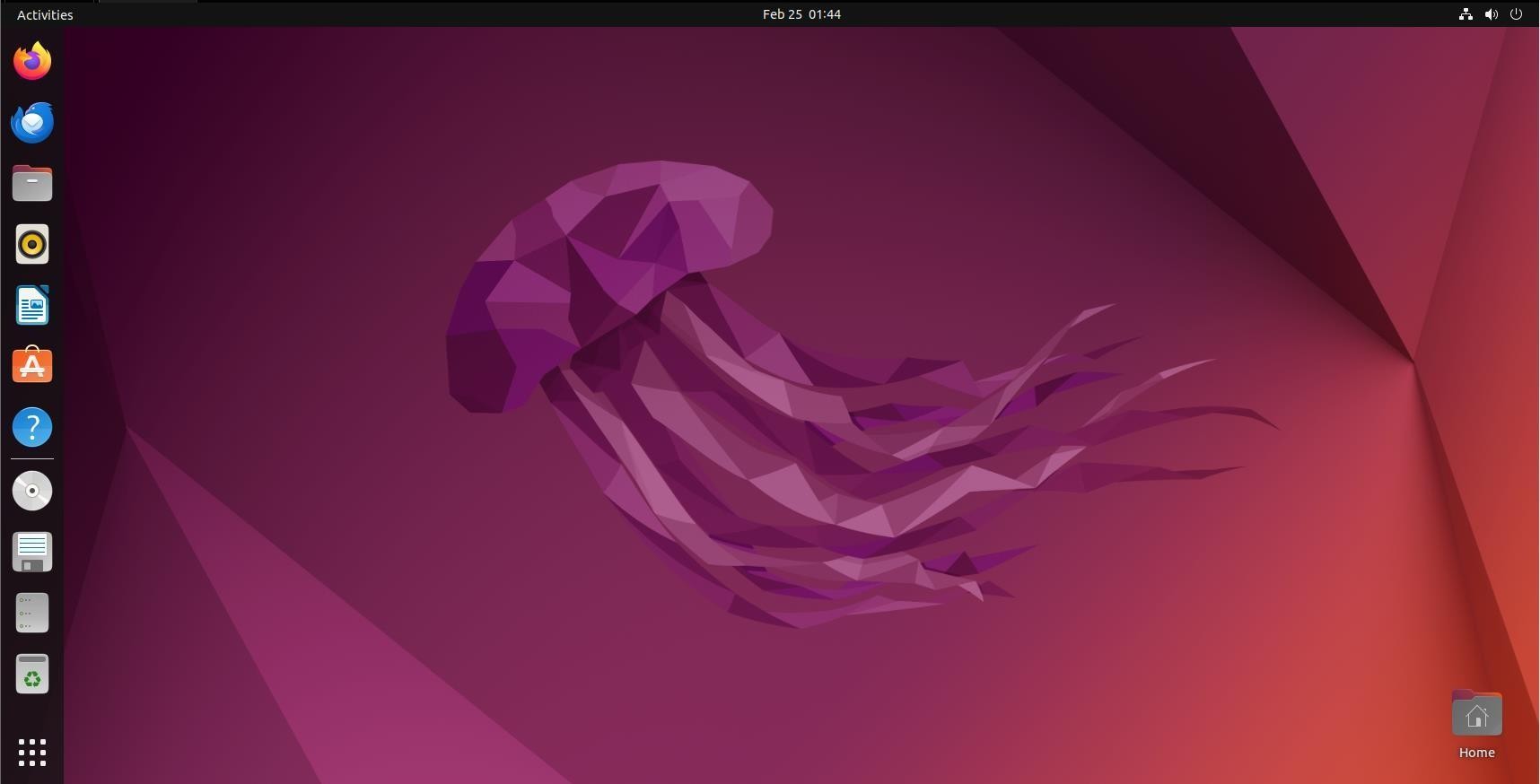
Now, set up your profile.



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



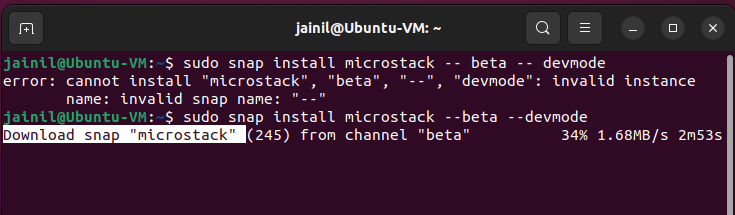
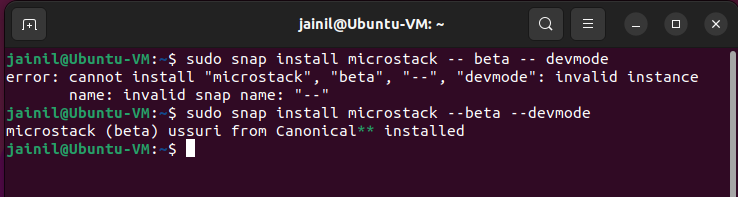
After logging in, you'll see the Ubuntu desktop.



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

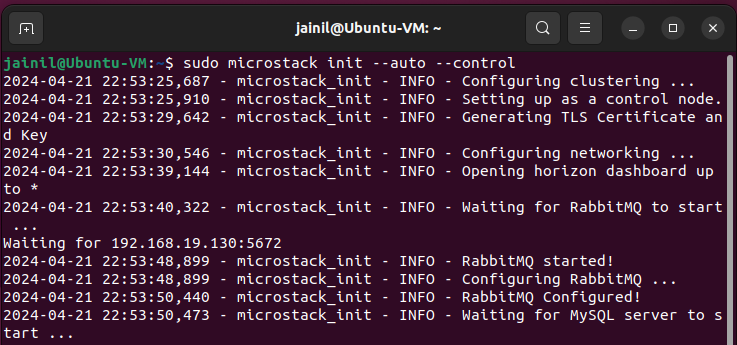
**Step 2:**

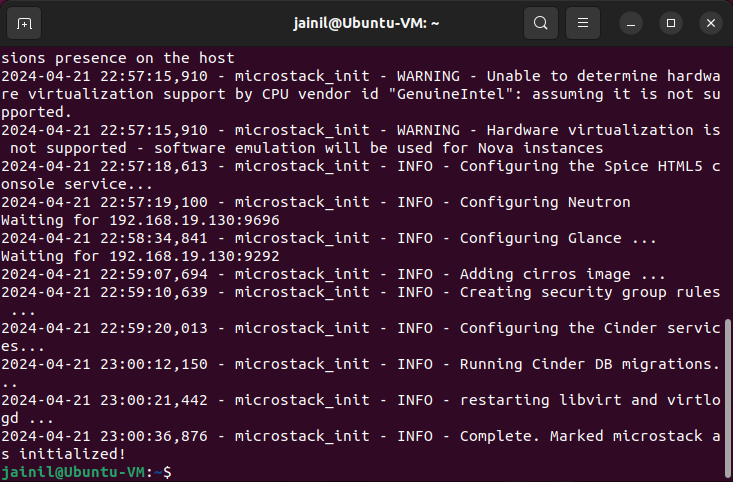
**Step 1:** Install OpenStack(microstack) on Ubuntu :

**Syntax:** sudo snap install microstack --beta –devmode

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

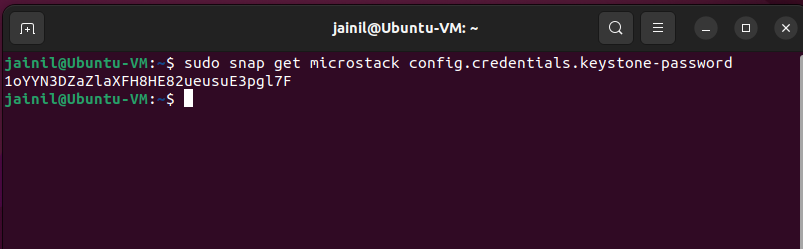
**Syntax:** sudo microstack init --auto --control





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

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

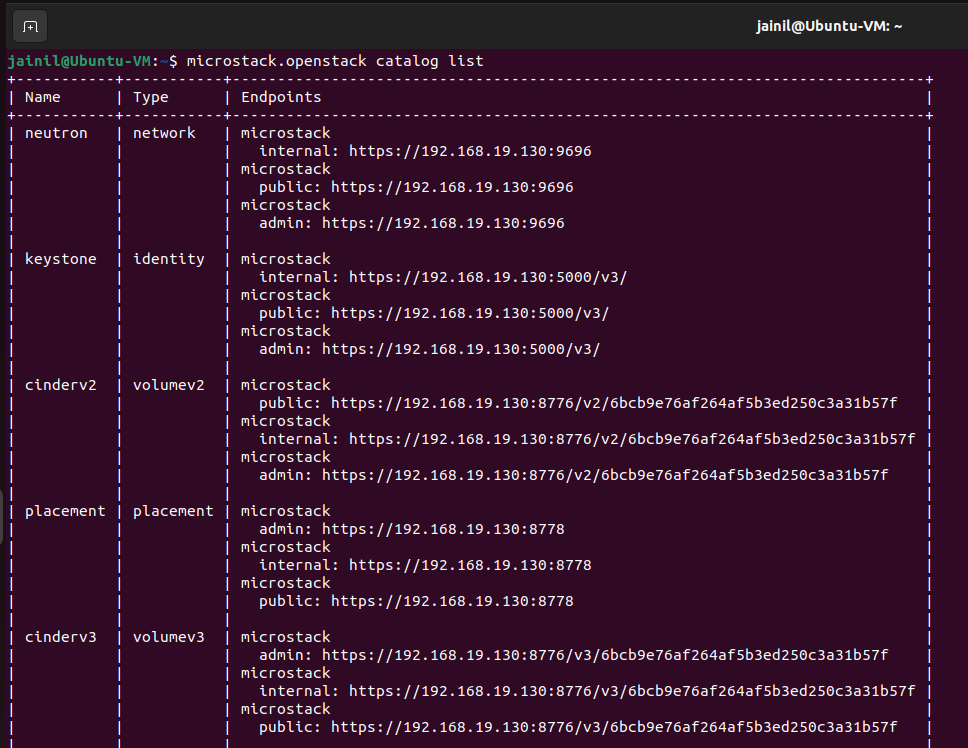


**User:** admin

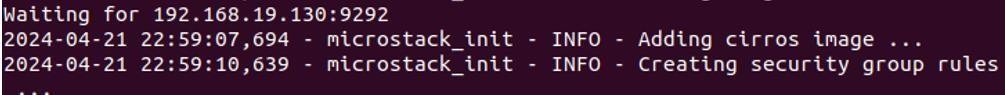
**Password:** 1oYYN3DZaZlaXFH8HE82ueusuE3pgl7F

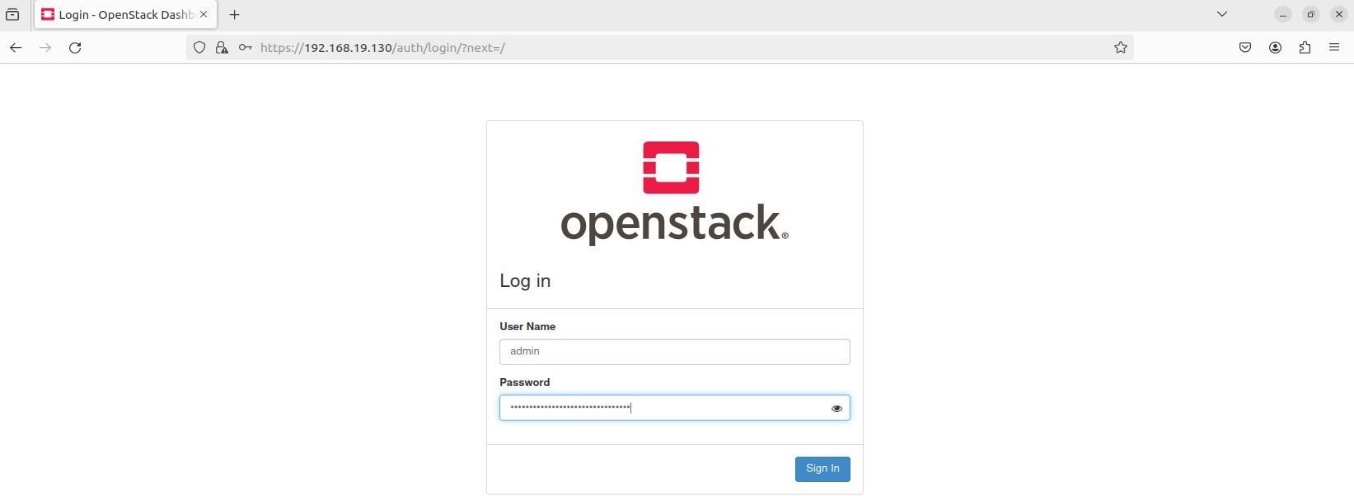
**Step 4:** List the available services in the OpenStack service catalog provided by MicroStack:

**Syntax:** microstack.openstack catalog list

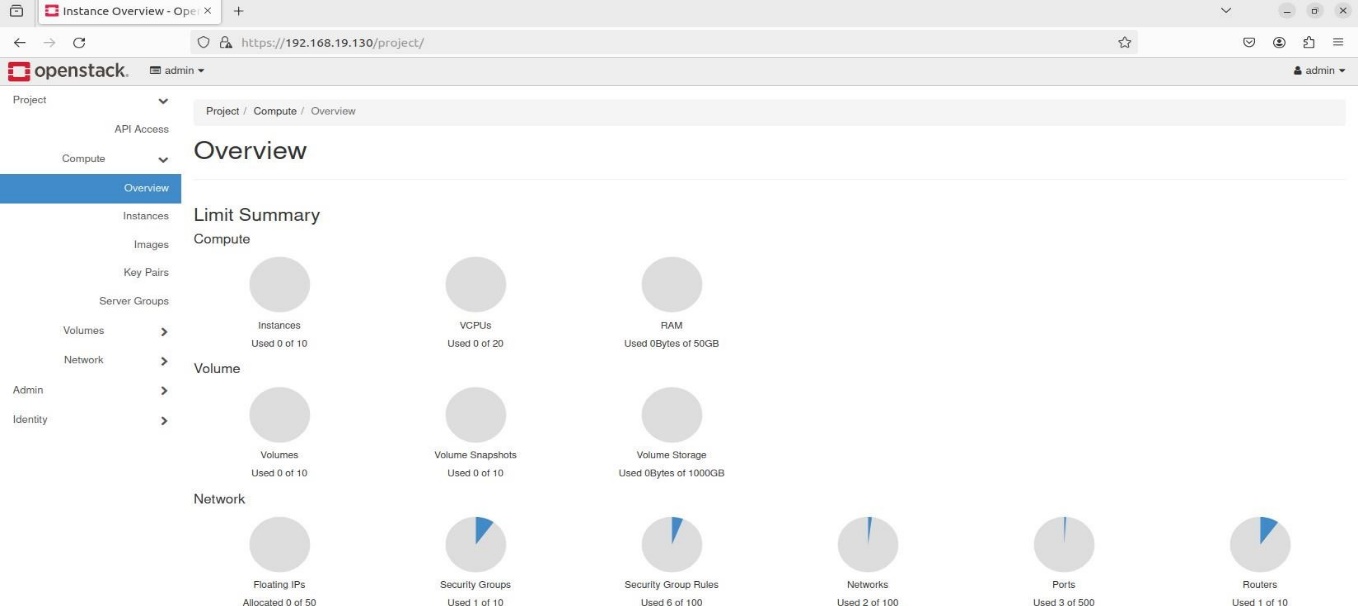


**Step 5:** Now open Firefox in Ubuntu and type the IP address given earlier :

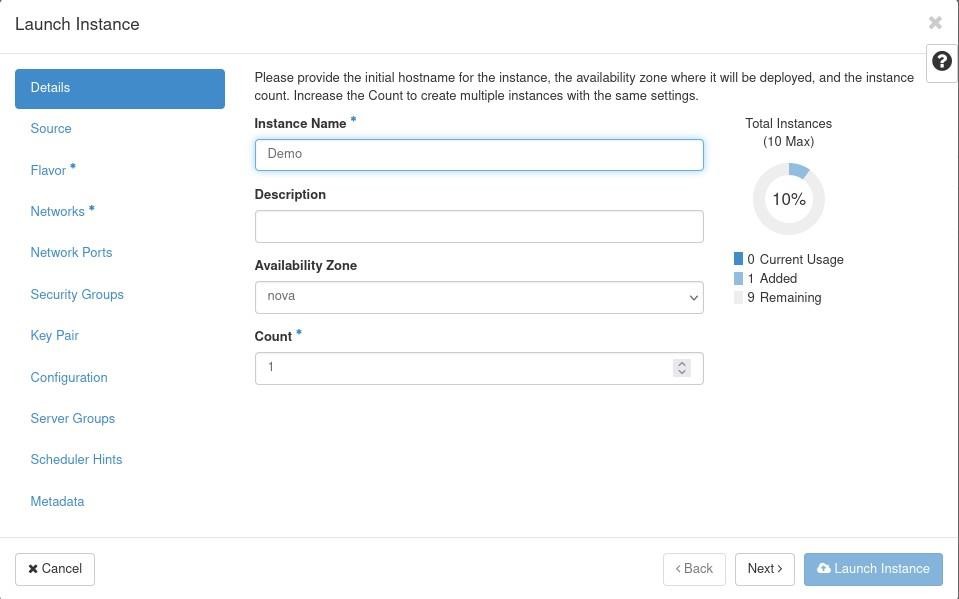
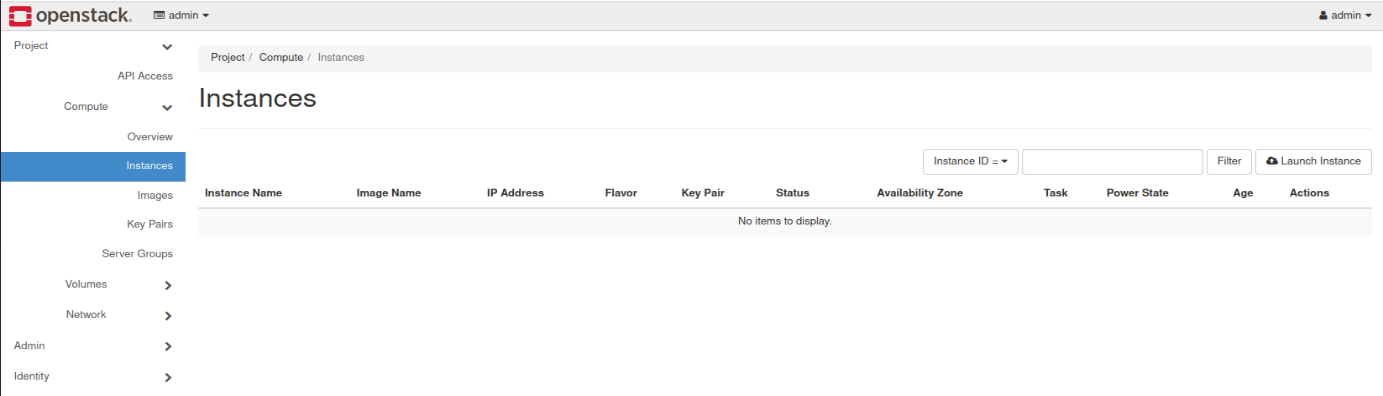


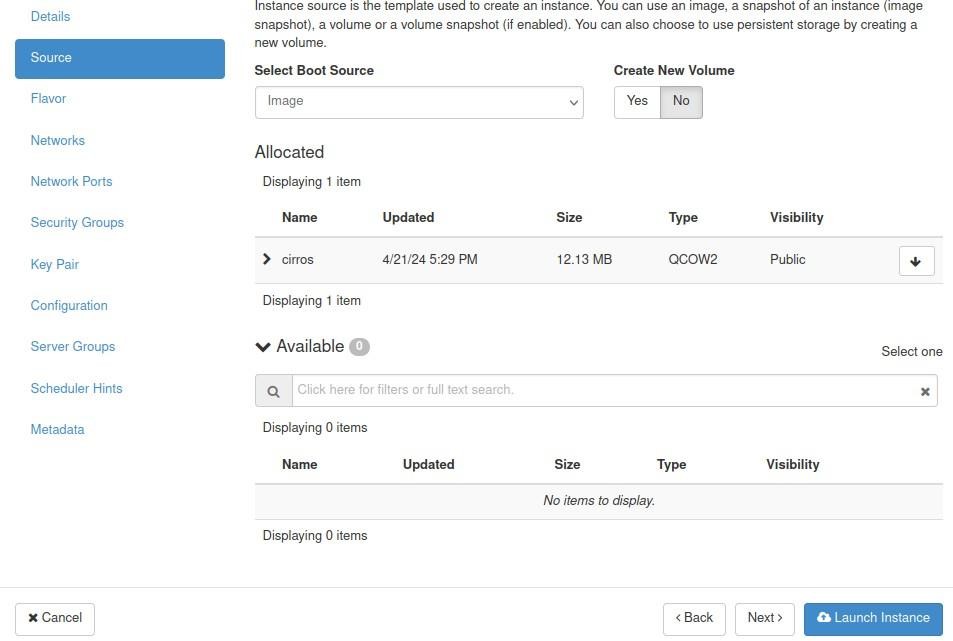


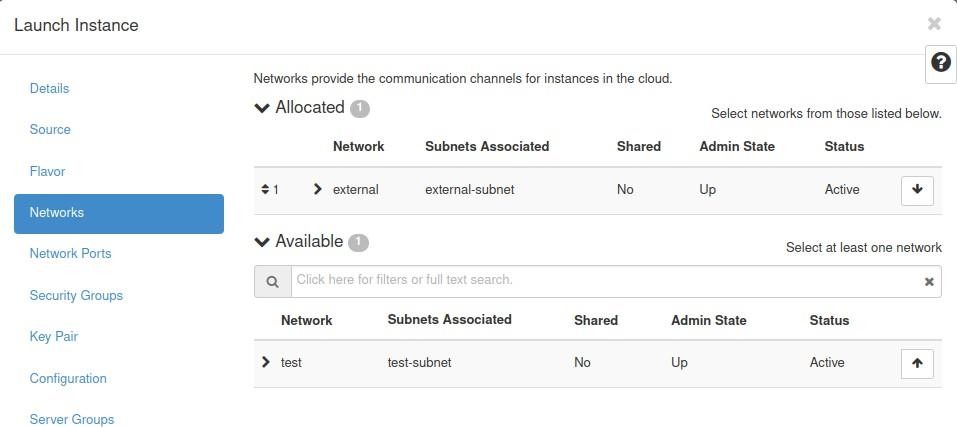
**OpenStack Dashboard:**



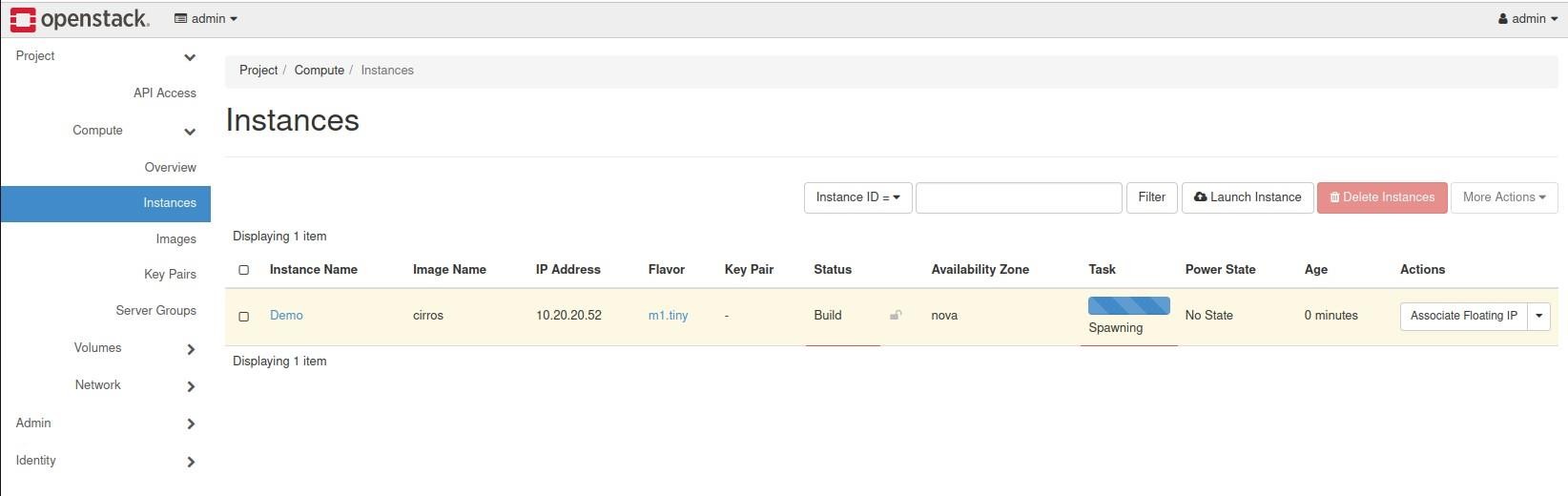
**Checking for instances:**



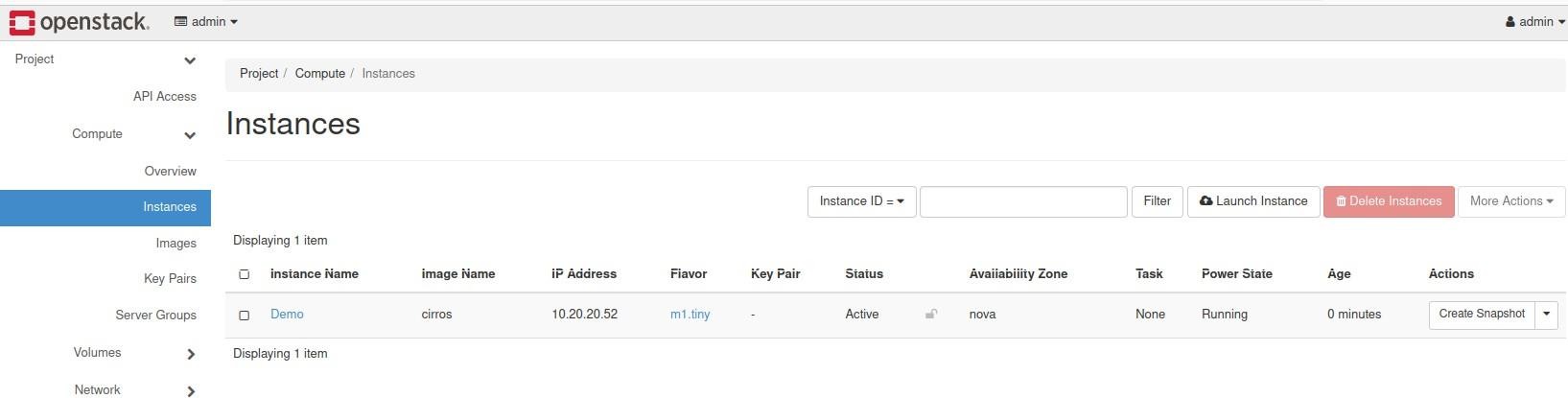
****



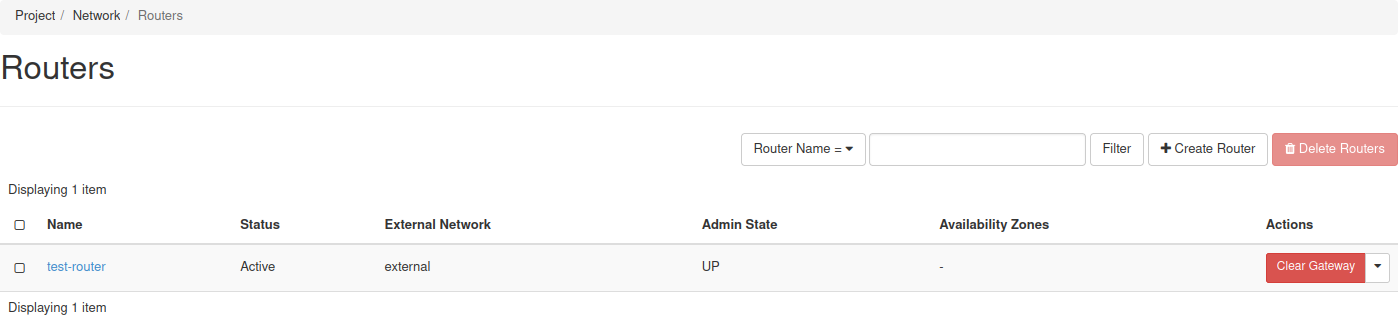
## **Instances is Spawning:**



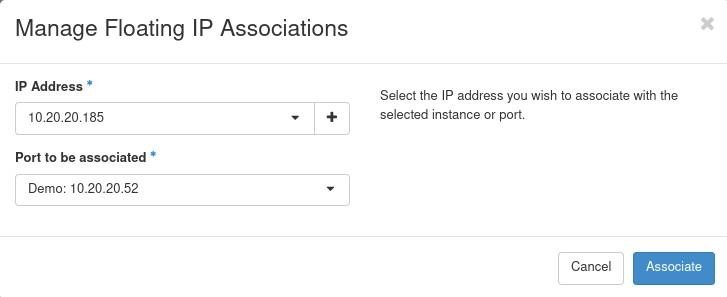
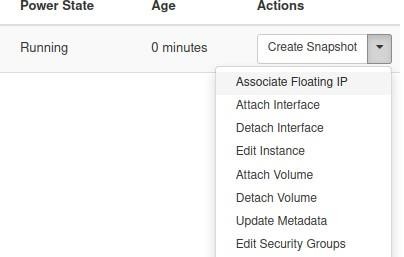
**Instances is ready:**



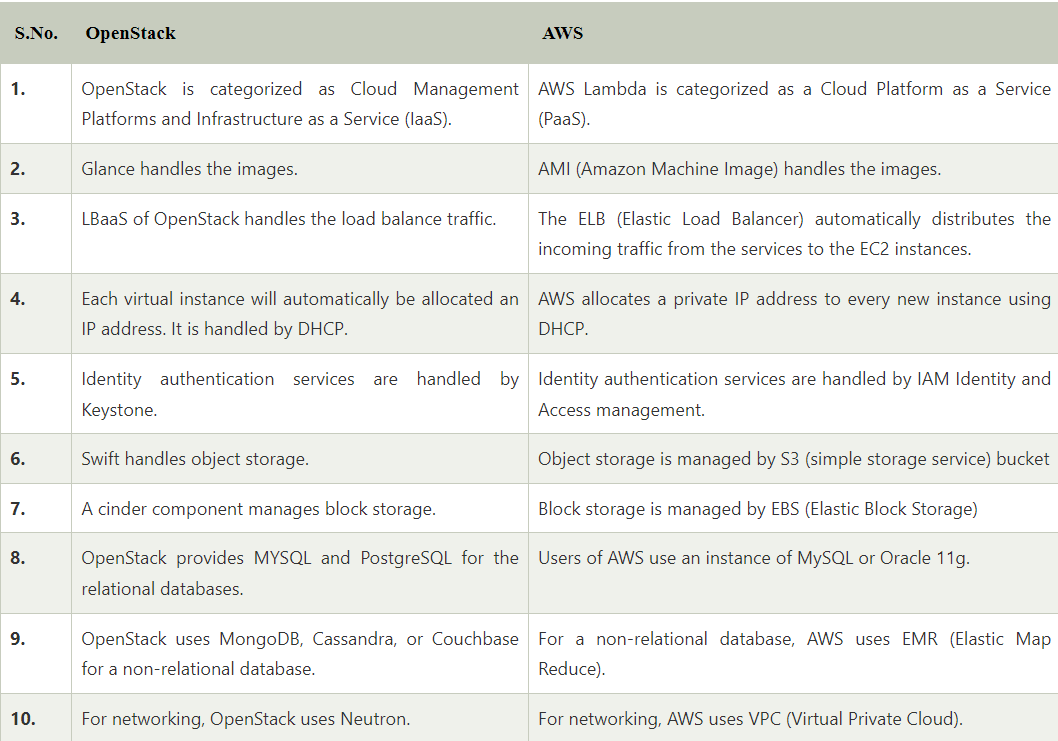
## **Set Routers Gateway:**



**Add the floating IP address :**



1. **COMPARSION WITH RELEVENT TECHNOLOGY AND SOFTWARE:**

****

1. **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.