

PROFESSIONAL TRAINING REPORT

Sathyabama Institute of Science and Technology

(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Technology Degree in Information Technology

By

Sakthi Vinayak

40120077



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600119. TAMILNADU

April 2023



SATHYABAMA

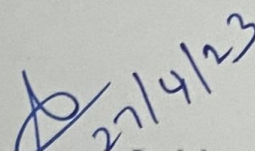
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

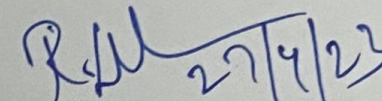
DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **Sakthi Vinayak(40120077)** who carried out the project entitled "**ESTABLISHING TENANT NETWORK IN OPENSTACK**" under our supervision from (January 2023 to April 2023).

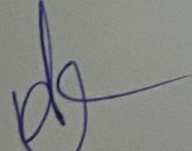

Internal Guide

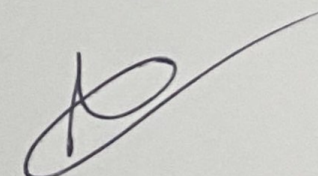
Dr.A.Sivasangari M.E., Ph.D.,


Head of the Department

Dr.R.Subhashini M.E., Ph.D.,

Submitted for Viva-voce Examination held on 29/4/23


Internal Examiner

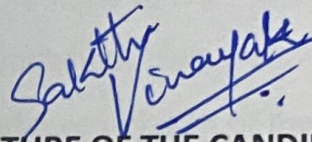

External Examiner

DECLARATION

I, **Sakthi Vinayak** hereby declare that the project report entitled **"ESTABLISHING TENANT NETWORK IN OPENSTACK"** done by me under the guidance of **Dr.A.Sivasangari, M.E., Ph.D.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Information technology.

DATE: 29/4/23

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.T.Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr.R.Subhashini, M.E.,Ph.D., Head of the Department of Information Technology** for providing me necessary details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project guide **Dr.A.Sivasangari, M.E., Ph.D.**, for her valuable guidance, Suggestions, and constant encouragement paved the way for the successful completion of the project work.

I wish to express my thanks to all teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE



इलेक्ट्रॉनिक्स एवं
सूचना प्रौद्योगिकी मंत्रालय
MINISTRY OF
ELECTRONICS AND
INFORMATION TECHNOLOGY
सत्यमेव जयते

Ref.No.: 19 /FSPCC/BC001/04/2023



COURSE COMPLETION LETTER

It is certified that

Sakthi Vinayak

Sathyabama Institute of Science and Technology

has successfully undergone the Bridge Course Training Program on

Cloud Computing Technology

Under "FutureSkills PRIME" Programme

commenced from 11 Mar, 2023

Shri. V. Sridharan
Chief Investigator
Cloud Computing Technology
Lead Resource Centre

L R PRAKASH

Digitally signed by L. R. PRAKASH
Date: 2023.04.26 02:34:01
+05'30'

Cdr.(Retd.) L.R. Prakash
Senior Director
Centre Head
C-DAC, Chennai

Abstract

In recent times, the concept of Cloud Computing has gained immense popularity and momentum. Simply put, cloud computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the Internet (“the cloud”) to offer faster innovation, flexible resources, and economies of scale. You typically pay only for cloud services you use, helping lower your operating costs, run your infrastructure more efficiently and scale as your business needs change.

The main pillars of cloud computing are Storage, Compute, and Networking. Out of these, Data storage, in particular, is a very important and valuable research field in cloud computing. Cloud storage security is now hugely popular. A recent survey showed that 95% of IT professionals are using cloud storage. This is not just a tech fad, it's here to stay. That number is expected to continue to grow rapidly. Storing, organizing, and accessing your digital files, photos, and videos in a central and secure storage space can be challenging. It's often difficult to store documents efficiently without taking up vast amounts of physical space for records storage or using valuable resources for records management.

As a result, document management systems have moved online to cloud file storage systems that also include document scanning tools and built-in document retention features. With more people opting to go paperless or work exclusively in the cloud, online document storage can be the solution that provides you with organized, accessible files anywhere you are. In this Mini-Project, we have implemented a local file storage system that acts as a cloud to store documents, images, videos files, audio files, etc on the local server. It is implemented as a secure file storage system with the help of UUID encryption in Django.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	6
	LIST OF FIGURES	9
1	INTRODUCTION	11
	1.1 CLOUD COMPUTING	11
	1.2 CLOUD DELIVERY MODELS	11
	1.3 CLOUD ENVIRONMENTS	12
2	AIM AND SCOPE	14
	2.1 AIM	14
	2.2 SCOPE	14
	2.3 ADVANTAGES OF CLOUD COMPUTING	14
	2.4 CHALLENGES IN CLOUD COMPUTING	15
3	LITERATURE SURVEY	17
	3.1 OPENSTACK ARCHITECTURE	17
	3.2 DYNAMIC AND ENERGY-EFFICIENT CONSOLIDATION OF VM USING OPENSTACK	18
4	METHODOLOGY	19
	4.1 SYSTEM REQUIREMENTS	19

	4.2 METHODOLOGY	20
	4.3 UBUNTU INSTALLATION	23
	4.4 KVM INSTALLATION	27
	4.5 OPENSTACK INSTALLATION	29
5	RESULTS AND DISCUSSION	34
	5.1 CREATE TENANT IN OPENSTACK AND LAUNCH INSTANCES	34
6	CONCLUSION	39

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
2.1	Challenges in Cloud Computing	15
4.1	Components of OpenStack	22
4.2	Rufus USB Creator	23
4.3	Diskpart CDM	24
4.4	Disk Management	25
4.5	Boot Manager	25
4.6	Installation Type	26
4.7	Setting Password	26
4.8	GRUB Menu	27
4.9	Check Processor	27
4.10	Verify Kernel	27
4.11	Verify KVM Module	28
4.12	Updating repository	28
4.13	qemu-kvm	28
4.14	libvirt-bin	28
4.15	bridge-utils	29
4.16	add user stack	29
4.17	echo statement	29
4.18	installing packages	29
4.19	Login using stack using OpenStack	30

4.20	Run Devstack	30
4.21	Openstack Dashboard	30
4.22	Uploading Images	31
4.23	Creating Network	31
4.24	Creating Instance	32
4.25	Volume creation	32
4.26	Mounting Volume	33
5.1	Create project	34
5.2	Create new user	34
5.3	Create network	35
5.4	Create router	35
5.5	Network topology	36
5.6	Create security group	36
5.7	Manage rules	37
5.8	Floating IPs	37
5.9	Create Instance	38
5.10	External connectivity	38

CHAPTER-1

INTRODUCTION

1.1 CLOUD COMPUTING

Cloud Computing is the delivery of computing services such as servers, storage, databases, networking, software, analytics, intelligence, and more, over the Cloud (Internet). The cloud environment provides an easily accessible online portal that makes handy for the user to manage the compute, storage, network, and application resources.

1.2 CLOUD DELIVERY MODELS

Cloud resources are provided in a variety of different delivery models that offer customers different levels of support and flexibility.

- **Infrastructure as a service**

IaaS is the on-demand delivery of computing infrastructure, including operating systems, networking, storage, and other infrastructural components. Acting much like a virtual equivalent to physical servers, IaaS relieves cloud users of the need to buy and maintain physical servers while also providing the flexibility to scale and pay for resources as needed. IaaS is a popular option for businesses that wish to leverage the advantages of the cloud and have system administrators who can oversee the installation, configuration, and management of operating systems, development tools, and other underlying infrastructure that they wish to use. However, IaaS is also used by developers, researchers, and others who wish to customize

the underlying infrastructure of their computing environment. Given its flexibility, IaaS can support everything from a company's computing infrastructure to web hosting to big data analysis.

- **Platform as a Service**

PaaS provides a computing platform where the underlying infrastructure (such as the operating system and other software) is installed, configured, and maintained by the provider, allowing users to focus their efforts on developing and deploying apps in a tested and standardized environment. PaaS is commonly used by software developers and developer teams as it cuts down on the complexity of setting up and maintaining computer infrastructure, while also supporting collaboration among distributed teams. PaaS can be a good choice for developers who don't have the need to customize their underlying infrastructure, or those who want to focus their attention on development rather than DevOps and system administration.

- **Software as a Service**

SaaS providers are cloud-based applications that users access on demand from the internet without needing to install or maintain the software. Examples include GitHub, Google Docs, Slack, and Adobe Creative Cloud. SaaS applications are popular among businesses and general users given that they're often easy to adopt, accessible from any device, and have free, premium, and enterprise versions of their applications. Like PaaS, SaaS abstracts away the underlying infrastructure of the software application so that users are only exposed to the interface they interact with.

1.3 CLOUD ENVIRONMENTS

Cloud services are available as public or private resources, each of which serves different needs.

- **Public Cloud**

The public cloud refers to cloud services (such as virtual machines, storage, or applications) offered publicly by a commercial provider to businesses and individuals. Public cloud resources are hosted on the commercial provider's hardware, which users access through the internet. They are not always suitable for organizations in highly-regulated industries, such as healthcare or finance, as public cloud environments may not comply with industry regulations regarding customer data.

- **Private Cloud**

The private cloud refers to cloud services that are owned and managed by the organization that uses them and available only to the organization's employees and customers. Private clouds allow organizations to exert greater control over their computing environment and their stored data, which can be necessary for organizations in highly-regulated industries. Private clouds are sometimes seen as more secure than public clouds as they are accessed through private networks and enable the organization to directly oversee their cloud security. Public cloud providers sometimes provide their services as applications that can be installed on private clouds, allowing organizations to keep their infrastructure and data on premise while taking advantage of the public cloud's latest innovations.

- **Hybrid Cloud & Multi cloud**

Many organizations use a hybrid cloud environment which combines public and private cloud resources to support the organization's computing needs while maintaining compliance with industry regulation. Multi cloud environments are also common, which entail the use of more than one public cloud provider (for example, combining Amazon Web Services and Digital Ocean).

CHAPTER-2

AIM AND SCOPE

2.1 AIM

Private or public, the goal of cloud computing is to provide easy, scalable access to computing resources and IT services. Cloud infrastructure involves the hardware and software components required for proper implementation of a cloud computing model.

2.2 SCOPE

The availability of on-demand computing resources, particularly computing power and data storage, without having the user manage it directly. In other words, cloud computing is the process of sharing such resources through the internet.

2.3 ADVANTAGES OF CLOUD COMPUTING

- **Back-up and restore data**

Once the data is stored in the cloud, it is easier to get back up and restore that data using the cloud.

- **Improved collaboration**

Cloud applications improve collaboration by allowing groups of people to quickly and easily share information in the cloud via shared storage.

- **Excellent accessibility**

Cloud allows us to quickly and easily access store information anywhere, anytime in the whole world, using an internet connection. An internet cloud infrastructure increases organization productivity and efficiency by ensuring that our data is always accessible.

- **Low maintenance cost**

Cloud computing reduces both hardware and software maintenance costs for organizations.

- **Mobility**

Cloud computing allows us to easily access all cloud data via mobile.

- **Services in the pay-per-use model**

Cloud computing offers Application Programming Interfaces (APIs) to the users for access services on the cloud and pays the charges as per the usage of service.

- **Unlimited storage capacity**

Cloud offers us a huge amount of storage capacity for storing our important data such as documents, images, audio, video, etc. in one place.

- **Data security**

Data security is one of the biggest advantages of cloud computing. Cloud offers many advanced features related to security and ensures that data is securely stored and handled.

2.4 CHALLENGES IN CLOUD COMPUTING

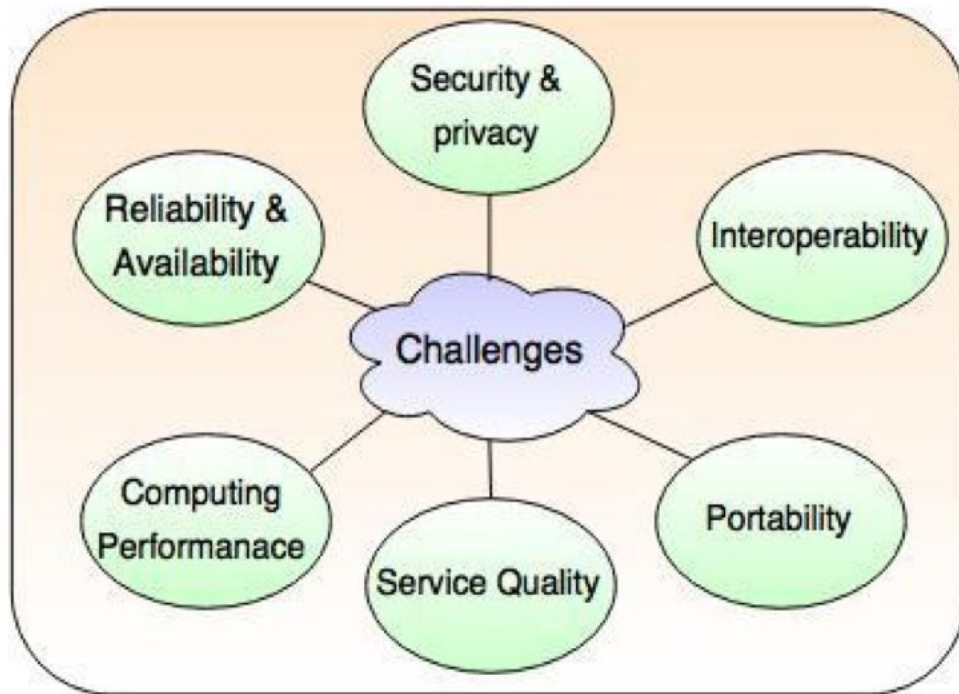


Fig 2.1 Challenges in cloud computing

- **Security and Privacy**

- Security and privacy are the main challenge in cloud computing.
- These challenges can reduced by using security applications, encrypted file systems, data loss software.

- **Interoperability**

- The application on one platform should be able to incorporate services from the other platform. This is known as Interoperability.
- It is becoming possible through web services, but to develop such web services is complex.

- **Portability**

- The applications running on one cloud platform can be moved to new cloud platform and it should operate correctly without making any changes in design, coding.
- The portability is not possible, because each of the cloud providers uses different standard languages for their platform.

- **Service Quality**

The Service-Level Agreements (SLAs) of the providers are not enough to guarantee the availability and scalability. The businesses disinclined to switch to cloud without a strong service quality guarantee.

- **Computing Performance**

- High network bandwidth is needed for data intensive applications on cloud, this results in high cost.
- In cloud computing, low bandwidth does not meet the desired computing performance.

- **Reliability and Availability**

Most of the businesses are dependent on services provided by third-party, hence it is mandatory for the cloud systems to be reliable and robust.

CHAPTER- 3

LITERATURE SURVEY

3.1 OVERVIEW OF OPENSTACK ARCHITECTURE

OpenStack is a architecture is a cloud computing platform that provides virtualized computing resources on pay-per-use basis. It is an open-source architecture that offers scalability and efficient resource management, making it a sustainable and robust private cloud solution. The architecture consists of several essential services, including compute, networking, storage, Identity, Image service, telemetry, orchestration, and database. These services work together to provide a complete cloud infrastructure that can be adapted to any hardware infrastructure. OpenStack leverages its modular architecture and simplicity to implement not only in major

datacentre but also in entry-level hardware providing enterprises with a massively scalable cloud infrastructure.

OpenStack architecture consists of several essential components that work together to provide a complete cloud infrastructure. These components are divided into five groups based on their characteristics, such as computing, networking, storing, shared, and supporting services. Here is a brief overview of each component:

1. Compute: This component provides virtual machines(vms) and manages their lifecycle.
2. Networking: This component provides network connectivity between VMs and external networks.
3. Storage: This component provides block storage and object storage services.
4. Identity: This component provides authentication and authorization services for OpenStack users.
5. Image service: this component provides a catalog of virtual machine images that can be used to create new VMs.
6. Telemetry: This component collects data about the OpenStack environment's performance and usage.
7. Orchestration: This component automates the deployment of complex application on OpenStack.
8. Database: This component provides database services for OpenStack applications.

In addition to these core components, there are also supporting services such as messaging, Dashboard, and DNS that are essential for OpenStack's operation but not part of its core architecture.

3.2 DYNAMIC & ENERGY-EFFICIENT CONSOLIDATION OF VM USING OPENSTACK

Dynamic consolidation of virtual machines improves resource utilization and reduce energy consumption in cloud data centres by consolidating VMs onto fewer physical servers, which allows the remaining idle servers to be powered off or put into a low-power state. This reduces the overall energy consumption of the data centre while maintaining the same level of service quality. The experiment

results confirm that dynamic VM consolidation is able to significantly reduce energy consumption in an IaaS cloud with a limited performance impact. The framework is designed for dynamic and energy-efficient consolidation of virtual machines in OpenStack clouds.

The paper proposes an architecture and open-source implementation of OpenStack neat, which can be configured to use custom VM consolidation algorithms and transparently integrates with existing OpenStack deployments without modifying their configuration. The paper also proposes a benchmark suite for evaluating and comparing dynamic VM consolidation algorithms. Dynamic consolidation of virtual machines improves resource utilization and reduces energy consumption in cloud data centres by consolidating VMs onto fewer physical servers, which allows the remaining idle servers to be powered off or put into a low-power state.

CHAPTER-4

METHODOLOGY

4.1. SYSTEM REQUIREMENTS

4.1.1. *Ubuntu*

The latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years of free security and maintenance updates, guaranteed until April 2027.

- GHz dual-core processor or better.
- GB system memory.
- 25 GB of free hard drive space.

- Internet access is helpful.
- Either a DVD drive or a USB port for the installer media.

4.1.2 Openstack:

- Ubuntu
 - Ubuntu 16.04 LTS (Xenial Xerus)
 - Linux kernel version 3.13.0-34-generic or later is required.
- CentOS (support is experimental)
 - Centos 7, fully updated.
 - Linux kernel version 3.10.0 or later.
- openSUSE (support is experimental)
 - Leap 42.X, fully updated.
 - Linux kernel version 4.4.X or later

4.2 METHODOLOGY

4.2.1 OpenStack

OpenStack is a cloud OS that is used to control the large pools of computing, storage

and networking resources within a data centre. Open Stack is open-source and free software. This is basically used for cloud computing and deployed as an IaaS. 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.

4.2.2 Components of Openstack

Major components of OpenStack are given below:

- **Compute** - 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** - 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**- 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** -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**- 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** - 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** - 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** - It is used to meter the usage and report it to OpenStack's individual users. So basically, telemetry provides billing services to OpenStack's individual users.
- **Orchestration** - 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** - It offers storage of the file to a virtual machine. This component gives an infrastructure for managing and provisioning file shares.
- **Elastic Map-reduce** - 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.

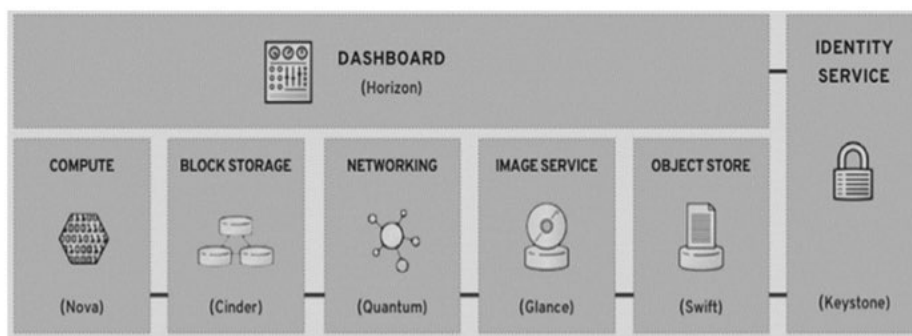


Fig 4.1 Components of OpenStack

4.3 UBUNTU INSTALLATION

Ubuntu is the best Linux distro for beginners. So many Linux beginners prefer to dual boot Windows 10 and Ubuntu, instead of completely removing Windows.

1. Get Ubuntu and create the bootable USB

As we going to install Ubuntu, so first we need to get the ISO file, get it from the official ubuntu website , After downloading the Ubuntu ISO, we need to create a bootable USB to install from it. There's many way to do that, in windows 10, we can use the Rufus USB creator, download Rufus

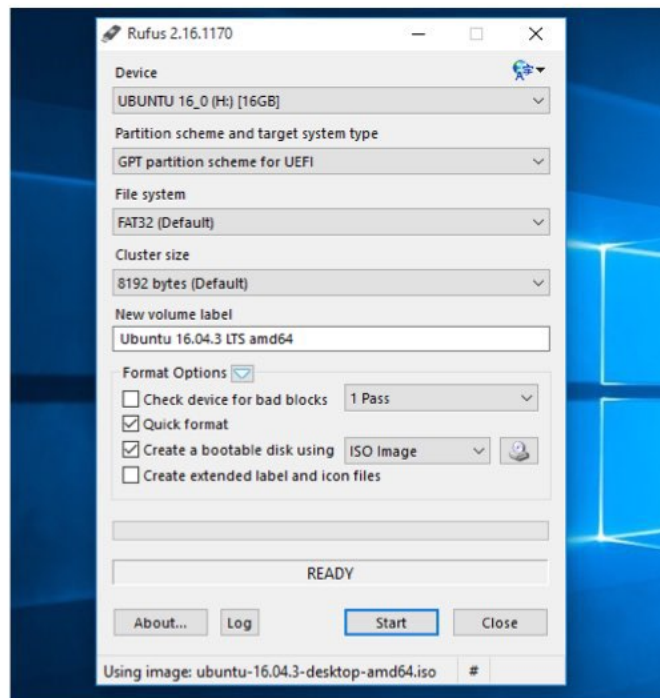


Fig 4.2 Rufus USB Creator

2. Analyze the system

Although this step is optional, but it's to know our system before altering anything. First check the hard disk partition type. We can use the diskpart command on windows to check the HDD, first open up the CMD as administrator and type the command below one by one.

```
Administrator: Command Prompt - diskpart
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Windows\system32>diskpart

Microsoft DiskPart version 10.0.10240

Copyright (C) 1999-2013 Microsoft Corporation.
On computer: DESKTOP-02DFD26

DISKPART>

DISKPART> list disk

   Disk ###  Status       Size      Free      Dyn  Gpt
   -----  -
   Disk 0    Online        465 GB         0 B
   Disk 1    Online        37 GB    1024 KB          *
```

Fig 4.3 Diskpart CDM

3. Partition the disk on windows 10

We need to free up some HDD space to dual boot ubuntu and windows 10 , at least 6 GB is required to install. However we can go with 8 GB, but about 20 GB for the linux filesystem root / partition is highly recommended. First open up the Disk Management tool from the settings application. Select the hard drive partition you want to shrink. As example, if our C drive has enough free space, then select it. Shrink the volume to desired size, anything between 10GB to 100GB should be enough.

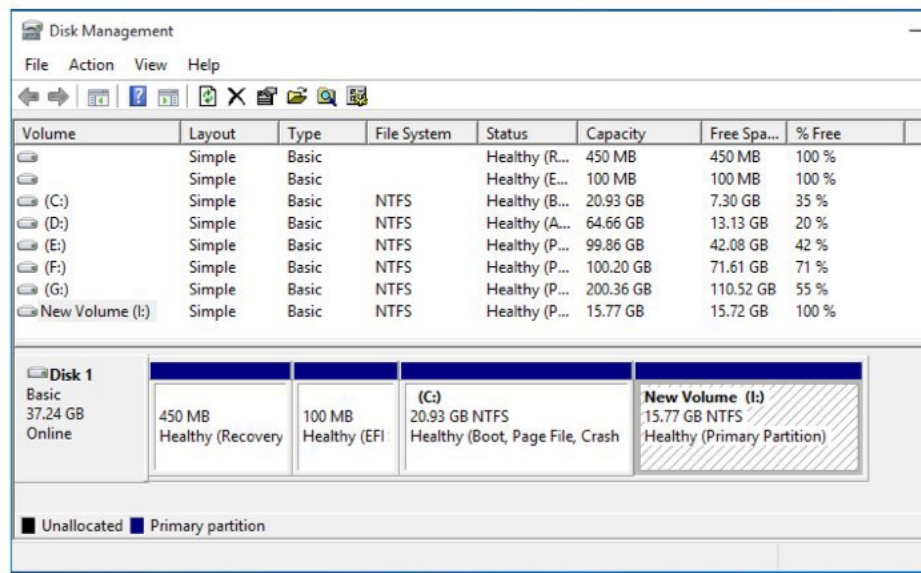


Fig 4.4 Disk management

4. Reboot the computer and boot from USB

Now reboot the PC and boot it from the USB drive. we need to tap the F12, F10 or F8 key many times to select the boot device. On Dell, Acer, Lenovo or Toshiba laptops, it's the F12 key in most cases.

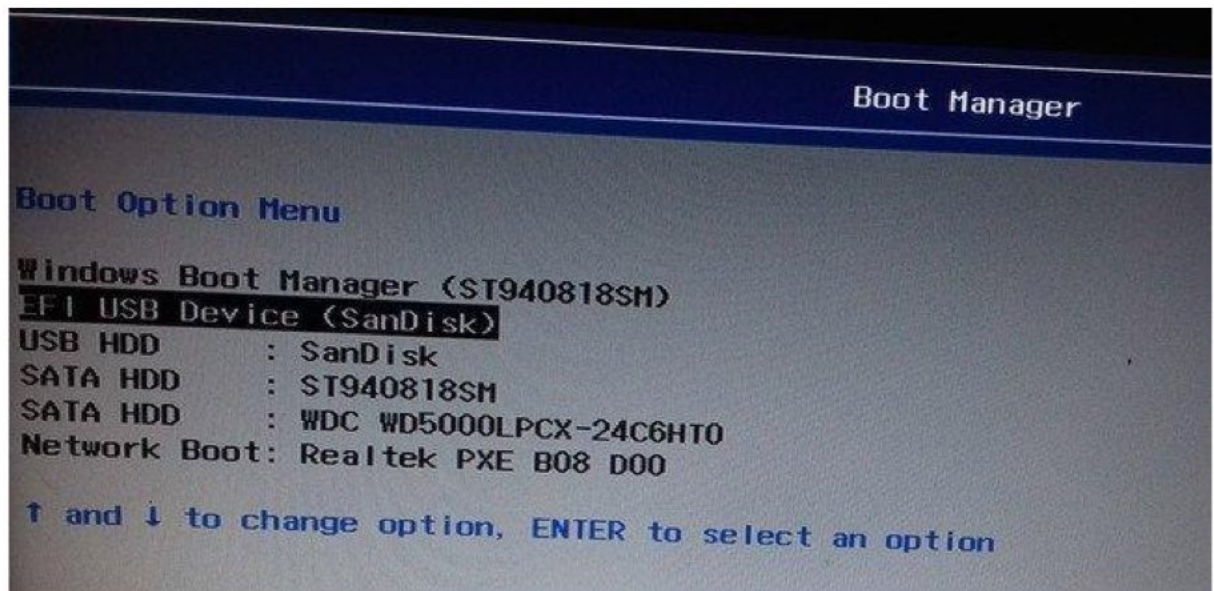


Fig 4.5 Boot manager

5. Dual boot windows 10 and Ubuntu

If we not selected the first option, then just click on the Install Ubuntu shortcut on the desktop, else the system will start the installer directly.

- The first step is to select the language, choose we preferred language and press the continue button.
- Next we will be asked to connect to internet if not connected, this step is optional, we can skip this step by selecting the first ubuntu install option.
- Now the installer will ask we to install the updates and third party software like graphics drivers or multimedia codecs. We can skip this step as well, just press Continue.
- The next step is Installation type, important to dual boot windows 10 and ubuntu. As the PC already has windows 10, we must select the Something else option, highlighted on the screen shot below.

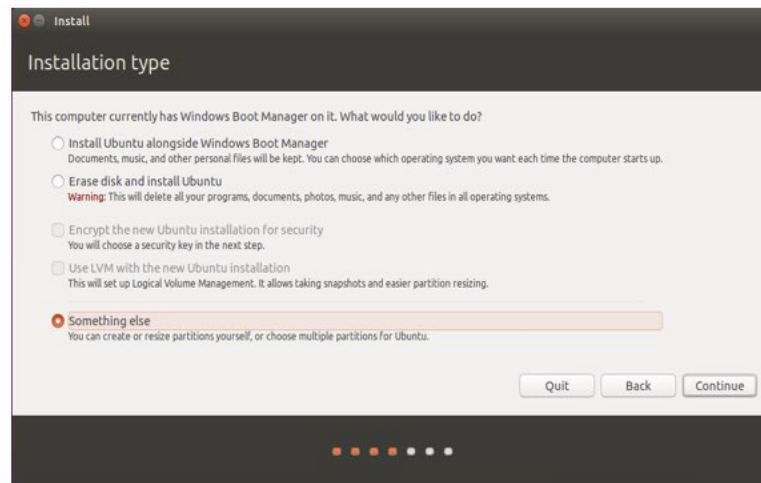


Fig 4.6 Installation Type

- Next we want to fill some basic information about our computer name, password and so on, a screenshot below.

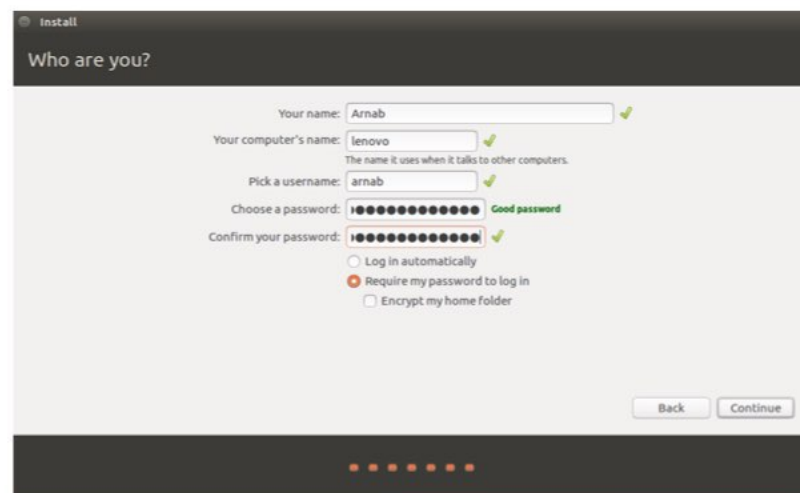


Fig 4.7 Setting Password

- We have to press the Continue button again, and we are almost done. The installation process will take some time, between 10 to 20 minute, till then watch the slide show. Reboot now and draw out the installation USB drive right after seeing the OEM logo on the screen.

6. Testing the new dual boot Ubuntu installation

After rebooting, we can see the GRUB boot menu, listing Ubuntu and windows.

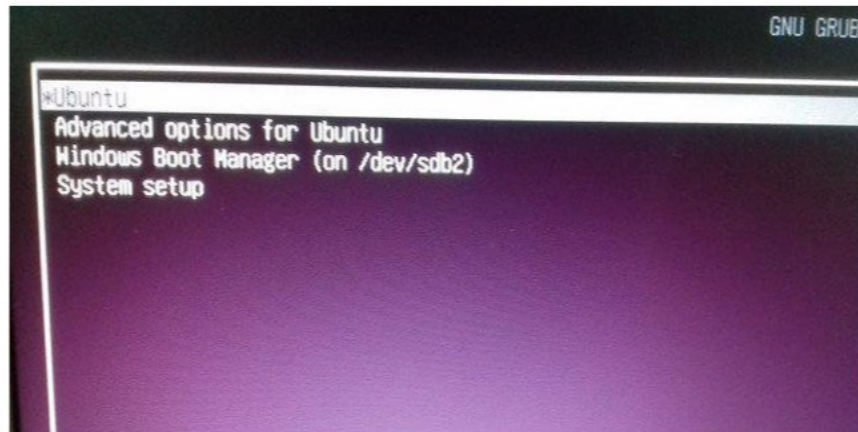


Fig 4.8 GRUB Menu

4.4 KVM INSTALLATION

1. Use a 64 bit kernel

Running a 64 bit kernel on the host operating system is recommended but not required. To serve more than 2GB of RAM for your VMs, you must use a 64-bit kernel. On a 32-bit kernel install, you'll be limited to 2GB RAM at maximum for a given VM.

2. To check if your processor is 64-bit, execute:

```
cloud@openstack:~$ egrep -c 'lm' /proc/cpuinfo
2
cloud@openstack:~$
```

Fig 4.9 Check processor is 64-bit

3. To verify if the running kernel is 64-bit:

```
root@meghvmserve:/home/cloud# uname -a
Linux meghvmserve 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u2 (2019-11-11) x86_64 GNU/Linux
root@meghvmserve:/home/cloud#
```

Fig 4.10 Verify kernel is 64-bit

4. Verify if the KVM module is loaded:

```
root@meghvmserve:/home/cloud# lsmod | grep kvm
kvm_intel          233472    80
kvm                749568    1 kvm_intel
irqbypass         16384    10 kvm
root@meghvmserve:/home/cloud#
```

Fig 4.11 Verify KVM module

5. Updating Repository:

```
root@openstack:~# source proxy
root@openstack:~# apt-get update
Hit:1 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:5 http://security.ubuntu.com/ubuntu bionic-security/main i386 Packages [513 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [805 kB]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/main amd64 DEP-11 Metadata [46.1 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [687 kB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/universe i386 Packages [631 kB]
Get:10 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 DEP-11 Metadata [49.2 kB]
Get:11 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 DEP-11 Metadata [2,464 B]
Fetched 2,823 kB in 21s (133 kB/s)
Reading package lists... Done
root@openstack:~#
```

Fig 4.12 Updating Repository

6. qemu -kvm

```
root@openstack:~# apt-cache search qemu-kvm
qemu-kvm - QEMU Full virtualization on x86 hardware
root@openstack:~#
```

Fig 4.13 qemu-kvm

7. libvirt-bin

```

root@openstack:~# apt-get install qemu-kvm
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  cpu-checker ibverbs-providers ipxe-qemu ipxe-qemu-256k-compatible-efi-roms
  libaio1 libcacard0 libfdt1 libibverbs1 libiscsi7 libnl-route-3-200 librados2
  librbd1 librdmacm1 libsd1.2debian libspice-server1 libusbredirparser1
  libxen-4.9 libxenstore3.0 msr-tools qemu-block-extra qemu-system-common
  qemu-system-x86 qemu-utils seabios sharutils
Suggested packages:
  samba vde2 sgabios ovmf debootstrap sharutils-doc bsd-mailx | mailx
The following NEW packages will be installed:
  cpu-checker ibverbs-providers ipxe-qemu ipxe-qemu-256k-compatible-efi-roms
  libaio1 libcacard0 libfdt1 libibverbs1 libiscsi7 libnl-route-3-200 librados2
  librbd1 librdmacm1 libsd1.2debian libspice-server1 libusbredirparser1
  libxen-4.9 libxenstore3.0 msr-tools qemu-block-extra qemu-kvm
  qemu-system-common qemu-system-x86 qemu-utils seabios sharutils
0 upgraded, 26 newly installed, 0 to remove and 263 not upgraded.
Need to get 13.7 MB of archives.
After this operation, 58.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 libiscsi7 amd64 1.17
.0-1.1 [55.4 kB]

```

Fig 4.14 libvirt-bin

8. bridge-utils

```

root@openstack:~# apt-get install bridge-utils qemu-system virt-manager
Reading package lists... Done
Building dependency tree
Reading state information... Done
bridge-utils is already the newest version (1.5-1ubuntu1).
bridge-utils set to manually installed.
The following additional packages will be installed:
  gir1.2-appindicator3-0.1 gir1.2-gtk-vnc-2.0 gir1.2-libosinfo-1.0
  gir1.2-libvirt-glib-1.0 gir1.2-spiceclientglib-2.0 gir1.2-spiceclientgtk-3.0
  libgovirt-common libgovirt2 libgtk-vnc-2.0-0 libgvnc-1.0-0 libosinfo-1.0-0
  libphodav-2.0-0 libphodav-2.0-common libpython-stdlib libpython2.7
  libpython2.7-minimal libpython2.7-stdlib libspice-client-glib-2.0-8
  libspice-client-gtk-3.0-5 libusbredirhost1 libvirt-glib-1.0-0 osinfo-db
  python python-asn1crypto python-cairo python-certifi python-cffi-backend
  python-chardet python-cryptography python-dbus python-enum34 python-gi
  python-gi-cairo python-idna python-ipaddr python-ipaddress python-libvirt
  python-libxml2 python-minimal python-openssl python-pkg-resources
  python-requests python-six python-urllib3 python2.7 python2.7-minimal
  qemu-slof qemu-system-arm qemu-system-mips qemu-system-misc qemu-system-ppc
  qemu-system-s390x qemu-system-sparc spice-client-glib-usb-acl-helper
  virt-viewer virtinst
Suggested packages:
  libosinfo-l10n gstreamer1.0-plugins-bad gstreamer1.0-libav python-doc

```

Fig 4.15 bridge-utils

4.5 OPENSTACK INSTALLATION:

1. adduser stack

```
root@openstack:~# adduser stack
Adding user 'stack' ...
Adding new group 'stack' (1001) ...
Adding new user 'stack' (1001) with group 'stack' ...
Creating home directory '/home/stack' ...
Copying files from '/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for stack
Enter the new value, or press ENTER for the default
  Full Name []: Openstack
    Room Number []:
    Work Phone []:
    Home Phone []:
      Other []:
Is the information correct? [Y/n] y
root@openstack:~#
```

Fig 4.16 adduser stack

echo "stack ALL=(ALL) NOPASSWD: ALL">> /etc/sudoers

```
root@openstack:~# echo "stack ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
root@openstack:~#
```

Fig 4.17 echo statement

2. Installing required packages: Sudo , git

```
root@openstack:~# apt-get install git sudo
Reading package lists... Done
Building dependency tree
Reading state information... Done
sudo is already the newest version (1.8.21p2-3ubuntu1.2).
sudo set to manually installed.
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs
  git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 251 not upgraded.
Need to get 4,741 kB of archives.
After this operation, 34.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Fig 4.18 installing packages

3. Login in as stack user: su – stack using cd Devstack

```
cloud@openstack:~$ git clone https://opendev.org/openstack/devstack -b stable/ussuri
Cloning into 'devstack'...
remote: Enumerating objects: 46244, done.
remote: Counting objects: 100% (46244/46244), done.
remote: Compressing objects: 100% (21131/21131), done.
remote: Total 46244 (delta 32691), reused 37526 (delta 24408)
Receiving objects: 100% (46244/46244), 9.51 MiB | 1.41 MiB/s, done.
Resolving deltas: 100% (32691/32691), done.
cloud@openstack:~$
```

Fig 4.19 Login

Set the administrative password. This password is used for the admin and demo accounts set up as OpenStack users. Set the MySQL administrative password. The default here is a random hex string which is inconvenient if you need to look at the database directly for anything. Set the RabbitMQ

password. Set the service password. This is used by the OpenStack services (Nova, Glance, etc) to authenticate with Keystone

4. Run DevStack: `./stack.sh`

```
++ declare -r -g _DEVSTACK_FUNCTIONS=1
++++ dirname /home/stack/devstack/functions
+++ cd /home/stack/devstack
+++ pwd
++ FUNC_DIR=/home/stack/devstack
++ source /home/stack/devstack/functions-common
++++ set +o
++++ grep xtrace
+++ _XTRACE_FUNCTIONS_COMMON='set -o xtrace'
+++ set +o xtrace
++ source /home/stack/devstack/inc/inl-config
++++ set +o
++++ grep xtrace
+++ INC_CONF_TRACE='set -o xtrace'
+++ set +o xtrace
++ source /home/stack/devstack/inc/meta-config
++++ set +o
++++ grep xtrace
+++ _XTRACE_INC_META='set -o xtrace'
+++ set +o xtrace
```

Fig 4.20 Run DevStack

5. OpenStack Dashboard

The image shows the OpenStack Dashboard login page. At the top is the OpenStack logo, which consists of a red square with a white 'O' inside, followed by the text 'openstack.' in a sans-serif font. Below the logo is the text 'Log in'. There are two input fields: 'User Name' and 'Password'. The 'User Name' field has a cursor in it. The 'Password' field has a toggle icon on the right. At the bottom right is a blue button labeled 'Sign In'.

Fig 4.21 OpenStack Dashboard

6. Uploading Images

A sample image is placed in /root filename is: Project -> compute -> images Click Create Image Button.

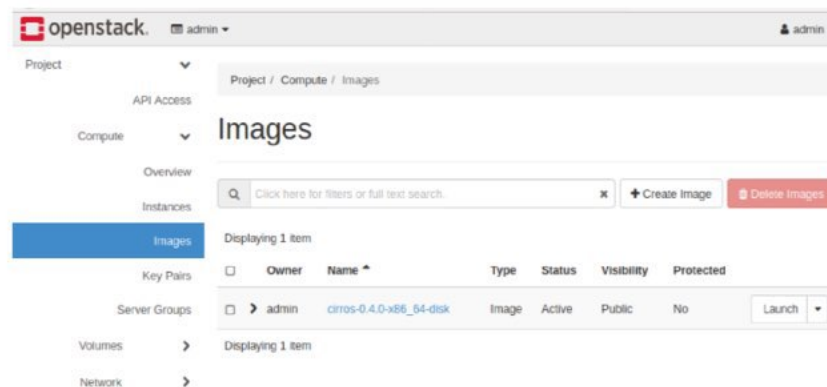
The image shows the OpenStack Dashboard 'Images' page. The top navigation bar includes the OpenStack logo, 'admin', and a user profile icon. The left sidebar has a menu with 'Project', 'API Access', 'Compute', 'Images' (selected), 'Key Pairs', 'Server Groups', 'Volumes', and 'Network'. The main content area is titled 'Images' and has a breadcrumb 'Project / Compute / Images'. Below the title is a search bar and two buttons: 'Create Image' and 'Delete Images'. A table displays the list of images. The table has columns: Owner, Name, Type, Status, Visibility, and Protected. One image is listed: 'admin' (Owner), 'cirros-0.4.0-x86_64-disk' (Name), 'Image' (Type), 'Active' (Status), 'Public' (Visibility), and 'No' (Protected). There is a 'Launch' button next to the image.

Fig 4.22 Uploading Images

Give the image name, Description, Browse the image file. From the drop down select qcow2, in Image requirements Architecture is X86_64, Minimum Disk is 1GB and Minimum RAM is 34. In Image sharing, Visibility is Public and Protected is no and click create Image.

7. Creating Network

- Project->Network-> Network Topology, Click Create Network Button. In Network tab, give the Network Name and Admin State is UP and Click next
- In Subnet tab, Give the Subnet Name, Network Address, Select the IP Version as IPV4 and give the gateway IP and click Next.
- In Subnet Details, enable DHCP and click Create.

Create Network

Network Subnet Subnet Details

Network Name
Cloud1

Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.

☒ Enable Admin State ⓘ

☐ Shared

☒ Create Subnet

Availability Zone Hints ⓘ
nova

Cancel < Back Next >

Fig 4.23 Creating Network

8. Creating Instance

- Select Boot Source -> Image and Select No in Create New Volume and Select the Image.
- Select the Network in which instance to be launched and Click Launch Instance.

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
☐	Cloud	cirros-0.4.0-x86_64-disk	192.168.100.100	m1.nano	-	Active	nova	None	Running	0 minutes	Create Snapshot

Displaying 1 item

Fig 4.24 Creating instance

9. Volume Creation

Project -> Volumes -> Volumes: Click create volume button.

Volumes

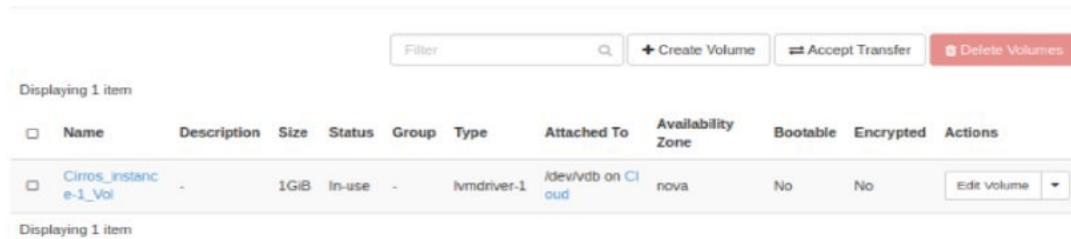


Fig 4.25 Volume Creation

- Enter the Volume Name, Size of the volume to be created and click create volume button.
- Select the volume and click on the action drop down menu and click manage attachments.
- Select the attach to instance, it display the list of instances running,
- The volume is attached to cloud instance in the device /dev/vdb

10. Mounting Volume in Instance

- Open virt-manager in terminal and click on the instance
- Login as cirros user, the give sudo -i.

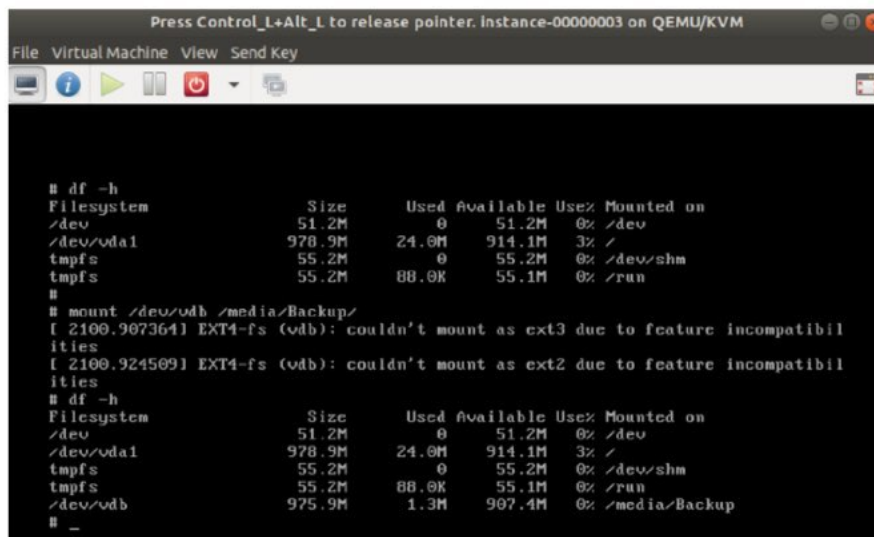


Fig:4.26 Mounting volume

CHAPTER-5

RESULTS AND DISCUSSION

5.1 Create tenant in OpenStack and launch instances:

5.1 Create project tenant and assign tenant user.

Login as admin to Horizon (dashboard) on controller node using web browser. Identity -> Projects -> Create Project, type name (i.e.abc) and description.

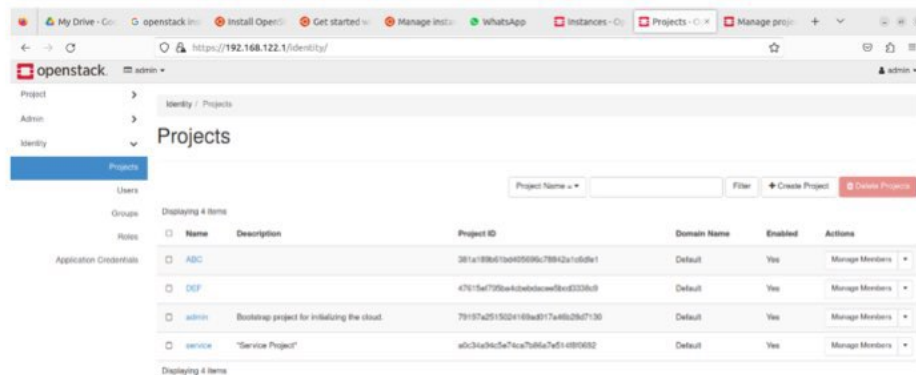


Fig 5.1 create project

5.2 Create new user and assign him to the newly created project tenant.

Identity -> Users -> Create User: assign user (i.e. abc) as _member_ of the newly created project tenant (abc).

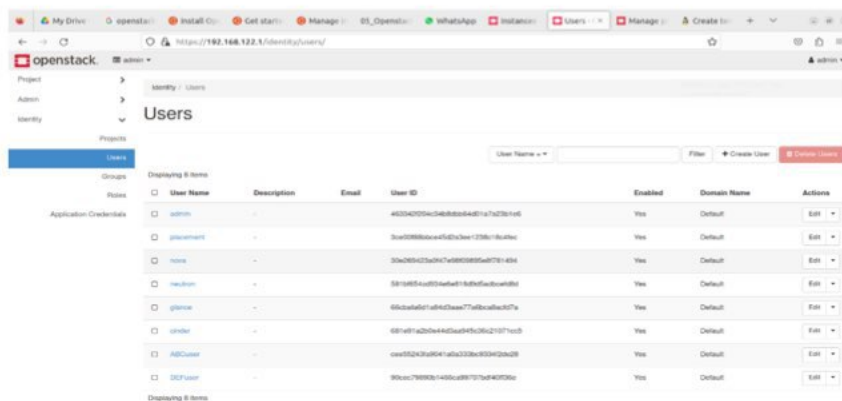


Fig 5.2 Create new user

5.3 Create networks and routers.

Login to dashboard as admin. First configure the private network, in the dashboard menu go to: Project -> Networks -> Create Network.

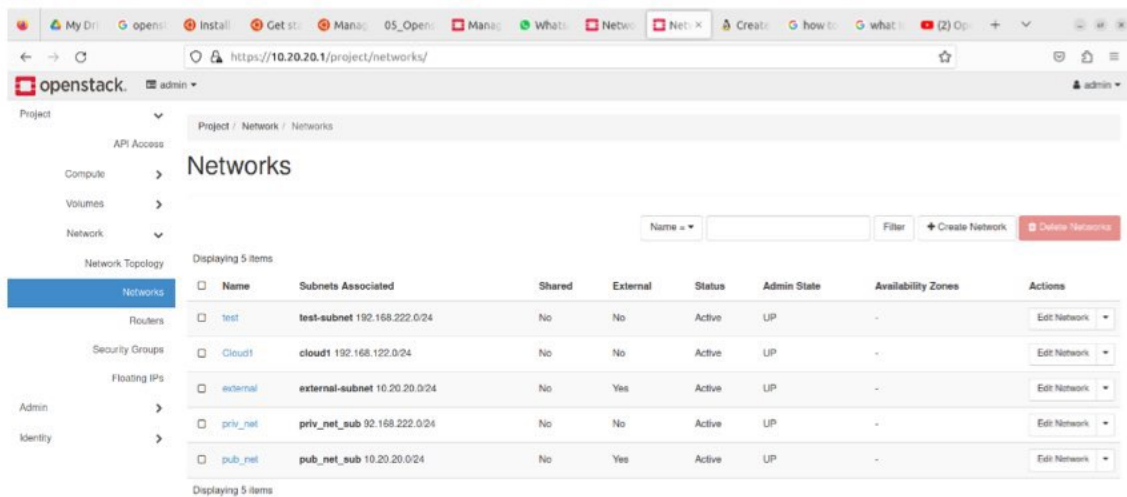


Fig 5.3 Create network

Log out as admin and log in as abc, Project -> Network -> Routers -> Create Router.

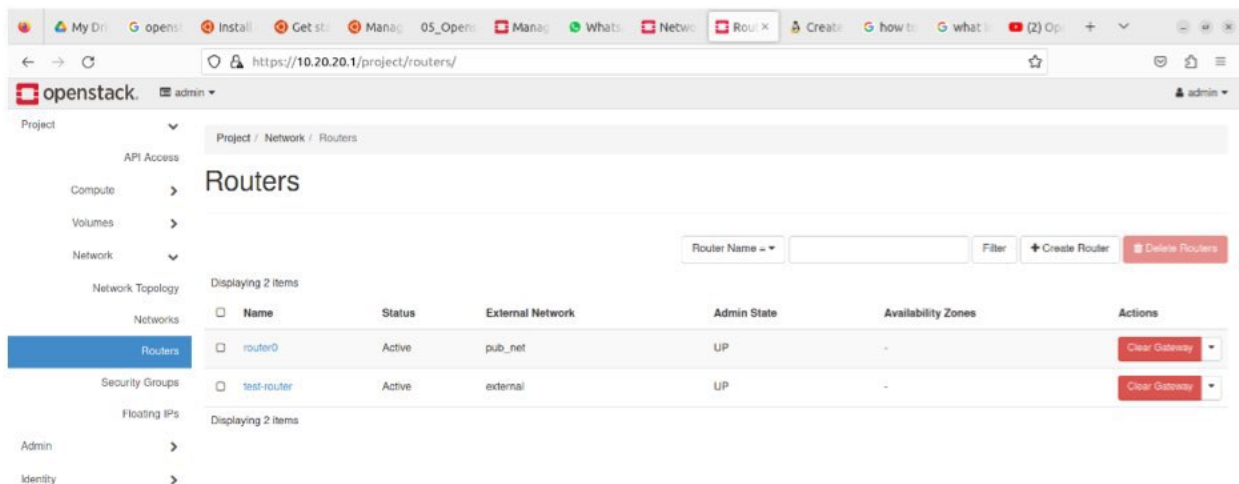


Fig 5.4 Create router

Let's check now the network connections in our OpenStack cloud. Project -> Network -> Network Topology.

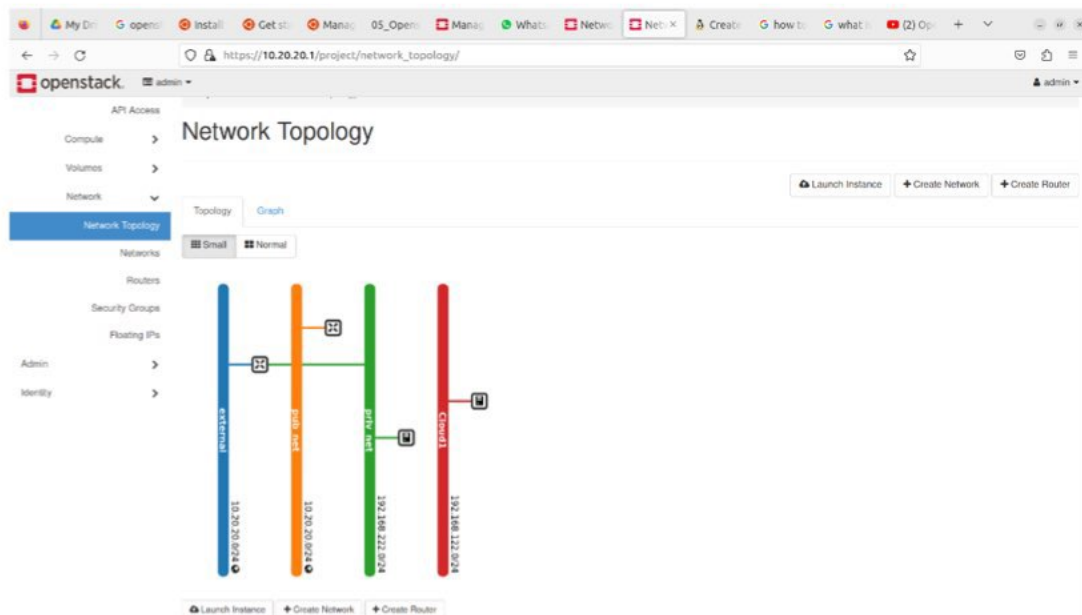


Fig 5.5 Network topology

5.4 Create Security Groups

Security Groups control network access to / from instances inside the tenant. Let's create new security group and add security rules to the group. Project -> Compute -> Access & Security -> Create Security Group.

Name	Security Group ID	Description	Actions
default	cb5d8d17-c2a3-4f12-9fe8-64e4a368863a	Default security group	Manage Rules
project_sec_group	c3a6c5b-e99f-4caa-e549-3ae542047ed	Default security group	Manage Rules

Fig 5.6 Create security group

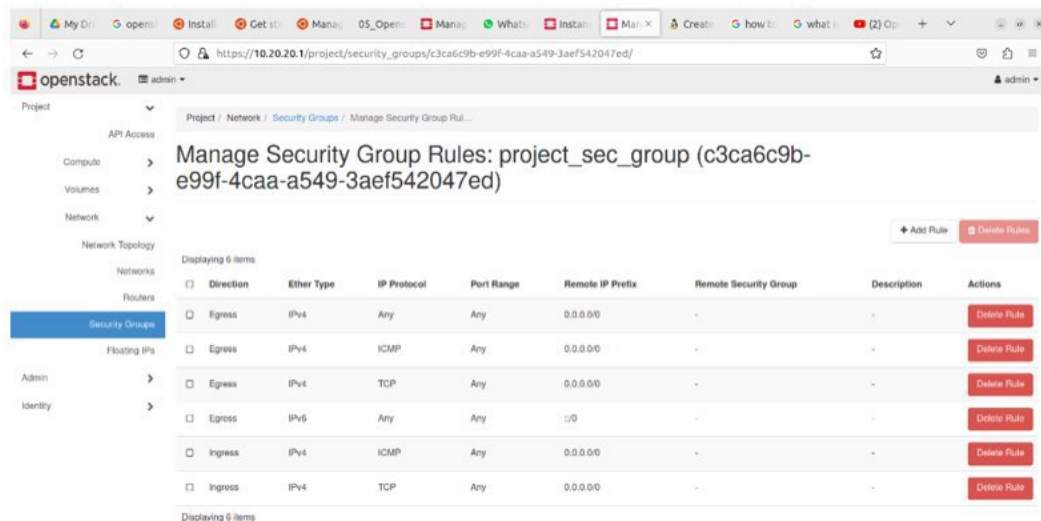


Fig 5.7 Manage rules

5.5 Assign Floating IPs to the Project Tenant

Floating IPs allow access to Instances from public network and WAN. To assign new IPs to the Project tenant go to: Project -> Compute -> Access & Security -> Floating IPs -> Allocate IP to The Project.

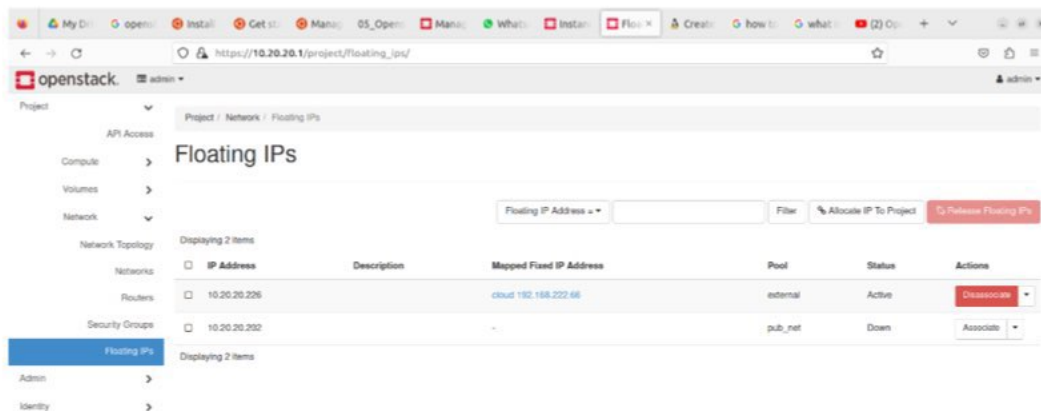


Fig 5.8 Floating Ips

5.6 Create Instances

Now when we have all the need components, we can create Instances (virtual machines running inside cloud). Let's create 2 Instances, go to: Project -> Compute -> Instances -> Launch Instance.

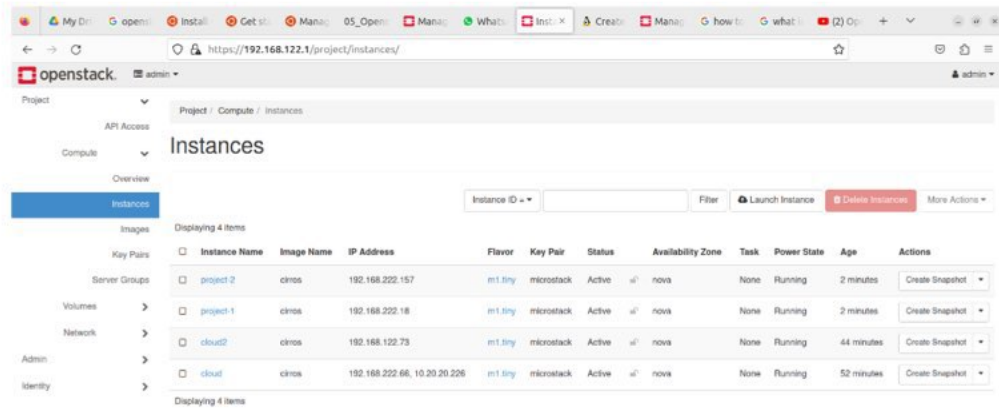
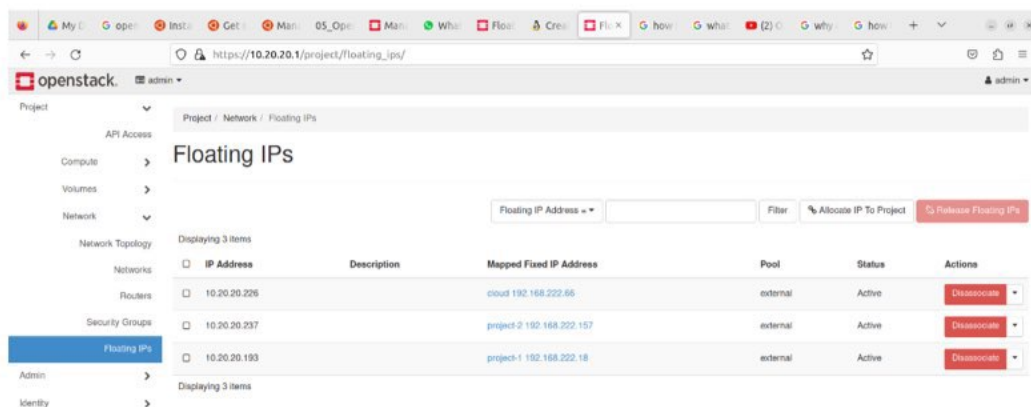


Fig 5.9 Create instance

5.7 Instance internal / external connectivity testing

Test connectivity of instances from public network, you can connect from controller or compute node to instance via ssh.



https://10.20.20.1/project/floating_ips/

Fig 5.10 external connectivity

CHAPTER-6

CONCLUSION

Cloud computing is recently new technological development that has the potential to have a great impact on the world. It has many benefits that it provides to its users and businesses. For example, some of the benefits that it provides to businesses, is that it reduces operating cost by spending less on maintenance and software upgrades and focus more on the businesses itself. But there are other challenges the cloud computing must overcome. People are very skeptical about whether their data is secure and private. There are no standards or regulations worldwide provided data through cloud computing. Europe has data protection laws but the US, being one of the most technological advance nation, does not have any data protection laws. Users also worry about who can disclose their data and have ownership of their data. But once, there are standards and regulation worldwide, cloud computing will revolutionize the future.

OpenStack is a cloud OS that is used to control the large pools of computing, storage, and networking resources within a data centre. OpenStack is an open-source and free software platform. This is essentially used and implemented as an IaaS for cloud computing. OpenStack allows users to install virtual machines that take care of different tasks for managing a cloud environment on the go. OpenStack cloud computing makes horizontal scaling easy, which means functions that have benefit from running in parallel can serve more users by spinning up Occurrence. Based on the report by market research future, OpenStack continues to grow at the compound annual growth rate (CAGR) of 29%. Its global business, including solutions and services, is expected to reach \$8M in 2023.

REFERENCES

1. <https://www.pcsuggest.com/dual-boot-windows-10-and-ubuntu-uefi/>
2. <https://fspcloud.boschn.in:8443/lab1.html>
3. <https://tuxfixer.com/create-project-tenant-in-openstack-and-launch-instances/>
4. <https://www.openstack.org/>
5. <https://www.redhat.com/en/topics/openstack>