

Shirish Shetty 2103164 C32

Thadomal Shahani Engineering College

Bandra (W.), Mumbai- 400 050.

CERTIFICATE

Certify that Mr./Miss Shirish Shetty

of COMPS Department, Semester VI with

Roll No. 2103164 has completed a course of the necessary experiments in the subject CCL under my

supervision in the **Thadomal Shahani Engineering College**

Laboratory in the year 2023 - 2024

Teacher In-Charge

Head of the Department

Date 28/3/24

Principal

CONTENTS

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1.	Introduction and overview of cloud computing	1 - 8	19/01/24	
2.)	To study and implement Hosted Virtualization using Virtual Box & KVM.	9 - 21	29/01/24	
3.)	To study and implement Bare metal Virtualization using Xen	22 - 25	02/02/24	
4.)	To study and implement Infrastructure as Service using AWS (EC2)	26 - 37	09/02/24	SP
5.)	To study and implement platform as Service using AWS Elastic Beanstalk	38 - 47	23/2/24	23/2/24
6.)	To study and implement storage as a service using Own /AWS S3.	48 - 62	26/1/24	
7.)	To study and implement Identity Access Management practices on AWS	63 - 80	01/03/24	
8.)	To study and implement Database as a Service on AWS RDS	81 - 100	11/03/24	
9.)	To study and implement Containerization using Docker	101 - 108	15/03/24	
10.)	To study and implement container orchestration using Kubernetes	109 - 118	18/03/24	
11.)	Assignment - 1	119 - 122		22/2/24
12.)	Assignment - 2	123 - 124		
13.	mini project - 1	125 - 131	8/4/24	

Experiment No - 1

Cloud Computing Lab.

- Definition of cloud computing

The term cloud refers to a network or the internet. It is technology that uses remote server on the internet to store, manage and access data online rather than local drives. The data can be anything such as files, images, documents, audio, video, and more.

- Characteristics of cloud computing

- (i) On-Demand self-service.

- self service means that the cloud user can acquire the service independently, without going through an IT department, call center or middle man.

To support self-service

- The cloud provider must have an automated interface, such as web portal or mobile app
- The user should be able to access the interface at any time.
- The user should be able to cancel cloud service at any time.

- (ii) Broad Network Access.

- The cloud service must be broadly available

over communication network. Users should be able to access from any location & internet-enabled device.

3.) Resource Pooling

multiple customers share the cloud service resources in a multi-tenancy model. This model raises privacy and security concerns, so users must protect their cloud data and assets by taking necessary security precautions.

4.) Rapid Elasticity

Elasticity refers to flexibility of cloud service to scale up or down automatically to meet the user's needs. That allows to access the right level and kind of resources, including processing power, memory, network bandwidth and storage to accommodate the user's varying workloads.

5.) measured service

A measured cloud service provides a metering capability that underpins the provider's pay-as-you-go pricing model. This model provides users with greater transparency & control over their cloud costs.

- NIST cloud computing model

⇒ National Institute of Standards and Technology states that cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.

- Different model of CCL (Service & Deployment)

⇒ Service models :-

- Software as a Service (SaaS) :- The capability provided to the consumer is use the provider application running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as web browser or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating system, storage, or even individual application capabilities, with possible exception of limited user specific application configuration settings.
- Platform as a Service (PaaS) :- The capability provided to consumer - created or acquired application created using programming languages, libraries, services and tool supported by provider. The consumer

Shirish Shetty 2103164 C32

does not manage or control the underlying cloud infrastructure including network, server, operating system & storage, but has control over deployed application and possibly configuration settings for application hosting environment.

Infrastructure as a Service (IaaS) : - The capability provided to consumer is to provision processing, storage, network, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software which can include operating system & applications. The consumer does not manage or control the underlying cloud infrastructure but has a control over operating system, storage & deployed applications & possibly limited control of selected networking components (e.g host firewalls)

⇒ **Deployment Models :-**

Private cloud : - The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers. It may be owned, managed & operated by organization, a third party, or some combination of them, and it may exist on or off premises.

Community cloud : - The cloud infrastructure is provisioned for exclusive use by a specific community of consumer from organization that have shared concerns. It may be owned, managed & operated by one or more of organization in the community, a third party, or some combination of them, and it may exist on / off premises.

Hybrid cloud :- The cloud infrastructure is a composition of two or more distinct cloud infrastructures that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability.

- Explanation of architecture of cloud computing with suitable diagram

The cloud architecture is divided into 2 parts i.e

- 1.) Frontend
- 2.) Backend

The basic architecture of cloud computing is combination of SOA and EDA. Client infrastructure, application, service, runtime, cloud, storage, infrastructure, management & security all these are components of cloud computing architecture.

- 1.) Frontend :-

Frontend of cloud architecture refers to client side of cloud computing system. Means it contains all user interfaces and applications which are used by the client to access the cloud.

computing services. For eg, use of web browser the cloud platform.

- Client infrastructure - Client infrastructure is a part of frontend component. It contains application & user interfaces which are required to access cloud platforms
- Basically GUI to interact with cloud.

2.) Backend :-

Backend refers cloud itself which is used by service provider. It contains resources well manages the resources & provides security mechanism. Along with this, it includes huge storage, virtual application, virtual machines etc.

1.) Application -

Application in backend refers to software or platform to which client accesses. Means it provides a service in backend as per the client requirement.

2.) Service :-

Service in backend refers to major three types of cloud based services like SaaS, PaaS and IaaS. Also manages which type of services the user accesses.

3) Runtime Cloud :-

Runtime cloud in backend provides the execution & runtime platform / environment to VM.

4) Storage :-

Storage in backend provides flexible & scalable storage service and management of stored data.

5) Infrastructure :-

Cloud infrastructure in backend refers to hardware & software component of cloud like it includes servers, storage, network devices, virtualization software etc.

6) Management -

Management in backend refers to management of backend component like application service, runtime cloud, storage infrastructure and other security mechanism.

7) Security :

Security in backend refers to implementation of different security mechanism in backend of secure cloud resources, systems, files and infrastructure to end-users.

8.) Internet :-

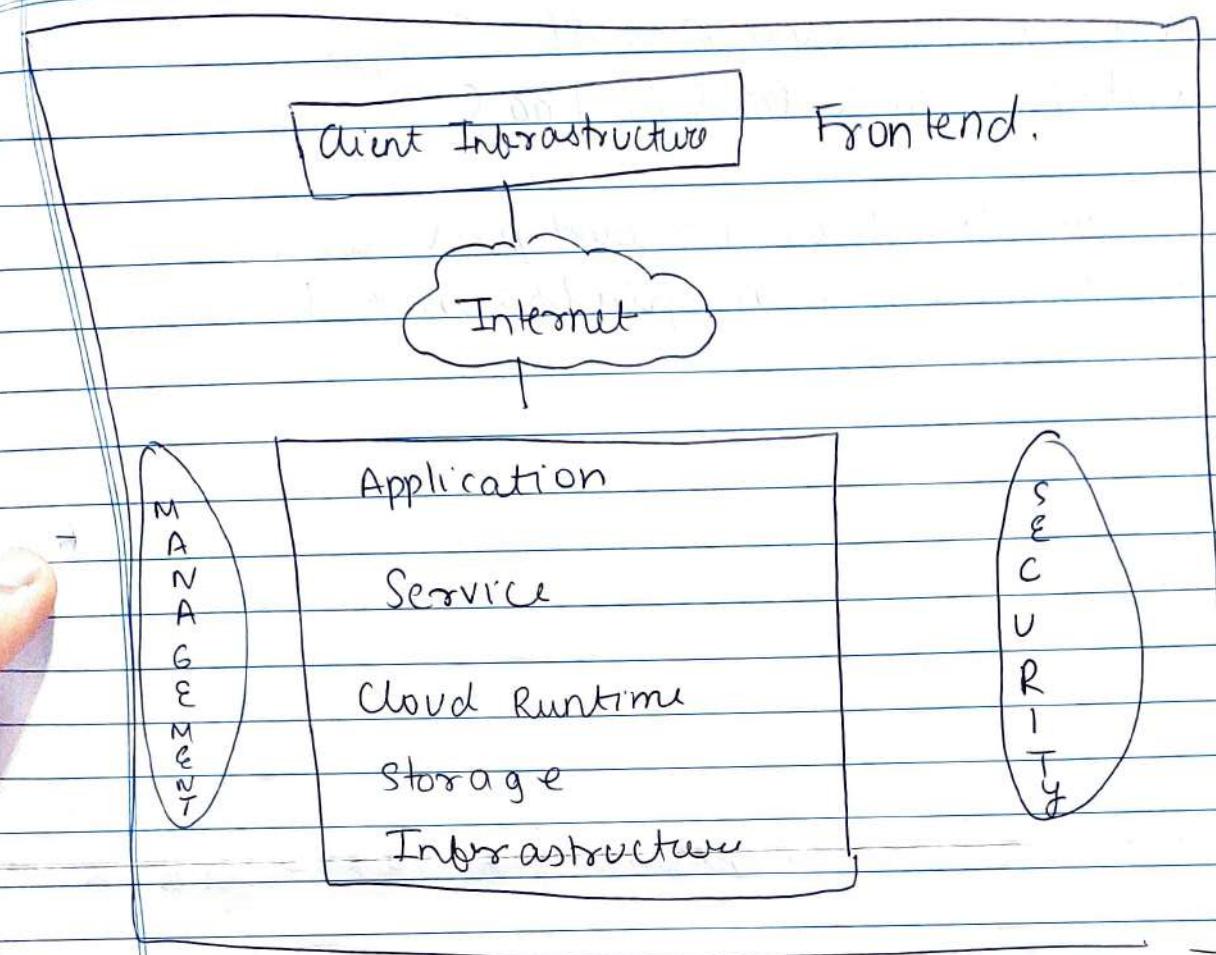
Connection act as medium or a bridge between frontend or backend & establishes interaction & communication between frontend & backend.

9.) Database :-

Database in backend refers to provide DB for storing/storing DS such as SQL, NoSQL databases. services include Amazon RDS, Google Cloud SQL.

10.) Networking - Networking in backend services that provides networking infrastructure for application in cloud, such as load balancing, DNS & virtual private networks.

11.) Analytics:- Analytics in backend services that provides analytic capabilities for data in cloud such as warehousing, business intelligence & machine learning.



6.) Benefits and limitation of cloud computing

Benefits

- 1.) Cost saving through, pay as you go model, reduced need for on premises hardware
- 2.) To increase scalability and flexibility to meet changing business model
- 3.) Enhanced collaborations and productivity.

- 4.) Service in backend are of 3 type of cloud based like SaaS, PaaS & IaaS
- 5.) Runtime cloud in backend provides the execution & runtime platform to VM.

Limitations :-

- 1.) Limited control

Cloud infra is completely owned & monitored by service provider.

- 2.) Security :-

Although cloud provides use best security standard to store information, but there is still risk.

- 3.) Cloud computing stored data in cloud.

so we need internet connection to access it.

(A)
26/2/24

Experiment NO - 2

Aim :- To study and implement Hosted virtualization using virtualBox & KVM

Theory :-

- Virtualization in cloud computing :-
- Virtualization is creation of virtual rather than actual version of something, such as a server, a desktop, a storage device, an operating system or network resources.
- In other words, virtualization is a technique, which allow sharing of single physical instance of a resource or an application among multiple customers and organization.
- It does this by assigning a logical name to physical storage & providing a pointer to that physical resource when demanded.
- A virtual machine provide an environment that is logically separated from underlying hardware.
- The machine on which the virtual machine is going to create is known as Host Machine and that virtual machine is referred as a Guest Machine.

- Virtualization play important role in CCL, normally in cloud computing, user share the data present in clouds like application etc, but actually with the help of virtualization user share the infrastructure.

° Benefits of virtualization :-

- Reduced cost of IT infrastructure : - with data being gathered and maintained on virtual server instead of one or more physical server. One lack of rigorous electricity use can help lower overhead bills.

- User can access applications and servers remotely, with users able to access apps and data center from anywhere on any device, the scalability for an organization can increase exponentially.

- Increase Scalability : - Virtualization in cloud allows for great vertical scaling, which involves growing & reallocating features such as memory, bandwidth and CPU cores associated with your network.

- Flexible data transfers : - Data can be transferred to and retrieved from virtual server at any moment. Both users & cloud vendors will not need to waste their time trying to locate different devices where data may be

- what is Hypervisor? Give examples of Hypervisors
 - A hypervisor is also known as virtual machine monitor or VMM
 - The hypervisor is a piece of software that allows us to build and run virtual machines which are abbreviated as VMs.
 - A hypervisor allows a single host computer to support multiple virtual machines (VMs) by sharing resources including memory and processing.
 - Hypervisors allow the use of more of system's available resource and provide greater versatility because the guest VMs are independent of host hardware which is one of major benefits of Hypervisors.
 - In other words this implies that they can be quickly switch between servers.
 - Eg :- Type 1 hypervisors include oracle VM server for XEN, SPARC, Oracle VM Server for X86 and ~~Microsoft~~ Hyper - V
- * Example types of Hypervisors.

Type 1 Hypervisor:-

a. A Type 1 hypervisor operates directly on host hardware to monitor the hardware and guest VMs, and is referred to as bare metal.

Comparison between KVM and VirtualBox

KVM and VM are both virtualization solutions, but they have some key differences in terms of their architecture, usage & features.

1.) Type of hypervisor

-KVM:- It type-1 hypervisor, which means it runs directly on hardware without the need of host OS. KVM is integrated into Linux kernel, leveraging the Linux kernel for resource management & hardware support.

-VirtualBox:- It is type-2 hypervisor, which means it runs on top of host OS. VirtualBox needs a host OS like windows, linux, Mac OS to provide virtualization services.

2.) Platform support

-KVM: Primarily designed for Linux environment. It is tightly integrated with Linux kernel & work best on Linux-based system.

- b.) Typically, they do not require the installation of software ahead of time instead, you can install directly on the hardware.
- c.) Type I hypervisors are more complex & have less hardware requirement to run adequately because of this it mostly chosen by IT operation & data center computing.
 - a.) Eg:- Oracle VM series of XEN (PARC), oracle VM server for x86, microsoft & VMWare) ESX/ESXi

- Type II Hypervisor

- a.) It is also called hosted hypervisor because it installed of existing OS and they are not more capable of running more virtual tasks.
- b.) People use it for basic development, testing and simulation.
- c.) If a security flaw found inside the host OS, it can potentially compromise all running VMs. This is why Type 2 hypervisor cannot be used for data computing, and they are designed for end-user system where securing is user of concern.

3) Performance :-

- KVM: generally considered to have better performance, especially for enterprise level visualization scenarios, it benefit from being part of Linux kernel & has direct access to hardware resources.
- VirtualBox: - May have slightly lower performance compared to KVM due to additional layer provided by the host operating system.

4) Licensing :-

- KVM: - Open source and part of Linux kernel. It is distributed under the GNU GPL.
- VirtualBox: Also open source it is released under the GNU GPL the open source edition & personal use and evaluation license (PUEL) for full featured version.

(A)

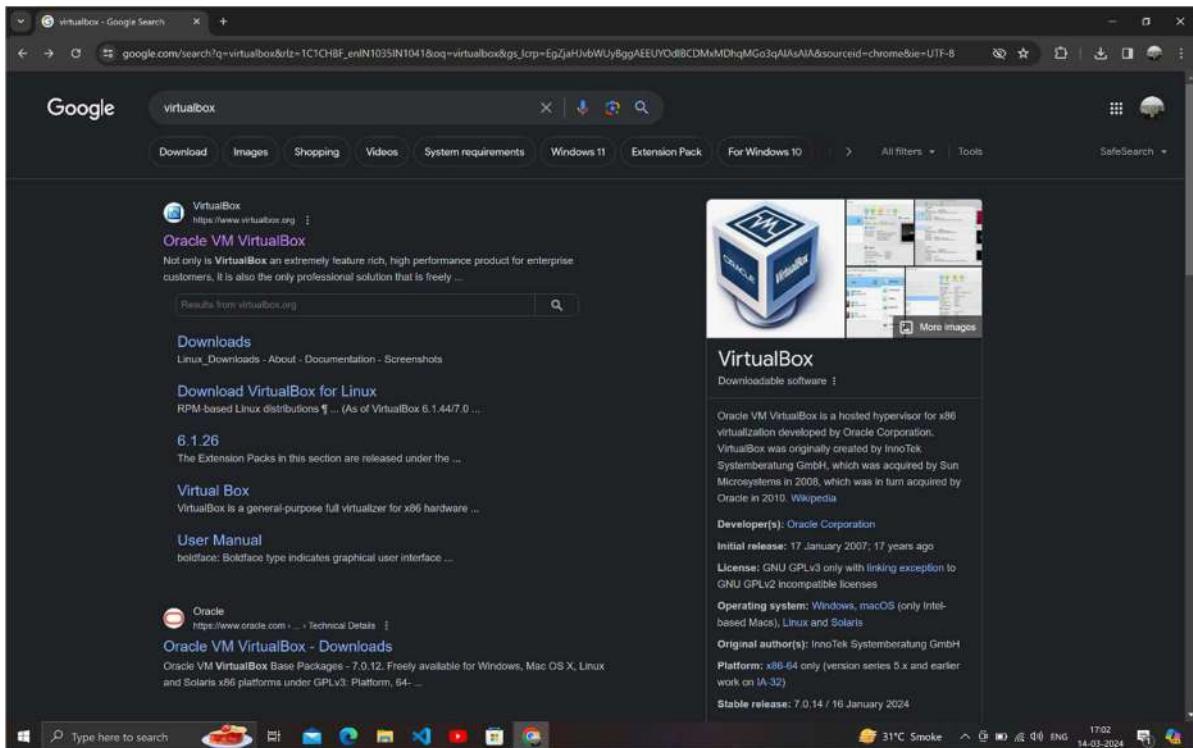
SF
26/12/24

EXPERIMENT NO: 2

AIM: To study and implement Hosted Virtualization using VirtualBox and KVM.

Steps for installation and working of VirtualBox is given below:

1. Search for VirtualBox Oracle in the Browser of your choice.



2. Click on the link by Oracle and download VirtualBox 7.0.

Welcome to VirtualBox.org!

VirtualBox is a powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 3. See "About VirtualBox" for an introduction.

Presently, VirtualBox runs on Windows, Linux, macOS, and Solaris hosts and supports a large number of guest operating systems including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Vista, 7, 8, Windows 10 and Windows 11), DOS/Windows 3.x, Linux (2.4, 2.6, 3.x, 4.x, 5.x and 6.x), Solaris and OpenSolaris, OS/2, OpenBSD, NetBSD and FreeBSD.

VirtualBox is being actively developed with frequent releases and has an ever growing list of features, supported guest operating systems and platforms it runs on. VirtualBox is a community effort backed by a dedicated company: everyone is encouraged to contribute while Oracle ensures the product always meets professional quality criteria.

Download VirtualBox 7.0

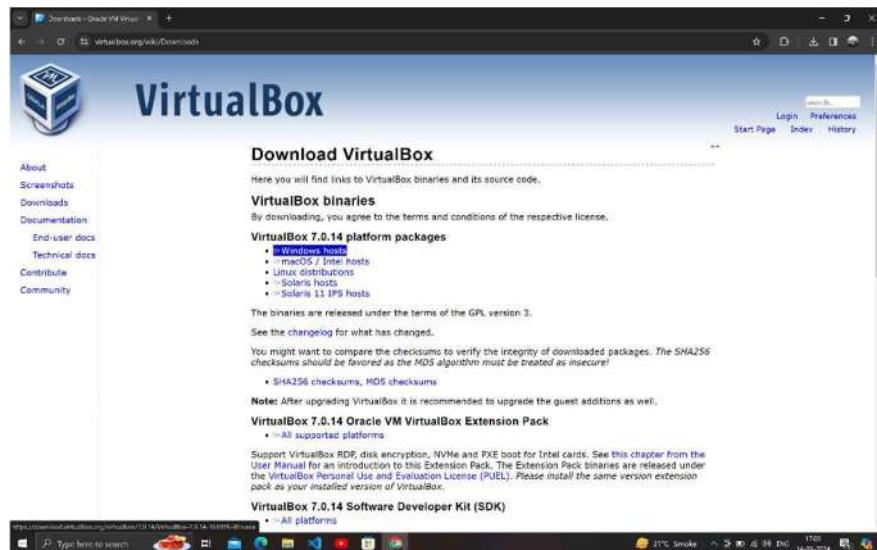
Hot picks:

- Pre-built virtual machines for developers at [Oracle Tech Network](#)
- Hyperbox Open-source Virtual Infrastructure Manager [project site](#)

News Flash

- Important February 8th, 2024**
We're hiring!
Looking for a new challenge? We're hiring a VirtualBox Principal Software Developer (Germany, UK, US, Romania, and other locations).
- New January 16th, 2024**
VirtualBox 7.0.14 released!
Oracle today released a 7.0 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- New January 16th, 2024**
VirtualBox 6.1.50 released!
Oracle today released a 6.1 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- New October 17th, 2023**
VirtualBox 7.0.12 released!
Oracle today released a 7.0 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- New October 17th, 2023**
VirtualBox 6.1.52 released!

- Under VirtualBox platform packages, select the OS of your choice. (Here Windows Hosts).



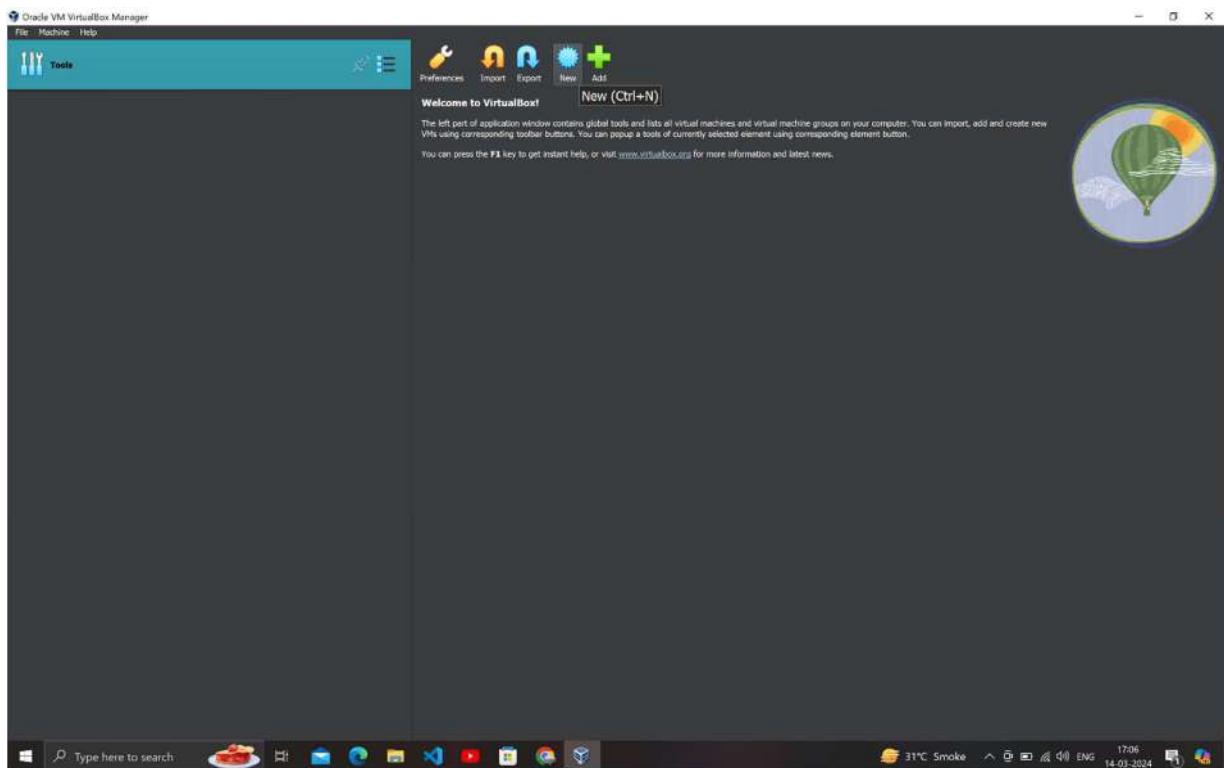
- Open the installation setup from downloads. Install VirtualBox with default settings and click finish to complete the installation.



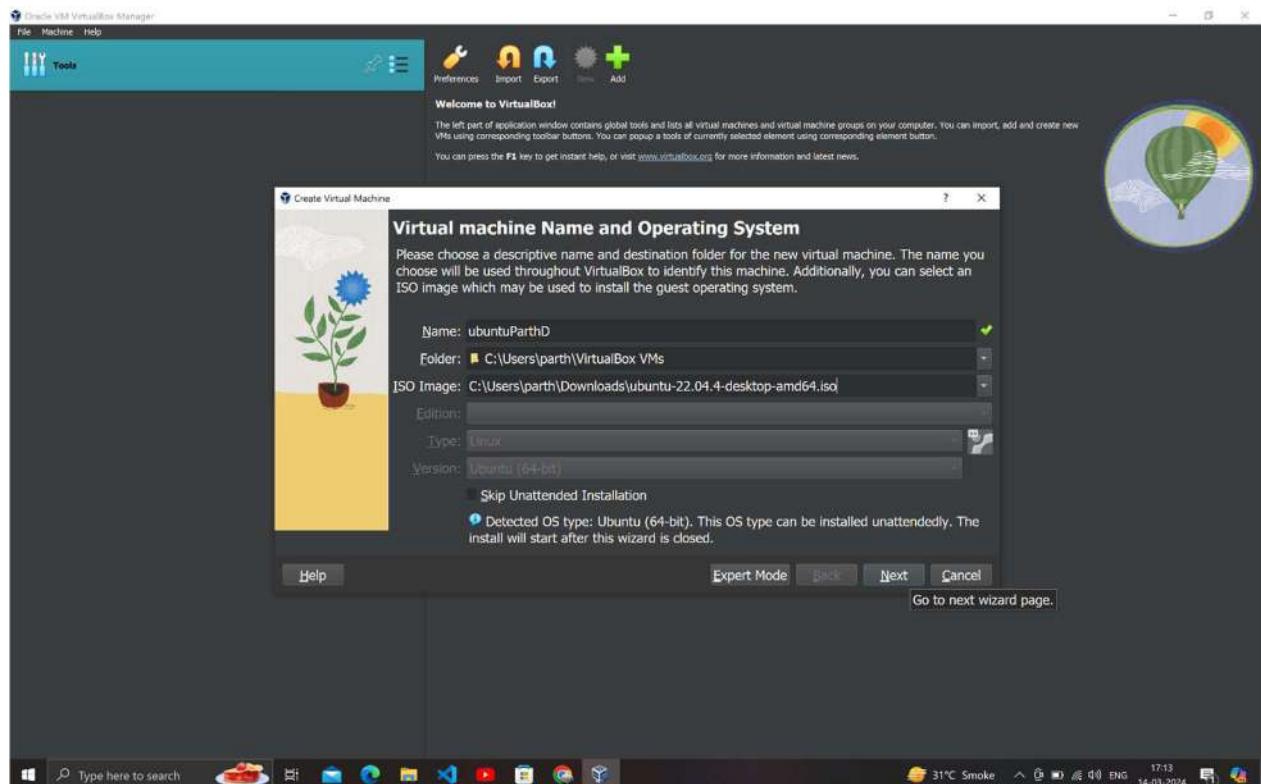
- Before launching VirtualBox, we must download the ISO file for the OS we want to use on the Virtual Machine i.e. on the Host OS(Windows). We will download Ubuntu as it is lightweight, small and easily downloadable OS. Go to the official site of Ubuntu and download the latest Linux Ubuntu LTS OS.



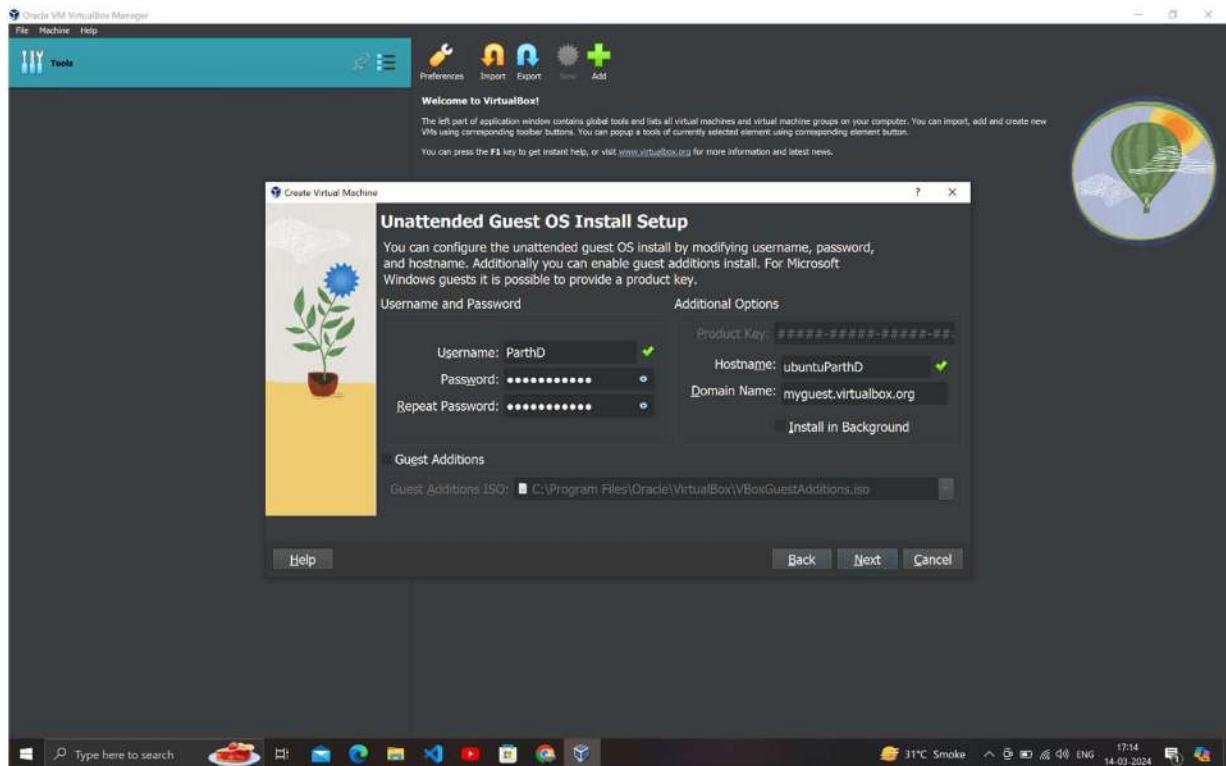
6. Open the VirtualBox. An interface will appear as given below. Click on NEW to start a new Virtual OS.



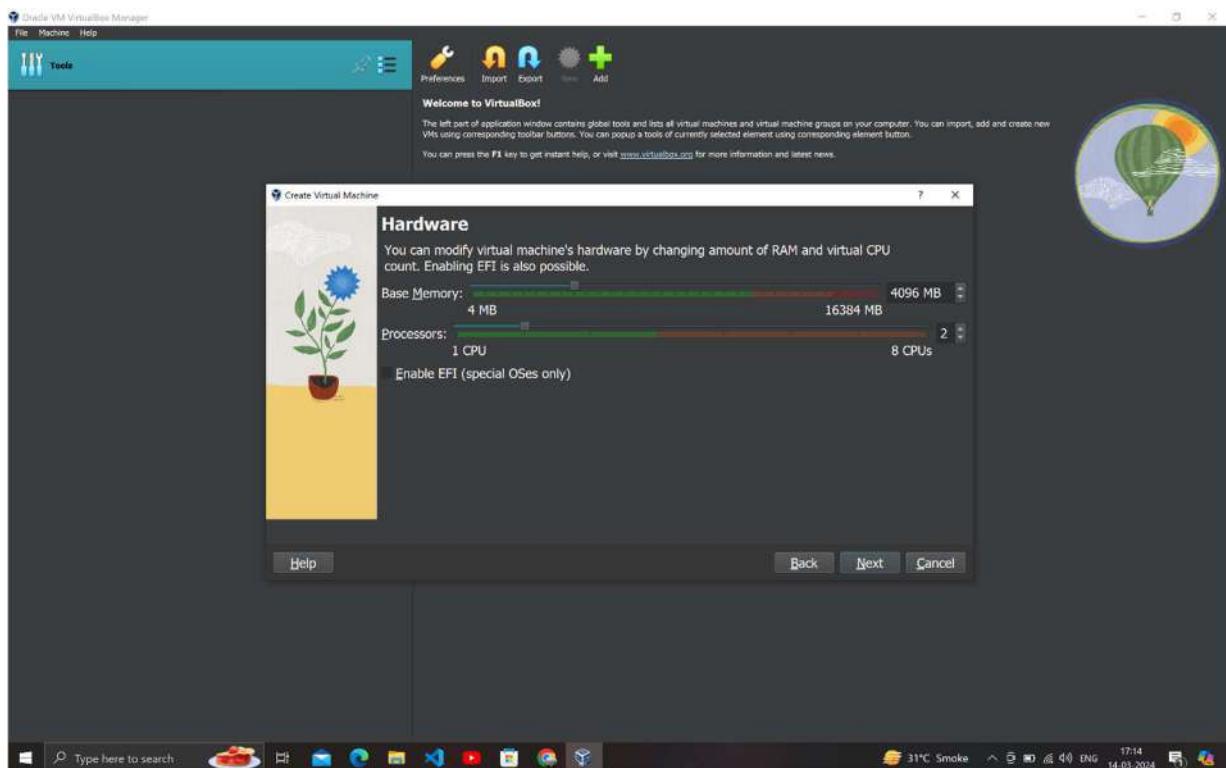
7. Enter the Virtual Machine name, the path to the folder of VMbox and under ISO Image section select the Ubuntu ISO which was downloaded in the earlier steps. Further select next.



- Fill the username and password for the Virtual OS. Remember the password as it will be used for login in further steps.

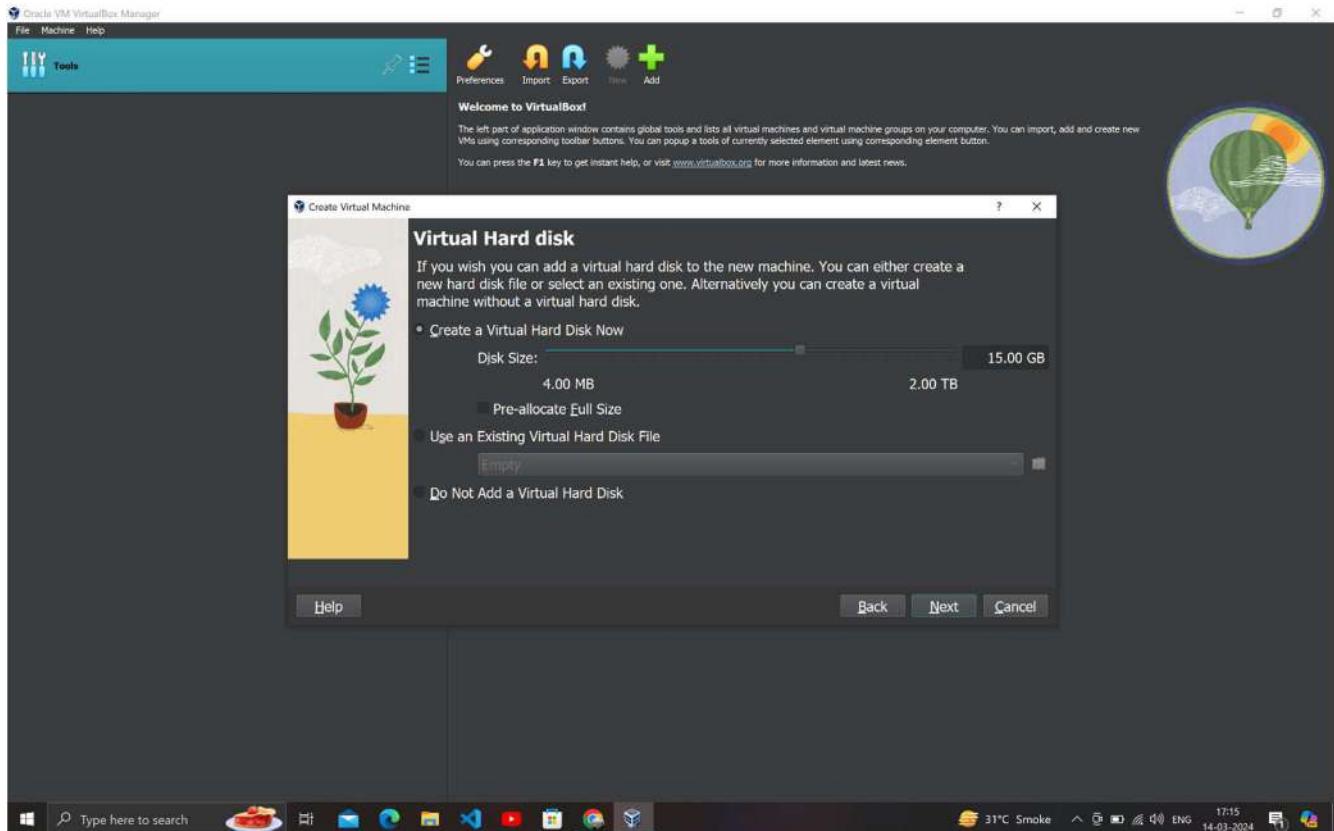


- Set the base memory/RAM to be utilized for hosting the OS. Here, I have set it to 4GB (4096MB). Also set the number of processors to be used for the same. I have assigned to processors for the same.

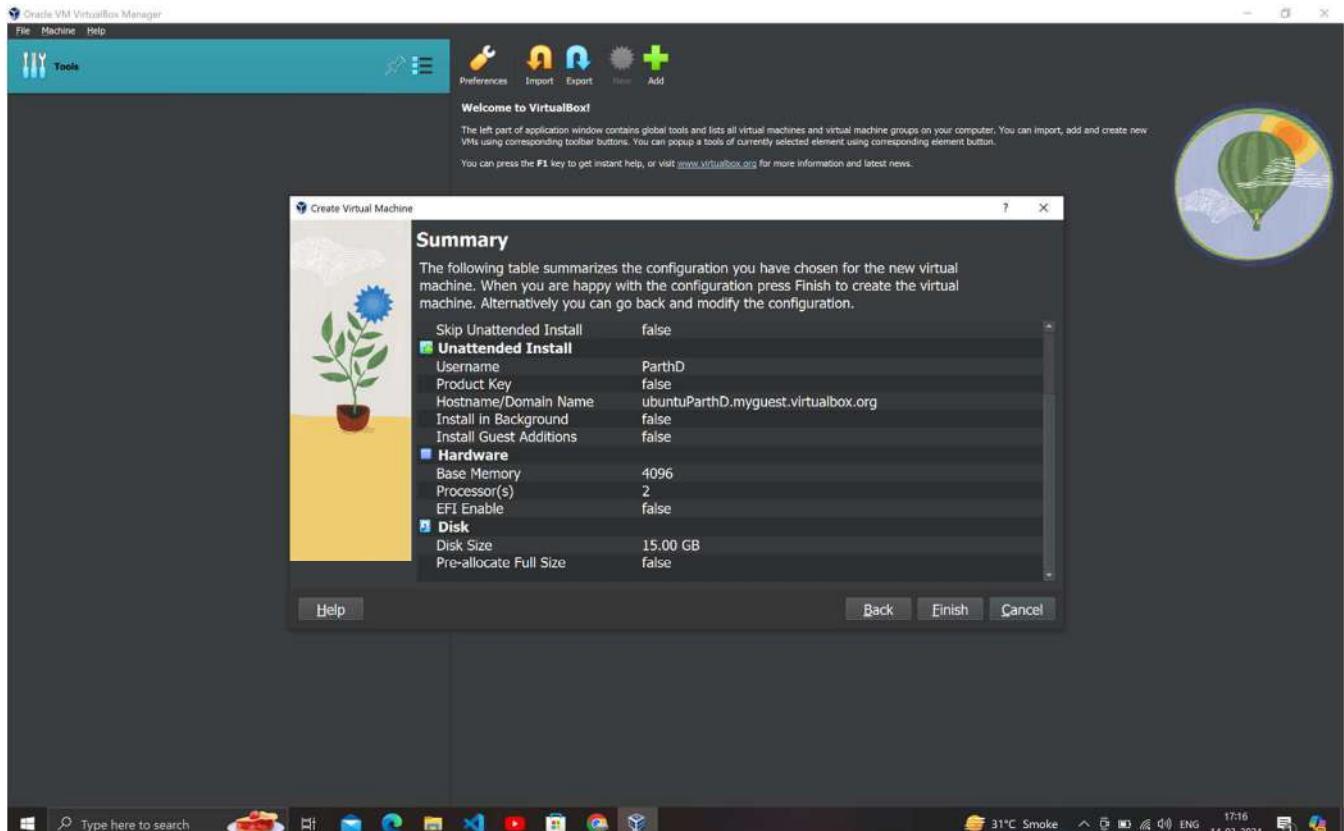


Shirish Shetty 2103164 C32

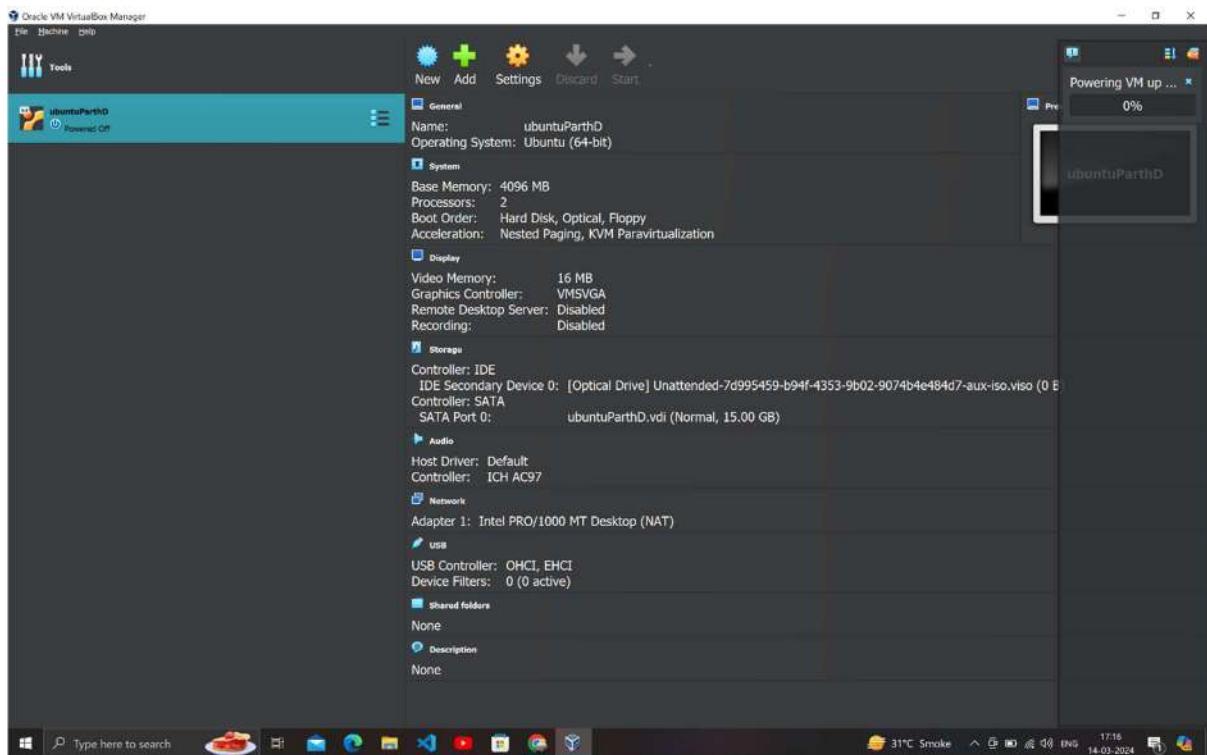
10. Assign the storage or Virtual Hard Disk for your Virtual Machine. I have set it to 15 GB.



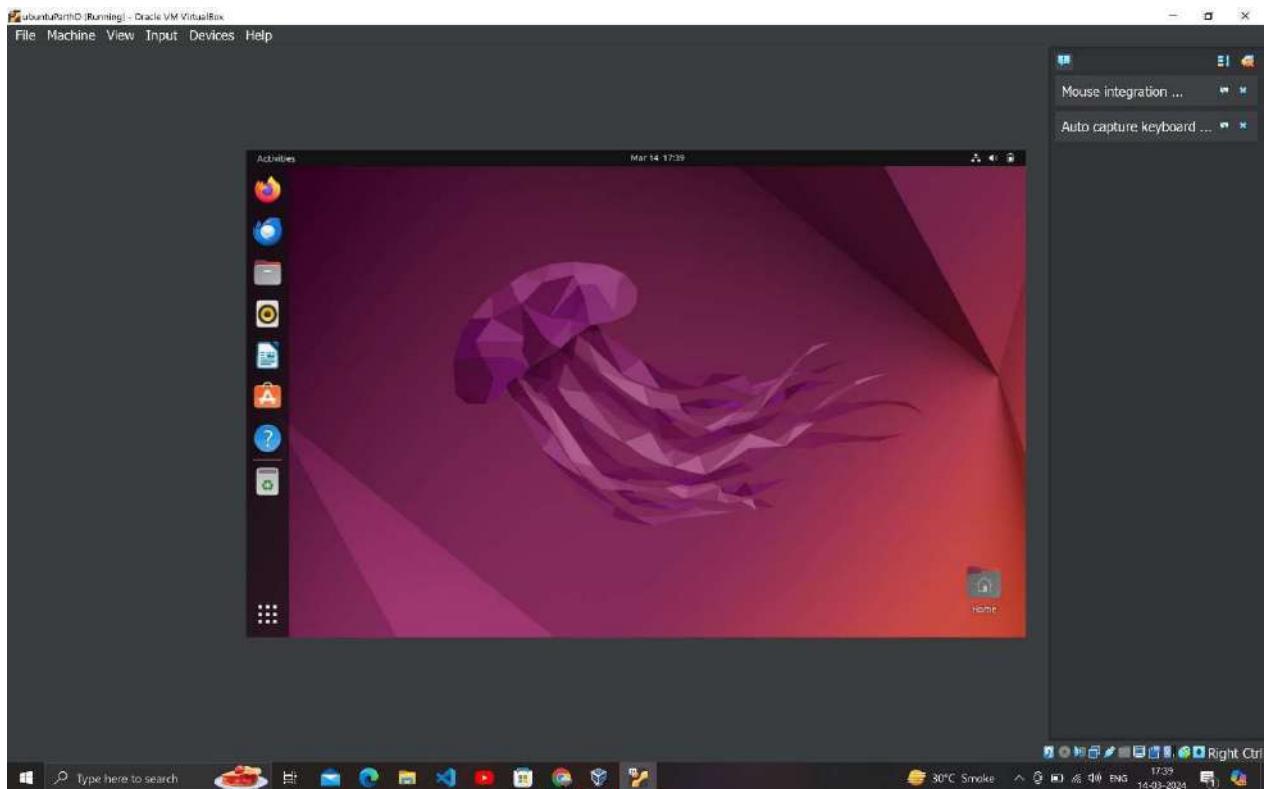
11. Analyse the summary provided. It contains all the information about the Virtual Machine such as RAM utilized, Storage assigned, Processors undertaken and used etc. Make changes if needed. Click finish to launch the Virtual Machine.



12. Wait for the virtual machine to power up. A small pop-up window will appear. Maximize the window in-order to get the perfect view for our newly launched Ubuntu OS.



13. Enter the correct username and password as set earlier and wait for the completion of installation. After the installation is complete, enter the password and login to get into the desktop of Ubuntu.



14. Open the Linux Terminal aka the Command Prompt. Write some basic Linux commands to check if the OS is properly working. Use the following commands:
- cd Desktop (To get into the Desktop)
 - ls -l (To list all the directories present in the Desktop. It also shows the permissions for read and write.
 - mkdir (To create a new Directory in desktop).
 - Touch filename (to create a new file)
 - Cat > filename (to create a new file and write in it).
 - Cat filename (to view the contents in the file).
 - Rmdir (to remove/delete the folder)
 - ps aux (to view all the processes working in the pc),

```

Activities Terminal Mar 14 17:45
parthd@ubuntuParthD: ~/Desktop/NewTestFolder
parthd@ubuntuParthD: ~ ls -l
total 36
drwxr-xr-x 2 parthd parthd 4096 Mar 14 17:38 Desktop
drwxr-xr-x 2 parthd parthd 4096 Mar 14 17:38 Documents
drwxr-xr-x 2 parthd parthd 4096 Mar 14 17:38 Downloads
drwxr-xr-x 2 parthd parthd 4096 Mar 14 17:38 Music
drwxr-xr-x 2 parthd parthd 4096 Mar 14 17:38 Pictures
drwxr-xr-x 2 parthd parthd 4096 Mar 14 17:38 Public
drwx----- 3 parthd parthd 4096 Mar 14 17:38 .snap
drwxr-xr-x 2 parthd parthd 4096 Mar 14 17:38 Templates
drwxr-xr-x 2 parthd parthd 4096 Mar 14 17:38 Videos
parthd@ubuntuParthD: ~
parthd@ubuntuParthD: ~ cd Desktop
parthd@ubuntuParthD: ~/Desktop$ mkdir NewTestFolder
parthd@ubuntuParthD: ~/Desktop$ ls -l
total 4
drwxrwxr-x 2 parthd parthd 4096 Mar 14 17:41 NewTestFolder
parthd@ubuntuParthD: ~/Desktop$ cd NewTestFolder
parthd@ubuntuParthD: ~/Desktop/NewTestFolder$ touch FirstFileinUbuntu.txt
parthd@ubuntuParthD: ~/Desktop/NewTestFolder$ ls -l
total 0
-parw-r-- 1 parthd parthd 0 Mar 14 17:41 FirstFileinUbuntu.txt
parthd@ubuntuParthD: ~/Desktop/NewTestFolder$ cat FirstFileinUbuntu.txt
cat: FirstFileinUbuntu: No such file or directory
parthd@ubuntuParthD: ~/Desktop/NewTestFolder$ cat > secondFile
Hello in the second file
It is a long day
ParthC12_2103032
^C
parthd@ubuntuParthD: ~/Desktop/NewTestFolder$ ls -l
total 4
-rw-rw-r-- 1 parthd parthd 0 Mar 14 17:41 FirstFileinUbuntu.txt
-rw-rw-r-- 1 parthd parthd 59 Mar 14 17:44 secondFile
parthd@ubuntuParthD: ~/Desktop/NewTestFolder$ cat secondFile
Hello in the second file
It is a long day
ParthC12_2103032
parthd@ubuntuParthD: ~/Desktop/NewTestFolder$
```

```

Activities Terminal Mar 14 17:46
parthd@ubuntuParthD: ~/Desktop/NewTestFolder$ cd ..
parthd@ubuntuParthD: ~/Desktop$ cd ..
parthd@ubuntuParthD: $ cd Desktop
parthd@ubuntuParthD: ~/Desktop$ mkdir newTest
parthd@ubuntuParthD: ~/Desktop$ ls -l
total 8
drwxrwxr-x 2 parthd parthd 4096 Mar 14 17:46 newTest
drwxrwxr-x 2 parthd parthd 4096 Mar 14 17:44 NewTestFolder
parthd@ubuntuParthD: ~/Desktop$ rmdir newTest
parthd@ubuntuParthD: ~/Desktop$ ls -l
total 4
drwxrwxr-x 2 parthd parthd 4096 Mar 14 17:44 NewTestFolder
parthd@ubuntuParthD: ~/Desktop$ cd ..
parthd@ubuntuParthD: $ ps aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.2 0.3 184140 13032 ? Ss 17:38 0:01 /sbin/init splash
root 2 0.0 0.0 0 0 ? S 17:38 0:00 [kthreadd]
root 3 0.0 0.0 0 0 ? I< 17:38 0:00 [rcu_gp]
root 4 0.0 0.0 0 0 ? I< 17:38 0:00 [rcu_par_gp]
root 5 0.0 0.0 0 0 ? I< 17:38 0:00 [slub_flushhw]
root 6 0.0 0.0 0 0 ? I< 17:38 0:00 [netns]
root 8 0.0 0.0 0 0 ? I< 17:38 0:00 [kworker/0:0-H-events_highpri]
root 9 0.0 0.0 0 0 ? I 17:38 0:00 [kworker/0:1-events]
root 11 0.0 0.0 0 0 ? I< 17:38 0:00 [mm_percpu_wq]
root 12 0.0 0.0 0 0 ? I 17:38 0:00 [rcu_tasks_kthread]
root 13 0.0 0.0 0 0 ? I 17:38 0:00 [rcu_tasks_rude_kthread]
root 14 0.0 0.0 0 0 ? I 17:38 0:00 [rcu_tasks_trace_kthread]
root 15 0.0 0.0 0 0 ? S 17:38 0:00 [ksoftirqd/0]
root 16 0.0 0.0 0 0 ? I 17:38 0:00 [rcu_preempt]
root 17 0.0 0.0 0 0 ? S 17:38 0:00 [migration/0]
root 18 0.0 0.0 0 0 ? S 17:38 0:00 [idle_inject/0]
root 19 0.0 0.0 0 0 ? S 17:38 0:00 [cpuhp/0]
root 20 0.0 0.0 0 0 ? S 17:38 0:00 [cpuhp/1]
root 21 0.0 0.0 0 0 ? S 17:38 0:00 [idle_inject/1]
root 22 0.0 0.0 0 0 ? S 17:38 0:00 [migration/1]
root 23 0.0 0.0 0 0 ? S 17:38 0:00 [ksoftirqd/1]
root 24 0.0 0.0 0 0 ? I 17:38 0:00 [kworker/1:0-cgroup_destroy]
root 25 0.0 0.0 0 0 ? S 17:38 0:00 [kdevtmpfs]
root 26 0.0 0.0 0 0 ? I 17:38 0:00 [inet_frag_wq]
root 27 0.0 0.0 0 0 ? I< 17:38 0:00 [kworker/u4:1-writelback]
root 28 0.0 0.0 0 0 ? I 17:38 0:00 [kworker/u4:1-writelback]
```

Experiment No - 3

Aim:- To study and implement Base-metal virtualization using Xen.

Theory :-

- 1.) Functions performed by Bare-metal Hypervisor.
 - Hardware Abstraction :- The base metal hypervisors provide a layer of abstraction between physical hardware and the virtualized environments
 - Resource Allocation :- The Bare-Metal hypervisor dynamically allocate CPU, memory, storage and network-bandwidth among VMs
 - Isolation :- Bare metal Hypervisors use techniques such as memory protection, extension to isolate VM.
 - Virtual machine Management :- Bare-metal hypervisor offer comprehensive management capabilities for VMs.
 - Performance Monitoring . the performance of both host system and VM is essential for bottlenecks and optimising resource usage

Security Management :- Bare-metal hypervisors implement various security mechanism to protect VMs from unauthorized malware and threats

- Device emulation :- Virtual hardware devices by bare-metal hypervisors emulate the functionality of physical devices
- High availability :- Bare-metal hypervisors offer high availability feature to ensure continuous operation of critical workloads

Q2) Compare Hosted & Bare-metal Hypervisors

Criteria	Base metal	Hosted
AKA	Native or Type 1	Type - 2
Deb	Runs directly on system with VMs running on them	Run on conventional OS.
Virtualization	Hardware	O.S
Operation	Guest OS and apps runs on hypervisors	Runs as an app on host O.S.
Stability	Better scalability	Not so much
Setup	Complicated	Easy.

Examples:- VMWare for Base-metal and sunsVB for Type-2

Q3) Explain the following terms.

- Horizontal scaling also known as scaling out is the process of increasing one of nodes and machine in resource pool for a sequential price of logic to be processed in parallel across numerous devices horizontal scaling scales by breaking it then into smaller chunks and delegating the logic to new machine

- Vertical scaling :- Also known as scaling up is process of increasing power of increasing the power of existing system such as CPU or RAM to meet the rising demands. Because there is no need to alter the logic vertical scaling is simpler.

- Auto scaling :- Also referred as "Automatic scaling" is a cloud computing technique for dynamically allocating computational resources. Depending on the load to server farm or pool, the no of server that are active typically vary automatic as user needs punctuate.

- Load Balancing :- It is essential technique used in cloud computing to optimize resource utilization & ensure that no of single resources is overburdened with traffic it is process of

distributing workloads across multiple computing resources such as server, virtual machines or container to achieve better performance, availability and scalability

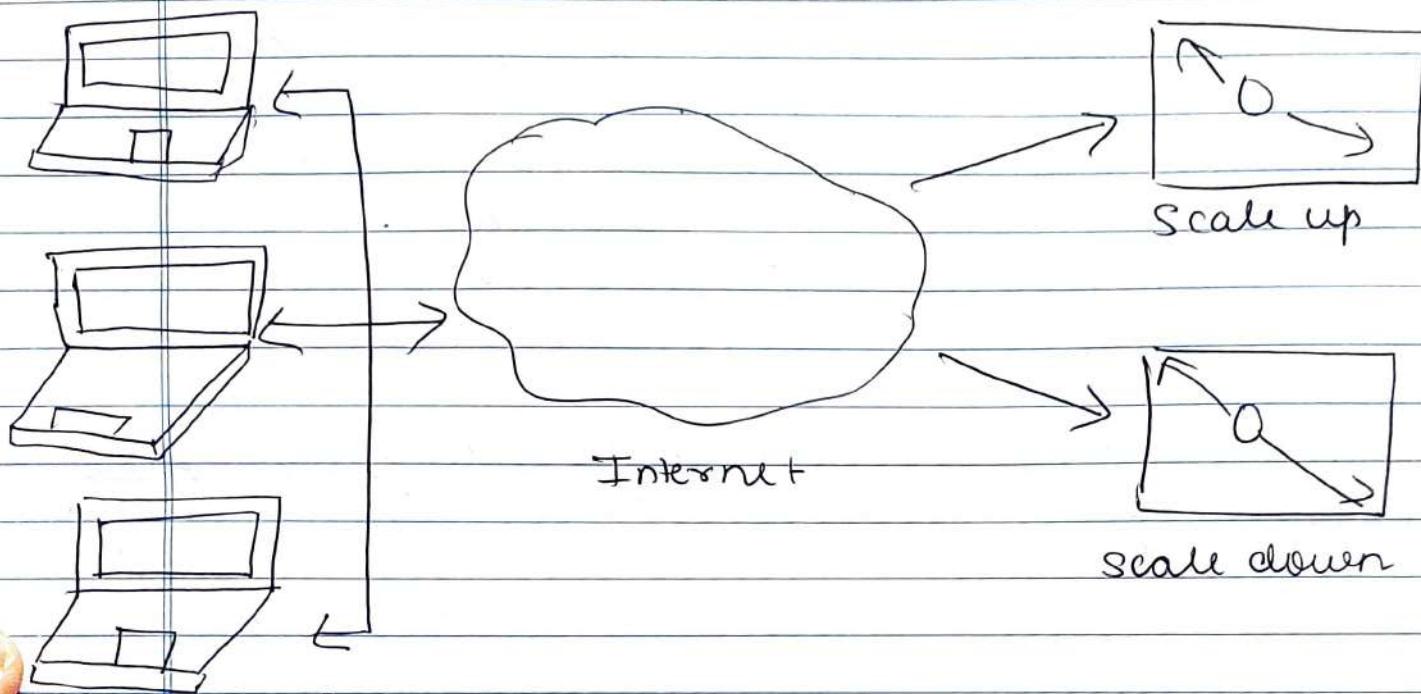


Fig.) How Auto scaling works.

- High availability load balancing insures, that there is no single point of failure in system which provide high availability and fault tolerance to handle severe failure.
- Scalability :- It makes it easier to scale resources up or down as needed helps to handle spikes in traffic or changes in demands.

EXPERIMENT NO: 3

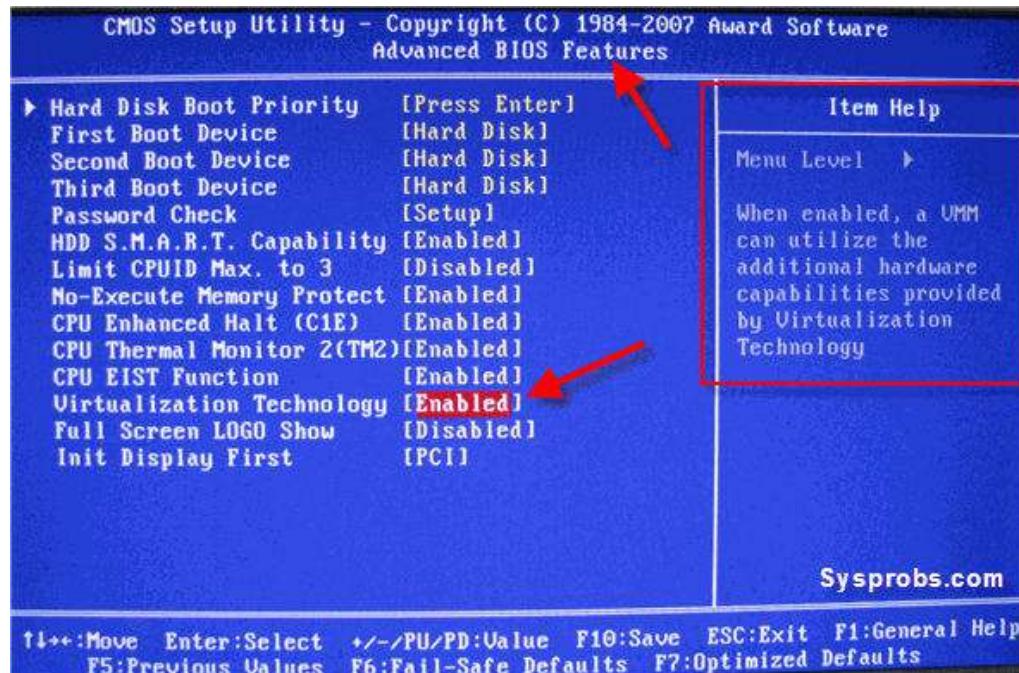
How to install XenServer? Metal Virtualization using XEN.

Step1: Install XenServer

- i) Insert Bootable CD into CDROM or Bootable Pendrive and make first boot device from BIOS.



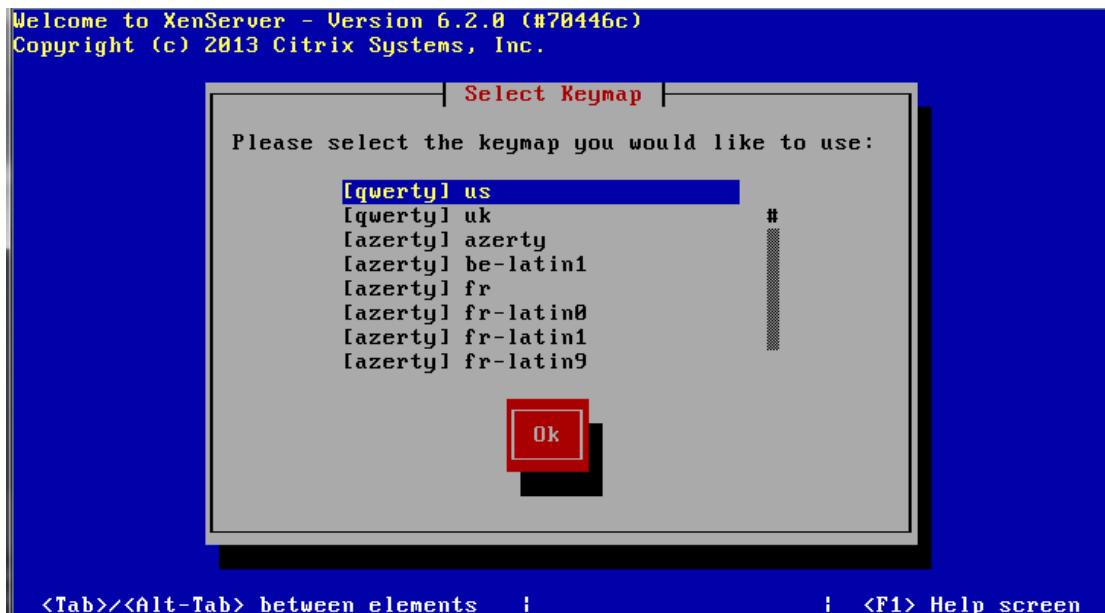
ii) Press F2 for advance options and Make Enabled Virtualization Technology from BIOS



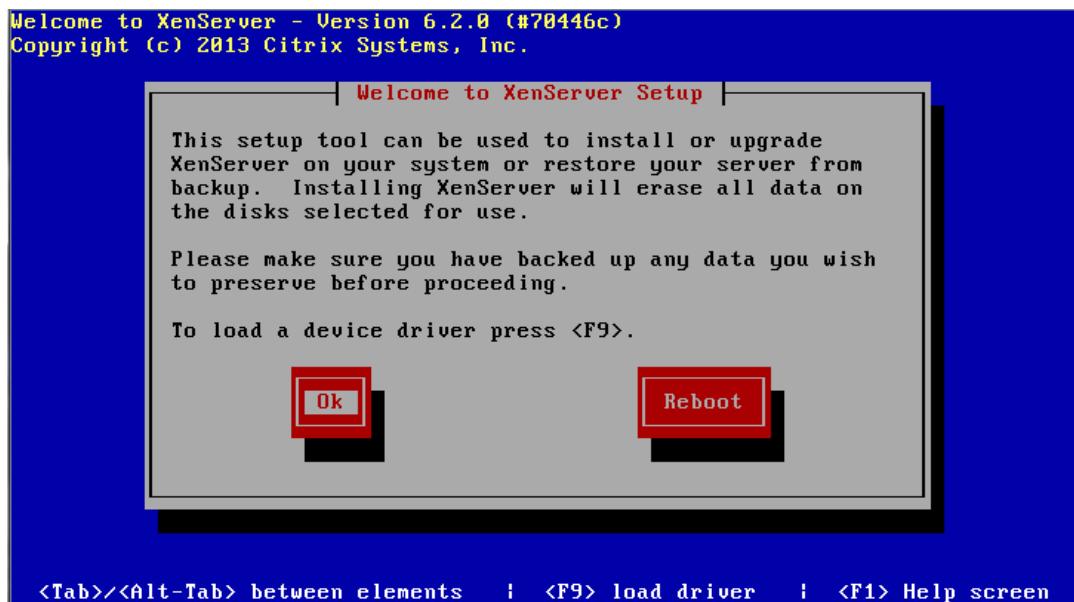
Save and Reboot



iii we begin by choosing the keymap i.e Keyboard Layout



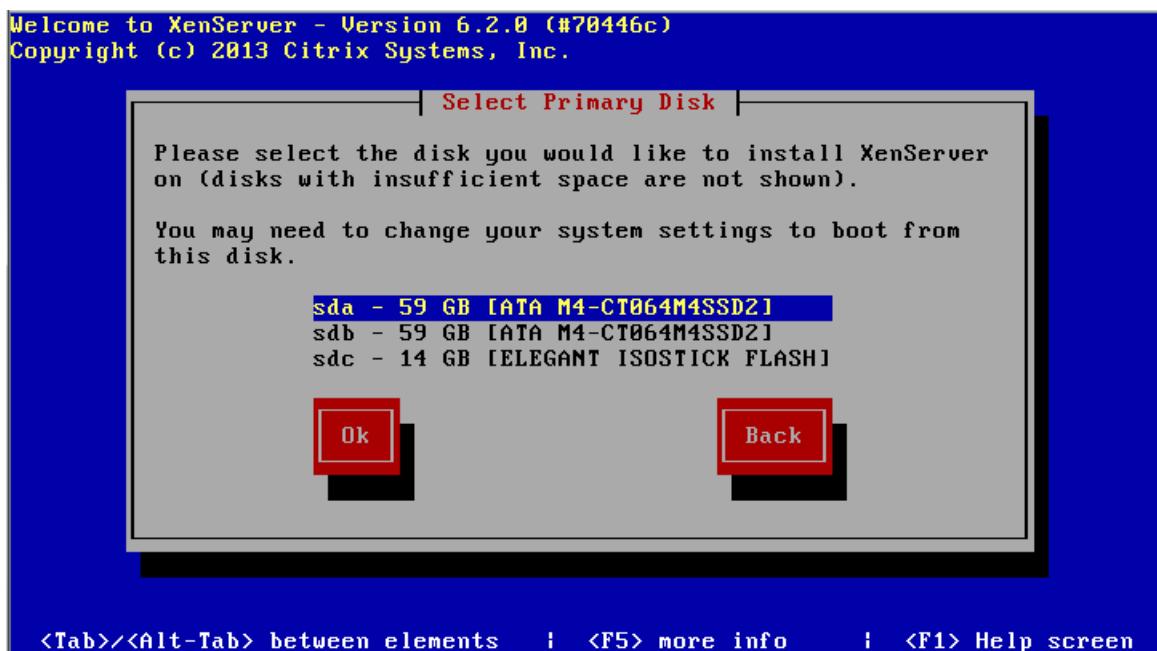
iv) Press Enter to load Device Drivers



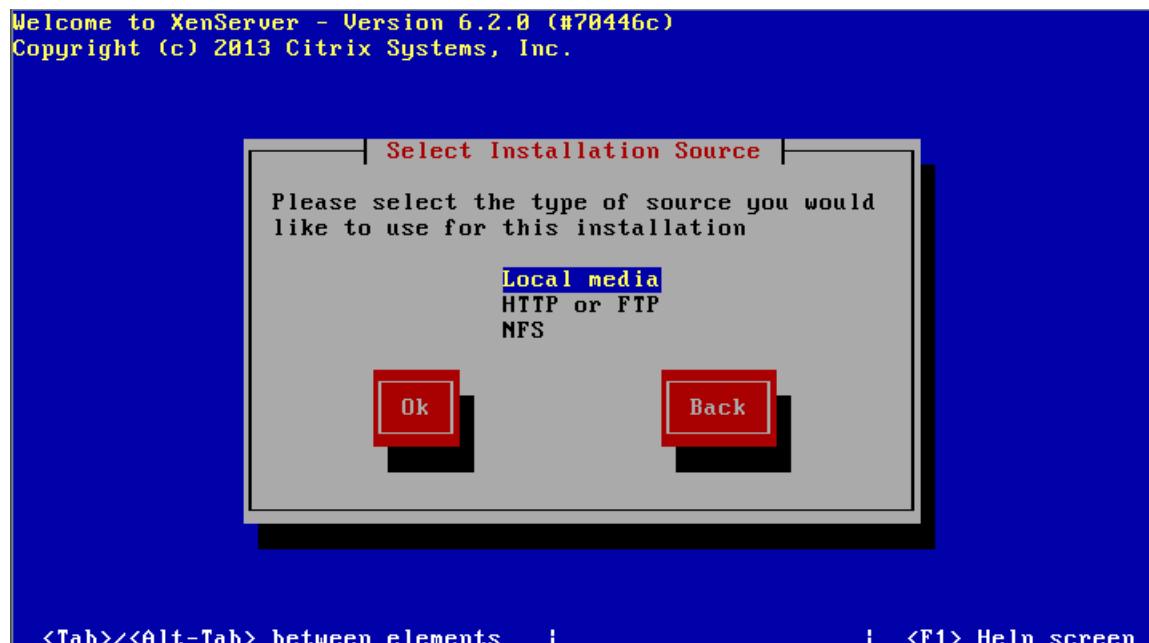
V) Press Enter to accept End User License Agreement



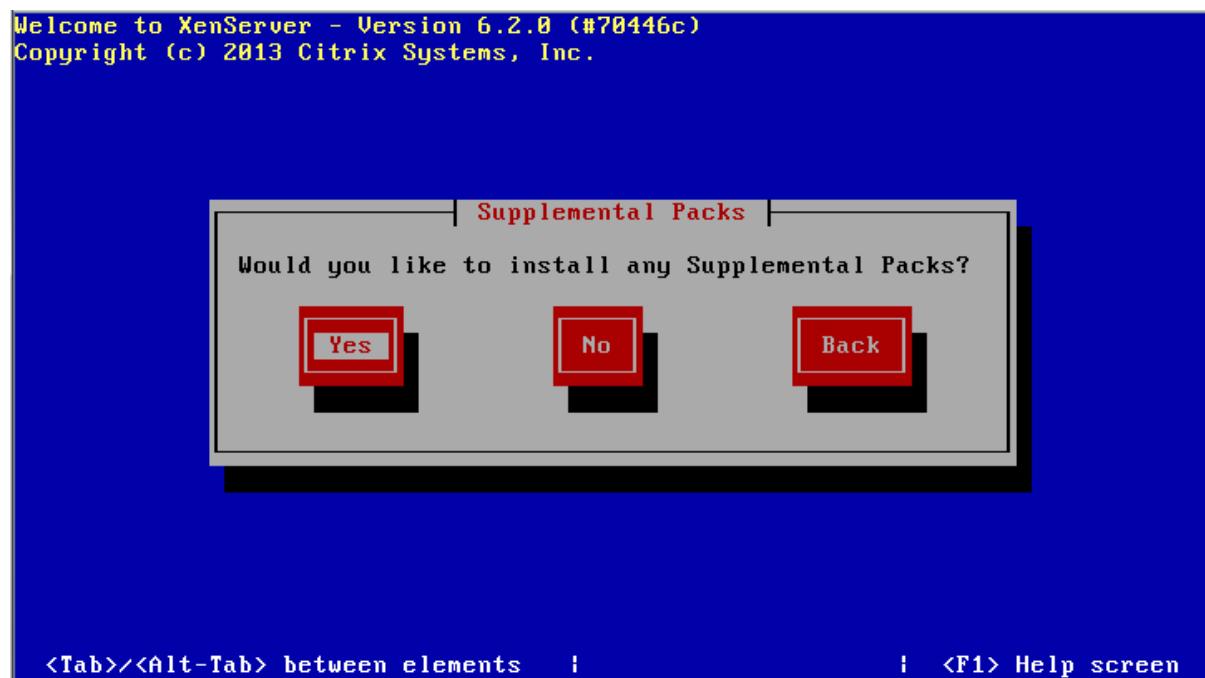
Vi) Select Appropriate disk on which you want to install Xenserver



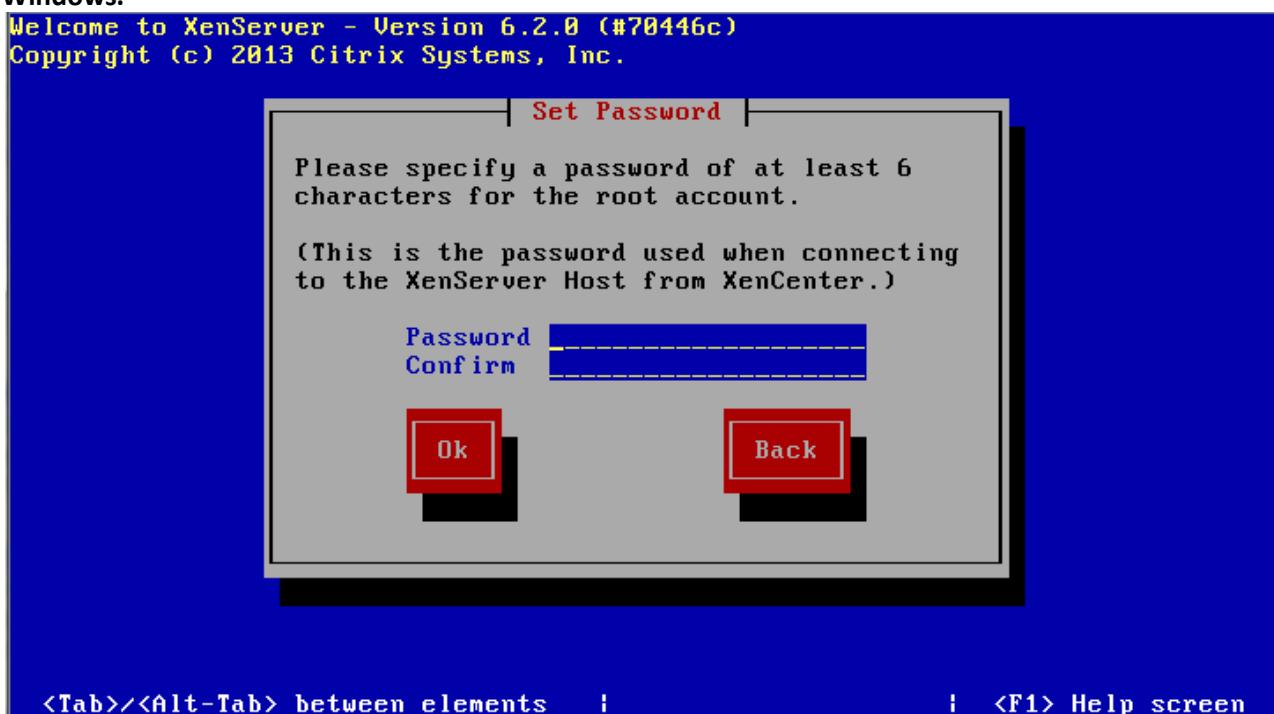
vii) Select appropriate Installation Media



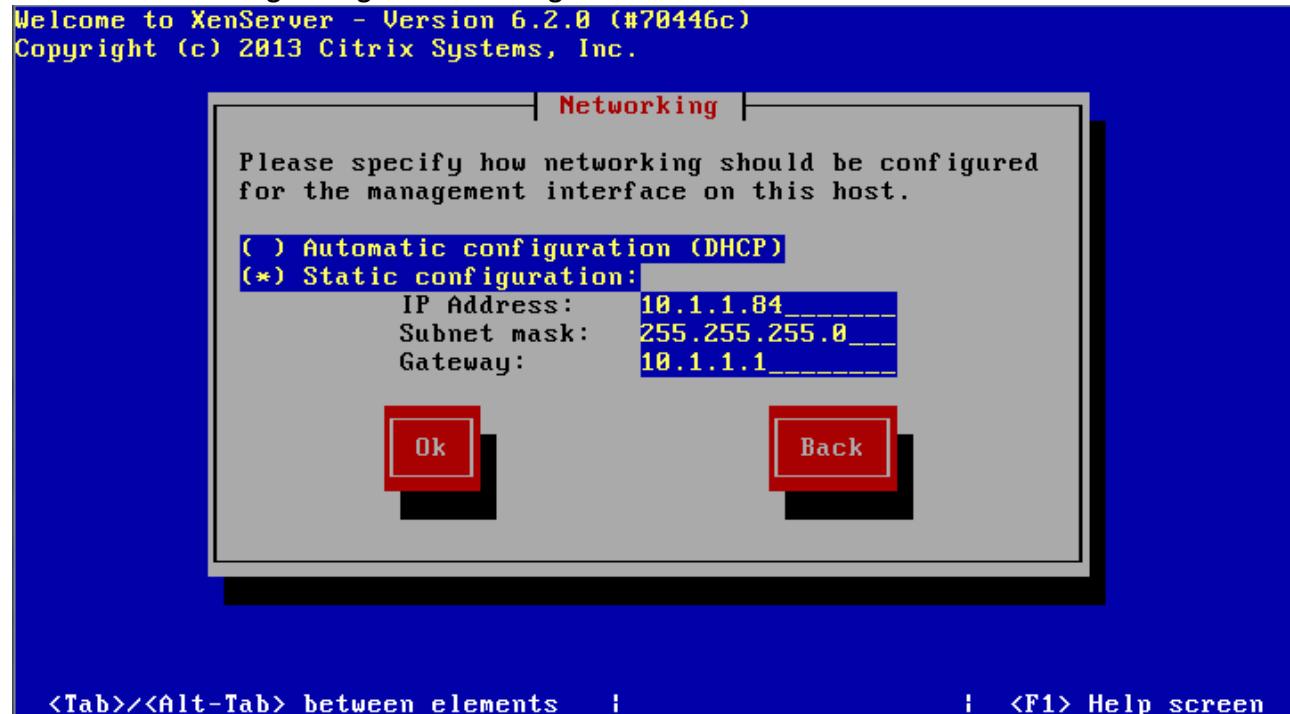
viii) Press yes to Select additional packages for installation , Otherwise Press No



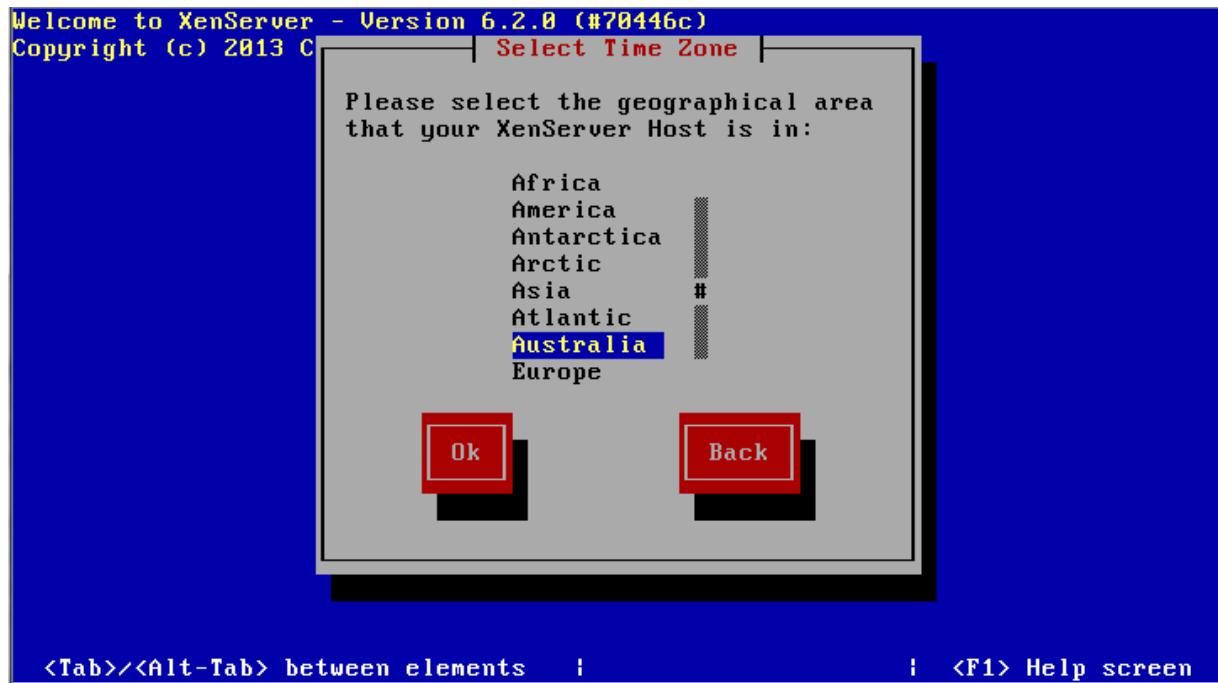
viii) Pick a root password – you will need this for logging in via the console or via the XenServer client on Windows.



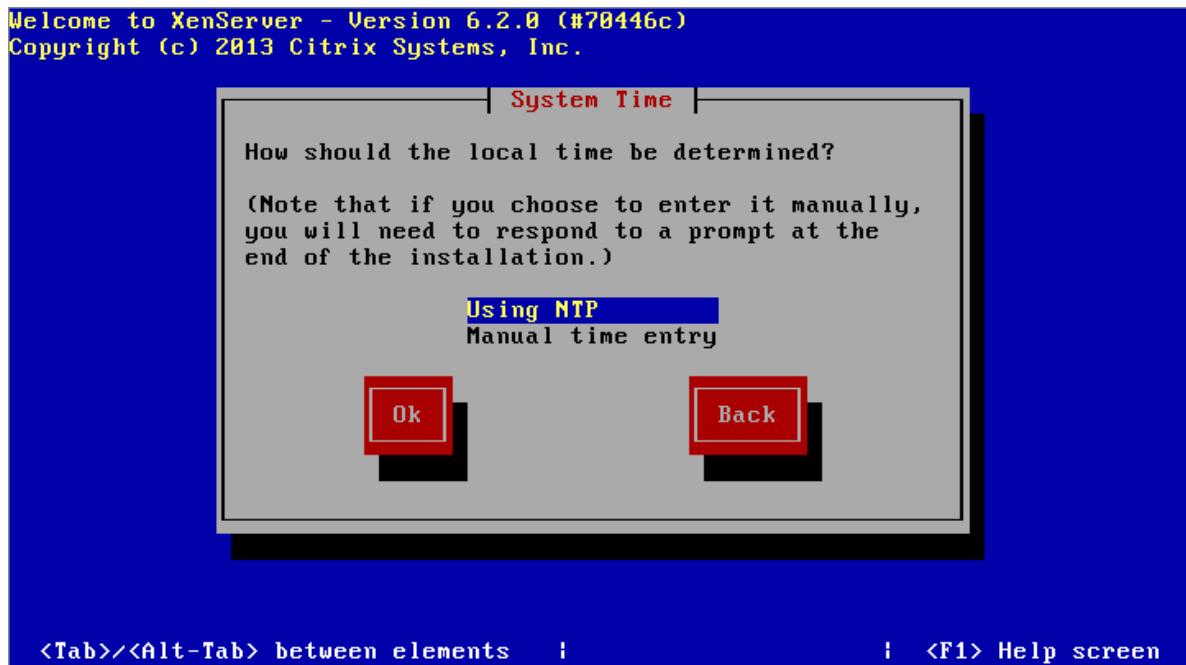
ix) The motherboard we are installing have two Ethernet ports, both of which are supported by XenServer. Choose the one you wish to use for the management network – you can change this later. Here we get to choose the networking settings for our management network.



x) Select Time zone

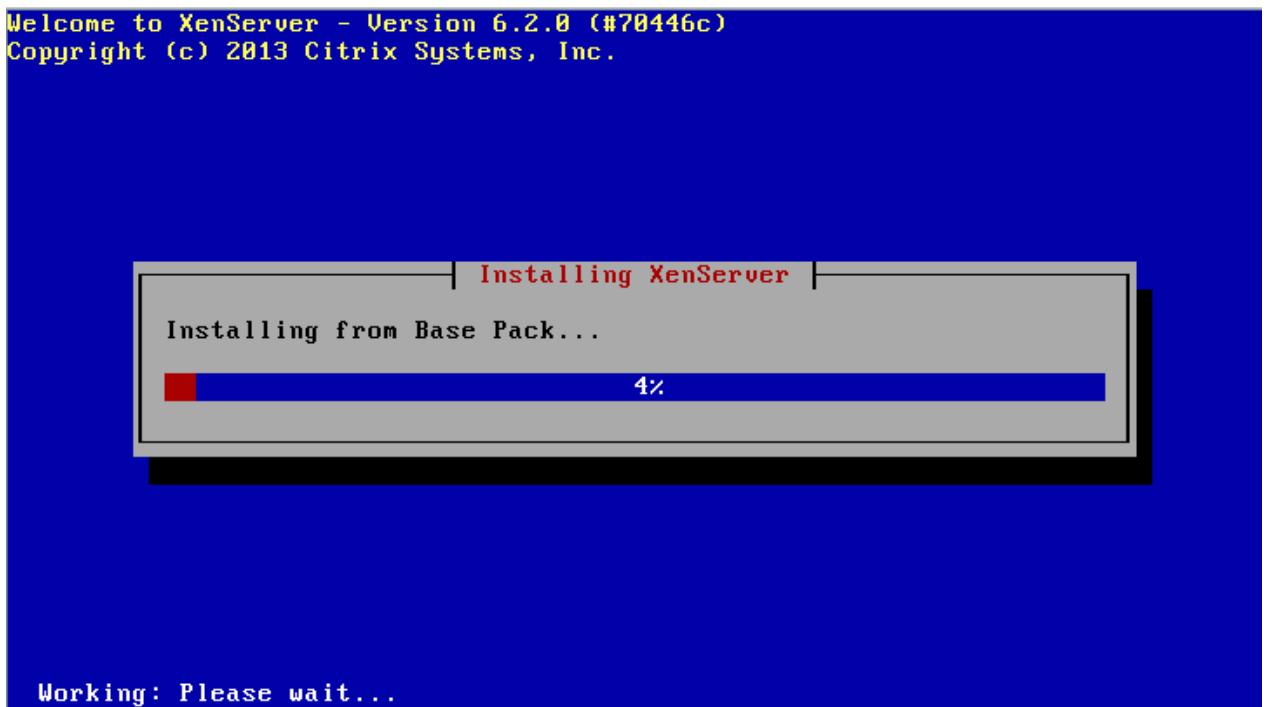
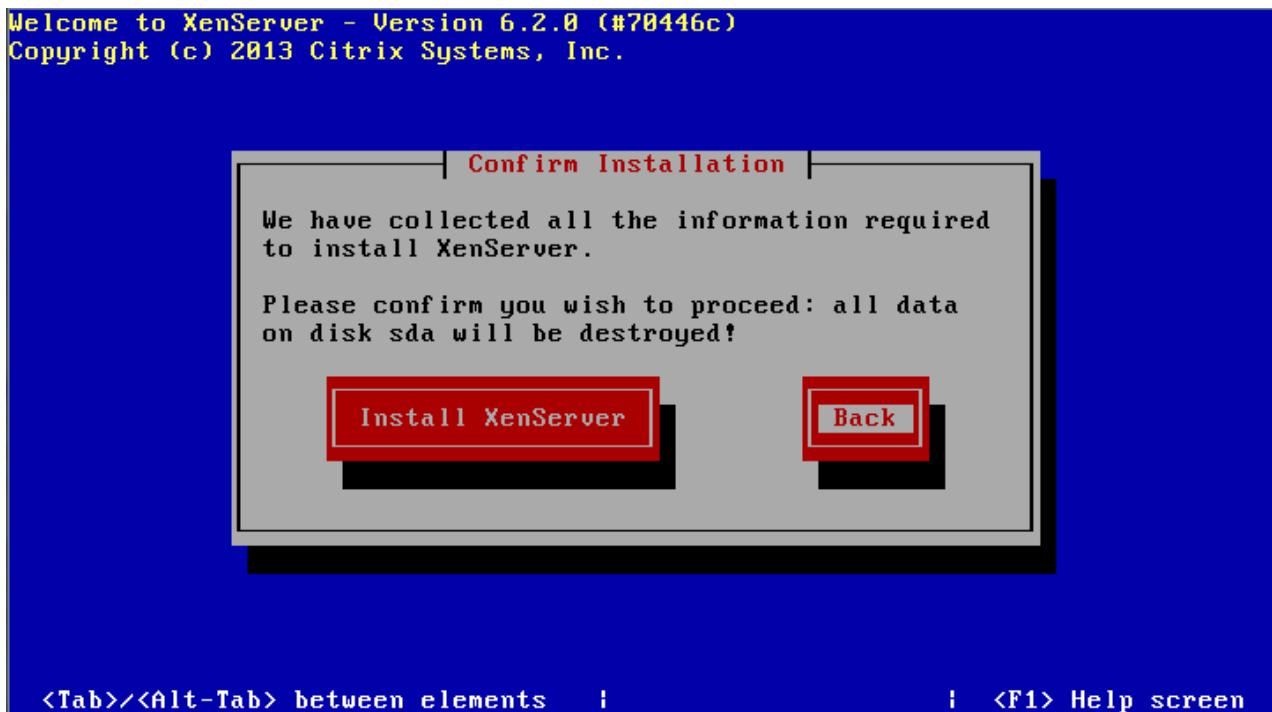


Xi)Specify NTP Server address and start installation to set your time .



Shirish Shetty 2103164 C32

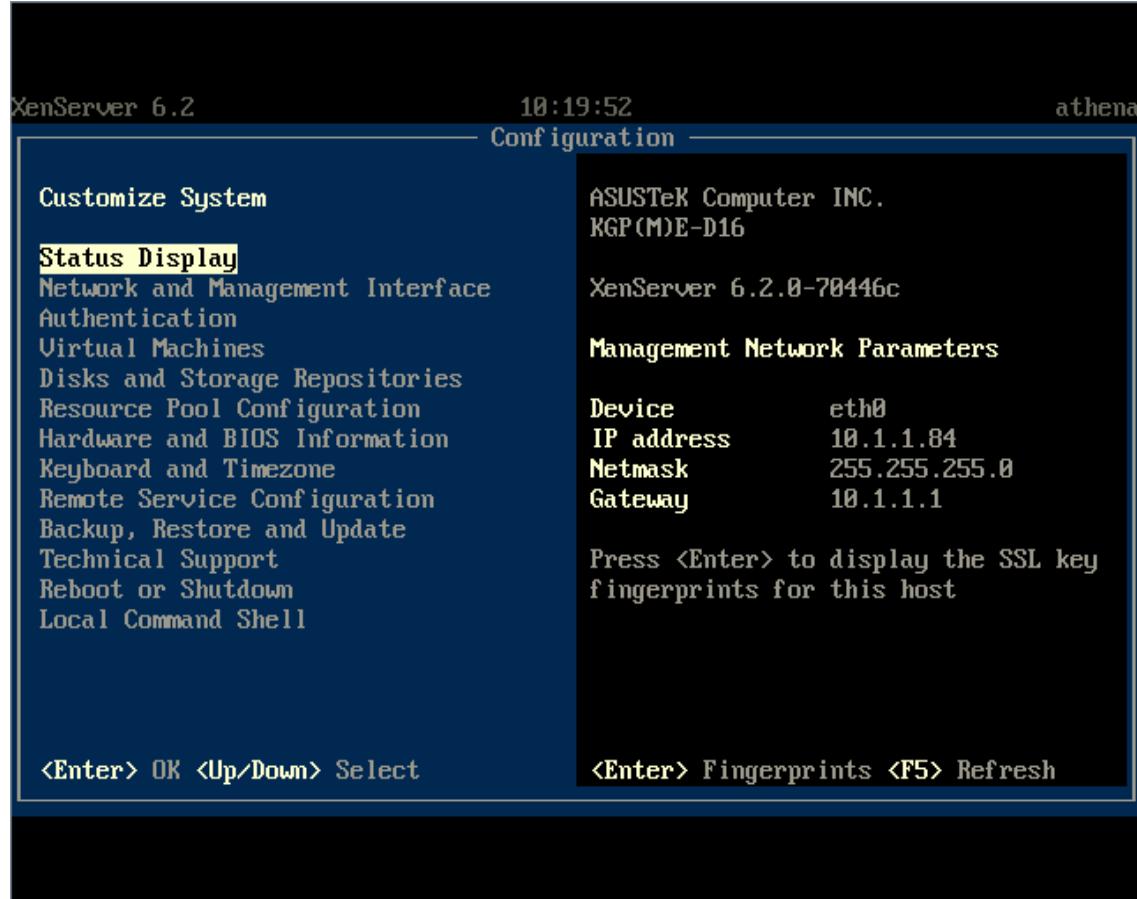
Xii) Press Enter to start installation of XenServer



This is the loading screen for installed new XenServer .

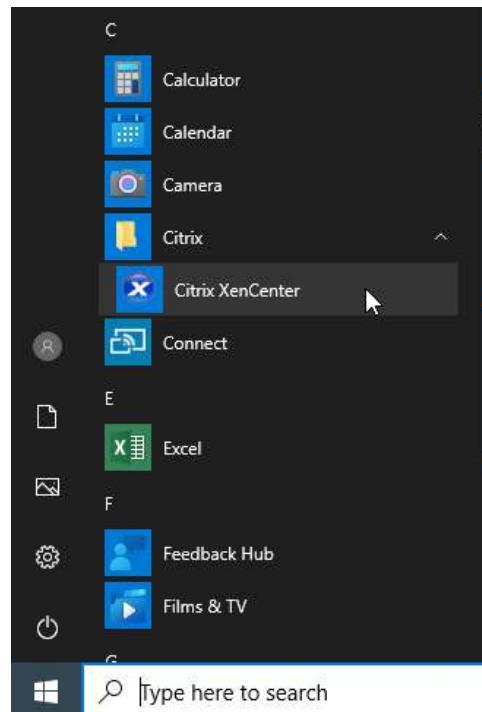


The console screen for XenServer

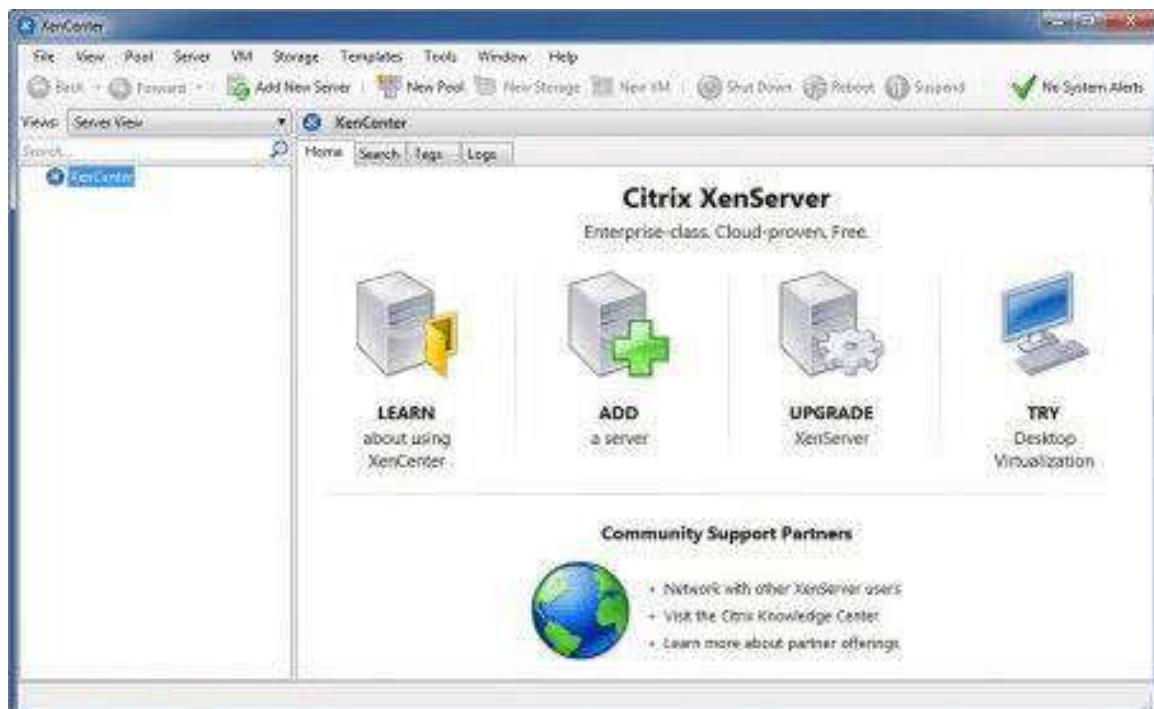


Step 2 : Connect XenCenter to XenServer

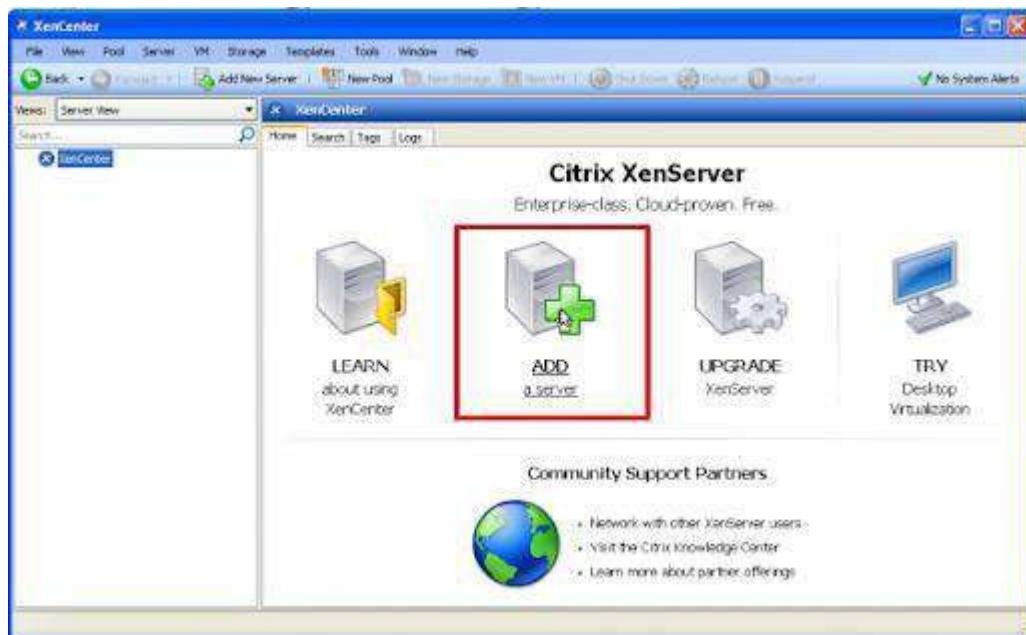
- i) Download the XenCenter a management utility from XenServer IP address as a URL on browser. Install XenCenter and open it from start Menu of Windows on Machine 2.



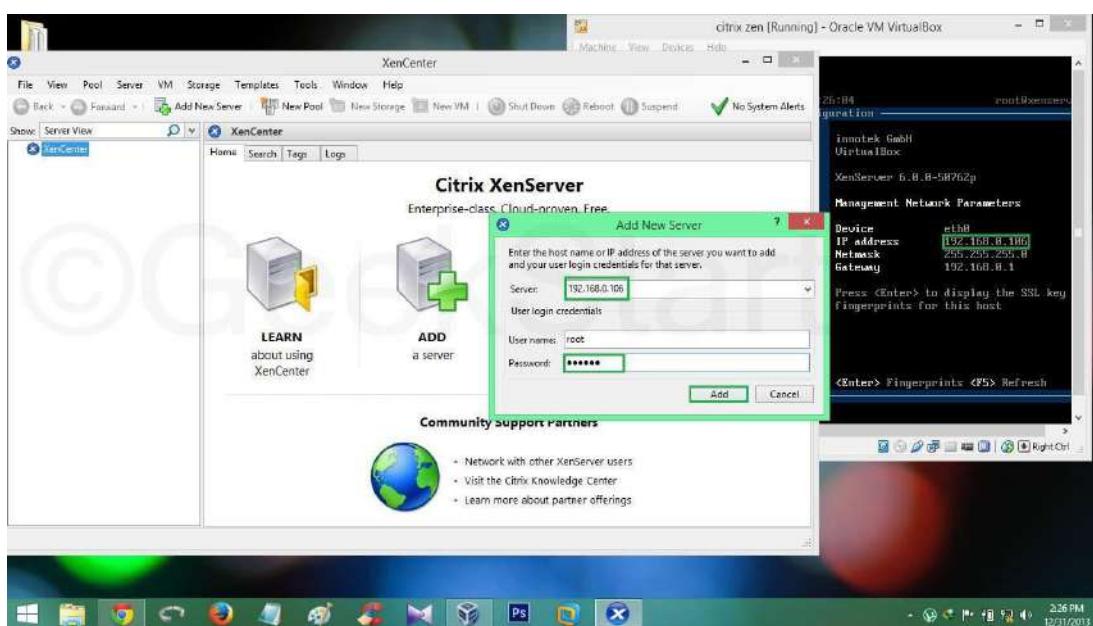
- ii) Once you click on Citrix XenCenter ..It Looks like as below:



- iii) To Connect to the XenServer host which is configured earlier, click **ADD a Server**



- iv) Enter IP address of XenServer and Enter User login credentials and click Add



v) Once you clicked on Add , it will ask to configure a master password for all the XenServers.



vi) Xenserver is added now to XenCenter

XenCenter

File View Pool Server VM Storage Templates Tools Window Help

Back Forward Add New Server New Pool New Storage New VM Shut Down Reboot

Views: Server View

Search...

XenCenter
bj-xenserver
DVD drives
Local storage
Removable storage

bj-xenserver Overview

Name	CPU Usage	Used Memory
bj-xenserver Default install of XenServer	17% of 2 CPUs	841 of 2048 MB

Step 3: Create Local ISO Storage .

Now before creating Virtual Machine we have to Create storage repository which is nothing but shared directory on XenCenter which holds all iso files and which is required to install Operating system on XenServer .

Using Local command Shell

i) First View Xen Directory Structure

```
# df - h
```

ii) Now make folder to store ISO images

```
# mkdir /var/ISO_images
```

iii) Using **wget** command download iso files .

for example:

```
#wget http://releases.ubuntu.com/16.04/ubuntu-16.04.6-desktop-amd64.iso
```

OR

USING PENDRIVE (need to Mount Pendrive)

i) #mkdir /mnt/myusb

ii) #ls -l /dev/sdb1 to check drive for removable disk.

iii) #mount -t vfat -o rw,users /dev/sdb1 /mnt/ myusb

iv) #cd / mnt/myusb

v) # ls (content of pendrive)

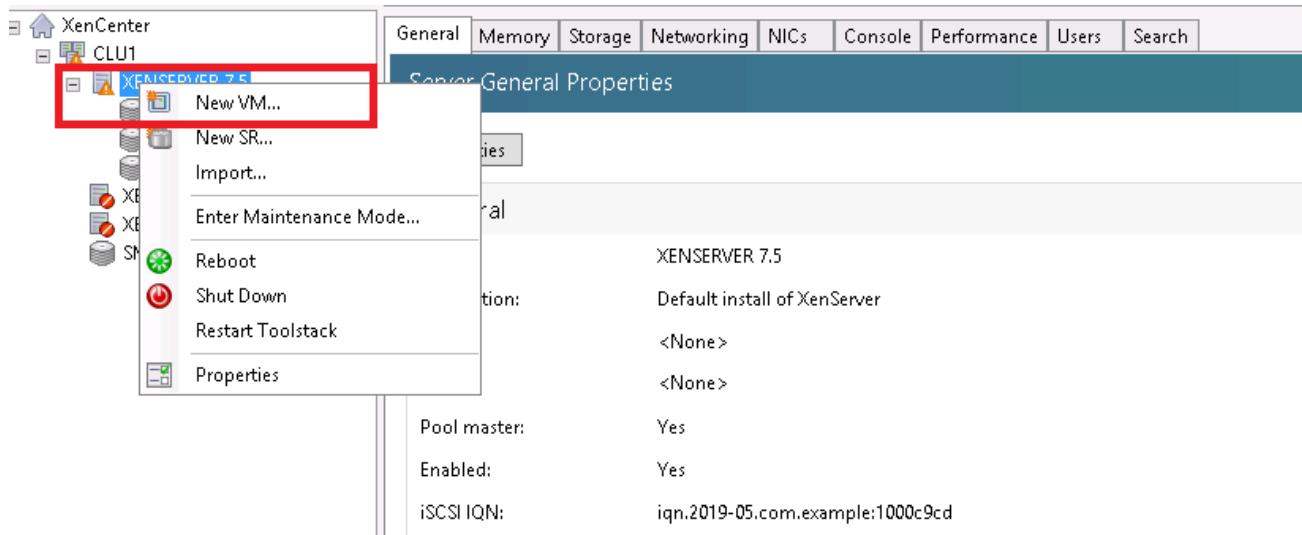
vi) # cp Ubuntu-16.04.5-desktop-i386.iso /var/ISO_images

vii) Reboot or shutdown Xenserver from Xencenter or from console of Xenserver

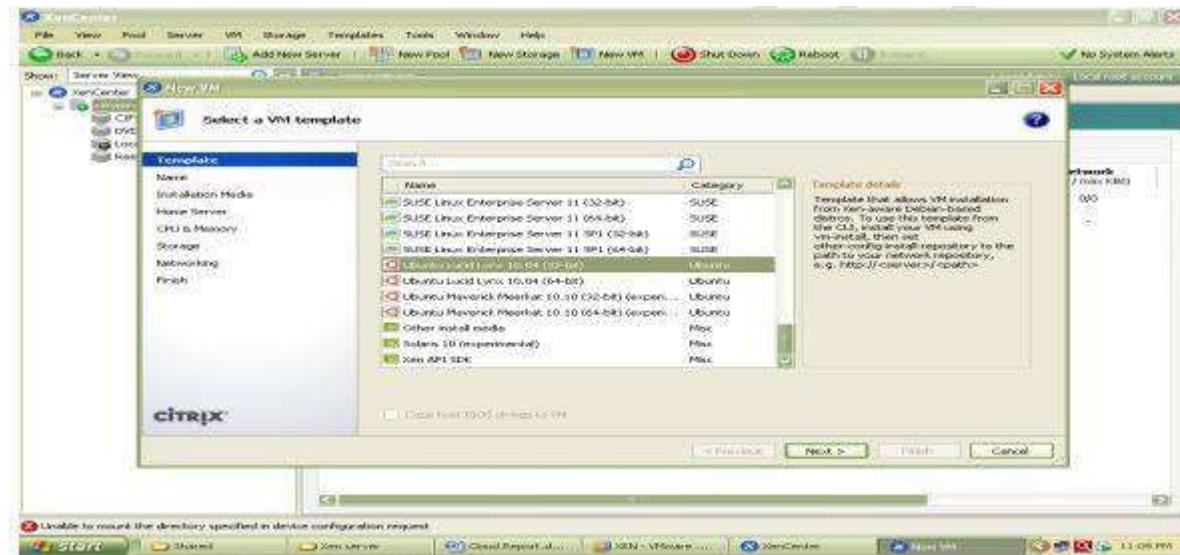
Shirish Shetty 2103164 C32

Step 4: Installation of Virtual Machine from Xencenter

i) Right click on Xenserver icon on Xen center and select New VM.



ii) Select VM template



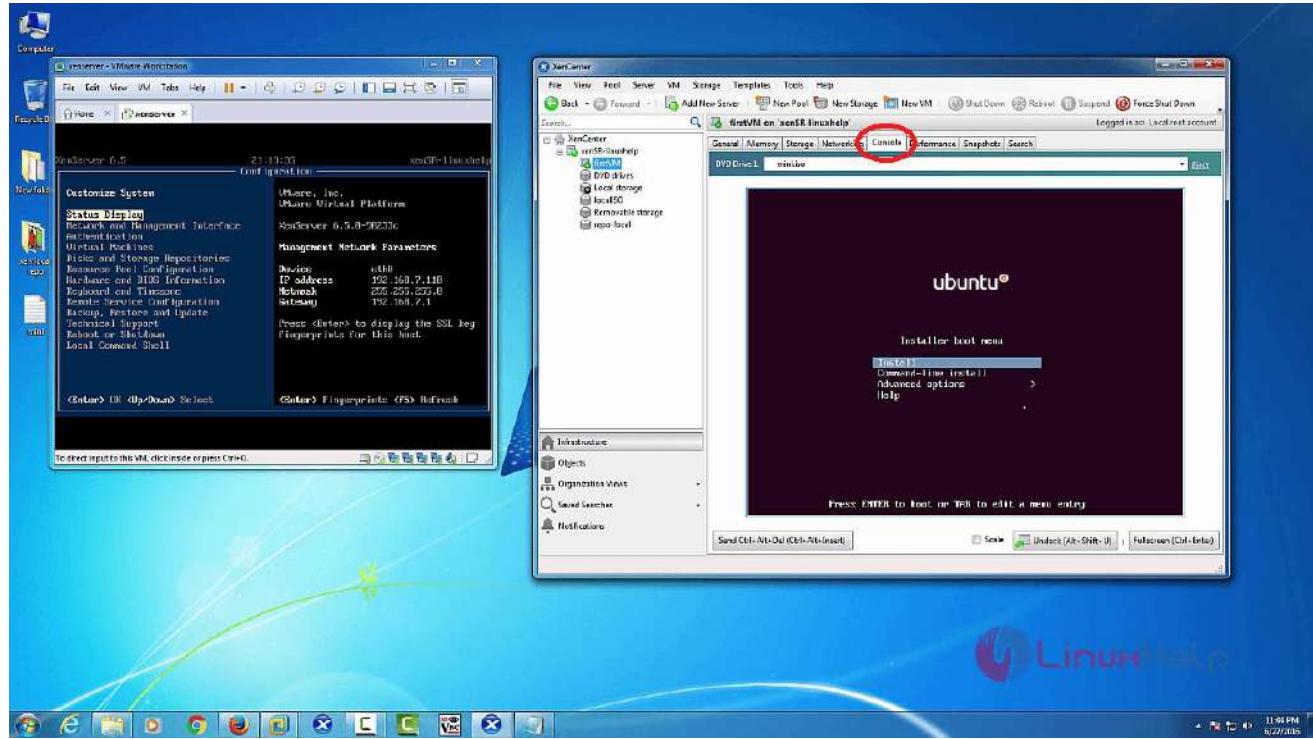
iii) Name the virtual machine

iv) Locate the operating system installation media To select appropriate OS iso file.

v) Allocate CPU and memory to VM

vi) Select networking

v) finish



Thus we have successfully created a virtual machine in xenServer using XenCenter tool.

Experiment No - 4

Aim:- To study and implement infrastructure as a service using AWS.

Theory :-

E C 2 :-

Amazon Elastic Compute Cloud , a VM service in AWS for running customizable virtual computer

Features :- On-demand computing , scalable capacity, no upfront hardware investments, easy app deploy and scaling based on traffic size

AMI :- Amazon Machine Image required to launch instances. Contains EBS snapshots, root template launch permissions etc.

Storage :- AMIs are stored in Amazon S3, identified by unique IDs can be created from scratch or bundled from existing EC2 instances .

Types of EC2 computing instances :-

General Purpose :- Balanced compute, memory and networking resources for diverse workloads like web service.

Computer Optimized :- High performance processors for compute-bound tasks like batch processing.

Memory Optimized :- Fast performance for memory intensive applications like databases.

Accelerated Computing :- Hardware accelerator for task like ml and graphics processing

Storage Optimized :- High , sequential read/ write access for large data sets.

Elastic IP Address :- These are static IPv4 address for dynamic cloud computing. You can also use them in DNS records for domain pointing. They are public IPv4 address and reachable from net.

Remote Desktop Control(RDP) :- It allows remote usage of desktop computer. It is commonly used, especially with Windows. User can access edit files run apps. Remote desktop access is more about accessing physical desktops remotely.

Implementation:-

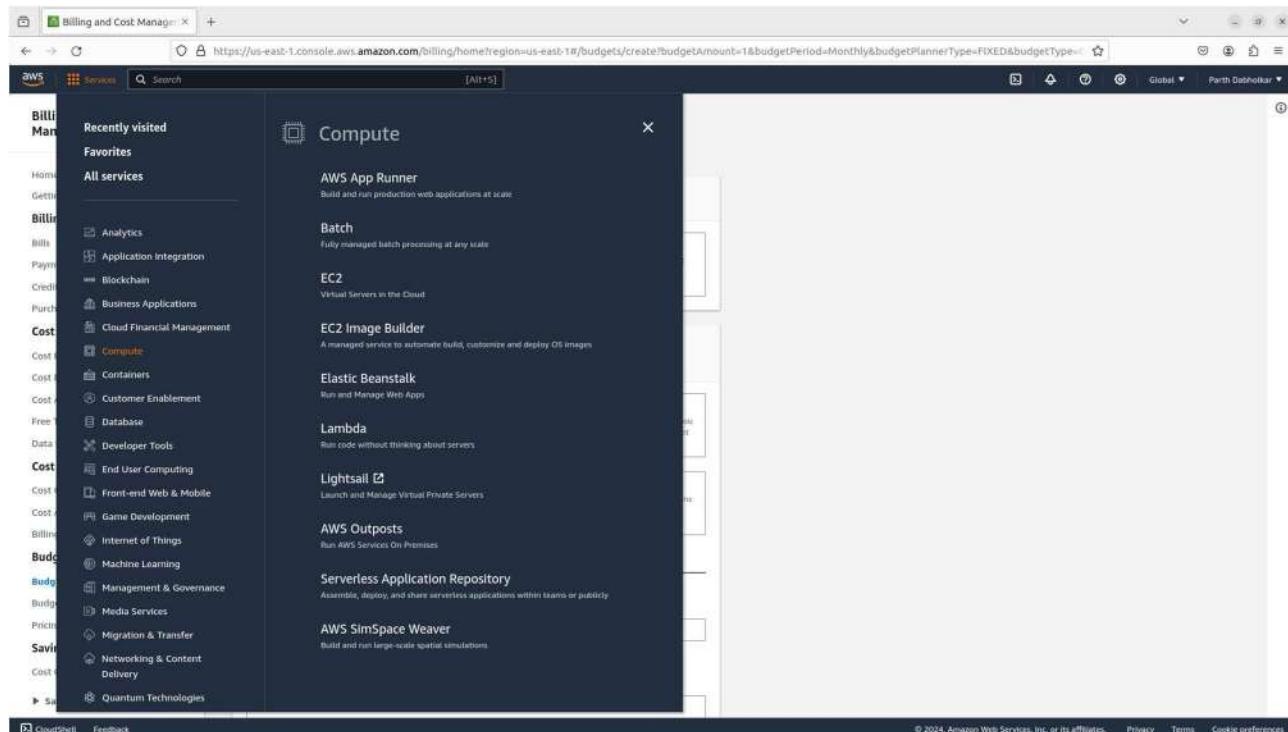
Open EC2 page on amazon, click launch instance, select Ubuntu Server 20.0.4 and select Instance Type, Configure Installation Details add a new volume and tag.

Conclusion:- Hence an EC2 instance was setup and run on Amazon Web Services.

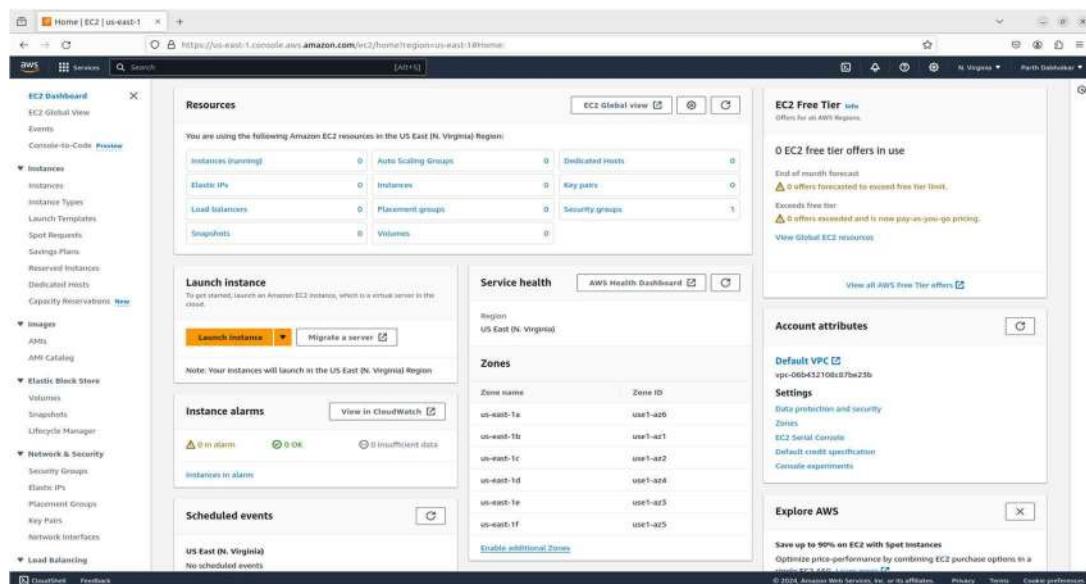
EXPERIMENT NO: 4

AIM: To study and implement Infrastructure as a Service using AWS (EC2).

Step 1: Log in to the account. Select the proper server from right top. Here we have selected asia-mumbai. Click on the services option left to the search bar. A drop down appears. Select Compute option and click on EC2 service.



Step 2 : Check the dashboard. All resources must be assigned with value 0 except the Security Groups which must be 1. Then Click on the Launch Instance.



Step 3: Name the EC2 Instance/Virtual Machine with a suitable name. For Eg: “MyFirstWebServer”.

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Name and tags' step, the 'Name' field contains 'MeraPehlaWebServer'. There is also a link to 'Add additional tags'.

Step 4 : Select the appropriate AMI(Amazon Machine Image) such as Windows or Ubuntu according to your need. Since Ubuntu is fast and open source, we use Ubuntu AMI.

The screenshot shows the AWS 'Application and OS Images (Amazon Machine Image)' catalog. Under the 'Quick Start' section, the 'Ubuntu' icon is selected. A detailed view of the 'Ubuntu Server 22.04 LTS (HVM), SSD Volume Type' AMI is shown, including its ID (ami-03f4878755434977f), architecture (64-bit (x86)), and a 'Verified provider' badge.

Shirish Shetty 2103164 C32

Step 5 : Select the instance type. It gives the description about the number of Virtual CPUs used, Amount of Memory, Storage etc. For the Ubuntu AMI used. We use the t2.micro which is a free tier service.

The screenshot shows the 'Instance type' section of an AWS instance creation wizard. The 't2.micro' option is selected, highlighted with a red border. To the right of the selection, it says 'Free tier eligible'. Below the selection, there is descriptive text about the instance type, including its family (t2), vCPU count (1), memory (1 GiB), current generation status (true), and base pricing for On-Demand Linux, Windows, RHEL, and SUSE. There is also a note about additional costs for pre-installed software. On the far right, there are buttons for 'All generations' and 'Compare instance types'.

Step 6 : Key pair can be used to securely connect to our instance. We shall proceed without a key.

The screenshot shows the 'Key pair (login)' section of the AWS instance creation wizard. A note states that a key pair is used for secure connection and that access to the selected key pair is required before launching. Below this, there is a dropdown menu for 'Key pair name - required' with two options: 'Proceed without a key pair (Not recommended)' (selected) and 'Default value'. To the right of the dropdown is a 'Create new key pair' button with a plus sign icon.

Shirish Shetty 2103164 C32

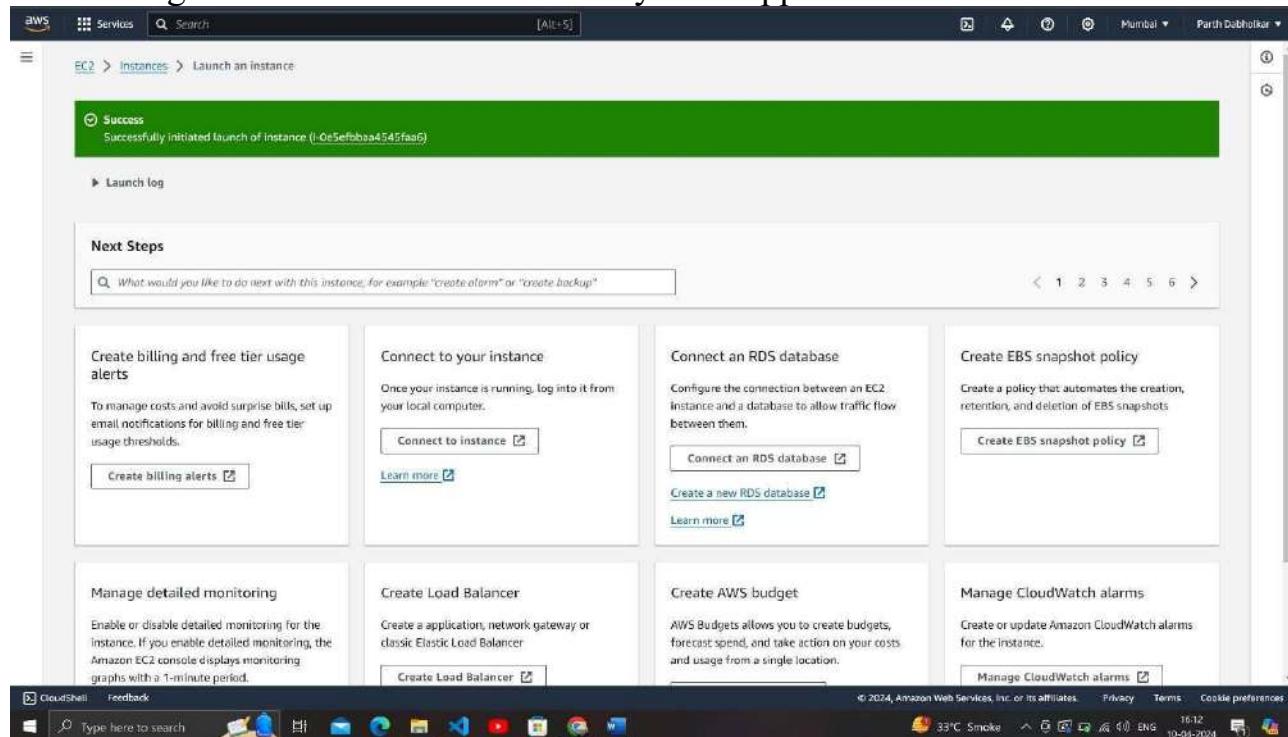
Step-7: In the network settings, select Create security groups. Allow SSH traffic from all IP addresses by selecting the option.

The screenshot shows the 'Network settings' section of the AWS Launch Wizard. It includes fields for Network (vpc-00dffeb2613906df4), Subnet (No preference), Auto-assign public IP (Enable), and Firewall (security groups). A note states 'Additional charges apply when outside of free tier allowance'. Below these, there are two options: 'Create security group' (selected) and 'Select existing security group'. A note below says 'We'll create a new security group called 'launch-wizard-1' with the following rules:' followed by three checkboxes: 'Allow SSH traffic from Anywhere' (selected), 'Allow HTTPS traffic from the internet' (unchecked), and 'Allow HTTP traffic from the internet' (unchecked). A warning message at the bottom right says 'Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' with a close button.

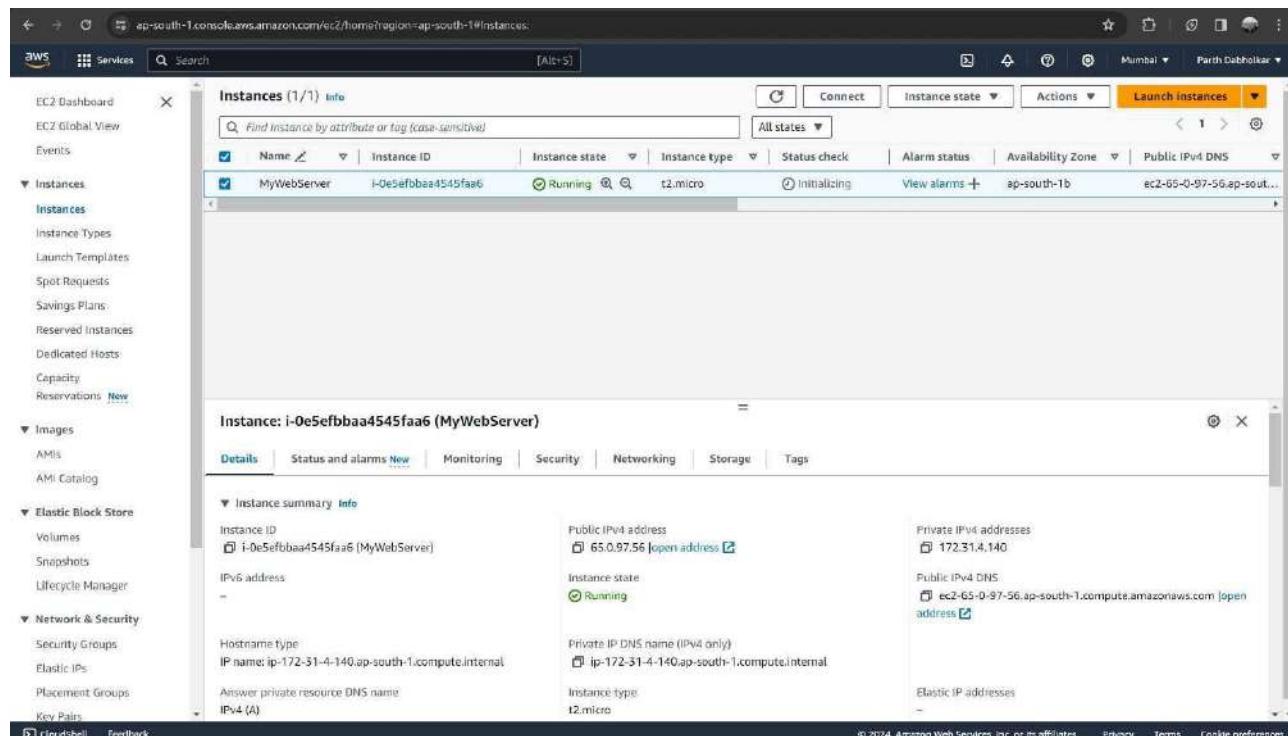
Step-8: In the configure storage section, let all the fields have default values. Nothing to be changed in it.

The screenshot shows the 'Configure storage' section of the AWS Launch Wizard. It displays a single volume configuration: 1x 8 GiB gp2 Root volume (Not encrypted). A note indicates 'Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage' with a close button. Below this is an 'Add new volume' button. At the bottom, a note says 'Click refresh to view backup information' with a refresh icon, followed by 'The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.' and '0 x File systems' with an 'Edit' button.

Step-9: Click on launch instance in the bottom right corner of the screen. A green alert stating instance launched successfully must appear.

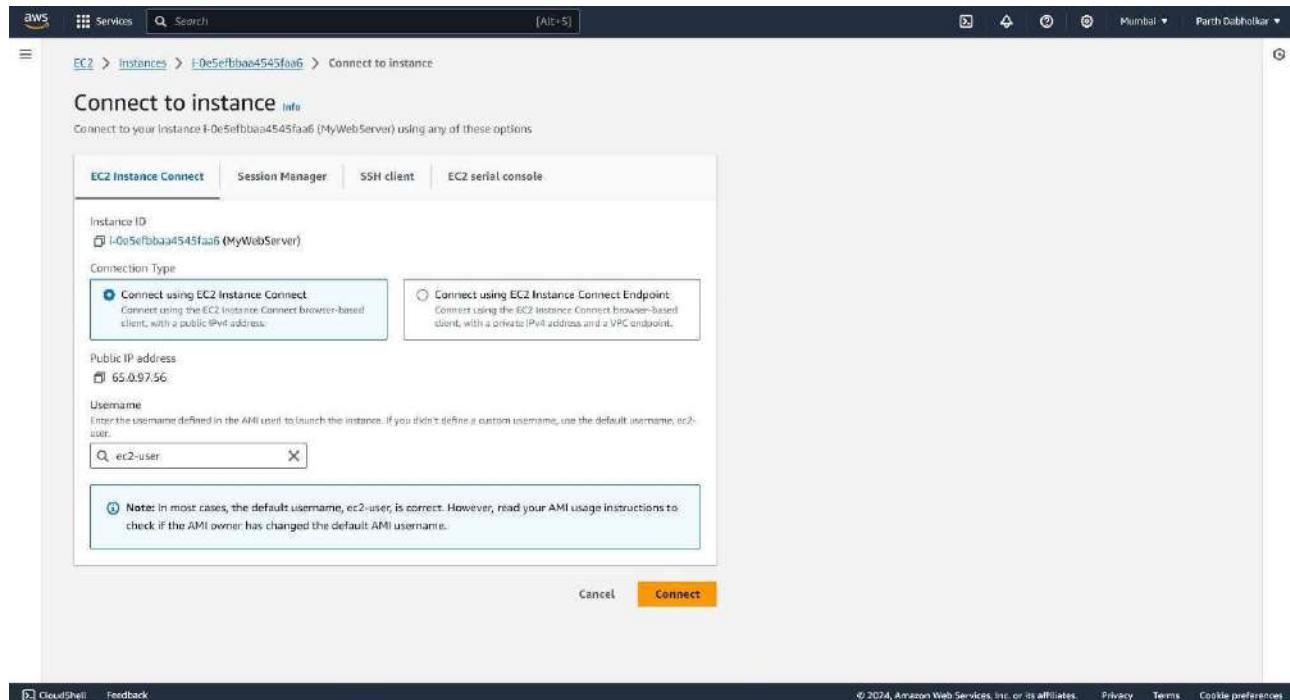


Step-10: Select the view instances on bottom right-hand corner of the screen. This will redirect you to a new page showing all the instances. Select the instance you have created by ticking the radio box beside it. Then click Connect.

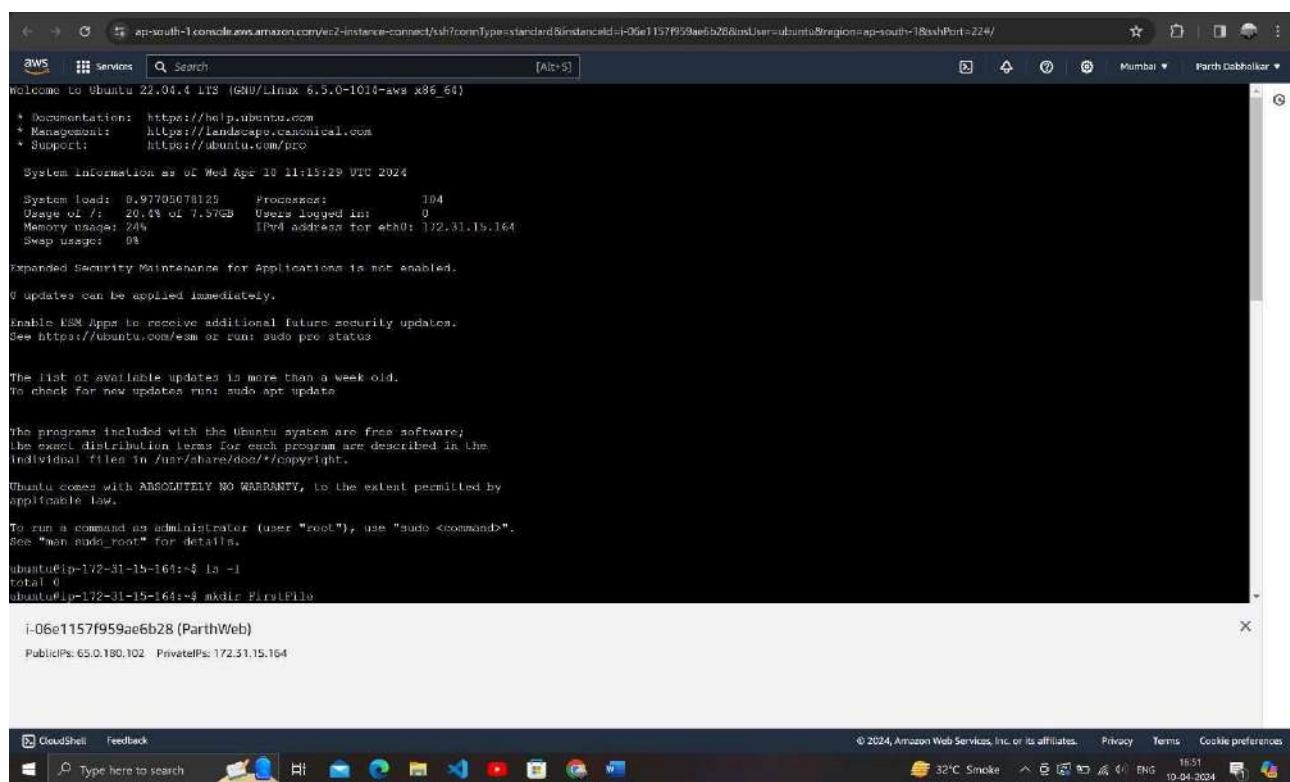


Shirish Shetty 2103164 C32

Step-11: You will be redirected to a new screen. In the connect to instance section, select Connect using EC2 Instance and hit Connect on the bottom right of the screen.



Step-12: Wait for the instance to launch. Once launched, an ubuntu terminal will be displayed on the screen. This is the virtualization of Ubuntu in AWS Cloud.



Shirish Shetty 2103164 C32

Step-13: Perform any Ubuntu/Linux commands to test the Ubuntu Terminal. Some of the commands used in this experiment are:

- mkdir: to create a directory
 - ls -l: to list all the directories present in the system
 - cd: to enter a directory
 - cat > filename: to create a text file and enter data into it.
 - cat filename: to display the contents of the file.
 - ifconfig: to get the network address of the system.
 - ps: to display all the active process in ubuntu system.
 - rmdir: to delete a directory from the system.
 - rm: to remove a file from directory.

```
aws s3 ls s3://parth-deshkar@172.31.15.164/ --recursive
```

```
aws [AWS] Services Q Search [Artif] Home Member Port Dashboards
arn:aws:lambda:ap-south-1:1571959ae6b28:User:ClientRegion(ap-south-1) [Port 224]
Hello to the first text file in AWS from Parth Bhaloker
On to the next one
arn:aws:lambda:ap-south-1:1571959ae6b28:User:ClientRegion(ap-south-1) [Port 224]
total 0
-rw-r--r-- 1 ubuntu ubuntu 4096 Apr 10 11:17 FirstText
ubuntu@ip-172-31-15-164:~/FirstFile$ cd ..
total 1
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 10 11:17 Firstfile
ubuntu@ip-172-31-15-164:~/FirstFile$ rm FirstText
ubuntu@ip-172-31-15-164:~/FirstFile$ cd ..
total 1
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 10 11:17 Firstfile
ubuntu@ip-172-31-15-164:~/FirstFile$ ls -l
total 13
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 10 11:17 Firstfile
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 10 11:16 Secondfile
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 10 11:16 Thirdfile
ubuntu@ip-172-31-15-164:~/FirstFile$ ls -l
total 13
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 10 11:17 Firstfile
ubuntu@ip-172-31-15-164:~/FirstFile$ ls -l
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 10 11:17 Firstfile
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 10 11:16 Secondfile
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 10 11:16 Thirdfile
ubuntu@ip-172-31-15-164:~/FirstFile$ tail Firstfile
Hello to the first text file in AWS from Parth Bhaloker
On to the next one
ubuntu@ip-172-31-15-164:~/FirstFile$ port FirstText
Hello to the first text file in AWS from Parth Bhaloker
ubuntu@ip-172-31-15-164:~/FirstFile$ cat FirstText
ubuntu@ip-172-31-15-164:~/FirstFile$ cd ..
ubuntu@ip-172-31-15-164:~/FirstFile$ ps
 PID TTY          PSR    USER      COMMAND
 1555 pts/0    00:00:00 bash
 1556 pts/0    00:00:00 bash
ubuntu@ip-172-31-15-164:~/FirstFile$ ls -l
total 8
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 10 11:17 FirstFile
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 10 11:16 Secondfile
ubuntu@ip-172-31-15-164:~/FirstFile$ ls -l
total 8
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 10 11:17 FirstFile
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 10 11:16 Secondfile
ubuntu@ip-172-31-15-164:~/FirstFile$
```

CloudBees Jenkins © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookies preferences 32°C Shreveport, LA 10:41 AM, 2024 16:02

Step-14: After performing all the operations, begin with deletion of instance. First select the instance and then in the instance state, select stop the instance. Once stopped, again select the instance and go to instance state actions. Select terminate instance and wait for an alert stating instance terminated successfully.

The screenshot shows the AWS EC2 Instances page. The left sidebar includes options like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, and Key Pairs. The main content area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
MyWebServer	i-0e5efbaa4545faa6	Terminated	t2.micro	-	View alarms	ap-south-1b	-
ParthWeb	i-06e1157f959ae6b28	Stopping	t2.micro	Initializing	View alarms	ap-south-1b	ec2-65-0-180-102.ap-s

A success message "Successfully stopped i-06e1157f959ae6b28" is shown above the table. Below the table, a message says "Instance: i-06e1157f959ae6b28 (ParthWeb)".

Terminate instance?

⚠️ On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

Are you sure you want to terminate these instances?

Instance ID	Termination protection
i-06e1157f959ae6b28 (ParthWeb)	Disabled

To confirm that you want to terminate the instances, choose the terminate button below. Instances with termination protection enabled will not be terminated. Terminating the instance cannot be undone.

Cancel **Terminate**

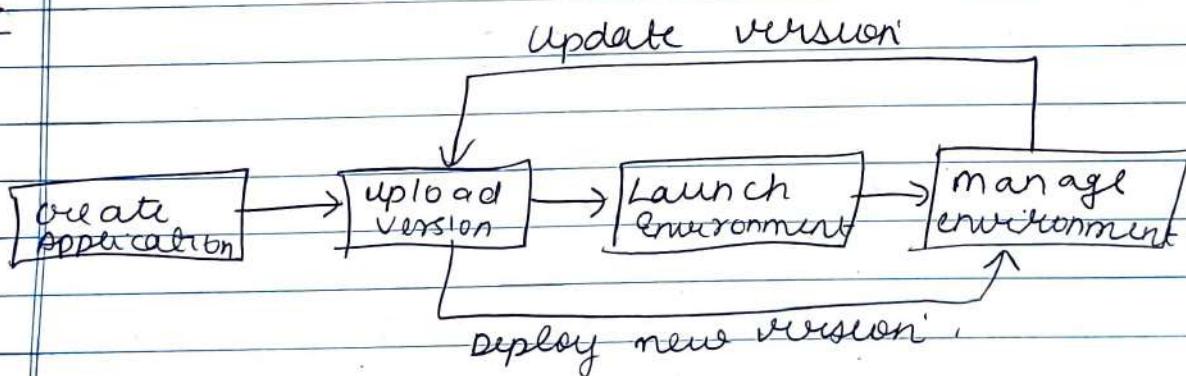
Experiment No:- 5

Aim:- To study and implement Platform as service using AWS Elastic Beanstalk Service

Theory:-

Amazon Web Services (AWS) Elastic Beanstalk Services:-

With Elastic Beanstalk, one can quickly deploy and control apps in AWS cloud without having to learn about infrastructure that runs these apps. Elastic Beanstalk reduces restricting choice or ctrl. You simply upload your application, and it automatically handles details of capacity provisioning, load balancing, scaling & app health monitoring.



Elastic Beanstalk supports app developed in Go, Java, .NET, NodeJS, PHP, Python and Ruby. When you deploy your application, it builds selected supported platform version and.

provisions one or more AWS resources, such as Amazon EC2 instances to run your apps

EBS supports a variety of programming language & framework like Java, .NET, Node.js, Python, Ruby, PHP, Go, Docker.

Compare EC2 & Elastic Beanstalk.

EC2

1.) It provides raw virtual machine in cloud. User have full control over OS & networking

2.) User are responsible for deploying application managing SW updates

3.) Supports a wide range of OS & services

4.) Pricing based on selected instance type & usage.

EBS

1.) It abstracts away the solution infrastructure details. and provides platform as a service soln.

2.) EBS takes care of underlying infrastructure

3.) Supports specific programming language & frameworks

4.) Pricing is based on resource used by underlying infra.

5) more effort needed to handle increased traffic

5) Monitors performance & provides easier scalability.

Elastic Load Balancing :-

ELB automatically distributes your incoming traffic across multiple targets, such as EC2 instances, containers and IP addresses, in one or more zones. It monitors health of its registered targets and routes traffic only to healthy targets. Elastic Load Balancing scales your load balancer capacity automatically in response to change in traffic.

Steps for deploying web apps/web services on AWS Elastic Beanstalk.

Step 1:- Login to AWS console & go to Elastic Beanstalk

Step 2:- Click on create application

Step 3:- Write app information :- Name, tag, platform etc.

Step 4:- In app code, select sample application & then click on button create application

Step 5:- Click on environment → check the health of environment till it becomes 'OK'

Step 6:- Click the UP!

(A) SRI

26/3/2018

Conclusion:-

By performing the above experiment, we were able to understand EBS and its applications & deploy services on EBS.

EXPERIMENT NO: 5

AIM: To study and Implement Platform as a Service using AWS Elastic Beanstalk Service.

Step1 : Login to AWS console and go to Elastic Beanstalk

The screenshot shows the AWS Services search interface. A search bar at the top contains the text 'ebs'. Below it, a sidebar lists services under 'Amazon S3' such as Buckets, Access Points, Object Lambda Access, Multi-Region Access P, Batch Operations, and Access analyzer for S3. The main search results show 'Services (15)' and 'See all 15 results'. Two services are highlighted: 'Elastic Beanstalk' with the subtext 'Run and Manage Web Apps' and 'EC2' with the subtext 'Virtual Servers in the Cloud'.

Step 2: Click on Create Application

The screenshot shows the 'Welcome | Elastic Beanstalk' page. It features a dark header with the AWS logo and a navigation bar with links like 'Compute', 'Logs', 'Metrics', 'CloudWatch Metrics', 'CloudWatch Logs', and 'CloudWatch Metrics Insights'. The main content area has a heading 'Amazon Elastic Beanstalk' with the subtext 'End-to-end web application management.' Below this is a 'Get started' section with a button labeled 'Create application'. To the right is a 'Pricing' section stating 'There's no additional charge for Elastic Beanstalk. You pay for Amazon Web Services resources that we create to start and run your web application, like Amazon S3 buckets and Amazon EC2 instances.' At the bottom right is a 'Getting started' section with a link 'Launch a web application'.

Step 3: Write Application information : Name, Tag,Platform etc.

The screenshot shows the 'Application information' configuration screen for a new application named 'Mynewappshilpa'. The application name field contains 'Mynewappshilpa'. Below it is a note: 'Up to 100 Unicode characters, not including forward slash (/)'. The 'Application tags' section allows adding up to 50 tags. One tag is currently added: 'name' with 'selfWebSite' as its value. There are buttons for 'Add tag' and 'Remove tag'. A note at the bottom states: 'Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. Learn more'.

Step 4: In Application Code: select **sample application** and then Click on button **Create Application**

This will take a few minutes.

Step 5: Click on Environments -> Check the health of Environment wait till it becomes 'OK'

Elastic Beanstalk Environments

All environments

Environment name	Health	Application name	Date created	Last modified	URL	Run vers.
Apptestshilpa-env	Pending	apptestshilpa	2022-02-22 23:26:06 UTC+0530	2022-02-22 23:28:04 UTC+0530	Apptestshilpa-env.eba-rrf38rza.ap-south-1.elasticbeanstalk.com	

Elastic Beanstalk Environments

All environments

Environment name	Health	Application name	Date created	Last modified	URL	Run vers.
Apptestshilpa-env	Ok	apptestshilpa	2022-02-22 23:26:06 UTC+0530	2022-02-22 23:29:14 UTC+0530	Apptestshilpa-env.eba-rrf38rza.ap-south-1.elasticbeanstalk.com	Sam Appl

Step 6: Click the URL

Elastic Beanstalk Environments

All environments

Health	Application name	Date created	Last modified	URL	Running versions	Platform	Platform state
Ok	apptestshilpa	2022-02-22 23:26:06 UTC+0530	2022-02-22 23:29:14 UTC+0530	Apptestshilpa-env.eba-rrf38rza.ap-south-1.elasticbeanstalk.com	Sample Application	PHP 8.0 running on 64bit Amazon Linux 2	Supp



💡 What's Next?

- [AWS Elastic Beanstalk overview](#)
- [Deploying AWS Elastic Beanstalk Applications in PHP Using Eb and Git](#)
- [Using Amazon RDS with PHP](#)
- [Customizing the Software on EC2 Instances](#)
- [Customizing Environment Resources](#)

💡 AWS SDK for PHP

- [AWS SDK for PHP home](#)
- [PHP developer guide](#)
- [AWS SDK for PHP on GitHub](#)

To Delete the application and Environment (Select it and in **Action** -Delete/Terminate : give conformation)

Experiment No - 6

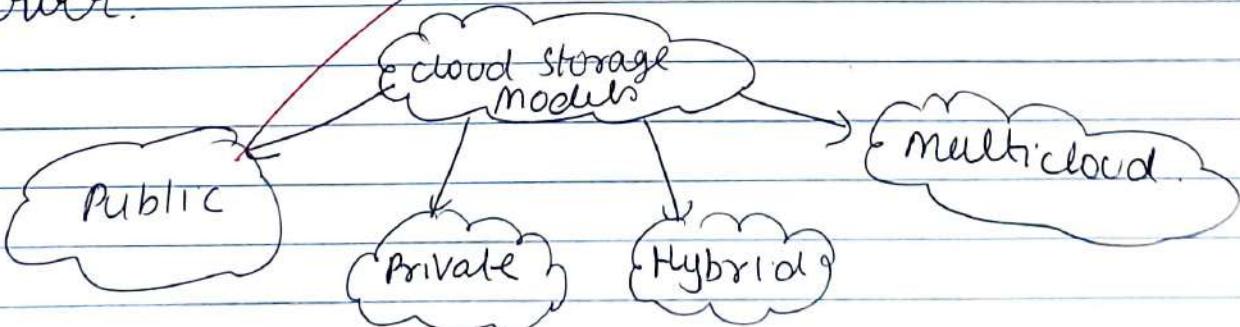
Aim :- To study and implement storage as a service using own cloud.

Theory :-

Explain concept of Cloud storage :-

- Cloud storage is a mode of computer data storage in which digital data is stored on servers in off-site locations. The services are maintained by a third-party provider who is responsible for hosting, managing and securing data stored on its services. It is always accessible via public or private internet connections.

- Cloud storage enables organization to store, access and maintain data so that they do not need to own and operate their own data centers, moving expenses from a capital expenditure model to operational. Users upload data to services via an internet connection, where it is saved on virtual machines on physical server.



Advantages of Cloud Storage :-

- Total cost of ownership
- Elasticity and Scalability
- Storing and Access Flexibility
- Robust Security

Disadvantages of Cloud Storage

- Delays in traffic due to internet connections.
- Relinquishment of control over access and management of data
- Public cloud providers aim to ensure continuous availability, outages sometimes do occur, making stored data unavailable.
- Explain Owncloud and its features:-

Owncloud gives you universal access to your files through a web interface or WebDAV. It also provides a platform to easily view & sync your contacts, calendars and bookmarks across all your devices and enables basic editing right on the web. Installation has minimal server requirement, doesn't need special permissions & is quick.

Features:-

- ① Access your stored files, folders, contacts, photo galleries, calendars and more from your mobile device, your desktop or a web browser.
 - ② Keep your files, contacts, photo galleries, calendar and more synchronized amongst your devices.
 - ③ Share your data with others publicly or privately. It is your data, do what you want with it.
 - ④ Own cloud automatically saves old file versions; you configure how much to save.
 - ⑤ With encryption application, all files stored on own cloud services are encrypted to your password.
 - ⑥ You can easily move your owncloud user account between own cloud instances and have a backup ready when you need it.
- (2) Advantages and limitation of storage as a service.

Shirish Shetty 2103164 C32

Advantages of STaaS:-

- The monthly or annual costs of STaaS are usually much more affordable than expenses of installing physical infrastructure.
- STaaS allow clients to store data in different location to improve chances of positive disaster recovery results.

Disadvantages of STaaS:-

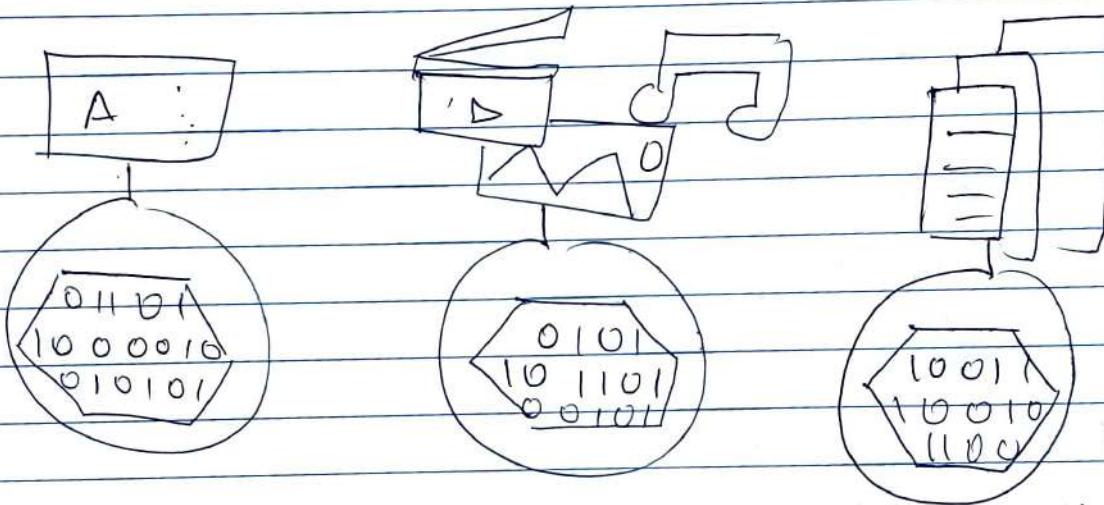
- If an organization uses STaaS for long enough, the subscription cost may exceed the installation cost of physical infrastructure.
- Provides open cloud infrastructure that varies. clients can use, meaning customization option may be limited.

Q Explain different types of storage like object storage, block level storage etc. -

~~storage solution can be broadly categorized based on how data is organized, accessed & managed. Hence are the main types.~~

① Object Storage

- Structure :- Data is managed as object within a flat namespace called a bucket. Each object includes the data itself metadata and globally unique identifiers.
- User Case :- Ideal for storing unstructured data like photos, video and backup.
- Example :- Amazon S3, Google Cloud Storage, Microsoft Azure Blob storage.

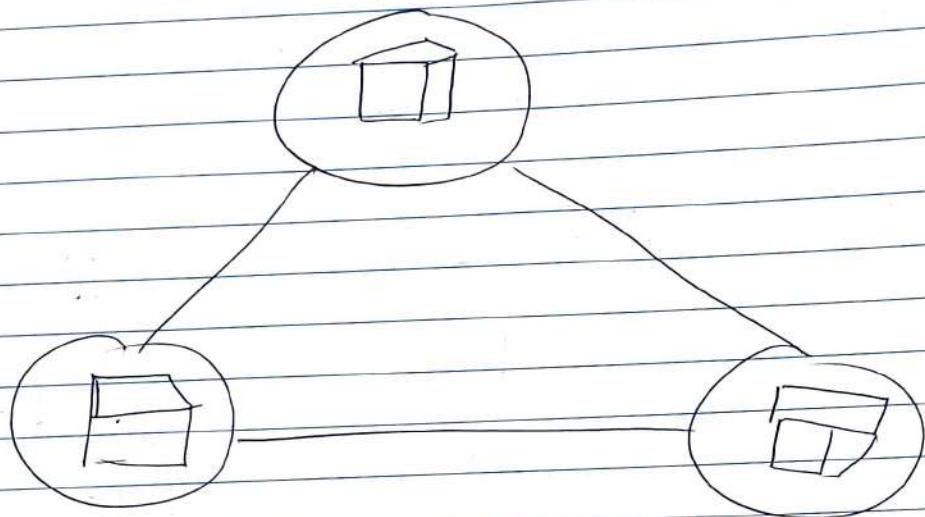


② Block storage :-

- i) Structure :- Data is divided into fixed sized block, each with unique identifier. Block storage devices are attached to source or virtual machine, appearing to operating system as local disk device

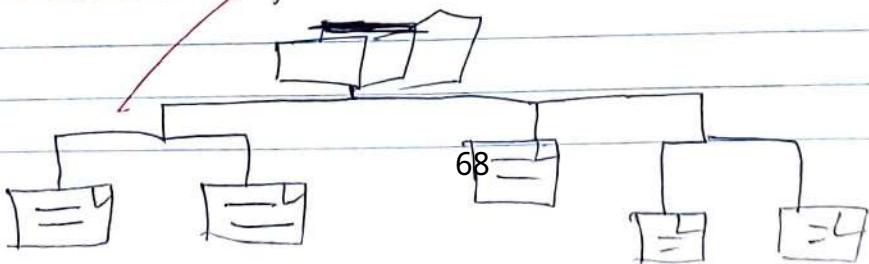
Use cases :- Ideal for DB, file system & applications that require high performance and low latency

Example :- Amazon elastic Blockstore (EBS), Google Persistent Disk, Microsoft Azure Disk storage



③ File storage:-

- Structure - Data is stored in files within hierarchical file and directory structure, similar to traditional file system on computer
- Use cases - Suitable for application & workloads that require shared file access, such as file sharing, content management system
- Example - Amazon Elastic File System, Google Cloud Filestore, Microsoft Azure files

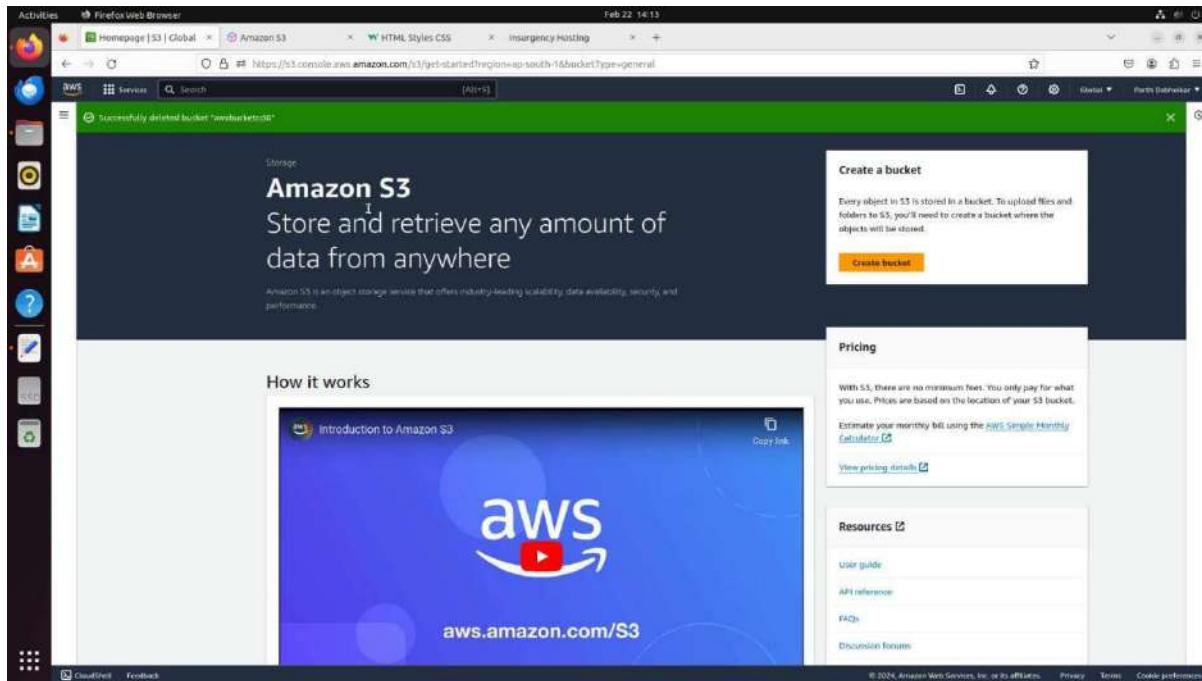


- Q.) Explain 5 popular storage as a services vendors along with their service(S3 is mandatory).
- 1.) Amazon Simple Storage Service(S3) :- Amazon offer a range of storage solution for businesses, including Amazon S3, an industry-leading product that help companies scale storage resource efficiently.
 - 2.) Drop box :- Dropbox offer a suite of product to help with efficient data storage & organization, streamlined document work-flows, automatic backup and restoration.
 - 3.) Google workspace :- Google offer enterprise-grade cloud storage services with Google workspace including a wide range of essential business application that make storing & sharing data easier with same platform.
 - 4.) IBM cloud object storage :- IBM offer different cloud storage solution to support growing business needs.
 - 5.) Microsoft cloud :- Microsoft cloud is a performed solution for a data storage used by more than 85% of fortune 500 companies.

EXPERIMENT NO: 6

AIM: To study and implement storage as a service using Own Cloud/AWS S3.

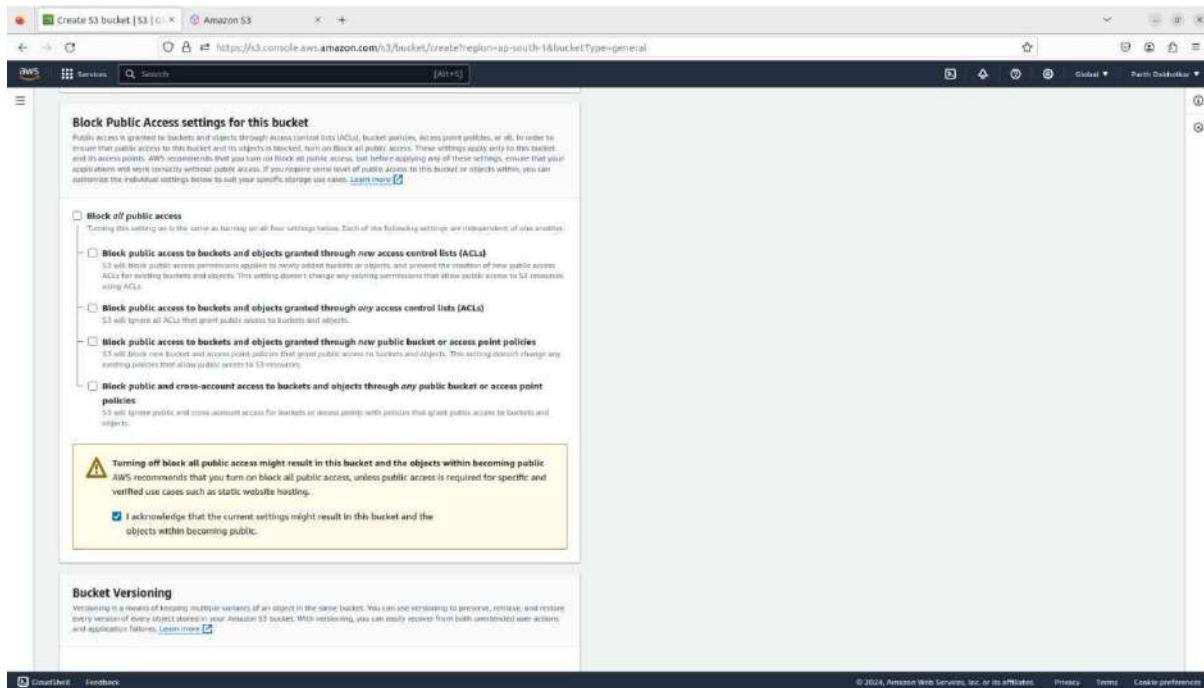
Step-1: Click on create bucket.



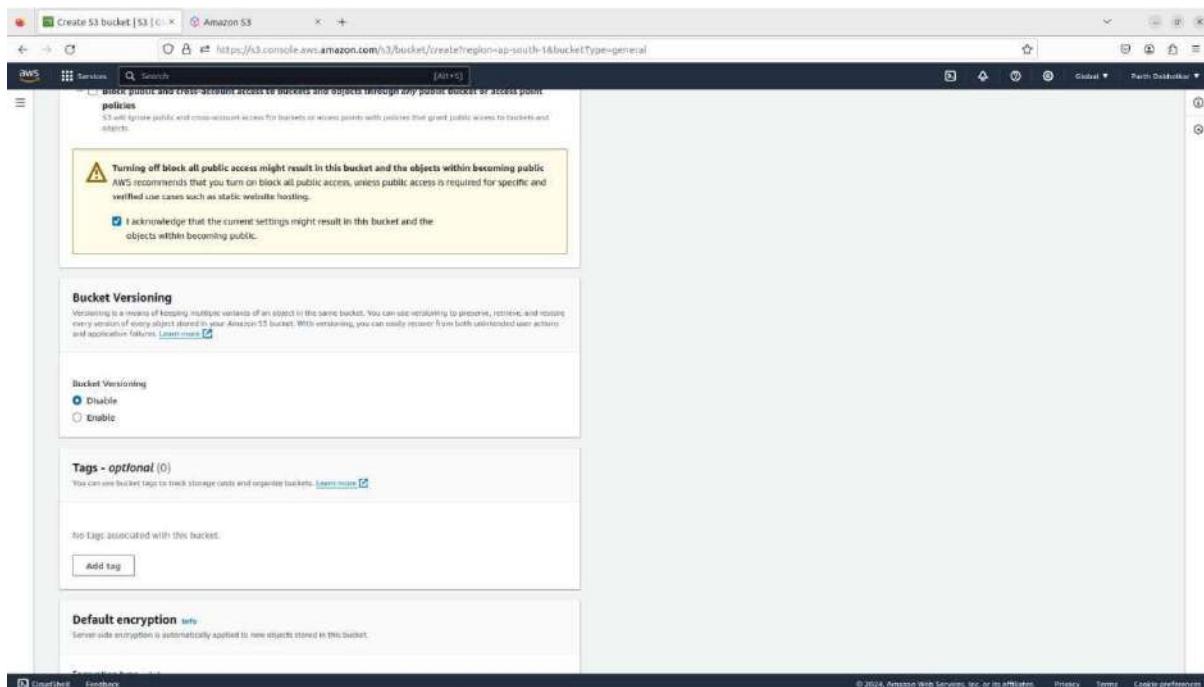
Step-2: Give Bucket name & select region for storage. Keep object ownership setting as ACLs Disabled as by-default.

The screenshot shows a browser window with the URL <https://s3.console.aws.amazon.com/s3/bucket/create?region=ap-south-1&bucketType=general>. The page title is 'Create bucket'. It has two main sections: 'General configuration' and 'Object Ownership'. In 'General configuration', the 'AWS Region' is set to 'Asia Pacific (Mumbai) ap-south-1', the 'Bucket name' is 'AwsbucketCCL6', and the 'Copy settings from existing bucket - optional' section is empty. In 'Object Ownership', the 'Object ownership info' section states 'Object ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.' It shows two options: 'ACLs disabled (recommended)' (selected) and 'ACLs enabled'. The 'Object ownership' dropdown is set to 'Bucket owner enforced'. At the bottom, there is a 'Block Public Access settings for this bucket' section with a note about public access through various policies. The footer includes links for 'Unsubscribe' and 'Feedback'.

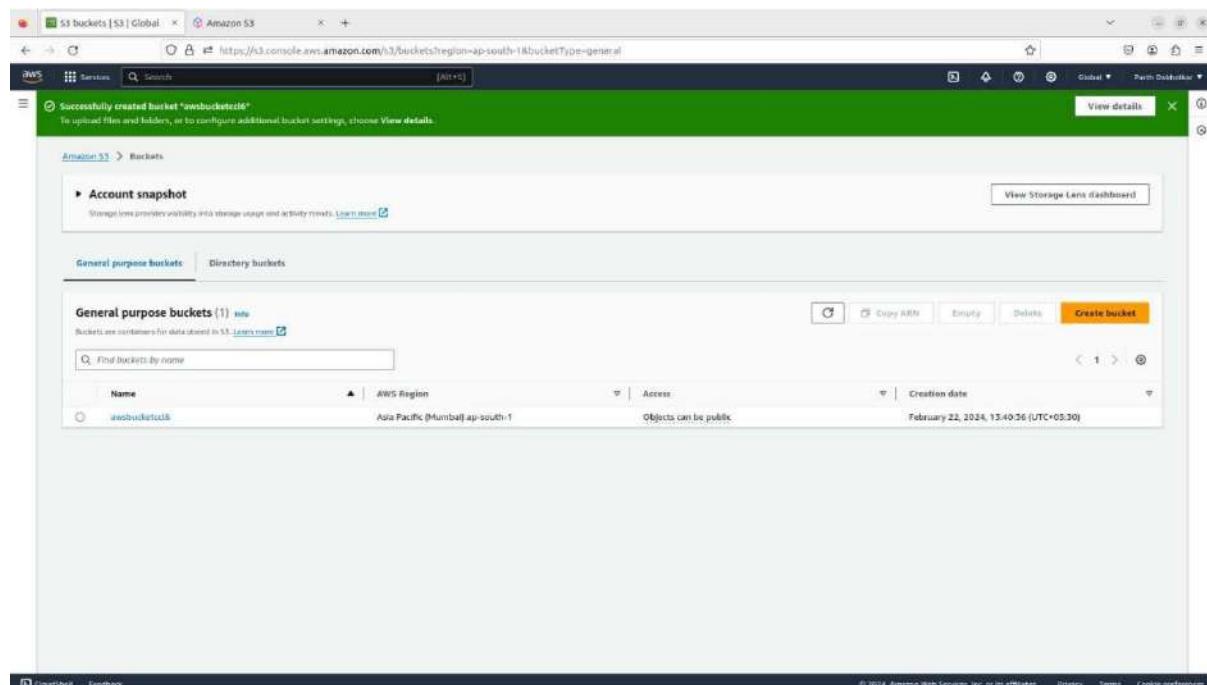
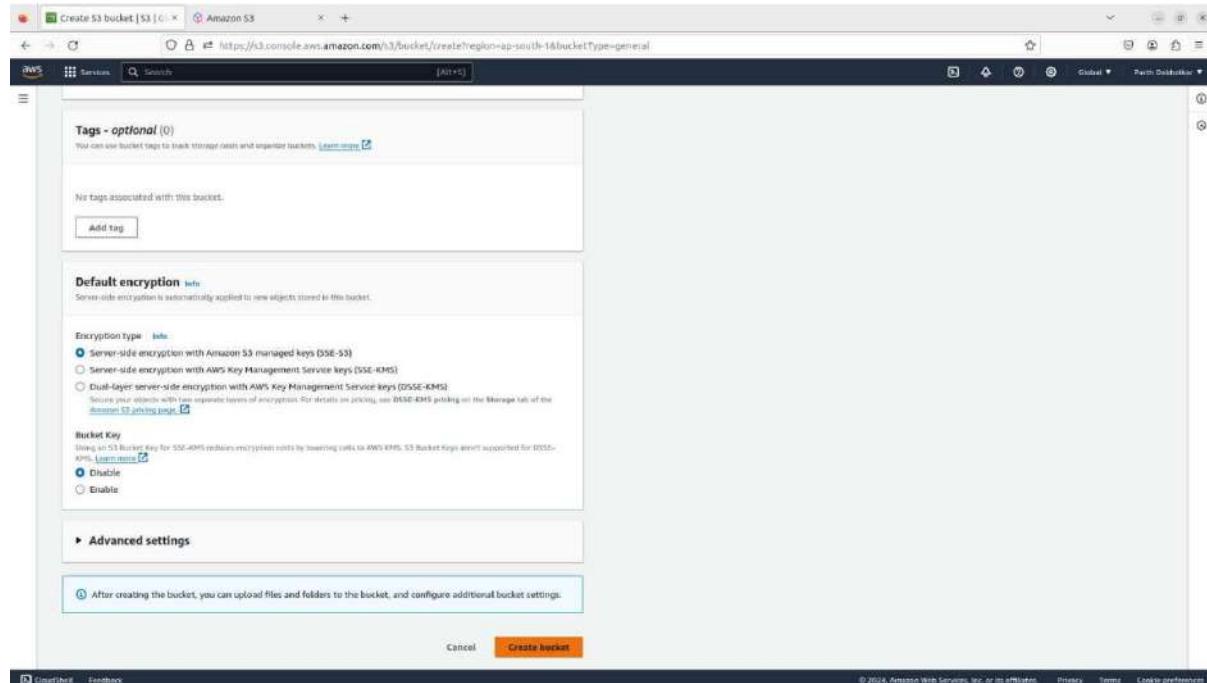
Step-3: Disable block all public access checkbox. Select the checkbox for Turning off block all public access might result in this bucket and the objects within becoming public.



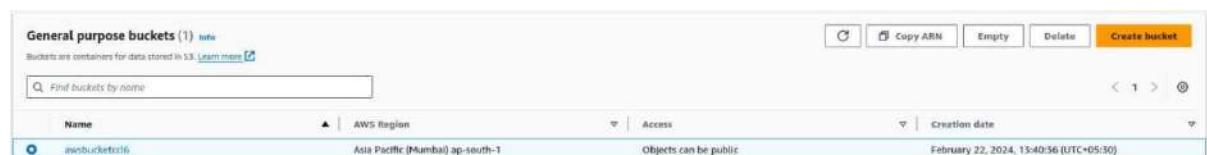
Step-4: Keep bucket versioning as disabled and add tags if required.



Step-5: Keep default encryption disabled and click on create bucket button. Your Bucket will be successfully created.



Step-6: Now click on the bucket that you have created.

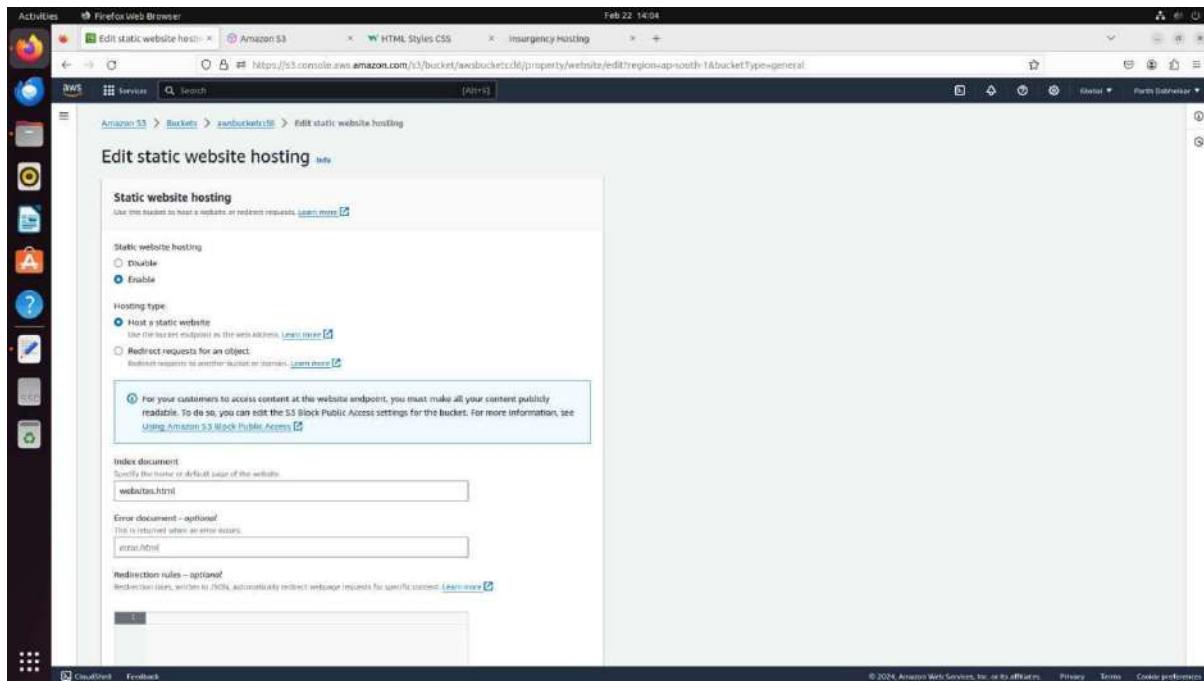


Step-7: You can either create a folder here or upload an existing file in the bucket. now click on upload button and click on add files button browse your local machine and select which file you need to upload on S3 next click on upload button at bottom right end. You can check the upload status on your screen.

The screenshot shows the AWS S3 console's 'Upload' interface. In the 'Files and folders' section, there is one item: 'websites.html'. Below it are 'Remove', 'Add files', and 'Add folder' buttons. The 'Destination' section shows 's3://newbuckettest00'. Under 'Permissions', there is an option to 'Grant public access and access to other AWS accounts...'. At the bottom right is a large orange 'Upload' button.

The screenshot shows the AWS S3 console after the upload has completed. A green header bar says 'Upload succeeded'. Below it, a summary table shows the upload details: Destination 's3://newbuckettest00' with 1 file (712.0 B) [100.00%] succeeded. The 'Files and folders' section lists 'websites.html' with a size of 712.0 B and a status of 'Succeeded'. The 'File' column shows a green checkmark icon.

Step-8: Select properties and scroll down to Static website hosting option which is disabled now click on Edit option on right side. Enable the radio button and specify the file name in Index document which you have added in S3.



Step-9: In bucket policy click on Edit option. After clicking on edit button paste the following code in bucket policy :

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "PublicReadGetObject",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": [  
                "s3:GetObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::Bucket-Name/*"  
            ]  
        }  
    ]}
```

]
}

Note-Make sure that you add your bucket name in the code above

Scroll down and click on Save Changes button

The screenshot shows the 'Edit bucket policy' page in the AWS S3 console. The policy JSON is displayed:

```
1  "Version": "2012-10-17",
2  "Statement": [
3    {
4      "Effect": "Allow",
5      "Principal": "*",
6      "Action": "GetObject",
7      "Resource": "arn:aws:s3:::awsbucketcc16/*"
8    }
9  ]
```

On the right, there's a 'Select statement' dropdown and a 'Add new statement' button.

The screenshot shows the 'Permissions' tab of the AWS S3 bucket configuration. A green success message says 'Successfully edited bucket policy.' Below it, the 'Block public access (bucket settings)' section is shown:

Block public access (bucket settings)
Off
Individual Block Public Access settings for this bucket

The 'Bucket policy' section shows the same JSON policy as the previous screenshot.

Step-10: Open your html file and click on Object URL.

Activities Firefox Web Browser Feb 22 14:10

websites.html - Object Inspector Amazon S3 HTML Styles CSS Insurgency Hosting

https://s3.console.aws.amazon.com/s3/object/awsbucketcc0?region=ap-south-1&bucketType=general&prefix=websites.html

Amazon S3 > Buckets > awsbucketcc0 > websites.html

websites.html

Properties Permissions Versions

Object overview

Owner: aafba70103540b011ec4dafe0f0150774707712fe0cdff9458935d54009605050da

AWS Region: Asia Pacific (Mumbai) ap-south-1

Last modified: February 22, 2024, 14:00:25 (UTC+05:30)

Size: 772.0 B

Type: html

Key: websites.html

S3 URI: s3://awsbucketcc0.s3.ap-south-1.amazonaws.com/websites.html

Amazon Resource Name (ARN): arn:aws:s3:::awsbucketcc0/websites.html

Etag tag (Etag): f0dd903b1cf0e1b3f535604606c54ac

Object URL: https://awsbucketcc0.s3.ap-south-1.amazonaws.com/websites.html

Object management overview

The following bucket properties and object management configurations impact the behavior of this object.

Bucket properties

Bucket Versioning: When enabled, multiple versions of an object can be stored in the bucket to easily recover from unanticipated user action and application failure. **Disabled**

Management configurations

Replication status: None. A replication rule is applied for an object. The replication status indicates the progress of the operation.

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookies preferences

Activities Firefox Web Browser Feb 22 14:10

Insurgency Hosting Amazon S3 HTML Styles CSS Insurgency Hosting

https://awsbucketcc0.s3.ap-south-1.amazonaws.com/websites.html

Welcome to my hosted website!!

Hosting on Amazon S3 Object Storage Service

Click Me

Step-11: Now begin with the bucket deletion process. Now for delete files click on checkbox of your file and then click on Delete Button. Write permanently delete and click on delete object button.

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with various AWS services like Access Grants, Object Lambda Access Points, Multi-Region Access Points, IAM Access Analyzer for S3, Storage Lens, Dashboards, and AWS Organizations settings. The main area shows a list of objects in the 'awsbucketcc16' bucket. There is one object named 'websites.html' listed. At the top right of the object list, there are several buttons: Copy S3 URL, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload. The 'Delete' button is highlighted in red.

This screenshot shows the 'Delete objects' confirmation dialog box overlaid on the S3 console. The dialog contains a warning message about deleting objects from a folder and a note that deleted objects can't be undone. Below this is a 'Specified objects' section showing the same 'websites.html' file. At the bottom, there's a 'Permanently delete objects?' section with a text input field containing 'permanently delete' and a large orange 'Delete objects' button.

The screenshot shows the AWS S3 console with a success message: "Successfully deleted objects" and "View details below". Below this, a summary table shows:

Name	Successfully deleted	Failed to delete
sh://wwwbucket010	1 object, 712.0 B	0 objects

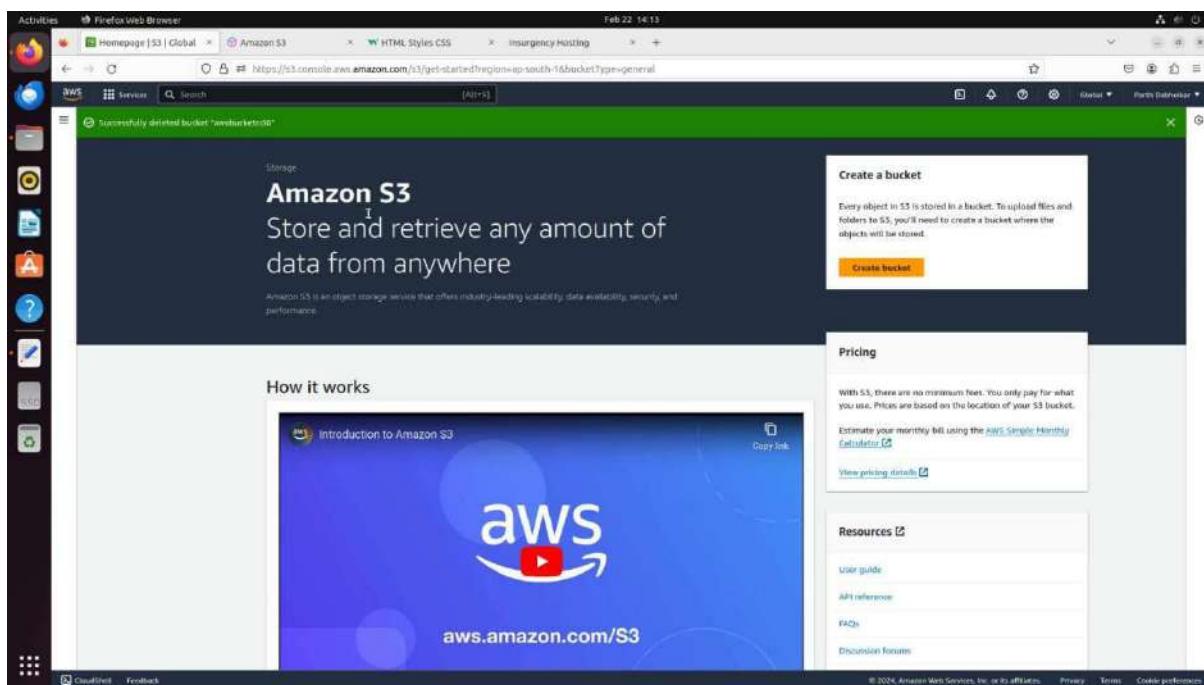
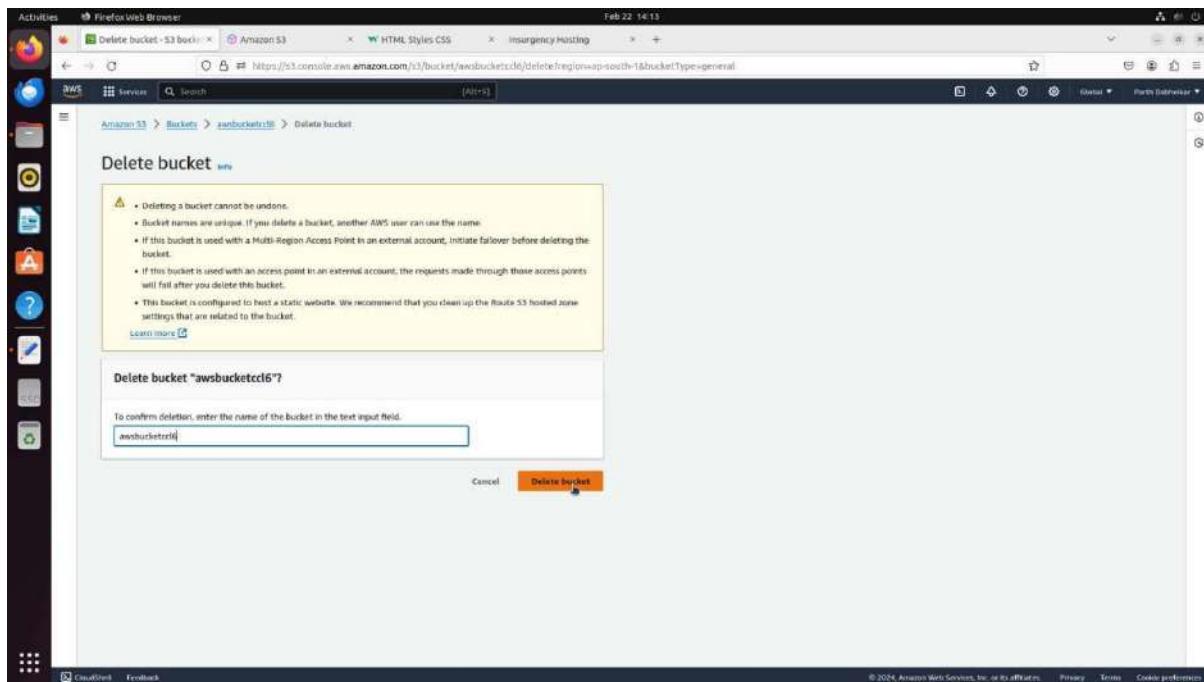
Below the summary, there are tabs for "Failed to delete" (0) and "Configuration". The "Failed to delete" tab shows a table with no data: "No objects failed to delete".

Step-12: Now come to Amazon S3 tab and select your bucket and then click on delete button. Write down your bucket name in delete bucket tab and click on delete button at bottom right. You can see that the bucket is deleted.

The screenshot shows the AWS S3 Buckets page. On the left, the navigation pane includes "Amazon S3", "Buckets", "Storage Lens", and "AWS Marketplace for S3". The main area displays an "Account snapshot" and a table of "General purpose buckets". The table shows one bucket named "insurancetest01" with the following details:

Name	AWS Region	Access	Creation date
insurancetest01	Asia Pacific (Mumbai) ap-south-1	Public	February 22, 2024, 11:40:56 (UTC+05:30)

At the top right of the table, there are buttons for "Copy ARN", "Empty", "Delete", and "Create bucket".



Experiment - 7

Aim :- To study & implement Identity & Access Management (IAM) practices on AWS

Theory :- AWS Identity & Access Management (IAM) is a web service that helps you securely control access to AWS resources. With IAM, you can centrally manage permission that controls which AWS resources user can access. You can control who is authenticated (signed in) & authorized (has permissions) to use resources.

* Features of IAM

→ Shared Access to Your AWS Account

- You can grant other people permission to administer & use resources in your AWS account without having to share your password or access key.

→ Granular permission

- . You can grant different permission to different people for different resources
- for eg :- You give complete access to ECS & S3 in AWS, But give ready only access to other

→ Multi - Factor Authentication (MFA)

- You can add 2 factors authentication to your account & to individual user for extra security.

- With MFA you or your users must provide not only password or AccessKey to work with your account, but also a code from specially configured device.

* Accessing IAM

→ AWS management console

- The console is browser-based interface to manage IAM & AWS resources. For more information about accessing IAM through the console.

→ AWS command line tool

- You can use the AWS command line tool to issue commands at your system command line to perform IAM & AWS tasks.

Command Line Tool

AWS CLI

(command

line interface)

AWS Tools for

window power shell

- After signing into console, you can use AWS CloudShell from your browser to run CLI or SDK commands

→ AWS SDK

- AWS provides SDK that consist of that
 - consists of libraries & sample code for various programming language & platform
- Easy to access IAM & AWS.

Q) Compare between root user & IAM users

Feature	Root user	IAM user
① Access ⇒	Has complete access to all AWS resources & services	Acess is limited to what permission are explicitly granted.
② Sign In ⇒	uses email address & password	Specify URI or access key for API I need.
③	using the root user for everyday task is discouraged	Secure approach to managing AWS resources.
④ Security & Recommended	to activate multi-factor authentication for enhance security	Recommended for secure resource management within AWS

Roles vs Policies

Parameter

Roles

Policies

Definition Define a set of permission that grant access to resources & actions.

Define rules & restriction that control access to resource and govern the behaviour of user.

Purpose To elegant access without showing security credentials.

To specify allocation of detailed actions or resources.

flexibility Enables temporary access for different scenarios.

Access control through detailed permission.

Practices used for temporary user account or application specific access.

Follow the principle of least privilege regularly review and refine.

Explain inline & custom policies in AWS.

~~Inline policies :-~~

- ① Defn :- policies embedded directly into IAM entities for exclusive associations.
- ② Use cases :- ensuring specific permission for individual identities.

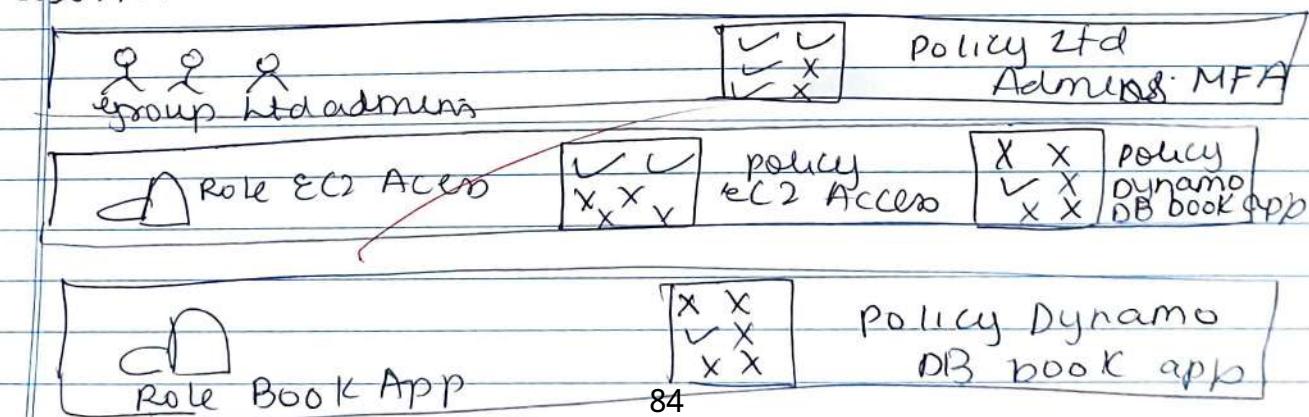
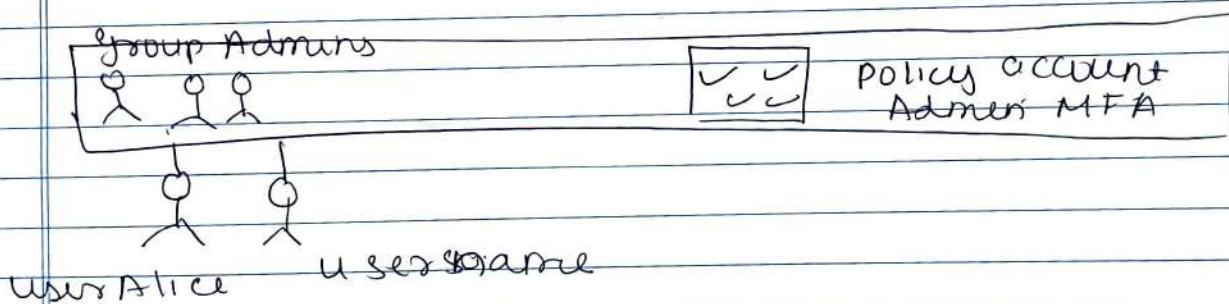
④ temporary Accs :- ideal for precise temporary permission

Custom Policies in AWS :-

- ① Definition:- standalone policies managed within your AWS Account.
- ② Administration:- customizable and updatable to meet evolving needs.
- ③ Flexibility :- Granular control over permission and access level.
- ④ Best Practices :- Recommended for Security & least privilege Accs.

Inline Policies

example AWS account



Explain MFA (multifactor authentication) in AWS.

Multifactor authentication in AWS is a security best practice that adds an extra layer of protection to user accounts beyond user password. With MFA enabled user can required to provide a second authentication factor in addition to their username and password, to access AWS resources. This second factor known as "something they have" can be a physical device like a security key or a virtual authentication app that generates time-based one-time passwords. By combining multiple factors, MFA enhances security by significantly reducing the risk of unauthorized access to AWS account & resources.

(AS)
SF
18/3/2021

1)

Experiment - 8.

Aim:- To study & implement database as a service on SQL database using AWS RDS.

Theory :- Database as a service (DBaaS) is a fully managed way to have database hosted in the cloud.

- With significant benefits compared to on-premises database. E.g. more than maintenance and upgrades are handled by the service provider, allowing user to quickly benefit from database service. DBaaS is gaining popularity as organization transition to cloud databases. Some notable organization transition to cloud databases some notable DBaaS providers & service include:-

- ① IBM Cloud :- Offer managed database services like IBM Cloudant, a scalable cloud database ~~or~~ on Apache CouchDB & IBM Db2 a fully managed cloud DB with AI capabilities
- ② MongoDB Atlas :- A non-relational DB services that is easily scalable and offer everything required to operate a database in cloud, including provisioning, license support & maintenance.

③ Microsoft Azure and Google Cloud :- These providers manage infrastructure & platform services allowing clients to focus on building complex application

NoSQL / SQL supported by AWS :-

AWS provides a comprehensive range of SQL and NoSQL databases to address various data storage and management requirements. These databases offer high performance, scalability, security features & flexibility to meet diverse application needs.

* SQL databases :-

① Amazon Aurora :- A highly scalable relational database service compatible with MySQL and PostgreSQL, offering performance and availability benefits.

② Amazon Redshift :- A powerful data warehouse solution designed for running complex data analyses on large datasets efficiently.

* NoSQL database :-

① Amazon DocumentDB :- A fully managed NoSQL database service compatible with MongoDB, optimized for managing JSON data at scale.

- Q) Amazon DynamoDB :- A proprietary key value and document database service known for its single digit millisecond latency at any scale and global tablets for eg data replication across multiple AWS Regions.
- ③ Amazon Neptune - A graph database service ideal for managing relationships among data, supporting popular graph modules like property S

Explain following terms in AWS wrt DBaaS.

- ① Storage types :- In AWS DBaaS refer to different categories of database available to user, such as key value stores, document databases, time series, graph databases and ledger databases. Each type is optimized for specific data structure and access pattern providing flexibility in choosing the right database for diverse application requirements.

Endpoint :- The endpoint in AWS refer to URL that application use to connect to database instance. It serves as entry point for accessing the ~~database~~ source.

Read Replica :- Read Replica in AWS
 Read Replica : - Read Replica in AWS are copies of a source database that can be used for read-heavy workloads. They help offload read operations from primary databases improving performances.

Single AZ and multi AZ instances :- In AWS, DBaaS single availability zone (AZ) instance operate within a single data center for basic deployment scenarios. On other hand multi AZ instances span multiple availability zones across different data centers enhancing fault tolerance & availability.

Difference between RDS and Aurora

Parameter	Amazon RDS	Amazon Aurora
Database Engines Supported	Supports more database engines like MySQL, SQL, PostgreSQL and Oracle	Support MySQL and PostgreSQL compatible DB
Performance	Performance relies on the specific database engine being used	Generally out performance RDS especially for read intensive workload.

Bring:- Commercial alternatives
with pay-as-you-go pricing

Storage instance
type and usage

Replication:- Support read
replicas for various
database engines

Support up to
15 replicas
with millisecond
replication

Conclusion:- Successfully implemented database
as a service on SQL databases using AWS RDS

(A)

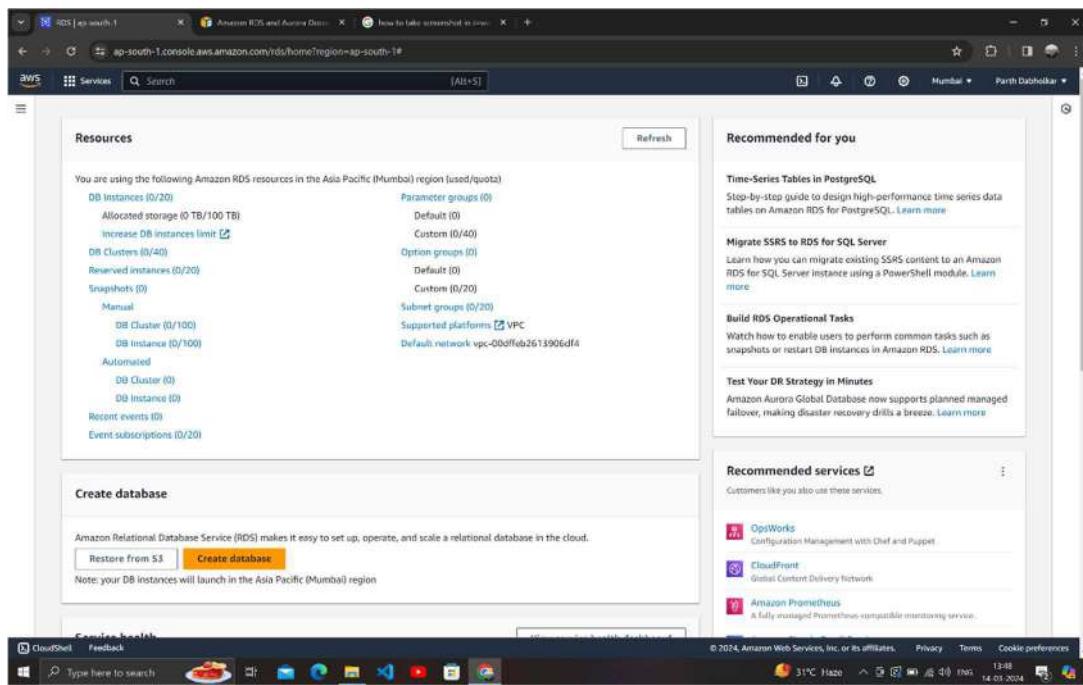
SI
26/3/2024

EXPERIMENT NO: 8

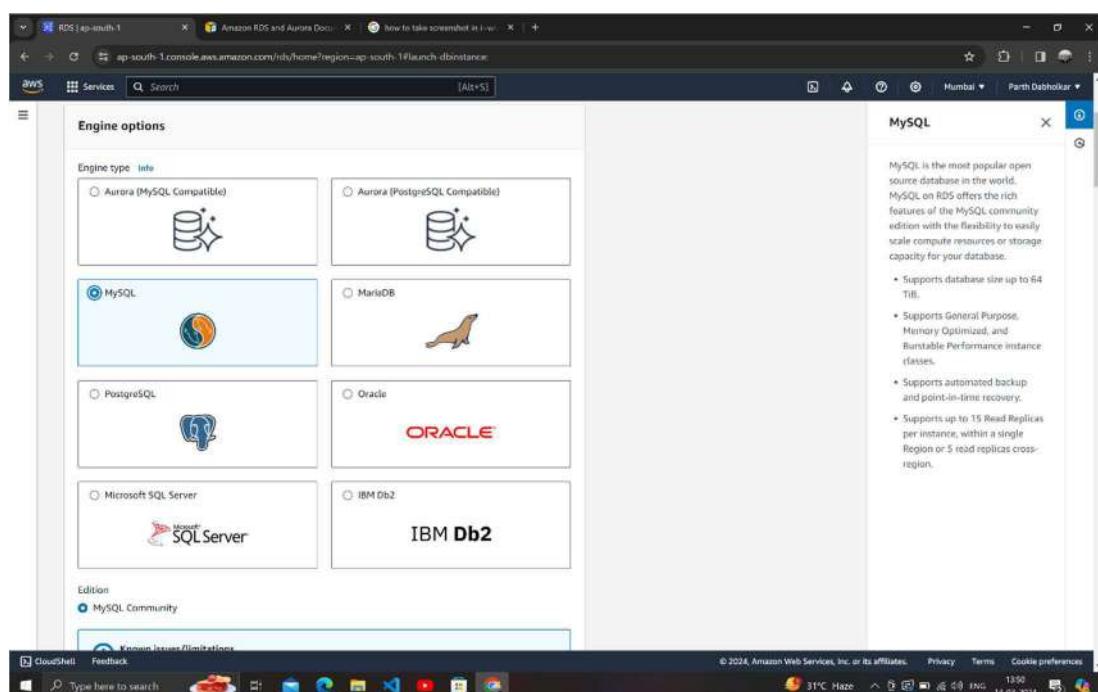
AIM: To study and implement database as a service on AWS RDS(Relational Database Services).

Steps to implement database as a service on AWS RDS is given below:

1. Login to Amazon AWS. Select the services section in the navbar. A dropdown appears. Select RDS in the dropdown menu. A new window appears. Select the create database option in the window.



2. A new window appears. Select the database engine to be used. Here we select the MySQL option. Also select the proper version of MySQL.



3. In the templates section, select the Free-Tier in-order to prevent any costs.

The screenshot shows the AWS RDS MySQL instance creation wizard. In the 'Templates' section, the 'Free tier' option is selected, which is highlighted with a blue border. The 'Free tier' description states: "Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS." To the right of the template selection, there is a sidebar titled 'MySQL' with a brief overview and a bulleted list of features.

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

4. In the settings section, enter the Database name, master password. Remember the password for further use.

The screenshot shows the 'Settings' section of the AWS RDS MySQL instance creation wizard. Under 'DB instance identifier', the value 'test-database-parth' is entered. Under 'Master username', the value 'admin' is entered. A note below states: "If you manage the master user credentials in Secrets Manager, some RDS features aren't supported." Below these fields are sections for generating a password and entering a master password twice.

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

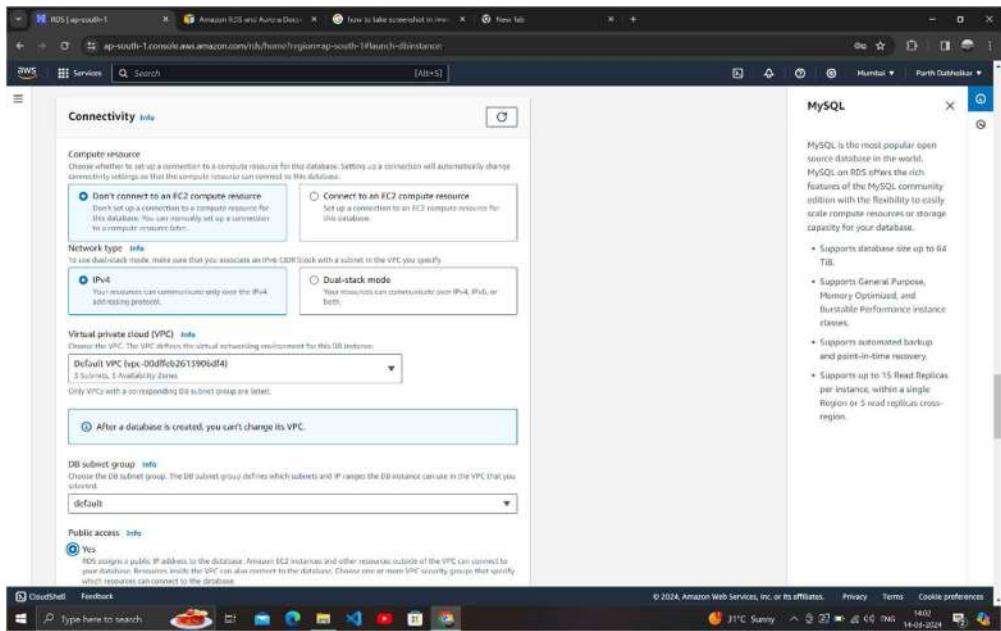
5. In instance selection, select db.t3.micro which consists of 2 CPUs and 1GB RAM.

The screenshot shows the AWS RDS console for launching a new DB instance. The 'Instance configuration' section is active, displaying the selected instance class 'db.t3.micro' (2 vCPUs, 1 GiB RAM, Network: 2,085 Mbps). The MySQL sidebar on the right provides a brief overview of MySQL, mentioning its popularity as an open-source database and its rich features like General Purpose, Memory Optimized, and Burstable Performance instance classes, automated backup, and support for up to 15 Read Replicas.

6. In the Storage section, do not change anything. Keep the default settings and proceed further.

The screenshot shows the 'Storage' configuration step in the AWS RDS console. The allocated storage is set to 20 GiB. A note in the storage section states: 'After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes.' The MySQL sidebar remains visible on the right side of the screen.

7. In the connectivity section select Do not connect to EC2 compute resources. Also set the public access to Yes. Let the other settings be as default. Select the create database option.



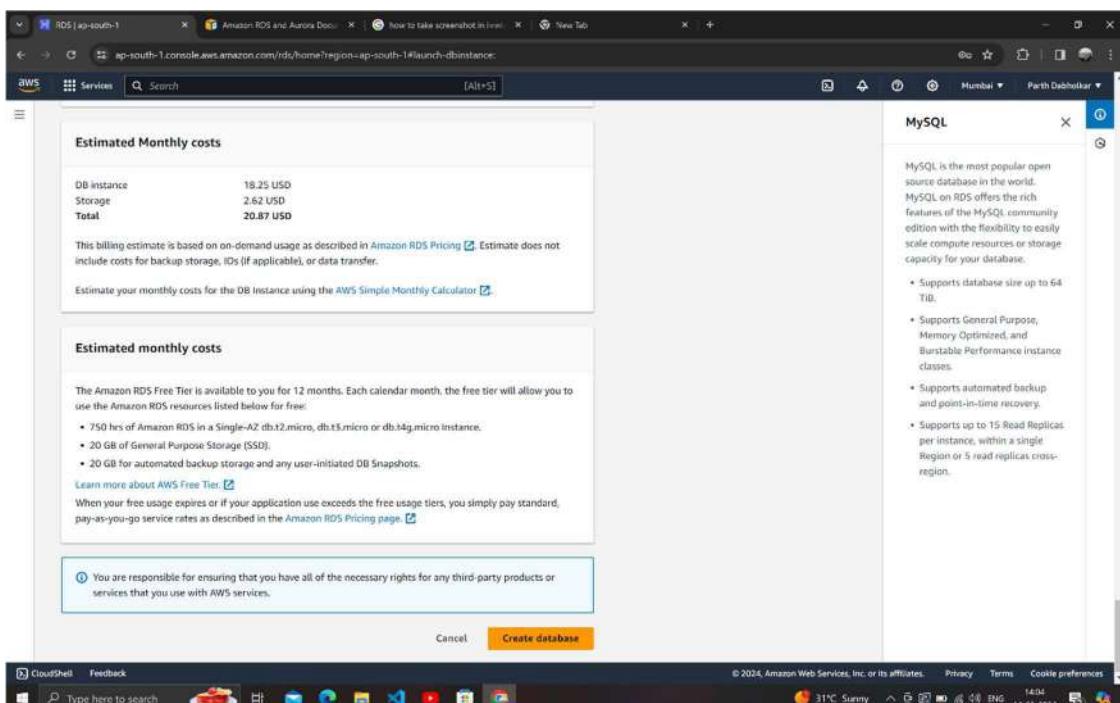
Public access Info

Yes

RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

No

RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

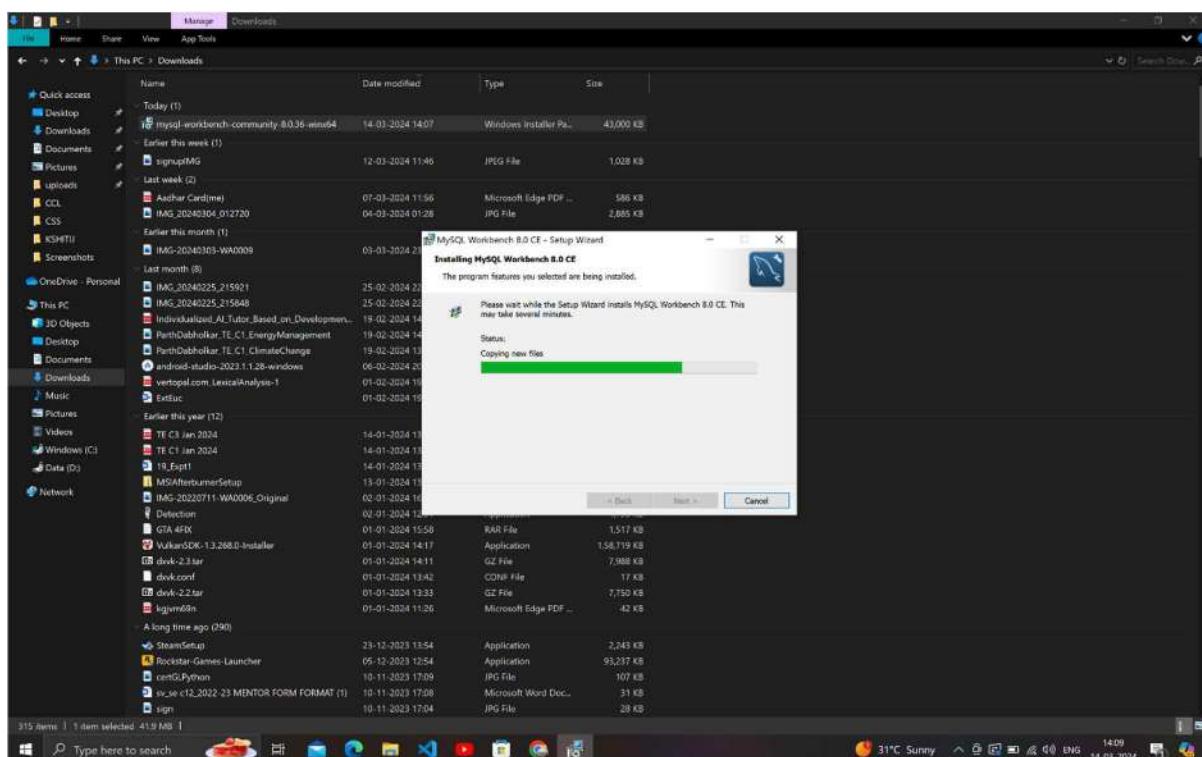
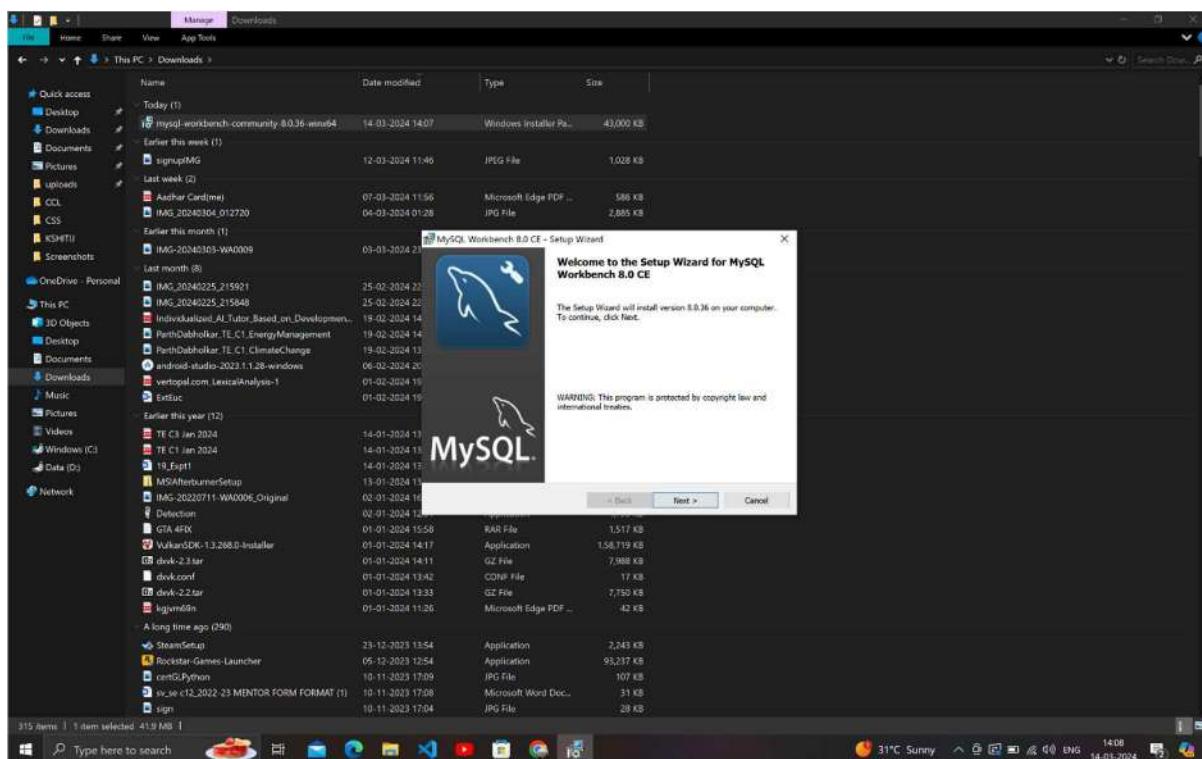


8. The creation of database will take some time. Until the database is created, go to Google and search for MySQL Workbench.

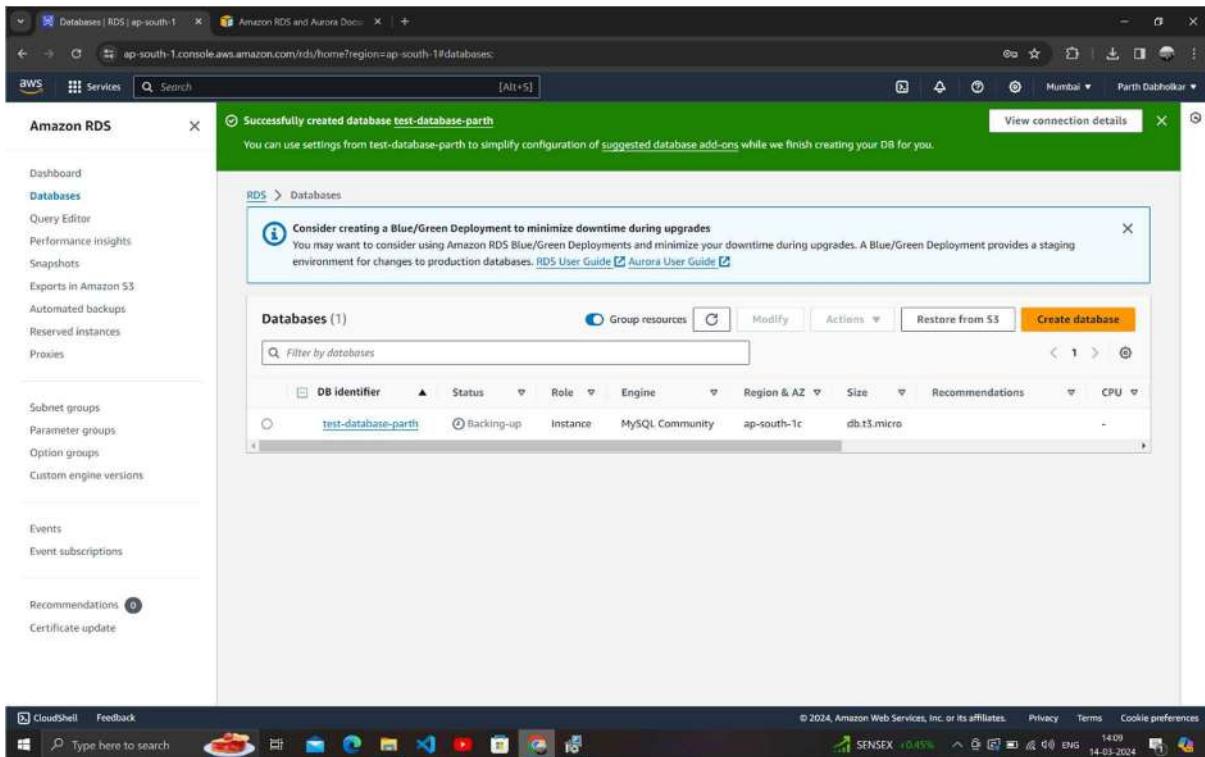
The screenshot shows the Amazon RDS console in a web browser. On the left, there's a sidebar with options like Dashboard, Databases, Performance Insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Events, Event subscriptions, Recommendations, and Certificate update. The main area is titled 'Creating database test-database-parth' with a note that it might take a few minutes to launch. Below this, there's a 'Databases (1)' section with a table. The table has columns: DB identifier, Status, Role, Engine, Region & AZ, Size, Recommendations, CPU, and Actions. One row is shown for 'test-database-parth' with a status of 'Creating', engine 'MySQL Community', size 'db.t3.micro', and other details. A tooltip suggests considering a Blue/Green deployment for downtime minimization. At the bottom right of the main area, there's a 'Create database' button. The browser's address bar shows the URL for the RDS home page. The taskbar at the bottom of the screen shows various pinned icons.

The screenshot shows a Google search results page for 'mysql workbench'. The first result is a link to the MySQL website with the title 'MySQL Workbench'. The second result is a snippet from Wikipedia with the title 'MySQL Workbench' and a detailed description of what it is and its features. Other search results include links for MySQL Workbench manual, developer zone, and migration tools. The browser's address bar shows the search query. The taskbar at the bottom of the screen shows various pinned icons.

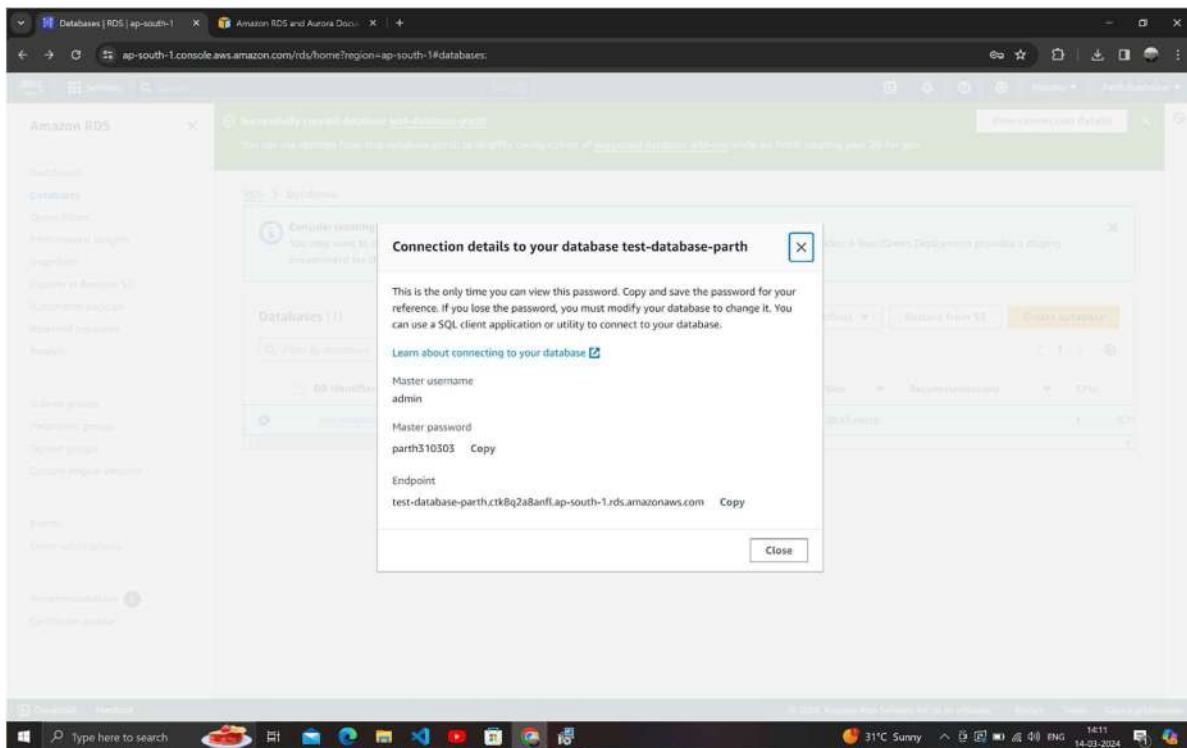
9. Select the link and click download. In MySQL Community download, select the download option in the bottom section of the screen. You will be redirected to a new screen. Select “No just continue my download” and wait for the download to be completed. After installation select finish and close the setup.



10. Go back to AWS portal and check whether the database is created. A green alert will be displayed.



11. Click on view connection details to see the details of the created RDS database in AWS. The details will contain a Host endpoint and master password which will be useful in connection of cloud database to MySQL workbench.



12. Copy the Host endpoint from the Connectivity and Security section. This link will be used in database connection with MySQL workbench.

The screenshot shows the AWS RDS console for an instance named 'ap-south-1'. The left sidebar is collapsed. The main area has tabs: Connectivity & security (selected), Monitoring, Logs & events, Configuration, Maintenance & backups, Tags, and Recommendations. Under 'Connectivity & security', there are three sections: Endpoint & port, Networking, and Security. The Endpoint section shows 'Endpoint' as 'test-database-parth.ctk02a8anfl.ap-south-1.rds.amazonaws.com' and 'Port' as '3306'. The Networking section shows 'Availability Zone' as 'ap-south-1c' and 'VPC' as 'vpc-00dfbeb2613906df4'. The Security section shows 'VPC security groups' as 'default (sg-0b25ef9c0ff06611d)' with 'Active' checked, and 'Publicly accessible' set to 'Yes'. Other details like Subnets and Certificate authority are also listed. Below this, a 'Connected compute resources' section is shown with a count of 0. The bottom of the screen shows the Windows taskbar with various icons and system status.

13. Open MySQL workbench. Click on plus sign beside MySQL Connectivity. A new window will pop-up. Fill the connection name as the name of the database. In the hostname, paste the endpoint link copied in earlier step. Set the port number to 3306. Also fill the username as "admin". Try to select test connection. It will not work.

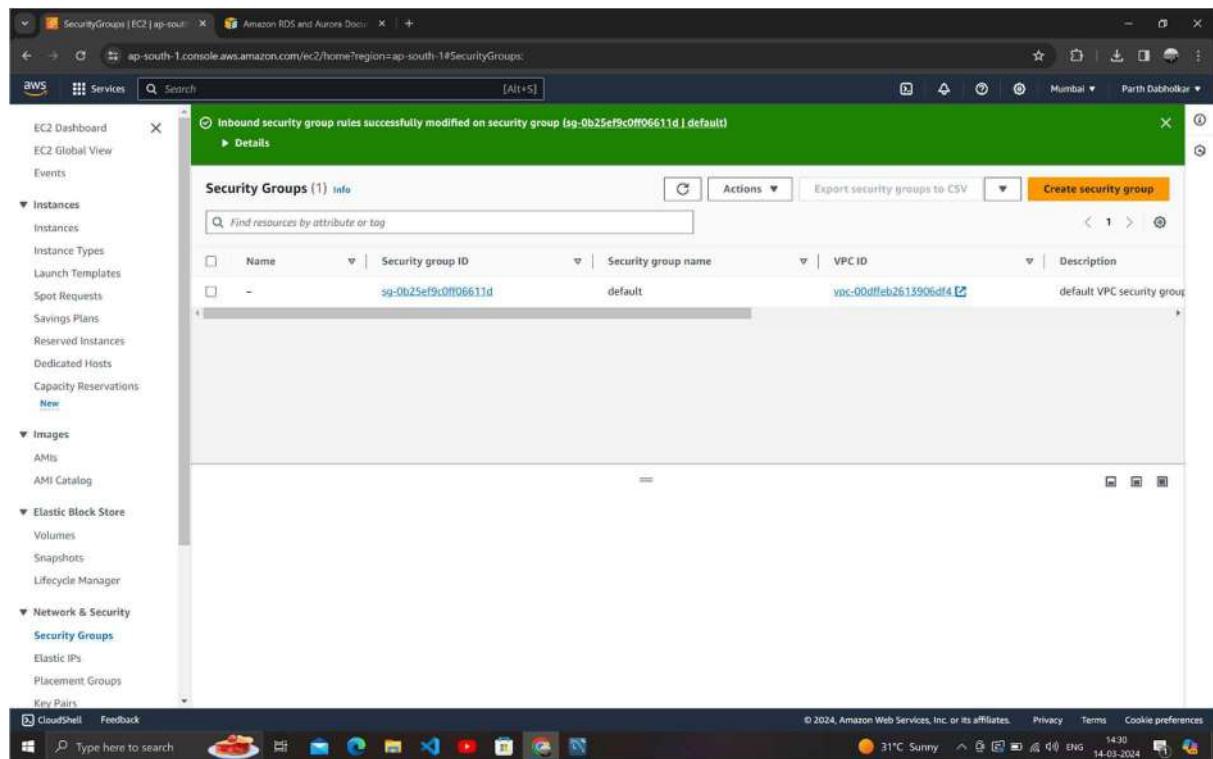
The screenshot shows the MySQL Workbench interface. On the left, there's a sidebar with icons for CloudShell, Feedback, and a search bar. The main area has tabs: File, Edit, View, Database, Tools, Scripting, Help. A central panel displays the message 'Welcome to MySQL Workbench'. Overlaid on this is a 'Setup New Connection' dialog box. The dialog has fields for 'Connection Name' (set to 'test-database-parth'), 'Connector Method' (set to 'Standard (TCP/IP)'), 'Hostname' (set to 'test-database-parth.ctk02a8anfl.ap-south-1.rds.amazonaws.com'), 'Port' (set to '3306'), 'Username' (set to 'admin'), and 'Password' (set to 'Store in Vault...'). Other tabs in the dialog include 'Advanced' and 'Schemas'. At the bottom of the dialog are buttons for 'Configure Server Management...', 'Test Connection', 'Cancel', and 'OK'. The bottom of the screen shows the Windows taskbar with various icons and system status.

14. Go to security groups in database. Select the security group present in the section. In the bottom section, select Inbound Rules and click Edit Inbound Rules.

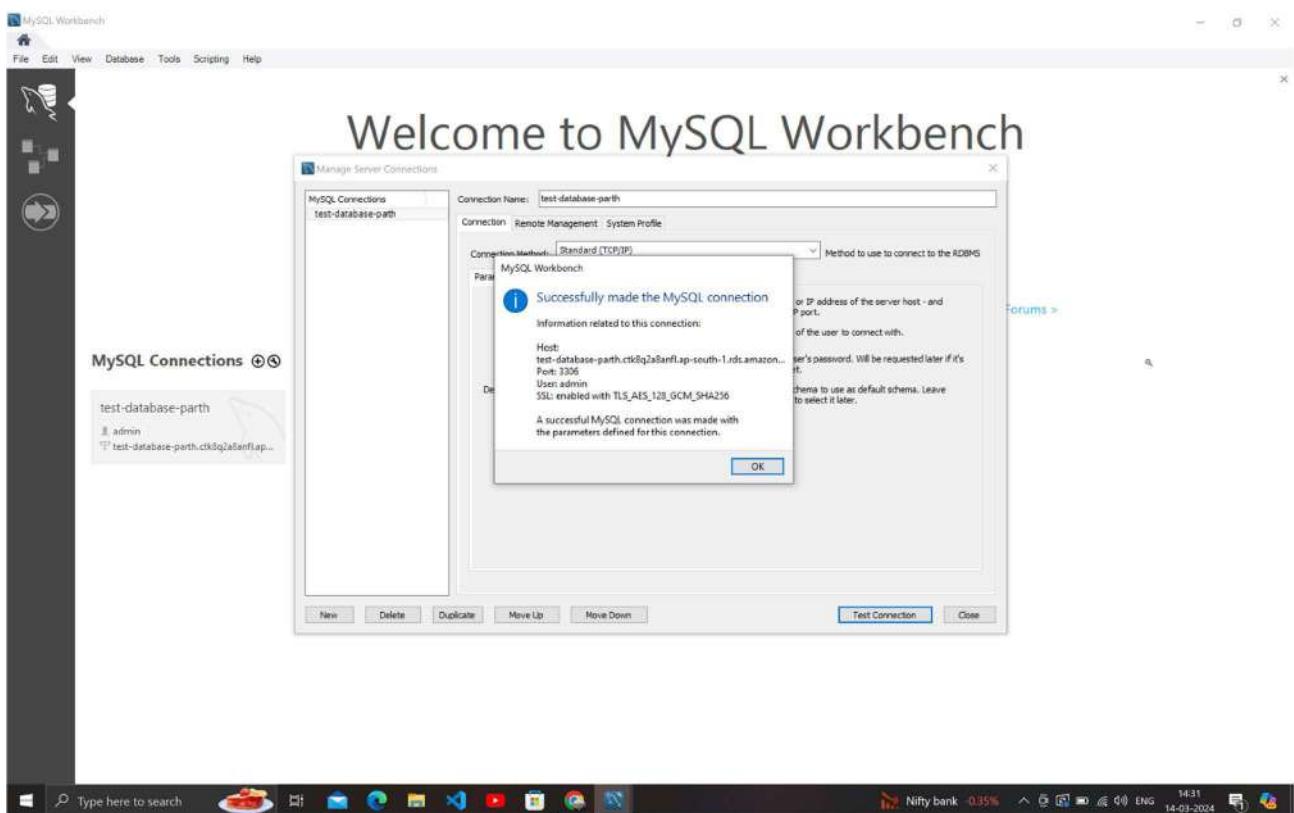
The screenshot shows the AWS EC2 console with the 'Security Groups' page open. A single security group, 'sg-0b25ef9c0ff06611d', is listed under the 'default' VPC. The 'Inbound rules' tab is selected, displaying one rule: 'sg-0cc34ec00db5c23d5' allowing all traffic from anywhere ('0.0.0.0/0').

15. Select add rule. In the type field, select IPV4 – All traffic in order to allow anyone to access the database. In the source field select “Anywhere” for all ip addresses. Click Save Rules.

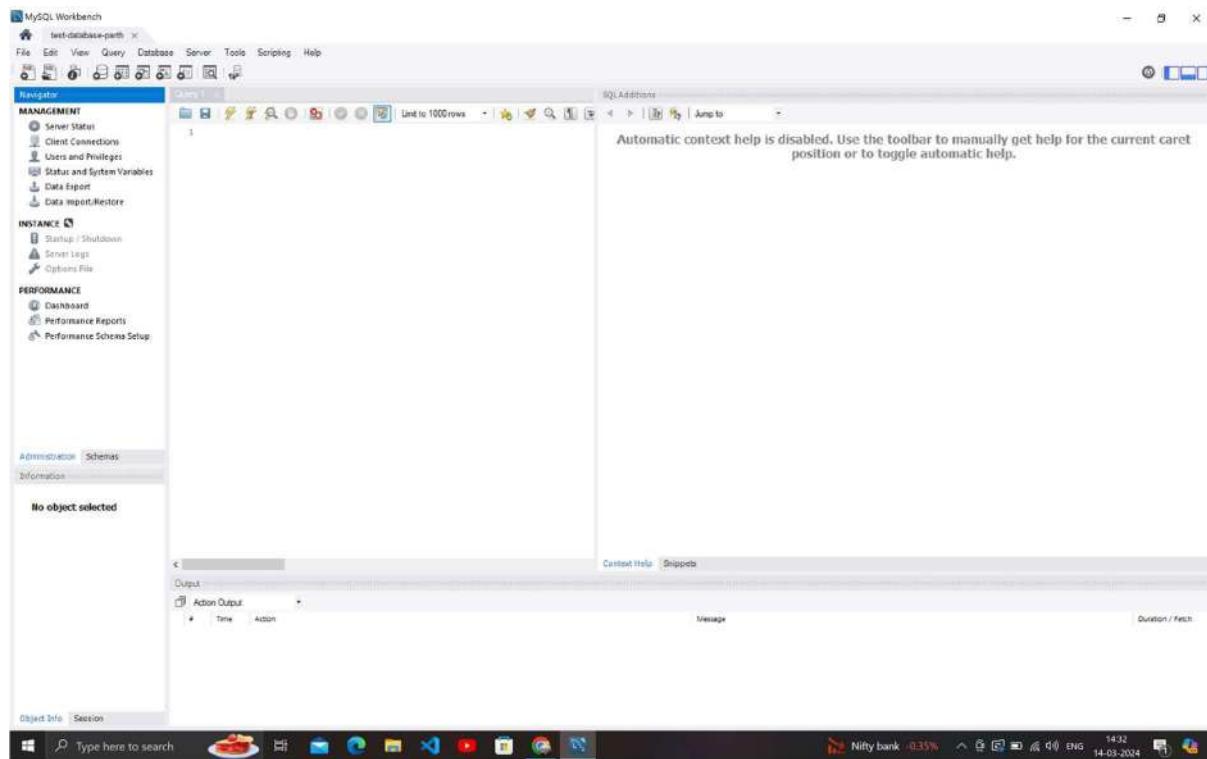
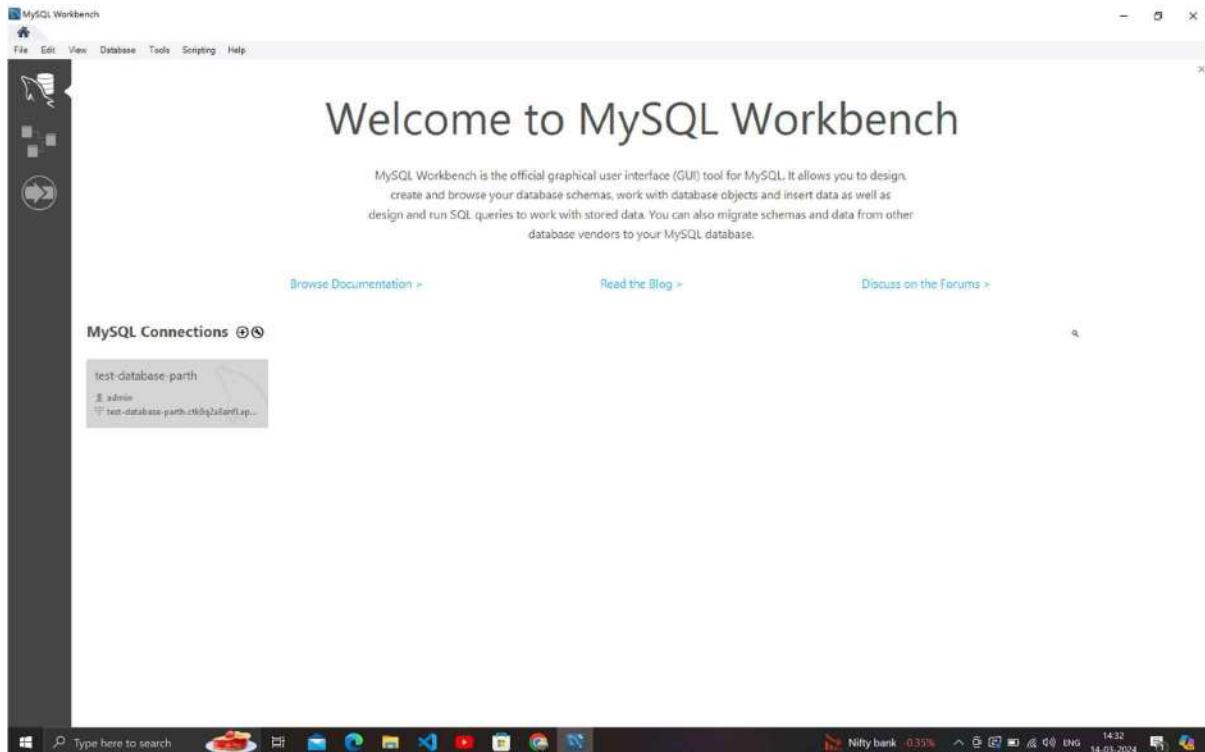
The screenshot shows the 'Edit inbound rules' dialog box for the 'sg-0b25ef9c0ff06611d - default' security group. It displays two existing rules and a new rule being added. The new rule has 'All traffic' selected for Type and '0.0.0.0/0' selected for Source. A warning message at the bottom states: '⚠️ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' Buttons for 'Cancel', 'Preview changes', and 'Save rules' are at the bottom right.



16. Go back to MySQL workbench. Fill the password section. Now select the Test Connection. You will see a pop up stating Connection was Successful.

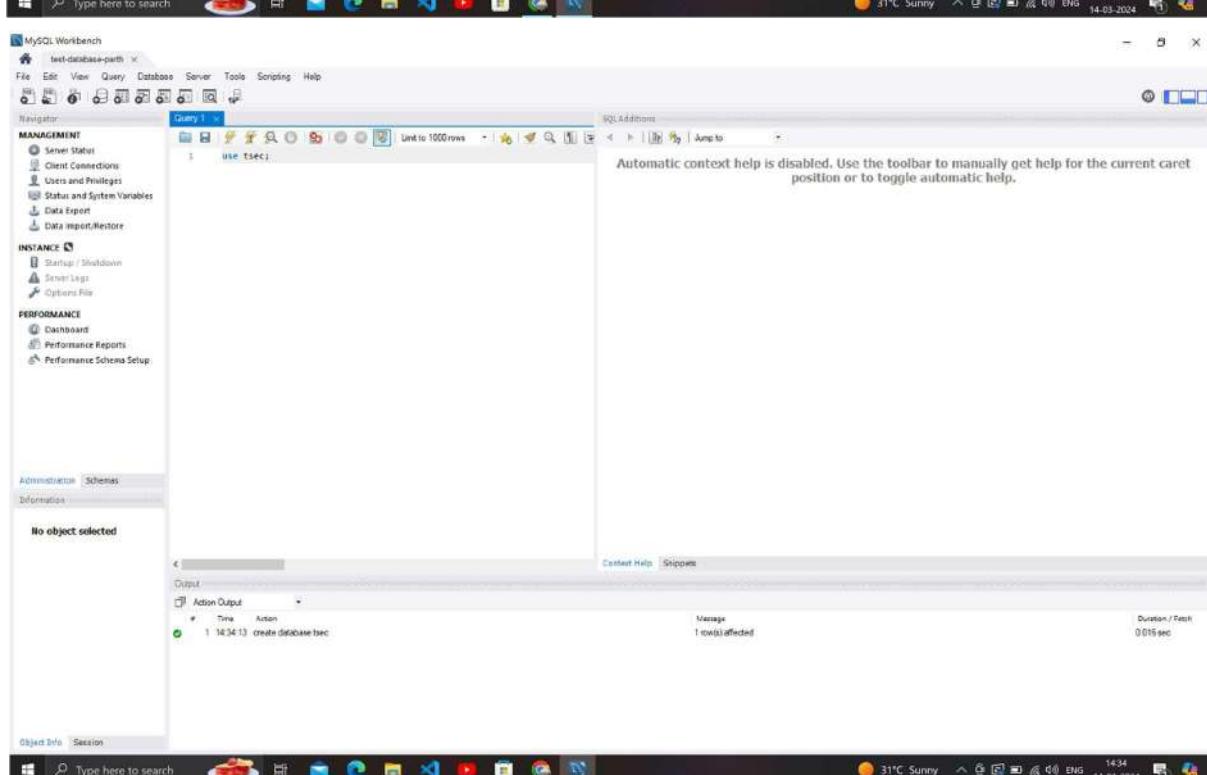
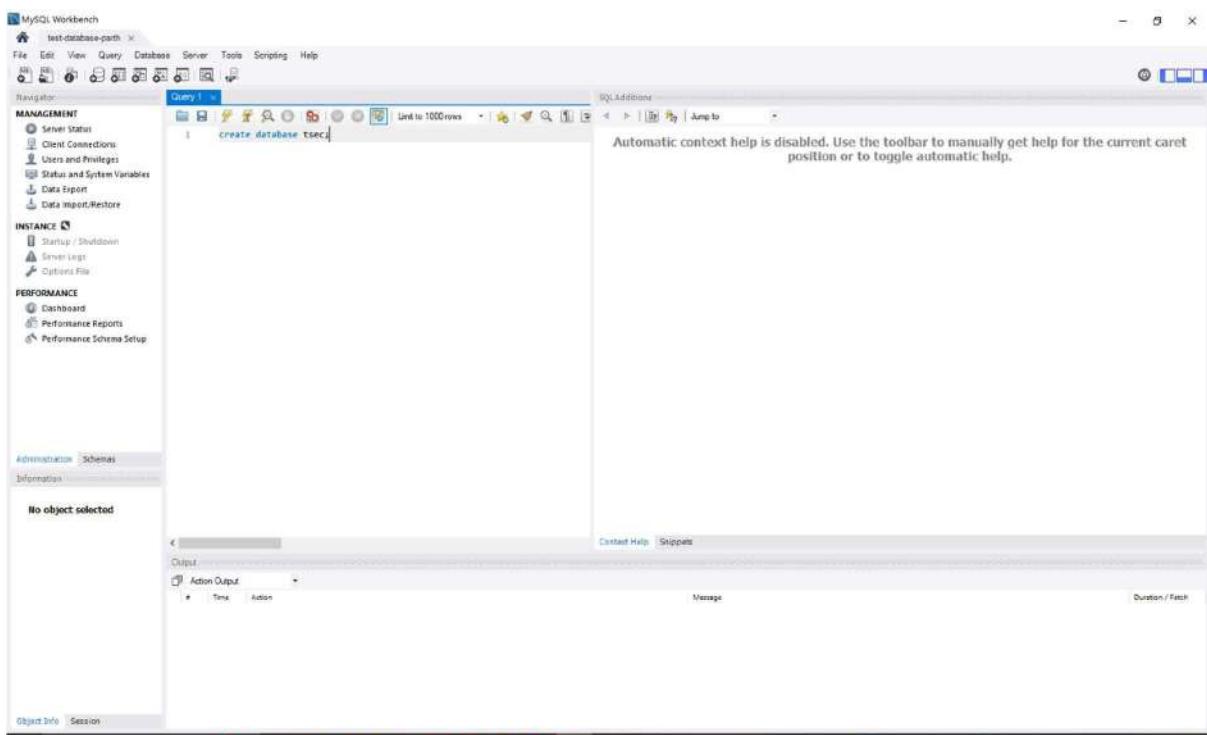


17. Select the database in the dashboard. A new editor instance will be created as follows.



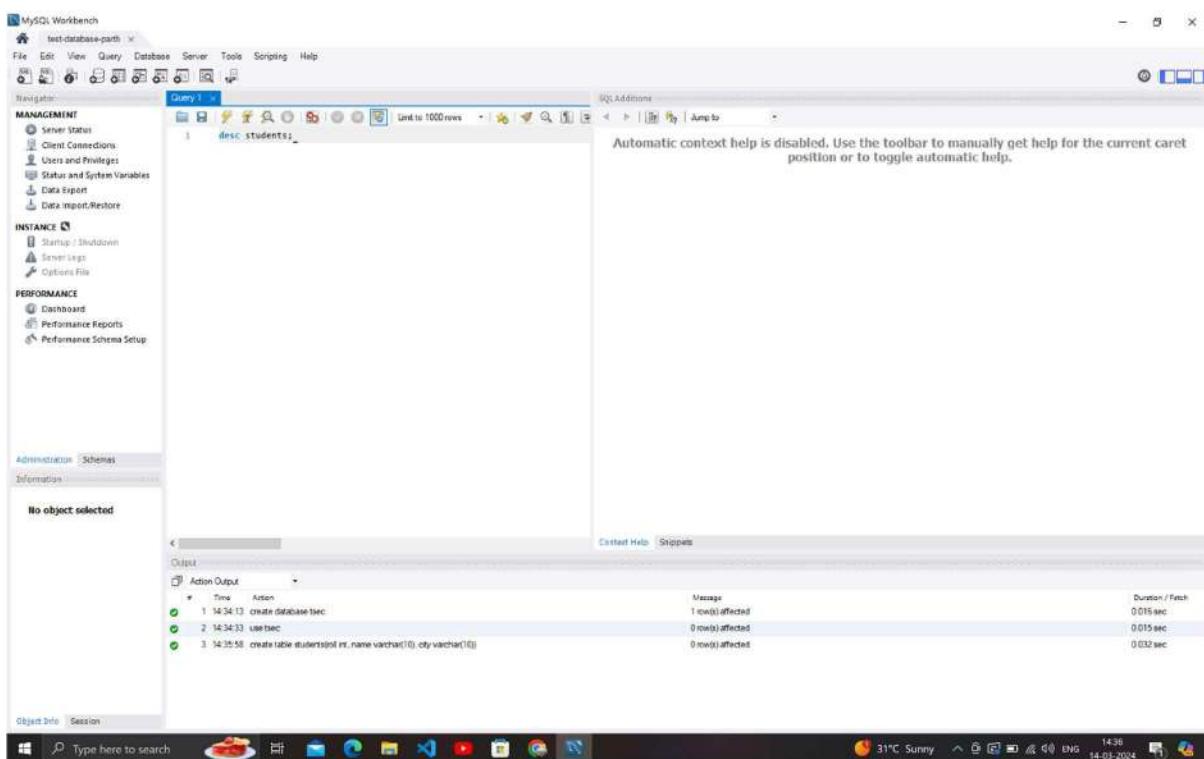
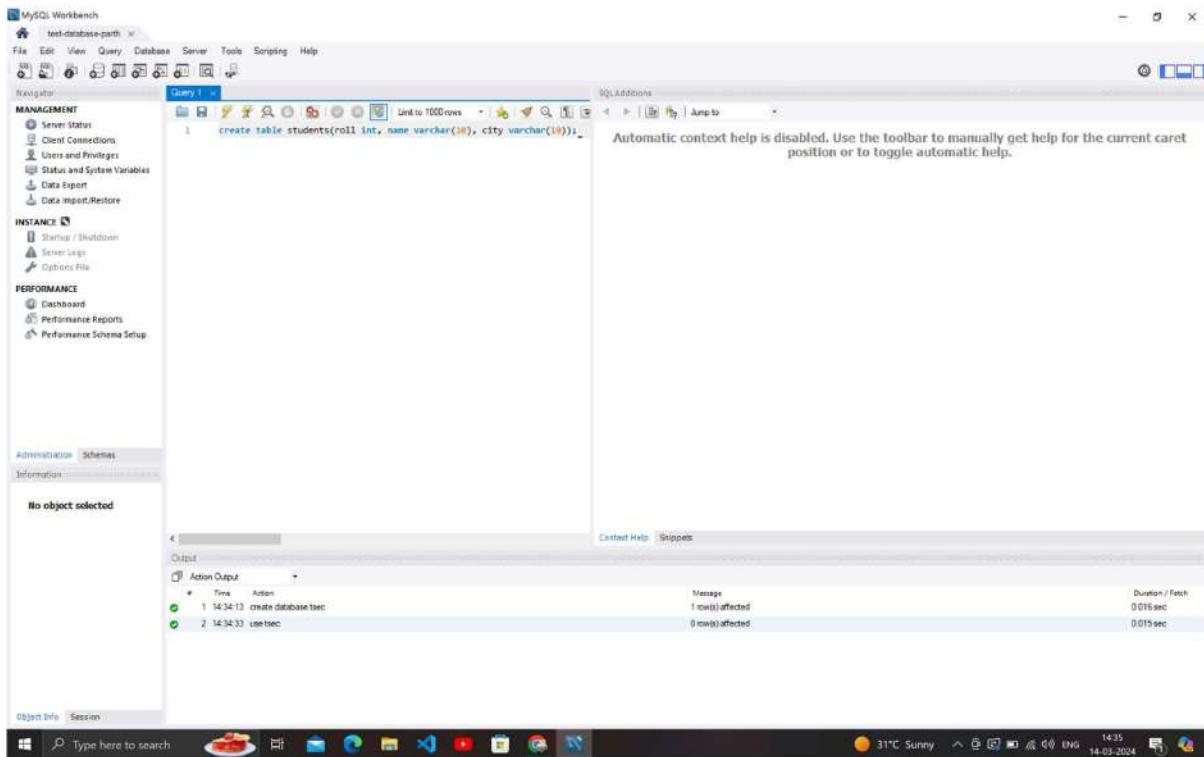
18. Write the following query in the editor:

- a. Create database tsec;
- b. Use tsec;
- c. Create table students(roll int, name varchar(10), city varchar(10));
- d. Insert into students values(32, 'Parth Dabholkar', 'Thane');
- e. Insert into students values(34, 'Vedant Devkar', 'Mumbai');
- f. Desc students;
- g. Select * from students;



Shirish Shetty 2103164 C32

Fath Dabholkar C12 2103052



Shirish Shetty 2103164 C32

The screenshot shows the MySQL Workbench interface. In the 'Query' editor, the following SQL statements were run:

```
1 * create database tec
2 * use tec
3 * create table students(id int, name varchar(10), city varchar(10))
4 * desc students
```

The 'Result Grid' pane displays the structure of the 'students' table:

Field	Type	Null	Key	Default	Extra
id	int	YES	NO		
name	varchar(10)	YES	NO		
city	varchar(10)	YES	NO		

The 'Result 1' pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	14:34:13	create database tec	1 row(s) affected	0.016 sec
2	14:34:33	use tec	0 rows(s) affected	0.015 sec
3	14:35:58	create table students(id int, name varchar(10), city varchar(10))	0 rows(s) affected	0.032 sec
4	14:36:45	desc students	3 rows(s) returned	0.016 sec / 0.000 sec

The screenshot shows the MySQL Workbench interface. In the 'Query' editor, the following SQL statements were run:

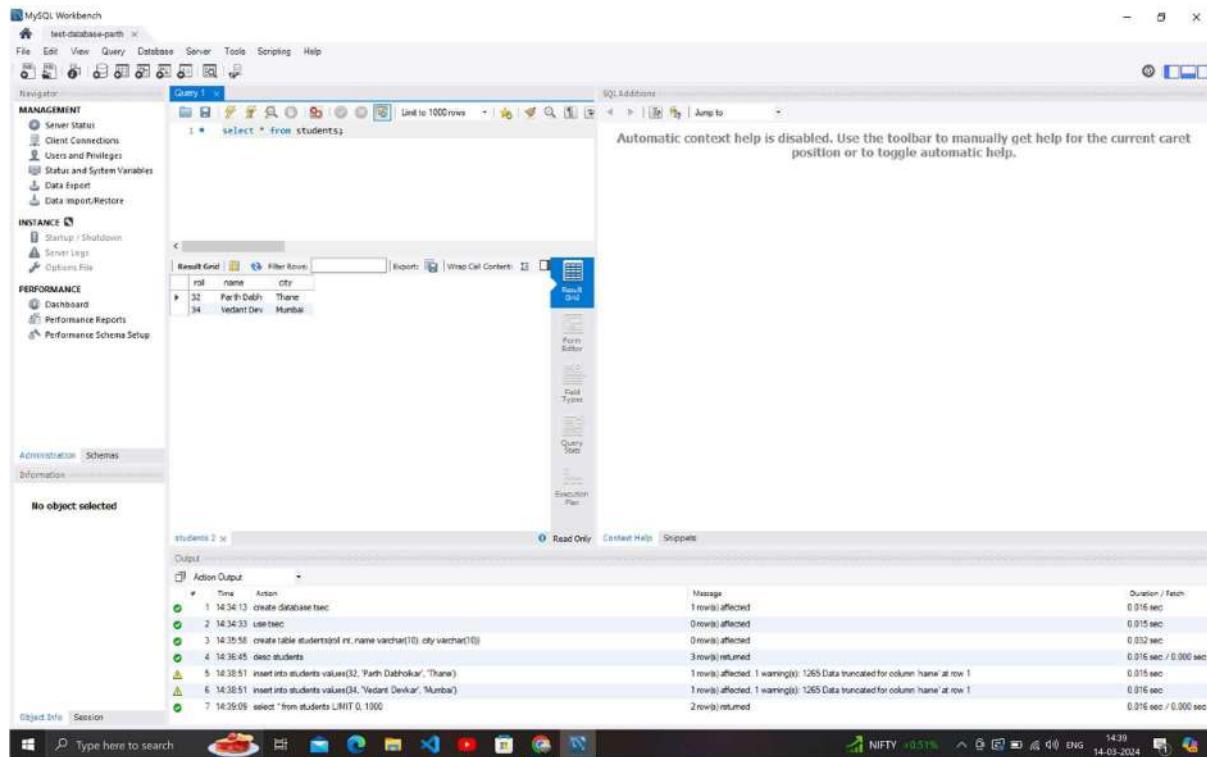
```
1 * insert into students values(1, "Parth Dabholkar", "Thane")
2 * insert into students values(2, "Vedant Devkar", "Mumbai")
```

The 'Result Grid' pane displays the structure of the 'students' table:

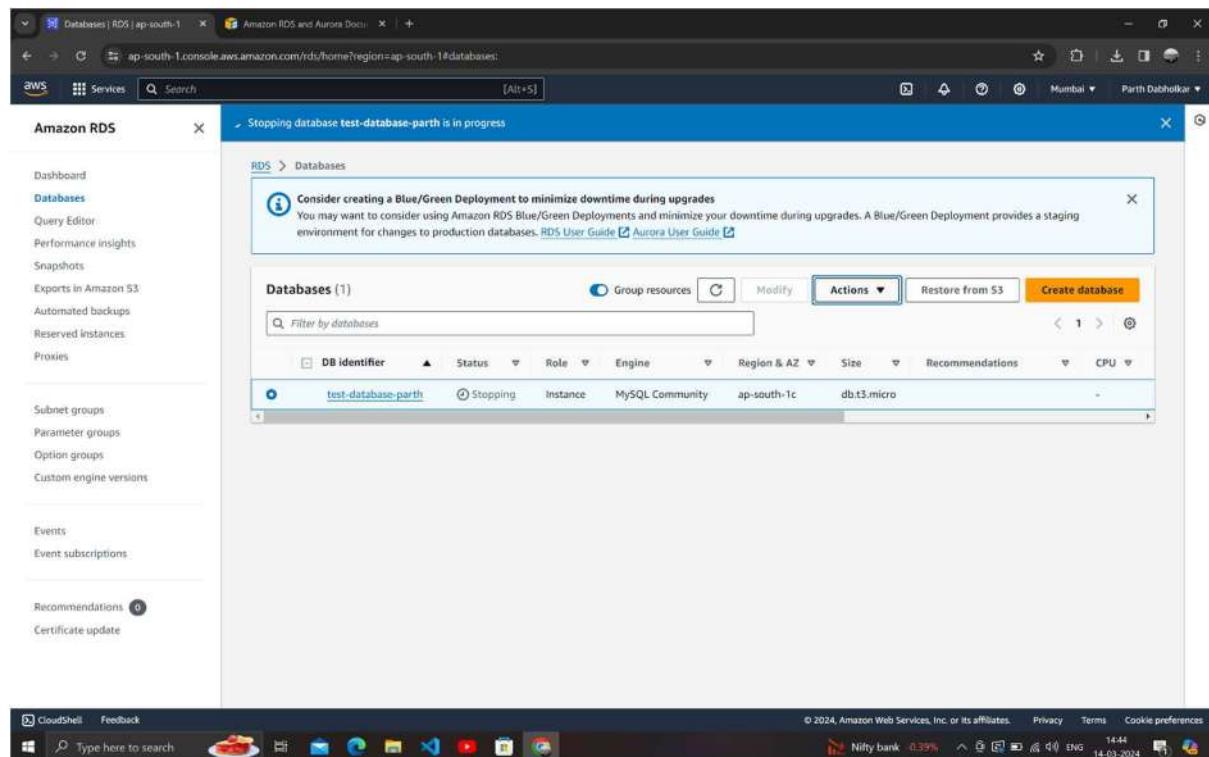
Field	Type	Null	Key	Default	Extra
id	int	YES	NO		
name	varchar(10)	YES	NO		
city	varchar(10)	YES	NO		

The 'Result 1' pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	14:34:13	create database tec	1 row(s) affected	0.016 sec
2	14:34:33	use tec	0 rows(s) affected	0.015 sec
3	14:35:58	insert into students(id, name, city) values(1, "Parth Dabholkar", "Thane")	1 row(s) affected	0.032 sec
4	14:36:45	insert into students(id, name, city) values(2, "Vedant Devkar", "Mumbai")	1 row(s) affected	0.016 sec / 0.000 sec
5	14:36:45	desc students	3 rows(s) returned	

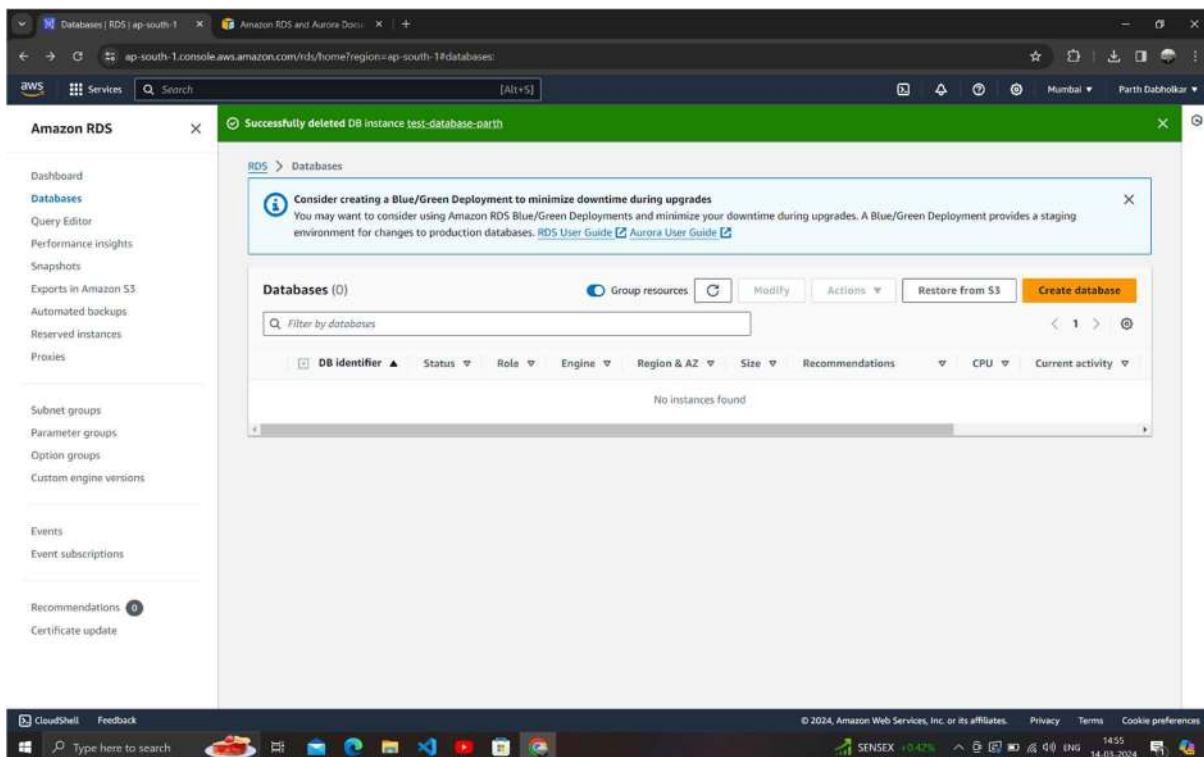
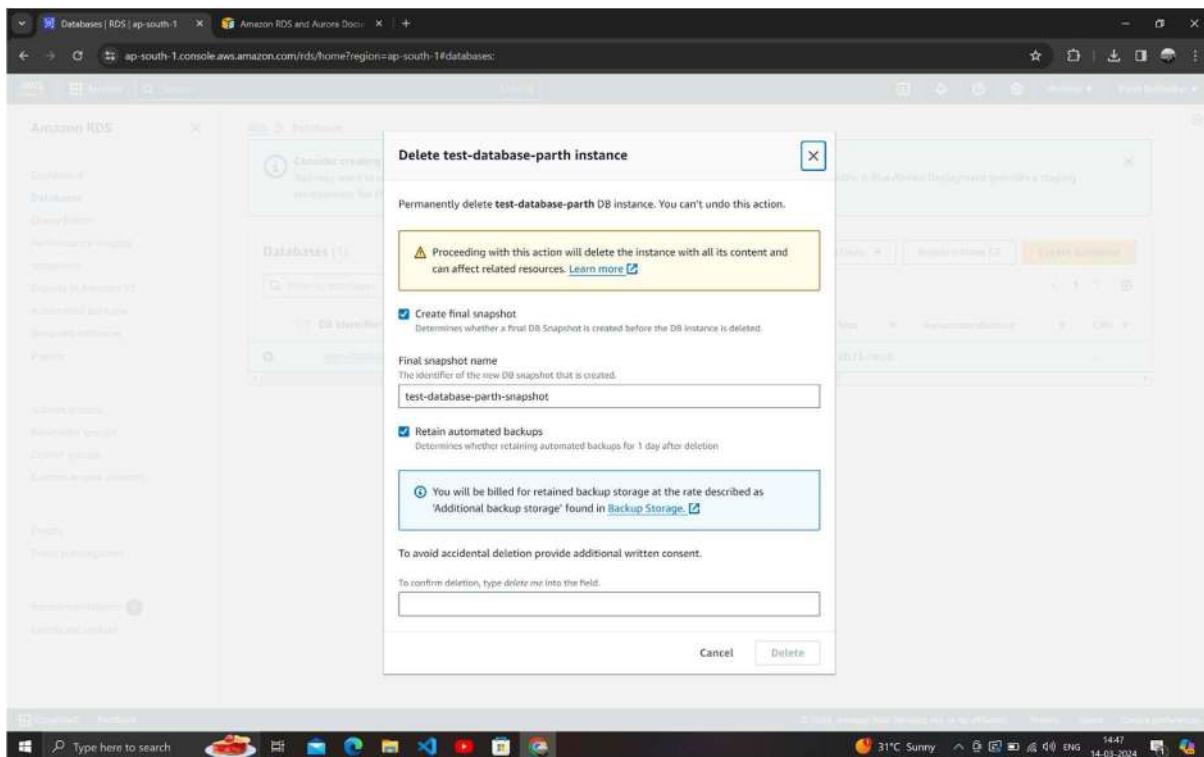


19. After completion of all queries, go back to the AWS portal. Start the deletion procedure. Select the database and select Stop Temporarily. After the database status shows STOP, select Delete database and follow the given steps.



Shirish Shetty 2103164 C32

Railit Dabholkar C12 2103032



Experiment No: 9

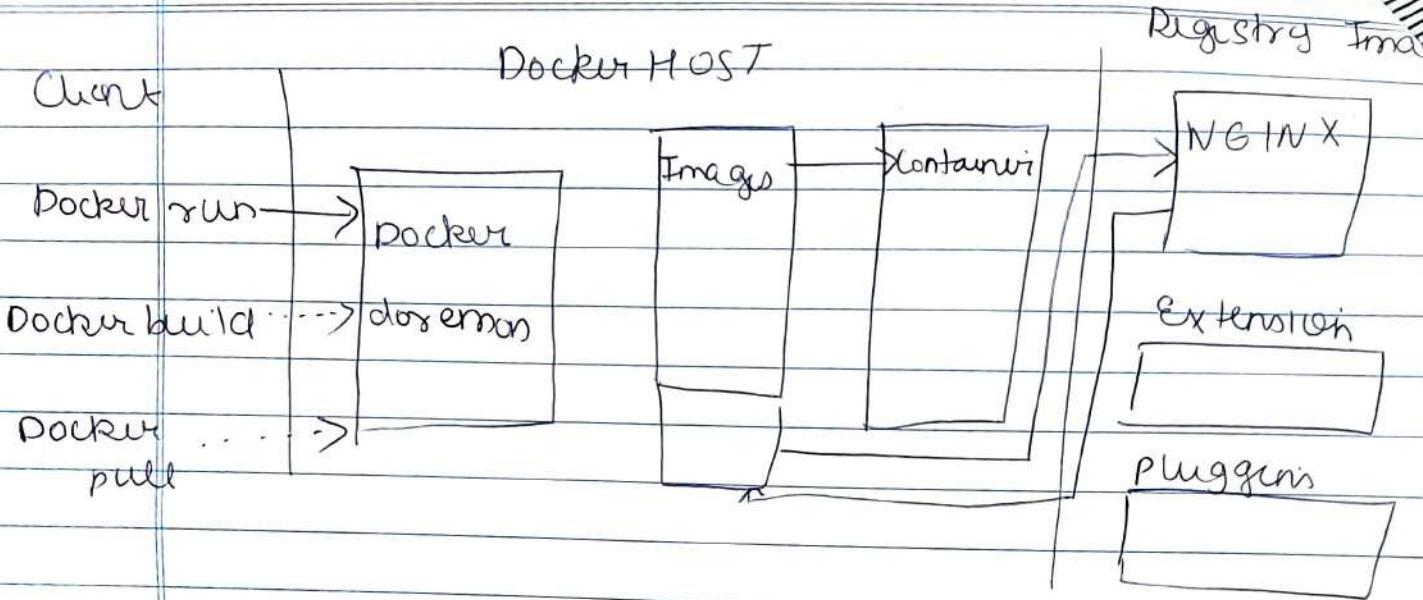
Aim:- To study and implementation containerization using Docker.

Theory :-

Docker :-

Docker is a platform that allow developer to package applications and their dependencies into containers for easy deployment occur different environment. This architecture consist of following component :-

- ① Docker Engine :- The core of Docker that create and manages containers.
- ② Docker Images :- Templates used to create container containing application code and dependencies.
- ③ Docker Container :- Instance of Docker images that run application in isolated environment.
- ④ Docker Registry :- Stores Docker images for sharing & distribution.
- ⑤ Docker Compose :- Tool for defining and running multi-container Docker application



Benefits of containerization:-

Containerization:- Offer Various benefits including

- ① Portability :- Easy movement of application between different environments.
- ② Isolation :- Ensure application runs independently without affecting each other.
- ③ Resource :- Allow running multiple isolated application on same host system.

Explain following w.r.t Docker

- Container A - lightweight standalone executable package of software that includes everything needed to run an application, containers are isolated environments where application runs.

Images :- Templates used to create containers. They include the application code, runtime, system tools, libraries & setting needed to run an application.

Docker file :- A text document that contains a series of commands a user could issue on command line to assemble an image.

Q) Virtual machine and containers

Virtual Machine

- ⑥ VM is a piece of software that allows you to install other software inside of it so you control it virtually opposed to installing the software directly on computer

- ⑦ Application running on VM system or hypervisor can run different OS.

- ⑧ VM virtualizes the computer system meaning its hardware

Containers

- ⑨ Container is software that allows different functionalities of an application independently.

- ⑩ While application running in a container environment share a single OS.

- ⑪ While container virtualizes operating system or the software

- | | |
|--|---|
| ④ VM uses a lot of system memory | ④ Container regular requires less memory |
| ⑤ Eg:- Type I hypervisor, KVM and VMWare | ⑤ Eg - PhotonOS and docker, RancherOS. |

Q:- Docker Image VS Docker Container

Image	Container
① It is blueprint of container	① It is an instance of image
② Image is a logical entity	② Container is a real world entity
③ Images are created only once	③ Container are created any number of time using an image
④ Images do not require computing resource to work	④ Container require computing resource to run as they run with Docker Virtual Machine
⑤ One cannot connect to images as these images are like snapshot	⑤ In this one cannot connect them and execute the command
⑥ Sharing of Docker image is possible	⑦ Sharing of Container is not possible directly.

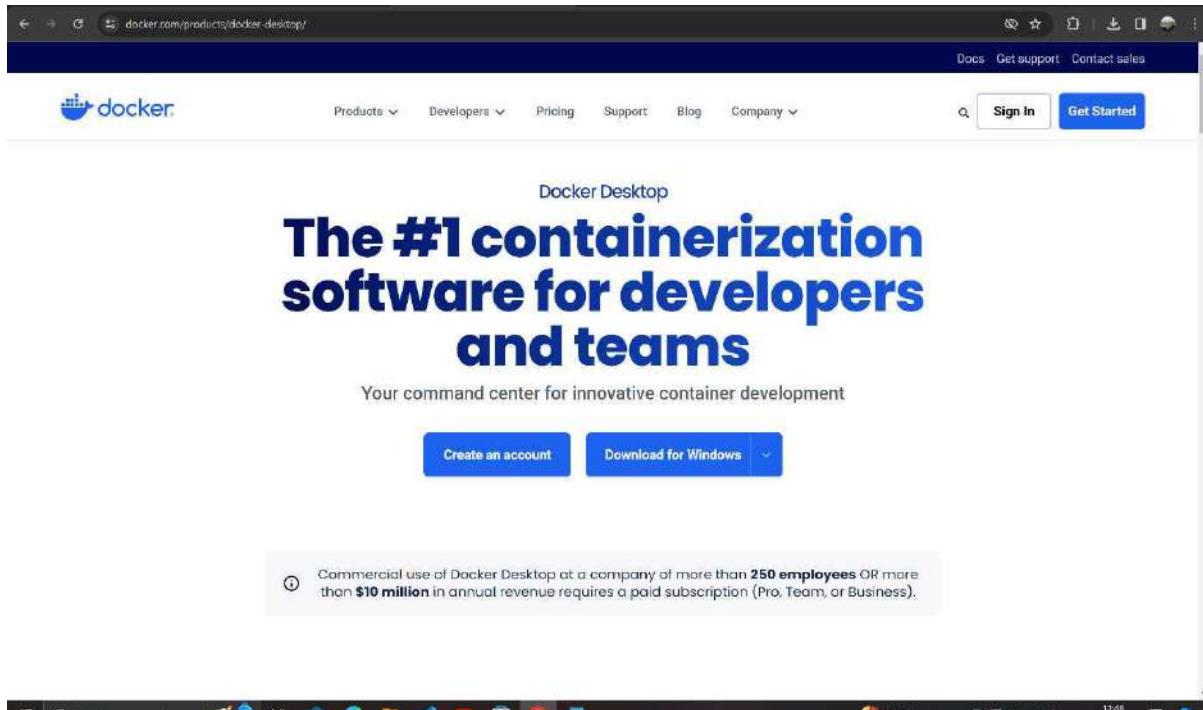
A+

ST
26/3/19

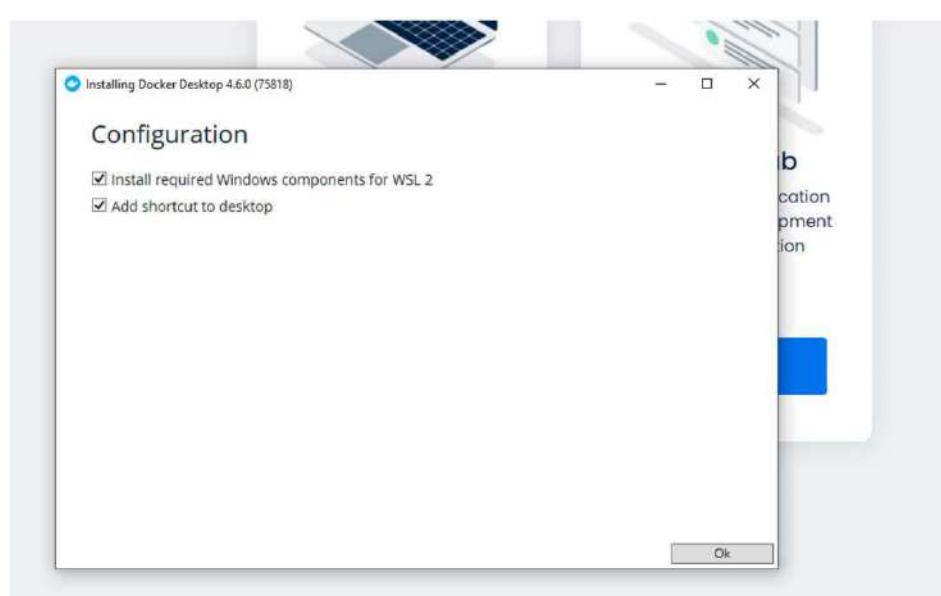
EXPERIMENT NO: 9

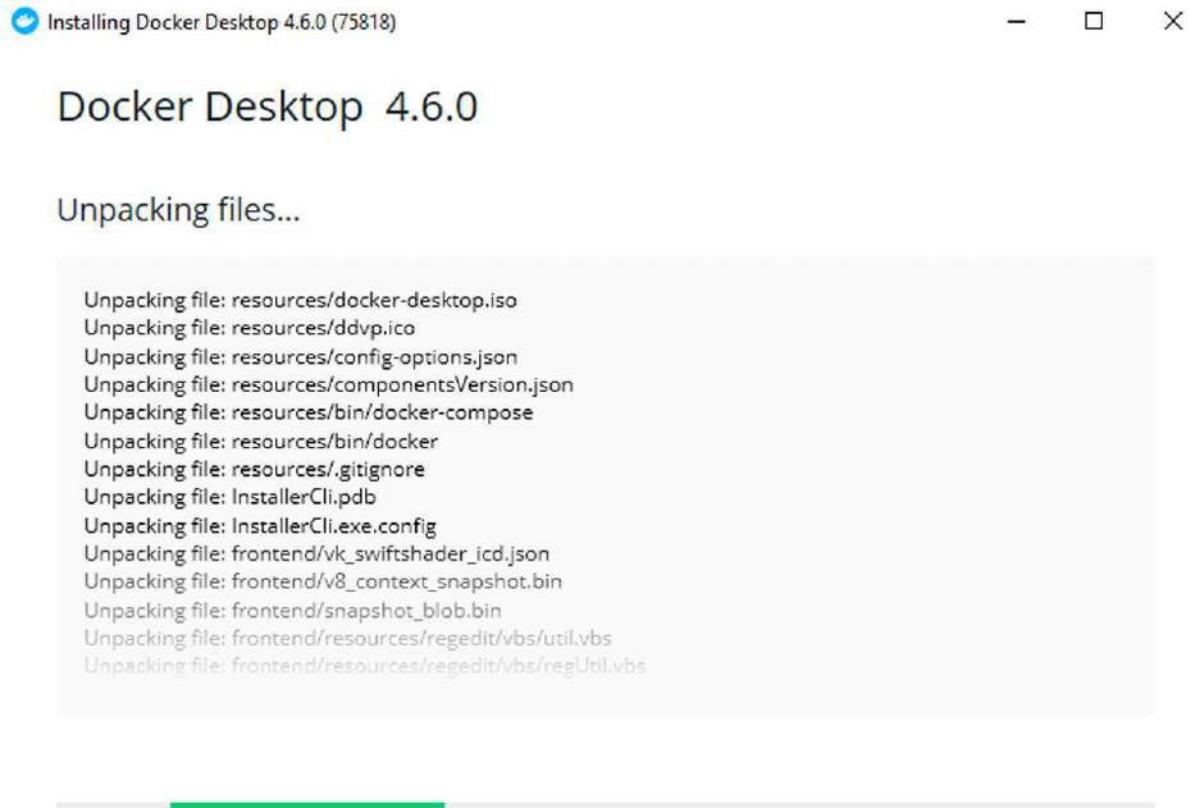
AIM: To study and implement containerization using Docker.

Step-1: Open docker.com on any browser. Scroll down and click on 'Get started for free' button. A new tab appears. Click on download Docker Desktop for Windows and wait for it to be downloaded.

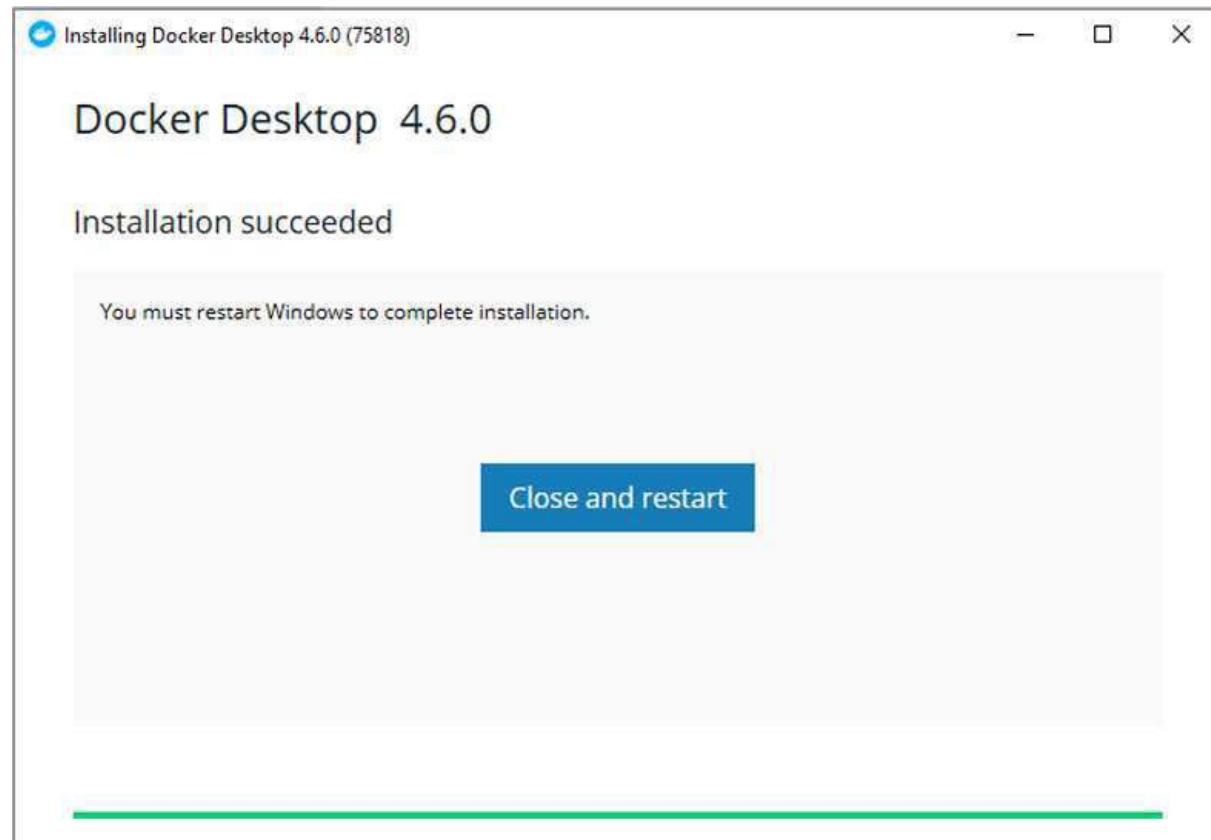


Step-2: Once the setup is downloaded from the website, open the setup wizard on your device. Start the installation process for Docker Desktop.

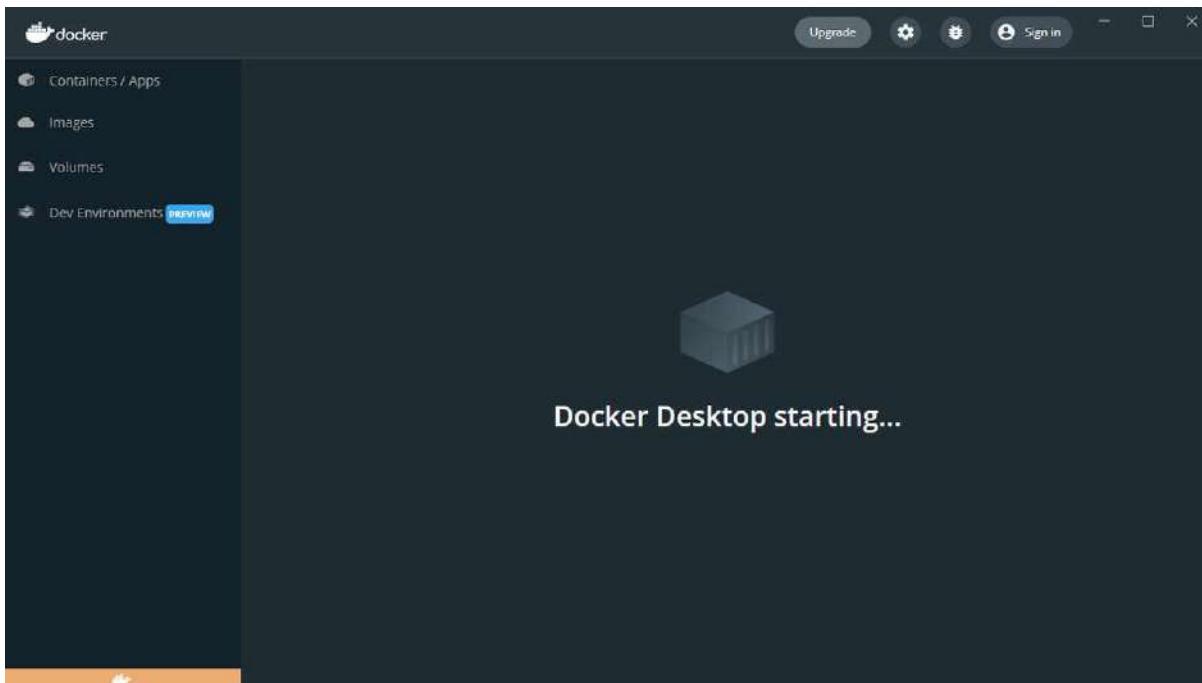
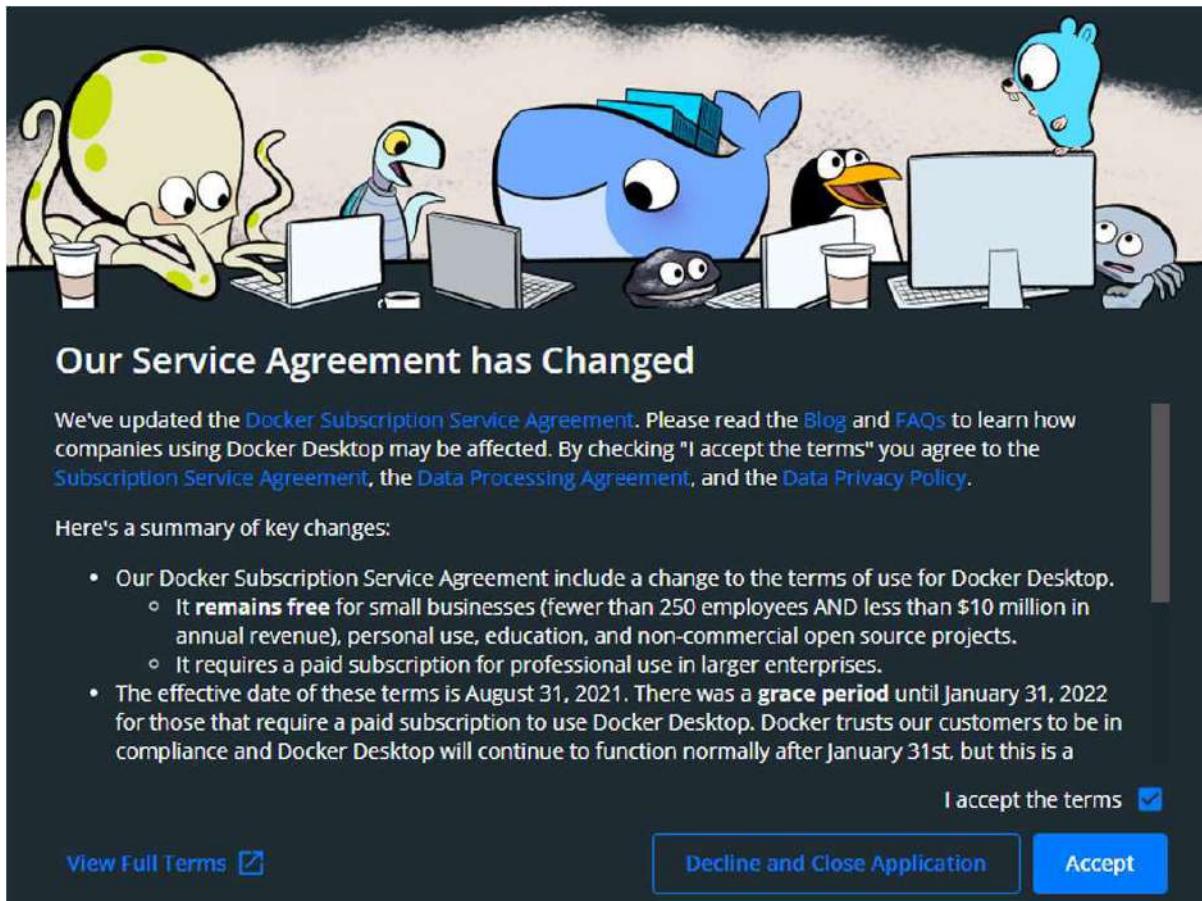




Step-3: After the installation is completed, click on finish the setup. To start the Docker Dashboard, restart your computer device.



Step-4: Now start the Docker Application on your device. Accept all the terms and conditions and move further. A dashboard will open.



Step-5: The following window will pop up. We must install WSL-2 in order to start the Docker Dashboard. Click on the following link for installation.

Click on the link - <https://aka.ms/wsl2kernel>.



Download the WSL2 Linux kernel update package for x64 machines.

Step 4 - Download the Linux kernel update package

1. Download the latest package:

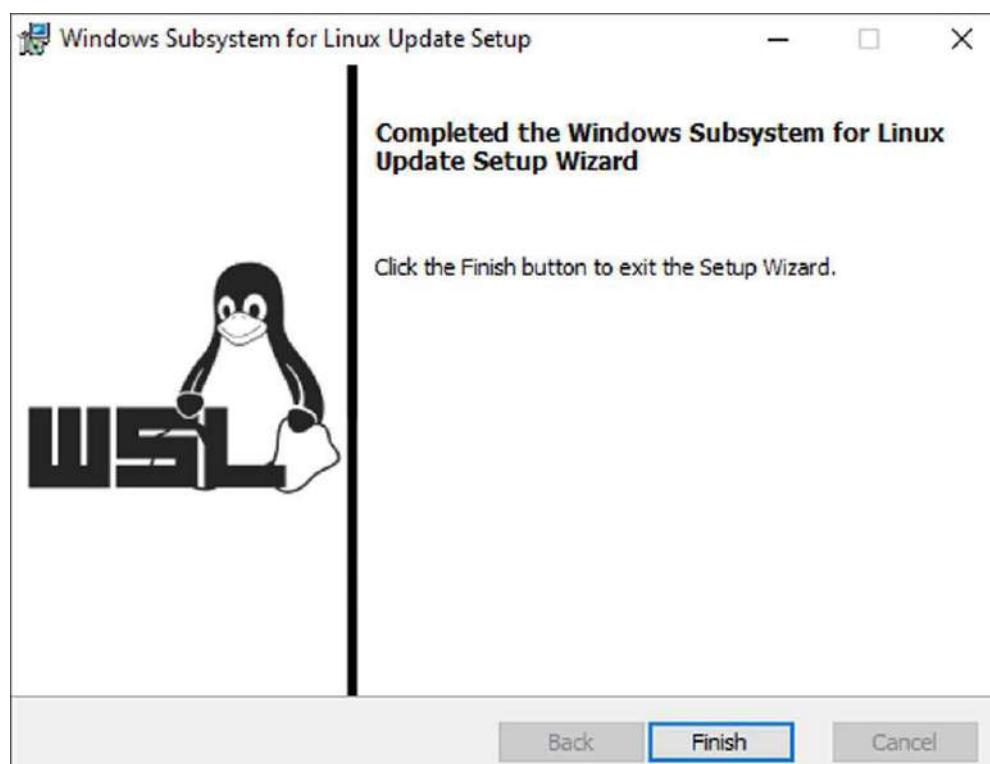
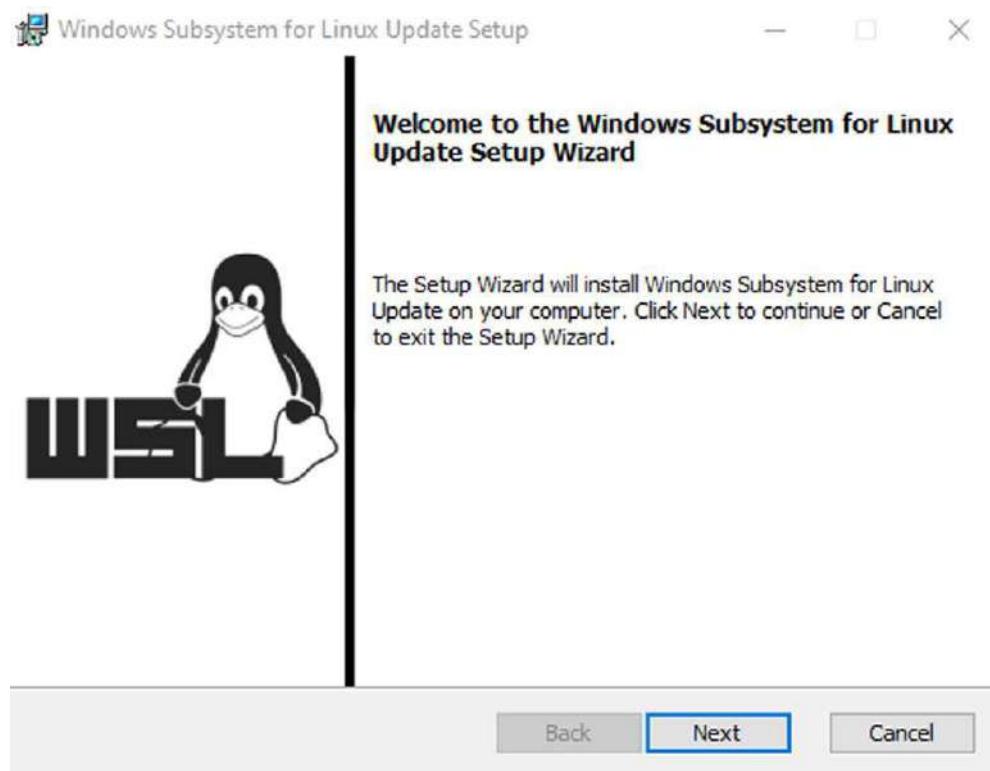
- [WSL2 Linux kernel update package for x64 machines](#)

ⓘ Note

If you're using an ARM64 machine, please download the [ARM64 package](#) instead. If you're not sure what kind of machine you have, open Command Prompt or PowerShell and enter: `systeminfo | find "System Type"`. **Caveat:** On non-English Windows versions, you might have to modify the search text, translating the "System Type" string. You may also need to escape the quotations for the find command. For example, in German `systeminfo | find ""Systemtyp""`.

2. Run the update package downloaded in the previous step. (Double-click to run - you will be prompted for elevated permissions, select 'yes' to approve this installation.)

Step-6: Once the download is complete, open the WSL-2 installation setup. Complete the installation process and click finish.



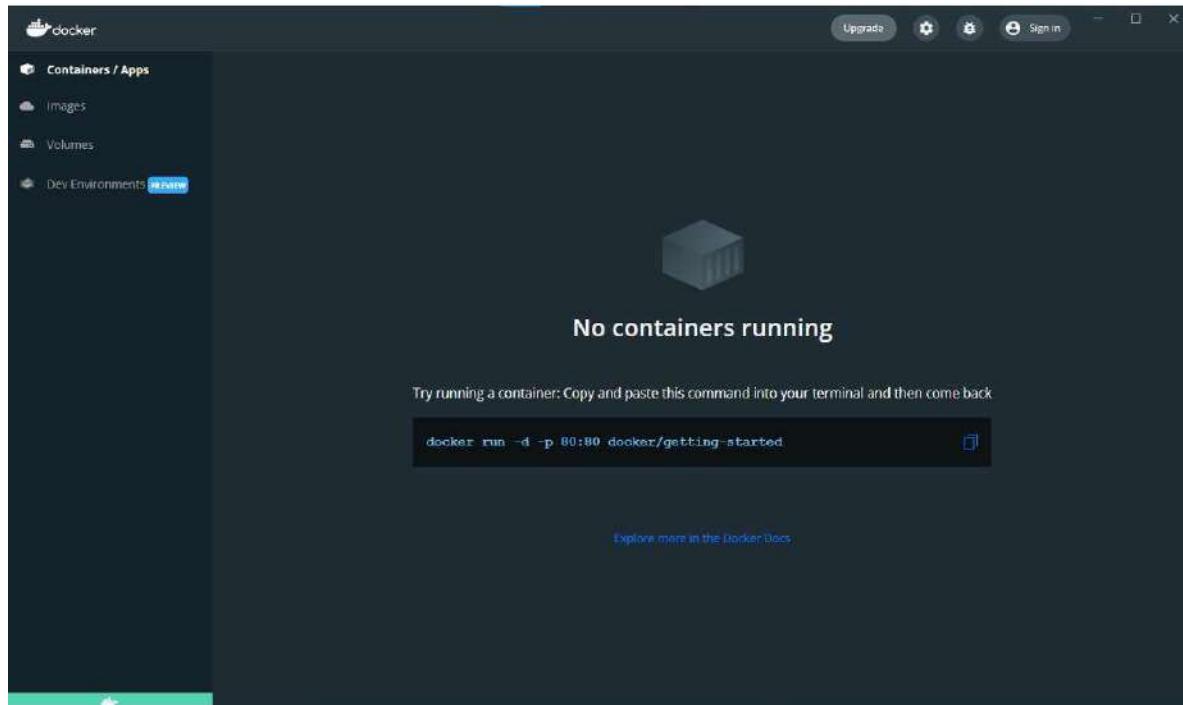
Step-7: Open Windows Powershell as an Administrator.

Run the following command:

```
wsl --set-default-version 2
```

```
PS C:\WINDOWS\system32> wsl --set-default-version 2
For information on key differences with WSL 2 please visit https://aka.ms/wsl2
The operation completed successfully.
PS C:\WINDOWS\system32> _
```

Now click on restart. Docker should now restart. The following window should pop up:



Step-8: Open Command Prompt, run the following commands:

docker ps

docker container ls -a

docker container rm 33fefebf03cd //copy docker id for remove but first

(Use your container ID in the above command)

stop your docker

- docker container stop 33fefebf03cd

- docker container rm 33fefebf03cd

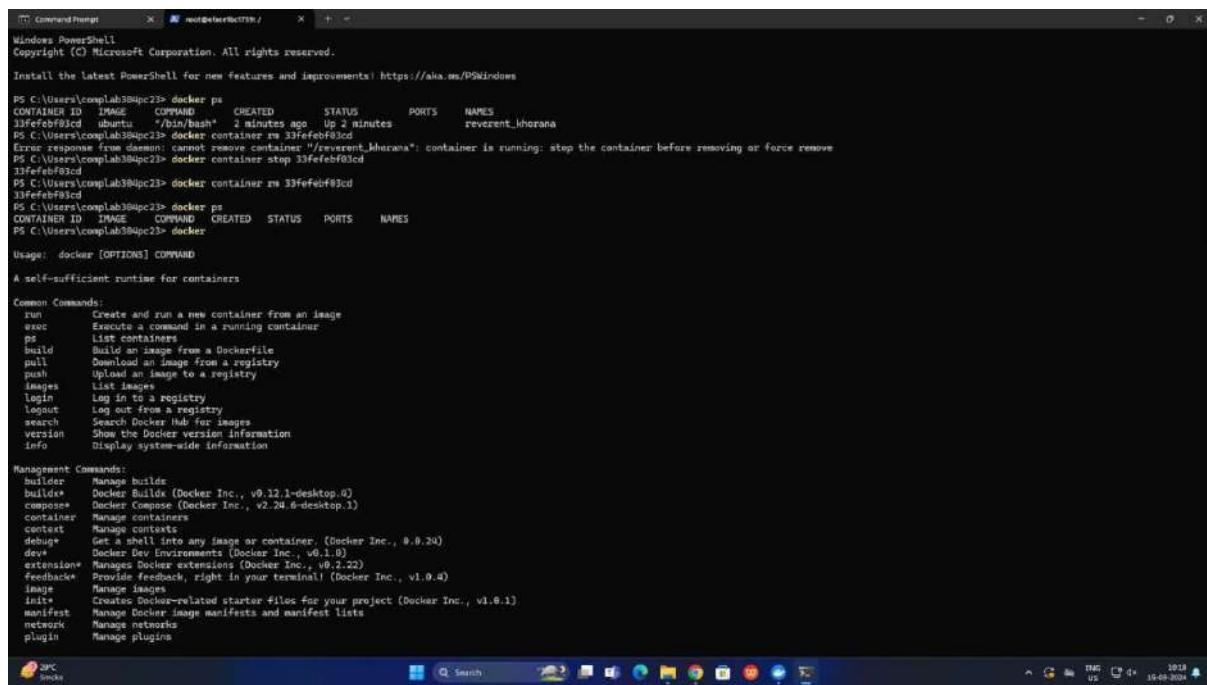
- docker ps - docker //list all docker commands

- docker images

- docker image rm ca2b0f26964c // copy image id from previous output

(Use your image ID in the above command)

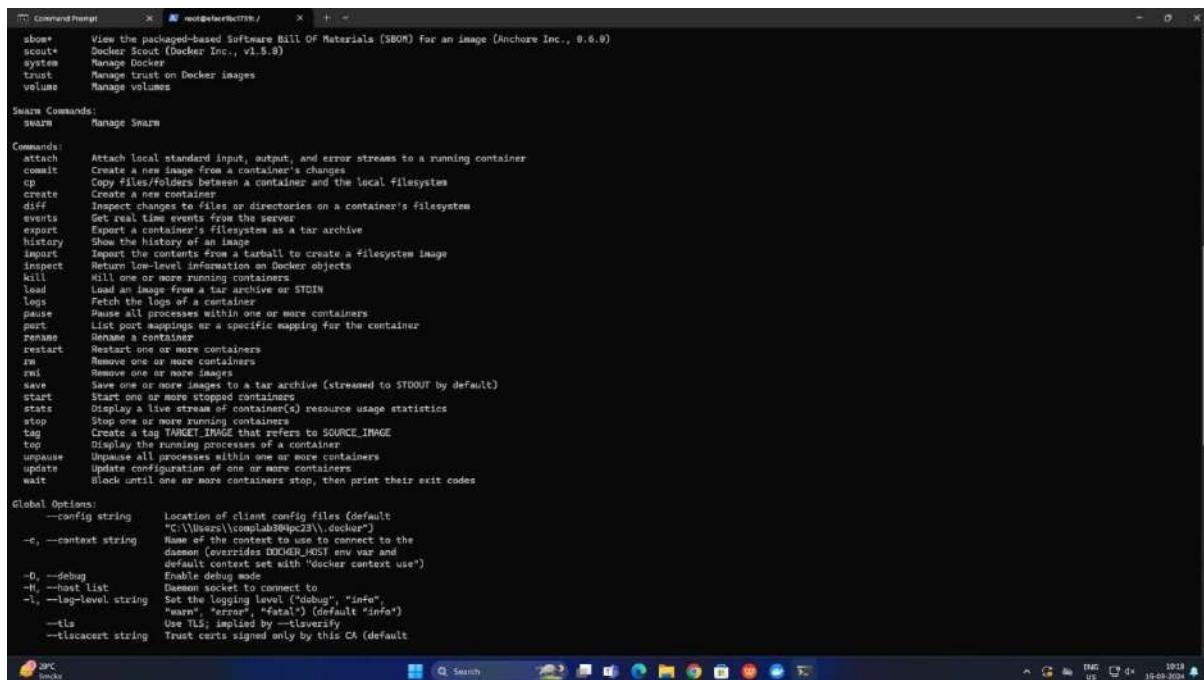
- docker run -it ubuntu /bin/bash //check output



```
PS C:\Users\compLab304pc> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
33fefebf03cd ubuntu "/bin/bash" 2 minutes ago Up 2 minutes reverent_khorana
PS C:\Users\compLab304pc> docker container rm 33fefebf03cd
Error response from daemon: cannot remove container "reverent_khorana": container is running: stop the container before removing or force remove
PS C:\Users\compLab304pc> docker containers stop 33fefebf03cd
33fefebf03cd
PS C:\Users\compLab304pc> docker container rm 33fefebf03cd
33fefebf03cd
PS C:\Users\compLab304pc> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
PS C:\Users\compLab304pc> docker
Usage: docker [OPTIONS] COMMAND
A self-sufficient runtime for containers

Common Commands:
run      Create and run a new container from an image
exec    Execute a command in a running container
ps      List containers
build   Build an image from a Dockerfile
pull    Download an image from a registry
push    Upload an image to a registry
logs    List logs
login   Log in to a registry
logout  Log out from a registry
search   Search Docker Hub for images
version  Show the Docker version information
info     Display system-wide information

Management Commands:
builder Manage builds
buildx* Docker Buildx (Docker Inc., v0.12.1-desktop.0)
compose* Docker Compose (Docker Inc., v2.24.6-desktop.1)
container Manage containers
context  Manage contexts
debug*  Get a shell into any image or container (Docker Inc., v0.0.24)
dev    Docker development environment (Docker Inc., v0.1.0)
extension* Manages Docker extensions (Docker Inc., v0.2.22)
feedback* Provide feedback, right in your terminal! (Docker Inc., v1.0.4)
image   Manage images
init*  Creates Docker-related starter files for your project (Docker Inc., v1.0.1)
manifest Manage Docker image manifests and manifest lists
network Manage networks
plugin  Manage plugins
```



```

Command Prompt  X  root@eface1bc1798: ~ + =
sbom*   View the packaged-based Software Bill Of Materials (SBOM) for an image (Anchore Inc., v0.6.0)
scout*  Docker Scout (Docker Inc., v1.5.8)
system* Manage Docker
trust*  Manage trust on Docker images
volume* Manage volumes

Shared Commands:
  swarm  Manage Swarm

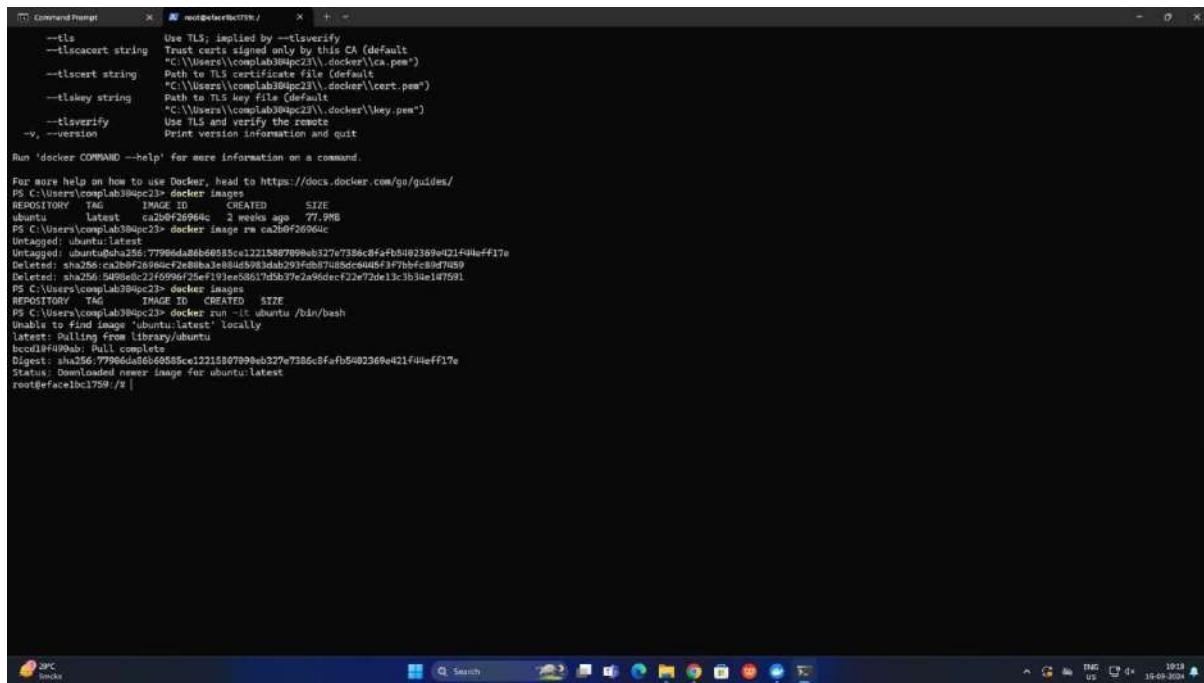
Commands:
  attach   Attach local standard input, output, and error streams to a running container
  commit* Create a new image from a container's changes
  cp      Copy files/folders between a container and the local filesystem
  create* Create a new container
  diff    Inspect changes to files or directories on a container's filesystem
  events* Get real time events from the server
  export* Export a container's filesystem as a tar archive
  history Show the history of an image
  import* Import the contents from a tarball to create a filesystem image
  inspect* Return low-level information on Docker objects
  kill    Kill one or more running containers
  load    Load an image from a tar archive or STDIN
  logs*  Fetch the logs of a container
  pause*  Pause all processes within one or more containers
  port*   List port mappings or a specific mapping for the container
  rename* Rename a container
  restart Restart one or more containers
  rm     Remove one or more containers
  rmi    Remove one or more images
  save*  Save one or more images to a tar archive (streamed to STDOUT by default)
  start* Start one or more containers
  stats* Display a live stream of container(s) resource usage statistics
  stop   Stop one or more running containers
  tag    Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
  top    Display the running processes of a container
  unpause* Unpause all processes within one or more containers
  update* Update configuration of one or more containers
  wait   Block until one or more containers stop, then print their exit codes

Global Options:
  --config string      Location of client config files (default "C:\Users\complab304pc23\.docker")
  -c, --context string  Name of the context to use to connect to the daemon via TCP (Docker's --host var and default context set with "docker context use")
  -D, --debug          Enable debug mode
  -H, --host list       Daemon socket to connect to
  -l, --log-level string Set the logging level ("debug", "info",
                         "warn", "error", "fatal") (default "info")
  --tls               Use TLS; implied by --tlscacert
  --tlscacert string   Trust certs signed only by this CA (default "C:\Users\complab304pc23\.docker\ca.pem")
  --tlscert string     Path to TLS certificate file (default "C:\Users\complab304pc23\.docker\cert.pem")
  --tlskey string      Path to TLS key file (default "C:\Users\complab304pc23\.docker\key.pem")
  --tlsv1.2             Use TLS and verify the remote
  -v, --version        Print version information and quit

Run 'docker COMMAND --help' for more information on a command.

For now help on how to use Docker, head to https://docs.docker.com/guide/
PS C:\Users\complab304pc23> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu latest ca2b0f26964c 2 weeks ago 77.9MB
PS C:\Users\complab304pc23> docker image rm ca2b0f26964c
Untagged: ubuntu:latest
Untagged: ubuntu@77986da8b6655c42213b97ff9e3377e7394cf4fb80b2389e021404ff1f7e
Deleted: sha256:77986da8b66555c42213b97ff9e3377e7394cf4fb80b2389e021404ff1f7e
Deleted: sha256:77986da8b66555c42213b97ff9e3377e7394cf4fb80b2389e021404ff1f7e
Status: Downloaded newer image for ubuntu:latest
root@eface1bc1798: ~ |

```



```

Command Prompt  X  root@eface1bc1798: ~ + =
--tls           Use TLS; implied by --tlscacert
--tlscacert string  Trust certs signed only by this CA (default "C:\Users\complab304pc23\.docker\ca.pem")
--tlscert string   Path to TLS certificate file (default "C:\Users\complab304pc23\.docker\cert.pem")
--tlskey string     Path to TLS key file (default "C:\Users\complab304pc23\.docker\key.pem")
--tlsv1.2           Use TLS and verify the remote
-v, --version      Print version information and quit

Run 'docker COMMAND --help' for more information on a command.

For now help on how to use Docker, head to https://docs.docker.com/guide/
PS C:\Users\complab304pc23> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu latest ca2b0f26964c 2 weeks ago 77.9MB
PS C:\Users\complab304pc23> docker run -it ubuntu /bin/bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
b2cd10f490ab: Pull complete
Digest: sha256:77986da8b665555c42213b97ff9e3377e7394cf4fb80b2389e021404ff1f7e
Status: Downloaded newer image for ubuntu:latest
root@eface1bc1798: ~ |

```

Experiment No: 10

Aim:- To study and implement container orchestration using Kubernetes

-Theory:-

Container orchestration tools like Kubernetes play a crucial role in modern environments due to complexity and scale of containerized application. Below is a detailed information about it is provided.

Kubernetes commonly known as K8s, is an open source system design for automating the deployment, scaling and management of containerized applications. Kubernetes leverages over 10 years of Google experience in running production workloads incorporating best practices from community to provide a robust container orchestration platform.

Need for Container Orchestration Tool.

Container orchestration tools are essential for managing, deployment, scaling, networking and availability of containerized applications. With the rise of microservices architecture and cloud-native application, Container orchestration tool have become indispensable for orchestrating complex application across multiple host or cloud environments.

What is Kubernetes?

Kubernetes is an open source container orchestration platform developed by Google, now maintained by Cloud Native Computing Foundation. It automates the deployment, scaling and management of containerized applications. Kubernetes offer such set of features that include automatic rollout, self healing, horizontal scaling & more. It provides a router system for operating containerized application in production environments.

Kubernetes component, working and Architecture, Kubernetes consists of several key components

- ① **Master Node** :- Controls the cluster and manages its state
- ② **Worker Nodes** :- Run application containers and communicate with master node
- ③ **Pods** smaller deployable units in Kubernetes containing one or more containers
- ④ **Services** :- Define networking and load balancing for pods.
- ⑤ **ReplicaSets** :- Ensure a specified number of pods replicas are running.
- ⑥ **Controller** :- Manages the state of clusters
- ⑦ **Volumes** - Provide Persistent storage for containers

Kubernetes follows a client server architecture, where master node controls the cluster state and worker nodes run application containers. The master node includes components like API Server, Scheduler, Controller Manager etc. Worker nodes have the Kubernetes, Kube-proxy and container runtime.

Difference between POD & node

POD	Node
① smallest deployable unit in Kubernetes	① worker machine in Kubernetes.
② Contains one or more container	② Provides CPU & memory resource for Pods.
③ Stores resource like networking storage	③ Can be physical or virtual machine
④ Designed to be short lived and disposable	④ Components include Kubernetes & container runtime
⑤ Automatically scheduled to run on a node	⑤ Responsible for managing containers and downloading them.
⑥ facilitates to locate containers	⑥ execute pods and provide necessary infrastructure

Q) Compare Kubernetes and Docker Swarm

Kubernetes

- ① Kubernetes offers more extensive and flexible networking model through Network plugins and Golang support.
- ② Has a steeper learning curve due to advanced features & flexibility.
- ③ Supports autoscaling based on resource utilization.
- ④ Provides a customizable approach to load balancing allowing for specific configuration.
- ⑤ Requires the installation of Kubectl command line tool for cluster management.

Docker swarm

- ① Provides simpler built-in overlay networking capabilities for easier management.
- ② Easy to learn, start with making it, lightweight, and straightforward tool for beginners.
- ③ Emphasizes scaling quickly without supporting auto scaling ability.
- ④ Offers automatic load balancing mechanism built in seamless communication between containers.
- ⑤ No need for an additional command line tool specifically for cluster management.

(A)

122
26/3 home

EXPERIMENT NO: 10

AIM: To study and implement container orchestration using Kubernetes.

Step-1: Before starting the minikube installation, it is recommended to install all available updates on your system. Run following command.

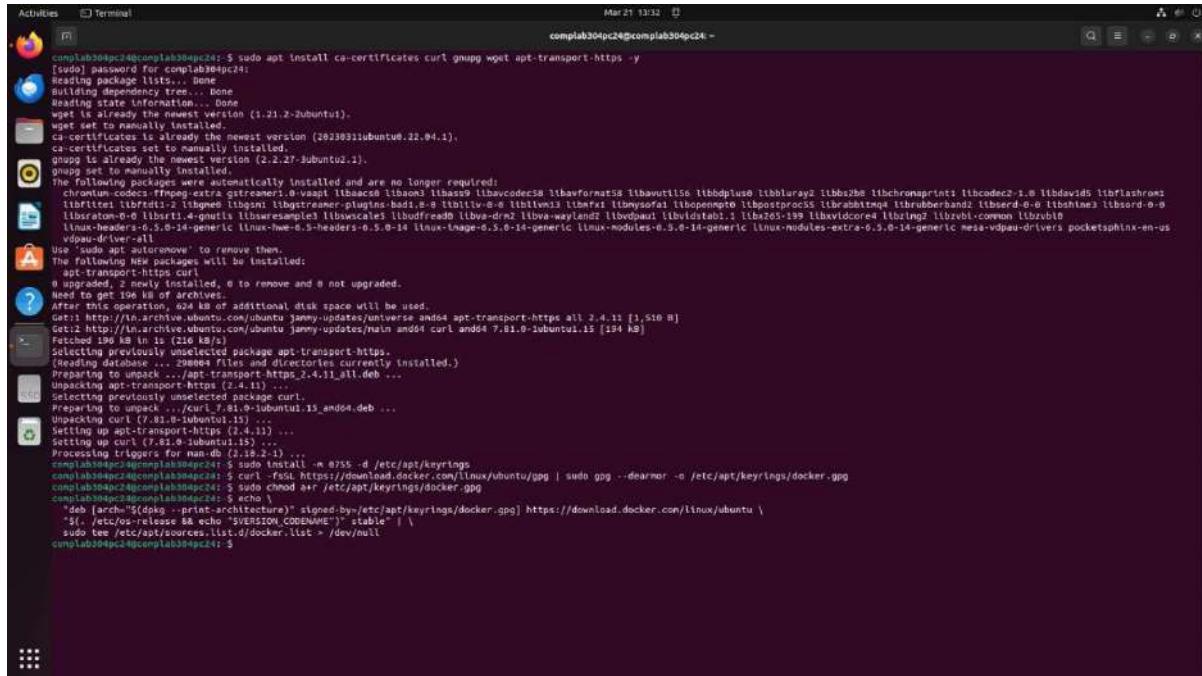
```
$ sudo apt update
```

```
$ sudo apt upgrade -y
```

```
Activities Terminal Mon Jul 24 13:19:22 UTC 2023 comlab304pc24@comlab304pc24: ~
comlab304pc24:~$ sudo apt update
[sudo] password for comlab304pc24:
Get:1 https://packages.microsoft.com/repos/vscode stable InRelease [3,594 B]
Get:2 https://packages.microsoft.com/repos/vscode stable/main amd64 Packages [26.9 kB]
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [139 kB]
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [116 kB]
Hit:6 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [189 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [594 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,302 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy-security/main i386 Packages [404 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu jammy-security/main amd64 Packages [109 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [269 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [1,019 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,059 kB]
Get:14 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [601 kB]
Get:15 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [144 kB]
Get:16 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [59.2 kB]
Get:17 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [244 kB]
Get:18 http://in.archive.ubuntu.com/ubuntu jammy-backports/main i386 Packages [59.2 kB]
Get:19 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [229 kB]
Get:20 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [1,591 kB]
Get:21 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [1,591 kB]
Get:22 http://archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.0 kB]
Get:23 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [28.4 kB]
Get:24 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe i386 Packages [17.2 kB]
Get:25 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [10.2 kB]
Get:26 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [1,591 kB]
Get:27 http://security.ubuntu.com/ubuntu jammy-security/universe i386 Packages [599 kB]
Get:28 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [851 kB]
Get:29 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [162 kB]
Fetched 12,3 MB in 5s (2,359 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
196 packages can be upgraded. Run 'apt list --upgradable' to see them.
comlab304pc24@comlab304pc24: ~
```

Step-2: Install Docker. Minikube requires either docker or VirtualBox, in this post, we will be installing docker on Ubuntu 22.04 system. Run the following set of command one after the another to docker apt repository.

```
$ sudo apt install ca-certificates curl gnupg wget apt-transport-https -y  
$ sudo install -m 0755 -d /etc/apt/keyrings  
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o  
/etc/apt/keyrings/docker.gpg  
$ sudo chmod a+r /etc/apt/keyrings/docker.gpg  
$ echo \  
"deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]  
https://download.docker.com/linux/ubuntu \  
"$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
$ sudo apt update
```



Step-3: Install docker by running the following command.

```
$ sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

The screenshot shows a terminal window titled "Activities Terminal" with the command \$ sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin. The output lists various packages being downloaded from the Ubuntu archive, including docker-ce, docker-ce-cli, containerd.io, docker-buildx-plugin, docker-compose-plugin, and several dependencies like libcurl4, libltdl, and libcurl. The progress bar indicates the download and extraction of files, with a total size of 124 MB.

Step-4: Add your local user to docker group so that your local user run docker commands without sudo.

```
$ sudo usermod -aG docker $USER
```

```
$ newgrp docker
```

The screenshot shows a terminal window titled "Activities Terminal" with the command \$ sudo usermod -aG docker \$USER followed by \$ newgrp docker. This creates a new group "docker" and adds the current user to it. The user then logs out and back in to apply the changes. The terminal then shows the status of the Docker service with the command \$ systemctl status docker. It displays the service is active (running) since Mar 21 2024, with tasks at 18 and memory usage around 29.7M. The log output shows Docker starting up and initializing containers.

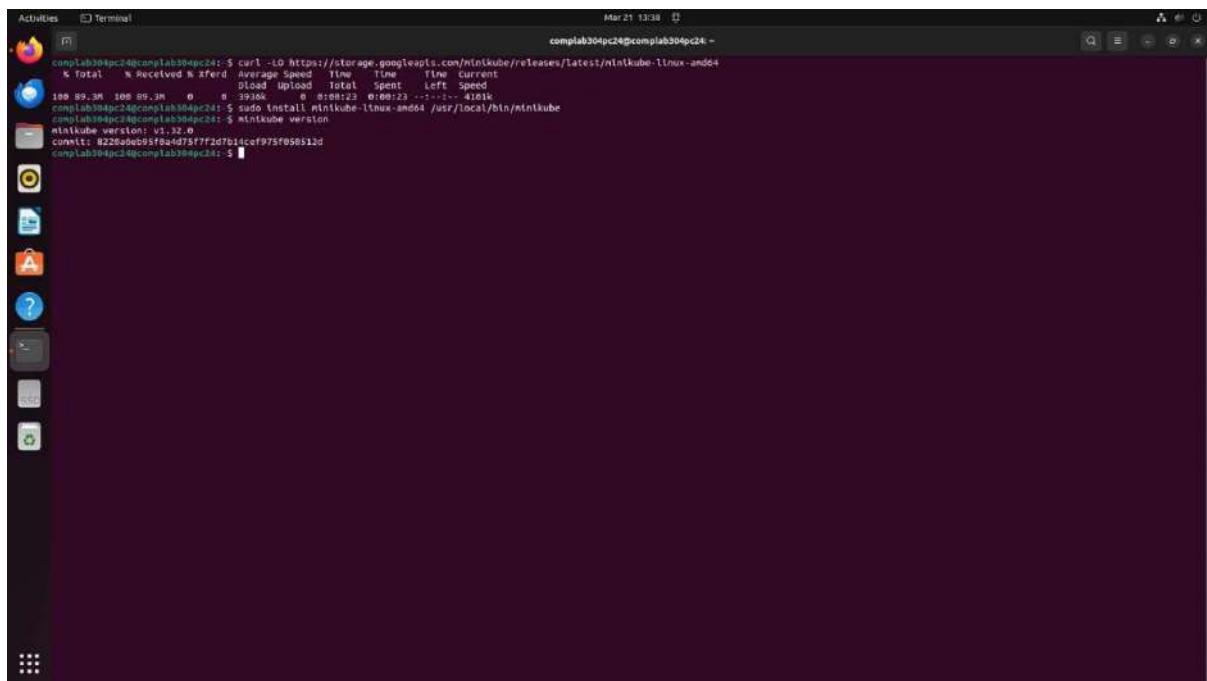
Step-5: Download and install Minikube Binary. To download and install minikube binary, run following commands,

```
$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
```

```
$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
```

To verify the minikube version, run

```
$ minikube version
```



Step-6: Install Kubectl Tool. Kubectl is a command line tool, used to interact with your Kubernetes cluster. So, to install kubectl run beneath curl command.

```
$ curl -LO https://storage.googleapis.com/kubernetes-release/release/' curl -s  
https://storage.googleapis.com/kubernetes-  
release/release/stable.txt` /bin/linux/amd64/kubectl
```

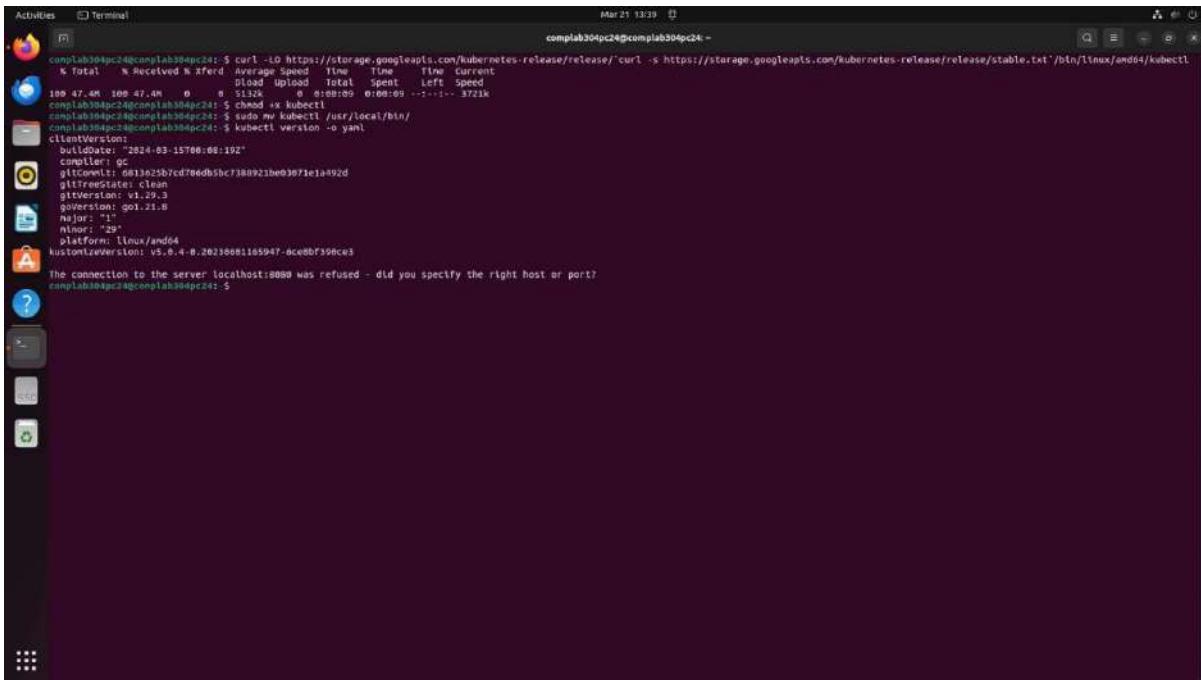
Next, set the executable permission on it and move to /usr/local/bin

```
$ chmod +x kubectl
```

```
$ sudo mv kubectl /usr/local/bin/
```

Verify the kubectl version, run

```
$ kubectl version -o yaml
```



```

Activities Terminal Mar 21 13:39
complab304pc24@complab304pc24: ~ curl -LO https://storage.googleapis.com/kubernetes-release/release/curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt /bin/linux/amd64/kubectl
  % Total    % Received % Xferd  Average Speed   Time   Time  Current
                                 Upload   Total   Spent  Left  Speed
 100 47.4M  100 47.4M    0     0  5132x      0  0:00:09  <---->  372k
complab304pc24@complab304pc24: ~ chmod +x kubectl
complab304pc24@complab304pc24: ~ sudo mv kubectl /usr/local/bin/
complab304pc24@complab304pc24: ~ kubectl version -o yaml
clientVersion:
  version: v1.22.0
  gitCommit: 88244-03-15T00:00:19Z
  compiler: gc
gitCommit: 0813e25b7cd76eddb5bc7388921be03e71e1a492d
gitTreeState: clean
gitVersion: v1.22.0
imageBuild: go1.18.0
imageMajor: "1"
imageMinor: "22"
platform: linux/amd64
kubernetesVersion: v1.22.0-0.20230805165947-0ce0bf390ce3

The connection to the server localhost:8080 was refused - did you specify the right host or port?
complab304pc24@complab304pc24: ~

```

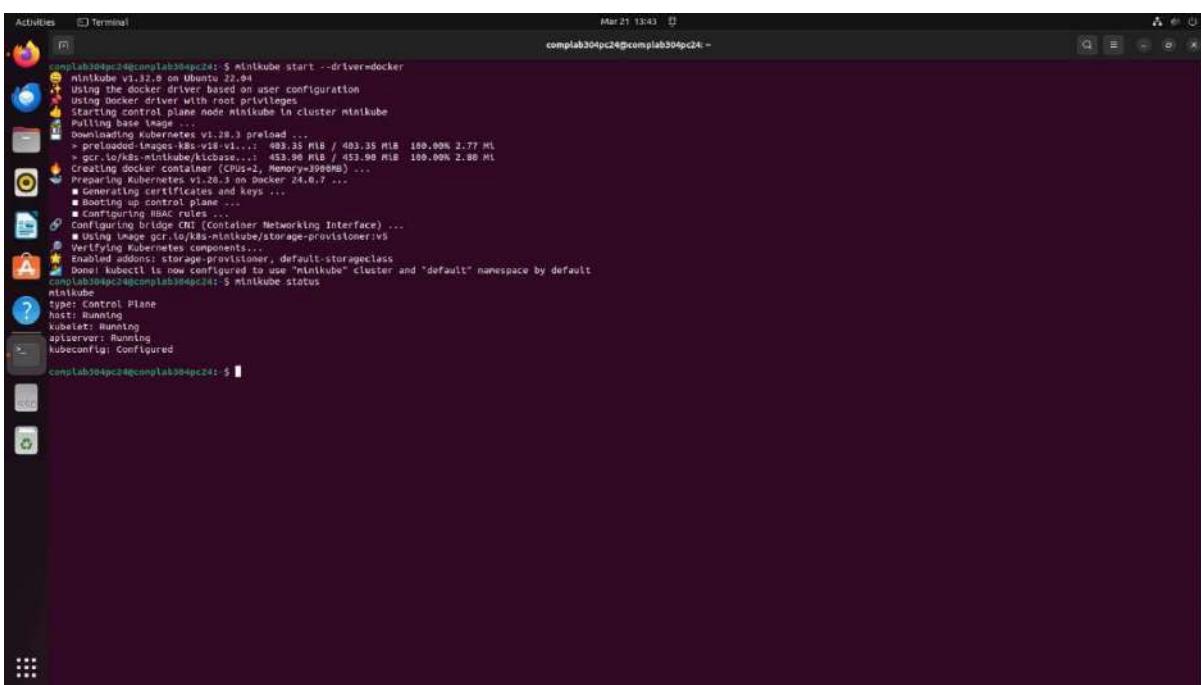
Step-7: Start Minikube Cluster. Now that Minikube is installed, start a Kubernetes cluster using the following command:

```
$ minikube start --driver=docker
```

This command initializes a single-node Kubernetes cluster, and it might take a few minutes to download the necessary components.

Once the minikube has started, verify the status of your cluster, run

```
$ minikube status
```



```

Activities Terminal Mar 21 13:43
complab304pc24@complab304pc24: ~ minikube start --driver=docker
minikube v1.32.0 on Ubuntu 22.04
Using Docker driver with root privileges
Starting control plane node minikube in cluster minikube
Pulling base image ...
Downloaded image k8s:v1.28.3 preload ...
> preloaded-image-k8s:v1.28.3 ...
  400.35 MB / 403.25 MB 100.00% 2.77 MiB/s
  453.90 MB / 453.90 MB 100.00% 2.88 MiB/s
Creating docker container (CPU=2, Memory=1996MB) ...
Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
  ■ Generating certificates and keys ...
  ■ Booting up the control plane ...
  ■ Configuring bridge CNI (Container Networking Interface) ...
  ■ Using Image gcr.io/k8s-minikube/storage-provisioner:v5
  ■ Verifying Kubernetes components...
  Enabled addons: storage-provisioner, default-storageclass
  2023-03-21 13:43:20.000000000 +0000 UTC [INFO][minikube.go:100] Configuring kubelet to use "minikube" cluster and "default" namespace by default
complab304pc24@complab304pc24: ~ minikube status
minikube
  type: Control Plane
  host: Running
  kubelet: Running
  apiserver: Running
  kubeconfig: Configured
complab304pc24@complab304pc24: ~

```

Step-8: Interact with your Minikube Cluster. Use kubectl to interact with your Minikube Kubernetes cluster. For example, you can check the nodes in your cluster:

```
$ kubectl get nodes
```

```
$ kubectl cluster-info
```



The screenshot shows a terminal window with the following command history and output:

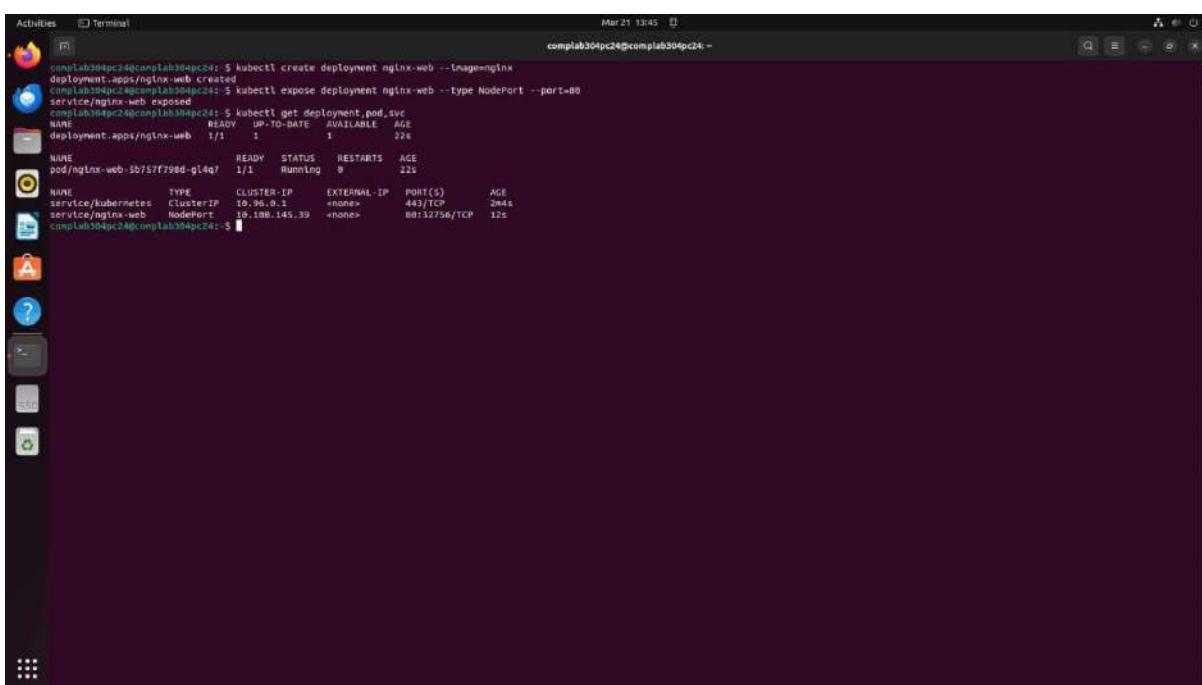
```
complab304pc24@complab304pc24: ~$ kubectl get nodes
NAME      STATUS    ROLES   AGE     VERSION
minikube   Ready    control-plane   2d       v1.22.0
complab304pc24@complab304pc24: ~$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.98.2:8443
CoreDNS is running at https://192.168.98.2:8443/api/v1/namespaces/kube-system/services/kube-dns:proxy
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
complab304pc24@complab304pc24: ~$
```

Step-9: Try to deploy a sample nginx deployment, run following set of commands.

```
$ kubectl create deployment nginx-web --image=nginx
```

```
$ kubectl expose deployment nginx-web --type NodePort --port=80
```

```
$ kubectl get deployment,pod,svc
```



The screenshot shows a terminal window with the following command history and output:

```
complab304pc24@complab304pc24: ~$ kubectl create deployment nginx-web --image=nginx
deployment.apps/nginx-web created
complab304pc24@complab304pc24: ~$ kubectl expose deployment nginx-web --type NodePort --port=80
service/nginx-web exposed
complab304pc24@complab304pc24: ~$ kubectl get deployment,pod,svc
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-web   1/1     1           1           22s
NAME          READY   STATUS    RESTARTS   AGE
pod/nginx-web-5b757f798d-g1q47   1/1     Running   0           22s
NAME        TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   10.96.0.1   <none>        443/TCP   2d44s
service/nginx-web   NodePort   10.108.145.39  <none>     80:32756/TCP   22s
complab304pc24@complab304pc24: ~$
```

Step-10: Managing Minikube Addons. If you want to add some additional functionality to your Kubernetes cluster like Kubernetes dashboard, ingress controller and more. You can enable these with addons. To view all the available addons, run

```
$ minikube addons list
```

ADDON NAME	PROFILE	STATUS	MAINTAINER
ambassador	minikube	disabled	3rd party (Ambassador)
auto-pause	minikube	disabled	minikube
cloud-spanner	minikube	disabled	Google
csi-hostpath-driver	minikube	disabled	Kubernetes
dashboard	minikube	disabled	Kubernetes
default-storageclass	minikube	disabled	3rd party (MetalLB)
gfk	minikube	disabled	3rd party (Elastic)
freshpod	minikube	disabled	Google
gcp-auth	minikube	disabled	Google
gitserver	minikube	disabled	minikube
gvisor	minikube	disabled	3rd party (Knative.io)
helm	minikube	disabled	3rd party (Helm)
helm-tiller	minikube	disabled	3rd party (Inacel)
inacel	minikube	disabled	[eloginacel.com]
ingress	minikube	disabled	Kubernetes
ingress-dns	minikube	disabled	Kubernetes
inspektoR-gadget	minikube	disabled	3rd party
istio	minikube	disabled	3rd party (Istio)
istio-provisioner	minikube	disabled	3rd party (Istio)
kubeflow	minikube	disabled	3rd party (Kong HQ)
kubevirt	minikube	disabled	3rd party (KubeVirt)
logteweaver	minikube	disabled	3rd party (unknown)
metallb	minikube	disabled	3rd party (MetalLB)
network-device-server	minikube	disabled	3rd party
nvidia-device-plugin	minikube	disabled	3rd party (Nvidia)
nvidia-driver-installer	minikube	disabled	3rd party (Nvidia)
nvidia-gpu-device-plugin	minikube	disabled	3rd party (Nvidia)
olm	minikube	disabled	3rd party (Operator Framework)
open-security-policy	minikube	disabled	3rd party (unknown)
portainer	minikube	disabled	3rd party (Portainer.io)
registry	minikube	disabled	minikube
registry-aliases	minikube	disabled	3rd party (unknown)
registry-creds	minikube	disabled	3rd party (UPMC Enterprises)
storage-provisioner	minikube	disabled	minikube
storage-provisioner-gluster	minikube	disabled	3rd party (Gluster)
storage-provisioner-rancher	minikube	disabled	3rd party (Rancher)
volumesnapshots	minikube	disabled	Kubernetes

Step-11: In order to enable addons, run

```
$ minikube addons enable dashboard
```

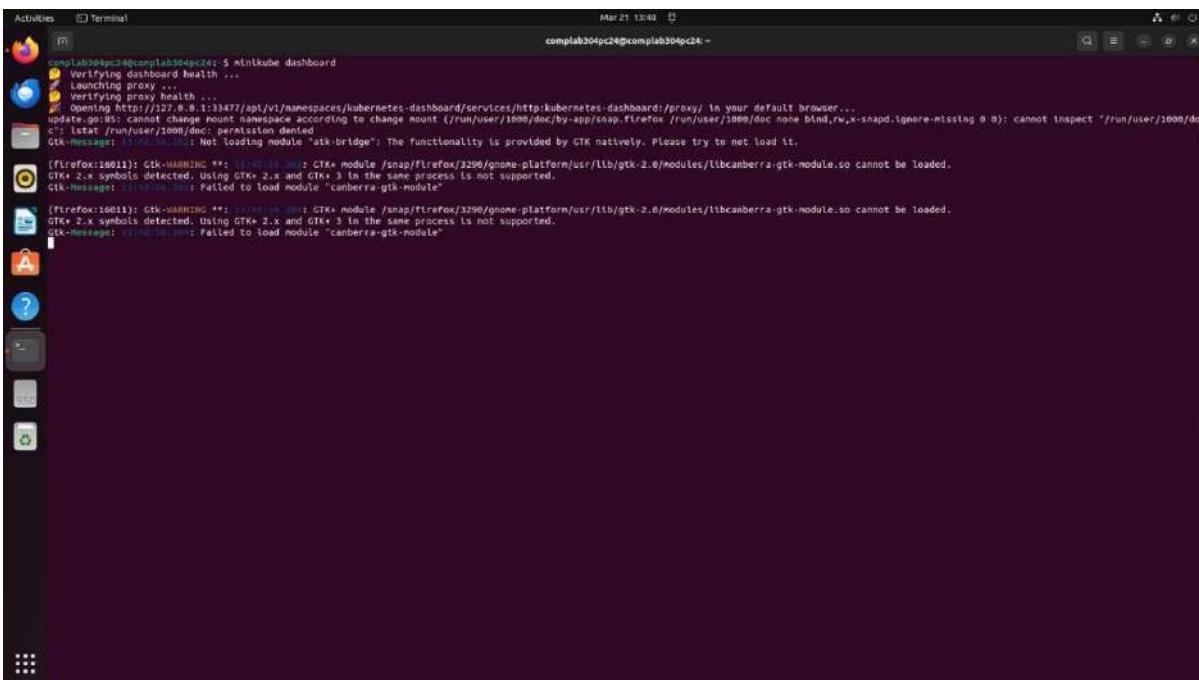
```
$ minikube addons enable ingress
```

```
minikube addons enable dashboard
dashboard is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at https://github.com/kubernetes/minikube/blob/master/OWNERS
Using Image docker.io/kubernetes-sul/metrics-scraper:v1.8.8
Some dashboard features require the metrics-server addon. To enable all features please run:
minikube addons enable metrics-server

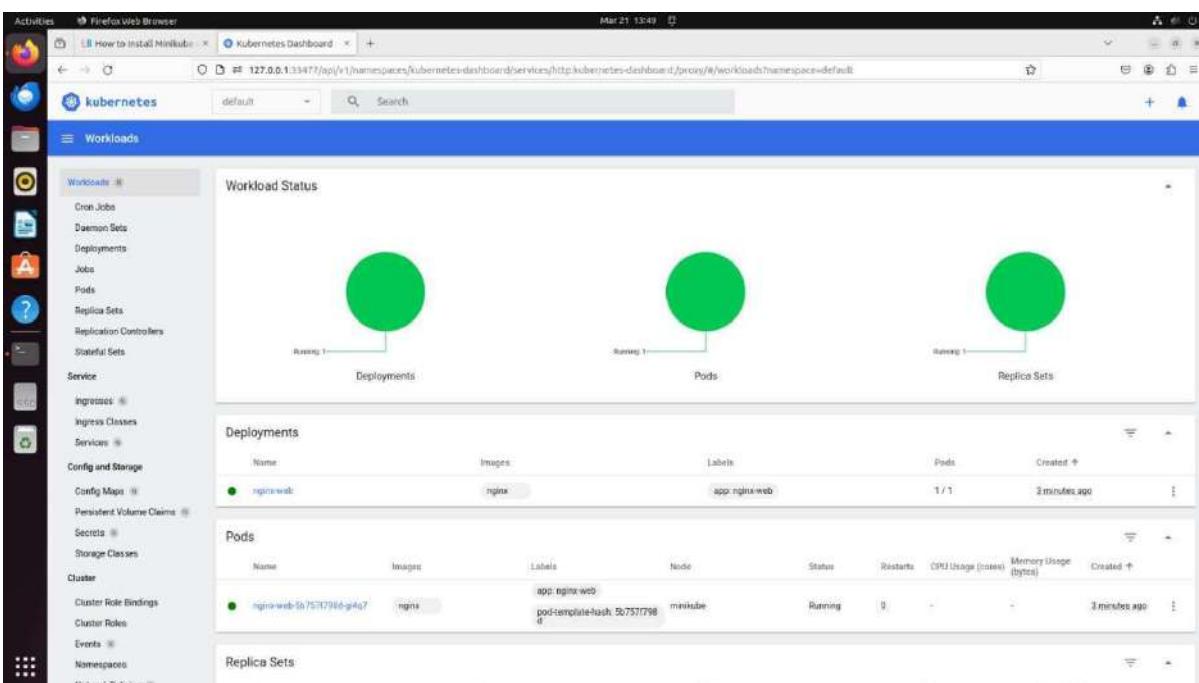
minikube addons enable ingress
The 'dashboard' addon is enabled
minikube addons enable ingress
ingress is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at https://github.com/kubernetes/minikube/blob/master/OWNERS
Using Image registry.k8s.io/ingress-nginx/controller:v1.9.4
■ Using Image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20231011-6b53cabe0
■ Using Image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20231011-6b53cabe0
Verifying ingress addon...
The 'ingress' addon is enabled
```

Step-12: To start the Kubernetes dashboard run below command, it will automatically launch the dashboard in the web browser as shown below:

```
$ minikube dashboard
```



```
Activities Terminal Mar 21 13:48
complab304pc24@complab304pc24: ~
$ minikube dashboard
Verifying dashboard health ...
Verifying proxy health ...
Opening http://127.0.0.1:33477/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
update.go:85: cannot change mount namespace according to change mount (/run/user/1000/doc/by-app/firefox /run/user/1000/doc none bind,rw,x-snapd.ignore-missing o ): cannot inspect '/run/user/1000/doc': lstat /run/user/1000/doc: permission denied
Gtk-Message: 13:48:00.000: Failed to load module "atk-bridge": The functionality is provided by GTK natively. Please try to not load it.
(Gtk:10011): Gtk-WARNING **: 13:48:00.000: GTK+ 2.x symbols detected. Using GTK+ 2.x and GTK+ 3 in the same process is not supported.
Gtk-Message: 13:48:00.000: Failed to load module "canberra-gtk-module"
(Gtk:10011): Gtk-WARNING **: 13:48:00.000: GTK+ 2.x symbols detected. Using GTK+ 2.x and GTK+ 3 in the same process is not supported.
Gtk-Message: 13:48:00.000: Failed to load module "canberra-gtk-module"
```



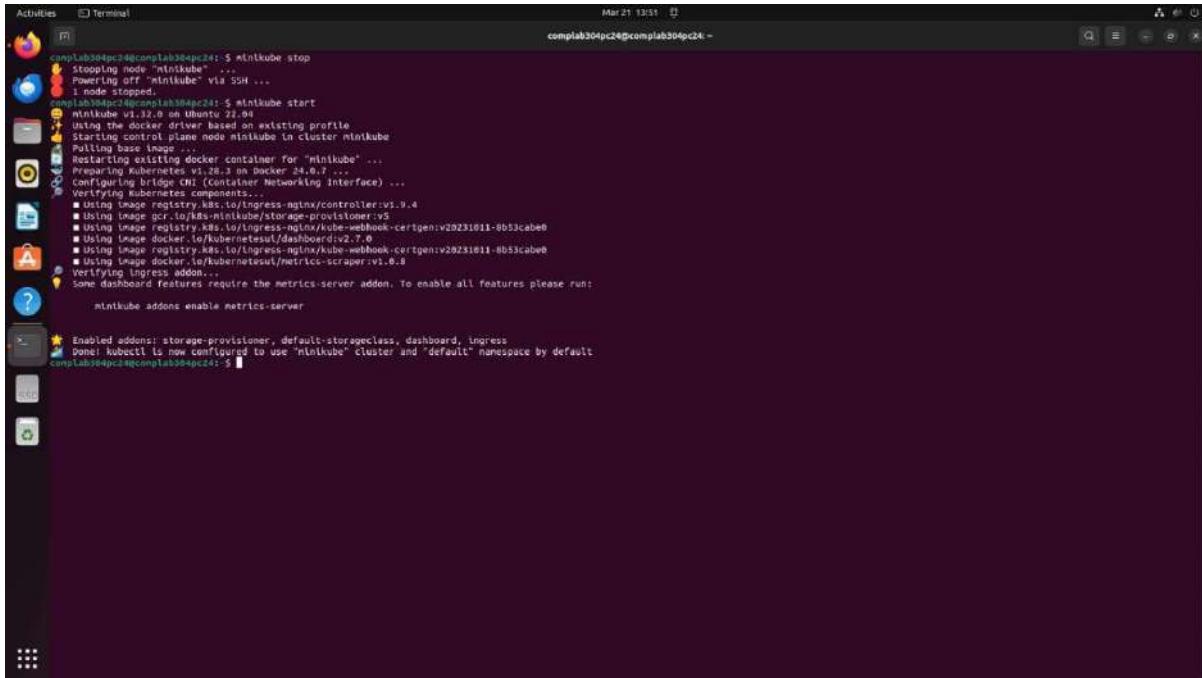
Step-13: Managing Minikube Cluster. To stop and start the minikube cluster, run beneath commands.

```
$ minikube stop
```

```
$ minikube start
```

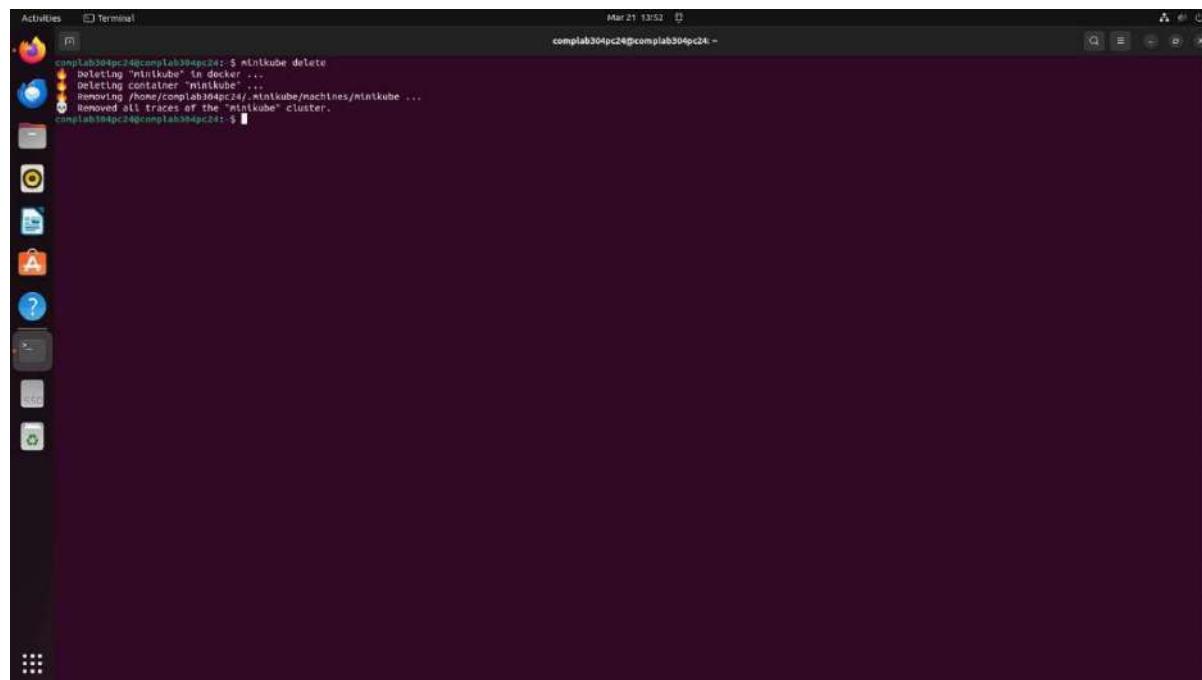
In order to delete the minikube cluster, run

```
$ minikube delete
```



```
Activities Terminal Mar 21 13:51 commlab304pc24@commlab304pc24: ~
commlab304pc24@commlab304pc24: ~$ minikube stop
Stopping "minikube"
Powering off "minikube" via SSH ...
1 node stopped.
commlab304pc24@commlab304pc24: ~$ minikube start
Starting the VM...
Using the existing driver based on existing profile
Starting control plane node minikube in cluster minikube
Pulling base image ...
Restarting existing docker container for "minikube" ...
Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
Creating host port mappings (container ports 443, 2375, 2376, 8080 to networking interface) ...
Verifying Kubernetes components...
  ● Using Image registry.k8s.io/ingress-nginx/controller:v1.9.4
  ● Using Image gcr.io/k8s-minikube/storage-provisioner:v5
  ● Using Image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20231011-8b53cabef
  ● Using Image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20231011-8b53cabef
  ● Using Image docker.io/kubernetes-sa/metrics-scraper:v1.8.8
Verifying Ingress addon...
Some dashboard features require the metrics-server addon. To enable all features please run:
  minikube addons enable metrics-server

* Enabled addons: storage-provisioner, default-storageclass, dashboard, ingress
Done! Kubernetes is now configured to use "minikube" cluster and "default" namespace by default
commlab304pc24@commlab304pc24: ~$
```



```
Activities Terminal Mar 21 13:52 commlab304pc24@commlab304pc24: ~
commlab304pc24@commlab304pc24: ~$ minikube delete
Deleting cluster "minikube" ...
  Deleting container "minikube" ...
    Removing '/home/commlab304pc24/.minikube/machines/minikube' ...
  Removed all traces of the "minikube" cluster.
commlab304pc24@commlab304pc24: ~$
```

Written Assignment No 1

(a) i.) What is Fog computing? Explain need of Fog computing.

⇒ Fog computing is distributed computing paradigm that extends cloud computing capabilities to the 'edge' of the network, closer to where data is generated & consumed by intelligent devices in IoT.

Key factors during need for Fog computing.

① Reduced Latency:-

→ Real time application demanding multisecond response can't rely on cloud processing due to network delays.

② Bandwidth limitation

→ Large scale data transmission from massive IoT deployment can overwhelm network bandwidth.

③ Security & privacy

→ Sensitive data may require local processing to comply with privacy regulation or security requirement or security device requirements.

(i) Device heterogeneity :- Edge devices have device computational & storage capabilities. Fog computing adapts to this heterogeneity.

(ii) Improved reliability :-

Fog computing provides distributed intelligence, making applications less susceptible to single points.

a2) Explain the architecture of Fog Computing.

→ A Fog computing architecture typically consists

- smart devices: IoT devices that generate & collect data
- Fog Nodes: - Computing devices deployed at the edge.
- Fog Aggregators: - Nodes aggregating data from multiple fog nodes, performing higher level processing connecting to cloud using high speed internet connection.

a3) Explain application of Fog computing:-

- Fog computing in diverse domain, including

- Smart cities :- Real time traffic management smart guides, environmental monitoring.
- Industrial IoT :- Predictive maintenance, remote asset monitoring, anomaly detection, real time process control.
- Connected health care :- Remote patient monitoring, medical imaging analysis
- Autonomous vehicles :- Real time collision avoidance, path optimization, traffic signal coordination.

Q4.) Explain Fog, edge & cloud computing.

The terms are often used interchangeably, but subtle difference exists :-

- Edge computing :- Processing data directly on a near the device itself, with minimal reliance on other computing resource.
- Fog computing :- Processing data close to source but potentially involving multiple devices.
- Cloud Computing Centralized processing & storage in large data centers offering massive scalability & powerful compute resource.

Choosing the right paradigm depends on factor like :-

- Required latency:- Edge for ultra low latency fog over real time & near-real time, cloud over but before latency.
- Data volume & processing need: Edge for using small data, fog for moderate data
- Cost & complexity:- Edge is simplest, fog offer a balance.

Q 5.) Case study of fog computing

Scenario :- A smart city implement a real time traffic management system

- Edge:- Traffic sensor collect data .
- Fog Nodes:- Process sensor data, identifies
- Fog Aggregation:- Aggregate traffic data from multiple fog nodes, optimizes city road traffic flow
- Cloud :- Stores historical traffic data provides long-term insights & trends

Benefits :-

- Reduced traffic congestion
- Improved air quality
- Shorter travel time
- Enhanced safety.

Additional considerations:-

- Security & privacy concern require careful data handling practice
- Standards & interoperability are still evolving
- Fog Computing cost effectiveness compared to cloud only solution varies based on specific use cases.

(A)

S.P.
26/3/2021

Written Assignment - 2

- Q.) Explore & compare the similar types of service provided by AWS, Azure & Google Cloud Platform.

Theory :-

To explore and compare the similar types of services provided by AWS, Azure and Google Cloud Platform, there are ten key service offered by each cloud provider along with example :-

1.) Compute Services

- Service :- Amazon Elastic Compute Cloud (EC2)
- Description :- Provide reusable compute capacity in cloud.
- Example :- Running virtual server for various application like website, mobile apps & databases.

2.) Storage Services :-

- Service :- Amazon Simple Storage (S3)
- Description :- Object storage service for storing and retrieving data
- Example :- Storing media files, backup and data archives securely.

3.) Database Services

- Services : Amazon Relational Database Service (RDS)
- Description :- Managed relational Database service
- Example :- Hosting MySQL, PostgreSQL or SQL Server databases

4.) Networking Services

- Services :- Amazon Virtual Private Cloud (VPC)
- Description :- Securely connect AWS resources to an isolated virtual network.
- Example :- Creating a private network for application with specific security requirements

5.) Machine Learning Services

- Service :- Amazon SageMaker
- Description :- Fully managed service to build, train and deploy machine learning model.
- Example :- Developing predictive analytics model for business insights

6.) Compute Service

- Service - Azure Virtual Machine
- Description - On demand scalable computing resource
- Example :- Hosting Windows or Linux virtual machine in cloud.

7.) ~~Azure Blob Storage~~ :- Provides object storage for unstructured data

Example :- Storage media files like images and video.

8.) Azure SQL Database :- Managed relational database service

Example :- Hosting a SQL server database

9.) Google Compute Engine :- Offers virtual machine for computing resource

Example :- Running application on Google infrastructure.

10.) Google Cloud Storage :- Provides scalable object storage for data

Example :- Storing and serving static static website content

11.) Azure Functions

- Supports serverless computing for event-driven application

Eg:- Running code in response to triggers

12.) Azure Virtual Network

- Enables private and secure communication between Azure resources

- Example - Creating a virtual network for Azure services.

13.) Google Virtual Private Cloud (VPC) :- Offer networking functionality for creating isolated environments.

Example:- Setting up a secure network for GCP.

There are many more.

These services represent a selection of core offerings from AWS, Azure & Google Cloud Platform, showcasing their capabilities in providing essential cloud computing resources & services for various business needs.

*Thadomal Shahani Engineering
College, Bandra (West)*

TOPIC : “ Todo List using AWS Lambda DynamoDb and Amplify Services ”

SUBJECT : CLOUD COMPUTING LAB MINI PROJECT

SUPERVISOR : TANUJA MAAM

GROUP MEMBERS :

- Shirish Shetty 2103164
- Adarsh Shukla 2103167

Description of problem statement:

To meet our objective of developing an easy-to-use task management application using AWS services, we carefully selected and integrated the following components:

1. **AWS Amplify:** This service allowed us to effortlessly host the user interface of our application. We leveraged its simplified configuration and automated deployment features, enabling quick setup and updates of our application's frontend. By connecting directly to other AWS services, Amplify facilitated seamless integration and management.
2. **DynamoDB:** As our primary database solution, DynamoDB efficiently stored and managed the structured and semi-structured data generated by our application users. Its flexible data model and auto-scaling capabilities were well-suited for our serverless workload. By utilizing DynamoDB, we ensured high performance and reliability while accommodating the dynamic nature of our application.
3. **AWS Lambda:** We employed Lambda to handle custom backend logic triggered by various events within our application. This included tasks such as processing user input from the frontend or reacting to changes in the DynamoDB database. By decoupling our backend logic into individual Lambda functions, we promoted modularity and simplified debugging. This architecture also ensured eventual consistency and reduced coupling between different components of our application.

By effectively utilizing AWS Amplify, DynamoDB, and AWS Lambda, we created a robust and user-friendly task management application. The seamless integration of these services enabled us to focus on delivering a smooth user experience while ensuring scalability, reliability, and cost-effectiveness.

on complex scenarios.

Overall, combining AWS, Amplify, DynamoDB, and Lambda yields powerful benefits, promoting agile development, operational flexibility, and financial prudence while delivering first-class experiences to users

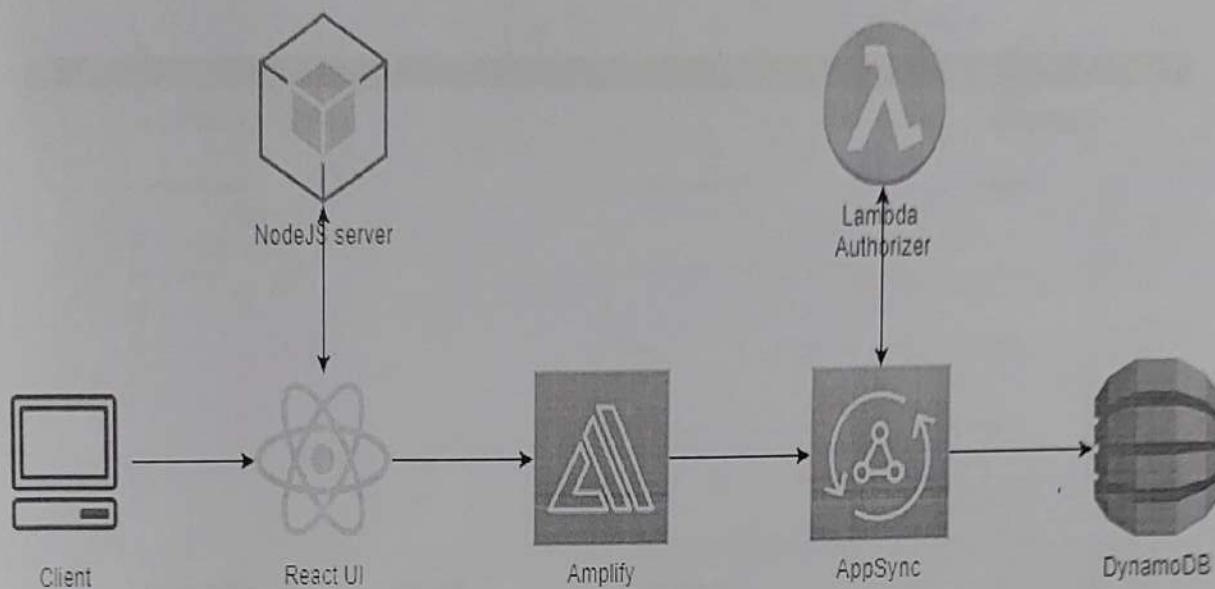
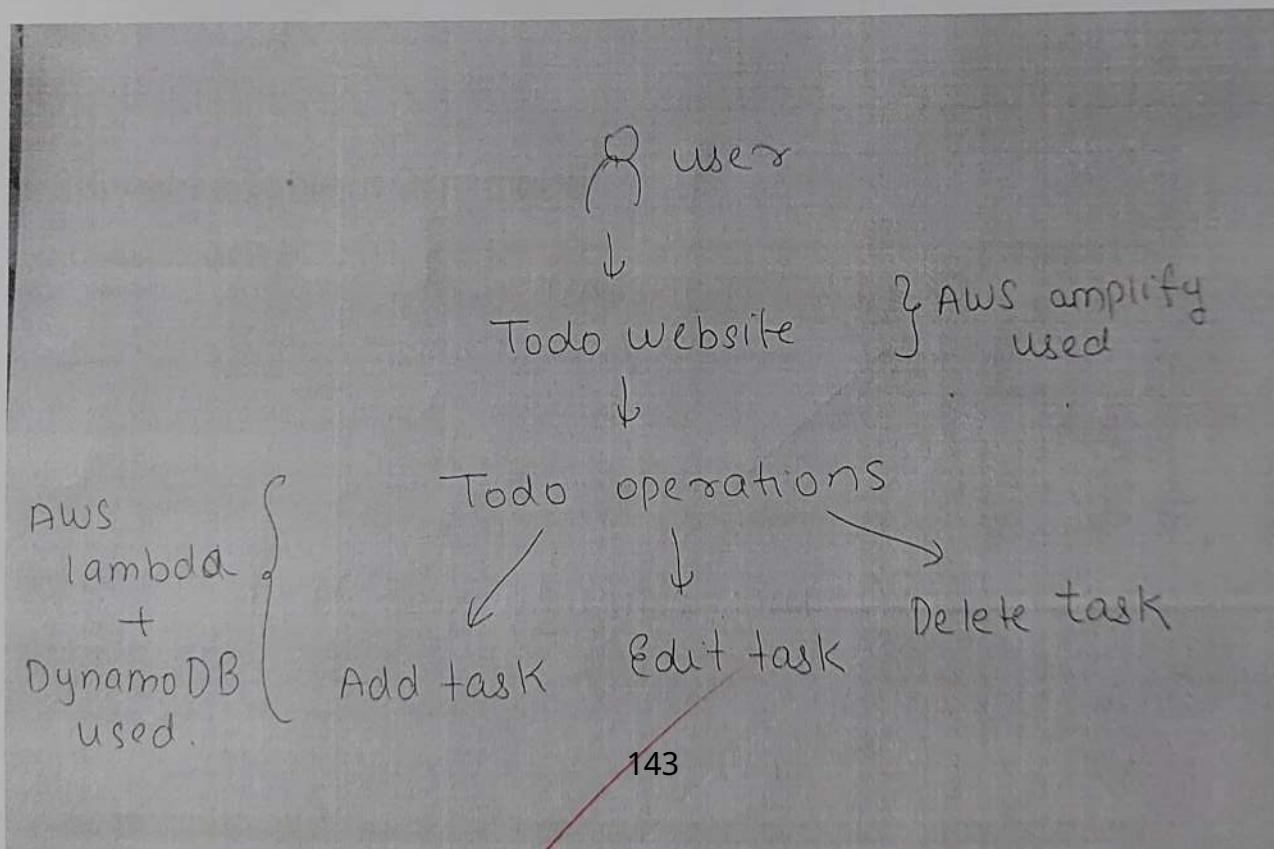
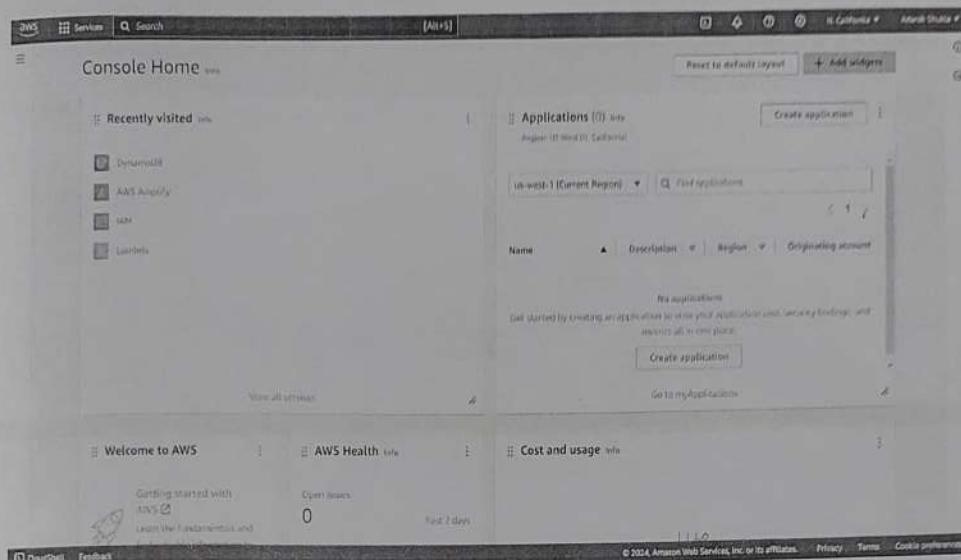


Fig) React App working with Aws Services

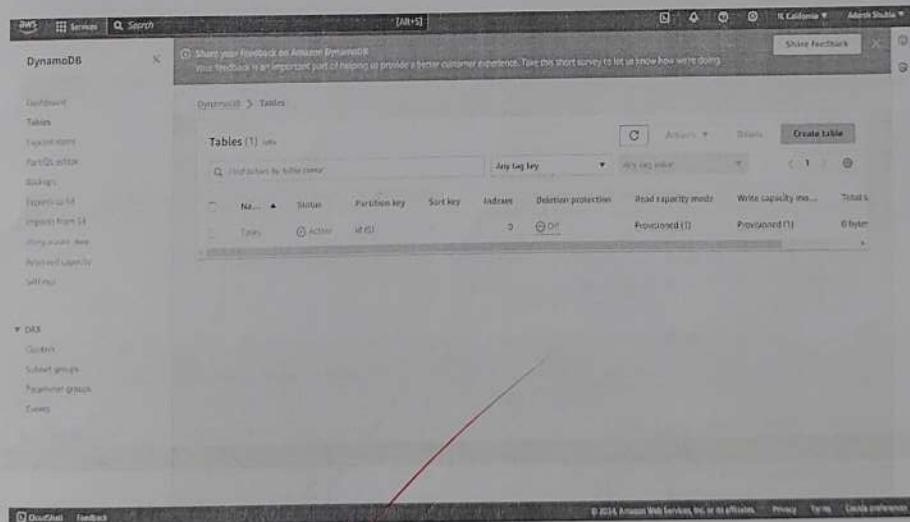


Major Steps And Codes

- 1) Create a AWS Account and go to the Console



- 2) Create a Table in DynamoDb



3) Host the UI part in Aws Amplify

The screenshot shows the AWS Amplify console interface for the 'CCL-ui' app. On the left, a sidebar lists various settings like App settings, General, Amplify Studio settings, and Monitoring. The main area displays the app's homepage with a heading 'CCL-ui'. Below it, a message says 'The app homepage lists all deployed frontend and backend environments.' A button 'Actions' is visible. In the center, there are tabs for 'Hosting environments' and 'Backend environments'. Under 'Hosting environments', a section titled 'main | ui' shows 'Continuous deploys set up (Edit)'. Below this, three circular icons represent the build pipeline: 'Provision' (checkmark), 'Build' (checkmark), and 'Deploy' (checkmark). At the bottom, there are links for 'Documentation' and 'Support', along with deployment details: 'Last deployment' (3/27/2024, 2:45:32 PM), 'Last commit' (Auto-build | GitHub), and 'Preview' (Disabled). The footer includes standard AWS links like 'Feedback' and copyright information.

4) Deploy the api and create lambda function in Aws Lambda

The screenshot shows the AWS Lambda console for the 'task-list-api' function. The top navigation bar includes 'Lambda', 'Functions', and 'task-list-api'. The main area has tabs for 'Function overview' (selected) and 'Info'. On the left, there are buttons for 'Diagram' and 'Template'. The 'Function overview' tab shows a box for 'task-list-api' with a 'Layers' section containing '(0)'. To the right, there are buttons for 'Throttle', 'Copy ARN', and 'Actions'. Below these are sections for 'Description', 'Last modified' (1 hour ago), 'Function ARN' (arn:aws:lambda:us-west-1:891377318888:function:task-list-api), and 'Function URL' (info). A red arrow points from the 'task-list-api' box towards the 'Function URL' section. The right side of the screen features a 'Create a simple web app' tutorial with steps: 'Build a simple web app consisting of a Lambda function with a function URL, that outputs a webpage.', 'Invoke your function through a function URL.', and a 'Start tutorial' button. The footer includes 'Code', 'Test', 'Monitor', 'Configuration', 'Aliases', 'Versions', 'Upload from', 'Code source', and 'Feedback'.

```
● ● ●  
export const API_URL = "https://7brlmecg6rz2gxmf4fczjpekoecgqlt.lambda-url.us-west-1.on.aws/task";
```

Fig) API URL with Lambda Url and entry point

```

import express from "express";
import serverless from "serverless-http";
import cors from "cors";
import { fetchTasks, createTasks, updateTasks, deleteTasks } from "./task.js";

const app = express();
const port = 3001;

app.use(express.json());

if (process.env.DEVELOPMENT) {
    app.use(cors());
}

app.get("/", (req, res) => {
    res.send("Hello World!");
});

app.get("/task", async (req, res) => {
    try {
        const tasks = await fetchTasks();
        res.send(tasks.Items);
    } catch (err) {
        res.status(400).send(`Error fetching tasks: ${err}`);
    }
});

app.post("/task", async (req, res) => {
    try {
        const task = req.body;
        const response = await createTasks(task);
        res.send(response);
    } catch (err) {
        res.status(400).send(`Error creating tasks: ${err}`);
    }
});

app.put("/task", async (req, res) => {
    try {
        const task = req.body;
        const response = await updateTasks(task);
        res.send(response);
    } catch (err) {
        res.status(400).send(`Error updating tasks: ${err}`);
    }
});

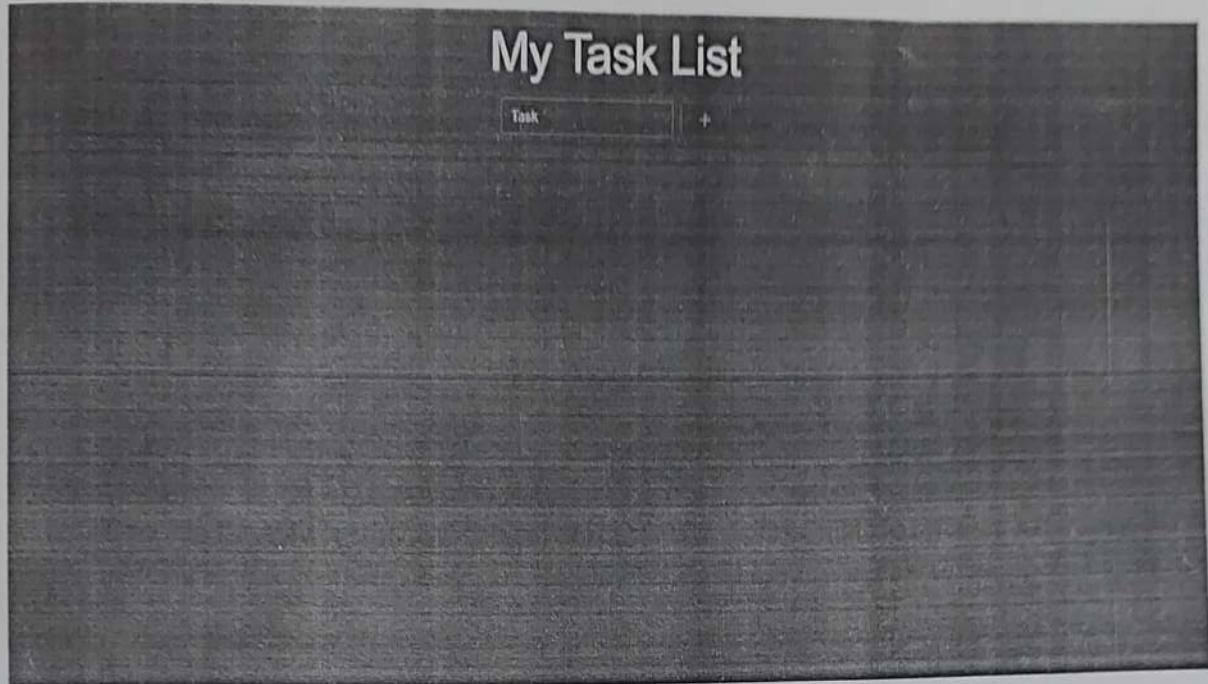
app.delete("/task/:id", async (req, res) => {
    try {
        const { id } = req.params;
        const response = await deleteTasks(id);
        res.send(response);
    } catch (err) {
        res.status(400).send(`Error deleting tasks: ${err}`);
    }
});

if (process.env.DEVELOPMENT) {
    app.listen(port, () => {
        console.log(`Example app listening on port ${port}`);
    });
}

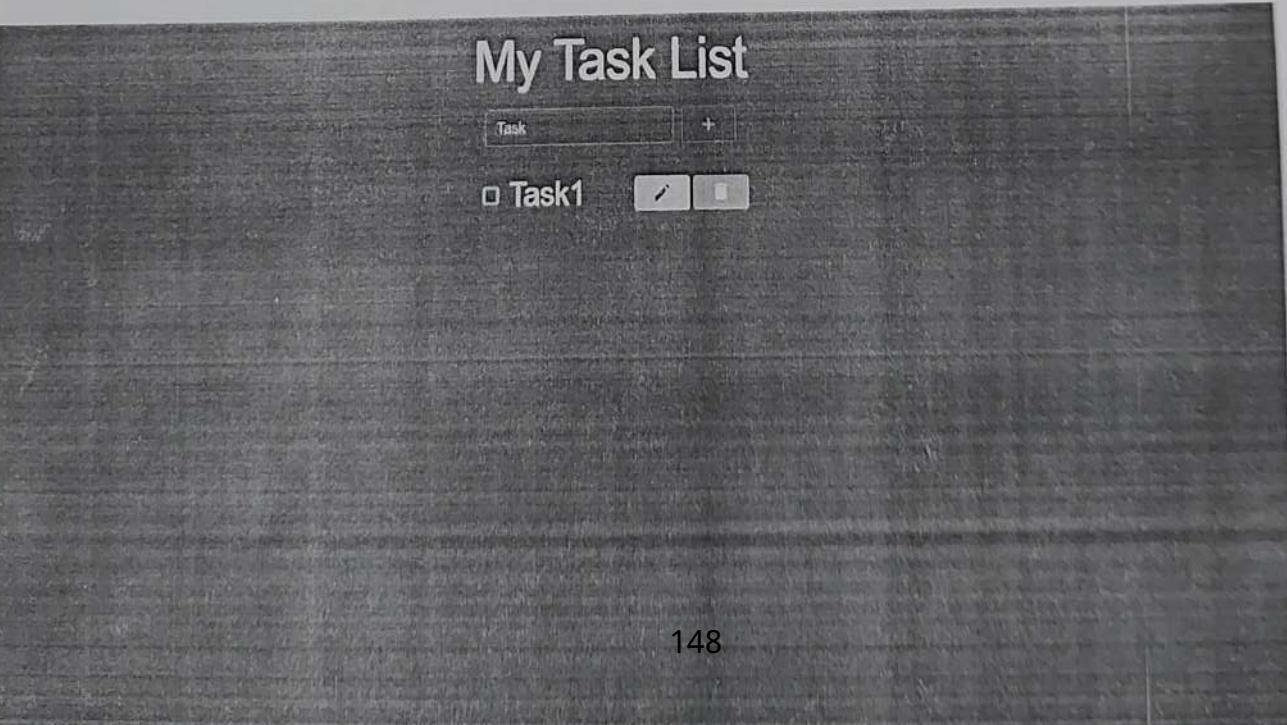
export const handler = serverless(app);

```

Screenshots of project in sequence



Task List Ui Hosted Live



My Task List

Shirish Shetty 2103164 C32

Task

+

Task1



Edit Task

Task

Task11



Edit of the task

My Task List

Shirish Shetty 2103164 C32

Task

+

Task11



Completion of the task

DynamoDB > Explore items > Edit item

Form | JSON view

Edit item

You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. Learn more [\[?\]](#)

Add new attribute [\[▼\]](#)

Attributes	Type	Action
Attribute name: id - Primary key Value: e0000332-edb7-46d1-87c3-239325ee2d5a	String	Remove
Attribute name: completed Value: True	Boolean	Remove
Attribute name: name Value: Task11	String	Save and close

Cancel Save Save and close

Live Updation Done in Dynamo Db

By utilizing AWS services like Amplify, DynamoDB, and Lambda, we've built a complex setup. Here's what we've achieved:

- Easy Building Blocks: AWS Amplify provides ready-made tools to create the user interface of your app and deploy it online effortlessly. It's like having a Lego kit to quickly assemble your project.
- Efficient Data Storage: DynamoDB acts as a smart storage system where your app stores information. It's like having a super-fast, organized filing cabinet for your data.
- Smooth Operations: AWS Lambda helps in running your app smoothly. It's like having a personal assistant who takes care of tasks behind the scenes, ensuring everything runs without a hitch.
- Cost-Efficiency: With AWS, you only pay for what you use. This means you don't waste money on resources you don't need, making it budget-friendly for students or anyone on a tight budget.
- Reliable Infrastructure: AWS services are designed to be highly reliable and available, ensuring your app is always accessible to users whenever they need it.

Overall, by leveraging these AWS services, we created a robust and efficient application infrastructure, setting us up for success in your projects and learning journey without getting bogged down by technical complexities.

A
28/3/22