

Thadomal Shahani Engineering College

Bandra (W.), Mumbai- 400 050.

© CERTIFICATE ©

Certify that Mr./Miss Parth Dabholkar of Computer Department, Semester VI with Roll No. 2103032 has completed a course of the necessary experiments in the subject Cloud Computing under my supervision in the **Thadomal Shahani Engineering College** Laboratory in the year 2023 - 2024

Teacher In-Charge

Head of the Department

Date 13/11/2023

Principal

CONTENTS

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1.	Introduction, Overview of Cloud Computing	1 - 9	18/11/24	
2.	To study and implement Host virtualization using Virtual Box	10 - 22	19/11/24	
3.	To study and implement Bare Metal virtualization using KVM.	23 - 42	25/11/24	
4.	To study and implement Infrastructure as a Service using AWS.	43 - 53	21/12/24	
5.	To study and implement Platform as a Service using Beanstalk.	54 - 60	8/12/24	
6.	Study & Implement Storage as a Service	61 - 74	22/12/24	
7.	Study & Implement IAM.	75 - 83	23/12/24	DAT
8.	Study and Implement Database as a service on AWS RDS.	84 - 101	29/12/24	13/4/2024
9.	Study and Implement Containerization using Docker.	102 - 113	11/1/24	
10.	Study and implement container orchestration using Kubernetes	114 - 124	7/1/24	
11.	Assignment 1	125 - 130	14/1/24	
12.	Assignment 2	131 - 136	15/1/24	
13.	Mini Project	137 - 149	28/1/24	

Experiment no: 1

Aim: Introduction and overview of cloud computing.

• Definition of cloud computing

Cloud computing is a revolutionary paradigm that transforms the way we access, store and manage digital information and computing resources. At its core, cloud computing involves the delivery of a variety of IT services, including computing power, storage, networking, databases, analytics and software, over the internet.

National Institute of Standards and Technology (NIST) defines Cloud Computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Cloud computing is a general term for anything that involves delivering hosted services over the internet. These are broadly divided into three categories: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service.

Let us discuss different characteristics of cloud computing as described by NIST:

- Characteristics of cloud computing:

- i) On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service.
- ii) Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (Eg: Mobile Phones, Tablets, laptops and workstations).
- iii) Resource Pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location of the provided resources but may be able to specify location at a higher level of abstraction.
- iv) Rapid elasticity: Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.
- v) Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service.

- NIST cloud computing model

National Institute of Standard and Technology states that cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g.: networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

- Different models of Cloud Computing

- * Service Models:

Software as a service (SaaS): The capability provided to the consumer is to use provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as web browser (e.g.: web-based email), or a program interface. The consumer does not manage or control underlying cloud infrastructure including networks, servers, operating systems, storage or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

Platform as a Service (PaaS): The capability provided to the consumer is to deploy onto the cloud infrastructure consumer created or acquired applications created using programming languages, libraries, services and tools supported by the provider. The consumer does not manage or control the underlying

cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

Infrastructure as a Service (IaaS): the capability provided to the consumer is to provision processing, storage, networks, and other fundamentals computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage and deployed applications and possibly limited control of select 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 (e.g. business team/units). It may be owned, managed and operated by organization, a third party, or some combination of them, and it may exist on or off premises.

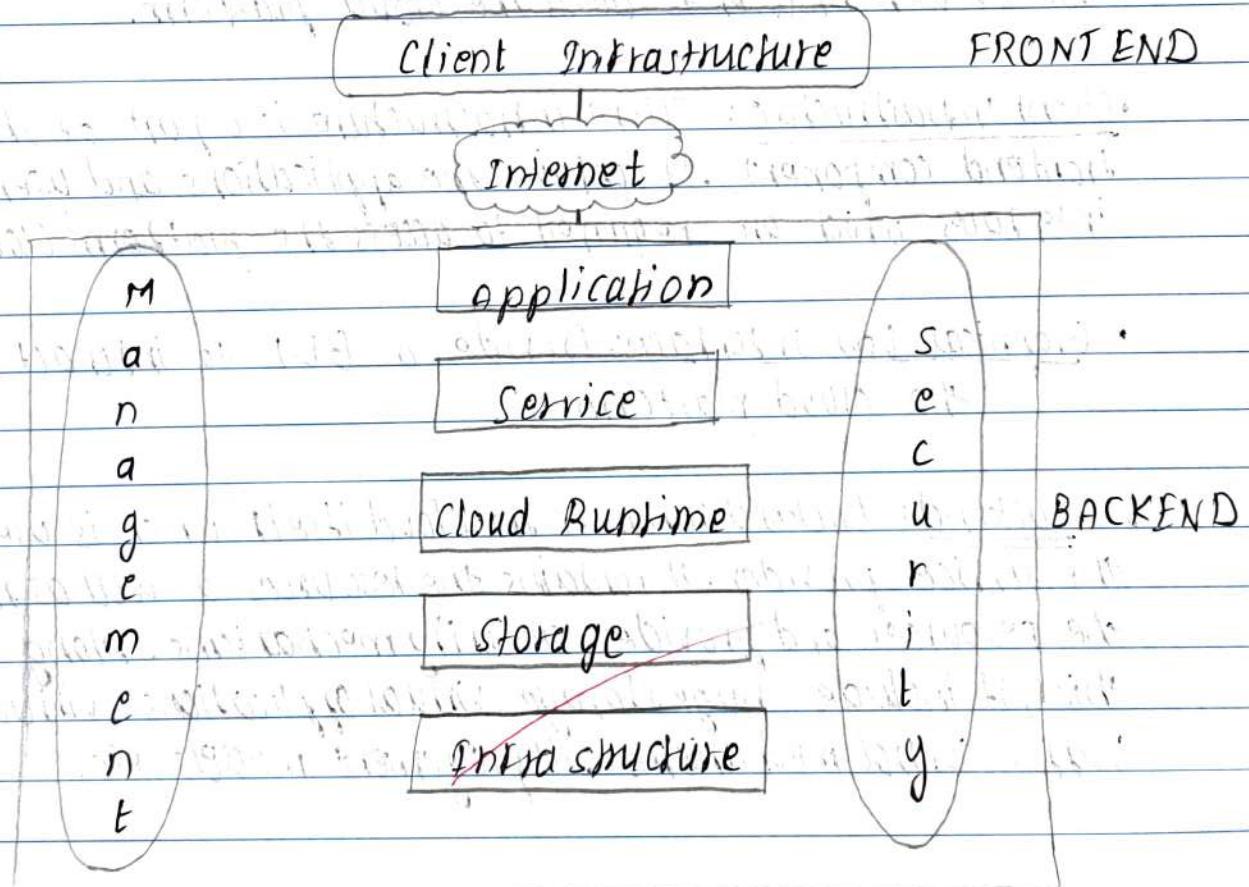
Community cloud: the cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g. mission, security requirements, policy and compliance considerations). It may be owned, managed, and operated by one or more organizations in the community, a third party, or some combination of them, and

it may exist on or off premise.

Public Cloud: The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

Hybrid Cloud: The cloud infrastructure is a composition of two or more distinct cloud infrastructure (private, community or public) that remains 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

Architecture of cloud computing is combination of both SOA (Service Oriented Architecture) and EDA (Event Driven Architecture). Client infrastructure, application, service, runtime, cloud, storage, infrastructure, management and security all these are the components of cloud computing architecture.

1. Frontend: Frontend of the cloud architecture refers to the client side of cloud computing system. It contains all the user interface and applications which are used by the client to access the cloud computing services/resources. For example, use of web browser to access the cloud platform.

• Client Infrastructure: Client infrastructure is a part of the frontend component. It contains the applications and user interfaces which are required to access the platform cloud.

• Graphical User Interface: Provides a GUI to interact with the cloud platform.

2. Backend: Backend refers to the cloud itself which is used by the service provider. It contains the resources as well as manages the resources and provides security mechanisms. Along with this, it includes huge storage, virtual applications, virtual machines, traffic control mechanisms, deployment models etc.

1. Application: Application in backend refers to a software or platform to which client accesses. Means it provides the services in backend as per the client requirement.
2. Service: Service in backend refers to the major three types of clouds based services like SaaS, PaaS and IaaS. Also manages which type of service the user accesses.
3. Runtime Cloud: Runtime cloud in backend provides the execution and runtime platform environment to virtual machine.
4. Storage: Storage in backend provides flexibility and scalable storage service and management of stored data.
5. Infrastructure: Cloud infrastructure in backend refers to the hardware and software components like servers, storage, network devices etc.
6. Management: Refers to management of backend components like application, service, runtime cloud, storage, infrastructure and other security mechanism.
7. Security: Refers to implementation of different security mechanisms in backend for secure cloud resources, systems, files and infrastructure to end users.
8. Internet: Acts as a medium or a bridge between frontend and backend and establishes interaction between frontend and backend.

9. Database: Refers to provide database for storing structured data, such as SQL or noSQL databases. Eg are Amazon AWS, Microsoft Azure, Google Cloud, One Drive etc.

10. Networking: Provides networking infrastructure for the application in the cloud, such as load balancing, DNS and virtual private networks.

11. Analytics: Provides analytical capabilities of data in cloud, such as warehousing, business intelligence and machine learning.

• Benefits and Limitations of cloud computing

1. Cost Efficiency: Cloud computing eliminates the need for significant upfront investments in hardware and infrastructure, allowing users to pay for only the resources they consume.

2. Scalability: Users can easily scale their computing resources up or down based on demand, ensuring optimal performance without overprovisioning.

3. Flexibility: They offer a wide range of application and services, providing flexibility for businesses to choose and customize solutions that meet their specific needs.

4. Accessibility: Users can access cloud services from anywhere with an internet connection, promoting remote work and collaboration.

5. Automatic Updates: Cloud providers handle server maintenance, security updates, and software patches, ensuring that users always have access to the latest features and improvements.

* Limitations:

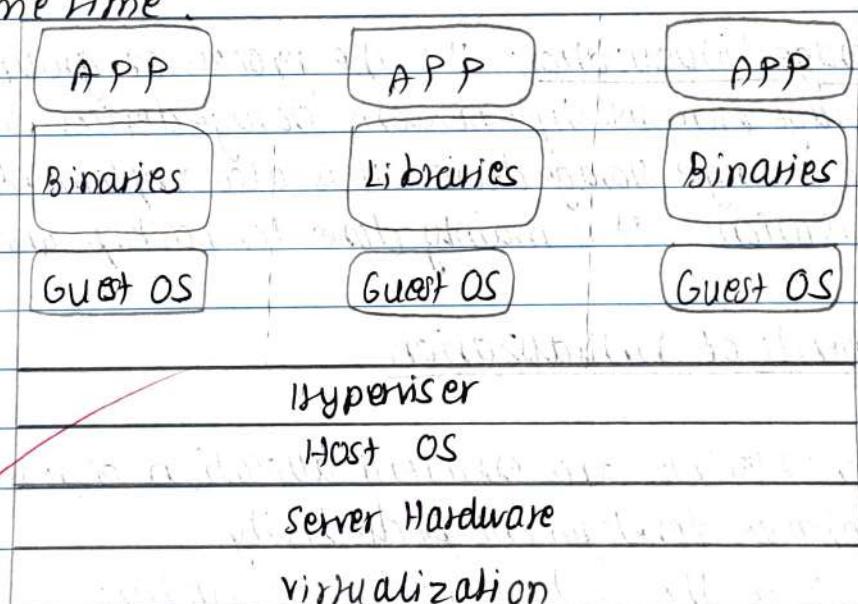
1. Security concerns: Storing data on remote servers raises security issues, and business must trust cloud providers to implement robust security measures.
2. Downtime: Reliance on internet connectivity means that if the network or cloud provider experiences downtime, users may lose access to applications and data.
3. Limited Customization: Some cloud services may have limitations in terms of customization, making it challenging for businesses with specific requirements.
4. Dependency on Service Providers: Organizations may become dependent on the reliability and performance of cloud service providers, and any issues on their end can impact the user's experience.
5. Potential for Hidden Costs: While cloud computing can be cost-effective, unexpected expenses may rise, such as data transfer costs or additional fees for exceeding usage limits.

(A+) 15/3/2024

Experiment no: 2

AIM: To study and implement Hosted virtualization using Virtual Box and KVM.

- Virtualization in Cloud Computing: Virtualization is a technique of how to separate a service from the underlying physical delivery of that service. It is a process of creating a virtual version of something like computer hardware. It involves using specialized software to create a virtual or software created version of computing resource rather than actual version of the same resource. With the help of virtualization, multiple operating systems and applications can run on the same machine and its same hardware at the end same time, increasing the utilization and flexibility of hardware. In other words, one of the main cost effective, hardware reducing, and energy saving techniques by cloud providers is virtualization. Virtualization allows sharing a single physical instance among multiple clients at the same time.



Types of Virtualization:

1. Hardware virtualization: when the machine software or rmm is directly installed on the hardware system is known as hardware virtualization. The main job of hypervisor is to control and monitor the processor, memory and other hardware resources. After virtualization on hardware system, we can install different operating systems and run different applications on OS.
2. Operating system virtualization: when rmm is installed on the Host Operating system instead of directly on the hardware system is known as operating system virtualization. Used for testing applications on different platforms of OS.
3. Server virtualization: when rmm is installed on the server system it is known as server virtualization. It is done because a single physical server can be divided into multiple servers on the demand basis and balancing the load.
4. Storage virtualization: It is the process of grouping the physical storage from multiple network storage devices so that it looks like a single storage device. It is also implemented using software applications. It is mainly done for backup and recovery purposes.

Benefits of Virtualization

1. More flexible and efficient allocation of resources.
2. Enhance development productivity
3. It lowers the cost of IT infrastructure.

4. Remote access and rapid scalability.
5. High availability and disaster recovery.
6. Pay per use of IT infrastructure on demand.
7. Enables running multiple operating systems.

- Hyper Visor:

A hypervisor is a form of virtualization software used in cloud computing hosting to divide and allocate resources on various piece of hardware. The program which provides partitioning, isolation or abstraction is called virtualization hypervisor. The hypervisor is a hardware virtualization technique that allows multiple guest operating systems (OS) to run on a single host system at the same time. A hypervisor is sometimes also called a Virtual Machine Manager (VMM).

Hypervisors allow the use of more of a system's available resources and provide greater IT versatility because the guest VMs are independent of the host hardware which is one of the major benefits of the Hypervisor. They can be quickly switched between servers. It helps to reduce:

- 1. Space Efficiency
- 2. Energy uses
- 3. Maintenance Requirements of the servers.

Examples: Oracle VM Virtual Box, VMware Server Workstation.

~~Microsoft Virtual PC, KVM~~

~~QEMU~~

~~Parallels~~

Types of Hypervisors:

1. Type-1 Hypervisors (Bare Metal Hypervisors):

Guest OS	Guest OS	Guest OS
----------	----------	----------

Hypervisor

System Hardware.

These hypervisors are also called as bare-metal hypervisors, run directly on the host's hardware without the need of a separate operating system. They have direct access to underlying hardware resources, providing high performance and efficiency.
 Eg: VMWare, Microsoft Hyper-V, Xenserver and KVM.

2. Type-2 Hypervisors (Hosted Hypervisors):

Guest OS	Guest OS
Hypervisor	
Host OS	

Host OS

Hardware

These hypervisors, also known as hosted hypervisors run on the top of a conventional operating system, much like other software applications. They rely on host operating system for hardware access and resource management.
 Eg: VMWare Player, Oracle Virtual Box, Parallel Desktop for MAC.

Comparison Between Virtual Box and KVM.

Feature	Virtual Box	KVM (Kernel Based VM)
1. Type	Virtual Box is a Type-1 Hypervisor (Hosted Hypervisor)	KVM is a Type-1 Hypervisor (Bare-Metal Hypervisor)
2. Host Operating Systems.	Windows, macOS, Linux.	Linux (Requires specific kernel modules).
3. Guest Operating Systems.	Windows, macOS, Linux, BSD, Solaris	Linux, Windows, macOS, BSD, Solaris.
4. Performance	Lower performance compared to KVM.	Higher performance due to direct hardware access.
5. Management Interface	Graphical User Interface (GUI) and Command Line.	Command Line Interface (CLI).
6. Live Migration	Limited support for live migration.	Supports Migration
7. Scalability	Limited scalability for large deployments.	Highly scalable for enterprise environment.
8. Community Support	Large community and extensive documentation	Strong community support and integration with Linux ecosystem.

9. Integration with cloud platforms.	Supports integration with cloud platforms.	Often used as the underlying hypervisor for cloud platforms.
10. Ease of Installation	Simple installation with a graphical installer.	Requires manual configuration and kernel module installation.
11. Resource Overhead	May have higher resource overhead due to additional layers of virtualization.	Lower resource overhead, as it operates closer to hardware level.
12. USB Device Support	Offers seamless USB device support for connecting peripherals to VMs.	Requires configuration for USB device passthrough to VMs.
13. 3D Graphics support	Provides 3D graphics support for accelerated graphics performance in VMs.	Limited 3D Graphics support, primarily optimized for server workloads.
14. Licensing	Free and open source platform.	Free and Open Source Platform

A⁺

29
15/3/2024

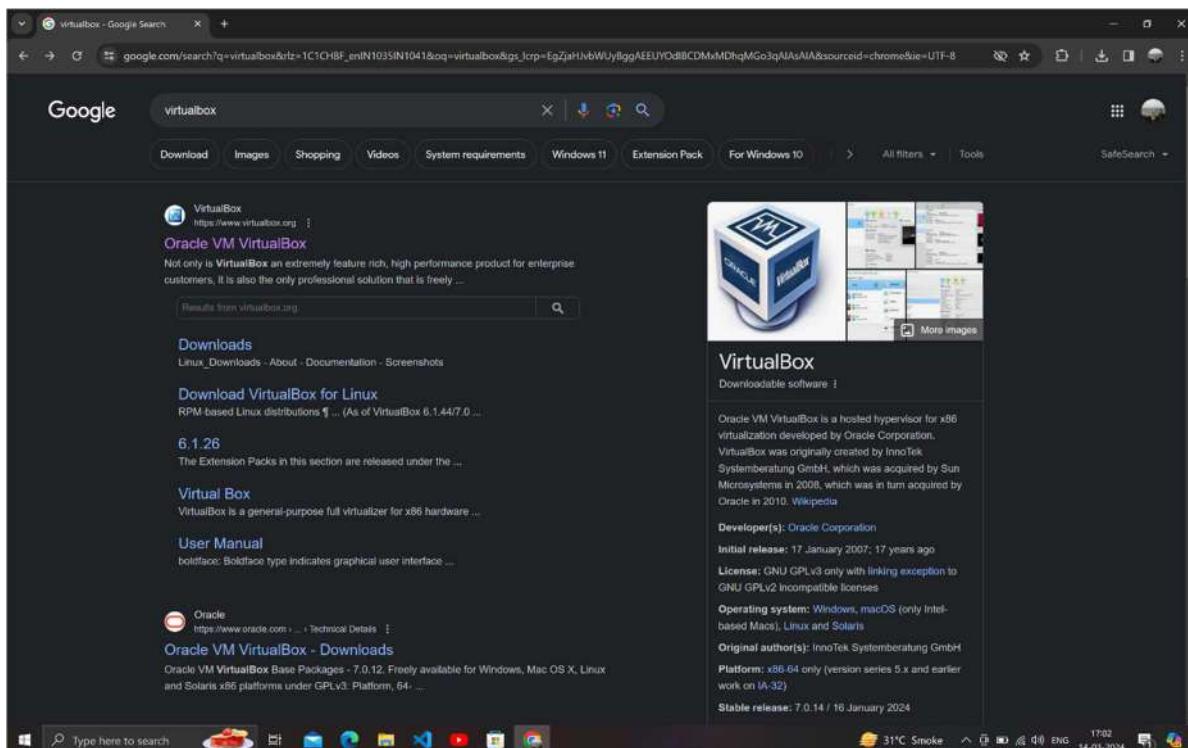
EXPERIMENT NO: 2

Parth Dabholkar C12 2103032

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

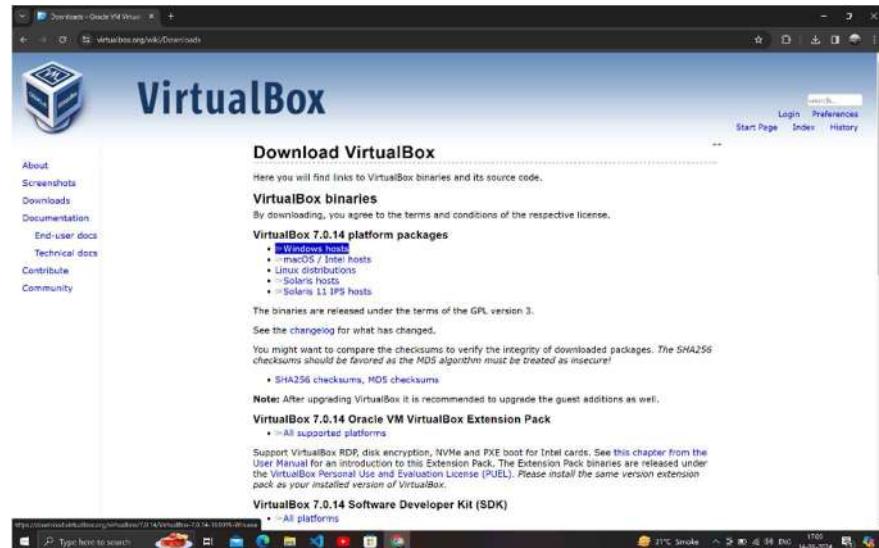
Hot picks:

- 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 6.1.48 released!

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



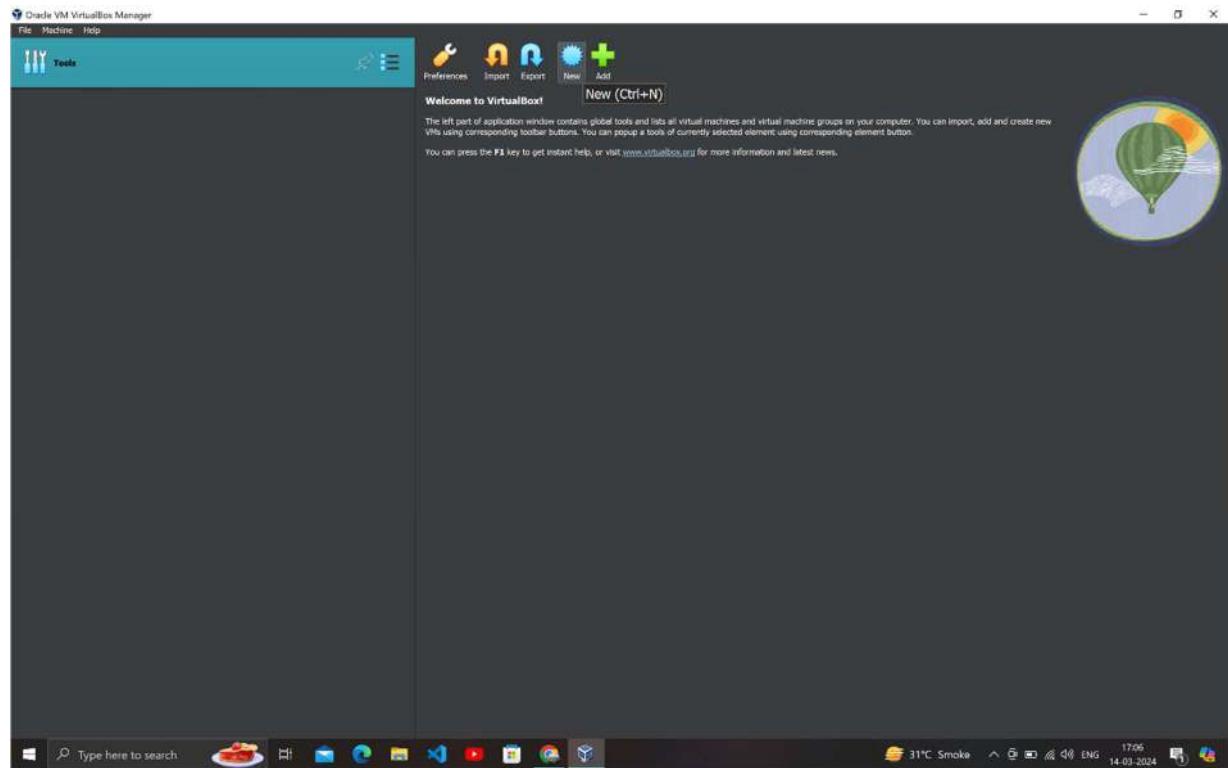
4. Open the installation setup from downloads. Install VirtualBox with default settings and click finish to complete the installation.



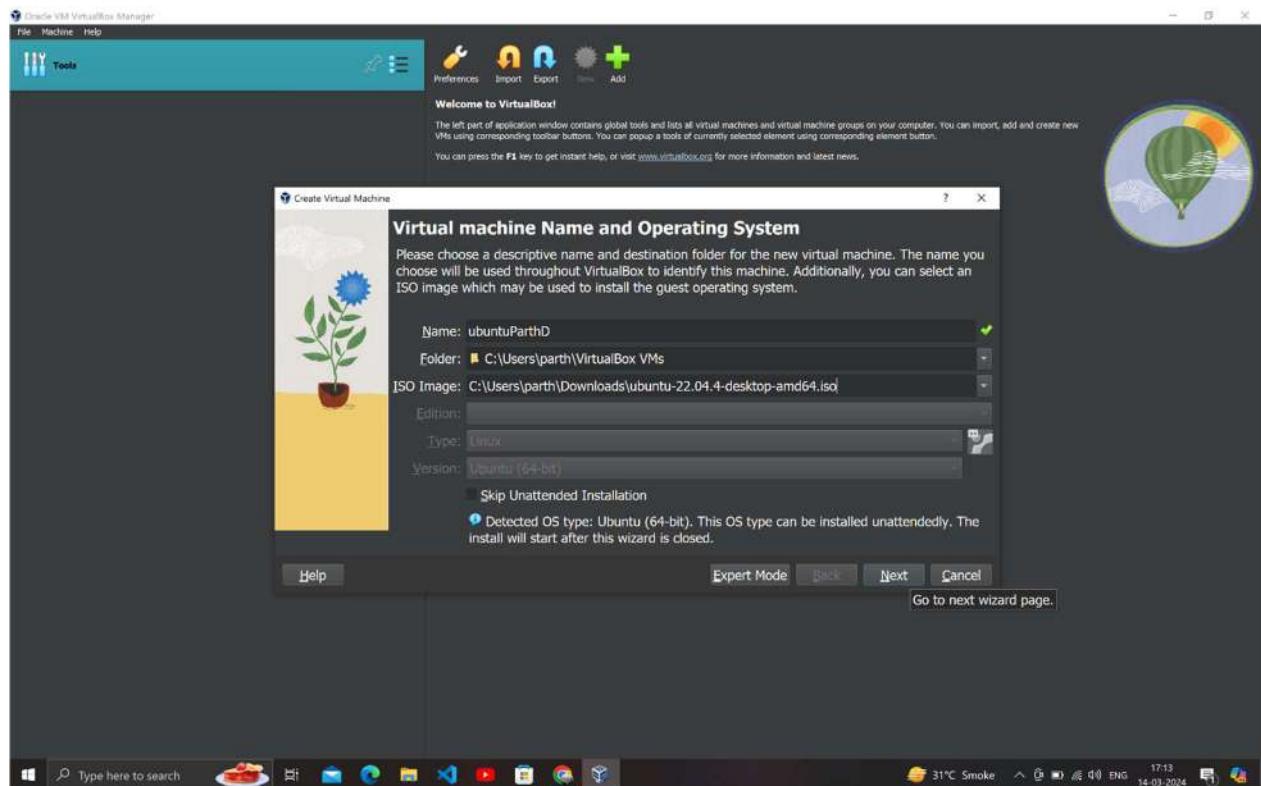
5. Before launching VirtualBox, we must download the ISO file for the OS we want to use on the Virtual Machine i.e. on the Host OS(Windows). We will download Ubuntu as it is lightweight, small and easily downloadable OS. Go to the official site of Ubuntu and download the latest Linux Ubuntu LTS OS.



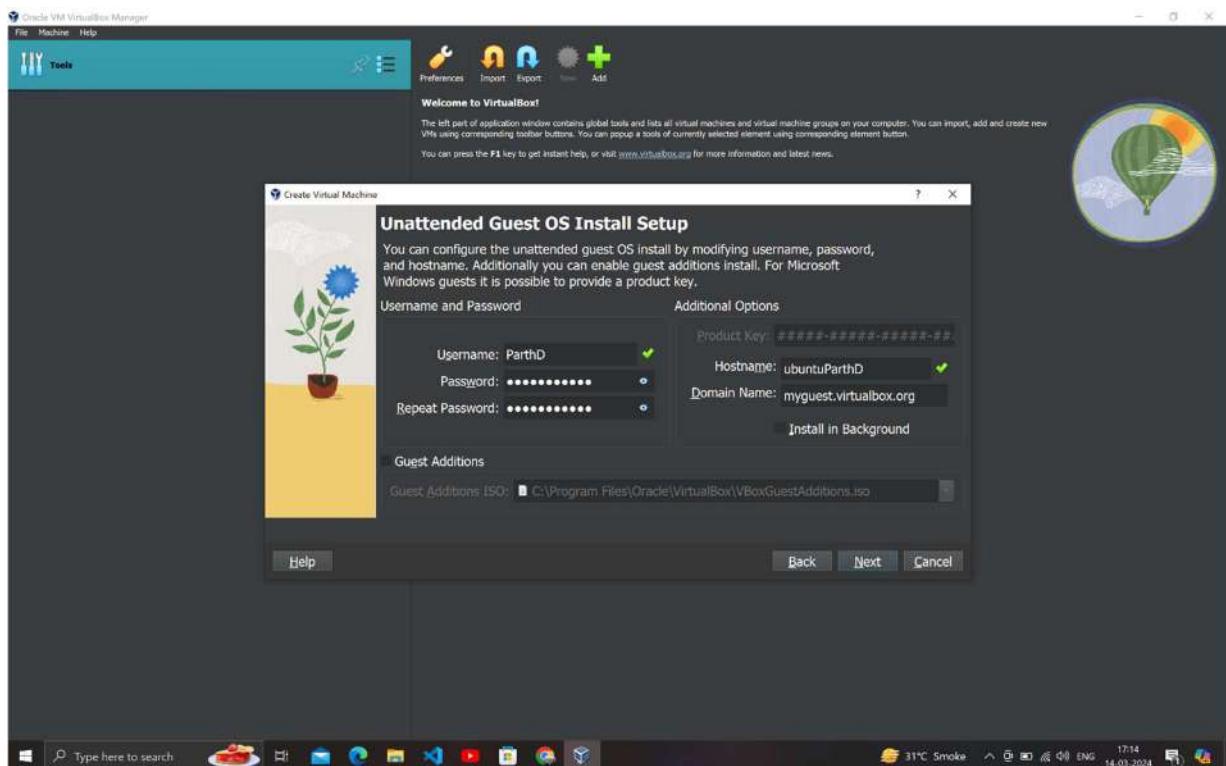
6. Open the VirtualBox. An interface will appear as given below. Click on NEW to start a new Virtual OS.



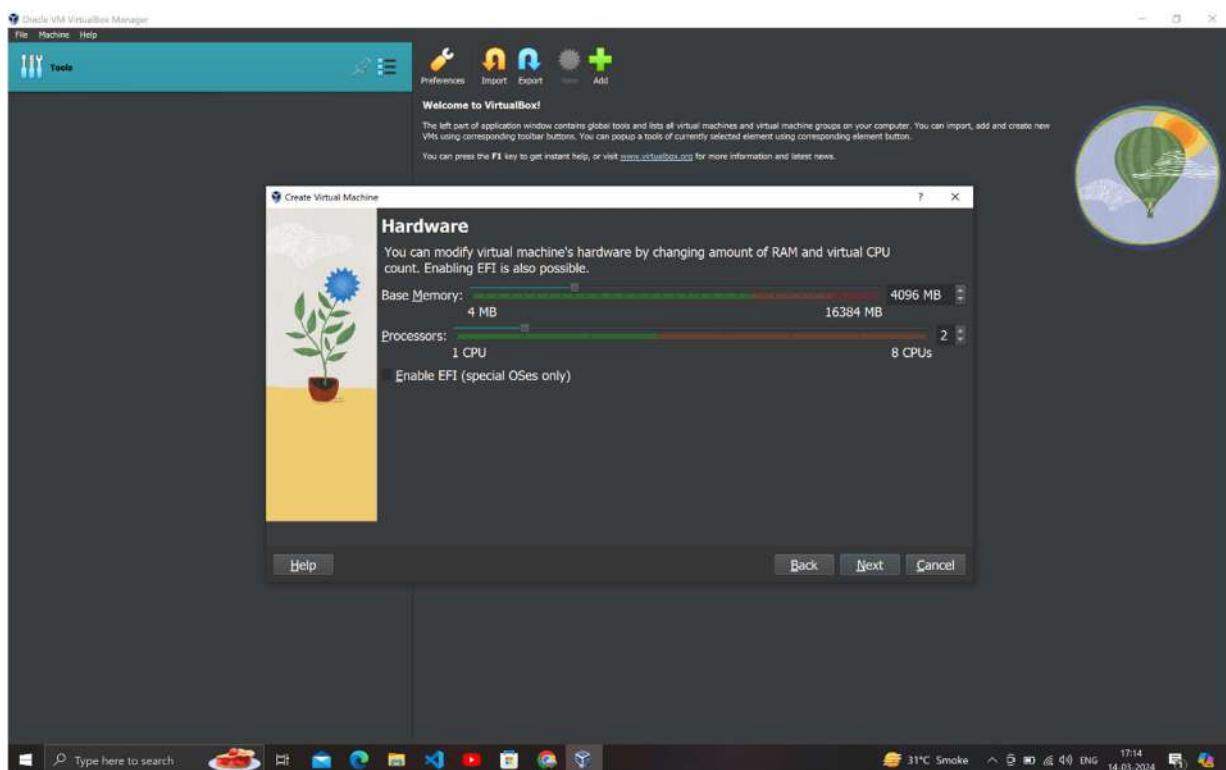
7. Enter the Virtual Machine name, the path to the folder of VMbox and under ISO Image section select the Ubuntu ISO which was downloaded in the earlier steps. Further select next.



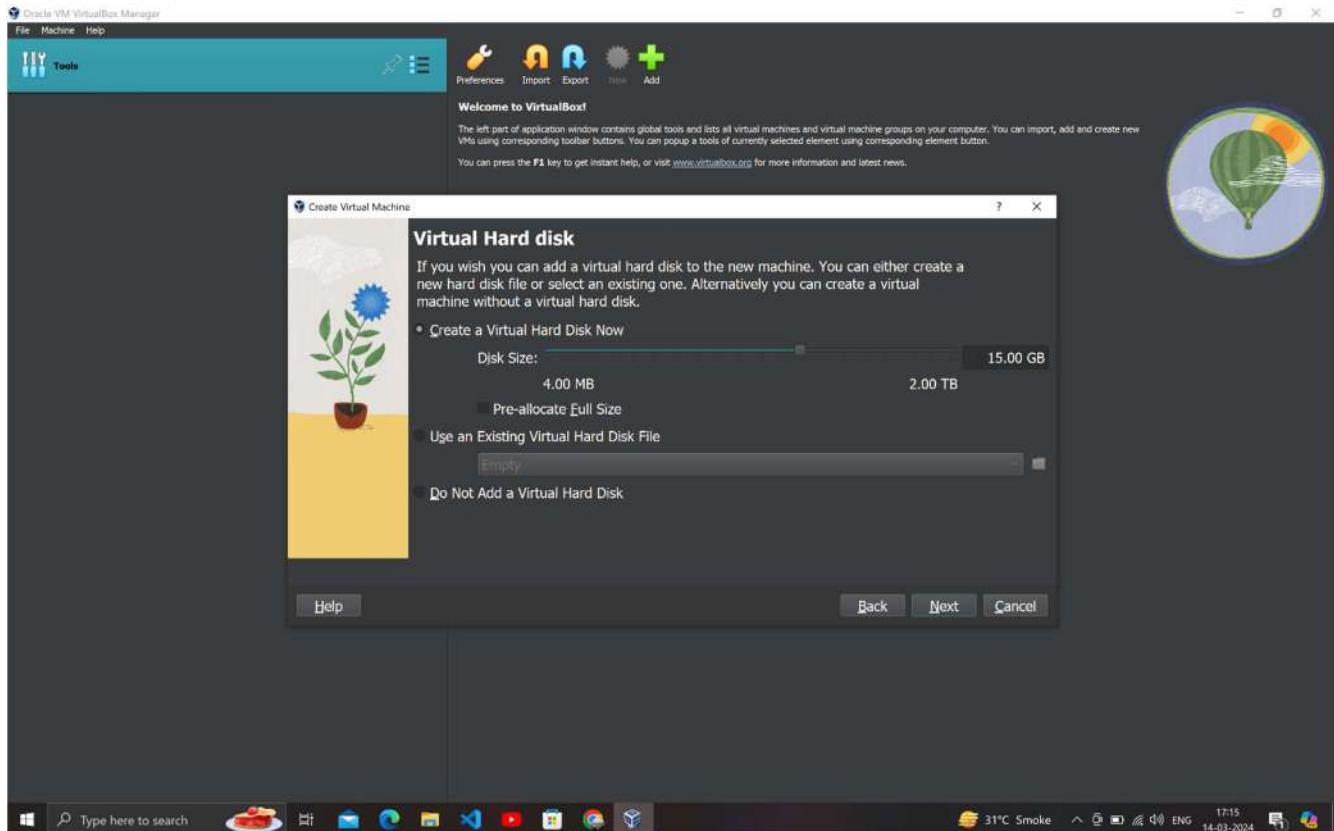
8. Fill the username and password for the Virtual OS. Remember the password as it will be used for login in further steps.



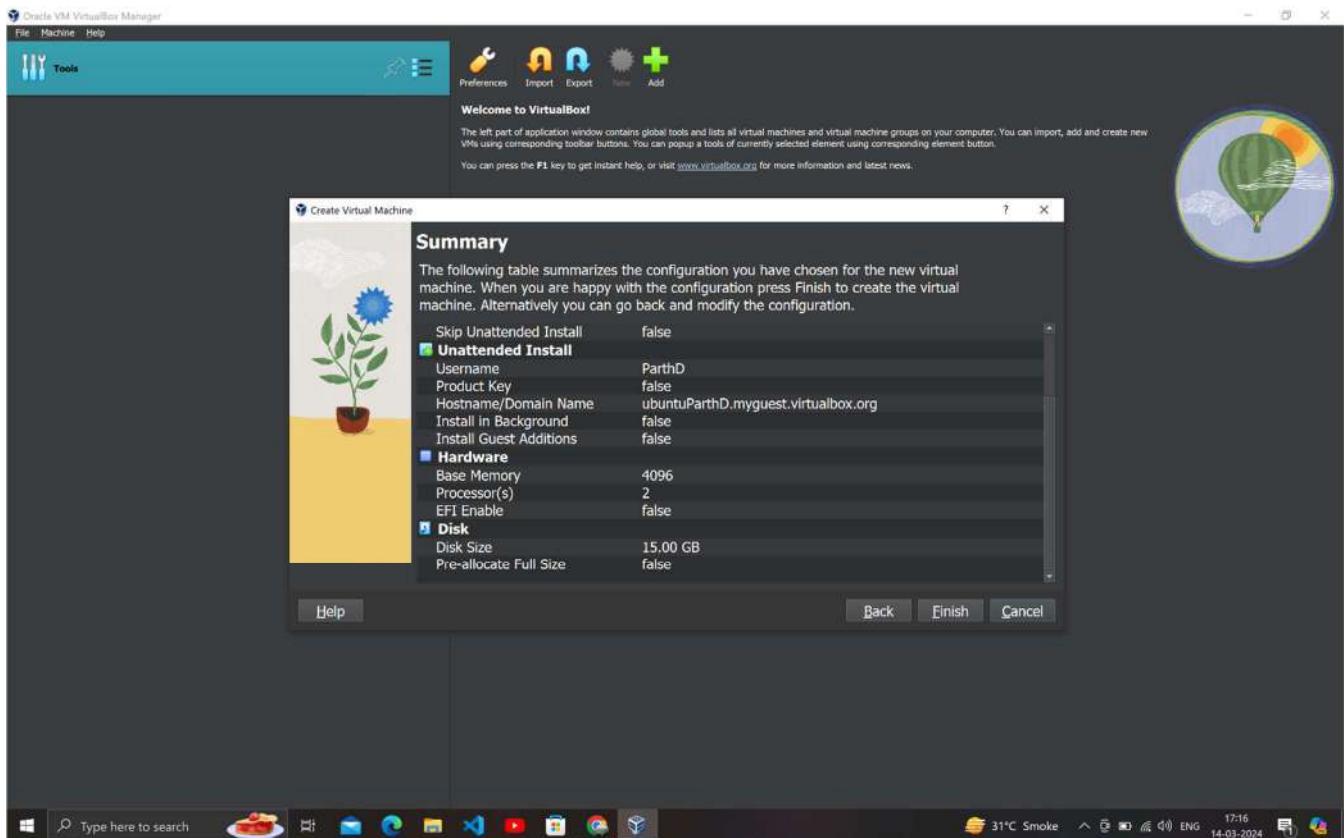
9. Set the base memory/RAM to be utilized for hosting the OS. Here, I have set it to 4GB (4096MB). Also set the number of processors to be used for the same. I have assigned to processors for the same.



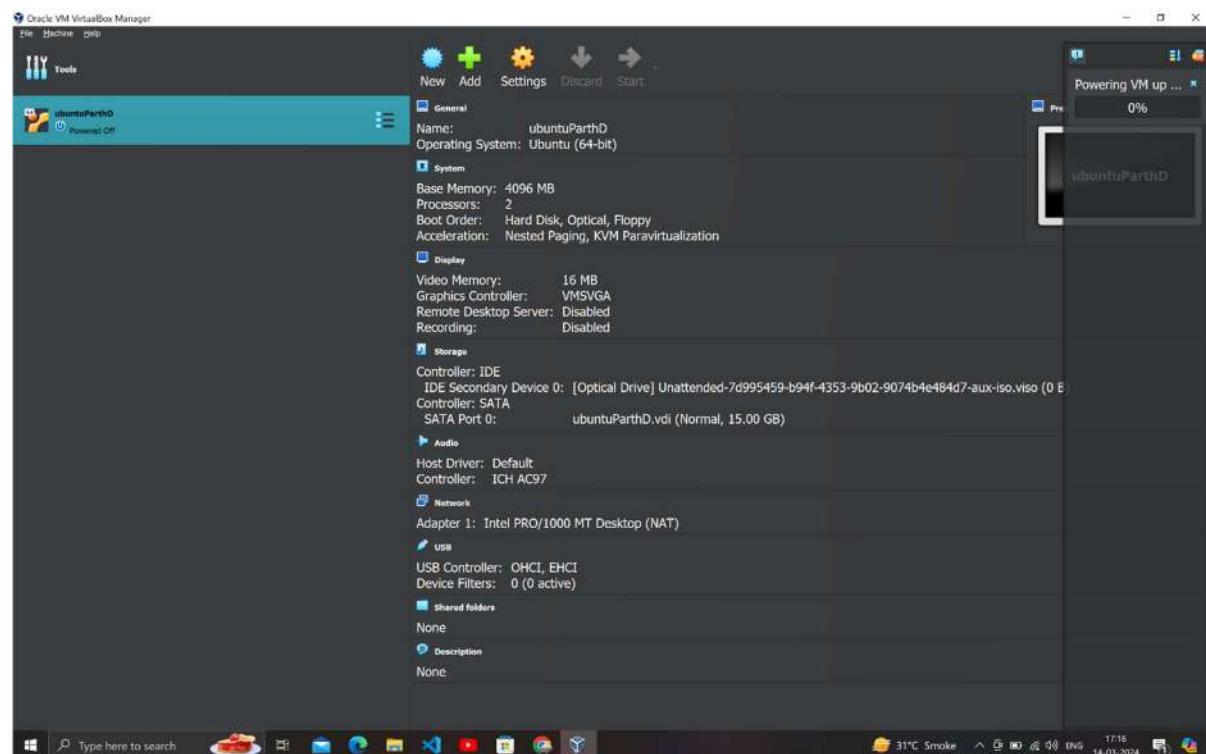
10. Assign the storage or Virtual Hard Disk for your Virtual Machine. I have set it to 15 GB.



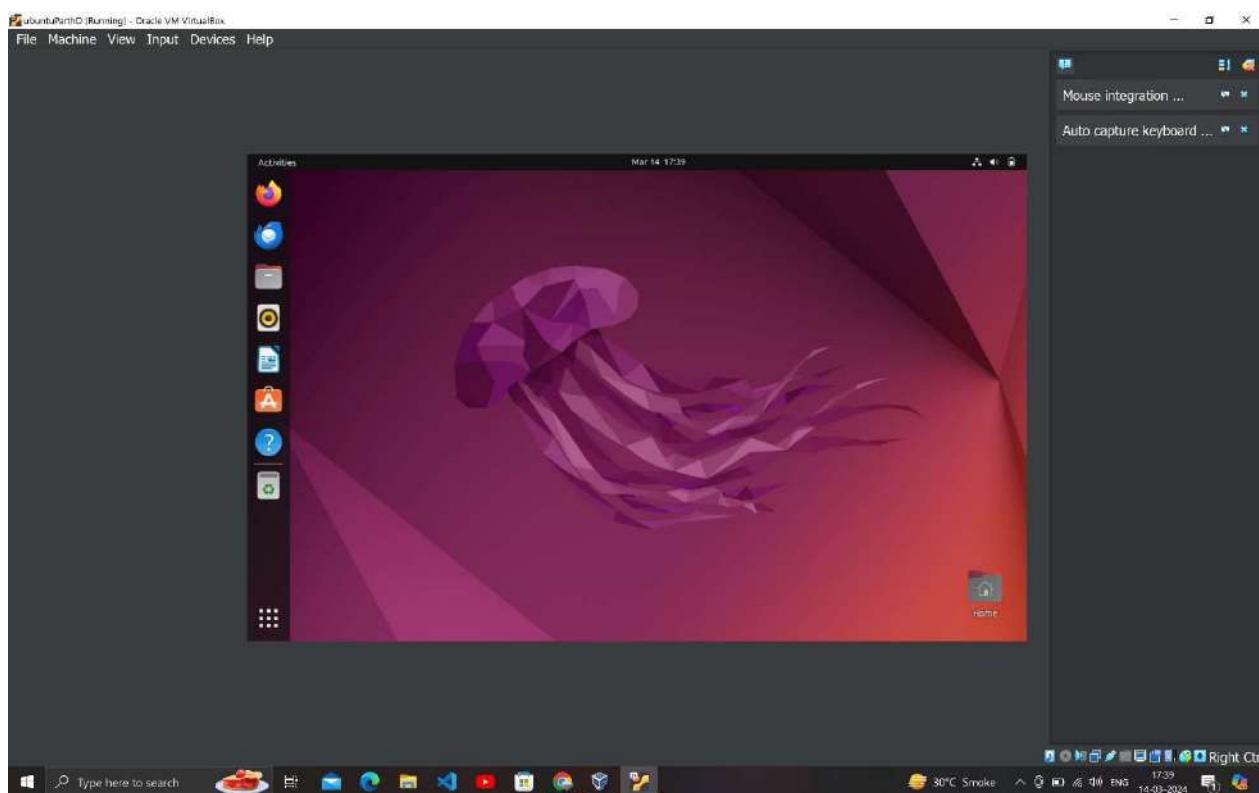
11. Analyse the summary provided. It contains all the information about the Virtual Machine such as RAM utilized, Storage assigned, Processors undertaken and used etc. Make changes if needed. Click finish to launch the Virtual Machine.



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 .xmp
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
-rw-rw-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$ 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_flushwg]
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-hv-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 27 0.0 0.0 0 0 ? I< 17:38 0:00 [inet_frag_wq]
root 28 0.0 0.0 0 0 ? I 17:38 0:00 [kworker/u4:1-writeback]
```

Experiment no: 3

AIM: To study and implement bare-metal visualization using zen.

- functions performed by Bare-Metal Hypervisors

Bare metal hypervisors, also known as Type1 hypervisors, are software systems that run directly on the physical hardware of a computer without needing a separate underlying operating system. They enable the virtualization of hardware resources to support the creation management of multiple virtual machines (VMs) on a single physical machine.

Here are the main functions performed by bare metal hypervisors:

1. Hardware Abstraction: Bare metal hypervisors abstract the underlying physical hardware, allowing multiple virtual machines to run independently on the same physical server. This abstraction enables VMs to access virtualized hardware resources such as CPU, memory, storage and network interfaces.
2. Resource Allocation: They allocate physical hardware resources among virtual machines according to pre-defined policies and configurations. This includes managing CPU scheduling, memory allocation, disk I/O and network bandwidth to ensure optimal performance and resource utilization across VMs.

3. Isolation: Bare metal hypervisors provide strong isolation between virtual machines, ensuring that each VM operates independently of others and cannot directly interfere with or access resources belonging to other VMs. This isolation enhances security and stability by preventing potential conflicts and unauthorized access.

4. Virtual Machine Management: They facilitate the creation, provisioning, monitoring and management of virtual machines. Administrators can create and configure VMs, allocate resources.

5. Performance Optimization: They optimize performance by implementing techniques such as paravirtualization, where certain guest operating system functions are modified to run more efficiently in a virtualized environment. Bare metal hypervisors also employ various optimization strategies such as memory ballooning, CPU pinning, and disk caching to improve overall system performance and responsiveness.

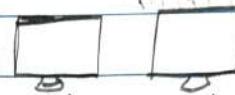
• Compare Hosted and Bare-Metal Hypervisors

Criteria	Bare Metal Hypervisor	Hosted Hypervisor
Definition	Runs directly on the system where virtual machines function.	Operates on conventional operating system.
Virtualization	Hardware virtualization	OS virtualization.

Operation	Guest OS and applications run on the hypervisor.	functions as application on host OS.
Scalability	More scalability	Limited scalability
Setup/Installation	simple, if your hardware supports the application	Easier than Type 1 setup because of existing OS.
System Independence	Having direct access to a virtual machine which hosts it.	Not giving direct access to hardware and resources.
Speed	faster	slower
Performance	higher because of no middle layer.	comparatively low due to extra overheads.
Security	More secured	Less secured
Use Cases	server virtualization, cloud computing, high performance computing	Development, Testing, Desktop virtualization
Examples	VMware, ESXi, Microsoft Hyper V, Citrix, XenServer.	VMware workstation, Player, Microsoft Virtual PC, Sun's Virtual Box.

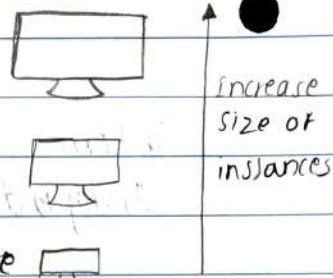
Horizontal scaling

- Horizontal Scaling: Also referred to as scaling out, horizontal scaling improves the cloud's throughput by adding new computer infrastructure. At a basic level, scaling out can mean adding new computing nodes or machines to enhance the data processing and storage capabilities. Compared to vertical scaling, horizontal scaling is quicker and easier to accomplish.

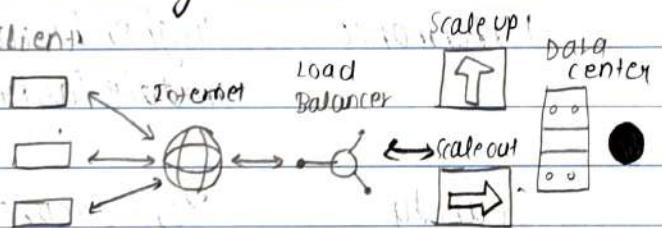


Vertical scaling

- Vertical scaling: vertical scaling enhances the technical specifications of existing infrastructure by adding or replacing CPU, HDD or other components. Decommissioning existing systems and replacing them with higher capability infrastructure would also qualify as vertical scaling or 'scaling up'. Compared to scaling out it may take longer and period of downtime. But it is cheaper compared to Horizontal scaling.



- Auto Scaling: It the capacity provided by cloud computing platforms to automatically adjust the number of compute resources allocated to the application based on its workload. It involves setting up policies or rules that define when additional resources should be provisioned and released.



AUTO Scaling

- Load Balancing: Load Balancing is the process of distributing incoming network traffic across multiple servers or resources to ensure optimal utilization and performance. The primary goal of load balancing is to prevent any single server from becoming overloaded, improving the overall reliability and availability of a system.

AT
15/3/2024

EXPERIMENT NO: 3

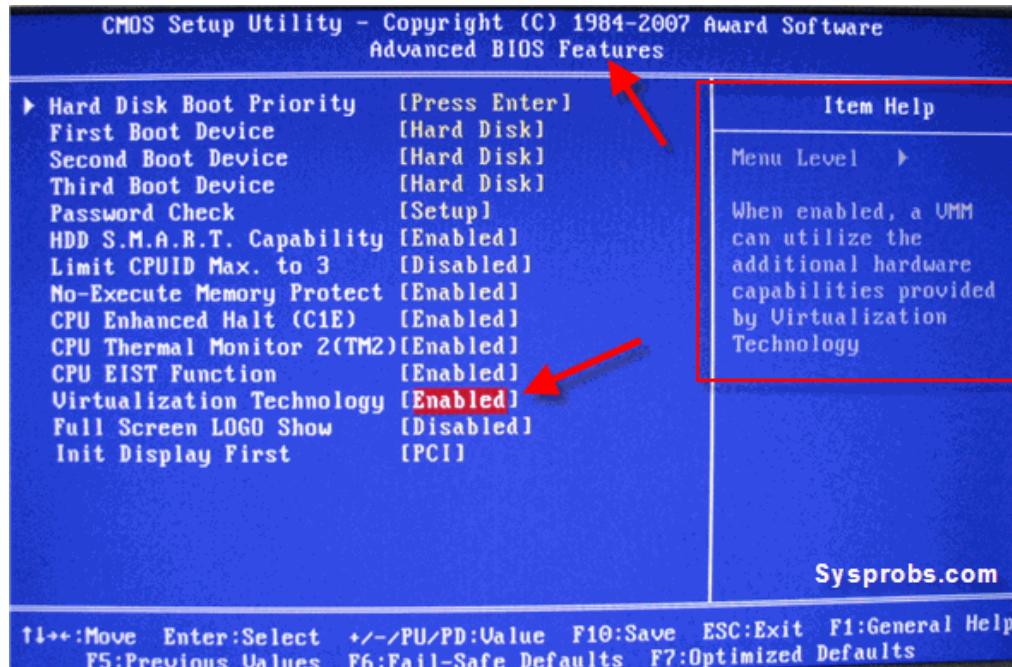
AIM: To study and implement Bare-metal Virtualization using XEN.

Step1: Install XenServer

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



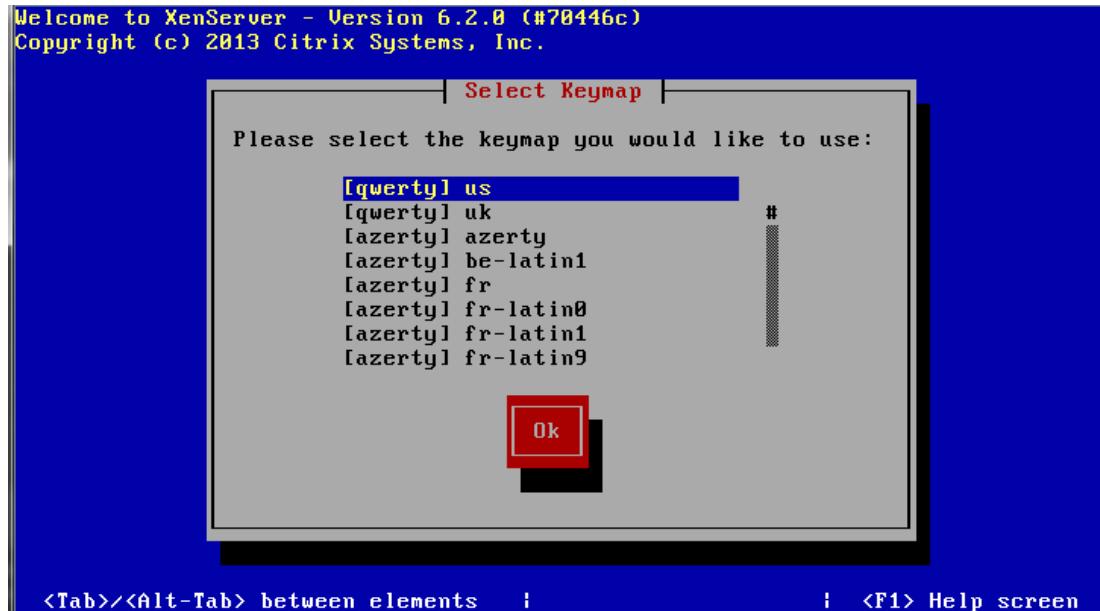
ii) Press F2 for advance options and Make Enabled Virtualization Technology from BIOS



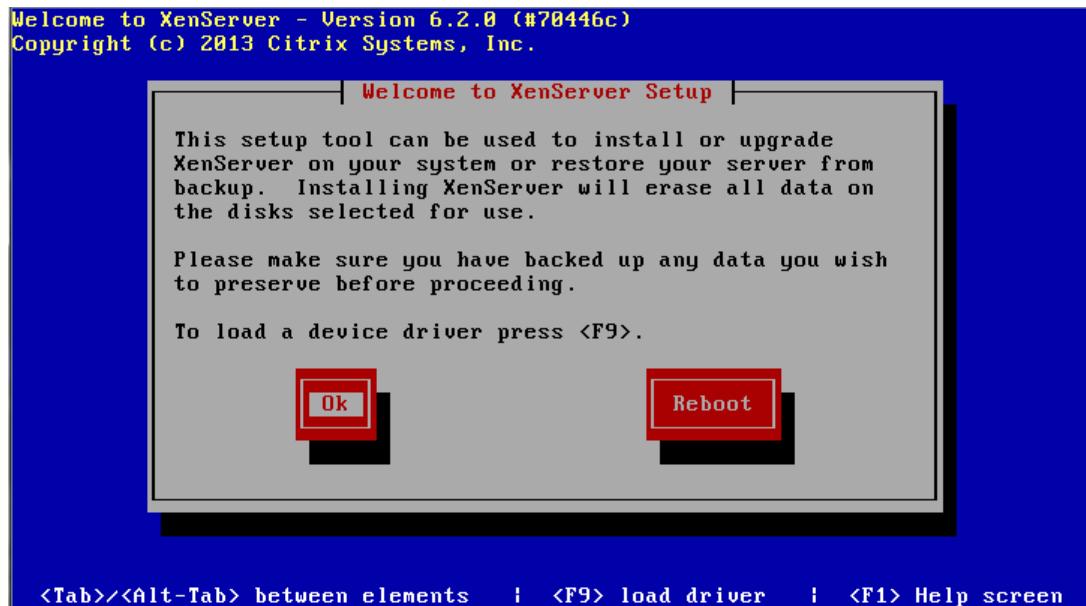
Save and Reboot



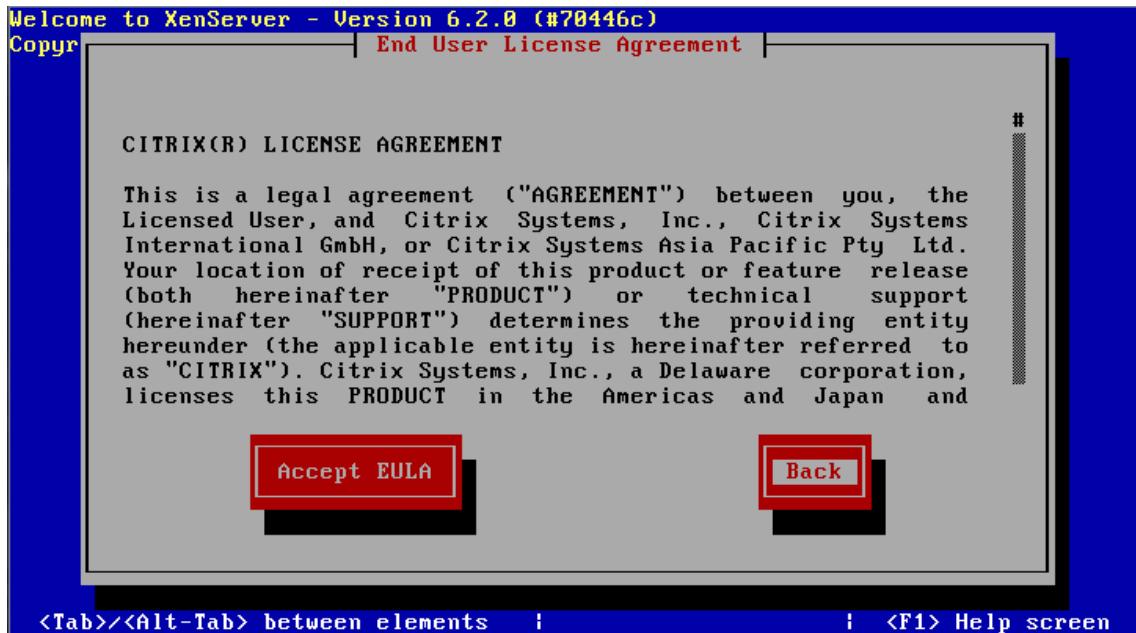
iii we begin by choosing the keymap i.e Keyboard Layout



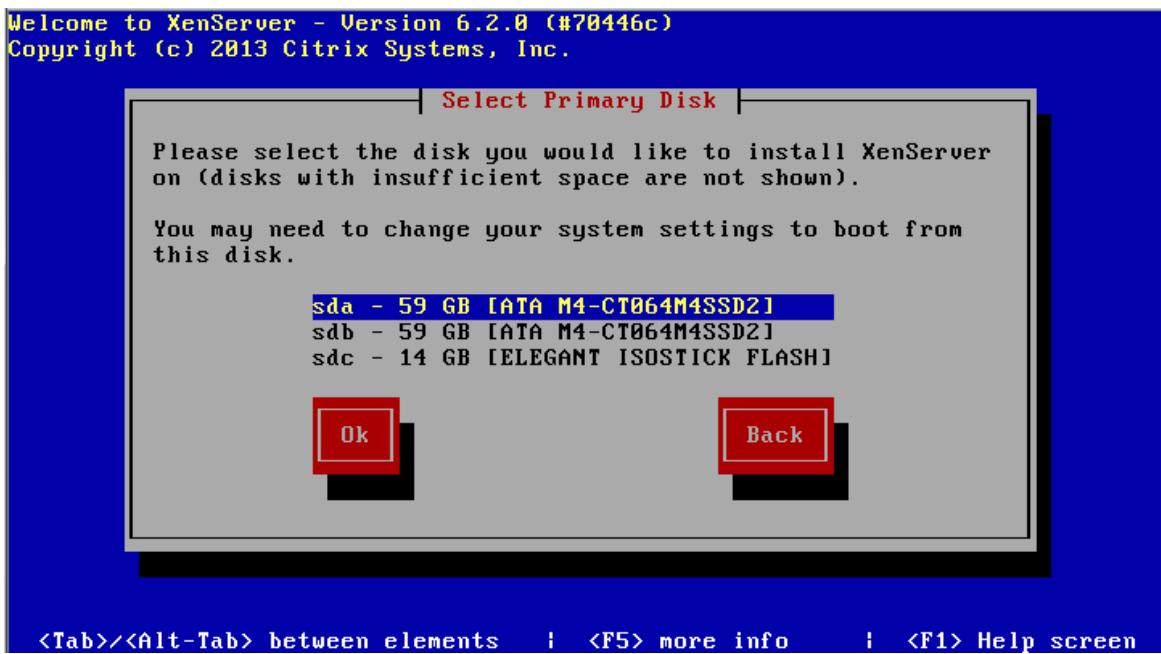
iv) Press Enter to load Device Drivers



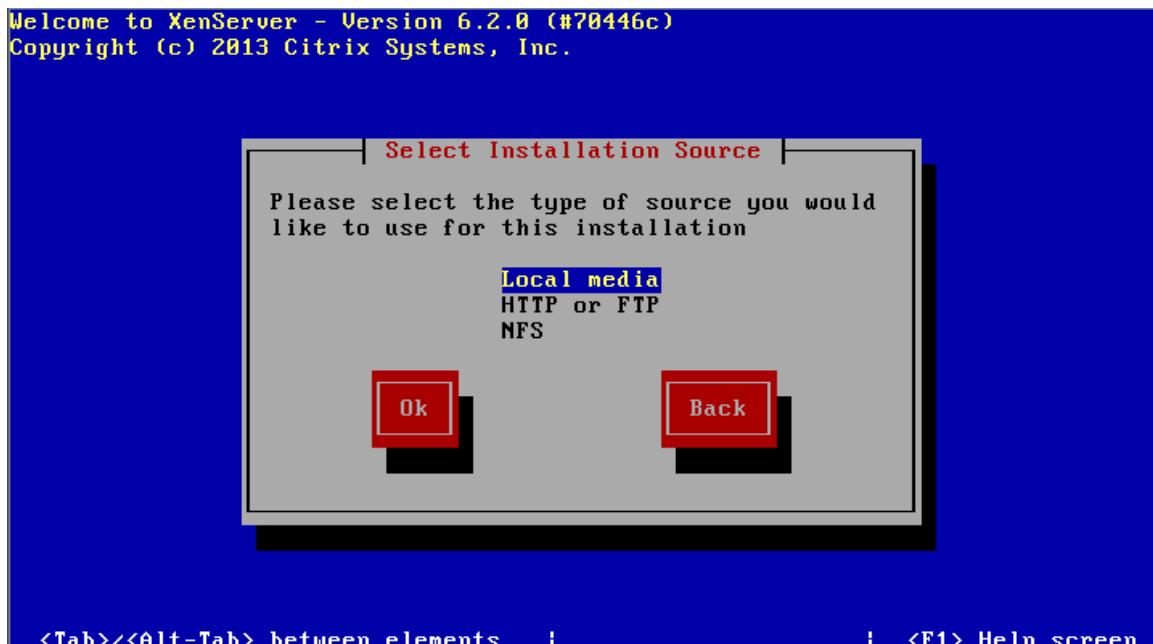
V) Press Enter to accept End User License Agreement



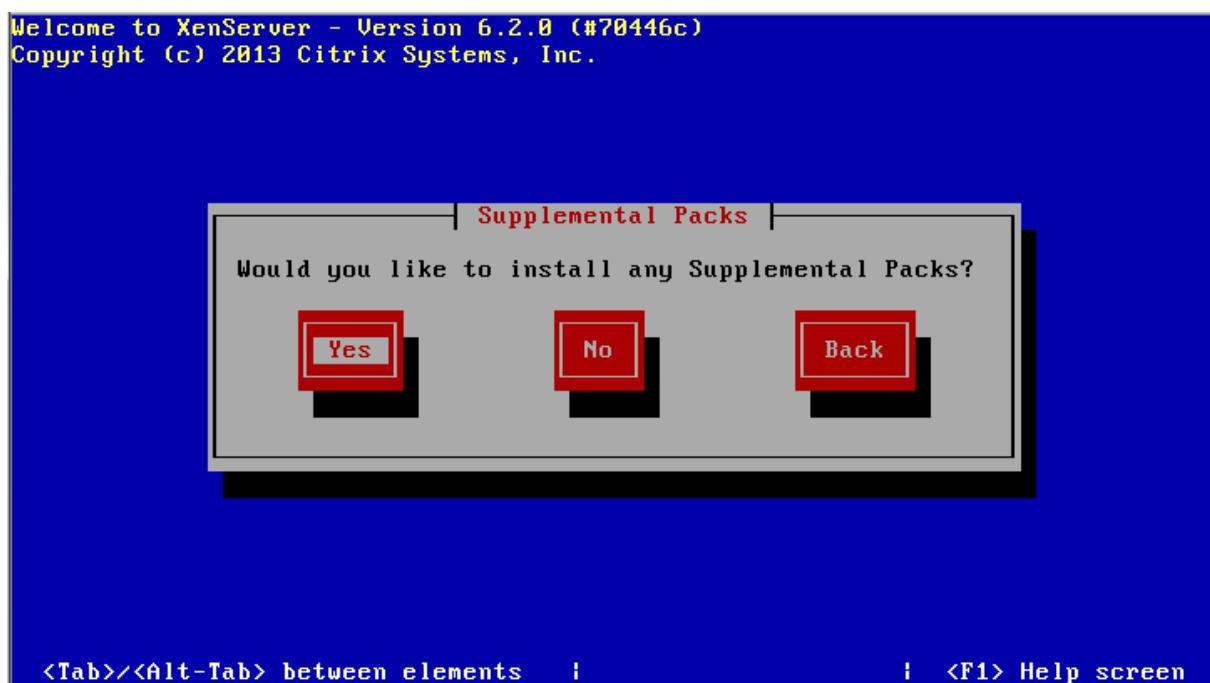
Vi) Select Appropriate disk on which you want to install Xenserver



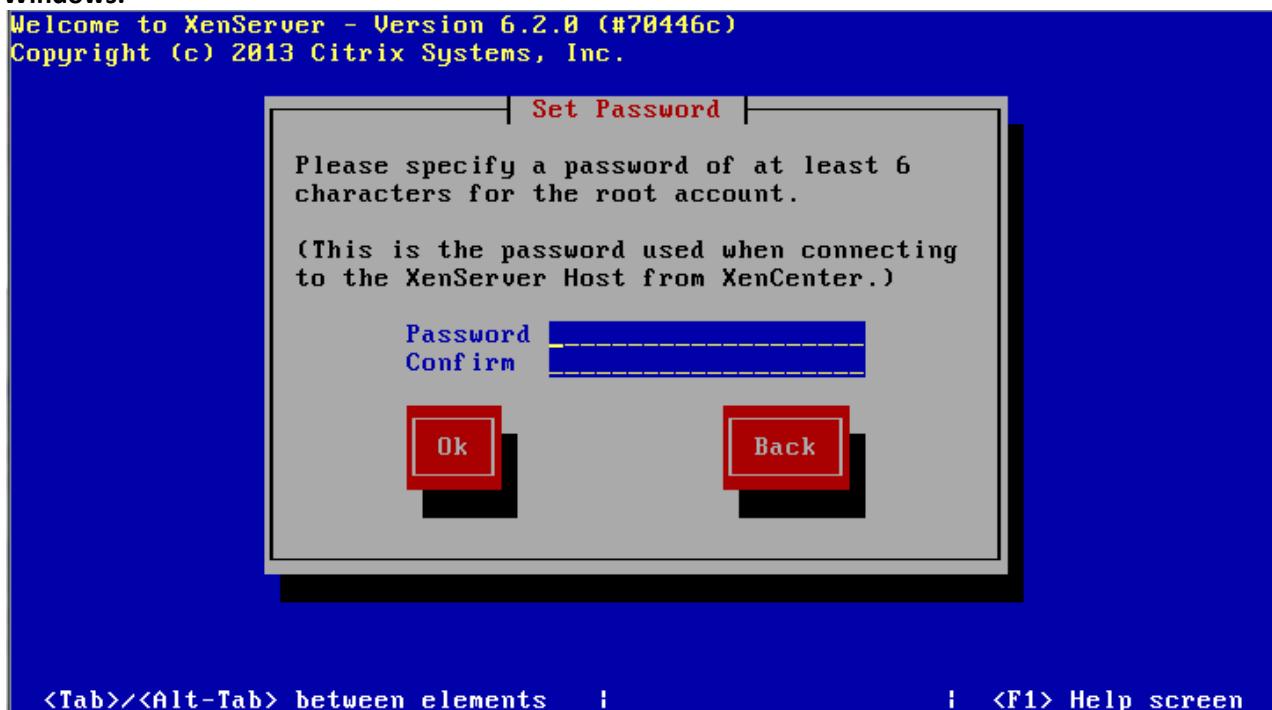
vii) Select appropriate Installation Media



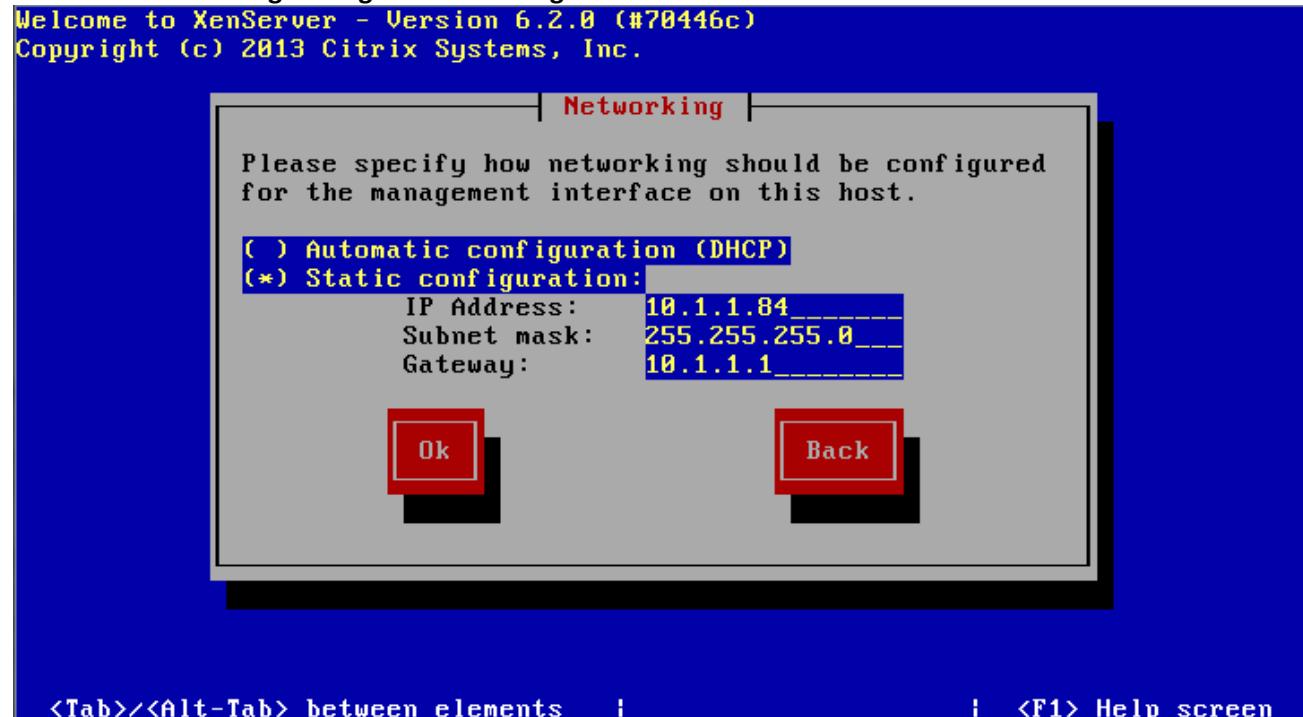
viii) Press yes to Select additional packages for installation , Otherwise Press No



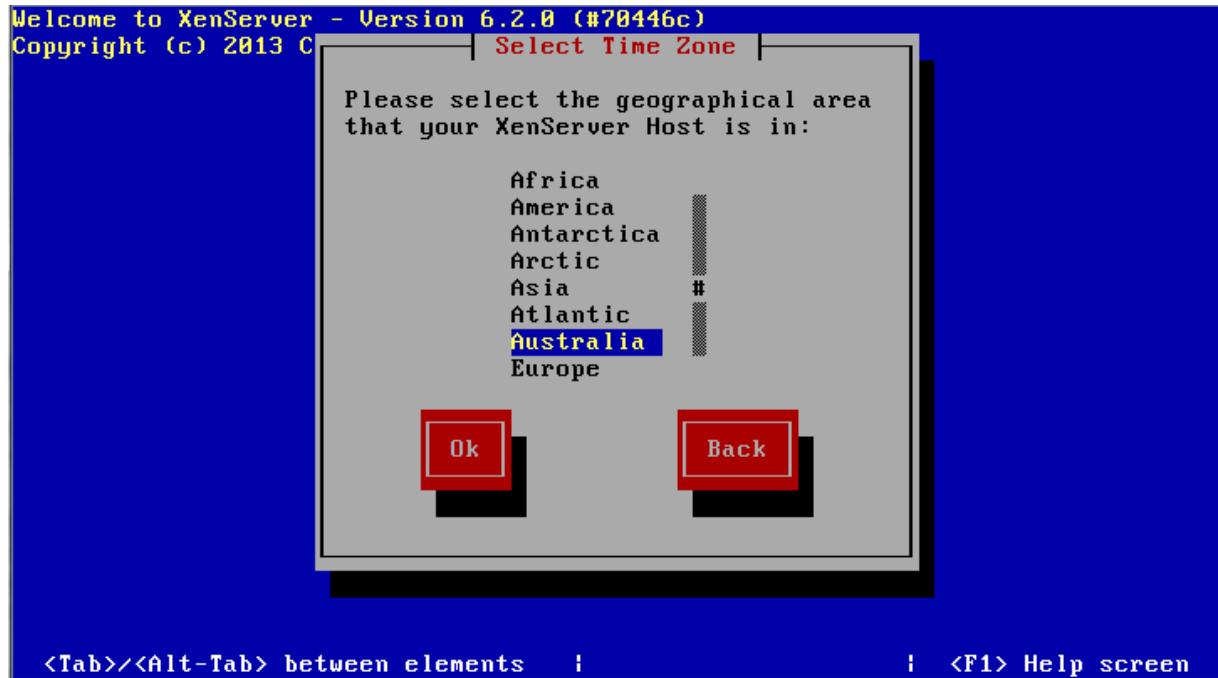
viii) Pick a root password – you will need this for logging in via the console or via the XenServer client on Windows.



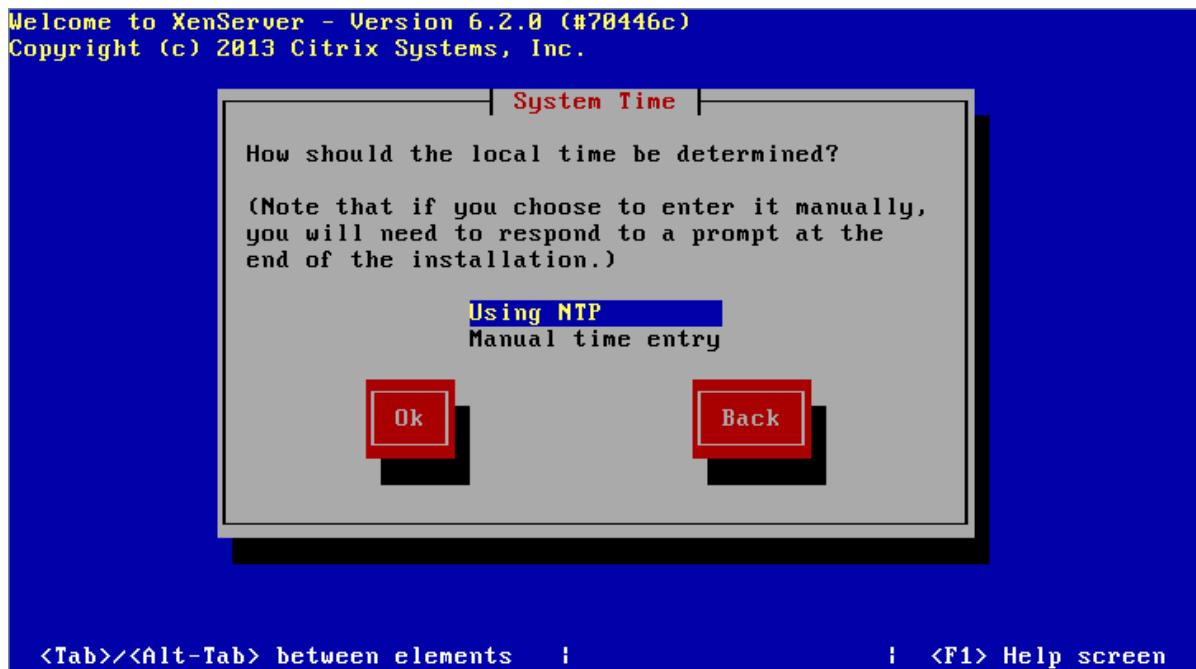
ix) The motherboard we are installing have two Ethernet ports, both of which are supported by XenServer. Choose the one you wish to use for the management network – you can change this later. Here we get to choose the networking settings for our management network.



x) Select Time zone



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



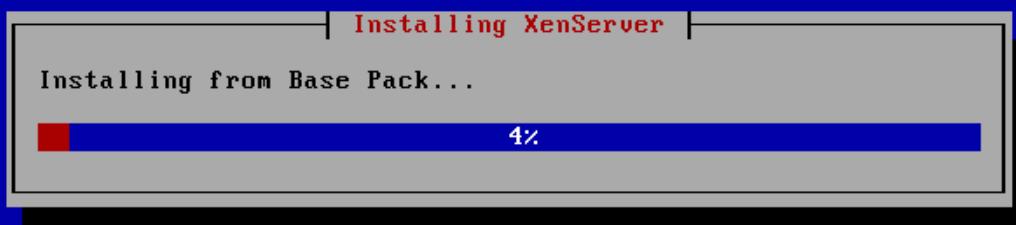
Xii) Press Enter to start installation of XenServer

Welcome to XenServer - Version 6.2.0 (#70446c)
Copyright (c) 2013 Citrix Systems, Inc.



<Tab>/<Alt-Tab> between elements | | <F1> Help screen

Welcome to XenServer - Version 6.2.0 (#70446c)
Copyright (c) 2013 Citrix Systems, Inc.

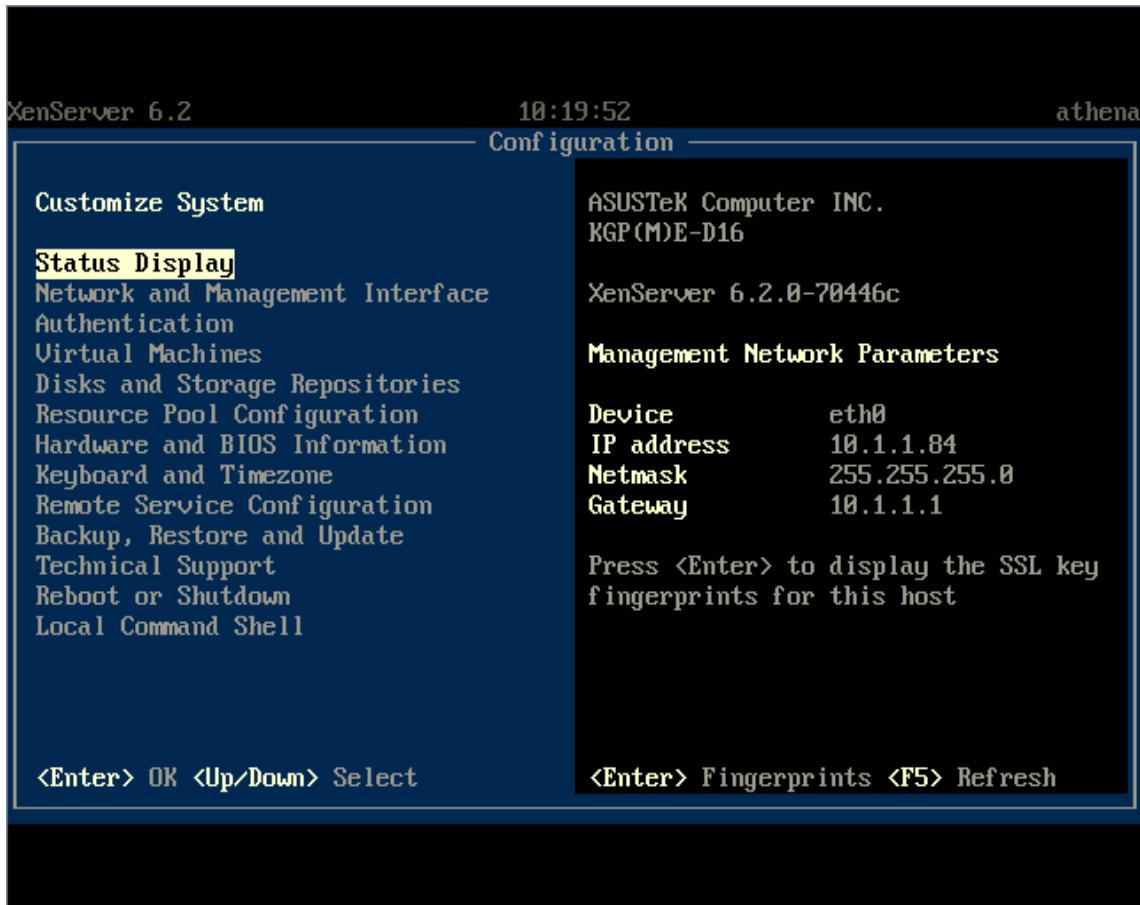


Working: Please wait...

This is the loading screen for installed new XenServer .

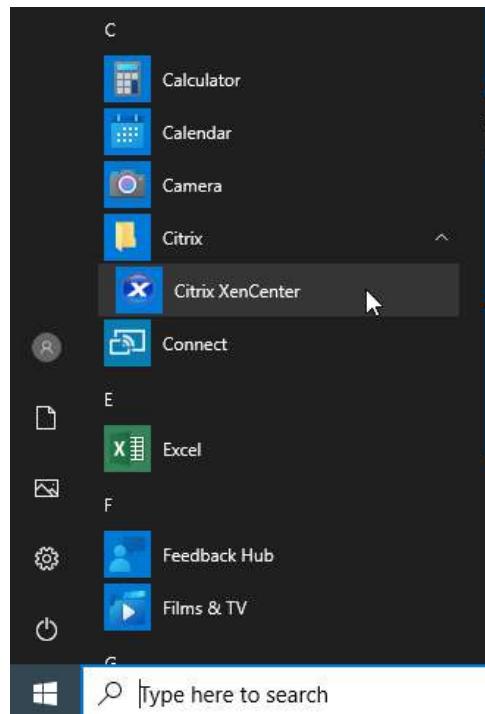


The console screen for XenServer

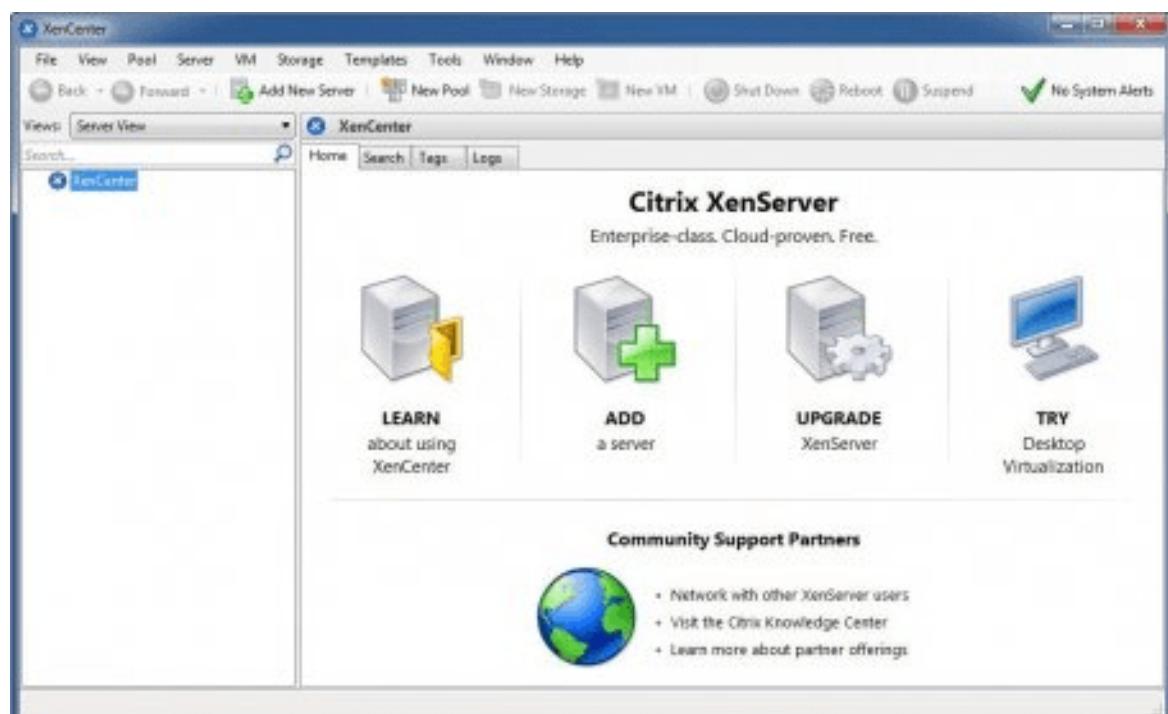


Step 2 : Connect XenCenter to XenServer

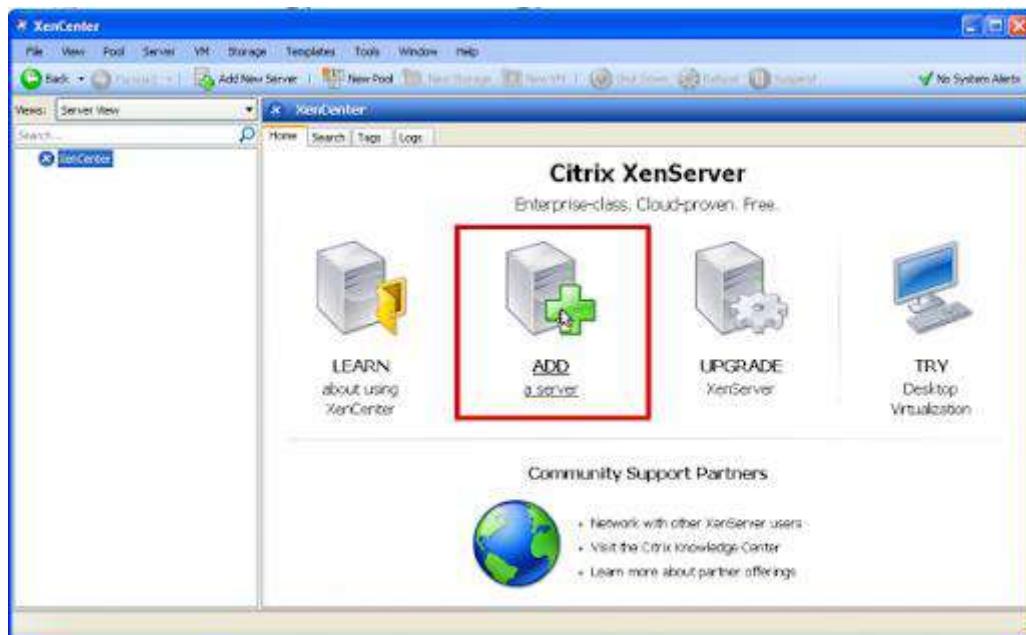
i) Download the XenCenter a management utility from XenServer IP address as a URL on browser. Install XenCenter and open it from start Menu of Windows on Machine 2.



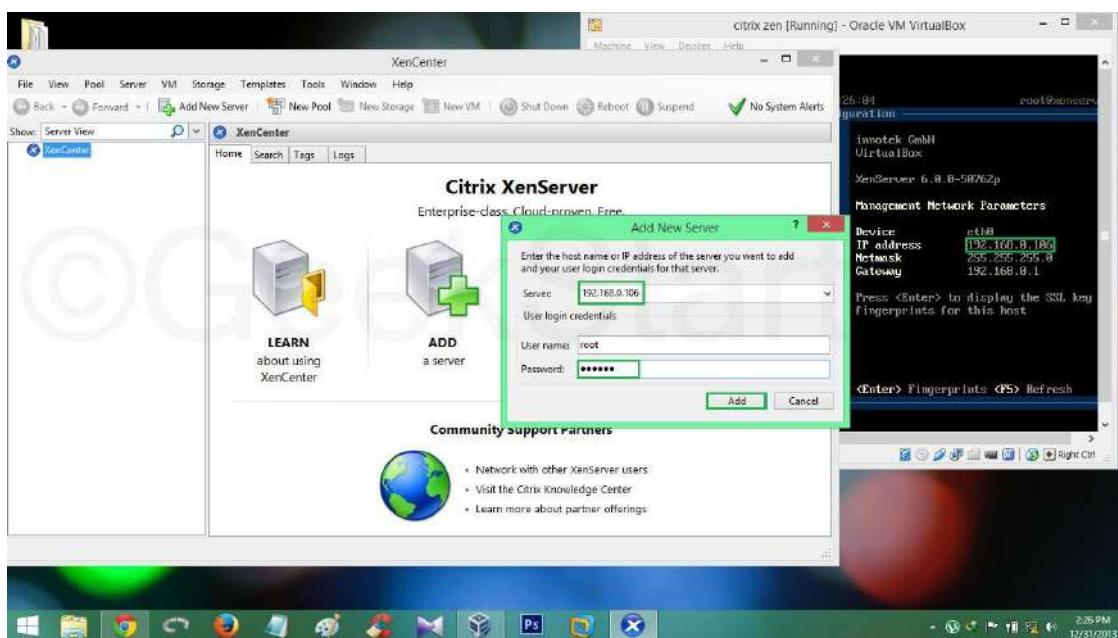
ii) Once you click on Citrix XenCenter ..It Looks like as below:



- 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

A screenshot of the XenCenter management interface. The title bar says "XenCenter". The menu bar includes File, View, Pool, Server, VM, Storage, Templates, Tools, Window, Help. The toolbar has icons for Back, Forward, Add New Server, New Pool, New Storage, New VM, Shut Down, Reboot. The "Views" dropdown is set to "Server View". The main pane shows a tree view under "XenCenter" with a single node "bj-xenserver" expanded, showing "DVD drives", "Local storage", and "Removable storage". To the right, the "bj-xenserver Overview" window is open, showing the "bj-xenserver" entry in the table. The table has columns: Name, CPU Usage, and Used Memory. The entry shows "Name: bj-xenserver", "CPU Usage: 17% of 2 CPUs", and "Used Memory: 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

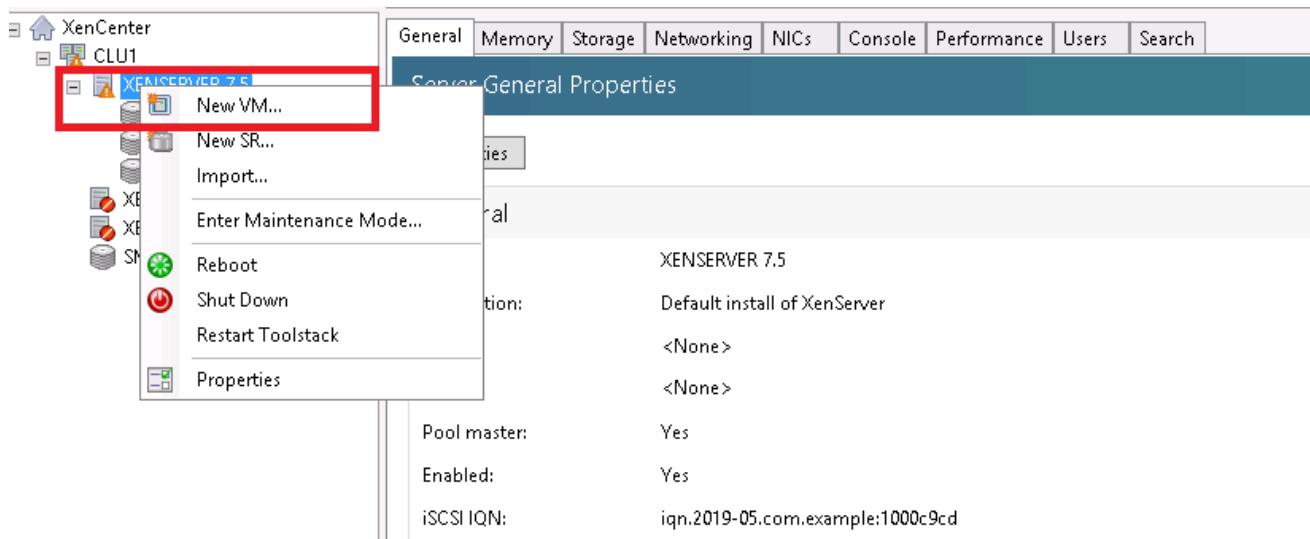
v) # ls (content of pendrive)

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

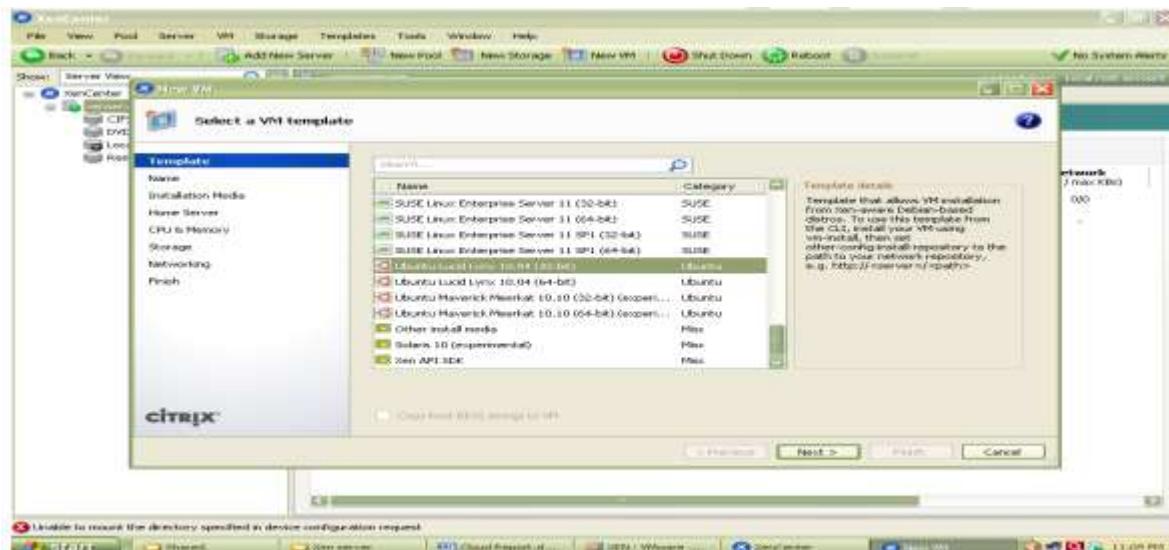
vii) Reboot or shutdown Xenserver from XenCenter or from console of Xenserver

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



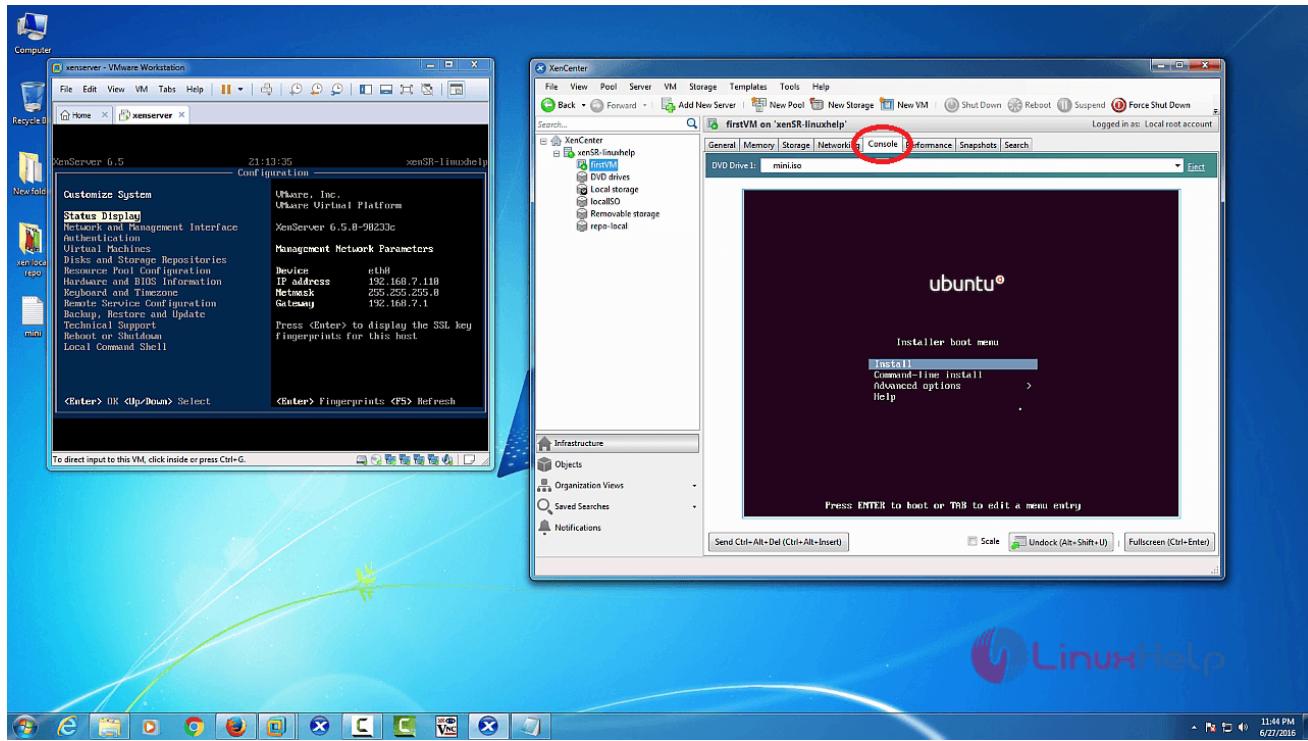
iii) Name the virtual machine

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 (IaaS) using AWS.

* Amazon EC2

EC2 stands for Elastic Compute Cloud. EC2 is an on-demand computing service on the AWS cloud platform. Under computing, it includes all the services a computing device can offer to you along with the flexibility of a virtual environment. It allows the user to configure their instances as per their requirements i.e. to allocate RAM, ROM, and storage according to the need of the current task. Even the user can dismantle the virtual device once its task is completed and it is no more required.

* Amazon Machine Image (AMI):

An Amazon Machine Image is a supported and maintained image provided by AWS that provides information required to launch an instance. You must specify an AMI when you launch an instance. You can launch multiple instances from a single AMI when you require multiple instances with the same configuration. You can use different AMIs to launch instances when you require instances with different configurations.

- Launch permissions that control which AWS accounts can use the AMI to launch instances.
- A block device mapping that specifies the volumes to attach to the instance when it's launched.

* Types of EC2 computing instances:

1) General Purpose:

Provide a balance of compute, memory, and networking resources. Ideal for applications that use resources in equal proportions such as web servers and code repositories.

2) Compute Optimized:

Ideal for compute-intensive applications that benefit from a higher ratio of compute to memory. Well-suited for batch processing workloads, media transcoding, HPC etc.

3) Memory Optimized:

Ideal for memory-intensive applications that benefit from more memory such as workloads that process large datasets from memory.

4) Storage Optimized:

Designed for workloads requiring high sequential read and write access to very large datasets on local storage.

* Elastic IP address:

An Elastic IP address is a static IPv4 address associated with AWS account.

Unlike traditional static IP addresses, Elastic IP addresses can be easily remapped to any instance in the AWS accounts VPC. Elastic IP addresses are particularly useful for hosting applications.

* Remote Desktop Protocol:

Remote Desktop Protocol is a proprietary protocol developed by Microsoft that allows users to remotely access and control windows based systems over network. In the context of AWS EC2 instances running windows server operating system.

There are several advantages to RDP. One advantage is that it does not require a VPN. It also keeps data stored securely on the user's desktop, instead of storing it on cloud servers or on the user's unsecured personal devices.

Furthermore, RDP enables companies with a legacy on-premises IT setup to allow their employees to work from home.

However, RDP may cause users to experience lag, especially if their local internet connection is slow. This can frustrate employees and cut down on their productivity. RDP also has some security vulnerabilities that leave it open to cyber attacks.

✓

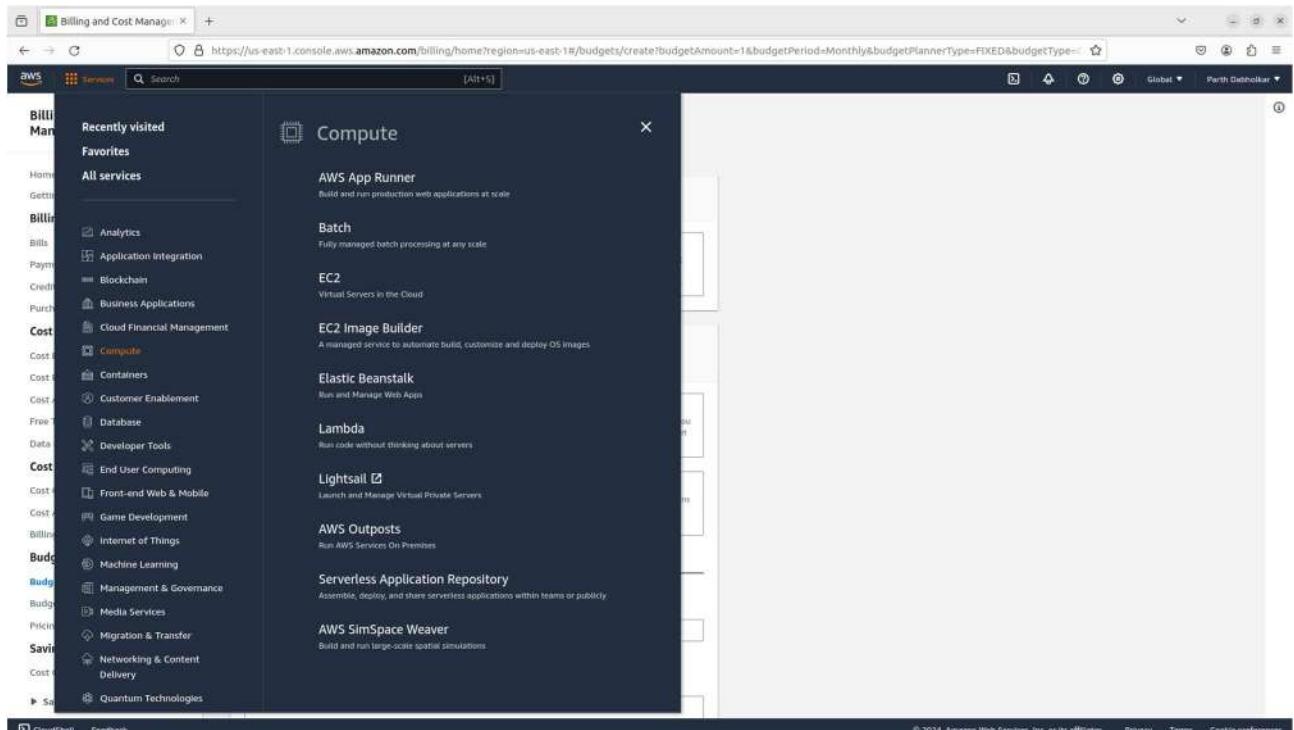
(A)

EP
21/3/24

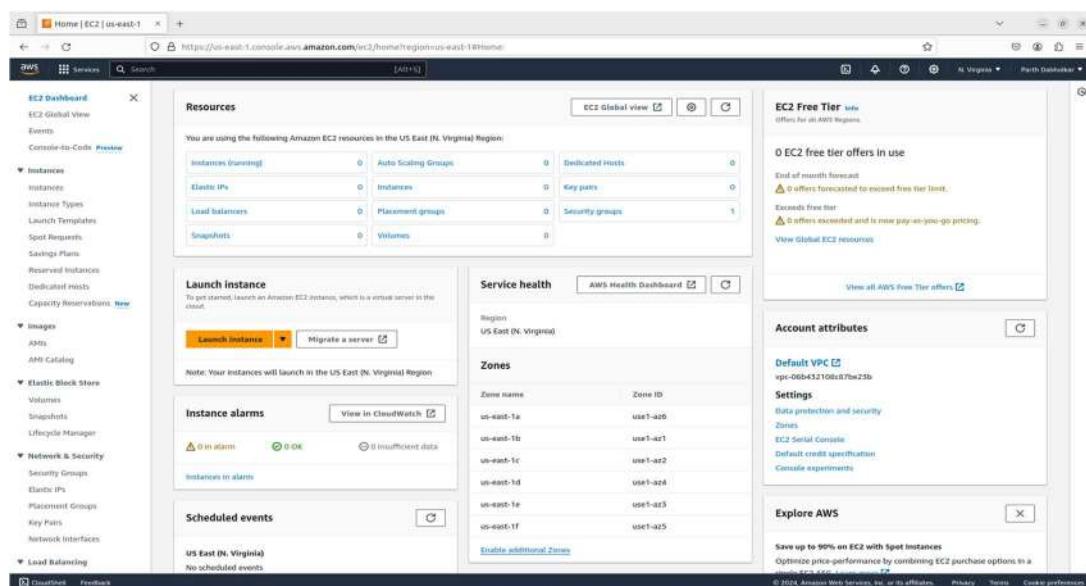
EXPERIMENT NO: 4

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 console with the path: EC2 > Instances > Launch an instance. The main title is "Launch an instance". Below it, a sub-section titled "Name and tags" is shown. A "Name" field contains the value "MeraPehlaWebServer". To the right of the name field is a link "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 AMI selection page. The main heading is "Application and OS Images (Amazon Machine Image)". Below it, a sub-section titled "Quick Start" shows icons for various operating systems: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. A search bar at the top says "Search our full catalog including 1000s of application and OS images". On the right, there is a "Browse more AMIs" button with the note "Including AMIs from AWS, Marketplace and the Community". Below the quick start section, a specific AMI is highlighted: "Ubuntu Server 22.04 LTS (HVM), SSD Volume Type". The details for this AMI include: "ami-03f4878755434977f (64-bit (x86)) / ami-077885f59ecb77b84 (64-bit (Arm))", "Virtualization: hvm", "ENA enabled: true", "Root device type: ebs", and "Free tier eligible". The "Description" section below states: "Canonical, Ubuntu, 22.04 LTS, amd64 Jammy Image build on 2023-12-07". At the bottom, the "Architecture" is listed as "64-bit (x86)", the "AMI ID" is "ami-03f4878755434977f", and there is a "Verified provider" badge.

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 the AWS instance creation wizard. The 't2.micro' option is selected, highlighted with a blue border. To the right of the selection, it says 'Free tier eligible'. Below the selection, there is descriptive text about the instance family, pricing, and supported operating systems. On the far right, there are buttons for 'All generations' and 'Compare instance types'.

Additional costs apply for AMIs with pre-installed software

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. Below, there is a dropdown menu for 'Key pair name - required' with the option 'Proceed without a key pair (Not recommended)' selected. To the right of the dropdown, there is a 'Default value' button and a 'Create new key pair' button.

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

Network | [Info](#)

vpc-00dffeb2613906df4

Subnet | [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP | [Info](#)

Enable

Additional charges apply when outside of [free tier allowance](#)

Firewall (security groups) | [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

We'll create a new security group called '**launch-wizard-1**' with the following rules:

Allow SSH traffic from Anywhere 0.0.0.0/0

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠️ 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. ✖

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

Configure storage | [Info](#) Advanced

1x GiB gp2 ▼ Root volume (Not encrypted)

ⓘ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage ✖

Add new volume

ⓘ Click refresh to view backup information ⟳

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems [Edit](#)

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

The screenshot shows the AWS EC2 'Launch an Instance' page. At the top, there is a green success message: "Successfully initiated launch of instance (i-0e5efbbbaa4545faa6)". Below this, there is a "Next Steps" section with several options:

- Create billing and free tier usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds. Includes a "Create billing alerts" button.
- Connect to your instance**: Once your instance is running, log into it from your local computer. Includes "Connect to instance" and "Learn more" buttons.
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. Includes "Connect an RDS database" and "Create a new RDS database" buttons.
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. Includes a "Create EBS snapshot policy" button.
- Manage detailed monitoring**: Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period. Includes a "Create Load Balancer" button.
- Create Load Balancer**: Create a application, network gateway or classic Elastic Load Balancer. Includes a "Create Load Balancer" button.
- Create AWS budget**: AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location. Includes a "Create AWS budget" button.
- Manage CloudWatch alarms**: Create or update Amazon CloudWatch alarms for the instance. Includes a "Manage CloudWatch alarms" button.

At the bottom of the page, there is a navigation bar with links for CloudShell, Feedback, and a search bar. The status bar at the bottom right shows the date and time as 10-04-2024, 16:12, and the location as Smoke.

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.

The screenshot shows the AWS EC2 Instances page. On the left, there is a sidebar with various navigation options like EC2 Dashboard, EC2 Global View, Events, Instances, 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, Key Pairs, and CloudShell.

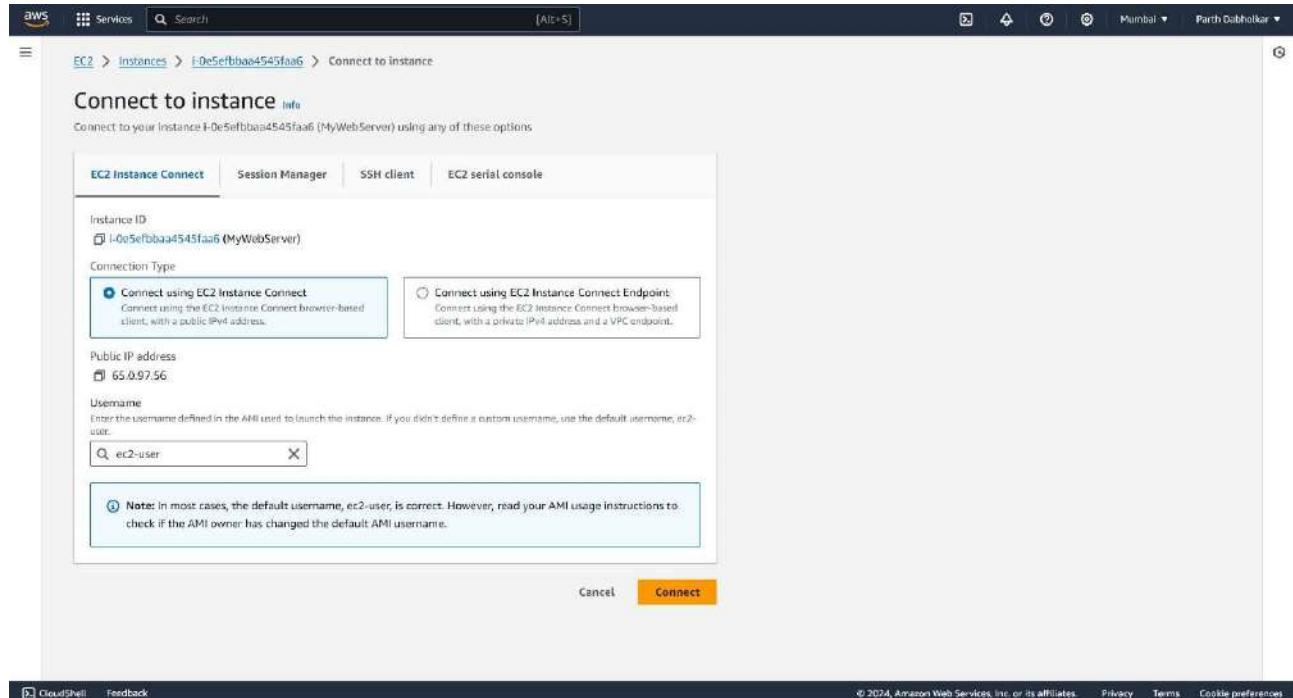
The main area displays the "Instances (1/1) Info" table. It shows one instance named "MyWebServer" with the ID "i-0e5efbbbaa4545faa6". The instance is listed as "Running" with a status check of "Initializing". It is located in the "ap-south-1b" availability zone and has a public IPv4 DNS of "ec2-65-0-97-56.ap-south-1.compute.amazonaws.com".

Below the table, there is a detailed view for the instance "i-0e5efbbbaa4545faa6 (MyWebServer)". The "Details" tab is selected, showing the following information:

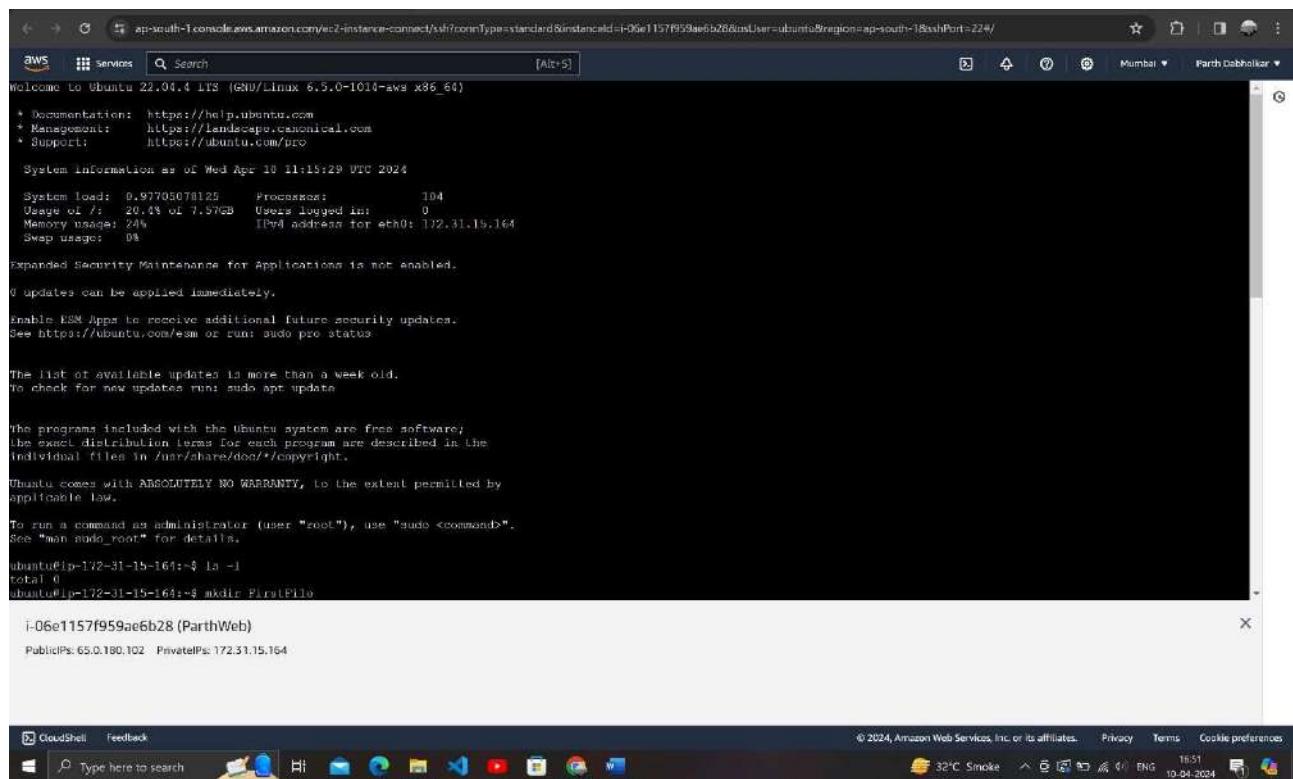
Attribute	Value
Instance ID	i-0e5efbbbaa4545faa6 (MyWebServer)
Public IPv4 address	65.0.97.56 [open address]
Private IPv4 addresses	172.31.4.140
Public IPv4 DNS	ec2-65-0-97-56.ap-south-1.compute.amazonaws.com [open address]
Private IP DNS name (IPv4 only)	ip-172-31-4-140.ap-south-1.compute.internal
Instance type	t2.micro
Elastic IP addresses	-

At the bottom of the page, there is a footer with links for CloudShell, Feedback, and a search bar. The status bar at the bottom right shows the date and time as 10-04-2024, 16:12, and the location as Smoke.

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.

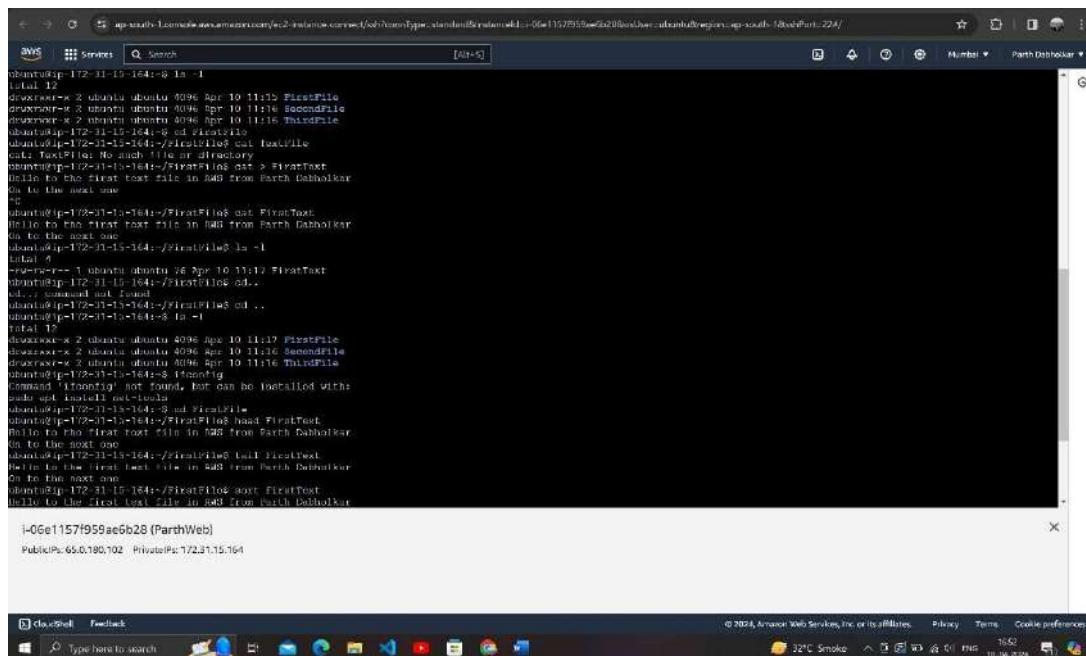


Step-12: Wait for the instance to launch. Once launched, an ubuntu terminal will be displayed on the screen. This is the virtualization of Ubuntu in AWS Cloud.



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

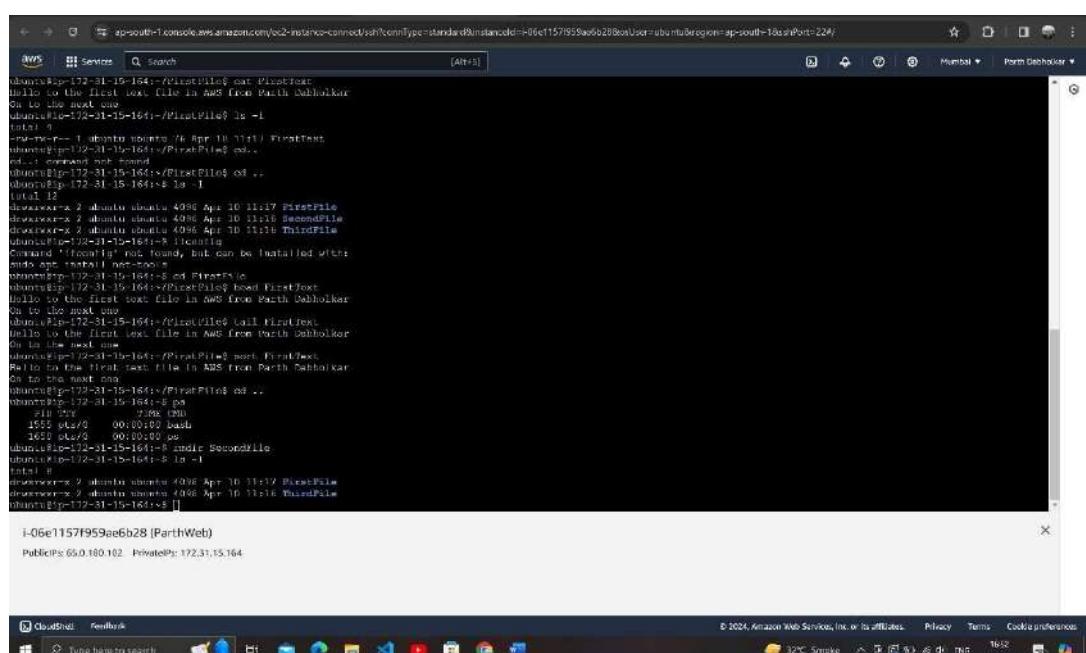
- mkdir: to create a directory
- ls -l: to list all the directories present in the system
- cd: to enter a directory
- cat > filename: to create a text file and enter data into it.
- cat filename: to display the contents of the file.
- ifconfig: to get the network address of the system.
- ps: to display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



```

ubuntu@ip-172-31-15-164:~$ ls -l
total 12
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 10 11:15 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:~/Documents$ cat Textfile1
cat: Textfile1: No such file or directory
ubuntu@ip-172-31-15-164:~/Documents$ cat FirstText
Hello to the first text file in AWS from Parth Dabholkar
On to the next one
ubuntu@ip-172-31-15-164:~/Documents$ ls -l
total 4
-rw-r--r-- 1 ubuntu ubuntu 76 Apr 10 11:12 FirstText
ubuntu@ip-172-31-15-164:~/Documents$ cd ..
cd: ..: command not found
ubuntu@ip-172-31-15-164:~/Firstfile$ cd ..
ubuntu@ip-172-31-15-164:~$ ls -l
total 12
drwxr-xr-x 2 ubuntu ubuntu 4096 Apr 10 11:12 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:~$ rm ifconfig
rm: ifconfig: command not found, but can be installed with
sudo apt install net-tools
ubuntu@ip-172-31-15-164:~$ rm Firstfile
ubuntu@ip-172-31-15-164:~/Firstfile$ head FirstText
Hello to the first text file in AWS from Parth Dabholkar
On to the next one
ubuntu@ip-172-31-15-164:~/Firstfile$ tail FirstText
Hello to the first text file in AWS from Parth Dabholkar
On to the next one
ubuntu@ip-172-31-15-164:~/Firstfile$ cat FirstText
Hello to the first text file in AWS from Parth Dabholkar
On to the next one
ubuntu@ip-172-31-15-164:~/Firstfile$ i
i-06e1157f959ae6b28 (ParthWeb)
PublicIP: 65.0.180.102 PrivateIP: 172.31.15.164

```



```

ubuntu@ip-172-31-15-164:~/Firstfile$ cat FirstText
Hello to the first text file in AWS from Parth Dabholkar
On to the next one
ubuntu@ip-172-31-15-164:~/Firstfile$ ls -l
total 1
-rw-r--r-- 1 ubuntu ubuntu 76 Apr 10 11:11 FirstText
ubuntu@ip-172-31-15-164:~/Firstfile$ cat FirstText
cat: FirstText: command not found
ubuntu@ip-172-31-15-164:~/Firstfile$ rm FirstText
rm: FirstText: command not found, but can be installed with
sudo apt install net-tools
ubuntu@ip-172-31-15-164:~/Firstfile$ head FirstText
Hello to the first text file in AWS from Parth Dabholkar
On to the next one
ubuntu@ip-172-31-15-164:~/Firstfile$ tail FirstText
Hello to the first text file in AWS from Parth Dabholkar
On to the next one
ubuntu@ip-172-31-15-164:~/Firstfile$ cat FirstText
Hello to the first text file in AWS from Parth Dabholkar
On to the next one
ubuntu@ip-172-31-15-164:~/Firstfile$ i
i-06e1157f959ae6b28 (ParthWeb)
PublicIP: 65.0.180.102 PrivateIP: 172.31.15.164

```

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, and Reservations. Below that are Images, AMIs, and AMI Catalog. Under Elastic Block Store are Volumes and Snapshots. Network & Security includes 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-06e1157f959ae6b28	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 modal window titled "Terminate instance?" is open at the bottom. It contains a warning message: "⚠ 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." Below the message is a question: "Are you sure you want to terminate these instances?". A table shows the instance ID "i-06e1157f959ae6b28 (ParthWeb)" and the "Termination protection" status as "Disabled". At the bottom, a note says: "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." Finally, there are "Cancel" and "Terminate" buttons.

EXPERIMENT NO:5

AIM: To study and implement Platform as a service (PaaS) using AWS Elastic Beanstalk service.

THEORY:

→ Amazon AWS Elastic Beanstalk Service:

AWS Elastic Beanstalk is a AWS managed service for web applications. Elastic Beanstalk is a pre-configured EC2 service that can directly take up your application code and environment configurations and use it to automatically provision and deploy the required resources within AWS to run the web applications. As a Platform as a Service (PaaS), it allows users to directly use a pre-configured server for their application.

* Language/Framework supported by BES:

Elastic Beanstalk supports a wide range of programming languages and frameworks, making it flexible and accessible to developers working with different languages such as:

- Java
- .NET
- Node.js
- Python
- Ruby
- PHP

* Compare EC2 and Elastic Beanstalk:

EC2

EBS

- | | |
|--|---|
| 1. It provides raw virtual machines (instances) in cloud users have full control over OS and networking. | 1. It abstracts away the solution infrastructure details and provides platforms as a service. |
| 2. Users are responsible for deploying applications managing server updates. | 2. EBS takes care of the underlying infrastructure. |
| 3. Supports a wide range of OS and services. | 3. supports specific programming languages and frameworks. |
| 4. Pricing is based on selected instance type and usage. | 4. Pricing based on resources used by underlying infrastructure. |
| 5. More effort needed to handle increased traffic. | 5. Monitors performance and 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 the health of its registered targets, and route traffic only to healthy targets.

* Steps for deploying web apps/web services on AWS Elastic Beanstalk:

Step 1: Login to AWS console and go to Elastic Beanstalk.

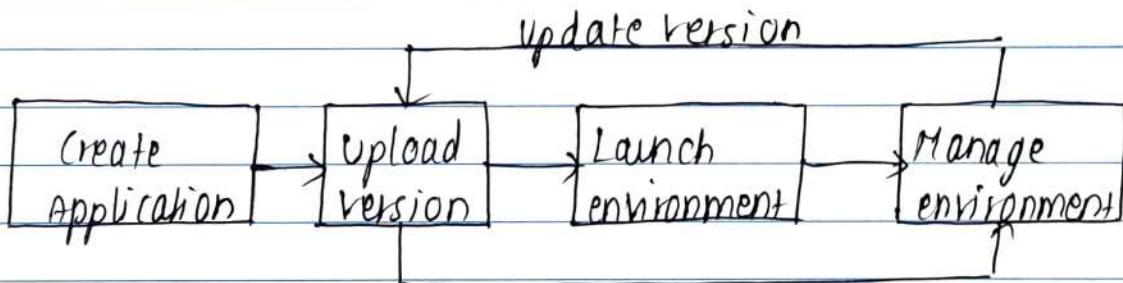
Step 2: click on create application.

Step 3: write app information : Name, tag, platform etc.

Step 4: In app code , select sample application and then click on button "Create Application".

Step 5: Click on environments → check the health of environments till it becomes 'OK'.

Step 6: Click on the URL.



Deploy new version

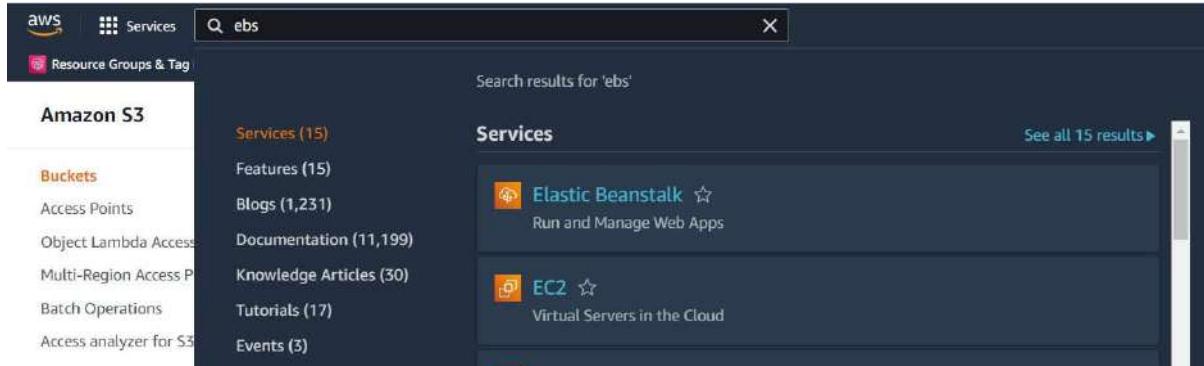
(A)

RP
21324

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



Step 2: Click on Create Application

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

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	Runvers
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	Runvers
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
Green	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	Support

The screenshot shows a blue header with the text "Congratulations!" in large white font. Below it, a message says: "Your AWS Elastic Beanstalk PHP application is now running on your own dedicated environment in the AWS Cloud". It also indicates "You are running PHP version 8.0.13" and "This environment is launched with Elastic Beanstalk PHP Platform". To the right, there are two sections: "What's Next?" with links to 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, and Customizing Environment Resources; and "AWS SDK for PHP" with links to AWS SDK for PHP home, PHP developer portal, and AWS SDK for PHP on GitHub.

Congratulations!

Your AWS Elastic Beanstalk PHP application is now running on your own dedicated environment in the AWS Cloud.

You are running PHP version 8.0.13

This environment is launched with Elastic Beanstalk PHP Platform

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 portal](#)
- [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.

* Cloud Storage:

Cloud storage is a cloud computing model that enables storing data and files on the internet through a cloud computing provider that you access either through a public internet or a dedicated private network connection. The provider securely stores, manages and maintains the storage servers, infrastructure and network to ensure you have access to data when you need it at virtually unlimited scale, and with elastic capacity. Cloud storage removes the need to buy and manage your own data storage infrastructure, giving you agility, scalability and durability, with anytime, anywhere data access.

* Owncloud and its Features:

Owncloud is a free open-source software project for content collaboration and sharing and syncing of files in distributed and federated enterprise scenarios. It allows companies and remote end-users to organize their documents on servers, computers and mobile devices and work with them collaboratively while keeping a centrally organized and synchronized state.

* Advantages and Limitations:

* Advantages:

1. Scalability: Cloud storage eliminates the need for support or down based on user demand.

2. Accessibility: Users can access their data from anywhere with internet connection.

3. Cost-effectiveness: Cloud storage eliminates the need for support up front hardware investments and allows organization to pay only for storage.

* Limitations:

1. Dependency on connectivity: Cloud storage requires a stable internet.

2. Data security: Storing sensitive data in the cloud raises concerns about data security.

3. Vendor Lock-in: Switching between cloud storage providers can be challenging due to formats.

* Different types of storage:

✓ 1. Object storage: Object storage is a storage architecture that manages data as objects.

2. Block storage: Block storage divides data into blocks and stores them as separate entities with unique identifiers.

* Popular storage as a service vendors:

1. Amazon Simple Storage Service (S3):

Amazon offers a range of storage services for businesses including Amazon S3, an industry-leading product that helps companies scale storage resources efficiently. This object-storage solution needs fluctuating business needs with high-performance computing applications. It can be easily integrated with Amazon Web Services (AWS) and other third-party services for better data insights.

2. Dropbox:

Dropbox offers a suite of products to help with efficient data storage and organization, streamlined document workflows, automatic backup and restorations, secure sharing and quick transfers. Used by more than 600,000 teams globally, the company offers various plan options for business of all sizes as well as custom solutions to meet your business goals.

3. Google workspace:

Google offers enterprise-grade cloud storage services with Google Workspace, including a wide range of essential business applications that make storing, accessing and sharing data easier within the same platform. This flexible and innovative solution boosts business productivity and coordination with easy file sharing, centralized administration and real-time collaboration.

4. IBM Cloud Object Storage:

IBM offers efficient cloud storage solutions to support growing business needs. IBM cloud storage object can accommodate exponential data growth with reduced storage costs and allow flexible resiliency options across multiple geographic locations.

5. Microsoft Cloud:

Microsoft Cloud is a preferred solution for data storage used by more than 85% of Fortune 500 companies. It offers a productivity platform with user-centric, AI-powered solutions for a secure and sustainable cloud storage system.

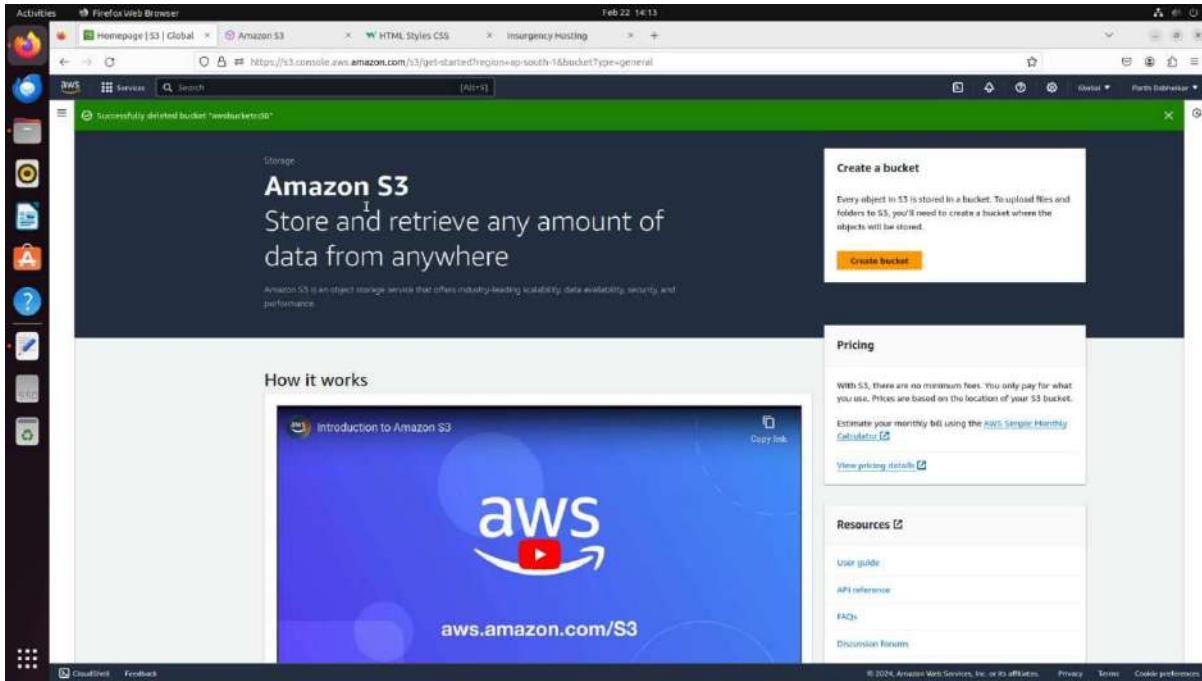
A+

2/3/24

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.

General configuration

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

Bucket name: AwsbucketCCLE

Object Ownership info

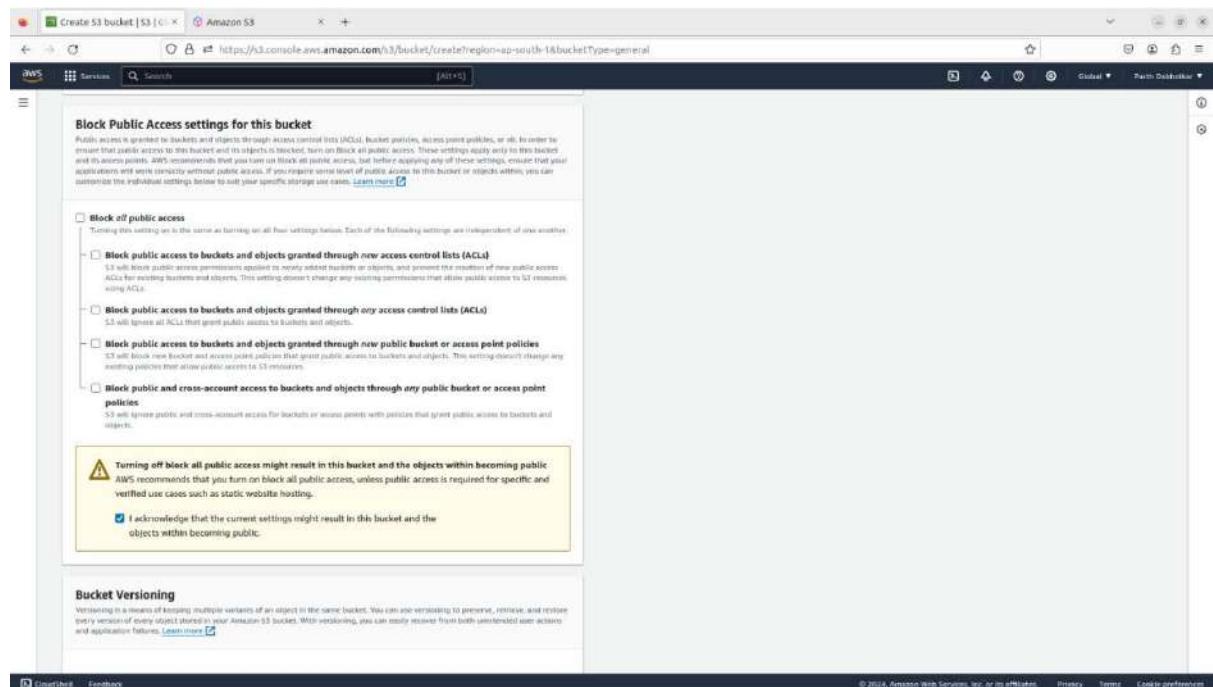
ACLs disabled (Recommended): All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using entry policies.

ACLs enabled: Objects in this bucket can be owned by other AWS accounts. Access to the bucket and its objects can be specified using ACLs.

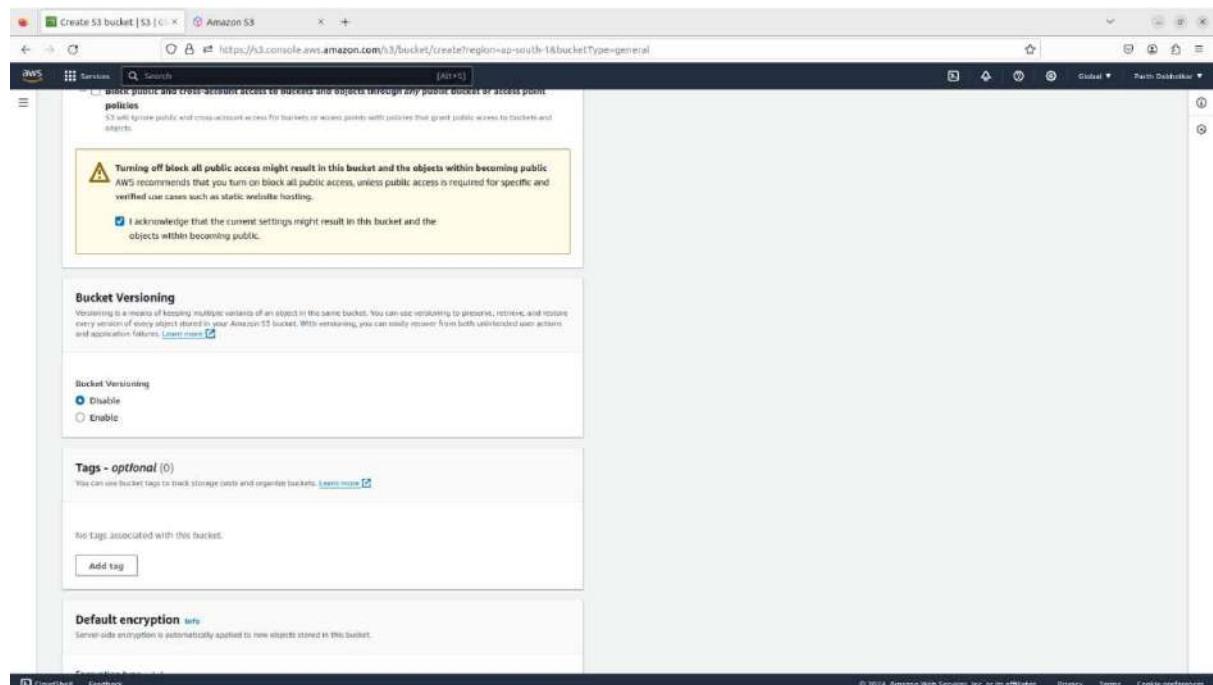
Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, or access point policies, or all. In order for access to your objects to be granted, they must be bucket or object level grants. These settings apply only to this bucket.

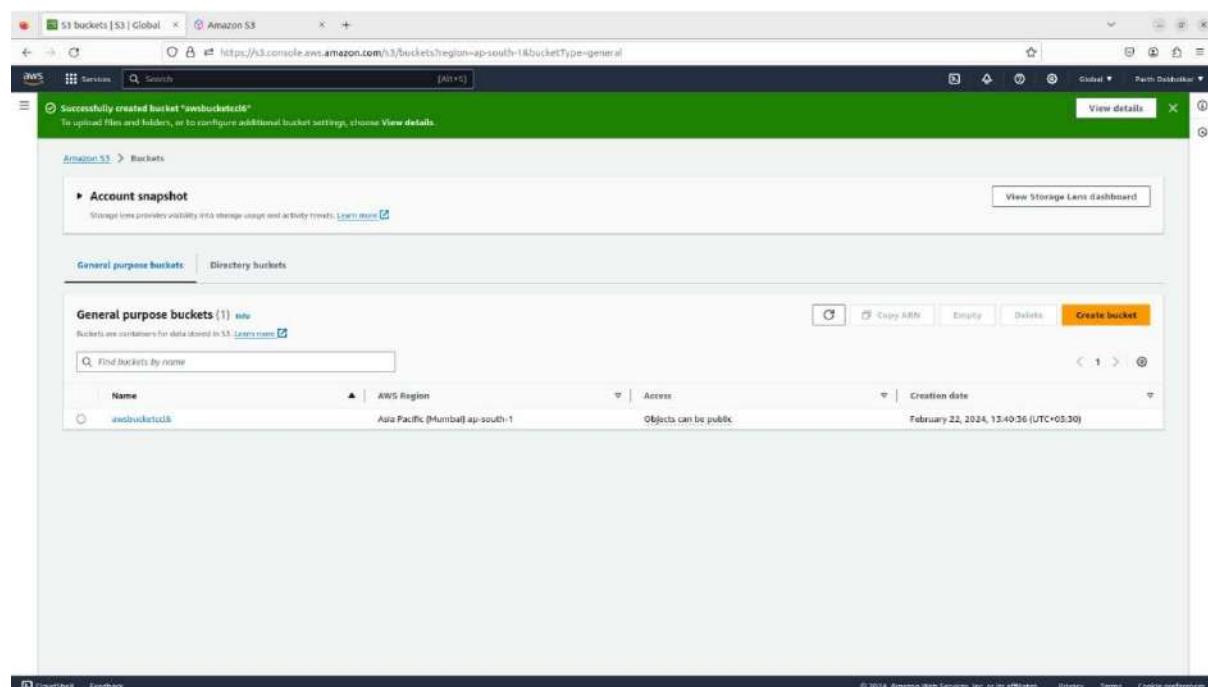
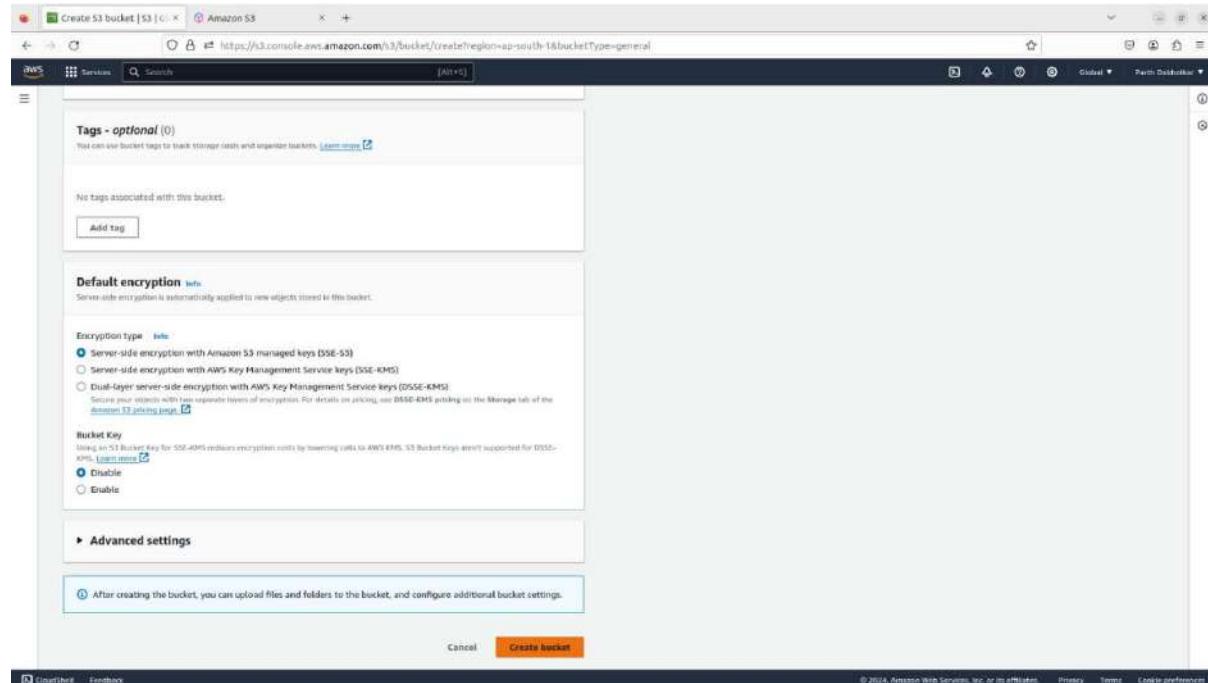
Step-3: Disable block all public access checkbox. Select the checkbox for Turning off block all public access might result in this bucket and the objects within becoming public.



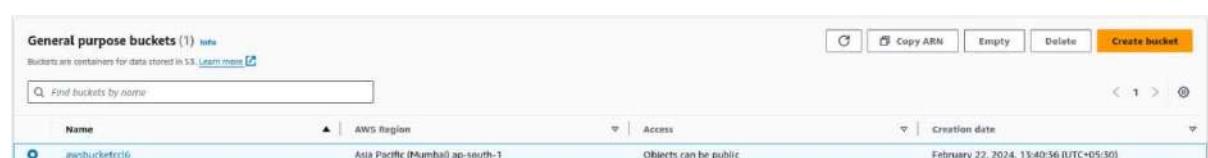
Step-4: Keep bucket versioning as disabled and add tags if required.



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 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 'Upload' interface. A file named 'websites.html' is selected for upload. The destination is set to 's3://newbuckettest0'. The 'Upload' button is highlighted in orange.

The screenshot shows the 'Upload: status' interface after the file has been uploaded successfully. The summary shows 1 file uploaded (websites.html) with 100% success rate. The 'File and folders' tab is selected, showing the uploaded file in a table.

Name	Folder	Type	Size	Status
websites.html		text/html	712.0 B	Succeeded

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.

The screenshot shows the 'Edit static website hosting' configuration page in the Amazon S3 console. The 'Static website hosting' section is active, with the 'Enable' radio button selected. The 'Index document' field is set to 'index.html'. A note at the bottom of the section states: 'For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access.' Below this, there's a 'Error document - optional' field containing 'error.html'.

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

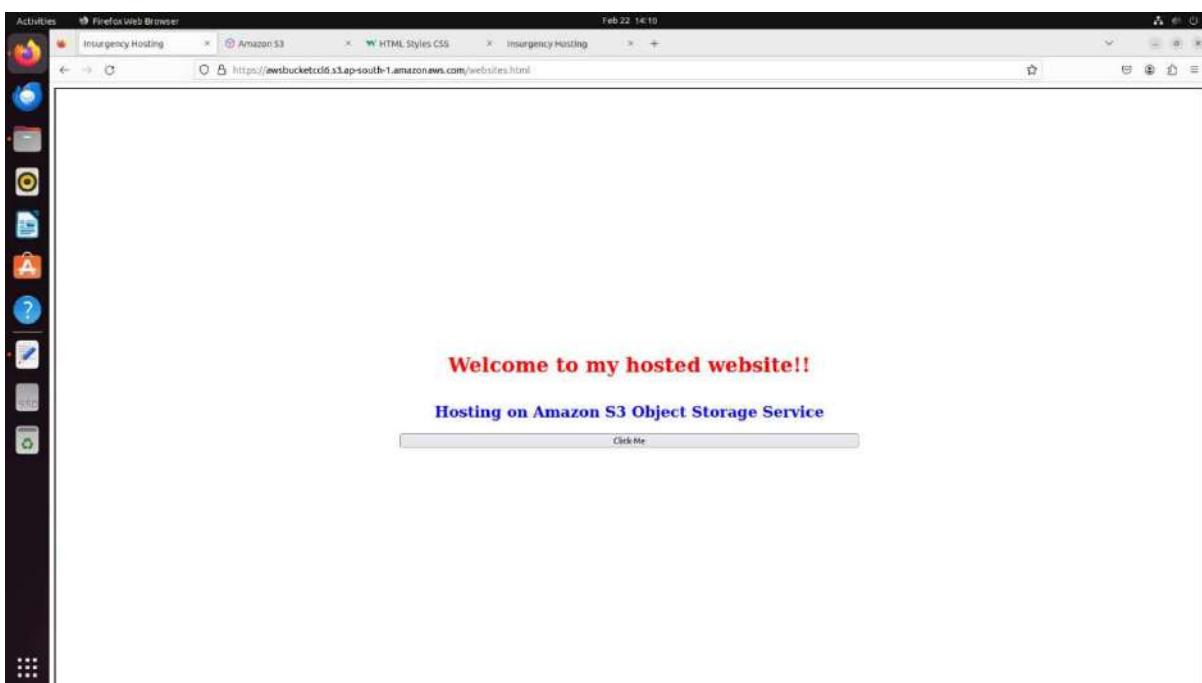
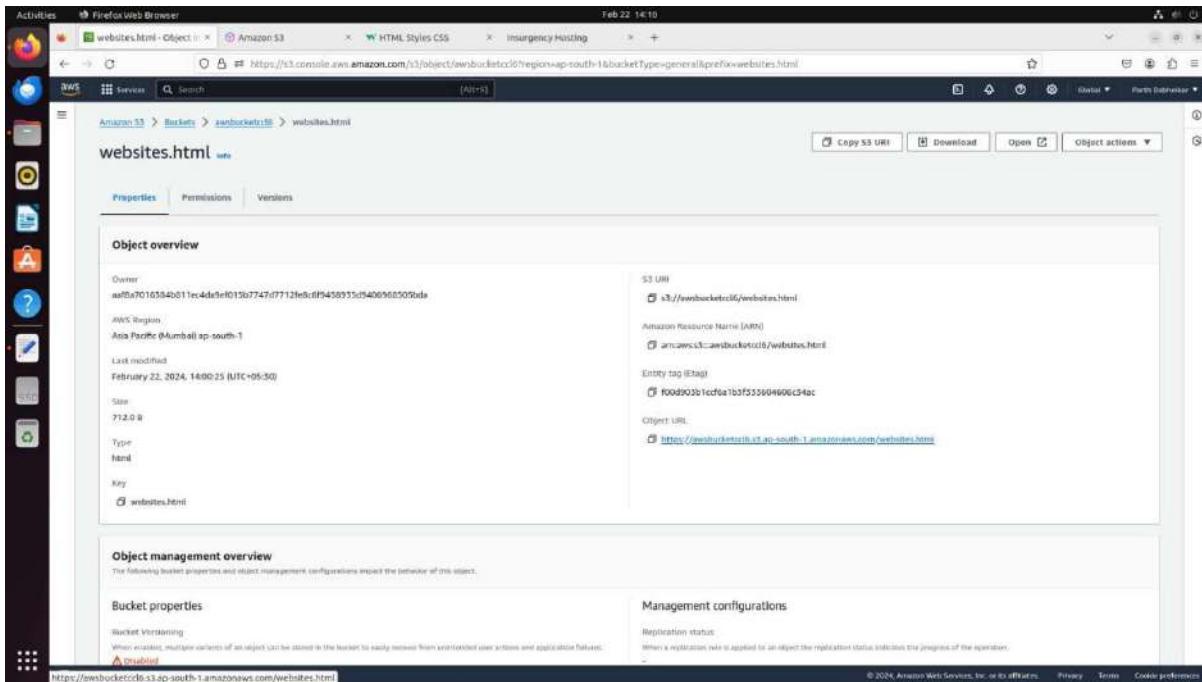
The screenshot shows the AWS Lambda function configuration page. The 'Handler' field contains the following code:

```

    1  "version": "2012-10-17",
    2  "statement": [
    3      {
    4          "principal": "arn:aws:lambda:ap-south-1::awsbucketcc16",
    5          "action": "PutObject",
    6          "resource": "arn:aws:s3:::awsbucketcc16/*"
    7      }
    8  ]
  
```

The screenshot shows the AWS S3 bucket permissions settings page. The 'Block public access' section is set to 'Off'. The 'Individual Block Public Access settings for this bucket' link is visible.

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



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.

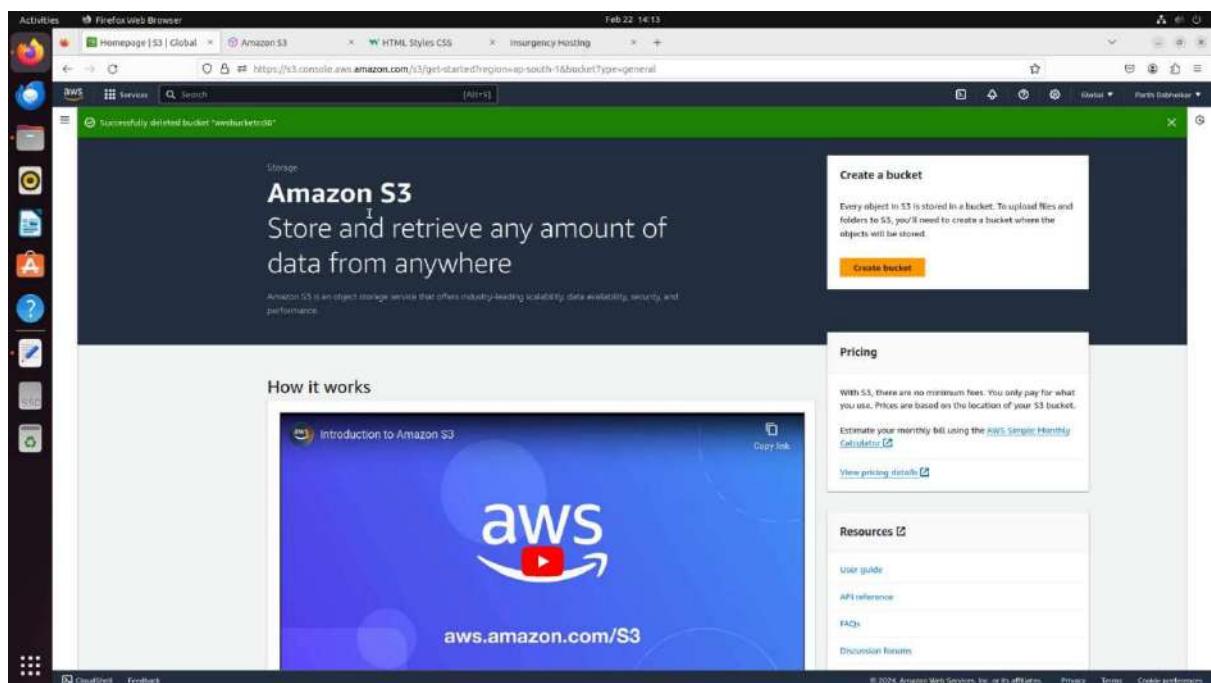
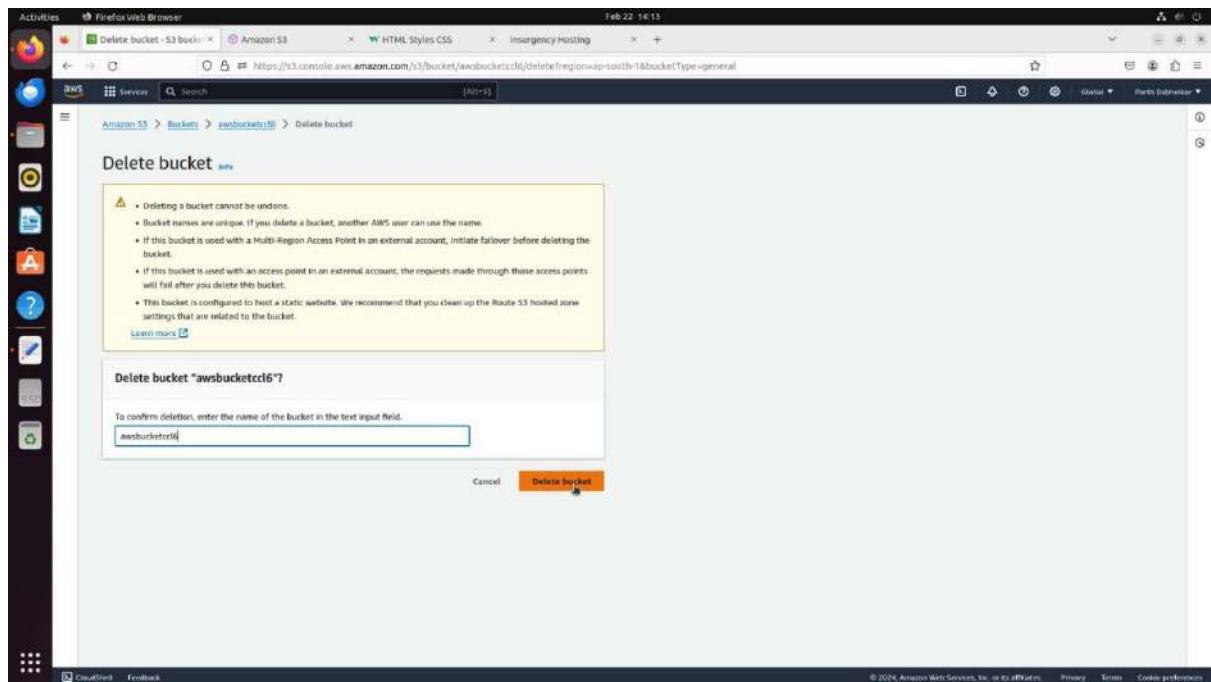
The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with various AWS services like Buckets, Access Grants, and Storage KMS. The main area shows a list of objects in the 'awsbucketcc16' bucket. One object, 'websites.html', is selected. The toolbar at the top has several buttons: Copy S3 URL, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload. The 'Delete' button is highlighted in orange. Below the toolbar is a search bar and a table showing the object details: Name (websites.html), Type (HTML), Last modified (February 22, 2024, 14:00:25 (UTC+05:30)), Size (712.0 B), and Storage class (Standard).

This screenshot shows the 'Delete objects' confirmation dialog. It displays a warning message about deleting objects and folders, followed by a 'Learn more' link. Below that is a section titled 'Specified objects' showing the selected object 'websites.html'. At the bottom, there's a section for confirming the deletion with the text 'To confirm deletion, type permanently delete in the text input field.' A text input field contains the text 'permanently delete'. There are 'Cancel' and 'Delete objects' buttons at the bottom right.

The screenshot shows the AWS S3 console in Firefox. A modal window titled 'Delete objects: status' is open, indicating that 1 object was successfully deleted from the source 's3://newbucket005'. The 'Failed to delete' tab shows 0 objects. The 'Configuration' tab is also visible.

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 console in Firefox. The left sidebar is expanded, showing options like Buckets, Storage Lens, and Feature spotlight. The main area displays the 'General purpose buckets' list, which contains one entry: 'insurgency-hosting'. The 'Delete' button is highlighted in the toolbar above the table.



EXPERIMENT: NO:7

AIM: To study and Implement Identity and Access Management practices on AWS.

THEORY:

* Concept and need of access management:

Access management is the process of controlling and managing user access to resources within an organization IT infrastructure.

* IAM and its components:
IAM is a service provided by AWS that enables users to securely control access to AWS.

- **Users:** Individuals within an organization who interact with AWS resources.

- **Groups:** Collections of users who have the same set of permissions.

- **Roles:** Entities that define a set of permissions and can be assumed by users.

* Compassion in AWS:

i) Root Users and other IAM:

Root users have unrestricted access to all AWS resources and services within the account.

Comparison Roles and Policies:

- Roles defines set of permission that can be assumed by users, AWS services.

* Inline and Custom Policies:

• Inline Policies:

inline policies are policies that are embedded directly in IAM users, groups or roles.

• Custom Policies:

custom policies are stand-alone policies that are created independently of IAM entities.

* MFA in AWS:

MFA is an additional layer of security that requires users to provide two or more forms of authentication before gaining access to AWS resources.

✓ A

SP
21/3/24

Step 1: Check the EC2 dashboard to see if any instances are running in the background or not. If running, delete/ terminate the instance.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like 'EC2 Global View', 'Events', 'Instances' (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, and New), 'Images' (with AMIs), and 'CloudShell' and 'Feedback' buttons. The main content area has a 'Resources' section with a table showing counts for various EC2 components: Instances (running) 0, Auto Scaling Groups 0, Dedicated Hosts 0, Elastic IPs 0, Instances 0, Key pairs 0, Load balancers 0, Placement groups 0, Security groups 1, Snapshots 0, and Volumes 0. To the right of this is an 'EC2 Free Tier' section with a summary: 0 EC2 free tier offers in use, End of month forecast (0 offers forecasted to exceed free tier limit), and Exceeds free tier (0 offers exceeded and is now pay-as-you-go pricing). There are also links to 'View Global EC2 resources' and 'View all AWS Free Tier offers'. At the bottom of the dashboard are standard AWS footer links: © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

Step 2: Go the IAM Dashboard in the AWS services in the navbar.

The screenshot shows the AWS IAM Dashboard. The left sidebar includes 'Identity and Access Management (IAM)', a search bar, and sections for 'Dashboard', 'Access management' (User groups, Users, Roles, Policies, Identity providers, Account settings), and 'Access reports'. The main area features a 'Security recommendations' section with two items: 'Add MFA for root user' (marked with a red exclamation point) and 'Root user has no active access keys' (marked with a green checkmark). Below this is an 'IAM resources' section titled 'Resources in this AWS Account' with a table showing counts: User groups 0, Users 0, Roles 2, Policies 0, and Identity providers 0. The bottom of the page follows the standard AWS footer pattern with links to © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

Step 3: Create on create option under Account Alias and give a valid name.

The screenshot shows the AWS IAM console with a modal window titled "Alias Created for ParthD". The modal displays the following information:

- AWS Account**
- Account ID: 851725389754
- Account Alias: ParthD
- Sign-in URL for IAM users in this account: https://parthd.signin.aws.amazon.com/console

Below the modal, the "Quick Links" section is visible, which includes a link to "My security credentials". The left sidebar shows the navigation menu for IAM, including "Dashboard", "Access management", "Access reports", and "Identity and Access Management (IAM)". The bottom of the screen includes standard AWS footer links: CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

Step 4: Click on create user option in the users section of IAM Dashboard.

The screenshot shows the AWS IAM console with the "Users" section open. The interface includes:

- Users (0) Info**: An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
- A search bar labeled "Search".
- A table header with columns: User name, Path, Group, Last activity, MFA, and Pass. A note below the header says "No resources to display".
- Standard AWS footer links: CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

The left sidebar shows the navigation menu for IAM, including "Dashboard", "Access management", "Access reports", and "Identity and Access Management (IAM)".

Step 5: Enter the all the user details, set permissions if required, review the details and then create the IAM user.

The screenshot shows the 'Retrieve password' step of the IAM user creation process. A green header bar at the top indicates 'User created successfully'. Below it, a message says 'You can view and download the user's password and email instructions for signing in to the AWS Management Console.' On the left, a vertical navigation menu lists 'Step 1 Specify user details', 'Step 2 Set permissions', 'Step 3 Review and create', and 'Step 4 Retrieve password'. The main content area is titled 'Console sign-in details' and contains fields for 'Console sign-in URL' (https://parthd.signin.aws.amazon.com/console), 'User name' (ParthD), and 'Console password' (a masked password). A 'Show' link is available for the password field. A 'Email sign-in instructions' button is also present. At the bottom right are 'Cancel', 'Download .csv file', and 'Return to users list' buttons. The footer includes links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

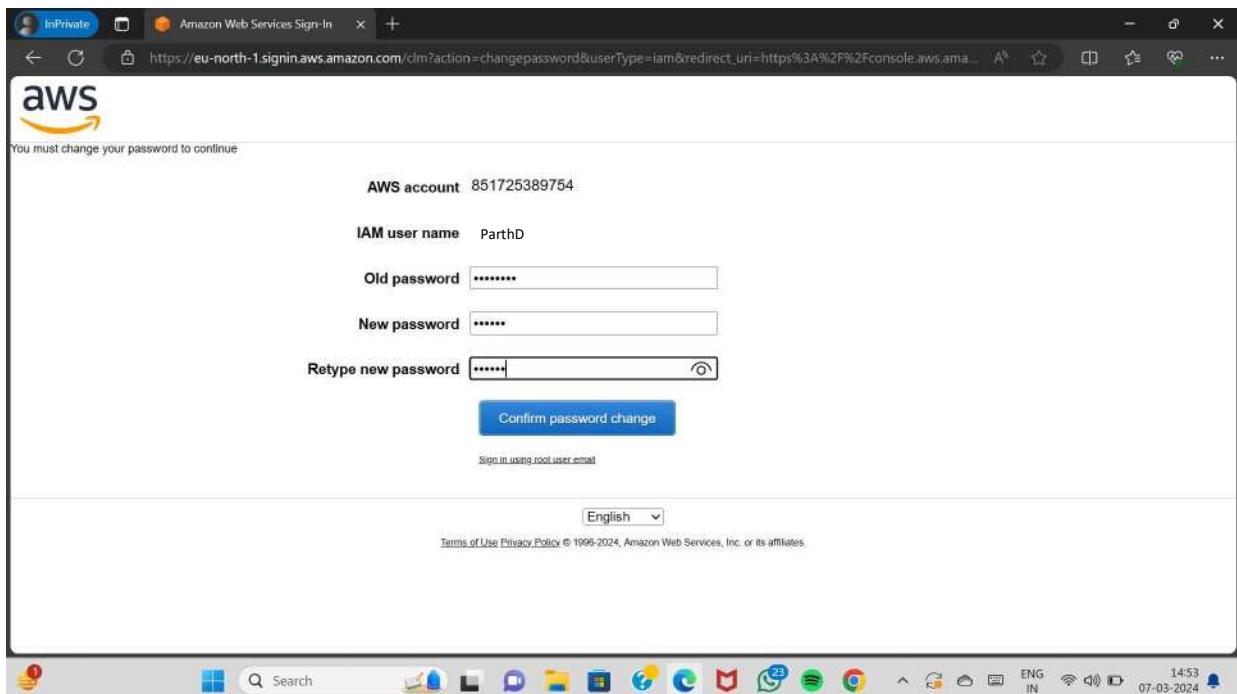
This screenshot shows the same 'Retrieve password' step as the previous one, but with a different password. The 'Console password' field now displays 'xY6m!3jq' with a 'Hide' link. The rest of the interface is identical, including the green success message, the navigation menu, and the 'Console sign-in details' section. The footer links are also the same.

Step 6: Check the IAM Users dashboard. The username must be visible.

The screenshot shows the AWS Identity and Access Management (IAM) service interface. On the left, a sidebar menu includes options like Dashboard, Access management (with sub-options User groups, Users, Roles, Policies, Identity providers, Account settings), and Access reports. The main content area is titled "Users (1) Info" and contains a table with one row for "ParthD". The table columns include User name, Path, Group, Last activity, MFA, and Password. A "Create user" button is located at the top right of the user list. The bottom of the screen displays standard AWS footer links: CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

Step 7: Now the IAM user has been created. Enter the account details and signin using IAM account details.

The screenshot shows a Microsoft Edge browser window. The address bar displays a URL starting with "https://eu-north-1.sigin.aws.amazon.com/oauth?response_type=code&client_id=am%3Aaws%3Asignin%3A%3Aconsole%2F...". The main content area features the AWS sign-in form with fields for "Account ID (12 digits) or account alias" (filled with "ParthD"), "IAM user name" (filled with "ParthD"), and "Password" (filled with "*****"). There is also a "Remember this account" checkbox and a "Sign in" button. To the right of the sign-in form is a promotional banner for "Amazon Lightsail" with the text "Lightsail is the easiest way to get started on AWS" and a "Learn more »" button. The banner features a cartoon robot character giving a thumbs up. The bottom of the screen shows the Windows taskbar with various pinned icons and system status indicators.



Step 8: Once logged-in using IAM account, we can see that the user does not have any permission to use the AWS services.

A screenshot of the AWS EC2 Dashboard. The left sidebar shows navigation options like EC2 Global View, Events, Instances (with sub-options like Instances, Instance Types, Launch Templates, etc.), Images (AMIs, AMI Catalog), and CloudShell. The main content area is titled "Resources" and displays a message: "You are using the following Amazon EC2 resources in the Asia Pacific (Sydney) Region:". Below this is a grid of resource status cards: Instances (running) 0, Auto Scaling Groups (API Error), Dedicated Hosts (API Error); Elastic IPs (API Error), Instances (API Error), Key pairs (API Error); Load balancers (API Error), Placement groups (API Error), Security groups (API Error); Snapshots (API Error), Volumes (API Error). Below the grid, there's a "Launch instance" section with a "Launch instance" button and a "Migrate a server" button. To the right is a "Service health" section with a "AWS Health Dashboard" button and a note: "Region: Asia Pacific (Sydney)". The bottom of the screen shows the Windows taskbar with pinned icons and the date/time (07-03-2024, 14:58).

The screenshot displays two browser windows side-by-side, both showing the AWS interface.

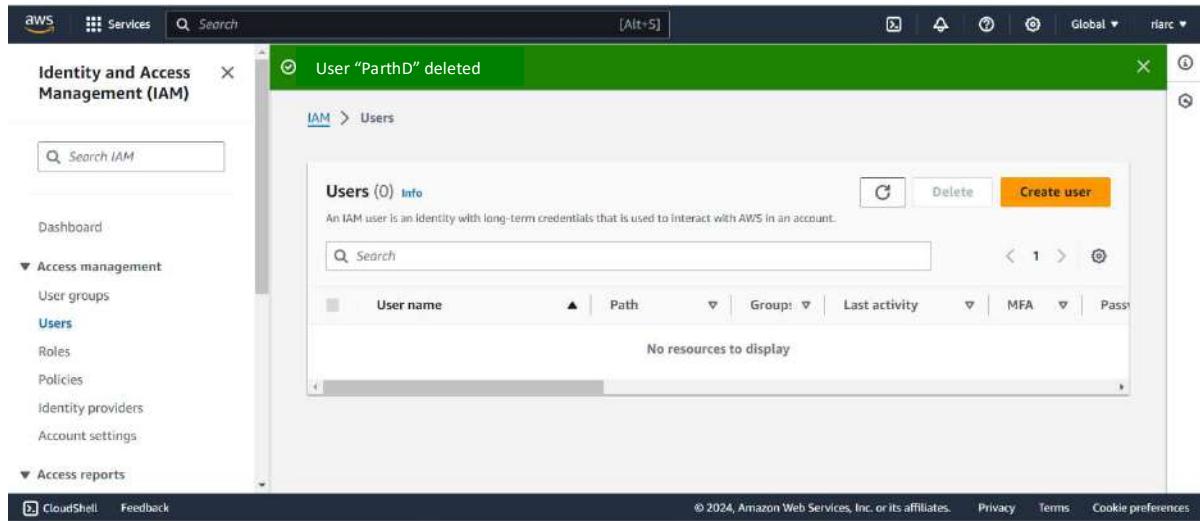
AWS EC2 Instances Page:

- URL:** https://ap-southeast-2.console.aws.amazon.com/ec2/home?region=ap-southeast-2#instances
- Header:** InPrivate, Instances | EC2 | ap-southeast-2
- Left Sidebar:** EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations (New), Images (AMIs, AMI Catalog).
- Main Content:** Instances Info tab, Find Instance by attribute or tag (case-sensitive) search bar, Instance state dropdown set to Running, Actions dropdown, Launch instances button. A message states: "You are not authorized to perform this operation. User: arn:aws:iam::851725389754:user/ria-c is not authorized to perform: ec2:DescribeInstances because no identity-based policy allows the ec2:DescribeInstances action". Below this is a "Select an instance" section.
- Bottom:** CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, Cookie preferences.

AWS S3 Homepage:

- URL:** https://s3.console.aws.amazon.com/s3/get-started?region=ap-southeast-2
- Header:** InPrivate, Homepage | S3 | Global
- Left Sidebar:** Storage (selected), Amazon S3, Store and retrieve any amount of data from anywhere.
- Content:** A dark-themed landing page for Amazon S3. It features a "Create a bucket" button and a "Pricing" section. The "Create a bucket" section explains that every object in S3 is stored in a bucket and provides a link to create one. The "Pricing" section states that there are no minimum fees and provides details about price calculations based on location.
- Bottom:** CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, Cookie preferences.

Step 9: Now delete the account by going to IAM dashboard. Select the checkbox and click on delete. A green alert box stating user deleted will be displayed. This confirms the account is being deleted.



EXPERIMENT NO: 8

AIM: To study and implement Database as a service on SQL database using AWS RDS.

THEORY:

Database as a service is cloud computing service model where database management tasks are outsourced to third-party provider.

Providers and services:

1. Amazon Web Service: Amazon RDS, Amazon Aurora
2. Microsoft Azure: Azure SQL Database, cosmos DB
3. Google Cloud Platform

SQL/NoSQL Database supported by AWS:

AWS supports both SQL and NoSQL.

- SQL Databases: Amazon RDS supports SQL database such as MySQL.

- NoSQL database: AWS provides services like Amazon Dynamo DB.

* Difference Between RDS and Aurora:

- **RDS:** It is a managed service that supports various relational databases like MySQL, PostgreSQL.
- **Aurora:** Aurora is high performance MySQL and PostgreSQL-compatible relational database built for cloud.
- terms in AWS with respect to DBaaS

1) Storage types:

General Purpose (SSD) : suitable for a broad range of database and workloads.

2) End points:

An endpoint is the URL used to connect to a database instance. It consists of DNS name and port number.

3) Snapshots:

A snapshot is a backup of database instance stored in Amazon S3.

4) Read Replica:

A read replica is a copy of the primary database.

5) Single AZ: Database instance deployed in single availability zone.

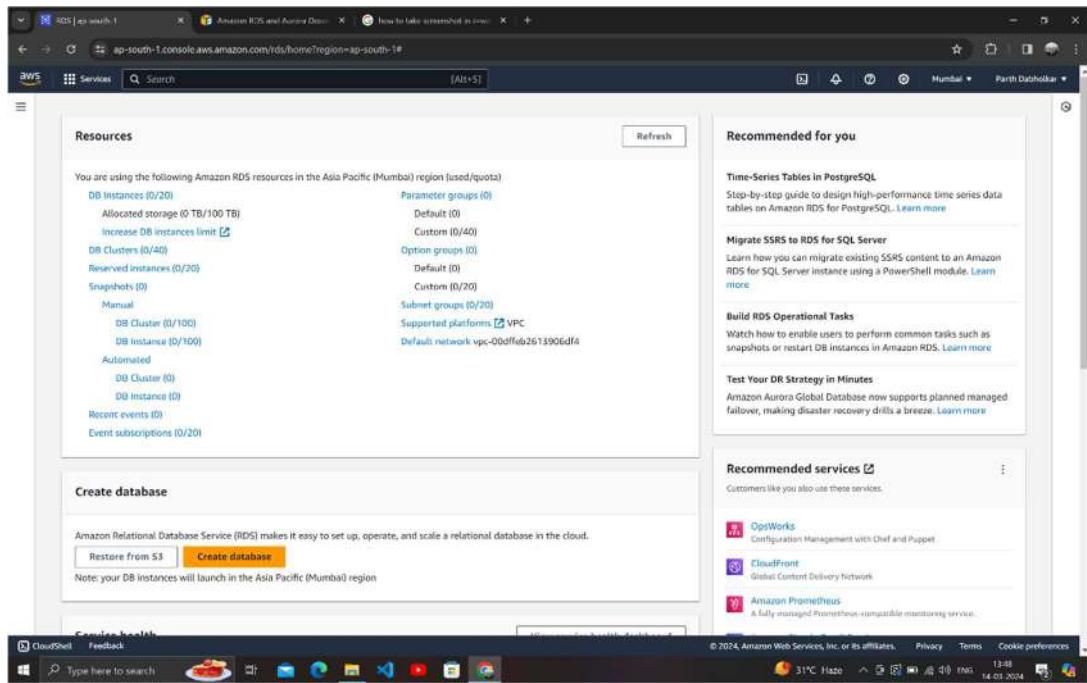
✓ (A+) CP
21/3/24

EXPERIMENT NO: 8

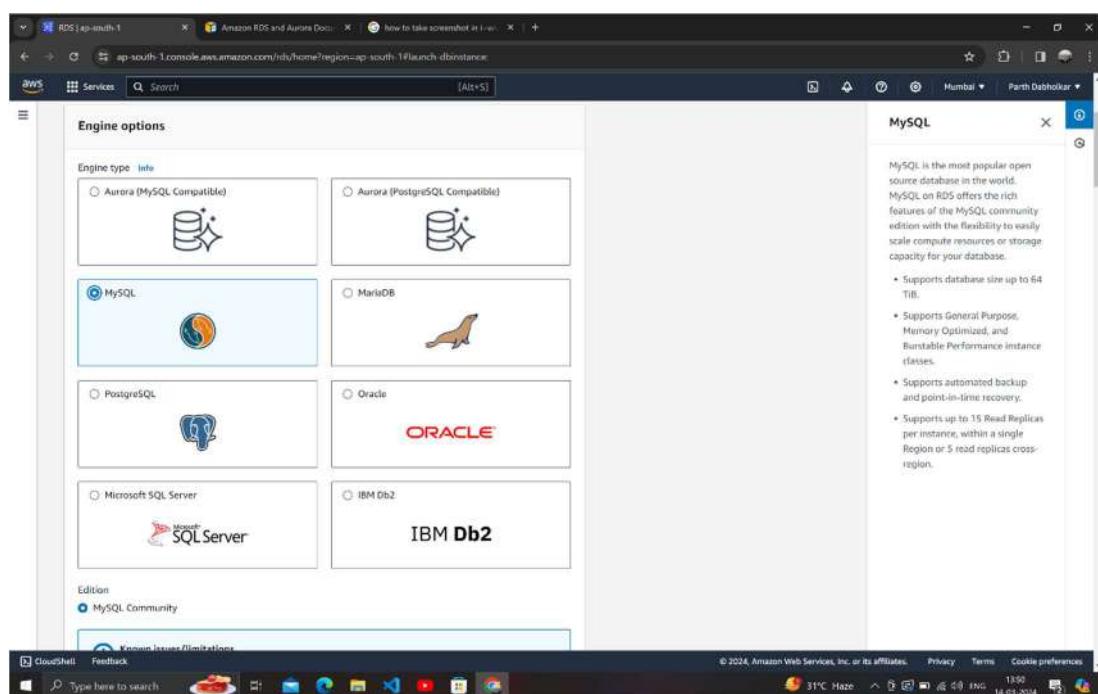
AIM: To study and implement database as a service on AWS RDS(Relational Database Services).

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 described as being intended for development use outside of a production environment. The 'Production' and 'Dev/Test' options are also shown. To the right, a detailed description of MySQL is provided, highlighting its popularity and various features like support for up to 64 TiB and automated backup.

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. The 'DB instance identifier' is set to 'test-database-parth'. Under 'Master username', 'admin' is entered. A note states that managing master user credentials in Secrets Manager is not supported. Below, there are fields for 'Master password' and 'Confirm master password', both of which have been filled with '*****'. To the right, a detailed description of MySQL is provided, highlighting its popularity and various features like support for up to 64 TiB and automated backup.

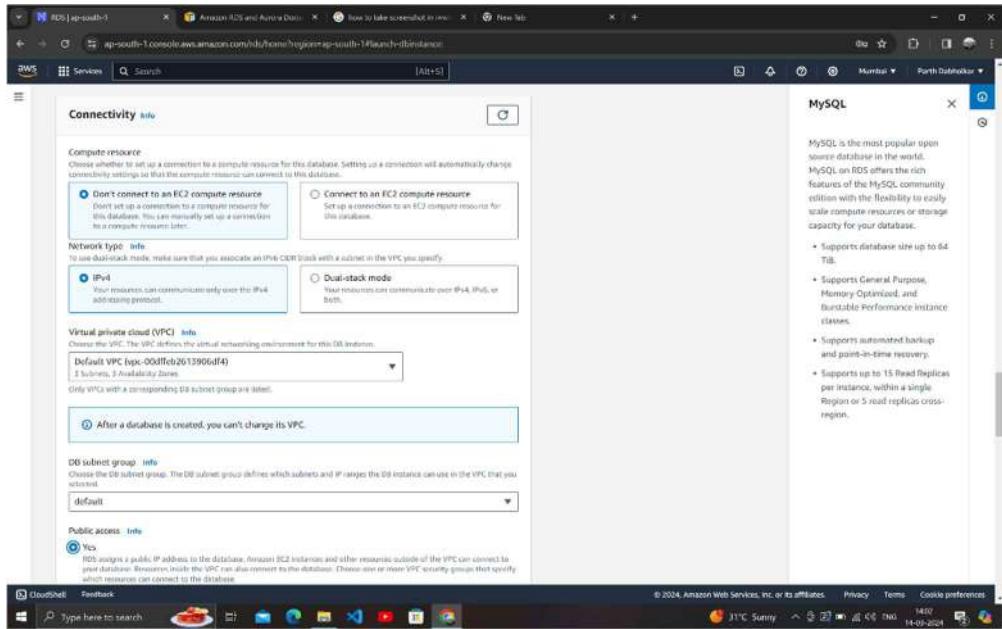
5. In instance selection, select db.t3.micro which consists of 2 CPUs and 1GB RAM.

The screenshot shows the AWS RDS MySQL instance configuration page. On the left, under 'Instance configuration', the 'DB instance class' dropdown is set to 'db.t3.micro'. This class is described as having 2 vCPUs, 1 GiB RAM, and Network: 2,085 Mbps. On the right, a sidebar titled 'MySQL' provides information about MySQL, including its popularity as an open-source database and its features like support for up to 64 TiB, General Purpose, Memory Optimized, and Burstable Performance instance classes, automated backup, and up to 15 Read Replicas per instance. The bottom of the screen shows a Windows taskbar with various icons and system status.

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

The screenshot shows the AWS RDS MySQL instance configuration page. The 'Storage' section is visible, showing the 'Storage type' as 'General Purpose SSD (gp2)' and the 'Allocated storage' as '20 GiB'. A note indicates that after modifying storage, the DB instance will be in storage optimization. On the right, the MySQL sidebar is present. The bottom of the screen shows a Windows taskbar with various icons and system status.

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.



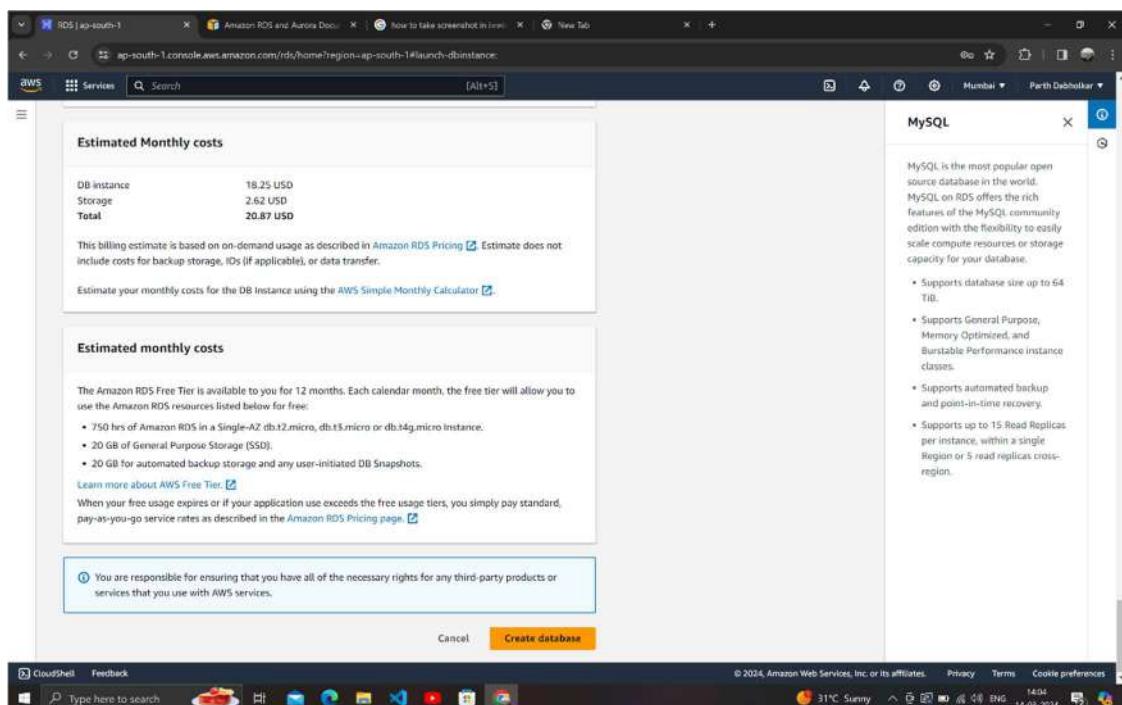
Public access Info

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.

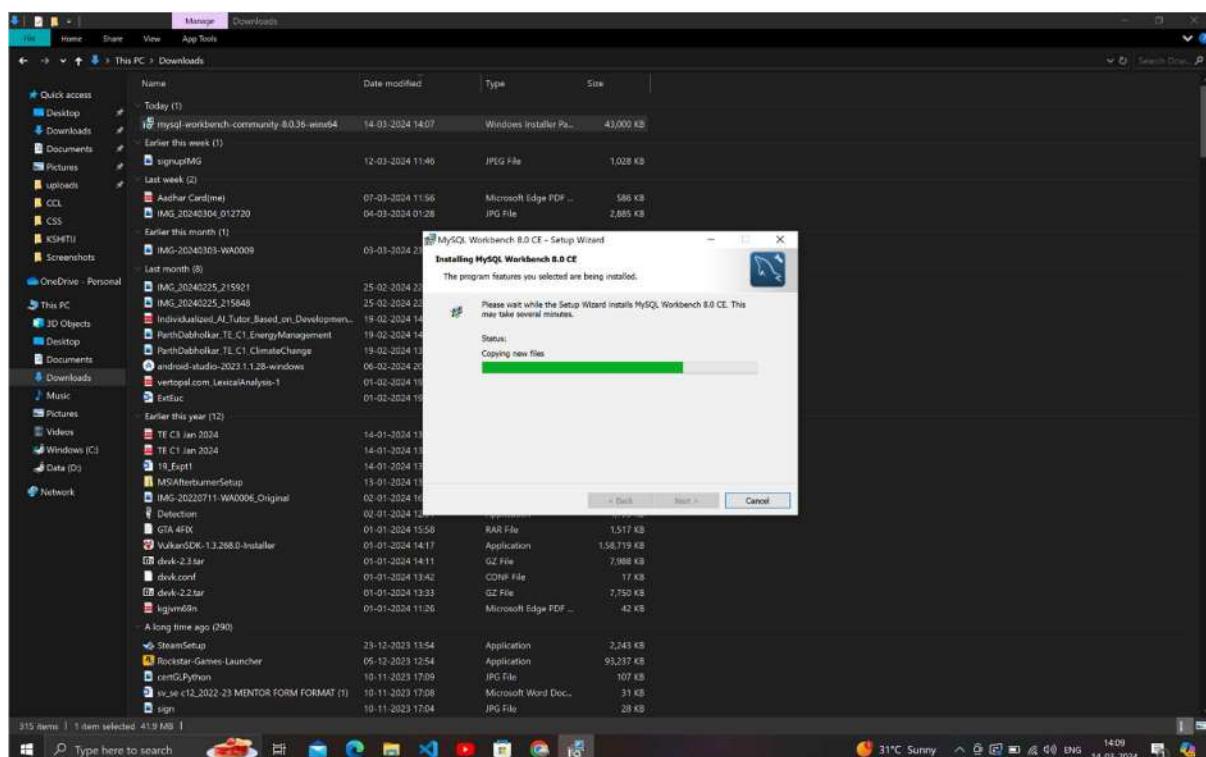
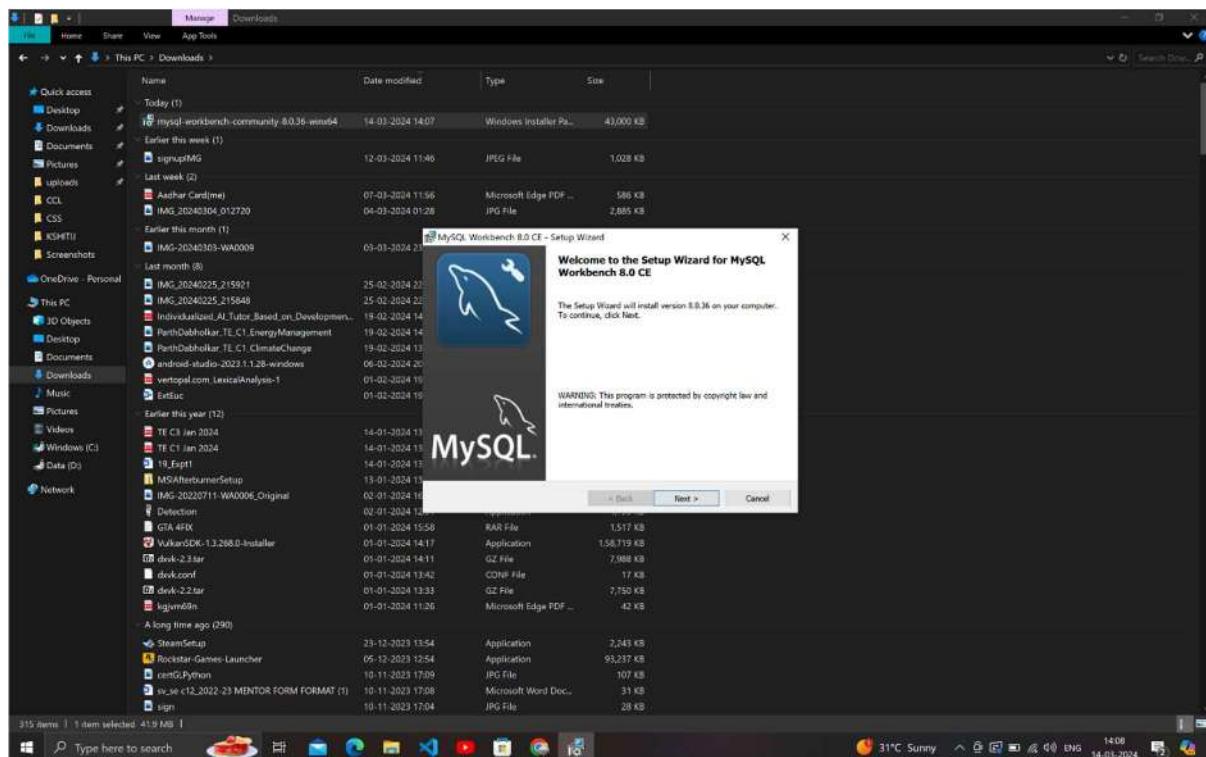


- The creation of database will take some time. Until the database is created, go to Google and search for MySQL Workbench.

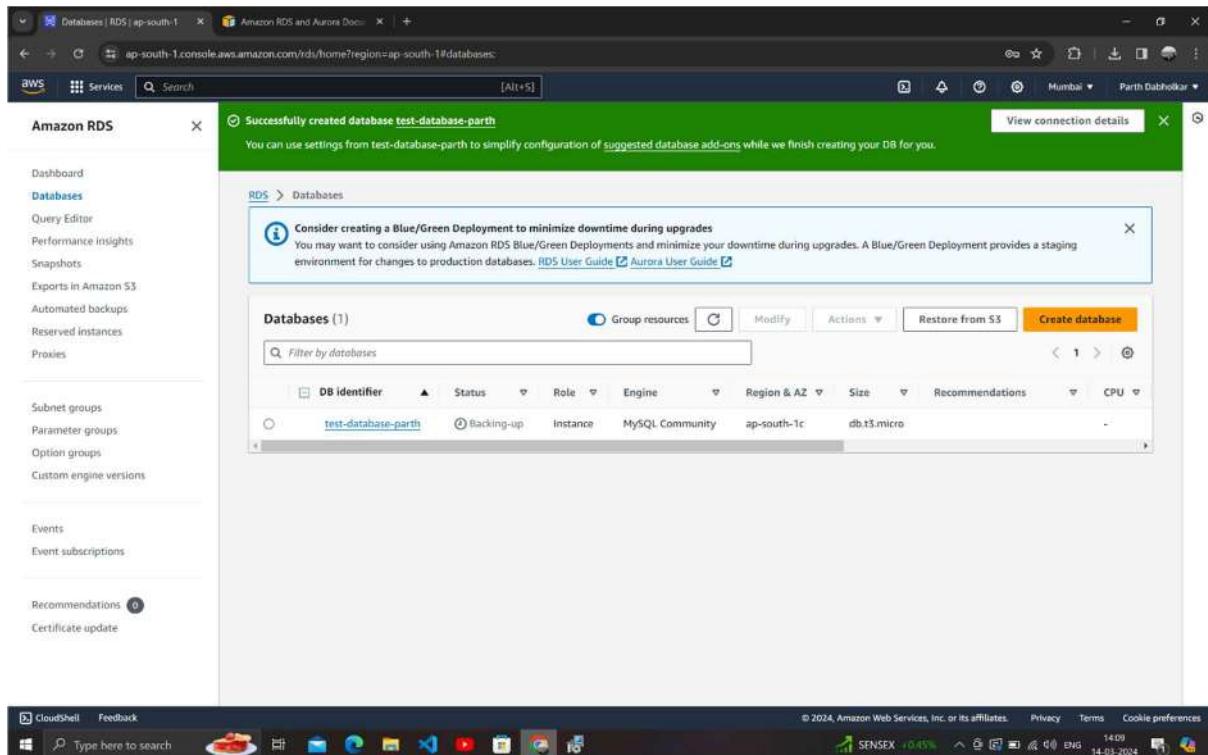
The screenshot shows the Amazon RDS console in a web browser. On the left, there's a sidebar with options like Dashboard, Databases, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, 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. It includes a message about Blue/Green Deployments. Below that is a table titled 'Databases (1)' with one entry: 'test-database-parth' (Status: Creating, Instance: MySQL Community, Engine: db.t5.micro). At the bottom right of the main area is a 'Create database' button.

The screenshot shows a Google search results page for 'mysql workbench'. The top result is a link to the MySQL website with the title 'MySQL Workbench'. The snippet describes MySQL Workbench as a unified visual tool for database architects, developers, and DBAs. It mentions features like data modeling, SQL development, and database design. To the right of the snippet, there's a box with the heading 'MySQL Workbench' and a brief description: 'MySQL Workbench is a visual database design tool that integrates SQL development, administration, database design, creation and maintenance into a single integrated development environment for the MySQL database system.' It lists programming languages (Python, C++, C, C#, Objective-C), license (GPL), operating systems (Windows, OS X and Linux), and a stable release date (8.0.22 / 2020-09-12). Below the snippet, there are links for 'People also search for' including XAMPP, DBeaver, Visual Studio Code, and MySQL.

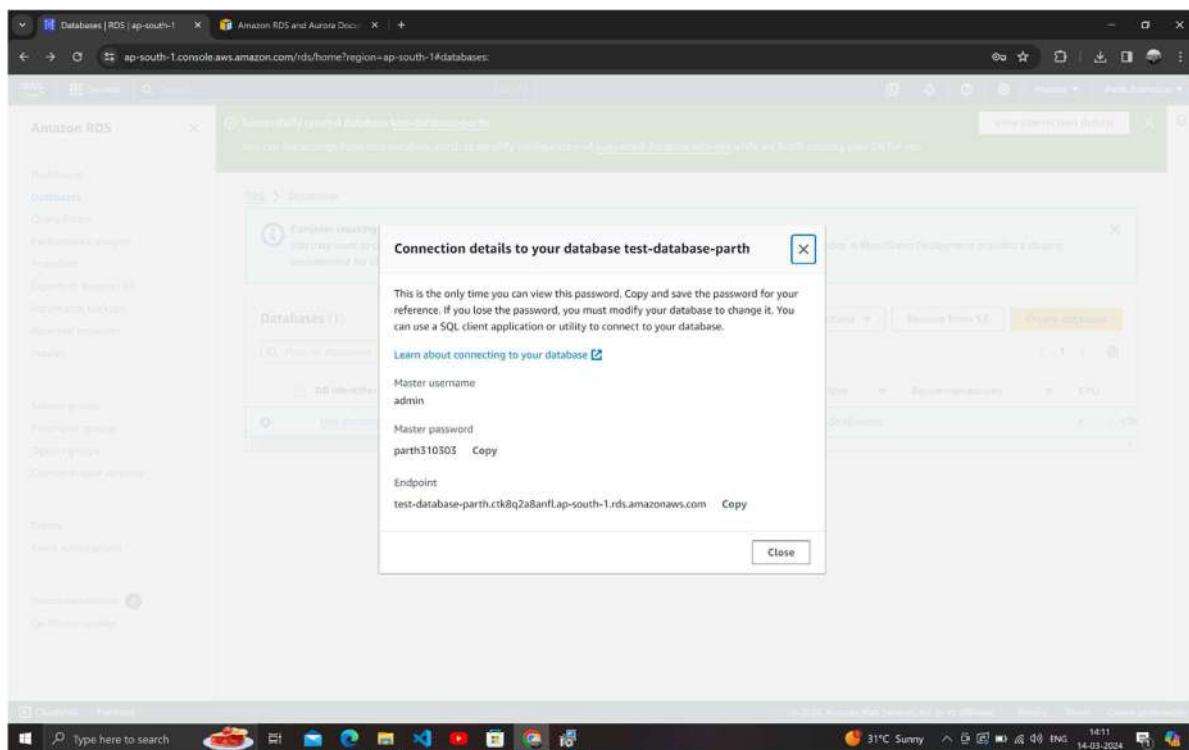
- Select the link and click download. In MySQL Community download, select the download option in the bottom section of the screen. You will be redirected to a new screen. Select “No just continue my download” and wait for the download to be completed. After installation select finish and close the setup.



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 Amazon RDS console for the 'ap-south-1' region. On the left, there's a sidebar with options like Dashboard, Databases (which is selected), Query Editor, 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 content area has tabs for Connectivity & security, Monitoring, Logs & events, Configuration, Maintenance & backups, Tags, and Recommendations. Under Connectivity & security, there are three sections: Endpoint & port, Networking, and Security. In the Endpoint & port section, the Endpoint is 'test-database-parth.ctk8q2a8anfl.ap-south-1.rds.amazonaws.com' and the Port is '3306'. In the Networking section, it shows the Availability Zone as 'ap-south-1c' and the VPC as 'vpc-00dffeb2613906df4'. In the Security section, it lists the VPC security groups as 'default (sg-0b25ef9c0ff06611d)' with status 'Active', and the Publicly accessible setting as 'Yes'. The Certificate authority is 'rds-ca-rsa2048-g1'. The Certificate authority date is 'May 20, 2021, 00:10 (UTC+05:30)' and the DB instance certificate expiration date is 'March 14, 2025, 14:07 (UTC+05:30)'. The Network type is 'IPv4'. Below this, there's a section for Connected compute resources with a count of 0.

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 File, Edit, View, Database, Tools, Scripting, and Help. The main area has a title 'Welcome to MySQL Workbench'. Below it, there's a 'MySQL Connections' section with a '+' icon. A 'Setup New Connection' dialog box is open in the center. The dialog has a title 'MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design...'. It asks for a 'Connection Name' (set to 'test-database-parth'), a 'Connection Method' (set to 'Standard (TCP/IP)'), and a 'Default Schema' (empty). Under 'Parameters', it shows 'Hostname' as 'test-database-parth.ctk8q2a8anfl.ap-south-1.rds.amazonaws.com', 'Port' as '3306', 'Username' as 'admin', and 'Password' as 'Store in Vault...'. There are 'Test Connection' and 'OK' buttons at the bottom. The status bar at the bottom shows 'Nifty midcap +1.53%' and the date '14-03-2024'.

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.

Name	Security group ID	Security group name	VPC ID	Description
-	sg-0b25ef9c0ff06611d	default	vpc-00ddffeb2613906df4	default VPC security group

Inbound rules (1)

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-0cc34ec00db5c23d5	-	All traffic	All	All

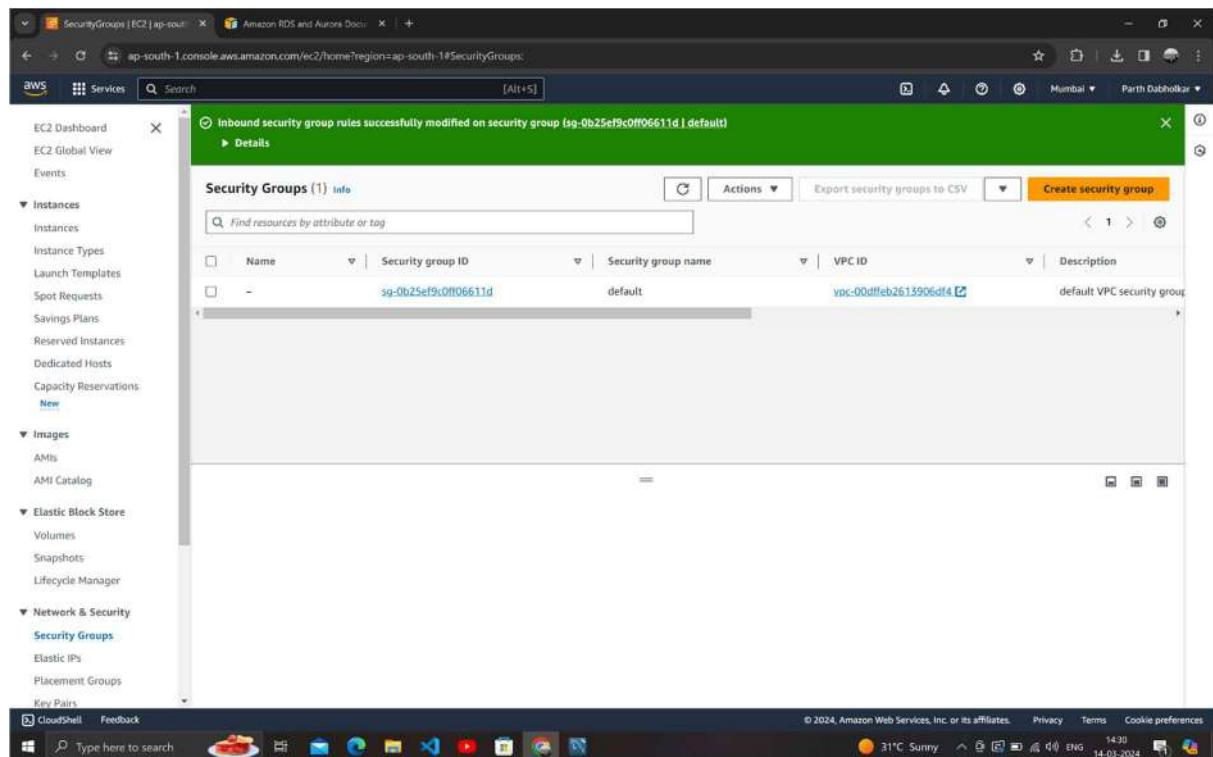
15. Select add rule. In the type field, select IPV4 – All traffic in order to allow anyone to access the database. In the source field select “Anywhere” for all ip addresses. Click Save Rules.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0cc34ec00db5c23d5	All traffic	All	All	Custom	sg-0b25ef9c0ff06611d
-	All traffic	All	All	Anywh...	0.0.0.0/0

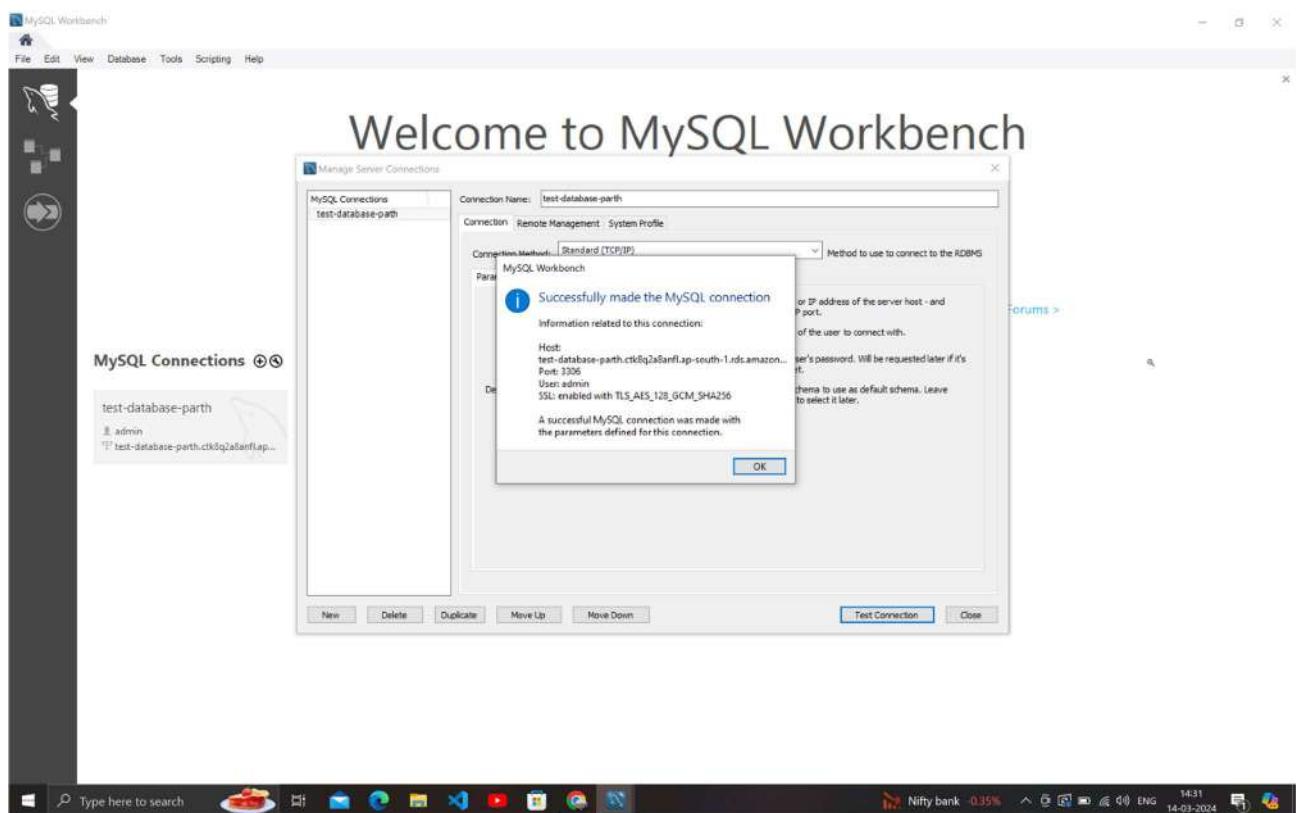
Add rule

⚠️ 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.

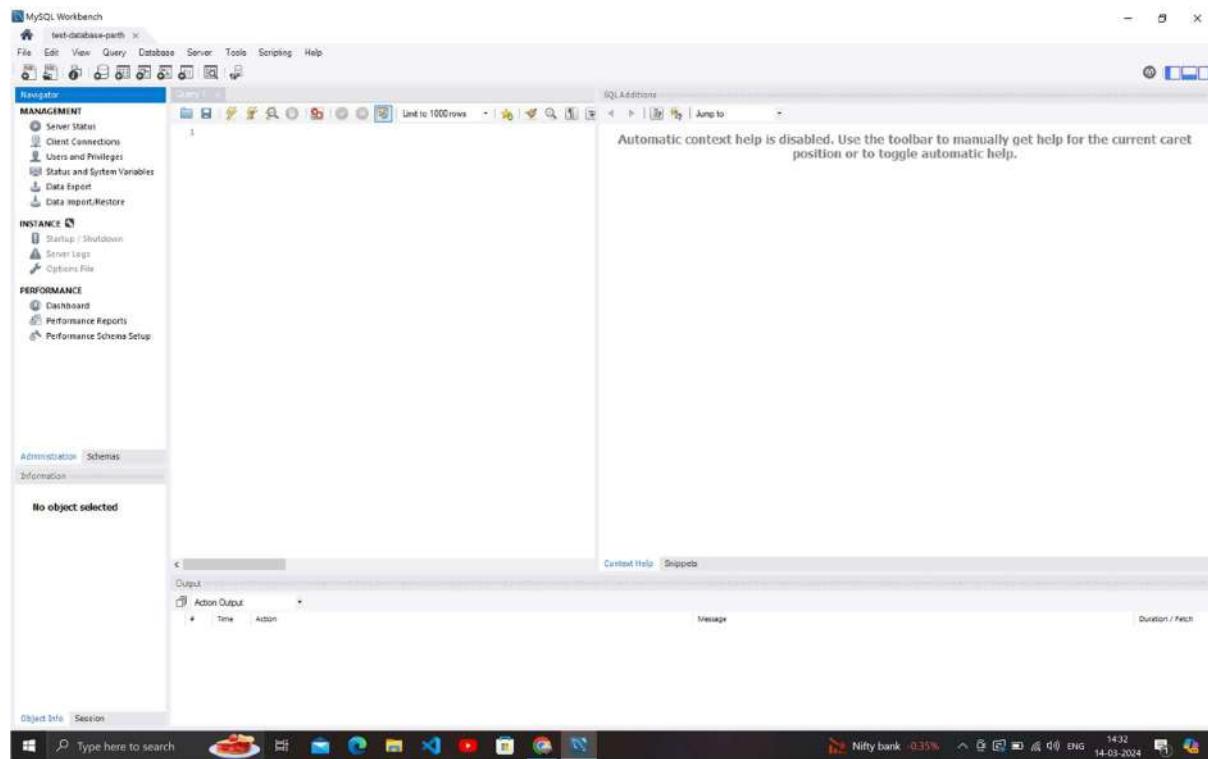
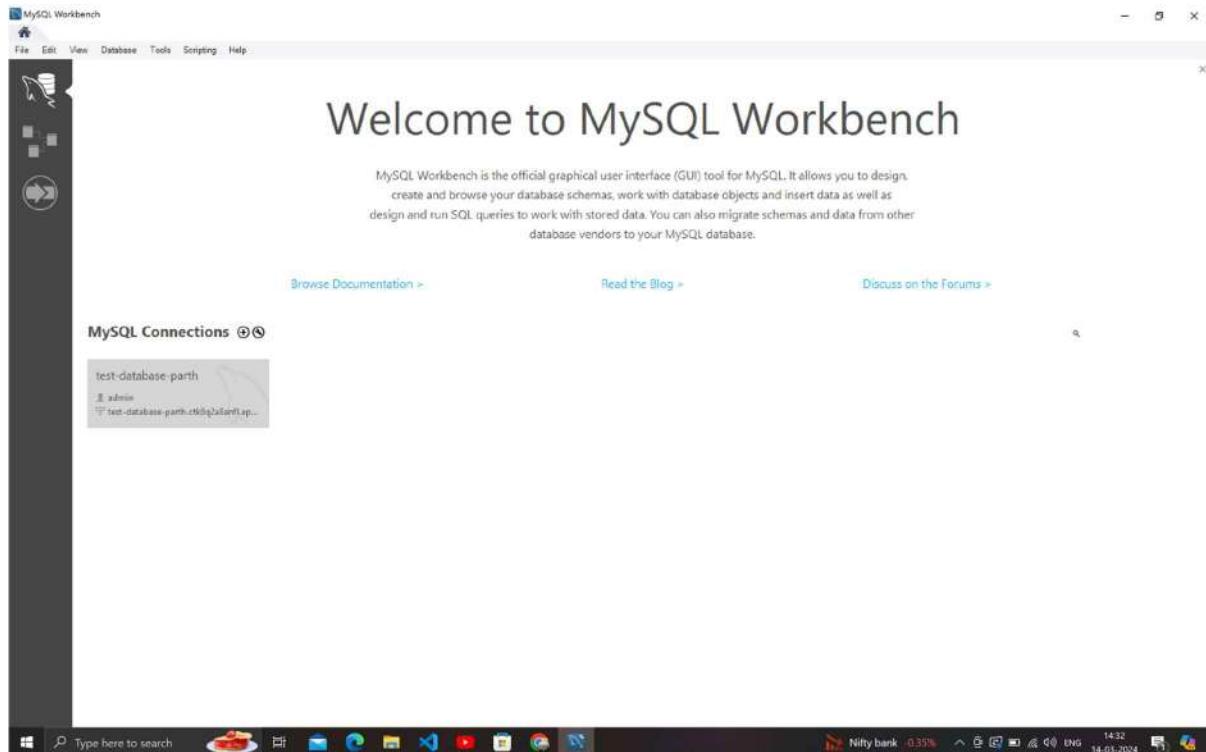
Cancel Preview changes Save rules



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.

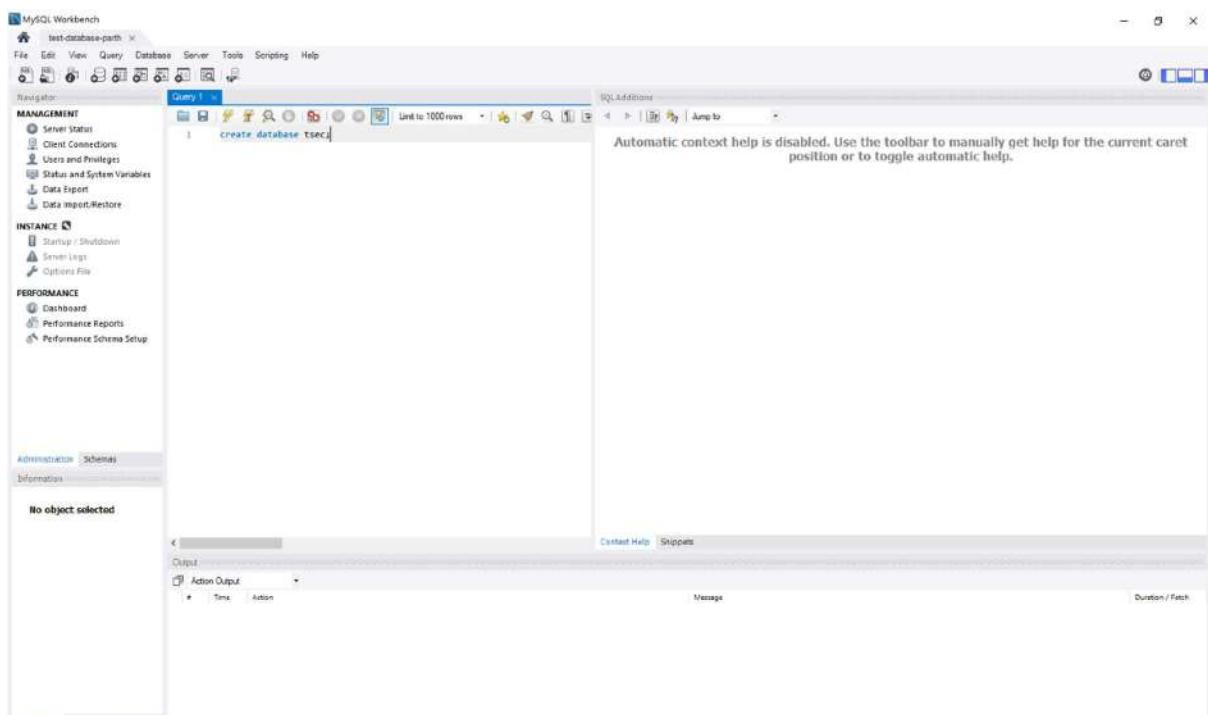


17. Select the database in the dashboard. A new editor instance will be created as follows.

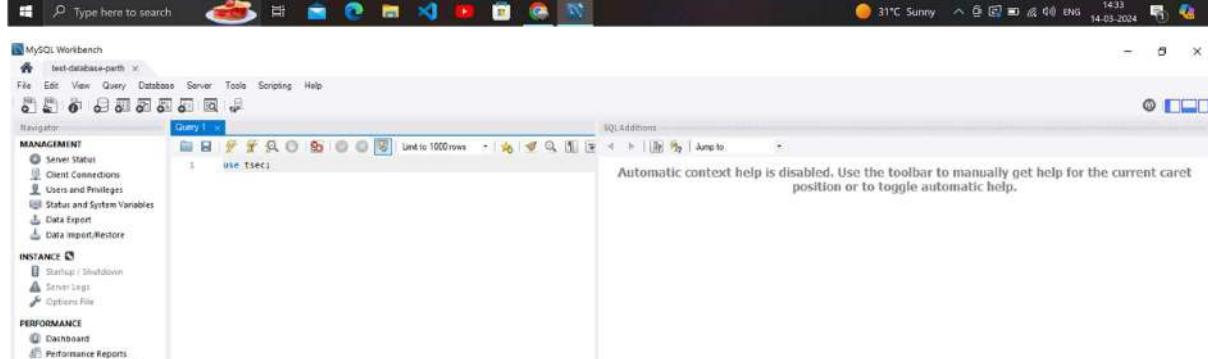


18. Write the following query in the editor:

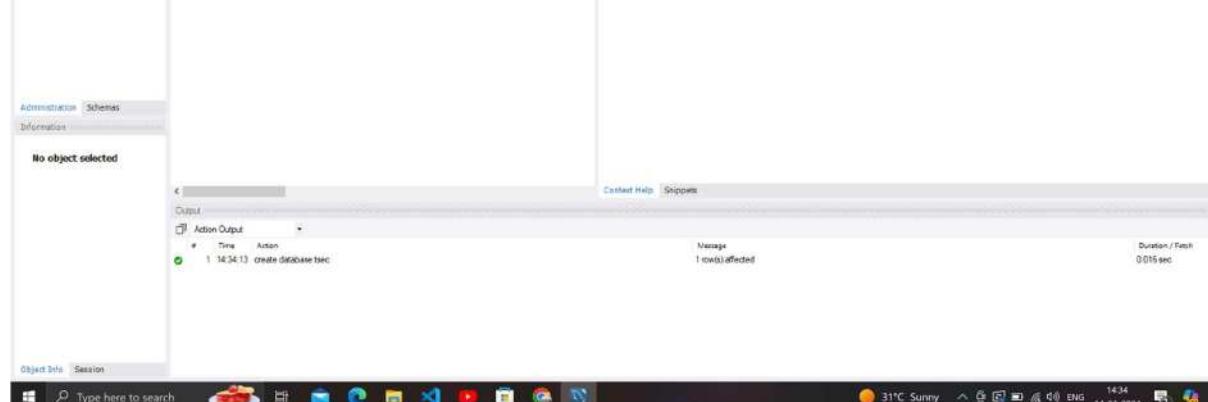
- Create database tsec;
- Use tsec;
- Create table students(roll int, name varchar(10), city varchar(10));
- Insert into students values(32, 'Parth Dabholkar', 'Thane');
- Insert into students values(34, 'Vedant Devkar', 'Mumbai');
- Desc students;
- Select * from students;



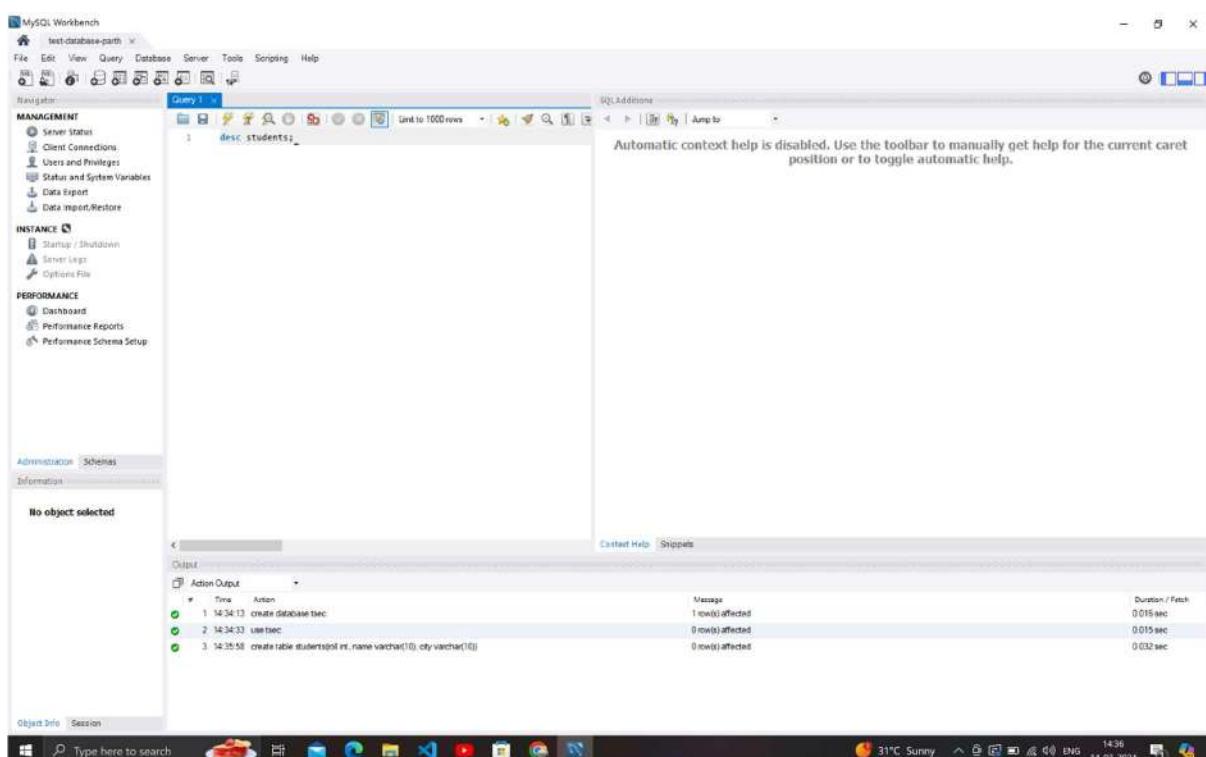
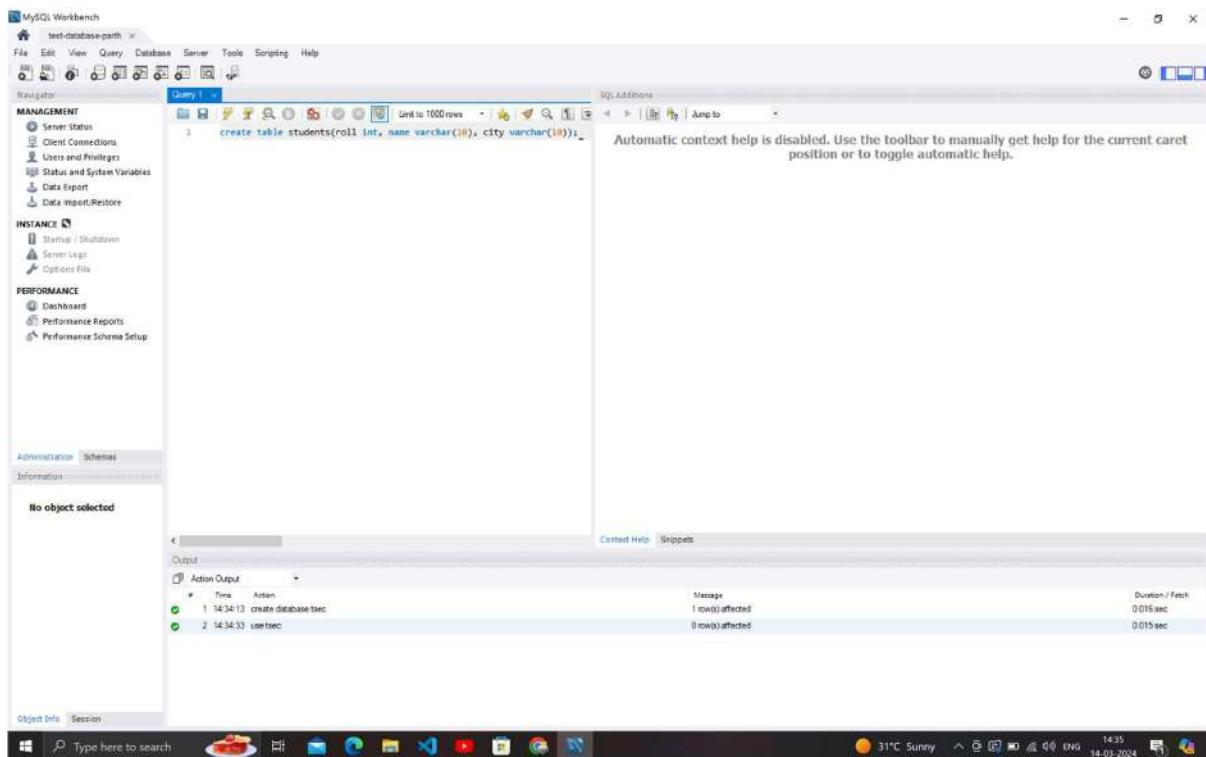
The screenshot shows the MySQL Workbench interface. In the top-left pane, under the 'MANAGEMENT' section, the 'Database' tab is selected. A query window titled 'Query 1' contains the command: 'create_database tsec;'. The status bar at the bottom right indicates '1 row(s) affected'.



The screenshot shows the MySQL Workbench interface. In the top-left pane, under the 'MANAGEMENT' section, the 'Database' tab is selected. A query window titled 'Query 1' contains the command: 'use tsec;'. The status bar at the bottom right indicates '1 row(s) affected'.



The screenshot shows the MySQL Workbench interface. In the top-left pane, under the 'MANAGEMENT' section, the 'Database' tab is selected. A query window titled 'Query 1' contains the command: 'create table students(roll int, name varchar(10), city varchar(10));'. The status bar at the bottom right indicates '1 row(s) affected'.



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

```

CREATE DATABASE test-database-path;
USE test-database-path;
CREATE TABLE students (
    roll INT,
    name VARCHAR(10),
    city VARCHAR(10)
);
DESC students;

```

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

Field	Type	Null	Key	Default	Extra
roll	int	YES			
name	varchar(10)	YES			
city	varchar(10)	YES			

The 'Action Output' pane shows the results of the executed statements:

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

```

INSERT INTO students values(1, "Parth Dabholkar", "Thane");
INSERT INTO students values(2, "Vedant Devkar", "Mumbai");

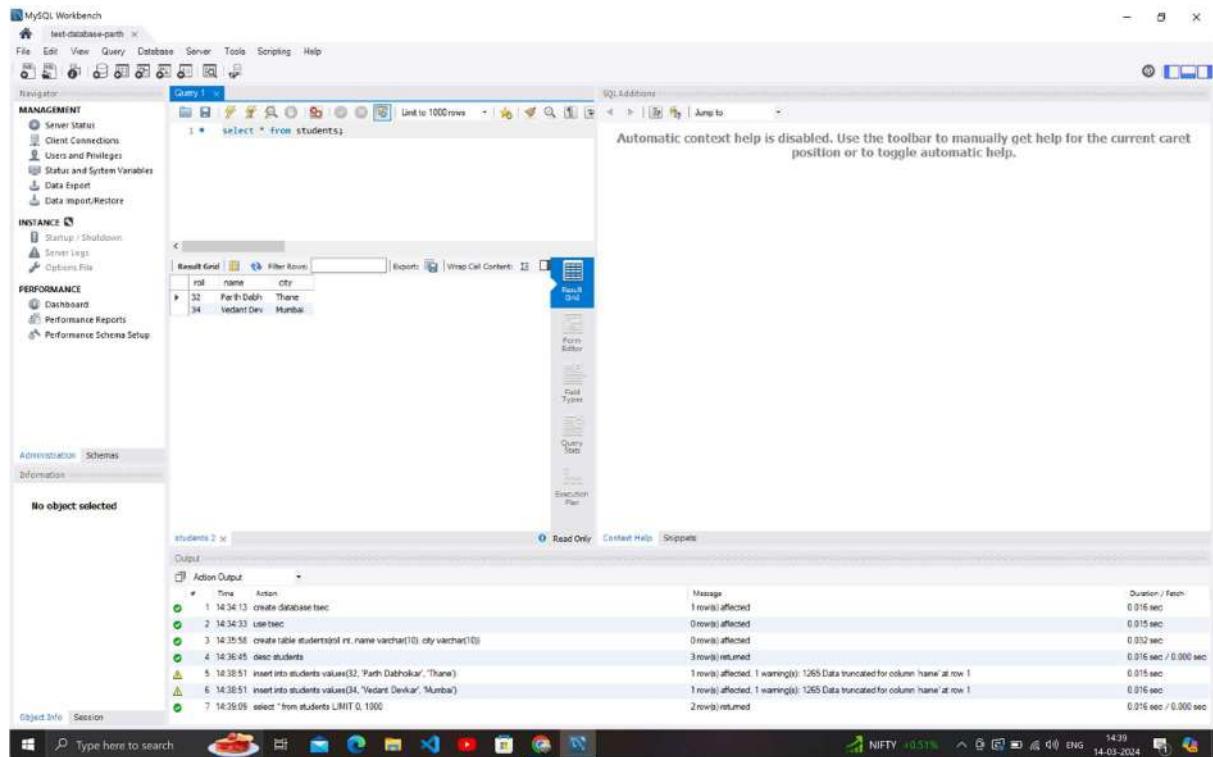
```

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

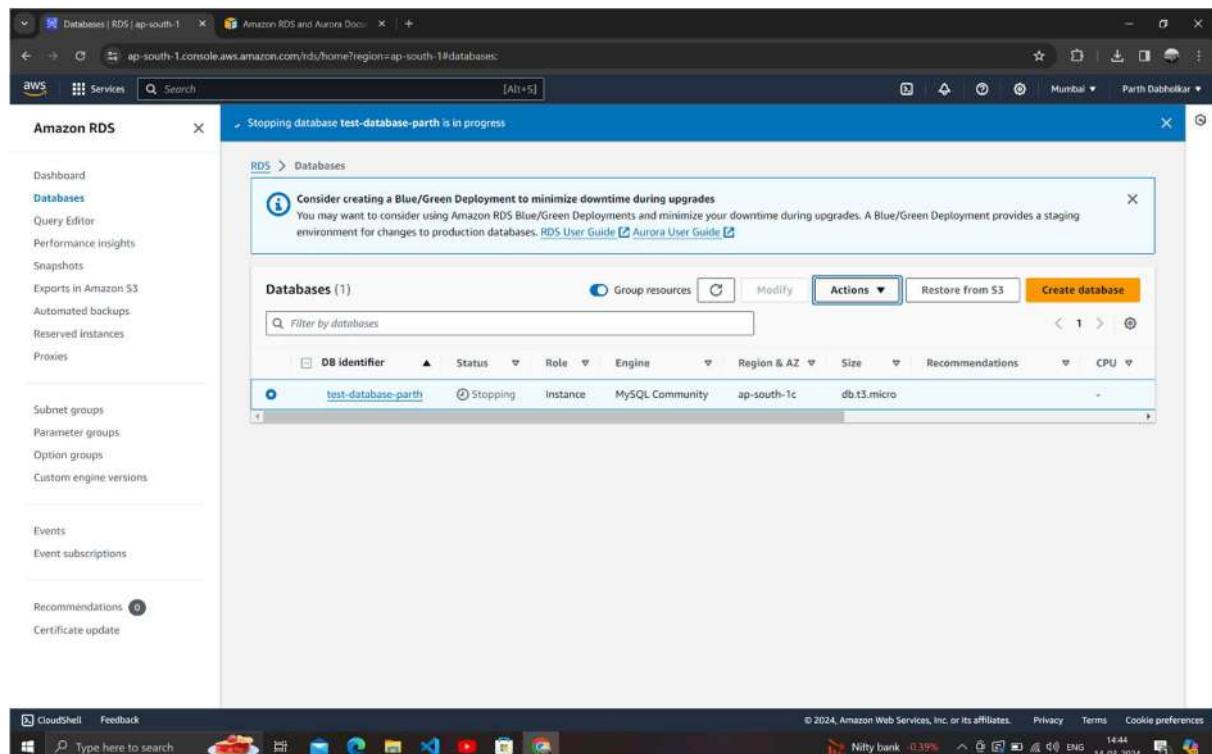
Field	Type	Null	Key	Default	Extra
roll	int	YES			
name	varchar(10)	YES			
city	varchar(10)	YES			

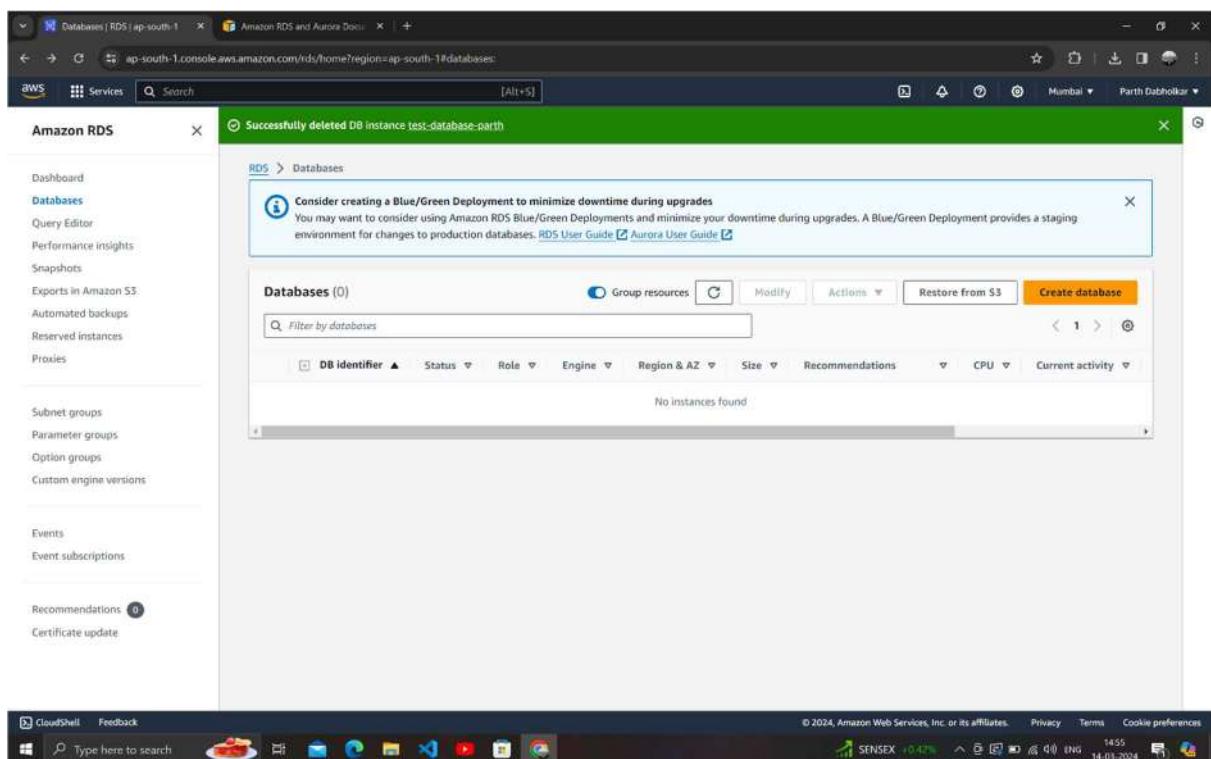
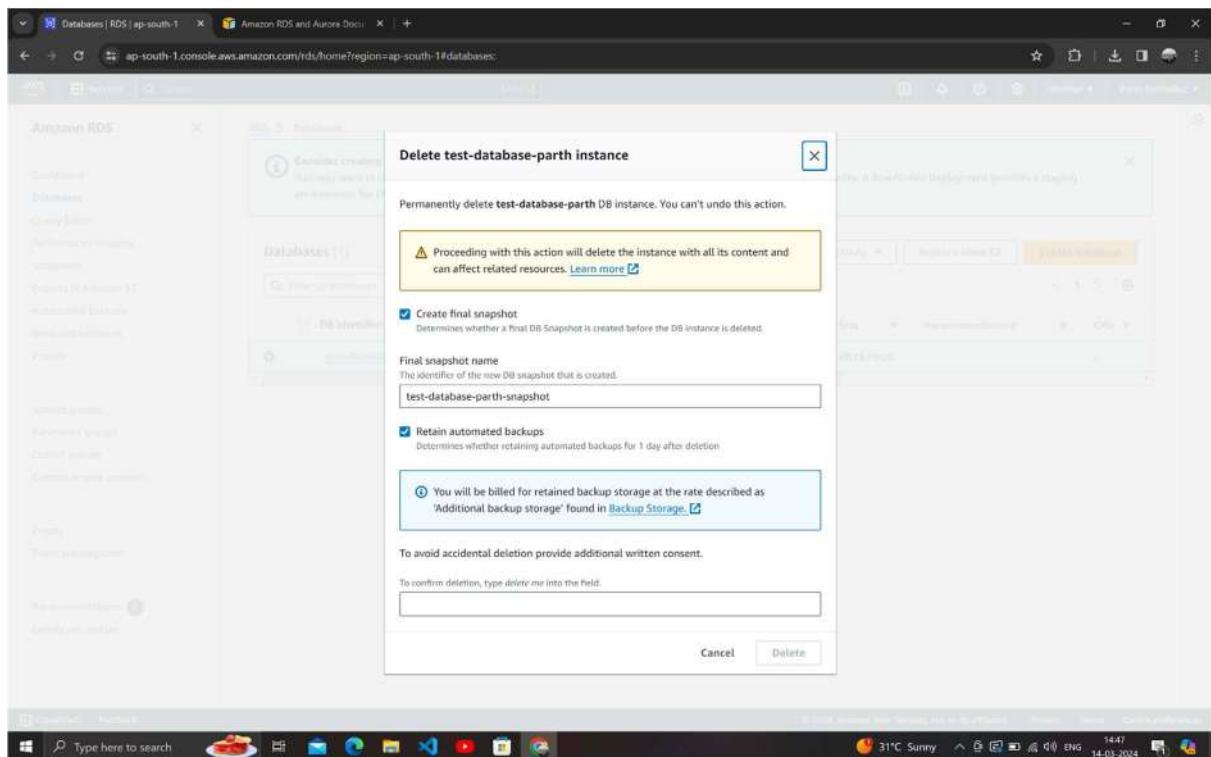
The 'Action Output' pane shows the results of the executed statements:

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



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.



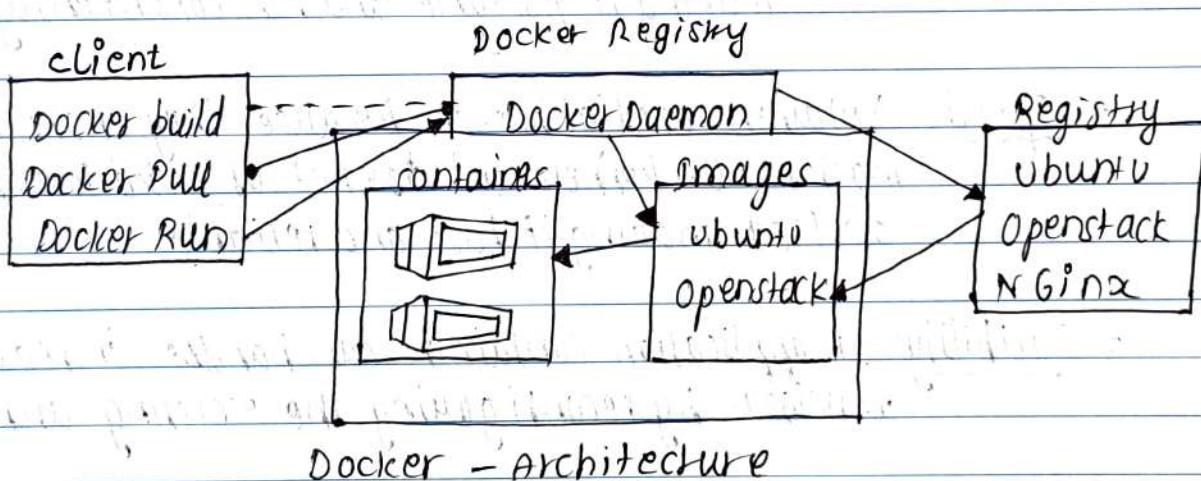


EXPERIMENT NO: 9

AIM: to study and implement Containerization using Docker.

Docker and its Architecture:

Docker is an open source software platform used to create, deploy and manage virtualized application containers on a common operating system (OS), with an ecosystem of allied tools. Docker gives software developers a faster and more efficient way to build and test containerized portions of an overall software applications. Each container contains all elements needed to build a software component and ensure it's built, tested and deployed smoothly. Docker enables portability for when these packaged containers are moved to different servers or environments.



* Docker Client: With the help of Docker client, the docker can interact with the users. The Docker command uses the Docker API. The Docker client can communicate with multiple Docker Daemons.

- * Docker Host: A Docker host is a type of machine that is responsible for running more than one container. It comprises of the Docker Daemon, Images, Containers, Networks and storage.
- * Docker Registry: All the Docker Images are stored in the Docker registry. There is a public registry which is known as Docker Hub that can be used by anyone. One can also run a private registry also.
- Benefits of Containerization:
 1. Portability: An application container creates an executable software package abstracted away from the Host OS. Hence, it is not dependent on Host OS and thus making it portable and run consistently.
 2. Speed: Containers are lightweight because they share host machines OS kernels and do not subjected to extra overload making it fast and reliable.
 3. Scalability: An application container can handle increasing workloads by reconfiguring the existing architecture.
 4. Agility: Developers using DevOps tools and processes for rapid app development and enhancement.
 5. Efficiency: Since the software running in containerized environments shares the host machine's OS kernel, developers can share the app layers across containers.

- Docker container: Containers are created from docker images as they are ready applications. With the help of Docker API or CLI, we can start, stop or delete, or move a container. A container can access only those resources which are defined in the image unless additional access is defined during the building of an image in the container.
- Docker Images: An image contains instructions for creating a docker container. It is just a read only template. It is used to store and ship applications. Images are an important part of the docker experience as they enable collaboration between developers in any way which is not possible earlier.
- Dockerfile: The Dockerfile uses DSL (Domain specific Language) and contains instructions for generating a Docker Image. Dockerfile will define the processes to quickly produce an image. While creating your application, you should create a Dockerfile in order since the Docker daemon runs all the instructions from top to bottom.

Aspect	Containers	Virtual Machines
Technology	Docker uses containerization technology.	Virtual Machines run on hypervisor technology.
Isolation	Lightweight, share OS Kernel.	Full isolation, each VM has its own OS.

Resource Usage	More efficient, shares OS resources.	Less efficient, requires more resources.
Startup time	Faster startup time	Slower startup time.
Deployment	Faster deployment	Slower deployment.
Portability	High portable across environments	Less portable due to dependencies.
Aspect	Images	Containers
Definition	A read-only template used to create containers.	An instance of an image that is running as a process.
Nature	static, immutable files.	Dynamic, mutable instances.
Purpose	Used to create containers.	Running instance of an image.
Content	Includes the application code, runtime libraries and dependencies.	Contains the running application processes along with dependencies.
Lifecycle	Exists before the container is created.	Created from an image and runs until stopped or deleted.

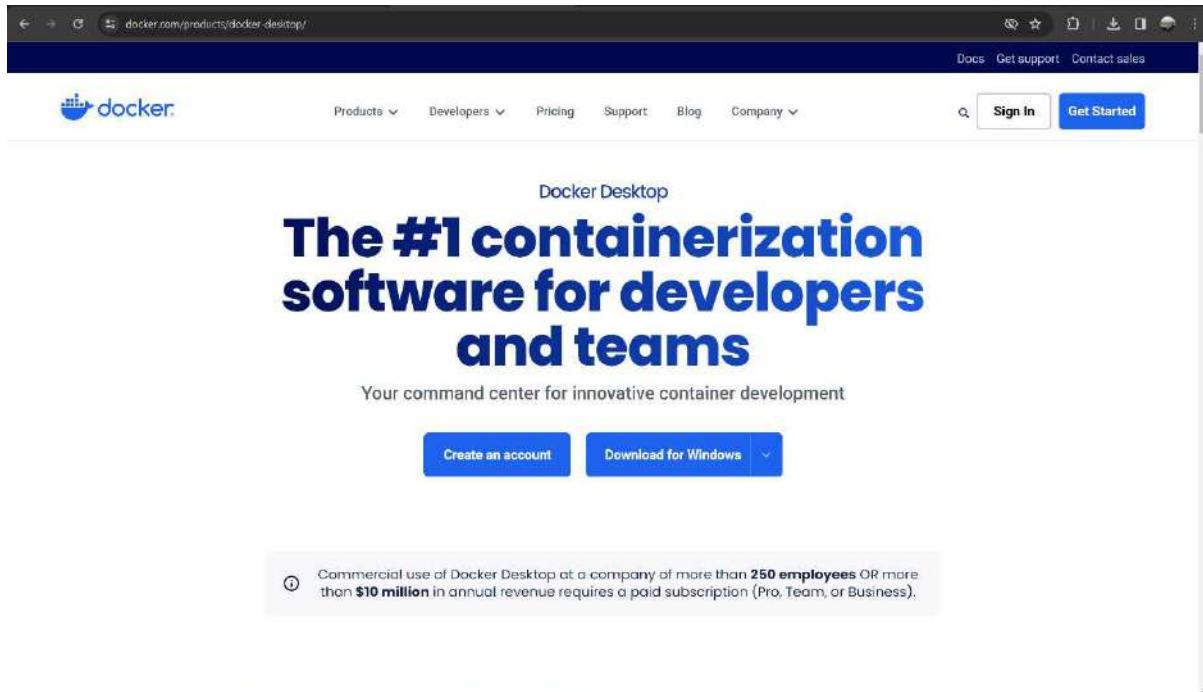
(A)

SP 27/3/2024

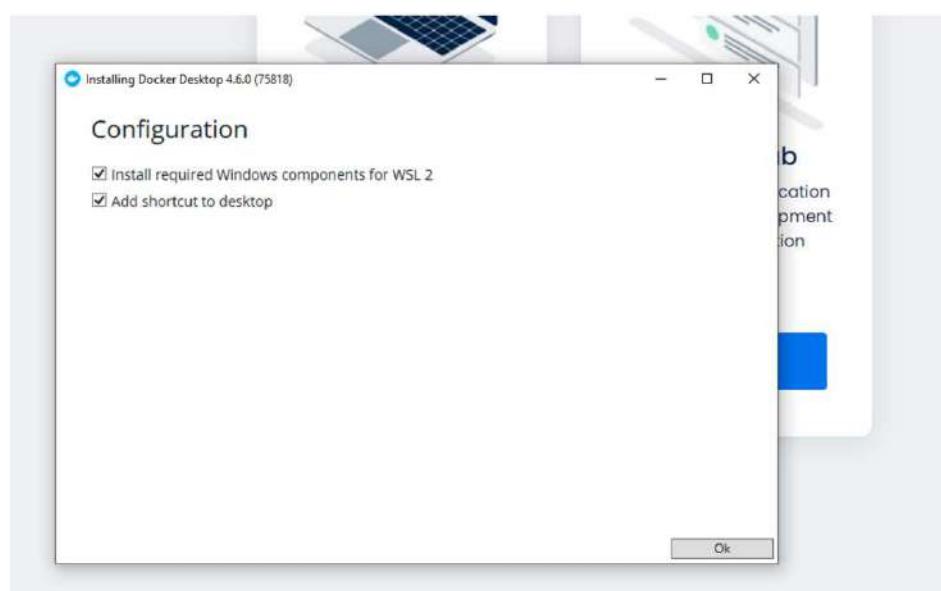
EXPERIMENT NO: 9

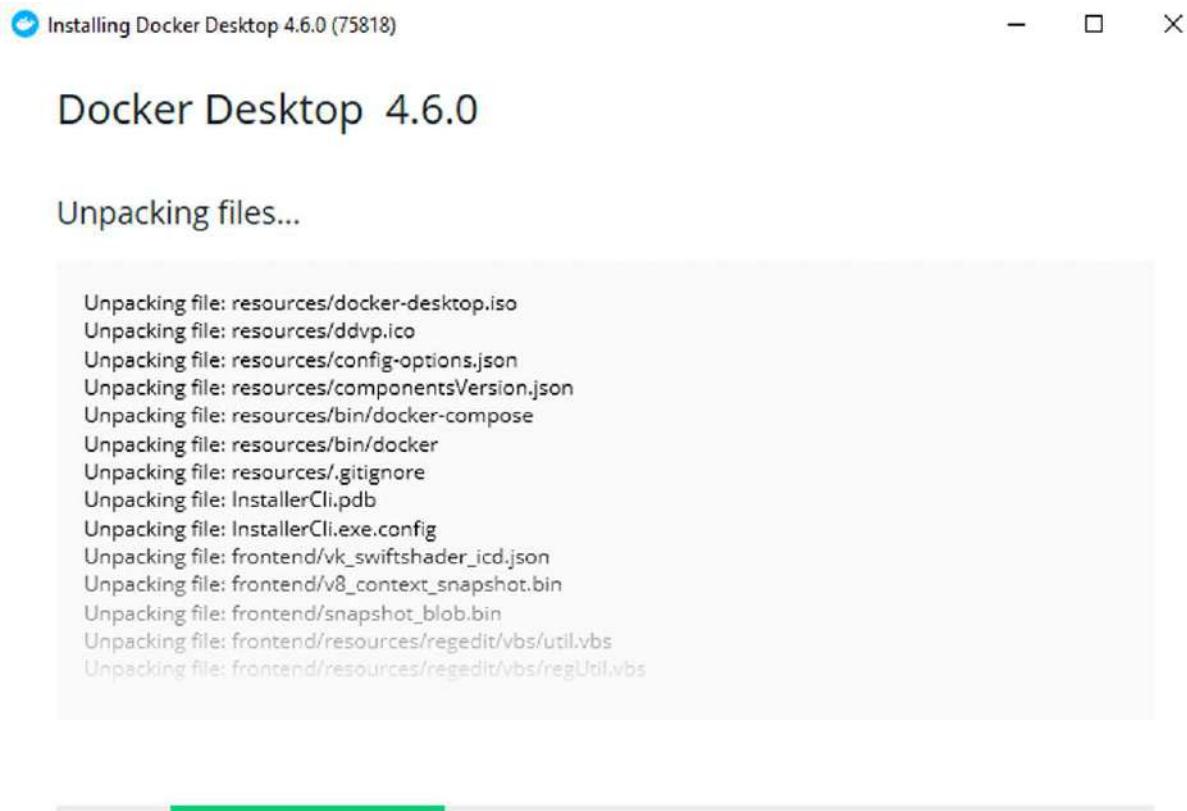
AIM: To study and implement containerization using Docker.

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.

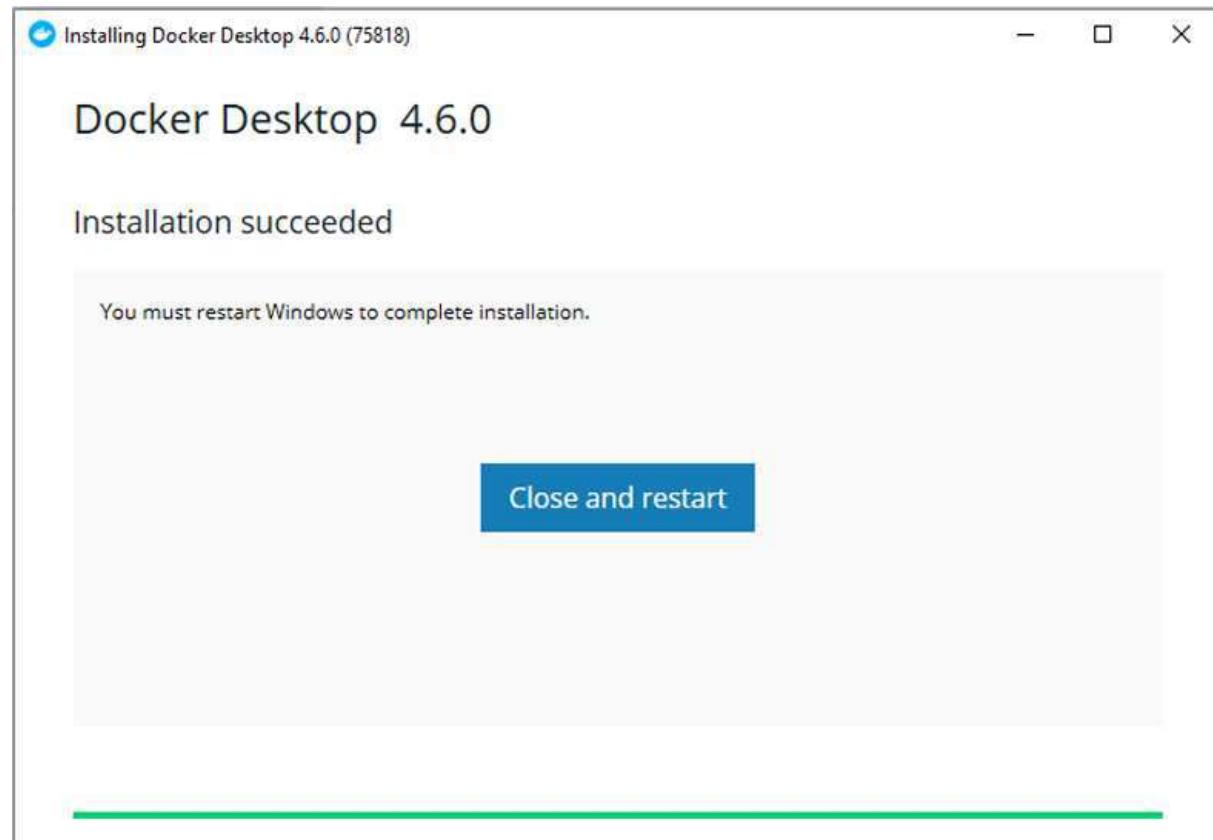


Step-2: Once the setup is downloaded from the website, open the setup wizard on your device. Start the installation process for Docker Desktop.

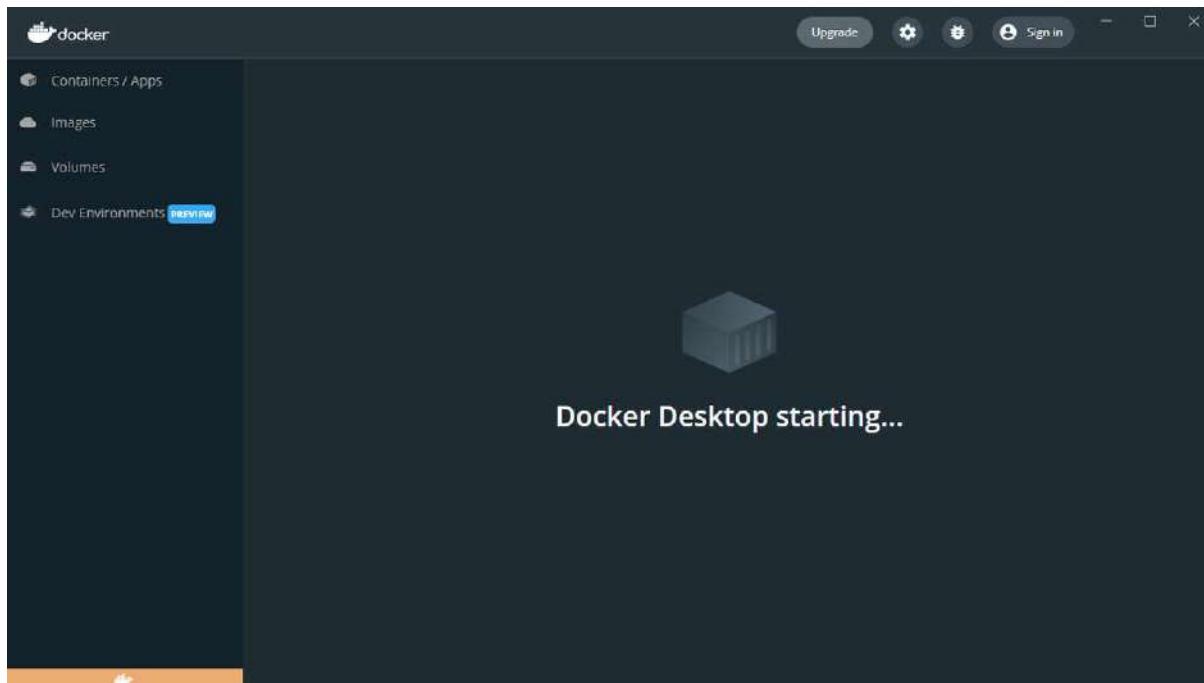
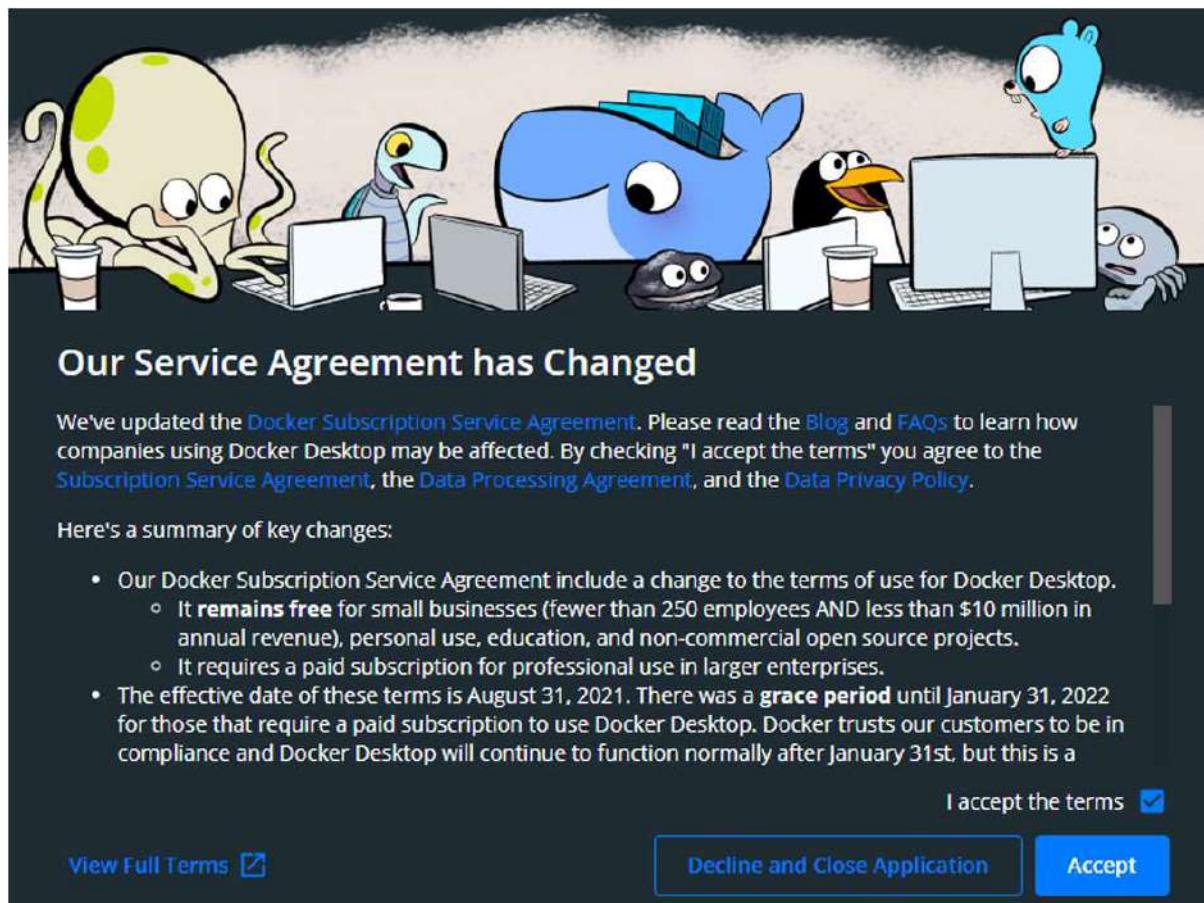




Step-3: After the installation is completed, click on finish the setup. To start the Docker Dashboard, restart your computer device.



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:

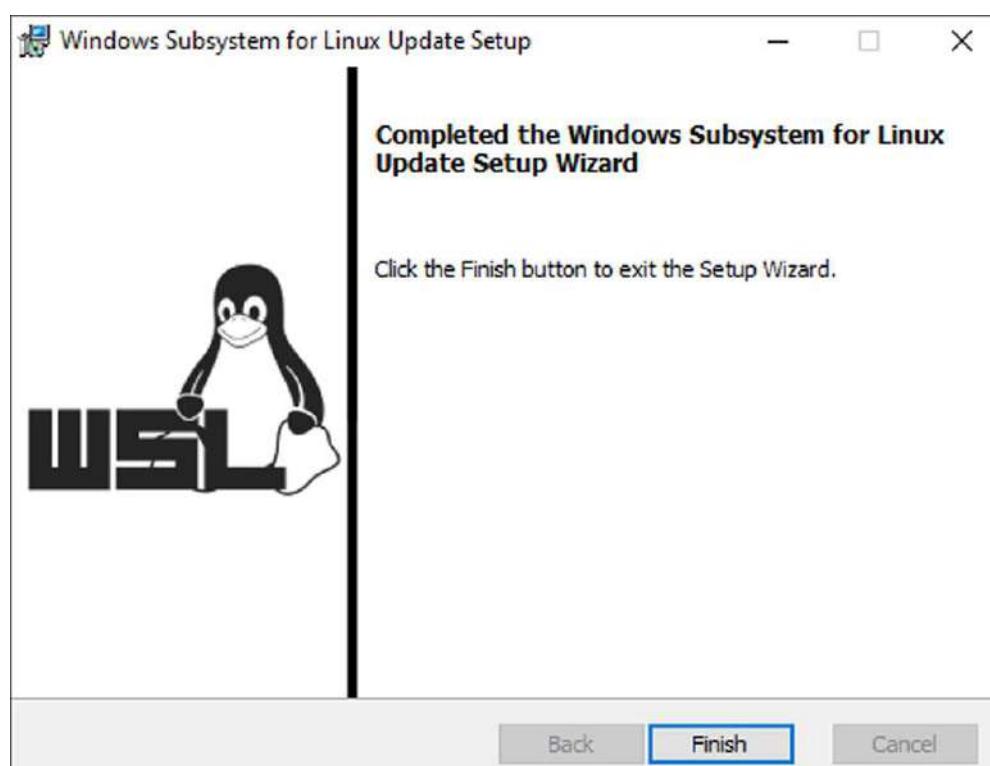
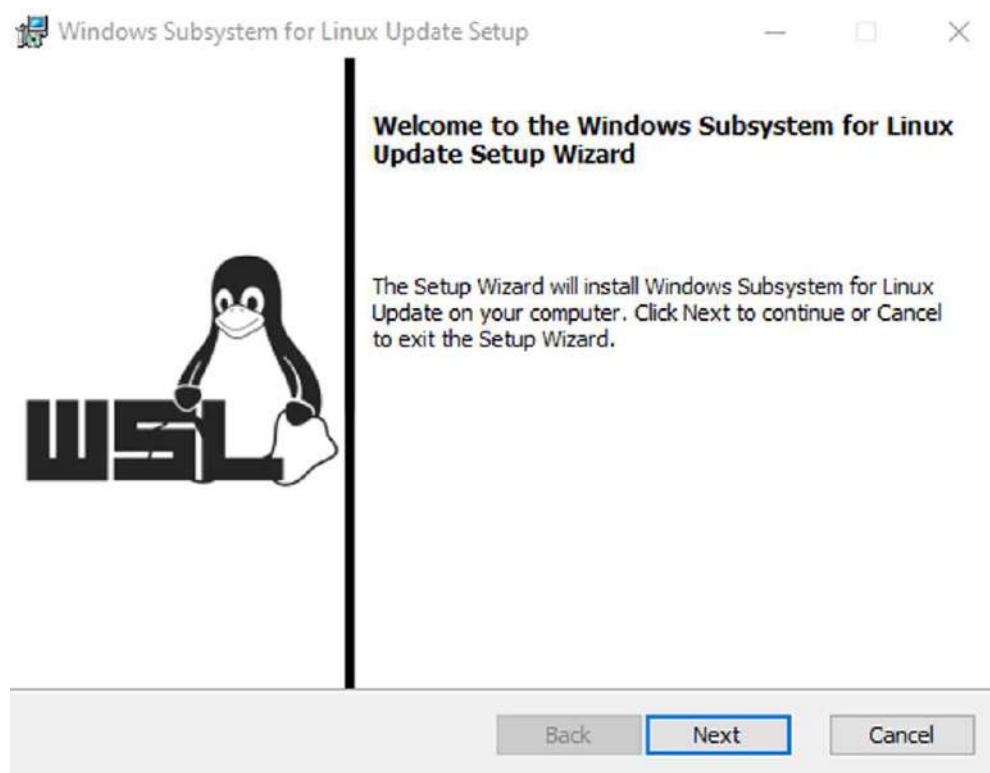
- [WSL2 Linux kernel update package for x64 machines](#)

ⓘ 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.



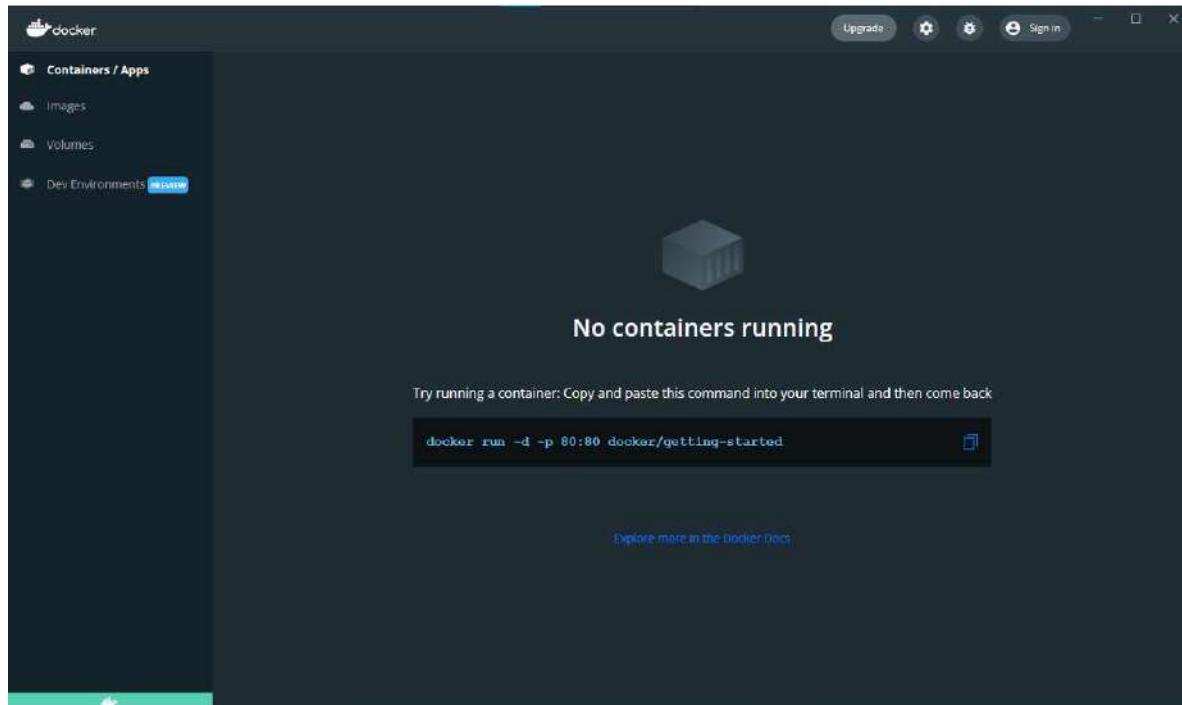
Step-7: Open Windows Powershell as an Administrator.

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

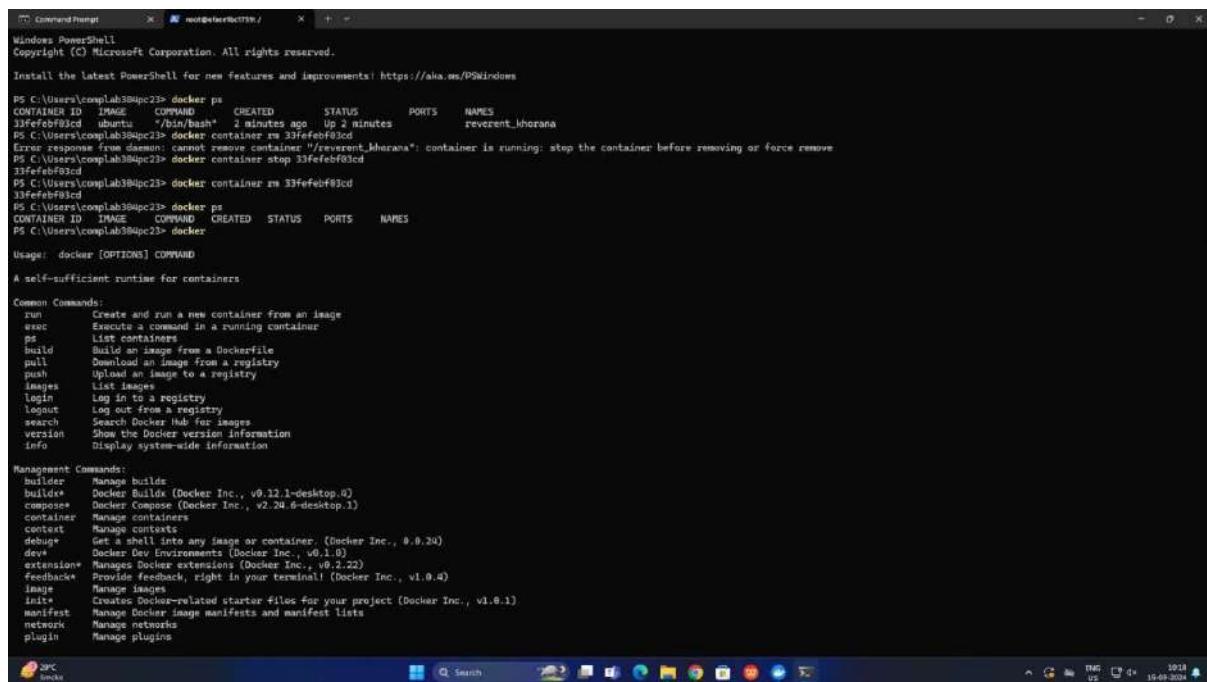
- docker ps - docker //list all docker commands

- 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\complab30\pc> 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\complab30\pc> 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\complab30\pc> docker containers stop 33fefebf03cd
33fefebf03cd
PS C:\Users\complab30\pc> docker container rm 33fefebf03cd
33fefebf03cd
PS C:\Users\complab30\pc> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
PS C:\Users\complab30\pc> 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)
dns     Docker DNS (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., v1.5.9)
system*    Manage Docker
trust*     Manage trust on Docker images
volume*   Manage volumes

Smart 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 Docker's HTTP API endpoint (default "complab304pc23")
  -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 --tlsv1.2
  --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 ca2b9f26964c 2 weeks ago 77.9MB
PS C:\Users\complab304pc23> docker image rm ca2b9f26964c
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:77996da8b66055c12213b97899eb3277e7394c0f4fb8082389e021404ff1f17e
Untagged: ubuntu@sha256:77996da8b66055c12213b97899eb3277e7394c0f4fb8082389e021404ff1f17e
Deleted: sha256:54909e8c2f059a675eF193e59617db37e2a95cccf22a72de13c3b3a61d7591
PS C:\Users\complab304pc23> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
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: sha256:77996da8b66055c12213b97899eb3277e7394c0f4fb8082389e421f44eff17e
Status: Downloaded newer image for ubuntu:latest
root@eface1bc1798: ~ |
```

```

Command Prompt  X  root@eface1bc1798: ~ + =
--tls           Use TLS; implied by --tlsv1.2
--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 ca2b9f26964c 2 weeks ago 77.9MB
PS C:\Users\complab304pc23> docker image rm ca2b9f26964c
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:77996da8b66055c12213b97899eb3277e7394c0f4fb8082389e021404ff1f17e
Untagged: ubuntu@sha256:77996da8b66055c12213b97899eb3277e7394c0f4fb8082389e021404ff1f17e
Deleted: sha256:54909e8c2f059a675eF193e59617db37e2a95cccf22a72de13c3b3a61d7591
PS C:\Users\complab304pc23> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
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: sha256:77996da8b66055c12213b97899eb3277e7394c0f4fb8082389e421f44eff17e
Status: Downloaded newer image for ubuntu:latest
root@eface1bc1798: ~ |
```

EXPERIMENT NO:10

AIM: to study and implement container orchestration using Kubernetes.

THEORY:

- Need for container orchestration tool:
Containerized applications offer numerous benefits, including portability, scalability and consistency. However, managing these applications at scale can be challenging without proper orchestration. Container Orchestration tools address their challenges by automating various aspects of container management, including deployment, scaling, load balancing, service discovery and health monitoring.
- Kubernetes:
Kubernetes is an open source container orchestration platform developed by Google. It provides a robust framework for automating the deployment, scaling and managing of containerized applications. Kubernetes abstracts away the complexity of managing containers at scale, allowing users to focus on developing and destroying, deploying applications without worrying about the underlying infrastructure. Some key features of Kubernetes include automatic scaling, self-healing, service discovery, rolling updates and declarative configuration management.

- Kubernetes Components:

Kubernetes architecture consists of several components that work together to manage containerized workloads:

Master Node: Oversees the entire cluster, scheduling tasks and keeping everything in order.

Worker Nodes: Where the actual containers run, following orders from the masters.

Pods: tightly knit groups of containers sharing resources on a single node. Think of them as a small farm for your cows.

- Comparison between Pods and Nodes:

Pod: A pod is the smallest deployable unit in Kubernetes, representing one or more containers that share the same network namespace and storage volume. Pods are scheduled and run on nodes within the Kubernetes cluster.

Node: A node is a physical or virtual machine that forms the part of the Kubernetes cluster. Nodes host pods and run the Kubernetes components necessary for cluster management, such as Kubelet, kube-proxy and container runtime.

- Comparison of Kubernetes with Docker Swarm

Kubernetes: More popular and scalable, but steeper learning curve.

Docker Swarm: Simple to use for smaller deployments.

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.

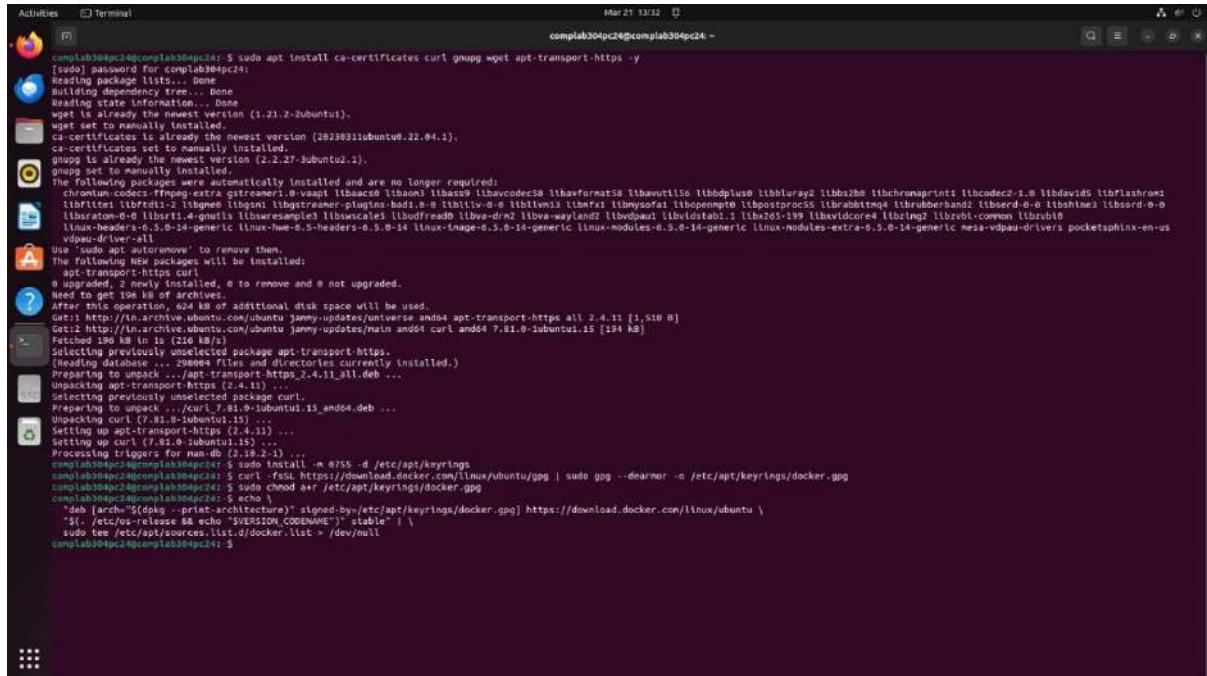
```
$ sudo apt update
```

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

```
Activities Terminal Mar 21 13:19
complab304pc24@complab304pc24: ~ $ sudo apt update
[sudo] password for complab304pc24:
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
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [139 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [116 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 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 [1,392 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy-security/main i386 Packages [461 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu jammy-security/main amd64 Packages [269 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [1,019 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [1,071 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,859 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 [1,049 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu jammy-backports/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 amd64 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://in.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/main Translation-en [1,049 kB]
Get:27 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [1,049 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)
Building dependency tree... Done
Reading state information... Done
196 packages can be upgraded. Run 'apt list --upgradable' to see them.
complab304pc24@complab304pc24: ~ $
```

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
```

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
```

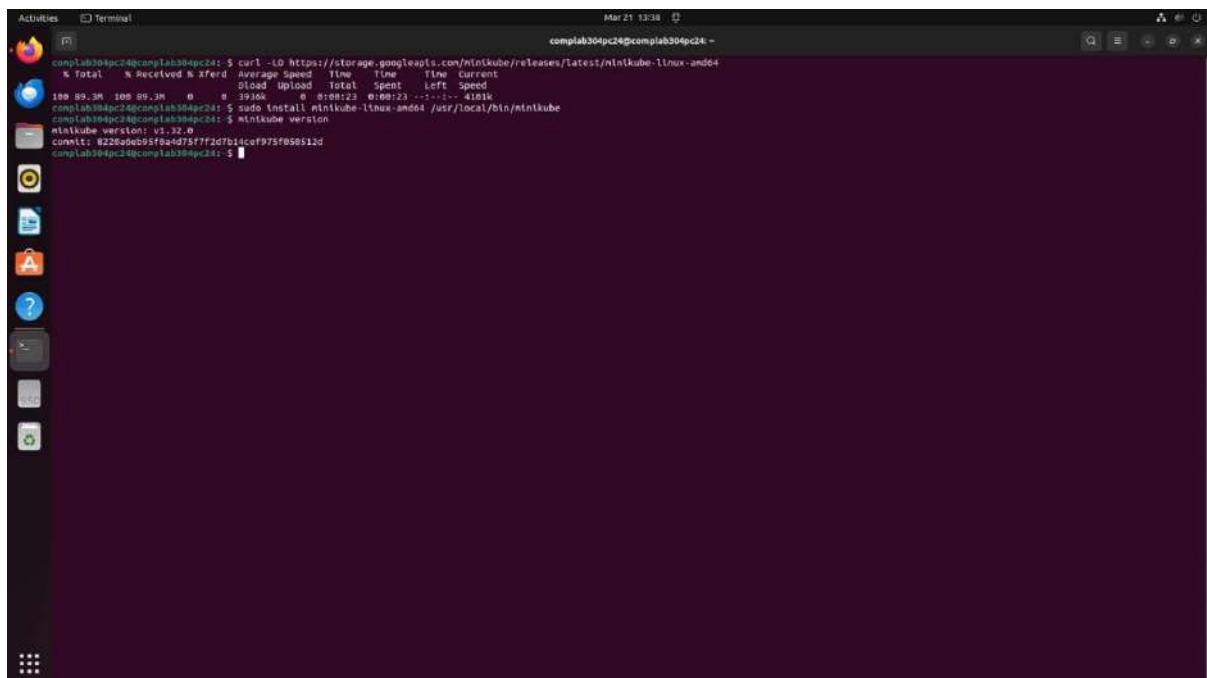
Step-5: Download and install Minikube Binary. To download and install minikube binary, run following commands,

```
$ 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
```



```
Activities Terminal Mar 21 13:38 comlab304pc24@comlab304pc24: ~
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
Total 100 89.3M 100 89.3M 0 0 3930K 0 0:00:23 0:00:23 --:--:-- 418k
curl: (29) HTTP request failed! (418)
comlab304pc24@comlab304pc24: ~$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
comlab304pc24@comlab304pc24: ~$ minikube version
minikube version: v1.32.0
commit: 8228ade095f0a4d75f77f2d7b14cef975f058512d
comlab304pc24@comlab304pc24: ~$
```

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
curl: (18) transfer closed with 0 bytes remaining to read
Total: 100 47.4M
  % Received % Xferd  Average Speed   Time   Time  Current
                                         Dload Upload Total spent Left Speed
100 47.4M  100 47.4M    0     0  5132K      0  0:00:09  --:--:-- 3721K
complab304pc24@complab304pc24: ~ chmod +x kubectl
complab304pc24@complab304pc24: ~ sudo mv kubectl /usr/local/bin/
complab304pc24@complab304pc24: ~ kubectl version -o yaml
clientVersion:
  buildDate: "2024-03-15T00:00:19Z"
  compiler: 'gc'
  gitCommit: 'e13a25b7cd7e6db5bc7380921be03671e1a49d2'
  gitTreeState: 'clean'
  goVersion: 'v1.29.3'
  goVersion: 'go1.21.8'
  major: "1"
  minor: "29"
  platform: 'linux/amd64'
kustomizeVersion: v3.0.4-0.20230801105947-ace0bf399ce3

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.

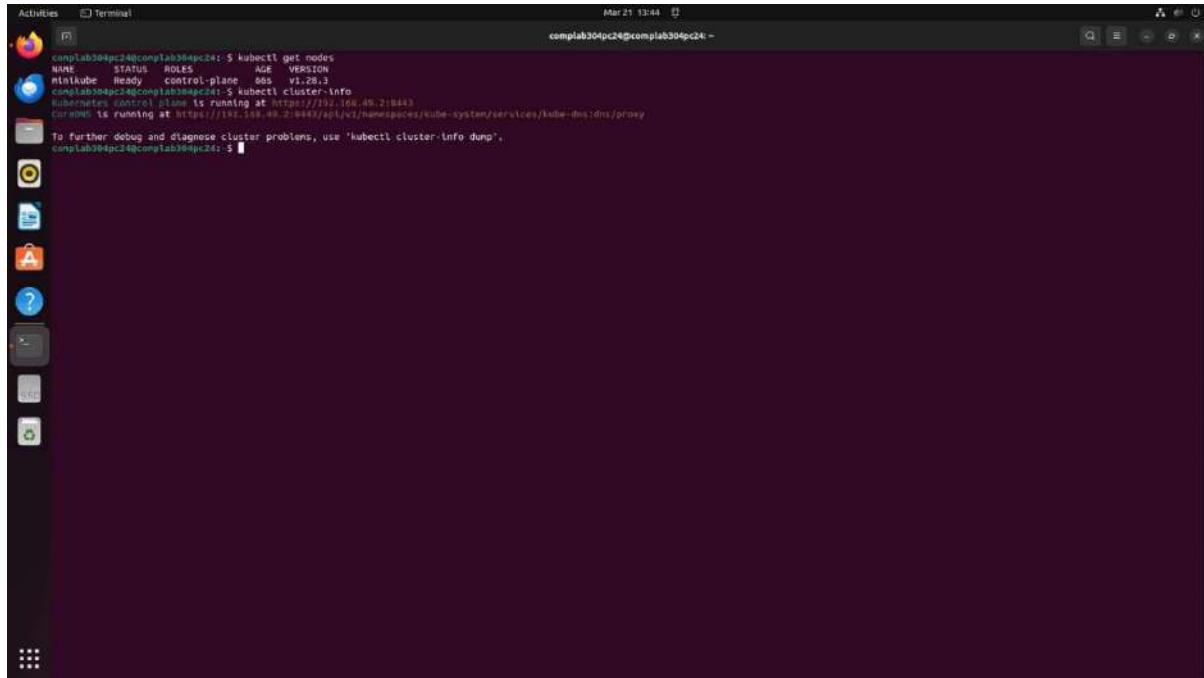
Once the minikube has started, verify the status of your cluster, run

```
$ minikube status
```

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 titled "Terminal" with the command "Mar 21 13:44". Inside the terminal, the following kubectl commands and their outputs are displayed:

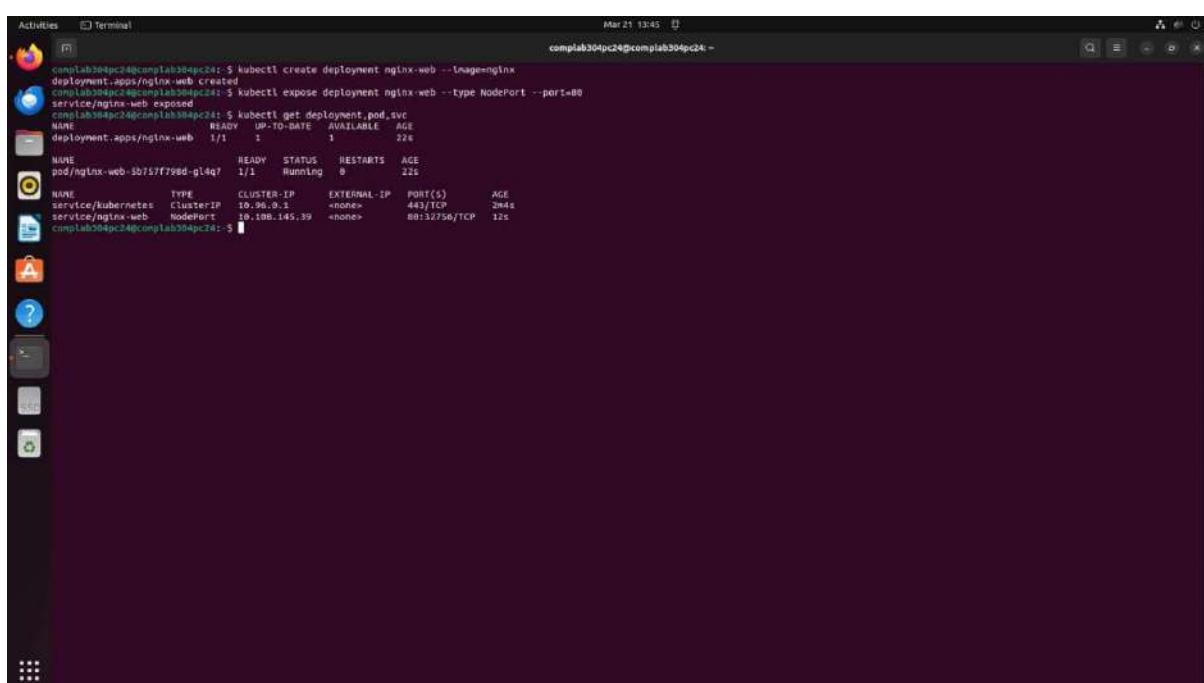
```
complab304pc24@complab304pc24: ~$ kubectl get nodes
NAME      STATUS    ROLES   AGE     VERSION
minikube   Ready    control-plane   2d       v1.22.0
complab304pc24@complab304pc24: ~$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-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 titled "Terminal" with the command "Mar 21 13:45". Inside the terminal, the following kubectl commands and their outputs are displayed:

```
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
expose "nginx-web" created
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   2h44s
service/nginx-web   NodePort   10.108.145.39  <none>     80:32756/TCP   12s
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)
efk	minikube	disabled	3rd party (Elastic)
freshpod	minikube	disabled	Google
gcp-auth	minikube	disabled	Google
gitserver	minikube	disabled	minikube
gvisor	minikube	disabled	3rd party (Knative)
helm	minikube	disabled	3rd party (Helm)
helm-tiller	minikube	disabled	3rd party (InAccel)
inacel	minikube	disabled	[login@inacel.com]
ingress	minikube	disabled	Kubernetes
ingress-dns	minikube	disabled	minikube
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)
kubewlir	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)
oam	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	3rd party
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

```

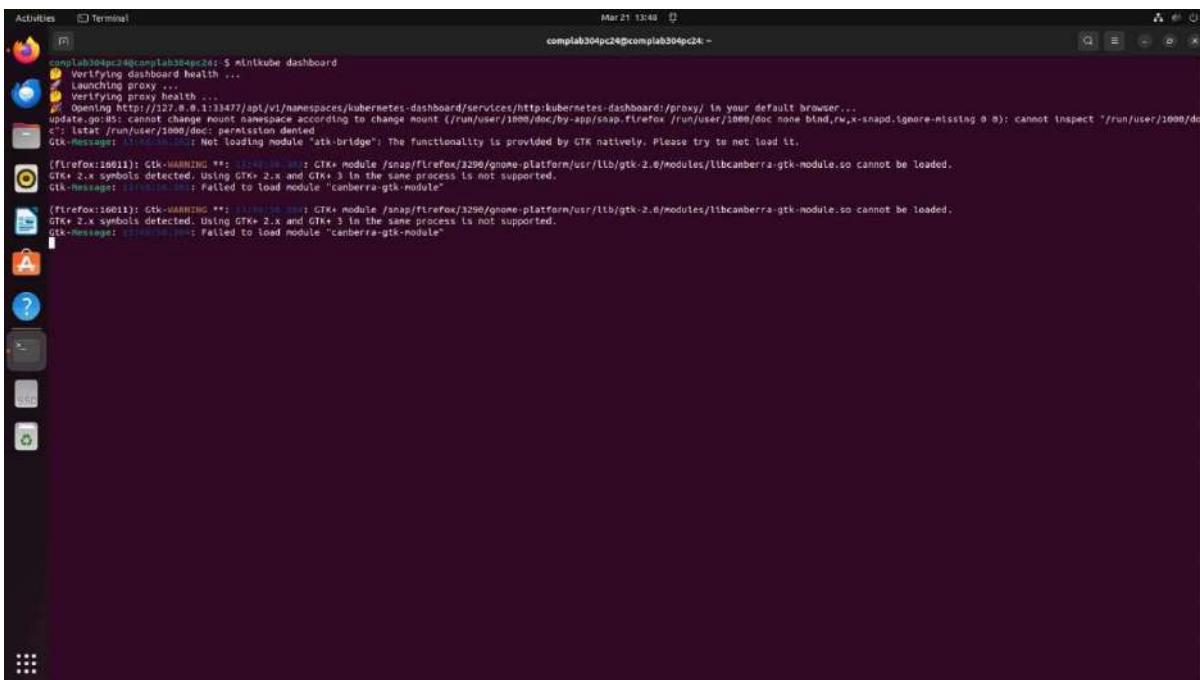
minikube addons enable dashboard
dashboards is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can review the list of minikube maintainers at https://github.com/kubernetes/minikube/blob/master/OWNERS
Using Image docker.io/kubernetsus/metrics-scraper:v1.0.8
Some dashboard 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 review 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
```



The screenshot shows the Kubernetes Dashboard interface running in a Firefox browser. The main view is the 'Workloads' section, which displays the status of Deployments, Pods, and Replica Sets. Each category has a large green circle indicating they are all running. The 'Deployments' section shows one deployment named 'nginx-web' using the 'nginx' image, with one pod running. The 'Pods' section shows one pod named 'nginx-web-5b7f179b6-gAq7' using the 'nginx' image, running on node 'minikube'. The 'Replica Sets' section also shows one replica set with one pod running. The left sidebar provides navigation links for other dashboard sections like Services, Config and Storage, and Cluster.

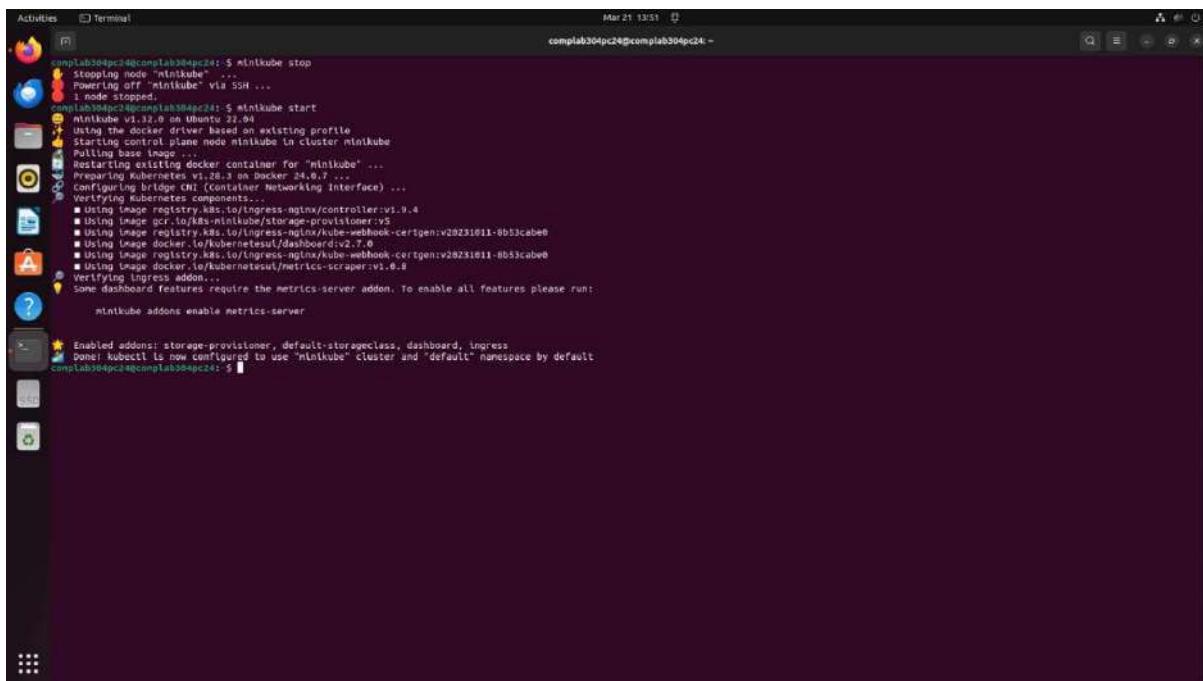
Step-13: Managing Minikube Cluster. To stop and start the minikube cluster, run beneath commands.

```
$ minikube stop
```

```
$ minikube start
```

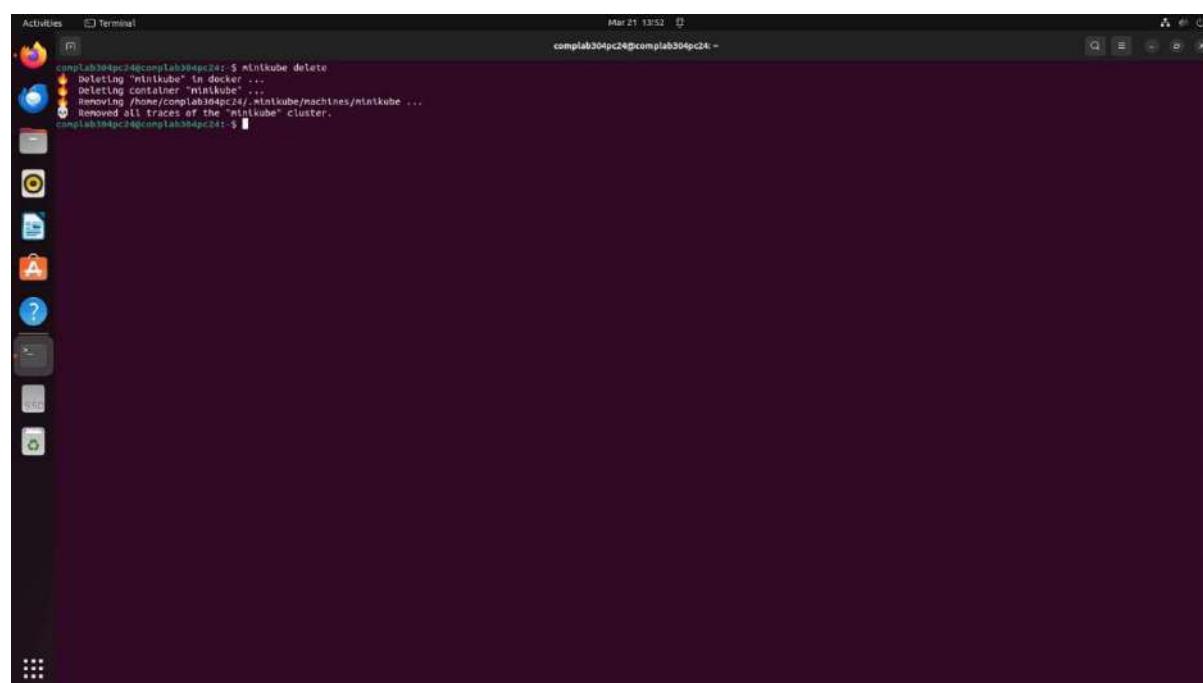
In order to delete the minikube cluster, run

```
$ minikube delete
```



```
Activities Terminal Mar 21 13:51 complab304pc24@complab304pc24: ~
complab304pc24@complab304pc24: $ minikube stop
Stopping "minikube"
Powering off "minikube" via SSH ...
1 node stopped.
complab304pc24@complab304pc24: ~$ minikube start
Starting "minikube"...
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 network "minikube" (with host-only 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:v20231011-0b53cabef
  Using Image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20231011-0b53cabef
  Using Image docker.io/kubernetes-selinux/metrics-scraperv1.8.8
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
complab304pc24@complab304pc24: ~$
```



```
Activities Terminal Mar 21 13:52 complab304pc24@complab304pc24: ~
complab304pc24@complab304pc24: ~$ minikube delete
Deleting cluster "minikube" ...
  Removing container "minikube" ...
  Removing /home/complab304pc24/.minikube/machines/minikube ...
  Removed all traces of the "minikube" cluster.
complab304pc24@complab304pc24: ~$
```

ASSIGNMENT NO:1

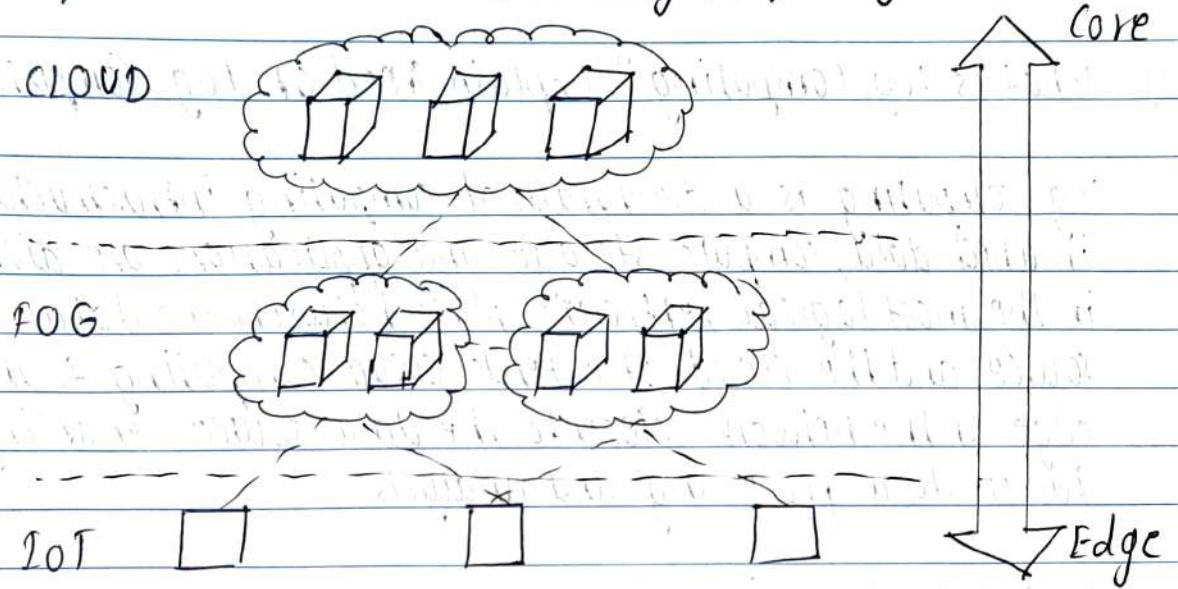
Q1] what is Fog Computing? Explain need of Fog Computing.

Fog computing is a decentralized computing infrastructure in which data, compute, storage and applications are distributed in the most logical, efficient place between the data source and the cloud. It extends cloud computing to the edge of the network, closer to the data source, enabling faster data processing and analysis.

Need of Fog Computing:

1. Latency: By processing data closer to where it's generated. Reduction. Fog computing reduces the time it takes for data to travel back and forth to the cloud, thereby decreasing latency and improve response times for time-sensitive applications like IoT devices, real-time analytics and autonomous vehicles.
2. Reliability: By distributing computing resources across the network, fog computing enhances reliability and resilience. It ensures that critical application can continue to operate even if there are disruptions in connectivity to the cloud.
3. Scalability: Fog computing provides a scalable information infrastructure that can accommodate the increasing number of IoT devices and applications.

Q2] Explain the architecture of Fog computing.



The three layers of fog computing architecture are given below:

- IoT Layer: This layer comprises IoT devices such as sensors or smartphones. These devices are usually geographically distributed and mainly aimed at sensing data and sending them to the upper layer for storage or processing.
- Fog Layer: Comprising many fog nodes, this layer is the core of fog computing architecture. Fog nodes can compute, transmit and temporarily store data, and they can be located anywhere between cloud and end devices.
- Cloud Layer: This layer is mainly composed of the centralized cloud infrastructure. It is composed of several servers with high computational and storage capabilities and provides different services.

Q3] Explain Applications of Fog Computing.

Fog computing finds applications across various industries and use cases, leveraging its ability to process data closer to the source. Some notable applications include:

1. Internet of Things (IoT): Fog Computing plays a crucial role in IoT deployments by providing real-time processing and analysis of data generated by IoT devices. It enables efficient management of connected devices, supports edge analytics and facilitates rapid decision-making in IoT.
2. Smart Grids: In the energy sector, fog computing is used for optimization operation of smart grids. By deploying computing resources at the edge of the grid, utilities can analyse data from sensors and meters in real-time.
3. Autonomous Vehicles: fog computing is essential for autonomous vehicles to process sensor data and make split decisions in real time.
4. Healthcare: fog computing improves healthcare delivery by enabling remote patient monitoring, telemedicine and healthcare analytics.
5. Retail: In the retail industry, fog computing enables personalized customer experiences, inventory management and supply chain optimization.

Q4] Explain fog, Edge and cloud computing.

Cloud Computing: Cloud computing involves storing and accessing data and applications over the internet instead of on local servers or personal computers. It provides on-demand access to a shared pool of configurable computing resources such as servers, storage, databases, networking, software and analytics, that can be rapidly provisioned and scaled with minimal management effort.

Examples of cloud computing services include Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

Edge Computing: Edge computing brings computation and data storage closer to the location where it is needed, typically at or near the source of data generation. It reduces the latency by processing data locally, near the edge of the network, instead of sending it to a centralized data centre or cloud. Edge computing is well suited for applications requiring real-time processing, such as IoT devices, autonomous vehicles and industrial automation. Eg: Routers, Gateways, edge servers, and IoT devices equipped with processing capabilities.

Fog Computing: Fog computing extends the capabilities of edge computing by providing a hierarchical architecture that spans from the edge to the cloud. It distributes

computing resources and services across a network continuum, from the cloud to the edge device, optimizing the placement of resources based on the specific requirements of applications and services. Fog computing aims to address the limitations of centralized cloud computing, such as latency, bandwidth constraints, reliability and privacy concerns. Fog computing is particularly suitable for IoT deployments, real-time analytics and applications requiring low-latency processing and distributed intelligence.

Q5] Case study of Fog Computing.

Scenario: Smart Agriculture with Fog Computing

Background: Agriculture is increasingly adopting digital technologies to improve efficiency, productivity and sustainability. Smart agriculture leverages IoT sensors, drones, satellites, imagery and data analytics to monitor crops, soil conditions, weather patterns and equipment performance. However, they face challenges such as latency, bandwidth constraints and reliability, especially in remote or rural areas with limited internet connectivity.

Solution: Fog computing provides a distributed computing infrastructure that addresses these challenges by processing data closer to the agricultural field. Here's how fog computing can be applied in a smart agriculture scenario:

1. Edge Devices and Sensors: IoT sensors deployed in the field collect data on moisture, temperature, humidity, and crop growth, weather conditions. These sensors are equipped with processing capabilities to perform basic data filtering and aggregation.
2. Edge Gateway: Edge Gateways are located in the field acts as intermediaries between the edge device and the fog node. They preprocess and filter the sensor data, performing initial analysis to identify actionable insights and anomalies.
3. Fog Nodes: Fog nodes, deployed in agricultural facilities or nearby infrastructure, serve as local processing hubs. They receive data from multiple edge gateways, perform more extensive data analysis, and execute advanced algorithms for predictive analysis, disease detection, irrigation scheduling, and crop optimization.
4. Decision Support System: The processed data and insights generated by the fog nodes are sent to a centralized decision system (DSS) supported system located in a cloud or a regional data center. The DSS aggregates data from multiple farms, performs cross-referencing, and provides farmers with actionable recommendations and alerts via web or mobile applications.

✓ A 27/3/24

Q

ASSIGNMENT NO: 2

- Q] Explore and compare the similar types of services provided by AWS, Azure and 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 services offered by each cloud provider along with their examples given below:-

1. Compute Services:

- AWS EC2: Offers resizable compute capacity in the cloud, allowing users to quickly scale instances based on demands.
- Azure Virtual Machines: Provides on demand scalable computing resources, enabling users to deploy virtual machines of various sizes and configurations.
- GCP Compute Engine: Offers virtual machines with customizable options for GPU, CPU, memory and storage, providing flexibility and control over computing resources.

2. Storage Services:

- AWS S3: Object storage service designed to store and retrieve any amount of data from anywhere on the web.
- Azure Blob Storage: Scalable object storage for unstructured data, suitable for serving documents, images, videos and more.

GCP cloud storage: Provides durable and highly available object storage for large datasets, with feature lifecycle management and versioning.

3. Database services:

- AWS RDS: Managed relational database service supporting multiple database engines such as MySQL, PostgreSQL, SQL Server and Oracle.
- Azure SQL Database: Fully managed relational database service offering built-in high availability and security features.
- GCP Cloud SQL: Managed database service for MySQL, PostgreSQL, and SQL Server with automated backups, replication and scaling capabilities.

4. Networking services:

- AWS VPC: Enables users to launch AWS resources in a virtual network isolated from other networks, providing control over IP addressing and routing.
- Azure Virtual Network: Allows users to create private networks in the Azure cloud, connecting VMs and other Azure resources securely.
- GCP VPC: Provides global, scalable and flexible virtual networking for GCP resources, with features like subnet creation and firewall rules.

5. Serverless Computing:

- AWS Lambda: Runs code in response to events without provisioning or managing servers, allowing for highly scalable and cost-effective application development.
- Azure Functions: Event driven serverless compute service that lets users write and deploy and run small pieces of code (functions) without worrying about infrastructure.
- GCP Cloud Functions: Executes functions in response to HTTP events triggered by HTTP requests, Pub/Sub messages, or cloud storage changes, enabling serverless application development.

6. Containers:

- AWS ECS: Fully managed container orchestration service for deploying, managing and scaling containerized applications using Docker containers.
- AZURE AKS: Managed Kubernetes service that simplifies deploying, managing and scaling containerized applications using Kubernetes.
- GCP GKE: Managed Kubernetes service that automates the deployment, scaling and management of containerized applications on Google Cloud Infrastructure Platform.

7. Big Data Analytics

- AWS EMR: Managed Big data platform for processing vast

amounts of data using popular frameworks such as Apache, Hadoop, Spark and HBase.

- Azure HD Insight: Fully managed cloud service for big data analytics using open-source frameworks such as Hadoop, Spark and Kafka.
- GCP BigQuery: Serverless, highly scalable data warehouse for analyzing large datasets using SQL queries with blazing-fast performance.

8. ML and AI:

- AWS SageMaker: Fully managed service for building, training and deploying machine learning models at scale.
- Azure Machine Learning: Cloud Based service for building, training and deploying machine learning models using any framework or language.
- GCP AI Platform: Provides tools and services for building, training and deploying machine learning models including TensorFlow, scikit-learn and XG-Boost.

9. Identity and Access Management (IAM):

- AWS IAM: Centralized Entity Management service that helps securely control access to AWS resources.
- Azure Active Directory (AAD): Cloud Based identity and access management service that enable for sign-on and multi-factor authentication for applications.
- GCP IAM: Manage Access control for GCP resources by defining who (identity) has what access (roles) for which resource.

10. Content Delivery and CDN:

- AWS CloudFront: Global content delivery network (CDN) service that accelerates the delivery of websites, APIs, video content and other web assets.
- Azure CDN: Delivers high-bandwidth content to users worldwide with low latency and high speed access to applications and content.
- GCP Cloud CDN: Low latency content delivery network (CCDN) service that caches content at Google's globally distributed edge locations to improve website and application performance.

11. Monitoring and Cloud Logging:

- AWS CloudWatch: Provides monitoring and observability for AWS resources and applications, allowing users to collect and track metrics, set alarms and gain insights through log analysis.
- Azure Monitor: Comprehensive monitoring solution for Azure resources and applications, offers metrics, logs and alerts to ensure optimal performance and availability.
- GCP Stackdriver: Monitoring, logging and diagnostic platform for GCP and hybrid cloud environments, offering visibility into performance, uptime and application health through metrics, logs and traces.

12. Content Management and Collaboration:

- AWS WorkDocs: Secure content collaboration service that enables users to store, share and collaborate on documents, spreadsheets and presentations in the cloud.
- Azure SharePoint Online: Cloud-based platform for content management and collaboration, providing features such as document library, team sites and workflow automation.
- GCP G-Suite: Suite of cloud-based productivity and collaboration tools including Gmail, Google Drive, Google Docs, Google Sheets and Google Slides, facilitating real-time collaboration and communication within organizations.

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

Thadomal Shahani Engineering College, Bandra

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

SUBJECT : CLOUD COMPUTING LAB MINI PROJECT

SUPERVISOR : DR. TANUJA SARODE

GROUP MEMBERS :

- **Ayush Bhanushali 2103018**
- **Maanav Chellani 2103030**
- **Parth Dabholkar 2103032**
- **Vedant Devkar 2103034**
- **Viraj Dighe 2103035**

Description of problem statement:

Our task involves developing a simplified to-do list application targeting diverse users who require an easy-to-use, intuitive interface for managing everyday tasks. Since our focus lies on implementing the solution using AWS services, we must prioritize selecting the right combination of tools and techniques offered by AWS to attain a balanced blend of convenience, effectiveness, and economic viability.

To accomplish this objective, following were used:

1. Utilized **AWS Amplify** to host the static web pages and user interfaces responsible for rendering the to-do list application. Leverage Amplify's inherent benefits, such as streamlined configuration, automated deployments, and direct connection to underlying AWS services.
 2. Selected **DynamoDB** as the NoSQL database solution to efficiently store and manage structured and semi-structured to-do list items generated by numerous users. Benefit from its flexible data model, auto-scaling properties, and exceptional performance characteristics tailored explicitly towards serverless workloads.
 3. Employed **AWS Lambda** to execute custom backend logic tied directly to specific triggers originating from either the Amplify frontend or DynamoDB itself. This approach ensures eventual consistency, reduces coupling between components, and promotes separation of concerns, ultimately leading to more straightforward debugging and enhanced modularity.
- .
- .

Block Diagram

Using AWS, Amplify, DynamoDB, and Lambda together brings several advantages:

1. Serverless Computing: With AWS Lambda, there is no need to worry about servers, operating systems, or patches. It lets developers run their code without having to manage infrastructure, resulting in reduced administration efforts and lower costs.
2. Auto-scaling: Both AWS Lambda and DynamoDB provide auto-scalability features, meaning resources grow or shrink according to actual traffic demands. This leads to better performance, higher availability, and less wasted expenditure compared to traditional fixed-capacity configurations.
3. Integrated Development Environment: Amplify simplifies setting up projects and connecting to AWS services. Its console offers prebuilt templates and components, speeding up development cycles and reducing common pitfalls.
4. Managed Databases: DynamoDB removes the burdens associated with managing databases, such as performing backups, restoring snapshots, patching software, or tuning indexes. Developers can concentrate solely on writing queries instead of worrying about administrative aspects.
5. Reliable Storage: DynamoDB guarantees durability by safely distributing redundant copies of data across multiple Availability Zones, mitigating risks caused by single point failures or natural disasters.
6. Cost Efficient: Pay only for what you utilize; neither Lambda nor DynamoDB charges for idle compute resources or disk space. Moreover, since there are no servers to maintain, expenses related to hardware procurement, installation, or decommissioning don't exist.
7. Performance Optimized: DynamoDB delivers predictably fast responses irrespective of the dataset size thanks to its internal engine designed around flash storage technology. Similarly, Lambda executes code close to the invoked service, thereby minimizing latencies.
8. Highly Secure: All three services—Amplify, Lambda, and DynamoDB—adhere strictly to established security best practices, offering multi-layered protections covering identity and access management, data encryption, firewalls, and intrusion detection.
9. Real-time Feeds: Through DynamoDB Streams, developers gain insights into recently added, modified, or removed records. Such knowledge can then trigger downstream actions using Lambda, enabling real-time analytics, messaging, and ETL processes.

10. Monitoring Capabilities: Each service exposes rich telemetry, empowering developers to measure application health, pinpoint bottlenecks, and troubleshoot issues effectively. Tools like Amazon CloudWatch, Application Insights, or X-Ray integrate seamlessly, shedding light 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

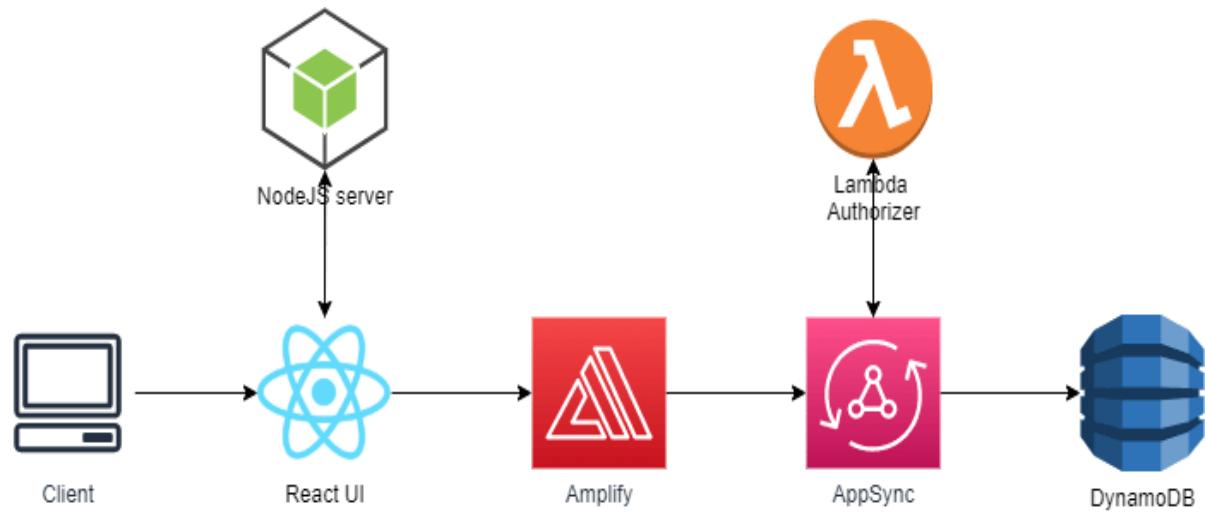


Fig) React App working with Aws Services

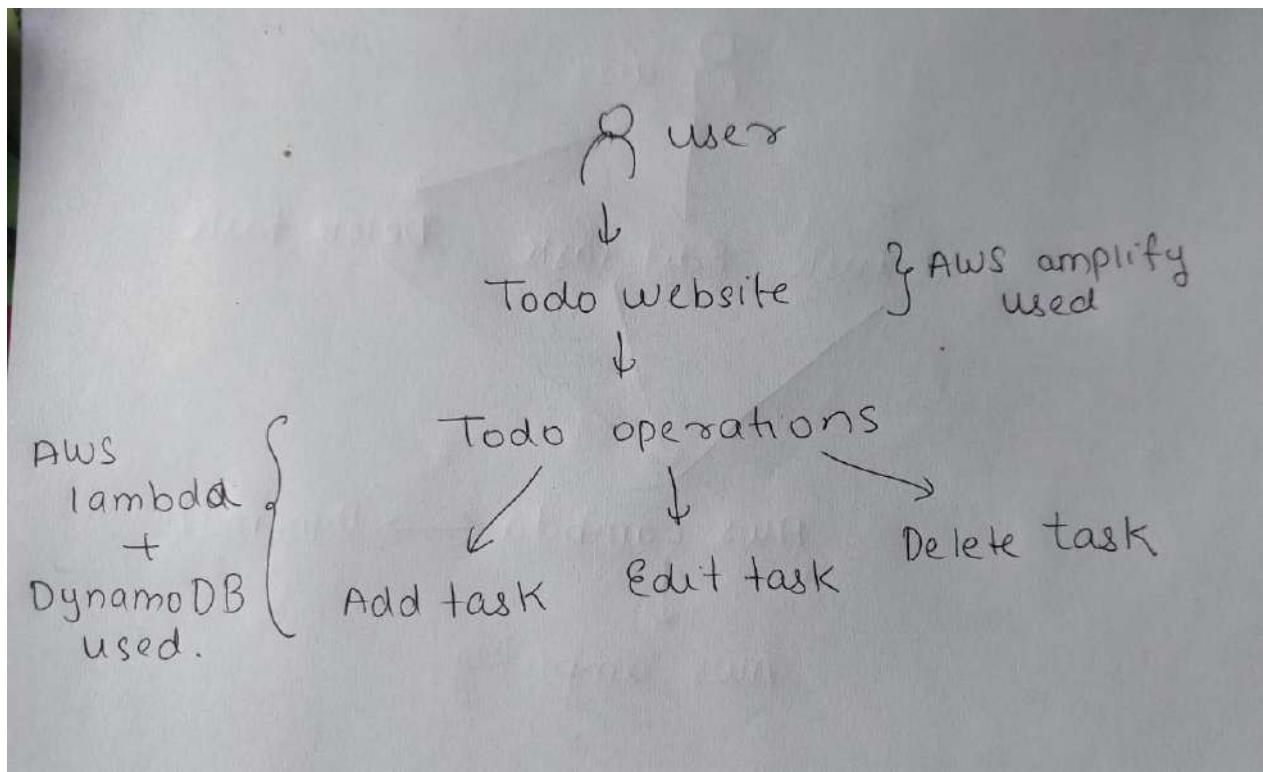
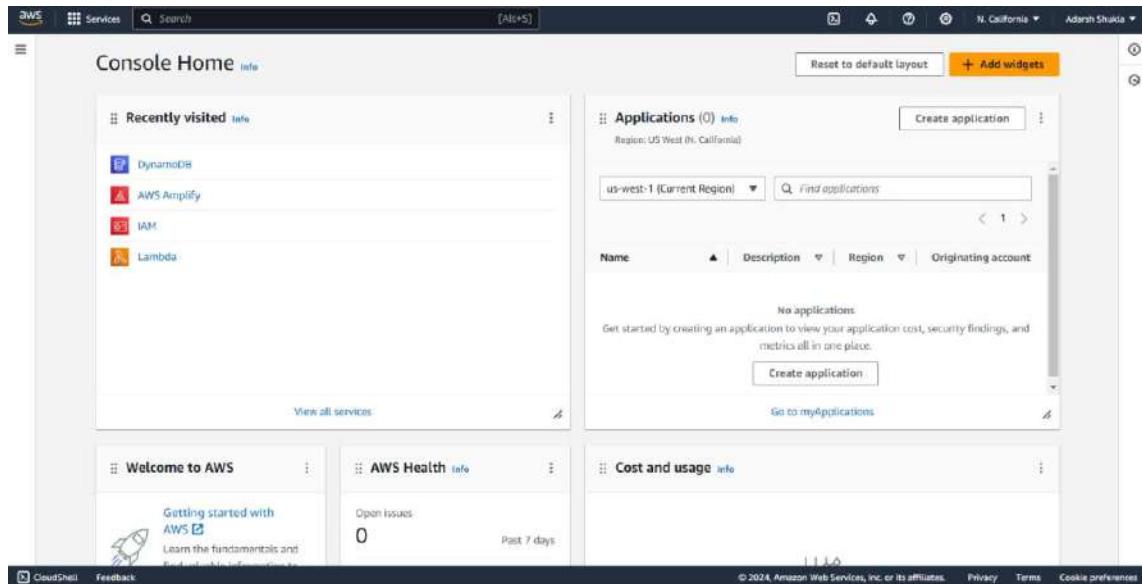


Fig) User Workflow Diagram

Major Steps And Codes

1) Create a AWS Account and go to the Console



2)Create a Table in DynamoDb

The screenshot shows the AWS DynamoDB service dashboard. On the left, there's a sidebar with options like Dashboard, Tables, Explore items, PartiQL editor, Backup, Exports to S3, Imports from S3, Integration, Reserved capacity, and Settings. Under the Tables section, there's a 'Create table' button. The main area shows a table named 'Tasks' with one item: {id: 'S', status: 'Active', task: 'OFF', deletion_protection: 'Off', read_capacity_mode: 'Provisioned', provisioned: 1, write_capacity_mode: 'Provisioned', provisioned_w: 0}. There are also columns for Status, Partition key, Sort key, Indexes, Deletion protection, Read capacity mode, Write capacity mode, Total size, and Last modified.

3) Host the Ui part in Aws Amplify

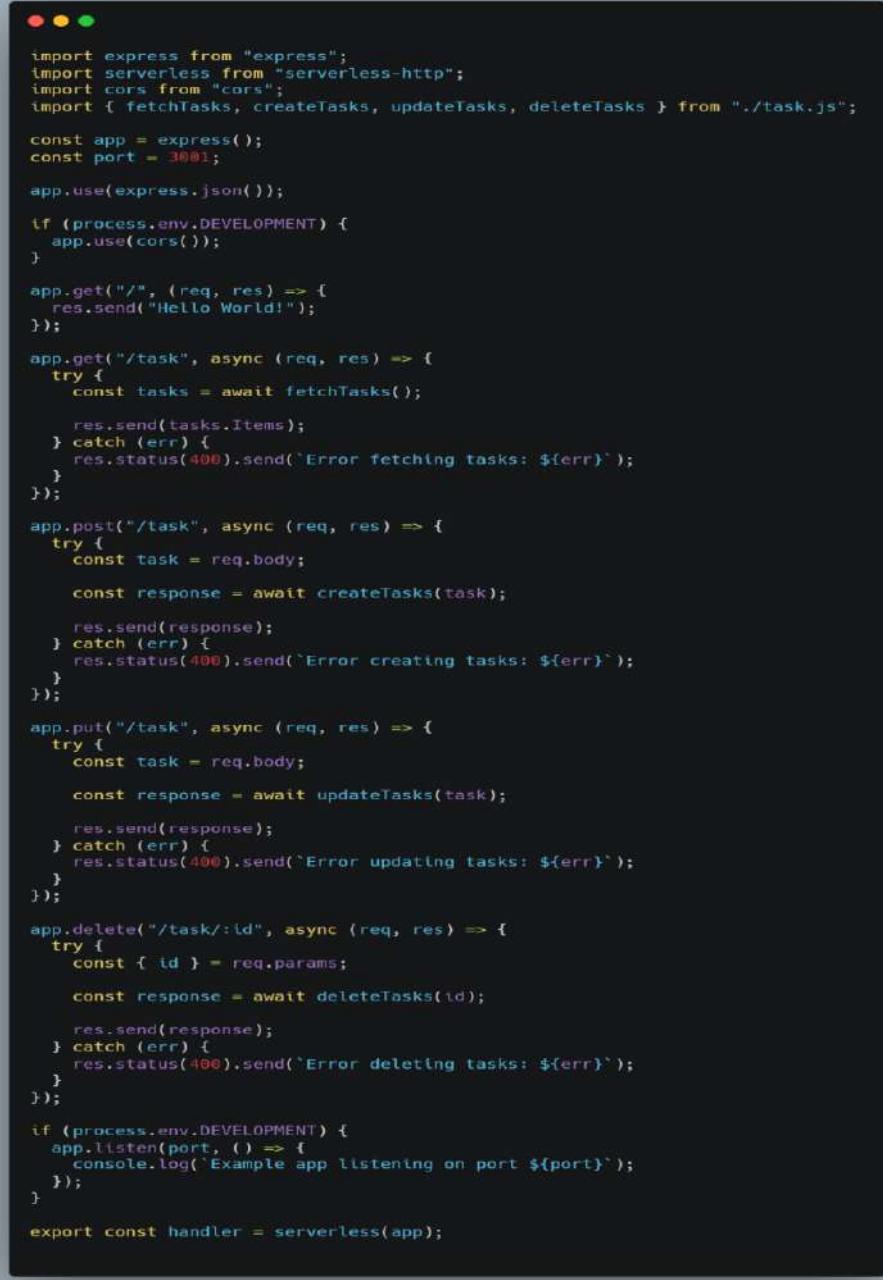
The screenshot shows the AWS Amplify service dashboard. On the left, there's a sidebar with options like All apps, App settings (General, Amplify Studio settings, Domain management, Build settings, Previews, Notifications, Environment variables, Access control, Monitoring, Rewrites and redirects, Custom headers), Documentation, and Support. The main area shows the 'CCL-ui' app homepage, which includes a 'Learn how to get the most out of Amplify Hosting' section, a 'Hosting environments' tab (selected), a 'Backend environments' tab, a 'Connect branch' button, and a 'main | ui' section showing a preview of the deployed UI at <https://main.cclui.amplifyapp.com>. It also shows deployment details: Last deployment (3/27/2024, 2:45:32 PM), Last commit (Auto-build | GitHub - main), and Previews (Disabled). A deployment flow diagram shows Provision, Build, and Deploy steps connected by green arrows with checkmarks.

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

The screenshot shows the AWS Lambda Functions console. In the top navigation bar, 'Lambda' is selected under 'Services'. The search bar contains 'Search' and the keyboard shortcut '[Alt+S]'. On the right, there are account details for 'N. California' and 'Adarsh Shukla'. Below the navigation, the function name 'task-list-api' is displayed. A 'Throttle' button, a 'Copy ARN' button, and an 'Actions' dropdown menu are visible. The main content area is titled 'Function overview' with tabs for 'Info' (selected) and 'Code source'. Under 'Info', there's a 'Diagram' tab showing a single function node labeled 'task-list-api' with '(0)' layers, and a 'Template' tab. To the right, there's a 'Description' section with fields for 'Last modified' (1 hour ago), 'Function ARN' (arn:aws:lambda:us-west-1:891377318888:function:task-list-api), and a 'Function URL' (https://7brlme...). A 'Tutorial' sidebar on the right is titled 'Create a simple web app' and lists steps to build and invoke a Lambda function.



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}`);
    });
}

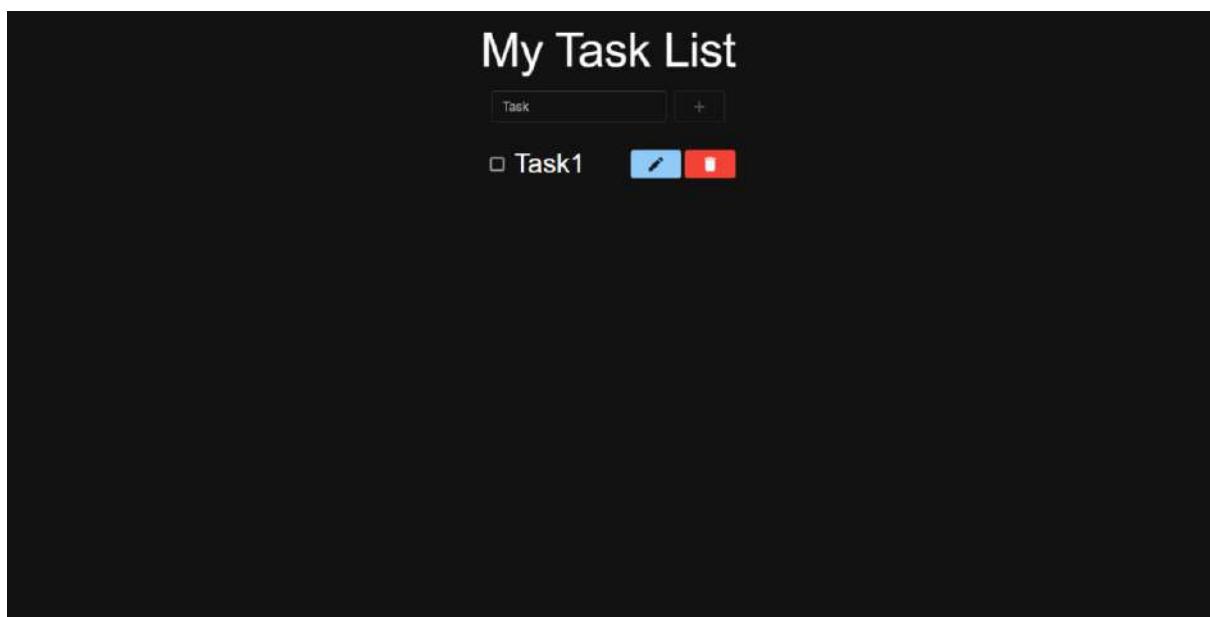
export const handler = serverless(app);
```

Fig) Express Code for handling Routes

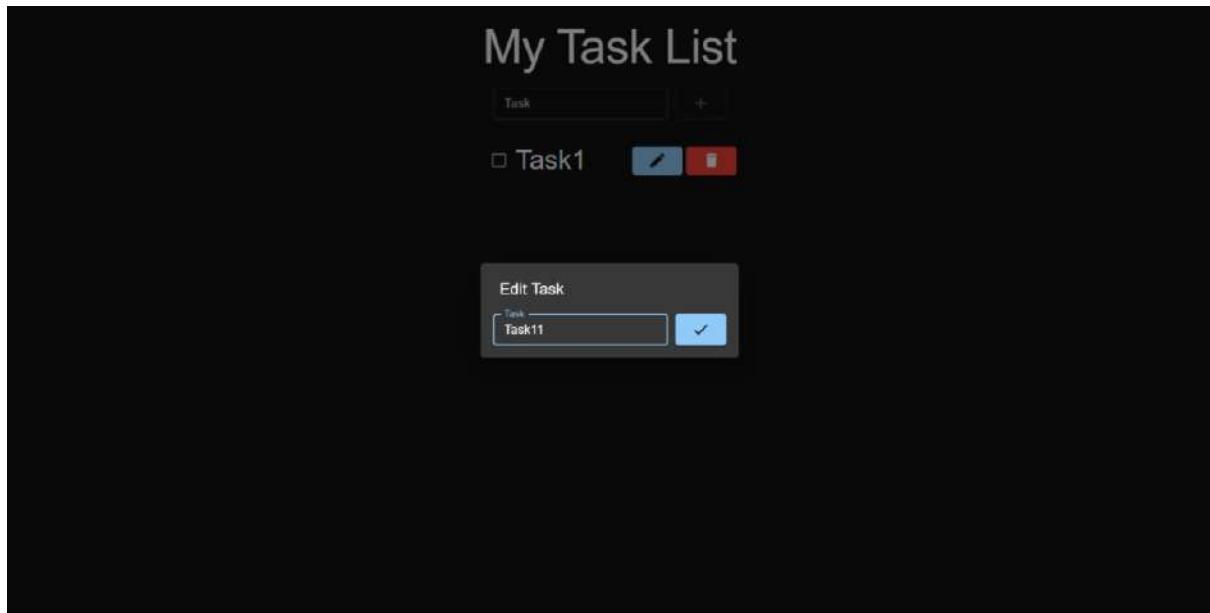
Screenshots of project in sequence



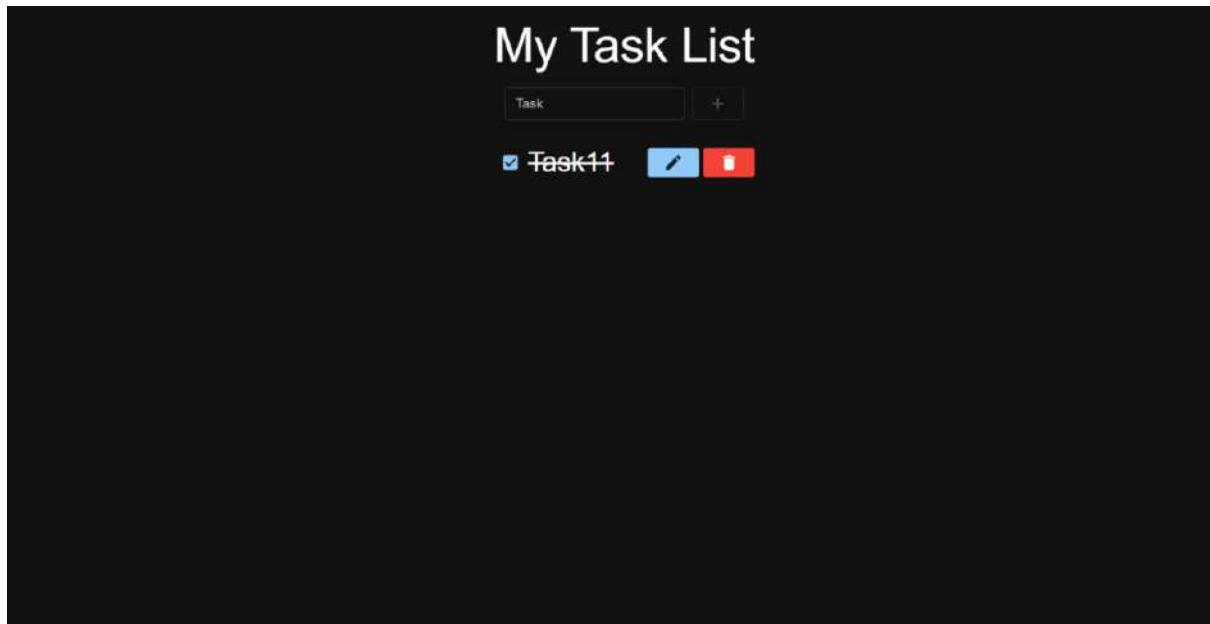
Task List Ui Hosted Live



Adding of the Task



Edit of the task



Completion of the task

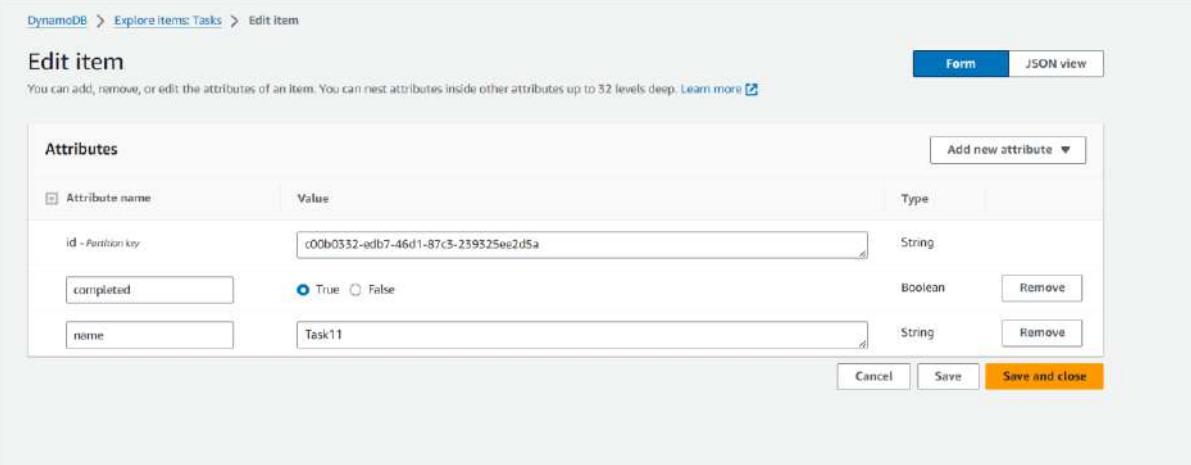
DynamoDB > Explore items: Tasks > Edit item

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 [↗](#)

Attributes		Type	Add new attribute ▾
<input type="checkbox"/> Attribute name	Value		
id - Partition key	c00b0332-edb7-46d1-87c3-239525ee2d5a	String	
completed	<input checked="" type="radio"/> True <input type="radio"/> False	Boolean	Remove
name	Task11	String	Remove

[Cancel](#) [Save](#) [Save and close](#)



Live Updation Done in Dynamo Db

Conclusion

By utilizing AWS services like Amplify, DynamoDB, and Lambda, we've built a application infrastructure without worrying about managing servers or complex setups. 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.