

Thadomal Shahani Engineering College

Bandra (W.), Mumbai- 400 050.

CERTIFICATE

Certify that Mr./Miss Om Anindha Shete of Computer Engg Department, Semester VI with Roll No. 2103163 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 10/09/2024

Principal

CONTENTS

SR. NO.	EXPERIMENTS	PAGE NO.	DATE	TEACHERS SIGN.
1.	Introduction and overview of cloud computing	1-10	19/1	
2.	To study and implement Hosted Virtualization using Virtual Box and KVM.	11-20	29/1	
3.	To study and implement Bare-metal Virtualization using Xen.	21-26	2/2	
4.	To study and implement Infra-structure as a EC2.	27-36	9/2	
5.	To study and implement Platform as a Service using AWS elastic Beanstalk.	37-44	23/2	JRF
6.	To study and implement Identity and Access Management practices on Aws.	53-76	26/2	10/4/2020
7.	To study and implement storage as a service using Own Cloud	45-58	11/3	
8.	To study and implement DB as a service on AWS RDS.	77-89	11/3	
9.	To study and implement Containerization using Docker.	90-96	15/3	
10.	To study and implement container orchestration using Kubernetes.	97-105	18/3	
11.	Assignment 1	106-113	22/2	
12.	Assignment 2	114-116	5/2	
13.	Mini-project.	117-128	8/4	

Experiment 1

Aim: Introduction and overview of cloud computing.

- Q.1 Definition of cloud computing**
- Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configuration computing resources (e.g., networks, servers) that can be rapidly provisioned and released in this provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

- Q.2 Characteristics of cloud computing**

- The essential characteristics of cloud computing are :-
- 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 provider.

2) Broad network access

→ Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms.

- For e.g.: mobile phones, tablets, laptops.

3) 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 independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction.

- For e.g.; storage, processing, memory and network bandwidth.

4) 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.

- ⇒ Measured services
- ⇒ Cloud systems automatically control and optimize resource use by leveraging a metering capability at (some) level of abstraction appropriate to the type of service.
- Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service
- For e.g.; storage, processing, bandwidth,

Q.3 NSIT cloud computing model

⇒

Essential characteristics	Broad Network Access	Rapid Elasticity	Measured services	On-demand services
---------------------------	----------------------	------------------	-------------------	--------------------

Resource POOLING

Service Models	Software as a service (SaaS)	Platform as a Service (PaaS)	Infrastructure as a Service (IaaS)
----------------	------------------------------	------------------------------	------------------------------------

Deployment Models	Public	Private	Hybrid	Community
-------------------	--------	---------	--------	-----------

- The service models are :-
 - 1) Software as a Service (SaaS)
 - 2) Platform as a Service (PaaS)
 - 3) Infrastructure as a Service (IaaS)

- Deployment Models are :-

- 1) Private cloud
- 2) Community cloud
- 3) Public cloud
- 4) Hybrid cloud.

Q.4 Different models of cloud computing

⇒

• Service Models

1) Software as a Service (SaaS)

⇒ The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure.

- The applications are accessible from various client devices through either a thin client interface, such as a web browser, or program interface.

2) 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 networks, servers, operating systems or storage, but has control over the deployed applications and possibly configurations setting for the application hosting environment.

3). Infrastructure as a Service (IaaS)

⇒ The capability provided to the consumer is to provision processing, storage, networks and other fundamental computing resources where the

consumer is to provision processing, storage, networks and other fundamental processing resources where the consumer is able to deploy and run arbitrary software, which can include operating system and applications.

Deployment Models

1) Private Cloud

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

2) Community Cloud

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

3) public cloud

→ The cloud infrastructure is provision for open use by the general public.

- It may be owned, managed and operated by a business, academic or government organizations or some combination of them.

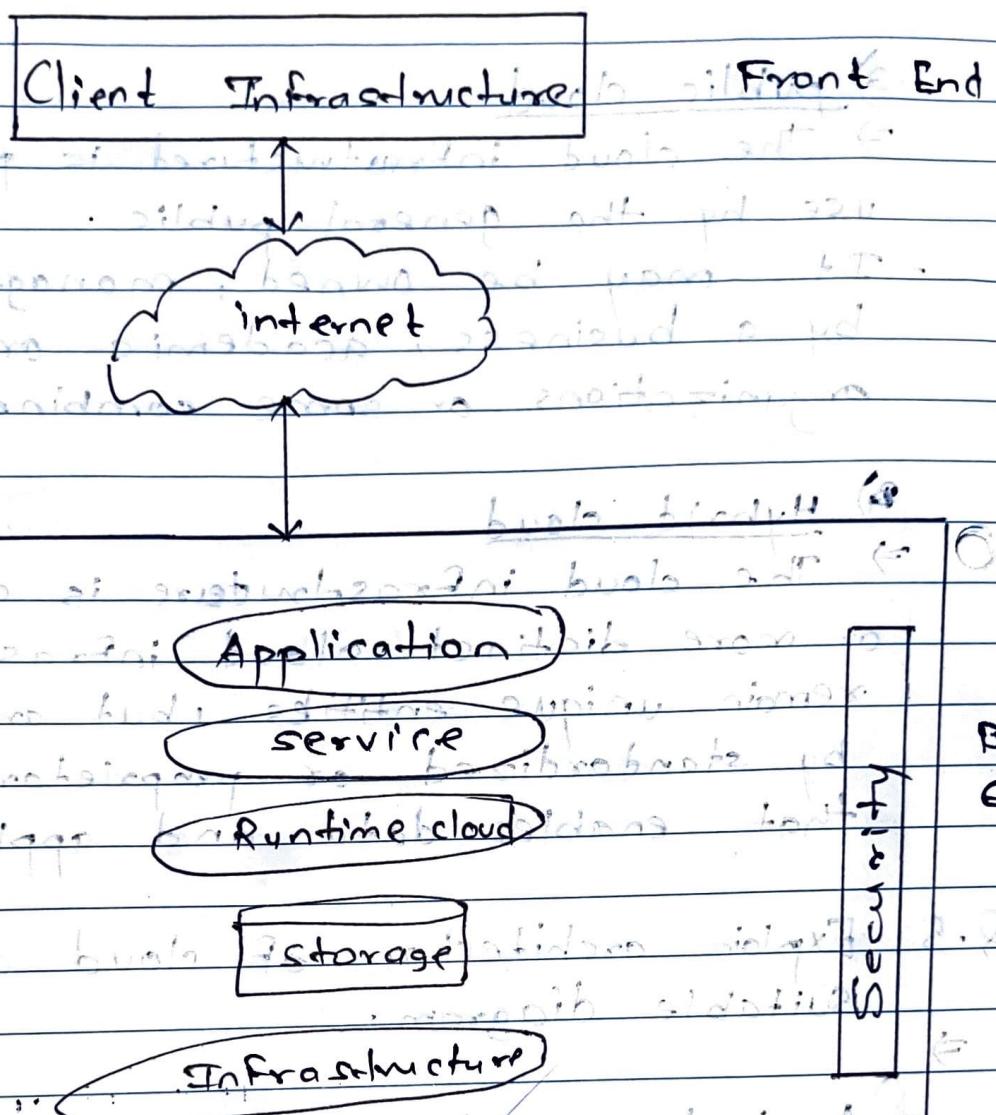
4) Hybrid cloud

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

Q.5 Explain architecture of cloud computing with suitable diagram.



- A cloud computing is a technology used by both small and large organizations to store the information in cloud and access it from anywhere at anytime using the internet connection.
- Cloud computing architecture is combination of service-oriented architecture and event-driven architecture.
- Cloud computing architecture is divided into the following two parts:
 - 1) Front End
 - 2) Back End



1) Front End

- ⇒ The front end is used by the client.
- It contains client-side interfaces and applications that are required to access the cloud computing platform.
- The front end includes web servers, thin and fat clients, tablets for mobile devices.

⇒ Back End

⇒ The back end is used by the service provider.

- It manages all resources that are required to provide cloud computing services.
- It includes a huge amount of data storage, security mechanism, virtual machines, deploying models, servers, traffic control mechanism, etc.

Q.6 Benefits and limitations of cloud computing.

⇒ Both maintained under A.L.T.P.H. on 17/12

- Benefits :

- 1) Once the data is stored in the cloud, it is easier to get back-up and restore that data using the cloud.
- 2) Cloud computing reduces both hardware and software maintenance costs for organizations.
- 3) Cloud computing allows us to easily access all cloud data via mobile.
- 4) Cloud computing offers APIs to the users for access services on the cloud and pays the charges as per the usage of services.
- 5) Cloud offers us a huge amount of storing capacity for storing our important data such as documents, images, audio, in one place.

- Limitations

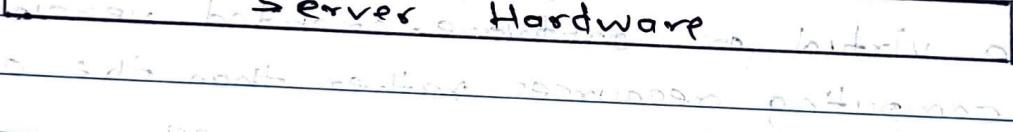
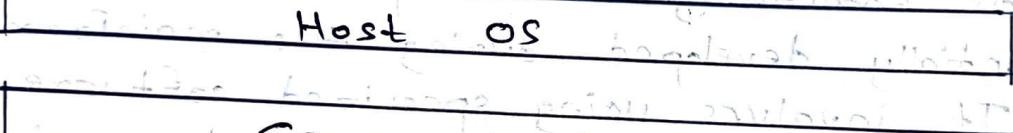
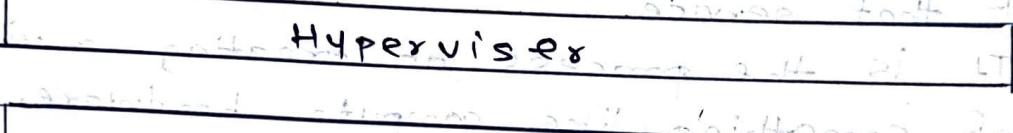
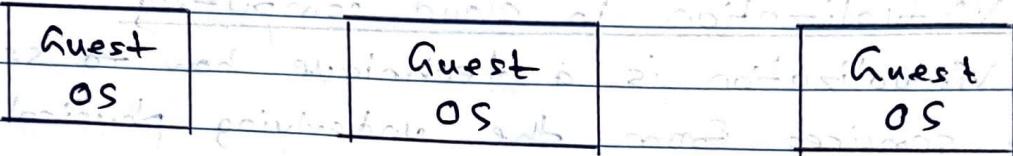
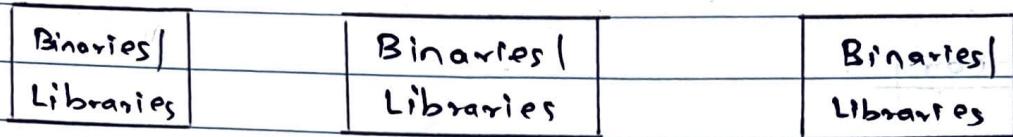
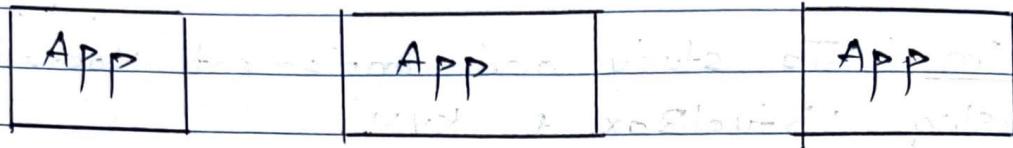
- Disadvantages of cloud computing:
- 1) Dependence on internet connectivity.
 - 2) Limited control and customization of the technology.
 - 3) Security concerns include the risk of data breaches and loss of sensitive information.
 - 4) Possibility of service disruptions and downtime.
 - 5) Limited compliance with specific regulations, such as HIPAA for healthcare data.

Experiment No. 2

Aim : To study and implement Hosted virtualization using VirtualBox & KVM.

Theory :

- Virtualization in cloud computing
 - Virtualization is a technique how to separate services from the underlying physical delivery of that service.
 - It is the process of creating a virtual version of something like computer hardware. It was initially developed during the mainframe era.
 - It involves using specialized software to create a virtual or software-created version of a computing resources rather than the 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 same time, increasing the utilization and flexibility of hardware.
 - In other words, one of the main cost-effective, hardware-reducing, and energy-saving techniques used by cloud providers is Virtualization.
 - Virtualization allows sharing of a single physical instance of a resource or an application among multiple customers and organizations at one time.



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) Enables running multiple operating systems.
- 7) Pay per use of the IT infrastructure on demand.

- Hypervisors

- A Hypervisor is a form of virtualization software used in cloud hosting to divide and allocate the resources on various pieces of hardware.
- The program which provides partitioning, isolation or abstraction is called a virtualization hypervisor.
- - The hypervisor is a hardware virtualization technique that allows multiple guest operating system to run on a single host system at the same time.
- A hypervisor is sometimes also called a virtual machine manager (VMM).
- For e.g., Amazon Elastic Compute Cloud (EC2) allows organizations to scale their cloud computing capabilities with Xen-based hypervisors.

- Types of Hypervisors

- - 1) TYPE - 1 Hypervisor! The hypervisor runs directly on the underlying host system.
 - It is also known as a "Native Hypervisor" or "Bare Metal Hypervisor".
 - It does not require any base server operating system.
 - It has direct access to hardware resources.
 - For e.g., VMware ESXi, Citrix XenServer, Microsoft Hyper-V hypervisor.

2) Type 2 Hypervisor : A host operating system runs on the underlying host system.

- It is also known as "Hosted Hypervisor"
- Such kind of hypervisors doesn't run directly over the underlying hardware rather than they run as an application in a Host system.
- Basically, the software is installed on an OS. Hypervisor asks the OS to make hardware calls.
- For e.g., VMware Players or Parallel desktops.

• Comparison between VirtualBox and KVM :

1) Performance

- ⇒ KVM, being a type 1 hypervisor, provides better performance compared to VirtualBox which is a type 2 hypervisor.
- KVM has direct access to the host hardware, resulting in faster execution and better overall performance.

2) Management

- ⇒ KVM is managed through command-line tools and APIs, making it suitable for advanced users and system administrators. VirtualBox, on the other hand, ~~KVM~~ provides a user-friendly graphical interface that makes it

easier for beginners to manage virtual machines.

3) Compatibility

- VirtualBox offers compatibility with various operating systems, including Windows, macOS and Linux, making it a popular choice for desktop virtualization. On the other hand, KVM is primarily used in Linux environments and is well-integrated with the Linux kernel.

4) Resource Allocation

- KVM allows for dynamic allocation of resources like CPU and memory to virtual machines.
- It also provides better control over resources utilization and allocation.
- In contrast, VirtualBox has limitations in resource management, making it less suitable for resource-intensive applications.

5) Community Support

- VirtualBox has a larger user community and a vast amount of online resources available for support and troubleshooting.
- KVM, on the other hand, has a more niche user base, mostly consisting of Linux enthusiasts and developers.

(A)

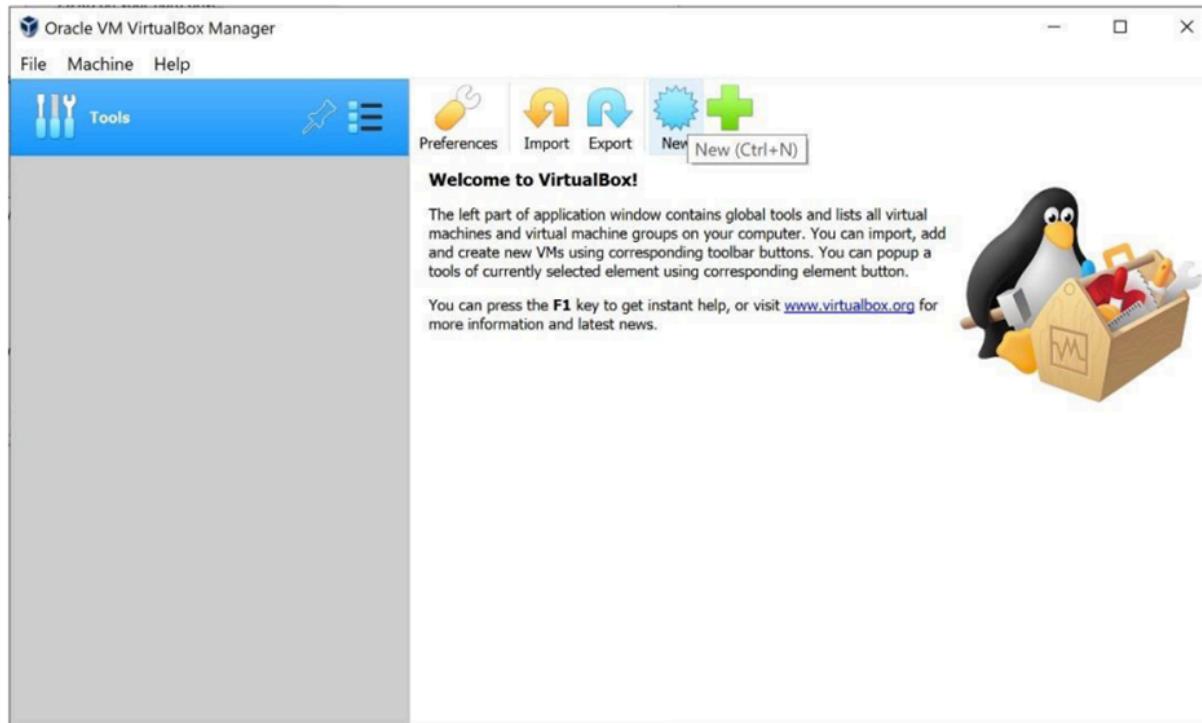
SP
26/2/24

Output:

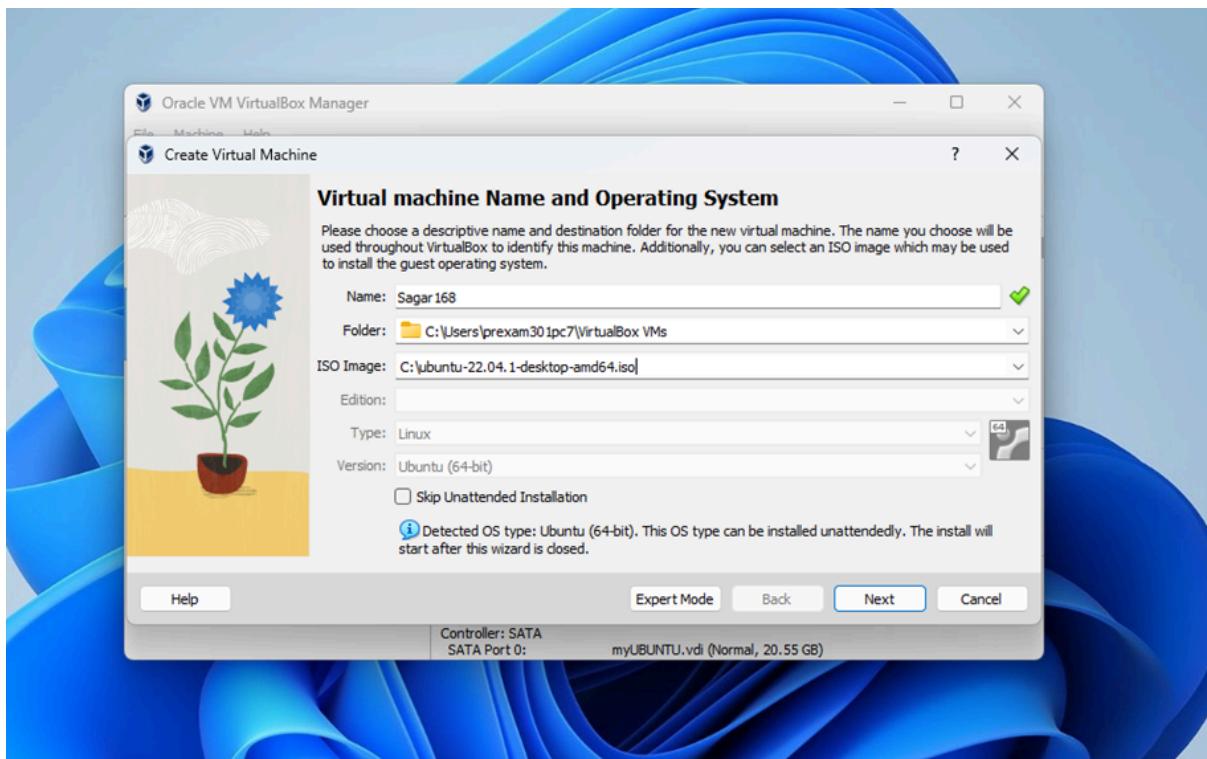
1. Download and install VirtualBox



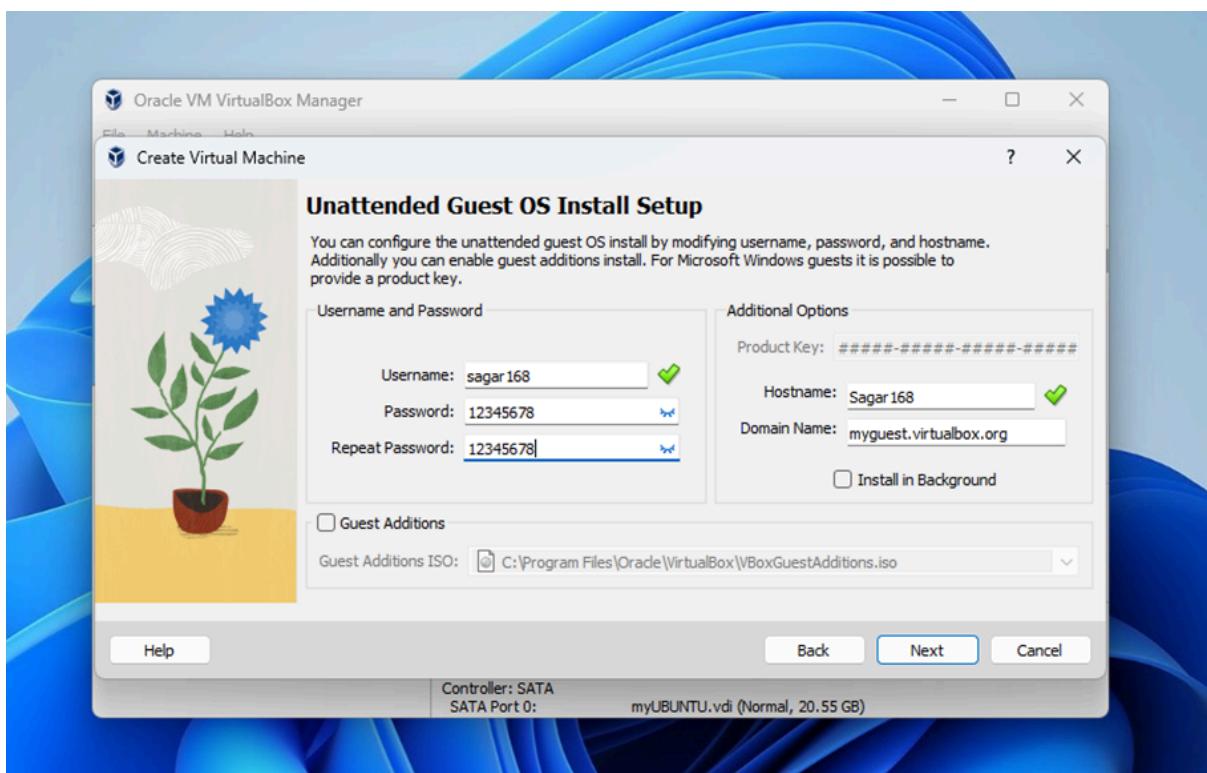
2. Create a new virtual machine.



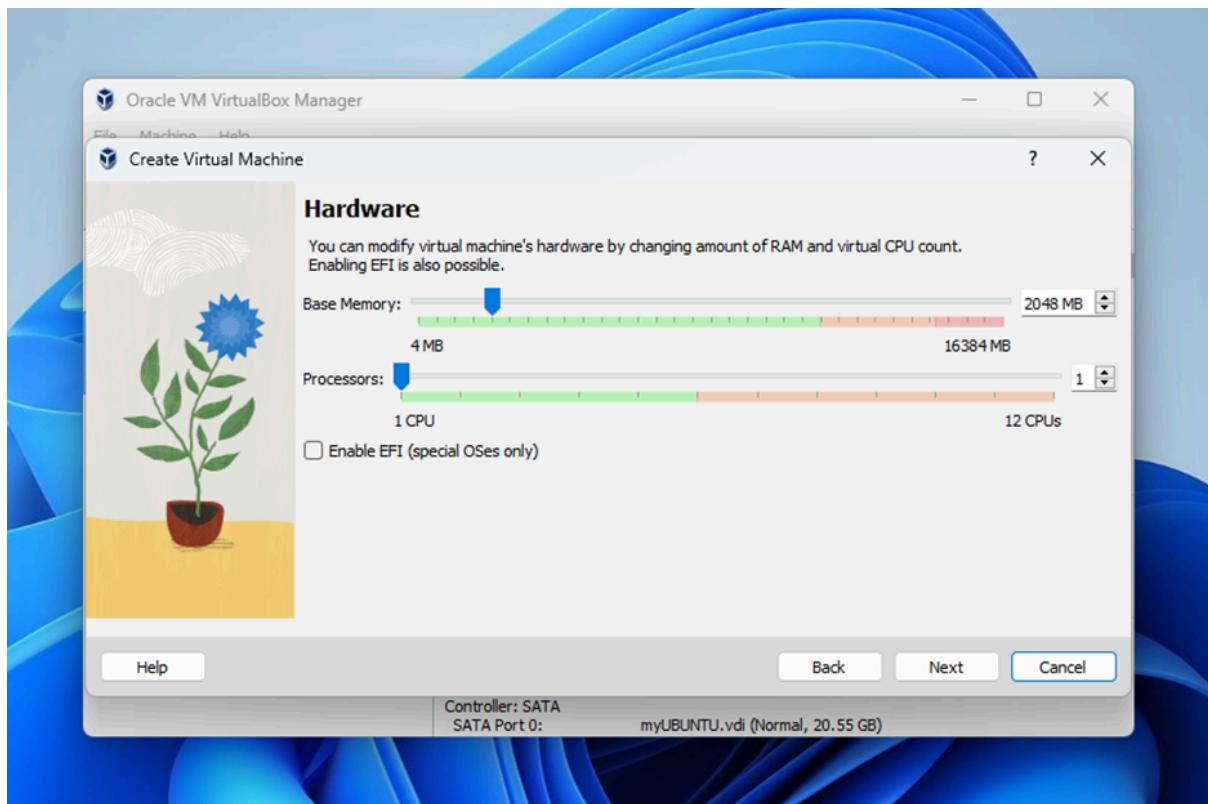
3. Choose the ISO file of the OS



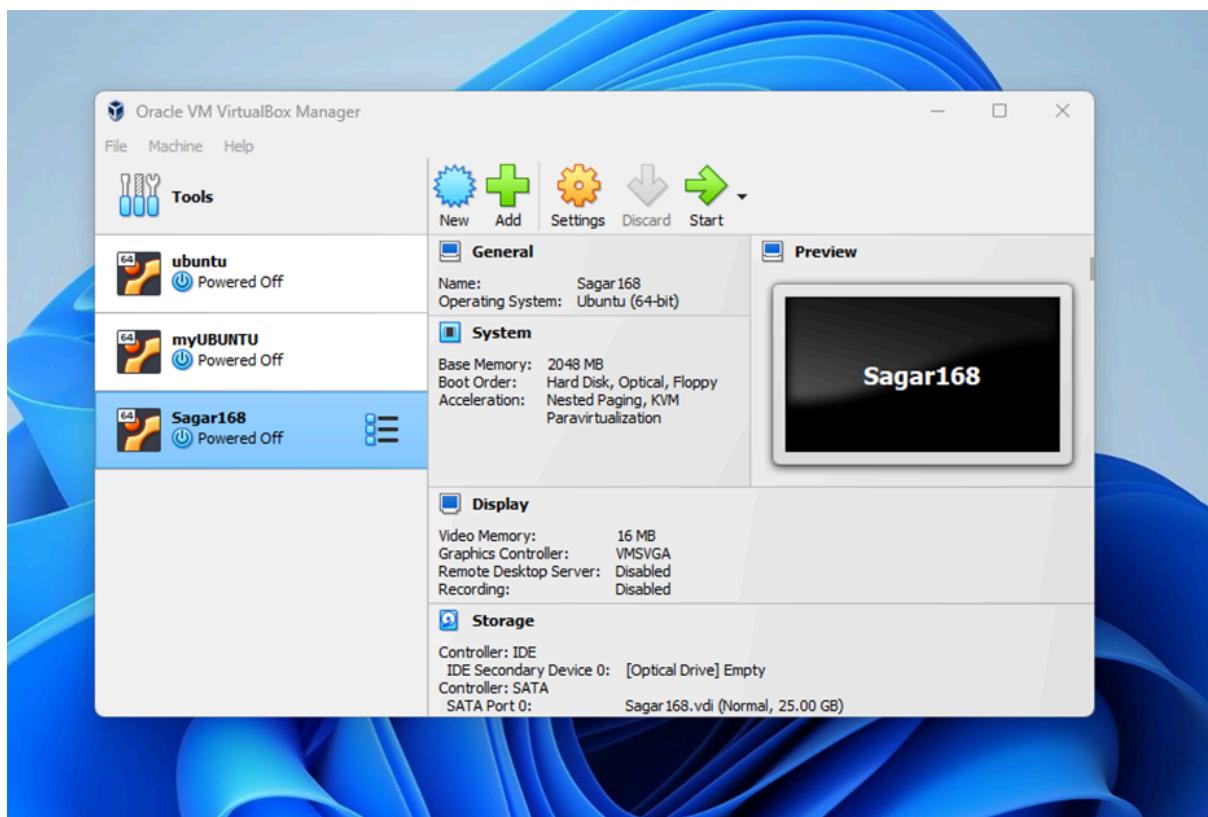
4. Complete the Setup



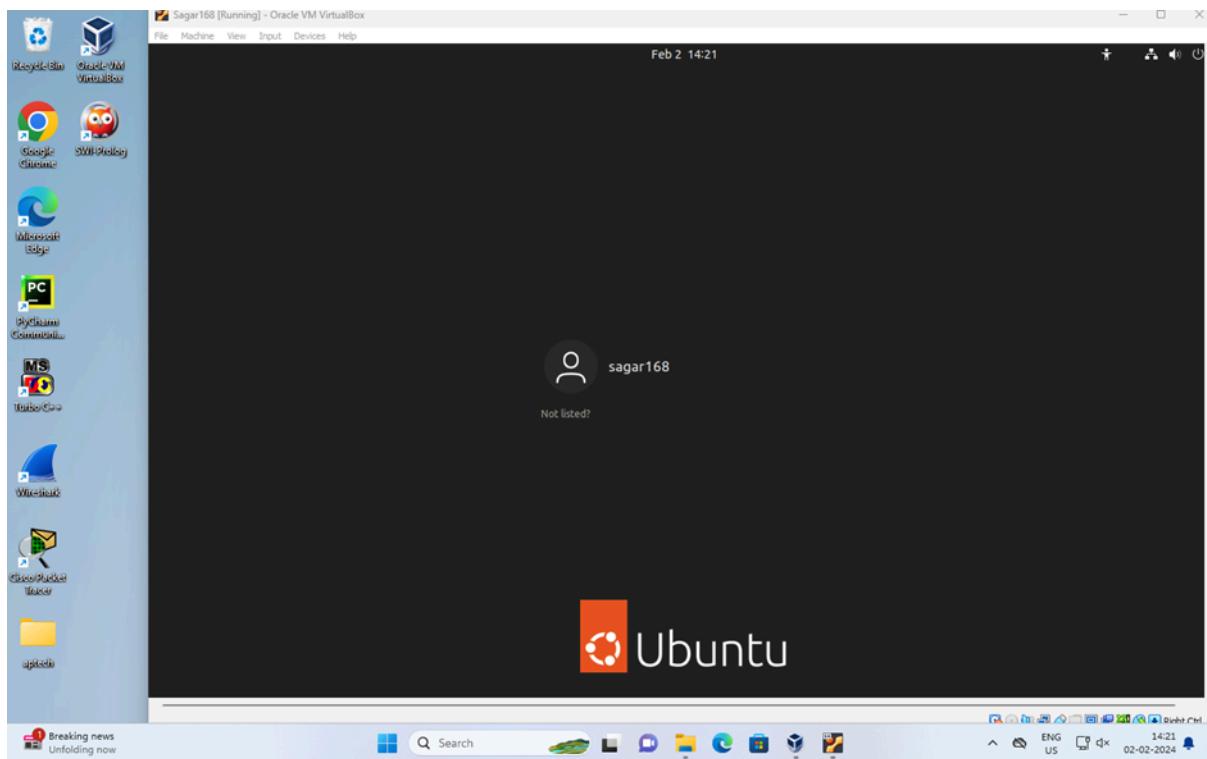
5. Select Ram and Virtual hard disk



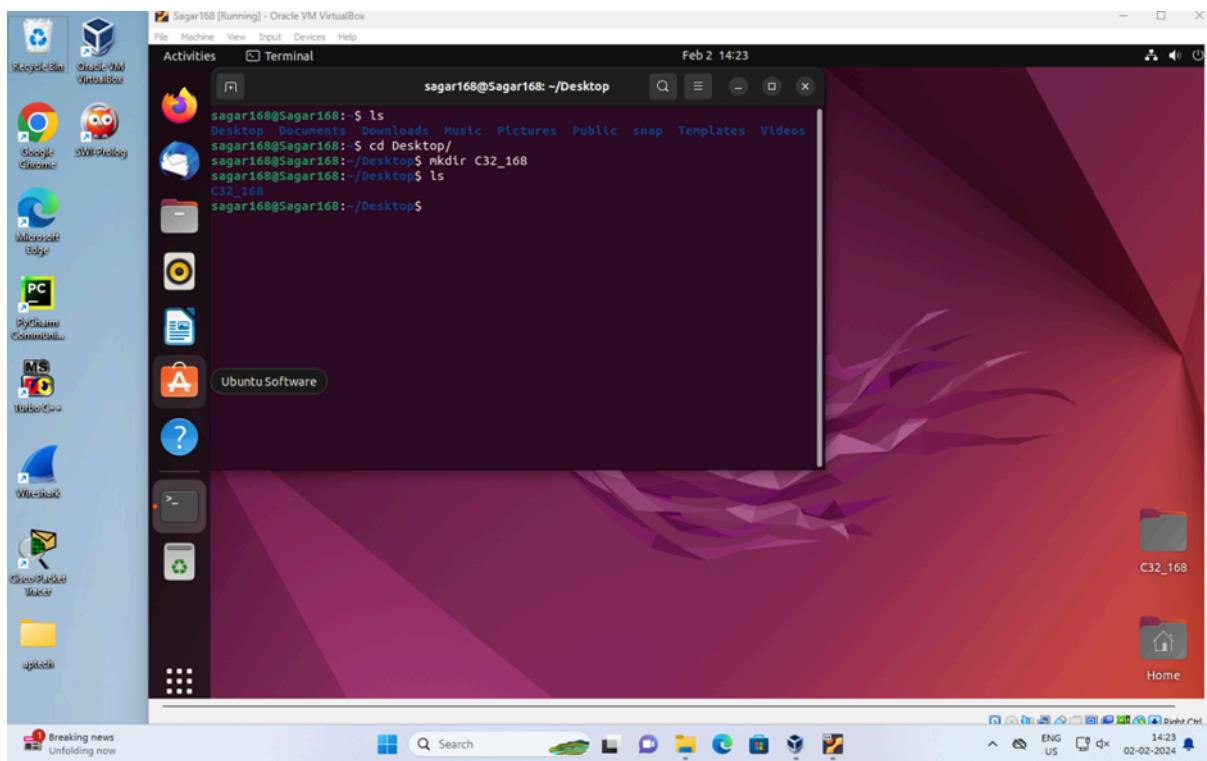
6. Click on Start and it will run Ubuntu OS



7. After The Device boots enter credentials



8. Perform Linux Commands



Experiment No : 3

Aim : To study and implement Bare-metal Virtualization with Xen.

Theory :

- Functions performed by Bare-metal hypervisors
 - ⇒ Bare-metal hypervisor, also known as Type-2 hypervisors, operate directly on the hardware without the need for an underlying operating system.

- Their main functions include:

1) Resource Management

- ⇒ Bare-metal hypervisors allocate hardware resources such as CPU, memory, and storage to multiple virtual machines running on the same physical host.

2) Isolation

- ⇒ They ensure strong isolation between VMs, preventing interference and providing security by separating the execution environments.

3) Performance Optimization

- ⇒ Bare-metal hypervisors optimize performance by directly interacting with hardware, reducing the overhead associated with a host operating system.

4) Hardware Virtualization

→ They enable multiple operating systems to run on a single physical machine by creating virtual instances of the hardware components.

• Compare Hosted and Bare-metal hypervisors:

Hosted Hypervisors	Bare-Metal Hypervisors
1) Operates on the conventional operating system.	1) Runs directly on the system where VMs function.
2) OS virtualization.	2) Hardware virtualization.
3) Functions as the application on the host.	3) Guest OS and applications runs on the hypervisor.
4) It is less scalable than the bare-metal hypervisor.	4) It is more scalable than the hosted hypervisor.
5) Easier than a type 1 setup because of the existing OS.	5) Simpler, if your hardware supports the application.
6) Speed is slower than bare-metal.	6) Speed is faster than hosted hypervisors.
7) For e.g., VMware ESXi, Microsoft Hyper V	7) For e.g., VMware ESXi, Microsoft Hyper V, Citrix XenServer.
Player Microsoft Virtual PC	
Sun's VirtualBox	

- Horizontal and Vertical Scaling:

- In horizontal scaling, increasing the number of instances of an application or services across multiple machines or nodes.
- For e.g., Adding more servers to a web application to handle increased user traffic.
- Vertical scaling can be defined as increasing the capacity of a single machine by adding more resources. (CPU, RAM)
- For e.g., Upgrading a server's RAM or CPU to handle a growing database.

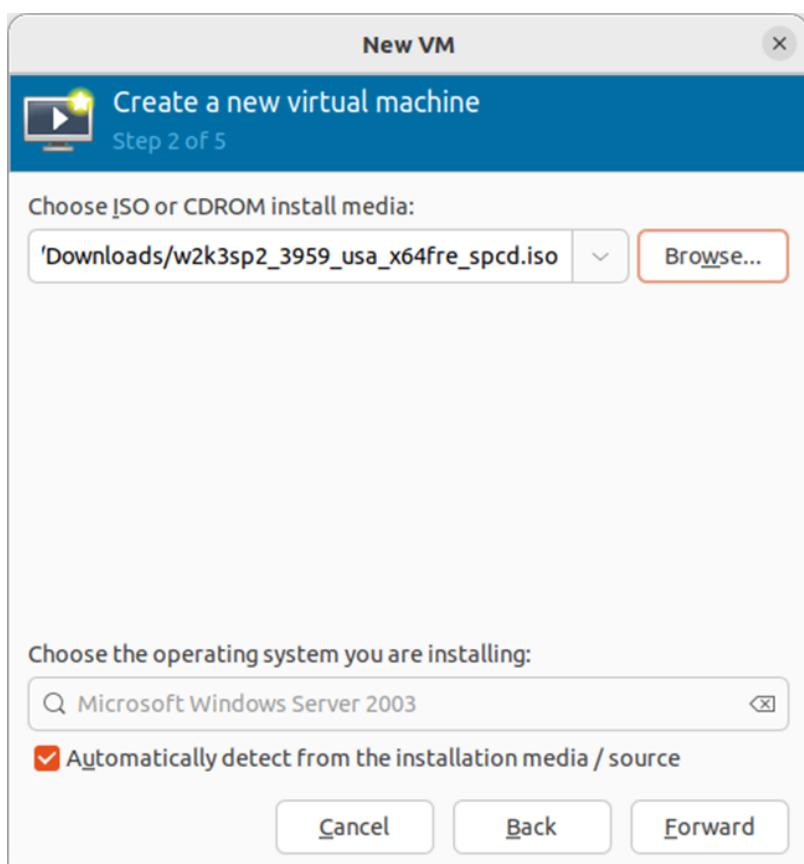
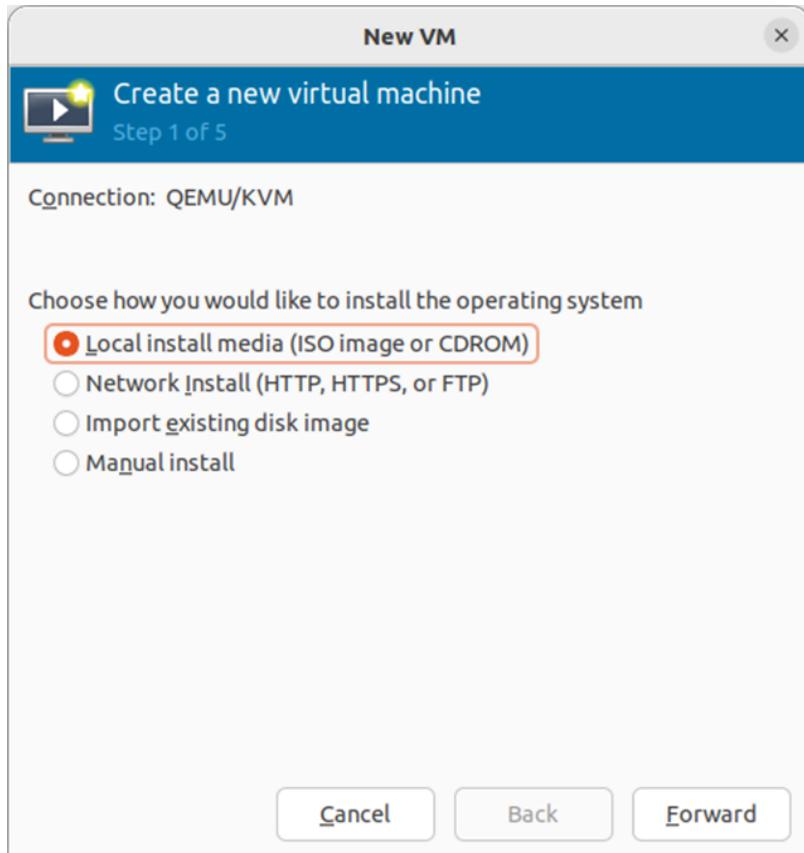
- Auto Scaling

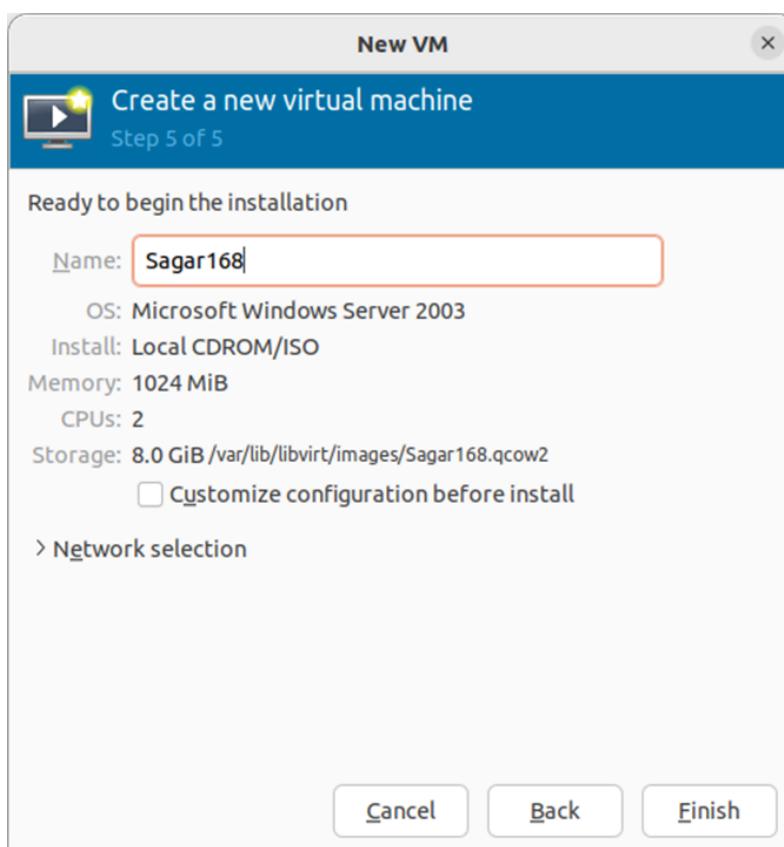
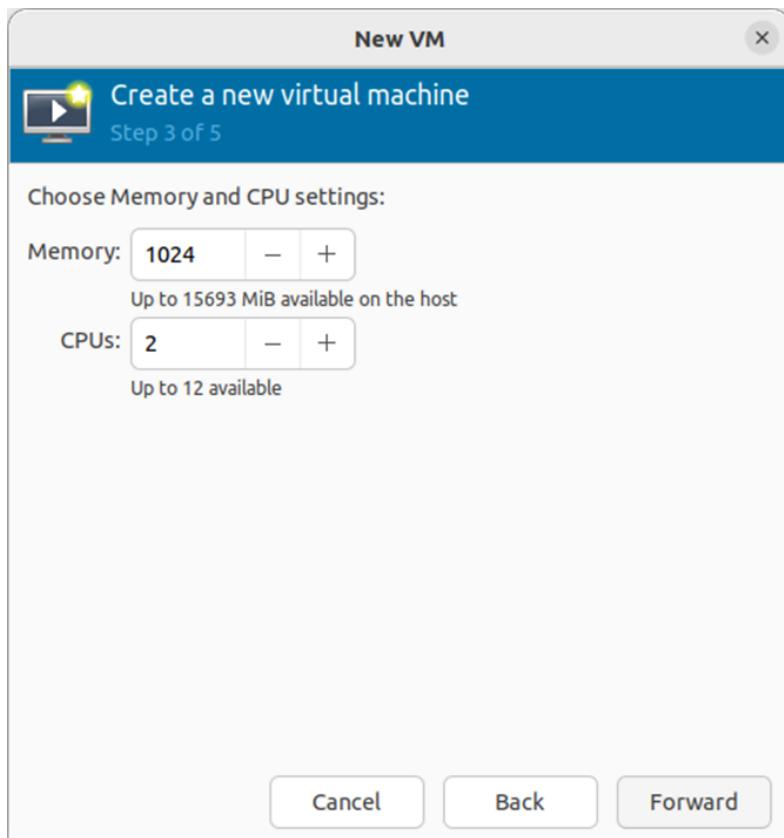
- Automatically adjusting the number of compute resources based on the demand, ensuring optimal performance and cost efficiency.
- For e.g., scaling up instances during peak hours and scaling down during low traffic.

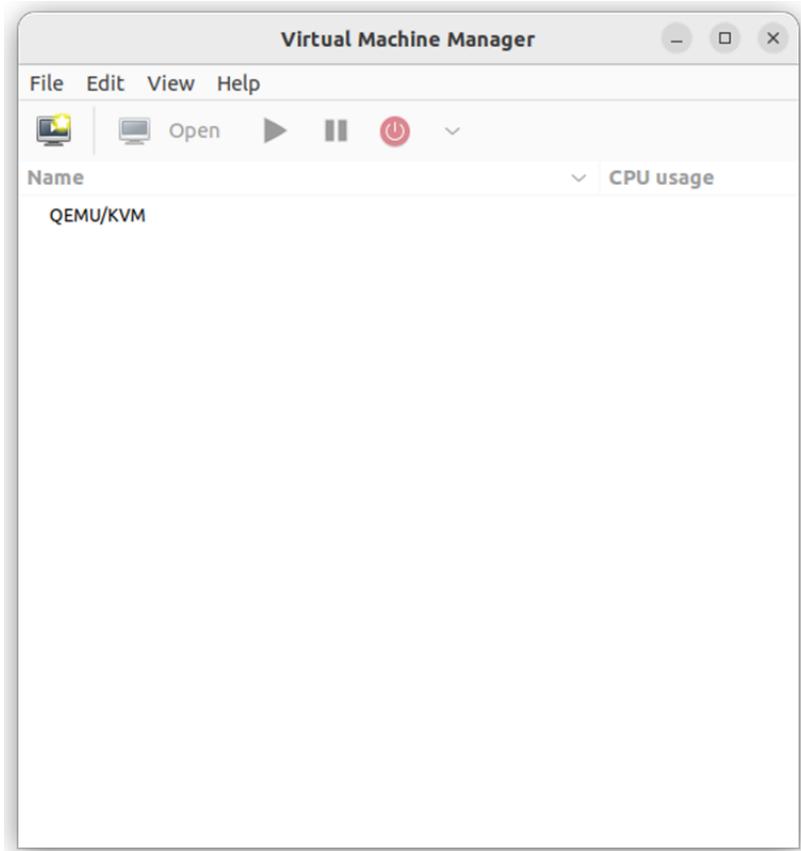
- Load Balancing

- Distributing incoming network traffic across multiple servers to ensure no single server is overwhelmed, optimizing resource utilization and improve reliability.
- For e.g., Distributing web traffic among multiple servers to prevent overloading any single server.

(A) ST
26/2/24

Output:





```
root@LAB301PC07: /home/student
Note, selecting 'qemu-system-x86' instead of 'qemu-kvm'.
bridge-utils is already the newest version (1.7-1ubuntu3).
libvirt-clients is already the newest version (8.0.0-1ubuntu7.8).
libvirt-daemon-system is already the newest version (8.0.0-1ubuntu7.8).
qemu-system-x86 is already the newest version (1:6.2+dfsg-2ubuntu6.16).
The following packages were automatically installed and are no longer required:
  libqt5help5 libqt5sql5 libqt5sql5-sqlite libqt5xml5 libsdl-ttf2.0-0
  linux-image-6.2.0-39-generic linux-modules-6.2.0-39-generic
  linux-modules-extra-6.2.0-39-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 57 not upgraded.
root@LAB301PC07:/home/student# virsh -c qemu:///system list
  Id  Name      State
  --
root@LAB301PC07:/home/student# virt-manager
root@LAB301PC07:/home/student# virsh -c qemu:///system list
virsh-c: command not found
root@LAB301PC07:/home/student# virsh -c qemu:///system list
  Id  Name      State
  --
  1  Sagar168   running
root@LAB301PC07:/home/student#
```

Experiment No : 4

Aim : To study and implement Infrastructure as a Service using AWS.

Theory : Building a system based on cloud computing.

- The objective of this experiment is to study and implement Infrastructure as a Service (IaaS) using AWS.

- The focus will be on Amazon EC2 including the creation and management of instances, Amazon Machine Images (AMI), various types of EC2 computing instances, Elastic IP address and accessing Windows virtual machines using Remote Desktop Protocol (RDP).

1) Amazon EC2

→ Amazon EC2 is a web service that provides resizable compute capacity in the cloud.

- It is designed to make web-scale cloud computing easier for developers.
- EC2 instances are virtual servers in the cloud that can run applications.

2) Amazon Machine Images (AMI)

→ An AMI is a pre-configured virtual machine image, which is used to create EC2 instances.

- It contains necessary information to launch an instances, including the OS, application server and applications.

3) Types of EC2 Computing Instances:

⇒ EC2 instances come in various types optimized for different use cases, including compute-optimized, memory-optimized, storage-optimized and GPU instances.

For e.g., t2.micro, m3.large, etc.

4) Elastic IP address

⇒ An elastic IP address is a static IPv4 address designed for dynamic cloud computing.

It can be associated with EC2 instances, providing a consistent IP address even if the instance is stopped and started.

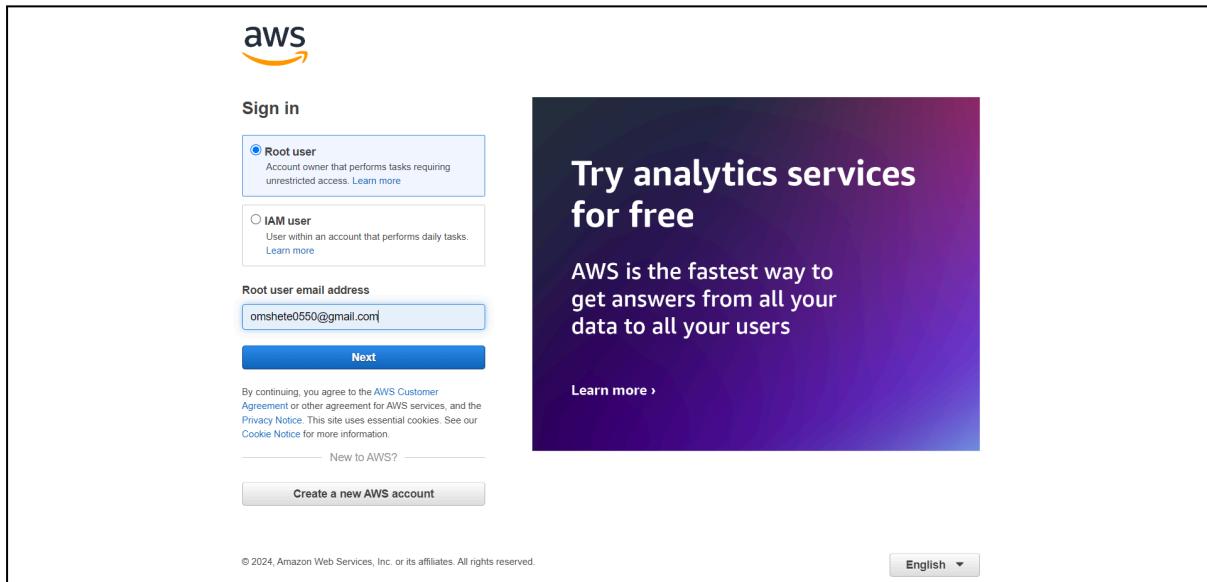
5) Remote desktop Protocol (RDP)

⇒ RDP is a proprietary protocol developed by Microsoft, which provides a user with a graphical interface to connect to another computer over a network connection.

(A) ~~Self Review~~

Output:

1. Create an AWS account and Login



2. On Homepage Select Launch a Virtual Machine

The screenshot shows the AWS homepage. The main navigation bar includes 'Services', a search bar, and a user profile 'omshete0550'. The homepage features several sections: 'Welcome to AWS' (with links to Getting started with AWS, Training and certification, and What's new with AWS?), 'AWS Health' (showing 0 open issues, 0 scheduled changes, and 0 other notifications), 'Cost and usage' (showing current month costs at \$0.00 and a bar chart for EC2 - Other showing costs for Nov 23, Jan 24, and Mar 24), 'Build a solution' (with links to Launch a virtual machine, Start migrating to AWS, Register a domain, Host a static web app, and Explore AWS), 'Explore AWS' (with links to AWS Support and Security), and 'Security' (Region: Europe (Stockholm)). At the bottom, there are links for CloudShell, Feedback, and various legal and preference buttons.

3. Select Launch Instance

The screenshot shows the AWS EC2 Dashboard for the Europe (Stockholm) Region. The left sidebar includes sections for Instances, Images, and Elastic Block Store. The main area displays a summary of resources: 0 Instances (running), 0 Auto Scaling Groups, 0 Dedicated Hosts, 0 Elastic IPs, 0 Instances, 0 Key pairs, 0 Load balancers, 0 Placement groups, 1 Security groups, 0 Snapshots, and 0 Volumes. Below this, there's a 'Launch instance' section with a large orange 'Launch instance' button and a 'Migrate a server' option. To the right, the 'Service health' section shows the service is operating normally. The bottom right corner contains account attributes and links to AWS Free Tier offers.

4. Select the Instance Specifications

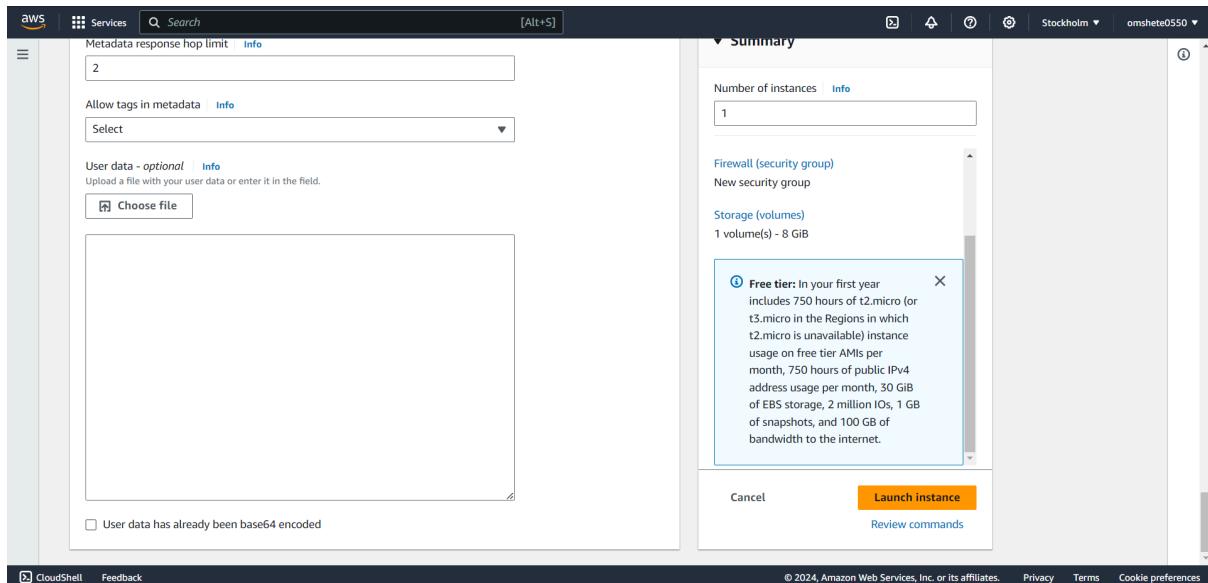
The screenshot shows the 'Launch an instance' wizard. The first step, 'Name and tags', has a 'Name' field containing 'My Server'. The second step, 'Application and OS Images (Amazon Machine Image)', shows a search bar and a grid of quick start AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. A tooltip for the Ubuntu AMI provides information about the free tier. The third step, 'Summary', shows 1 instance selected, the software image (AMI) as Canonical, Ubuntu, 22.04 LTS, the virtual server type as t3.micro, and the storage as 1 volume(s) - 8 GiB. The 'Launch instance' button is highlighted in orange.

The screenshots show the AWS CloudFormation console interface across three different stages of configuration:

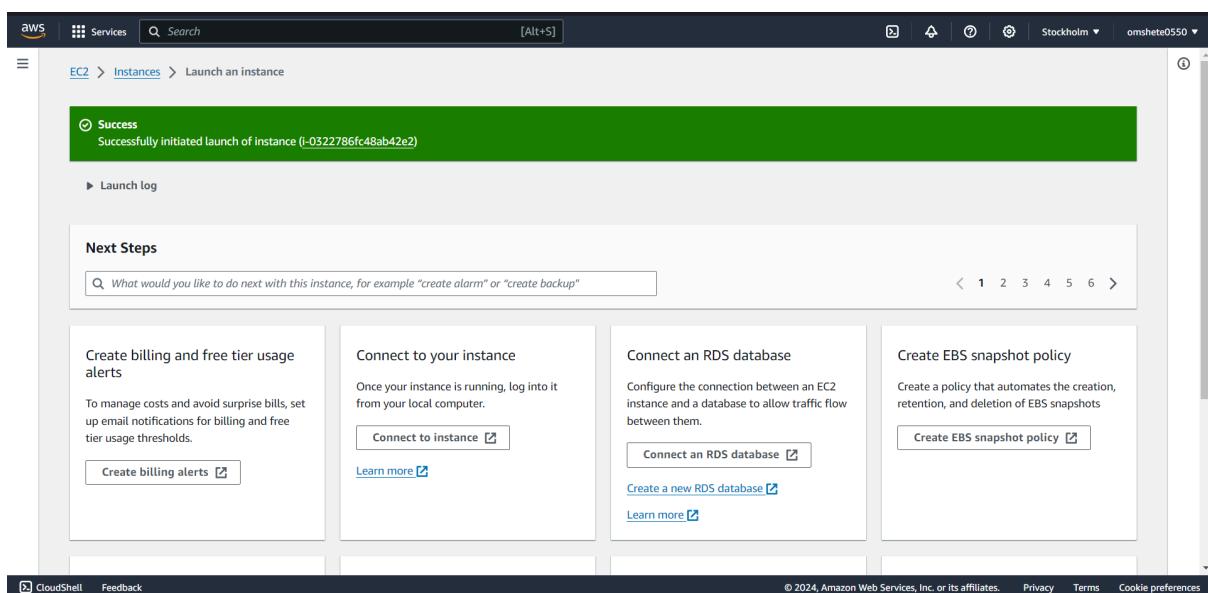
- Screenshot 1 (Top):** Shows the selection of an Amazon Machine Image (AMI) for a new instance type. The selected AMI is "Ubuntu Server 22.04 LTS (HVM), SSD Volume Type". It includes details like AMI ID (ami-0914547665e6a707c), Virtualization type (hvm), and Root device type (ebs). A tooltip for the "Free tier eligible" status is shown.
- Screenshot 2 (Middle):** Shows the configuration of a security group. It includes options for allowing HTTPS and HTTP traffic from the internet. A tooltip for the "Free tier" usage is shown.
- Screenshot 3 (Bottom):** Shows the configuration of storage volumes. It includes setting up a root volume (8 GiB gp2) and a tooltip for free tier EBS usage.

In all three screenshots, the right-hand panel displays a summary of the instance configuration, including the number of instances (1), software image (Ubuntu Server 22.04 LTS), virtual server type (t3.micro), firewall (New security group), and storage (1 volume(s) - 8 GiB). The "Launch instance" button is prominently displayed at the bottom of each screen.

5. Click on Launch Instance on the right



6. Instance Created Successfully



7. On EC2 Dashboard Click Instances(running)

Resources

You are using the following Amazon EC2 resources in the Europe (Stockholm) Region:

Instances (running)	1	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	1	Key pairs	0
Load balancers	0	Placement groups	0	Security groups	2
Snapshots	0	Volumes	1		

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Service health

AWS Health Dashboard

Region: Europe (Stockholm)

Status: This service is operating normally.

Zones

Zone name	Zone ID

Account attributes

Default VPC: vpc-02ed8cfea6738bce

Settings

Data protection and security

Zones

EC2 Control Groups

8. Click on Connect

Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

Instance state = running | All states

Details

Instance: i-0322786fc48ab42e2 (My Server)

Details | Status and alarms New | Monitoring | Security | Networking | Storage | Tags

Instance summary

Instance ID: i-0322786fc48ab42e2 (My Server)

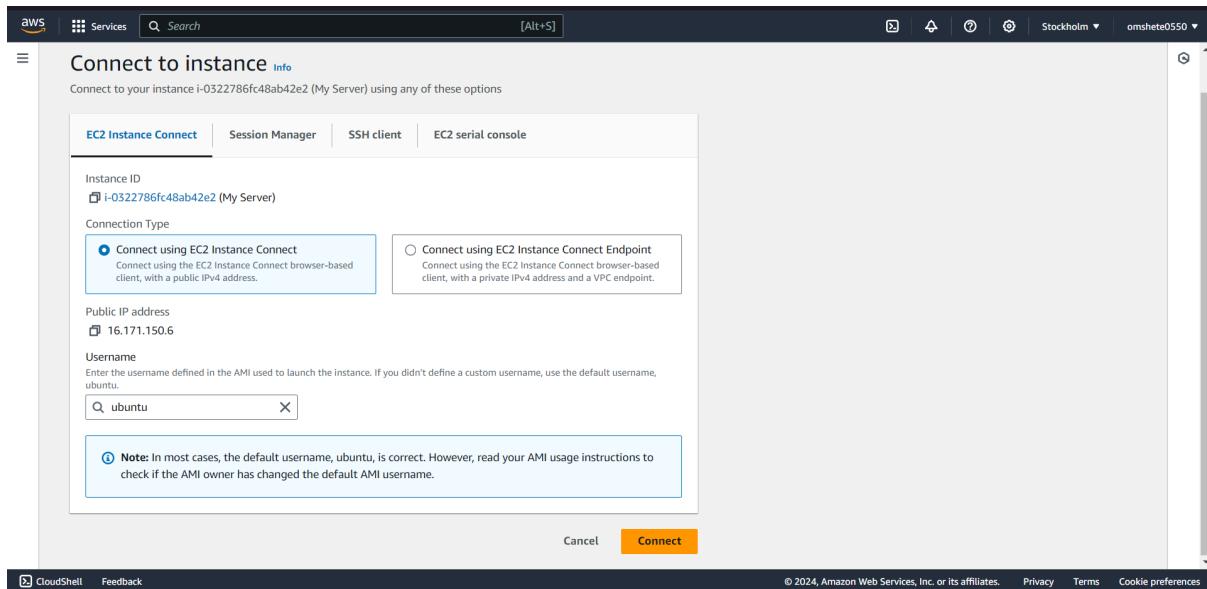
Public IPv4 address: 16.171.150.6 [open address]

Instance state: Running

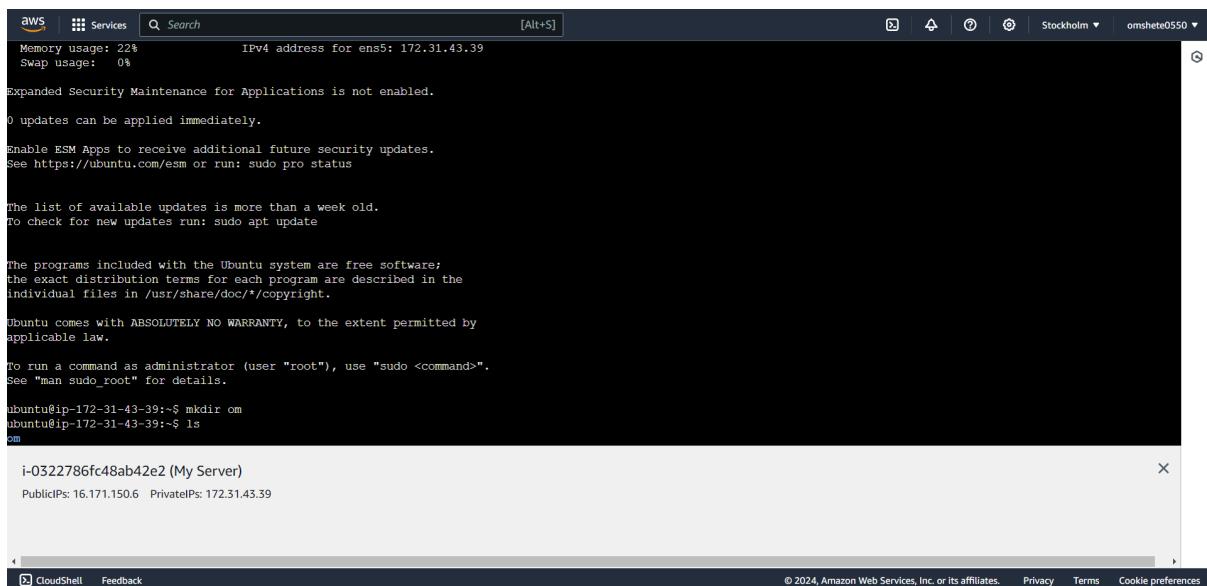
Private IP DNS name (IPv4 only): ip-172-31-43-39.eu-north-1.compute.internal

Private IPv4 addresses: 172.31.43.39

Public IPv4 DNS: ec2-16-171-150-6.eu-north-1.compute.amazonaws.com [open address]



9. After Connecting to the Instance you can run any Linux command



10. Now Go to EC2 Dashboard and Select Instance State

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Instances (with sub-options like Instances, Instance Types, Launch Templates, etc.), Images, and Elastic Block Store. The main area displays a table of instances. One instance is selected, labeled 'My Server' with the ID i-0322786fc48ab42e2. The instance is shown as 'Running' with the type t3.micro. A context menu is open over this instance, with the 'Terminate instance' option highlighted in orange.

11. Select Terminate Instance

The screenshot shows the AWS EC2 Instances page with the same setup as the previous screenshot. A modal dialog titled 'Terminate instance?' is displayed. It contains a warning message about EBS-backed instances losing root EBS volume data if termination protection is disabled. Below this, it asks if the user wants to terminate the instance. The 'Termination protection' setting is shown as 'Disabled'. At the bottom right of the dialog is a large orange 'Terminate' button.

The screenshot shows the AWS EC2 Instances page. At the top, a green banner indicates "Successfully terminated i-0322786fc48ab42e2". The main table lists one instance:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
My Server	i-0322786fc48ab42e2	Shutting-d...	t3.micro	2/2 checks passed	View alarms +	eu-north-1b	ec2-16-17-

Below the table, the "Details" tab is selected for the instance i-0322786fc48ab42e2. The "Instance summary" section shows the following details:

- Instance ID: i-0322786fc48ab42e2 (My Server)
- IPv6 address: -
- Hostname type: IP name: ip-172-31-43-39.eu-north-1.compute.internal
- Public IPv4 address: 16.171.150.6 [open address]
- Instance state: Shutting-down
- Private IP DNS name (IPv4 only): ip-172-31-43-39.eu-north-1.compute.internal
- Private IPv4 addresses: 172.31.43.39
- Public IPv4 DNS: ec2-16-171-150-6.eu-north-1.compute.amazonaws.com [open address]

The screenshot shows the AWS EC2 Dashboard. On the left, the navigation menu includes "Instances", "Launch instance", and "Launch a server". The main area displays the following information:

- Resources**: A summary of EC2 resources in the Europe (Stockholm) Region:

Instances (running)	0	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	1	Key pairs	0
Load balancers	0	Placement groups	0	Security groups	2
Snapshots	0	Volumes	1		
- Launch instance**: A button to start a new instance.
- Service health**: Shows the status of the AWS Health Dashboard.
- EC2 Free Tier**: Information about free tier offers.
- Account attributes**: Details about the default VPC and settings.

Experiment No: 5

Aim: To study and implement a platform as a service using AWS Elastic Beanstalk service.

Theory: It abstracts away the complexities of the infrastructure management, allowing developers to focus on writing code while AWS handles provisioning, load balancing, auto-scaling and other operational tasks.

- Amazon AWS Elastic Beanstalk (EBS)
- EBS is a cloud computing service provided by AWS that simplifies the management of scaling of web applications as services.
- It abstracts away the complexities of the infrastructure management, allowing developers to focus on writing code while AWS handles provisioning, load balancing, auto-scaling and other operational tasks.
- Key Features include:
 - 1) Easy development process, maintenance
 - 2) Developers can simplify uploading their application code as EBS automatically handles the deployment process, including provisioning the necessary resources.

1) Easy development process, maintenance

→ Developers can simplify uploading their application code as EBS automatically handles the deployment process, including provisioning the necessary resources.

2) Auto-scaling

→ EBS can automatically scale the number of EC2 instances based on application load, ensuring that the application remains available and optimized.

Cloud frameworks

3) Monitoring & Logging: One of the main features of EBS is its built-in monitoring and logging capabilities.

EBS provides developers with built-in monitoring and logging capabilities, allowing them to monitor application health and troubleshoot issues easily.

(283) ~~What are the main benefits of using EBS?~~

• Languages & Frameworks supported by EBS

• Has support for multiple languages and frameworks

1) AWS Elastic Beanstalk supports a variety of programming languages and frameworks, providing flexibility for developers to deploy their application using their technology stack.

2) EBS provides support for Docker containers, allowing developers to package their applications and dependencies into Docker containers, allowing developers to package their applications.

3) Ruby applications allow developers to develop web applications built with the Ruby on Rails framework.

4) Go applications are supported by EBS, enabling developers to deploy web applications written in the Go language.

- Elastic Load Balancing (ELB)

⇒ ELB is provided by AWS that automatically distributes incoming application traffic across multiple targets such as EC2 instances or Lambda functions to ensure optimum performance, availability and fault tolerance of the applications.

- Features include:

1) Distribution of incoming traffic

⇒ ELB automatically distributes incoming application traffic across multiple targets, ensuring that each target receives a balanced load of requests.

2) Scalability

⇒ ELB can automatically scale its request handling capacity in response to changes in incoming traffic.

3) High Availability

⇒ ELB enhances the availability of the application by automatically detecting unhealthy targets and rerouting traffic to healthy targets.

- Comparison between EC2 and PBS



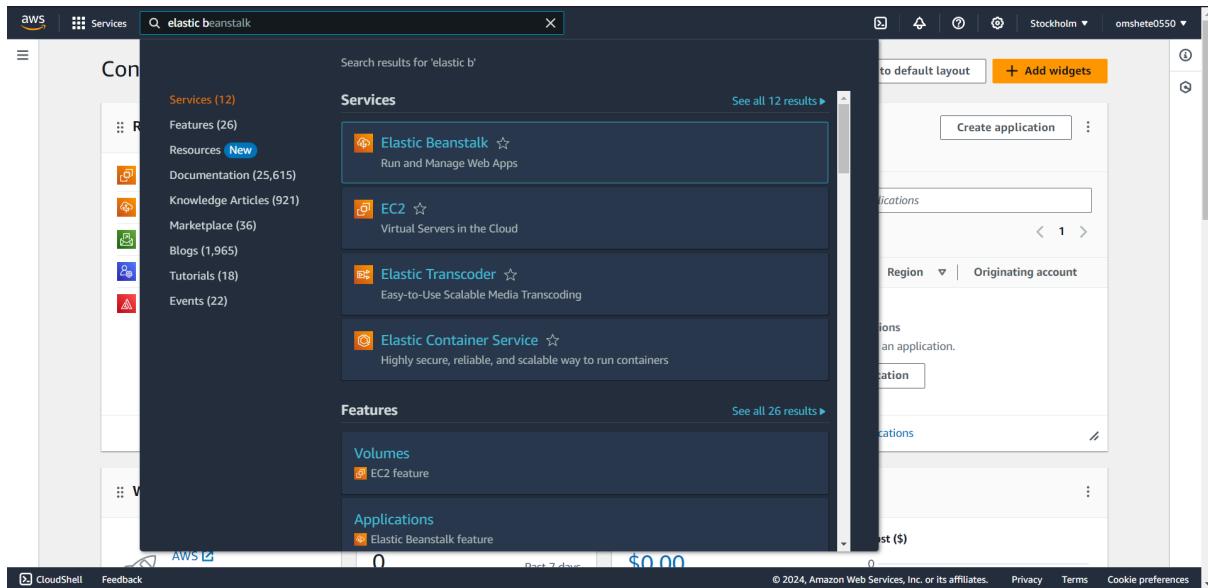
EC2 (AWS) provisioned vs EBS provisioned

- | | |
|---|--|
| <p>1) Provides full control over the virtual resources, allowing users to configure the EC2 instances and applications according to their needs.</p> <p>2) Requires manual configuration and management of all infrastructure components.</p> <p>3) Users are responsible for manually scaling EC2 instances application load.</p> <p>4) Users pay for the EC2 instances, other resources provisioned manually with pricing based on the interface types.</p> | <p>1) Abstracts away the infrastructure management, automatically provisioning, load balancing, auto scaling and application health monitoring.</p> <p>2) Simplifies deployment and management tasks by automatically provisioning and configuration.</p> <p>3) Automatically handles scaling based on the application load.</p> <p>4) Users pay for the underlying AWS resources provisioned by EBS along with any additional service fees.</p> |
|---|--|

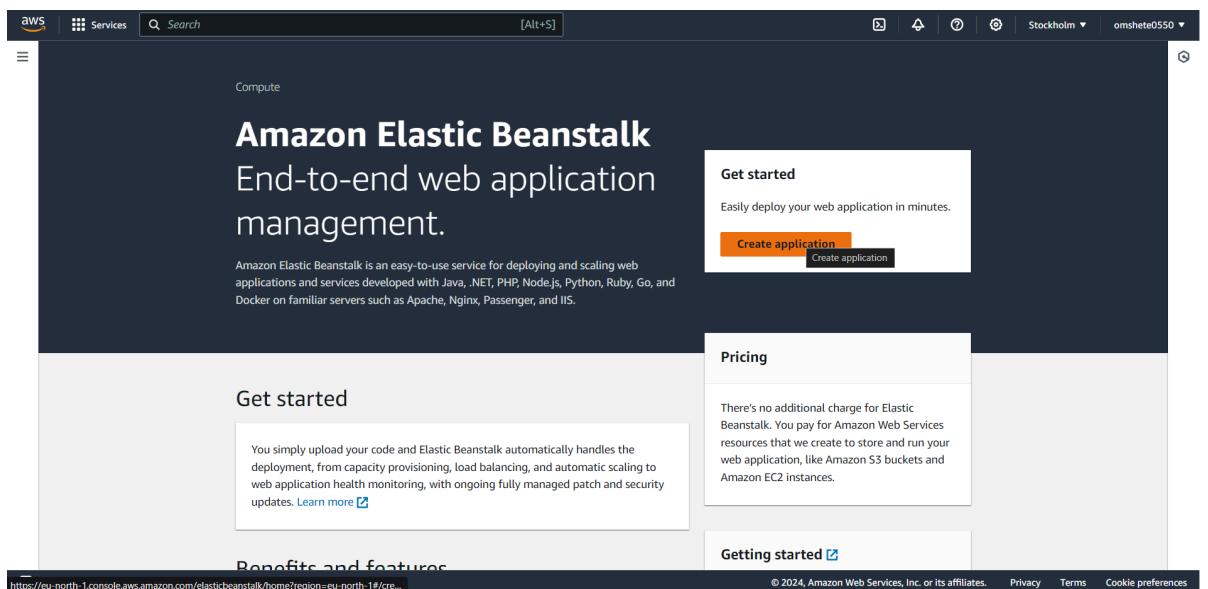
283 hrs 829 minutes (approx)

Output:

1. Login to the AWS console and go to Elastic Beanstalk



2. Click on Create Application



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

The screenshots show the configuration steps for creating an application on AWS Elastic Beanstalk:

- Environment tier:** Set to "Web server environment".
- Application information:** Application name is "MyApp".
- Application tags (optional):** A single tag named "My Server" is added.
- Platform:** Platform type is "Managed platform" (Node.js, Node.js 20 running on 64bit Amazon Linux 2023, 6.1.2 (Recommended)).
- Application code:** Sample application selected.
- Presets:** Configuration preset is "Single instance (free tier eligible)".

Configure service access [Info](#)

Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

Create and use new service role
 Use an existing service role

Existing service roles

Choose an existing IAM role for Elastic Beanstalk to assume as a service role. The existing IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role [C](#)

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

mykeypair [C](#)

EC2 instance profile

Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

[View permission details](#)

Cancel Skip to review Previous Next

4. Enter submit to create the application

Activated	100	Percentage
Command timeout	Deployment policy	Health threshold
600	AllAtOnce	Ok
Ignore health check	Instance replacement	false
Platform software		
Lifecycle	Log streaming	Logs retention
false	Deactivated	7
Rotate logs	Update level	X-Ray enabled
Deactivated	minor	Deactivated
Environment properties		
Key	Value	
GRADLE_HOME	/usr/local/gradle	
M2	/usr/local/apache-maven/bin	
M2_HOME	/usr/local/apache-maven	

Cancel Previous Submit

This will take a few minutes.

The screenshot shows the AWS Elastic Beanstalk console. The left sidebar shows 'Application: MyApp' and 'Environment: MyApp-env'. The main area displays the 'MyApp-env' environment overview, which includes sections for Environment overview and Platform. The Platform section shows the following details:

- Platform: Node.js 20 running on 64bit Amazon Linux 2023/6.1.2
- Running version: -
- Platform state: Supported

Below the overview, there are tabs for Events, Health, Logs, Monitoring, Alarms, Managed updates, and Tags. The Events tab shows 2 events, with a search bar and navigation controls.

5. Click on Environments -> Check Environment's health until it becomes 'OK'

The screenshot shows the AWS Elastic Beanstalk console for environment 'MyApp-env-1'. The main area displays the 'MyApp-env-1' environment overview, which includes sections for Environment overview and Platform. The Platform section shows the following details:

- Platform: Node.js 20 running on 64bit Amazon Linux 2023/6.1.2
- Running version: -
- Platform state: Supported

Below the overview, there are tabs for Events, Health, Logs, Monitoring, Alarms, Managed updates, and Tags. The Events tab shows 6 events, with a search bar and navigation controls. The events listed are:

Time	Type	Details
April 8, 2024 11:52:02 (UTC+5:30)	INFO	Deleting SNS topic for environment MyApp-env-1.
April 8, 2024 11:52:01 (UTC+5:30)	ERROR	Failed to launch environment.

Experiment No. 6

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

Theory:

- Explain the concept of cloud storage.
⇒ Cloud storage is made of computer data storage in which digital data is stored on servers in outside location.
- The servers are maintained by third party provider who is responsible for hosting, managing and security data stored on its infrastructure.
- Explain own cloud and its features.
⇒ Own cloud is self hosted, open source file syncing and sharing server like dropbox, google drive, box and more for the big organization, owncloud give you access to your file, calendar, contacts and other data.
- You can sync all between your own sync device and store file with others.
- Features
⇒
 - 1) Access your data
 - 2) Sync your data
 - 3) Share your data
 - 4) Versioning
 - 5) Encryption

- 6) Drag-and-Drop file Upload, photo or file.
- 7) Viewer for pdf files or presentation.
- 8) Application API.
- 9) Application store.

- Advantages:
 - 1) Easy to learn and implement.
 - 2) Cost efficient, less initial cost and less maintenance.
 - 3) Scalability.
 - 4) Accessibility, can access anywhere, anytime.
 - 5) Automatic updates.
 - 6) Redundancy.
 - 7) Collaborator.
 - 8) Flexibility.
 - 9) Managed security.

Limitations

- 1) Dependency on Internet connection.
- 2) Security concerns, hacking.
- 3) Data transfer speed may not be fast.
- 4) Service downtime.
- 5) Regulatory compliances.
- 6) Customization constraints.
- 7) Limited control.
- 8) Data location.

• Explain different types of storage like object storage, block level storage.

1) Object storage:



Transport : TCP / IP

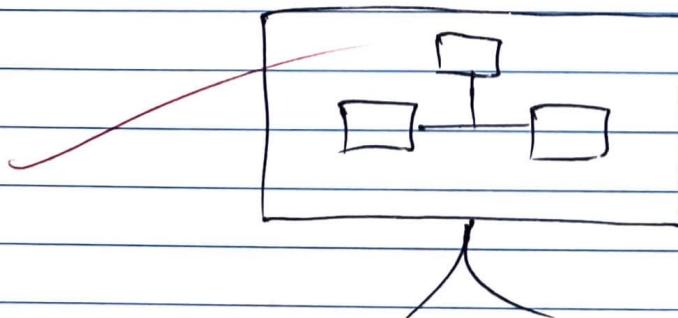
Interface : HTTP / REST

Use case : Easily scaling with no limits accessible across LAN and WAN

has no database layer : no transactions

DEFINITION : It's a way architecture for storing data which section data into unit objects.

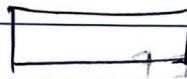
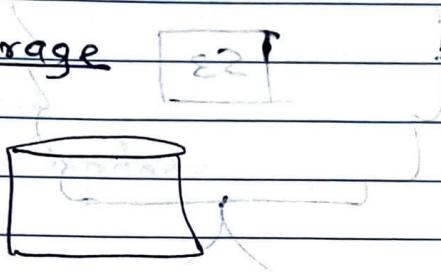
2) File storage:



- Transport : TCP / IP
- Interface : NFS / SMB
- Use case : Good performance, file sharing, etc.
- The data is stored in file, files are organized in folders and folders are organized into hierarchy of direction of sub direction.

3) Block storage

⇒



FC / FTT : Transport
F2FS / GPT : Structure

limit on data transfer : 32GB/s

- Transport : FC, serial connection

- Interface : Direct attached or SAN

- Use cases : Low latency, best for structure data, time and stable writes, voids

(A)

GP : supports SIF (S)

S13/2024



Output:

1. Click on Create bucket

The screenshot shows the AWS Cloud Home page with a search bar at the top containing 'buckets'. The search results are displayed under the 'Features' section, with 'Buckets' being the first result. Other features listed include Object storage, Multi-Region Access Points, and Account Block Public Access settings. To the right of the search results, there is a sidebar for 'myApplications' with a 'Create application' button and a 'Find applications' search bar.

The screenshot shows the 'Amazon S3 > Buckets' page. It displays an 'Account snapshot' with a link to the Storage Lens dashboard. Below this, there are tabs for 'General purpose buckets' and 'Directory buckets', with 'General purpose buckets' currently selected. It shows two buckets: 'elasticbeanstalk-eu-north-1-' and 'elasticbeanstalk-us-east-1-'. Each bucket entry includes its name, AWS Region, IAM Access Analyzer status, and creation date. A 'Create bucket' button is located at the top right of the bucket list table. The left sidebar of the S3 console is visible, showing options like Buckets, Access Grants, and Storage Lens.

2. Give Bucket name & select region for storage

The screenshot shows the 'Create bucket' page in the AWS Management Console. The 'General configuration' section is visible, containing fields for 'Bucket name' (set to 'myawsbucketcc16') and 'AWS Region' (set to 'Europe (Stockholm) eu-north-1'). The 'Bucket type' section offers two options: 'General purpose' (selected) and 'Directory - New'. The 'Object Ownership' section is partially visible at the bottom.

3. Keep object ownership setting as ACLs Disabled as by-default

The screenshot shows the 'Create bucket' page in the AWS Management Console, focusing on the 'Object Ownership' section. It displays two options: 'ACLs disabled (recommended)' (selected) and 'ACLs enabled'. The 'Object Ownership' section also indicates 'Bucket owner enforced'. The top navigation bar shows the AWS logo, services menu, search bar, and user information for 'omshete0550'.

4. Disable block all public access checkbox

The screenshot shows the 'Block Public Access settings for this bucket' section. At the top, there is a note about public access being granted through ACLs, bucket policies, access point policies, or all. Below this, the 'Block all public access' checkbox is unchecked. A detailed description follows, explaining that turning it on is equivalent to enabling four other settings. The four settings are listed as sub-options under the main checkbox:

- Block public access to buckets and objects granted through new access control lists (ACLs)**: Grants public access permissions applied to newly added buckets or objects, and prevents creation of new public access ACLs for existing buckets and objects.
- Block public access to buckets and objects granted through any access control lists (ACLs)**: Ignores all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**: Blocks new bucket and access point policies that grant public access to buckets and objects. Doesn't change existing policies.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**: Ignores public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

A warning message in a yellow box states: "Turning off block all public access might result in this bucket and the objects within becoming public. AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting." Below the message is a checkbox labeled "I acknowledge that the current settings might result in this bucket and the objects within becoming public." The bottom of the page includes standard AWS footer links: CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

5. Selecting the checkbox for Turning off blocks all public access might result in this bucket and the objects within becoming public

The screenshot shows the same 'Block Public Access settings for this bucket' section, but now the 'Block all public access' checkbox is checked. The warning message and acknowledgment checkbox are identical to the previous screenshot. The bottom of the page includes standard AWS footer links: CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

6. Keep bucket versioning as disabled and add tags if required

The screenshot shows the 'Bucket Versioning' configuration page for an S3 bucket. The 'Disable' option is selected. There is a section for 'Tags - optional (0)' with a link to learn more about bucket tags. Below it, there is a button to 'Add tag'. A note states 'No tags associated with this bucket.' At the bottom, there is a 'Default encryption' section with an 'Info' link, stating 'Server-side encryption is automatically applied to new objects stored in this bucket.' and an 'Encryption type' section with an 'Info' link.

7. Keep default encryption disabled and click on Create Bucket button

The screenshot shows the 'Create Bucket' configuration page. Under 'Default encryption', the 'Server-side encryption with Amazon S3 managed keys (SSE-S3)' option is selected. In the 'Bucket Key' section, the 'Disable' option is selected. Below these, there is a 'Advanced settings' section with a note: 'After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.' At the bottom right, there is a 'Create bucket' button.

You can now see the successful creation of your bucket

The screenshot shows the AWS S3 console with a green success message at the top: "Successfully created bucket 'myawsbucketcc16'. To upload files and folders, or to configure additional bucket settings, choose View details." Below this, the "General purpose buckets" section is displayed, showing three buckets: "elasticbeanstalk-eu-north-1-533267428271", "elasticbeanstalk-us-east-1-533267428271", and "myawsbucketcc16". The "myawsbucketcc16" bucket was created on April 8, 2024, at 11:08:45 UTC+05:30.

Name	AWS Region	IAM Access Analyzer	Creation date
elasticbeanstalk-eu-north-1-533267428271	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	March 13, 2024, 16:47:02 (UTC+05:30)
elasticbeanstalk-us-east-1-533267428271	US East (N. Virginia) us-east-1	View analyzer for us-east-1	February 26, 2024, 15:41:08 (UTC+05:30)
myawsbucketcc16	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	April 8, 2024, 11:08:45 (UTC+05:30)

8. Now click on the bucket that you have created

The screenshot shows the AWS S3 console with the "myawsbucketcc16" bucket selected in the list. The bucket details are visible on the right side of the screen.

Name	AWS Region	IAM Access Analyzer	Creation date
elasticbeanstalk-eu-north-1-533267428271	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	March 13, 2024, 16:47:02 (UTC+05:30)
elasticbeanstalk-us-east-1-533267428271	US East (N. Virginia) us-east-1	View analyzer for us-east-1	February 26, 2024, 15:41:08 (UTC+05:30)
myawsbucketcc16	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	April 8, 2024, 11:08:45 (UTC+05:30)

9. You can either create a folder here or upload an existing file in the bucket

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with options like 'Buckets', 'Storage Lens', and 'Feature spotlight'. The main area shows a bucket named 'myawsbucketcc16'. At the top, there are tabs for 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. Below the tabs, there's a toolbar with actions like 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. A search bar says 'Find objects by prefix'. A table below shows 'No objects' found. At the bottom right of the main area, there's a large 'Upload' button.

10. Now click on the upload button and click on the add files button browse your local machine and select which file you need to upload on S3 next click on the upload button at the bottom right end

The screenshot shows the AWS S3 console with the 'Upload' interface. A large text box in the center says 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' Below this, a table lists 'Files and folders (12 Total, 71.7 MB)' with columns for 'Name', 'Folder', and 'Type'. The table includes files like 'CCL_ASSIGNMENT1.pdf', 'CCL_ASSIGNMENT2.pdf', and several 'CCL_EXP' files. At the top of the upload interface, there are 'Add files' and 'Add folder' buttons. At the bottom right, there's a large 'Upload' button.

The screenshot shows the AWS S3 console with a progress bar at the top indicating "Uploading" with 7% completion. Below the progress bar, it says "Total remaining: 12 files: 66.8 MB (93.14%)", "Estimated time remaining: a minute", and "Transfer rate: 1.1 MB/s". A "Cancel" button is visible. The main area displays a table titled "Files and folders (12 Total, 71.7 MB)" with columns for Name, Folder, Type, Size, Status, and Error. All files listed are in the "Pending" status.

Name	Folder	Type	Size	Status	Error
CCL_ASSIGN...	CCL/	application/...	6.0 MB	In progress (85)	-
CCL_ASSIGN...	CCL/	application/...	4.4 MB	Pending	-
CCL_EXP1.pdf	CCL/	application/...	13.5 MB	Pending	-
CCL_EXP10....	CCL/	application/...	12.3 MB	Pending	-
CCL_EXP2.pdf	CCL/	application/...	9.8 MB	Pending	-
CCL_EXP3.pdf	CCL/	application/...	4.8 MB	Pending	-
CCL_EXP4.pdf	CCL/	application/...	3.6 MB	Pending	-
CCL_EXP5.pdf	CCL/	application/...	2.7 MB	Pending	-
CCL_EXP6.pdf	CCL/	application/...	2.3 MB	Pending	-
CCL_EXP7.pdf	CCL/	application/...	2.6 MB	Pending	-

Now you can check the upload status screen

The screenshot shows the AWS S3 console with a green header bar indicating "Upload succeeded" and "View details below.". Below the header, there are two tabs: "Files and folders" (selected) and "Configuration". The main area displays a table titled "Files and folders (12 Total, 71.7 MB)" with columns for Name, Folder, Type, Size, Status, and Error. All files listed are now in the "Succeeded" status.

Name	Folder	Type	Size	Status	Error
CCL_ASSIGN...	CCL/	application/...	6.0 MB	Succeeded	-
CCL_ASSIGN...	CCL/	application/...	4.4 MB	Succeeded	-
CCL_EXP1.pdf	CCL/	application/...	13.5 MB	Succeeded	-
CCL_EXP10....	CCL/	application/...	12.3 MB	Succeeded	-
CCL_EXP2.pdf	CCL/	application/...	9.8 MB	Succeeded	-
CCL_EXP3.pdf	CCL/	application/...	4.8 MB	Succeeded	-
CCL_EXP4.pdf	CCL/	application/...	3.6 MB	Succeeded	-
CCL_EXP5.pdf	CCL/	application/...	2.7 MB	Succeeded	-
CCL_EXP6.pdf	CCL/	application/...	2.3 MB	Succeeded	-
CCL_EXP7.pdf	CCL/	application/...	2.6 MB	Succeeded	-

Now click on the close button.
The screen will appear as below.

The screenshot shows the AWS S3 console interface. At the top, the navigation bar includes the AWS logo, Services, a search bar, and account information for Stockholm and user 'omsheete0550'. Below the navigation bar, the path 'Amazon S3 > Buckets > myawsbucketccl6' is displayed. The main content area is titled 'myawsbucketccl6' with an 'Info' link. A sub-header 'Objects (1) Info' is shown, along with a toolbar with icons for Copy, Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload. A message states: 'Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions.' A search bar labeled 'Find objects by prefix' is present. The object list table has columns for Name, Type, Last modified, Size, and Storage class. One object, 'CCL/' (a folder), is listed. At the bottom of the page, there are links for CloudShell, Feedback, and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

11. Select properties and scroll down to Static website hosting option which is disabled now click on Edit option on right side

The screenshot shows the 'Properties' page for the 'myawsbucketccl6' bucket in the AWS S3 console. The top navigation bar and account information are identical to the previous screenshot. The main content area is titled 'myawsbucketccl6 Properties'. It contains several sections: 'Object Lock' (disabled), 'Requester pays' (disabled), and 'Static website hosting'. The 'Static website hosting' section is specifically highlighted, showing the status 'Disabled' and a note: 'Use this bucket to host a website or redirect requests.' An 'Edit' button is located next to the 'Static website hosting' title. At the bottom of the page, there are links for CloudShell, Feedback, and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

12. Now come to the Amazon S3 tab select your bucket and then click on the delete button

The screenshot shows the AWS S3 console. In the 'General purpose buckets' section, there are three buckets listed:

- elasticbeanstalk-eu-north-1 (selected)
- elasticbeanstalk-us-east-1
- myawsbucketcc16

Actions available for each bucket include Copy ARN, Empty, Delete, and Create bucket.

The screenshot shows the 'Empty bucket' confirmation dialog for the 'myawsbucketcc16' bucket. It includes a warning message about deleting all objects, a note about lifecycle rules, and a text input field to confirm deletion.

Empty bucket

Permanently delete all objects in bucket "myawsbucketcc16"?

To confirm deletion, type *permanently delete* in the text input field.

permanently delete

Cancel **Empty**

The screenshot shows the 'Empty bucket: status' confirmation dialog. It displays a summary of the deletion process, showing 12 objects deleted successfully and 0 failed to delete.

Summary

Source	Successfully deleted	Failed to delete
s3://myawsbucketcc16	12 objects, 71.7 MB	0 objects

Failed to delete (0)

No failed object deletions

The screenshot shows the 'Delete bucket' confirmation dialog in the AWS S3 console. At the top, there is a warning box containing instructions about deleting buckets. Below it, the message 'Delete bucket "myawsbucketcl6"' is displayed, followed by a text input field containing 'myawsbucketcl6'. At the bottom, there are 'Cancel' and 'Delete bucket' buttons.

The screenshot shows the 'General purpose buckets' list in the AWS S3 console. A green banner at the top indicates 'Successfully deleted bucket "myawsbucketcl6"'. The table below lists two buckets: 'elasticbeanstalk-eu-north-1-533267428271' and 'elasticbeanstalk-us-east-1-533267428271'. The table includes columns for Name, AWS Region, IAM Access Analyzer, and Creation date.

Name	AWS Region	IAM Access Analyzer	Creation date
elasticbeanstalk-eu-north-1-533267428271	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	March 13, 2024, 16:47:02 (UTC+05:30)
elasticbeanstalk-us-east-1-533267428271	US East (N. Virginia) us-east-1	View analyzer for us-east-1	February 26, 2024, 15:41:08 (UTC+05:30)

Experiment No: 7

Aim: To study and implement Identity and access management (IAM) portion practices on AWS

Theory: ~~with respect to cloud computing~~

- Explain the concept and need of access management.
- → Access management (AM) is process of identifying and controlling and managing authorized or specified user access to a system, application or basically IT instances.
- It is a broad concept that encompasses all policies, methodologies and tools to maintain access privilege.
- AM is primarily an information security, IT and data governance process used in granting access to valid users or prohibiting involved users.
- Typically, AM is used in conjunction with identity access management (IAM).
- Identity management creates, provision and controls the different users, roles and groups and policies whereas AM ensures that those roles and policies are allowed.
- An IAM-based application states the different user roles and their policies.



Fundamentals

Explain IAM and its components.

- IAM is an identity and access management is the security discipline that enables the right individuals to access the right resources at the right reasons.
- These resources could be tools required to complete a job, access to database with mission critical transaction services and applications hosted in cloud.
- Components in IAM include
 - 1) Authentication
 - 2) Authorization
 - 3) User Management
 - 4) Central User Repository

It can be categorized into four components:-

- 1) Authentication
- 2) Authorization
- 3) User Management
- 4) Central User Repository

Compare the following in AWS:

- Root User and AWS IAM User
- Root user is the first cloud service identity created by default when you create your cloud service provider account.
- It is important to note all cloud services providers have some form of root elements.
- AWS IAM user can be created by a root user or another IAM user who has entitlements to create additional IAM user.

- Can authenticate on start a remote session using their credentials or alias.
- Can corresponds to human application process on another machine based identity.

2) Roles and Policies

⇒

Cloud Roles:

- i) Intended for granting temporary access for entities like application of services.
- ii) AWS resources.
- iii) Temporary access.
- iv) Defines who can assume the role.
- v) Provides a way to grant precise permissions to a specific entity.

Cloud Policies:

- i) Defines permission and can be attached to user groups or roles.
- ii) Attached to the IAM users/groups.
- iii) Permanent permission.
- iv) This relationship is not applicable as they do not involve.
- v) Defines a set of permissions that can be attached to multiple entities.

- Explain inline and custom policies in AWS
 - An inline policy is a policy created for a single IAM identity.
 - Inline policies maintain a strict one-to-one relationship between a policy and identity either you can create a policy and embed it in an identity.
- An AWS managed policy is a standalone policy that is created and administered by AWS and doesn't have its own ARN that includes the policy name.
- You can't create a standalone policy in your own AWS account that you can attach to principal entities.

• Explain Multifactor Authentication in AWS

⇒ It is an AWS entity identity and Access Management best practice that requires a second authentication factor for high additional security.

Written Within (A) S 5/3/2024

Output:

1. Login to AWS console Make sure to check all Ec2 dashboard parameters

The screenshot shows the AWS EC2 Dashboard for the Europe (Stockholm) Region. The left sidebar includes sections for EC2 Global View, Events, Instances, Images, Elastic Block Store, and Networks & Security. The main area displays the following data:

- Resources:** Instances (running) 0, Auto Scaling Groups 0, Dedicated Hosts 0, Elastic IPs 0, Instances 1, Key pairs 1, Load balancers 0, Placement groups 0, Security groups 2, Snapshots 0, Volumes 0.
- Launch instance:** Buttons for "Launch instance" and "Migrate a server". Note: Your instances will launch in the Europe (Stockholm) Region.
- Service health:** Region Europe (Stockholm), Status (green) This service is operating normally.
- Zones:** Zone name, Zone ID.
- Offer usage (monthly):** Linux EC2 Instances (749.884722 hours remaining), Storage space on EBS (30 GB remaining).
- Account attributes:**

2. Go to IAM dashboard

The screenshot shows the AWS IAM Dashboard. The left sidebar includes sections for Identity and Access Management (IAM), Access management, Access reports, and CloudShell. The main area displays the following information:

- Security recommendations:**
 - Add MFA for root user (warning icon)
 - Root user has no active access keys (green checkmark icon)
- IAM resources:** User groups 0, Users 0, Roles 3, Policies 0, Identity providers 0.
- AWS Account:** Account ID 533267428271, Account Alias Create, Sign-in URL https://533267428271.signin.aws.amazon.com/console.
- Quick Links:** My security credentials (Manage your access keys, multi-factor authentication (MFA) and other credentials).
- Tools:** CloudShell, Feedback.

3. Click on create option under Account Alias and give a valid name; save changes

The screenshot shows the AWS IAM Dashboard. A modal window titled "Create alias for AWS account 533267428271" is open. In the "Preferred alias" field, the value "myaliascl7" is entered. Below the field, a note states: "Must be not more than 63 characters. Valid characters are a-z, 0-9, and - (hyphen)." A "New sign-in URL" is provided: <https://myaliascl7.signin.aws.amazon.com/console>. A note below the URL says: "IAM users will still be able to use the default URL containing the AWS account ID." At the bottom right of the modal is a yellow "Create alias" button.

The screenshot shows the AWS IAM Dashboard after the alias has been created. A green banner at the top left reads: "Alias myaliascl7 created for this account." The main dashboard area shows the "Security recommendations" section with one item: "Add MFA for root user". The "IAM resources" section displays statistics: 0 User groups, 0 Users, 3 Roles, 0 Policies, and 0 Identity providers. The "Quick Links" and "Tools" sections are also visible on the right.

4. Click on “users” in the left column

The screenshot shows the AWS Identity and Access Management (IAM) service interface. The left sidebar has a tree view with 'Identity and Access Management (IAM)' selected. Under 'Access management', 'Users' is also selected. The main content area is titled 'Users (0) Info' and contains a message: 'An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.' Below this is a search bar and a table header with columns: User name, Path, Group, Last activity, MFA, Password age, and Console last sign-in. A message 'No resources to display' is centered below the table. At the top right, there are 'Create user', 'Delete', and other navigation buttons.

5. Click on the Create Users button

The screenshot shows the 'Specify user details' step of the IAM User creation wizard. On the left, a sidebar lists steps: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password). The main area is titled 'Specify user details' and contains a 'User details' section. It includes a 'User name' input field with 'Om_Shete' typed in, a note about character restrictions, and a checked checkbox for 'Provide user access to the AWS Management Console - optional'. Below this is a box titled 'Are you providing console access to a person?' with two options: 'Specify a user in Identity Center - Recommended' (radio button unselected) and 'I want to create an IAM user' (radio button selected). A note for the second option states: 'We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.' At the bottom, there are 'Console password', 'Autogenerated password', and 'Custom password' options, with 'Custom password' selected. The footer includes standard AWS links like CloudShell and Feedback.

Are you providing console access to a person? [Alt+S]

User type

- Specify a user in Identity Center - Recommended
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.
- I want to create an IAM user
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

Console password

- Autogenerated password
You can view the password after you create the user.
- Custom password
Enter a custom password for the user.

 - Must be at least 8 characters long
 - Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & * () _ + - (hyphen) = [] { } { }

Show password

Users must create a new password at next sign-in - Recommended
Users automatically get the IAMUserChangePassword policy to allow them to change their own password.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel Next

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

IAM > Users > Create user

Step 1 [Specify user details](#)

Step 2 **Set permissions**

Step 3 [Review and create](#)

Step 4 [Retrieve password](#)

Permissions options

- Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function. [Learn more](#)
- Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
- Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Get started with groups
Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. [Learn more](#)

Create group

▶ Set permissions boundary - optional

Cancel Previous Next

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws Services Search [Alt+S]

Step 2 [Set permissions](#)

Step 3 [Review and create](#)

Step 4 [Retrieve password](#)

User details

User name Om_Shete	Console password type Custom password	Require password reset Yes
-----------------------	--	-------------------------------

Permissions summary

Name	Type	Used as
IAMUserChangePassword	AWS managed	Permissions policy

Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

Key	Value - optional
NewUser	Om_Shete

Add new tag
You can add up to 49 more tags.

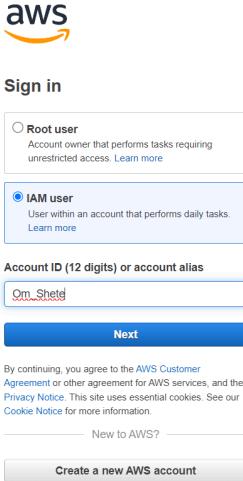
Cancel Previous Create user

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS Management Console with the IAM service selected. A green success banner at the top states "User created successfully" and provides instructions to view and download the user's password and email instructions for signing in to the AWS Management Console. Below the banner, the "Create user" step is completed. The "Retrieve password" section displays the console sign-in URL (<https://myaliascd7.sigin.aws.amazon.com/console>), user name (Om_Shete), and a masked console password with a "Show" link. Navigation links on the left include Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password). Action buttons at the bottom right include "Cancel", "Download .csv file", and "Return to users list".

The screenshot shows the Windows File Explorer interface. The left sidebar shows navigation paths like Home, Om - Personal, Desktop, Documents, Pictures, Downloads, and This PC. The main area displays three recently downloaded files: "Om_Shete_create_details" (Excel document icon), "mykeypair.pem" (text file icon), and "CCL_EXP6" (Chrome icon). A "Today" folder is also visible. The status bar at the bottom indicates there are 31 items.

Logging in as the new User & Checking their permissions
Enter the new user's name and psw saved earlier



The screenshot shows the AWS sign-in interface. It features two radio button options: "Root user" (unchecked) and "IAM user" (checked). Below the "IAM user" option is a note: "User within an account that performs daily tasks." A text input field contains the account alias "Om_Shete". A blue "Next" button is positioned below the input field. To the right of the sign-in form is a purple banner with the text "AWS Support" and "Save time and move faster with expert guidance and assistance". At the bottom of the sign-in form, there is a link to "Create a new AWS account".



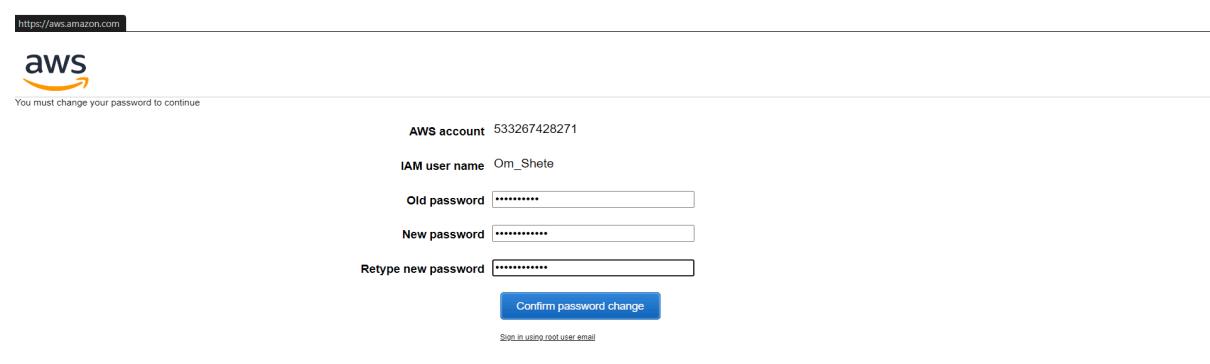
The screenshot shows the AWS Support landing page. It has a purple gradient background with the "AWS Support" logo at the top. Below it, the text "Save time and move faster with expert guidance and assistance" is displayed. A blue "Select a Support plan >" button is located on the right side. At the bottom left, there is a link to "New to AWS?".



The screenshot shows the "Sign in as IAM user" page. It includes fields for "Account ID (12 digits) or account alias" (containing "myaliascc17"), "IAM user name" (containing "Om_Shete"), and "Password" (containing "*****"). There is also a "Remember this account" checkbox and a "Sign in" button. Below the form, links for "Sign in using root user email" and "Forgot password?" are visible.



The screenshot shows the Amazon Lightsail landing page. It features a dark background with a bright orange and yellow swoosh graphic. The text "Amazon Lightsail" is prominently displayed, followed by "Lightsail is the easiest way to get started on AWS". A blue "Learn more >" button is present. To the right, there is a cartoon illustration of a robot holding a thumbs-up.



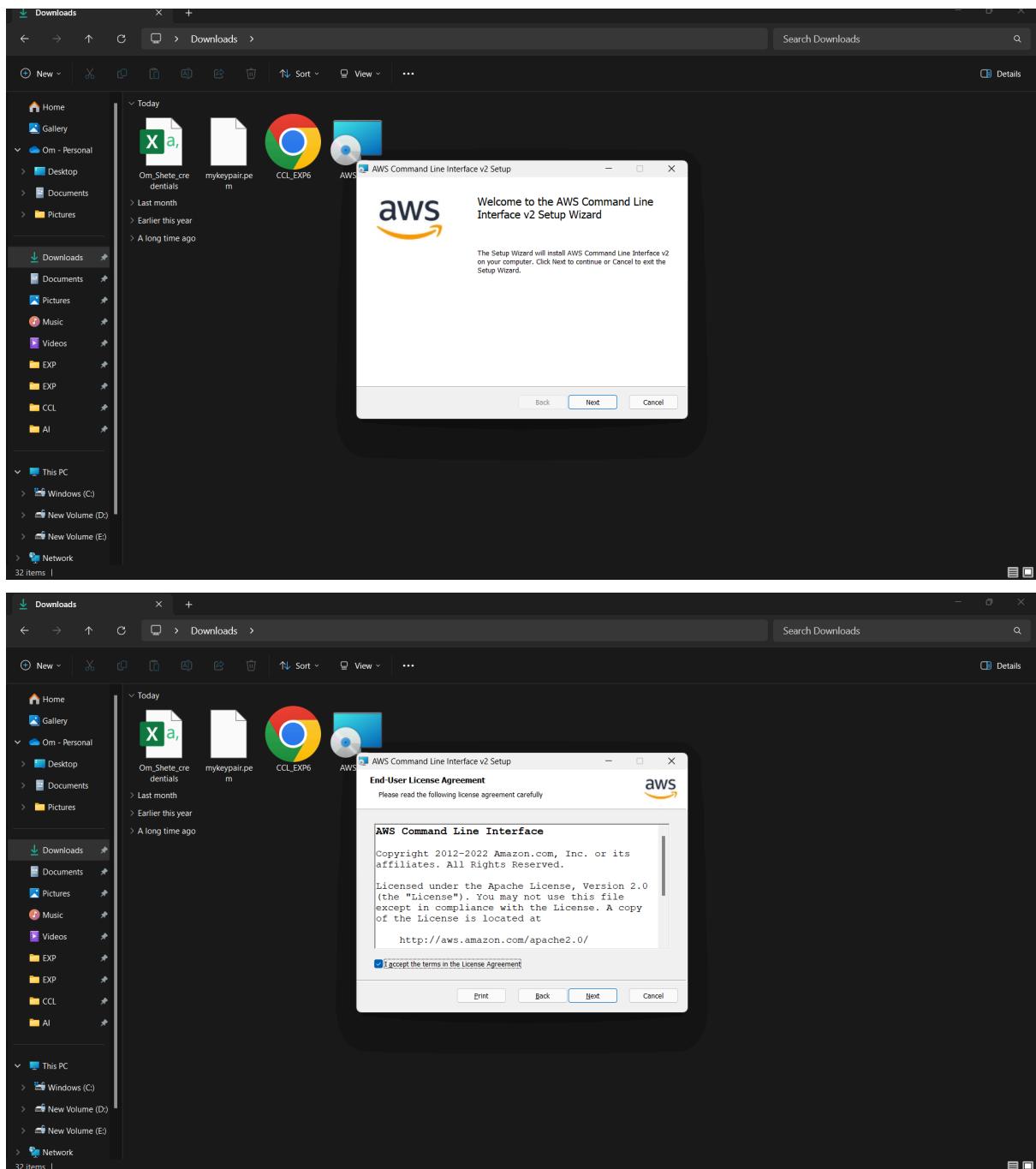
The screenshot shows the "Change Password" page for the IAM user "Om_Shete". It displays the AWS account number "533267428271" and the IAM user name "Om_Shete". Three password input fields are shown: "Old password" (containing "*****"), "New password" (containing "*****"), and "Retype new password" (containing "*****"). A blue "Confirm password change" button is at the bottom. Below the form, there is a link "Sign in using root user email".

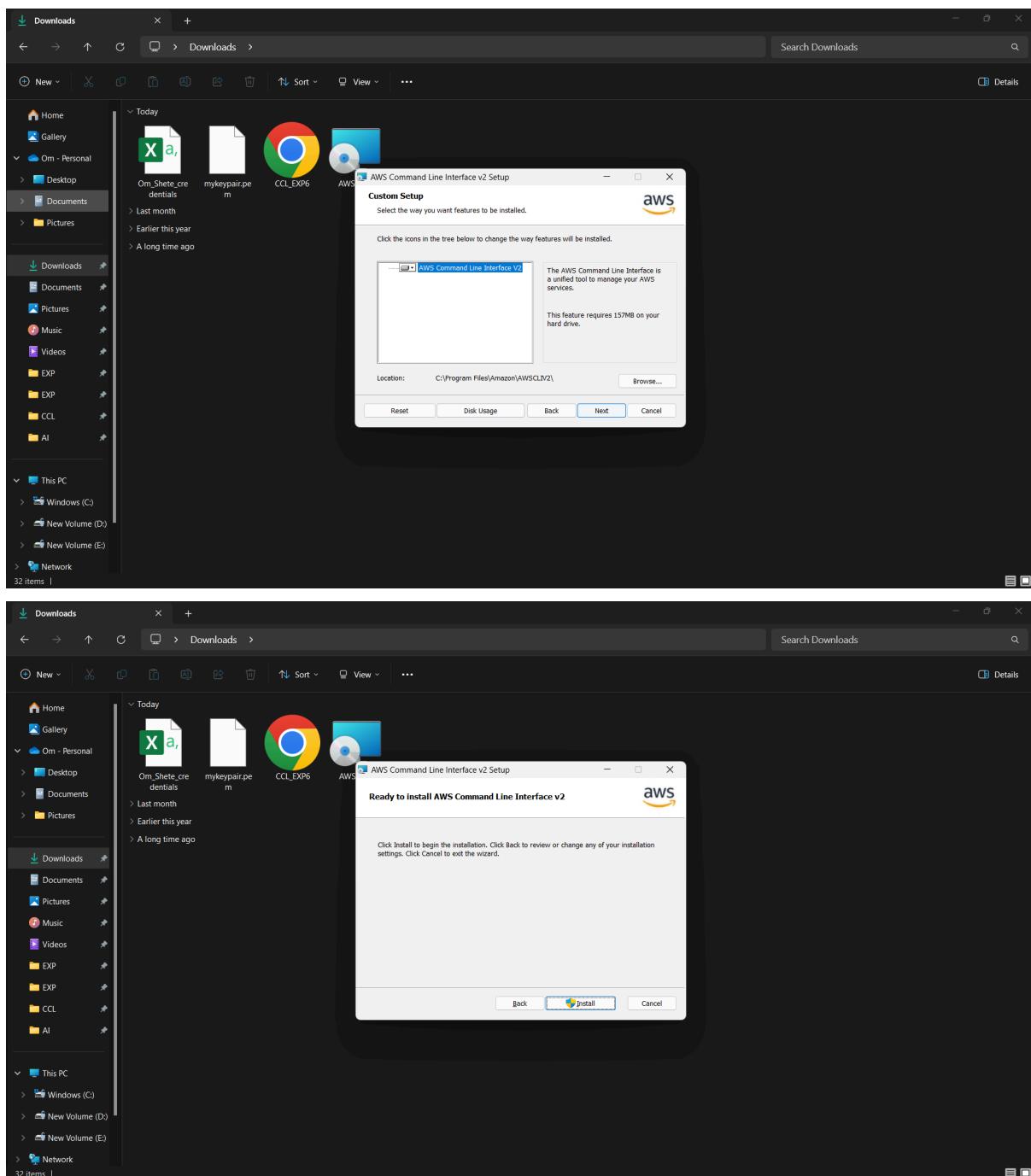
After logging in, you will notice that you don't have permission to do anything yet

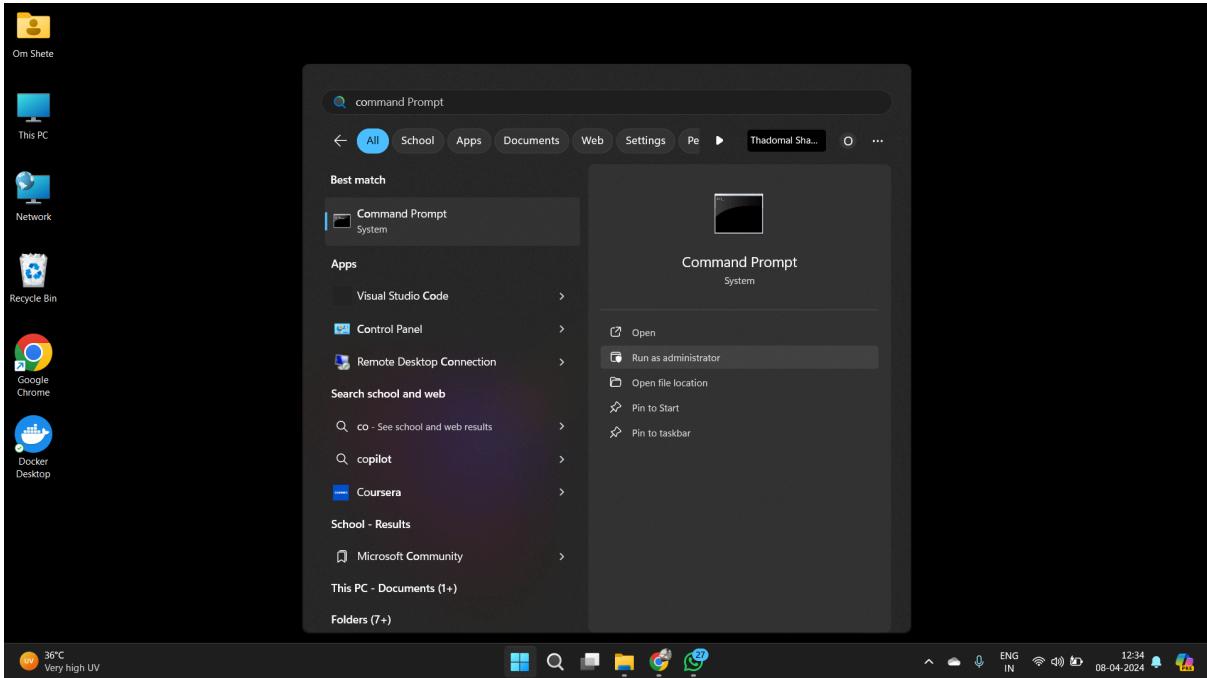
The image consists of three vertically stacked screenshots of the AWS console, all showing an "Access denied" error message.

- Screenshot 1: AWS Console Home**
Shows the "Recently visited" section with a note: "No recently visited services". Below it, a link to "View all services". To the right, the "Applications" section shows a table with one row, which has a red border and displays the "Access denied" message.
- Screenshot 2: EC2 Dashboard**
Shows the "Instances" section. A search bar at the top contains the query "Instance state = running". Below it, a table header includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. A blue error message box states: "You are not authorized to perform this operation. User: arn:aws:iam::533267428271:user/Om_Shete is not authorized to perform: ec2:DescribeInstances because no identity-based policy allows the ec2:DescribeInstances action".
- Screenshot 3: Amazon S3**
Shows the "Buckets" section. A search bar at the top contains the query "Find buckets by name". Below it, a table header includes columns for Name, AWS Region, IAM Access Analyzer, and Creation date. A blue error message box states: "You don't have permissions to list buckets. After you or your AWS administrator has updated your permissions to allow the s3>ListAllMyBuckets action, refresh this page. Learn more about Identity and access management in Amazon S3".

Type “AWS CLI” in a new window of any browser and go to its the main page of AWS regarding the same Click on 64-bit hyperlink in the RHS column under the Windows section and download, install the AWS CLI







Type aws configure, it will ask for a few inputs; AWS Access Key ID and Key are the ones which we saved earlier Default region name is whichever region AWS you are using; in case of Mumbai, its: apsouth-1 The output format is json in our case

```
C:\ Administator: Command Prompt
Microsoft Windows [Version 10.0.22631.3374]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>aws configure
AWS Access Key ID [None]: AKIAKYKJXGOXV4Z3JQKJ
AWS Secret Access Key [None]: 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
Default region name [None]: Stockholm
Default output format [None]: json

C:\Windows\System32>
```

The next two steps are OPTIONAL: aws --version aws s3 ls

```
C:\Windows\System32>aws --version
aws-cli/2.15.36 Python/3.11.8 Windows/10 exe/AMD64 prompt/off

C:\Windows\System32>aws s3 ls
Could not connect to the endpoint URL: "https://s3.Stockholm.amazonaws.com/"

C:\Windows\System32>
```

Go in the security credentials tab under Users of IAM Dashboard

The screenshot shows the AWS IAM Dashboard. In the top navigation bar, the user 'omschte0550' is selected. The main area shows a user named 'Om_Shete'. Under the 'Security credentials' tab, it lists an 'Access key 2' with the option to 'Create access key'. Below this, the 'Permissions' tab is selected, showing a list of 'Permissions policies (1)'. One policy, 'IAMUserChangePassword', is listed as 'AWS managed' and attached 'Directly'. There are also sections for 'Permissions boundary (not set)' and 'Generate policy based on CloudTrail events'.

The screenshot shows the 'Policies' section of the AWS IAM console. It displays the details of the 'IAMUserChangePassword' policy. The 'Policy details' section shows the policy was created on November 15, 2016, at 05:55 UTC+05:30, and last edited on November 16, 2016, at 04:48 UTC+05:30. The ARN is listed as 'arn:aws:iam::aws:policy/IAMUserChangePassword'. The 'Permissions' tab is selected, showing a table titled 'Permissions defined in this policy'. It lists one service, 'IAM', with an 'Access level' of 'Limited: Read, Write' and a 'Resource' of 'Multiple'. A link to 'Show remaining 406 services' is present.

Click on the “Manage” Hyperlink

The image consists of three vertically stacked screenshots from the AWS IAM console, illustrating the steps to manage MFA for a user.

Screenshot 1: User Summary Page

This screenshot shows the "Summary" tab for the user "Om_Shete". Key details include:

- ARN:** arn:aws:iam::533267428271:user/Om_Shete
- Console access:** Enabled without MFA
- Created:** April 08, 2024, 12:49 (UTC+05:30)
- Last console sign-in:** Never
- Access key 1:** AKIAXYKJXGOV4Z3JQKJ - Active (Never used. Created today.)
- Access key 2:** Create access key

Screenshot 2: Select MFA Device Page

This screenshot shows the "Select MFA device" step. It includes:

- MFA device name:** mydevice
- MFA device options:**
 - Authenticator app:** Selected (Authenticate using a code generated by an app installed on your mobile device or computer.)
 - Security Key:** Unselected (Authenticate using a code generated by touching a YubiKey or other supported FIDO security key.)

Screenshot 3: Set up Authenticator App Page

This screenshot shows the "Set up device" step for the "Authenticator app". It includes:

- Step 1:** Install a compatible application such as Google Authenticator, Duo Mobile, or Authy app on your mobile device or computer. See a list of compatible applications.
- Step 2:** Scan the QR code using your authenticator app. Open your authenticator app, choose Show QR code on this page, then use the app to scan the code. Alternatively, you can type a secret key.
- Step 3:** Fill in two consecutive codes from your MFA device.

Buttons at the bottom include: Cancel, Previous, and Add MFA.

Enter two of the codes which are shown in the Google Authenticator App over a span of 30 secs each; click on Assign MFA Button

The screenshot shows the AWS Identity and Access Management (IAM) console. On the left, the navigation pane is visible with sections like 'Identity and Access Management (IAM)', 'Access management', 'Users', 'Access reports', and 'Organization activity'. The main content area displays a summary for a user named 'Om_Shete'. A prominent green banner at the top states 'MFA device assigned' with a note about registering up to 8 MFA devices. Below this, the 'Summary' section provides details: ARN (arn:aws:iam::533267428271:user/Om_Shete), Console access (Enabled with MFA), and two access keys (Access key 1 and Access key 2). The 'Security credentials' tab is selected. The 'Console sign-in' section shows a console sign-in link (https://myaliascl7.signin.aws.amazon.com/console) and a console password (Updated 25 minutes ago). The 'Multi-factor authentication (MFA)' section shows one MFA device assigned, with buttons for 'Remove', 'Resync', and 'Assign MFA device'.

Experiment No : 8

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

Theory:

Explain the concept of Database as a Service.
List some Database as a service provider along with name of service.

- ⇒ Database as a service (DBaaS) is a cloud computing service model that provides users with access to managed database system.
- Instead of setting up and maintaining their own database infrastructure, users can leverage a cloud provider's resources to store, manage and access their data.
- Some well-known DBaaS providers include:
 - 1) Amazon Web Services (AWS)
 - ⇒ Amazon RDS, Amazon Aurora
 - 2) Microsoft Azure
 - ⇒ Azure SQL database, Cosmos DB

3) Google Cloud Platform

⇒ cloud SQL, firestore

Cloud storage, cloud database, Bigtable

Cloud machine learning, Cloud functions

Cloud storage, Cloud functions

Cloud databases and storage

8.1. AWS Relational Database Services

- o List the most popular database supported by AWS and its architecture.
- AWS supports various SQL and NoSQL databases, including:
 - 1) SQL database: MySQL, PostgreSQL, SQL Server, MariaDB, Oracle, SQLite, Aurora MySQL.
 - 2) NoSQL database: Amazon DynamoDB.
 - 3) Difference between RDS and Aurora.

Amazon RDS	Aurora
1) Amazon RDS is one of the managed relational database services that supports multiple database engines such as MySQL, PostgreSQL, MariaDB, Oracle and SQL Server.	2) Aurora is a MySQL and PostgreSQL compatible relational database engine built for the cloud.
2) Aurora replicates data six ways across three availability zones by default, providing greater durability and redundancy.	2) Aurora's storage system is highly available with multiple instances; allowing for faster replication and failover compared to RDS where each instance has its own storage.

* Explain the following terms with AWS RDS.

1) DBaaS (Amazon Relational Database Service)

⇒ DBaaS provides a managed database service that allows you to focus on your application logic and data storage needs.

2) Storage types:

⇒ AWS offers various storage types for databases, including General Purpose SSD (GP2), Provisioned IOPS SSD (T1) (standard), Magnetic (standard), and Amazon Aurora storage.

3) These options differ in terms of performance, durability, and cost.

2) Endpoint

⇒ In AWS RDS, an endpoint refers to the network address used to access a database instance. Applications connect to the database stored either over the internet or within the AWS network.

3) Snapshot

⇒ A snapshot is a point-in-time copy of a database instance. It captures the entire state of a database at the moment the snapshot is taken.

- Snapshots are used for backup, recovery, and replication purpose.

- 4) Read Replicas → primed with notes
- A read replica is a copy of a source database instance that allows read-only access to the data.
- Read replicas can be used to offload read operations from the primary database, improving performance and scalability.
- 5) Single AZ and Multi AZ instances
- In AWS, a single Availability Zone instance runs in a single data center within a specific AWS region.
- In contrast, a multi-AZ instance replicates data synchronously across multiple availability zones within the same region, providing high availability and fault tolerance in case of a failure in one AZ.

Ax 18/3/2024

In your writing, if you write with a pen, then it is handwritten and if you write with a computer, then it is typed. If you write with a pen, then it is handwritten and if you write with a computer, then it is typed.

Output:

Amazon RDS Dashboard

The dashboard shows the following resources in the EU North (Stockholm) region:

- DB Instances (0/40)
- Allocated storage (0 TB/100 TB)
- DB Clusters (0/40)
- Reserved instances (0/40)
- Snapshots (0)
 - Manual: DB Cluster (0/100), DB Instance (0/100)
 - Automated: DB Cluster (0), DB Instance (0)
- Recent events (0)
- Event subscriptions (0/20)

Recommended services

- OpsWorks
- Amazon Prometheus
- Simple Queue Service
- Route 53
- Lightsail

Additional information

- Getting started with RDS
- Overview and features
- Documentation
- Articles and tutorials
- Data import guide for MySQL
- Data import guide for Oracle
- Data import guide for SQL Server
- New RDS feature announcements
- Pricing
- Forums

Create database

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

Service health

Current status: Amazon Relational Database Service (Stockholm) - Service is operating normally.

Database Preview Environment

Get early access to new DB engine versions. The

MySQL

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

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

Edition

MySQL Community

Known issues/limitations
Review the [Known issues/limitations](#) to learn about potential compatibility issues with specific database versions.

Engine version [Info](#)
View the engine versions that support the following database features.

Hide filters

Show versions that support the Multi-AZ DB cluster [Info](#)
Create a Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.

Show versions that support the Amazon RDS Optimized Writes [Info](#)
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine Version
MySQL 8.0.35

Templates
Choose a sample template to meet your use case.

Production
Use defaults for high availability

Dev/Test
This instance is intended for

Free tier
Use RDS Free Tier to develop new

MySQL

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

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

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Availability and durability

Deployment options [Info](#)
The deployment options below are limited to those supported by the engine you selected above.

Multi-AZ DB Cluster
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.

Multi-AZ DB instance (not supported for Multi-AZ DB cluster snapshot)
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.

Single DB instance (not supported for Multi-AZ DB cluster snapshot)
Creates a single DB instance with no standby DB instances.

Settings

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.
database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB instance.
admin

MySQL

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

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

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.
database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB instance.
admin

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management
You can use AWS Secrets Manager or manage your master user credentials.

Managed in AWS Secrets Manager - most secure
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

Self managed
Create your own password or have RDS create a password that you manage.

Auto generate password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / * @

Confirm master password [Info](#)

MySQL

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

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

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

The screenshots illustrate the step-by-step process of creating a new Amazon RDS DB instance:

- Storage Configuration:** Shows the selection of an instance class (db.t3.micro), storage type (General Purpose SSD (gp2)), and allocated storage (20 GiB). A note indicates that after modification, the DB instance will be in storage-optimization.
- VPC Configuration:** Selects a Default VPC (vpc-02ed8cfea6738bce) and a DB subnet group (default). A note states that after creation, the VPC cannot be changed.
- Security Group Configuration:** Chooses an existing VPC security group (Choose existing) and specifies an existing VPC security group (default). Other options like RDS Proxy and Certificate authority are also shown.

Additional configuration

Database options: encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

Estimated monthly costs

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

[Learn more about AWS Free Tier](#)

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the [Amazon RDS Pricing page](#).

Master password

Specify a string that defines the password for the master user. Master Password must be at least eight characters long, as in "mypassword".

Notes

You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

Create database

Creating database database-1

Your database might take a few minutes to launch. You can use settings from database-1 to simplify configuration of [suggested database add-ons](#) while we finish creating your DB for you.

Introducing Aurora I/O-Optimized

Aurora's I/O-Optimized is a new cluster storage configuration that offers predictable pricing for all applications and improved price-performance, with up to 40% cost savings for I/O-intensive applications.

Databases (1)

DB identifier	Status	Role	Engine	Region & AZ	Size	Recommendations	CPU
database-1	Creating	Instance	MySQL Community	-	db.t3.micro	-	-

Consider creating a Blue/Green Deployment to minimize downtime during upgrades

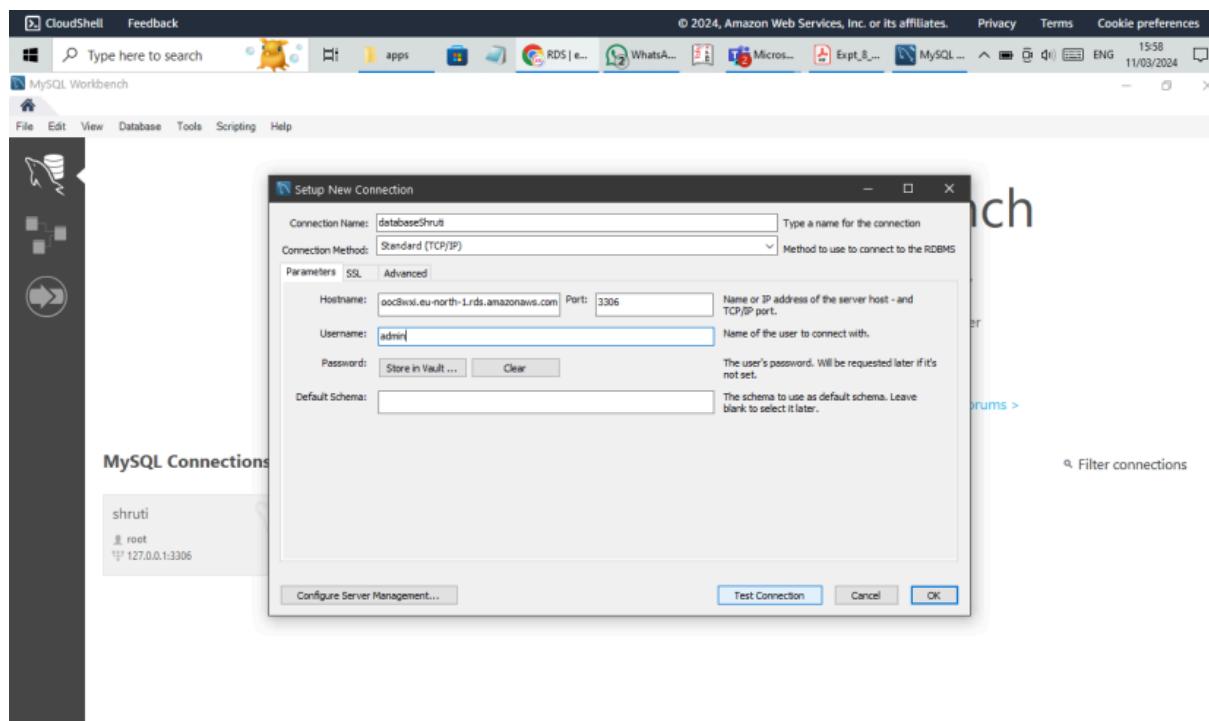
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. [RDS User Guide](#) | [Aurora User Guide](#)

Summary

DB identifier	database-1	Status	Available	Role	Instance	Engine	MySQL Community	Recommendations
CPU	21.46%	Class	db.t3.micro	Current activity	0 Connections	Region & AZ	eu-north-1b	

Connectivity & security

Endpoint	Availability Zone	Security
database-1.cjukokoegozoy.eu-north-1.rds.amazonaws.com	eu-north-1b	VPC security groups default (sg-0239f055b28cc5f49) Active
Port	VPC	Publicly accessible Yes
3306	vpc-02ed8cfeaa6738bce	



VPC dashboard EC2 Global View

Filter by VPC: Select a VPC

Virtual private cloud

Your VPCs

- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- Endpoints

Your VPCs (1) Info

Name	VPC ID	State	IPv4 CIDR
-	vpc-001671399ec9d2700	Available	172.31.0.0/16

Select a VPC above

EC2 > Security Groups > sg-01031f3c06b9fd512 - default > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

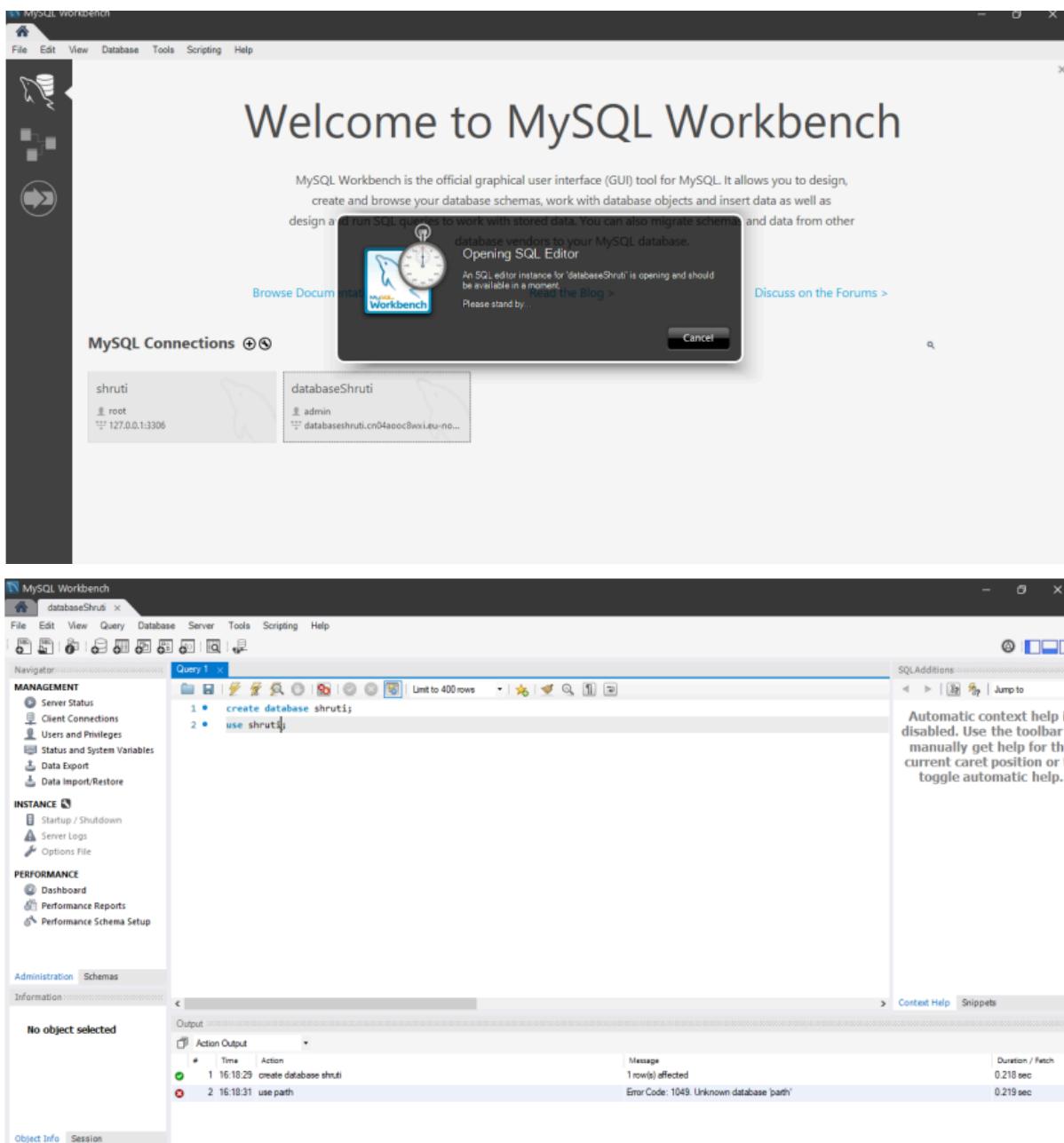
Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0eb19093ec456da26	All traffic	All	All	C... <input type="button" value="X"/>	<input type="text"/> Delete

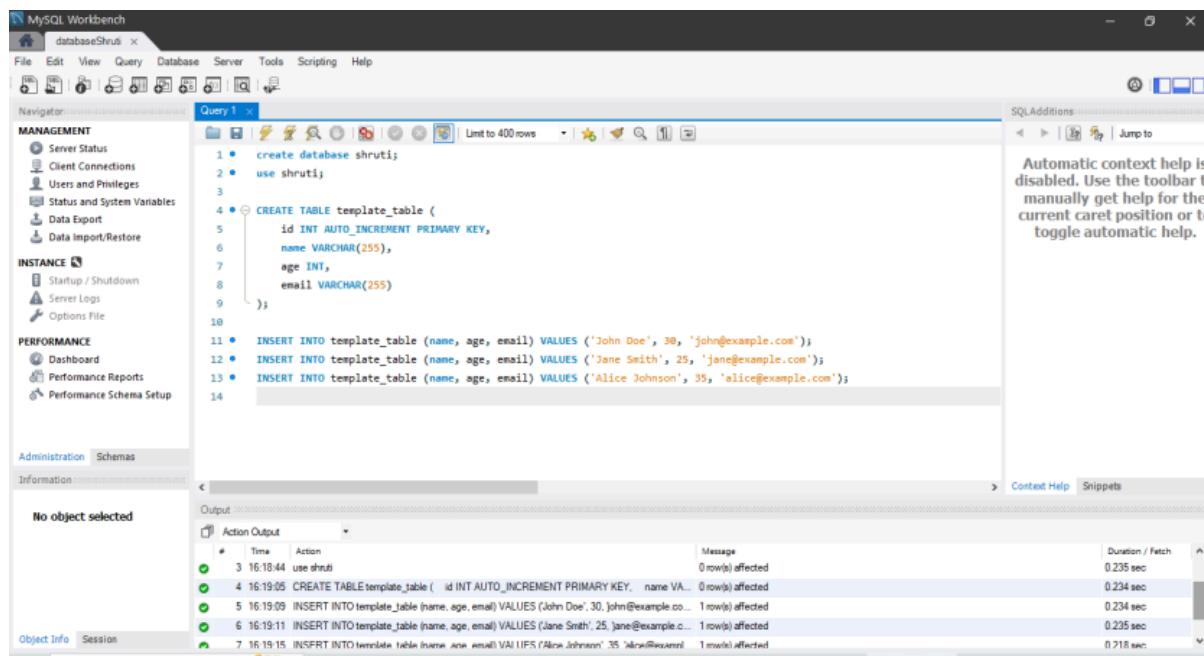
Add rule

The screenshot shows the AWS EC2 Dashboard with a success message at the top: "Inbound security group rules successfully modified on security group (sg-01031f3c06b9fd512 | default)". Below this, the "Security Groups" section is displayed, showing one item: "databaseShruti". A search bar and a "Create security group" button are also visible.

The screenshot shows the "Manage Server Connections" dialog box in MySQL Workbench. The connection name is "databaseShruti". The connection method is set to "Standard (TCP/IP)". Parameters include Hostname: "databaseshruti.cn04aoc8xi.eu-north-1.rds.amazonaws.com", Port: "3306", Username: "admin", and Password: "admin". The Default Schema is left blank. Buttons for "Test Connection" and "Close" are at the bottom.

The screenshot shows the same "Manage Server Connections" dialog box, but now it displays a confirmation message: "Successfully made the MySQL connection". It provides information about the connection parameters and a note that a successful MySQL connection was made. An "OK" button is present at the bottom of the message box.





The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the following SQL code:


```

1 •  create database shruti;
2 •  use shruti;
3
4 •  CREATE TABLE template_table (
5     id INT AUTO_INCREMENT PRIMARY KEY,
6     name VARCHAR(255),
7     age INT,
8     email VARCHAR(255)
9 );
10
11 •  INSERT INTO template_table (name, age, email) VALUES ('John Doe', 30, 'john@example.com');
12 •  INSERT INTO template_table (name, age, email) VALUES ('Jane Smith', 25, 'jane@example.com');
13 •  INSERT INTO template_table (name, age, email) VALUES ('Alice Johnson', 35, 'alice@example.com');
14
            
```
- Output Window:** Shows the execution log with the following entries:

Action	Time	Message	Duration / Fetch
use shruti	3 16:19:44	0 row(s) affected	0.234 sec
CREATE TABLE template_table (id INT AUTO_INCREMENT PRIMARY KEY, name VA...	4 16:19:05	0 row(s) affected	0.234 sec
INSERT INTO template_table (name, age, email) VALUES ('John Doe', 30, 'john@example.co...	5 16:19:09	1 row(s) affected	0.235 sec
INSERT INTO template_table (name, age, email) VALUES ('Jane Smith', 25, 'jane@example.co...	6 16:19:11	1 row(s) affected	0.235 sec
INSERT INTO template_table (name, age, email) VALUES ('Alice Johnson', 35, 'alice@example.com')	7 16:19:15	1 row(s) affected	0.218 sec
- Result Grid:** Displays the data from the template_table:

	id	name	age	email
1	1	John Doe	30	john@example.com
2	2	Jane Smith	25	jane@example.com
3	3	Alice Johnson	35	alice@example.com
- Action Output:** Shows the following errors:

Action	Time	Message	Duration / Fetch
SELECT * FROM template_table	8 16:19:18	Error Code: 1054. You have an error in your SQL syntax; check the manual that corresponds ...	0.235 sec
SELECT * FROM template_table	9 16:19:31	Error Code: 1054. You have an error in your SQL syntax; check the manual that corresponds ...	0.219 sec
select * from template_table LIMIT 0, 400	10 16:20:24	3 row(s) returned	0.219 sec / 0.000 sec

eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#databases:

Stop DB instance temporarily

You are stopping this DB instance for up to 7 days. You can restart the DB instance manually at any time. To stop the DB instance permanently, save it in a snapshot and delete it. [Learn more](#)

Acknowledgement

I acknowledge that the DB instance will restart automatically after 7 days, on March 18, 2024, 16:21 (UTC+05:30).

Snapshot - optional

Save the DB instance in a snapshot
The snapshot enables you to restore the DB instance to its last state before it was stopped.

Cancel **Stop temporarily**

[Create database](#)

Region & AZ Size

eu-north-1b db.t3.micro

[CloudShell](#) [Feedback](#)

[eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#databases:">eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#databases:](#)

Delete database-1 instance

Permanently delete database-1 DB instance. You can't undo this action.

⚠ Proceeding with this action will delete the instance with all its content and can affect related resources. [Learn more](#)

Create final snapshot
Determines whether a final DB Snapshot is created before the DB instance is deleted.

Final snapshot name
The identifier of the new DB snapshot that is created.

Retain automated backups
Determines whether retaining automated backups for 1 day after deletion

ⓘ You will be billed for retained backup storage at the rate described as 'Additional backup storage' found in Backup Storage. [Learn more](#)

To avoid accidental deletion provide additional written consent.

To confirm deletion, type *delete me* into the field.

Cancel **Delete**

[Create database](#)

Size Recommendations CPU

db.t3.micro

[CloudShell](#) [Feedback](#)

[eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#databases:">eu-north-1.console.aws.amazon.com/rds/home?region=eu-north-1#databases:](#)

Deleting DB instance database-1

Databases (1)

Group resources Modify Actions **Create database**

Filter by databases

DB identifier	Status	Role	Engine	Region & AZ	Size	Recommendations	CPU
database-1	Deleting	Instance	MySQL Community	eu-north-1b	db.t3.micro		

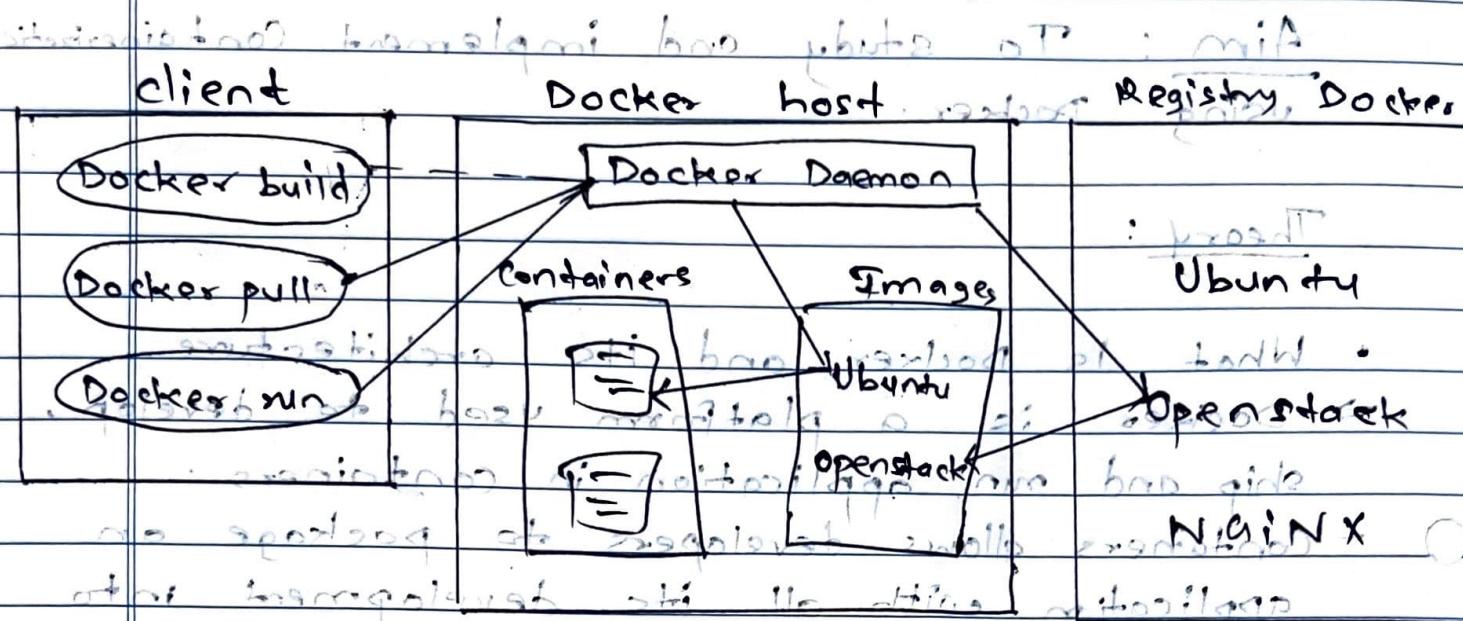
CloudShell Feedback

Experiment No: 9

Aim: To study and implement Containerization using Docker.

Theory:

- What is Docker, and its architecture
- Docker is a platform used to develop, ship and run application in containers.
- Containers allows developers to package an application with all its development into a standardized unit for software development.
- Docker makes use of client-server architecture. The Docker client talks with the docker daemon which helps in building, running, and distributing the docker containers.
- The Docker client runs with the daemon on the same system or we can connect the Docker client with the Docker daemon remotely.
- With the help of REST API over a UNIX socket or a network, the docker client and daemon interact with each other.
- Docker daemon manages all the services by communication with other daemon.
- It manages docker objects such as images, containers, networks and volumes with the help of the API requests of Docker.



- Benefits of Containerization:
 - ⇒ Containerization offers several advantages in:
 - 1) Portability: Containers encapsulate an application and its dependencies, making them portable across different environments from the build environment to production.
 - 2) Isolation: It is a superior deployment method.
 - ⇒ Containers provide isolated environments for applications, ensuring that changes or dependencies of one container do not affect other containers in the stack.

3) Scalability: Containers can be easily scaled up or down based on the demand, allowing for efficient resource utilization.

• Explain the following with Docker:

1) Containers: A container is a lightweight, standalone, executable package that includes everything needed to run a piece of software, including the code, runtime, system tools, system libraries and settings.

2) Images: Images are read-only templates used to create containers. They contain the application code, dependencies, and other necessary files and metadata required to run the app.

3) Dockerfile: A Dockerfile is a text file that contains instructions for building a Docker image. It specifies the base image, environment variables, commands to run during the image build process and other configuration settings necessary to create the image.

Q) Compare the following : Virtual machine & Container

Ans) Container and virtual machine

Virtual Machine vs Container

- | | |
|---|---|
| 1) Application running on VM system or hypervisor, are in a different OS. | 1) While application running in a container environment shares a single OS. |
| 2) VM virtualizes the computer system, meaning its hardware. | 2) While a container virtualizes the OS or software only. |
| 3) VM size is very large, generally in GB. | 3) While the size of container is very light. |
| 4) E.g., Type 1 hypervisor bare KVM, Xen, etc. | 4) E.g., Rancher, OS, PhotonOS, etc. |

Q) Image and container is same? Or

Ans) Both are not same. Both have some differences.

Image

- 1) It is a blueprint of the container.
- 2) Image is a logical identity, no physical.
- 3) Images are created only once no matter how many times it is used.
- 4) Sharing of docker image is possible.

Container

- 1) It is the instance of the image.
- 2) The container is real world identity.
- 3) The containers are created any kind of time using an image.
- 4) Sharing of containers is not exactly possible.

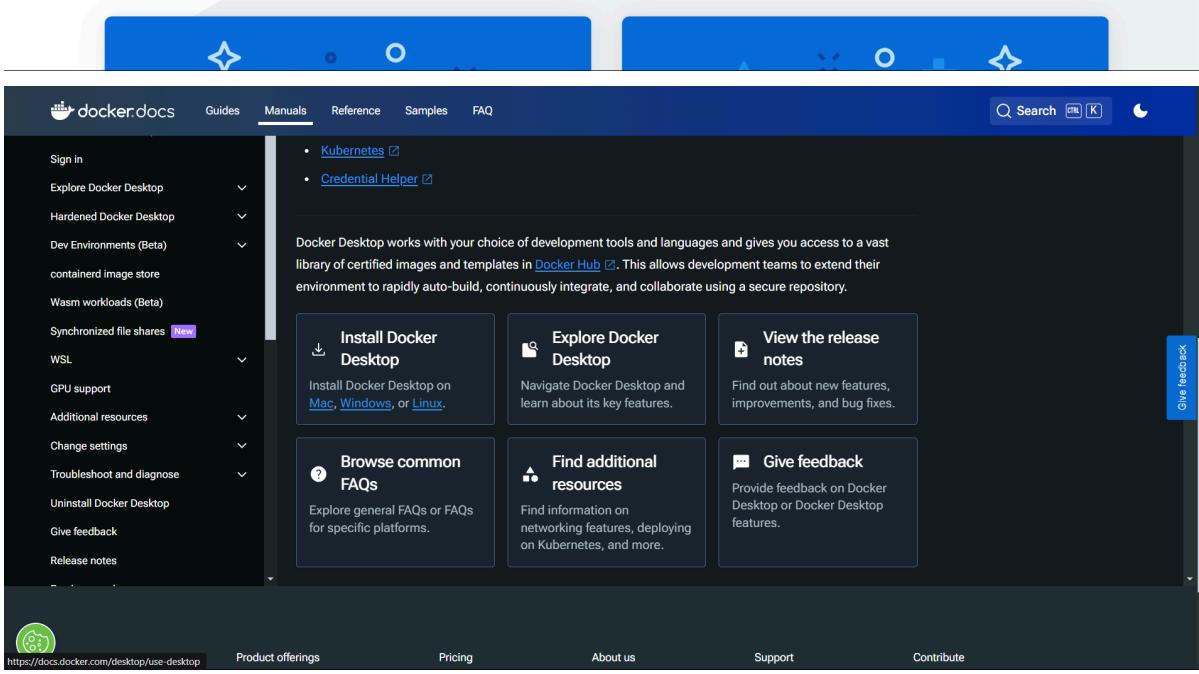
AT

SP
93/3/2024

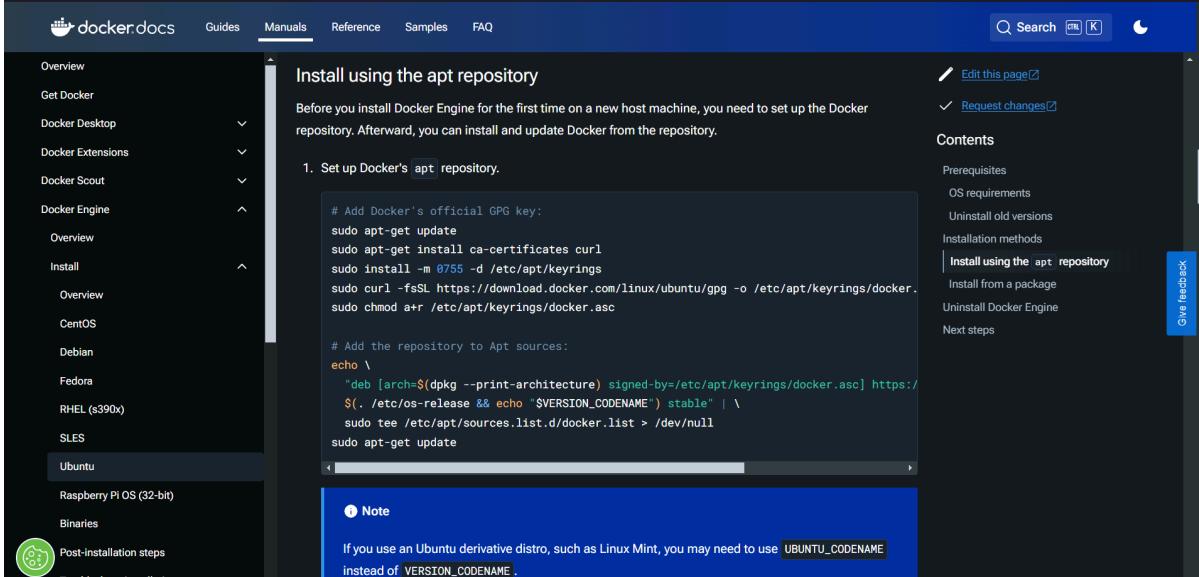
Output:



The screenshot shows the Docker homepage with the title "Get Started with Docker" and a subtitle "Build applications faster and more securely with Docker for developers". It features two main calls-to-action: "Learn how to install Docker" and "Download for Windows".



This screenshot shows the "Docker Desktop" documentation page under the "Manuals" tab. It includes sections for Kubernetes and Credential Helper, and a central content area about Docker Desktop's capabilities. There are also several interactive cards: "Install Docker Desktop", "Explore Docker Desktop", "View the release notes", "Browse common FAQs", "Find additional resources", and "Give feedback".



This screenshot shows the "Ubuntu" section of the Docker Engine documentation. It starts with a heading "Install using the apt repository" and a note about setting up the Docker repository. Below this is a code block for adding the Docker GPG key and repository to Apt sources. A "Note" section at the bottom explains that for Ubuntu derivatives like Linux Mint, users should use `UBUNTU_CODENAME` instead of `VERSION_CODENAME`. The sidebar on the left lists various Docker documentation categories.

The screenshot shows the Docker documentation page for Ubuntu. The left sidebar lists various operating systems and platforms. The main content area is titled "Install the Docker packages." It shows two options: "Latest" and "Specific version". Below this, it says "To install the latest version, run:" followed by a command line snippet: `$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-`. Step 3 says "Verify that the Docker Engine installation is successful by running the `hello-world` image." Below this is another command line snippet: `$ sudo docker run hello-world`. A note explains: "This command downloads a test image and runs it in a container. When the container runs, it prints a confirmation message and exits." A "Tip" section at the bottom suggests handling errors if run without root.

The screenshot shows the Docker Desktop application window. The left sidebar has icons for Containers, Images, Volumes, Builds, Dev Environments (BETA), Docker Scout, and Extensions. The main area is titled "Containers" with a "Give feedback" link. It features a graphic of three overlapping containers. Below it, a heading says "Your running containers show up here" with the subtext "A container is an isolated environment for your code". There are two cards: one titled "What is a container?" with a duration of "5 mins" and another titled "How do I run a container?" with a duration of "6 mins". At the bottom, there's a "View more in the Learning center" link. The footer shows "Engine running", system resources (RAM 1.64 GB, CPU 0.25%), and a signed-in status. The version is v4.28.0.

Name	Tag	Status	Created	Size	Actions
ubuntu ca2b0f26964c	latest	In use	17 days ago	77.86 MB	

Showing 1 item

walkthroughs

- How do I run a container? 6 mins
- Run Docker Hub images 5 mins

[View more in the Learning center](#)

```
root@e56cf5d99e5: /
```

```
student@LAB301PC05:~$ docker run ubuntu
student@LAB301PC05:~$ sudo docker pull ubuntu:20.04-
invalid reference format
student@LAB301PC05:~$ sudo docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
^Cstudent@LAB301PC05:~$ sudo docker pull ubuntu:latest
latest: Pulling from library/ubuntu
bccd10f490ab: Pull complete
Digest: sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

What's Next?
1. Sign in to your Docker account → docker login
2. View a summary of image vulnerabilities and recommendations → docker scout quickview ubuntu:latest
student@LAB301PC05:~$ sudo docker run -ti --rm ubuntu
root@e56cf5d99e5:/#
```

Experiment No: 10

Aim: To Study and implement container orchestration using Kubernetes.

Theory:

- 1) Explain need of container orchestration tool
 - ⇒ Container orchestration tools are needed to manage deployment scale, and automate containers effectively.
 - They ensure containers run as intended, manage resource allocation, handle networking, storage, and maintain high availability across a cluster of machines.
 - Container orchestration tools automate various tasks involved in managing containers, such as deployment, scaling, load balancing, health monitoring, and recovery.
 - As the number of containers and the size of the infrastructure grows, manual management becomes impractical.

2) What is Kubernetes? Describe its features.

- ⇒ Kubernetes is an open-source container orchestration platform designed to automate the deployment, scaling, and management of containerized applications.

It's features are :-

- i) Automated rollouts and rollbacks
- ii) Service discovery and load balancing.

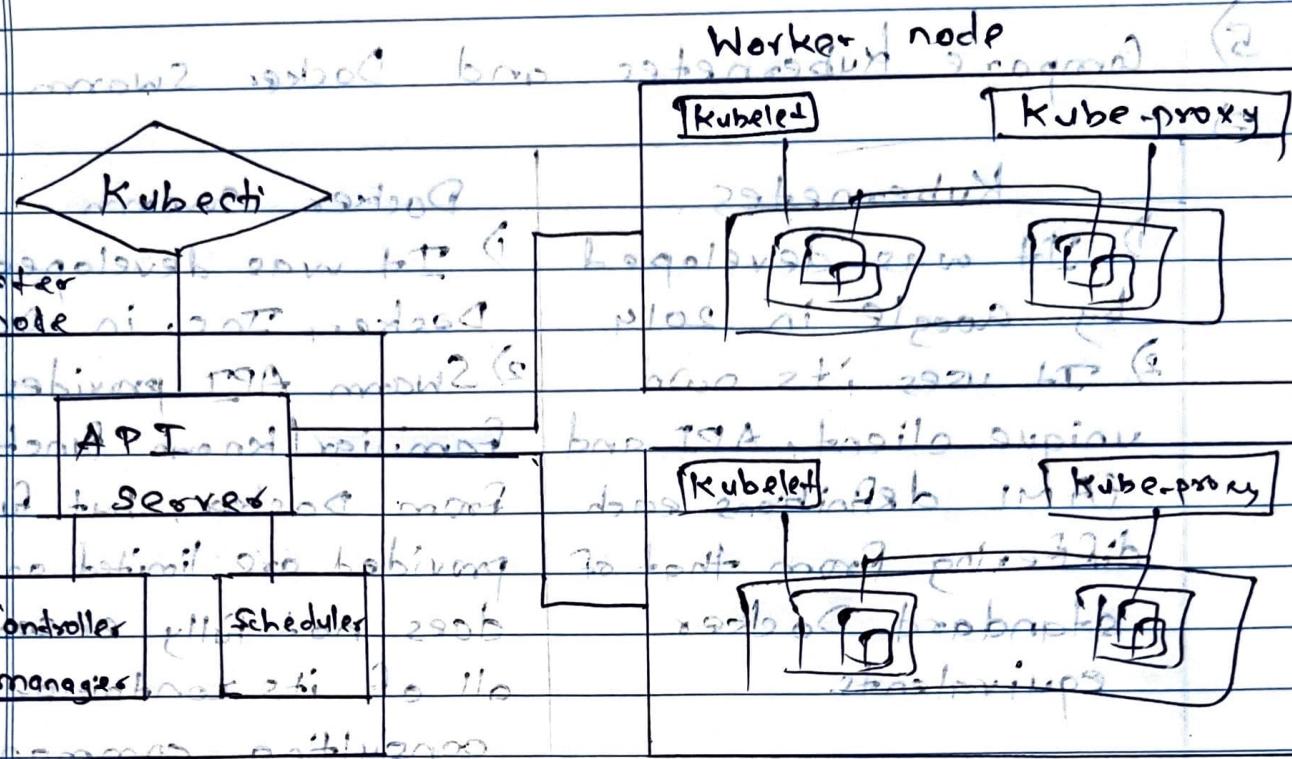
Cloud Formation

- iii) Storage orchestration: $\frac{1}{2}$ OT 1 min
- iv) Self-healing: $\frac{1}{2}$ min
- v) Secret and configuration management
- vi) Automatic bin packing
- vii) Batch execution
- ix) Horizontal scaling to less weight
- x) Multi-IP & data distribution
- x) Designed for extensibility

Q) Explain Kubernetes components & its working mechanism and its architecture.

- Ans) Kubernetes components include the Master and Worker nodes to establish a cluster. The Master node oversees the cluster and manages its state through various components such as the API server, controller manager, scheduler, and etcd. Worker nodes host the containers and run the necessary Kubernetes components like kubelet, kube-proxy, and container runtime.

Kubernetes has a client-server architecture and has master and worker nodes, with the master being installed on a single Linux system and the nodes on many Linux workstations. It follows a simple bootstrap mechanism where each node has persistent storage.



abnormal, broad ratio no (e) from a stronger LTO (e)

4) Differences b/w between PODs and nodes

↳ nodes is a single physical machine or virtual machine
label, pod and POD, in is sub podNode windows

1) A pod is smallest deployable unit in Kubernetes

2) It represents a single instance of a running process in your cluster.

3) PODs are ephemeral and can be created, destroyed or replicated dynamically based on workload requirement.

1) A node is a physical or virtual machine in a Kubernetes cluster.

2) It is the underlying infrastructure where POD runs

3) Nodes are responsible for running and managing PODs providing the necessary computing, networking and storage resources.

5) Compare Kubernetes and Docker Swarm

Kubernetes

- 1) It was developed by Google in 2014
- 2) It uses its own unique client, API and YAML definitions each differing from that of standard Docker equivalents.

Docker Swarm

- 1) It was developed by Docker, Inc. in 2013
- 2) Swarm API provides many familiar/known functionalities from Docker, but functionalities provided are limited and it does not fully encompass all of its containers' consulting commands.

- 3) It supports a more complex, flexible architecture with stronger service guarantees due to which performance slows down in comparison to Kubernetes.
- 4) It supports auto-scaling.

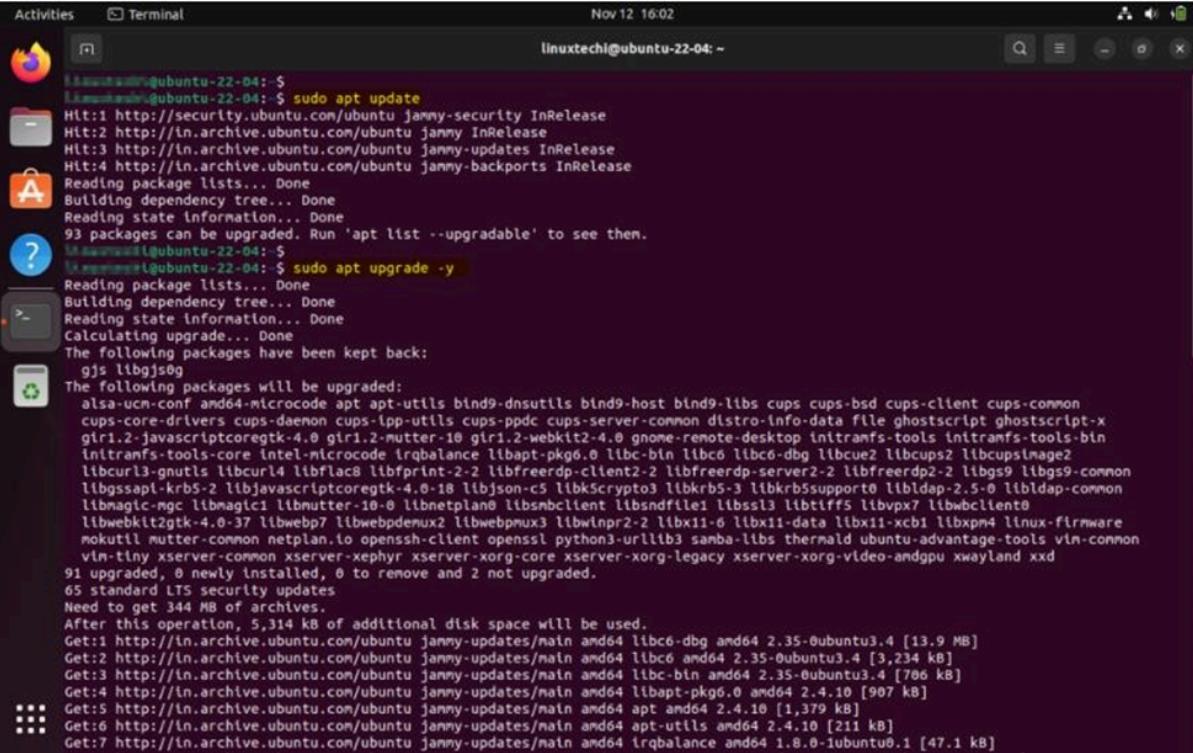
- 3) On other hand, supports simple architecture, so in terms of sheer speed, it always has an added advantage.
- 4) It cannot do auto-scaling.

some Q&A while doing in class

(A)

(B)

(C)

Output:


```

Activities Terminal Nov 12 16:02
linuxtechi@ubuntu-22-04:~$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
93 packages can be upgraded. Run 'apt list --upgradable' to see them.
linuxtechi@ubuntu-22-04:~$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  gjs libgjs0g
The following packages will be upgraded:
  alsu-ucm-conf amd64-microcode apt apt-utils bind9-dnsutils bind9-host bind9-libs cups cups-bsd cups-client cups-common
  cups-core-drivers cups-daemon cups-ipp-utils cups-ppdc cups-server-common distro-info-data file ghostscript ghostscript-x
  gir1.2-javascriptcoregtk-4.0 gir1.2-mutter-10 gir1.2-webkit2-4.0 gnome-remote-desktop intransfs-tools intransfs-tools-bin
  intransfs-tools-core intel-microcode irqbalance libapt-pkg6.0 libc-bin libc6 libc6-dbg libcue2 libcurl2 libcupsmime2
  libcurl3-gnutls libcurl4 libflac8 libfprint-2-2 libfreerdp-client2-2 libfreerdp-server2-2 libfreerdp2-2 libgs9 libgs9-common
  libgssapi-krb5-2 libjavascriptcoregtk-4.0-18 libjson-c5 libksyptool libkrb5-3 libkrb5support0 libldap-2.5-0 libldap-common
  libmagic-noc libmaglci1 libmutter-10-0 libnetplan8 libnmbclient libnsdfile1 libssl3 libtiff5 libvpx7 libwbclient0
  libwebkit2gtk-4.0-37 libwebp7 libwebpdmux2 libwebpmux3 libwinpr2-2 libxii-6 libxii-data libxii-xcb1 libxpm4 linux-firmware
  mokutil mutter-common netplan.io openssh-client openssl python3-urllib3 samba-libs thermald ubuntu-advantage-tools vim-common
  vlm-tiny xserver-common xserver-xephyr xserver-xorg-core xserver-xorg-legacy xserver-xorg-video-andgpu xwayland xxd
91 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
65 standard LTS security updates
Need to get 344 MB of archives.
After this operation, 5,314 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libbc6-dbg amd64 2.35-0ubuntu3.4 [13.9 MB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libbc6 amd64 2.35-0ubuntu3.4 [3,234 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libbc6-bin amd64 2.35-0ubuntu3.4 [786 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapt-pkg6.0 amd64 2.4.10 [907 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apt amd64 2.4.10 [1,379 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apt-utils amd64 2.4.10 [211 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 irqbalance amd64 1.8.0-1ubuntu0.1 [47.1 kB]

[1] 11866 ? 0:00 sudo apt install ca-certificates curl gnupg wget apt-transport-https -y
[sudo] password for linuxtechi:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wget is already the newest version (1.21.2-2ubuntu1).
wget set to manually installed.
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
ca-certificates set to manually installed.
gnupg is already the newest version (2.2.27-3ubuntu2.1).
gnupg set to manually installed.
The following NEW packages will be installed:
  apt-transport-https curl
0 upgraded, 2 newly installed, 0 to remove and 2 not upgraded.
Need to get 196 kB of archives.
After this operation, 623 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 apt-transport-https all 2.4.10 [1,510 B]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64 7.81.0-1ubuntu1.14 [194 kB]
Fetched 196 kB in 2s (81.8 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 182047 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.4.10_all.deb ...
Unpacking apt-transport-https (2.4.10) ...
Selecting previously unselected package curl.
Preparing to unpack .../curl_7.81.0-1ubuntu1.14_amd64.deb ...
Unpacking curl (7.81.0-1ubuntu1.14) ...
Setting up apt-transport-https (2.4.10) ...
Setting up curl (7.81.0-1ubuntu1.14) ...
Processing triggers for man-db (2.10.2-1) ...
linuxtechi@ubuntu-22-04:~$ sudo install -m 0755 -d /etc/apt/keyrings
linuxtechi@ubuntu-22-04:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
linuxtechi@ubuntu-22-04:~$ sudo chmod a+r /etc/apt/keyrings/docker.gpg
linuxtechi@ubuntu-22-04:~$ 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

```

```
linoxbuzz@ubuntu-22-04: $ sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras git git-man liberror-perl libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs
  git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-rootless-extras docker-compose-plugin git git-man
  liberror-perl libslirp0 pigz slirp4netns
0 upgraded, 12 newly installed, 0 to remove and 2 not upgraded.
Need to get 118 MB of archives.
After this operation, 430 MB of additional disk space will be used.
Get:1 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io amd64 1.6.24-1 [28.6 MB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.10 [954 kB]
Get:5 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-buildx-plugin amd64 0.11.2-1-ubuntu.22.04-jammy [28.2 MB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.10 [3,166 kB]
Get:7 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-cli amd64 5:24.0.7-1-ubuntu.22.04-jammy [13.3 MB]
Get:8 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce amd64 5:24.0.7-1-ubuntu.22.04-jammy [22.6 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libslirp0 amd64 4.6.1-1bulld1 [61.5 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 slirp4netns amd64 1.0.1-2 [28.2 kB]
Get:11 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-rootless-extras amd64 5:24.0.7-1-ubuntu.22.04-jammy [9,030 kB]
Get:12 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-compose-plugin amd64 2.21.0-1-ubuntu.22.04-jammy [11.9 MB]
Fetched 118 MB in 8s (14.4 MB/s)
```

```
linoxbuzz@ubuntu-22-04: $ sudo usermod -aG docker $USER
linoxbuzz@ubuntu-22-04: $ newgrp docker
linoxbuzz@ubuntu-22-04: $ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2023-11-12 16:20:53 IST; 1min 54s ago
    TriggeredBy: ● docker.socket
      Docs: https://docs.docker.com
 Main PID: 3114 (dockerd)
    Tasks: 8
   Memory: 27.3M
      CPU: 1.065s
     CGroup: /system.slice/docker.service
             └─3114 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Nov 12 16:20:51 ubuntu-22-04 systemd[1]: Starting Docker Application Container Engine...
Nov 12 16:20:51 ubuntu-22-04 dockerd[3114]: time="2023-11-12T16:20:51.771454830+05:30" level=info msg="Starting up"
Nov 12 16:20:51 ubuntu-22-04 dockerd[3114]: time="2023-11-12T16:20:51.774072108+05:30" level=info msg="detected 127.0.0.53 nameserver"
Nov 12 16:20:52 ubuntu-22-04 dockerd[3114]: time="2023-11-12T16:20:52.051794664+05:30" level=info msg="Loading containers: start."
Nov 12 16:20:53 ubuntu-22-04 dockerd[3114]: time="2023-11-12T16:20:53.028261474+05:30" level=info msg="Loading containers: done."
Nov 12 16:20:53 ubuntu-22-04 dockerd[3114]: time="2023-11-12T16:20:53.23111103+05:30" level=info msg="Docker daemon" commit=311b9f>
Nov 12 16:20:53 ubuntu-22-04 dockerd[3114]: time="2023-11-12T16:20:53.233235025+05:30" level=info msg="Daemon has completed initialization"
Nov 12 16:20:53 ubuntu-22-04 dockerd[3114]: time="2023-11-12T16:20:53.351693568+05:30" level=info msg="API listen on /run/docker.sock"
Nov 12 16:20:53 ubuntu-22-04 systemd[1]: Started Docker Application Container Engine.
```

```
linoxbuzz@ubuntu-22-04:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload Total   Spent    Left Speed
100 89.3M  100 89.3M    0     0  9.8M      0  0:00:09  0:00:09  --:--:-- 13.4M
linoxbuzz@ubuntu-22-04:~$ 
linoxbuzz@ubuntu-22-04:~$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
linoxbuzz@ubuntu-22-04:~$ 
linoxbuzz@ubuntu-22-04:~$ minikube version
minikube version: v1.32.0
commit: 8220a6eb95f0a4d75f7f2d7b14cef975f050512d
linoxbuzz@ubuntu-22-04:~$ 
linoxbuzz@ubuntu-22-04:~$ 
```

```
linuxbuzz@ubuntu-22-04:~$ curl -LO https://storage.googleapis.com/kubernetes-release/release/'curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt'/bin/linux/amd64/kubectl
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload Total   Spent    Left Speed
100 47.5M  100 47.5M    0     0  8524k      0  0:00:05  0:00:05  --:--:--  9.8M
linuxbuzz@ubuntu-22-04:~$ 
linuxbuzz@ubuntu-22-04:~$ chmod +x kubectl
linuxbuzz@ubuntu-22-04:~$ 
linuxbuzz@ubuntu-22-04:~$ sudo mv kubectl /usr/local/bin/
linuxbuzz@ubuntu-22-04:~$ 
linuxbuzz@ubuntu-22-04:~$ kubectl version -o yaml
clientVersion:
  buildDate: "2023-10-18T11:42:52Z"
  compiler: gc
  gitCommit: a8a1abc25cad87333840cd7d54be2efaf31a3177
  gitTreeState: clean
  gitVersion: v1.28.3
  goVersion: go1.20.10
  major: "1"
  minor: "28"
  platform: linux/amd64
kustomizeVersion: v5.0.4-0.20230601165947-6ce0bf390ce3

The connection to the server localhost:8080 was refused - did you specify the right host or port?
linuxbuzz@ubuntu-22-04:~$ 
```

```
linuxbuzz@ubuntu-22-04: $ minikube start --driver=docker
└── minikube v1.32.0 on Ubuntu 22.04 (vbox/amd64)
    └── Using the docker driver based on user configuration
    └── Using Docker driver with root privileges
    └── Starting control plane node minikube in cluster minikube
    └── Pulling base image ...
        └── Downloading Kubernetes v1.28.3 preload ...
            > preloaded-images-k8s-v18-v1...: 403.35 MiB / 403.35 MiB 100.00% 7.23 Mi
            > gcr.io/k8s-minikube/kicbase...: 453.90 MiB / 453.90 MiB 100.00% 7.36 Mi
    └── Creating docker container (CPUs=2, Memory=2200MB) ...
    └── Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
        └── Generating certificates and keys ...
        └── Booting up control plane ...
        └── Configuring RBAC rules ...
    └── Configuring bridge CNI (Container Networking Interface) ...
        └── Using image gcr.io/k8s-minikube/storage-provisioner:v5
    └── Verifying Kubernetes components...
    └── Enabled addons: storage-provisioner, default-storageclass
    └── Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
linuxbuzz@ubuntu-22-04:~$
```

```
linuxbuzz@ubuntu-22-04:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

```
linuxbuzz@ubuntu-22-04:~$
```

```
linuxbuzz@ubuntu-22-04: $ kubectl get nodes
NAME      STATUS   ROLES   AGE     VERSION
minikube  Ready    control-plane   5m49s   v1.28.3
linuxbuzz@ubuntu-22-04: $
linuxbuzz@ubuntu-22-04: $ 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:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
linuxbuzz@ubuntu-22-04: $
linuxbuzz@ubuntu-22-04:~$
```

```
linuxbuzz@ubuntu-22-04:~$ kubectl create deployment nginx-web --image=nginx
deployment.apps/nginx-web created
linuxbuzz@ubuntu-22-04:~$ kubectl expose deployment nginx-web --type=NodePort --port=80
service/nginx-web exposed
linuxbuzz@ubuntu-22-04:~$ kubectl get deployment,pod,svc
NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-web     1/1     1           1           40s

NAME                         READY   STATUS    RESTARTS   AGE
pod/nginx-web-5b757f798d-qnbzq 1/1     Running   0          39s

NAME              TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes  ClusterIP   10.96.0.1   <none>       443/TCP   12m
service/nginx-web  NodePort   10.104.243.119  <none>       80:30523/TCP 27s
linuxbuzz@ubuntu-22-04:~$
```

```
linuxbuzz@ubuntu-22-04:~$ 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 | enabled | ✓ Kubernetes |
| efk | minikube | disabled | 3rd party (Elastic) |
| freshpod | minikube | disabled | Google |
| gcp-auth | minikube | disabled | Google |
| gvisor | minikube | disabled | minikube |
| headlamp | minikube | disabled | 3rd party (kinvolk.io) |
| helm-tiller | minikube | disabled | 3rd party (Helm) |
| inacel | minikube | disabled | 3rd party (InAccel) [info@inacel.com] |
| ingress | minikube | disabled | Kubernetes |
| ingress-dns | minikube | disabled | minikube |
| inspektor-gadget | minikube | disabled | 3rd party (inspektor-gadget.io) |
| istio | minikube | disabled | 3rd party (Istio) |
| istio-provisioner | minikube | disabled | 3rd party (Istio) |
| kong | minikube | disabled | 3rd party (Kong HQ) |
| kubeflow | minikube | disabled | 3rd party |
| kubevirt | minikube | disabled | 3rd party (KubeVirt) |
| logviewer | minikube | disabled | 3rd party (unknown) |
| metallb | minikube | disabled | 3rd party (MetallLB) |
| metrics-server | minikube | disabled | Kubernetes |
| nvidia-device-plugin | minikube | disabled | 3rd party (NVIDIA) |
| nvidia-driver-installer | minikube | disabled | 3rd party (Nvidia) |
| nvidia-gpu-device-plugin | minikube | disabled | 3rd party (Nvidia) |
| olm | minikube | disabled | 3rd party (Operator Framework) |
| pod-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) |
+-----+-----+-----+-----+
linuxbuzz@ubuntu-22-04:~$ minikube addons enable dashboard
💡 dashboard is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
■ Using image docker.io/kubernetesui/dashboard:v2.7.0
■ Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
💡 Some dashboard features require the metrics-server addon. To enable all features please run:
minikube addons enable metrics-server

★ The 'dashboard' addon is enabled
linuxbuzz@ubuntu-22-04:~$ minikube addons enable ingress
💡 ingress is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
■ Using image registry.k8s.io/ingress-nginx/controller:v1.9.4
■ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20231011-8b53cabe0
■ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20231011-8b53cabe0
💡 Verifying ingress addon...
★ The 'ingress' addon is enabled
linuxbuzz@ubuntu-22-04:~$ minikube dashboard
💡 Verifying dashboard health ...
💡 Launching proxy ...
💡 Verifying proxy health ...
💡 Opening http://127.0.0.1:36529/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
```

The screenshot shows the Kubernetes Dashboard interface running in a Firefox browser. The URL is 127.0.0.1:36529/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/port/. The dashboard is titled "kubernetes" and shows the "Workloads" section. On the left sidebar, there are links for Workloads, Service, Config and Storage, and others like Cron Jobs, Daemon Sets, Deployments, etc. The main area displays "Workload Status" with three tabs: Deployments, Pods, and Replica Sets. The "Deployments" tab shows one deployment named "nginx-web" with the image "nginx" and labels "app: nginx-web". The "Pods" tab shows one pod named "nginx-web-5b757f798d-qnbzq" with the image "nginx" and labels "app: nginx-web", "pod-template-hash: 5b757f798d", and "node: minikube". The pod status is "Running" with 0 restarts. The "Replica Sets" tab is empty.

Assignment No: 1

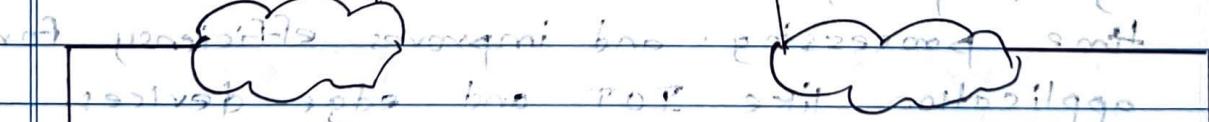
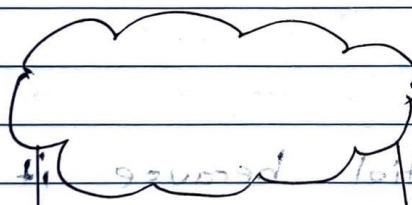
Q.1 What is fog computing? and explain the need of fog computing.

Fog computing is the term coined by Cisco that refers to extending cloud computing to the edge of the enterprise's network.

Thus, it is also known as Edge Computing or Fogging.

It facilitates the operations of computing, storage and networking services between end devices and computing data centers.

→ Advantages of fog computing



- 1) The devices comprising the fog infrastructure are known as fog nodes.
- 2) In fog computing, all the storage capabilities, computing capabilities, data along with the applications are placed between the cloud and the physical host.
- 3) All these functionalities are placed more towards the host. This makes processing faster as it is done almost at the point where data is created.
- 4) It improves the efficiency of the system and is also used to ensure increased security.

- Need:

It is essential because it addresses the limitations of centralized cloud computing.

By bringing computation and storage closer to the edge of network.

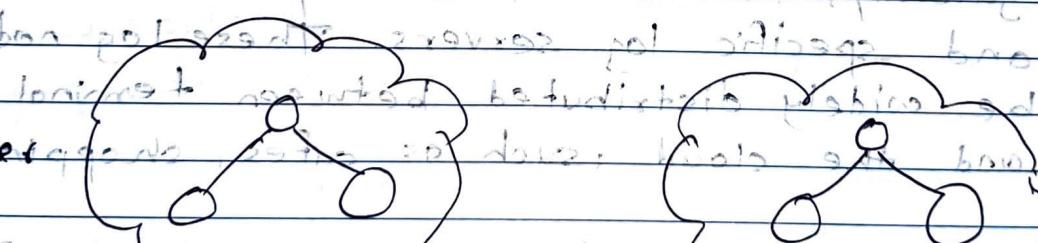
- Fog computing reduces latency, enhances real-time processing, and improves efficiency for application like IoT and edge devices.
- This decentralized approach also enhances privacy and security by processing data closer to its source, reducing the need for extensive data transfers to centralized servers.

Q.2

Explain the architecture of fog computing.

The fog computing architecture is generally composed of three different working layers, namely a terminal layer, a fog layer, and a cloud layer.

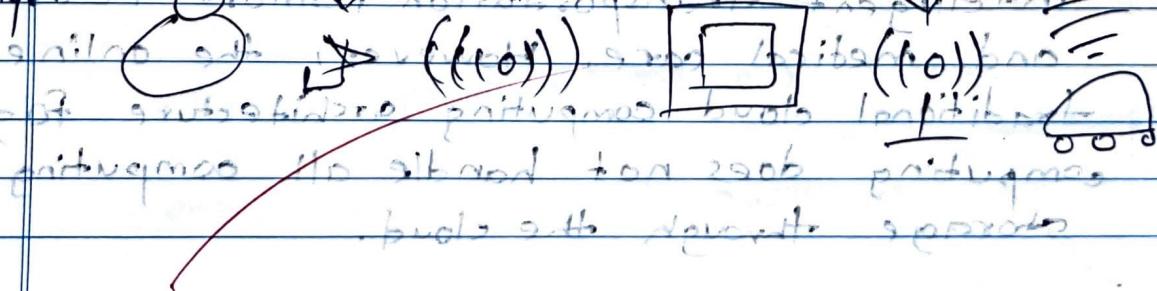
The three-layer architecture of fog computing is as follows:



To solve a problem in terminal, it can be solved in a local way.

It can be solved in a local way or it can be solved in a central way.

Edge and terminal are local solving, not central solving.



1) Edge layer :- This layer is closest to end users and end devices and consists of various IoT nodes or intelligent devices, such as sensors, mobile phones, intelligent vehicles, smart cards and readers. What is special is that although these devices have the capability of computing and usually only use these devices to carry out intelligent sending of entity objects or events and upload the collected sensing data to the upper layer for subsequent processing and storage.

2) Fog layer :- This layer is located at the edge of network and consists of a large no. of fog nodes. These fog nodes usually contain routers, gateways, switches, access points, base stations and specific log servers. These log nodes can be widely distributed between terminal devices and the cloud, such as cafes, shopping, etc.

3) Cloud layer :- Cloud layer is composed of multiple servers and storage devices with high performance to provide various intelligent application services such as smart home, intelligent transportation, smart factory and medical care. However, the online traditional cloud computing architecture fog computing does not handle all computing and storage through the cloud.

Q.3 Explain applications of Fog computing. (6)
⇒ Fog finds applications in various domains:-

1) IoT

⇒ Facilitates real-time processing of data from connected devices, reducing latency and enhancing overall efficiency.

2) Smart cities

⇒ Manages and analyzes data from sensors, cameras and other devices to maintain city services, traffic flow and resource utilization.

3) Healthcare

⇒ Enables real-time monitoring of patient data, supports remote diagnostics and facilitates quicker response time in critical situations.

4) Manufacturing

⇒ Enhances operational efficiency by providing real-time analytics for monitoring equipment, optimizing production processes and reduce downtime.

5) Retail

⇒ Supports personalized and location-based services, improves inventory management and enhances the overall customer shopping experience.

- 6) Edge AI \rightarrow 3.0 smartgrid intelligent 8.0
 \Rightarrow Facilitates non-artificial intelligent processing.

TOP 6

- 7) Energy Management 1.0 \rightarrow 2.0
 \Rightarrow Optimizes energy consumption by analyzing data from smartgrids and devices.

Q.4 Compare Fog and Cloud computing. 6

Fog computing	Cloud computing
1) Fog computing has a low latency.	1) Cloud computing has high latency compared to cloud.
2) Response time of the system is high.	2) Response time of the system is low.
3) Fog computing has high security.	3) Cloud computing has less security compared to fog computing.
4) Success speed is high even more compared to cloud computing.	4) Success speed is high depending on the VM connectivity.
5) Multiple data sources and devices.	5) Multiple data sources can be integrated.
6) Mobility is high.	6) In cloud computing, the mobility is limited.
7) Supported in fog computing (Location awareness).	7) Partially supported in cloud computing (location awareness).

Q.5 Case study of Fog Computing b/w 6

→ ~~we have application b/w its implementation & IoT is~~
~~with the advancement of computing has a lot~~
~~relation about to capabilities of its own~~

Introduction:

The IoT connects objects through the Internet without human involvement. IoT includes sensors, smartphones, etc.

- Cloud computing with its storage and processing power is vital for IoT applications, but it has limitation like centralization. To address this, fog computing bring data and computation closer to users, improving latency and quality of service.

Architecture of Fog Computing

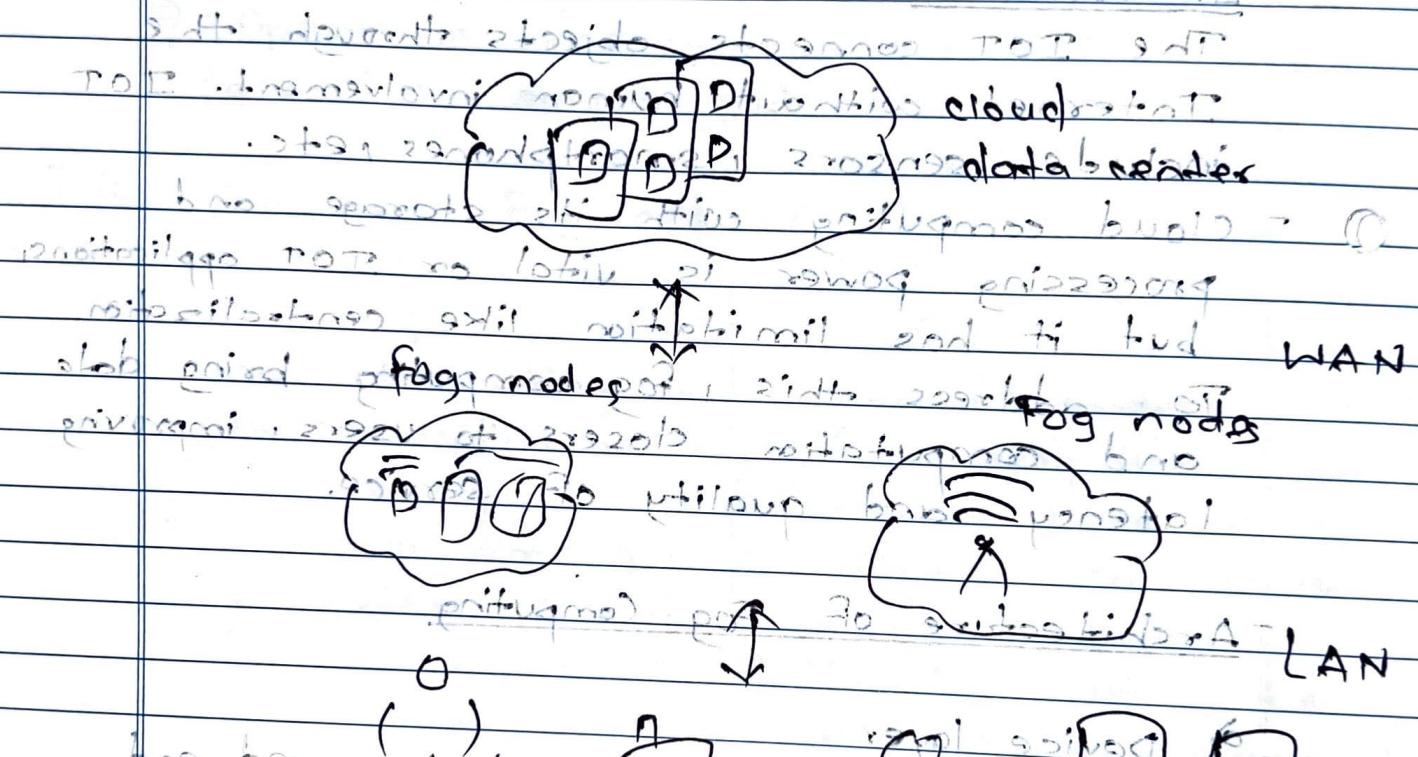
a) Device layer

⇒ Contains different IoT devices and end devices such as mobile phones, smart vehicles, cards, etc. These devices are distributed geographically.

b) Fog Layer

⇒ This layer plays an important role in transmission between devices and cloud computing layers.

- c) Cloud layer(s) part → photo 920 7.2
⇒ It corresponds to cloud intelligence and can store and process massive amounts of data, depending on the capabilities of data center



Characteristics of fog computing

- 1) Geographical distribution
- 2) Decentralization
- 3) Location Awareness
- 4) Real-time interaction
- 5) Low latency.

(R)

Ex 26/3/2020

Assignment No: 2

Aim: Explore and compare the similar types of services provided by AWS, Azure and Google Cloud platforms.

⇒ Amazon Web Service (AWS), Microsoft Azure, and Google Cloud Platform (GCP) are the three major players in the cloud computing market, each offering a wide range of services to meet the diverse needs of businesses and developers.

1) Compute Service:

a) AWS

⇒ Amazon Elastic Compute Cloud (EC2) provides resizable compute capacity in the cloud.

b) Azure

⇒ Azure Virtual Machines (VMs) offers scalable computing resources with various configuration.

c) GCP

⇒ Google Compute Engine (GCE) allows users to create virtual machines in Google's data center.

2) Storage Services:

a) AWS

⇒ Amazon Simple Storage Service (S3) provides scalable object storage.

Cloud Computing

b) Azure

⇒ Azure blob storage offers massively scalable object storage for unstructured data.

Amazon S3 (AWS) gives us the same (⇒)

c) GCP methods build storage blobs

⇒ Google Cloud Storage (GCS) provides durable

and highly available object storage.

Amazon S3 (AWS) and Google Cloud Storage

3) Database Services how centralized

a) AWS

⇒ Amazon relational database service (RDS) offers managed relational databases, like

MySQL, PostgreSQL, etc. (AWS)

blobs and other storage solutions integrated

b) Azure

⇒ Azure SQL database provides fully managed relational databases (Azure SQL) with automatic scaling, redundancy, failover

c) GCP

⇒ Google Cloud SQL offers managed MySQL, PostgreSQL and SQLite server databases.

blobs and other storage solutions integrated

d) Networking Services:

a) AWS

⇒ Amazon Virtual Private Cloud (VPC) enables users to launch AWS resources on a virtual network.

b) Azure

⇒ Azure functions provides serverless Azure virtual network allows users to provide isolated networks in the cloud.

c) GCP

⇒ Google virtual Private Cloud (VPC) offers global virtual networking for Google Cloud resources.

5) Serverless Computing:

a) AWS

⇒ AWS lambda allows users to run code without provisioning or managing servers.

b) Azure

⇒ Azure functions provides serverless compute for event-driven applications.

c) GCP

⇒ Google cloud functions enables users to run event-driven functions without managing infrastructure.

① SP

26/3/2024

CCL MINI-PROJECT

Title: Urban Garden

**Mohib Abbas Sayed 2103158
Hamza Sayyed 2103159
Om Shete 2103163**

(A)
2021
10/12/2021

Description of problem statement

Many people want to buy plants and are directly concerned with the nursery store. But sometimes people do not know specific information about certain plants and the seller does not have technical skills. Build an online nursery store website so that customers can compare prices, view descriptions, and add reviews to a product that helps the customers for a pleasant shopping experience. After the order is placed, an order confirmation report can be viewed by the client for review.

The main objective of this E-Commerce Website project is to create an online nursery store website that helps customers compare prices, view descriptions, can comment on customization ideas, and follow planting tips that promote gardening. Customer service is essential. So, each customer should have a pleasant shopping experience.

1. Offer personalized recommendations for plants based on the customer's preferences and location.
2. Showcase the variety of plants available for purchase.
3. To attract new customers
4. Increase sales by providing an easy-to-use e-commerce platform.
5. Provide helpful resources and guides for gardening enthusiasts.
6. Promote sustainability and eco-friendly practices.

We've containerized our online nursery store project using Docker, encapsulating our application and its dependencies into portable units known as containers. This approach offers flexibility, enabling us to deploy and manage our application consistently across different environments.

For hosting, we've chosen Amazon Web Services (AWS) as our platform and opted to deploy our containers on Amazon Elastic Compute Cloud (EC2) instances. This allows us to have more control over the underlying infrastructure while still benefiting from the scalability and reliability of AWS.

To start, we set up EC2 instances to serve as our hosting environment. We select instance types based on our resource requirements and configure networking and security settings to ensure a secure and accessible environment for our containers.

With our EC2 instances ready, we proceed to deploy our containerized application using Docker. We create and manage Docker containers on these instances, leveraging tools like Docker Compose or Docker Swarm for orchestration.

Deployment is straightforward; we simply SSH into our EC2 instances, pull the Docker images containing our application and run the containers. We can manage multiple containers across different EC2 instances, ensuring high availability and fault tolerance.

To monitor and maintain our application, we utilize AWS CloudWatch for logging and monitoring. CloudWatch allows us to track metrics like CPU utilization and memory usage, set up alarms for automatic notifications, and analyze container logs for debugging purposes.

Overall, by dockerizing our project and hosting it on AWS EC2 instances, we achieve a scalable, reliable, and easily manageable infrastructure. This setup ensures smooth operation of our online nursery store, providing customers with a seamless shopping experience while allowing us to efficiently manage our resources and scale as needed.

Requirement Specification

Understanding the needs of both customers and sellers, we have outlined comprehensive requirements to ensure the successful development of GreenThumb Hub. From user-friendly browsing features to robust backend management tools, our specifications cover all aspects necessary for a seamless online shopping experience.

User Requirements:

- Ability to browse plant varieties: Users should be able to explore a wide range of plant varieties available in the online nursery store.
- Access to detailed plant descriptions including care instructions: Each plant listing should include comprehensive descriptions providing information such as plant species, origin, growth habits, sunlight and water requirements, and care instructions.
- Price comparison feature: Users should be able to compare prices of similar plant varieties offered by different sellers.
- Review system for customers: Users should be able to leave reviews and ratings for purchased plants.
- Order placement and viewing of order confirmation report: Users should be able to easily place orders for selected plants through a seamless checkout process. After placing an order, users should receive a confirmation email containing details of their order.

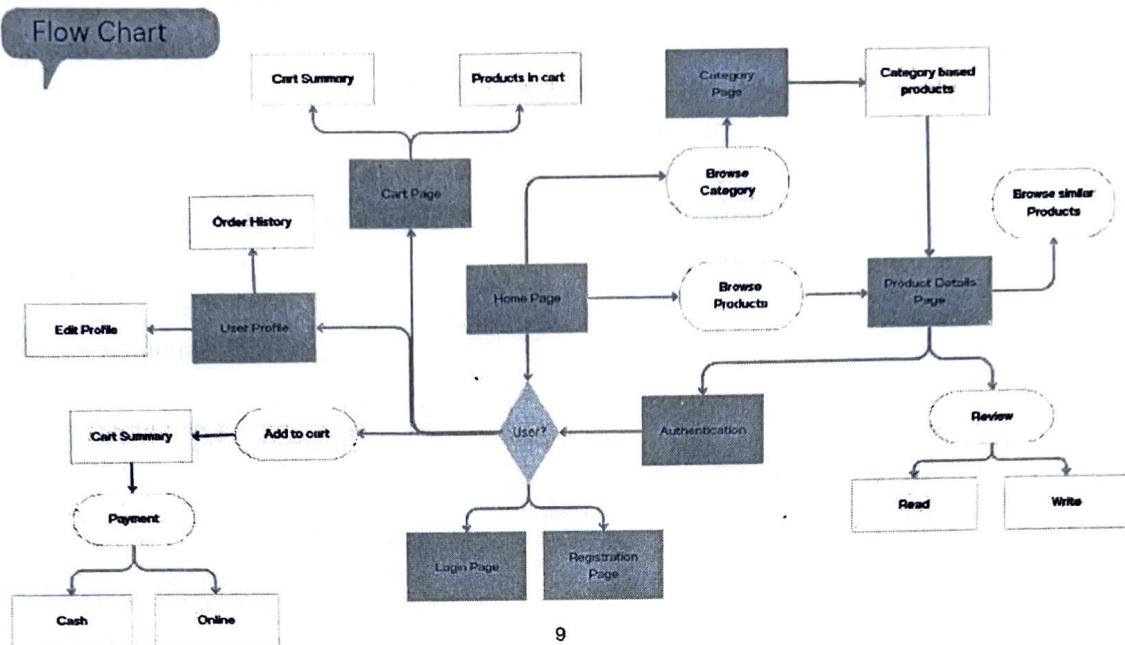
Seller Requirements:

- Easy product management interface: Sellers should have access to an intuitive dashboard for managing their product listings. This interface should allow sellers to add new plant varieties, update descriptions and prices, and remove listings as needed.
- Ability to update plant descriptions and prices: Sellers should be able to edit plant descriptions to provide accurate and up-to-date information to customers. They should also have the flexibility to adjust prices based on market conditions and inventory levels.
- Access to customer reviews and ratings: Sellers should have visibility into customer reviews and ratings for their products. This feedback enables sellers to assess customer satisfaction levels and make improvements if necessary.

Technical Requirements:

- Dockerized application for easy deployment and scalability: The application should be containerized using Docker to ensure consistency in deployment across different environments. Containerization simplifies deployment and scaling processes, making it easier to manage the application's lifecycle.
- Hosting on AWS EC2 for reliability and scalability: The application should be hosted on Amazon Elastic Compute Cloud (EC2) instances to ensure reliability, scalability, and high availability. EC2 instances can be easily scaled up or down to accommodate changes in demand, ensuring optimal performance for users.
- Integration of payment gateway for online transactions: The application should integrate with a secure payment gateway to facilitate online transactions. This integration enables users to make purchases using credit/debit cards, digital wallets, or other payment methods securely.
- User authentication and account management system: The application should have robust user authentication mechanisms to ensure secure access to user accounts. Users should be able to register for accounts, log in securely, and manage their profiles, including updating personal information and viewing order history.

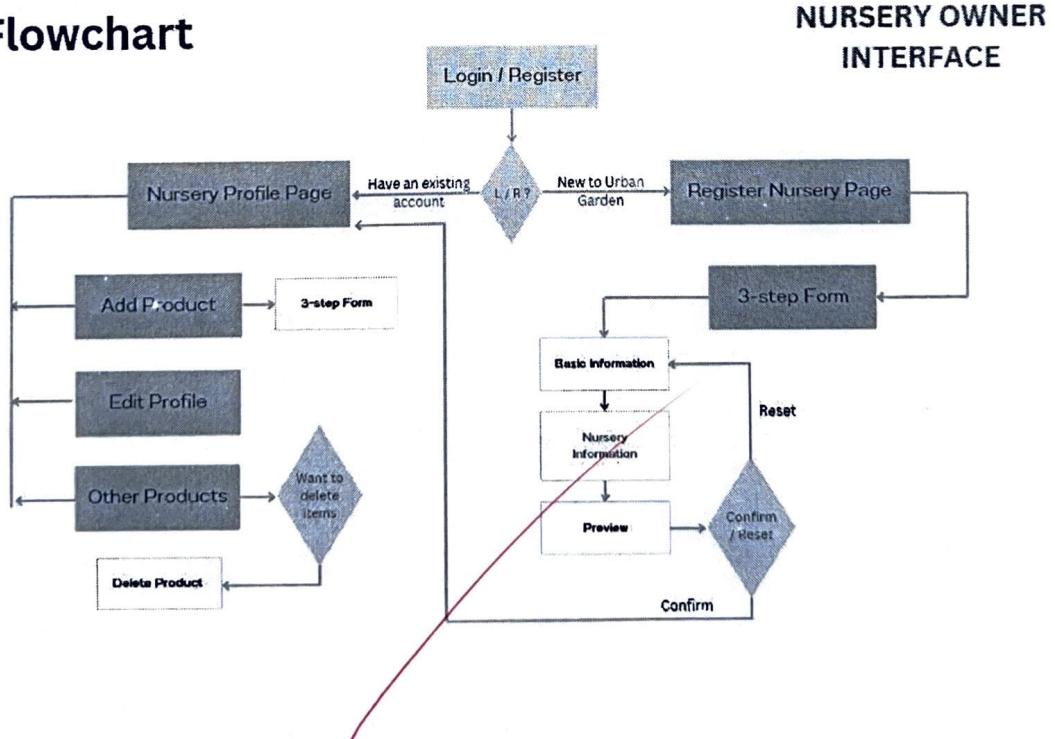
Block diagram /Architecture Diagram



9

A visual representation of the architecture illustrates how different components of Urban Graden interact, from the frontend user interface to the backend databases and external services such as payment gateways. This architecture ensures scalability, reliability, and security throughout the system.

Flowchart



Main code/Major steps

Step 1: Install Docker

Install Docker Engine: Go to the official Docker website (<https://docs.docker.com/get-docker/>) and follow the instructions to download and install Docker Engine for your operating system.

Verify Installation: After installation, open a terminal (or command prompt) and run the following command to verify that Docker is installed correctly:

```
docker --version
```

Step 2: Dockerize Each Component

Create Dockerfile for MongoDB:

Create a file named Dockerfile in your MongoDB directory with the following content:

```
FROM mongo:latest
```

Create Dockerfile for Node.js/Express Backend:

Create a file named Dockerfile in your Node.js/Express project directory with the following content:

```
FROM node:12.19.0
```

```
WORKDIR /app
```

```
COPY package*.json ./
```

```
RUN npm install
```

```
COPY ..
```

```
ENV PORT=8800
```

```
ENV JWT="8hEnPGeoBqGUT6zksxt4G95gW+uMdzwe7EVaRnp0xRI="
```

```
ENV MONGO_URL="mongouri"
```

```
EXPOSE 8800
```

```
CMD [ "npm", "start" ]
```

Create Dockerfile for React.js Frontend:

Create a file named Dockerfile in your React.js project directory with the following content:

```
FROM node:14-slim
```

```
WORKDIR /app
```

```
COPY ./package.json ./  
COPY ./package-lock.json ./
```

```
RUN npm install
```

```
COPY ..
```

```
EXPOSE 3000
```

```
CMD [ "npm", "start" ]
```

Step 3: Create Docker Compose Configuration

Create a docker-compose.yml file in the root directory of your project:

```
version: "3.8"  
services:  
  mongodb:  
    image: "mongo"  
    volumes:  
      - data:/data/db  
  server:  
    build: ./server  
    ports:  
      - "8800:8800"  
    volumes:  
      - logs:/app/logs  
      - ./server:/app  
      - /app/node_modules  
    depends_on:  
      - mongodb  
  client:  
    build: ./client  
    ports:  
      - "3000:3000"  
    volumes:  
      - ./client/src:/app/src  
    stdin_open: true  
    tty: true  
    depends_on:  
      - server
```

volumes:

data:

logs:

Step 4: Build and Run Docker Containers

Open a terminal/command prompt in the project directory.

Run the following command to build the Docker images and start the containers:

docker-compose up -build

Once the containers are running, you can access your application at <http://localhost> in your web browser.

Example of a response from the Server:

Screenshots of the project in sequence

```
D:\SEME4PROJ\Urban-Garden> docker-compose up
[+] Running 2/3
  ✓ Network urban-garden_default     Created
  ✓ Container urban-garden-mongodb-1 Created
  - Container urban-garden-server-1   Creating

```

```
Starting the development server...
client-1 |
```

```
Dockerfile docker-compose.yml
client-1 |
client-1 |
```

```
DIRECTORY URBAN-GARDEN
> .git
> client
> node_modules
> public
> src
  .dockerignore
  .gitignore
  Dockerfile
  package-lock.json
  package.json
> server
```

```
docker-compose.yml
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
```

```
client-1 • Line 5: 'StandardImageList' is defined but never used no-unused-vars
client-1 • Line 11:7: 'loading' is assigned a value but never used no-unused-vars
client-1 • Line 29:16: 'setData' is assigned a value but never used no-unused-vars
client-1 • src/Products/SingleProductPage/SingleProductPage.jsx
client-1 • Line 22:6: React Hook useEffect has a missing dependency: 'Fetching'. Either include it or remove the dependency
array react-hooks/exhaustive-deps
client-1 • src/Products/SingleProductPage/SingleProductPage.jsx
client-1 • Line 11:17: 'loading' is assigned a value but never used no-unused-vars
client-1 • Line 11:26: 'error' is assigned a value but never used no-unused-vars
client-1 • src/UserProfile/UserProfile.jsx
client-1 • Line 26:6: React Hook useEffect has a missing dependency: 'getWindowDimensions'. Either include it or remove the dependency
array react-hooks/exhaustive-deps
```

```

PS D:\SEMPROJ\Urban-Garden> docker-compose up
[+] Running 3/0
  ✓ Container urban-garden-mongodb-1  Running
  ✓ Container urban-garden-server-1   Running
  ✓ Container urban-garden-client-1  Running
Attaching to client-1, mongodb-1, server-1

```

Containers

Search for images, containers, volumes... Ctrl+K

Container CPU usage: 0.67% / 800% (8 cores available)

Container memory usage: 901.83MB / 3.63GB

Show charts ▾

Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
urban-garden-mongodb-1	mongodb:3.6.10	Running (3/3)	0.67%	39 minutes ago	Logs Stop Delete	
urban-garden-server-1	urban-garden-server:7c7a3b49	Running	0.65%	39 minutes ago	Logs Stop Delete	
urban-garden-client-1	urban-garden-client:7c7a3b49	Running	0%	39 minutes ago	Logs Stop Delete	

Showing 4 items

urban-garden.vercel.app

Home How To Use Why UG Log In Sign Up

Best Choice

Planting Success, One Click Away.

Discover the ultimate convenience of setting up your online nursery store. Empower your customers with a secure navigation, detailed plant information and effortless purchases. Maximize sales and satisfaction in the growing market of plants and gardening supplies!

[Shop Now](#) [Learn More](#)

urbangarden.vercel.app/home

TRENDING

- Rose**
Black Rose all the way from Himalayas
★★★★★
₹200
[Add To Cart](#)
- Pilea Peperomioides**
Also known as the Chinese Money Plant, P...
★★★★★
₹30
[Add To Cart](#)
- Purple Bulbs**
Purple fresh Bulbs all the way from the ...
★★★★★
₹750
[Add To Cart](#)
- Ceramic pots**
These pots are made from glazed or ungl...
★★★★★
₹113
[Add To Cart](#)

urbangarden.vercel.app/Products/644f62c34fb23c47c673e6fc

Urban Garden

Categories: What are you looking for?

Gardening ▾ Plants ▾ Seeds ▾ Bulbs ▾ Pots ▾ Planters ▾ Soil & Fertilizer ▾ Pebbles ▾ Accessories ▾

Snake Plant
posted by 644f62c34fb23c47c673e6fc
★★★★★
₹559
Also known as mother-in-law's tongue, snake plants are incredibly tough and can survive even in low light conditions.

✓ Season: Winter
❖ Sunlight Requirement: indirect light
🕒 Water Frequency: 25 x 50 cm

[ADD TO MY LIST](#)

Urban Garden [Category/Gardening](#)

CATEGORIES

- Gardening
- Plants
- Seeds
- Bulbs
- Pots
- Soil & Fertilizer
- Pebbles
- Accessories

PRICE

The selected range is ₹ 0 - ₹ 500

CITY

City name:

<https://urbangarden.veetelapp.com/category/Products/644947182987ba7c8344>

GARDENING »

Jade Plant
These plants are native to South Africa and...
₹530 [Add To Cart](#)

Snake Plant
Also known as mother-in-law's tongue, sn...
₹559 [Add To Cart](#)

Snake Plant
Also known as mother-in-law's tongue, sn...
₹559 [Add To Cart](#)

Urban Garden [Category](#) What are you looking for?

ACCOUNT
1G_P9R

OVERVIEW

Your Orders

Order Placed	Total	Ship To	Mode Of Payment
2023-05-07T18:31:57.730Z	₹299	Shri. 0550	Cash On Delivery

Delivered 2023-05-07T18:31:57.730Z

Crushed granite
This type of pebble is made by crushing granite i...
Price: ₹99
Quantity: 3
Total Price: ₹297

Order Placed	Total	Ship To	Mode Of Payment
2024-07-22T06:12:58.371Z	₹2748	Shri. 0550	Cash On Delivery

Urban Garden [Category](#) What are you looking for?

Gardening **Plants** **Seeds** **Bulbs** **Pots** **Planters** **Soil & Fertilizer** **Pebbles** **Accessories**

Price Details

Total MRP:	₹ 559
Discount on MRP:	₹ 0.00
Coupon Discount:	₹ 0.00
Platform Fee:	₹ 20
Shipping Fee:	FREE
Total:	₹ 579

[PLACE ORDER](#)

Snake Plant
by 644947182987ba7c8344

1 ₹ 559/-

Conclusion

Blooming Delights emerges as a comprehensive online nursery store, facilitating a seamless shopping experience for gardening enthusiasts. With an extensive range of seeds, plants, tools, and related products, customers are empowered to explore, purchase, and engage with their gardening essentials. The platform not only enables easy comparison of prices but also encourages interaction through product ratings and comments, fostering a sense of community among users.

Moreover, the availability of detailed product descriptions aids in informed decision-making, while the provision of customization ideas and planting tips enhances the overall gardening experience. As we look to the future, our focus remains on continual improvement, with plans to expand product categories, introduce diverse payment modes for added convenience, and implement efficient order-related communication channels such as email or SMS updates.

By prioritizing customer satisfaction and innovation, Blooming Delights is poised to evolve into a premier destination for all gardening needs, catering to the aspirations and passions of green-thumbed enthusiasts worldwide.