

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

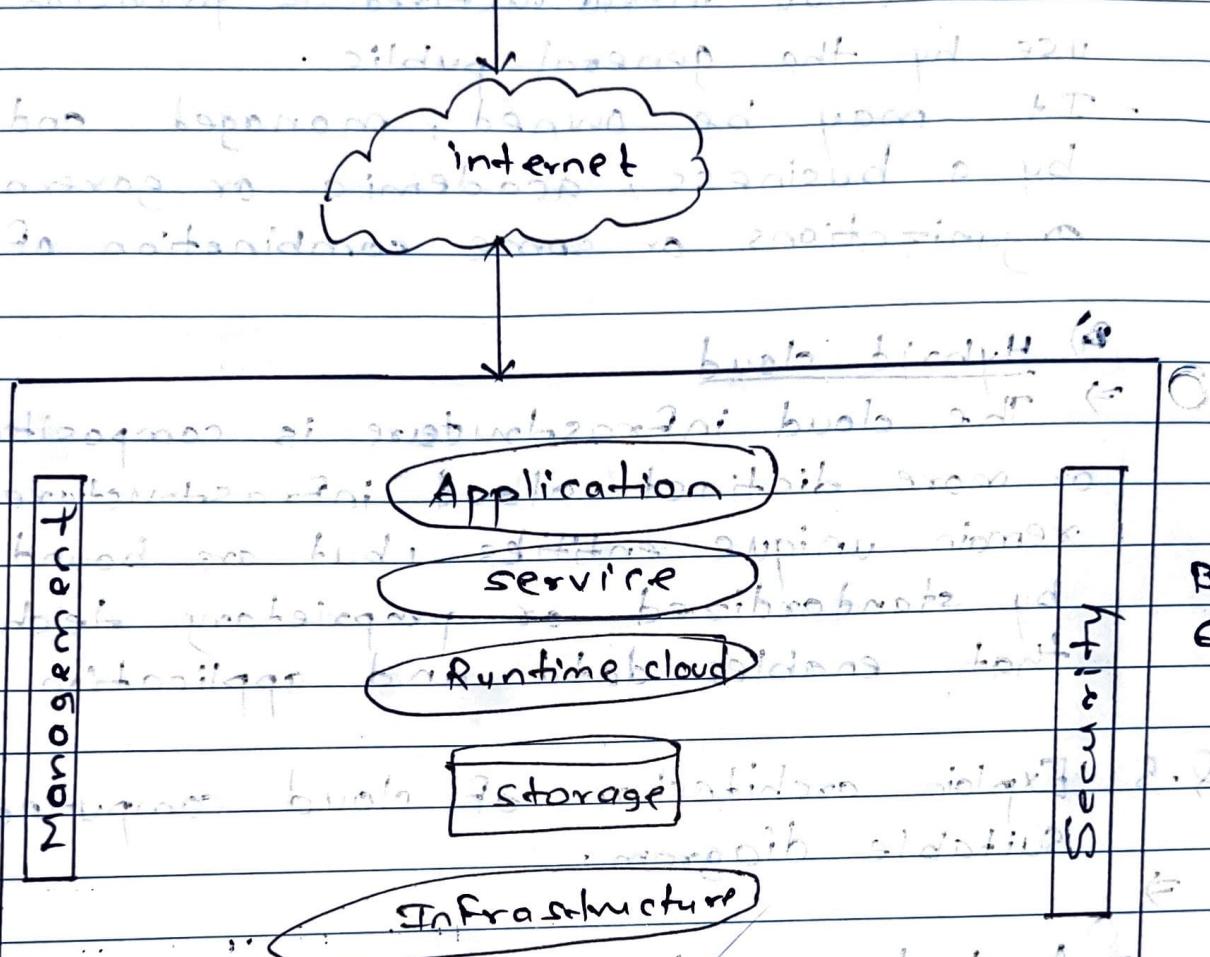
→ The cloud infrastructure is composition of two or more distinct cloud infrastructures that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability.

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

Client Infrastructure



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 maintainance costs & maintenance costs are low.

- 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

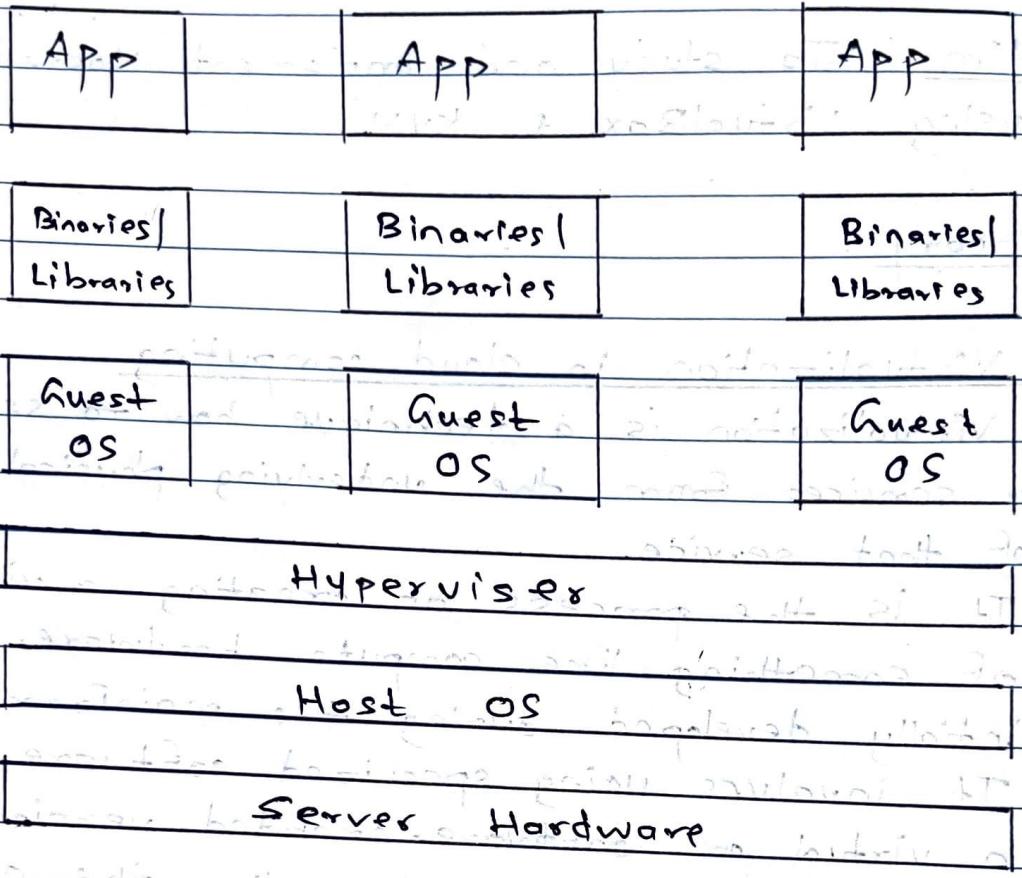
- Disadvantages of cloud computing:
- 1) Dependence on internet connectivity.
 - 2) Limited control and customization of the technology.
 - 3) Security concerns include the risk of data breaches and loss of sensitive information.
 - 4) Possibility of service disruptions and downtime.
 - 5) Limited compliance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

VirtualBox
Welcome to VirtualBox.org!

VirtualBox is a powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2. See "About VirtualBox" for an introduction.

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

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

Download VirtualBox 6.1

Hot picks:

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

ORACLE
Contact - Privacy policy - Terms of Use

News Flash

- We're hiring!** Looking for a new challenge? We're hiring a System Administrator/Quality Engineer (Germany).
- Important May 17th, 2021** We're hiring! Looking for a new challenge? We're hiring a VirtualBox senior developer in Berlin (Germany).
- New January 18th, 2022** VirtualBox 6.1.32 released! Oracle today released a 6.1 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- New November 22nd, 2021** VirtualBox 6.1.30 released! Oracle today released a 6.1 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- New October 19th, 2021** VirtualBox 6.1.28 released! Oracle today released a 6.1 maintenance release which improves stability and fixes regressions. See the Changelog for details.

[More information...](#)

2. Create a new virtual machine.

Oracle VM VirtualBox Manager

File Machine Help

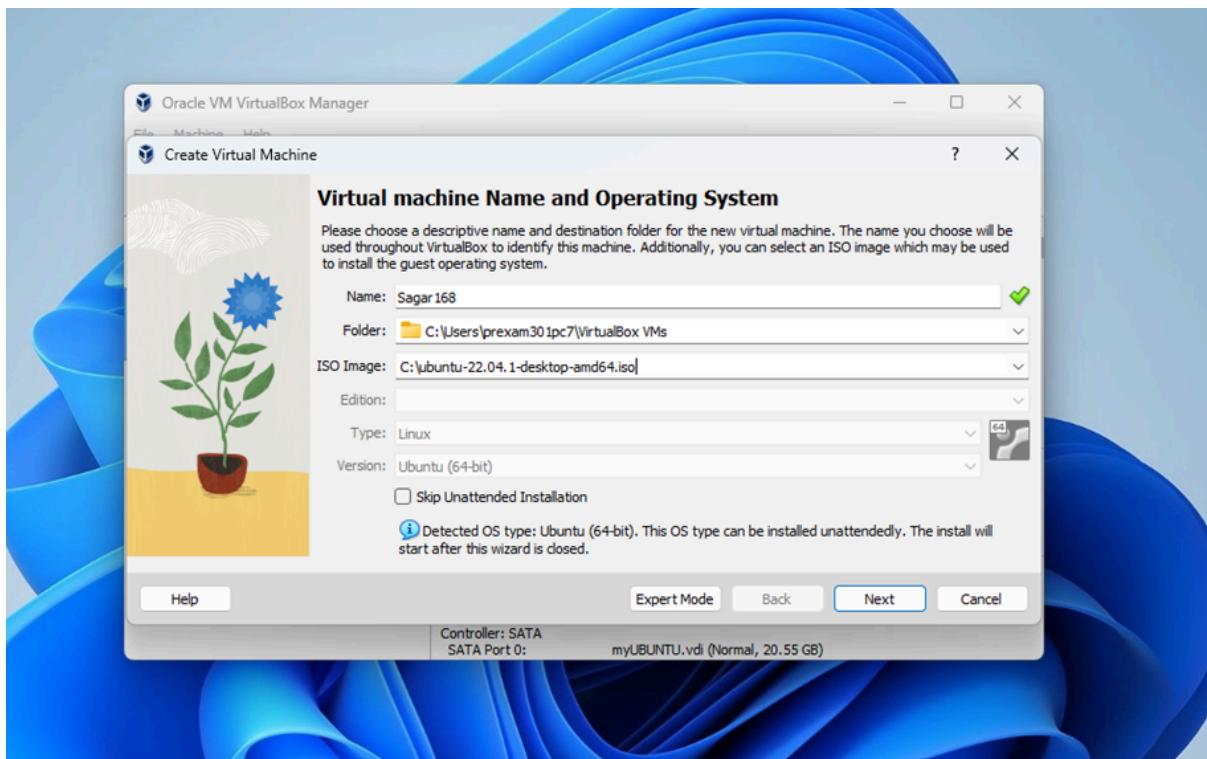
Tools Preferences Import Export New (Ctrl+N)

Welcome to VirtualBox!

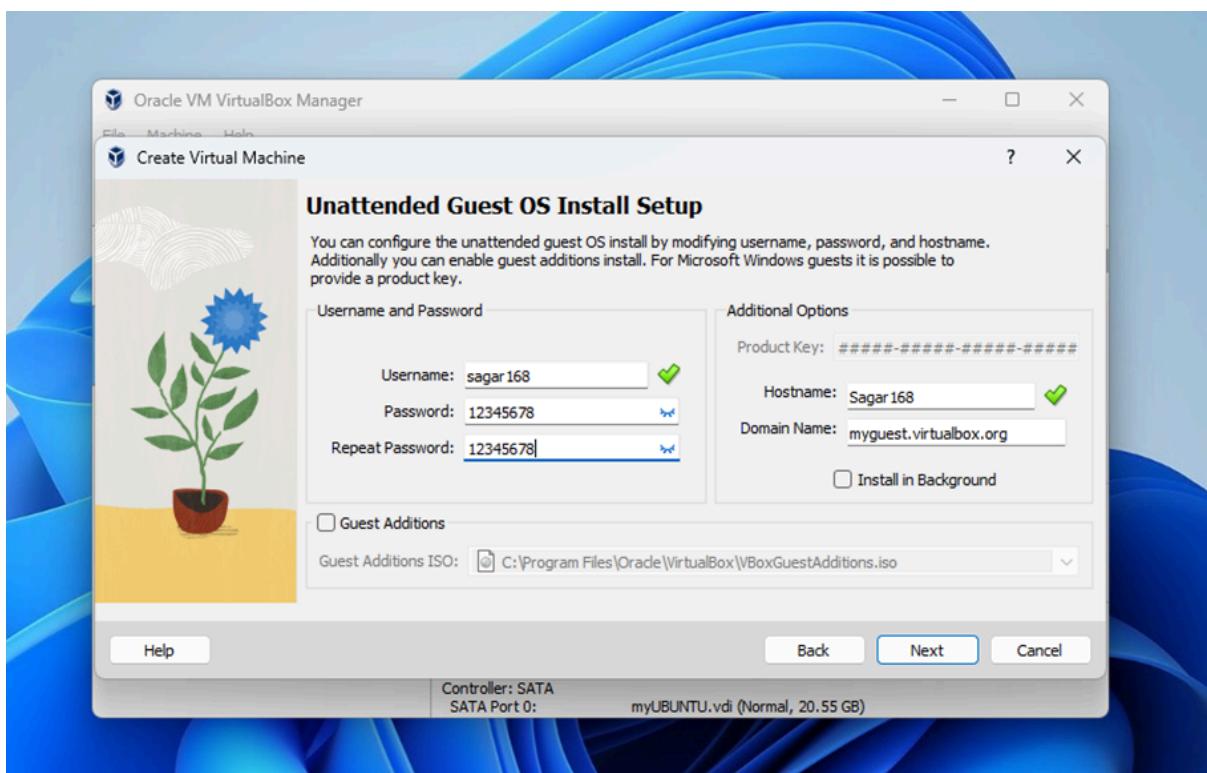
The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

You can press the F1 key to get instant help, or visit www.virtualbox.org for more information and latest news.

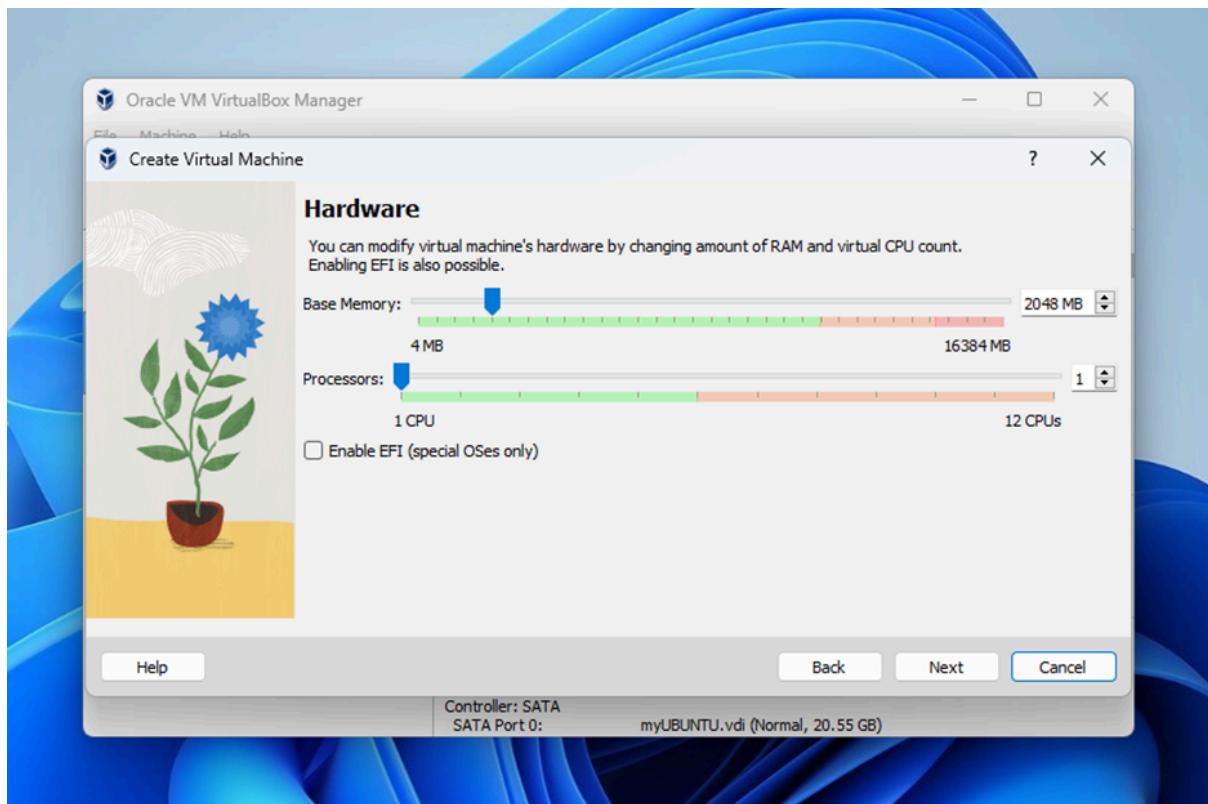
3. Choose the ISO file of the OS



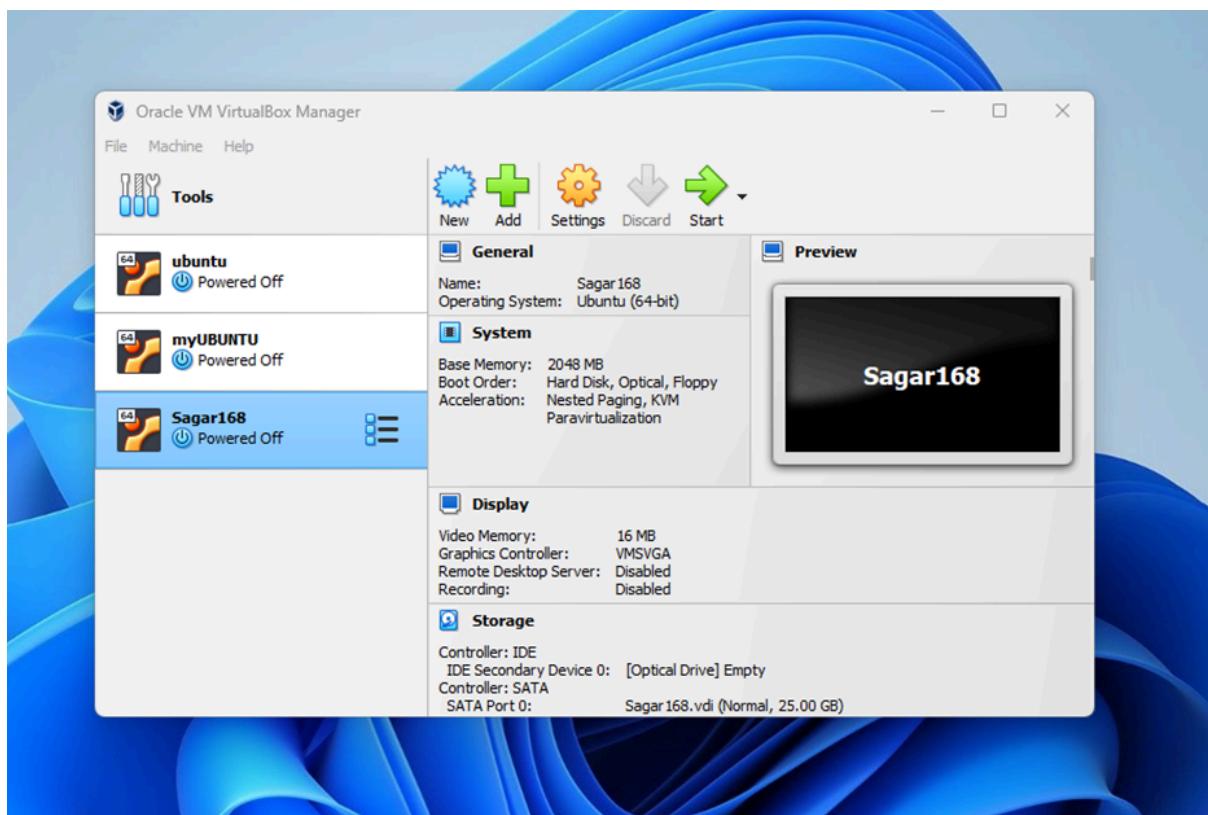
4. Complete the Setup



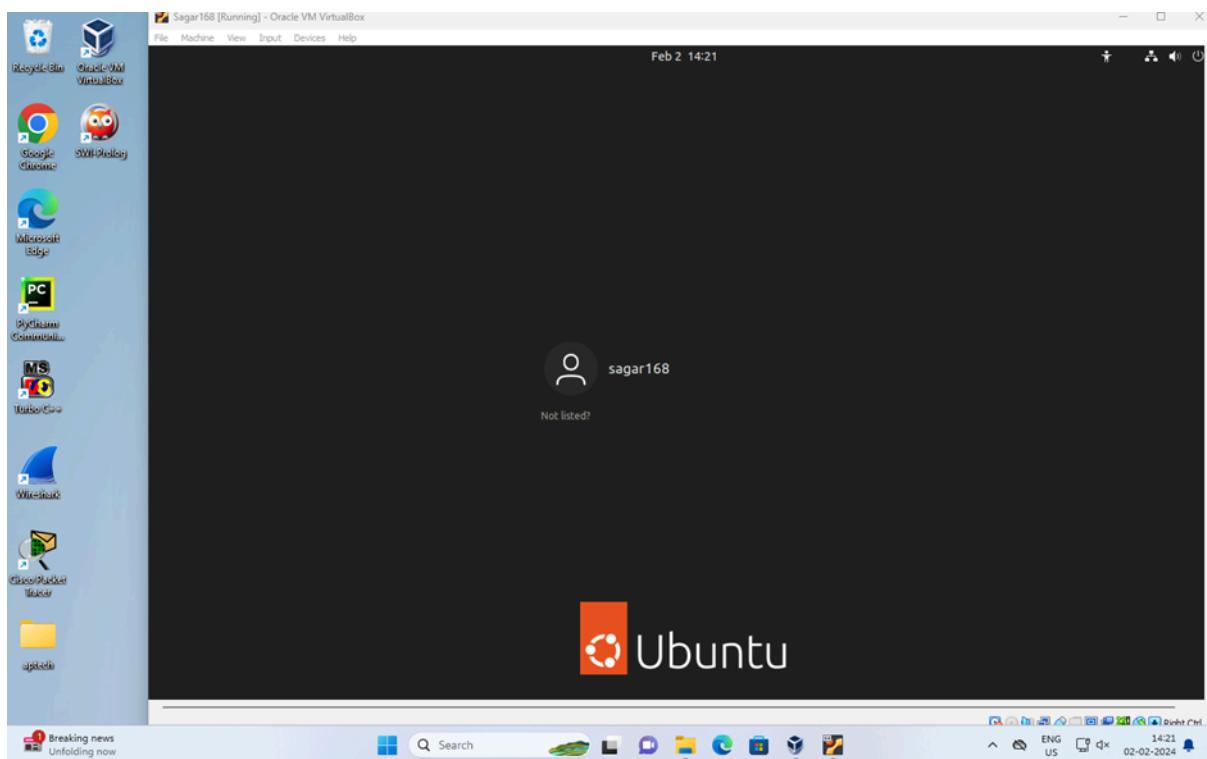
5. Select Ram and Virtual hard disk



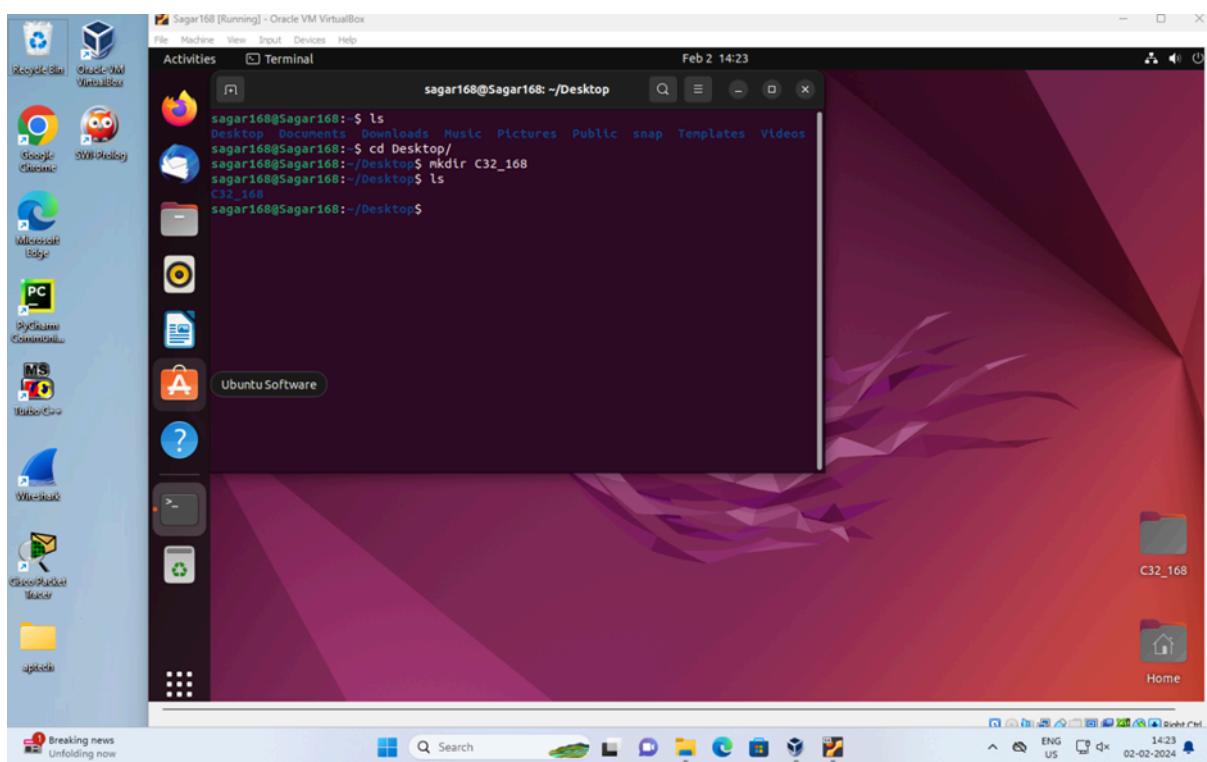
6. Click on Start and it will run Ubuntu OS



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
<ul style="list-style-type: none"> 1) Operates on the conventional operating system. 2) OS virtualization. 3) Functions as the application on the host. 4) It is less scalable than the bare-metal hypervisor. 5) Easier than a type 1 setup because of the existing OS. 6) Speed is slower than bare-metal. 7) For e.g., VMWare, EXSi, Microsoft Hyper V. 8) For e.g., VMWare Workstation Player, Microsoft Virtual PC, Sun's VirtualBox. 	<ul style="list-style-type: none"> 1) Runs directly on the system where VMs function. 2) Hardware virtualization. 3) Guest OS and applications runs on the hypervisor. 4) It is more scalable than the hosted hypervisor. 5) Simpler, if your hardware supports the application. 6) Speed is faster than hosted hypervisors. 7) For e.g., VMWare ESXi, Microsoft Hyper V, Citrix XenServer.

- 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

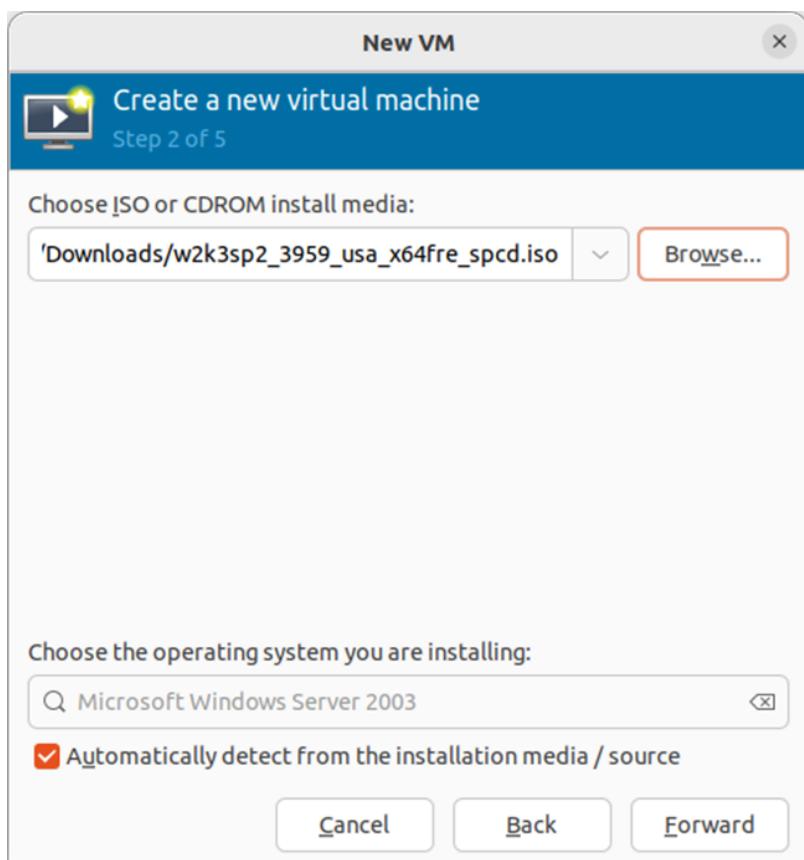
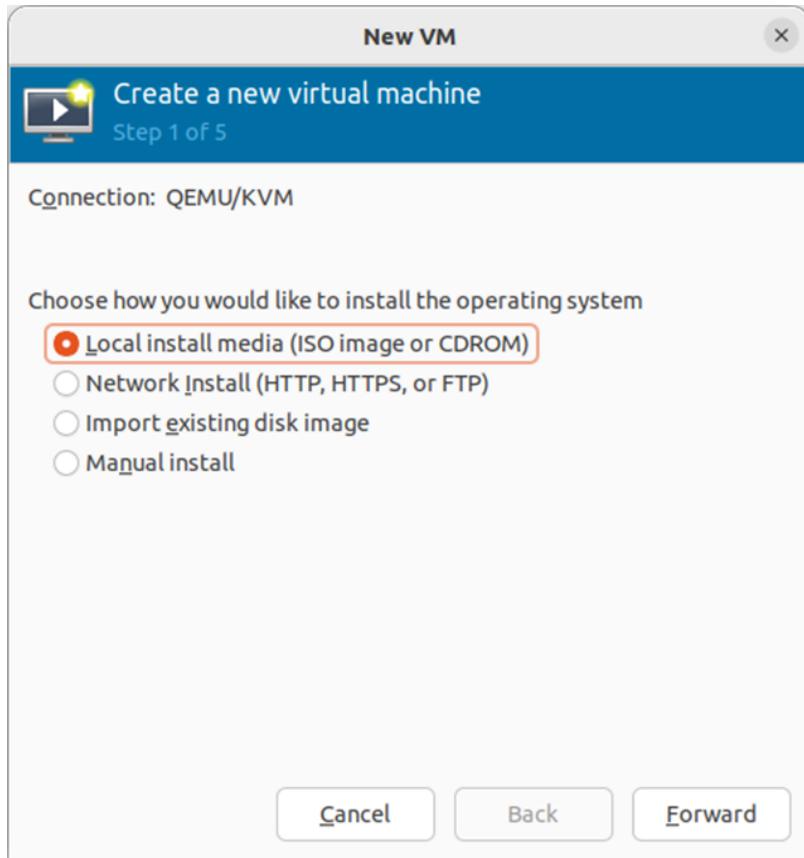
- Automatically adjusting the number of compute resources based on the demand, ensuring optimal performance and cost efficiency.
- For e.g., scaling up instances during peak hours and scaling down during low traffic.

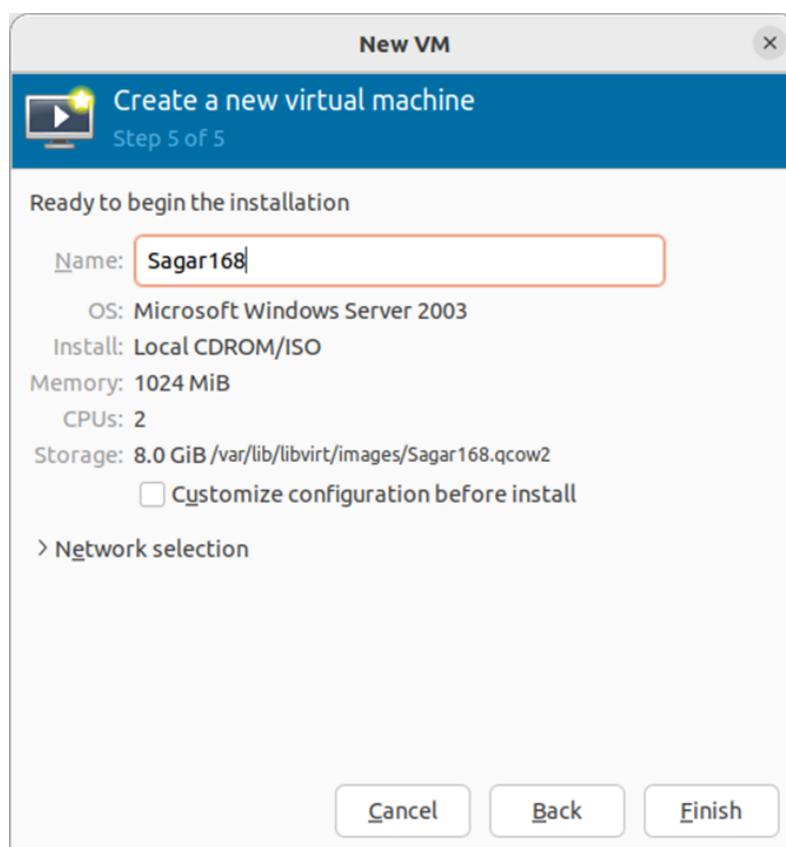
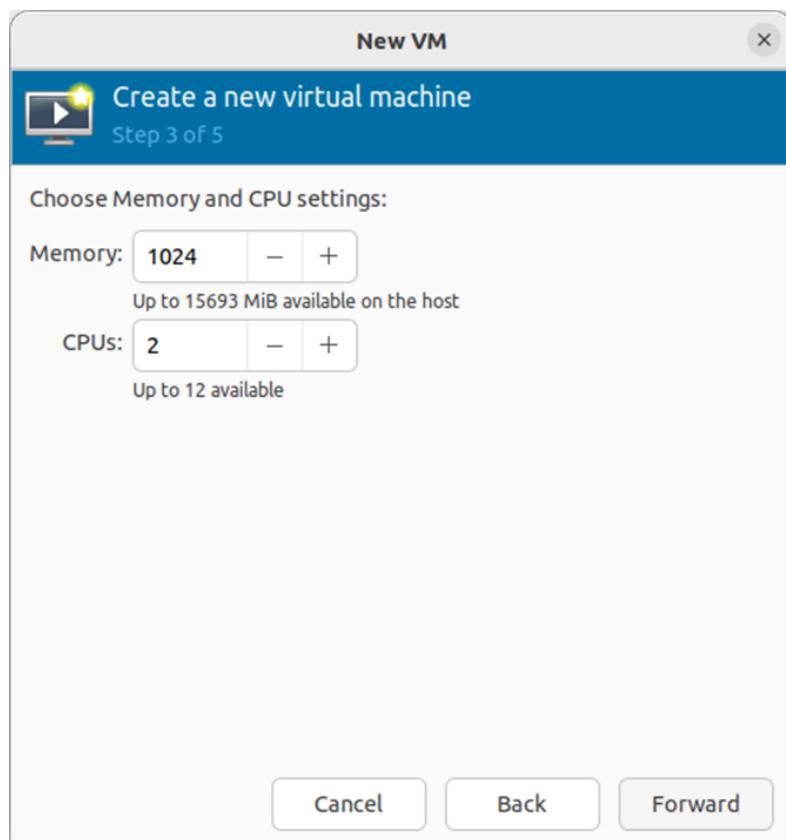
- Load Balancing

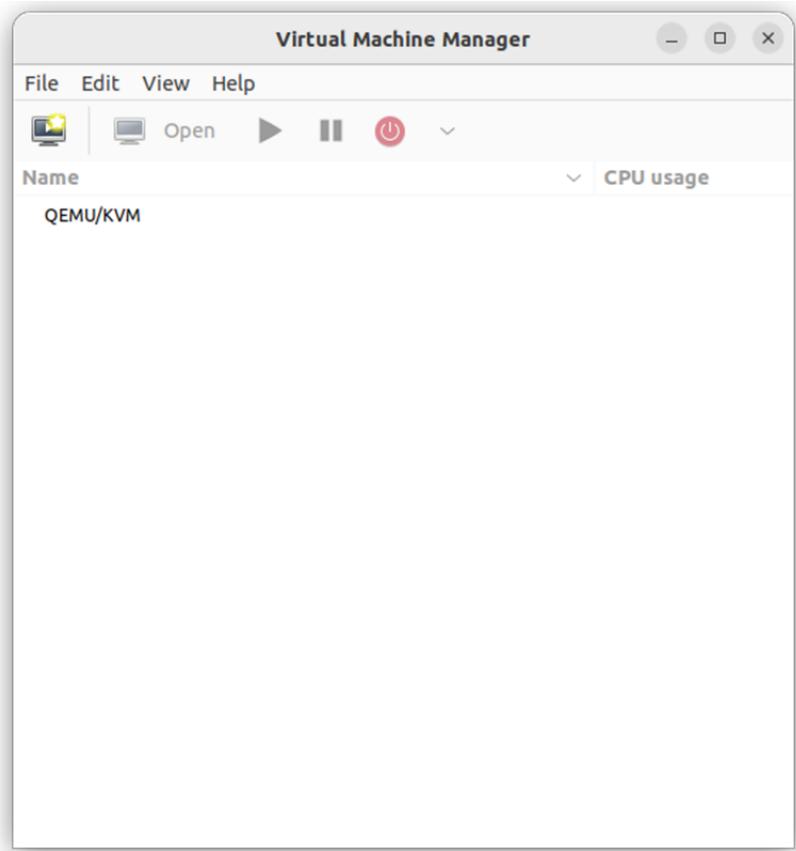
- Distributing incoming network traffic across multiple servers to ensure no single server is overwhelmed, optimizing resource utilization and improve reliability.
- For e.g., Distributing web traffic among multiple servers to prevent overloading any single server.

(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.

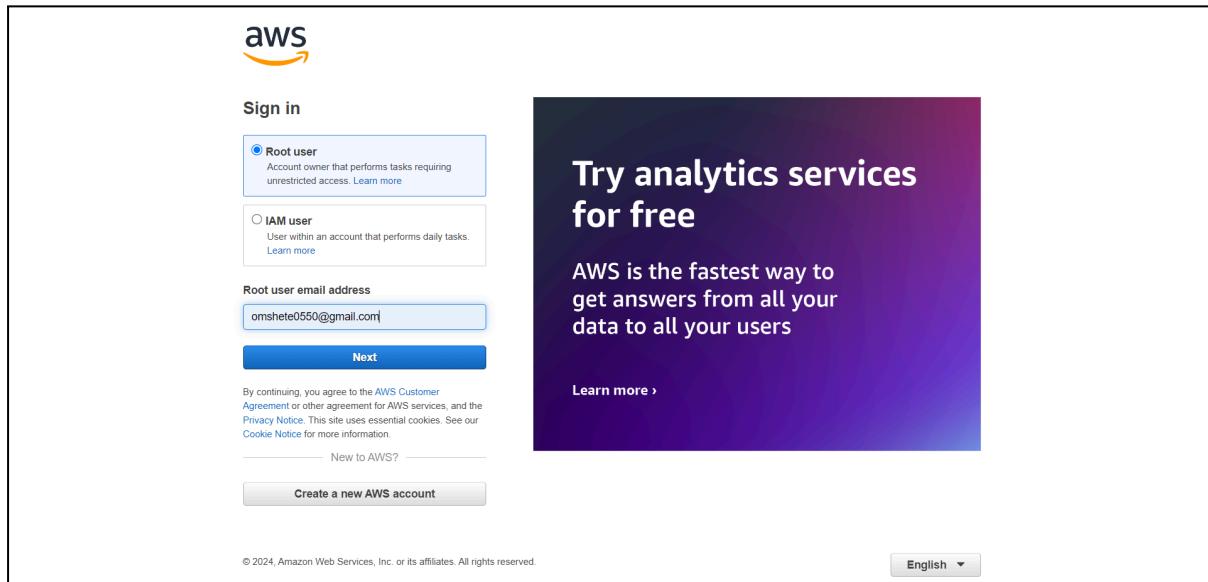
5) Remote desktop Protocol (RDP)

⇒ 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 dropdown for 'Stockholm' and '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 Fully managed benefits of Amazon...), 'Explore AWS' (with a link to AWS Support), and 'Security' (with a link to AWS Amplify Console). 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 with the following details:

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

Instances (running)	0
Auto Scaling Groups	0
Dedicated Hosts	0
Elastic IPs	0
Instances	0
Key pairs	0
Load balancers	0
Placement groups	0
Security groups	1
Snapshots	0
Volumes	0
- Launch instance:** To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.
 - Launch instance** button (orange)
 - Migrate a server** button
 - Note: Your instances will launch in the Europe (Stockholm) Region
- Service health:** AWS Health Dashboard
 - Region: Europe (Stockholm)
 - Status: This service is operating normally.
- Account attributes:**
 - Default VPC: vpc-02ed8cfea6738bce
 - Settings
 - Data protection and security
 - Zones

4. Select the Instance Specifications

The screenshot shows the "Launch an instance" wizard with the following steps completed:

- Step 1: Name and tags**
 - Name: My Server
 - Add additional tags
- Step 2: Application and OS Images (Amazon Machine Image)**
 - An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.
 - Search bar: Search our full catalog including 1000s of application and OS images
 - Quick Start: Buttons for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and a search icon.
 - Browse more AMIs: Including AMIs from Microsoft, Red Hat, and SUSE.
- Step 3: Summary**
 - Number of instances: 1
 - Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...read more
ami-0914547665e6a707c
 - Virtual server type (instance type): t3.micro
 - Firewall (security group): New security group
 - Storage (volumes): 1 volume(s) - 8 GiB
 - Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month.
 - Launch instance button (orange)

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
 ami-0914547665e6a707c (64-bit (x86)) / ami-0b559d60d5ba5053 (64-bit (Arm))
 Virtualization: hvm ENA enabled: true Root device type: ebs

Description
 Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2024-03-01

Architecture 64-bit (x86) **AMI ID** ami-0914547665e6a707c **Verified provider**

Instance type **t3.micro** **Family** t3 **2 vCPU** **1 GiB Memory** **Current generation: true** **Free tier eligible**

Configure storage 1x 8 GiB gp2 **Root volume (Not encrypted)**

Advanced details **Domain join directory** Select **Create new directory** **IAM instance profile** Select **Create new IAM profile**

Hostname type IP name **DNS Hostname** Enable IP name IPv4 (A record) DNS requests Enable resource-based IPv4 (A record) DNS requests Enable resource-based IPv6 (AAAA record) DNS requests

Instance auto-recovery Select **Shutdown behavior** Stop **Stop + Hibernate behavior** Info

Summary

Number of instances **Info** 1

Software Image (AMI) Canonical, Ubuntu, 22.04 LTS, ...read more
 ami-0914547665e6a707c

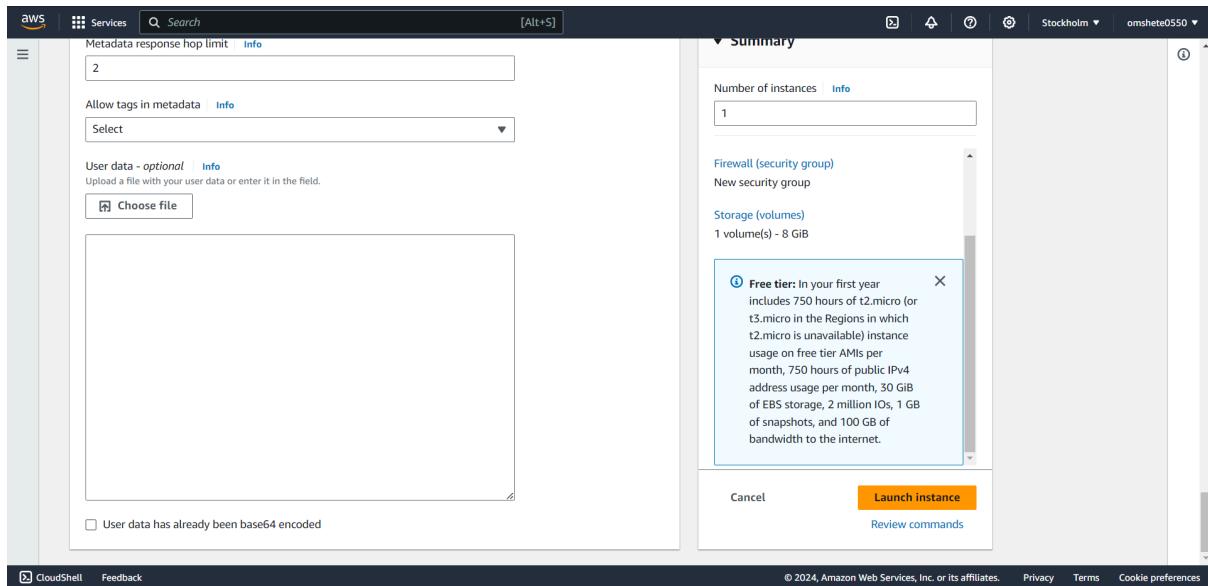
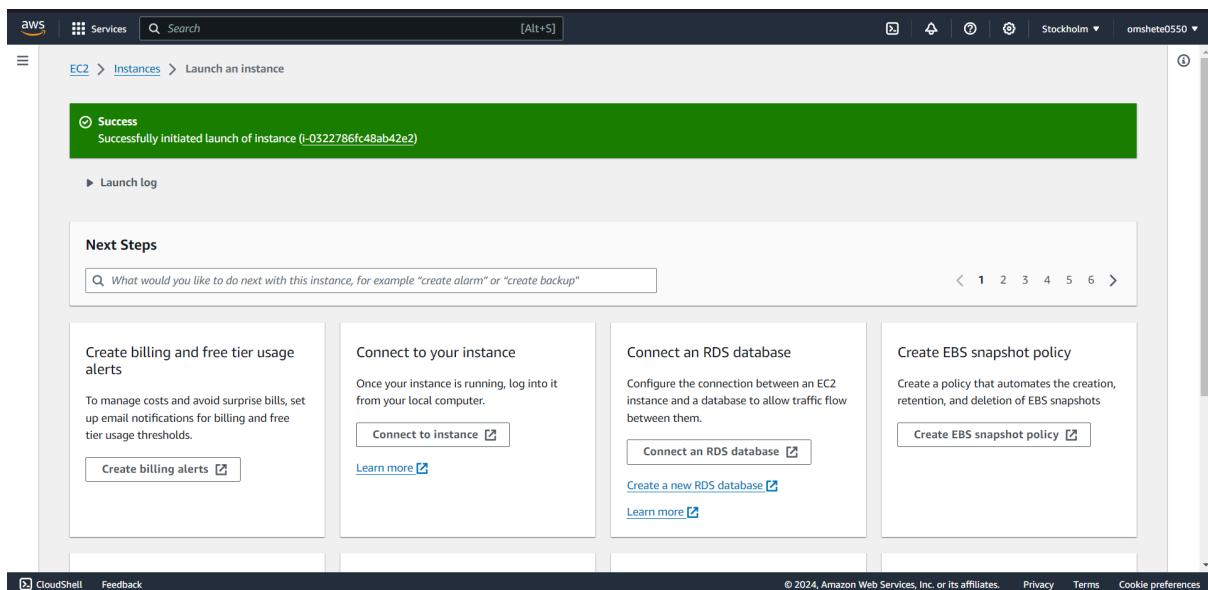
Virtual server type (instance type) t3.micro

Firewall (security group) New security group

Storage (volumes) 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Launch instance

5. Click on Launch Instance on the right**6. Instance Created Successfully**

7. On EC2 Dashboard Click Instances(running)

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 'Resources' section displays various Amazon EC2 resources: 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, and Volumes 1. Below this, there's a 'Launch instance' button and a note stating "Note: Your instances will launch in the Europe (Stockholm) Region". To the right, the 'Service health' section shows "This service is operating normally". The bottom right corner contains account attributes and a Default VPC entry.

8. Click on Connect

The screenshot shows the AWS EC2 Instances page. The left sidebar lists various EC2-related services. The main table shows one instance: "My Server" (i-0322786fc48ab42e2), which is "Running". The "Actions" column for this instance has a "Connect" button highlighted. The "Details" tab is selected for the instance, displaying its summary information, including its public IPv4 address (16.171.150.6), private IP address (172.31.43.39), and public DNS name (ec2-16-171-150-6.eu-north-1.compute.amazonaws.com). Other tabs include Status and alarms, Monitoring, Security, Networking, Storage, and Tags.

Connect to instance [Info](#)

Connect to your instance i-0322786fc48ab42e2 (My Server) using any of these options

EC2 Instance Connect Session Manager SSH client EC2 serial console

Instance ID
i-0322786fc48ab42e2 (My Server)

Connection Type
 Connect using EC2 Instance Connect
 Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

Connect using EC2 Instance Connect Endpoint
 Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
16.171.150.6

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.
ubuntu

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel Connect

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

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

```

Memory usage: 22%           IPv4 address for ens5: 172.31.43.39
Swap usage:  0%
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-43-39:~$ mkdir om
ubuntu@ip-172-31-43-39:~$ ls
om

i-0322786fc48ab42e2 (My Server)
PublicIPs: 16.171.150.6  PrivateIPs: 172.31.43.39
  
```

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

10. Now Go to EC2 Dashboard and Select Instance State

The screenshot shows the AWS EC2 Instances page. At the top, there is a search bar and a 'Launch instances' button. Below the search bar, there are filters for 'Instance state' (set to 'running'), 'Instance type' (set to 't3.micro'), and 'Status' (set to 'ms'). A table lists one instance: 'My Server' (Instance ID: i-0322786fc48ab42e2), which is 'Running'. To the right of the table, there are buttons for 'Stop instance', 'Start instance', 'Reboot instance', 'Hibernate instance', and 'Terminate instance'. The 'Terminate instance' button is highlighted with a yellow border. At the bottom of the page, there is a detailed view for the selected instance, showing its public and private IP addresses, instance state, and DNS names.

11. Select Terminate Instance

The screenshot shows the AWS EC2 Instances page with a 'Terminate instance?' dialog box overlaid. The dialog contains a warning message: 'On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.' Below this, it asks 'Are you sure you want to terminate these instances?'. A table shows the instance to be terminated: 'i-0322786fc48ab42e2 (My Server)' with 'Termination protection' set to 'Disabled'. At the bottom of the dialog, there is a 'Cancel' button and a prominent orange 'Terminate' button.

Om Shete - C3 - 2103163

The screenshot shows the AWS EC2 Instances page. A green banner at the top indicates "Successfully terminated i-0322786fc48ab42e2". The main table lists one instance: "My Server" (i-0322786fc48ab42e2), which is currently "Shutting-down". The instance type is t3.micro, status check is 2/2 checks passed, and it is in the eu-north-1b availability zone. The public IP is 16.171.150.6. The instance summary details show the instance ID, IPv4 address (16.171.150.6), instance state (Shutting-down), and private IP DNS name (ip-172-31-43-39.eu-north-1.compute.internal). The sidebar on the left includes sections for Instances, Images, and Elastic Block Store.

The screenshot shows the AWS EC2 Dashboard. The "Resources" section displays the following counts: 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, and Volumes 1. The "Launch instance" section contains a "Launch instance" button and a note that instances will launch in the Europe (Stockholm) Region. The "Service health" section shows the AWS Health Dashboard and indicates that the service is operating normally. The "Account attributes" section shows the Default VPC (vpc-02ed8cf8aa6738bce) and other account settings. The sidebar on the left includes sections for Instances, Images, and Elastic Block Store.

Experiment No: 5

Aim: To study and implement 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, 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: AWS Lambda provides built-in monitoring and logging capabilities, allowing developers to monitor application health and troubleshoot issues easily.

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

(283) ~~languages | Frameworks supported by EBS~~

~~languages supported by EBS~~

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 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> |
|---|--|

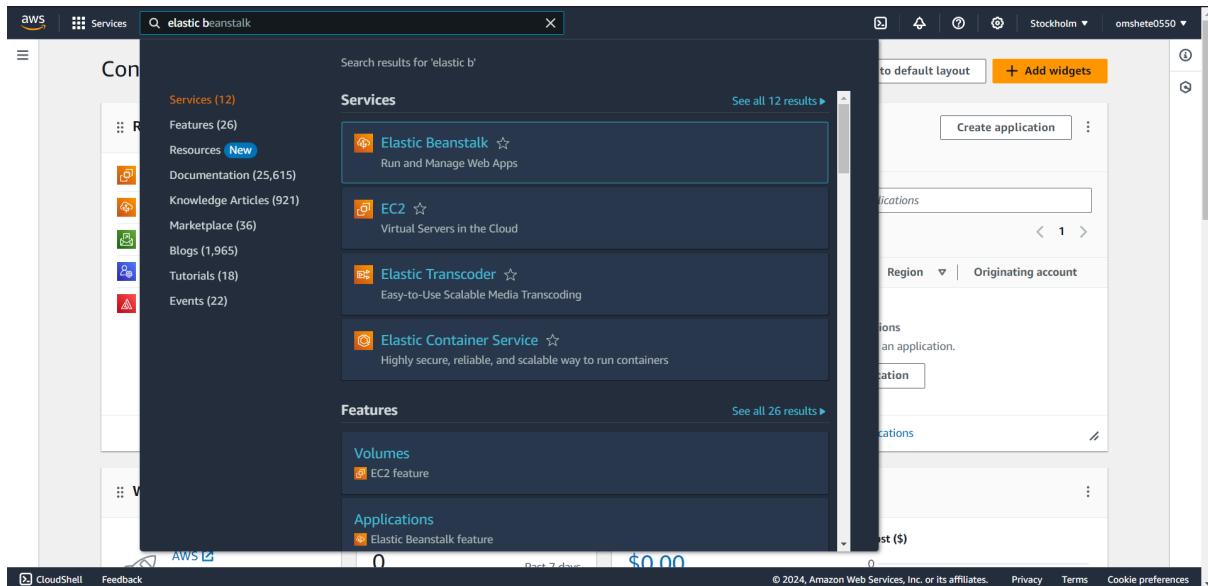
283 hrs 829 minutes (approx)

(A)

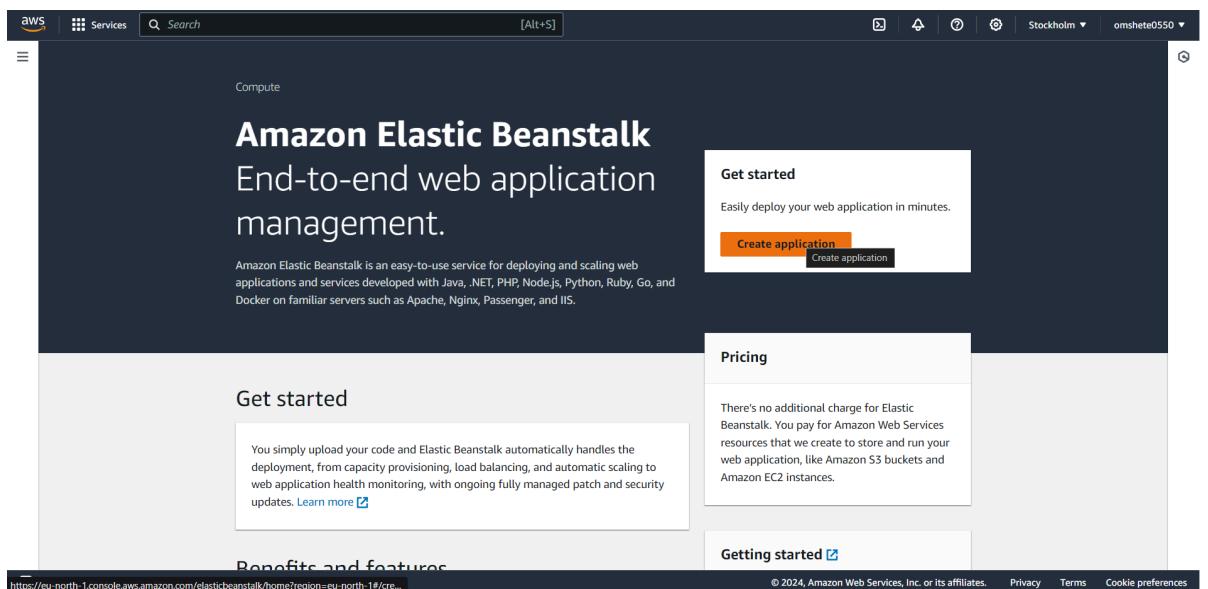
SP
IS/32

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.

Environment tier

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

- Web server environment Run a website, web application, or web API that serves HTTP requests. [Learn more](#)
- Worker environment Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information

Application name: MyApp

Application tags (optional)

Up to 50 tags. You can use tags to group and filter resources and is case-sensitive. [Learn more](#)

Key	Value
Name	My Server
Add new tag	

You can add 49 more tags.

Platform

Platform type

- Managed platform Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)
- Custom platform Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform: Node.js

Platform branch: Node.js 20 running on 64bit Amazon Linux 2023

Platform version: 6.1.2 (Recommended)

Application code

- Sample application
- Existing version Application versions that you have uploaded.
- Upload your code Upload a source bundle from your computer or copy one from Amazon S3.

Version: 4.2.2 (Recommended)

Presets

Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.

Configuration presets

- Single instance (free tier eligible)
- Single instance (using spot instance)
- High availability
- High availability (using spot and on-demand instances)
- Custom configuration

Next

Configure service access

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

EC2 key pair

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

mykeypair

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

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 top navigation bar includes the AWS logo, Services, a search bar, and account information for Stockholm (omshete0550). The main title is "Elastic Beanstalk" with a progress bar indicating "Elastic Beanstalk is launching your environment. This will take a few minutes." Below the title, the breadcrumb path is "Elastic Beanstalk > Environments > MyApp-env". The left sidebar shows "Application: MyApp" with options like Application versions, Saved configurations, Environment: MyApp-env (selected), Go to environment, Configuration, Events, Health, Logs, Monitoring, Alarms, Managed updates, and Tags. The main content area displays the "Environment overview" and "Platform" sections. The "Environment overview" table includes rows for Health (Unknown), Domain (-), Environment ID (e-i6mmy8dpc2), and Application name (MyApp). The "Platform" section shows Node.js 20 running on 64bit Amazon Linux 2023/6.1.2. Below these are tabs for Events, Health, Logs, Monitoring, Alarms, Managed updates, and Tags. The "Events" tab shows 2 items with a search bar and pagination controls. At the bottom, there are links for CloudShell, Feedback, and copyright information (© 2024, Amazon Web Services, Inc. or its affiliates.).

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

The screenshot shows the AWS Elastic Beanstalk console. The top navigation bar includes the AWS logo, Services, a search bar, and account information for Stockholm (omshete0550). The main title is "Elastic Beanstalk" with a progress bar indicating "Elastic Beanstalk is launching your environment. This will take a few minutes." Below the title, the breadcrumb path is "Elastic Beanstalk > Environments > MyApp-env-1". The left sidebar shows "Application: MyApp" with options like Application versions, Saved configurations, Environment: MyApp-env-1 (selected), Go to environment, Configuration, Events, Health, Logs, Monitoring, Alarms, Managed updates, and Tags. The main content area displays the "Environment overview" and "Platform" sections. The "Environment overview" table includes rows for Health (ok), Domain (-), Environment ID (e-m48pfjfkj), and Application name (MyApp). The "Platform" section shows Node.js 20 running on 64bit Amazon Linux 2023/6.1.2. Below these are tabs for Events, Health, Logs, Monitoring, Alarms, Managed updates, and Tags. The "Events" tab shows 6 items with a search bar and pagination controls. The first event is an INFO message from April 8, 2024, at 11:52:02 (UTC+5:30) about deleting an SNS topic. The second event is an ERROR message from April 8, 2024, at 11:52:01 (UTC+5:30) about failing to launch the environment. At the bottom, there are links for CloudShell, Feedback, and copyright information (© 2024, Amazon Web Services, Inc. or its affiliates.).

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: To begin with simplicity & user interface to show in separate build.

- 1) Cost efficient. Less cost in separate.
- 2) Scalability. National ability in O.
- 3) Accessibility. can access soft →
- 4) Automatic update in auto delivery.
- 5) Redundancy. Data has no concern.
- 6) Collaborator.
- 7) flexibility.
- 8) Managed security. less risk.

- Limitations: reverse engineering by others.

- 1) Dependency on Internet & bandwidth.
- 2) Security concerns about viruses.
- 3) Data transfer speed may not be.
- 4) Service downtime due to less reliable.
- 5) Regulatory compliances.
- 6) Customization constraints.
- 7) Limited control over app.
- 8) Data location.

→ ~~not available~~

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

1) Object storage:



Transport : TCP / IP

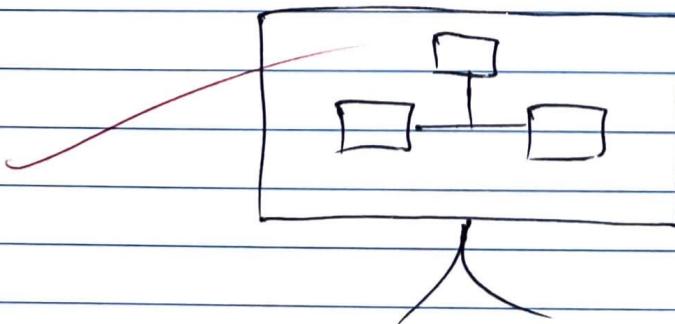
Interface : HTTP / REST

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

has no database layer : no transactions

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

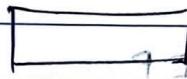
2) File storage:



- 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

Fibre Channel : Interface

limit on data transfer : 2Gb/s

- Transport : FC, serial connection

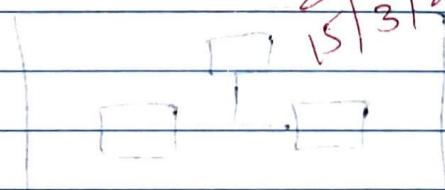
- Interface : Direct attached or SAN

- Use cases : Low latency, best for structure data, time and stable with no voids.

(A)

FC : supports SFP (S)

SFP



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' section and a table of 'General purpose buckets'. There are two buckets listed: 'elasticbeanstalk-eu-north-1-' and 'elasticbeanstalk-us-east-1-'. Each bucket entry includes a 'Name', 'AWS Region', 'IAM Access Analyzer', 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'.

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)

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, including the 'AWS Region' (set to 'Europe (Stockholm) eu-north-1'), 'Bucket type' (set to 'General purpose'), and 'Bucket name' (set to 'myawsbucketcc16'). The 'Object Ownership' section at the bottom is collapsed.

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

The screenshot shows the 'Create bucket' page in the AWS Management Console. The 'Object Ownership' section is expanded, showing the 'ACLs disabled (recommended)' option selected. Other options like 'ACLs enabled' are also shown. The 'Object Ownership' section also includes the note: 'Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.'

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 'Block all public access':

- **Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- **Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore 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**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- **Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

A yellow warning box contains the text: "Turning off block all public access might result in this bucket and the objects within becoming public". It also includes a note from AWS recommending to turn on block all public access unless required for specific use cases like static website hosting. At the bottom, there is a checkbox for acknowledging the consequences of turning off the setting.

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 rest of the interface remains identical to the previous screenshot, including the detailed description of the four sub-settings and the yellow warning box at the bottom.

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 interface. At the top, a green banner displays the message "Successfully created bucket 'myawsbucketcc16'". Below the banner, the "General purpose buckets" tab is selected. A table lists three buckets: "elasticbeanstalk-eu-north-1-533267428271" (created March 13, 2024), "elasticbeanstalk-us-east-1-533267428271" (created February 26, 2024), and "myawsbucketcc16" (created April 8, 2024). The "Create bucket" button is visible at the top right of the table area.

8. Now click on the bucket that you have created

This screenshot shows the same AWS S3 console interface as the previous one, but with a slight difference. The "myawsbucketcc16" bucket is now highlighted in blue, indicating it has been selected. The other two buckets remain unselected. The rest of the interface, including the sidebar and the table structure, is identical to the previous screenshot.

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 'Upload' (which is highlighted in orange), 'Create folder', 'Delete', 'Actions', 'Copy S3 URI', 'Copy URL', 'Download', and 'Open'. 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

This screenshot shows the 'Upload' step in the AWS S3 console. It displays a list of files and folders to be uploaded. A large blue dashed box at the top 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)'. The table has columns for 'Name', 'Folder', and 'Type'. Each row contains a checkbox, the file name, its folder path, and its type (all listed as 'application/'). There are also 'Remove', 'Add files', and 'Add folder' buttons above the table. A search bar 'Find by name' is present. At the bottom right, there's a large 'Upload' button.

The screenshot shows the AWS S3 console with a progress bar at 7% for an upload task. Below the progress bar, it says "Total remaining: 12 files: 66.8 MB (93.14%)". A table lists 12 files and folders, all currently in a "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 after the upload has completed successfully. A green banner at the top says "Upload succeeded" and "View details below". Below the banner, the file list shows all 12 files and folders with a "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, there's a navigation bar with the AWS logo, a search bar, and account information for 'Stockholm' and 'omsheste0550'. Below the navigation bar, the URL shows 'Amazon S3 > Buckets > myawsbucketccl6'. The main area is titled 'myawsbucketccl6' with a 'Info' link. A horizontal menu bar below the title includes 'Objects' (which is underlined), 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. Under the 'Objects' section, there's a sub-menu with 'Objects (1) Info'. It shows one object: 'CCL/' (Type: Folder). There are buttons for 'Actions' (with options like Copy S3 URI, Copy URL, Download, Open, Delete, Create folder, and Upload), a search bar ('Find objects by prefix'), and sorting options for Name, Type, Last modified, Size, and Storage class. At the bottom of the page, there are links for CloudShell, Feedback, and a copyright notice: '© 2024, Amazon Web Services, Inc. or its affiliates.' followed by 'Privacy', 'Terms', and '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. At the top, there's a message: 'Amazon S3 Transfer acceleration is not available for your bucket because it is located in an unsupported Region. [Learn more](#)'. Below this, there are three sections: 'Object Lock' (status: Disabled), 'Requester pays' (status: Disabled), and 'Static website hosting' (status: Disabled). Each section has an 'Edit' button to its right. The 'Static website hosting' section also contains a note: 'Use this bucket to host a website or redirect requests. [Learn more](#)'. At the bottom of the page, there are links for CloudShell, Feedback, and a copyright notice: '© 2024, Amazon Web Services, Inc. or its affiliates.' followed by 'Privacy', 'Terms', and '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 top navigation bar, 'Amazon S3' is selected. Below it, there's an 'Account snapshot' section with a link to 'View Storage Lens dashboard'. Under 'General purpose buckets', there's a table listing three buckets:

Name	AWS Region	IAM Access Analyzer	Creation date
elasticbeanstalk-eu-north-1-553267428271	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	March 13, 2024, 16:47:02 (UTC+05:30)
elasticbeanstalk-us-east-1-553267428271	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)

Actions for the selected bucket ('myawsbucketcc16') 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 confirmation step where the user types 'permanently delete' into a text input field. Buttons for 'Cancel' and 'Empty' are at the bottom.

The screenshot shows the 'Empty bucket: status' confirmation dialog. It displays a summary table with the following data:

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

Below this, a 'Failed to delete (0)' section shows a table with no data, indicating 'No failed object deletions'.

Delete bucket Info

Delete bucket "myawsbucketcl6"?

To confirm deletion, enter the name of the bucket in the text input field.

myawsbucketcl6

Cancel **Delete bucket**

https://eu-north-1.console.aws.amazon.com/console/home?region=eu-north-1

Successfully deleted bucket "myawsbucketcl6"

General purpose buckets Info **Create bucket**

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)

CloudShell Feedback

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.

✓ 2020 MAF

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 IAM

- 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) Trust 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. Key statistics include 0 Instances (running), 0 Auto Scaling Groups, 0 Dedicated Hosts, 0 Elastic IPs, 1 Instances, 1 Key pairs, 0 Load balancers, 0 Placement groups, 2 Security groups, 0 Snapshots, and 0 Volumes. The 'Launch instance' section allows launching an instance in the Europe (Stockholm) Region. The 'Service health' section shows the AWS Health Dashboard with the status 'This service is operating normally.' The 'Offer usage (monthly)' section tracks Linux EC2 Instances and Storage space on EBS. The 'Account attributes' section provides account-level information.

2. Go to IAM dashboard

The screenshot shows the AWS IAM Dashboard. Under 'Security recommendations', it lists 'Add MFA for root user' and 'Root user has no active access keys'. The 'IAM resources' section displays 0 User groups, 0 Users, 3 Roles, 0 Policies, and 0 Identity providers. The 'AWS Account' sidebar shows the Account ID (533267428271), Account Alias (Create), and Sign-in URL (https://533267428271.signin.aws.amazon.com/console). The 'Quick Links' section includes a link to 'My security credentials'. The 'Tools' section includes CloudShell and Feedback links.

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

The screenshot shows the AWS IAM Dashboard. On the left, there's a sidebar with various navigation options like 'Access management', 'User groups', 'Users', etc. In the center, there's a 'Security recommendations' section with a warning about 'Add MFA for root user' and a note that 'Root user has no active access keys'. Below that is an 'IAM resources' section showing 0 User groups, 0 Users, 3 Roles, 0 Policies, and 0 Identity providers. At the top right, there's a modal window titled 'Create alias for AWS account 533267428271'. It has a 'Preferred alias' input field containing 'myaliascl7', a note that it must be between 1 and 63 characters, and a 'Create alias' button.

This screenshot shows the same IAM dashboard after the alias has been created. The success message 'Alias myaliascl7 created for this account.' is displayed at the top. The rest of the interface is identical to the previous screenshot, including the security recommendations, IAM resources summary, and the 'Create alias' modal which is now closed.

4. Click on “users” in the left column

The screenshot shows the AWS Identity and Access Management (IAM) service interface. In the left sidebar, under the 'Access management' section, the 'Users' option is selected. The main content area is titled 'Users (0) Info' and contains a message stating, '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 note at the bottom of the table says, 'No resources to display'. At the top right of the main area are 'Create user', 'Delete', and other navigation icons.

5. Click on the Create Users button

The screenshot shows the 'Specify user details' step of the AWS IAM 'Create user' wizard. On the left, a sidebar lists steps: Step 1 (selected), 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?'. It contains 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 Keypairs, or a backup credential for emergency account access.' At the bottom of the 'User details' section, there are three password options: 'Console password', 'Autogenerated password', and 'Custom password'.

Are you providing console access to a person?

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

Set permissions

Add user to an existing group, or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

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.
- 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](#)

Set permissions boundary - optional

Cancel **Previous** **Next**

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

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 <input type="text" value="NewUser"/>	Value - optional <input type="text" value="Om_Shete"/>	Remove
---	---	---------------

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

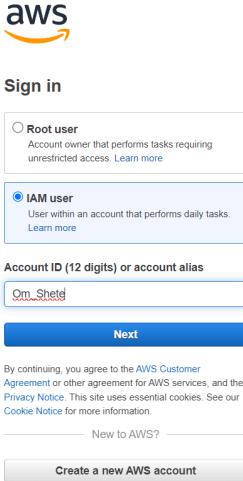
Cancel **Previous** **Create user**

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 details, including the URL (<https://myaliascd7.signin.aws.amazon.com/console>), user name (Om_Shete), and a masked console password. Buttons for "Email sign-in instructions" (with a link icon), "Cancel", "Download .csv file", and "Return to users list" are visible. On the left sidebar, the "Specify user details" step is highlighted.

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

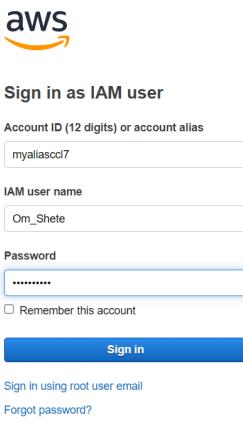
Logging in as the new User & Checking their permissions
Enter the new user's name and psw saved earlier



The screenshot shows the AWS sign-in interface. It has two radio button options: "Root user" (unchecked) and "IAM user" (checked). Below the "IAM user" option is a text input field containing "Om_Shete". A blue "Next" button is at the bottom. To the right of the input fields, there is a link "New to AWS?" and a button "Create a new AWS account". At the bottom, a copyright notice reads "© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved." and a language selector "English".



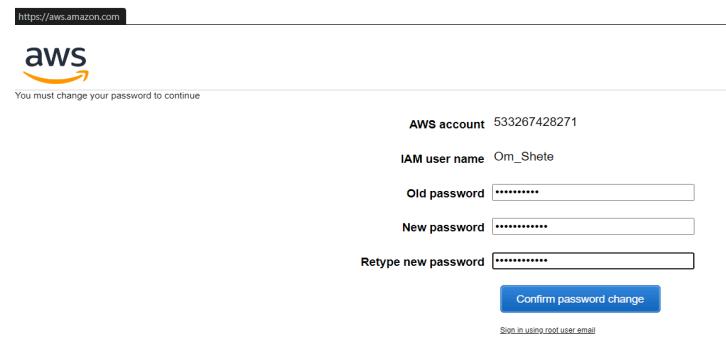
The screenshot shows the AWS Support landing page. It features a purple gradient background with the text "AWS Support" in white. Below it is the tagline "Save time and move faster with expert guidance and assistance". There is a "Select a Support plan >" button. At the bottom, there is a copyright notice "© 1996-2024, Amazon Web Services, Inc. or its affiliates." and a language selector "English".



The screenshot shows the "Sign in as IAM user" interface. It includes fields for "Account ID (12 digits) or account alias" (containing "myaliascd7"), "IAM user name" (containing "Om_Shete"), and "Password" (containing "*****"). There is a "Remember this account" checkbox, which is unchecked. A blue "Sign in" button is at the bottom. Below the form, there are links for "Sign in using root user email" and "Forgot password?". At the bottom, there is a copyright notice "© 1996-2024, Amazon Web Services, Inc. or its affiliates." and a language selector "English".



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". There is a "Learn more >" button and a cartoon robot icon holding a thumbs-up. At the bottom, there is a copyright notice "© 1996-2024, Amazon Web Services, Inc. or its affiliates." and a language selector "English".



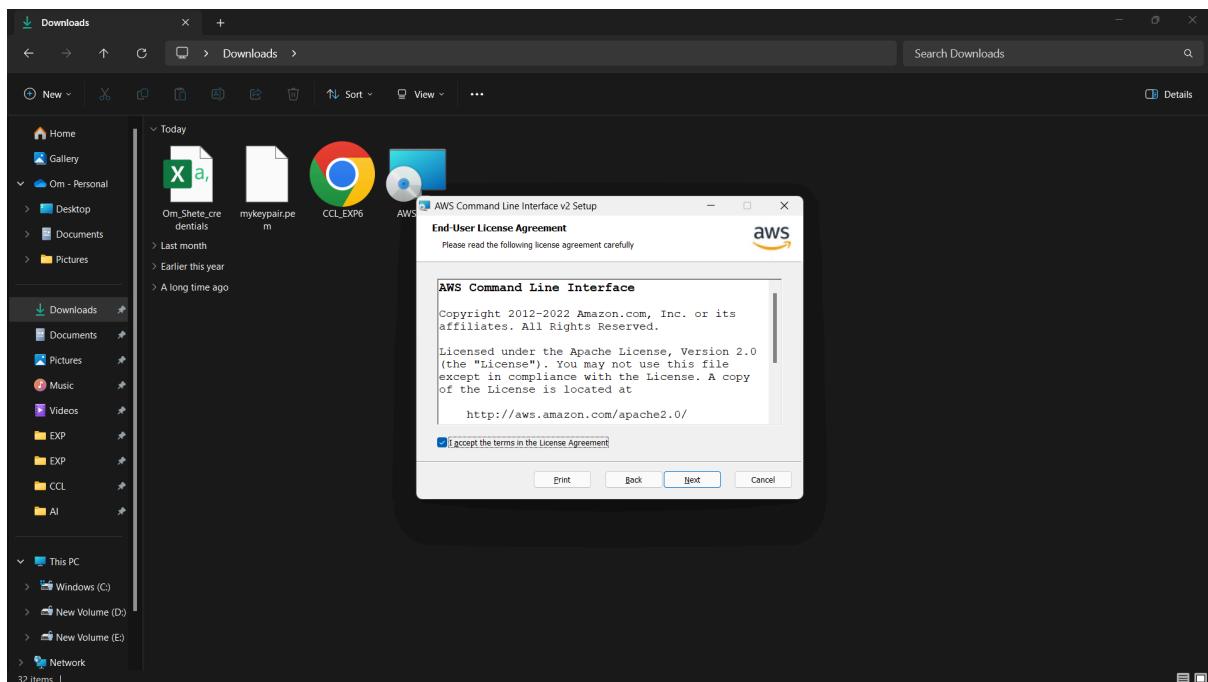
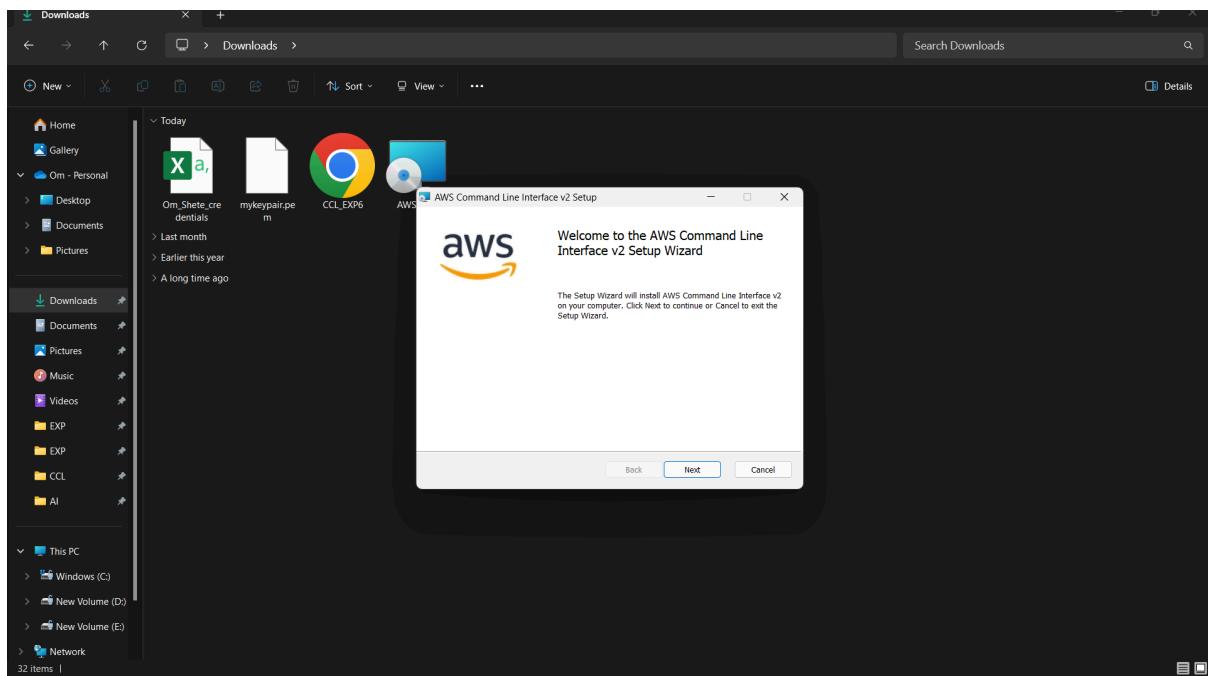
The screenshot shows the "Change Password" interface. It displays the AWS account number "533267428271" and the IAM user name "Om_Shete". It has three password input fields: "Old password" (containing "*****"), "New password" (containing "*****"), and "Retype new password" (containing "*****"). A blue "Confirm password change" button is at the bottom. Below the buttons, there is a link "Sign in using root user email" and a copyright notice "© 1996-2024, Amazon Web Services, Inc. or its affiliates." and a language selector "English".

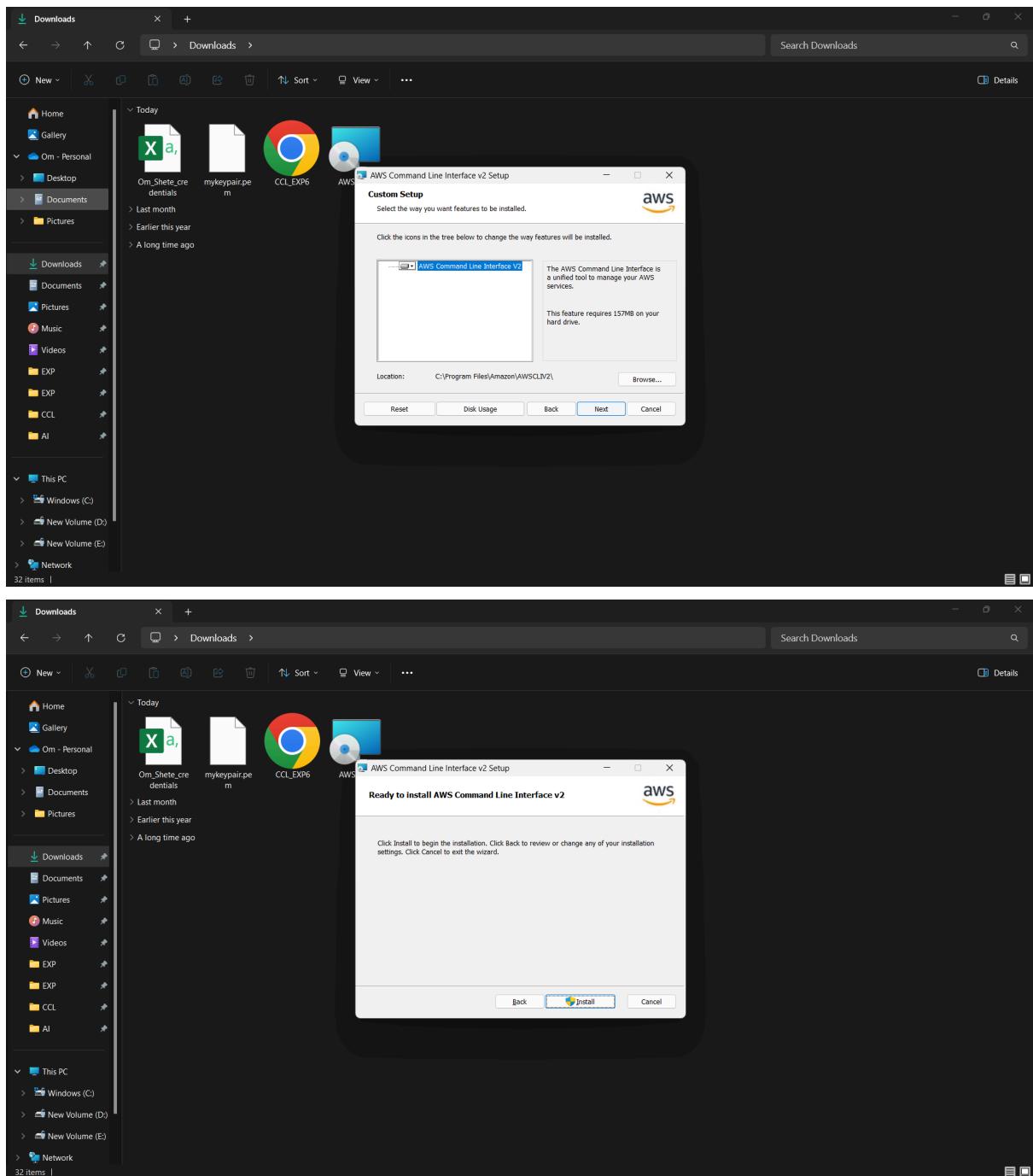
After logging in, you will notice that you don't have permission to do anything yet

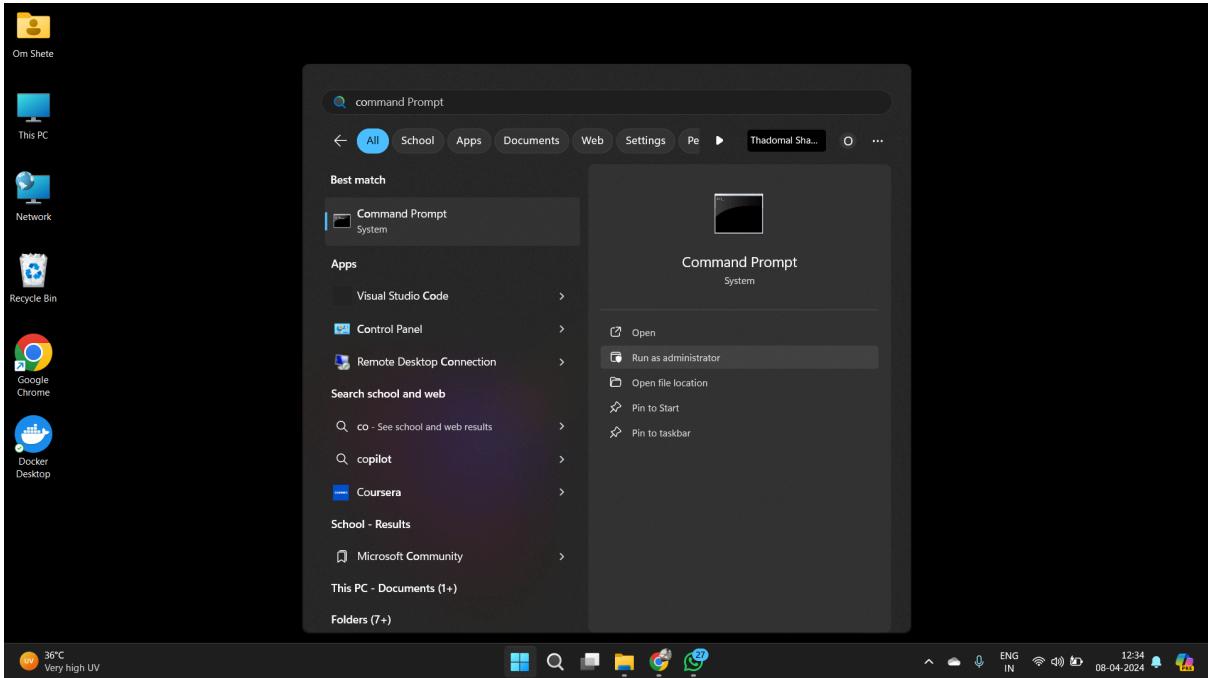
The image consists of three vertically stacked screenshots of the AWS console, all showing an "Access denied" error message.

- Screenshot 1: AWS Console Home**
The "Recently visited" section shows a placeholder icon and the message "No recently visited services". Below it, links for EC2, S3, RDS, and Lambda are shown. The "Applications" section shows a table with one row, but the details are obscured by a red box containing the "Access denied" message.
- Screenshot 2: EC2 Dashboard**
The left sidebar shows navigation options like EC2 Dashboard, EC2 Global View, Events, Instances, Images, and Elastic Block Store. The main "Instances" section has a search bar and a table header. A prominent red box highlights the error message: "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**
The left sidebar shows Buckets, Storage Lens, and other S3 features. The main "Buckets" section includes an "Account snapshot" summary and a table for managing buckets. A red box highlights the error message: "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 it’s 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:\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. On the left, there's a navigation sidebar with options like Dashboard, Access management (Users, Roles, Policies, Identity providers, Account settings), Access reports (Access Analyzer, External access, Unused access, Analyzer settings, Credential report, Organization activity), and Service control policies (SCPs). The main area is titled "Identity and Access Management (IAM)". It shows basic user information: Created (April 08, 2024, 12:49 UTC+05:30), Last console sign-in (Never), and an Access key (Access key 2, with a "Create access key" button). Below this, there are tabs for Permissions, Groups, Tags, Security credentials (which is selected), and Access Advisor. Under the "Permissions" tab, it says "Permissions policies (1)". A table lists one policy: "Policy name: IAMUserChangePassword" (Type: AWS managed, Attached via Directly). There are buttons for "Add permissions" and "Create inline policy". Below this, there's a section for "Permissions boundary (not set)" and another for "Generate policy based on CloudTrail events". At the bottom, there's a URL (https://us-east-1.console.aws.amazon.com/iam/home?region=eu-north-1#/users/details/Om_Shete/add-permissions) and standard AWS footer links.

This screenshot shows the "Policies" page under the IAM section. The navigation sidebar is identical to the previous screenshot. The main content shows the "IAMUserChangePassword" policy. The "Policy details" section includes the Type (AWS managed), Creation time (November 15, 2016, 05:55 UTC+05:30), Edited time (November 16, 2016, 04:48 UTC+05:30), and ARN (arn:aws:iam::aws:policy/IAMUserChangePassword). Below this, there are tabs for Permissions, Entities attached, Policy versions (2), and Access Advisor. The "Permissions defined in this policy" section shows "Allow (1 of 407 services)" for the IAM service with an access level of Limited: Read, Write. There are buttons for "Summary" and "JSON". At the bottom, there's a URL (https://us-east-1.console.aws.amazon.com/iam/policies/IAMUserChangePassword?region=eu-north-1#), standard AWS footer links, and a "CloudShell" button.

Click on the “Manage” Hyperlink

Screenshot 1: IAM User Summary

Screenshot 2: Select MFA device

Screenshot 3: Authenticator app setup

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) service interface. On the left, there's a navigation sidebar with options like Dashboard, Access management (User groups, Roles, Policies, Identity providers, Account settings), Access reports (Access Analyzer, External access, Unused access, Analyzer settings), Credential report, Organization activity, CloudShell, and Feedback.

The main content area displays the 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). It also shows the creation date (April 08, 2024) and last console sign-in (Never).

Below the summary, tabs for Permissions, Groups, Tags, Security credentials (which is selected), and Access Advisor are visible. The "Console sign-in" section contains a "Console sign-in link" (https://myaliascl7.signin.aws.amazon.com/console), a "Console password" (updated 25 minutes ago), and the last console sign-in (Never).

The "Multi-factor authentication (MFA)" section shows one MFA device assigned. Buttons for Remove, Resync, and Assign MFA device are present. At the bottom of the page, there are links for Privacy, Terms, and Cookie preferences.

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

Firebase, real-time database, Cloud Functions

Cloud storage, Cloud functions

8.1 AWS Relational Database

- 1) 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.	1) 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 DBaaS.

DBaaS is a fully managed service that provides a secure and reliable database solution for your applications.

⇒ Database solutions are provided by Amazon RDS.

1) Storage types:

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

These options differ in terms of performance,

durability, and cost.

2) Endpoint

⇒ In AWS DBaaS, an endpoint refers to the network address used to access a database instance.

Endpoints are used by applications to 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 → primarily with MySQL
- 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 Availability Zone 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 instance.

A*

Sep 2024

18/3/2024

In your writing, if you write with a red pen, it is considered as independent work, quoted in your own handwriting → original authorship by you

Output:

Amazon RDS Dashboard

The screenshot shows the Amazon RDS dashboard for the EU North (Stockholm) region. It displays various resources and metrics. A prominent blue banner at the top introduces the Aurora I/O-Optimized storage configuration.

Resources

- DB Instances (0/40)
- Allocated storage (0 TB/100 TB)
- Increase DB Instances limit
- 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: 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.

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.

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

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

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".

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](#)

Storage

Allocated storage GiB

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".

Virtual private cloud (VPC)

DB subnet group

Public access

VPC security group (firewall)

RDS Proxy

Certificate authority - optional

Screenshot 1: AWS RDS Free Tier Configuration

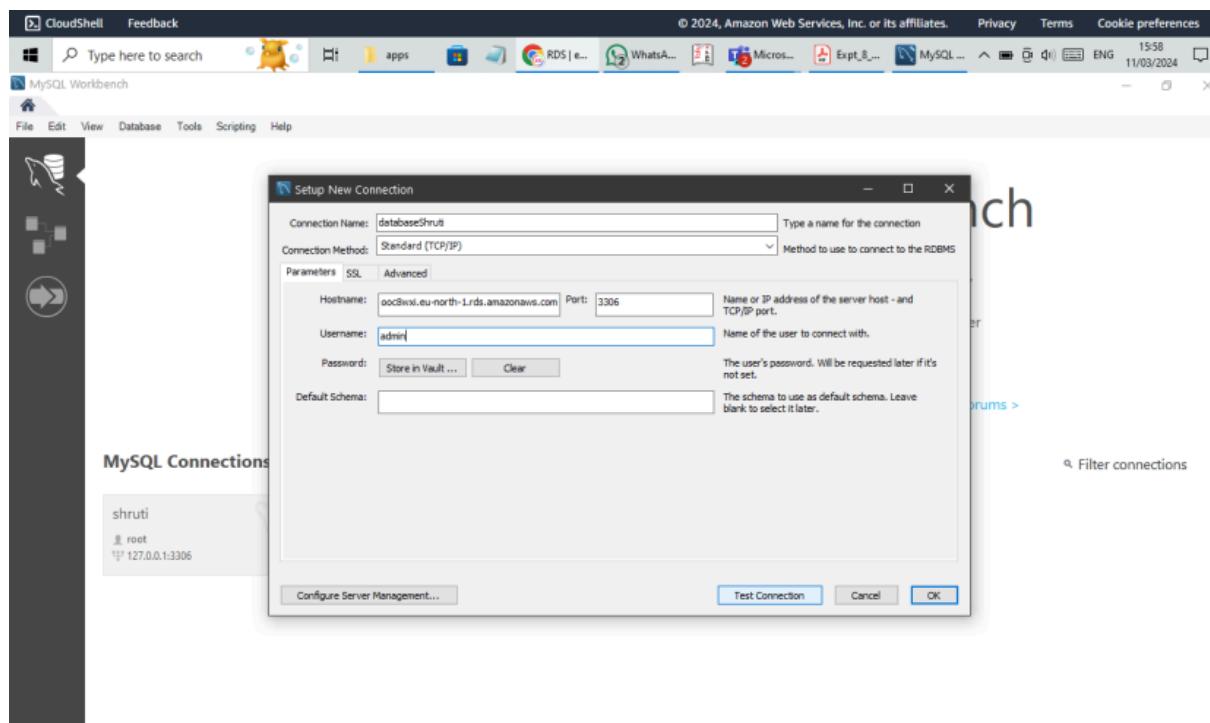
The screenshot shows the AWS RDS Free Tier configuration page. It includes sections for 'Enable Enhanced Monitoring', 'Additional configuration' (with notes on encryption, backup, and maintenance), 'Estimated monthly costs' (noting the free tier for 12 months), and a note about responsibilities. At the bottom is a 'Create database' button.

Screenshot 2: Creating a Database

This screenshot shows the 'Creating database database-1' step. It displays a progress bar and a note about using Aurora I/O-Optimized and Blue/Green Deployments. The 'Databases' table shows 'database-1' in the 'Creating' state.

Screenshot 3: Database Overview

This screenshot shows the 'database-1' database details page. It includes tabs for 'Summary', 'Connectivity & security', 'Monitoring', 'Logs & events', 'Configuration', 'Maintenance & backups', 'Tags', and 'Recommendations'. The 'Connectivity & security' tab is active, displaying information like the endpoint, VPC, and security groups.



VPC dashboard EC2 Global View

Filter by VPC: Select a VPC

Your VPCs (1) Info

VPC ID : **vpc-001671399ec9d2700** X Clear filters

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

Select a VPC above

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

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... X	sg-01031f3c06b9fd512

Add rule

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

EC2 Dashboard X Inbound security group rules successfully modified on security group (sg-01031f3c06b9fd512 | default) Details

EC2 Global View

Events

Instances

- Instances
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations
- New

Images

AMIs

CloudShell Feedback

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

MySQL Workbench

File Edit View Database Tools Scripting Help

Manage Server Connections

MySQL Connections

- shruti
- databaseShruti

Connection Name: databaseShruti

Connection Method: Standard (TCP/IP)

Parameters SSL Advanced

Hostname: databaseshruti.cn04aoc8exi.eu-north-1.rds.amazonaws.com Port: 3306

Username: admin

Password: Store in Vault ... Clear

Default Schema:

New Delete Duplicate Move Up Move Down Test Connection Close

MySQL Connection

shruti

root

127.0.0.1:3306

MySQL Workbench

MySQL Connections

- shruti
- databaseShruti

Connection Name: databaseShruti

Connection Method: Standard (TCP/IP)

MySQL Workbench

MySQL Workbench

Successfully made the MySQL connection

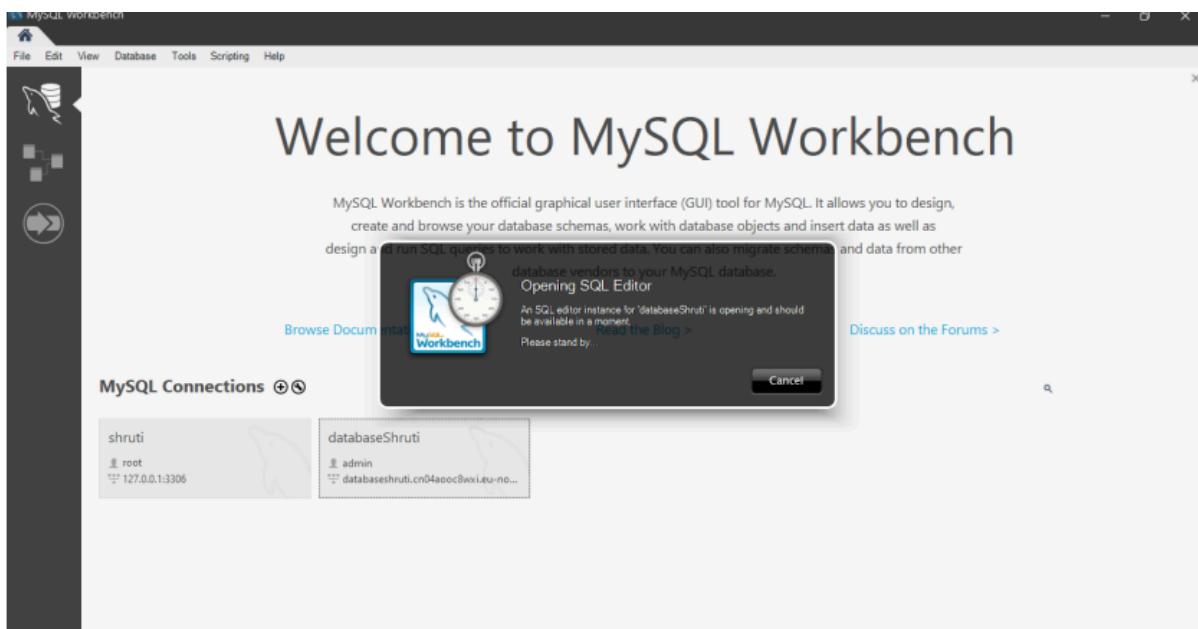
Information related to this connection:

Host: databaseshruti.cn04aoc8exi.eu-north-1.rds.amazonaws.com
Port: 3306
User: admin
SSL enabled with TLS_AES_128_GCM_SHA256

A successful MySQL connection was made with the parameters defined for this connection.

OK

New Delete Duplicate Move Up Move Down Test Connection Close



This screenshot shows the MySQL Workbench interface after connecting to the "databaseShruti" database. The title bar now says "databaseShruti". The left sidebar includes sections for MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), INSTANCE (Startup / Shutdown, Server Logs, Options File), and PERFORMANCE (Dashboard, Performance Reports, Performance Schema Setup). The main area has tabs for Administration and Schemas, with "Administration" selected. In the center, the "Query 1" tab shows the following SQL commands:

```
1 • create database shruti;
2 • use shruti;
```

The "Output" pane at the bottom displays the results of these queries:

Action	Time	Message	Duration / Fetch
create database shruti	1 16:18:29	1 row(s) affected	0.210 sec
use shruti	2 16:18:31	Error Code: 1049. Unknown database 'path'	0.219 sec

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	0.234 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

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

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


```

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
15 •  select * from template_table;
            
```
- 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
- Output Window:** Shows the execution log with the following entries:

Action	Time	Message	Duration / Fetch
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
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:

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](#)

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

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**

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.

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.

delete me

Cancel **Delete**

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

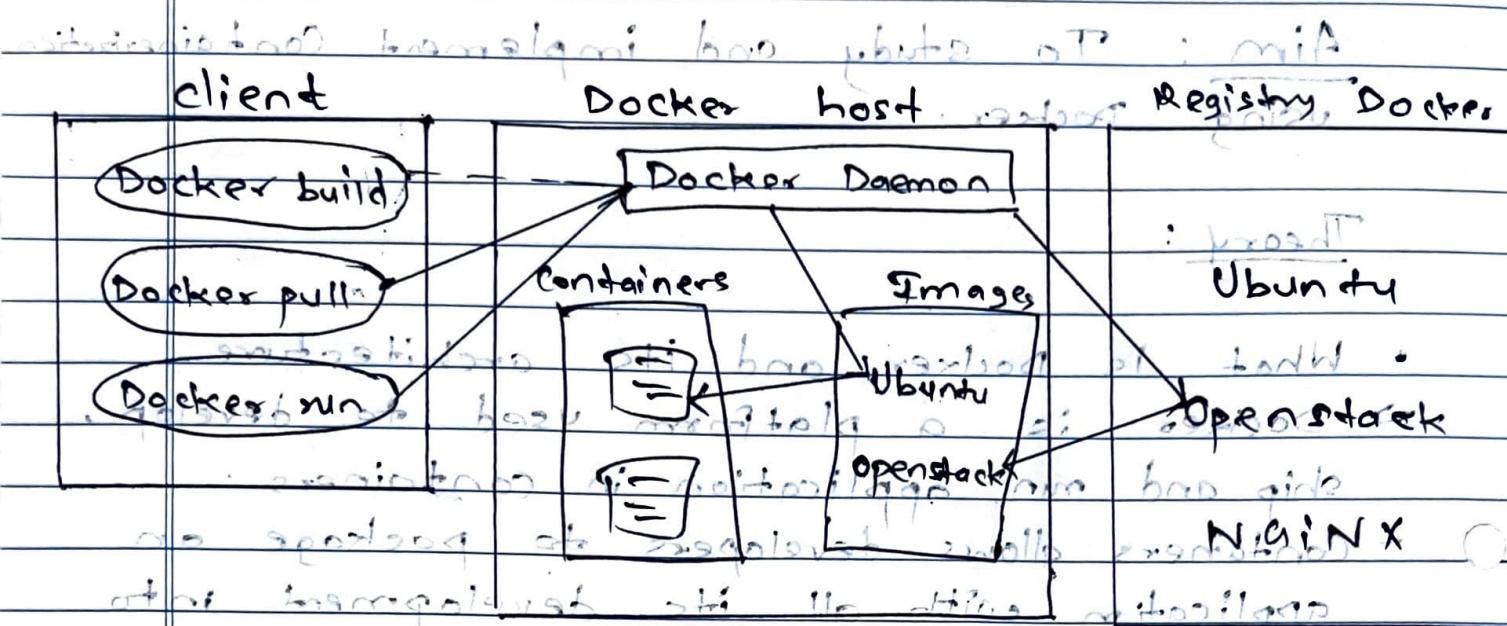
Group resources **Modify** Actions **Restore from S3** **Create database**

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.



Build → pull → run

- Benefits of Containerization

⇒ Containerization offers several advantages in

- Portability: Containers encapsulate an application and its dependencies, making them portable across different environments. From the development to production, the application does not need to change its environment.

b) Isolation → application cannot interact with other applications or dependencies outside the container.

⇒ Containers provide isolated environments for applications, ensuring that changes or dependencies outside the container do not affect others.

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 vs 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, similarities, differences

Ans :- Similarities :-

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.

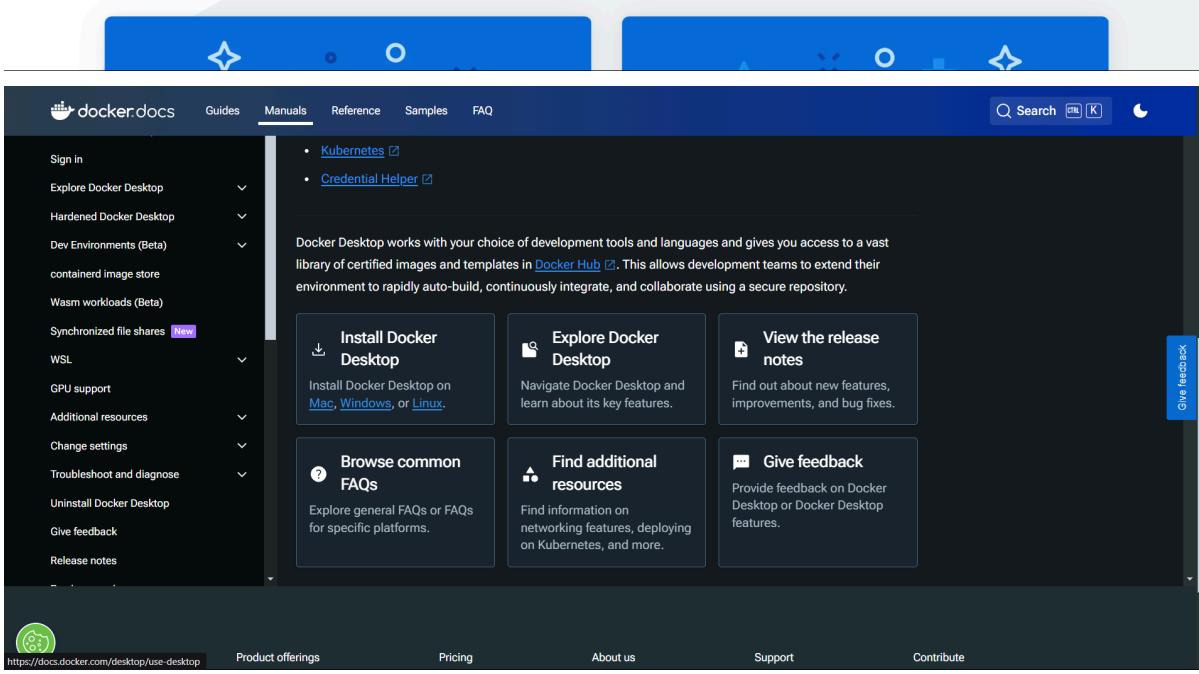
✓ (A)

SP
18/3/2024

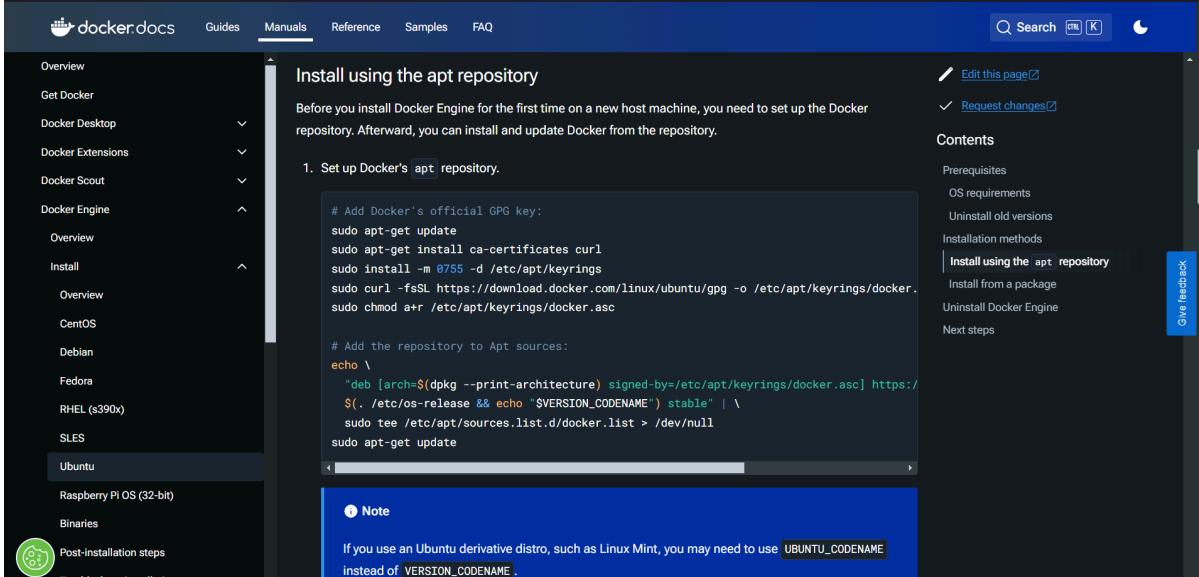
Output:



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 buttons: "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". Below these, there's a summary of Docker Desktop's capabilities and links to "Install Docker Desktop", "Explore Docker Desktop", "View the release notes", "Browse common FAQs", "Find additional resources", and "Give feedback".



This screenshot shows the "Ubuntu" documentation page for Docker Engine. It starts with an "Install using the apt repository" section, which contains a step-by-step guide and a terminal command for adding the Docker GPG key and repository. A note at the bottom states: "If you use an Ubuntu derivative distro, such as Linux Mint, you may need to use `UBUNTU_CODENAME` instead of `VERSION_CODENAME`". On the right side, there are links for "Edit this page", "Request changes", and a "Contents" sidebar.

The screenshot shows the Docker documentation page for Ubuntu. The left sidebar lists various operating systems: Overview, Get Docker, Docker Desktop, Docker Extensions, Docker Scout, Docker Engine, Overview, Install, Overview, CentOS, Debian, Fedora, RHEL (x86_64), SLES, Ubuntu, Raspberry Pi OS (32-bit), and Binaries. The 'Ubuntu' link is highlighted. Below the sidebar, the main content area has a blue header bar with the text: "If you use an Ubuntu derivative distro, such as Linux Mint, you may need to use `UBUNTU_CODENAME` instead of `VERSION_CODENAME`". The main content is divided into sections: "2. Install the Docker packages." (with "Latest" selected), "To install the latest version, run:" (containing a terminal command: `$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-`), "3. Verify that the Docker Engine installation is successful by running the `hello-world` image." (containing a terminal command: `$ sudo docker run hello-world`), and a "Tip" section with the text: "Receiving errors when trying to run without root?". On the right side, there's a "Contents" sidebar with links to Prerequisites, OS requirements, Uninstall old versions, Installation methods (highlighted), Install using the `apt` repository, Install from a package, Uninstall Docker Engine, and Next steps. There are also "Edit this page" and "Request changes" buttons at the top right.

The screenshot shows the Docker Desktop application window. The left sidebar includes icons for Containers, Images, Volumes, Builds, Dev Environments (BETA), Docker Scout, and Extensions (with a plus sign). The main area is titled "Containers" and features a central image of three overlapping containers. Below it, the text "Your running containers show up here" and "A container is an isolated environment for your code". At the bottom of this section are two cards: "What is a container?" (5 mins) and "How do I run a container?" (6 mins). A "View more in the Learning center" link is located between them. The bottom status bar shows "Engine running", system resources (RAM 1.64 GB, CPU 0.25%), and user status (Signed in). The version "v4.28.0" is also visible in the bottom right corner.

The screenshot shows the Docker desktop application's main interface. On the left is a sidebar with icons for Containers, Images, Volumes, Builds, Dev Environments (BETA), Docker Scout, Extensions (with a plus icon and 'Add Extensions'), and a three-dot menu. The main area is titled 'Images' with a 'Give feedback' link. It has tabs for 'Local' and 'Hub'. Below that is a search bar and a table with columns: Name, Tag, Status, Created, Size, and Actions. One item is listed: 'ubuntu' with tag 'latest', status 'In use', created '17 days ago', size '77.86 MB'. At the bottom, it says 'Showing 1 item'. Below the table is a 'walkthroughs' section with two cards: 'How do I run a container?' (with steps 1-4) and 'Run Docker Hub images' (with a 'docker hub image' icon). A link 'View more in the Learning center' is at the bottom.

The terminal window shows the following session:

```
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.

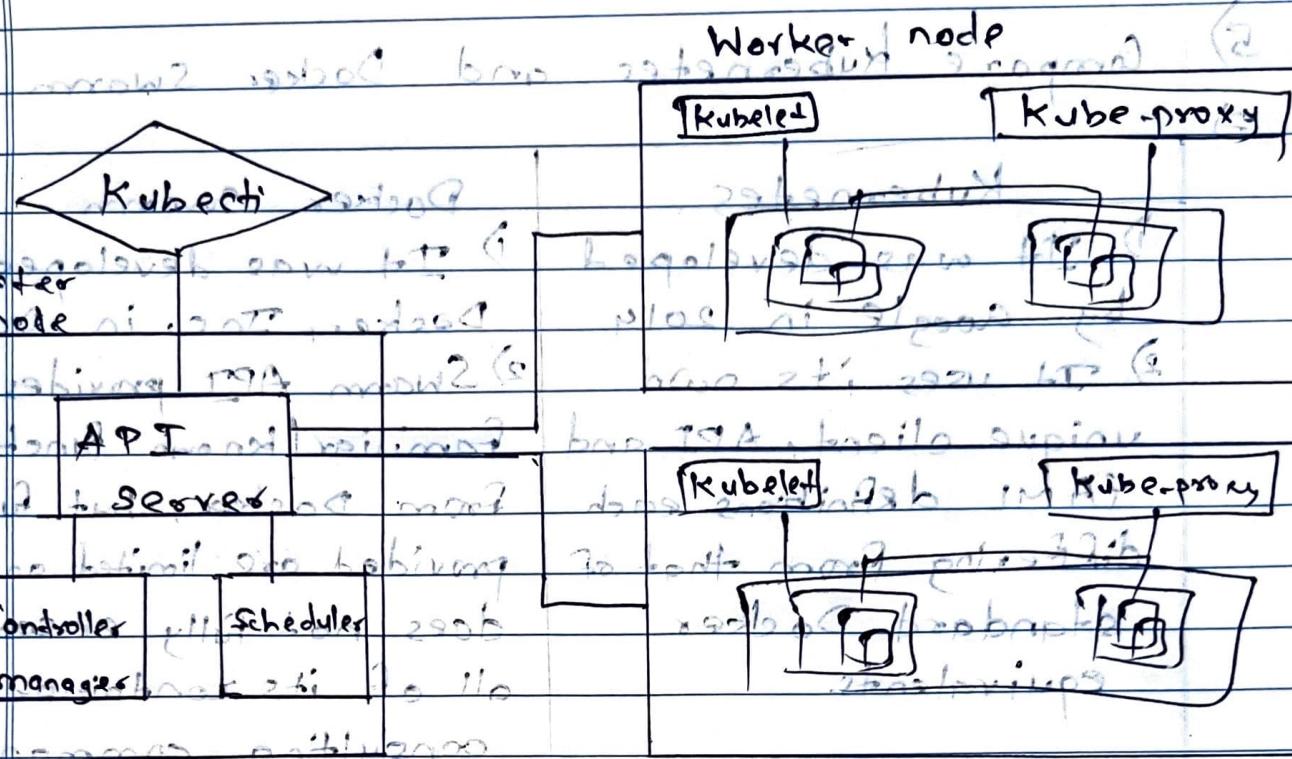
Cloud Formation

- 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: $\frac{1}{2}$ min
- vii) Batch execution
- ix) Horizontal scaling: $\frac{1}{2}$ min
- x) Multi-IP failover: $\frac{1}{2}$ min
- x) Designed for extensibility: $\frac{1}{2}$ min

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 status 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 (2) from a stronger LTO (2)

4) Differences between PODs and nodes

↳ nodes are separate physical or virtual machines
 ↳ pods are managed by kubelets in nodes

1) A pod is smallest deployable unit in Kubernetes (2)

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.

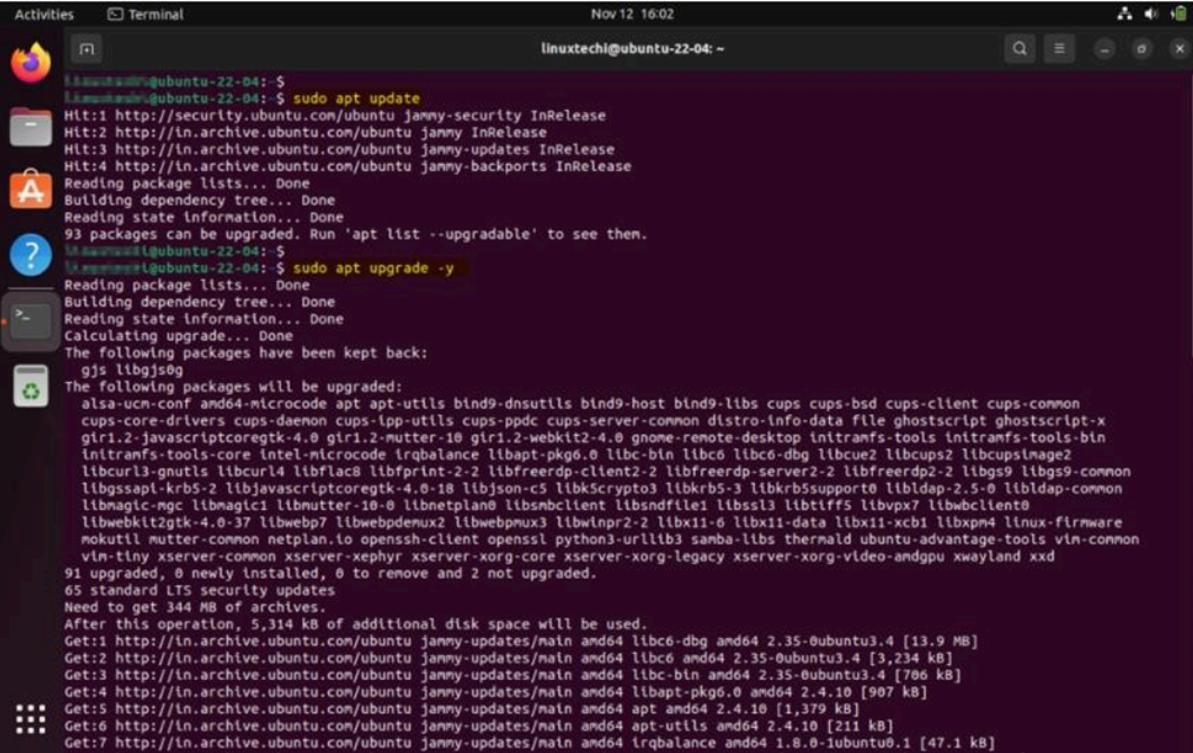
3) On other hand, supports a simple architecture, so in terms of sheer speed, it always has an added advantage.

4) It cannot do auto-scaling, but Docker can do auto-scaling.

5) It supports a more complex, flexible architecture with stronger service guarantees due to which performance slows down.

6) It supports auto-scaling.

7) It can do auto-scaling.

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 libcurlimage2
  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 libjansson-c5 libksyms 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 libc6-dbg amd64 2.35-0ubuntu3.4 [13.9 MB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libc6 amd64 2.35-0ubuntu3.4 [3,234 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libc6-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=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
  '$(cat /etc/os-release && echo "$VERSION_CODENAME")' stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
[1] 12000 ? 0:00 sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
[1] 12001 ? 0:00 sudo chmod a+r /etc/apt/keyrings/docker.gpg
[1] 12002 ? 0:00 sudo echo |
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
  '$(cat /etc/os-release && echo "$VERSION_CODENAME")' stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

```

Om Shete - C3 - 2103163

```
lunuxbuzz@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 https://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)
```

```
lunuxbuzz@ubuntu-22-04: $ sudo usermod -aG docker $USER
lunuxbuzz@ubuntu-22-04: $ newgrp docker
lunuxbuzz@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.
```

```
lunuxbuzz@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
lunuxbuzz@ubuntu-22-04:~$ 
lunuxbuzz@ubuntu-22-04:~$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
lunuxbuzz@ubuntu-22-04:~$ 
lunuxbuzz@ubuntu-22-04:~$ minikube version
minikube version: v1.32.0
commit: 8220a6eb95f0a4d75f7f2d7b14cef975f050512d
lunuxbuzz@ubuntu-22-04:~$ 
lunuxbuzz@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
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:~$
```

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:v20231011-8b53cabef0
  ■ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v20231011-8b53cabef0
💡 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 window. The URL is `127.0.0.1:36529/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/port/80`. The dashboard is titled "kubernetes" and displays the "Workloads" section. On the left sidebar, there are links for Workloads, Service, Config and Storage, and a general search bar. The main content area shows three tabs: "Workload Status", "Deployments", and "Replica Sets". Under "Deployments", a single deployment named "nginx-web" is listed, using the "nginx" image with labels "app: nginx-web" and "pod-template-hash: 5b757f798d". Under "Pods", a pod named "nginx-web-5b757f798d-qnbzq" is shown, also using the "nginx" image and matching the deployment's labels. The "Replica Sets" tab is currently empty.

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory (bytes)
nginx-web	nginx	app: nginx-web pod-template-hash: 5b757f798d	minikube	Running	0	-	-

Name	Images	Labels	Pods

Assignment No: 1

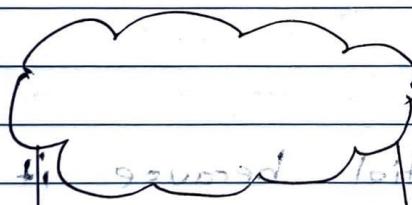
Q.1 What is fog computing? and explain the need of fog computing.

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.

Processing power at local level is less



Cloud node



Fog Node 1

Fog Node 2

End device		End device
------------	--	------------

- 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.

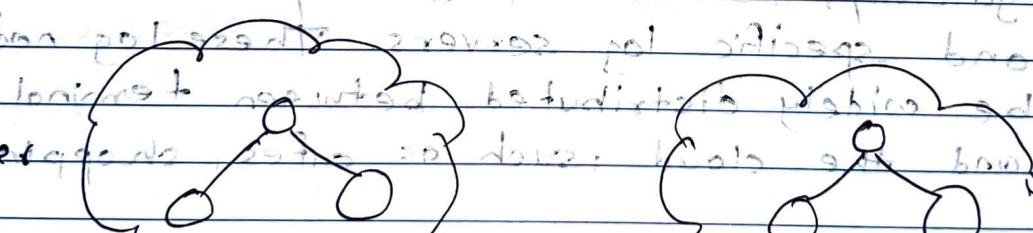
- Fog computing reduces latency, enhances real-time processing, and improves efficiency for application like IoT and edge devices.
- This decentralized approach also enhances privacy and security by processing data closer to its source, reducing the need for extensive data transfers to centralized servers.

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 this in natural ai model with a royal path

path to a goal → to solving has shown in

using distance values between path to state taken

with and changing values, condition, operating

on each path to a range of values for

fog layers to be used to find the path and

algorithm to have more useful in royal path

and also to get the best path

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.

- 1) Fog computing has a low latency.
- 2) Response time of the system is high.
- 3) Fog computing has high security.

- 4) Success speed is high even more compared to cloud computing.
- 5) Multiple data sources and devices.
- 6) Mobility is high.

- 7) Supported in fog computing (Location awareness), but not in cloud computing.

- | Fog Computing | Cloud Computing |
|---|-----------------|
| 1) Cloud computing has high latency compared to cloud. | |
| 2) Response time of the system is low. | |
| 3) Cloud computing has less security compared to fog computing. | |

- 4) Success speed is high depending on the VM connectivity.
- 5) Multiple data sources can be integrated.
- 6) In cloud computing, the mobility is limited.
- 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 b/w technologies and no problem~~

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

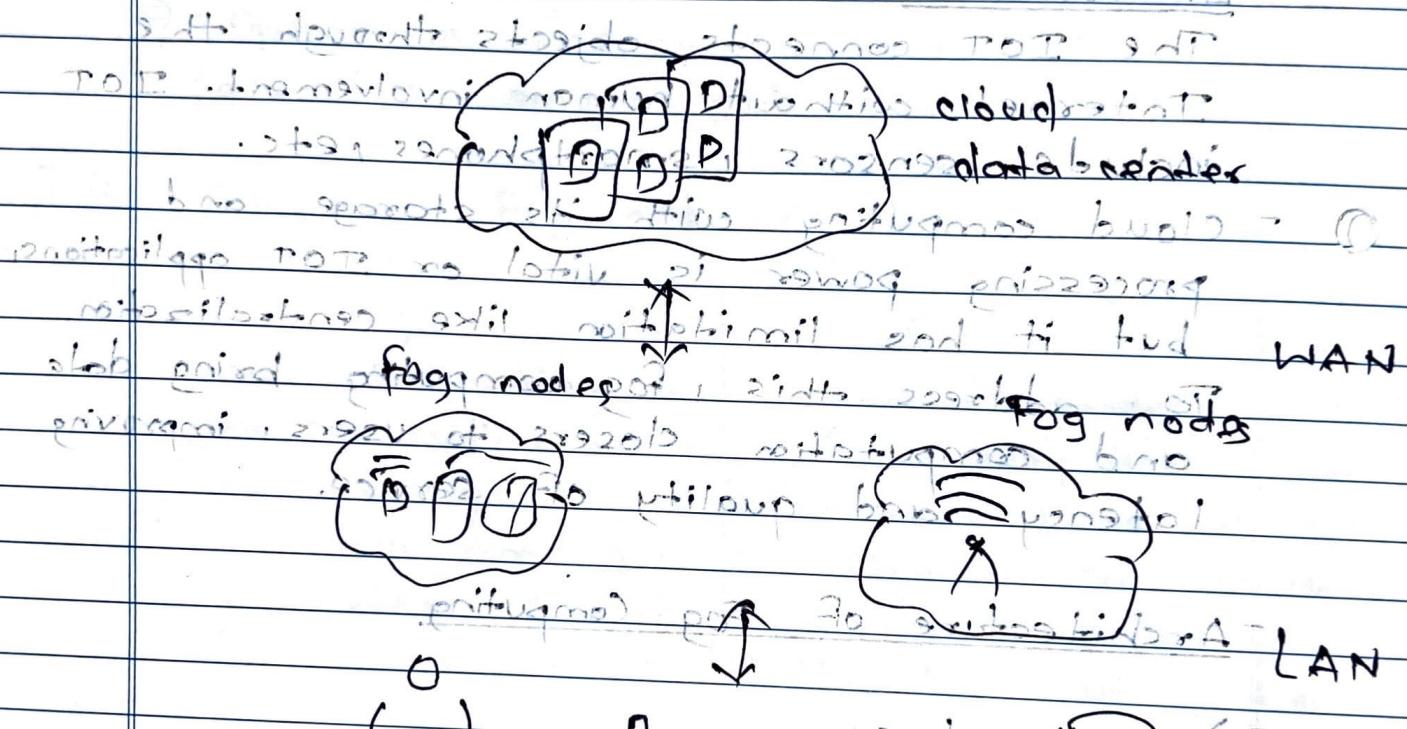
a) Device layer

⇒ 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.

- 3) 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 → noqmo hup broqra : mil

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

⇒ Google Cloud Storage (GCS) provides durable and highly available object storage.

3) Database Services bns 22222nized

a) AWS

⇒ Amazon relational database service (RDS) offers managed relational databases, like MySQL, PostgreSQL, etc. It provides managed services for MySQL, PostgreSQL, and Oracle databases.

b) Azure

⇒ Azure SQL database provides fully managed relational databases, including support for managed identities, automatic backups, and more.

c) GCP

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

4) Networking Services:

a) AWS

⇒ Amazon Virtual Private Cloud (VPC) enables users to launch AWS resources within 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 (GVP) 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 A

SJ
26/3/2024