



SHRI VILEPARLE KELAVANI MANDAL'S
SHRI BHAGUBHAI MAFATLAL POLYTECHNIC



Computer Engineering Department

A
Laboratory Manual

Cloud Application Development
(Course Code: CLD198924)

Semester: VI

Vision /Mission Statements

Institute Vision

SBM Polytechnic aspires to be the lead institute in providing need based technical education.

Institute Mission

- To provide state-of-the-art infrastructure and latest equipments for providing a stimulating learning environment.
- To prepare students to meet the dynamic needs of the industry by periodic reviewing and upgradation of curriculum through an interactive process with industry.
- To inculcate a spirit of excellence in terms of academic performance, research and innovation in faculty by providing appropriate support and incentive systems.
- To promote and support Co-Curricular, extra-curricular activities and industry interaction to make students socially sensitive and employable.

Department Vision

Create a sustainable academic environment to produce highly competent computer professionals of the future

Department Mission

1. To expose students to latest tools and technologies in computing.
2. To foster the professional development of students by providing excellence in education.
3. To Adapt rapid advancements in computing by engaging students in the lifelong learning.
4. To inculcate sound ethical, moral and social values amongst students for benefit of the society.

Program Educational objectives

After successful completion of the program students will be able to:

- 1. Identify, design and solve computing problems by applying knowledge in Computer engineering.**
- 2. Promote lifelong learning by integrating academic knowledge and practical applications.**
- 3. Depict effective team work and practical skills for holistic development**

Program Outcomes

- 1. Basic and Discipline specific knowledge: Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems**
- 2. Problem analysis: Identify and analyse well-defined engineering problems using codified standard methods**
- 3. Design/ development of solutions: Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs**
- 4. Engineering Tools, Experimentation and Testing: Apply modern engineering tools and appropriate technique to conduct standard tests and measurements**
- 5. Engineering practices for society, sustainability and environment: Apply appropriate technology in context of society, sustainability, environment and ethical practices.**
- 6. Project Management: Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities**
- 7. Life-long learning: Ability to analyse individual needs and engage in updating in the context of technological changes**

Program Specific Outcomes

PSO1: Demonstrate the fundamental knowledge in the areas of Operating system, Web Technology, Microprocessor based system and IOT by applying programming skills and developing applications.

PSO2: Administer and manage Open source, Networking, Security and Database domains to enhance student growth.

Course Outcomes

At the end of the semester student will be able to: -

1. Implement Virtualization
2. Create CLI for cloud and write application for various cloud services
3. Implement containerization.
4. Describe recent trends in cloud computing.

CO-PO Mapping

Course Outcomes (CO)	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PSO 1	PSO 2
CO 1	3	1	1	1		1	2		3
CO 2	3	1	3	2	1	2	2		3
CO 3	3	1	1	2	1		1		3
CO 4	3						1		3

INDEX

Subject: Cloud Application Development

Code: CLD198924

Course: - Information Technology

Total Contact hours for Practicals: 32 Hours

Sr. No.	Title of Experiment	Approx. Hrs required	COs
1	To implement virtual machine	2	CO1
2.	To install and configure AWS CLI	2	CO2
3.	To implement cloud networking and use AWS VPC	4	CO2
4.	Host a word press application using EC2	4	CO2
5.	Host a word press high available application using EC2 and ELB	2	CO2
6.	Host a static website using S3with Cloud Front CDN	4	CO2
7.	Use AWS Lambda to create A server less function	2	CO3
8.	Build a near real Time Event log System using Dockers Amazon SNS and SQS	4	CO3
9.	Write an application to log the data using Open Cloud platform	4	CO4
10.	Assignment on Public Clouds and Business model	2	CO2
11.	Assignment on Cloud services and component offered by Google & Azure	2	CO2
	TOTAL	32	

Experiment 1: Installing Linux on physical partition of machine having windows platform

Aim: To implement virtual machine (EC2) using AWS.

Lab Outcomes:

After undergoing this laboratory module, the student will be able to:

1. Create an AWS account
2. Implement a virtual machine (EC2) on AWS

Hardware/Software used: Client System, AWS Management Console

Theory:

Cloud Computing:

Cloud technology is the key driver of a paradigm shift from traditional IT services to next-generation computing solutions, redefining the way companies deliver software and applications, and customers use them.

The rapid proliferation of IoT (Internet of Things), 5G technologies, and edge computing, along with a rise in Big Data consumption, as well as the increased use of AI-powered, real-time data analytics, are significantly contributing to the adoption of cloud computing worldwide.

A cost-effective alternative to on-premises IT and the foundation of app-based, data-driven tech environments, cloud computing provides businesses with easy access to a variety of resources from anywhere, at any time, including databases, servers, storage, and virtual machines.

Cloud Platforms:

A Cloud platform hosts the server hardware and operating system in a web-based datacenter. The platform enables the coexistence of hardware and software, and it offers remote connectivity to compute services at scale.

Computer Engineering Department, S.B.M.Polytechnic, Mumbai.

There are several cloud platforms on the market. The services offered by these platforms range from providing complete app development frameworks to computing services, such as virtual desktops, servers, and storage.

1. AWS or Amazon Web Services,

AWS is a subsidiary of the American multinational Amazon, is one of the most well-integrated and widely adopted cloud platforms in the world.

Amazon Web Services offers a wide range of innovative cloud-based services, including Route 53, Amazon CloudWatch, Simple Email Service, CloudTrail, Relational Database Service, Simple Queue Service, AWS Key Management Service, Elastic Compute Cloud, Simple Storage Service, DynamoDB, AWS Data Transfer, Simple Notification Service, EC2, and Virtual Private Cloud.

2. Microsoft Azure

Azure was released in February 2010 by technology giant Microsoft to facilitate the development, testing, deployment, and management of services and applications through data centers owned and managed by Microsoft.

Azure integrates diverse cloud-based services, which include Windows Virtual Desktop, Microsoft Azure Stack, the Internet of Things, Developer Tools, Mixed Reality, Storage, Security, Databases, Containers, Networking, Migration, Artificial Intelligence, Machine Learning, Blockchain, and Analytics.

3. GCP (Google Cloud Platform)

GCP is one of the most popular cloud platforms that integrates an array of advanced features and solutions.

Some of the stand-out services provided by the Google Cloud Platform include the following: Firebase, G Suite, Orbitera, Apigee, Android Enterprise, Google Hardware, Chrome Enterprise, Google Identity, Google Maps Platform,

Computer Engineering Department, S.B.M.Polytechnic, Mumbai.

Serverless Computing, Storage, Networking, Databases, Containers, Migration, Compute, API Management, the Internet of Things, Multi-cloud, Machine Learning, and AI.

AWS Account:

The AWS Free Tier is automatically activated on each new AWS account. With the AWS Free Tier, you can try out some AWS services free of charge up to a specific maximum amount of usage each month.

Free Tier includes three different types of offerings

- Trials: These are short-term trial offers that start from the date that you activate a service. You pay standard rates after the trial period expires. For more information on how different services are charged.
- 12-Months free: These offers provide limited usage for 12 months after your initial sign-up date. You pay standard rates after your 12 months free usage term expires or if your application use exceeds the free tier limits.
- Always free: These offers are available to all AWS customers and don't expire at the end of your 12-month AWS Free Tier term.

Not all free usage is part of the AWS Free Tier. Some services offer free trials outside of AWS Free Tier. Other services offer some level of service for free, whether or not your account is covered under the AWS Free Tier.

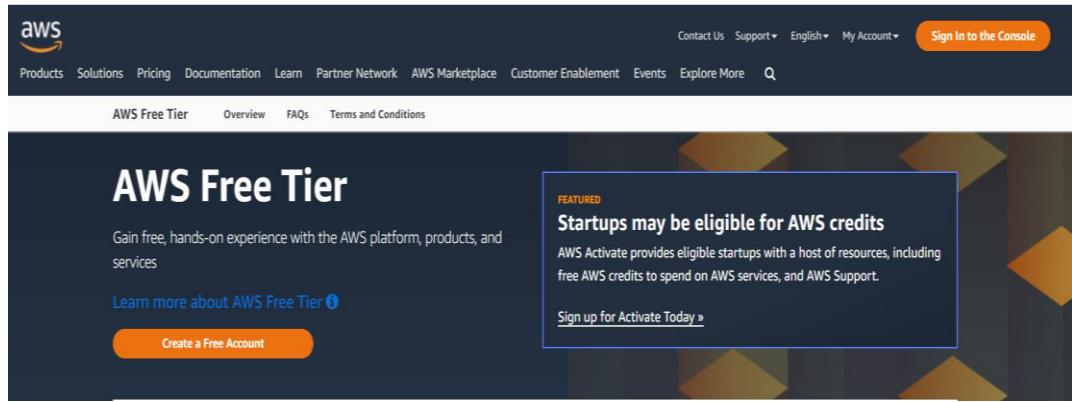
Problem Statement:

Create an AWS free tier account and deploy a virtual instance i.e. Linux and Windows.

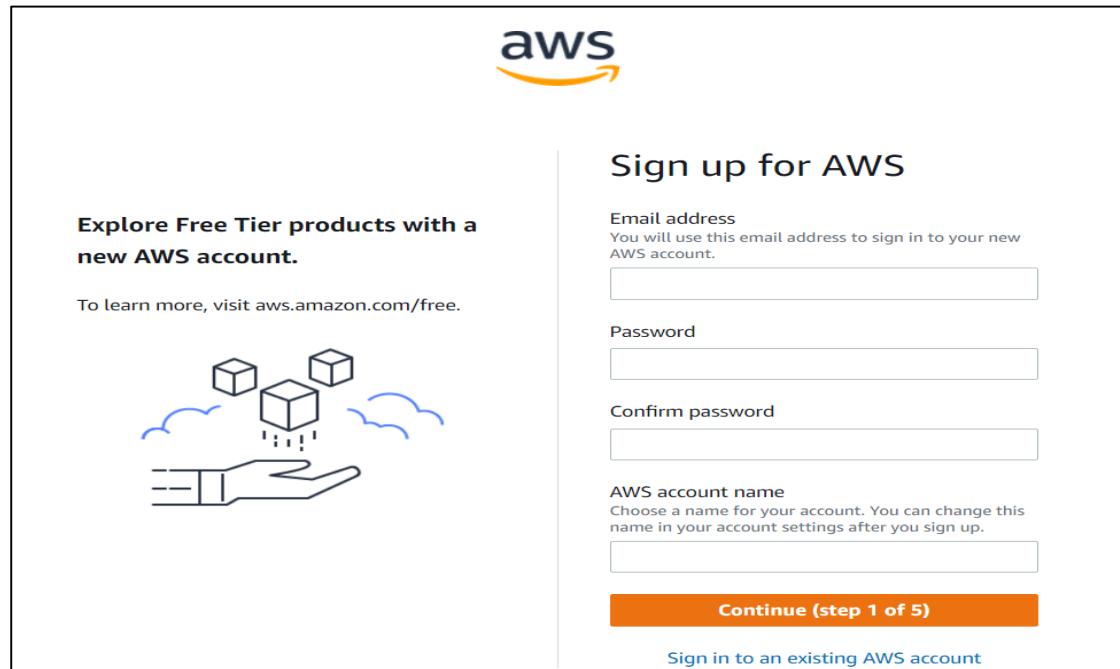
Procedure:

Steps of Creating an AWS free account:

1. First Open your web browser and navigate to AWS Free Tier Page & click on “Create Free Account” button



2. Issue the details which you want to use for login your AWS account and click on Continue



3. Contact Information

Select your AWS type (Professional/ Personal) Fill in the correct information to validate your account if you're going to create personal use then click on "Personal Account" else use "Company Account", Accepts the Terms and condition and then click on Create Account and Continue

Contact Information

All fields are required.

Please select the account type and complete the fields below with your contact details.

Account type Professional Personal

Full name

Phone number

Country/Region

Address

City

+ State / Province or region

Postal code

Check here to indicate that you have read and agree to the terms of the [AWS Customer Agreement](#)

Create Account and Continue

- Payment and PAN information: In this step, you must fill in your credit card /Debit Card info and billing address and click on Secure Submit.

Sign up for AWS

Secure verification

We will not charge you for usage below AWS Free Tier limits. We may temporarily hold up to \$1 USD (or an equivalent amount in local currency) as a pending transaction for 3-5 days to verify your identity.

Billing Information

Credit or Debit card number

AWS accepts all major credit and debit cards. To learn more about payment options, review our [FAQ](#).

Expiration date

Cardholder's name

CVV

Billing address

Use my contact address
xyz
abc lane, 12345
IN

Use a new address

Do you have a PAN?

Permanent Account Number (PAN) is a ten-digit alphanumeric number issued by the Indian Income Tax Department. This 10-digit number is printed on the front of your PAN card.

Yes
 No

You can sign on the Tax Settings Page on Billing and Cost Management Console to update your PAN information.

Verify and Continue [step 3 of 5]

You might be redirected to your bank's website to authorize the verification charge.

- it will take you to the payment gateway to validate your payment information and for your credit card/debit card verification, Amazon will charge the minimum price based on Country, For India, 2 INR.

Merchant details

Merchant Name: AMAZON INTERNET SERVICES

Date: JAN 10, 2020

Card Number: 4160 XXXX XXXX 6037

Total Charge: Rs. 2.00

Authenticate Transaction

OTP

Successfully sent the One Time Password to your Registered Mobile Number 86***9***29.

Enter OTP

.....

Resend OTP

CANCEL SUBMIT

Note- Please ensure that your latest mobile number/ email id is updated in the Bank records. Visit nearest Branch or call Customer Care for the same.

This page will automatically time out after 02:27 seconds

6. Phone verification: Here you will be taken to an identity verification page that will already have your phone number, so you just have to select either "Text message or Voice call" Provide a valid phone number, Solve the captcha and then click on Send SMS or Call Me Now(depending upon your selection).

Confirm your identity

Before you can use your AWS account, you must verify your phone number. When you continue, the AWS automated system will contact you with a verification code.

How should we send you the verification code?

Text message (SMS) Voice call

Country or region code

India (+91)

Cell Phone Number

Security check

Y5785C

Type the characters as shown above

Send SMS

7. After clicking on Send SMS or Call me Now, you will immediately receive a call or SMS from Amazon, for verification code, Enter your code then click on Verify Code.

Enter verification code

Enter the 4-digit verification code that you received on your phone.

8393

Verify Code

Having trouble? Sometimes it takes up to 10 minutes to receive a verification code. If it's been longer than that, [return to the previous page](#) and enter your number again.

8. Support plan: AWS support offers a selection of plans to meet your business needs. Select your suitable plan then click continue

Sign up for AWS

Select a support plan

Choose a support plan for your business or personal account. Compare plans and pricing examples . You can change your plan anytime in the AWS Management Console.

Basic support - Free

- Recommended for new users just getting started with AWS
- 24x7 self-service access to AWS resources
- For account and billing issues only
- Access to Personal Health Dashboard & Trusted Advisor



Developer support - From \$29/month

- Recommended for developers experimenting with AWS
- Email access to AWS Support during business hours
- 12 (business)-hour response times



Business support - From \$100/month

- Recommended for running production workloads on AWS
- 24x7 tech support via email, phone, and chat
- 1-hour response times
- Full set of Trusted Advisor best-practice recommendations



9. Registration Confirmation page



Congratulations
Thank you for signing up for AWS.
We are activating your account, which should only take a few minutes. You will receive an email when this is complete.
[Go to the AWS Management Console](#)
Sign up for another account or contact sales.

Steps of Creating an EC2 instance:

1. Login into AWS account:

The screenshot shows the AWS Sign In interface. On the left, under 'Root user', the 'Root user' radio button is selected. Below it, there's a field for 'Root user email address' containing 'username@example.com'. A large blue 'Next' button is centered. To the right, the Amazon ElastiCache for Redis landing page is displayed with the heading 'Amazon ElastiCache for Redis' and the subtext 'Boost application performance with microsecond latency'. It features a diagram of three interconnected hexagons representing a Redis cluster.

2. Select the EC2 Service:

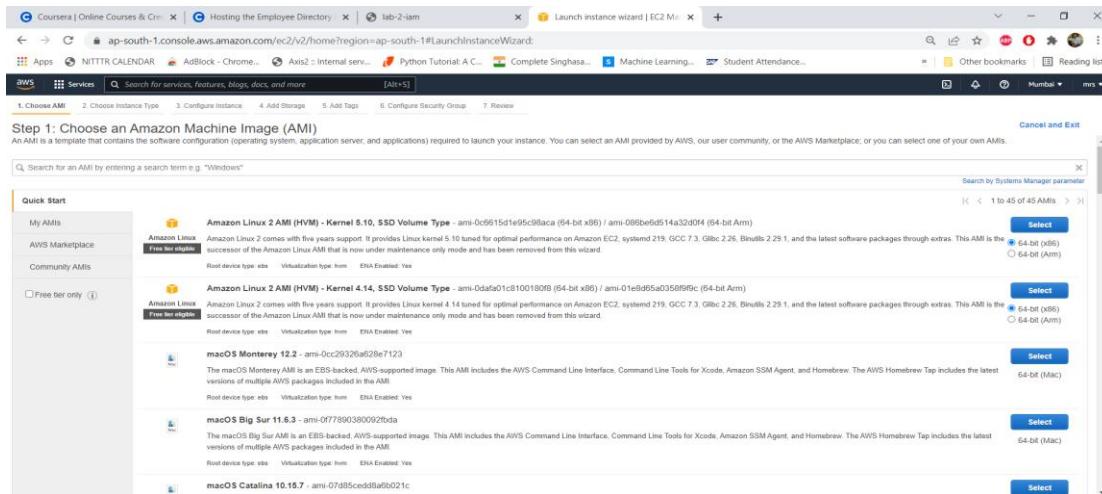
The screenshot shows the AWS Services dashboard. The 'All services' section is expanded, displaying various AWS services categorized into Compute, Management & Governance, Security, Identity, & Compliance, and more. To the right, there are promotional cards for the New AWS Console Home, Stay connected to your AWS resources on-the-go, and Explore AWS.

3. Click on “EC2 Dashboard” and Select “Create Instance”

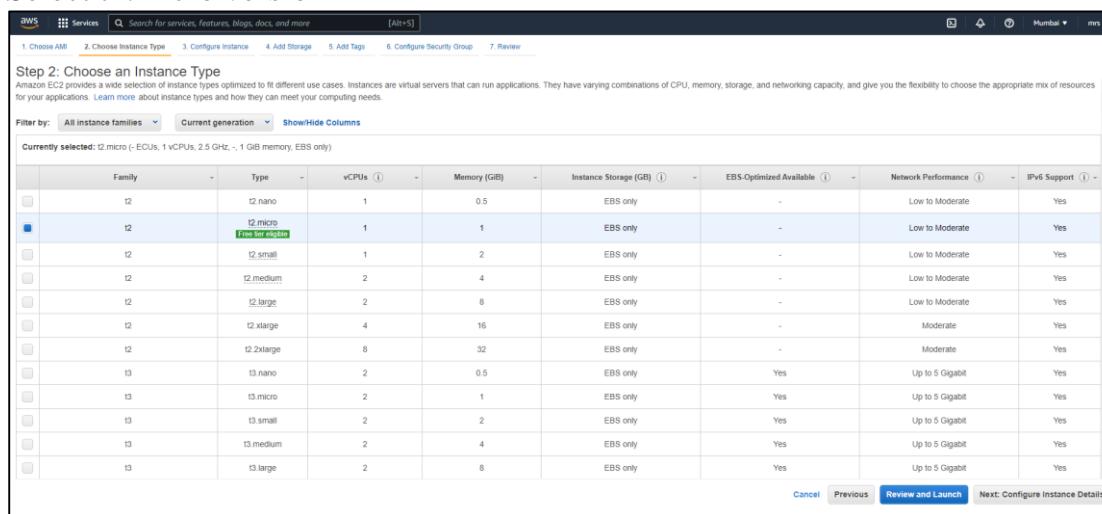
The screenshot shows the EC2 Dashboard. The left sidebar has 'EC2 Dashboard' selected. The main area displays 'Resources' and 'Account attributes'. Under 'Resources', there are tables for Instances (running), Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups, Snapshots, and Volumes. A callout box points to the 'Launch instance' button. The 'Account attributes' section includes 'Supported platforms' (VPC), 'Default VPC' (vpc-070c8b16e558c0bbb), and other settings like EBS encryption and Zones. The 'Explore AWS' section promotes AWS Graviton2 and Get Up to 40% Better Price Performance.

Cloud Application Development CLD198924

4. Select Amazon Linux Instance



5. Select t2.micro version



6. Click on “Add Storage”

Cloud Application Development CLD198924

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1

Purchasing option: Request Spot Instances

Network: vpc-070cb16e558c0bbb (default)

Subnet: No preference (default subnet in any Availability Zone)

Auto-assign Public IP: Use subnet setting (Enable)

Hostname type: Use subnet setting (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

Placement group: Add instance to placement group

Capacity Reservation: Open

Domain join directory: No directory

IAM role: None

Shutdown behavior: Stop Enable hibernation as an additional stop behavior

Enable termination protection: Protect against accidental termination

Monitoring: Enable CloudWatch detailed monitoring

7. Click on “Review & Launch”

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-0d2b1848f14ffd3cd	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

▼ Shared file systems

You currently don't have any file systems on this instance. Select "Add file system" button below to add a file system.

8. Click on “Launch”

Cloud Application Development CLD198924

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-0c6615d1e05c98aca

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

Security Groups

Security group name	Description
launch-wizard-1	launch-wizard-1 created 2022-02-12T20:54:23.512+05:30

Instance Details

Type: SSH, Protocol: TCP, Port Range: 22, Source: 0.0.0.0/0

Launch

9. Select the Key Value Pair

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair

Select a key pair

No key pairs found

No key pairs found
You don't have any key pairs. Please create a new key pair by selecting the [Create a new key pair](#) option above to continue.

Cancel **Launch Instances**

10. See the instance running state

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP
Edit Name	i-0e216acbacb161960	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-3-109-182-78.ap-s...	3.109.182.78	-

Instance: i-0e216acbacb161960

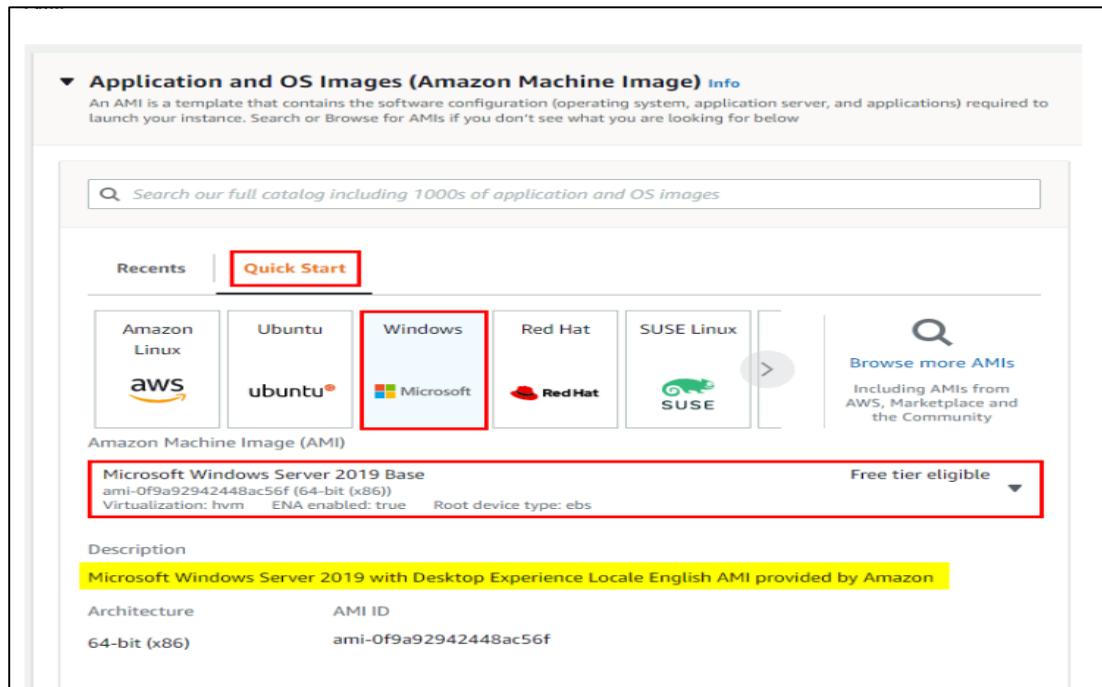
Details		Security	Networking	Storage	Status checks	Monitoring	Tags
Instance ID	i-0e216acbacb161960	Public IPv4 address	3.109.182.78 open address		Private IPv4 addresses	172.31.43.143	
IPv6 address	-	Instance state	Running		Public IPv4 DNS	ec2-3-109-182-78.ap-south-1.compute.amazonaws.com open address	
Hostname type	IP name: ip-172-31-43-143.ap-south-1.compute.internal	Private IP DNS name (IPv4 only)	ip-172-31-43-143.ap-south-1.compute.internal		Answer private resource DNS name	IPv4 (A)	

11. Click on “Connect”

```
Last login: Sat Feb 12 15:43:58 2022 from ec2-13-233-177-0.ap-south-1.compute.amazonaws.com
[ec2-user@ip-172-31-43-143 ~]$ ls
[ec2-user@ip-172-31-43-143 ~]$ echo "Hi"
Hi
[ec2-user@ip-172-31-43-143 ~]$ mkdir mrs
[ec2-user@ip-172-31-43-143 ~]$ ls
mrs
[ec2-user@ip-172-31-43-143 ~]$
```

Steps of Creating an EC2 instance (Windows Machine):

1. Select Windows Instance



2. Keep everything default and click on “Launch Instance”

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
MyEC2	i-0e216acbcb161960	Stopped	t2.micro	-	No alarms	ap-south-1a
-	i-0ad1622c9548b5553	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b

Instance: i-0ad1622c9548b5553

Details	Security	Networking	Storage	Status checks	Monitoring	Tags			
Instance summary Info <table border="1"> <tr> <td>Instance ID i-0ad1622c9548b5553</td> <td>Public IPv4 address 3.110.124.221 open address</td> <td>Private IPv4 addresses 172.31.3.184</td> </tr> </table>							Instance ID i-0ad1622c9548b5553	Public IPv4 address 3.110.124.221 open address	Private IPv4 addresses 172.31.3.184
Instance ID i-0ad1622c9548b5553	Public IPv4 address 3.110.124.221 open address	Private IPv4 addresses 172.31.3.184							

3. Click on “Connect” and click on “Download remote desktop file” from “RDP Client” tab

Connect to instance [Info](#)

Connect to your instance i-0ad1622c9548b5553 using any of these options

Session Manager [RDP client](#) [EC2 Serial Console](#)

You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

[Download remote desktop file](#)

When prompted, connect to your instance using the following details:

Public DNS	User name
------------	-----------

4. To get windows password, locate your .pem file

Get Windows password [Info](#)

Retrieve and decrypt the initial Windows administrator password for this instance.

To decrypt the password, you will need your key pair for this instance.

 Key pair associated with this instance
mrskey

Browse to your key pair:

 [Browse](#)

mrskey.pem
1.704KB

Or copy and paste the contents of the key pair below:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAr48uiPkTAriHn1ajlq9CmZ3c7dnyfrtjUkDrSyyHamXrMKk
e02bOP00ujSaWimSgyx2+9DFZ9L3p/UVZ2MRRdTn5ga2DhtDyEUZ5Pp2W7v81r// 
3CbODN0+ipLu2tHYkyVBvROrlzOkCX6tC+hnkdmPOVE3nRf0HXcfTeUljhjEHQo
i+xufHovb8KB8ak4Hi+NgV3cOkgggRpdlZlP2tC+BeQCwLGi91PdMdihSUJvAO+fV
oL2jEST7G+5pQRJPyUKXQyPYg5t3HRtsn6Uy7DpVUW38GNhWokpCY803mnFR1hAh
e9HyKsEOUHUzGPjQE7SERCFQoXL5UOmDounLjwiDAQABAoiBAEexNFszmkLTlQsc
z7OwauBzzBz4OEbIVPgMaBbxGK+qUQk410iPMaM9K3+igaHuNf8o0IVQxY3/fC1
```

[Cancel](#) [Decrypt password](#)

 **Password Decryption Successful**
The password for instance i-0ad1622c9548b5553 was successfully decrypted.

EC2 > Instances > i-0ad1622c9548b5553 > Connect to instance

Connect to instance [Info](#)

Connect to your instance i-0ad1622c9548b5553 using any of these options

Session Manager [RDP client](#) [EC2 Serial Console](#)

You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

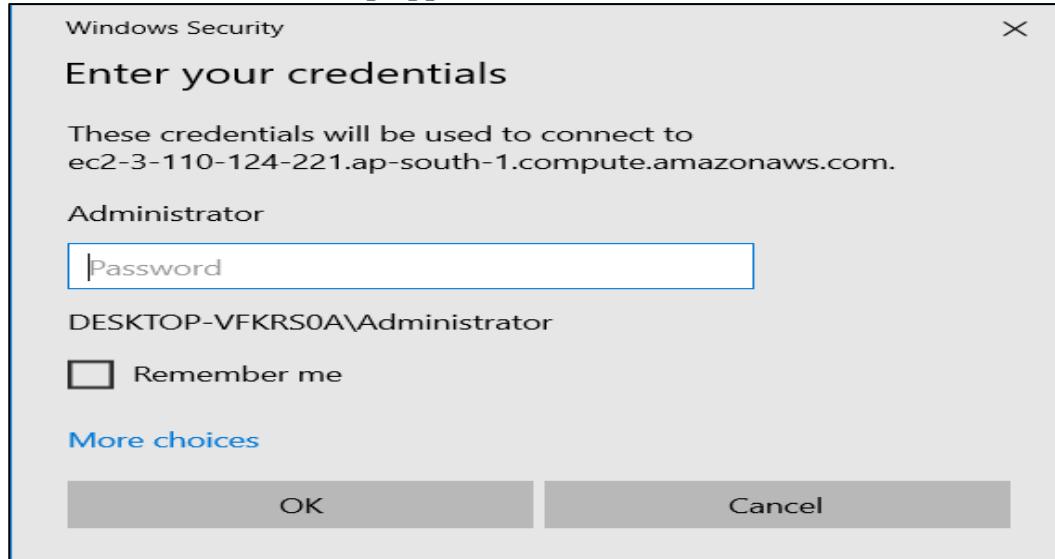
[Download remote desktop file](#)

When prompted, connect to your instance using the following details:

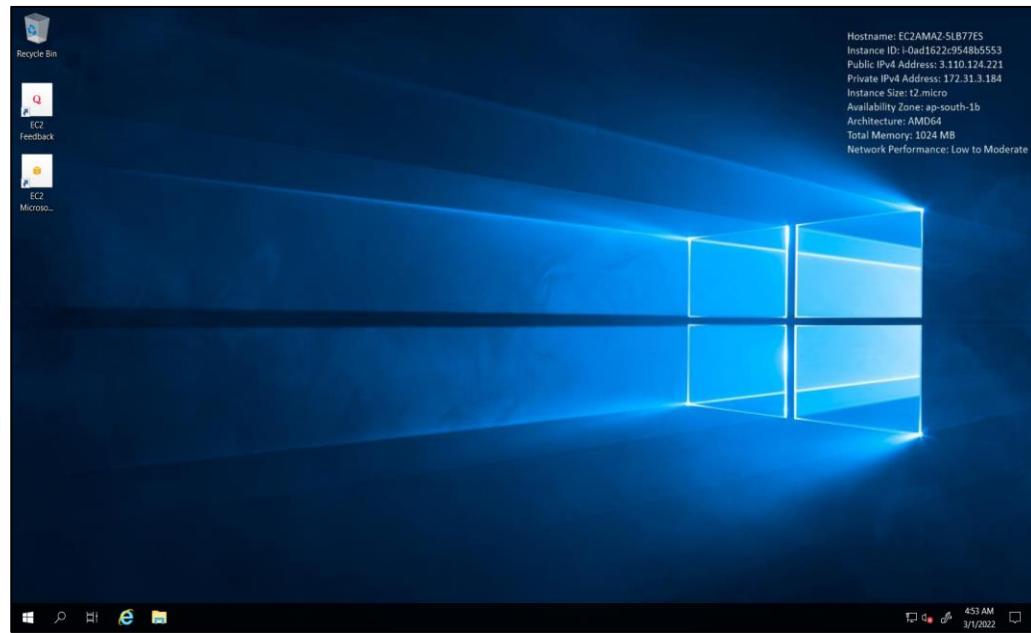
Public DNS	User name
 ec2-3-110-124-221.ap-south-1.compute.amazonaws.com	 Administrator
Password	
 Ogab.DITS(=RXIzs2ndZDwrsiu2BlcPs	

If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.

5. Run the remote desktop app



6. Give Permission to connect with remote machine



Output:

Few screen shots of AWS free tier account creation, EC2 Linux instance creation and EC2 Windows server instance creation

Conclusion:

Computer Engineering Department, S.B.M.Polytechnic, Mumbai.
Page 20 | 93

Questions:

1. Define cloud computing.
2. Describe various cloud platforms in brief.
3. State any 3 features/rules of AWS free tier account
4. Enlist the steps of creating AWS EC2 Linux instance.
5. Enlist the steps of creating AWS EC2 Windows instance.
6. Describe the process of creating a physical partition on windows OS.

EXPERIMENT NO: 02 To install and configure AWS CLI.

Aim: To install AWS CLI and configure AWS using aws commands.

Lab Outcomes:

After undergoing this laboratory module, the student will be able to:

1. Install AWS CLI
2. Configure AWS using aws commands

Hardware/Software used: Client System ,AWS CLI, AWS Management Console, Putty

Theory:

AWS CLI(Command Line Interface) :

AWS provides two ways of infrastructure configurations:

AWS web console:

It is a graphical method to connect to various AWS resources, their configuration, modification, etc. It is simple to use and does not require knowledge of scripting

AWS Command Line Interface:

It is an open-source unified tool to communicate with AWS resources and services using a command-line interface. We can manage all AWS services and control their behavior using this tool. We can also use CLI to automate AWS infrastructure resource and service management through scripts.

We can install it on Windows, Linux, macOS and Docker containers

The script provides us with the flexibility to manage multiple AWS resources, infrastructures effectively. We can use the script to deploy multiple resources without the need to go through a complete configuration wizard each time.

Example: If we want to download some data from S3, we have to select appropriate bucket on cloud Management Console which is tedious task. If we know the right command of CLI it is very easy task.

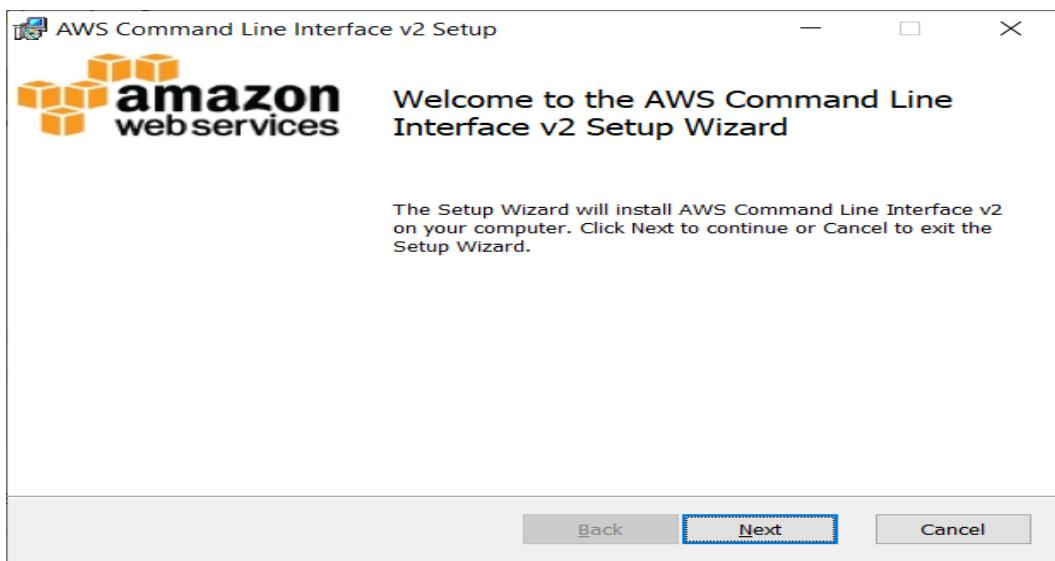
Procedure:

Computer Engineering Department, S.B.M.Polytechnic, Mumbai.

1. Download AWS CLI Version 2 from <https://awscli.amazonaws.com/AWSCLIV2.msi> (Windows) OR from AWS Management Console as shown below:

The screenshot shows the AWS CLI v2 landing page. At the top, there's a navigation bar with links for Products, Solutions, Pricing, Documentation, Learn, Partner Network, AWS Marketplace, Customer Enablement, Events, Explore More, and a search bar. On the right, there's a "Sign In to the Console" button. Below the navigation, there's a sidebar with "RESOURCES" and a link to "AWS Command Line Interface". Under "RELATED LINKS", there are links for Documentation, Tools, and Release Notes. A "Get Started with AWS for Free" section includes a "Create Free Account" button. The main content area is titled "AWS Command Line Interface" and contains text about the unified tool for managing AWS services. It highlights new features like improved installers and AWS Single Sign-On (SSO). Below the text are four icons: a large number 1, a document with lines, a document with a gear, and a cube. At the bottom, there are links for "Getting Started", "CLI Reference", "CLI Help", "Projects", and "Community". To the right, there are sections for "Windows", "MacOS", "Linux", and "Amazon Linux", each with a brief description of the download process.

2. Install the AWSCLIV2.msi on the System



3. Execute “aws –version” command to verify AWS CLI installation on CMD

```
C:\Users\manis>aws --version
aws-cli/2.4.21 Python/3.8.8 Windows/10 exe/AMD64 prompt/off

C:\Users\manis>
```

AWS CLI Commands:

- A. **Help command:** used to check its brief description, synopsis, and options to use.

```
C:\ Command Prompt - aws help
C:\Users\manis>aws help

aws
^^^

Description
*****
services.

Synopsis
*****
aws [options] <command> <subcommand> [parameters]
are shown in square brackets.

Options
*****
"--debug" (boolean)
Turn on debug logging.
"--endpoint-url" (string)
```

Note: We have to press Enter key to get the subsequent help statements

- ## B. Security credentials for AWS account

For the AWS CLI configuration, we require the following arguments:

Access Key and Secret Access Key

To connect with a user (root or IAM), we require login credentials. In AWS, we call it as access key (user) and secret access key (password).

Generating Access Key & Secret Key: Click on “Access key”

Cloud Application Development CLD198924

The screenshot shows the AWS Identity and Access Management (IAM) service in the AWS Management Console. The left sidebar lists various IAM management options like Dashboard, Access management, and Access reports. The main content area is titled 'Your Security Credentials' and provides instructions for managing account credentials. It highlights the 'Password' section, which contains a note about using an email and password for secure AWS access. Below this are sections for Multi-factor authentication (MFA), Access keys (access key ID and secret access key), CloudFront key pairs, X.509 certificate, and Account identifiers. A 'Sign out' button is located in the bottom right corner of the main content area.

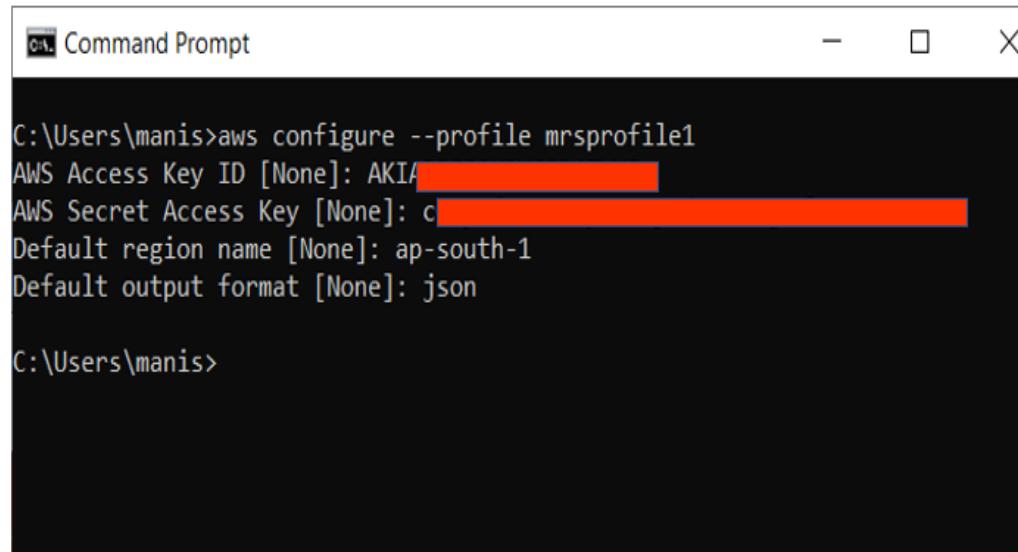
Click on “Create New Access Key”

This screenshot shows the 'Create New Access Key' dialog box. The left sidebar of the main IAM page is visible. The dialog box has a header 'Create Access Key'. Inside, there is a green checkmark icon followed by the message: 'Your access key (access key ID and secret access key) has been created successfully.' Below this, it says: 'Download your key file now, which contains your new access key ID and secret access key. If you do not download the key file now, you will not be able to retrieve your secret access key again.' There is also a note: 'Root user access keys provide unrestricted access to your entire AWS account. If you need long-term access keys, we recommend creating a new IAM user with limited permissions and generating access keys for that user instead.' At the bottom of the dialog are two buttons: 'Download Key File' and 'Close'.

Download the key:

This screenshot shows the confirmation message from the 'Create Access Key' dialog. It displays a green checkmark icon and the text: 'Your access key (access key ID and secret access key) has been created successfully.' It also includes the instruction: 'Download your key file now, which contains your new access key ID and secret access key. If you do not download the key file now, you will not be able to retrieve your secret access key again.' Below this, there is a note: 'Root user access keys provide unrestricted access to your entire AWS account. If you need long-term access keys, we recommend creating a new IAM user with limited permissions and generating access keys for that user instead.' At the bottom are 'Download Key File' and 'Close' buttons.

C. Configure AWS CLI using the CONFIGURE command



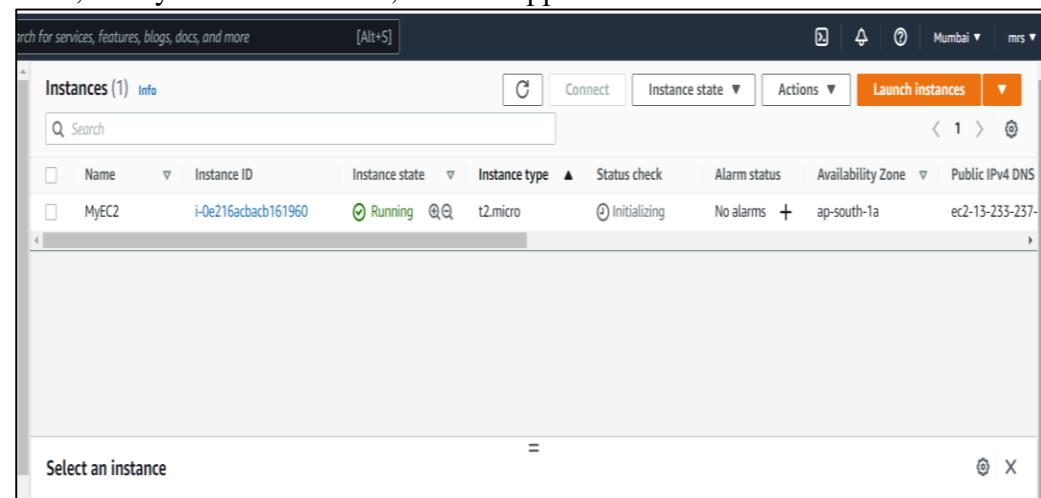
```
C:\Users\manis>aws configure --profile mrsprofile1
AWS Access Key ID [None]: AKIA[REDACTED]
AWS Secret Access Key [None]: c[REDACTED]
Default region name [None]: ap-south-1
Default output format [None]: json

C:\Users\manis>
```

We can see the configuration file (.json) as below:

```
1  [default]
2    region = Mumbai
3    output = json
4  [profile mrsprofile1]
5    region = ap-south-1
6    output = json
```

Now, start your EC2 instance, if it is stopped...



D. Describing EC2 instances using “aws aws ec2 describe-instances” command

```
C:\Users\manis>aws ec2 describe-instances
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-0c6615d1e95c98aca",
                    "InstanceId": "i-0e216acbcb161960",
                    "InstanceType": "t2.micro",
                    "KeyName": "mrskey",
                    "LaunchTime": "2022-02-12T15:30:23+00:00",
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "Placement": {
                        "AvailabilityZone": "ap-south-1a",
                        "GroupName": "",
                        "Tenancy": "default"
                    },
                    "PrivateDnsName": "ip-172-31-43-143.ap-south-1.compute.internal",
                    "PrivateIpAddress": "172.31.43.143",
                    "ProductCodes": [],
                    "PublicDnsName": "",
                    "State": {
                        "Code": 80,
                        "Name": "stopped"
                    },
                    "StateTransitionReason": "User initiated",
                    "SubnetId": "subnet-00e28c00be82029f2",
                    "VpcId": "vpc-070c8b16e558c0bbb",
                }
            ]
        }
    ]
}
```

aws ec2 describe-regions

```
C:\Users\manis>aws ec2 describe-regions
{
    "Regions": [
        {
            "Endpoint": "ec2.eu-north-1.amazonaws.com",
            "RegionName": "eu-north-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-south-1.amazonaws.com",
            "RegionName": "ap-south-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-west-3.amazonaws.com",
            "RegionName": "eu-west-3",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-west-2.amazonaws.com",
            "RegionName": "eu-west-2",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-west-1.amazonaws.com",
            "RegionName": "eu-west-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.ap-northeast-3.amazonaws.com",
            "RegionName": "ap-northeast-3",
            "OptInStatus": "opt-in-not-required"
        }
    ]
}
```

aws ec2 describe-regions --output table

DescribeRegions			
Endpoint	OptInStatus	RegionName	Regions
eu-north-1.amazonaws.com	opt-in-not-required	eu-north-1	eu-north-1
ap-south-1.amazonaws.com	opt-in-not-required	ap-south-1	ap-south-1
eu-west-3.amazonaws.com	opt-in-not-required	eu-west-3	eu-west-3
eu-west-2.amazonaws.com	opt-in-not-required	eu-west-2	eu-west-2
eu-west-1.amazonaws.com	opt-in-not-required	eu-west-1	eu-west-1

Note: We can run commands on EC2 Linux instance using SSH or Putty.

(Test these both the approaches practically in the lab session)

Output:

Few Screen shots of AWS CLI installation & configuration commands

Conclusion:

Questions:

- State different ways with which a AWS user can configure his instances/services
- Compare AWS Management Console with AWS CLI.
- State and describe the AWS CLI command used for configuring the region of an EC2 instance
- Enlist different output formats of “describe-regions” command.
- Discuss different ways to connect with EC2 Linux instance remotely on terminal mode.

EXPERIMENT NO: 03 To implement cloud networking using AWS VPC

Aim: To implement cloud networking using AWS VPC (Virtual Private Cloud)

Lab Outcomes:

After undergoing this laboratory module, the student will be able to:

1. Implement Virtual Private Cloud on AWS

Theory:

A VPC is a logically isolated virtual network, spanning an entire AWS Region, where your EC2 instances are launched. It gives all the benefits of the traditional network that you have for your own data center. Resources and applications are accessed through IPv4 or IPv6 in your AWS VPC. It provides scalable infrastructure in the AWS environment with complete control over your virtual network.

There are six core components which are fundamental to a VPC and will be created by a user or by AWS as part of a default VPC :

- VPC CIDR Block
- Subnet
- Gateways
- Route Table
- Network Access Control Lists (ACLs)
- Security Group

Note: Every AWS account created after Dec. 4 2013 supports VPCs and these accounts are assigned a default VPC in every AWS Region. These default VPCs are designed to make it easy for AWS users to set up networking for their EC2 instances

CIDR blocks:

You should choose CIDR blocks carefully; Amazon VPC can contain anywhere from 16 to 65,536 IP addresses. You can select your CIDR block according to the number of instances needed.

Subnet: A subnet is a **range of IP addresses in your VPC**. You can attach AWS resources, such as EC2 instances and RDS DB instances, to subnets. You can create subnets to group instances together according to your security and operational needs.

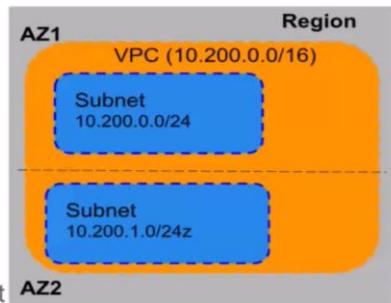
Public Subnet

- Has route for Internet
- Instances with Public IP can communicate to internet
- Ex: NAT, Web servers, Load balancer

Private Subnet

- No route to Internet
- Instances receives private IPs
- Typically uses NAT for instances to have internet access
- Ex: Database, App server

1 Subnet = 1 AZ



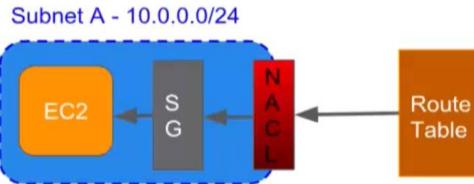
Note: You can not use 5 IP addresses
in given subnet
www.kvrik.com

Gateways:

An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between your VPC and the internet. It supports IPv4 and IPv6 traffic. It does not cause availability risks or bandwidth constraints on your network traffic.

Routing Table: The route table contains existing routes with targets other than a network interface, Gateway Load Balancer endpoint, or the default local route. The route table contains existing routes to CIDR blocks outside of the ranges in your VPC. Route propagation is enabled for the route table.

- Contains rules to route the traffic in/out of Subnets/VPC
- Main route table at VPC level
- Custom route table at Subnet level
- Each route table contains default immutable local route for VPC
- If no custom route table is defined then new subnets are associated with Main route table
- We can modify main route table



Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-id

Network Access Control Lists (ACLs): A network access control list (NACL) is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC.

- Stateless firewall - 2nd level of network security
- Works at Subnet level - applied to all instances
- Stateless
- Contains both Allow and Deny rules
- Rules are evaluated in the order of rule number
- Default NACL allows all inbound and outbound traffic



#Rule	Type	Protocol	Port	Source	Allow/Deny
100	HTTP	TCP	80	0.0.0.0/0	ALLOW
101	HTTPS	TCP	443	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	180.151.138.43/32	DENY

Security Groups: A security group controls the traffic that is allowed to reach and leave the resources that it is associated with. For example, after you associate a security group with an EC2 instance, it controls the inbound and outbound traffic for the instance. When you create a VPC, it comes with a default security group.

- Virtual Firewall - First level of defence
- Stateful - No need to explicitly allow return traffic
- Works at EC2 and RDS level
- Default - **allows all outbound traffic. No inbound traffic.**
- Can attach upto 5 security groups to single EC2 instance with 100 rules (in/outbound) per SG
- Can only specify **ALLOW** rules.
- Deny rules can not be specified
- Changes to rules are in effect immediately
- When target is Security group, allow inbound connection to all instances associated with target security group

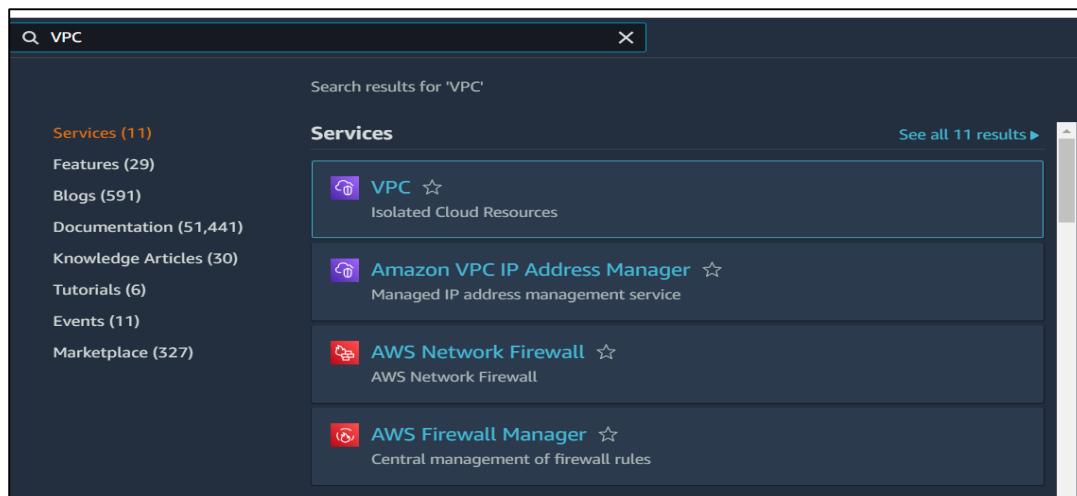


Security group inbound rules			
Type	Protocol	Port	Source
HTTP	TCP	80	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0
SSH	TCP	22	180.151.138.43/32

[www.kvriks](http://www.kvriks.com)

Procedure:

1. **Create a new VPC** : Search “VPC” inside Service search bar and click on VPC



Now click on “Your VPCs

Cloud Application Development CLD198924

The screenshot shows the AWS VPC Management Console. On the left, there's a sidebar with 'New VPC Experience' and 'VPC Dashboard'. Below that, under 'VIRTUAL PRIVATE CLOUD', 'Your VPCs' is highlighted with a red circle. Other options like 'Subnets' and 'Route Tables' are also listed. In the main area, there's a 'Launch VPC Wizard' button and a 'Resources by Region' section. A note says 'Note: Your Instances will launch in the Asia Pacific region.' To the right, there's a 'Service Health' section showing 'Amazon EC2 - Asia Pacific' with 'Service is operating normally' and a 'View complete service health details' link. At the bottom, there's a 'Settings' section with 'Zones' and 'Console Experiments'.

Click on “Create VPC”

This screenshot shows the same AWS VPC Management Console interface, but now it displays a list of 'Your VPCs'. There is one entry: 'VPC ID: vpc-070c8b16e558c0bbb', which is highlighted with a red circle. The rest of the interface remains the same, including the sidebar and the 'Create VPC' button.

VPC Wizard

The screenshot shows the 'Create VPC' wizard. The first step, 'VPC settings', is displayed. It has several sections: 'Resources to create' (radio buttons for 'VPC only' and 'VPC, subnets, etc.', with 'VPC only' selected), 'Name tag - optional' (text input field containing 'mrs-vpc'), 'IPv4 CIDR block' (radio buttons for 'IPv4 CIDR manual input' and 'IPAM-allocated IPv4 CIDR block', with 'IPv4 CIDR manual input' selected), 'IPv4 CIDR' (text input field containing '10.200.0.0/16'), 'IPv6 CIDR block' (radio buttons for 'No IPv6 CIDR block', 'IPAM-allocated IPv6 CIDR block', 'Amazon-provided IPv6 CIDR block', and 'IPv6 CIDR owned by me', with 'No IPv6 CIDR block' selected), and 'Tenancy' (dropdown menu set to 'Default').

Now, click on “Create VPC” button

Cloud Application Development CLD198924

S | Services | Search for services, features, blogs, docs, and more [Alt+S]

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create Info
Create only the VPC resource or create VPC, subnets, etc.

VPC only VPC, subnets, etc.

Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.

mrs-vpc

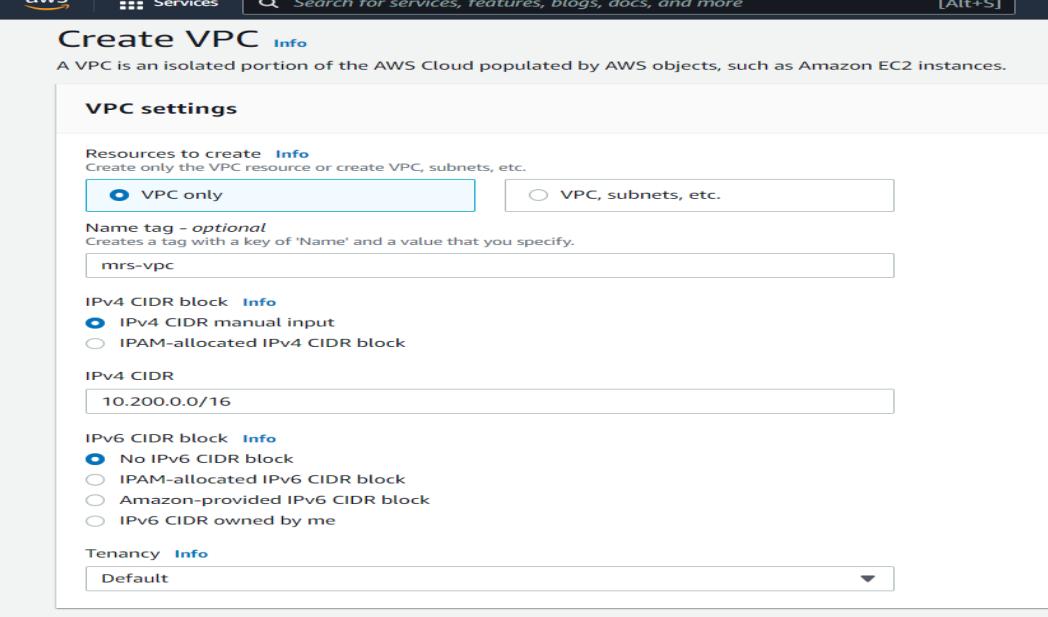
IPv4 CIDR block Info
 IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.200.0.0/16

IPv6 CIDR block Info
 No IPv6 CIDR block IPAM-allocated IPv6 CIDR block Amazon-provided IPv6 CIDR block IPv6 CIDR owned by me

Tenancy Info
Default

Feedback



VPC created successfully

AWS VPC on Udemy course by c | AWS VPC and Networking in dev... | Instances | EC2 Management Con... | VPC Management Console | ap-south-1.console.aws.amazon.com/vpc/home?region=ap-south-1#VpcDetails:VpcId=vpc-04b44ba4bda462df8

NITTR CALENDAR AdBlock - Chrome... Axis2 : Internal serv... Python Tutorial: A... Complete Singhasa... Machine Learning... Student Attendance... Maharashtra State... https://vincentarelb... Other bookmarks

w Mumbai mrs

New VPC Experience Tell us what you think

You successfully created vpc-04b44ba4bda462df8 / mrs-vpc

VPC Dashboard EC2 Global View New

Filter by VPC: Select a VPC

VIRTUAL PRIVATE CLOUD Your VPCs

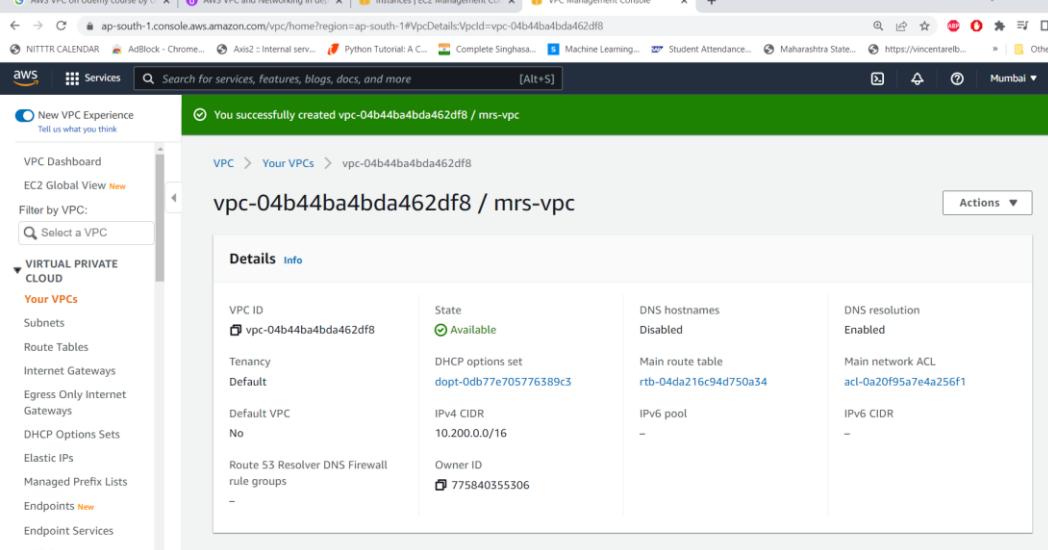
Subnets Route Tables Internet Gateways Egress Only Internet Gateways DHCP Options Sets Elastic IPs Managed Prefix Lists Endpoints New Endpoint Services NAT Gateways

vpc-04b44ba4bda462df8 / mrs-vpc Actions

Details Info

VPC ID	State	DNS hostnames	DNS resolution
vpc-04b44ba4bda462df8	Available	Disabled	Enabled
Tenancy	DHCP options set	Main route table	Main network ACL
Default	dopt-0db77e705776389c3	rtb-04da216c94d750a34	acl-0a20f95a7e4a256f1
Default VPC	IPv4 CIDR	IPv6 pool	IPv6 CIDR
No	10.200.0.0/16	-	-
Route 53 Resolver DNS Firewall rule groups	Owner ID		
-	775840355306		

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences



2. **Creating INTERNET GATEWAY:** Click on “Internet Gateways” Now click on “Create Internet gateway”

Name	Internet gateway ID	State	VPC ID	Owner
-	igw-05ee9399485d124e	Attached	vpc-070c8b16e558c0bbb	775840355306

Internet Gateway Wizard

Create internet gateway

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="mrs-igw"/> Remove

Add new tag
You can add 49 more tags.

Create Internet gateway

Gateway Created:

The following internet gateway was created: igw-03e3706a2331caf03 - mrs-igw. You can now attach to a VPC to enable the VPC to communicate with the internet.

igw-03e3706a2331caf03 / mrs-igw

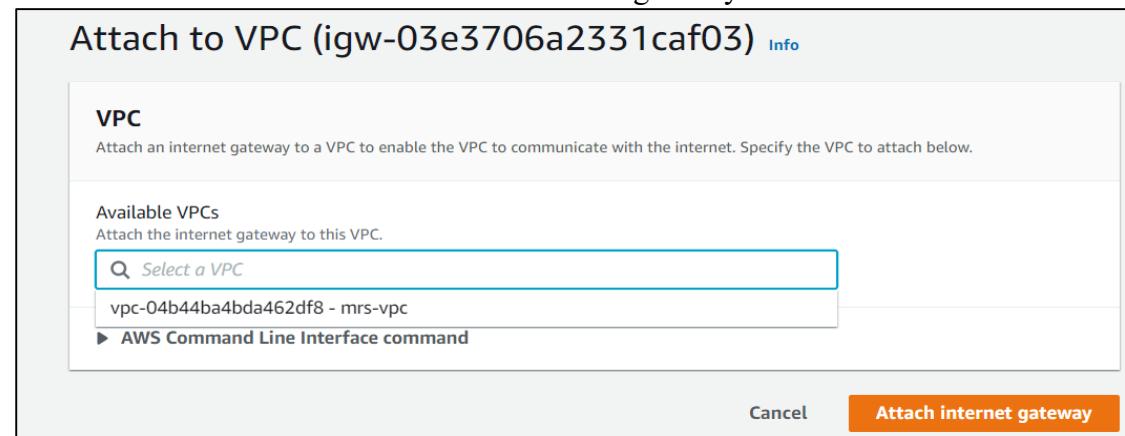
Details

Internet gateway ID <input type="text" value="igw-03e3706a2331caf03"/>	State <input checked="" type="radio" value="Attached"/> Attached	VPC ID <input type="text" value="-"/>	Owner <input type="text" value="775840355306"/>
---	---	--	--

Tags

Key	Value
Name	mrs-igw

- Select created VPC & Click on “Attach internet gateway”



Gateway successfully attached with VPC

- Creating SUBNETS:** Select Subnets and click on “Create subnet”

Create subnet Wizard

VPC > Subnets > Create subnet

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.
vpc-04b44ba4bda462df8 (mrs-vpc)

Associated VPC CIDRs
IPv4 CIDRs
10.200.0.0/16

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
mrs-public

The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
Asia Pacific (Mumbai) / ap-south-1a

IPv4 CIDR block Info
10.200.0.0/24

Tags - optional

Key	Value - optional	Remove
Name	mrs-public	Remove

Add new tag
You can add 49 more tags.

Remove
Add new subnet

Cancel **Create subnet**

Feedback

Subnet created successfully

⌚ You have successfully created 1 subnet: subnet-0703e354e1db39127

Subnets (1) Info

Subnet ID: subnet-0703e354e1db39127 X	Clear filters	Actions ▾	Create subnet
Subnet ID: subnet-0703e354e1db39127			
mrs-public	Available	vpc-04b44ba4bda462df8 mr...	10.200.0.0/24

5. **Creating ROUTING TABLE:** Select “Route Tables” and click on “Create route table”

The screenshot shows the AWS VPC Route Tables list page. It has a sidebar with 'New VPC Experience' and 'VPC Dashboard'. Under 'Route tables (2)', there is a table with columns: Name, Route table ID, Explicit subnet associations, Edge associations, Main, and VPC. Two entries are listed: 'rtb-045ffc37ba0632eb6' and 'rtb-04da216c94d750a34'. Both are associated with 'vpc-070c8b16e558c0bbb'.

Route Table Wizard

The screenshot shows the 'Create route table' wizard. In the 'Route table settings' section, the name is set to 'mrs-public-rt' and the VPC is set to 'vpc-04b44ba4bda462df8 (mrs-vpc)'. In the 'Tags' section, a tag 'Name' is added with the value 'mrs-public-rt'. At the bottom, there are 'Cancel' and 'Create route table' buttons.

Edit the route table and assign gateway

The screenshot shows the 'Edit routes' dialog. It displays a table with columns: Destination, Target, Status, and Propagated. One row is present: '10.200.0.0/16' with 'Target' set to 'local' and 'Status' as 'Active'. Another row is partially visible below it. At the bottom, there are 'Add route', 'Cancel', 'Preview', and 'Save changes' buttons.

6. Attaching SUBNETS: click on “Subnet associations”

Click “Edit subnet associations”, select your subnet and Save associations

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
mrs-public	subnet-0703e354e1db39127	10.200.0.0/24	-	Main (rtb-04da216c94d750a34)

7. Now Create and EC2 instance and attach created VPC & subnet

Step 3: Configure Instance Details
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of lower costs, and more.

Number of instances	<input type="text" value="1"/>	Launch into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	vpc-04b44ba4bda462df8 mrs-vpc	<input type="button" value="Create new VPC"/>
Subnet	subnet-0703e354e1db39127 mrs-public ap-south-1	<input type="button" value="Create new subnet"/> 251 IP Addresses available
Auto-assign Public IP	Use subnet setting (Enable)	
Hostname type	Use subnet setting (IP name)	
DNS Hostname	<input checked="" type="checkbox"/> Enable IP name IPv4 (A record) DNS requests <input checked="" type="checkbox"/> Enable resource-based IPv4 (A record) DNS requests <input type="checkbox"/> Enable resource-based IPv6 (AAAA record) DNS requests	

Output:

Students will remove print outs of some snap shots.

Conclusion:

Questions:

1. State the significance of Virtual Private Network.
2. Enlist and describe the types of subnets.
3. Differentiate between classful and classless IP addresses.
4. Compare NACL and Security groups.
5. Discuss the role of Gateway and Routing Table in networking.
6. Explain the process of configuring VPC on AWS Management console.

EXPERIMENT NO: 04 Host a word press application using EC2

Aim: To Host a word press application using EC2

Lab Outcomes:

After undergoing this laboratory module, the student will be able to:

1. Host a word press application using EC2 on Linux machine.

Hardware/Software used: AWS Management Console, EC2

Theory:

WordPress:

WordPress is a content management system (CMS) that allows you to host and build websites. WordPress contains plugin architecture and a template system, so you can customize any website to fit your business, blog, portfolio, or online store.

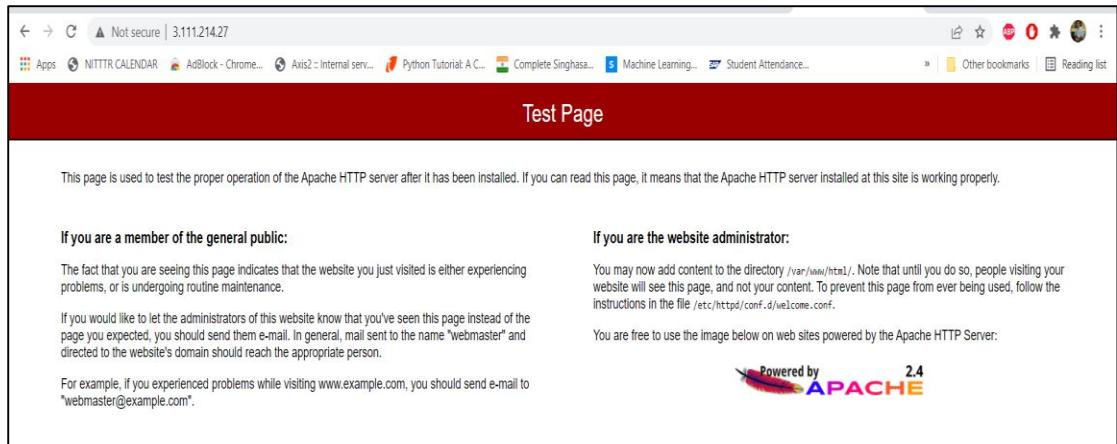
Procedure:

1. Connect to your EC2 instance
 2. Update software packages in your EC2 instance & install LMAB (mariadb & PHP)

3. Install Apache webserver in your EC2 instance.

```
55 postgresql      available [ -stable ]
60 mock2          available [ =stable ]
61 dnsmasq2.85   available [ =stable ]
* Extra topic has reached end of support.
[ec2-user@ip-172-31-43-143 ~]$ sudo yum install httpd -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.52-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.52-1.amzn2 for package: httpd-2.4.52-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.52-1.amzn2 for package: httpd-2.4.52-1.amzn2.x86_64
--> Processing Dependency: system-logos-httdp for package: httpd-2.4.52-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.52-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.52-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.52-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.52-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.52-1.amzn2.x86_64
--> Running transaction check
--> Package apr.x86_64 0:1.7.0-9.amzn2 will be installed
--> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.1-5.amzn2.0.2 for package: apr-util-1.6.1-5.amzn2.0.2.x86_64
--> Package generic-logos-httdp.noarch 0:18.0.0-4.amzn2 will be installed
--> Package httpd-filesystem.noarch 0:2.4.52-1.amzn2 will be installed
--> Package httpd-tools.x86_64 0:2.4.52-1.amzn2 will be installed
--> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
--> Package mod_http2.x86_64 0:1.15.19-1.amzn2.0.1 will be installed
--> Running transaction check
--> Package apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Finished Dependency Resolution
■
Dependencies Resolved
```

Now try to open your EC2 instance public IP address in a browser. If you see below test page, your webserver installation was successful.



2. Install MariaDB Server

Type the following command:

```
sudo vim /etc/yum.repos.d/mariadb.repo
```

```
cleaning up everything
Maybe you want: rm -rf /var/tmp/yum-ec2-user-za6TpP, to also free up space taken by orphane
[ec2-user@ip-172-31-43-143 ~]$ sudo yum install mariadb-server -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No package mariadb-server available.
Error: Nothing to do
[ec2-user@ip-172-31-43-143 ~]$ sudo systemctl status httpd
[mariadb]
name=MariaDB baseurl = http://yum.mariadb.org/10.4/centos7-amd64
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB gpgcheck=1
~
```

```
~
~
"/etc/yum.repos.d/mariadb.repo" [New] 3L, 137B written
[ec2-user@ip-172-31-43-143 ~]$ sudo yum makecache fast
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Repository 'mariadb': Error parsing config: Error parsing "gpgkey = 'https://yum.mariadb.org/RPM-GPG-KEY-MariaDB gpgcheck=1'": URL must be http, ftp, file or https not ""
amzn2-core
amzn2extra-docker
amzn2extra-kernel-5.10
amzn2extra-lamp-mariadb10.2-php7.2
amzn2extra-php7.2
Metadata Cache Created
[ec2-user@ip-172-31-43-143 ~]$
```

Now execute “sudo yum install mariadb-server mariadb-client” and test the process is running or not

```
[ec2-user@ip-172-31-43-143 ~]$ sudo systemctl start mariadb
[ec2-user@ip-172-31-43-143 ~]$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.2 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; vendor preset: disabled)
   Drop-In: /usr/lib/systemd/system/mariadb.service.d
             └─tokudb.conf
     Active: active (running) since Fri 2022-03-18 12:38:20 UTC; 7s ago
       Process: 4747 ExecStartPost=/usr/libexec/mysql-check-upgrade (code=exited, status=0/SUCCESS)
       Process: 4551 ExecStartPre=/usr/libexec/mysql-prepare-db-dir %n (code=exited, status=0/SUCCESS)
       Process: 4527 ExecStartPre=/usr/libexec/mysql-check-socket (code=exited, status=0/SUCCESS)
     Main PID: 4675 (mysqld)
       Status: "Taking your SQL requests now..."
      CGroup: /system.slice/mariadb.service
              └─4675 /usr/libexec/mysqld --basedir=/usr

Mar 18 12:38:20 ip-172-31-43-143.ap-south-1.compute.internal mysql-check-upgrade[4747]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so....ed.
Mar 18 12:38:20 ip-172-31-43-143.ap-south-1.compute.internal mysql-check-upgrade[4747]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so....ed.
Mar 18 12:38:20 ip-172-31-43-143.ap-south-1.compute.internal mysql-check-upgrade[4747]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so....ed.
Mar 18 12:38:20 ip-172-31-43-143.ap-south-1.compute.internal mysql-check-upgrade[4747]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so....ed.
Mar 18 12:38:20 ip-172-31-43-143.ap-south-1.compute.internal mysql-check-upgrade[4747]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so....ed.
Mar 18 12:38:20 ip-172-31-43-143.ap-south-1.compute.internal mysql-check-upgrade[4747]: ERROR: ld.so: object '/usr/lib64/libjemalloc.so....ed.
```

3. Change some permissions:

Change file permissions

Add your user (in this case, ec2-user) to the apache group.
sudo usermod -a -G apache ec2-user

Exit the EC2 connection and connect back
exit

```
## Connect back to your EC2 instance and verify your membership in  
the apache group  
groups
```

```
## Change the group ownership of /var/www and its contents to the  
apache group.  
sudo chown -R ec2-user:apache /var/www
```

```
## Change the directory permissions of /var/www and its subdirectories.  
sudo chmod 2775 /var/www && find /var/www -type d -exec sudo  
chmod 2775 {} \;
```

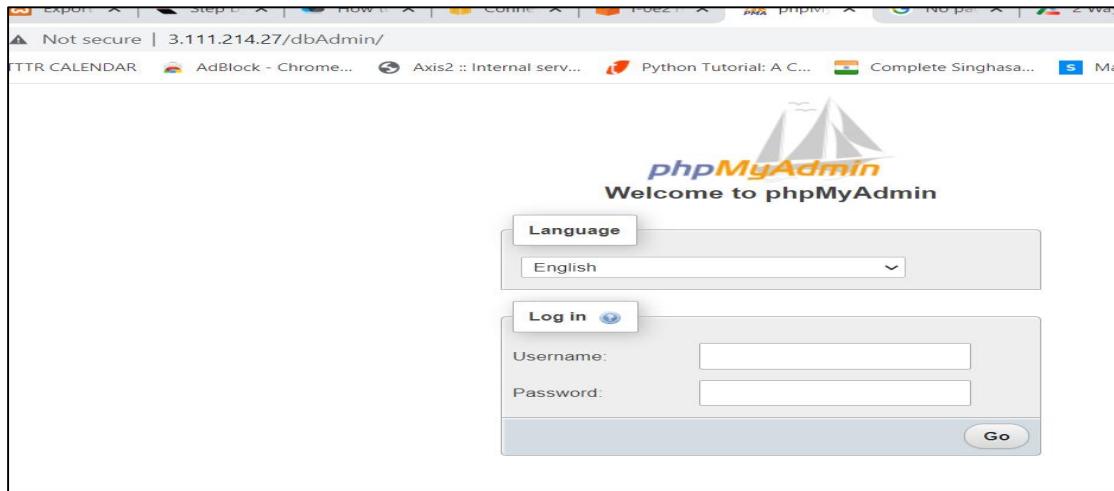
```
## Change the file permissions of /var/www and its subdirectories.  
find /var/www -type f -exec sudo chmod 0664 {} \;
```

```
[ec2-user@ip-172-31-43-143 html]$ sudo systemctl enable httpd  
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.  
[ec2-user@ip-172-31-43-143 html]$ sudo usermod -a -G apache ec2-user  
[ec2-user@ip-172-31-43-143 html]$
```

```
Last login: Fri Mar 18 12:11:27 2022 from ec2-13-233-177-0.ap-south-1.compute.amazonaws.com  
[ec2-user@ip-172-31-43-143 ~]$  
Amazon Linux 2 AMI  
[ec2-user@ip-172-31-43-143 ~]$  
https://aws.amazon.com/amazon-linux-2/  
16 package(s) needed for security, out of 29 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-43-143 ~]$ groups  
ec2-user adm wheel apache systemd-journal
```

```
[ec2-user@ip-172-31-43-143 ~]$ sudo chown -R ec2-user:apache /var/www  
[ec2-user@ip-172-31-43-143 ~]$ sudo chmod 2775 /var/www && find /var/www -type d -exec sudo chmod 2775 {} \;  
[ec2-user@ip-172-31-43-143 ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;  
[ec2-user@ip-172-31-43-143 ~]$
```

4. Now try to open phpMyAdmin in your browser using http://<ec2_public_ip>/<custom_name>



5. Install and configure WordPress in your EC2 instance

```
phpMyAdmin-5.1.3-all-languages/vendor/williamdes/mariadb-mysql-kbs/src/SlimData.php
phpMyAdmin-5.1.3-all-languages/yarn.lock
[ec2-user@ip-172-31-43-143 html]$ rm phpMyAdmin-latest-all-languages.tar.gz
[ec2-user@ip-172-31-43-143 html]$ ls
dbAdmin
[ec2-user@ip-172-31-43-143 html]$ cd dbAdmin/
[ec2-user@ip-172-31-43-143 dbAdmin]$ ls
babel.config.json config.sample.inc.php favicon.ico LICENSE README show_config_errors.php url.php
ChangeLog CONTRIBUTING.md index.php locale RELEASE-DATE-5.1.3 sql vendor
composer.json doc js package.json robots.txt templates yarn.lock
composer.lock examples libraries print.css setup themes
[ec2-user@ip-172-31-43-143 dbAdmin]$ ..
[ec2-user@ip-172-31-43-143 html]$ ls
dbAdmin
[ec2-user@ip-172-31-43-143 html]$ pwd
/var/www/html
[ec2-user@ip-172-31-43-143 html]$ cd
[ec2-user@ip-172-31-43-143 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-43-143 ~]$ wget https://wordpress.org/latest.tar.gz
--2022-03-18 13:15:50-- https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18722604 (18M) [application/octet-stream]
Saving to: 'latest.tar.gz'

100%[=====] 18,722,604 5.75MB/s in 4.3s
2022-03-18 13:15:56 (4.16 MB/s) - 'latest.tar.gz' saved [18722604/18722604]

[ec2-user@ip-172-31-43-143 ~]$ tar -xzf latest.tar.gz
[ec2-user@ip-172-31-43-143 ~]$
```

```
/var/www/html
[ec2-user@ip-172-31-43-143 html]$ cd
[ec2-user@ip-172-31-43-143 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-43-143 ~]$ wget https://wordpress.org/latest.tar.gz
--2022-03-18 13:15:50-- https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18722604 (18M) [application/octet-stream]
Saving to: 'latest.tar.gz'

100%[=====] 18,722,604 5.75MB/s  in 4.3s

2022-03-18 13:15:56 (4.16 MB/s) - 'latest.tar.gz' saved [18722604/18722604]

[ec2-user@ip-172-31-43-143 ~]$ tar -xzf latest.tar.gz
[ec2-user@ip-172-31-43-143 ~]$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 17
Server version: 10.2.38-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create user wpuser@localhost identified by wpuser@123
      -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'wpuser@123' at line 1
MariaDB [(none)]> create user 'wpuser'@'localhost' identified by 'wpuser@123';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]>
```

Create a database table

```
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 17
Server version: 10.2.38-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create user wpuser@localhost identified by wpuser@123
      -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax
* For information on other constants that can be used for debugging,
* visit the documentation.
*
* @link https://wordpress.org/support/article/debugging-in-wordpress/
*/
define( 'WP_DEBUG', false );

/* Add any custom values between this line and the "stop editing" line. */

/* That's all, stop editing! Happy publishing. */
/** Absolute path to the WordPress directory. */
if ( ! defined( 'ABSPATH' ) ) {
	define( 'ABSPATH', __DIR__ . '/' );
}

/** Sets up WordPress vars and included files. */
require_onceABSPATH . 'wp-settings.php';
"wordpress/wp-config.php" [dos] 96L, 2977B written
[ec2-user@ip-172-31-43-143 ~]$
```

Install php-gd

Cloud Application Development CLD198924

```
#<Directory "/var/www">
    AllowOverride All
    # Allow open access
    Require all granted
</Directory>

# Further relax access to the default document root:
"/etc/httpd/conf/httpd.conf" 362L, 11909B written
[ec2-user@ip-172-31-43-143 ~]$ sudo yum install php-gd -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Repository 'mariadb': Error parsing config: Error parsing "gpgkey = 'https://yum.mariadb.org/RPM-GPG-KEY-MariaDB gpgcheck=1'": URL must be http
p, ftp, file or https not ""
amzn2-core
Resolving Dependencies
--> Package php-gd.x86_64 0:7.2.34-1.amzn2 will be installed
--> Processing Dependency: libXpm.so.4()(64bit) for package: php-gd-7.2.34-1.amzn2.x86_64
--> Processing Dependency: libX11.so.6()(64bit) for package: php-gd-7.2.34-1.amzn2.x86_64
--> Running transaction check
--> Package libX11.x86_64 0:1.6.7-3.amzn2.0.2 will be installed
--> Processing Dependency: libX11-common >= 1.6.7-3.amzn2.0.2 for package: libX11-1.6.7-3.amzn2.0.2.x86_64
--> Processing Dependency: libxcb.so.1()(64bit) for package: libX11-1.6.7-3.amzn2.0.2.x86_64
--> Package libXpm.x86_64 0:3.5.12-1.amzn2.0.2 will be installed
--> Running transaction check
--> Package libX11-common.noarch 0:1.6.7-3.amzn2.0.2 will be installed
--> Package libxcb.x86_64 0:1.12-1.amzn2.0.2 will be installed
--> Processing Dependency: libXau.so.6()(64bit) for package: libxcb-1.12-1.amzn2.0.2.x86_64
--> Running transaction check
--> Package libXau.x86_64 0:1.0.8-2.1.amzn2.0.2 will be installed
--> Finished Dependency Resolution
Dependencies Resolved
```

6. Restart httpd and write wp-admin/install.php on web browser

Not secure | 3.111.214.27/wp-admin/install.php

NITTTR CALENDAR AdBlock - Chrome... Axis2 :: Internal serv... Python Tutorial: A C... Complete Singhasa... Machine Learning... Student Attendance...

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username

>Password Show

Your Email

Search engine visibility Discourage search engines from indexing this site

Important: You will need this password to log in. Please store it in a secure location.

Double-check your email address before continuing.

It is up to search engines to honor this request.

Install WordPress

7. Set title,user name, password, etc. on WordPress

Not secure | 3.111.214.27/wp-admin/install.php

NITTTR CALENDAR AdBlock - Chrome... Axis2 :: Internal serv... Python Tutorial: A C... Complete Singhasa... Machine Learning... Student Attendance...

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username

Password Show

Your Email

Search engine visibility Discourage search engines from indexing this site

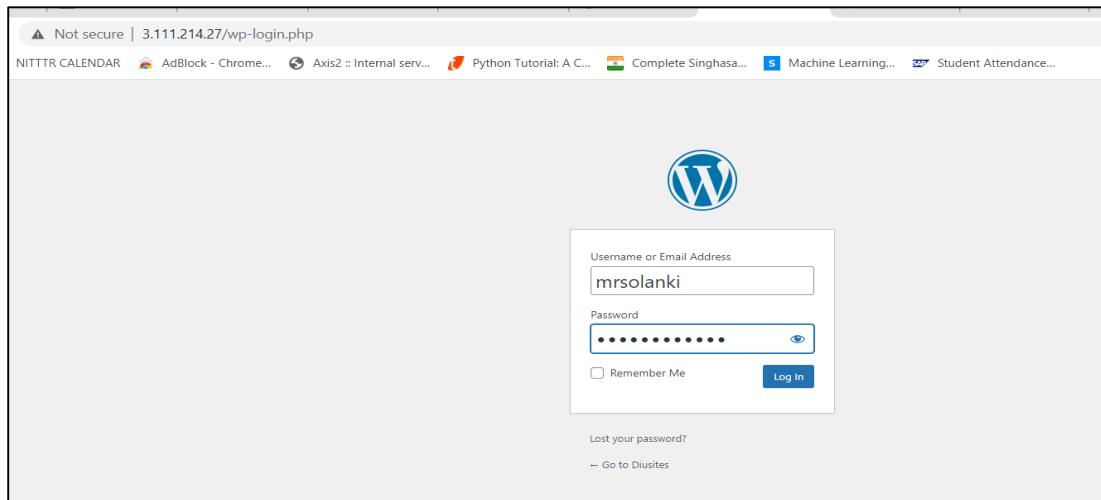
Important: You will need this password to log in. Please store it in a secure location.

Double-check your email address before continuing.

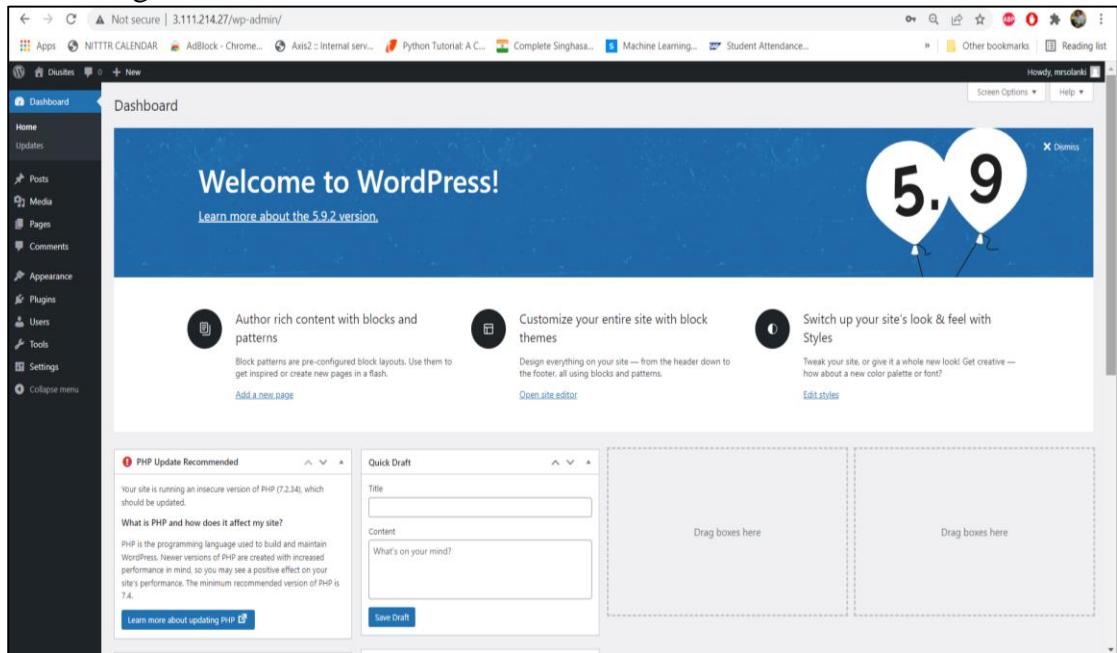
It is up to search engines to honor this request.

Install WordPress

8. Login in WordPress



9. Home Page of WordPress



Output:

Students will remove print outs of some snap shots.

Conclusion:

Questions:

1. Describe WordPress with its applications.
2. Enlist the steps to install WordPress on Linux machine Enlist the directory names of Linux File structure.
3. Define the term “LAMP”.
4. State the commands to test apache & mariadb services are running or not

Computer Engineering Department, S.B.M.Polytechnic, Mumbai.

EXPERIMENT NO: 05 Hosting a simple static web application using EC2 and Elastic Load Balancer (ELB)

Aim: To Host a simple static web application on EC2 using ELB

Lab Outcomes:

After undergoing this laboratory module, the student will be able to:

1. Host a simple static web application on EC2 using ELB

Hardware/Software used: AWS Management Console,EC2,ELB

Theory:

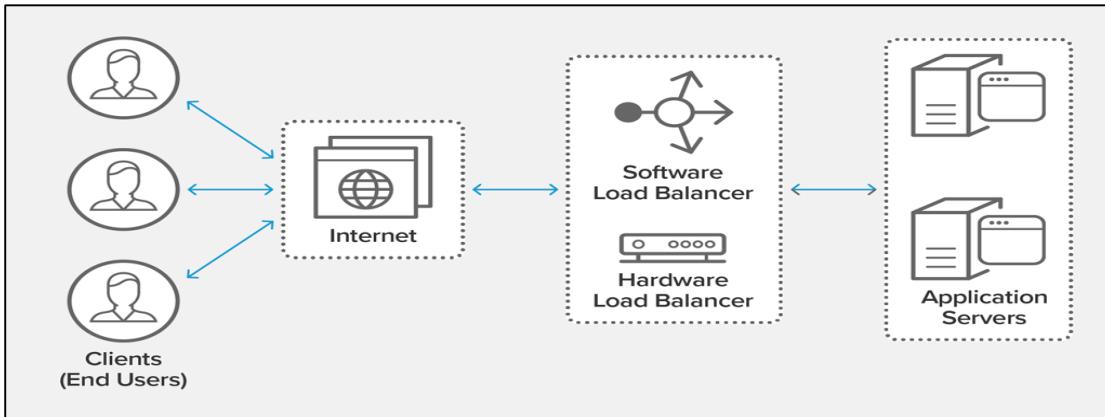
Load balancing refers to efficiently distributing incoming network traffic across a group of backend servers, also known as a *server farm* or *server pool*.

Modern high-traffic websites must serve hundreds of thousands, if not millions, of concurrent requests from users or clients and return the correct text, images, video, or application data, all in a fast and reliable manner. To cost-effectively scale to meet these high volumes, modern computing best practice generally requires adding more servers.

A load balancer acts as the “traffic cop” sitting in front of your servers and routing client requests across all servers capable of fulfilling those requests in a manner that maximizes speed and capacity utilization and ensures that no one server is overworked, which could degrade performance. If a single server goes down, the load balancer redirects traffic to the remaining online servers. When a new server is added to the server group, the load balancer automatically starts to send requests to it.

In this manner, a load balancer performs the following functions:

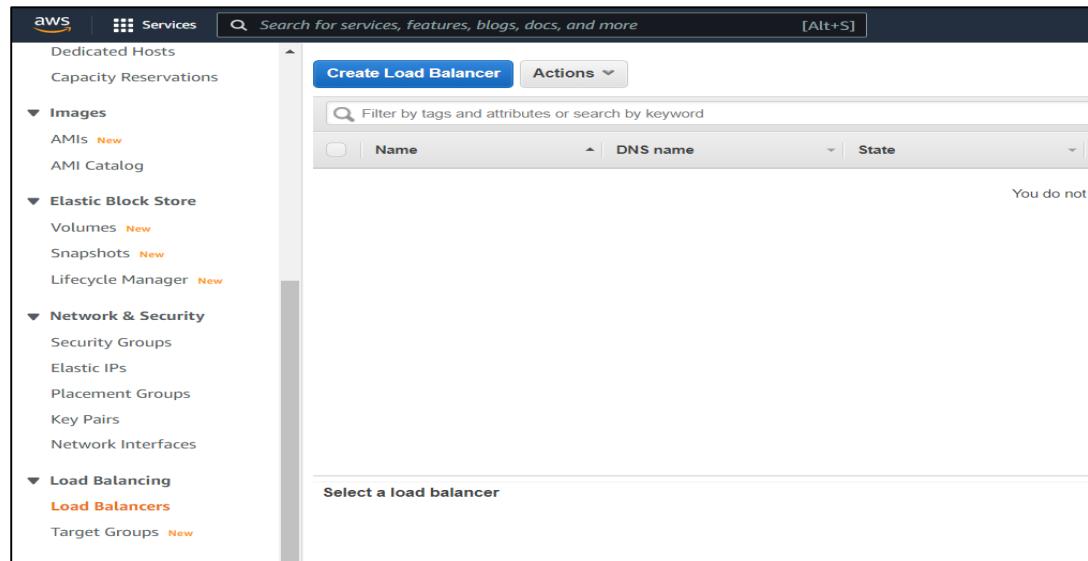
- Distributes client requests or network load efficiently across multiple servers
- Ensures high availability and reliability by sending requests only to servers that are online
- Provides the flexibility to add or subtract servers as demand dictates



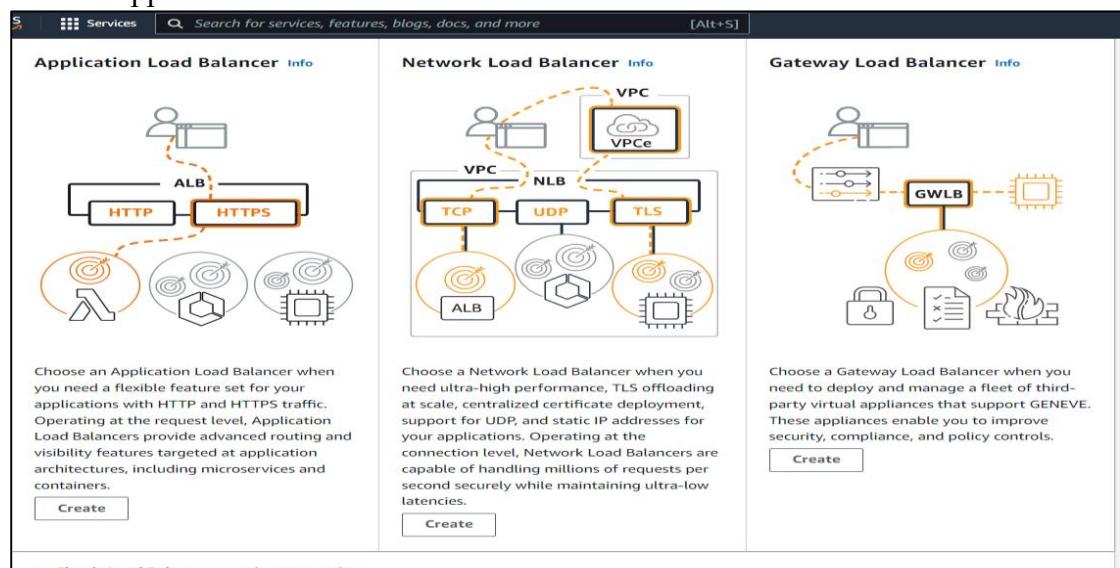
Procedure:

1. Create EC2 instance 1 with tag name WebServer 1 with AZ as ‘ap-south-1b’
2. Create EC2 instance 2 with tag name WebServer 2 with AZ as ‘ap-south-1a’
3. Write a simple html page i.e. HelloWorld as per steps enlisted as below:
Install httpd on both EC2 instances by setting permissions on /var/www/html as shown in wordpress deployment experiment
yum install httpd -y
Change the group ownership of /var/www and its contents to the apache group.
sudo chown -R ec2-user:apache /var/www

Change the directory permissions of /var/www and its subdirectories.
sudo chmod 2775 /var/www && find /var/www -type d -exec sudo chmod 2775 {} \;
Change the file permissions of /var/www and its subdirectories.
find /var/www -type f -exec sudo chmod 0664 {} \;
#Now Write index.html
echo '<h1> Response from Server-1</h1>' > /var/www/html/index.html
for WebServer2, write Server-2 in above message
#start apache server
sudo systemctl start httpd
4. Select “Load Balancers” and click on “Create Load Balancer”



Select “Application Load Balancer”



Provide basic configuration options

Create Application Load Balancer Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

► How Application Load Balancers work

Basic configuration

Load balancer name
Name must be unique within your AWS account and cannot be changed after the load balancer is created.
 A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info
Scheme cannot be changed after the load balancer is created.
 Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more [?]
 Internal
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type Info
Select the type of IP addresses that your subnets use.
 IPv4
Recommended for internal load balancers.
 Dualstack
Includes IPv4 and IPv6 addresses.

Select both the AZs

Network mapping Info
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC Info
Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To confirm the VPC for your targets, view your target groups [?]

-
vpc-070ccb16e558c0bbb
IPv4: 172.31.0.0/16

Mappings Info
Select at least one Availability Zone and one subnet for each zone. We recommend selecting at least two Availability Zones. The load balancer will route traffic only to targets in the selected Availability Zones. Zones that are not supported by the load balancer or VPC cannot be selected. Subnets can be added, but not removed, once a load balancer is created.

ap-south-1a

Subnet

IPv4 settings
Assigned by AWS

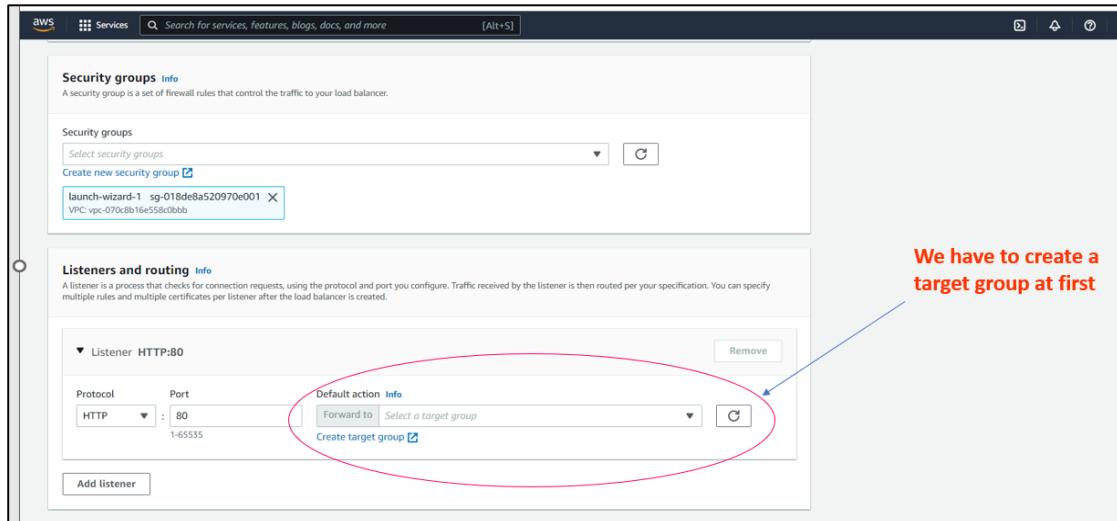
ap-south-1b

Subnet

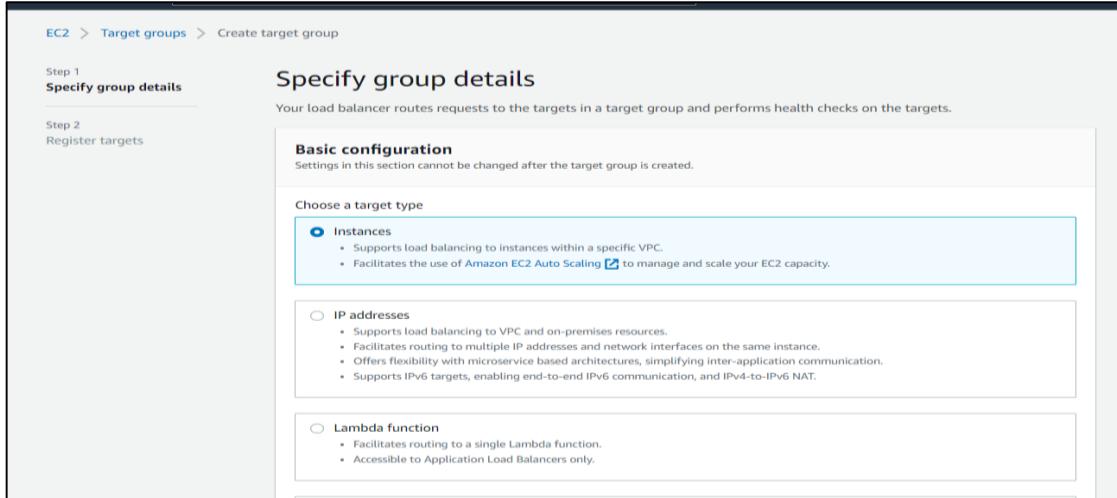
IPv4 settings

5. Create a target group at first

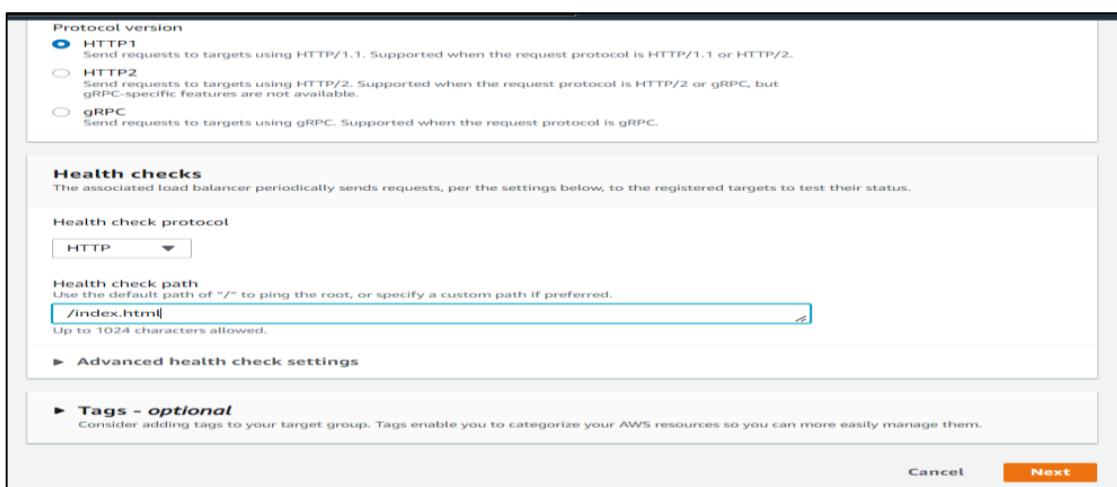
Cloud Application Development CLD198924



Specify group details



Select protocol & health check path



Select both Available instances under “Register targets”

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (2/2)

Instance ID	Name	State	Security groups	Zone	Subnet ID
i-0e8fece92775f498	WebServer2	running	launch-wizard-1	ap-south-1a	subnet-00e28c00be82029f2
i-0a761e5ce1eccabc7	WebServer1	running	launch-wizard-1	ap-south-1b	subnet-08cf3cd696b669848

Ports for the selected instances
Ports for routing traffic to the selected instances.
80
1-65535 (separate multiple ports with commas)

Include as pending below

Go to Load balancer tab and select created target group i.e. mrstg

Listeners and routing Info

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.

Listener HTTP:80

Protocol	Port	Default action
HTTP	80	Forward to mrstg <small>Target type: Instance, IPv4</small>

Add listener

Click on “Create Load Balancer”

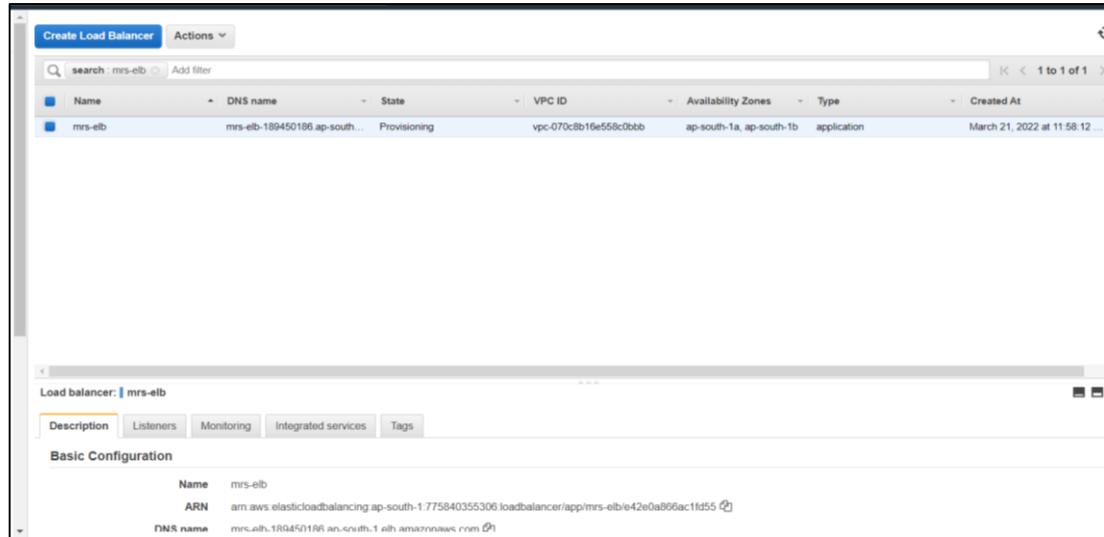
Summary

Review and confirm your configurations. Estimate cost

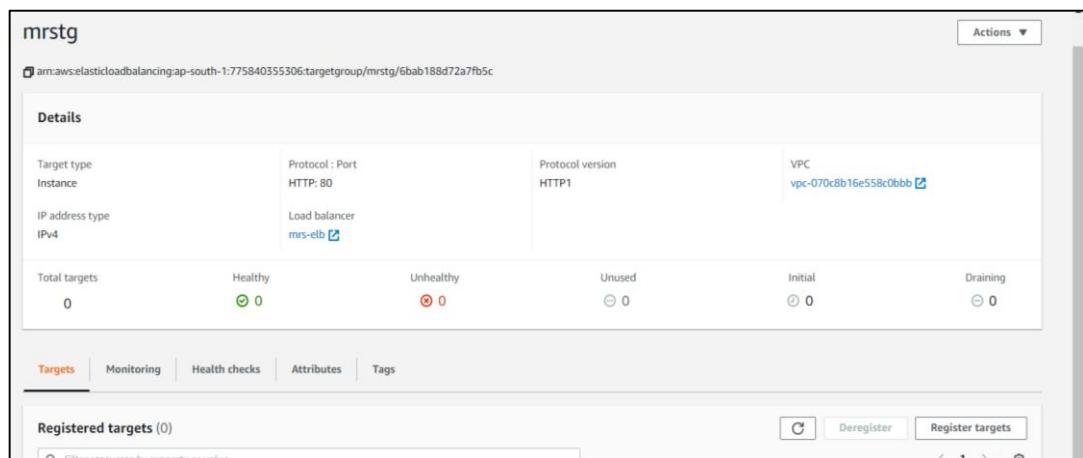
Basic configuration <small>Edit</small> mrs-elb • Internet-facing • IPv4	Security groups <small>Edit</small> • launch-wizard-1 sg-018de8a520970e001	Network mapping <small>Edit</small> VPC vpc-070c8b16e558c0bbb • ap-south-1a subnet-00e28c00be82029f2 • ap-south-1b subnet-08cf3cd696b669848	Listeners and routing <small>Edit</small> • HTTP:80 defaults to mrstg
Add-on services <small>Edit</small> None	Tags <small>Edit</small> Name mrsdemoleb		
Attributes <small>Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.</small>			

Create load balancer

Click on “View Load Balancers” and see the DNS name of the load Balancer



Click on “Registered Targets”



Select both EC2 instances i.e. WebServer1 and WebServer2

Cloud Application Development CLD198924

Available instances (2)

Instance ID	Name	State	Security groups	Zone	IPv4 address	Subnet ID
i-0efceee92775f498	WebServer2	running	launch-wizard-1	ap-south-1a	13.232.190.11	subnet-00e28c00be82029f2
i-0a761e3ce1eccabc7	WebServer1	running	launch-wizard-1	ap-south-1b	13.233.103.197	subnet-08cf3cd696b669848

Ports for the selected instances
Ports for routing traffic to the selected instances.
80
1-65535 (separate multiple ports with comma)

Include as pending below

Review targets

Click on “Include as pending below” and then click on “Register pending targets”

Targets (2)

Remove	Health status	Instance ID	Name	Port	State	Security groups	Zone	IPv4 address	Subnet ID
X	Pending	i-0efceee92775f498	WebServer2	80	running	launch-wizard-1	ap-south-1a	13.232.190.11	subnet-00e28c00be82029f2
X	Pending	i-0a761e3ce1eccabc7	WebServer1	80	running	launch-wizard-1	ap-south-1b	13.233.103.197	subnet-08cf3cd696b669848

2 pending

Cancel **Register pending targets**

Refresh the screen and you should get 2 healthy status

Details

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-070c8b16e558c0bbb
IP address type	Load balancer		
IPv4	mrs-elb		

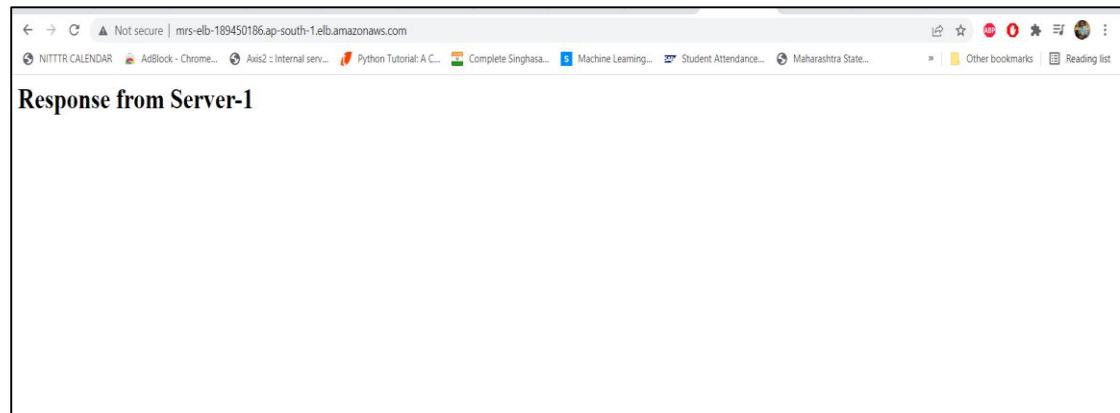
Total targets	Healthy	Unhealthy	Unused	Initial	Draining
2	2	0	0	0	0

Targets | Monitoring | Health checks | Attributes | Tags

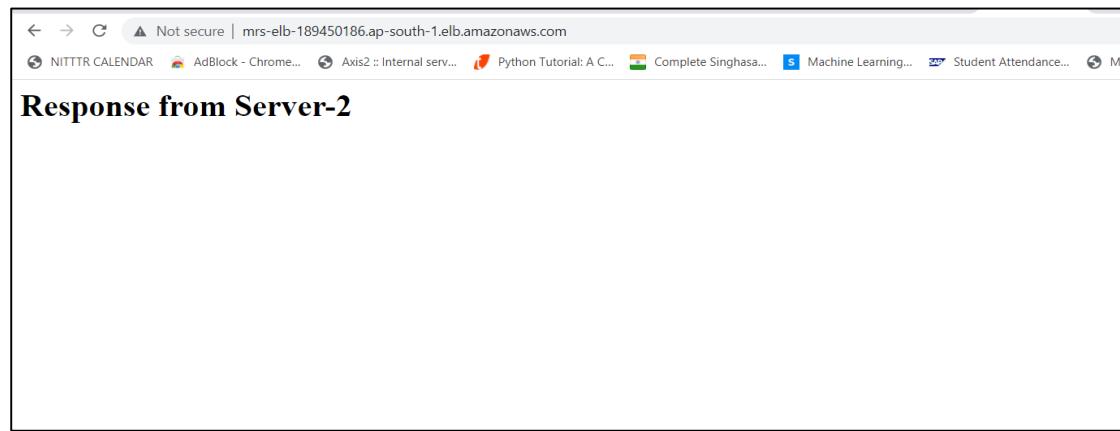
Registered targets (2)

Instance ID	Name	Port	Zone	Health status	Health status details
i-0a761e3ce1eccabc7	WebServer1	80	ap-south-1b	healthy	
i-0efceee92775f498	WebServer2	80	ap-south-1a	healthy	

6. Now copy elb DNS on the browser and see the response



Now Refresh the browser



Output:

Students will remove print outs of some snap shots.

Conclusion:

Questions:

1. State the need of Load Balancers
2. Describe the types of Load Balancers provided by AWS
3. Enlist the steps of creating target groups
4. Explain the process of creating Load Balancer to host a web page.

EXPERIMENT NO: 06 Hosting a static website using S3 with CloudFront CDN

Aim: To Host a static website using S3with CloudFront CDN

Lab Outcomes:

After undergoing this laboratory module, the student will be able to:

1. Host a static website using S3with CloudFront CDN

Hardware/Software used: AWS Management Console, AWS Cloud Front

Theory:

A content delivery network (CDN) refers to a geographically distributed group of servers which work together to provide fast delivery of Internet content.

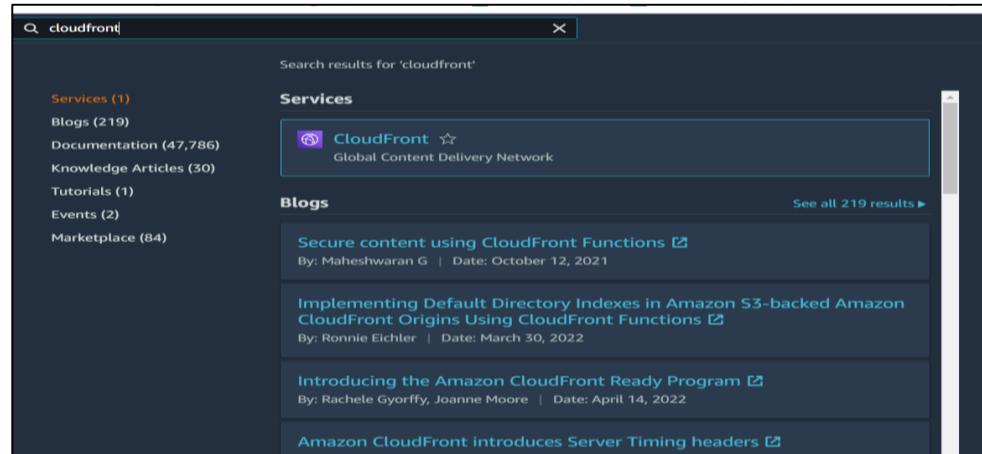
A CDN allows for the quick transfer of assets needed for loading Internet content including HTML pages, javascript files, stylesheets, images, and videos. Today the majority of web traffic is served through CDNs, including traffic from major sites like Facebook, Netflix, and Amazon.

Amazon CloudFront is a content delivery network (CDN) service built for high performance, security, and developer convenience. You can use Amazon CloudFront to do these tasks:

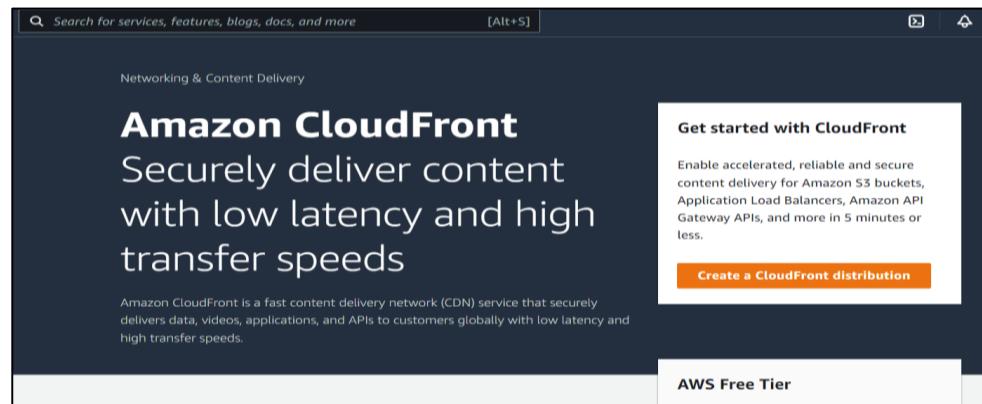
- Deliver data through 275+ globally dispersed points of presence (POPs) with automated network mapping and intelligent routing.
- Improve security with traffic encryption and access controls, and use AWS Shield Standard to defend against distributed denial-of-service (DDoS) attacks at no additional charge.
- Customize the code you run at the AWS network edge using serverless compute features to balance cost, performance, and security.
- Scale automatically to deliver software, game patches, and IoT updates with high transfer rates.

Procedure:

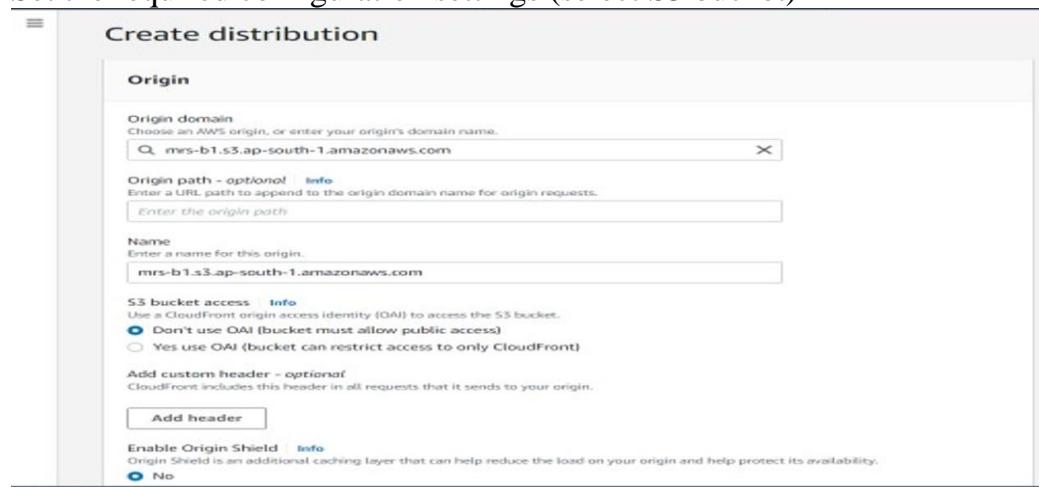
1. Search for “CloudFront”



2. Click in “Cloud Front Distribution” button



3. Set the required configuration settings (select S3 bucket)



Now, click on “Create Distributions”

The screenshot shows the 'Create Distribution' configuration page. It includes fields for 'Alternate domain name (CNAME) - optional', 'Custom SSL certificate - optional', 'Supported HTTP versions' (with 'HTTP/2' checked), 'Default root object - optional', 'Standard logging' (set to 'Off'), 'IPv6' (set to 'On'), and a 'Description - optional' field. At the bottom right are 'Cancel' and 'Create distribution' buttons.

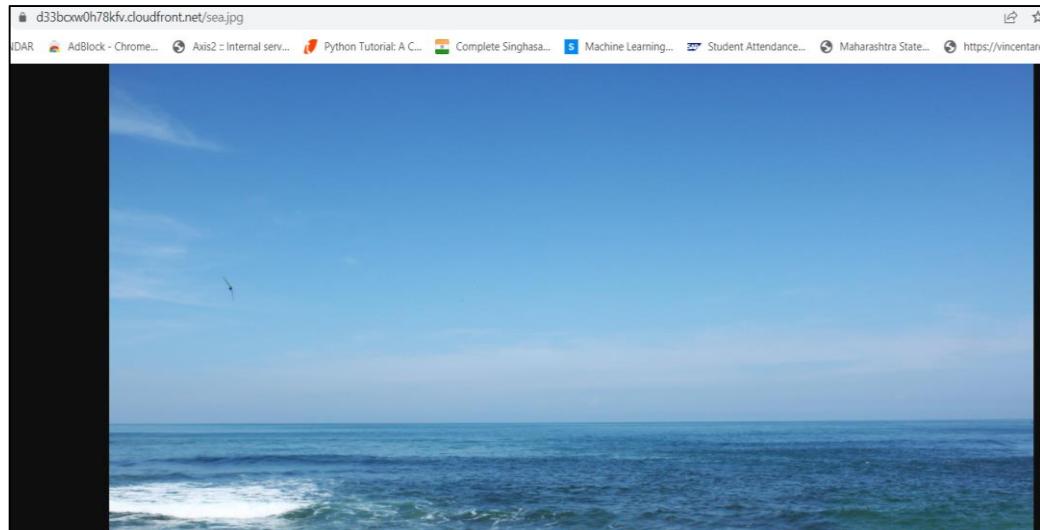
Distribution created

ID	Description	Domain name	Alternate domain ...	Origins	Status
E1VWOAN9J5GIAP	-	d33bcxw0h78kfv.cloudfront.net	-	mrs-b1.s3.ap-south-1.amazonaws.com	Enabled

Copy DNS name <https://d33bcxw0h78kfv.cloudfront.net/index.html> on the browser and see the output

The browser window displays the content of the S3 website. The main heading is 'My first S3 website'. Below it, the text reads 'It is so much easy to deploy websites with S3.' and 'This is version2.' The browser's address bar shows the URL: d33bcxw0h78kfv.cloudfront.net/index.html.

Access your image i.e. sea.jpg stored at S3 bucket



Output:

Students will remove print outs of some snap shots

Conclusion:

Questions:

1. Define CDN and describe its working.
2. Explain the working of AWS CloudFront
3. Discuss advantages of using CDN

EXPERIMENT NO: 07 Use AWS Lambda to create A server less function

Aim: To create a server less function using AWS Lambda.

Lab Outcomes:

After undergoing this laboratory module, the student will be able to:

1. Create a server less function using AWS Lambda.

Hardware/Software used: AWS Management Console, AWS Lamda

Theory:

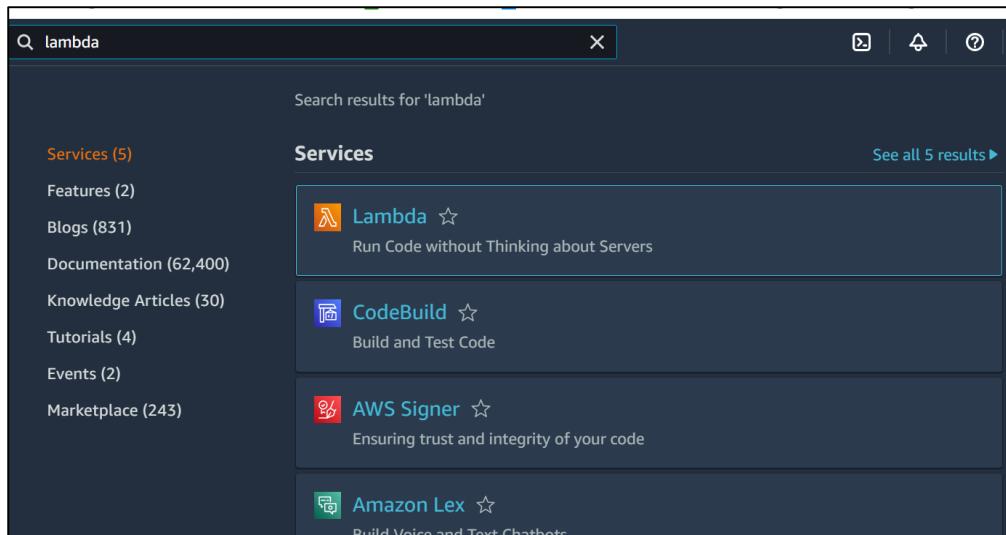
AWS Lambda is a service that allows developers and software companies to run code without having to set up and manage servers, which is often known as “serverless architecture.”

You send your function’s code to AWS, and they run it. You pay for the time the function actually gets executed, in a “pay as you go model.” Adopting AWS Lambda effectively eliminates your need for traditional computing services and infrastructure. That, in its turn, greatly reduces the cost and complexity of your IT operations, makes development times faster and scaling easier.

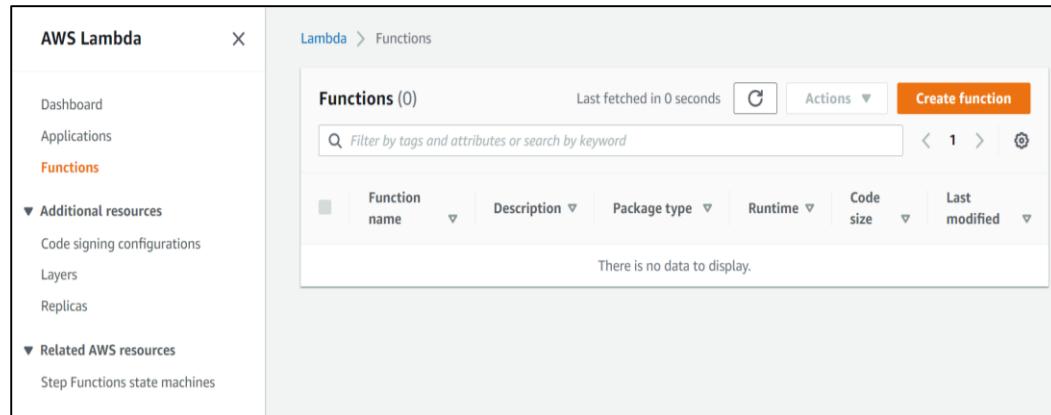
AWS Lambdas can be used today to run Node.js, Python, Java, Go, C#, and even PowerShell. Lambdas are stateless. They just run your function once, then dispose of everything.

Procedure:

1. Search “Lambda” service from search bar

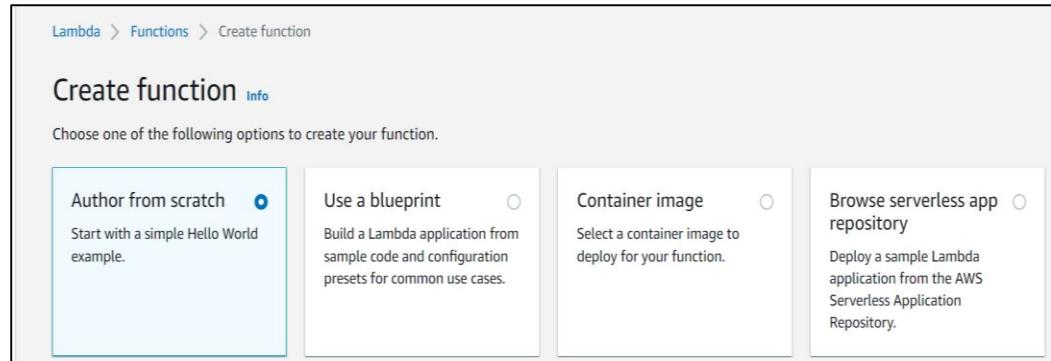


2. Inside Functions tab click on “Create function”



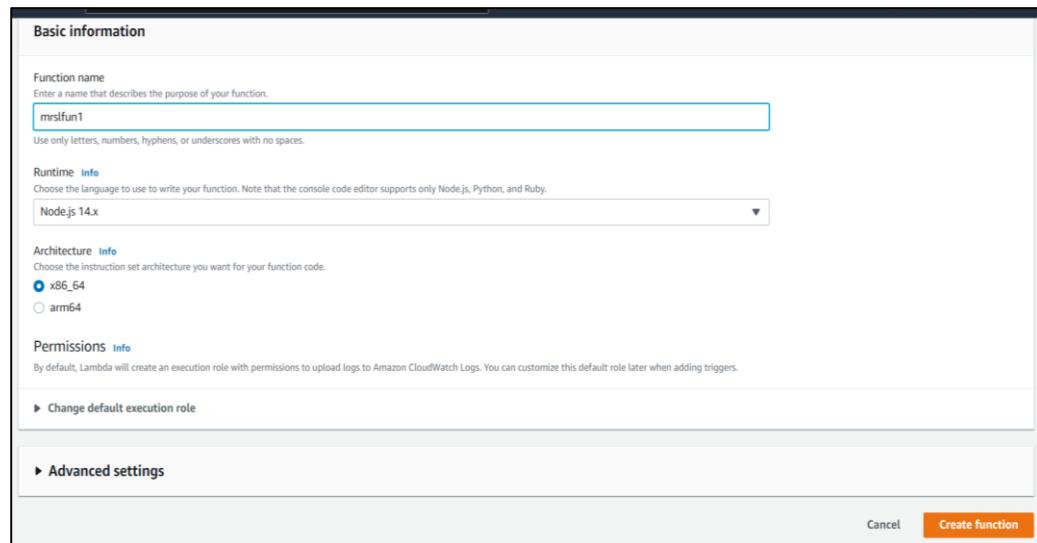
The screenshot shows the AWS Lambda Functions page. On the left, there's a sidebar with options like Dashboard, Applications, Functions (which is selected), Additional resources, and Related AWS resources. The main area is titled 'Functions (0)' and shows a table with columns for Function name, Description, Package type, Runtime, Code size, and Last modified. A message at the bottom says 'There is no data to display.' At the top right of the main area, there's a 'Create function' button.

3. Select “Author from Scratch” radio button



The screenshot shows the 'Create function' wizard. It has four options: 'Author from scratch' (selected, indicated by a blue border), 'Use a blueprint', 'Container image', and 'Browse serverless app repository'. Below each option is a brief description. The 'Author from scratch' option includes the sub-instruction: 'Start with a simple Hello World example.'

4. Give Basic Information and create a function



The screenshot shows the 'Basic information' step of the 'Create function' wizard. It includes fields for 'Function name' (mrsfun1), 'Runtime' (Node.js 14.x), 'Architecture' (x86_64 selected), and 'Permissions' (Change default execution role). There are also sections for 'Advanced settings' and a 'Create function' button at the bottom right.

5. Verification of function creation

The screenshot shows the AWS Lambda Functions console. A success message at the top states: "Successfully created the function mrslfun1. You can now change its code and configuration. To invoke your function with a test event, choose 'Test'." Below this, the function name "mrslfun1" is displayed. Under "Function overview", there is a thumbnail icon for "mrslfun1", a "Layers" section showing "(0)", and buttons for "+ Add trigger" and "+ Add destination". On the right side, there are some status indicators: "Desc -", "Last in 30", and "Func a".

6. Write "Hello World" code and specify Test parameters

The screenshot shows the AWS Lambda Code source editor. The "Code source" tab is selected. The "index.js" file contains the following code:

```
1 exports.handler = async (event) => {
2     // TODO implement
3     const response = {
4         statusCode: 200,
5         body: JSON.stringify('Hello from Lambda!'),
6     };
7     return response;
8 };
```

7. Response of Test case

The screenshot shows the AWS Lambda Test results console. The "Test" tab is selected. The "Execution result" section shows the status as "Succeeded" with a duration of "16.36 ms". The "Response" section displays the JSON output: { "statusCode": 200, "body": "\"Hello from Lambda!\""}". The "Function Logs" section shows the request and response logs.

```
START RequestId: 0da1c93e-9b6e-46d6-9e62-57b2301703d6 Version: $LATEST
END RequestId: 0da1c93e-9b6e-46d6-9e62-57b2301703d6
REPORT RequestId: 0da1c93e-9b6e-46d6-9e62-57b2301703d6 Duration: 16.36 ms Billed Duration:
Request ID
0da1c93e-9b6e-46d6-9e62-57b2301703d6
```

Example2 : Function performs Arithmetic Operations using Python

The screenshot shows the AWS Lambda function editor interface. The left sidebar shows a project named 'mrs12_python' containing a file 'lambda_function.py'. The main area displays the following Python code:

```
from __future__ import division
def lambda_handler(event, context):
    number1 = event['Number1']
    number2 = event['Number2']
    sum = number1 + number2
    product = number1 * number2
    difference = abs(number1 - number2)
    quotient = number1 / number2
    return {
        "Number1": number1,
        "Number2": number2,
        "Sum": sum,
        "Product": product,
        "Difference": difference,
        "Quotient": quotient
    }
```

Response Received from server

The screenshot shows the AWS Lambda function execution results. The 'Execution results' tab is selected, displaying the test event name 'mrs12_python' and the response object:

```
{
    "Number1": 10,
    "Number2": 20,
    "Sum": 30,
    "Product": 200,
    "Difference": 10,
    "Quotient": 0.5
}
```

Below the response, the 'Function Logs' section shows the request and response details:

```
START RequestId: 1f78e289-107c-4e4e-a2f3-f157a1f0ae25 Version: $LATEST
END RequestId: 1f78e289-107c-4e4e-a2f3-f157a1f0ae25
REPORT RequestId: 1f78e289-107c-4e4e-a2f3-f157a1f0ae25 Duration: 1.50 ms Billed Duration: 1.50 ms
Request ID
1f78e289-107c-4e4e-a2f3-f157a1f0ae25
```

Output:

Students will remove print outs of some snap shots

Conclusion:

Questions:

1. Describe “Serverless Computing”.
2. Enlist the languages supported by AWS Lambda to write functions.
3. State the possible types/formats of responses of AWS Lambda functions.
4. Discuss advantages of AWS Lambda.

EXPERIMENT NO: 08 Build a near real Time Event log System using Amazon SNS and SQS.

Aim: To Build a near real Time Event log System using Amazon SNS and SQS.

Lab Outcomes:

After undergoing this laboratory module, the student will be able to:

1. Build publisher service using AWS SNS and subscriber service using email, SMS and SQS.

Hardware/Software used: AWS Management Console, AWS SNS, AWS SQS. Docker

Theory:

AWS SNS:

Amazon Simple Notification Service (Amazon SNS) is a web service that makes it easy to set up, operate, and send notifications from the cloud. It provides developers with a highly scalable, flexible, and cost-effective capability to publish messages from an application and immediately deliver them to subscribers or other applications. It is designed to make web-scale computing easier for developers. Amazon SNS follows the “publish-subscribe” (pub-sub) messaging paradigm, with notifications being delivered to clients using a “push” mechanism that eliminates the need to periodically check or “poll” for new information and updates. With simple APIs requiring minimal up-front development effort, no maintenance or management overhead and pay-as-you-go pricing, Amazon SNS gives developers an easy mechanism to incorporate a powerful notification system with their applications.

AWS SQS:

Amazon SQS provides several advantages over building your own software for managing message queues or using commercial or open-source message queuing systems that require significant upfront time for development and configuration.

These alternatives require ongoing hardware maintenance and system administration resources. The complexity of configuring and managing these systems is compounded by the need for redundant storage of messages that ensures messages are not lost if hardware fails.

In contrast, Amazon SQS requires no administrative overhead and little configuration. Amazon SQS works on a massive scale, processing billions of messages per day. You can scale the amount of traffic you send to Amazon SQS up or down without any configuration. Amazon SQS also provides extremely high message durability, giving you and your stakeholders added confidence.

Procedure: Creating an Email & SMS message using SNS

1. Search for “Simple Notification Service”

The screenshot shows the AWS search interface with the query 'sns' entered. The search results page displays several services under the 'Services' category, with 'Simple Notification Service' being the first and most prominent result. It is described as a 'SNS managed message topics for Pub/Sub'. Other visible results include 'Route 53 Resolver', 'Route 53', and 'Amazon Simple Email Service'.

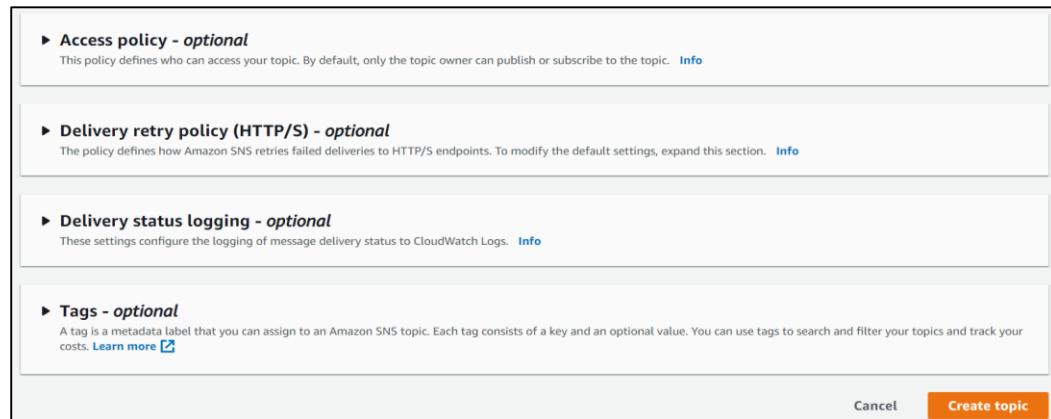
2. Create a new topic

The screenshot shows the 'Create topic' wizard for Amazon SNS. The left side features a brief introduction to SNS and its use cases. The right side contains a form to enter the topic name, with a placeholder 'MyTopic'. Below the form are buttons for 'Next step' and 'Start with an overview'. At the bottom, there is a 'Pricing' link.

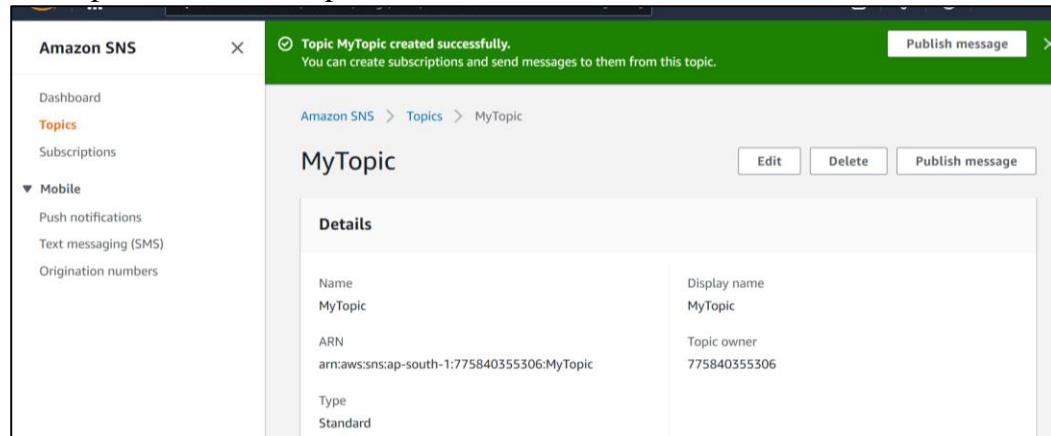
3. Select type of queue

The screenshot shows the 'Create topic' wizard with the 'Details' tab selected. It allows choosing between 'FIFO (first-in, first-out)' and 'Standard' queue types. The 'Standard' option is currently selected. Both options list their respective benefits. Below the queue type selection, there is a 'Name' field containing 'MyTopic' and a 'Display name - optional' field also containing 'MyTopic'.

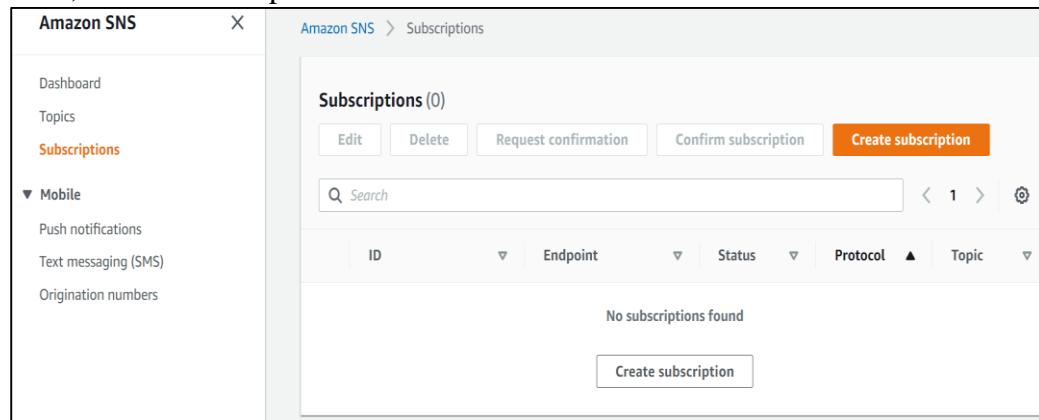
4. Select necessary options and create the topic



5. Description of created topic



6. Now, create subscription



7. Select topic ARN and Email endpoint

Create subscription

Details

Topic ARN
 X

Protocol
 The type of endpoint to subscribe
 ▼

Endpoint
 An email address that can receive notifications from Amazon SNS.
 ▼

i After your subscription is created, you must confirm it. [Info](#)

8. Create subscription

Amazon SNS X

▼ Subscription filter policy - optional
This policy filters the messages that a subscriber receives. [Info](#)

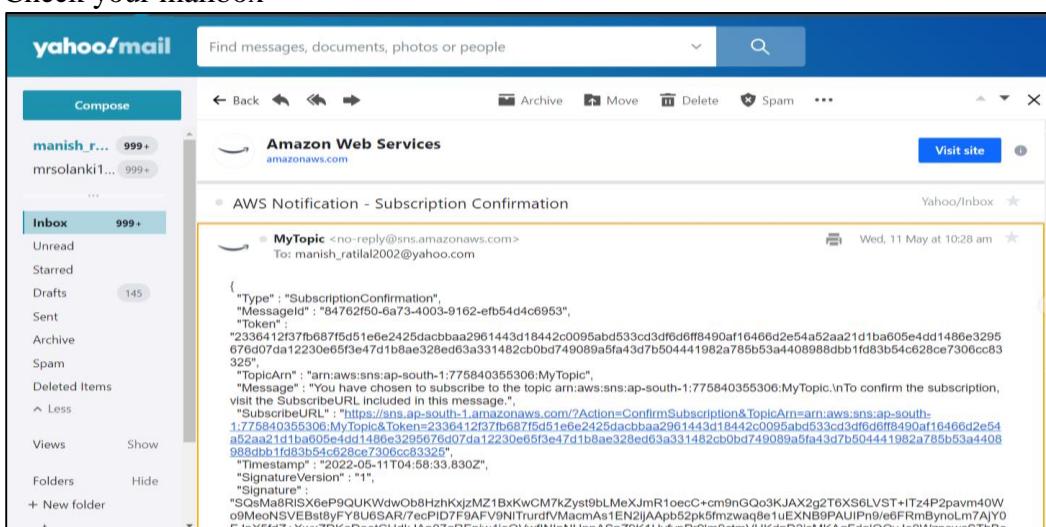
JSON editor

```
1 [
2   "MessageSource": [
3     "Medium"
4   ]
5
6 ]
```

► Redrive policy (dead-letter queue) - optional
Send undeliverable messages to a dead-letter queue. [Info](#)

Cancel Create subscription

9. Check your mailbox



The screenshot shows a Yahoo Mail inbox with several unread emails. One specific email is highlighted with a yellow border:

Amazon Web Services
amazonaws.com Visit site

AWS Notification - Subscription Confirmation

To: manish_ratilal2002@yahoo.com Wed, 11 May at 10:28 am

Type: SubscriptionConfirmation
MessageID: 84762f50-6a73-4003-9162-efb54d4c6953
Token: 23364123f7b687f5d51e6e2425daccba2961443d18442c0095abd533cd3df6d6ff8490af16466d2e5a52aa21d1ba605e4dd1486e3295676d07da12230e65f3e47d1b8ae328ed63a331482cb0bd749089a5fa43d7b50441982a785b53a4408988dbb1fd83b54c628ce7306cc8325
TopicArn: arn:aws:sns:ap-south-1:775840355306:MyTopic
Message: You have chosen to subscribe to the topic arn:aws:sns:ap-south-1:775840355306:MyTopic. To confirm the subscription, visit the [SubscribeURL](https://sns.ap-south-1.amazonaws.com/?Action=ConfirmSubscription&TopicArn=arn:aws:sns:ap-south-1:775840355306:MyTopic&Token=23364123f7b687f5d51e6e2425daccba2961443d18442c0095abd533cd3df6d6ff8490af16466d2e54988dbb1fd3b54c628ce7306cc8323) included in this message.
SubscribeURL: <https://sns.ap-south-1.amazonaws.com/?Action=ConfirmSubscription&TopicArn=arn:aws:sns:ap-south-1:775840355306:MyTopic&Token=23364123f7b687f5d51e6e2425daccba2961443d18442c0095abd533cd3df6d6ff8490af16466d2e54988dbb1fd3b54c628ce7306cc8323>,
Timestamp: 2022-06-11T04:58:33.830Z,
SignatureVersion: 1,
Signature: S0sMaBRISX6eP9QUKVdwOb8HzhKxjzMZ1BxKwCM7kZyst9bLMxJmR1oeC+Cm9nGQo3KJAX2g2T6XS6LVST+ITz4P2pavm40W09MeoNSVEBs8yFY8U6SAR/7ecPID7F9AFV9NITrurdvMacmAs1EN2lAAPb52pk5fmzwag8e1uEXNb9PAUPlPn9/e6FRmBynOl7AjY0EJnX6fdZ+Xwz2DKeDootChdkJaP07nREniw4lnOvlfNhpNhpnASnZ9k1UvvnPr9lm9ctmV/HKdnD2lsMKAnEdqI0QvJe0W7pswcSThRe

Creating SMS subscription

Create subscription

Details

Topic ARN
 X

Protocol
The type of endpoint to subscribe
 ▼

Endpoint
A mobile number that can receive notifications from Amazon SNS.

 **Sandbox destination phone numbers**
When in the sandbox, you can only deliver SMS to the sandbox destination phone numbers you have verified. [Learn more](#)

Add phone number

Add & verify phone number

Add a phone number

Add a destination phone number in SMS Sandbox in order to send a test message to the specified phone number.

Destination details

Phone number
 ▼

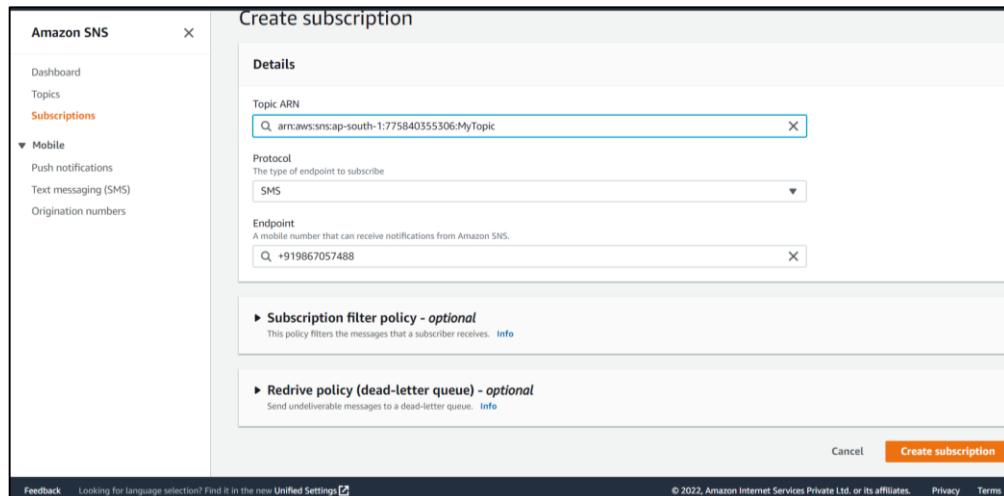
The phone number can have up to 20 digits, with a leading '+' and no spaces or hyphens (-).

Verification message language
 ▼

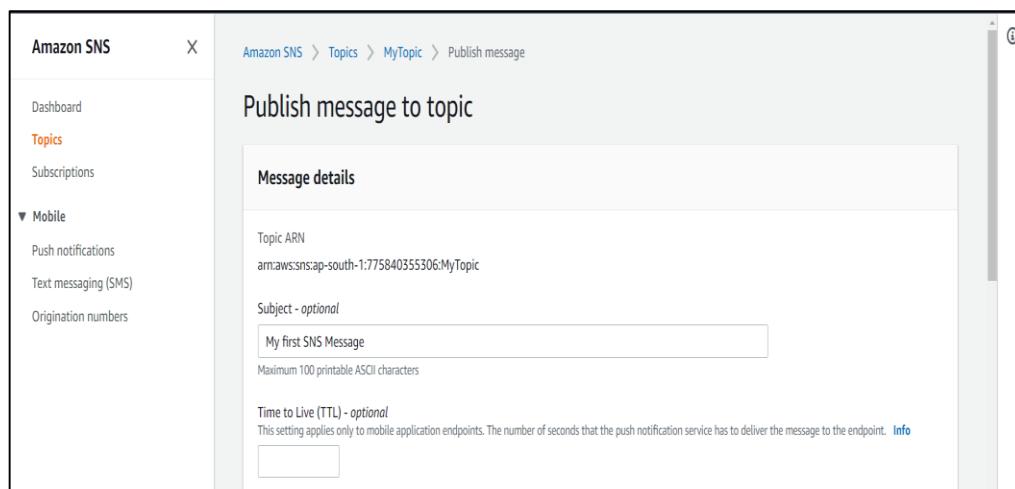
The language the verification message will be sent in.

Cancel Add phone number

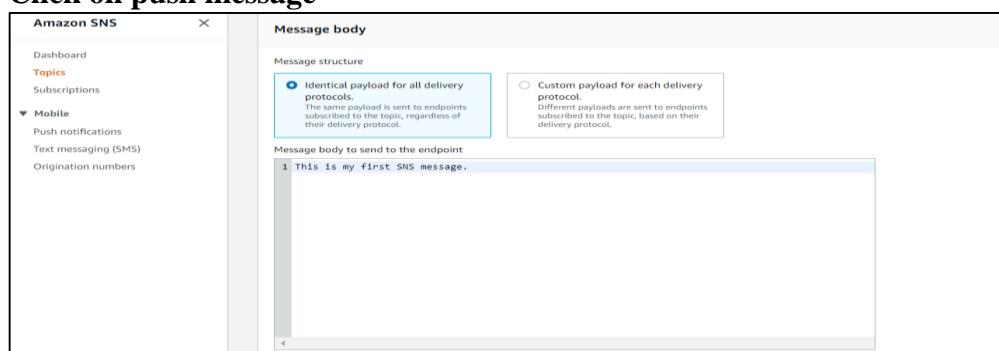
Select Topic ARN



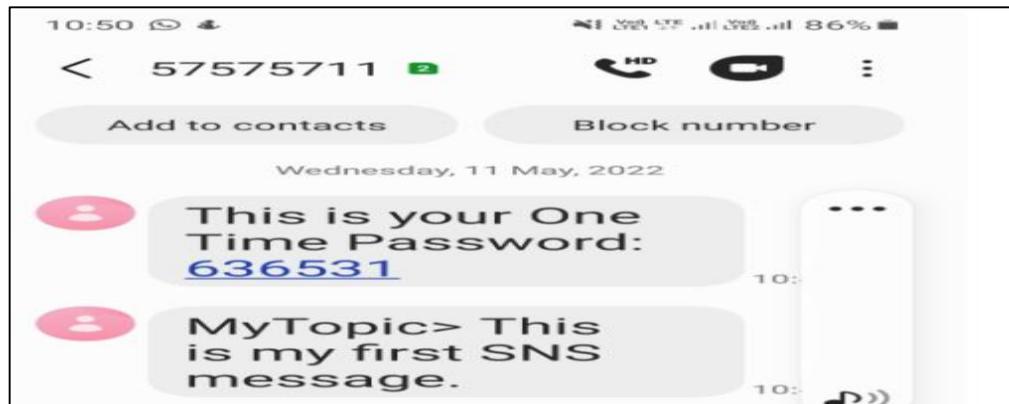
Click on “Publish message”



Click on push message



SMS received



Creating message using SQS

1. Search for SQS service

A screenshot of the AWS Management Console search interface. The search bar at the top contains the text 'sq'. On the left, there is a sidebar for 'Amazon SNS' with options like 'Dashboard', 'Topics', 'Subscriptions', and 'Mobile'. The main search results are titled 'Services (12)' and show 'Simple Queue Service' as the top result. Other services listed include 'Service Quotas', 'Simple Notification Service', and 'RDS'. Below the services, there are sections for 'Features' and 'Events'. On the right side of the search results, there are buttons for 'Edit', 'Delete', and 'Publish message'. A 'Create subscription' button is also visible.

2. Click on “Create Queue”

A screenshot of the Amazon SQS landing page. The page has a dark header with the text 'Application integration' and 'Amazon SQS' followed by 'A message queuing service'. Below the header, there is a brief description of what Amazon SQS does. To the right, there is a 'Get started' section with a 'Create queue' button. At the bottom of the page, there are sections for 'How it works' and 'Pricing (US)'.

3. Select appropriate options:

Create queue

Details

Type
Choose the queue type for your application or cloud infrastructure.

You can't change the queue type after you create a queue.

Standard [Info](#)
At-least-once delivery, message ordering isn't preserved

- At-least once delivery
- Best-effort ordering

FIFO [Info](#)
First-in-first-out delivery, message ordering is preserved

- First-in-first-out delivery
- Exactly-once processing

Name

A queue name is case-sensitive and can have up to 80 characters. You can use alphanumeric characters, hyphens (-), and underscores (_).

Configuration

Set the maximum message size, visibility to other consumers, and message retention. [Info](#)

Visibility timeout Info <input type="text" value="30"/> Seconds	Message retention period Info <input type="text" value="4"/> Days
Should be between 0 seconds and 12 hours.	
Delivery delay Info <input type="text" value="0"/> Seconds	Maximum message size Info <input type="text" value="256"/> KB
Should be between 0 seconds and 15 minutes.	
Receive message wait time Info <input type="text" value="0"/> Seconds	Should be between 0 and 20 seconds.

Access policy

Define who can access your queue. [Info](#)

► Redrive allow policy - Optional
Identify which source queues can use this queue as the dead-letter queue. [Info](#)

► Encryption - Optional
Amazon SQS provides in-transit encryption by default. To add at-rest encryption to your queue, enable server-side encryption. [Info](#)

▼ Dead-letter queue - Optional
Send undeliverable messages to a dead-letter queue. [Info](#)

Set this queue to receive undeliverable messages.
 Disabled
 Enabled

► Tags - Optional
A tag is a label assigned to an AWS resource. Use tags to search and filter your resources or track your AWS costs. [Learn more](#) 

[Cancel](#) [Create queue](#)

Looking for language selection? Find it in the new [Unified Settings](#) 

© 2022, Amazon Internet Services Private

4. Message created successfully

The screenshot shows the AWS SQS Queue MySQS1 details page. At the top, a green banner displays the message "Queue MySQS1 created successfully. You can now send and receive messages." Below the banner, the queue name is "MySQS1". The "Details" tab is selected. Key information includes:

- Name:** MySQS1
- Type:** Standard
- ARN:** arn:aws:sqs:ap-south-1:775840355306:MySQS1
- Encryption:** Disabled
- URL:** https://sns.ap-south-1.amazonaws.com/775840355306/MySQS1
- Dead-letter queue:** -

A "More" button is located at the bottom left.

5. Subscribe to Amazon Topic

The screenshot shows the AWS SNS Subscriptions page. A new subscription has been added to the queue. The subscription details are:

- Encryption:** Disabled
- URL:** https://sns.ap-south-1.amazonaws.com/775840355306/MySQS1
- Dead-letter queue:** Enabled

The "SNS subscriptions" tab is selected. Other tabs include Lambda triggers, Dead-letter queue, Monitoring, Tagging, Access policy, Encryption, and Dead-letter queue redrive tasks. The "Subscription region" dropdown is set to "ap-south-1". The "SNS subscriptions (0)" section shows a single entry with a "View in SNS" link, a "Delete" button, and an orange "Subscribe to Amazon SNS topic" button. A search bar and navigation controls are also present.

6. Specify SNS ARN

The screenshot shows the "Subscribe to Amazon SNS topic" dialog box. It contains the following fields:

- Amazon SNS topic:** To allow your queue to receive messages from an Amazon SNS topic, subscribe it to an Amazon SNS topic.
- Specify an Amazon SNS topic available for this queue:** A dropdown menu containing the ARN "arn:aws:sns:ap-south-1:775840355306:MyTopic".

At the bottom right are "Cancel" and "Save" buttons.

7. Subscribed successfully

The screenshot shows the AWS SQS Queue MySQS1 details page. The green banner at the top now displays the message "Subscribed successfully to topic arn:aws:sns:ap-south-1:775840355306:MyTopic.". The queue details are identical to the previous screenshot, including the ARN and Dead-letter queue settings.

8. Click on “Send and receive messages”

Amazon SQS > Queues > MySQS1

MySQS1

[Edit](#) [Delete](#) [Purge](#) [Send and receive messages](#) [Start DLQ redrive](#)

[Details](#) [Info](#)

Name MySQS1	Type Standard	ARN arn:aws:sqs:ap-south-1:775840355306:MySQS1
Encryption Disabled	URL https://sns.ap-south-1.amazonaws.com/775840355306/MySQS1	Dead-letter queue Enabled
More		

Send and receive messages

Send messages to and receive messages from a queue.

[Send message](#) [Info](#)

[Clear content](#) [Send message](#)

Message body
Enter the message to send to the queue.
This is the message for the Queue

Delivery delay [Info](#)
10 Seconds

Should be between 0 seconds and 15 minutes.

Message attributes - Optional [Info](#)

Name	String	Custom type	MediumMessage
Number	Number	float	3.14
Remove			

[Add new attribute](#)

Send and receive messages

Send messages to and receive messages from a queue.

[Send message](#) [Info](#)

[Clear content](#) [Send message](#)

Your message has been sent and can be received in 10 seconds. [View details](#) [X](#)

Message body
Enter the message to send to the queue.
This is the message for the Queue

Delivery delay [Info](#)
10 Seconds

Should be between 0 seconds and 15 minutes.

Message attributes - Optional [Info](#)

Name	String	Custom type	MediumMessage
Number	Number	float	3.14
Remove			

9. Check the SNS to see SQS is added as a subscriber

The screenshot shows the 'Subscriptions' page in the Amazon SNS console. The left sidebar has 'Subscriptions' selected. The main area displays a table of three confirmed subscriptions:

ID	Endpoint	Status	Protocol	Topic
fa53a192-129b-45cd-80d9-81320dd86cb6	manish_ratilal2002@yahoo.com	Confirmed	EMAIL-JSON	MyTopic
c9f2ffac-74a4-4a17-add2-3076d41367f0	+919867057488	Confirmed	SMS	MyTopic
a778144e-5af8-4be9-a916-264b468832f9	arn:aws:sqs:ap-south-1:775840355306:MySQ S1	Confirmed	SQS	MyTopic

Output:

Students will remove print outs of some snap shots

Conclusion:

Questions:

1. Describe the working of AWS SNS.
2. Describe the working of AWS SQS with types of queues.
3. Enlist the steps to create Email and SMS message service using AWS SNS
4. State the process of creating messaging using SQS.

EXPERIMENT NO: 09 Write an application to log the data using Open Cloud platform

Aim: To Write an application to log the data using Open Cloud platform

Lab Outcomes:

After undergoing this laboratory module, the student will be able to:

1. Create and setup private cloud using Open Source platform
2. Write an application to log data on open source cloud

Hardware/Software used: OpenStack Cloud Platform

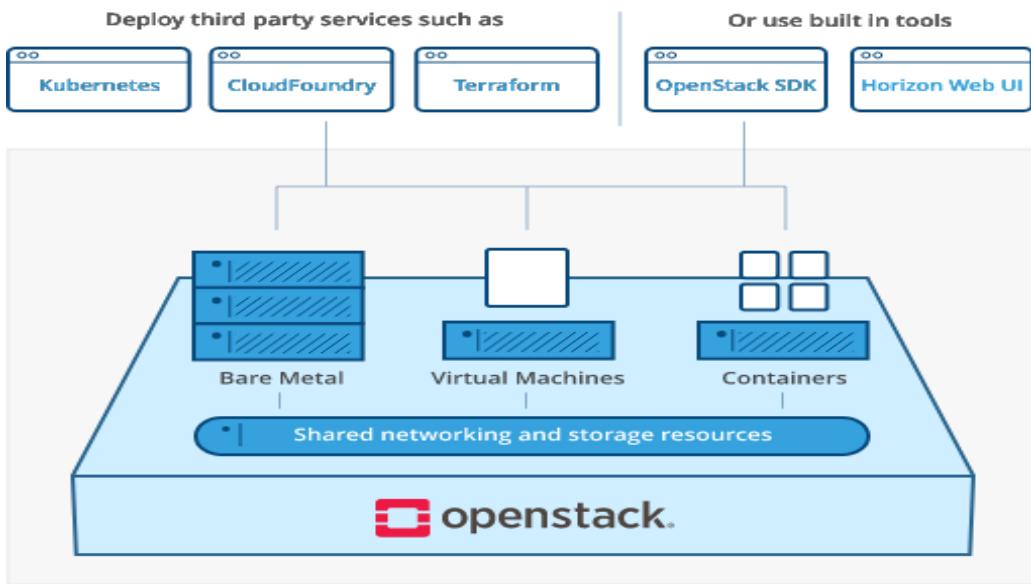
Theory:

OpenStack is an open source cloud computing platform that is used by organizations to manage and control large scale deployments of virtual machines, such as in a cloud computing or virtual private server environment. OpenStack is a popular choice for organizations because it is scalable, reliable, and provides a high degree of control over the underlying infrastructure.

Besides being used to manage deployments of virtual machines, OpenStack can also be used to manage storage and networking resources in a cloud environment.

Here are some key differences between AWS & OpenStack Cloud :

- OpenStack is an open source platform, while AWS is a proprietary platform.
- OpenStack offers more flexibility and customization options than AWS.
- OpenStack typically requires more technical expertise to set up and manage than AWS since you basically have to set up everything yourself.



Some of the more common OpenStack services are listed below:

Object Storage: OpenStack Object Storage (Swift) is a highly scalable, distributed object storage system.

Compute: OpenStack Compute (Nova) is a cloud computing fabric controller, which manages the allocation of compute resources.

Networking: OpenStack Networking (Neutron) is a system for managing networks and IP addresses.

Dashboard: The OpenStack Dashboard (Horizon) is a web-based interface for managing OpenStack resources.

Identity: OpenStack Identity (Keystone) is a system for managing user accounts and access control.

Image: OpenStack Image (Glance) is a service for storing and retrieving virtual machine images.

Block Storage: OpenStack Block Storage (Cinder) is a service for managing block storage devices.

Telemetry: OpenStack Telemetry (Ceilometer) is a service for collecting and storing metering data.

Orchestration: OpenStack Orchestration (Heat) is a service for orchestration and cloud formation.

Bare Metal: OpenStack Bare Metal (Ironic) is a service for provisioning and managing bare metal servers.

Data Processing: OpenStack Data Processing (Sahara) is a service for provisioning and managing Hadoop and Spark clusters.

One of the easiest ways to get started with OpenStack is by using the OpenMetal on-demand private cloud. This allows us to quickly deploy OpenStack to the cloud and

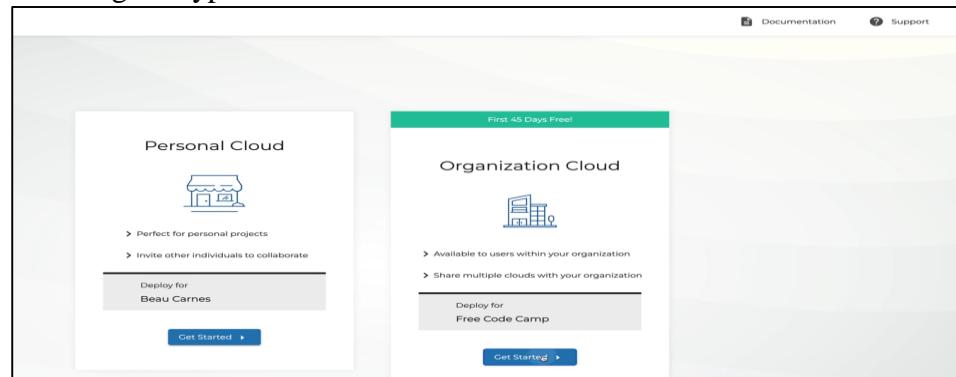
simplifies the setup process. OpenMetal provided a grant that made this tutorial possible.

Procedure: Creating an account and setting up cloud environment in OpenStack.

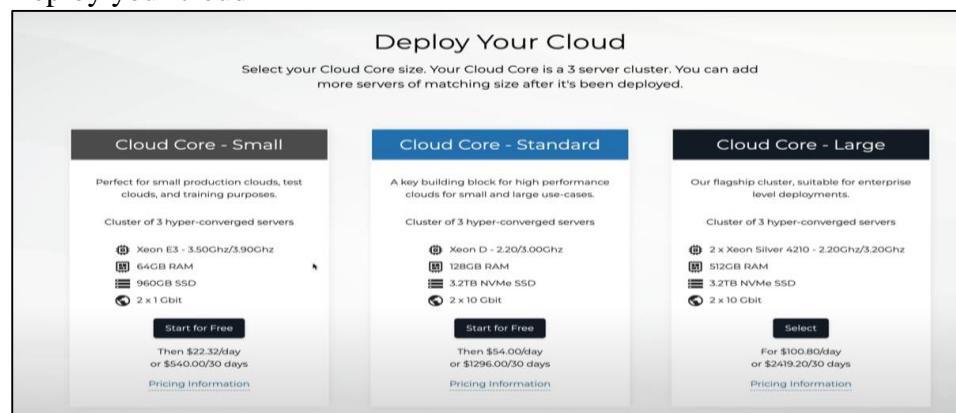
1. Creating an account with OpenStack

The screenshot shows the 'Account Setup' page of the OpenMetal website. On the left, there's a promotional banner with the text 'A few clicks away from creating your private cloud.' and 'Set up complex infrastructure in minutes.' Below the banner, there are two sign-in options: 'Continue with email' and 'sign in using Google'.

2. Selecting the type of cloud



3. Deploy your cloud



4. Provide the description of cloud

5. Provide public SSH key

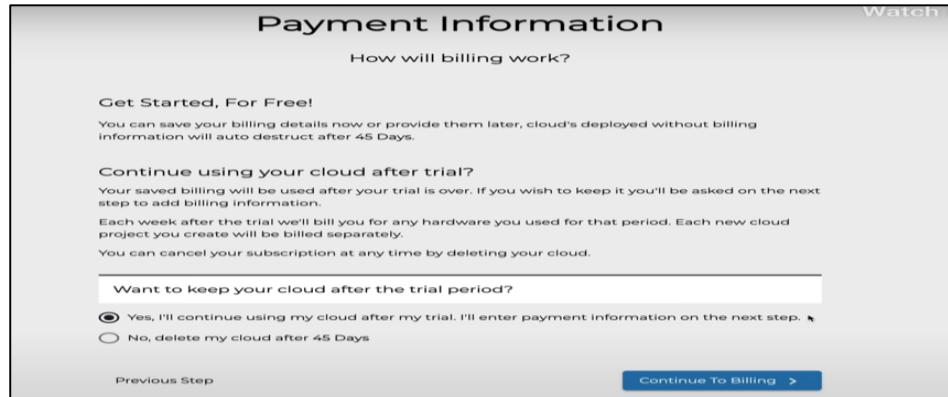
6. Generate SSH key

```
→ ~ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/beau/.ssh/id_rsa): █

→ .ssh ls
id_rsa.id_rsa.pub known_hosts
→ .ssh vim id_rsa.pub █
```

7. Get the key value from id_rsa.pub file

8. Fill the payment information



9. Use following commands to get the password for “Horizon” service

The screenshot shows the "My Cloud" dashboard. The "Status" section indicates the cloud is "RUNNING" and provides instructions for SSHing into the server. The "Access Details" section contains instructions for obtaining the Horizon administrator password via SSH and provides a command example:

```
$ ssh -i ~/.ssh/your_key_name root@173.231.217.21
```

```
→ ~ ssh -i ~/.ssh/id_rsa.pem root@173.231.217.21
Enter passphrase for key '/Users/beau/.ssh/id_rsa.pem':
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Fri Apr 29 14:45:50 2022 from 108.89.89.153
[root@modest-galliform ~]# grep keystone_admin_password /etc/kolla/passwords.yml
keystone_admin_password: 5d2vpI7BoXauA4ayzpxZxjD8ed0gTwBqV9
qAqged
[root@modest-galliform ~]#
```

10. Hardware resources are displayed under “Assets” tab

Cloud Application Development CLD198924

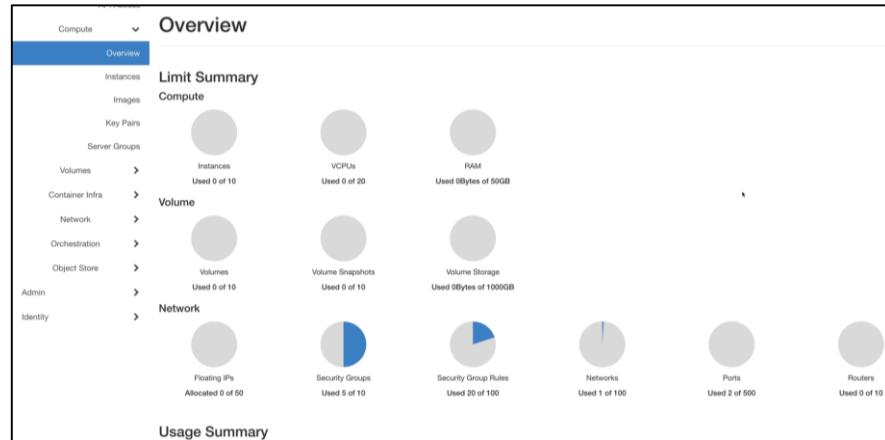
The screenshot shows a dashboard with the following sections:

- Hardware:** A table listing three Cloud Core - Small instances. Each instance has 8 cores, 64 GB RAM, and 1 TB storage. Hostnames include modest-galliform.local, gifted-badger.local, and hopeful-guineafowl.local. Public IP addresses are 173.231.217.21, 173.231.217.22, and 173.231.217.23.
- Inventory IP Address Blocks:** Shows a single CIDR range 173.231.217.6/28 with 14 hosts.
- Provider IP Address Blocks:** Shows a single CIDR range 173.231.255.32/28 with 14 hosts.

11. Login inside Horizon



12. Dashboard of the usage/performance of resources



Note: Now, we can use various services of OpenSatck as per our requirements

Output:

Students will remove printouts of some snapshots

Conclusion:

Questions:

1. Enlist any 3 open source cloud providers.
2. State any 2 differences between OpenStack and AWS cloud.
3. Describe the process of generating SSH key in OpenStack.
4. Describe any 3 services of OpenStack cloud.
5. State the number/quantity & capacity of hardware resources provided by OpenStack in free access.

EXPERIMENT NO: 10 Explain the business model of public cloud

Aim: To explain the business model of public cloud

Lab Outcomes:

After undergoing this laboratory module, the student will be able to:

1. Explain the business model of public cloud

Hardware/Software used: Literature Survey

Theory:

Public Clouds

A public cloud is an IT model where public cloud service providers make computing services – including compute and storage, develop-and-deploy environments, and applications available on-demand to organizations and individuals over the public internet. Some public cloudcomputing resources are available for free, while customers may pay for other resources through subscription or pay-per-usage pricing models.

There are various public cloud providers operating in the cloud infrastructure and services domain. Each provider has key services that are oriented towards applications and deployments of varying scales – from hobbyist applications to large scale enterprise servers and applications. Mostenterprise applications operate on the various services of the same infrastructure stack provided bythe provider. This ensures consistent and streamlined monitoring, security, access control and billing. The various public cloud providers used in the industry are -

- Google Cloud Platform (GCP)
- Amazon Web Services (AWS)
- Microsoft Azure
- Digital Ocean
- IBM Cloud (Kyndryl)
- Oracle Cloud
- Akamai Linode
- Tencent Cloud

Amazon Web Services, Google Cloud Platform, Microsoft Azure and Digital Ocean are moregeared towards hobbyists / individual applications, but also have the capacity to handle multi-tenantenterprise applications. IBM, Linode and Oracle are enterprise friendly cloud deployment providersthat provide high-performance, high-reliability compute environments that suit large-scale

application deployments. All competing cloud providers have a common set of services offered under various brand labels, such as -

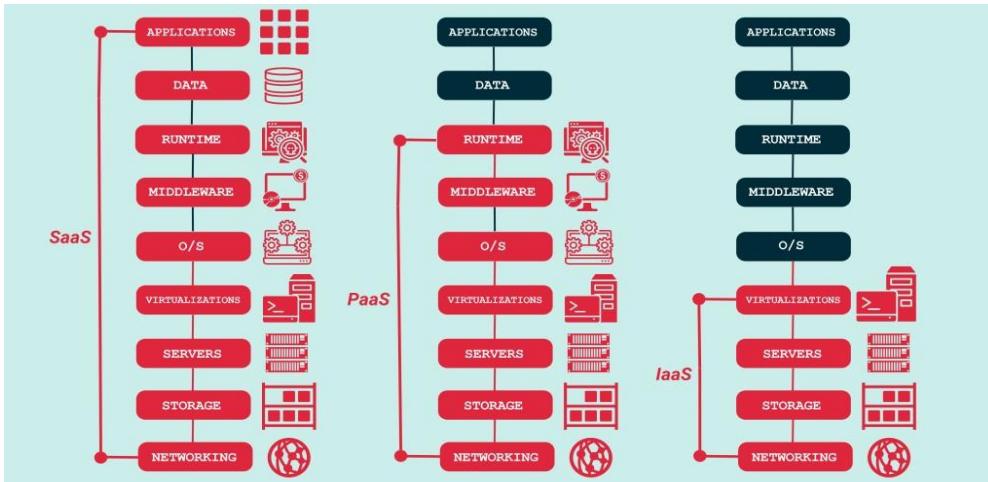
- Storage – AWS S3, GCP Cloud Storage, Azure Disk Storage
- Identity and Authorization – AWS IAM, GCP IAM, Azure Active Directory
- Database – AWS DynamoDB, GCP Firebase, Azure CosmosDB
- Virtual Machines – AWS EC2, GCP Compute Engine, Azure VMs
- Cloud Functions – AWS Lambda, GCP Cloud Functions, Azure Functions



Business Models

The business model of a cloud provider determines the operation stack, services offered as well as a financial revenue model for each service provided. Services provided by cloud infrastructure operators can be classified as follows -

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)
- Identity as a Service (IDaaS)
- Data as a Service (DaaS)



Each business model of a service includes a set of cloud-based components used in the service. For example -

- **Infrastructure as a Service** – IaaS primarily provides virtual hardware resources for cloud-based compute instances. Virtual Machines, such as EC2, Compute Engine, etc. are largely IaaS services. It is up to the client / customer to determine software configuration of the specific service. IaaS also includes resource scaling and management as a core component.
- **Platform as a Service** – PaaS provides a fully functioning cloud platform for customers to deploy their application on. Pre-configured VMs enable swift application deployments with minimal system configuration. PaaS is largely used in cloud deployments, such as websites, cloud services, etc. PaaS is built on top of the infrastructure of IaaS.
- **Software as a Service** – SaaS uses a comprehensive technology stack to provide individual software functions to end users. Customers do not have to take into consideration hardware or software configuration. A majority of services offered come under SaaS. Cloud providers develop various APIs, Frameworks and SDKs to improve cohesion of client applications with cloud services.
- **Identity as a Service** – Identity, Authentication, Authorization and Accounting management is a massive task for large-scale organizations to carry out manually. Cloud-based identity management tools, such as IAM or Active Directory, enable automated actions to bulk create, retrieve, update, and delete user identities and roles. Having a single identity provider conforming to industry standards enables one-click authentication and authorization, such as Single Sign On (SSO) functionality.
- **Data as a Service** – In an emerging new world, data has taken the center stage. DaaS provides centralized data repositories, cloud databases, AI & ML Datasets, mock data as well as application data security and consistency required in enterprise applications. DaaS is a data-focused extension of SaaS.

Various sub-services provided by cloud operators are largely extensions and additions to the SaaS stack. As real-world requirements keep growing, newer services are developed to meet enterprise constraints and demands.

Computer Engineering Department, S.B.M.Polytechnic, Mumbai.

Output:

Students will write the philosophy and business models of public cloud in the journal.

Conclusion:

Questions:

1. Describe the working of public clouds.
2. Enlist any 3 providers of public cloud with brief description of each.

EXPERIMENT NO: 11 Compare Cloud services and component offered by Google & Azure with AWS

Aim: To compare Cloud services and component offered by Google & Azure with AWS.

Lab Outcomes:

After undergoing this laboratory module, the student will be able to:

1. Compare Cloud services and component offered by Google, Azure and AWS.

Hardware/Software used: Literature Survey

Theory:

Google Cloud:

Google Cloud is a suite of cloud computing services offered by Google. It provides a range of services, including computing, storage, networking, data analytics, and machine learning, that can be used by individuals, businesses, and organizations of all sizes.

Google Cloud offers a wide range of components and services that help individuals and organizations manage their data and applications more efficiently. Here are some of the key components offered by Google Cloud:

1. **Compute Engine:** This is a virtual machine (VM) service that allows users to launch and manage virtual machines in the cloud. With Compute Engine, users can choose from a variety of pre-configured VMs or create their own customized VMs.
2. **Kubernetes Engine:** This is a managed container orchestration service that makes it easier to deploy and manage containerized applications. Kubernetes Engine is built on top of the popular open-source Kubernetes platform and provides features such as automatic scaling and self-healing.
3. **App Engine:** This is a platform as a service (PaaS) offering that enables developers to build and deploy web and mobile applications on a fully managed infrastructure. App Engine supports multiple programming languages and provides features such as automatic scaling and traffic splitting.
4. **Cloud Storage:** This is a highly scalable and durable object storage service that enables users to store and retrieve any amount of data from anywhere in the world. Cloud Storage is designed to support a wide range of use cases, from small-scale backups to large-scale content distribution.
5. **Cloud SQL:** This is a managed relational database service that makes it easier to set up, operate, and scale relational databases on Google Cloud. Cloud SQL supports MySQL, PostgreSQL, and SQL Server, and provides features such as automatic backups, replication, and failover.

6. **BigQuery:** This is a fully managed data warehouse service that enables users to analyze large datasets using SQL-like queries. BigQuery is designed to be fast and scalable, and can process petabytes of data in seconds.
7. **Cloud Functions:** This is a serverless compute service that enables users to run code in response to events, without having to manage servers or infrastructure. Cloud Functions supports multiple programming languages and can be used for a wide range of use cases, from data processing to mobile backends.

Overall, Google Cloud offers a comprehensive set of components and services that can help organizations of all sizes and types manage their data and applications more effectively and efficiently.

Advantages of Google Cloud Services:

1. **Scalability:** Google Cloud provides on-demand access to computing resources, which means that users can quickly scale up or down as needed, without having to worry about the underlying infrastructure.
2. **Cost-effectiveness:** Google Cloud offers a pay-as-you-go pricing model, which means that users only pay for the resources they use, without any upfront costs or long-term commitments.
3. **Flexibility:** Google Cloud supports a wide range of programming languages, frameworks, and tools, which makes it easier for users to build, deploy, and manage their applications.
4. **Security:** Google Cloud provides a range of security features, such as encryption, identity and access management, and network security, to help protect users' data and applications.
5. **Machine learning:** Google Cloud provides a range of machine learning services, such as image and speech recognition, natural language processing, and predictive analytics, that can be used to build intelligent applications.

Disadvantages of Google Cloud Services

1. **Complexity:** Google Cloud is a powerful and flexible platform, but it can also be complex and difficult to manage, especially for small businesses with limited IT resources. Setting up and configuring Google Cloud services may require technical expertise that is not always readily available to small businesses.
2. **Cost:** While Google Cloud's pay-as-you-go pricing model can be cost-effective for many businesses, it can also be unpredictable and difficult to manage for small businesses with limited budgets. Small businesses may struggle to manage their cloud costs effectively, especially if they do not have a clear understanding of their resource usage.
3. **Integration:** Small businesses may already have existing IT systems and applications that they need to integrate with Google Cloud services. This can be a complex and time-consuming process that requires technical expertise.
4. **Security:** While Google Cloud offers a range of security features, small businesses may be more vulnerable to security threats due to their limited

resources and expertise. Small businesses may struggle to implement the necessary security controls and protocols to protect their data and applications effectively.

Google Cloud Platform Services

Google Cloud Platform is continually expanding across the globe. Some of the significant services of GCP are:

- Compute Services
- Networking
- Storage Services
- Big Data
- Security and Identity Management
- Management Tools
- Cloud AI
- IoT (Internet of Things)

Some of the notable users of Google Cloud Services are **Spotify, Airbnb, Twitter, PayPal, Coco Cola, HSBC, The New York Times**

Microsoft Azure:

Microsoft Azure is a cloud computing platform and service offered by Microsoft. It provides a wide range of cloud-based services, including virtual machines, storage, analytics, machine learning, networking, databases, and more. Azure enables users to build, deploy, and manage applications and services on a global network of Microsoft-managed data centres.

Windows Azure was first announced October 28, 2008. A cloud computing operating system which was targeted at Business and Developers without additional coding. The original name Windows Azure was a deliberate response in competition to the Amazon EC2 and Google App Engine. Windows Azure was built from as an extension of the Windows NT which was the beginning of Microsoft Cloud Platform as a Service (PaaS). The Windows Azure was an internal project which went by the code name “Project RedDog”. Recognizing that its cloud computing service had moved far beyond Windows, the company renamed Windows Azure as Microsoft Azure in April 2014.

Components and services of Microsoft Azure

Microsoft Azure uses large-scale virtualization at Microsoft data centres worldwide and it offers more than 600 services. The services offered by Azure are categorized

Computer Engineering Department, S.B.M.Polytechnic, Mumbai.

in different components, below are the key components of Azure along with various services provided under them

1) Compute services

- a. Virtual Machines: Azure provides a wide range of virtual machines that can be used to host various applications and services (IaaS).
- b. Azure App Service: Platform as a service (PaaS) environment letting developers easily publish and manage websites. This service allows developers to build, deploy, and manage web and mobile applications on the cloud.
- c. Azure Kubernetes Service (AKS): This is a fully managed Kubernetes service that simplifies the deployment and management of containerized applications.

2) Azure Functions:

- Azure functions are used in serverless computing architectures where subscribers can execute code as an event driven Function-as-a-Service (FaaS) without managing the underlying server resources.

3) Storage services:

- Azure Storage: Azure provides various storage options to store and manage data on the cloud. Storage Services provides REST and SDK APIs for storing and accessing data on the cloud.

4) Data management:

- Azure SQL Database: This is a fully managed relational database service that enables users to run and scale highly available and secure SQL databases on the cloud.
- Azure Cosmos DB: This is a globally distributed, multi-model database service that provides fast and reliable access to data for applications that require low-latency data access.

5) Identity

- Azure Active Directory: This is a cloud-based identity and access management service that provides secure access to resources and applications.

6) Developer

- Azure DevOps: This is a set of services that enables teams to plan, develop, test, and deploy software applications.

7) Security

- Azure Security Center: This is a unified security management and advanced threat protection service that helps users prevent, detect, and respond to threats on the cloud.

8) Azure AI

- Microsoft Azure Machine Learning (Azure ML) provides tools and ML frameworks for developers to create their own machine learning and artificial intelligence (AI) services.

- Microsoft Azure Cognitive Services are a set of prebuilt APIs, SDKs and customizable services for developers like speech recognition, speaker recognition, face recognition, computer vision, OCR/form understanding, natural language processing and others

9) Azure Orbital

Azure Orbital is a ground station service to help customers move satellite data to the cloud and to provide global cloud connectivity, allowing mobile cloud computing ground stations to provide point-to-point cloud connectivity to remote locations where there is no existing ground infrastructure (such as energy, agricultural and military).

These are just a few examples of the many components offered by Microsoft Azure. It also provides services under Mobile, Communications, Messaging, Media, CDN (Content delivery network (CDN) for audio, video, applications, images, and other static files.), Internet of Things, and Azure blockchain workbench components.

Advantages of Microsoft azure:

With popular cloud services like AWS and Google Cloud offer many similar services and features, there are some advantages to using Microsoft Azure over

The others:

- Familiarity: Many businesses already use Microsoft products, such as Office 365 or Windows, and may find it easier to integrate these products with Azure, rather than with Google Cloud.
- Hybrid cloud: Azure provides a robust hybrid cloud solution, allowing users to seamlessly integrate their on-premises infrastructure with the cloud. This can be especially valuable for businesses that have significant investments in existing infrastructure.
- Enterprise-grade security and compliance: Azure provides built-in security features and compliance with various industry standards and regulations, such as GDPR, HIPAA, and ISO 27001.
- Support for Windows applications: Azure has strong support for Windows-based applications and services, including SQL Server, Active Directory, and .NET.

Disadvantages of Microsoft azure

Potential disadvantages to consider with Microsoft azure:

- Cost: Azure's pricing structure can be complex and difficult to understand, especially for businesses with complex usage patterns or requirements.
- Learning curve: Azure can be complex to set up and configure requiring a significant investment in time and resources to get up to speed.
- Limited support for non-Microsoft tools and platforms: It may not be the best

choice for businesses that rely heavily on non-Microsoft tools or platforms.

- Latency: Azure's global network of data centres may not be as extensive as some other cloud providers, which could lead to increased latency or slower performance for users in certain regions.
- Dependency on Microsoft: Because Azure is a Microsoft product, users may be locked into using other Microsoft products or services, which could limit their flexibility or ability to switch to other platforms in the future.

Major clients of Microsoft azure

Some of the major clients of Microsoft Azure include: Walmart, Coca-Cola, BMW, Samsung, Adobe.

Output:

Students will compare the cloud services and components offered by Google and Azure.

Conclusion:

Questions:

1. Describe any 5 widely used cloud components and services offered by Google.
2. Describe any 5 widely used cloud components and services offered by Azure.
3. Discuss the applications of AWS, Google and Azure cloud services as per the client requirements