SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution. Affiliated to Anna University, Chennai)
Kuniamuthur, Coimbatore - 641 008

DEPARTMENT OF INFORMATION TECHNOLOGY

22AD401 - CLOUD COMPUTING LABORATORY

Continuous Assessment Record

Submitted by

Name :……..……..

Register No :…….……

Degree & Branch :……..

Class & Semester :……

Academic Year :…….……

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution. Affiliated to Anna University, Chennai)
Kuniamuthur, Coimbatore - 641 008

DEPARTMENT OF INFORMATION TECHNOLOGY

22AD401 - CLOUD COMPUTING LABORATORY

Continuous Assessment Record

Submitted by

Name : ……..…….……..…… **Register No. :.....……**

Class/Semester :……. **Degree & Branch :……..**

Faculty In-charge **Head of the Department**

BONAFIDE CERTIFICATE

This is to certify that this record is the bonafide record of work done by Mr./Ms.

_____

During the academic year 2023–2024.

Submitted for the practical examination held on_____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DEPARTMENT OF INFORMATION TECHNOLOGY

22AD401-CLOUD COMPUTING LABORATORY

Record of laboratory

EVEN SEMESTER: 2023-2024

Name of the Faculty	Ms.A.Raihana

CONTINUOUS EVALUATION SHEET

REFERENCES RUBRICS TABLE

Criteria	Range of Marks			
	Excellent	Good	Average	Below Average
Aim (10 marks)	9-10	7-8	5-6	0-4
Procedure (30 Marks)	27-30	21-26	15-20	0-14
Demo (40 Marks)	37-40	31-36	25-30	0-24
Result (10 Marks)	9-10	7-8	5-6	0-4
Viva (10 Marks)	9-10	7-8	5-6	0-4
Overall Marks	**90-100**	**70-89**	**50-69**	**0-49**

INDEX

Ex.No	Name of the Experiment	Page No.
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution. Affiliated to Anna University, Chennai)
Kuniamuthur, Coimbatore - 641 008

DEPARTMENT OF INFORMATION TECHNOLOGY

22AD401-CLOUD COMPUTING LABORATORY

EVEN SEMESTER : 2023-2024

Components	EX 1	EX 2	EX 3	EX 4	EX 5	EX 6	EX 7	EX 8	EX 9	EX 10	EX 11	EX 12	EX 13	EX 14	EX 15
Aim (10 marks)															
Procedure (30 Marks)															
Demo (40 Marks)															
Result (10 Marks)															
Viva (10 Marks)															
TOTAL															
Consolidated Mark(100)															
Faculty Signature															

EX:NO -1

Create an S3 bucket with the following requirements

Create an S3 bucket in the Asia location of 'ap-south-1'(Mumbai).

Create a folder in the name of 'myfolder' in the created bucket.

Upload a file named 'mybiodata.pdf' under the folder 'myfolder'.

Make the object 'mybiodata.pdf' file accessible to everyone(publicly).

Aim:

To create an s3 bucket in ap-south-1 region and to create folder and upload a file to the bucket which can be accessible to everyone.

Description:

Bucket Creation:

1. Login to the AWS console.
2. Search for S3 service in search bar.
3. Click ‘Create bucket’ in S3 service.
4. Select ap-south-1 (Mumbai) AWS Region.
5. Give unique name to the bucket.
6. Enable ACL and change object ownership to Object Writer.
7. Allow public access to the bucket objects.
8. Finally Click ‘Create Bucket’ and your Bucket is created.

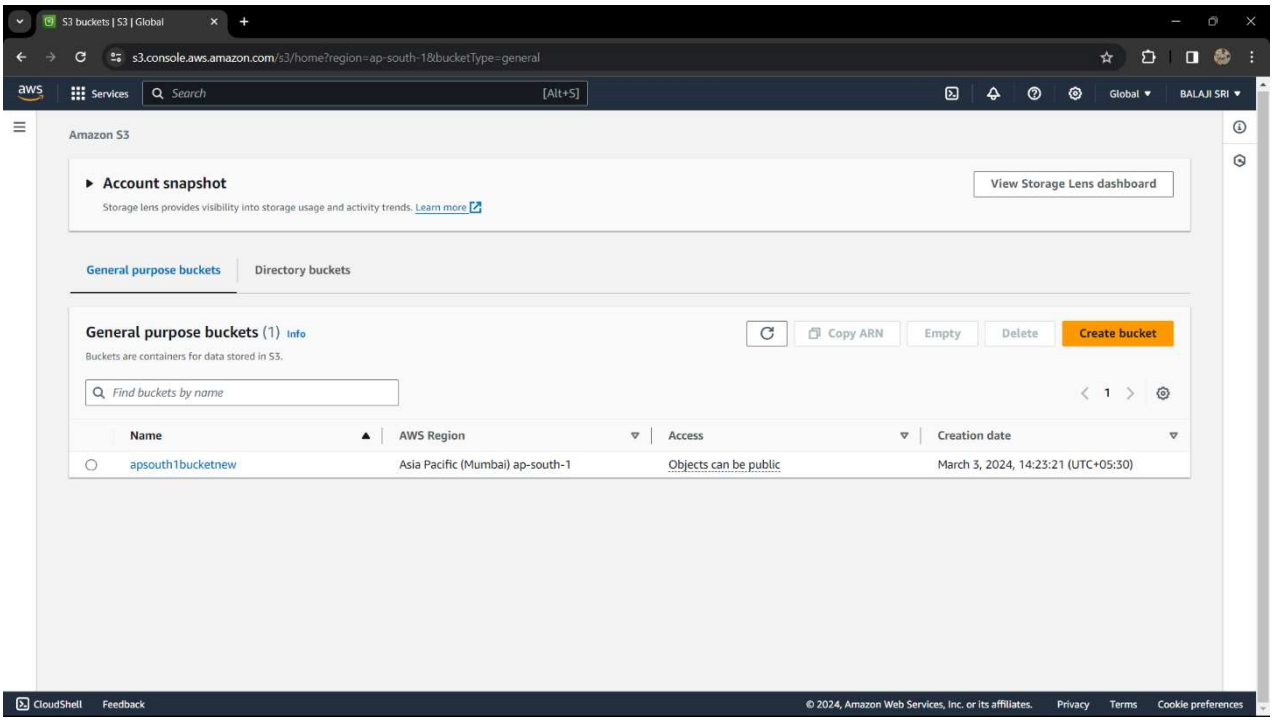
Folder Creation:

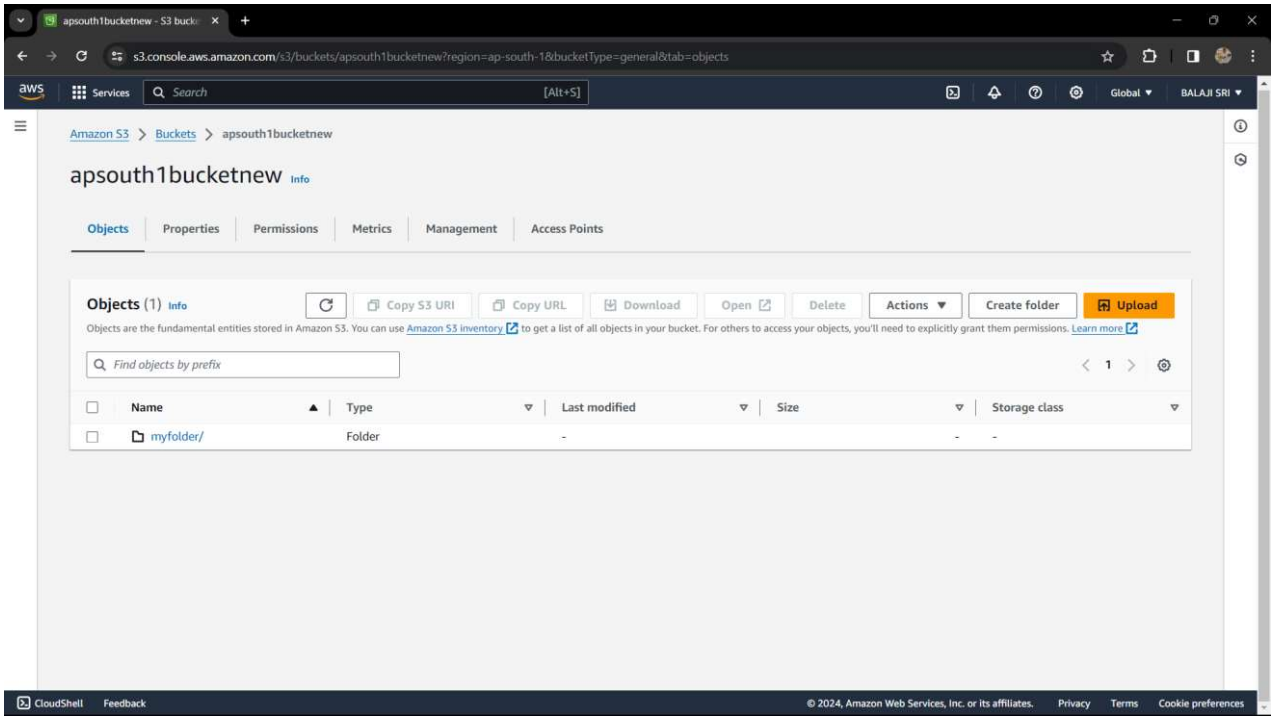
1. Select your bucket from the bucket list.
2. Click Create folder and create a folder with name ‘myfolder’.

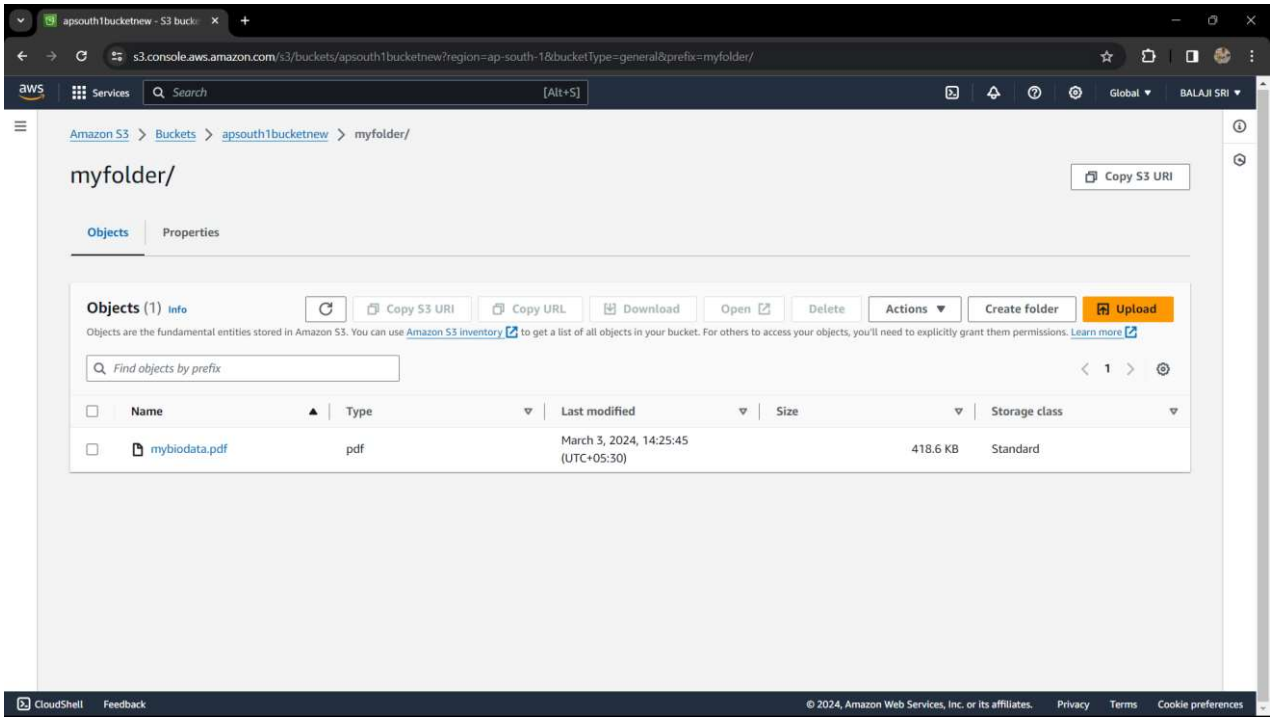
File Upload:

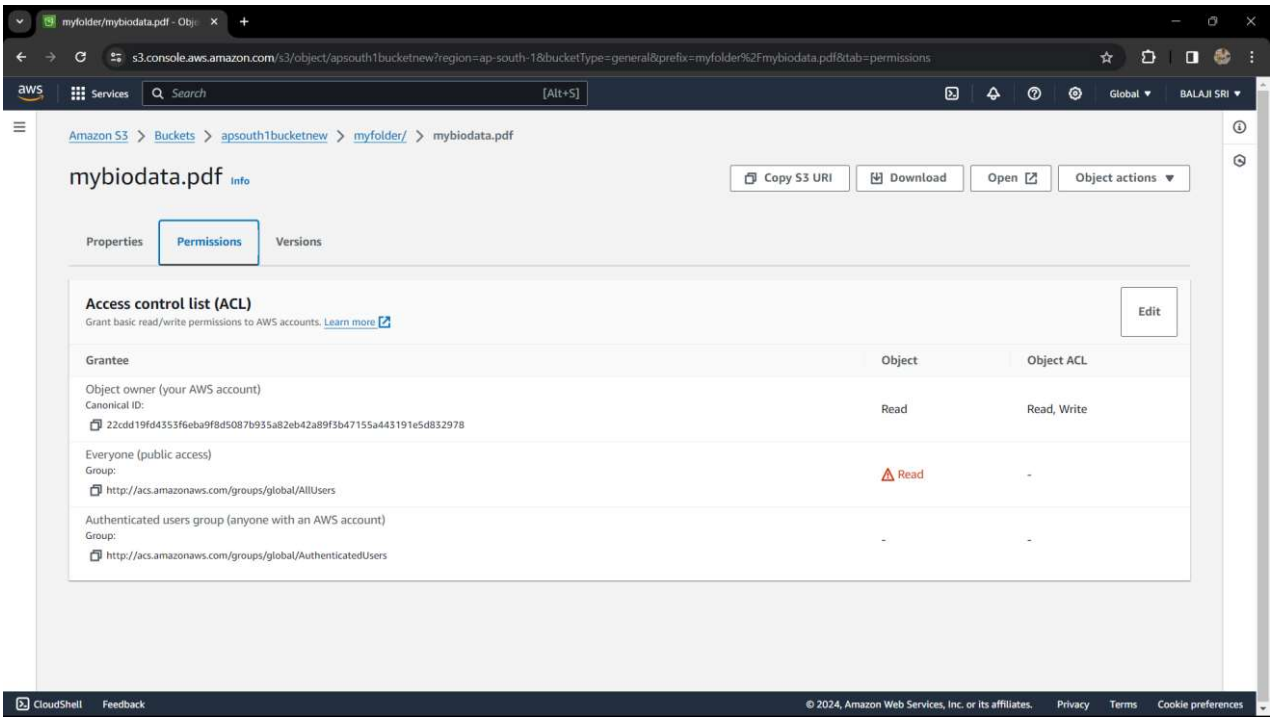
1. Select the folder ‘myfolder’ in your bucket and Click Upload to upload a file.
2. Now upload a pdf file ‘mybiodata’ from your local storage to the folder inside the created S3 bucket.
3. Click on the file inside the folder and go to its properties.
4. In properties edit the accessibility of the object with public access.

Output:


S3 buckets | S3 | Global
s3.console.aws.amazon.com/s3/home?region=ap-south-1&bucketType=general
Services
Search
[Alt+S]
Global
BALAJI SRI
Amazon S3
Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
General purpose buckets
Directory buckets
General purpose buckets (1) Info
Buckets are containers for data stored in S3.
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Name
AWS Region
Access
Creation date
apsouth1bucketnew
Asia Pacific (Mumbai) ap-south-1
Objects can be public
March 3, 2024, 14:23:21 (UTC+05:30)
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


apsouth1bucketnew - S3 bucket
s3.console.aws.amazon.com/s3/buckets/apsouth1bucketnew?region=ap-south-1&bucketType=general&tab=objects
Services
Search
[Alt+S]
Global
BALAJI SRI
Amazon S3 > Buckets > apsouth1bucketnew
apsouth1bucketnew Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (1) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Name
Type
Last modified
Size
Storage class
myfolder/
Folder
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences





Result:

Thus a S3 bucket is created in Mumbai region with a folder having a file with public access.

EX: NO- 2

Create an EC2 Instance in the us-east-1 region with the following requirements.

Give the Name tag of both EC2 instance & keypair as "ec2usecase1"(Name).

EC2 instance AMI should be "Amazon Linux 2".

Allow SSH traffic for taking putty remote connection.

Allow HTTP traffic from the internet for reaching website requests.

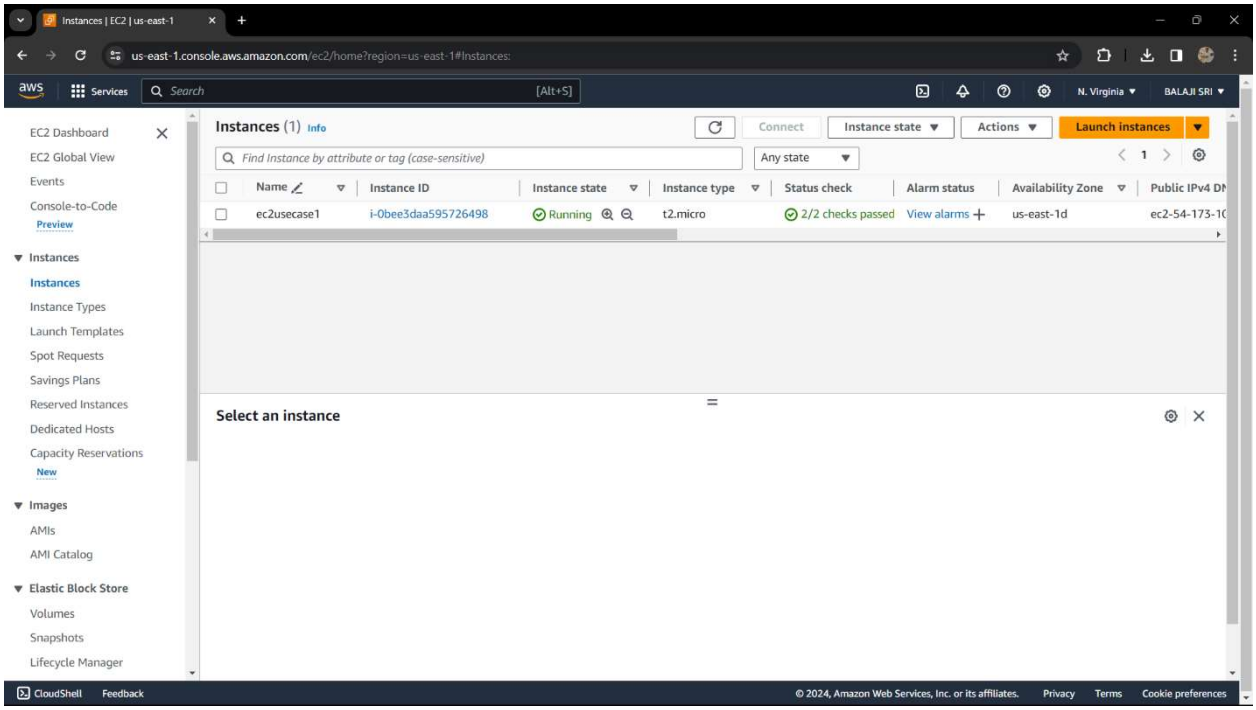
Aim:

To create an EC2 instance of name ‘ec2usecase1’ in the us-east-1 region with a key pair name of ‘ec2usecase1’. EC2 instance AMI should by ‘Amazon Linux’ and the instance should allow SSH and HTTP traffics.

Description:

1. Login to your AWS console.
2. Change your AWS region to us-east-1 (N. Virginia).
3. Search for EC2 service in Search bar.
4. Click ‘Launch Instance’ there.
5. Give a name for your EC2 instance. As given in question, the name of the EC2 instance should be ‘ec2usecase1’.
6. Select ‘Amazon Linux 2023’ as AMI for your instance.
7. Select Instance type. (For free tier account, select t2.micro).
8. Create a new key pair in the name of ‘ec2usecase1’.
9. Allow SSH and HTTP traffic to the instance.
10. If we want to change our root volume for the instance, we can configure it in Configure storage.

Output:



Result:

Thus, an EC2 instance of AMI ‘Amazon Linux’ and name and keypair as ‘ec2usecase1’ which allows SSH and HTTP traffics across the internet has been successfully created.

EX:NO -3

Create an IAM group called 'Network-L1-Team' with 'AmazonVPCReadOnlyAccess' and 'AWSNetworkManagerReadOnlyAccess' policies, then add an IAM user called 'Network-L1-User1' to the group.

The name of the IAM group should be 'Network-L1-Team'.

The name of the IAM user should be 'Network-L1-User1'.

The 'AmazonVPCReadOnlyAccess' policy should be attached.

The 'AWSNetworkManagerReadOnlyAccess' policy should be attached.

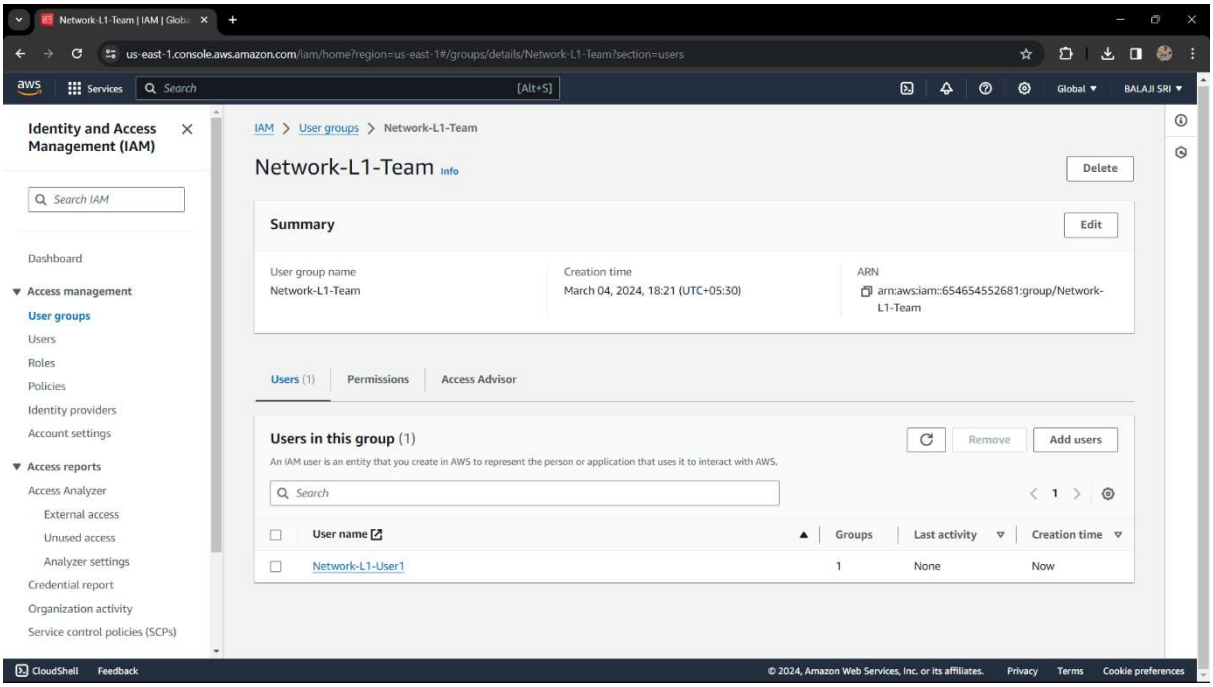
Aim:

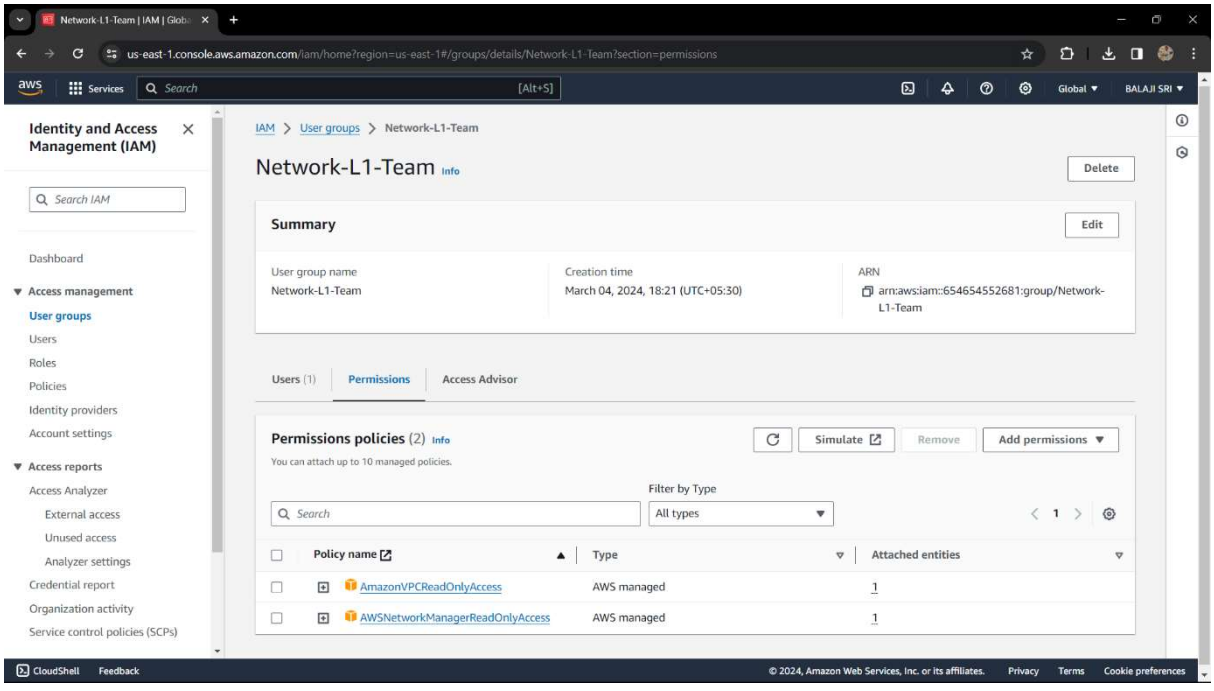
To create an IAM user group named as ‘Network-L1-Team’ with 'AmazonVPCReadOnlyAccess' and 'AWSNetworkManagerReadOnlyAccess' policies, then add an IAM user called 'Network-L1-User1' to the group.

Description:

1. Login to your AWS console.
2. Search for ‘IAM’ service in search box.
3. Click on ‘User groups’ in side bar, to create an IAM user group.
4. Click ‘Create group’.
5. Give ‘Network-L1-Team’ as name to the user group.
6. Attach following two policies to the user group.
    - 'AmazonVPCReadOnlyAccess'
    - ‘AWSNetworkManagerReadOnlyAccess'
7. Click Create Group and your group is created.
8. Click on ‘Users’ in side bar, to create an IAM user.
9. Give ‘Network-L1-User1’ as user name for IAM user, and click next.
10. In permissions options, select ‘Add user to the group’ and select the user group we already created to add our newly created IAM user to that group.
11. Click ‘Create user’ and thus a new IAM user has been created.

Output:





Result:

Thus, a user group with given name and permissions, and a user with given name has been created and the created user has been added to the user group successfully.

EX-NO: 4

Create a VPC with two public subnets as mentioned below, and also create an Internet Gateway and attach it with the created VPC and also add internet route in the Route Table

The VPC 'Name' tag must be "public-vpc" in the region "N. Virginia"

The VPC CIDR range must be "10.10.0.0/16".

The VPC "public-vpc" should have a public subnet with the CIDR "10.10.1.0/24" in

the "us-east-1a" zone

The VPC "public-vpc" should have another public subnet with the CIDR

"10.10.2.0/24" in the "us-east-1b" zone

Internet Gateway should be attached to the VPC "public-vpc"

The "public-vpc" route table should have Internet route entry

Aim:

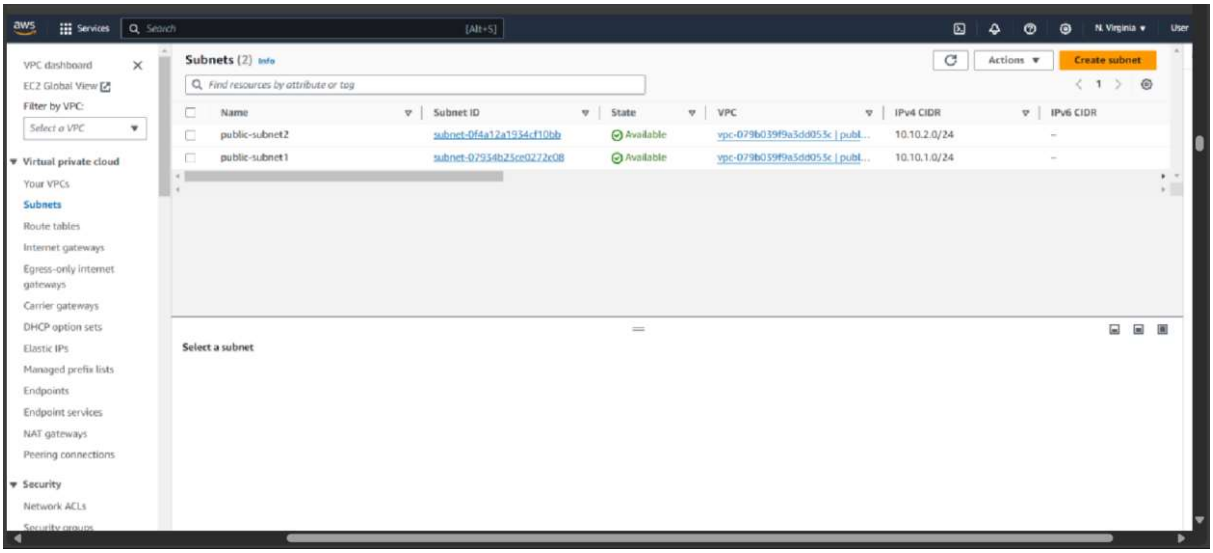
To create a VPC named "public-vpc" in the region "N. Virginia" with given CIDR range and other given requirements.

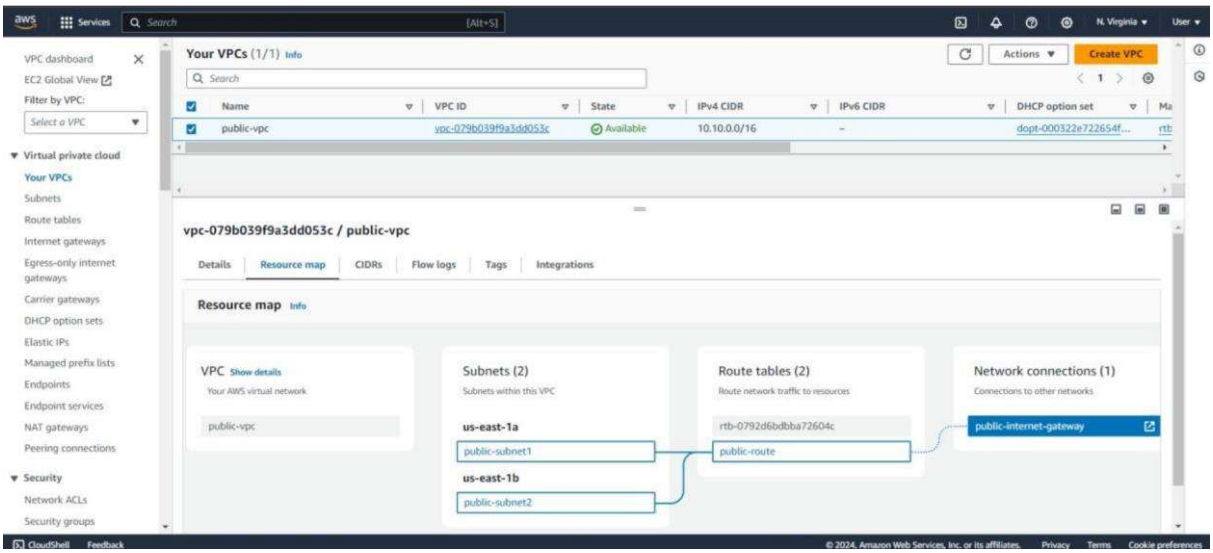
Description:

VPC Creation And its Connection:

a. Login to the AWS console.
b. Search for VPC service in search bar.
c. Click ‘Create VPC’ in create it with the name “public-vpc”
d. Select us-east-1 (N.Virginia) AWS Region.
e. Choose the CIDR Range for VPC as 10.10.0.0/16.
f. Create Public-subnet1 with cidr "10.10.1.0/24" in the "us-east-1a" and Public-subnet2 with cidr "10.10.2.0/24" in the "us-east-1b".
g. Make sure that route table is created for each subnet
h. Choose the number of availability zones to have NAT Gateways
i. Create a VPC endpoints and Enable DNS option.
j. Finally click “Create VPC” and VPC is created
k. Attach Internet Gateway is connected with created VPC using route table

Output :





Result :

Thus VPC named "public-vpc" is created in the region "N. Virginia" with given CIDR range and other given requirements.

EX:NO -5

Configure AWS CloudFront to deliver your website content from an S3 bucket in the ap-northeast-1 region to users in All edge Locations

Create a new S3 bucket in the region of "Tokyo".

Upload a file in the name of 'homepage.html'.

Make the object 'index.html' file accessible to everyone(publicly)

Create a CloudFront distribution with the created S3 Bucket as its Origin Domain.

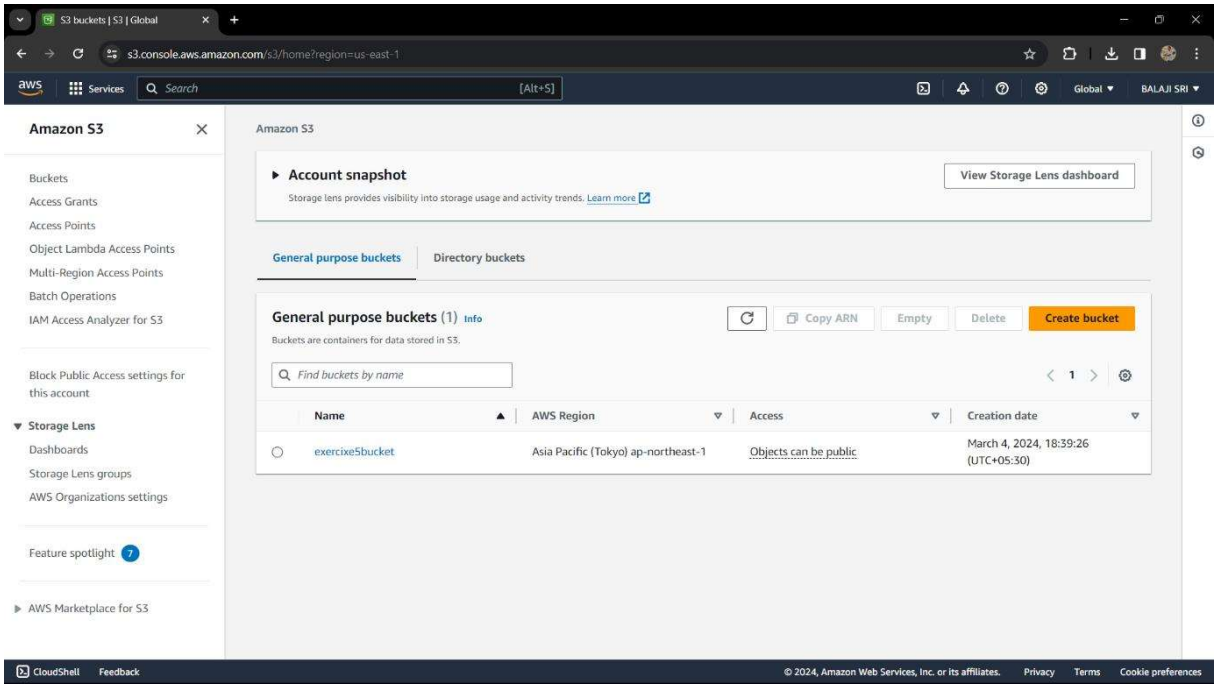
Aim:

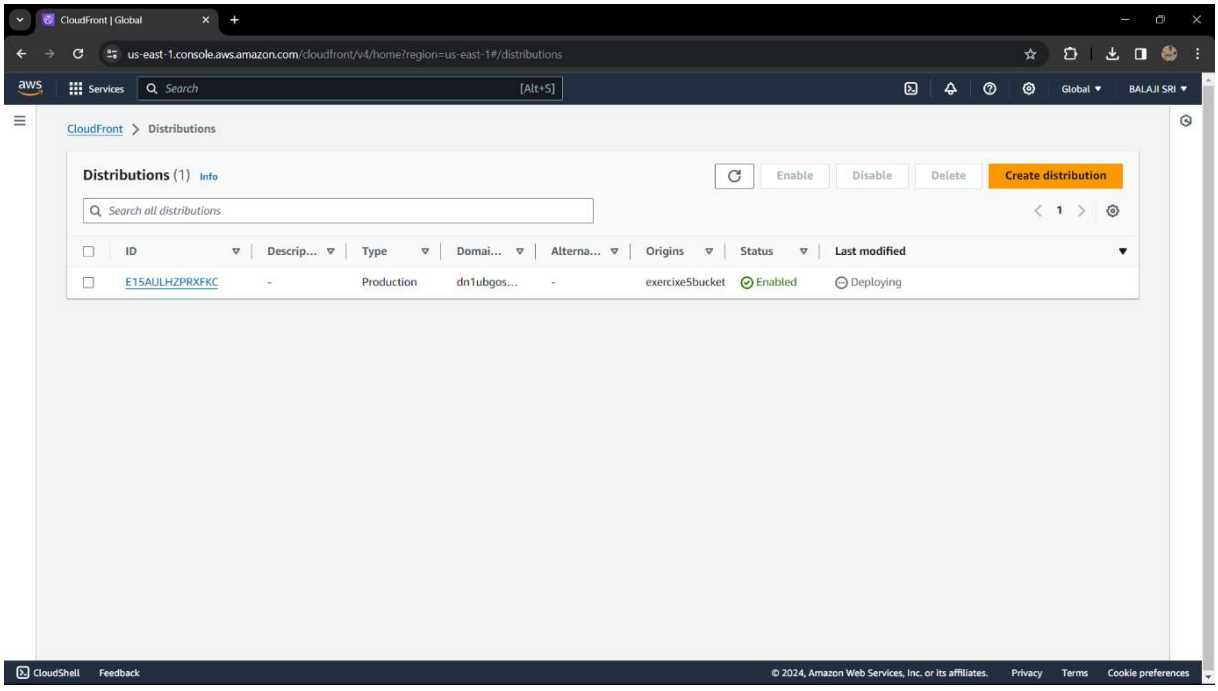
To Configure AWS CloudFront to deliver your website content from an S3 bucket in the ap-northeast-1 region to users in All edge Locations.

Description:

1. Login to your AWS console and search for S3 service in search bar.
2. Click ‘Create bucket’ and select ‘ap-northeast-1’ AWS region.
3. Give a unique name to your bucket.
4. Enable ACL in object ownership and select object writer and object owner.
5. Enable public access to the bucket.
6. Enable Bucket Versioning, if you want.
7. Finally, Click ‘Create Bucket’ to create your S3 bucket.
8. Select the bucket you created and upload a HTML file named ‘homepage’ to the bucket.
9. Before uploading the file, grant public read access to the file.
10. Search for ‘CloudFront’ int search bar and click ‘Create distribution’.
11. Choose your S3 bucket as origin domain.
12. In WAF, do not enable security protections.
13. Finally, click ‘Create Distribution’ to create cloud distribution.

Output:





Result:

Thus, a file is uploaded to a newly created S3 bucket in Tokyo region with public access and a cloud distribution has been created with the created S3 bucket as its origin domain.

EX:NO - 6

Create a public ECR with the name 'docker' in the region 'us-east-1' and push the docker image to the repository.

Create an EC2 instance with the name tag "dockerinstance1" in the "us-east-1" region

EC2 instance AMI should be "Amazon Linux 2023".

Allow HTTP traffic from the instance security group

Install docker in the "dockerinstance1"

Write a dockerfile with httpd as its base image and build the file

Create public repository in ECR with the name "docker"

Push the image to the created repository

The image should have 'latest' as its tag

Aim:

To create a Ec2 instance with Amazon Linux AMI and to enable HTTP traffic and to install docker in the instance and to write a dockerfile with httpd as a base image and to create a public repo in ECR named as Docker and to push the image to it and to view the image with latest tag.

Description:

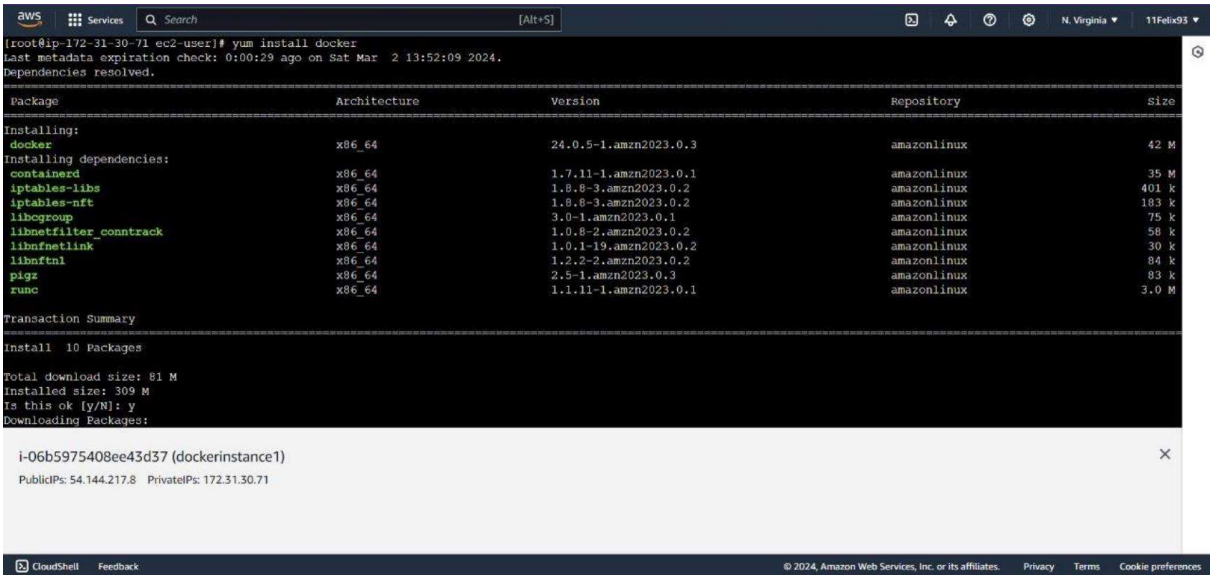
Instance Creation:

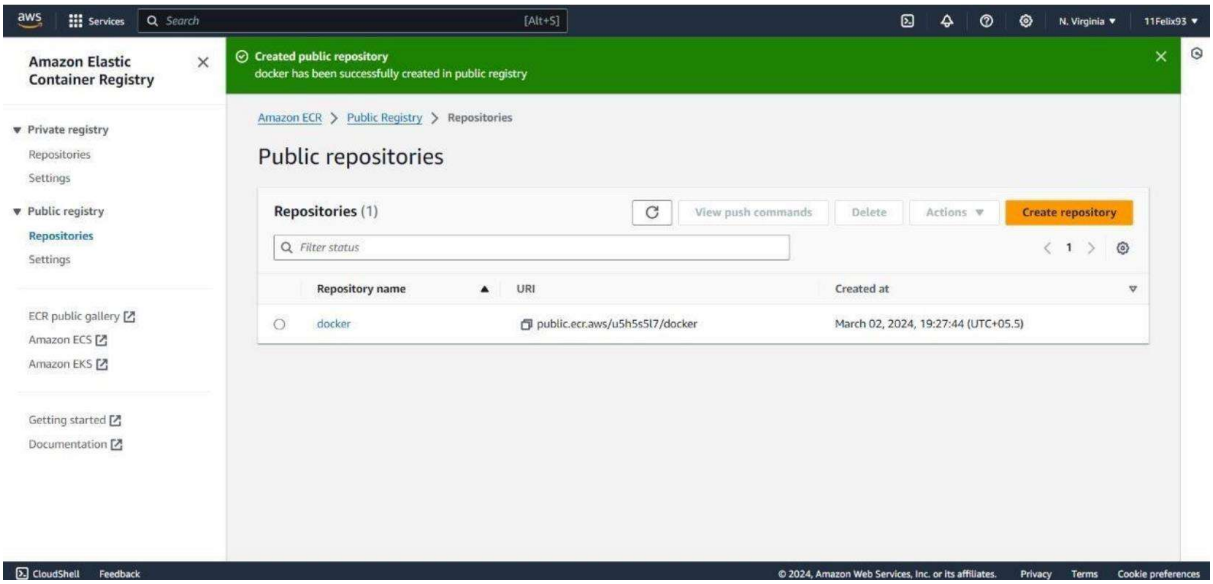
1. Login to the AWS console.
2. Search for EC2 service in search bar and click on “Launch instance”.
3. Select eu-east-1 (Paris) as AWS Region.
4. Give instance name and key pair name as “dockerinstance1”.
5. Choose Amazon Linux as AMI.
6. Enable SSH and HTTP requests in Networking and security groups.
7. Finally Click ‘Launch Instance’ and your Instance is created.

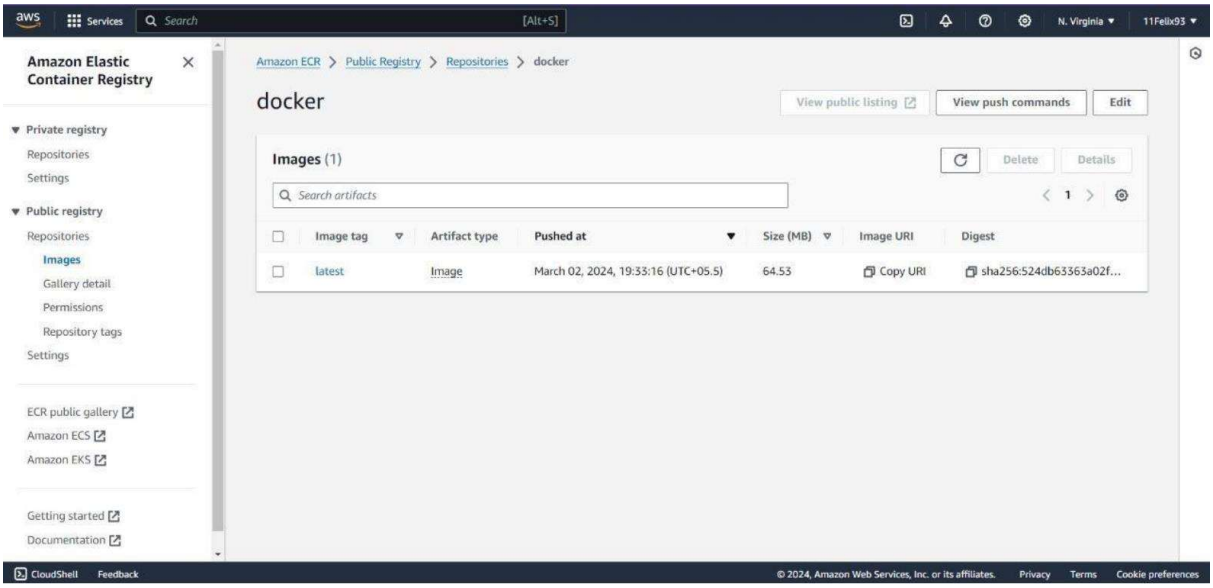
Docker Installation with ECR:

1. Install docker in terminal with basic linux commands.
2. Write a dockerfile with httpd as its base image.
3. Create a public repository in ECR with the name “docker”
4. Push the created docker image in the “docker” repository
5. Make sure the Image have a tag called “latest” in it.

Output :


aws
Services
Search
[Alt+S]
N. Virginia
11Felix93
[root@ip-172-31-30-71 ec2-user]# yum install docker
Last metadata expiration check: 0:00:29 ago on Sat Mar 2 13:52:09 2024.
Dependencies resolved.
Package Architecture Version Repository Size
Installing:
docker x86_64 24.0.5-1.amzn2023.0.3 amazonlinux 42 M
Installing dependencies:
containerd x86_64 1.7.11-1.amzn2023.0.1 amazonlinux 35 M
iptables-libs x86_64 1.8.8-3.amzn2023.0.2 amazonlinux 401 k
iptables-nft x86_64 1.8.8-3.amzn2023.0.2 amazonlinux 183 k
libcgroup x86_64 3.0-1.amzn2023.0.1 amazonlinux 75 k
libnetfilter_conntrack x86_64 1.0.8-2.amzn2023.0.2 amazonlinux 58 k
libnfnetlink x86_64 1.0.1-19.amzn2023.0.2 amazonlinux 30 k
libnftnl x86_64 1.2.2-2.amzn2023.0.2 amazonlinux 84 k
pigz x86_64 2.5-1.amzn2023.0.3 amazonlinux 83 k
runc x86_64 1.1.11-1.amzn2023.0.1 amazonlinux 3.0 M
Transaction Summary
Install 10 Packages
Total download size: 81 M
Installed size: 309 M
Is this ok [y/N]: y
Downloading Packages:
i-06b5975408ee43d37 (dockerinstance1)
PublicIPs: 54.144.217.8 PrivateIPs: 172.31.30.71
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Services
Search
[Alt+S]
N. Virginia
11Felix93
Amazon Elastic Container Registry
Created public repository
docker has been successfully created in public registry
Amazon ECR > Public Registry > Repositories
Public repositories
Private registry
Repositories
Settings
Public registry
Repositories
Settings
ECR public gallery
Amazon ECS
Amazon EKS
Getting started
Documentation
Repositories (1)
View push commands
Delete
Actions
Create repository
Filter status
1
Repository name
URI
Created at
docker
public.ecr.aws/u5h5s5l7/docker
March 02, 2024, 19:27:44 (UTC+05.5)
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



Result :

Thus, EC2 instance with Amazon Linux AMI is created and HTTP traffic is enabled and docker is installed and written with httpd as a base image and the image with latest tag is pushed into a public repository called “docker”.

EX:NO - 7

You are working on a project that involves developing and deploying a web application using containers on a cloud-based platform. You have been asked to create and manage the containers and container images for this project. You are using Docker as your containerization technology.

Evaluate the process for creating and removing containers and container images in Docker.

Aim :

To evaluate the process for creating and removing containers and container images in docker.

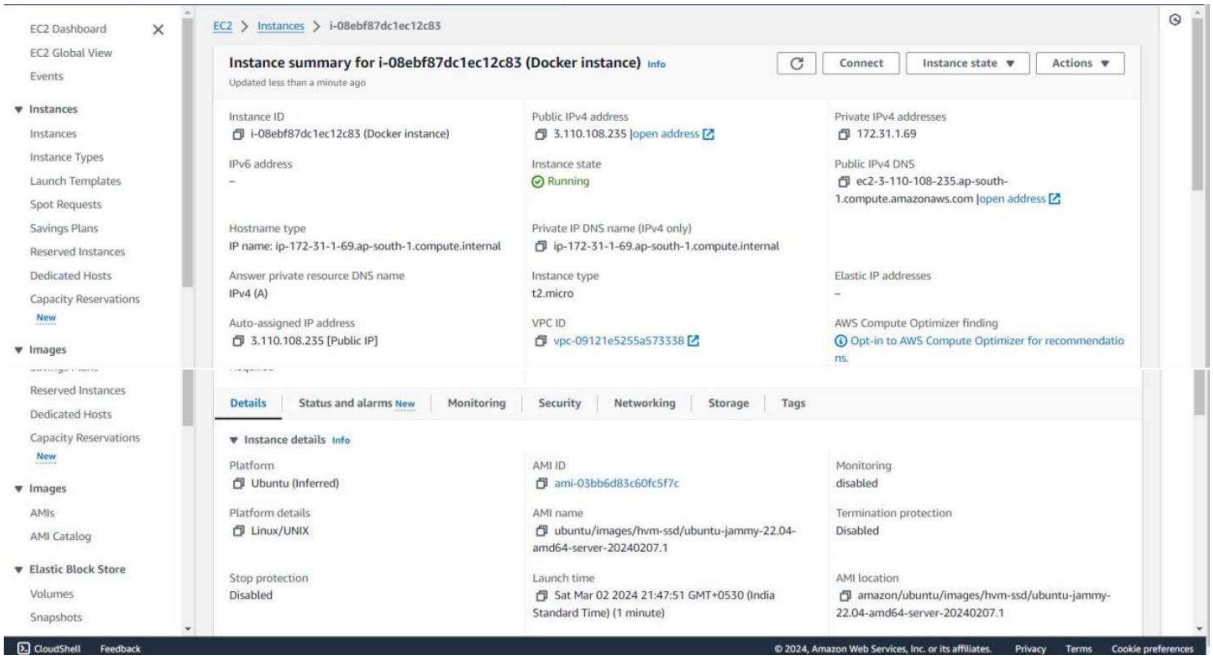
Description :

Docker is an open-source project that uses several resource-isolation features of the Linux kernel to sandbox an application, its dependencies, configuration files, and interfaces inside of an atomic unit called a container. This allows a container to run on any host with the appropriate kernel components, while shielding the application from behavioral inconsistencies due to variances in software installed on the host

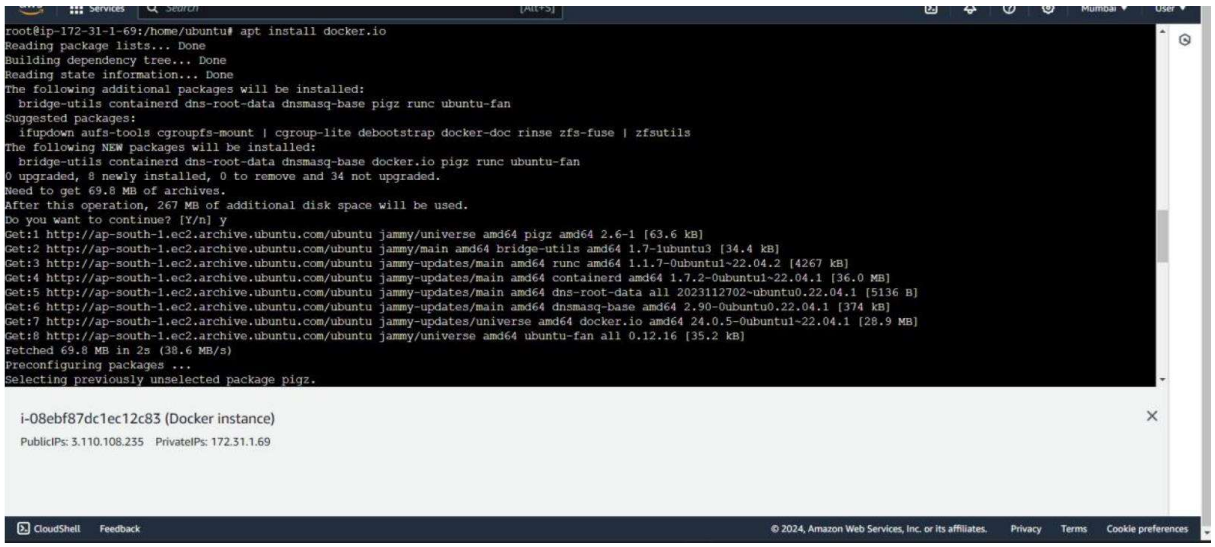
There are many ways in which using containers on AWS can benefit your organization. Docker has been widely employed in use cases such as distributed applications, batch jobs, continuous deployment pipelines, and etc. The use cases for Docker continue to grow in areas like distributed data processing, machine learning, streaming media delivery and genomics

Process :

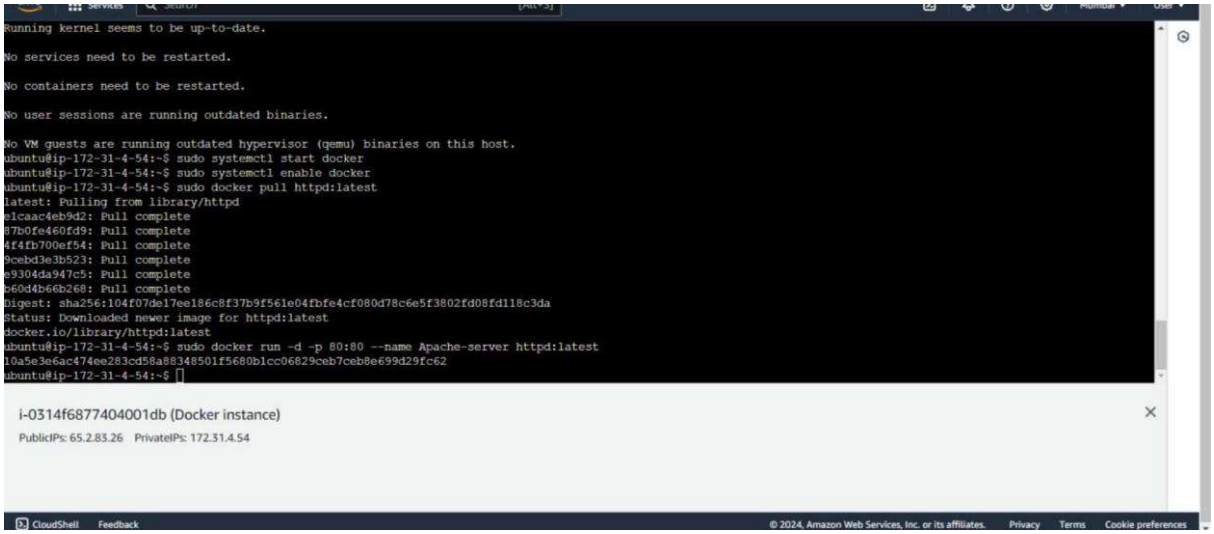
1. Creating a EC2 instance with Ubuntu AMI named as “Dockerinstance”



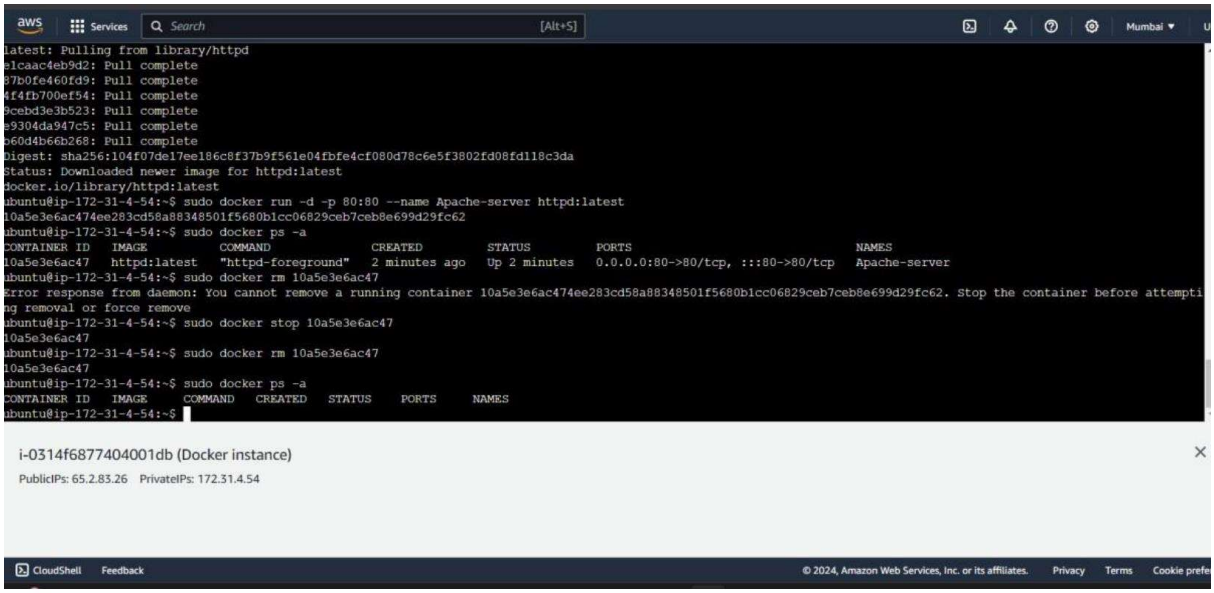
2. Loging in and installing Docker in the instance



3. Building and Running the image in docker



4. Stopping and Removing the Running Container



Result :

Thus, the process for creating and removing containers and container images in docker is evaluated.

EX:NO -8

Create a three VPC in the region ca-central-1 and enable peer to peer connections with the following requirements.

Requirements:

Name should be vpc-1, vpc-2 and vpc-3 respectively.

CIDR ranges should be 10.0.0.0/16 ,11.0.0.0/16 and 12.0.0.0/16 for the above created VPCs.

Create a one public subnet in each.

Connect vpc-1, vpc-2 as first connection.

Connect vpc-2, vpc-3 as second connection.

Connect vpc-3, vpc-1 as third connection.

Aim:

To Create a three VPC in the region ca-central-1 and enable peer to peer connections with the given requirements

Description:

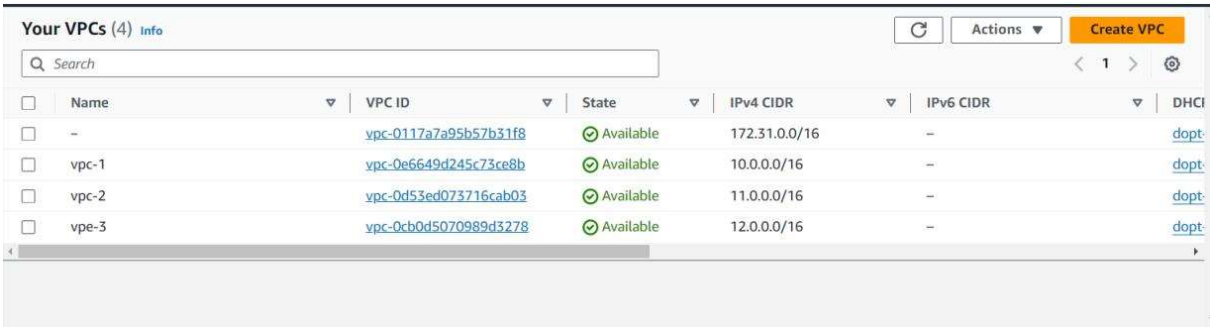
VPC Creation:

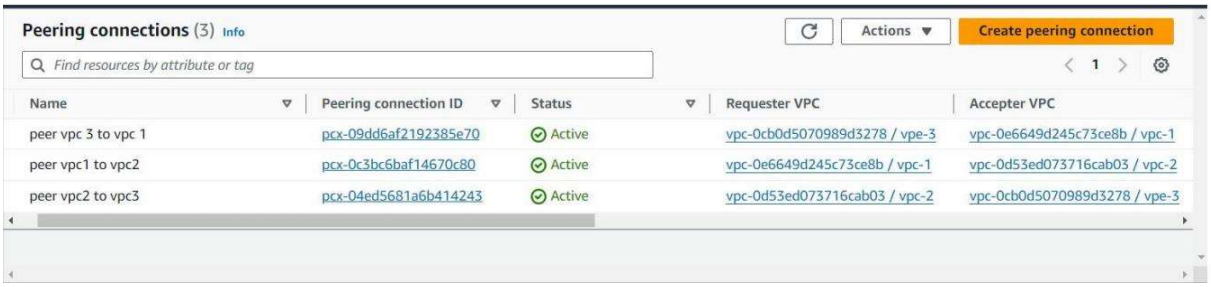
1. Login to the AWS console.
2. Click 'Create VPC' in VPC service for three specific VPC's.
3. Select ca-central-1 as AWS Region.
4. Give its name as "vpc-1", "vpc-2", "vpc-3" respectively.
5. Choose the CIDR Range as 10.0.0.0/16 ,11.0.0.0/16 and 12.0.0.0/16 respectively.
6. Create 1 subnet for each VPC.
7. Finally click "Create VPC" and VPC is created.

VPC Peering Connection :

1. Create a VPC peering connection
2. Connect vpc-1, vpc-2 as first connection.
3. Connect vpc-2, vpc-3 as second connection.
4. Connect vpc-3, vpc-1 as third connection.
5. Finally Create and establish the peering between VPC's.

Output :





Result :

Three VPC in the region ca-central-1 is created and enabled for peer to peer connections with the given requirements

EX:NO - 9

Create a docker container with the following requirements.

Requirements:

Give tag name and key name as “ubuntu-docker”.

EC2 instance AMI should be “ubuntu”.

Allow SSH for SSH client for instance connection.

Allow HTTP traffic from the internet for reaching website requests.

Install docker.

Pull docker Apache image from docker hub.

Docker container name should be “Apache-server” and run docker in 8080 port, forward it to 80;

Aim:

To create a Docker container with the given requirements.

Description:

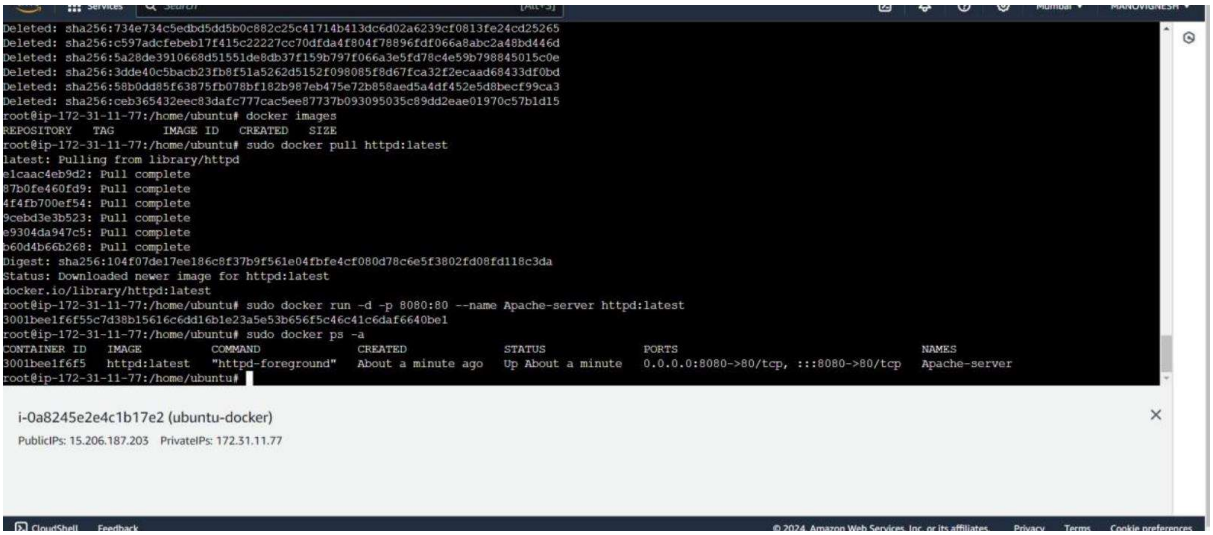
Instance Creation:

1. Login to the AWS console.
2. Search for EC2 service in search bar and click on “Launch instance”.
3. Select eu-east-1 (Paris) as AWS Region.
4. Give instance name and key pair name as “ubuntudocker”.
5. Choose ubuntu as AMI.
6. Enable SSH and HTTP requests in Networking and security groups.
7. Finally Click ‘Launch Instance’ and your Instance is created.

Docker Installation:

1. Install docker in terminal with basic linux commands.
2. Write a dockerfile with apache as its base image.
3. Pull the Apache server from dockerhub
4. Docker container name should be “Apache-server” and run docker in 8080 port and forward it to 80.

Output :



Result :

Thus, the docker is created with the given requirements.

EX:NO - 10

Create a docker container with the following requirements.

Requirements:

Tag name and key name: "ubuntu-docker".

EC2 instance AMI: Ubuntu.

Allow SSH for SSH client for instance connection.

Allow HTTPS traffic from the internet for secure connections.

Install Docker.

Build Docker image for Apache2 server with index.html using Dockerfile.

Run Docker container from the built image.

Aim:

To create a Docker container with the given requirements.

Description:

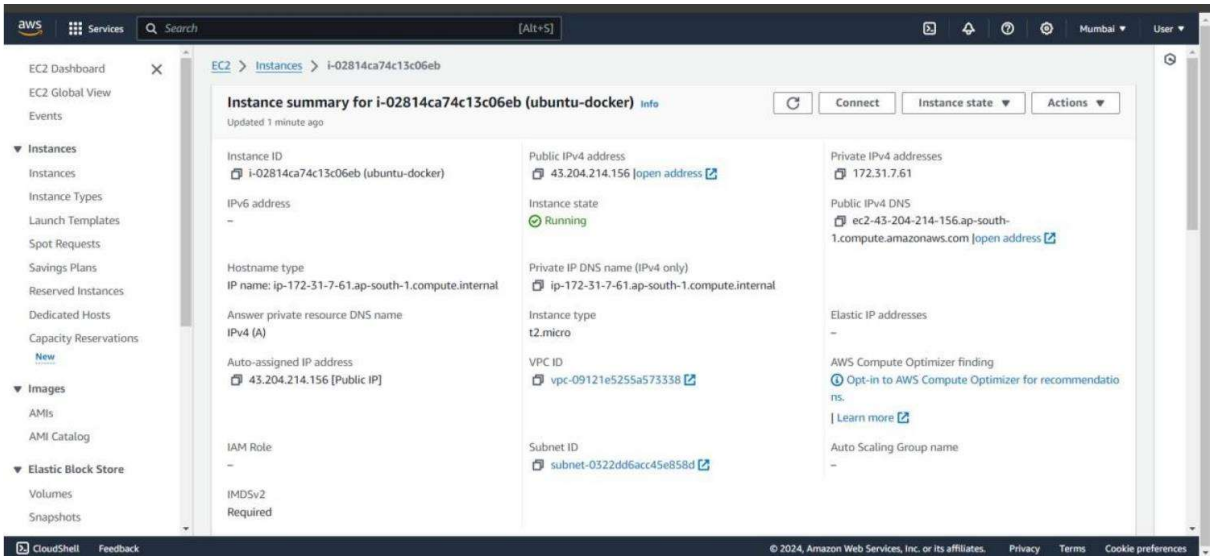
Instance Creation:

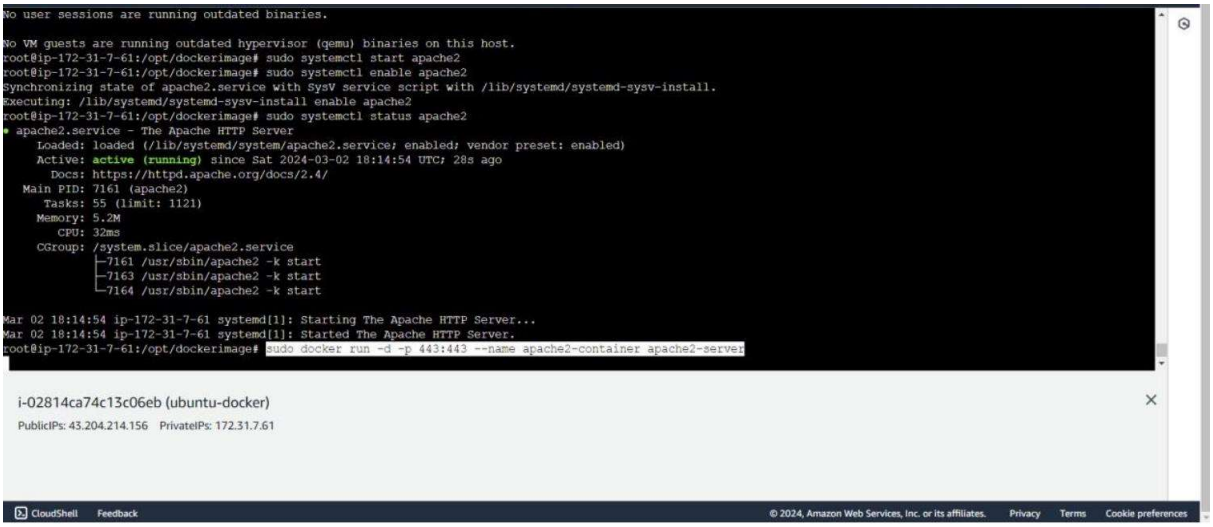
1. Login to the AWS console.
2. Search for EC2 service in search bar and click on “Launch instance”.
3. Select eu-east-1 (Paris) as AWS Region.
4. Give instance name and key pair name as “ubuntu-docker”.
5. Choose ubuntu as AMI.
6. Enable SSH and HTTP requests in Networking and security groups.
7. Finally Click ‘Launch Instance’ and your Instance is created.

Docker Installation:

1. Install docker in terminal with basic linux commands.
2. Write a dockerfile with apache as its base image.
3. Build docker image for Apache server using Dockerfile.
4. Run the docker container with built image.

Output :





Result :

Thus, the docker is created with the given requirements.

EX:NO -11

Create a EC2 instance (ap-south-1) region, attach the volume and take snapshot (EBS) with the following requirements.

Give the name tag of both ec2 instance and keypair as “ap-ec2”.

Ec2 instance AMI should be Amazon Linux 2.

Create a EBS volume of 5 Gb and attach the of the ap-ec2.

Take snapshot of the EBS volume attached.

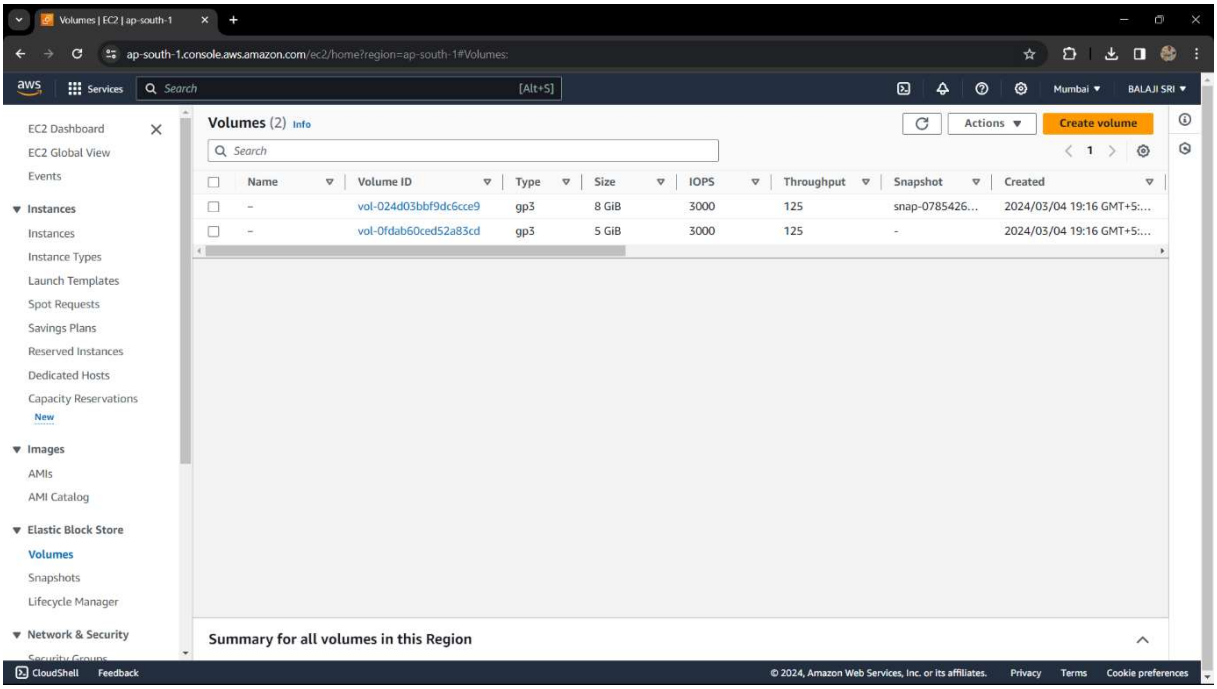
Aim:

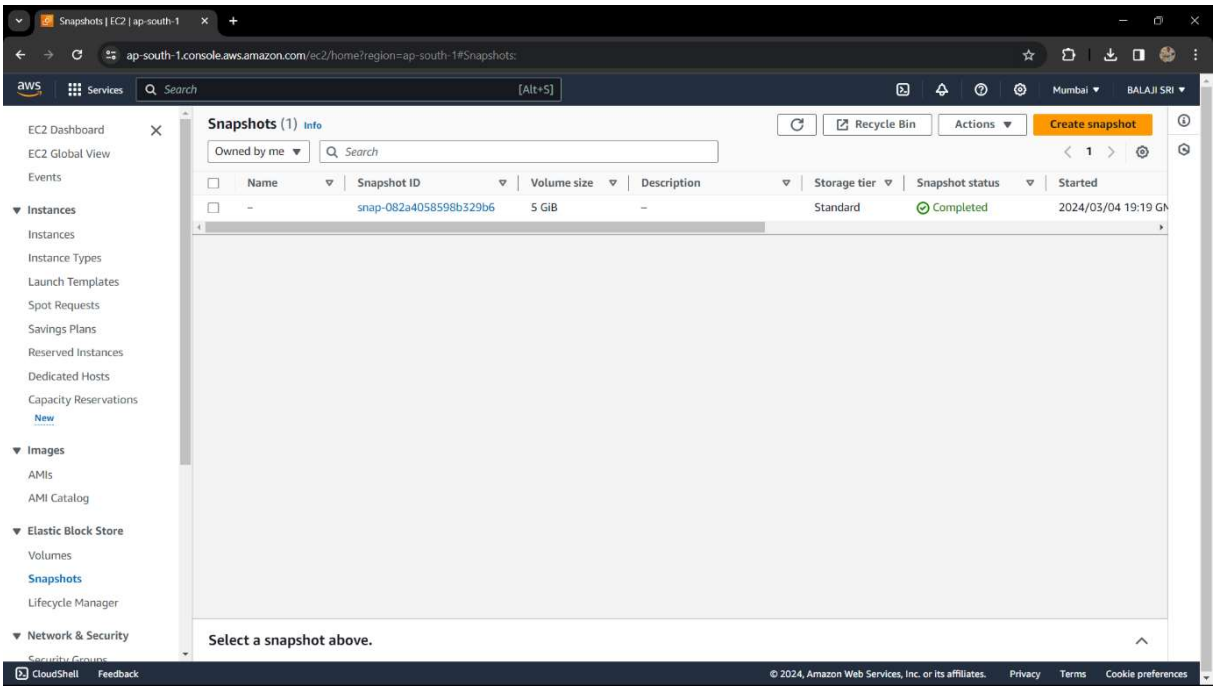
To create an EC2 instance int ap-south-1 region and to attach the volume and take snapshot with the given requirements.

Description:

1. Login to your AWS console and Search for EC2 service in search bar.
2. Change your AWS region to ap-south-1 (Mumbai).
3. Click ‘Launch Instances’ to create a new instance.
4. Give name of EC2 instance as ‘ap-ec2’.
5. Select ‘Amazon Linux 2023’ as AMI for your EC2 instance.
6. Select ‘t2.micro’ as your instance type. (only instance type available for free tier account).
7. Create a keypair with the name of ‘ap-ec2’.
8. Allow SSH, HTTP traffic in network to your EC2 instance.
9. In configure storage, Add new EBS volume of size 5 GB.
10. Finally, Click ‘Launch Instance’ to create your instance.
11. In side bar, you can find ‘Snapshots’. Click on that and Create Snapshot.
12. Select resource type as ‘Volume’.
13. Select your EBS volume ID and thus finally create snapshot.

Output:





Result:

Thus an EC2 instance has been created in Mumbai region with an Elastic Block Store of size 5GB and the snapshot for that EBS volume has been successfully taken.

EX:NO -12

Create EC2 windows instance in eu-west-3 region with given requirements.

Requirements:

Give the name tag of both ec2 instance and key pair as “windows-server”.

Enable http and RDP in security group.

Login into window server using RDP.

Enable IIS and show the IIS page output with the help of public IP.

Aim:

To create an ec2 instance in eu-west-3 region and enabling IIS to show the IIS page output with help of public IP.

Description:

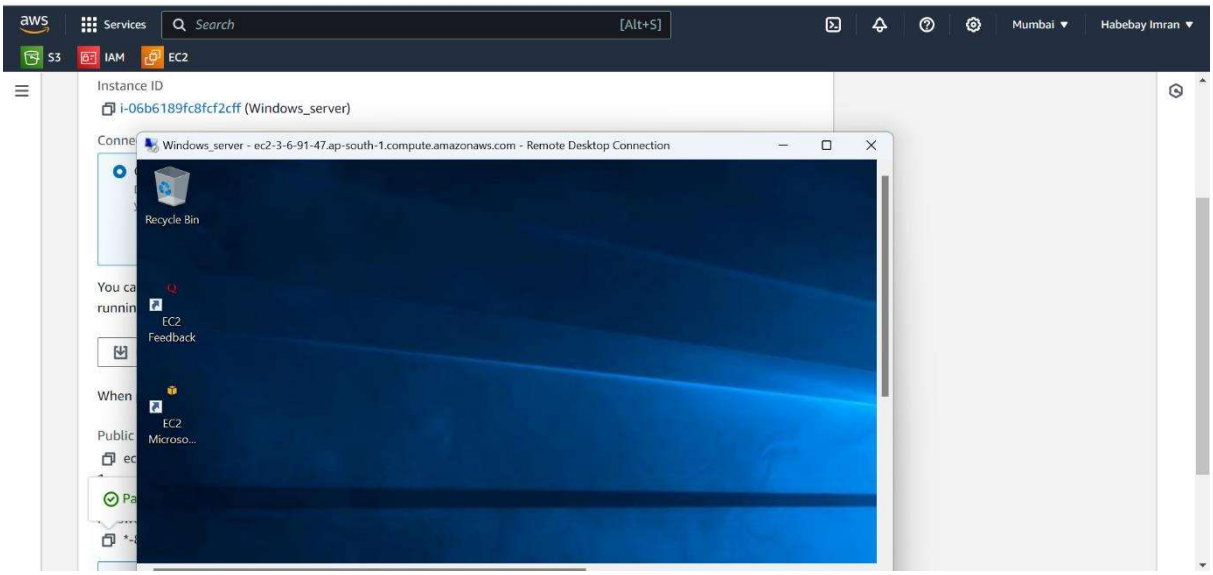
Instance Creation:

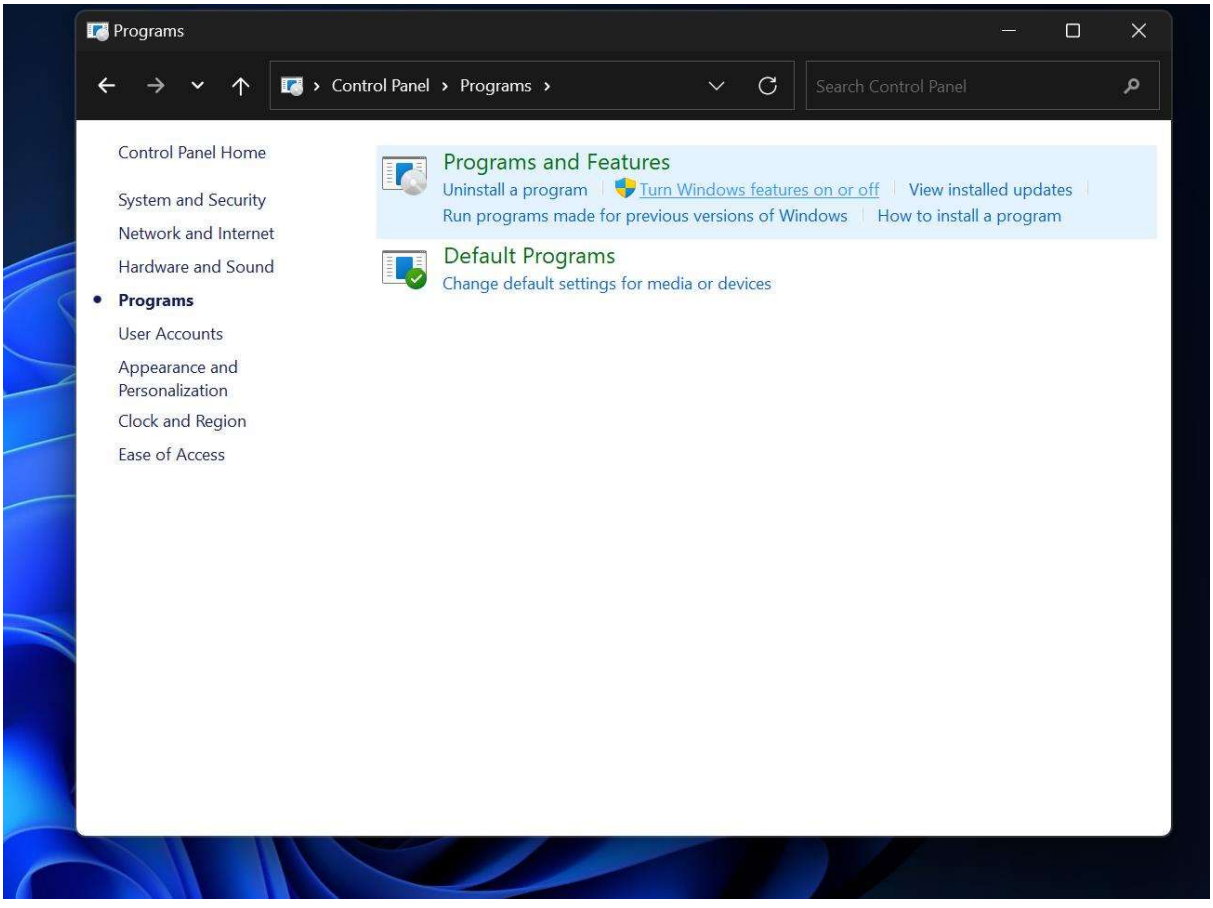
1. Login to the AWS console.
2. Search for EC2 service in search bar.
3. In Instance section, Click ‘Launch Instance’.
4. Select eu-west-3 (Paris) as AWS Region.
5. Give instance name as “windows-server”.
6. Choose Windows as AMI.
7. Choose t2-micro or t3-micro as instance type.
8. Enable RDP and HTTP requests in Networking and security groups.
9. Finally Click ‘Launch Instance’ and your Instance is created.

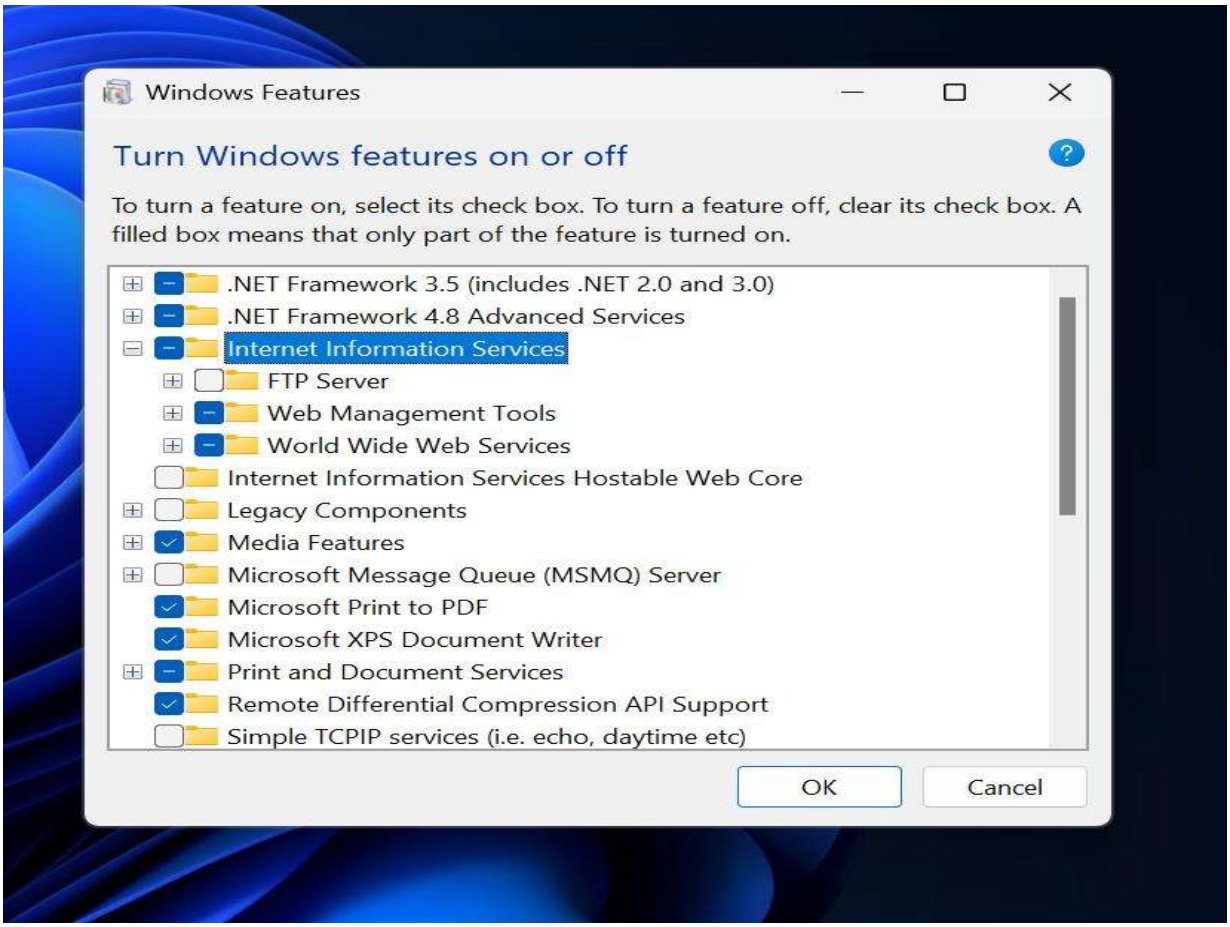
Instance Login and IIS Access:

1. Connect the instance using RDP client by downloading file and decrypting the password
2. Enter the password to open instance window
3. In Instance Window, Go to Programs In Control Panel.
4. Select “Turn Windows Features On or Off”
5. Search For IIM and Enable it.

Output:


aws
Services
Search
[Alt+S]
Mumbai
Habebay Imran
S3
IAM
EC2
Instance ID
i-06b6189fc8fcf2cff (Windows_server)
Windows_server - ec2-3-6-91-47.ap-south-1.compute.amazonaws.com - Remote Desktop Connection
Recycle Bin
EC2 Feedback
EC2 Microso...


Programs
Control Panel › Programs ›
Search Control Panel
Control Panel Home
System and Security
Network and Internet
Hardware and Sound
Programs
User Accounts
Appearance and Personalization
Clock and Region
Ease of Access
Programs and Features
Uninstall a program
Turn Windows features on or off
View installed updates
Run programs made for previous versions of Windows
How to install a program
Default Programs
Change default settings for media or devices



Result:

Thus an EC2 instance is created in eu-west-3 region and logged in using instance and IIS is enabled.

EX:NO -13

Create an EC2 Amazon Linux instance in (ap-northeast-1) Tokyo and launch another instance using the backup AMI of Japanese server with the following requirements:

Requirements:

Give the name tag of both ec2 instance and keypair as " japanese-server".

Enable ssh and http in security group.

Create an image backup from running server with the name (server-backup).

Launch another server using the created AMI backup.

Aim:

To create an ec2 instance in ap-northeast-1 region and launching another instance using the backup AMI of first instance.

Description:

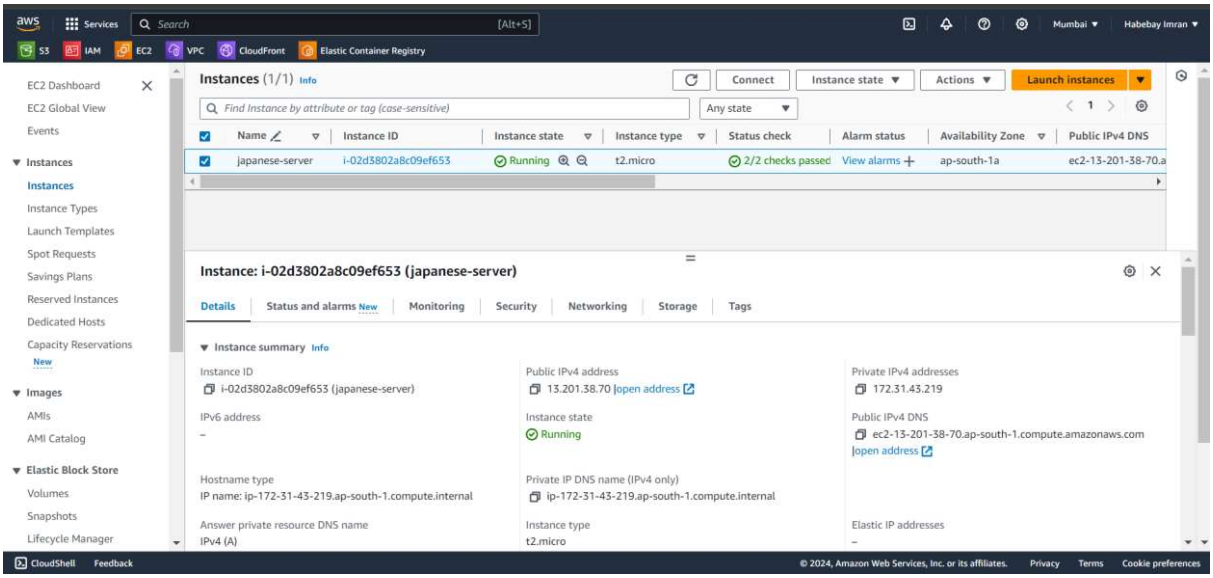
Instance Creation With Amazon Linux:

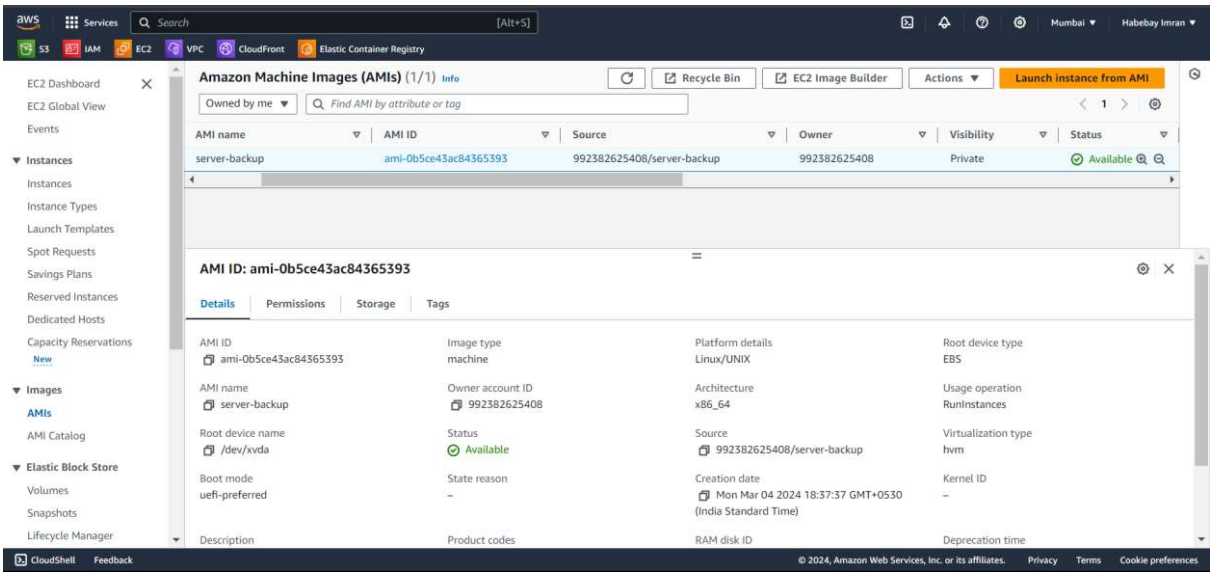
1. Login to the AWS console.
2. Search for EC2 service in search bar.
3. In Instance section, Click 'Launch Instance'.
4. Select ap-northeast-1 (Tokyo) as AWS Region.
5. Give instance name as "japanese-server".
6. Choose Amazon as AMI.
7. Choose t2-micro or t3-micro as instance type.
8. Create a key pair in the same name as instance
9. Enable SSH and HTTP requests in Networking and security groups.
10. Finally Click 'Launch Instance' and your Instance is created.

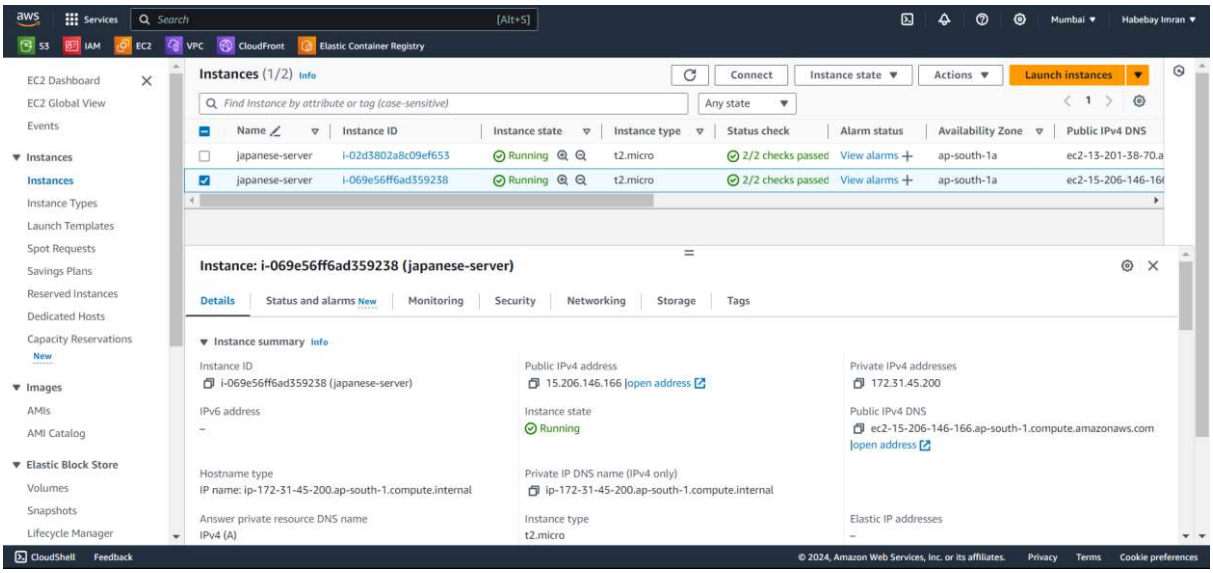
Instance Creation With AMI Backup:

1. Click on Actions in the created instance and create a image in the name "server-backup"
2. Launch new instance with same name.
3. Choose the Backup AMI for AMI
4. Apply the same properties as first instance.
5. Finally Click 'Launch Instance' and your Instance is created.

Output:


Instances (1/1) Info
Connect
Instance state
Actions
Launch instances
japanese-server
i-02d3802a8c09ef653
Running
t2.micro
2/2 checks passed
ap-south-1a
Instance: i-02d3802a8c09ef653 (japanese-server)
Details
Status and alarms New
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-02d3802a8c09ef653 (japanese-server)
Public IPv4 address
13.201.38.70 |open address
Private IPv4 addresses
172.31.43.219
IPv6 address
Instance state
Running
Public IPv4 DNS
ec2-13-201-38-70.ap-south-1.compute.amazonaws.com
|open address
Hostname type
IP name: ip-172-31-43-219.ap-south-1.compute.internal
Private IP DNS name (IPv4 only)
ip-172-31-43-219.ap-south-1.compute.internal
Answer private resource DNS name
IPv4 (A)
Instance type
t2.micro
Elastic IP addresses


Amazon Machine Images (AMIs) (1/1) Info
Recycle Bin
EC2 Image Builder
Actions
Launch instance from AMI
Owned by me
server-backup
ami-0b5ce43ac84365393
992382625408/server-backup
992382625408
Private
Available
AMI ID: ami-0b5ce43ac84365393
Details
Permissions
Storage
Tags
AMI ID
ami-0b5ce43ac84365393
Image type
machine
Platform details
Linux/UNIX
Root device type
EBS
AMI name
server-backup
Owner account ID
992382625408
Architecture
x86_64
Usage operation
RunInstances
Root device name
/dev/xvda
Status
Available
Source
992382625408/server-backup
Virtualization type
hvm
Boot mode
uefi-preferred
State reason
Creation date
Mon Mar 04 2024 18:37:37 GMT+0530
(India Standard Time)
Kernel ID
Description
Product codes
RAM disk ID
Deprecation time



Result:

Thus an EC2 instance is created in ap-northeast-1 region and an another EC2 instance is created with the backup AMI of first instance

EX-NO: 14

Create a VPC infrastructure with the following requirements.

Requirements:

Choose “ap-south-1” as VPC region and VPC name should be “vpc-south”

Your VPC must have 65,536 IP with the starting IP of10.0.0.0.

VPC should contains at least one private subnet in each zone.

Create routing table, each private subnet should have separate routing table.

And connect it with the internet using NAT gateway.

Aim:

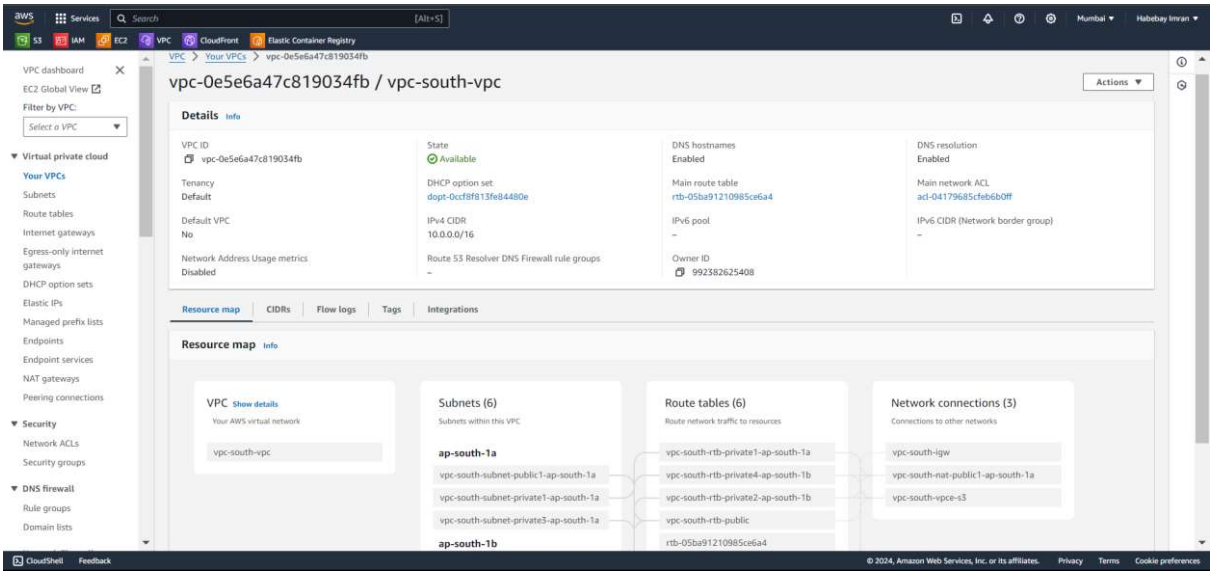
To create an VPC in ap-south-1 region with the given requirements

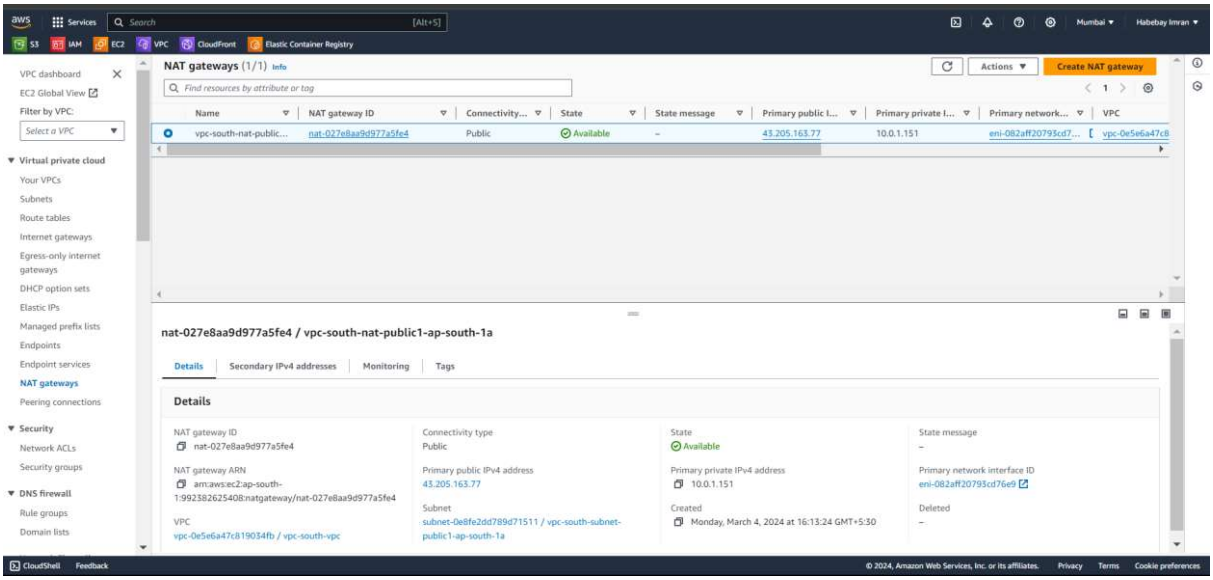
Description:

VPC Creation And NAT Gateway Connection:

a. Login to the AWS console.
b. Search for VPC service in search bar.
c. Click ‘Create VPC’ in VPC service.
d. Select ap-south-1 (Mumbai) AWS Region.
e. Give “vpc-south” as its name.
f. Choose the CIDR Range as 10.0.0.0/16 with 65,536 IP’s.
g. Choose the number of availability zone to provision subnets.
h. Choose the number of public and private subnets.
i. Make sure that route table is created foe each private subnet
j. Choose the number of availability zones to have NAT Gateways
k. Create a VPC endpoints and Enable DNS option.
l. Finally click “Create VPC” and VPC is created
m. Make sure that NAT Gateway is connected with created VPC.

Output:





Result:

Thus VPC is created in ap-south-1 region with a specified CIDR range and having specific route table for private subnets and connected to internet using NAT Gateway.

EX-NO: 15

Create a VPC in the region ap-northeast-1 with the following requirements.

Requirements:

Choose “ap-northeast-1” as VPC region and VPC name should be “vpc-northeast”.

Your VPC must have 12.0.0.0/14 CIDR range.

Your VPC must have 1 public subnets and 2 private subnets with the proper IP CIDR range.

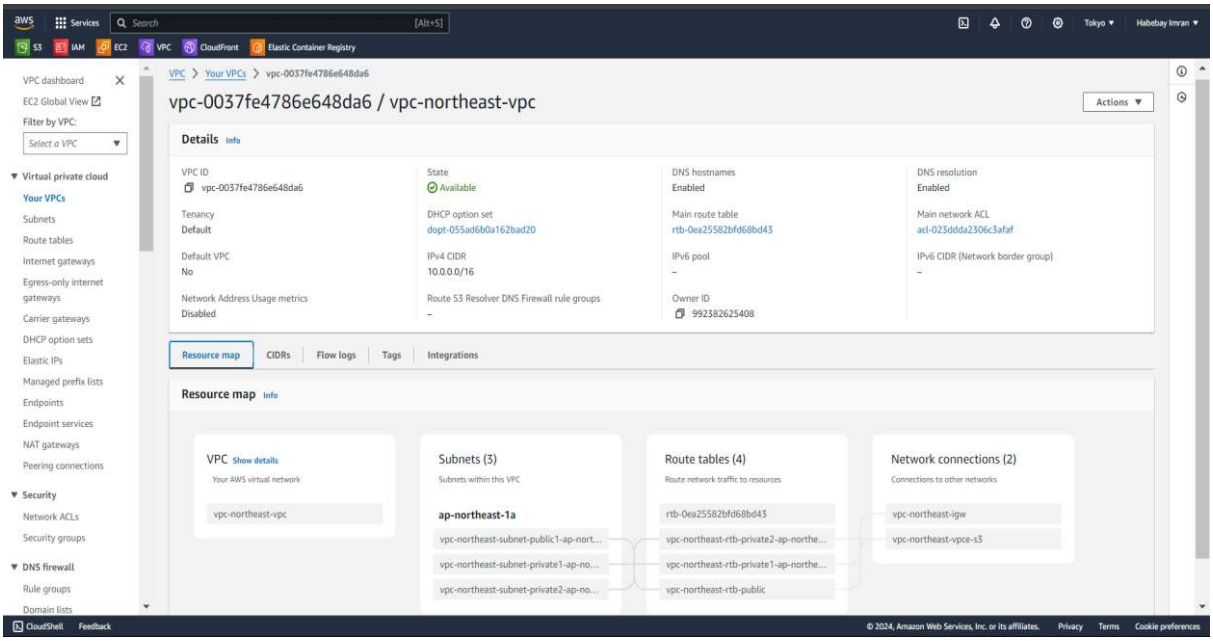
Aim:

To create an VPC in ap-northeast-1 region with the given requirements

Description:

a. Login to the AWS console.
b. Search for VPC service in search bar.
c. Click ‘Create VPC’ in VPC service.
d. Select ap-northeast-1 (Tokyo) AWS Region.
e. Give “vpc-northeast” as its name.
f. Choose the CIDR Range as 12.0.0.0/14.
g. Choose the number of availability zone as 1 to provision subnets.
h. Choose the number of public subnets as 1.
i. Choose the number of private subnets as 2.
j. Choose the number of availability zones to have NAT Gateways
k. Create a VPC endpoints
l. Enable DNS options
m. Finally click “Create VPC” and VPC is created

Output:



Result:

Thus VPC is created in ap-northeast-1 region with a specified CIDR range and having 1 public and 2 private subnets.