PRACTICAL 1

Create an AWS account, azure account and google cloud account.

Step 1:
First Open your web browser and navigate to AWS Free Tier Page

Step 2:
On middle click of Create a Free Account

Step 3:
Issue the details which you want to use to log in to your AWS account and click on Continue

- **Email address**: Provide the mail id which hasn't been registered yet with Amazon AWS.
- **Password:** Type your password.
- **Confirm password**: Authenticate the password.
- **AWS Account name**: Choose a name for your account. You can change this name in your account settings after you sign up

Step 4:

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).

Step 5:
Enter your Purpose of Account Registration

Step 6:
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.

Step 7:

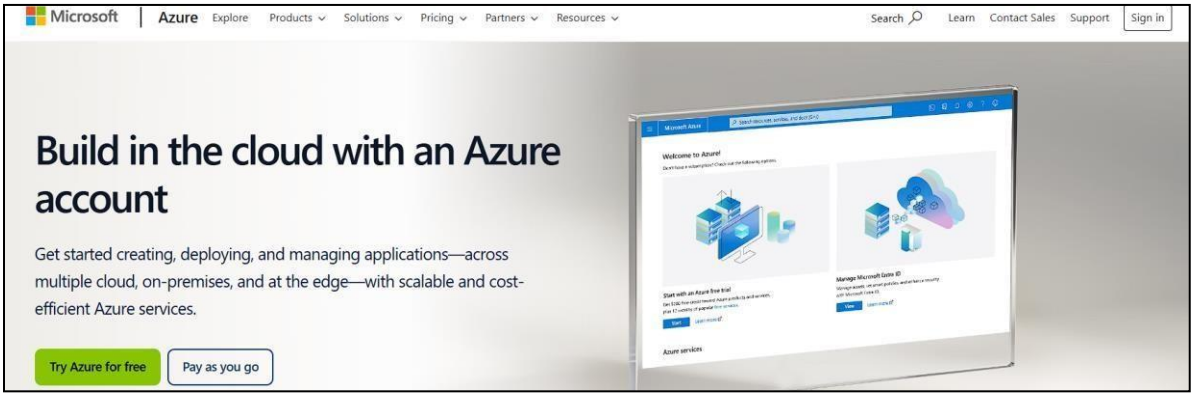
Support plan: AWS support offers a selection of plans to meet your business needs.
Select your suitable plan then click continue.

Step 8:

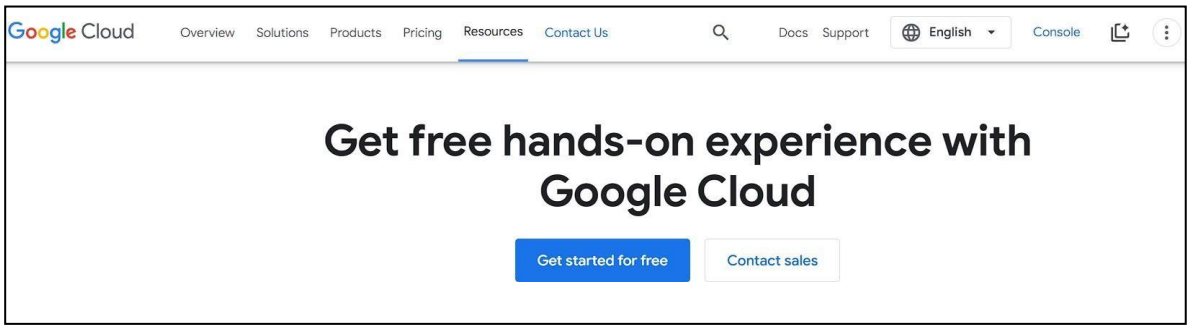
Registration Confirmation page.

Once you complete all the above steps and processes. You'll get the confirmation page below. Now your account will be processed for activation. It may take somewhere between 30 minutes to 1 hour for you to receive an email confirmation that your Amazon Cloud Services account has been activated

- **Azure Account:**
  Go to Azure Sign-
  Up.
  Click Start Free.
  Provide details, verify identity, and enter billing information.



- **Google Cloud Account:**
  Go to Google Cloud Sign-Up.
  Click Get Started for Free.
  Complete registration with a Google account, verify billing information, and log in.



PRACTICAL 2

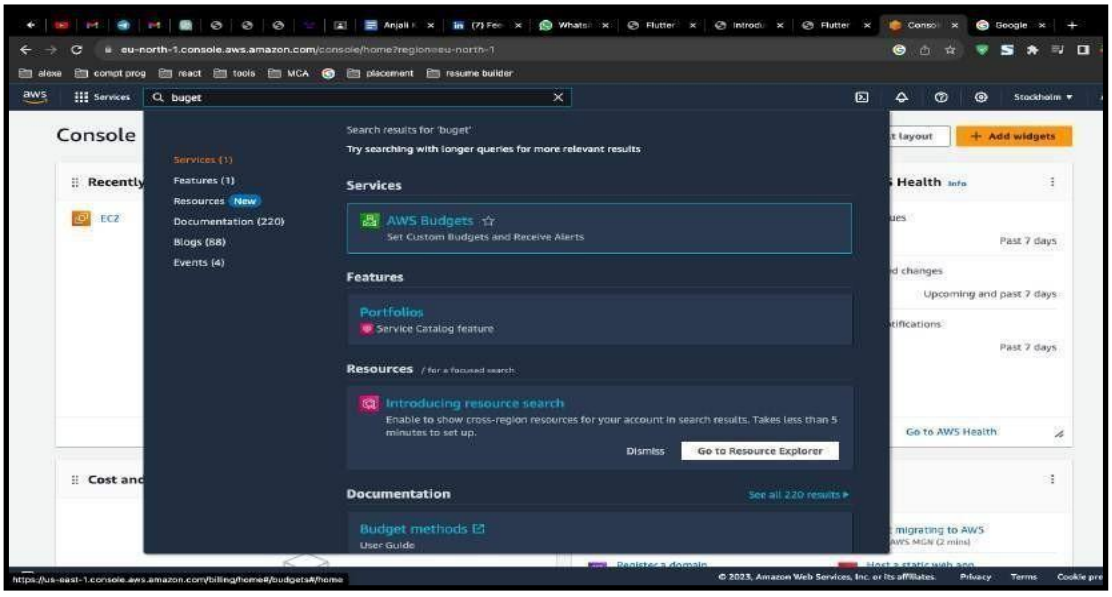
Set up Budget to an AWS account.

Objective: The primary object of AWS Budgets is to help AWS customers manage and control their cloud spending effectively. AWS Budgets is a cost management tool provided by Amazon Web Services (AWS) to help organizations set and track budgets for their AWS spending. Its main objectives are:

- Cost Monitoring: AWS Budgets provides insights into your AWS cost and usage data, allowing you to monitor your spending in real-time. This helps you keep track of your expenses and ensure they align with your budgetary goals.
- Budget Setting: The service allows you to set specific budgets for different aspects of your AWS usage, such as overall costs, service costs, or specific cost and usage patterns. You can create custom budgets that align with your business objectives.
- Cost Alerts: AWS Budgets enable you to set up cost and usage alerts. When your actual spending approaches or exceeds your budget thresholds, AWS Budgets will notify you via email or SNS (Simple Notification Service). This helps you take timely action to avoid unexpected overages.

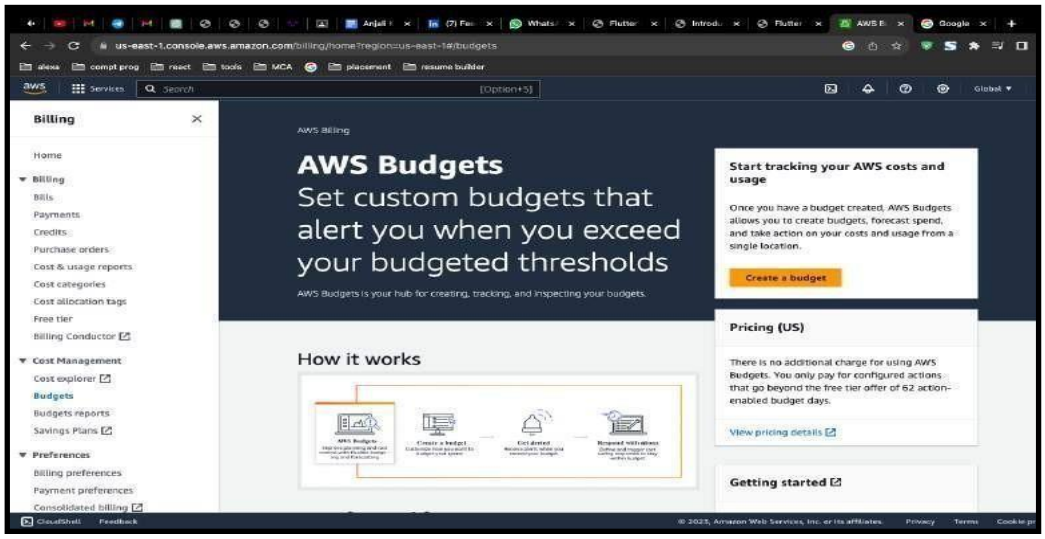
Step 1:
Sign in to the AWS Management Console and open the AWS Cost Management console at https://console.aws.amazon.com/cost-management/home.

Step 2:
In the navigation pane, choose Budgets.



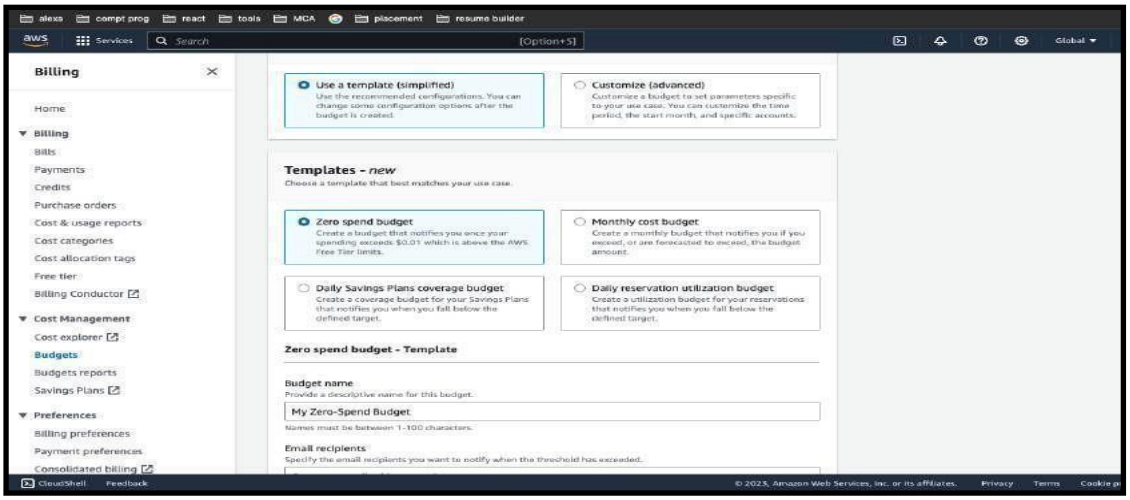
Step 3:

At the top of the page, choose Create budget.



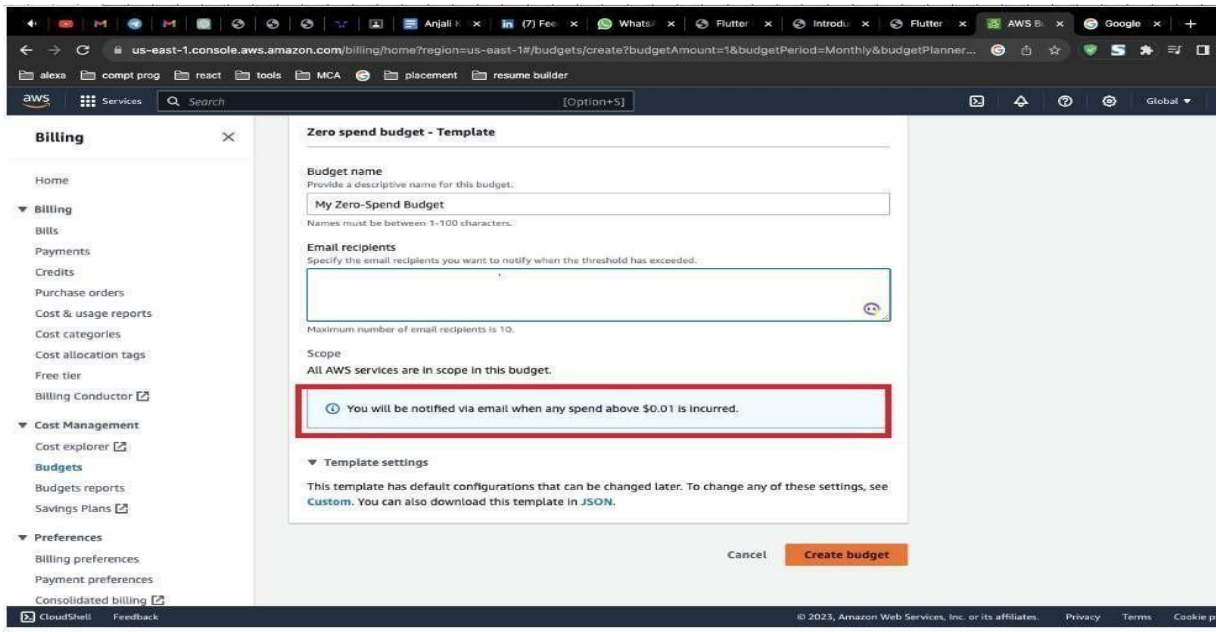
Step 4:

Under Details, for Budget name, enter the name of your budget. Your budget name must be unique within your account. It can contain A-Z, a-z, spaces, and the following characters:



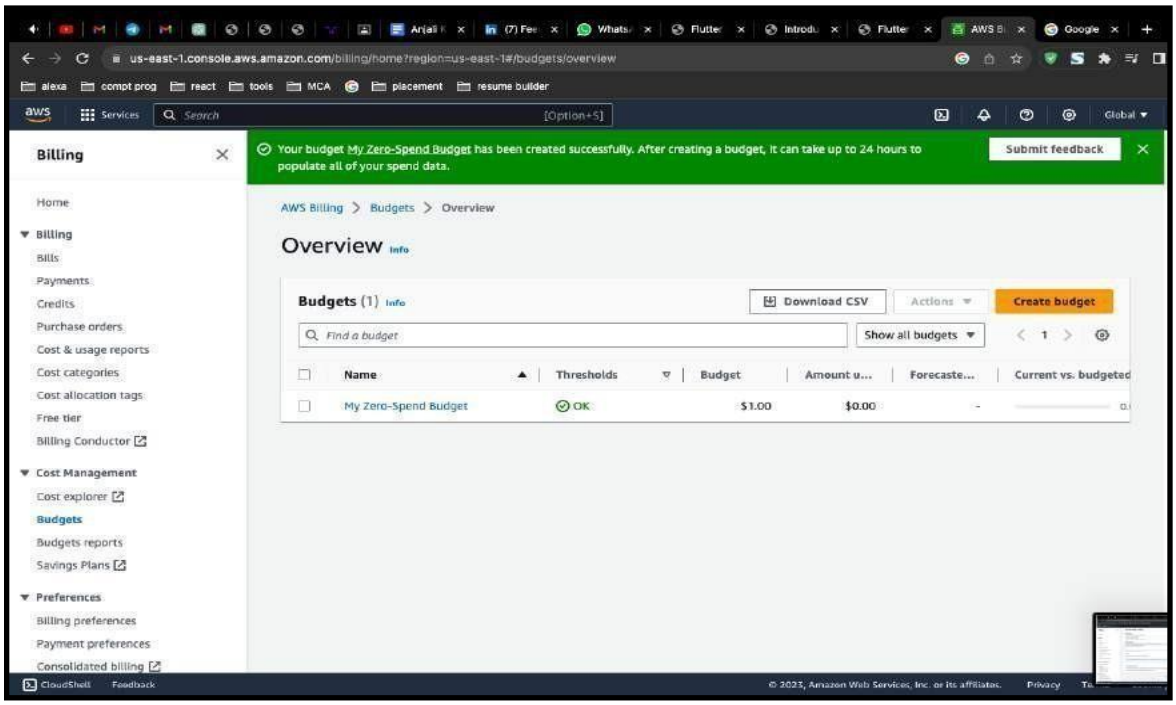
Step 5:

Under Set alert threshold, for Threshold, enter the amount that must be reached for you to be notified. This can be either an absolute value or a percentage.



Step 6:

Your Aws Budget is Created



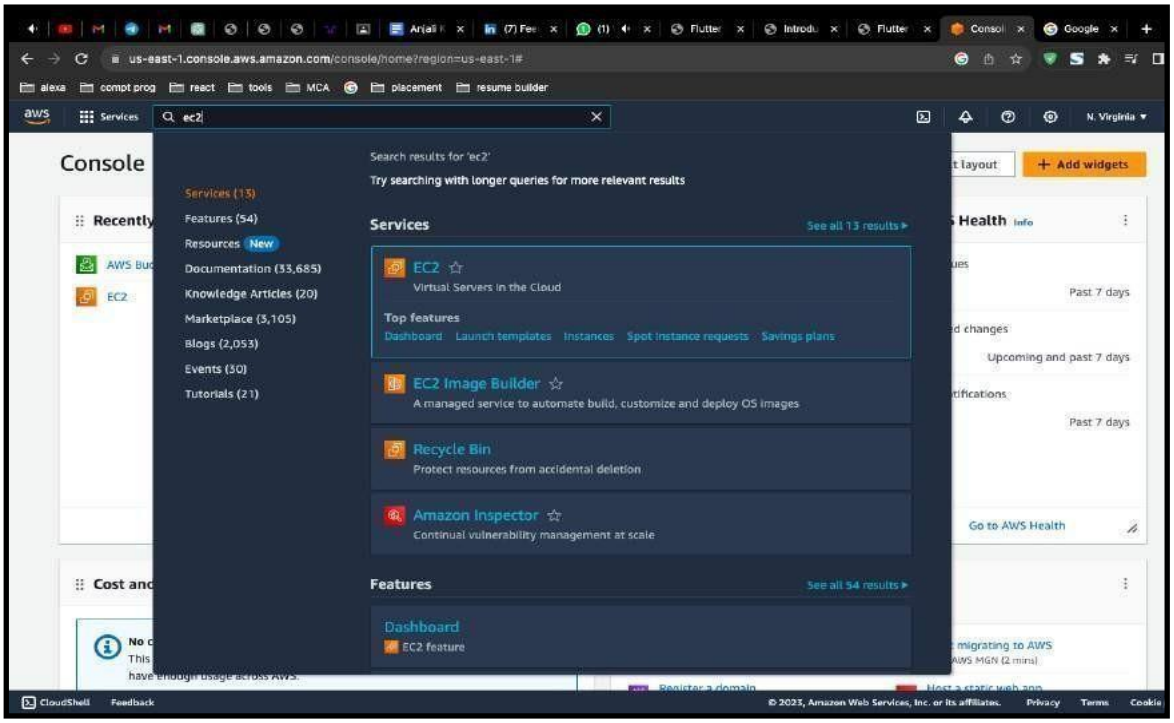
PRACTICAL 3

Launch a WINDOW SERVER INSTANCE with t2.micro.instance type and create a security group by using EC2

Objective: Launching a Windows Server instance in AWS EC2 serves a variety of purposes, depending on your specific needs and use cases. Here are some common reasons for launching a Windows Server in AWS EC2:

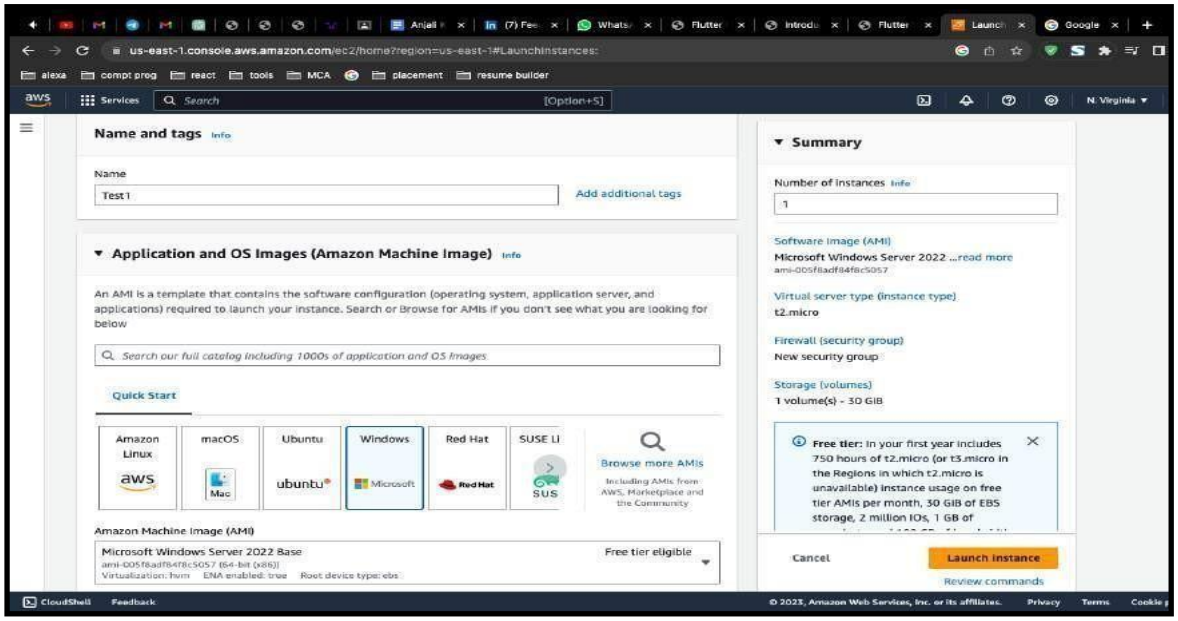
- Application Hosting: You can host Windows-based applications, including web servers, database servers, content management systems, and custom applications, on Windows Server instances in EC2.
- Development and Testing: Windows Server instances are ideal for development and testing environments. You can create isolated development environments, test software, and simulate production environments on-demand.
- Data Analysis and Reporting: Organizations often use Windows Servers in EC2 for data analysis, data warehousing, and generating reports using tools like SQL Server, Power BI, or custom analytics software.

Step1: Once logged in, navigate to the EC2 dashboard. You can do this by searching for "EC2" in the AWS Management Console's search bar or by selecting "Compute" and then "EC2" under the "Services" menu.



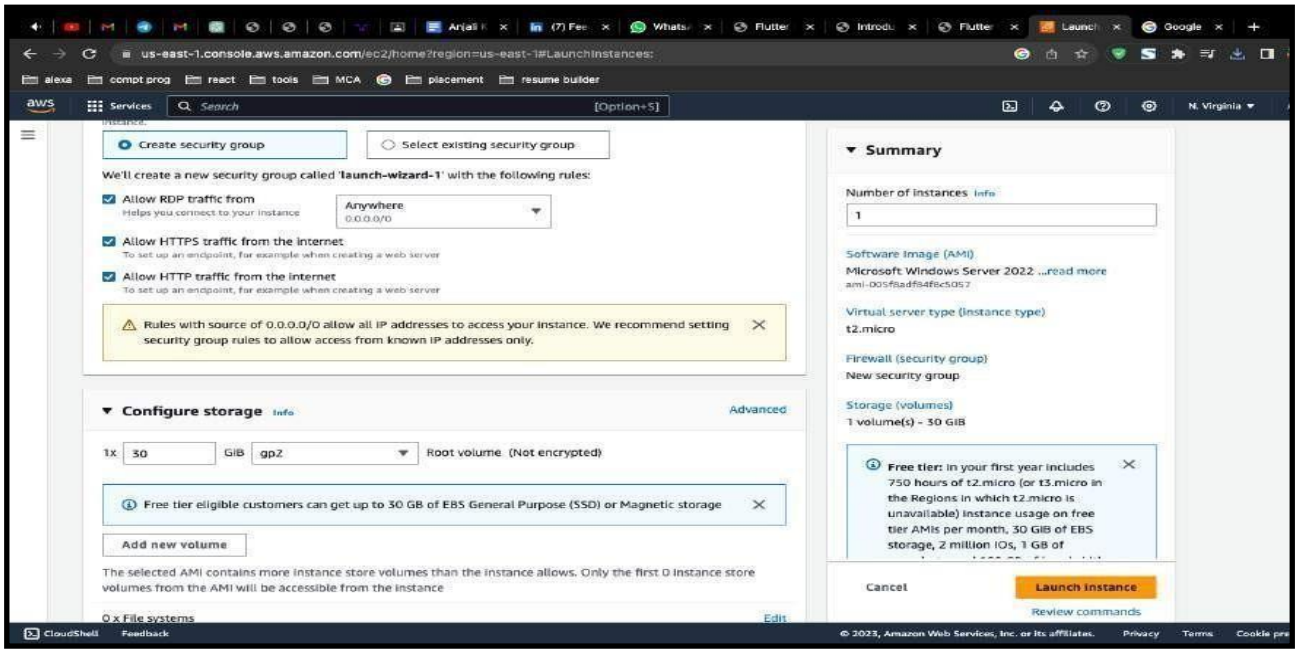
Step 2: Enter the name of your Aws Ec2 Instance

Step 3: In the "Choose an Amazon Machine Image (AMI)" step, search for a Windows Server AMI. AWS provides various Windows Server AMIs, including different Windows Server versions and editions. Select the one that suits your requirements.

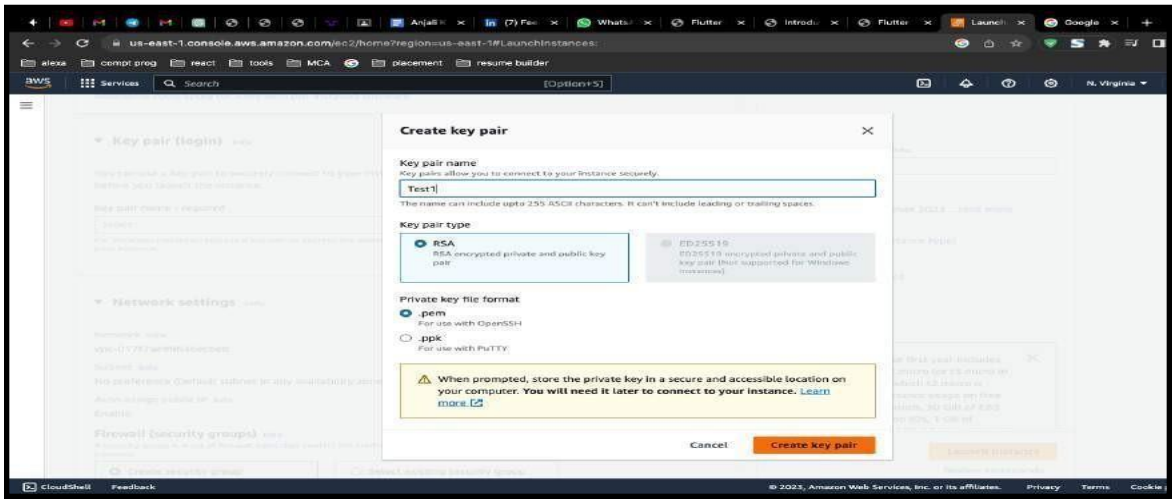


Step 4: In the "Add Storage" step, you can specify the size and type of root volume for your instance. You can also add additional volumes if necessary.

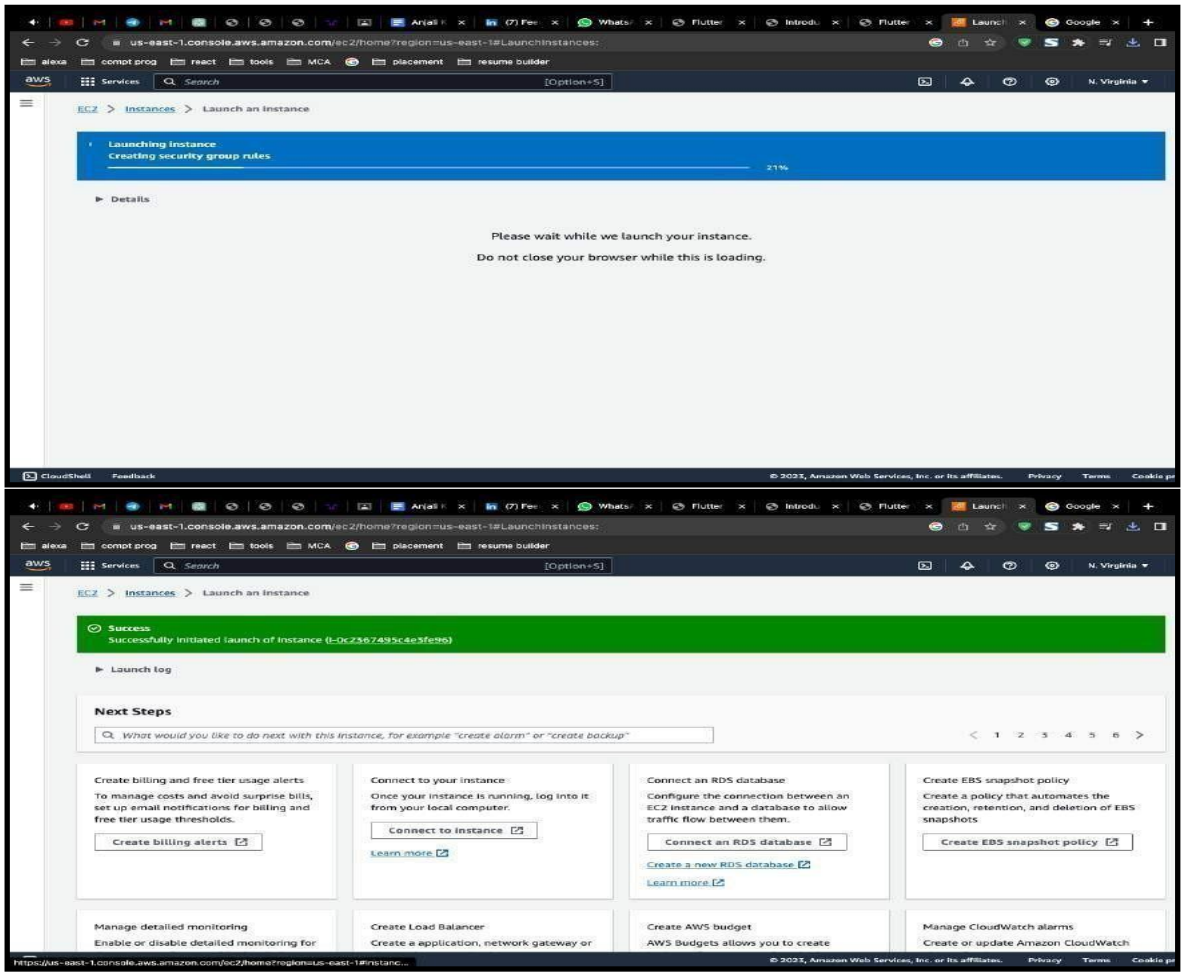
Step 5: In the "Configure Security Group" step, you'll need to configure the security group rules. Security groups act as firewalls to control inbound and outbound traffic to your instance. Ensure that you allow Remote Desktop Protocol (RDP) for Windows instances if you plan to access them remotely.



Step 6: If you haven't created a key pair, you will be prompted to create one. This key pair is used to securely access your Windows Server instance. Download and save the key pair (.pem file) in a secure location.



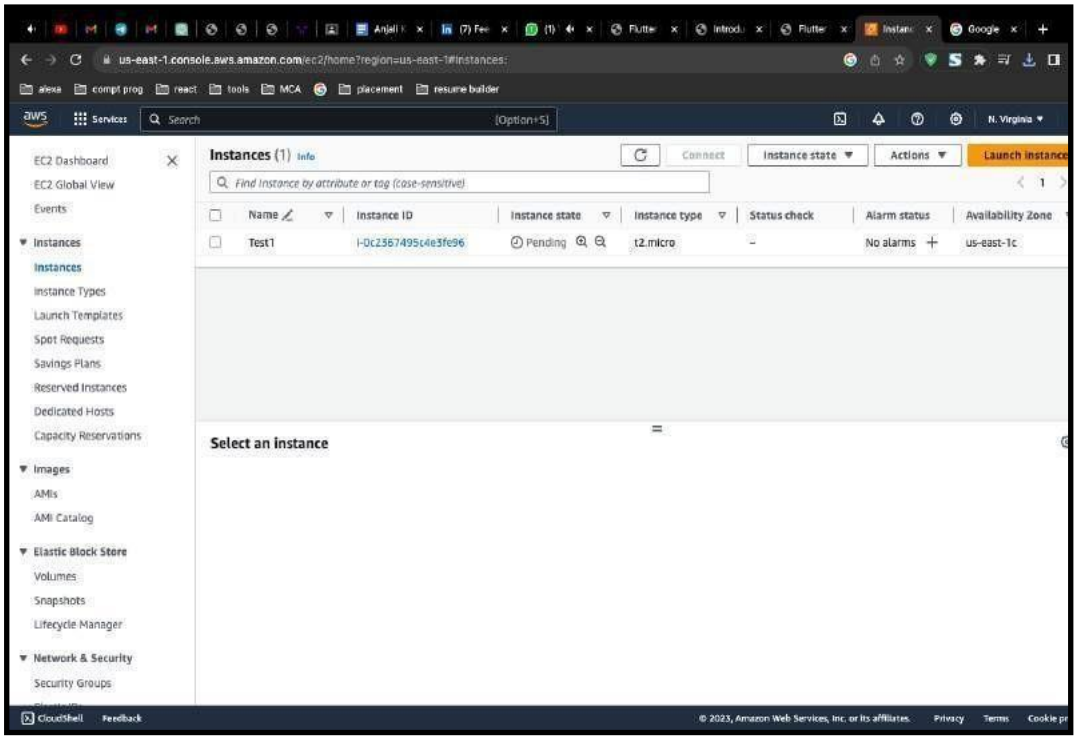
Step 7: After selecting or creating a key pair, click the "Launch Instances" button.



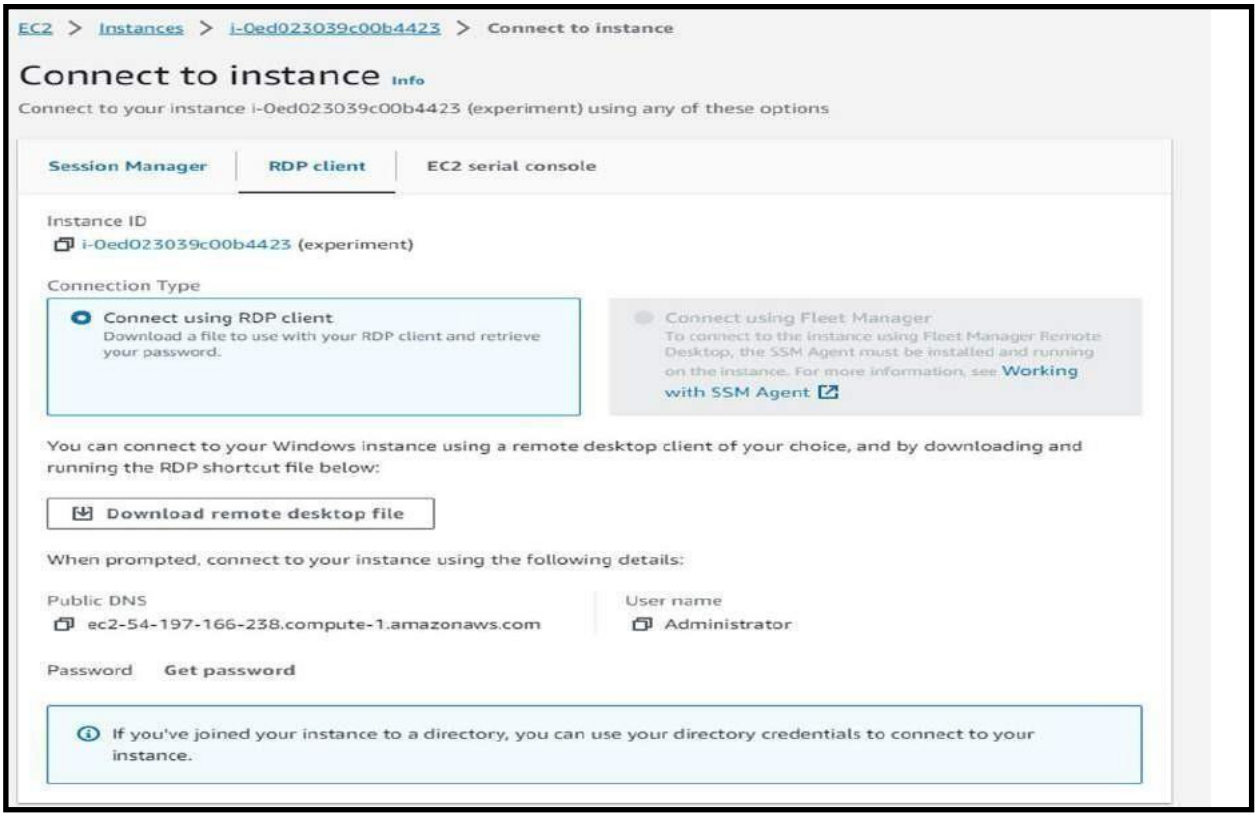
PRACTICAL 4

Connect the launch instance, 2/2 status check and decrypt password by using RDP client.

Step 1:
Select the Instance

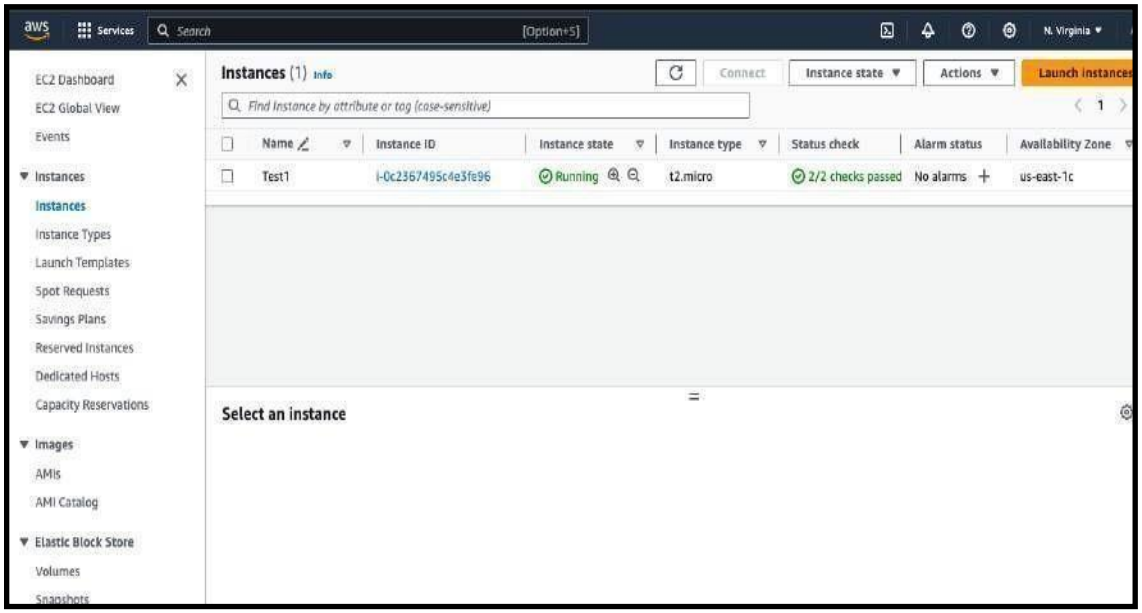


Step 2:
You'll be taken to the "Instances" view, where you can see the status of your Windows Server instance as it starts. Once the instance is in a "running" state, you can connect to it using RDP.



Step 3:

In the "Instances" view, you can see the status of your instance(s) as they transition from "pending" to "running." You can monitor the status changes there. Once the instance is in the "running" state, it means it has successfully launched, and you can then proceed to connect to it or use it for your intended purposes.

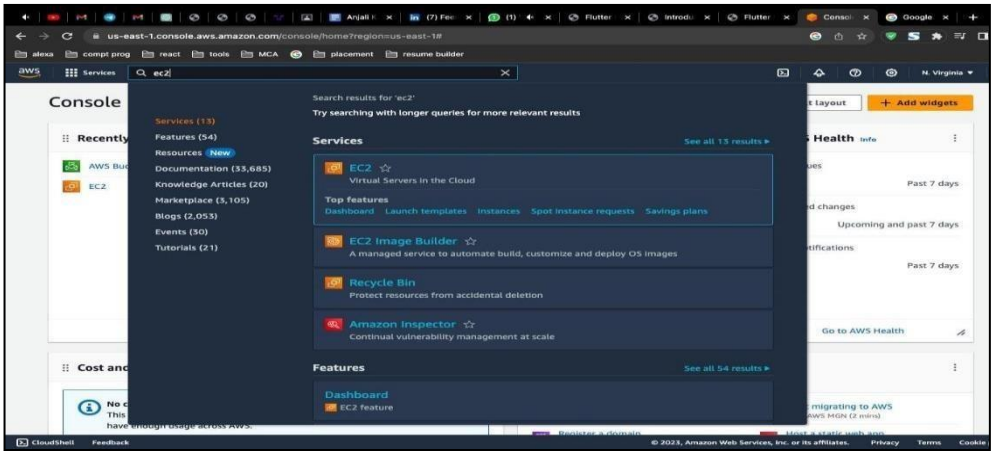


PRACTICAL 5

Launch a LINUX UBUNTU SERVER INSTANCE with t2.micro instance type and create a security group BY USING EC2.

Step 1: Log in to your AWS account and navigate to the EC2 service in the console.

Step 2: Click "Launch Instance" in the EC2 dashboard.

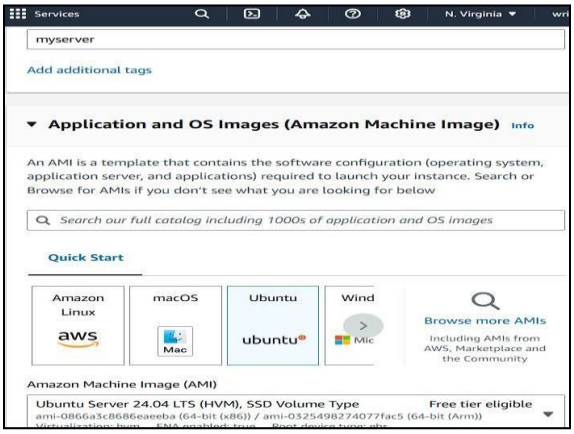


Step 3:

Choose an AMI:

Select "Ubuntu Server" from the "Amazon Machine Image (AMI)" list.

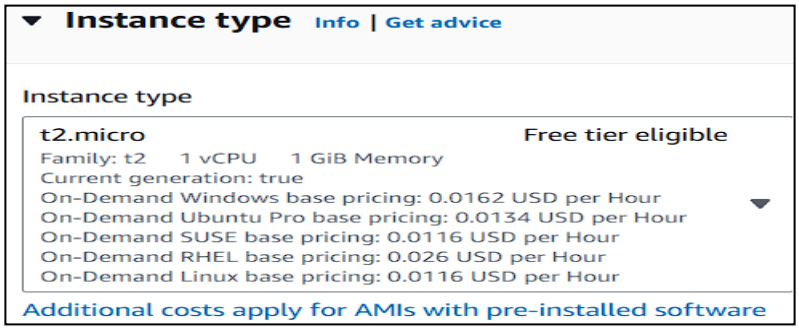
Ensure you choose the latest stable version of Ubuntu.



Step 4:

Select Instance Type:

Under "Instance type", select "t2.micro".



Step 5:

Configure Instance Details:

Name: Provide a descriptive name for your instance (e.g., "Ubuntu-Server").

Network Settings: Choose your preferred VPC and subnet.

Step 6:

Create a Security Group:

Create a new Security Group:

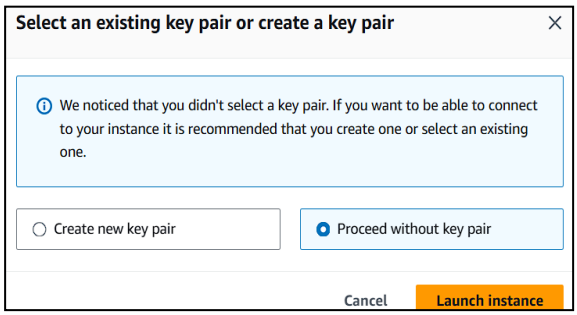
Click "Create a new security group" and give it a name (e.g., "Web Server").

Add Inbound Rules:

SSH (Port 22): Add a rule to allow inbound connections on port 22 from "Anywhere" (0.0.0.0/0) to access your instance via SSH.

HTTP (Port 80): If you plan to host a web server, add a rule to allow inbound connections on port 80 from "Anywhere" as well.

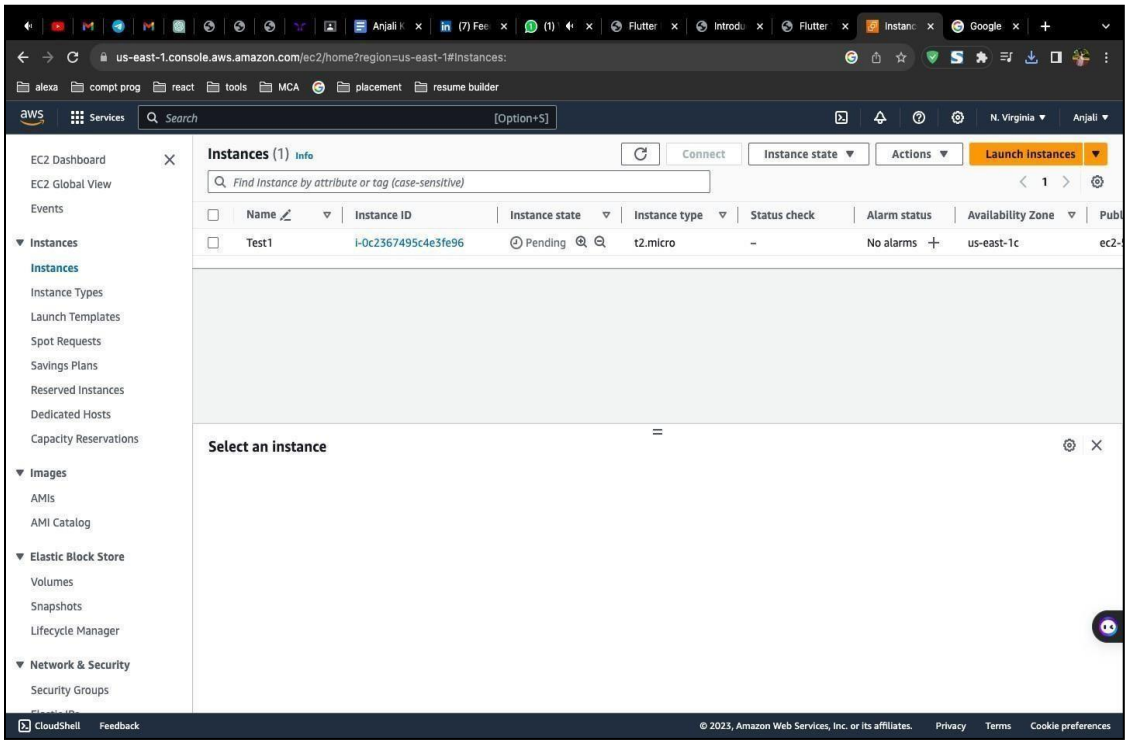
Step 7: Review and Launch.



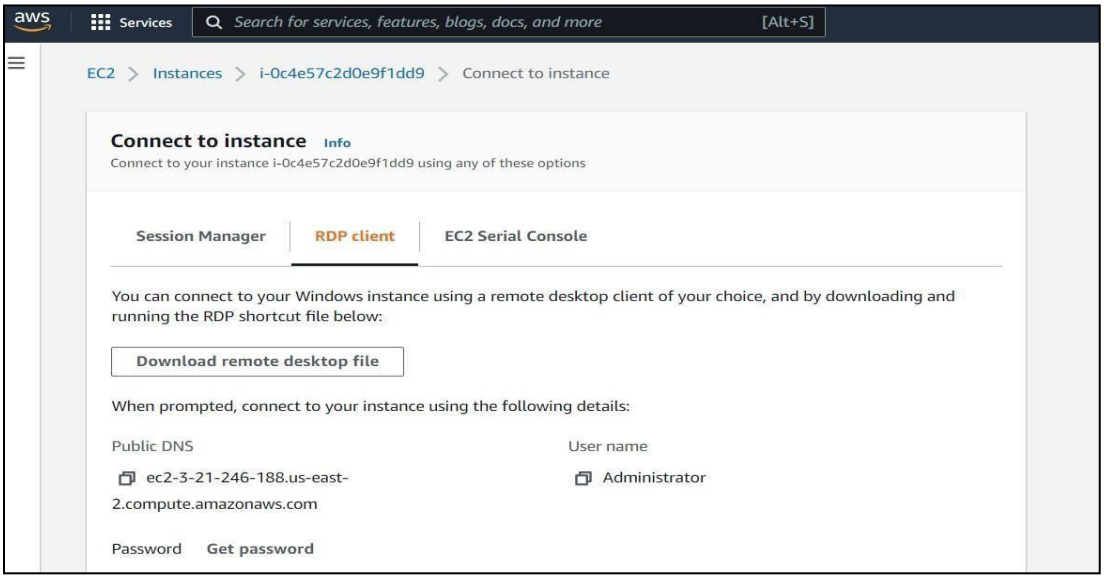
PRACTICAL 6

Connect the launch instance ,2/2 status check and decrypt password BY USING EC2 INSTANCE CONNECT.

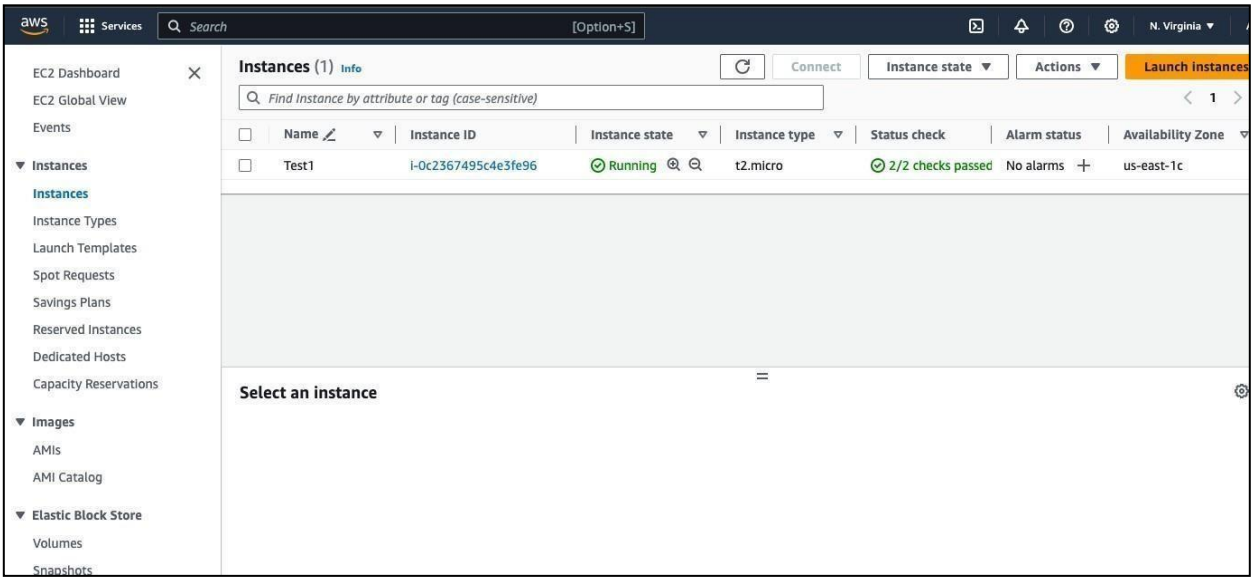
Step 1: Select the Instance



Step 2: You'll be taken to the "Instances" view, where you can see the status of your Windows Server instance as it starts. Once the instance is in a "running" state, you can connect to it using RDP.



Step 3: In the "Instances" view, you can see the status of your instance(s) as they transition from "pending" to "running." You can monitor the status changes there. Once the instance is in the "running" state, it means it has successfully launched, and you can then proceed to connect to it or use it for your intended purposes.

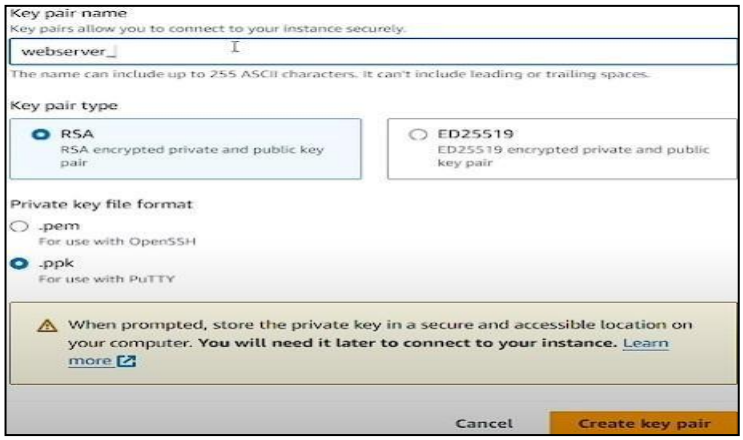


PRACTICAL 7

Connect the launch instance ,2/2 status check and decrypt password BY USING PUTTY AND SSH CLIENT.

Step 1: Convert the .pem File to a .ppk File (for PuTTY users)
If you are using PuTTY, you need to convert the .pem file to a .ppk file format.

Open PuTTYgen.
Click Load and select your .pem file.
Click Save private key to save it as a .ppk file.
Name and save the .ppk file, which will be used to connect to your instance in PuTTY.



Step 2: Check the Instance Status (2/2 Status Checks)
Log in to your AWS Management Console.
Go to EC2 Dashboard > Instances.
Ensure your instance is in the running state and that the 2/2 Status Checks have passed.
If they haven't, wait until both checks pass.

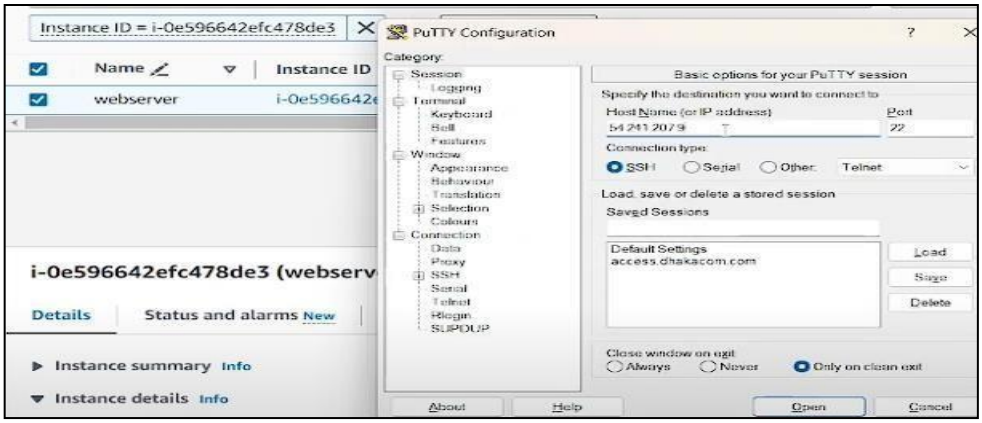
Step 3: Obtain the Public IP Address or DNS of the Instance
Select your instance in the EC2 Console.
Copy the Public IPv4 address or Public DNS (e.g., ec2-xx-xxx-xxx-xx.compute-1.amazonaws.com).

Step 4: Connect Using PuTTY (for Windows Users)
Open PuTTY.
In the Host Name (or IP address) field, enter ec2-user@<Public IP or DNS> (for Amazon Linux). Replace ec2-user with the correct username if using a different Linux distribution:

Ubuntu: ubuntu@<Public IP>
CentOS: centos@<Public IP>
Debian: admin@<Public IP>

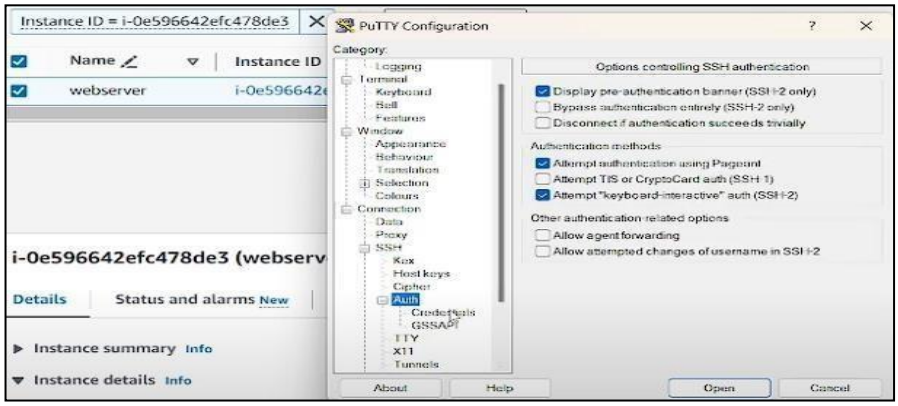
In the Category pane on the left, go to SSH > Auth.
Under Authentication parameters, click Browse and select your .ppk file.
Click Open to start the connection.

When prompted, accept the security alert to trust the connection.

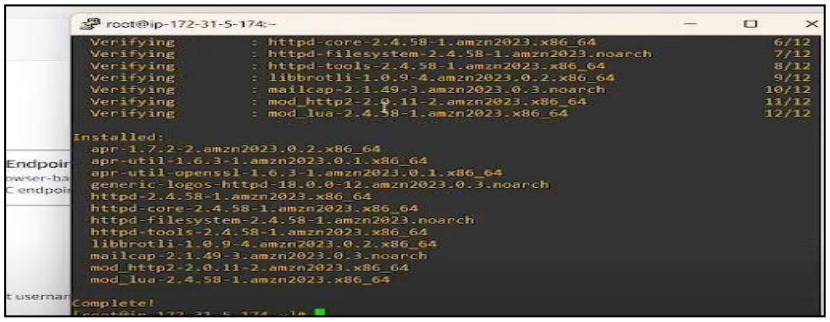


You should now be connected to your EC2 instance.

Step 5: Connect Using SSH Client (for Linux/Mac Users)
For users on Linux or MacOS, use the terminal with the SSH command. Make sure you have the .pem file ready.
Open a terminal.
Use the following SSH command to connect to the instance:
bash
Copy code
ssh -i /path/to/your-key.pem ec2-user@<Public IP or DNS>



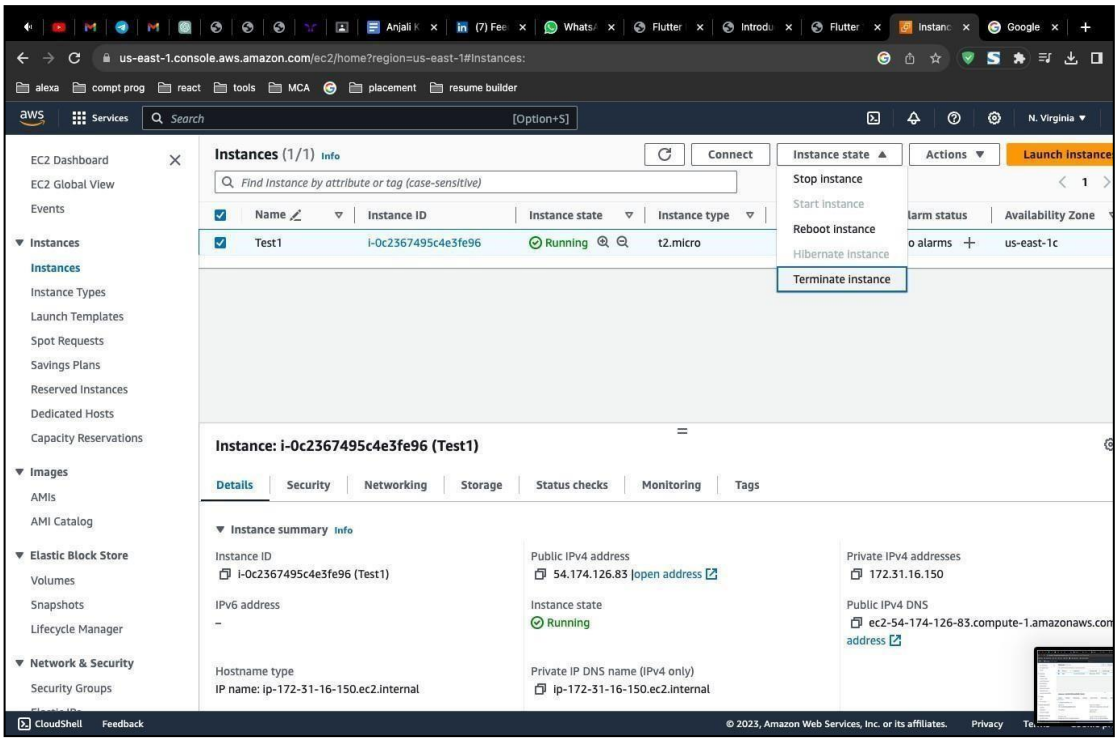
Step 6: Verify Connection
You should now be successfully connected to your EC2 instance via SSH. You'll see a terminal prompt indicating you're logged in.



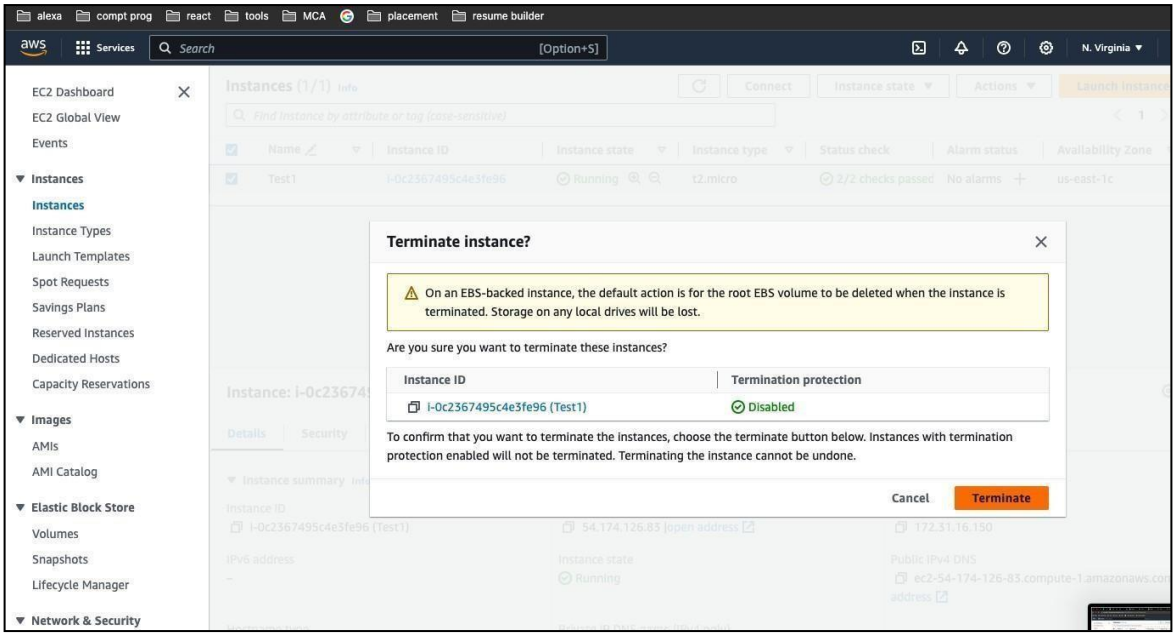
PRACTICAL 8

Terminate the launch instance and connect again by using RDP client.

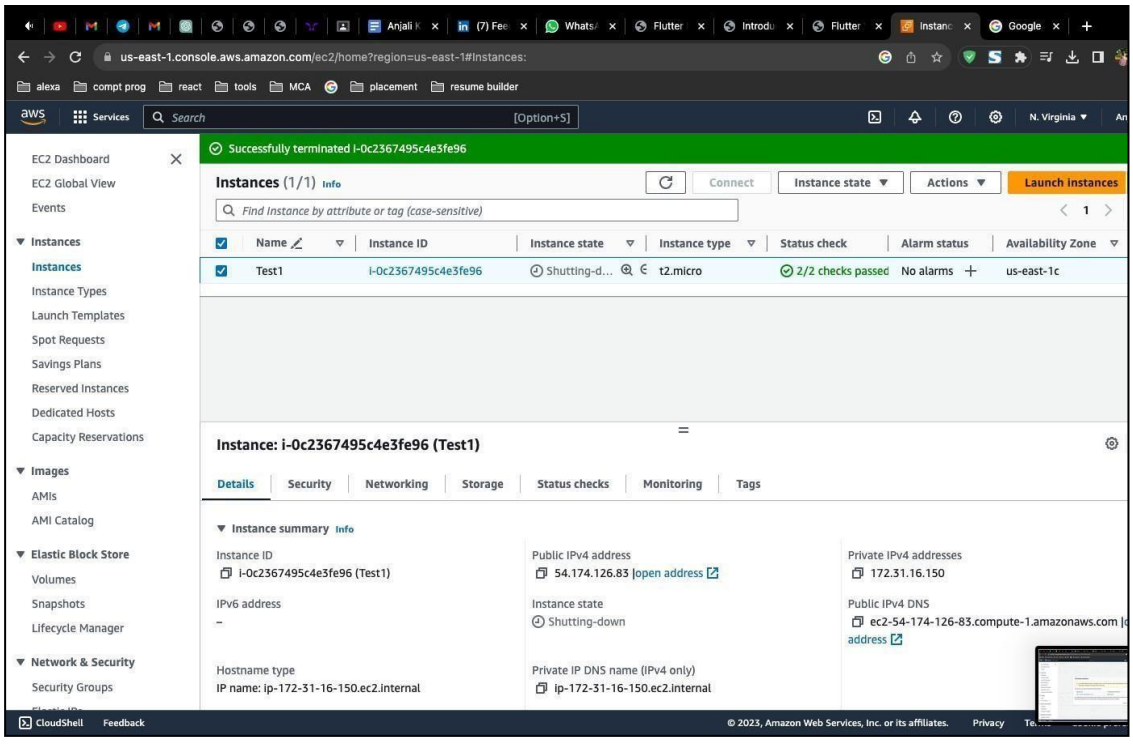
Step 1: In the EC2 dashboard, click on "Instances" in the left navigation pane to view a list of your running instances.



Step 2: With the instance selected, click the "Actions" button at the top of the dashboard, and from the dropdown menu, select "Instance State" and then choose "Terminate."



Step 3: AWS will now initiate the termination process. The instance will first be stopped if it was running, and then it will be permanently deleted. This process may take a few minutes.



PRACTICAL 9

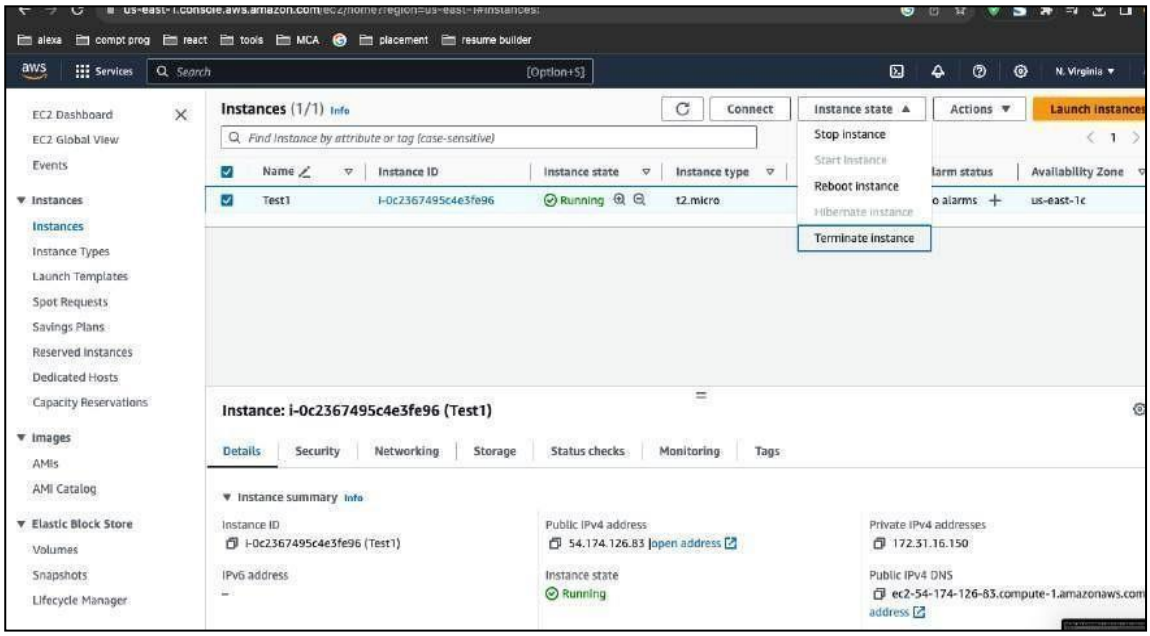
Delete the launched instance.

Step 1:

In the EC2 dashboard, click on "Instances" in the left navigation pane to view a list of your running instances. Locate the instance you want to terminate.

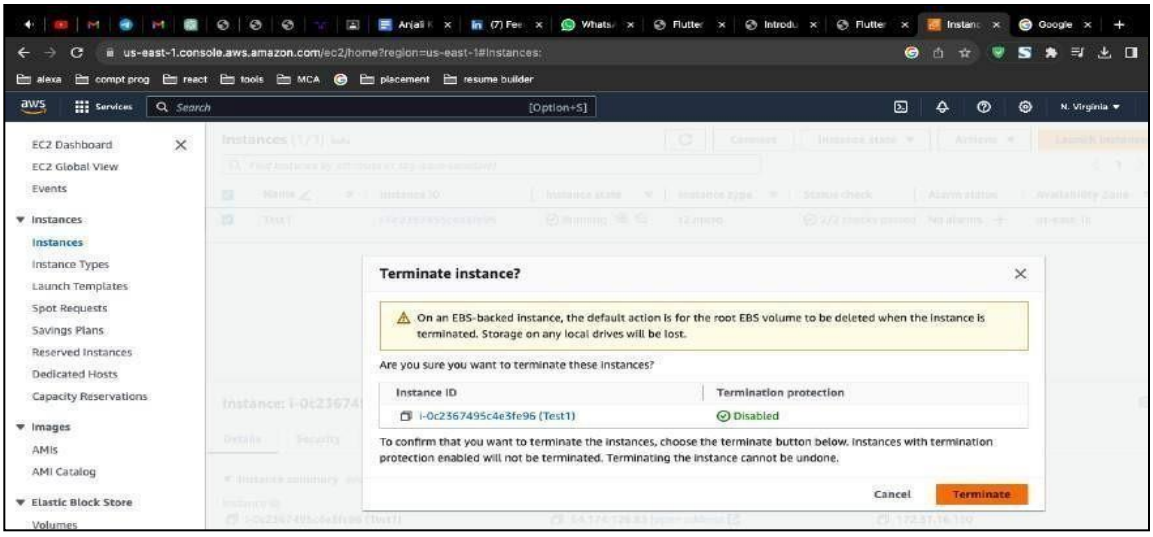
Step 2:

Click the checkbox next to the instance you want to terminate. It will become selected.



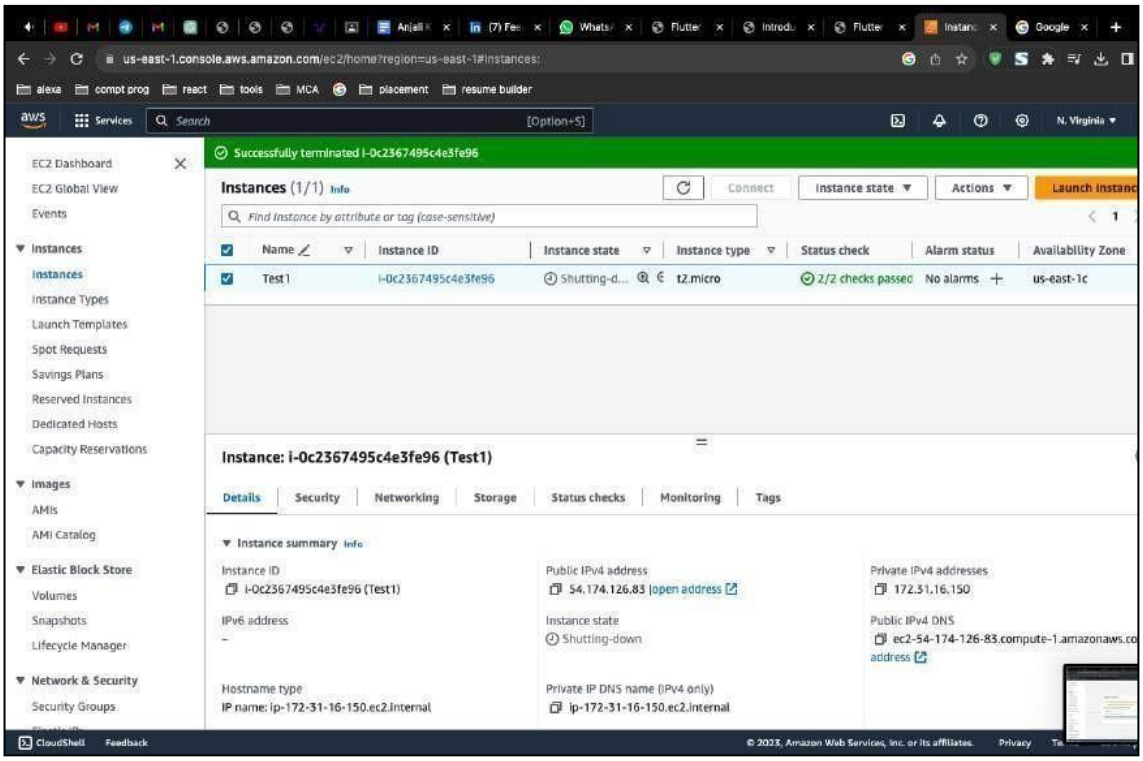
Step 3 :

With the instance selected, click the "Actions" button at the top of the dashboard, and from the dropdown menu, select "Instance State" and then choose "Terminate."



Step 4 :

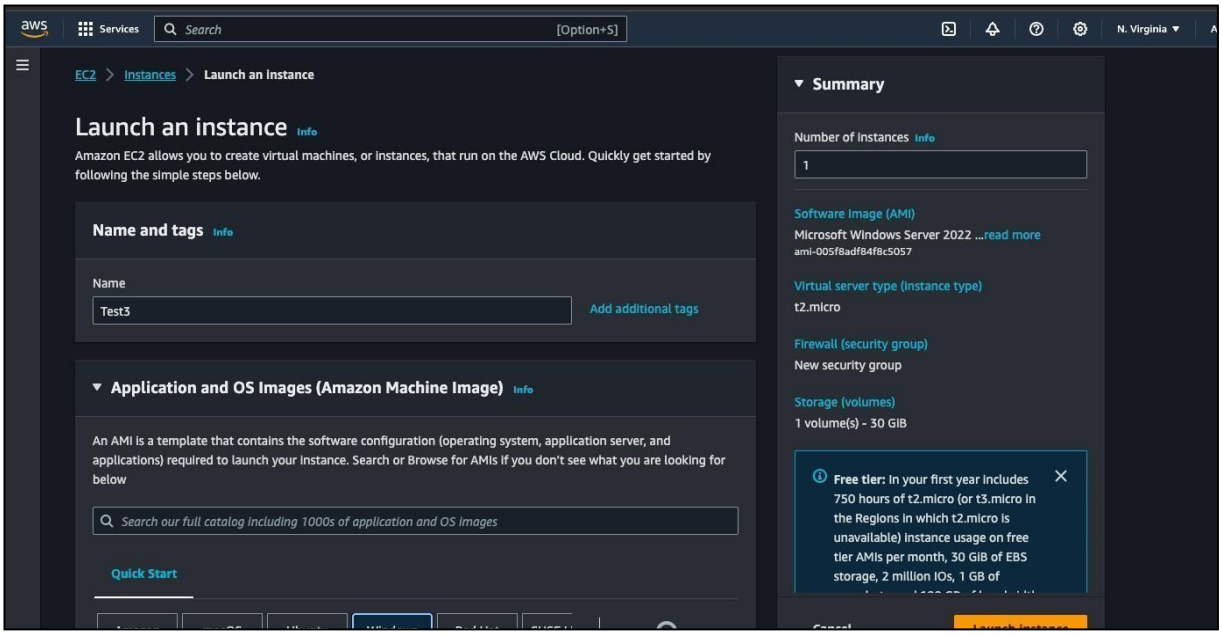
AWS will now initiate the termination process. The instance will be stopped if it was running, and then it will be permanently deleted. This process may take a few minutes.



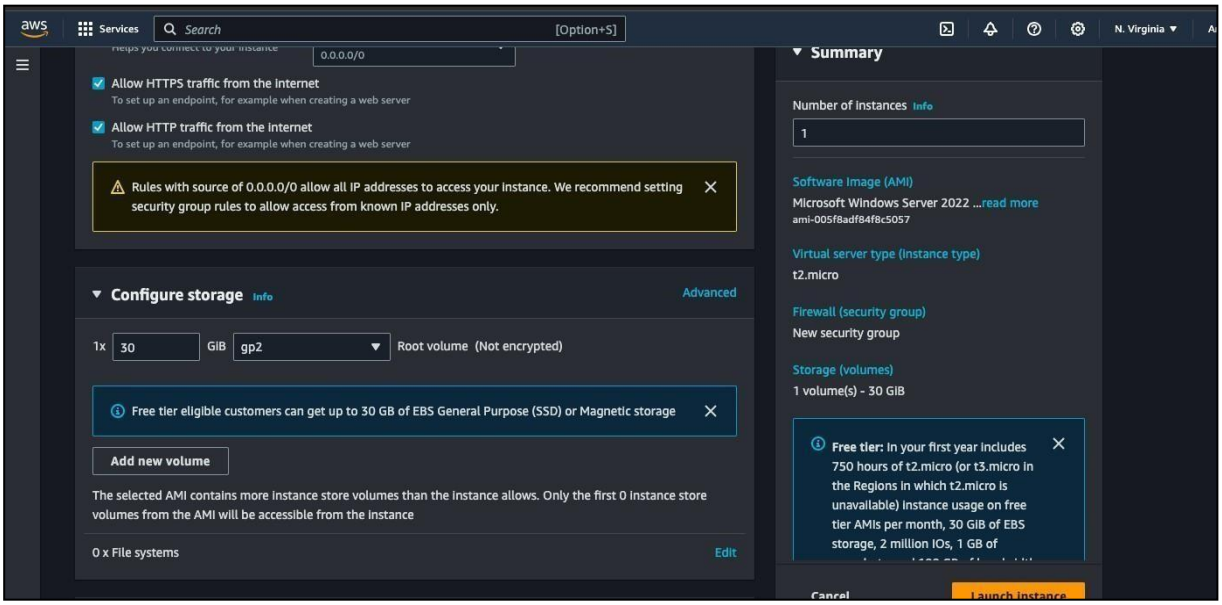
PRACTICAL 10

Host a static website on Windows server with IIS Manager on EC2 Launch Instance.

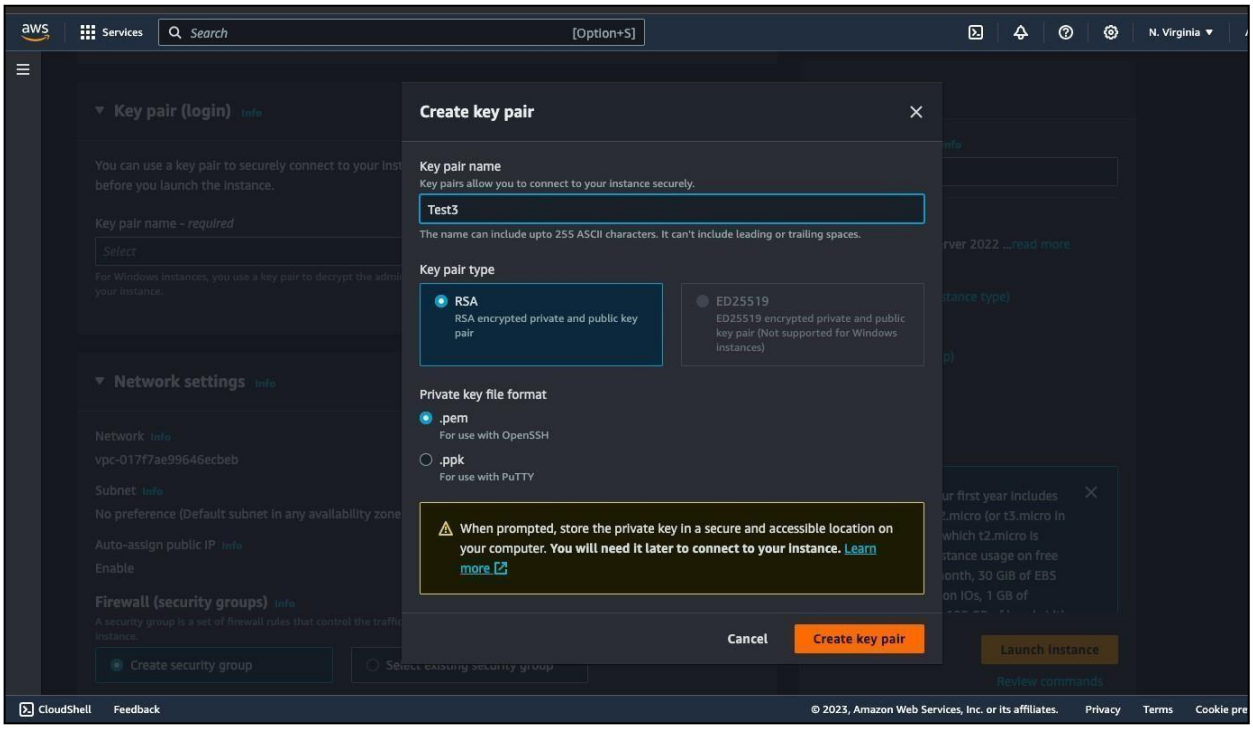
Step 1: Create a Instance



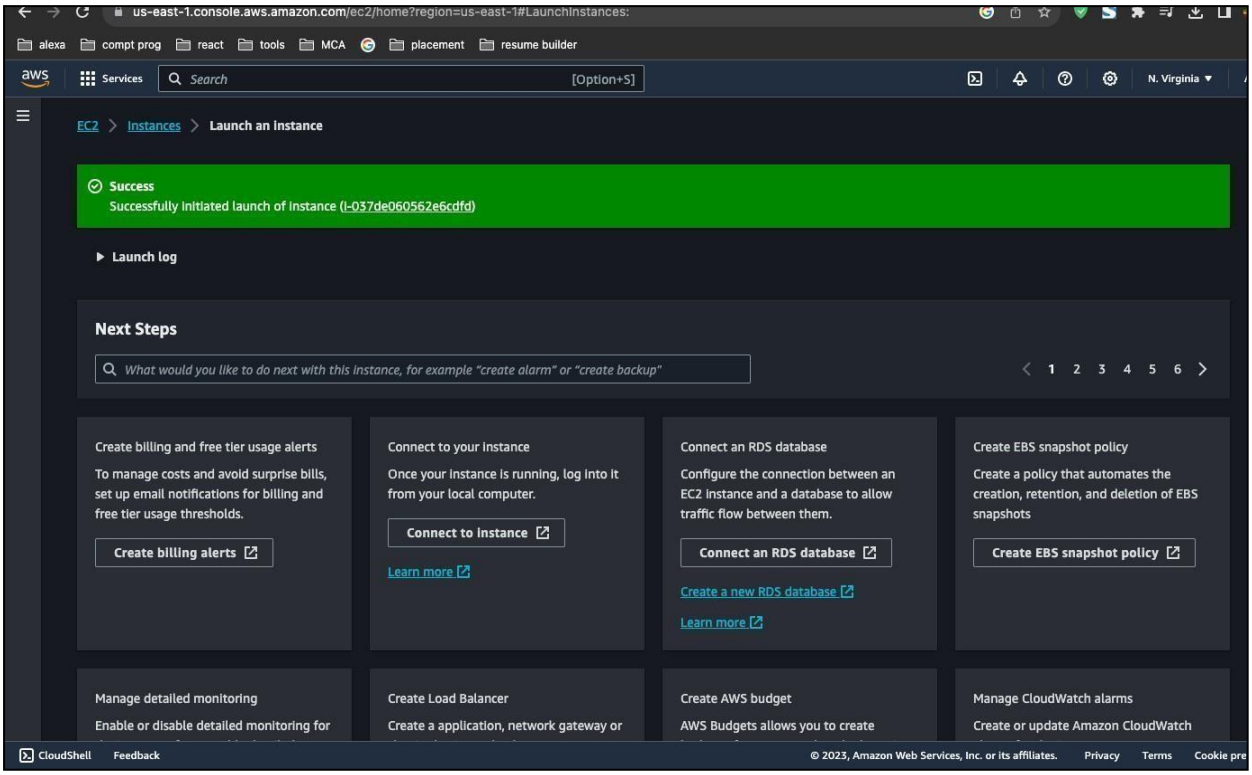
Step 2: In the "Configure Security Group" step, you'll need to configure the security group rules. Security groups act as firewalls to control inbound and outbound traffic to your instance. Ensure that you allow Remote Desktop Protocol (RDP) for Windows instances if you plan to access them remotely.



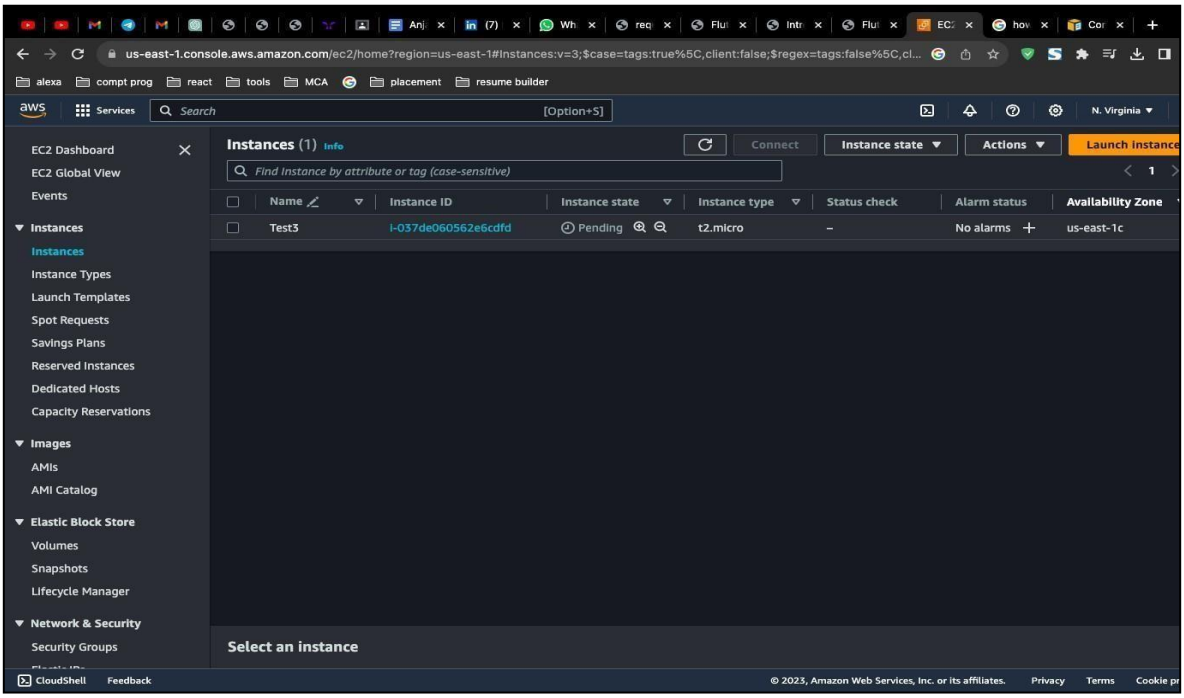
Step 3: Create a key pair for instance



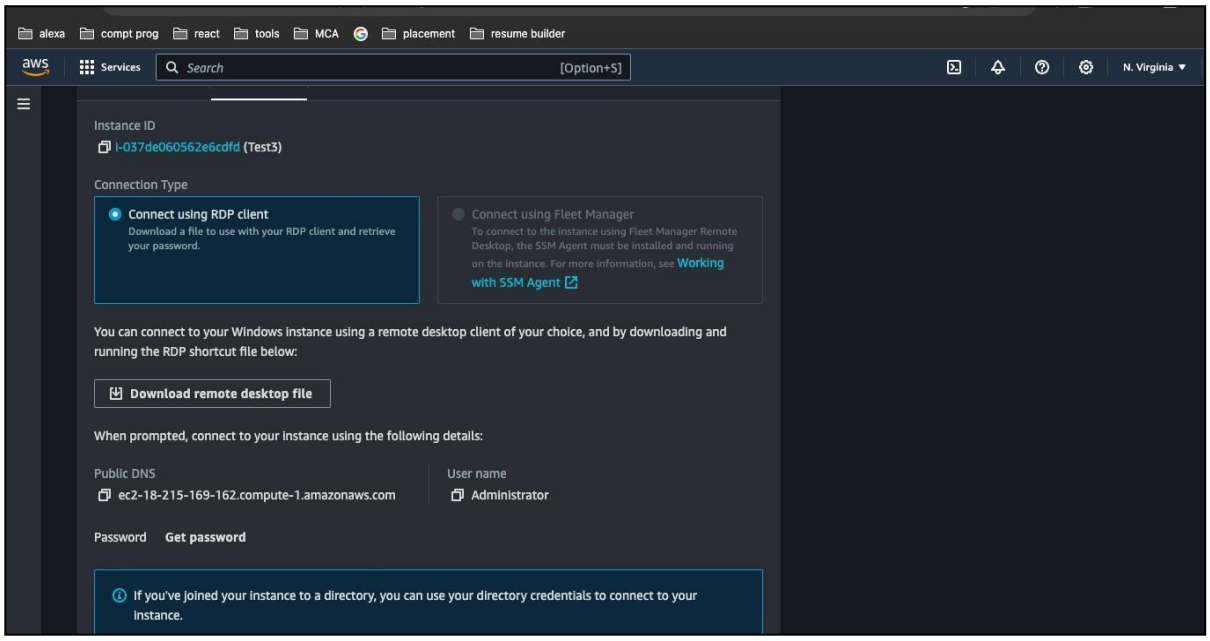
Step 4: Launch the instance



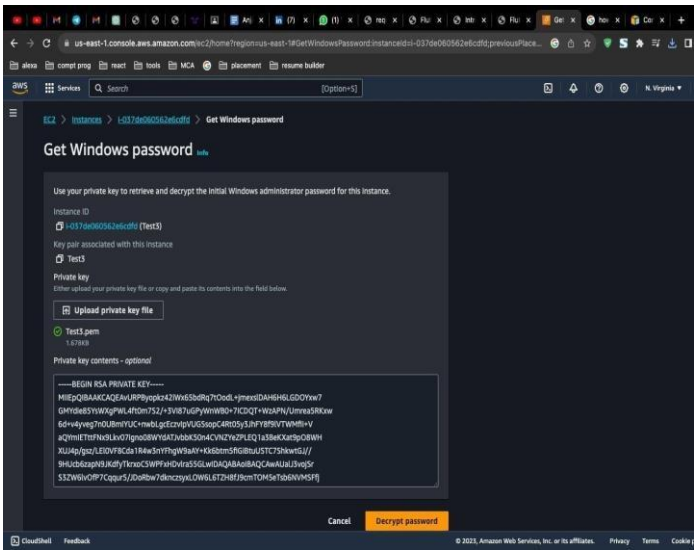
Step 5: Launch the instance

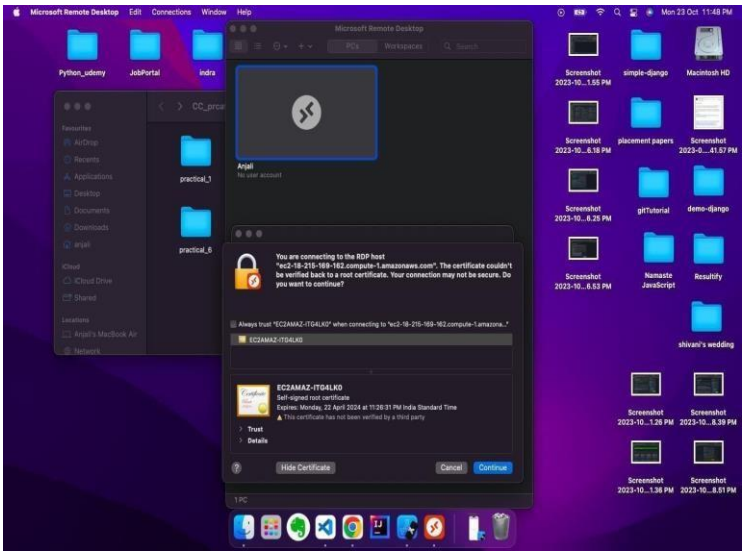


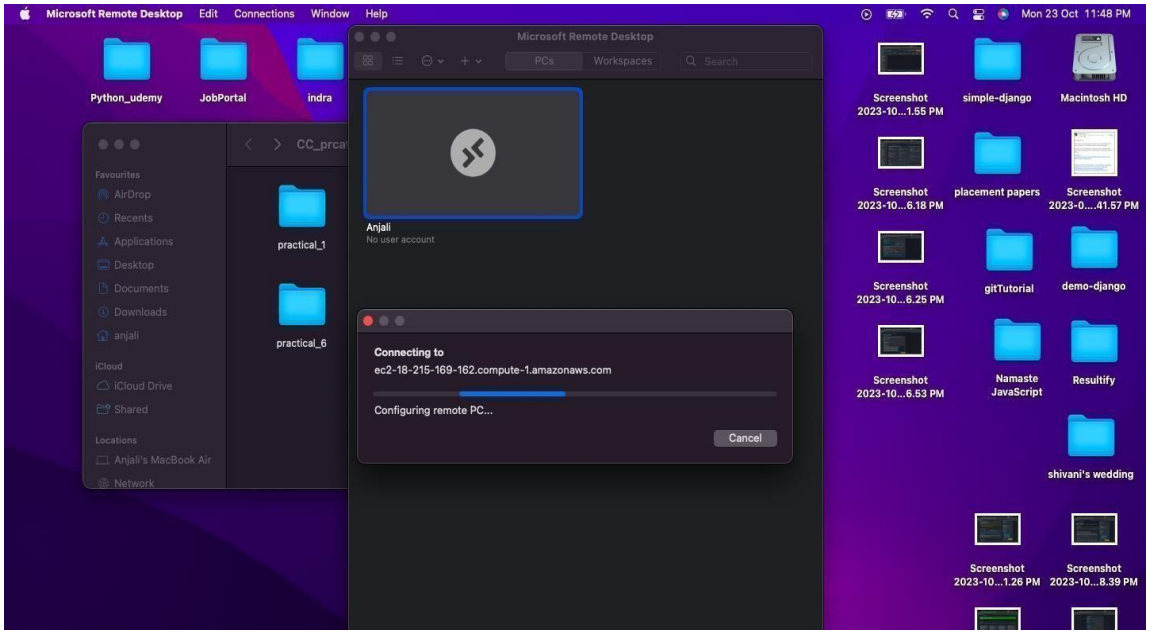
Step 6: Open your RDP client and configure a new connection with the public IP address or public DNS name of your Windows Server instance. You may also need to specify the username you want to use for the remote desktop session. By default, this is usually "Administrator."



Step 7: After configuring the connection, click "Connect" or "Connect" in your RDP client. You will be prompted to enter the administrator's username and password. If you haven't changed the password, you can retrieve it from the AWS Management Console.

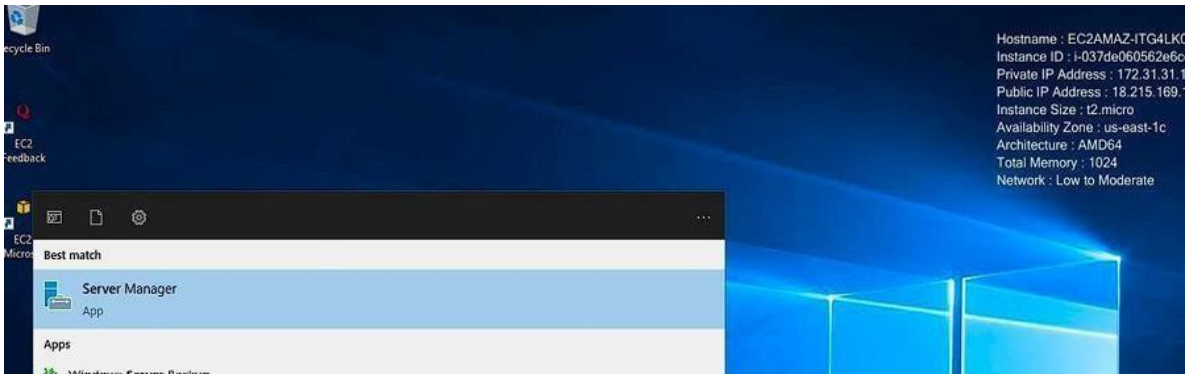




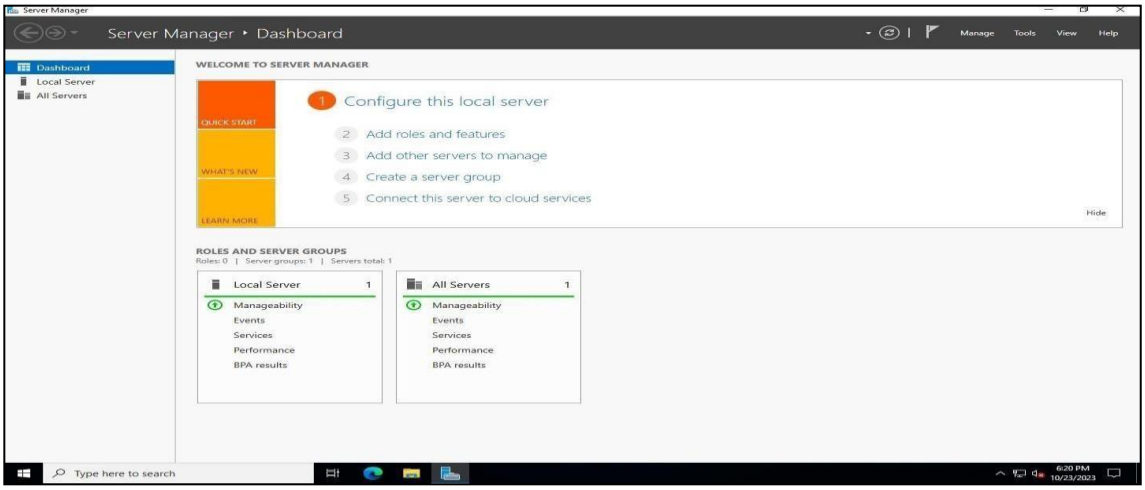


Step 8: To host a website, you'll need web server software. You can use Microsoft Internet Information Services (IIS) as a common choice for hosting websites on Windows Server. Follow these steps to install and configure IIS:

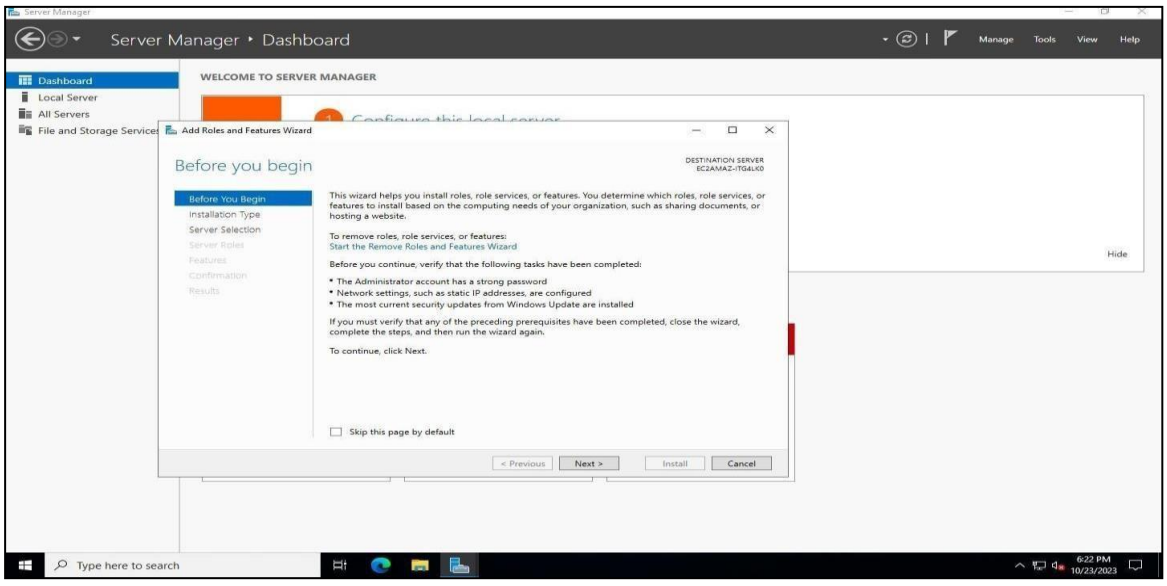
Step 9: In your Windows Server instance, open "Server Manager."



Step 10: Click on "Add roles and features" and follow the wizard to install the Web Server (IIS) role.



Step 11: Once IIS is installed, you can configure your website by creating a new site, specifying the content directory, and setting up bindings (e.g., domain names or IP addresses



PRACTICAL 11

Host a BCIIT SAMPLE website on Windows server with IIS Manager on EC2 Launch

Step 1: Connect to the Instance

Get the Public IP Address:

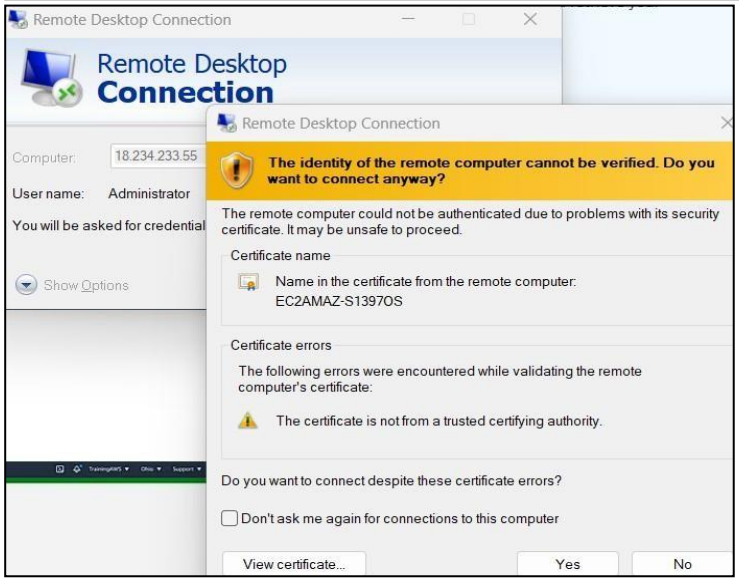
Once your instance is running, locate the Public IPv4 address on the EC2 dashboard.

Connect via RDP:

Open Remote Desktop Connection (Windows) or any RDP client.

Enter the public IP address and click Connect.

Use the username (Administrator) and the password generated from your key pair to log in.



Step2: Install IIS on Windows Server

Open Server Manager:

Once logged in, the Server Manager should open automatically. If not, search for it in the Start menu.

Add Roles and Features:

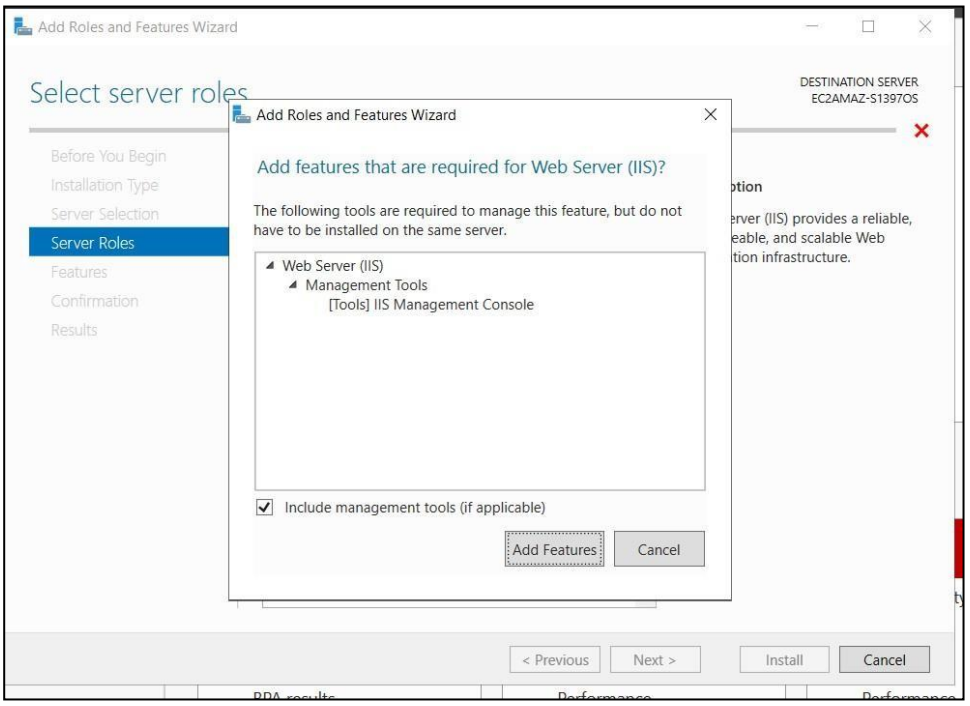
Click on Manage and select Add Roles and Features.

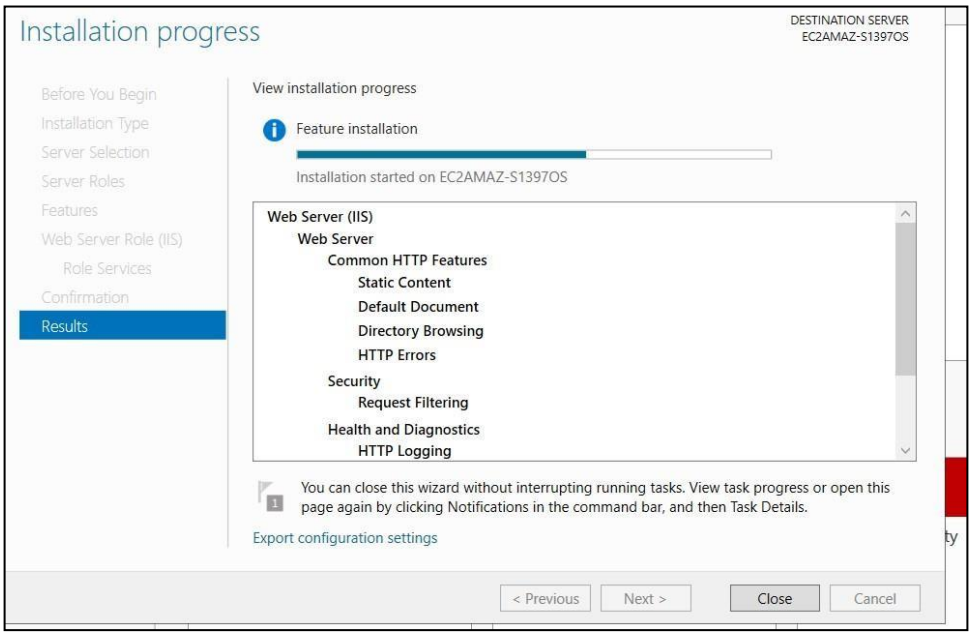
Click Next until you reach the Server Roles section.

Check the box for Web Server (IIS) and follow the prompts to add required features. Click Next through the wizard.

Click Install to begin the installation.

Wait for the installation to complete and then close the wizard.





Step 3: Prepare Your BCIIT Sample Website

Create a Directory for Your Website:

Navigate to C:\inetpub\wwwroot (default directory for IIS).

Create a new folder, e.g., BCIITSampleWebsite.

Upload Your Website Files:

Copy your BCIIT sample website files (HTML, CSS, JS, etc.) into the new folder.

You can use RDP file transfer, or if you have the files locally, you can transfer them via RDP or a file transfer method like WinSCP.

Step 4: Configure IIS to Host Your Website

Open IIS Manager:

Search for IIS Manager in the Start menu and open it.

Add a New Website:

In IIS Manager, expand the server node in the left pane and right-click on Sites.

Select Add Website.

Site name: Enter a name for your site (e.g., BCIITSampleWebsite).

Physical path: Browse to the folder you created in C:\inetpub\wwwroot\BCIITSampleWebsite.

Binding: Set the IP address to All Unassigned and Port to 80. Leave the Host name blank.

Click OK.

Step 5: Configure Security Group

Open Security Groups in AWS:

In the AWS Management Console, navigate to EC2 > Security Groups.

Select the security group associated with your instance.

Edit Inbound Rules:

Click on the Inbound rules tab.

Add a rule to allow HTTP traffic:

Type: HTTP

Protocol: TCP

Port: 80

Source: 0.0.0.0/0 (to allow access from anywhere)

Save the rules.

Step 6: Access Your BCIIT Sample Website

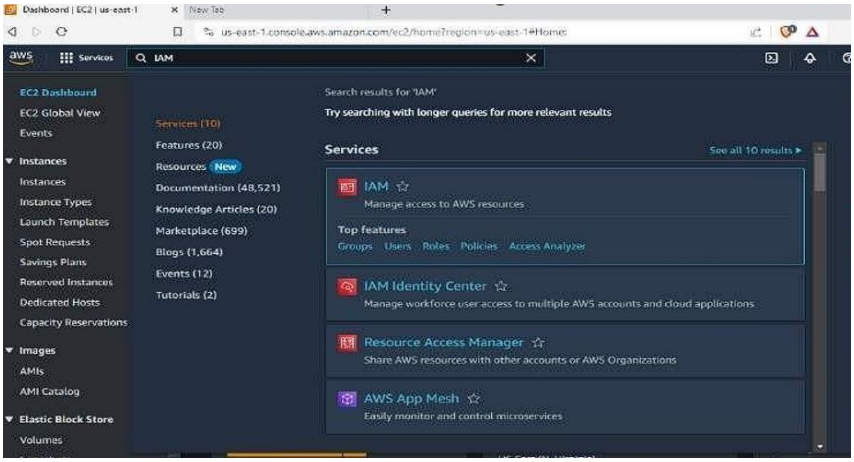
Open a Web Browser:

Enter the public IP address of your EC2 instance in the browser's address bar.

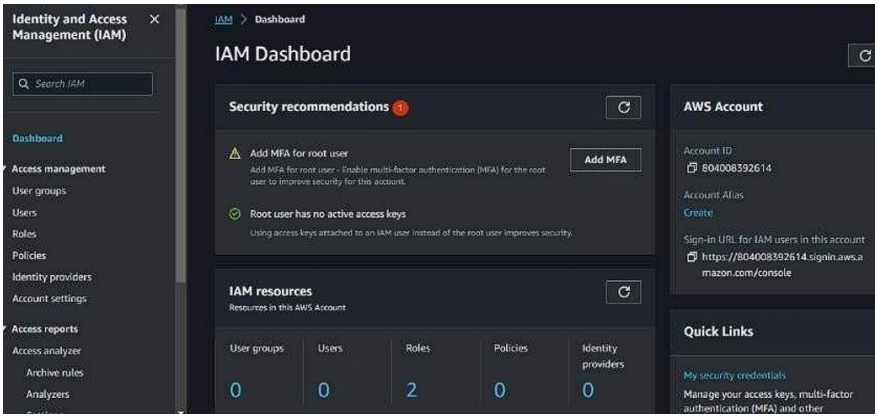
PRACTICAL 12

Create IAM user and grant in limited permission to IAM user by AWS route user

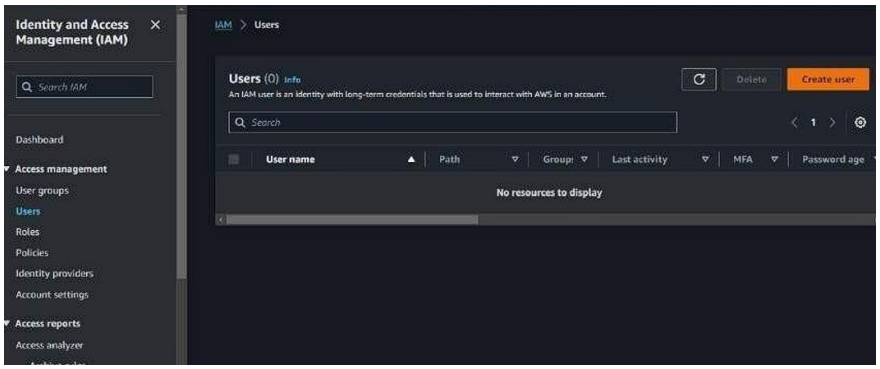
Step 1: Once logged in, navigate to the IAM (Identity and Access Management) Console. You can do this by searching for "IAM" in the AWS Management Console's search bar or by selecting "Security, Identity, & Compliance" and then "IAM" under the "Services" menu.



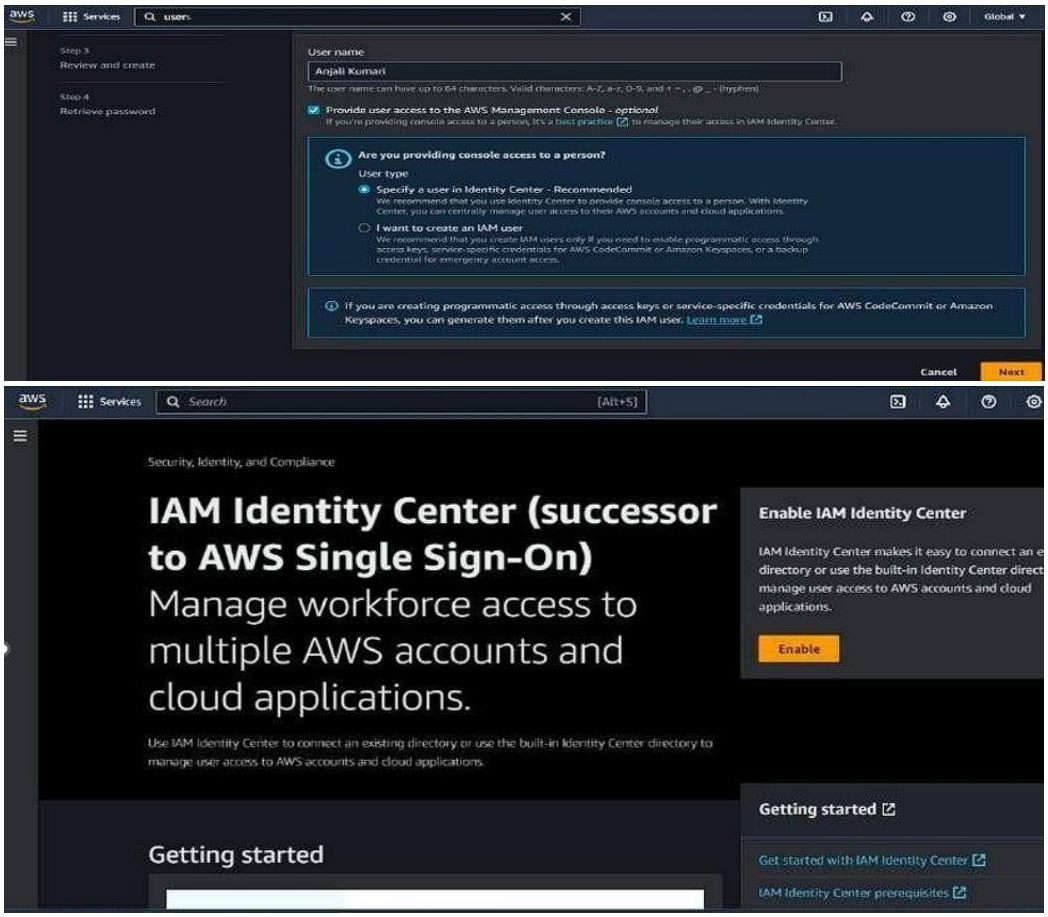
Step 2: In the IAM dashboard, click on "Users" in the left navigation pane to view the list of existing IAM users.



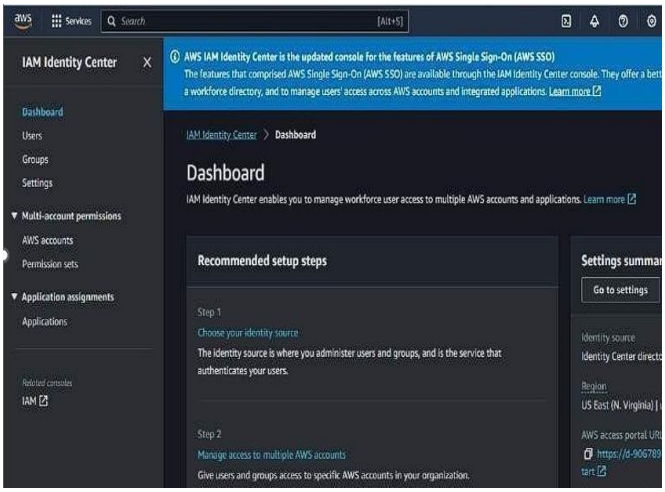
Step 3: To create a new IAM user, click the "Add user" button at the top of the dashboard.



Step 4: Username: Enter a unique name for the IAM user.



Step 5: After the user is successfully created, you'll see a confirmation page. This page provides important information, such as the user's access key and secret access key (if you selected "Programmatic access"). Make sure to download and securely store the access keys because they will only be displayed once.



Step 6: Give limited permission rights by navigating to permissions

PRACTICAL 13

Create a bucket by using S3 AWS service

Objective: The primary objective of Amazon S3 (Simple Storage Service) buckets in AWS is to provide a highly scalable, durable, and secure storage solution for a wide range of use cases. S3 buckets serve as containers for storing and managing data, and their key objectives include:

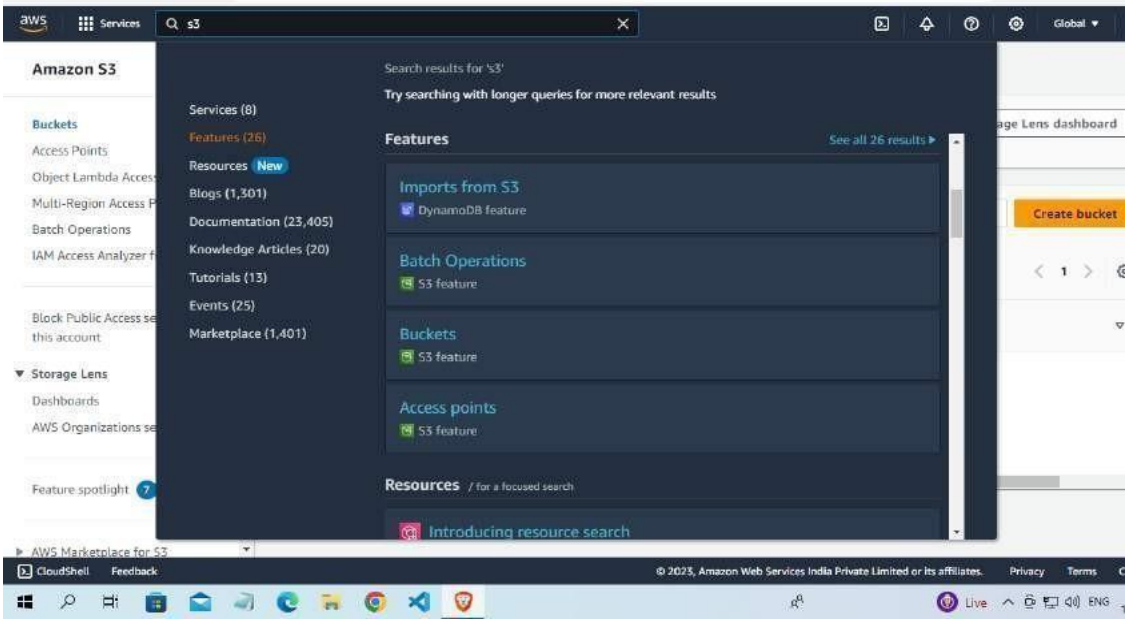
- Scalable Storage: S3 buckets can store an almost unlimited amount of data, making it suitable for organizations of all sizes. You can start with a small amount of storage and scale as needed without any disruption.

- Durability: Data stored in S3 buckets is designed to be highly durable. AWS replicates data across multiple Availability Zones, providing 99.999999999% (11 nines) durability. This means data is protected against hardware failures, and even if an entire Availability Zone goes down, your data is still safe.

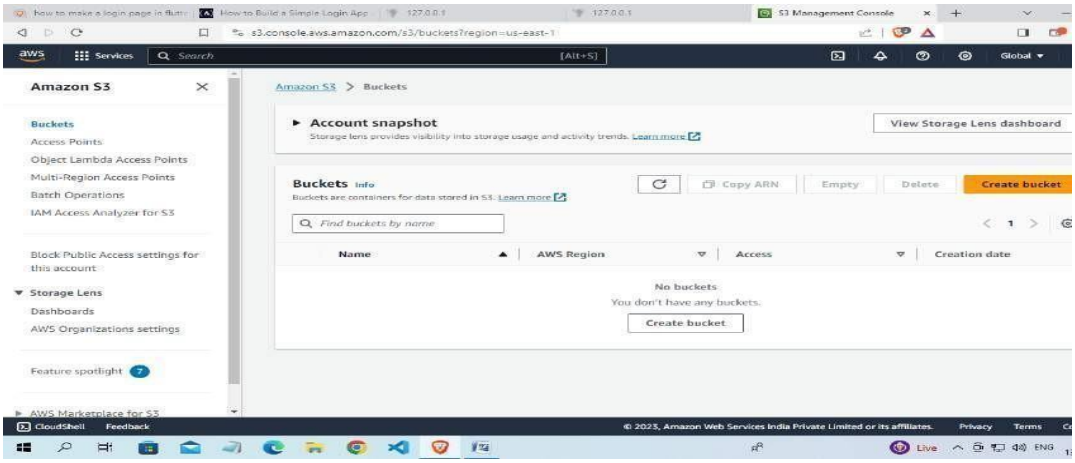
- Data Availability: S3 provides high data availability. Your data is accessible over the internet, and you can access it from anywhere with an internet connection. This makes it a suitable solution for content delivery and web hosting.

- Data Security: S3 buckets offer various security features, including access control through bucket policies, IAM roles, and Access Control Lists (ACLs). You can also enable server-side encryption to protect data at rest.

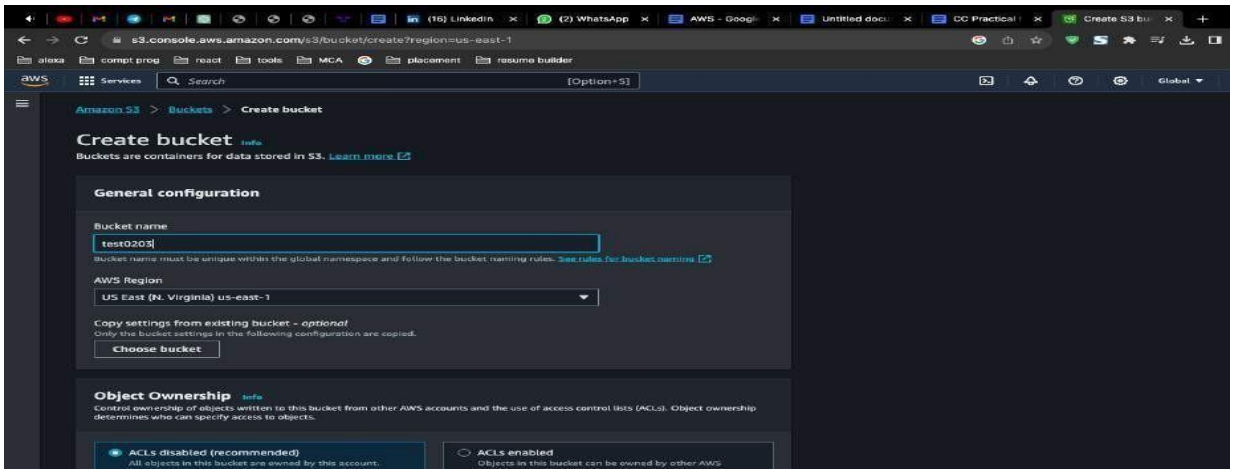
Step 1: Once logged in, navigate to the S3 dashboard. You can do this by searching for "S3" in the AWS Management Console's search bar or by selecting "Storage" and then "S3" under the "Services" menu.



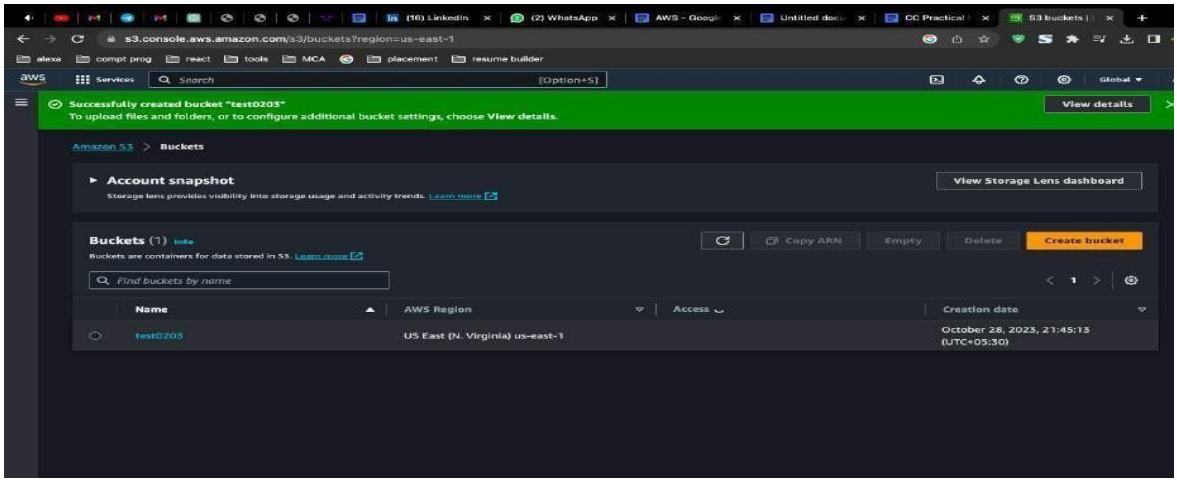
Step 2: Click the "Create bucket" button.



Step 3: In the "Bucket name" field, enter a unique and globally-unique name for your bucket. Bucket names must be unique across all of AWS.



Step 4: Review your configuration and click the "Create bucket" button to create your S3 bucket.

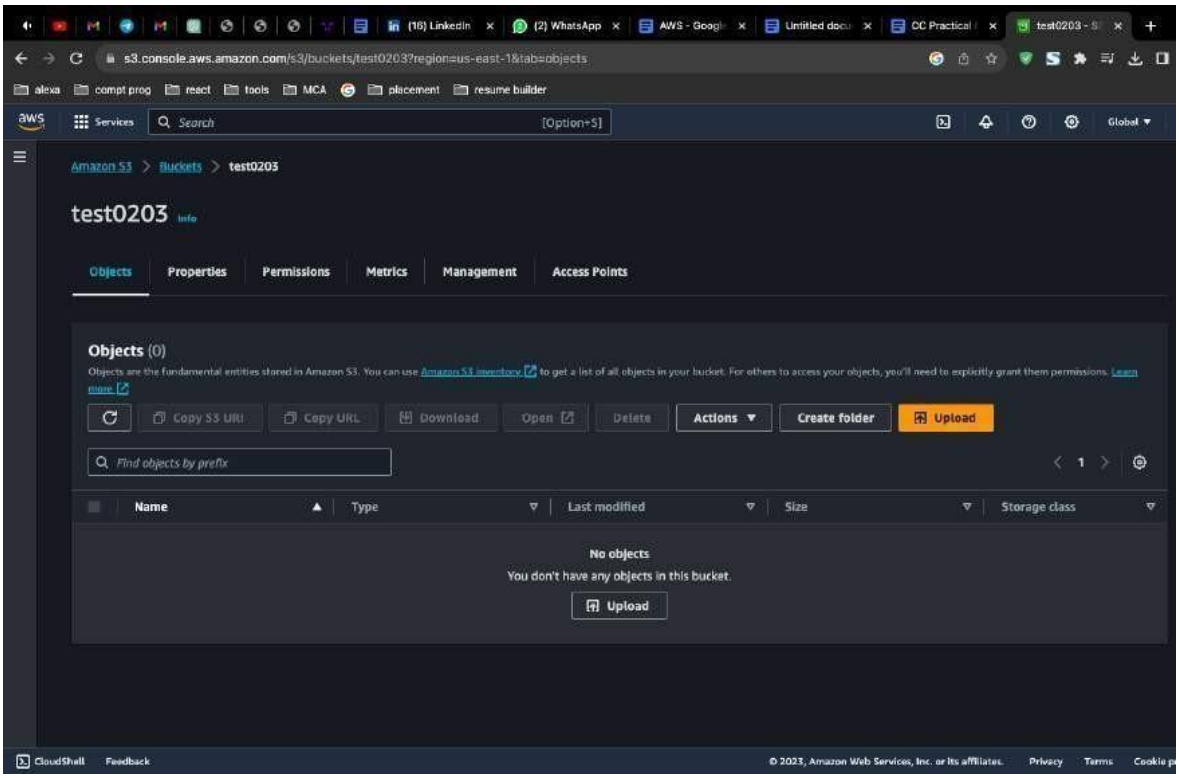


PRACTICAL 14

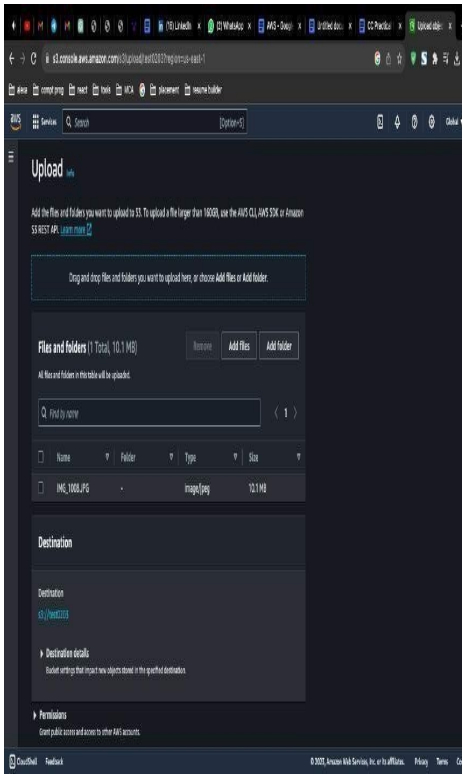
Upload an object on bucket created by using S3 AWS service

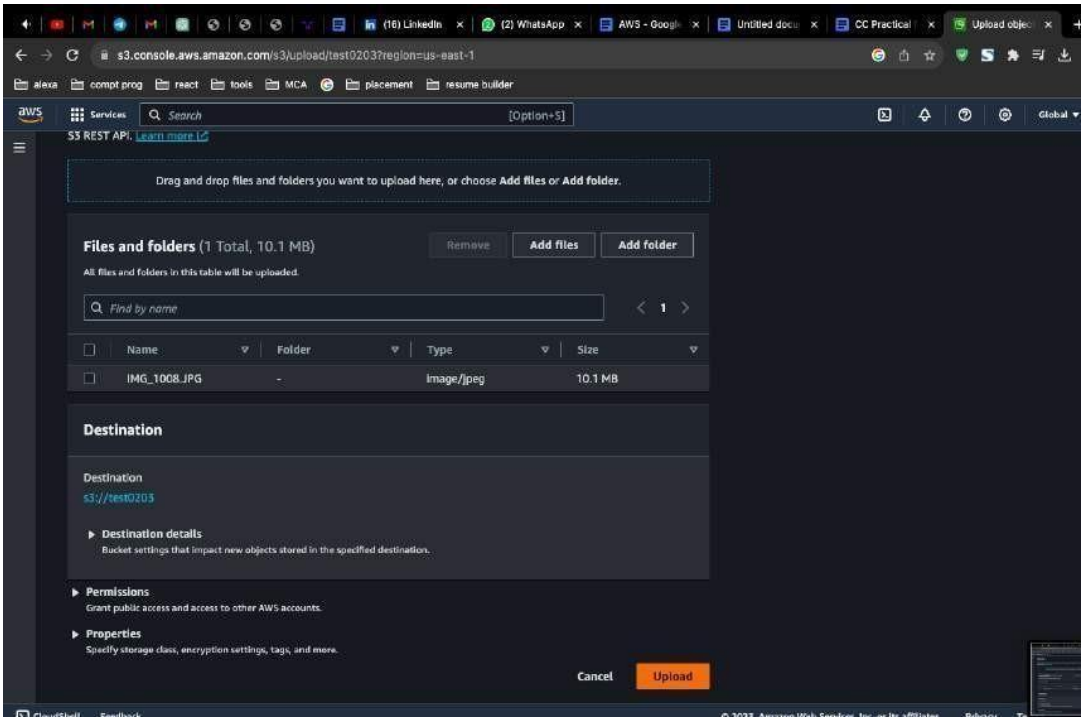
Objective: The objective of the practical task "Upload an object on bucket created by using S3 AWS service"
is to demonstrate how to upload files to an Amazon S3 bucket This can be useful for various purposes, such as hosting static websites, sharing files, or distributing public content.

Step 1: Select the object (file) you want to make publicly accessible by checking the checkbox next to its name.

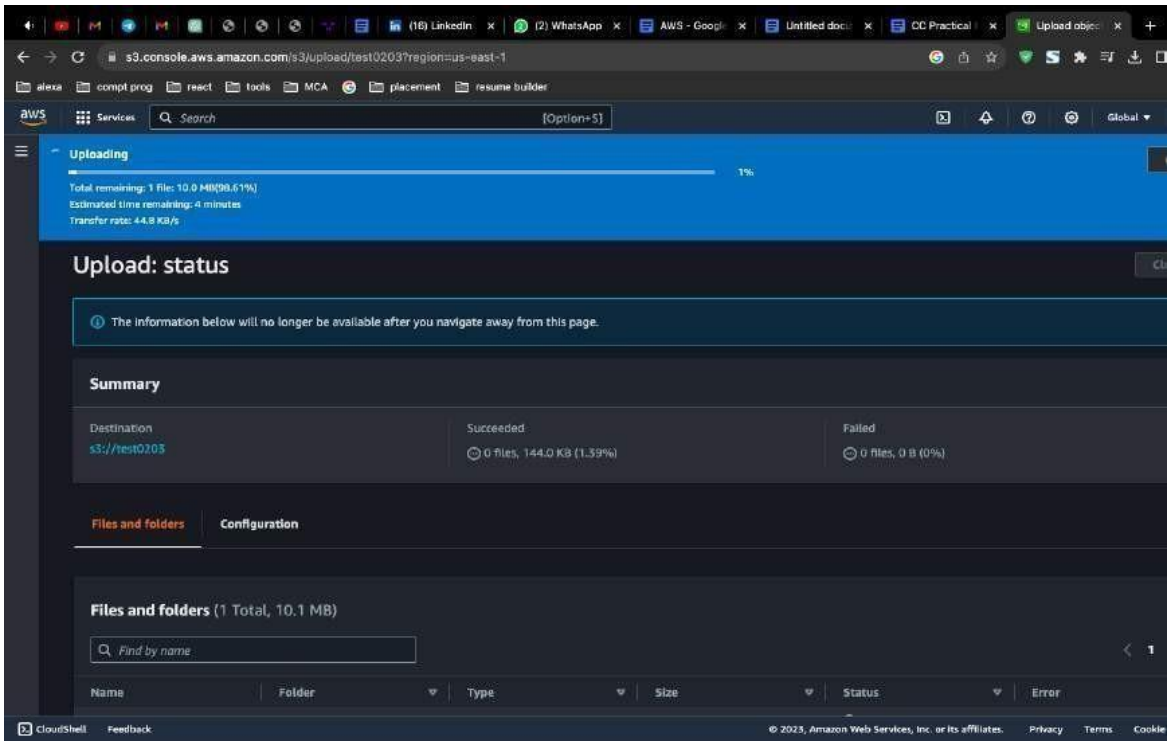


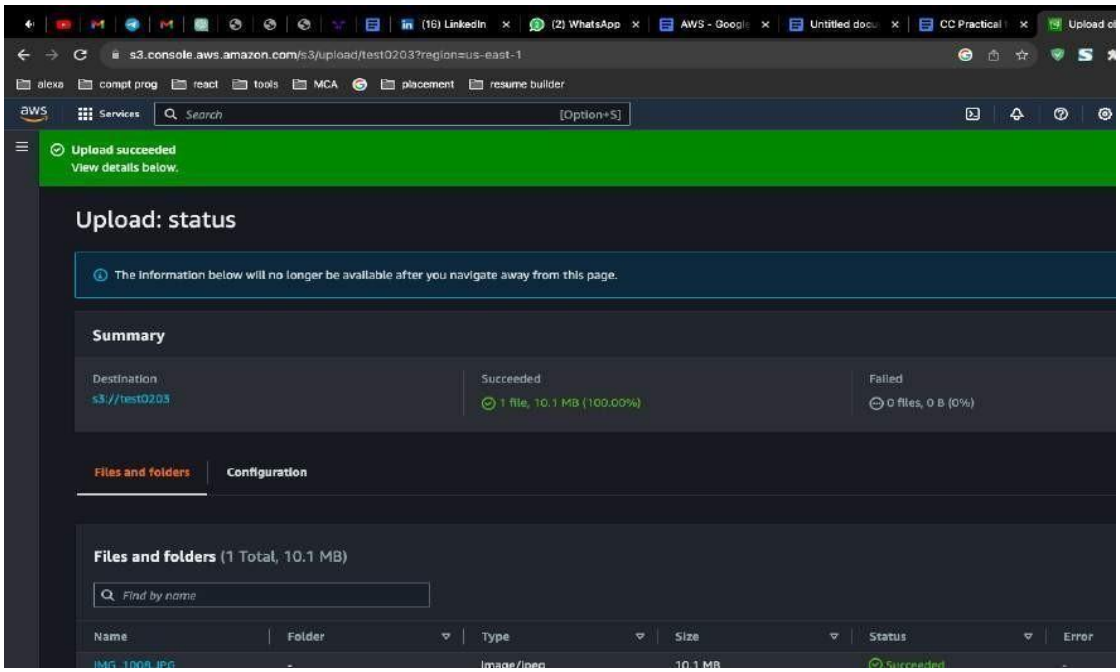
Step 2: Click the "Actions" button, and from the dropdown menu, select "Make public."
Note: If you want all objects in the bucket to be public, you can modify the bucket's access control settings to allow public access.


Upload
Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more
Drag and drop files and folders you want to upload here, or choose Add files or Add folder.
Files and folders (1 Total, 10.1 MB)
Remove
Add files
Add folder
All files and folders in this table will be uploaded.
Find by name
Name
Folder
Type
Size
IMG_1008.JPG
image/jpeg
10.1 MB
Destination
Destination
s3://test0203
Destination details
Bucket settings that impact new objects stored in the specified destination.
Permissions
Grant public access and access to other AWS accounts.


(16) LinkedIn
(2) WhatsApp
AWS - Google
Untitled docu
CC Practical
Upload objec
s3.console.aws.amazon.com/s3/upload/test0203?region=us-east-1
alexa
compt prog
react
tools
MCA
placement
resume builder
aws
Services
Search
[Option+S]
Global
S3 REST API. Learn more
Drag and drop files and folders you want to upload here, or choose Add files or Add folder.
Files and folders (1 Total, 10.1 MB)
Remove
Add files
Add folder
All files and folders in this table will be uploaded.
Find by name
1
Name
Folder
Type
Size
IMG_1008.JPG
-
image/jpeg
10.1 MB
Destination
Destination
s3://test0203
Destination details
Bucket settings that impact new objects stored in the specified destination.
Permissions
Grant public access and access to other AWS accounts.
Properties
Specify storage class, encryption settings, tags, and more.
Cancel
Upload

Step 3:
File Uploaded in Buket





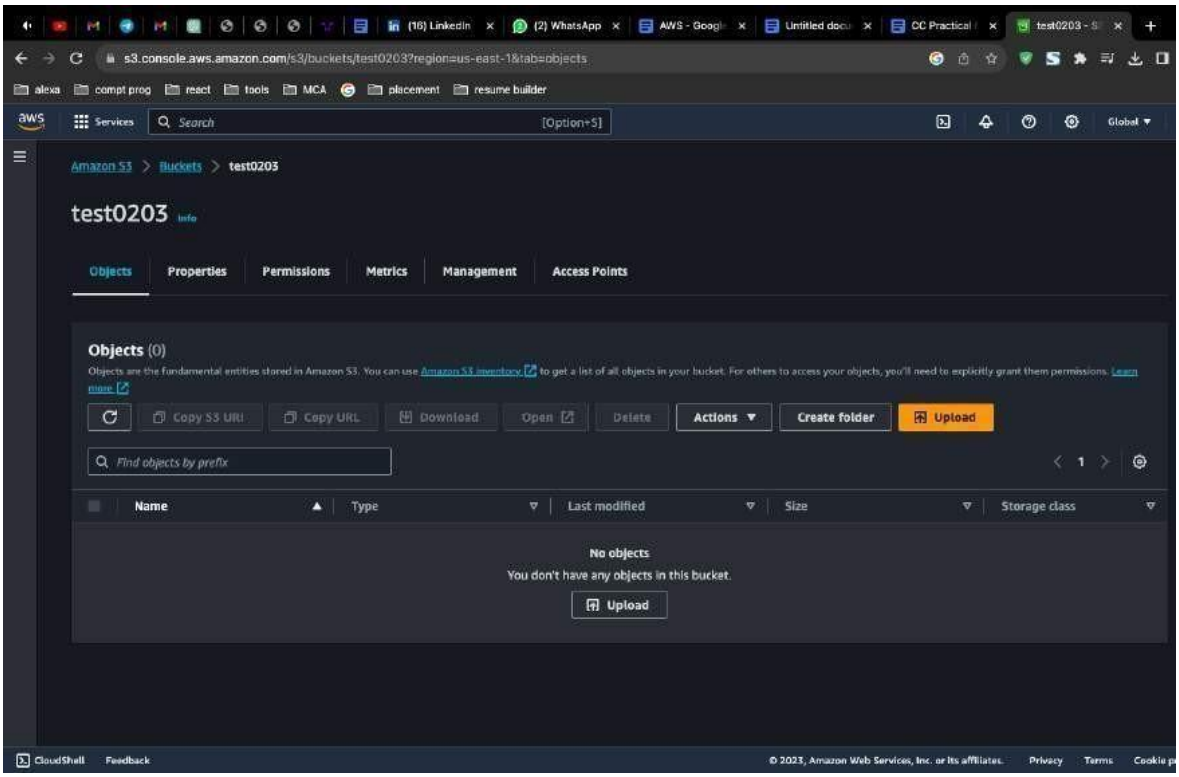
PRACTICAL 15

Create a bucket and allow public access on uploaded object by using object URL & S3 AWS service.

Objective : The objective of the practical task "Upload objects and enable public access via URL in a created Amazon S3 bucket" is to demonstrate how to upload files to an Amazon S3 bucket and configure the bucket settings to make those objects publicly accessible via a URL. This can be useful for various purposes, such as hosting static websites, sharing files, or distributing public content.

Step 1:

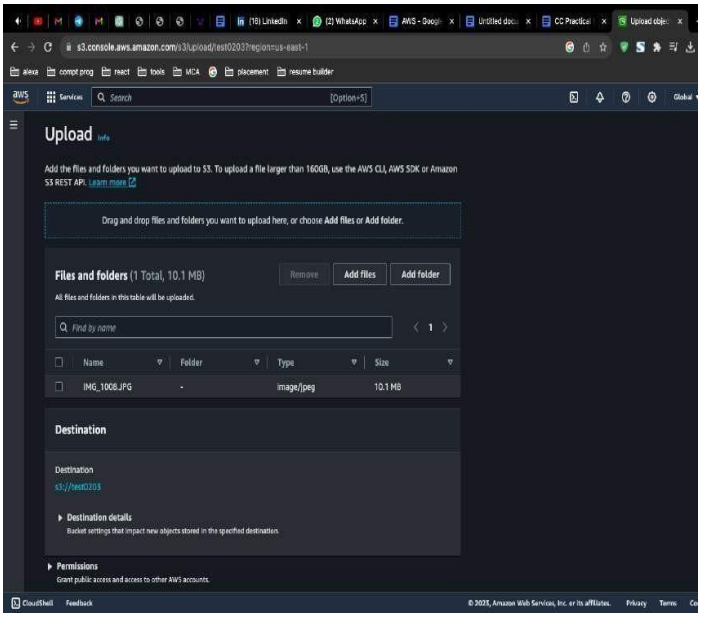
Select the object (file) you want to make publicly accessible by checking the checkbox next to its name.

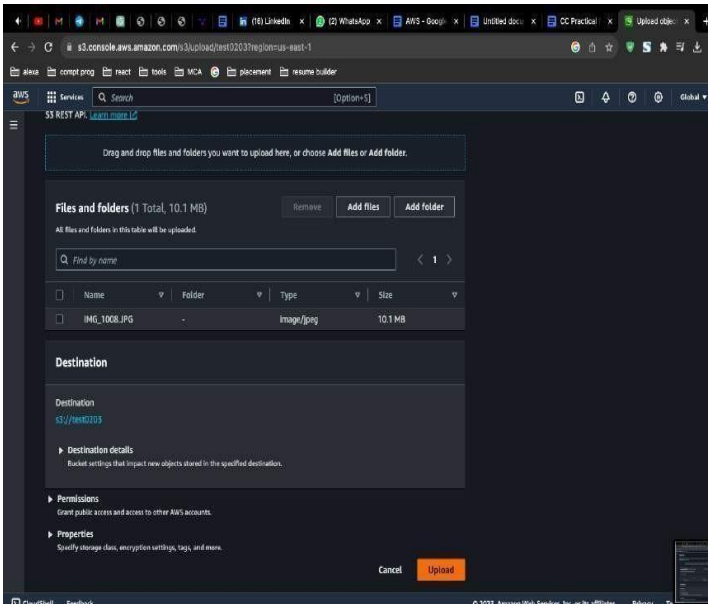


Step 2:

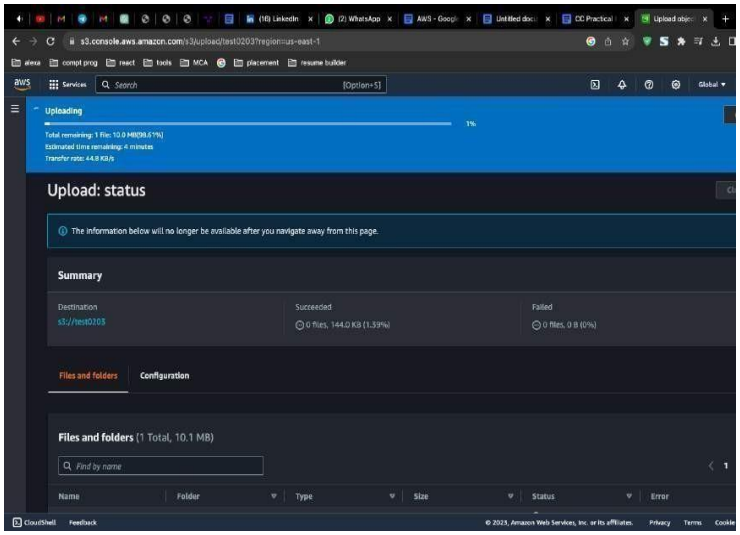
Click the "Actions" button, and from the dropdown menu, select "Make public."

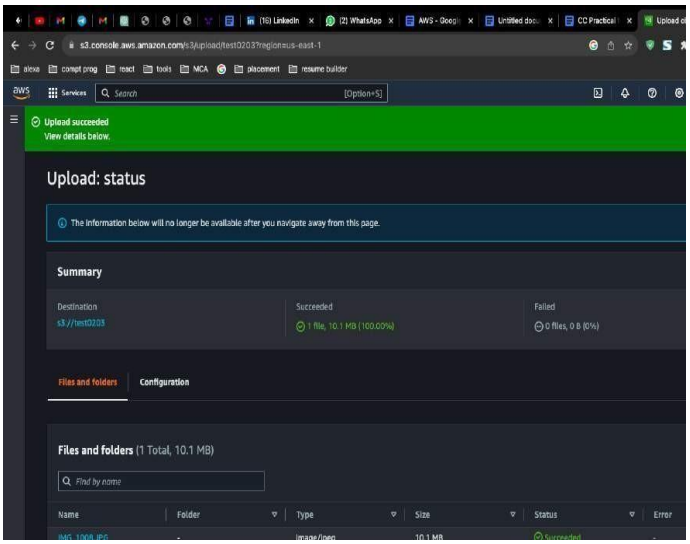
Note: If you want all objects in the bucket to be public, you can modify the bucket's access control settings to allow public access.



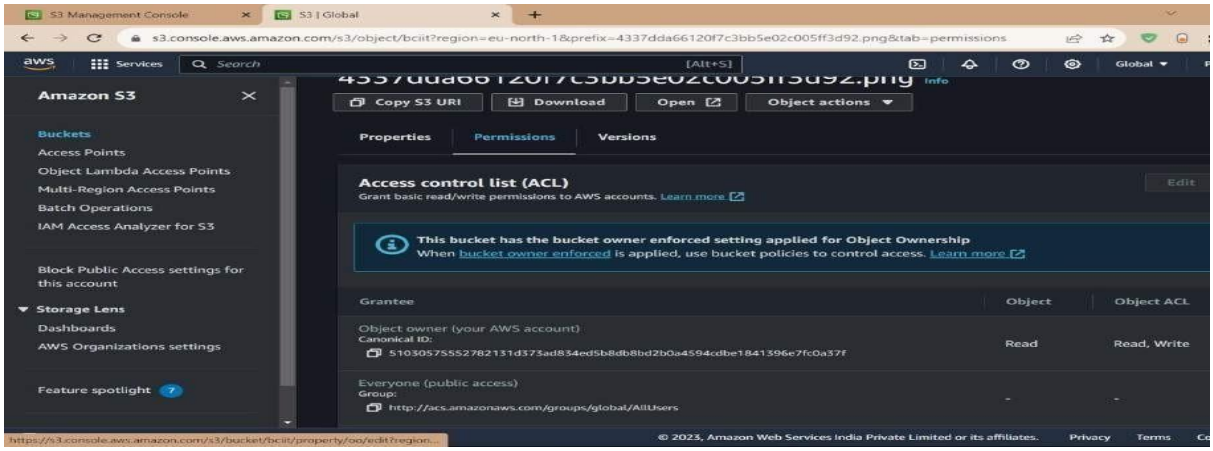


Step 3: File Uploaded in Buket

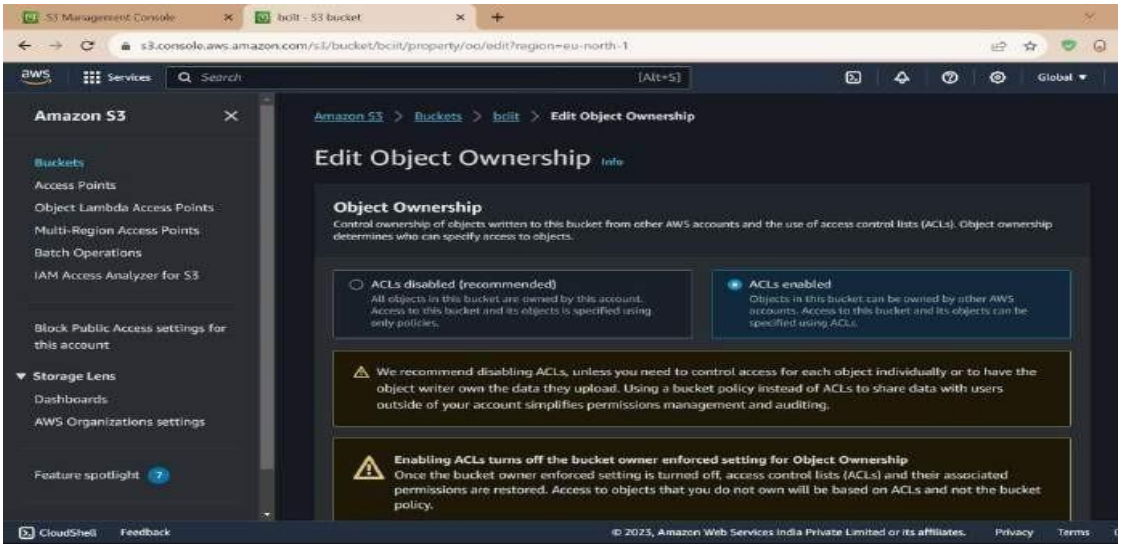




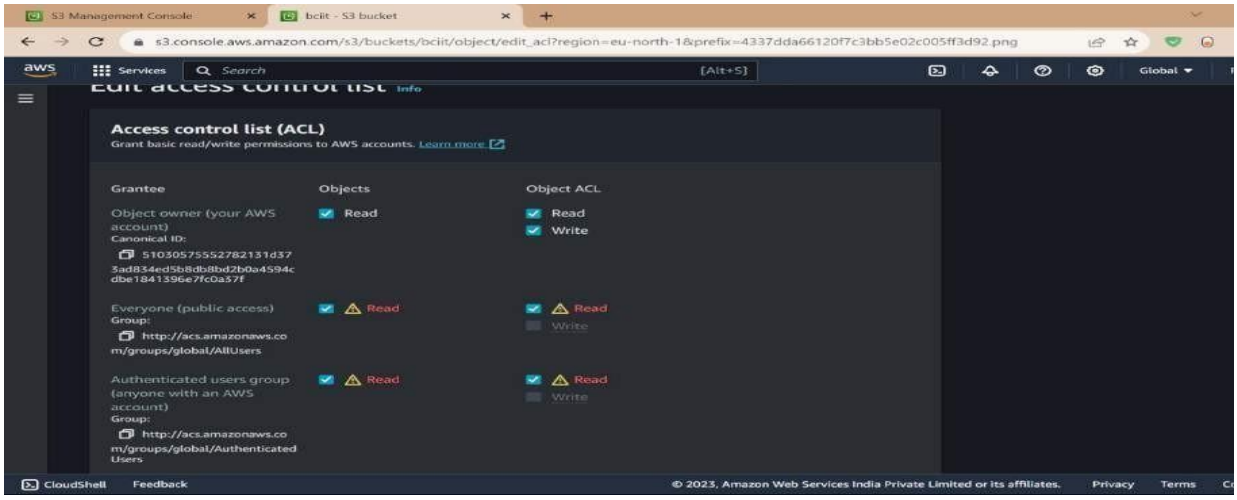
Step 5: Go to permissions and click on bucket owner enforced



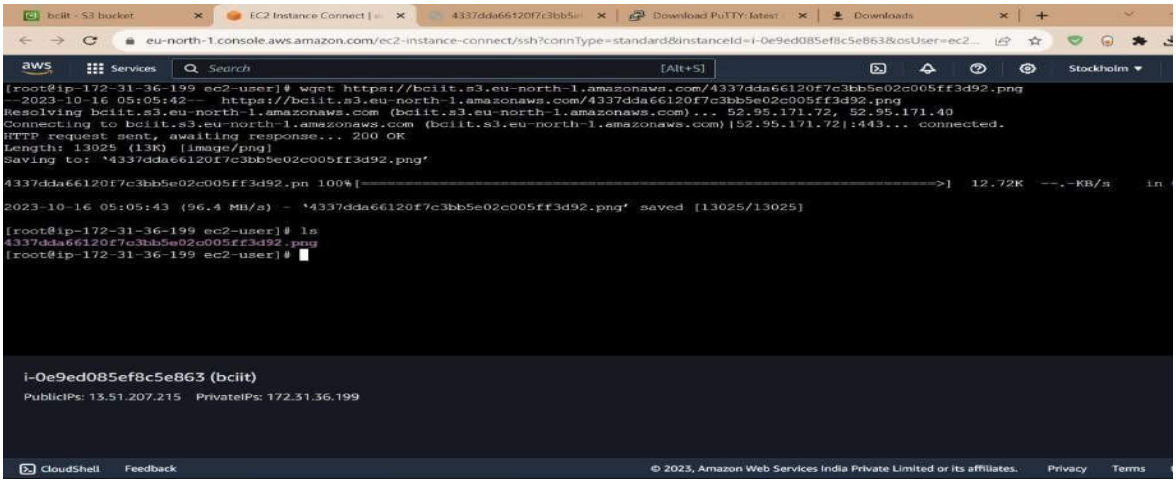
Step 6: Click on ACL2 enabled



Step 7: give access to read and write



Step 8: Navigate to S3 terminal and paste the object URL



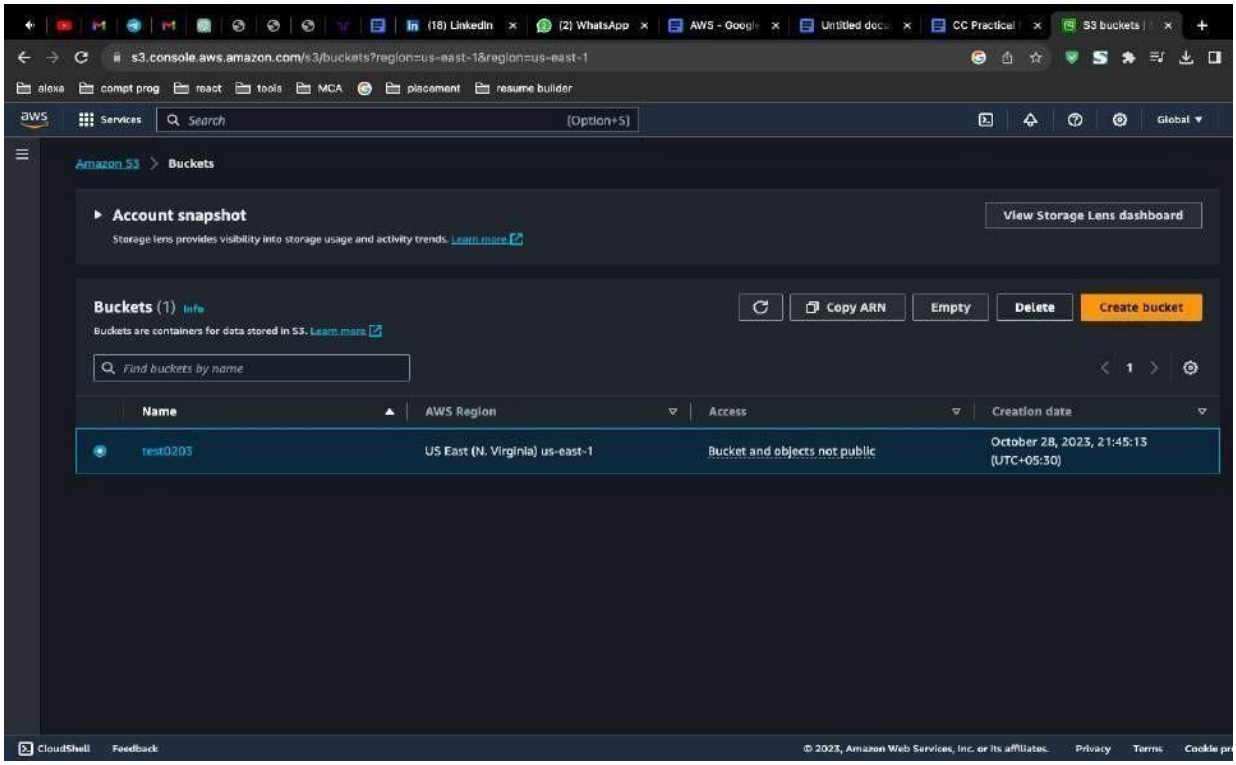
PRACTICAL 16

Delete the created object and its bucket.

Objective: The objective of deleting objects in an Amazon S3 (Simple Storage Service) bucket in AWS can vary depending on the specific use case and needs of the user. Here are some common objectives for deleting objects from an S3 bucket:

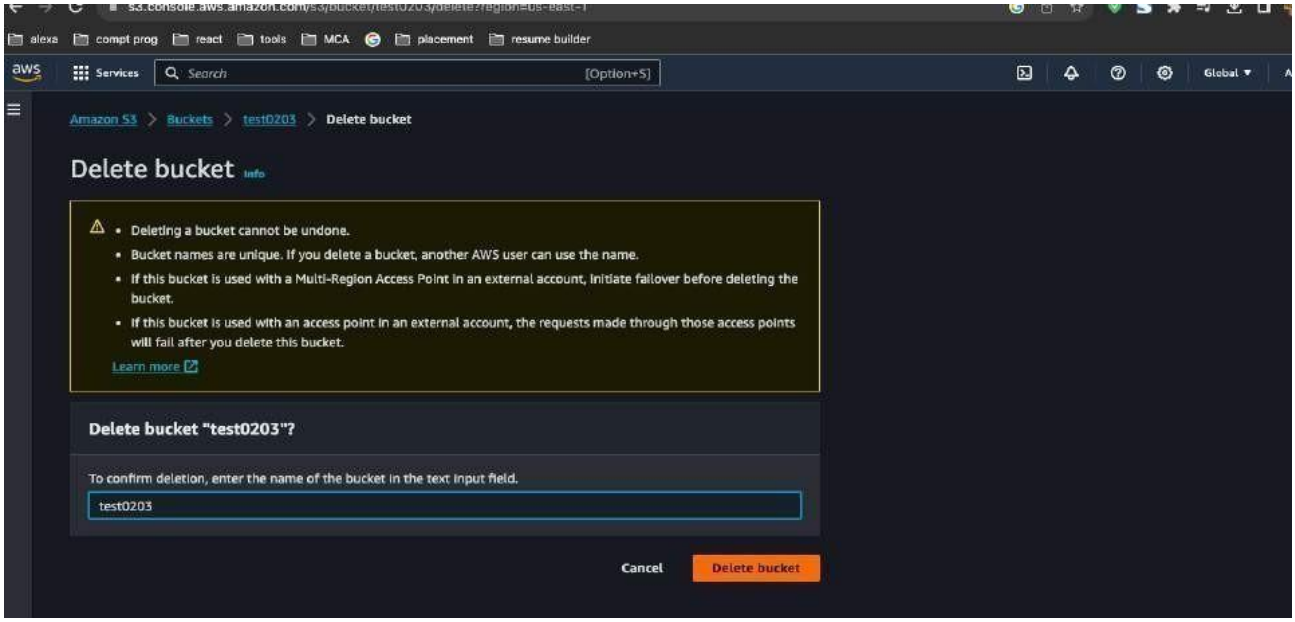
- Data Cleanup: Removing outdated or unnecessary objects to free up storage space and reduce storage costs. Over time, old versions of files or expired data can accumulate, and deleting them helps maintain an efficient storage environment.
- Security: Deleting sensitive or confidential data that is no longer needed to reduce the risk of unauthorized access or data breaches. This is especially important for compliance with data privacy regulations.
- Version Control: Managing versioned objects by removing older versions of files that are no longer relevant. This ensures that only the most up-to-date versions are retained.
- Archiving: Deleting objects that have been archived to more cost-effective storage classes, such as Amazon Glacier, when they are no longer needed in the original S3 bucket.
- Temporary Files: Removing temporary files or objects that were only needed for a specific task or process. This helps in keeping the bucket organized and reducing clutter.

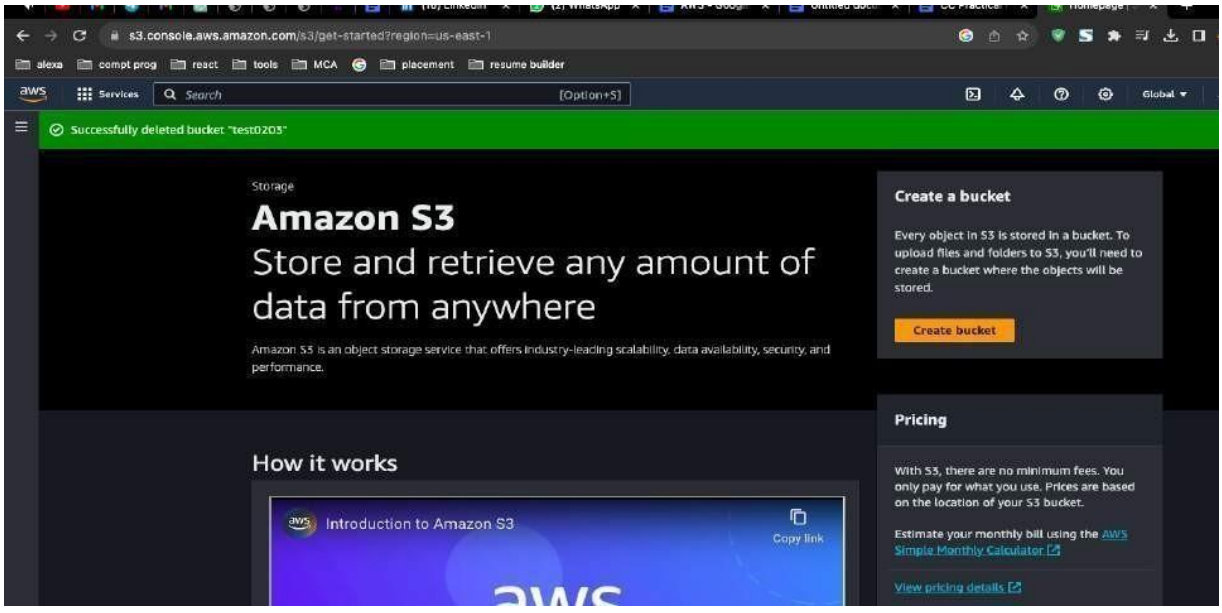
Step 1: In the S3 dashboard, click on the name of the bucket from which you want to delete objects.



Step 2: In the bucket, navigate the objects you want to delete. You can do this by clicking on the folders and subfolders, if applicable, to locate the objects.

- Select the objects you wish to delete by checking the checkboxes next to their names.
- Once the objects are selected, you can choose one of the following methods to delete them:
- Click the "Actions" button, then select "Delete" to delete the selected objects.
- Alternatively, you can simply press the "Delete" key on your keyboard after selecting the objects.
- Confirm the deletion by clicking "Delete" in the confirmation dialog.





PRACTICAL 17

Transfer the object file from S3 service to EC2 launched Linux server install GCC and wget commands in this regard on terminal

Objective: Transfer an object file from S3 to a Linux server running on EC2. Install GCC and wget on the server using terminal commands in AWS.

- Data Cleanup: Removing outdated or unnecessary objects to free up storage space and reduce storage costs. Over time, old versions of files or expired data can accumulate, and deleting them helps maintain an efficient storage environment.

- Security: Deleting sensitive or confidential data that is no longer needed to reduce the risk of unauthorized access or data breaches. This is especially important for compliance with data privacy regulations.

Step 1: Connect to your EC2 instance via SSH. Open a terminal application.
Type the ssh command followed by the username and IP address of your EC2 instance. For example:

```
ssh username@ec2-public-ip-address
```

Enter your password when prompted.

Step 2: Install wget:
Type the following command to update the package list:

Ubuntu/Debian:

```
sudo apt install wget
```

Step 3: Get the S3 object URL:

1. In the AWS Management Console, navigate to the S3 bucket containing the object file.
2. Right-click the object and select "Get Object URL".
3. Copy the URL to your clipboard.

Step 4: Transfer the object file to EC2 using wget:

1. In the SSH terminal window, type the following command, replacing URL_OF_S3_OBJECT with the URL you copied:

```
wget URL_OF_S3_OBJECT
```

2. Press Enter. The file will be downloaded to your current directory on the EC2 instance.

Step 5: Verify the file transfer:

1. Type the following command to list the files in your current directory:

```
ls -l
```

2. Make sure the object file is listed.

Step 6: Install GCC:

1. Update the package list again:
2. Install GCC using the appropriate command for your Linux distribution:

Ubuntu/Debian:

```
sudo apt install gcc
```

Step 7: Verify GCC installation:

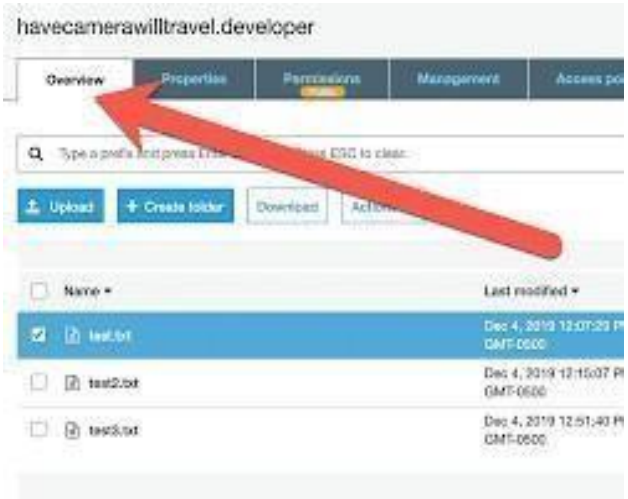
1. Type the following command to check the installed GCC version:

```
gcc -version
```

2. The command should display the installed GCC version.

Screenshots:

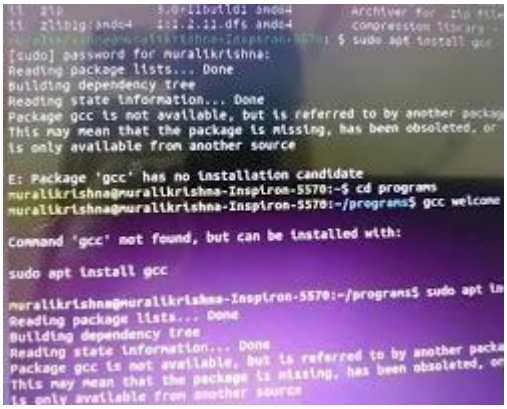
1. **Object URL in S3 Console:**



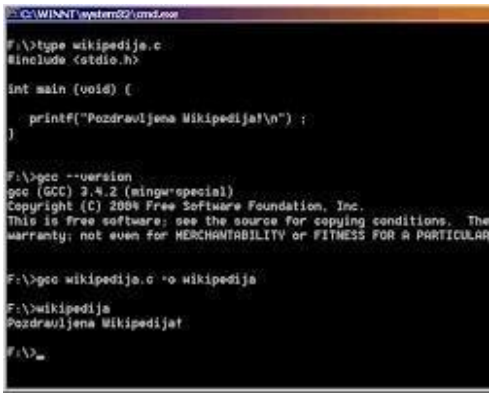
2. **wget command in terminal:**

3. File listing showing downloaded object:

4. GCC installation command:

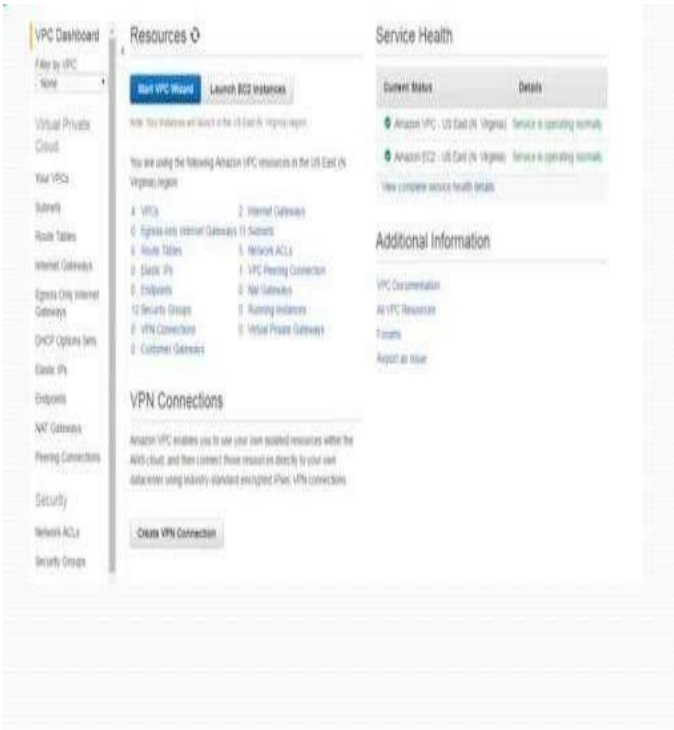


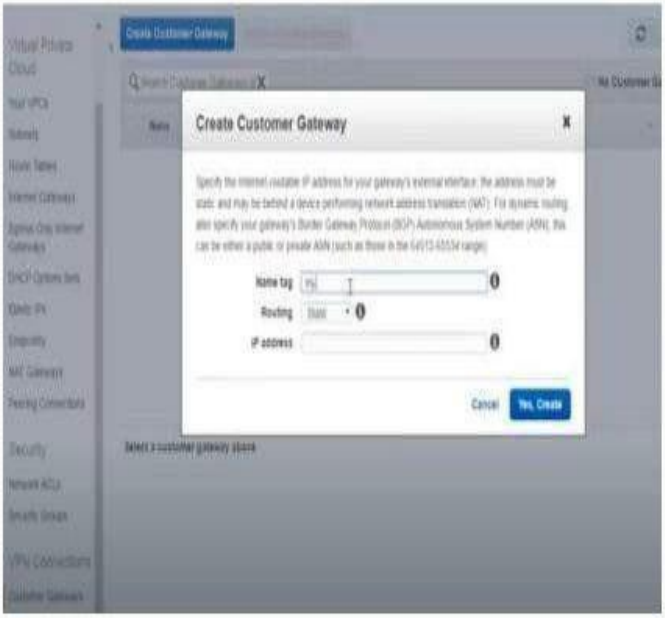
5. GCC version verification:

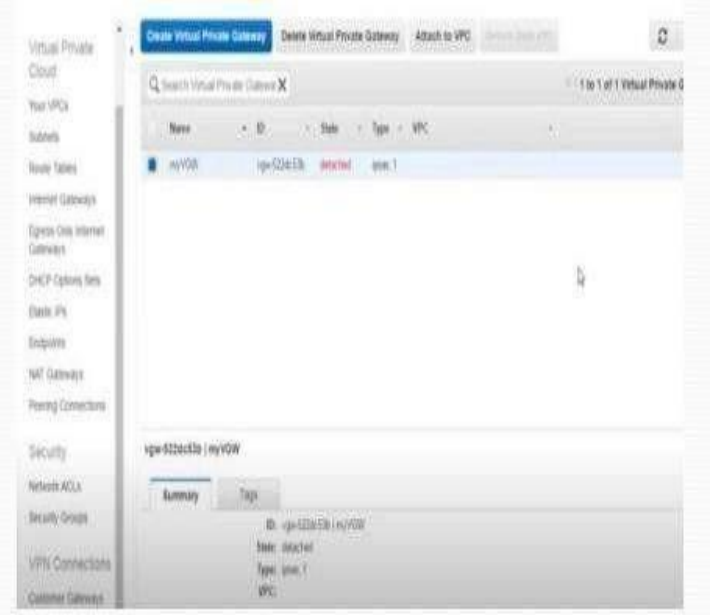


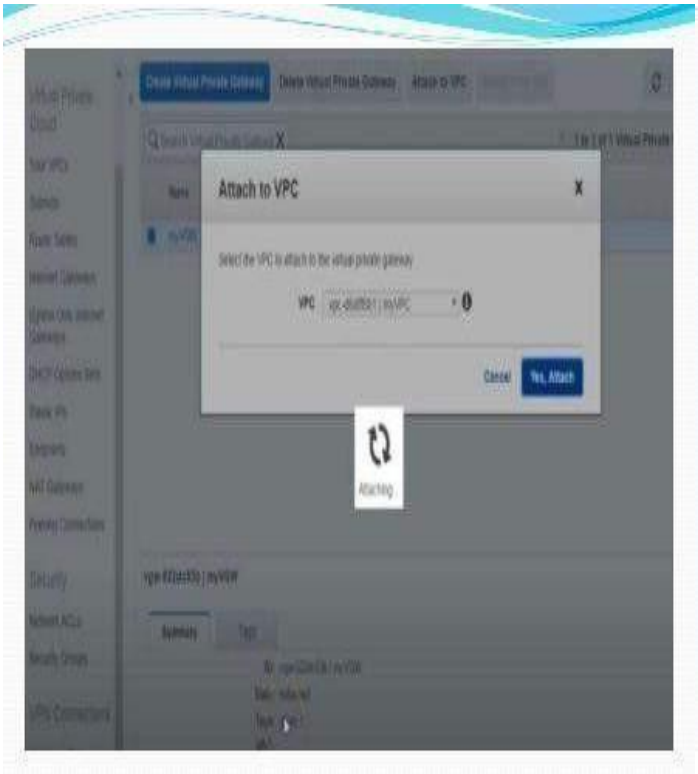
PRACTICAL 18

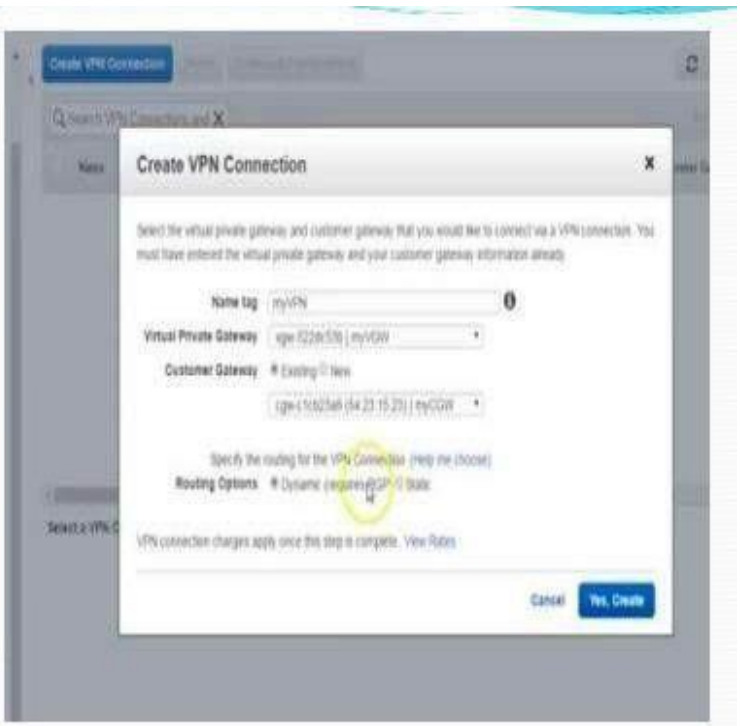
Create VPC and implement EC2 services on it.

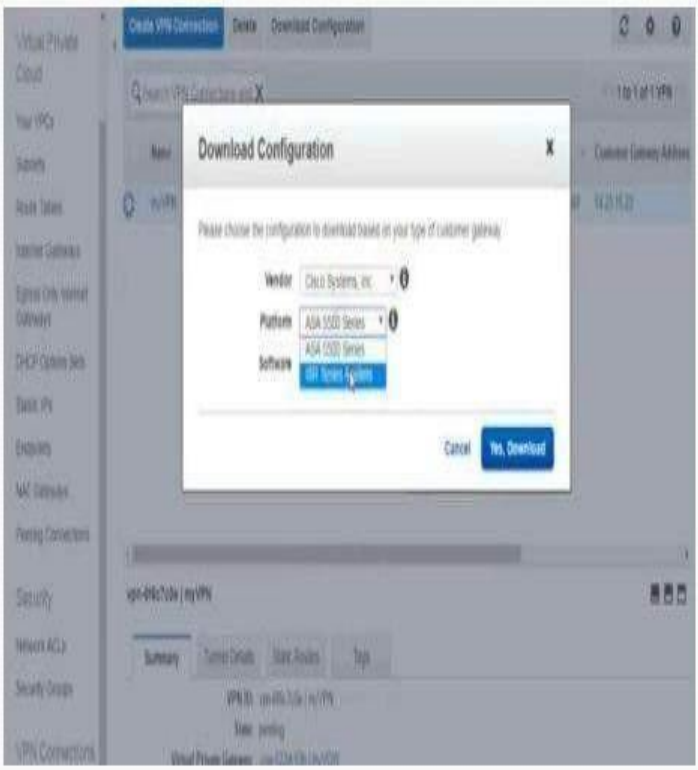


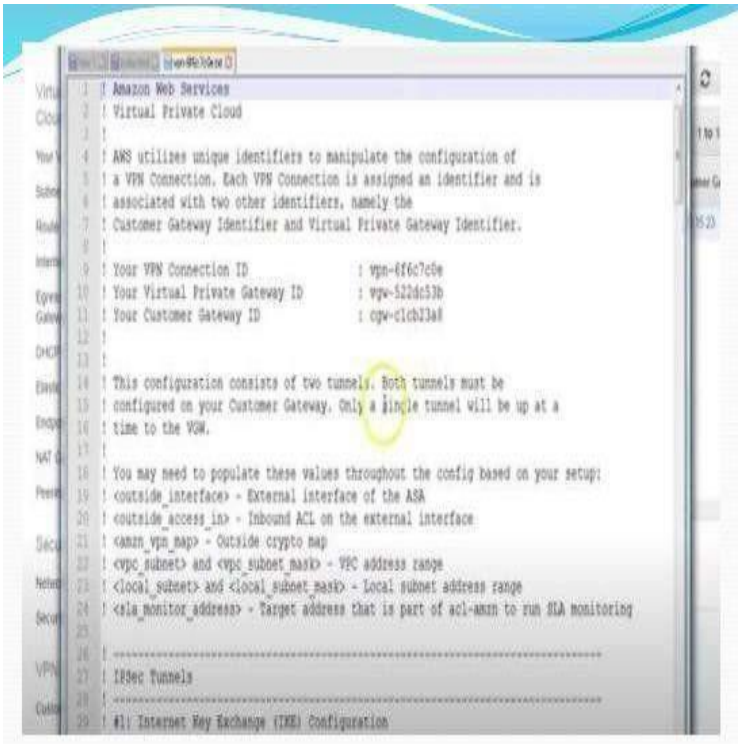





Attach to VPC
Cancel
Yes, Attach
Attaching

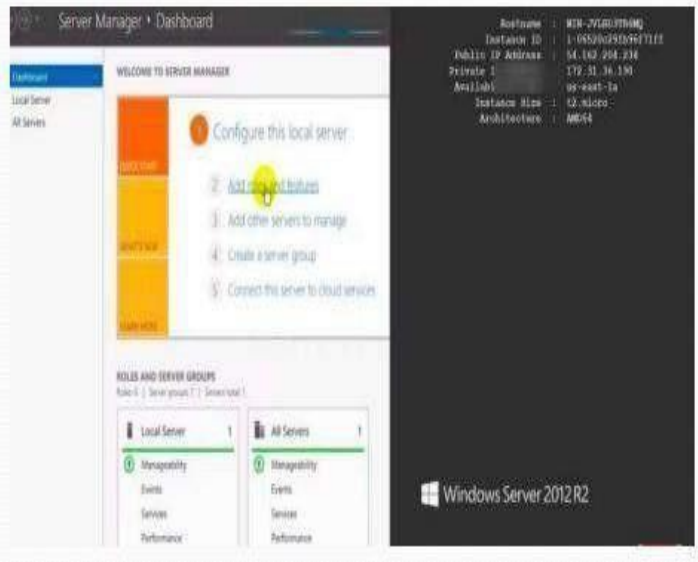

Create VPN Connection
Name tag
Virtual Private Gateway
Customer Gateway
Routing Options
Cancel
Yes, Create

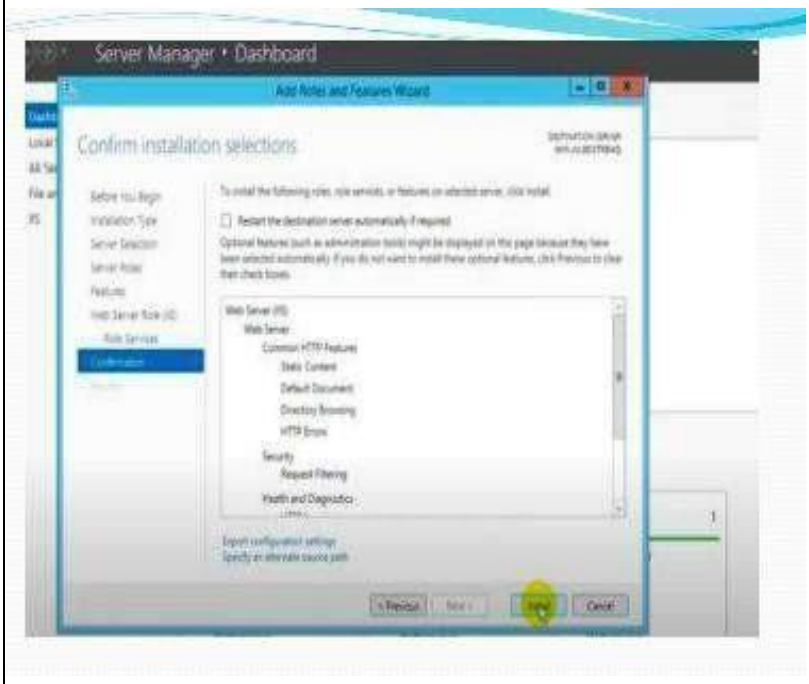

Download Configuration
Please choose the configuration to download based on your type of customer gateway.
Vendor
Platform
Software
Cancel
Yes, Download

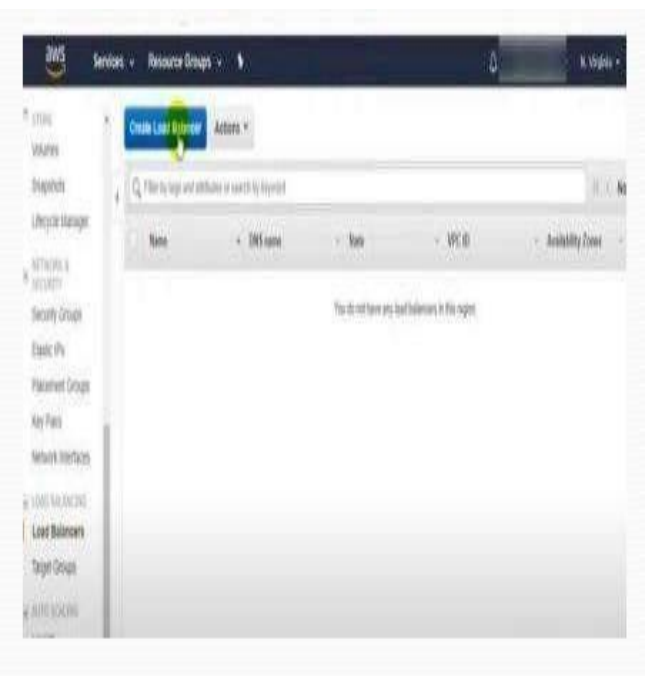

Amazon Web Services
Virtual Private Cloud
ANS utilizes unique identifiers to manipulate the configuration of
a VPN Connection. Each VPN Connection is assigned an identifier and is
associated with two other identifiers, namely the
Customer Gateway Identifier and Virtual Private Gateway Identifier.
Your VPN Connection ID : vpn-6f6c7c0e
Your Virtual Private Gateway ID : vgw-522dc53b
Your Customer Gateway ID : cgw-c1cb23a8
This configuration consists of two tunnels. Both tunnels must be
configured on your Customer Gateway. Only a single tunnel will be up at a
time to the VGW.
You may need to populate these values throughout the config based on your setup:
<outside_interface> - External interface of the ASA
<outside_access_in> - Inbound ACL on the external interface
<amzn_vpn_map> - Outside crypto map
<vpc_subnet> and <vpc_subnet_mask> - VPC address range
<local_subnet> and <local_subnet_mask> - Local subnet address range
<sla_monitor_address> - Target address that is part of acl-amzn to run SLA monitoring
IPSec Tunnels
#1: Internet Key Exchange (IKE) Configuration

PRACTICAL 19

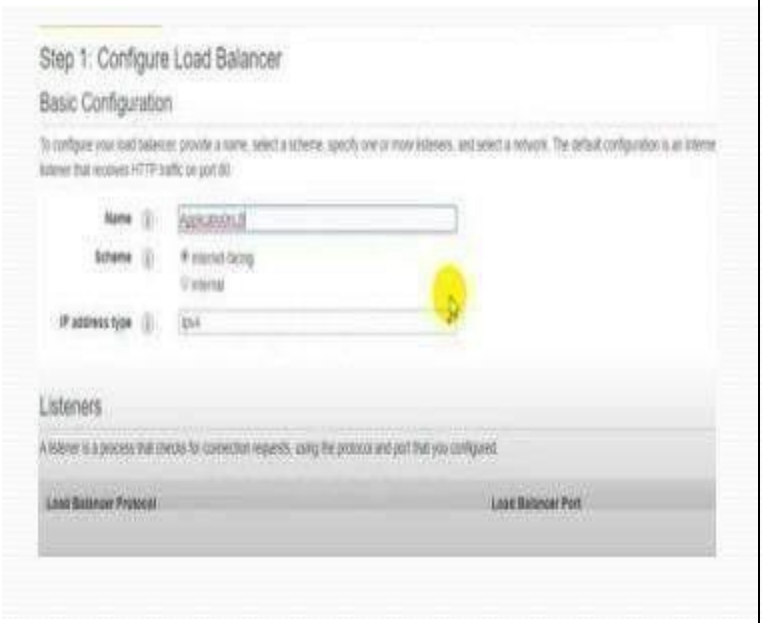
Implement & Configure load balancing with all necessary steps.

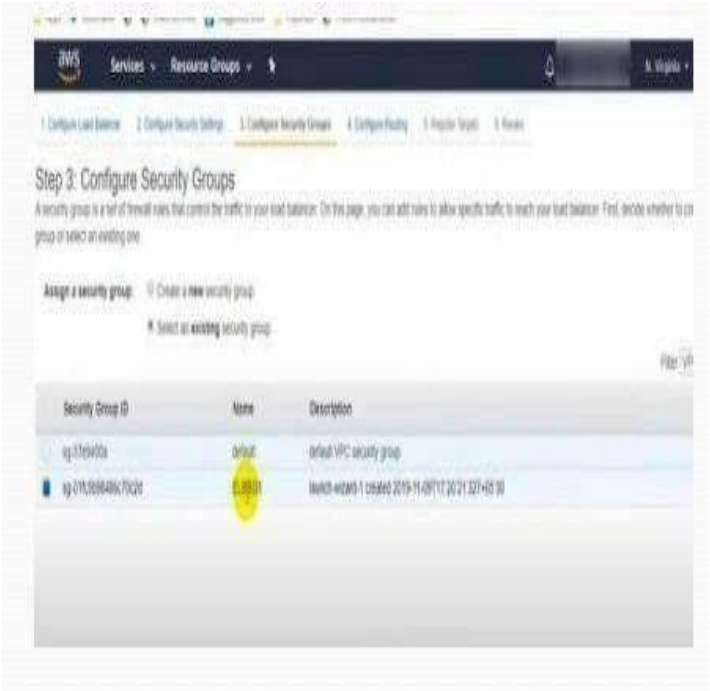


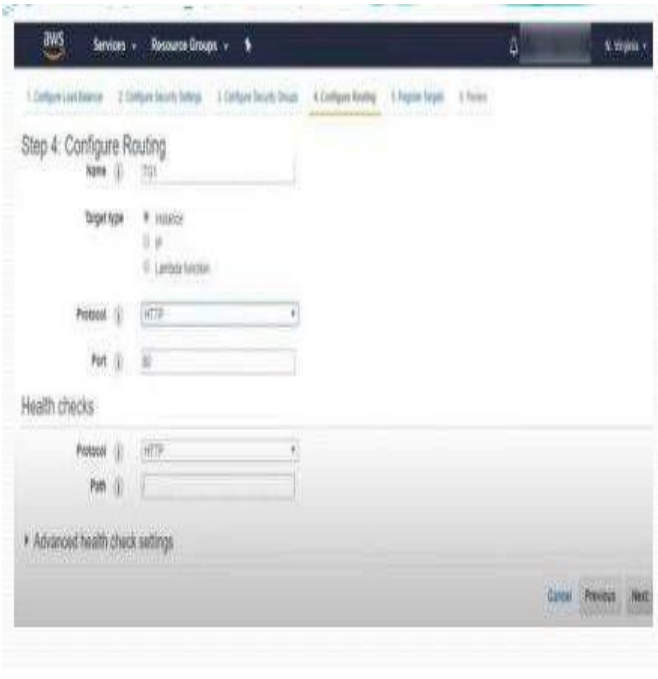


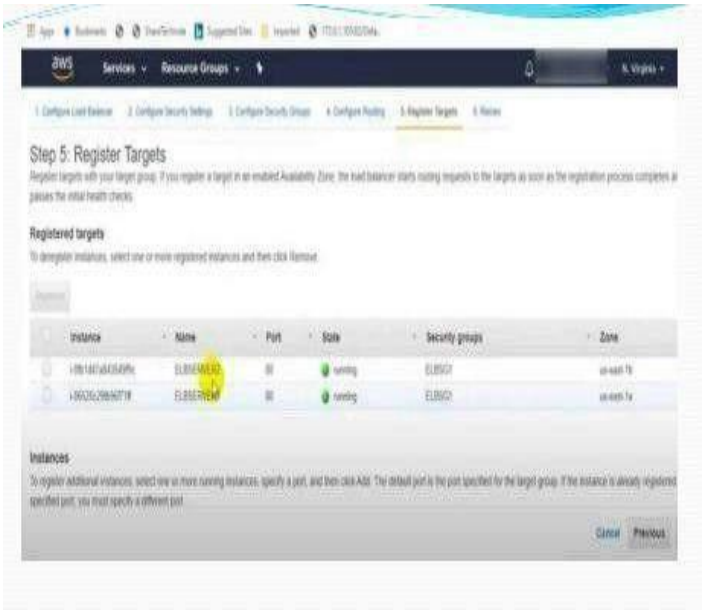


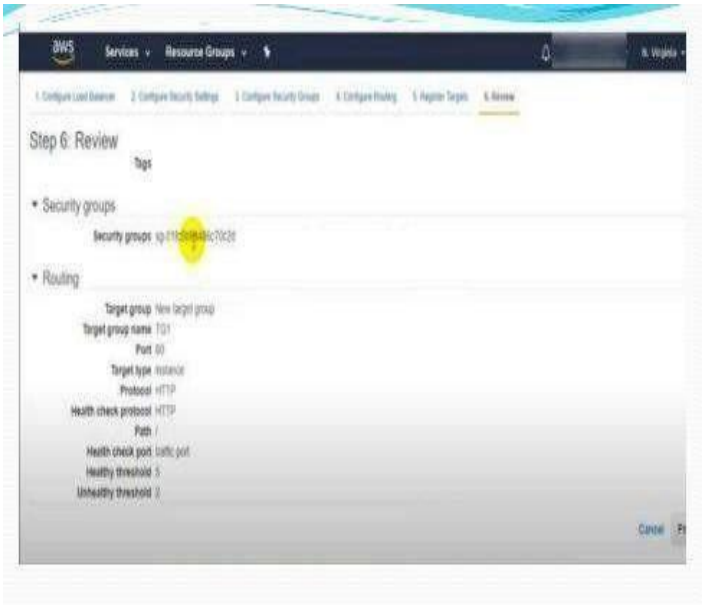

Select load balancer type
Application Load Balancer
HTTP
HTTPS
Create
Network Load Balancer
TCP
TLS
UDP
Create
Classic Load Balancer
PREVIOUS GENERATION
Create
Learn more

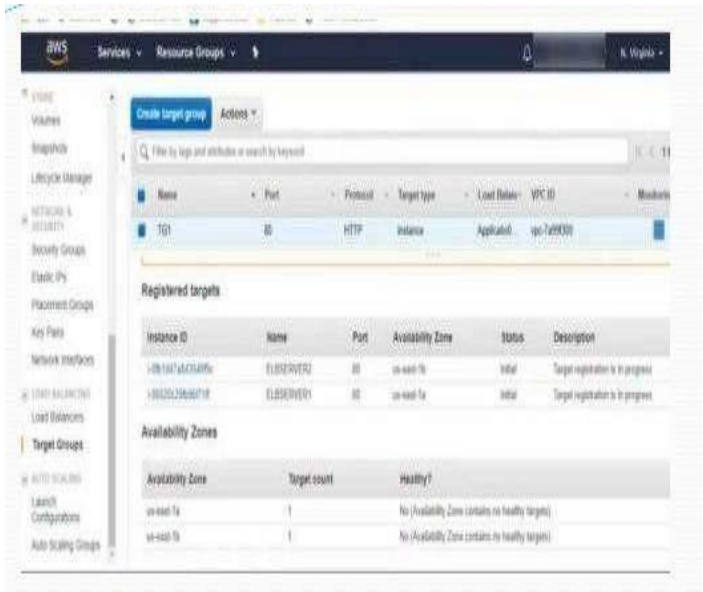

Step 1: Configure Load Balancer
Basic Configuration
Name
Scheme
IP address type
Listeners
A listener is a process that checks for connection requests, using the protocol and port that you configured.
Load Balancer Protocol
Load Balancer Port

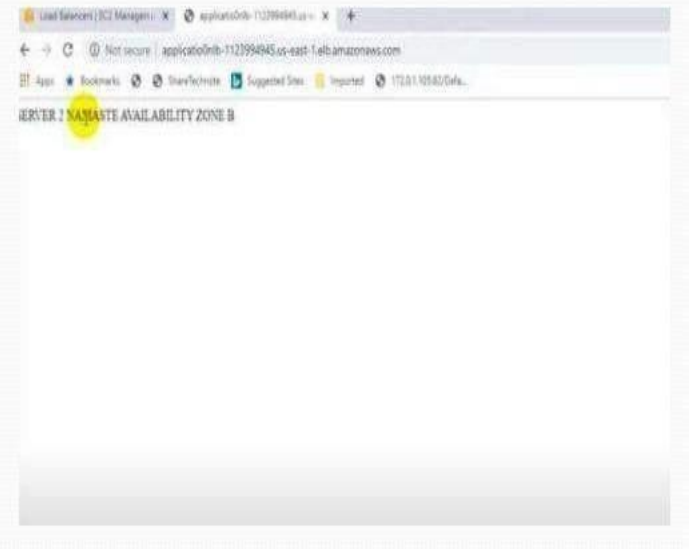

Services
Resource Groups
Step 3: Configure Security Groups
Assign a security group:
Security Group ID
Name
Description


Services
Resource Groups
Step 4: Configure Routing
Name
Target type
Protocol
Port
Health checks
Protocol
Path
Advanced health check settings
Cancel
Previous
Next


Step 5: Register Targets
Registered targets
Instances


Step 6: Review
Security groups
Routing


Create target group
Actions
Registered targets
Availability Zones


Not secure
applicatio0nlb-1123994945.us-east-1.elb.amazonaws.com
SERVER 2 NAJIASTE AVAILABILITY ZONE B


applicatio0nlb-1123994945.us-east-1.elb.amazonaws.com/
SERVER 1 HELLO AVAILABILITY ZONE A

PRACTICAL 20

How to handle a cloud shell. Explain it

Steps to use AWS Cloud shell:

Using AWS CloudShell is a convenient way to access the AWS Command Line Interface (CLI) and various AWS services directly from your web browser,

Here are the steps to use AWS CloudShell:

Login to the AWS Management Console:
Ensure you have an AWS account
and are logged into the AWS Management Console.

Access AWS CloudShell:
Once you're logged in, you can access AWS CloudShell
from the AWS Management Console. You can find it in the top-right corner of the AWS Management Console, labeled as "AWS CloudShell."

Initialize the Environment:
The first time you access CloudShell, it may take a moment to initialize your environment. Once it's ready, you'll be presented with a command-line interface.

Use the AWS CLI and AWS SDKS:
CloudShell comes pre-configured with the AWS CLI and various AWS SDKs.

1. You can use these tools to interact with AWS services. For example, you can run AWS CLI commands, Python scripts, or

2. use any of the supported SDKs to manage your AWS resources.
Customize Your Environment (optional): You can customize your CloudShell environment by installing additional packages

3. or configuring your shell as per your preferences. You can use package managers like pip, npm, or brew to install

Save Your Work: AWS CloudShell provides you with home directory
storage that is persistent, even across sessions. This means you can save your scripts, configuration files, and other data within your home directory.

Exit CloudShell: When you're done with your session, you can type exit to exit CloudShell. Your home directory data will persist for the next time you log in.

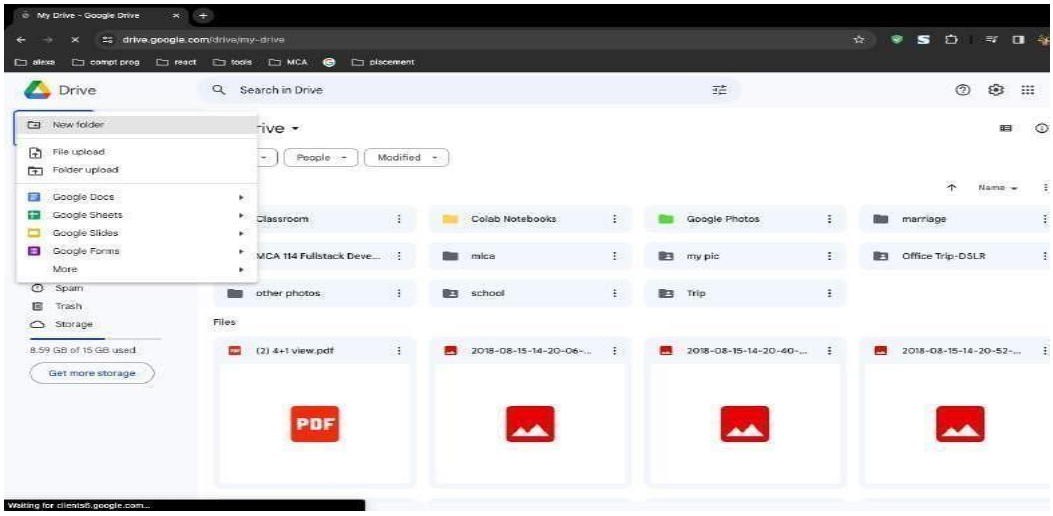
PRACTICAL 21

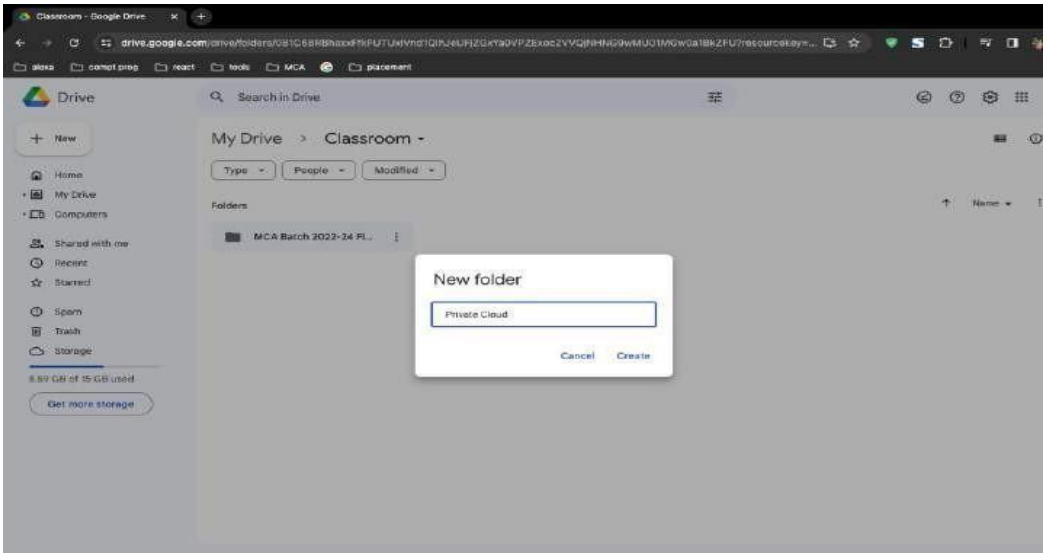
Create a private cloud on google drive and grant permission for the user.

Steps to Create a Private Cloud on Google Drive:

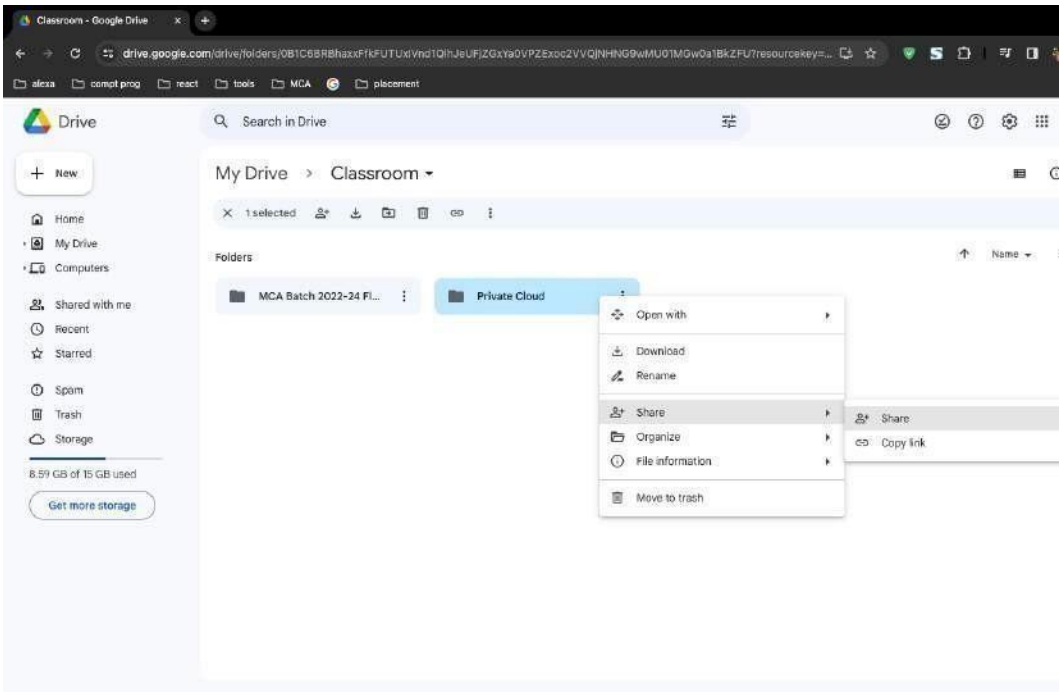
1. Sign in to Google Drive:
Open your web browser and go to Google Drive.
Sign in with your Google account or create one if you don't have it.

2. Create a New Folder:
Click on the "+ New" button on the left side.
Choose "Folder" to create a new folder.
Name the folder appropriately, e.g., "Private Cloud."

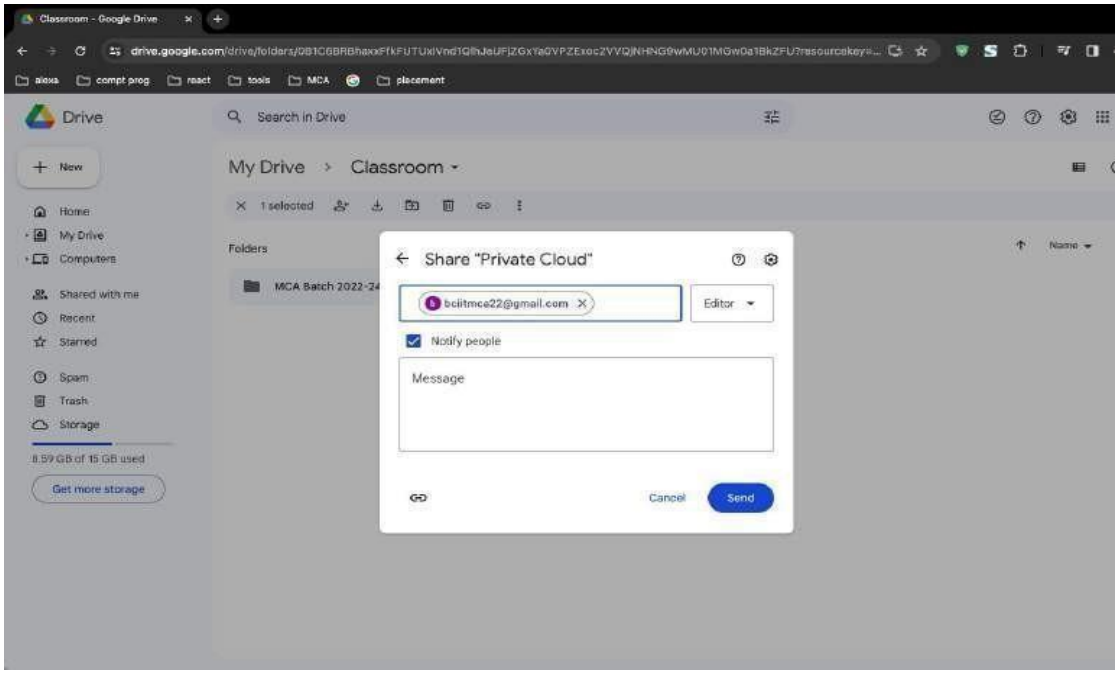




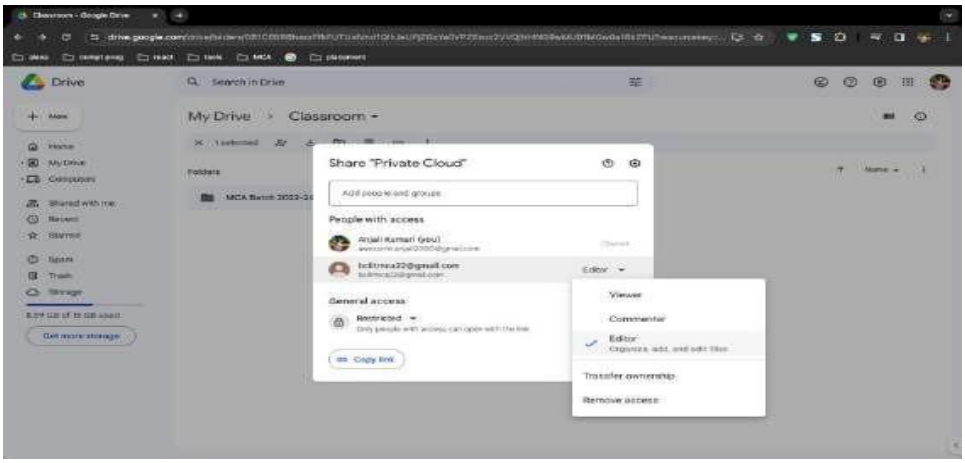
3. Share the Folder:
Right-click on the folder you just created.
Select "Share."



4. Add Users:
In the sharing dialog, enter the email addresses of the users you want to grant access to.
Choose the appropriate access level (e.g., Viewer, Commenter, Editor) based on the level of access you want to provide.

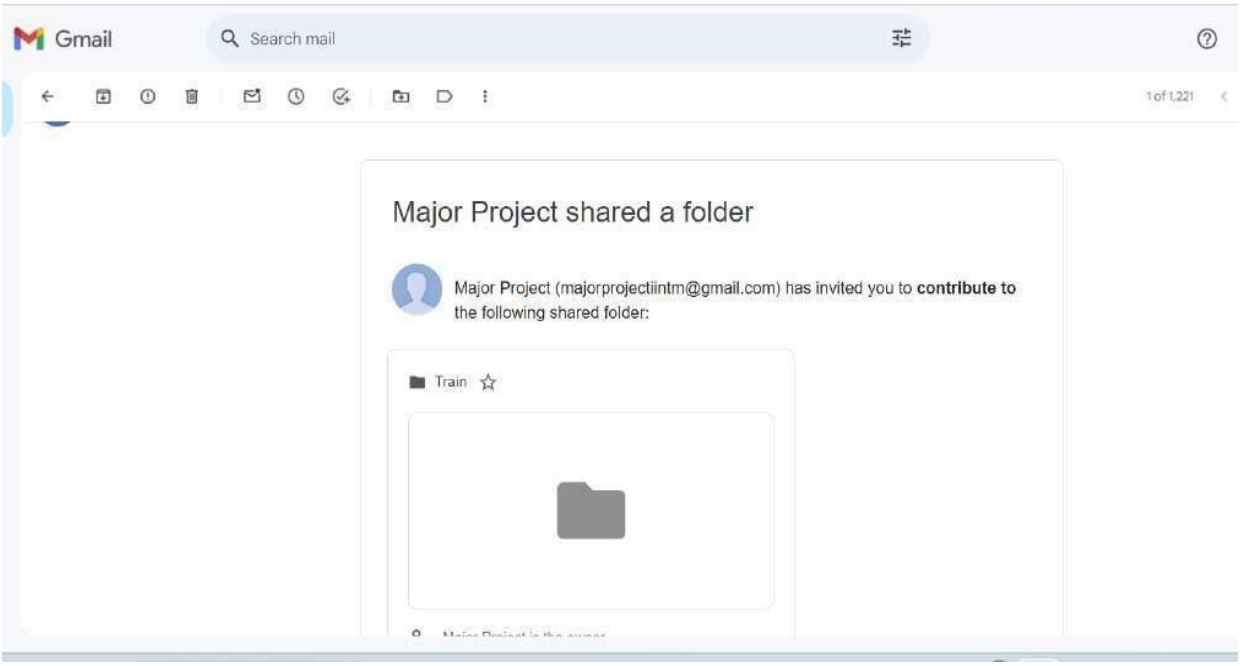


5. Configure Advanced Settings (Optional):
Click on "Advanced" in the sharing dialog.
Adjust settings like link sharing, preventing editors from changing access, etc.



6. Send Invitations:
Click on "Send" to send invitations to the specified email addresses.
Users will receive an email notification and can access the shared folder through their Google Drive.

7. User Access Management:
As the owner, you can manage access at any time by right-clicking on the folder, selecting "Share," and modifying permissions.

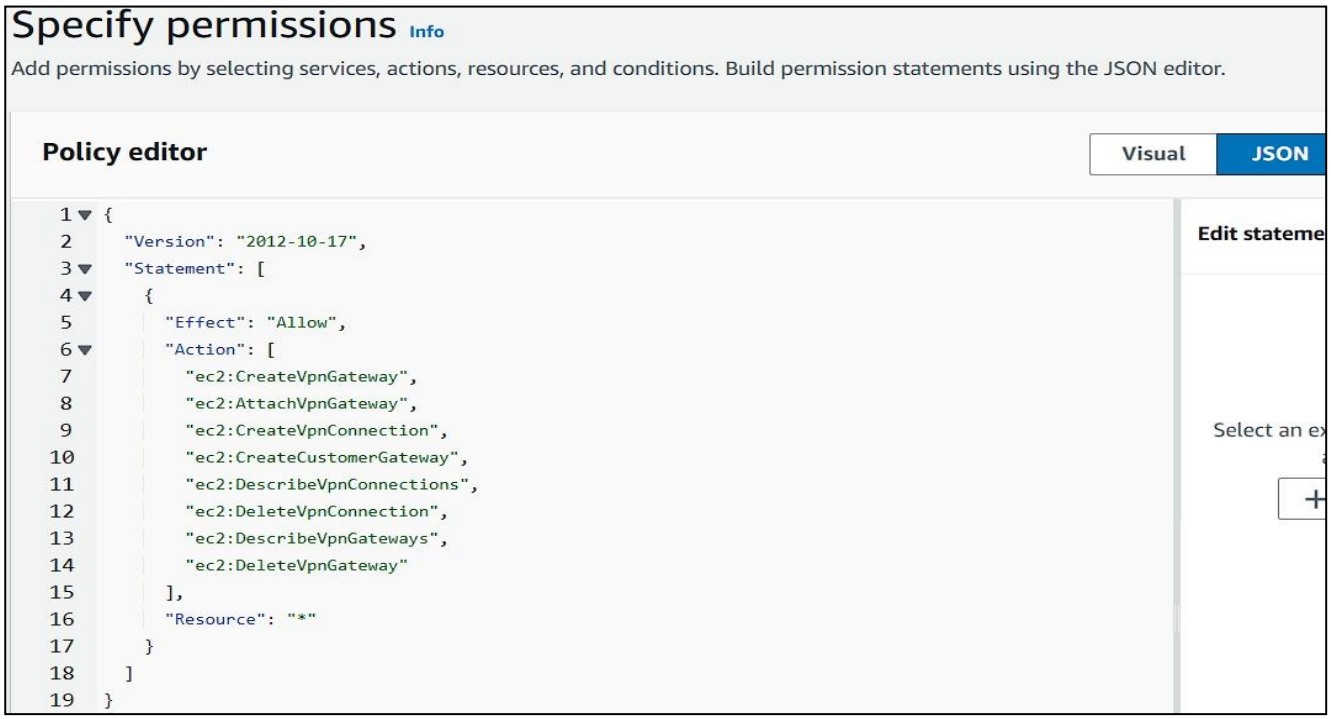


PRACTICAL 22

Setup VPN connection in IAM.

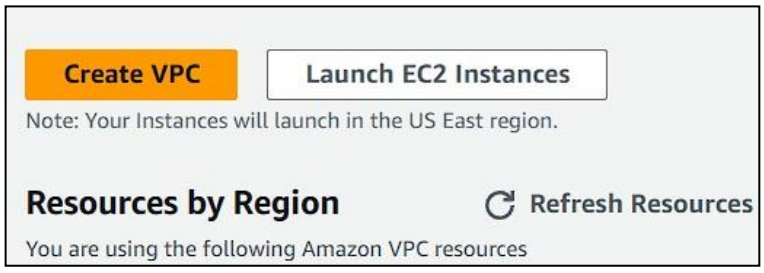
Step 1: Set Up an IAM Role or Policy for VPN Management
To allow users to manage VPN connections, you need to configure IAM permissions.
Go to IAM Console:
Open the IAM Console in the AWS Management Console.
Create a New Policy:
Navigate to Policies > Create Policy.

In the JSON editor, add permissions for VPC and VPN management:

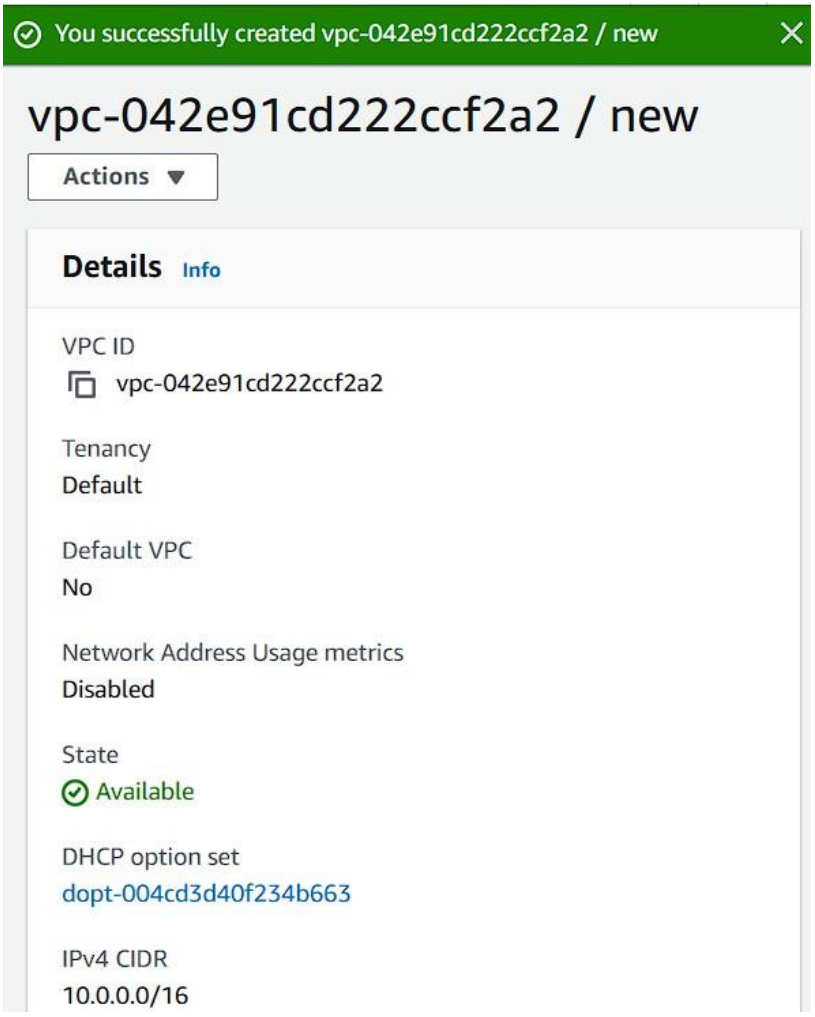


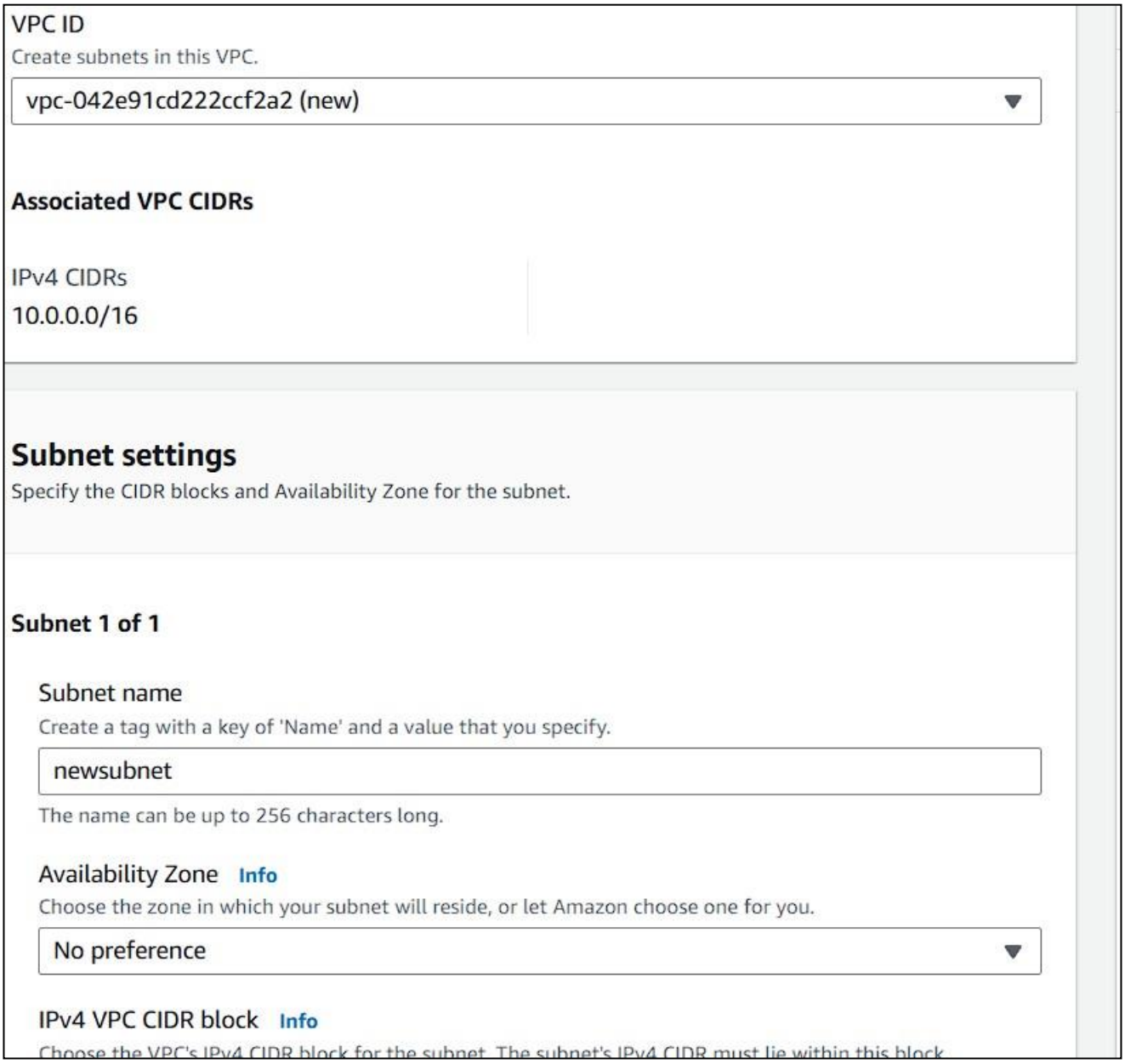
Click Next > Next > Create Policy.
Attach the Policy to a User or Role:
Go to Roles or Users, select the target entity, and attach the new policy to grant VPN setup permissions.

Step 2: Set Up the VPC and Subnet
Go to VPC Console:
Open the VPC Console in the AWS Management Console.
Create a VPC:
Go to Your VPCs > Create VPC.

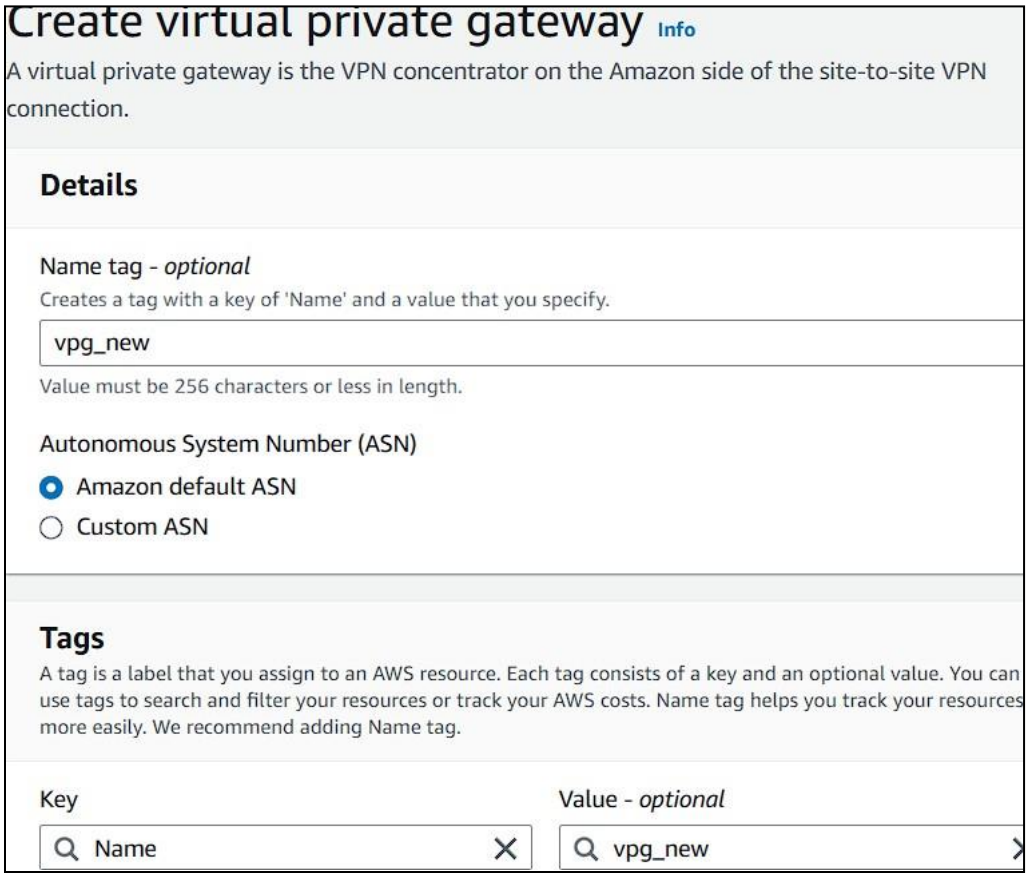


Enter a Name, IPv4 CIDR block (e.g., 10.0.0.0/16), and IPv6 CIDR block if needed.
Select Tenancy as Default and click Create VPC.
Create Subnets:
Go to Subnets > Create subnet.
Select the VPC you just created and specify an IPv4 CIDR block (e.g., 10.0.1.0/24).
Click Create.

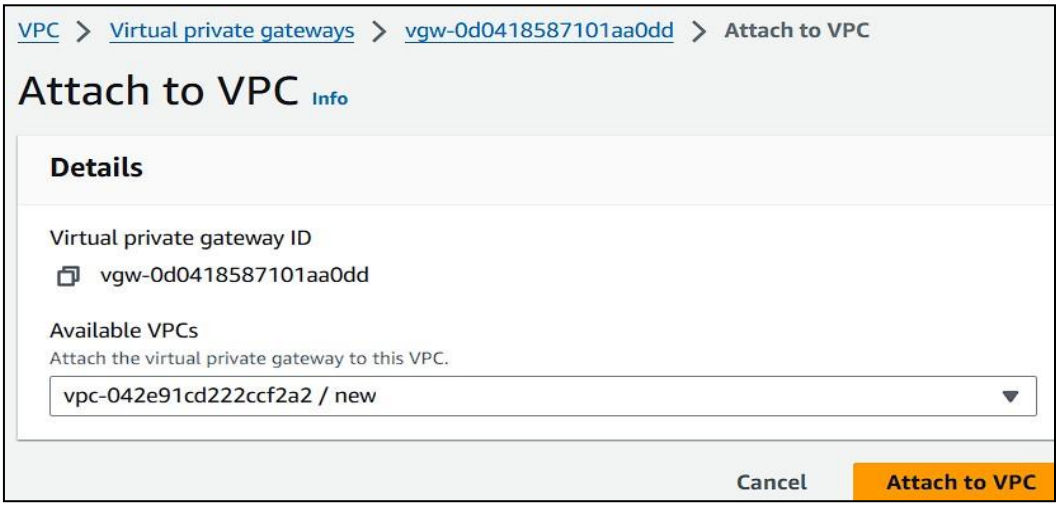



VPC ID
Create subnets in this VPC.
vpc-042e91cd222ccf2a2 (new)
Associated VPC CIDRs
IPv4 CIDRs
10.0.0.0/16
Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.
Subnet 1 of 1
Subnet name
Create a tag with a key of 'Name' and a value that you specify.
newsubnet
The name can be up to 256 characters long.
Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
No preference
IPv4 VPC CIDR block Info

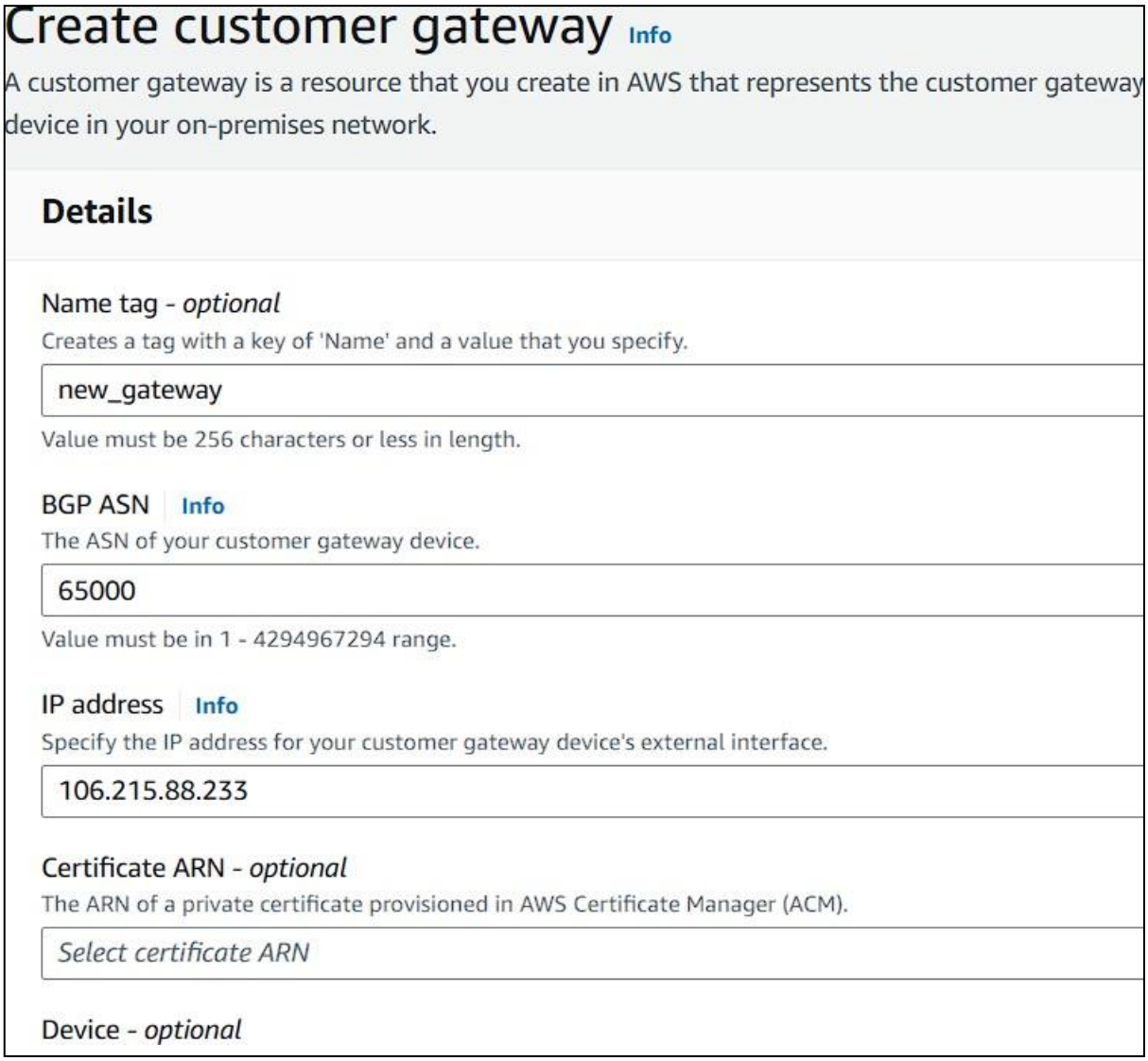
Step 3: Create a Virtual Private Gateway
Go to Virtual Private Gateways:
In the VPC Console, go to Virtual Private Gateways > Create Virtual Private Gateway.
Name your gateway, select Amazon side ASN (or leave default), and click Create.



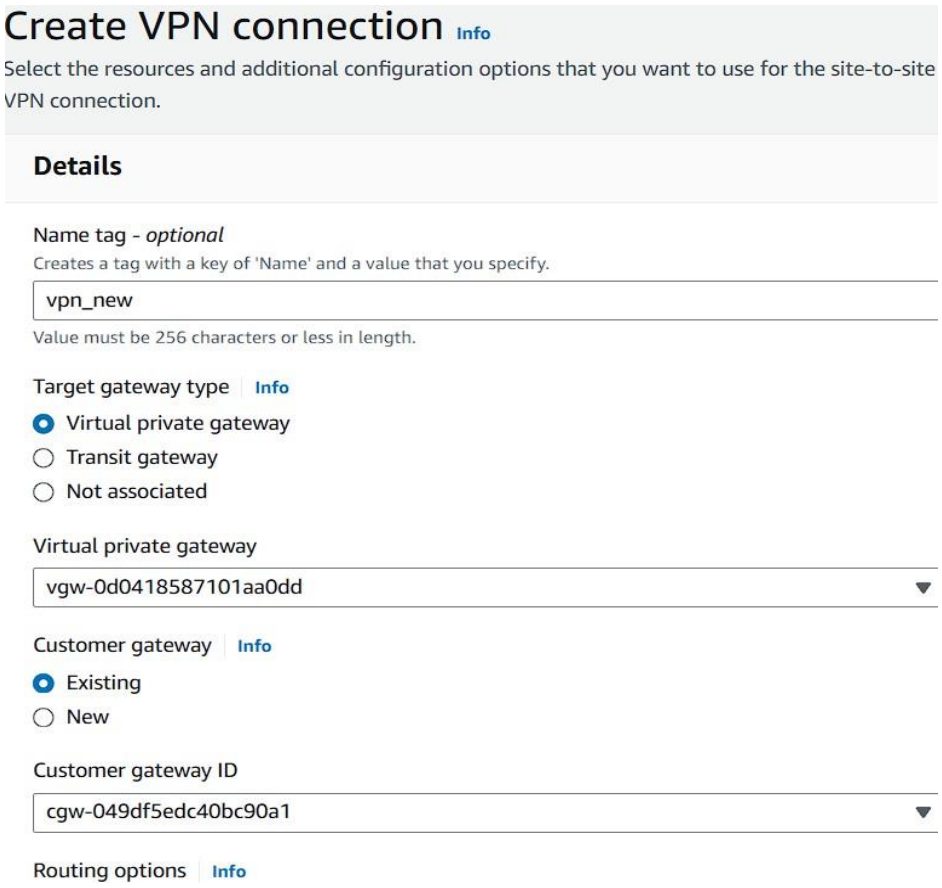
Attach the Virtual Private Gateway to the VPC:
Select the Virtual Private Gateway you created, click Actions > Attach to VPC.
Choose the VPC you created in Step 2 and click Attach.



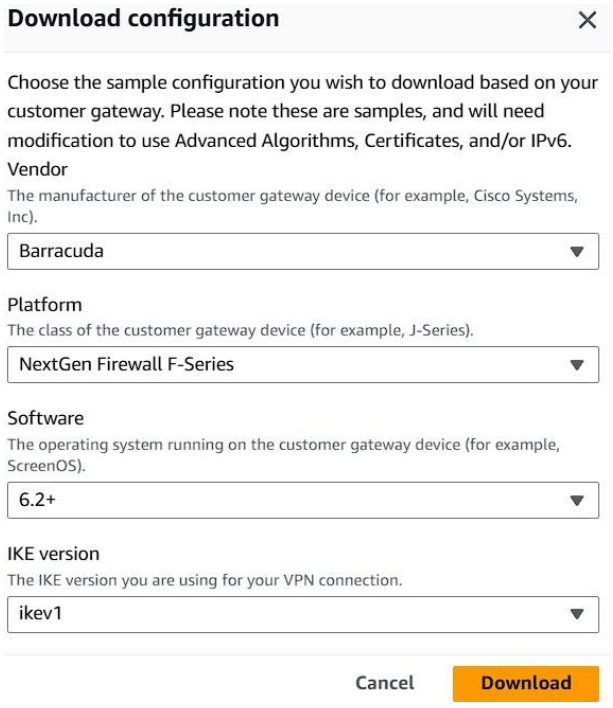
Step 4: Set Up Customer Gateway (On-premises Configuration)
Go to Customer Gateways:
In the VPC Console, go to Customer Gateways > Create Customer Gateway.



Step 5: Create the VPN Connection
Go to VPN Connections:
In the VPC Console, go to VPN Connections > Create VPN Connection.
Specify VPN Details:
Name tag: Add a name for the VPN connection.
Target Gateway Type: Select Virtual Private Gateway and choose the one you created earlier.
Customer Gateway: Select the customer gateway you created.
Routing Options: Select Static or Dynamic.
Static Routes: If Static, enter the on-premises CIDR.
Click Create VPN Connection.

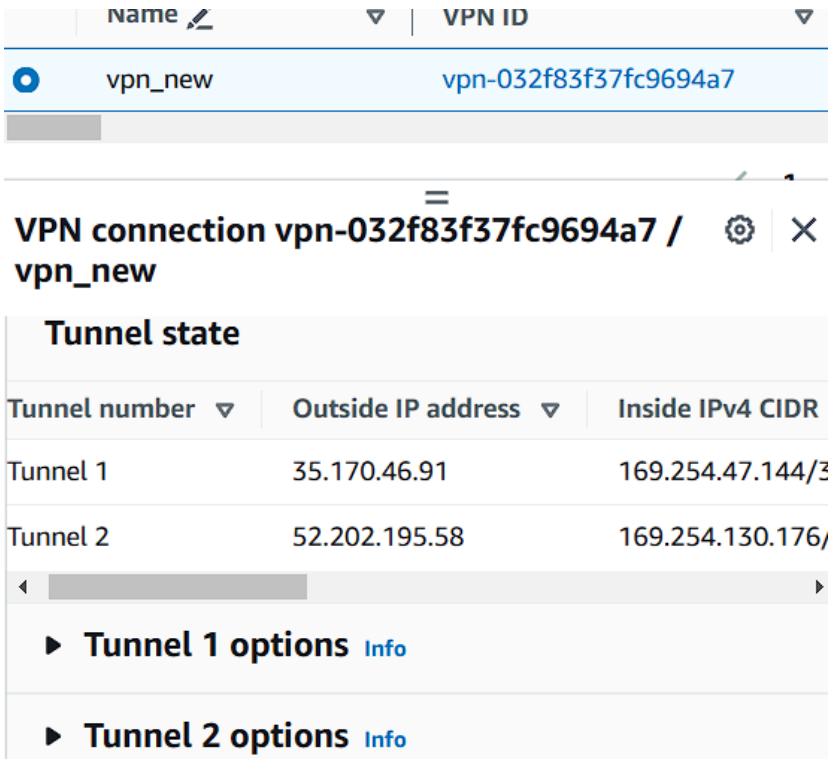


Download Configuration:
After creating the VPN connection, select it, go to Actions > Download Configuration.
Choose your on-premises vendor and model to download the configuration file for your device.



Step 6: Configure Security and Route Tables
Update Route Tables:
Go to Route Tables in the VPC Console.
Select the route table associated with your subnet, and click Edit routes.
Add a route that directs traffic to your on-premises CIDR through the Virtual Private Gateway.
Configure Security Groups:
Go to Security Groups and edit the rules to allow traffic as needed for the VPN.
Update Network ACLs (Optional):
If using Network ACLs, allow the appropriate inbound and outbound rules for VPN traffic.

Step 7: Test and Verify the VPN Connection
Check VPN Status:
In VPN Connections, check the Tunnel Status for the VPN connection. It should display UP if configured correctly.




```
64 bytes from bom07s37-in-f14.1e100.net (142.250.199.174): icmp_seq=6
64 bytes from bom07s37-in-f14.1e100.net (142.250.199.174): icmp_seq=7
64 bytes from bom07s37-in-f14.1e100.net (142.250.199.174): icmp_seq=8
64 bytes from bom07s37-in-f14.1e100.net (142.250.199.174): icmp_seq=9
64 bytes from bom07s37-in-f14.1e100.net (142.250.199.174): icmp_seq=10
64 bytes from bom07s37-in-f14.1e100.net (142.250.199.174): icmp_seq=11
```