

S.No	Practical	Page No.	Sign
1	Create an AWS account, azure account and google cloud account	1-2	
2	Set up a budget to an AWS account	3-5	
3	Launch a windows server instance with t2.micro instance type and create a security group by using EC2	6-8	
4	Connect the launch instances 2/2 status check and decrypt password by using RDP client	9-10	
5	Launch a LINUX UBUNTU SERVER INSTANCE with t2.micro instance type and create a security group BY USING EC2 ..	11-12	
6	Connect the launch instance ,2/2 status check and decrypt password BY USING EC2 INSTANCE CONNECT	13-14	
7	Connect the launch instance ,2/2 status check and decrypt password BY USING PUTTY AND SSH CLIENT	15-16	
8	Terminate the launch instance and connect again by using RDP client .	17-18	
9	Delete the launch instance	19-20	
10	Host a static website on Windows server with IIS Manager on EC2 Launch Instance.	21-25	
11	Host a BCIIT SAMPLE website on Windows server with IIS Manager on EC2 Launch Instance	26-28	
12	Create IAM user and grant in limited permission to IAM user by AWS route user	29-30	
13	Create a bucket by using S3 aws service	31-32	
14	Upload an object on bucket created by using S3 AWS service	33-35	
15	Create a bucket and allow public access on uploaded objects by using object URL and S3 aws surface	36-38	
16	Delete the object and bucket by using S 3 interface	39-40	
17	Transfer the object file from S3 service to EC2 launched Linux server install GCC and wget commands in this regard on terminal	41-43	
18	Create a VPC and implement EC2 services on it	44-45	
19	Implement and configure load balancing with all necessary steps	46-48	
20	How to handle a cloud shell explain it	49	
21	Create a private cloud on Google Drive and Grant restrict permissions for the user	50-52	
22	Setup VPN connection in IAM.	53-59	

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.

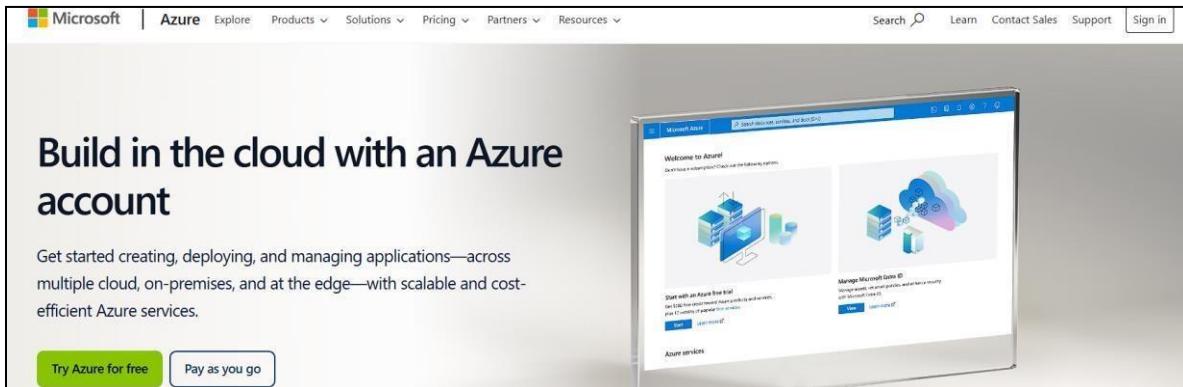
Step 8:

Registration Confirmation page.

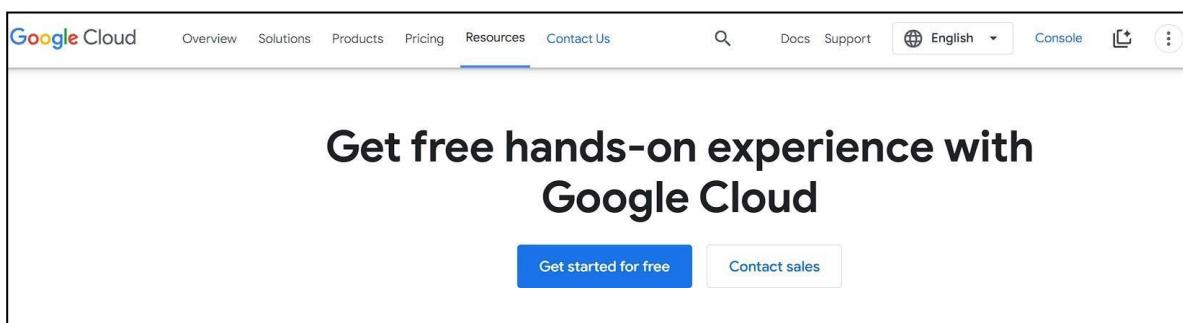
Once you complete all the above steps and processes. You'll get the confirmation page below. Now your account will be processed for activation. It may take somewhere between 30 minutes to 1 hour for you to receive an email confirmation that your Amazon Cloud Services account has been activated



- **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:

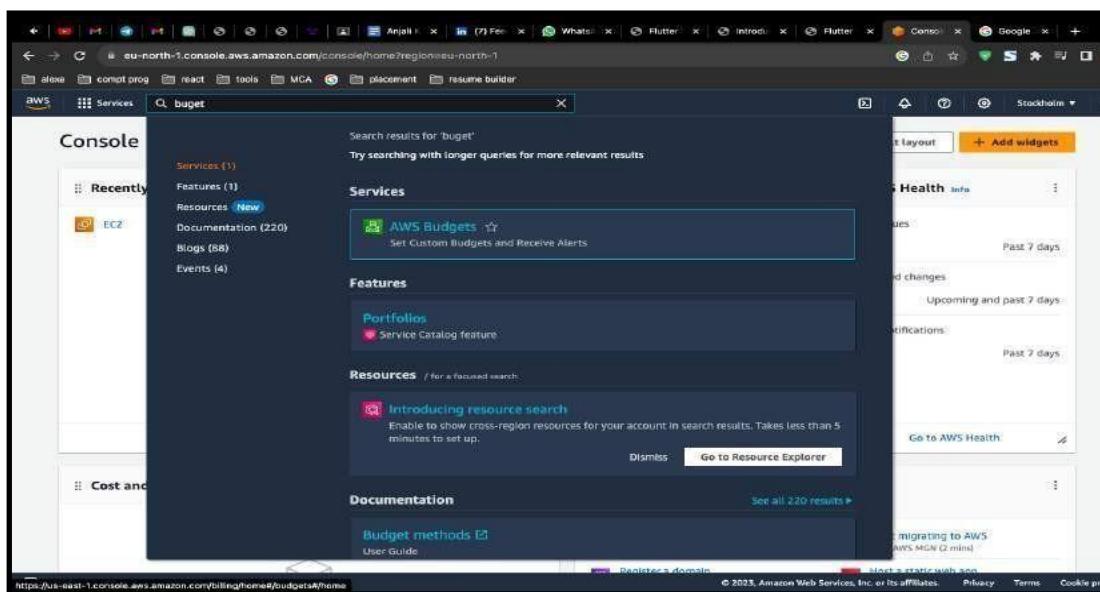
- **Cost Monitoring:** AWS Budgets provides insights into your AWS cost and usage data, allowing you to monitor your spending in real-time. This helps you keep track of your expenses and ensure they align with your budgetary goals.
- **Budget Setting:** The service allows you to set specific budgets 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.



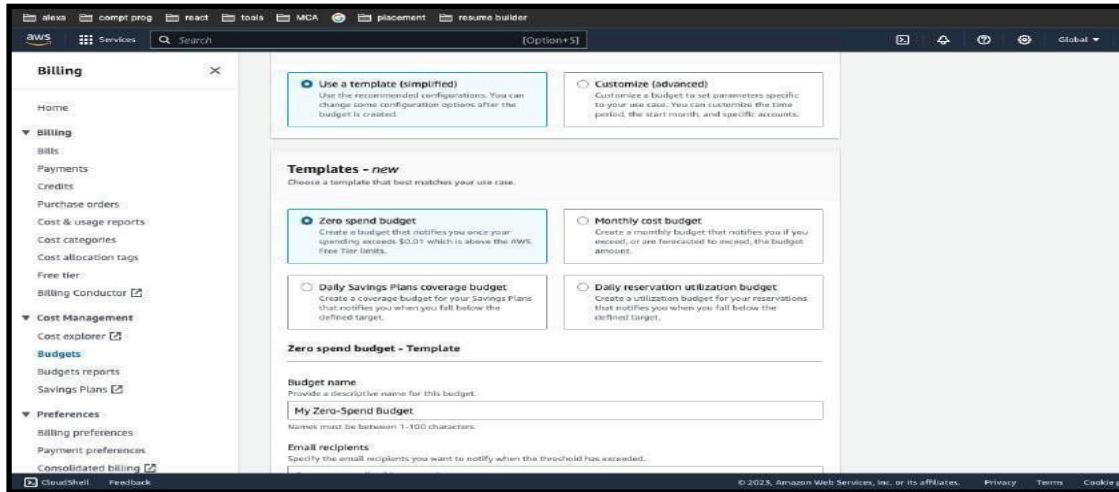
Step 3:

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.

Billing

Zero spend budget - Template

Budget name: My Zero-Spend Budget

Email recipients:

Scope: All AWS services are in scope in this budget.

You will be notified via email when any spend above \$0.01 is incurred.

Create budget

Step 6:

Your Aws Budget is Created

Your budget My Zero-Spend Budget has been created successfully. After creating a budget, it can take up to 24 hours to populate all of your spend data.

Billing

Overview

Budgets (1)

Name	Thresholds	Budget	Amount used	Forecasted	Current vs. budgeted
My Zero-Spend Budget	OK	\$1.00	\$0.00	-	0.0

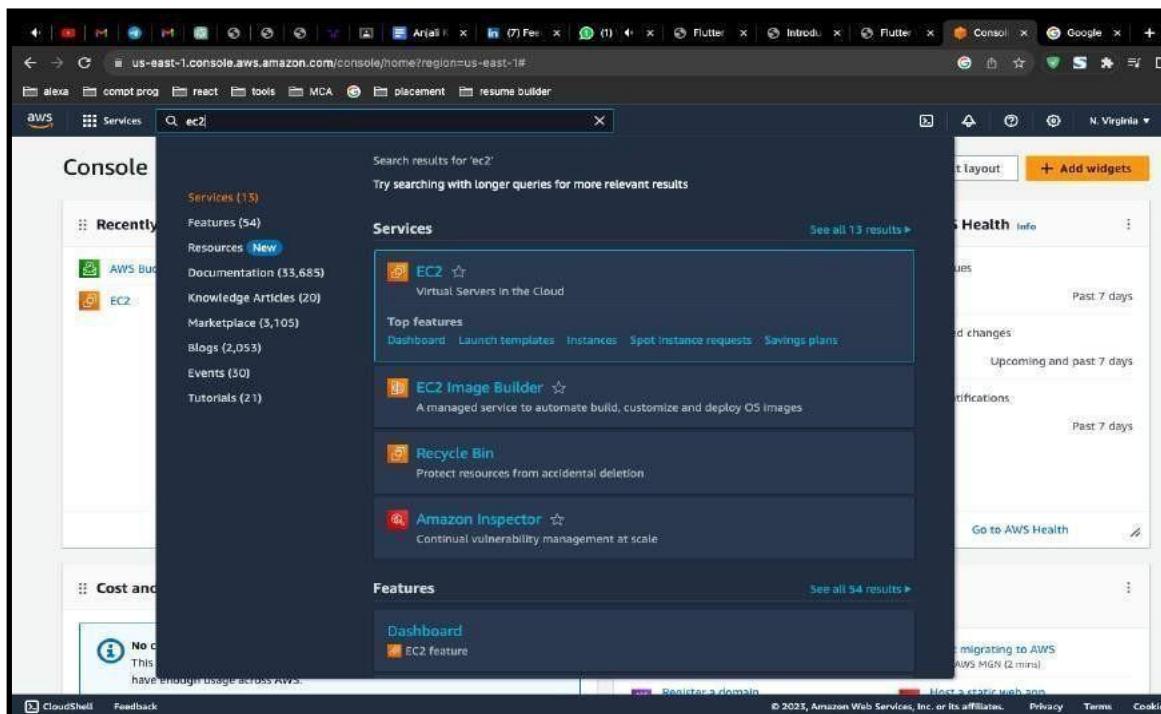
PRACTICAL 3

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:

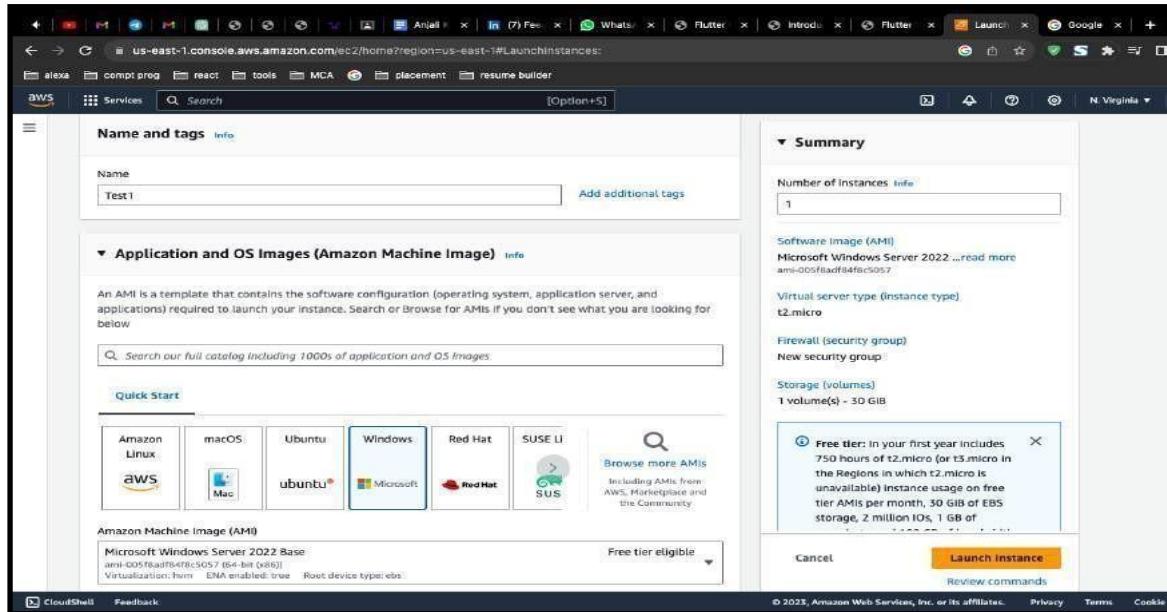
- Application Hosting: You can host Windows-based applications, including web servers, database servers, content management systems, and custom applications, on Windows Server instances in EC2.
- Development and Testing: Windows Server instances are ideal for development and testing environments. You 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.



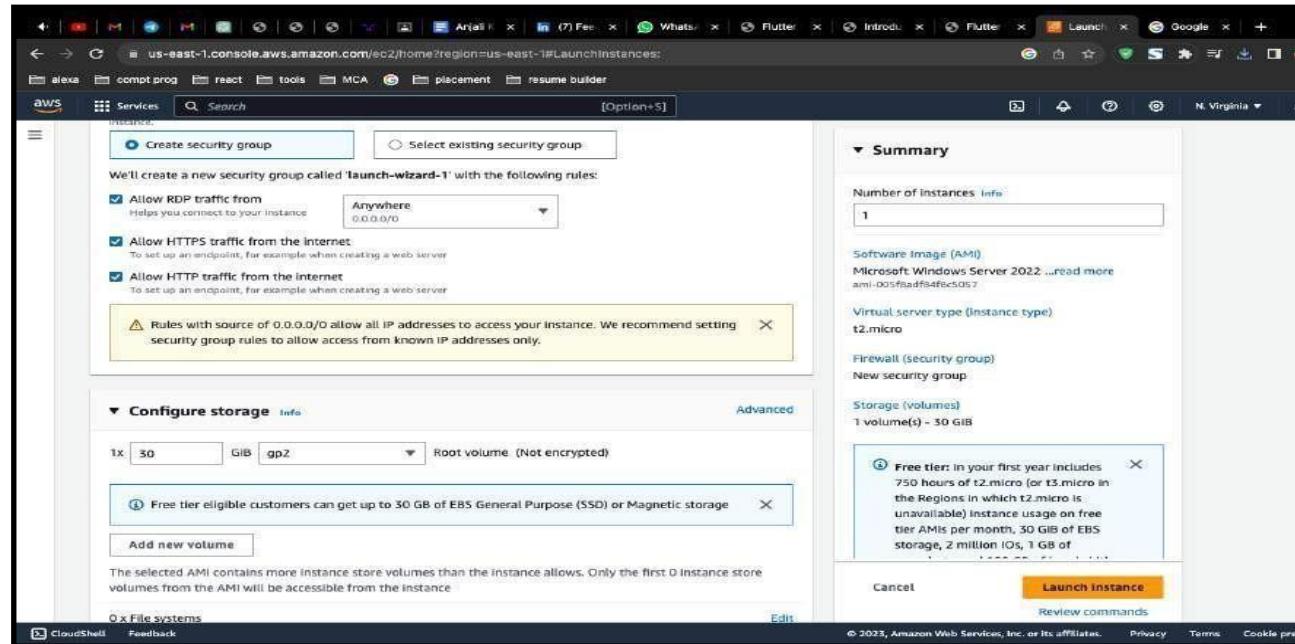
Step 2: Enter the name of your Aws Ec2 Instance

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.

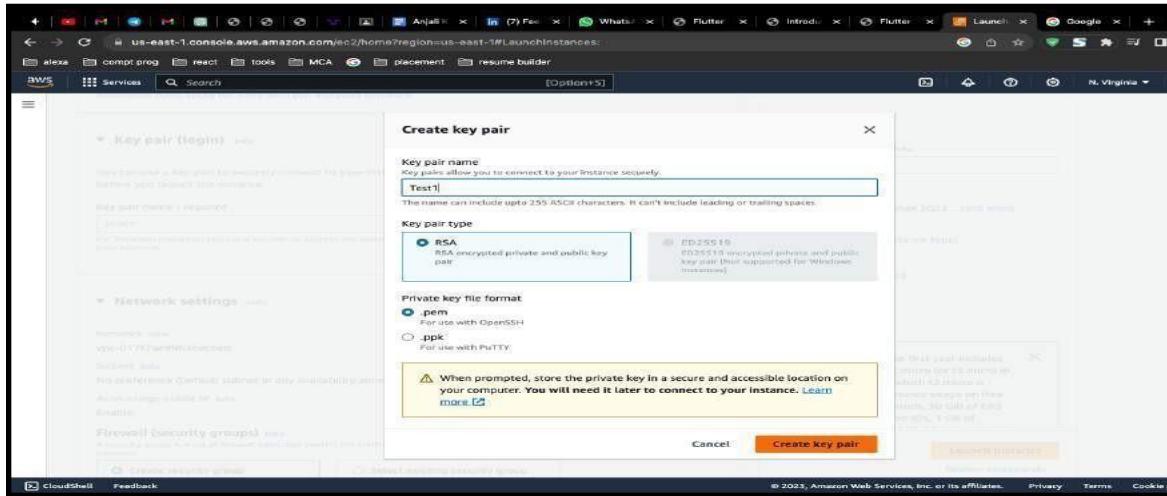


Step 4: In the "Add Storage" step, you can specify the size and type of root volume for your instance. You can also add additional volumes if necessary.

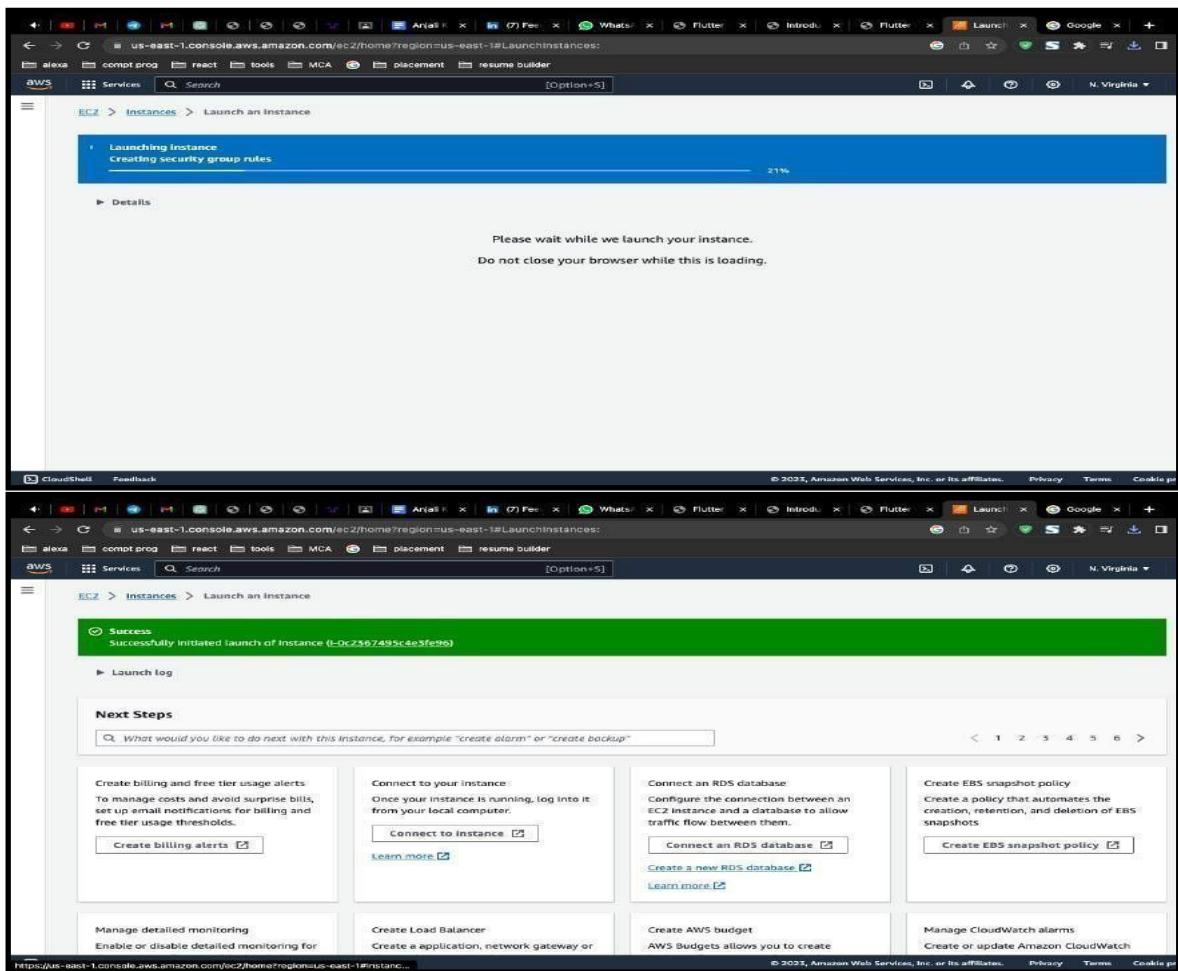
Step 5: In the "Configure Security Group" step, you'll need to configure the security group rules. Security groups act as firewalls to control inbound and outbound traffic to your instance. Ensure that you allow Remote Desktop Protocol (RDP) for Windows instances if you plan to access them remotely.



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.

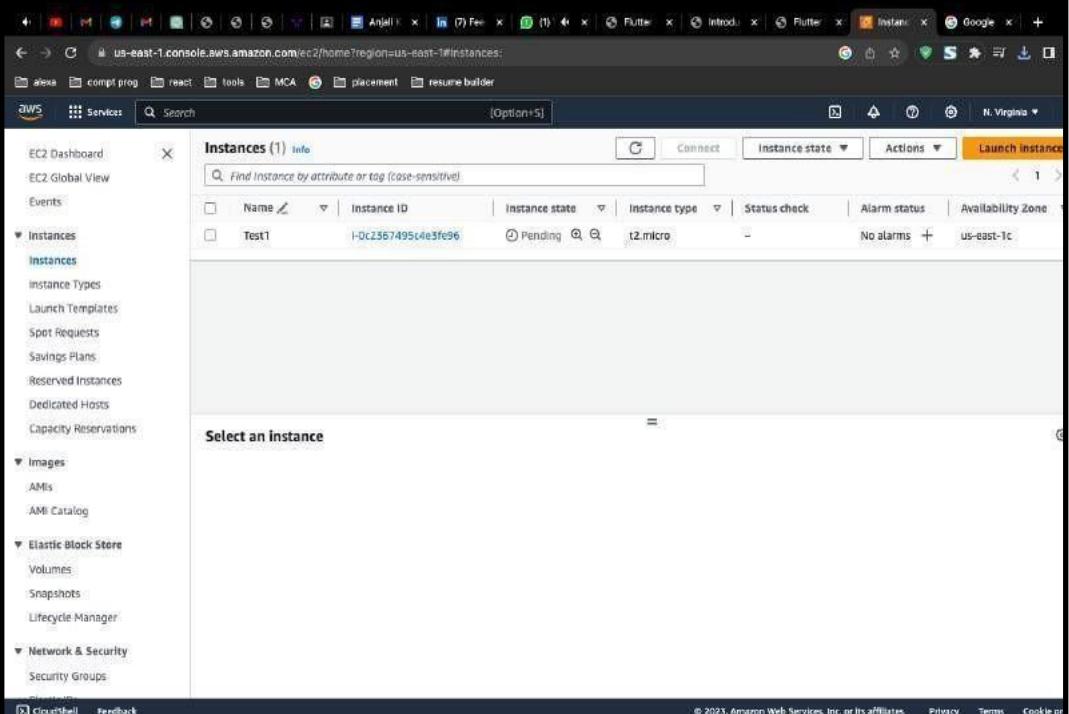


PRACTICAL 4

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

Step 1:

Select the Instance



The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various EC2-related options like Dashboard, Global View, Events, Instances, Images, Elastic Block Store, and Network & Security. The main area displays a table titled 'Instances (1) Info'. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. One row is visible for an instance named 'Test1' with the ID 'i-0c2367495c4e3fe96', which is currently 'Pending'. The 'Instance state' dropdown is set to 'Pending'. The 'Status check' column shows a green icon with a question mark. The 'Alarm status' column says 'No alarms'. The 'Availability Zone' is listed as 'us-east-1c'. At the top right of the table, there are buttons for 'Connect', 'Actions', and 'Launch instance'. Below the table, a large button says 'Select an 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.

EC2 > Instances > i-0ed023039c00b4423 > Connect to instance

Connect to instance Info

Connect to your instance i-0ed023039c00b4423 (experiment) using any of these options

Session Manager | **RDP client** | **EC2 serial console**

Instance ID
 i-0ed023039c00b4423 (experiment)

Connection Type

Connect using RDP client
 Download a file to use with your RDP client and retrieve your password.

Connect using Fleet Manager
 To connect to the instance using Fleet Manager Remote Desktop, the SSM Agent must be installed and running on the instance. For more information, see [Working with SSM Agent](#)

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

[!\[\]\(adaaaa51ac0f09d90a6641dac9fbfa7e_img.jpg\) Download remote desktop file](#)

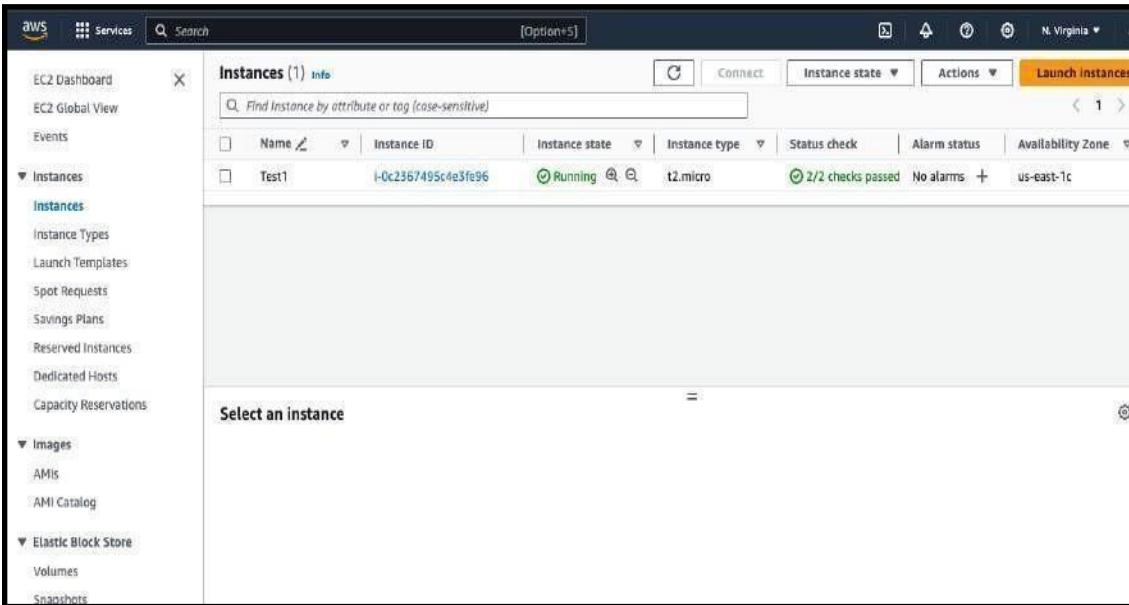
When prompted, connect to your instance using the following details:

Public DNS <input checked="" type="checkbox"/> ec2-54-197-166-238.compute-1.amazonaws.com	User name <input checked="" type="checkbox"/> Administrator
Password Get password	

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

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.



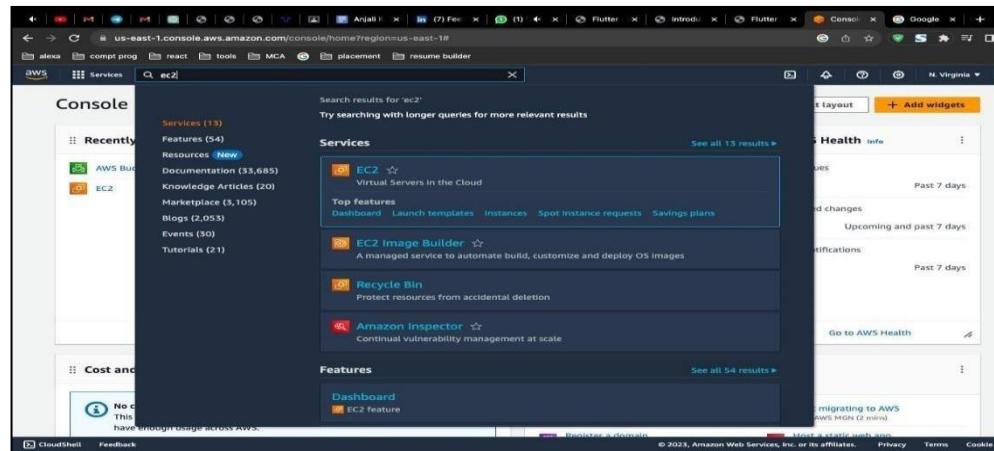
The screenshot shows the AWS EC2 Instances page. The left sidebar includes links for EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots). The main content area displays the "Instances (1) Info" section with a table. The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. One row is shown for an instance named "Test1" with the following details: Instance ID: i-0c2367495c4a3fe96, Instance state: Running, Instance type: t2.micro, Status check: 2/2 checks passed, Alarm status: No alarms, and Availability Zone: us-east-1c. Below the table, a message says "Select an instance".

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.

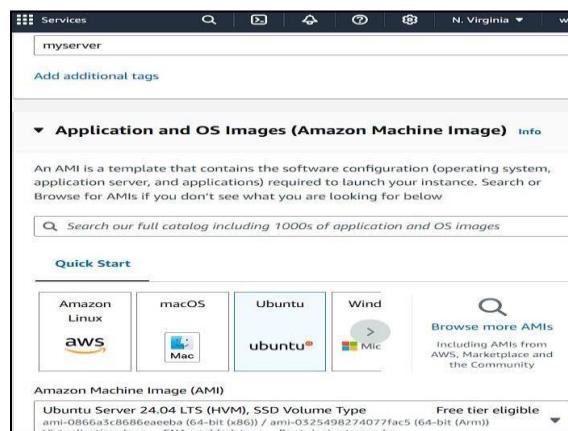


Step 3:

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".

Instance type [Info](#) | [Get advice](#)

t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory
Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

Additional costs apply for AMIs with pre-installed software

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.

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

We'll create a new security group called '**launch-wizard-1**' with the following rules:

- Allow SSH traffic from **Anywhere** 0.0.0.0/0
Helps you connect to your instance
- Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server
- Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

Step 7: Review and Launch.

Select an existing key pair or create a key pair

ⓘ We noticed that you didn't select a key pair. If you want to be able to connect to your instance it is recommended that you create one or select an existing one.

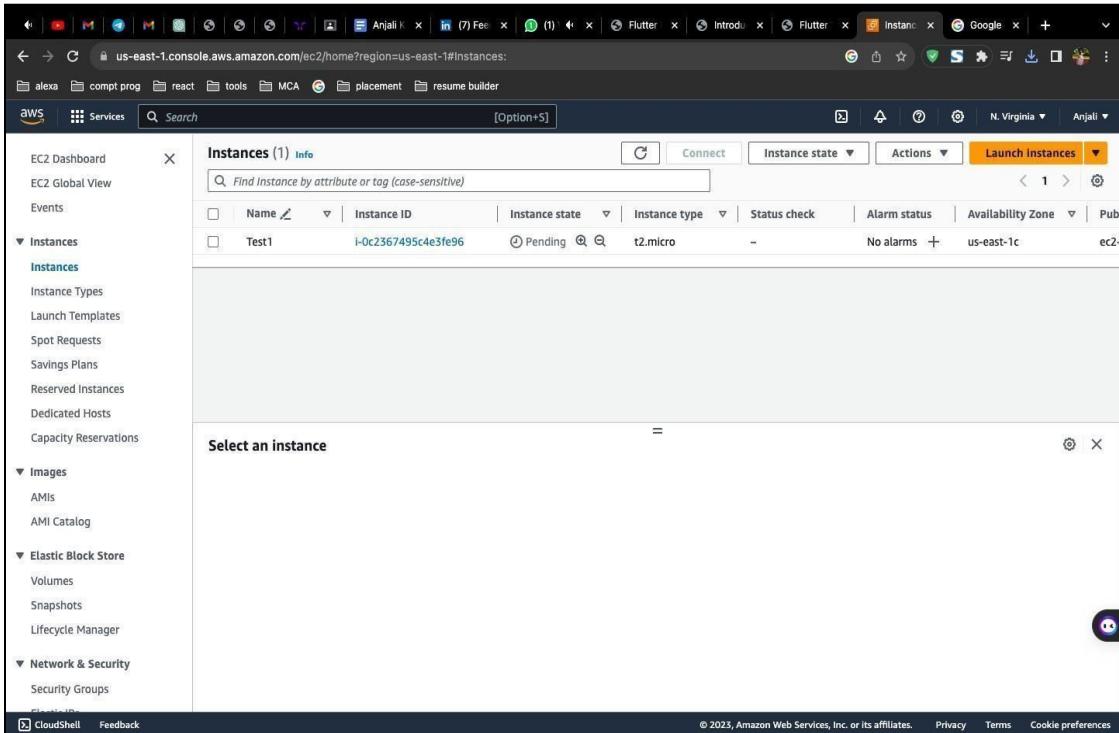
Create new key pair Proceed without key pair

[Cancel](#) [Launch instance](#)

PRACTICAL 6

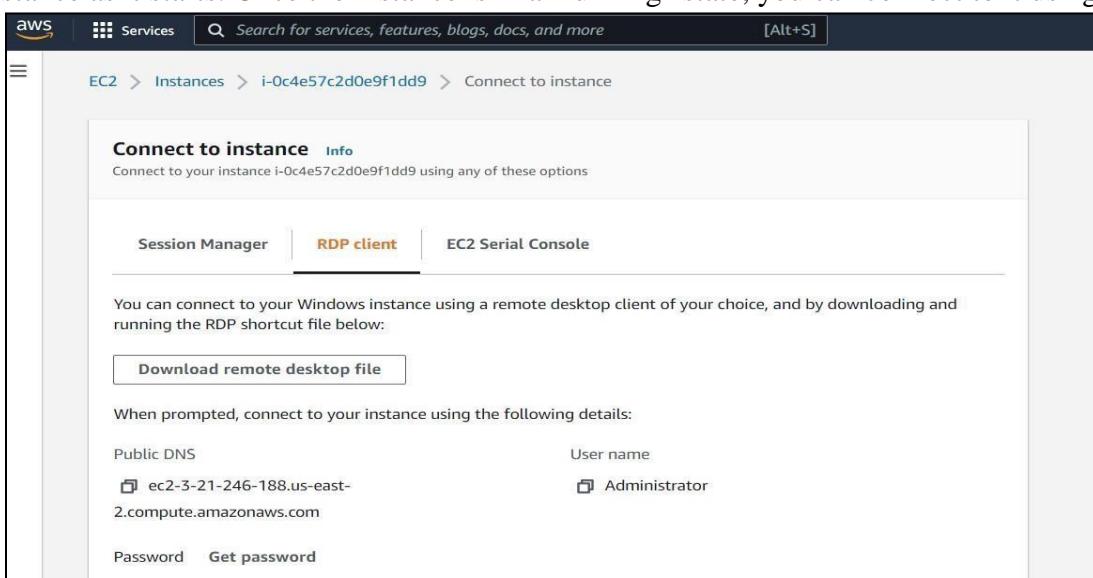
Connect the launch instance ,2/2 status check and decrypt password BY USING EC2 INSTANCE CONNECT.

Step 1: Select the Instance



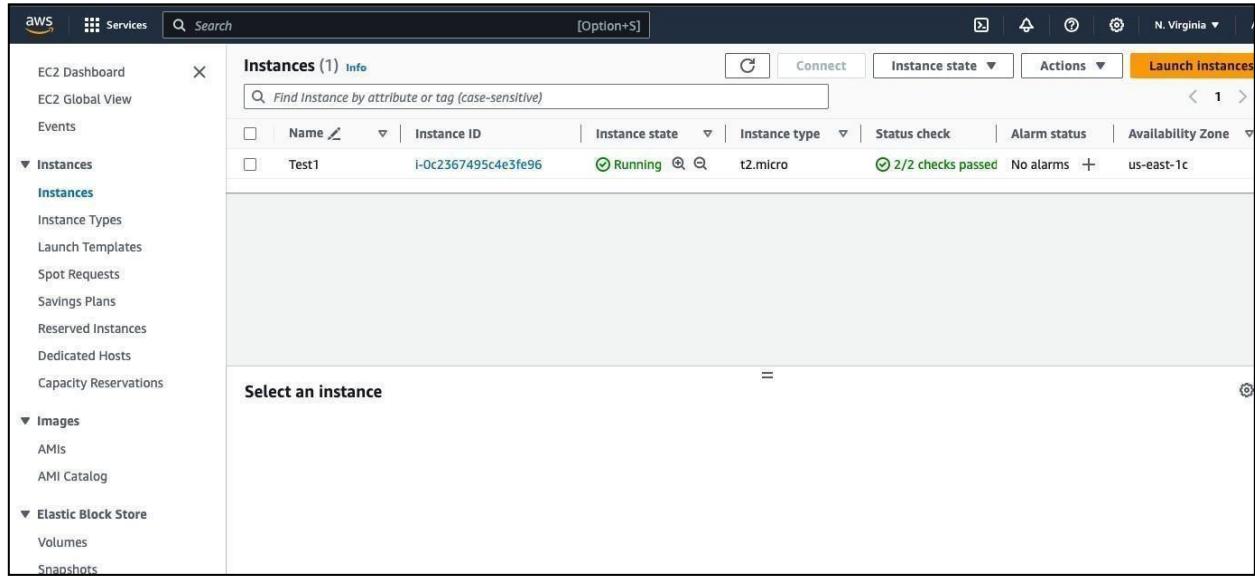
The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation options like EC2 Dashboard, Global View, Events, Instances (selected), Images, Elastic Block Store, and Network & Security. The main area displays a table titled 'Instances (1) Info'. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. One row is shown for an instance named 'Test1' with the ID 'i-0c2367495c4e3fe96', which is currently 'Pending'. The status check is '-' and it has no alarms. It's located in the 'us-east-1c' availability zone. At the bottom of the page, there are links for CloudShell and Feedback, and a footer with copyright information and links for Privacy, Terms, and Cookie preferences.

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.



The screenshot shows the 'Connect to instance' page for the 'Test1' instance. The top navigation bar includes the AWS logo, a search bar, and a link to 'Services'. Below the navigation, the breadcrumb trail shows 'EC2 > Instances > i-0c4e57c2d0e9f1dd9 > Connect to instance'. The main content area is titled 'Connect to instance' with an 'Info' link. It says 'Connect to your instance i-0c4e57c2d0e9f1dd9 using any of these options'. There are three tabs: 'Session Manager' (selected), 'RDP client' (highlighted in orange), and 'EC2 Serial Console'. A note below the tabs says 'You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:'. A button labeled 'Download remote desktop file' is present. Below this, instructions say 'When prompted, connect to your instance using the following details:'. It lists 'Public DNS' as 'ec2-3-21-246-188.us-east-2.compute.amazonaws.com', 'User name' as 'Administrator', and 'Password' with a 'Get password' link.

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.



The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main area displays a table titled "Instances (1) Info". The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. One row is present, showing "Test1" as the name, "i-0c2367495c4e3fe96" as the instance ID, "Running" as the instance state, "t2.micro" as the instance type, "2/2 checks passed" as the status check, "No alarms" as the alarm status, and "us-east-1c" as the availability zone. Below the table, a section titled "Select an instance" is visible.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Test1	i-0c2367495c4e3fe96	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c

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)

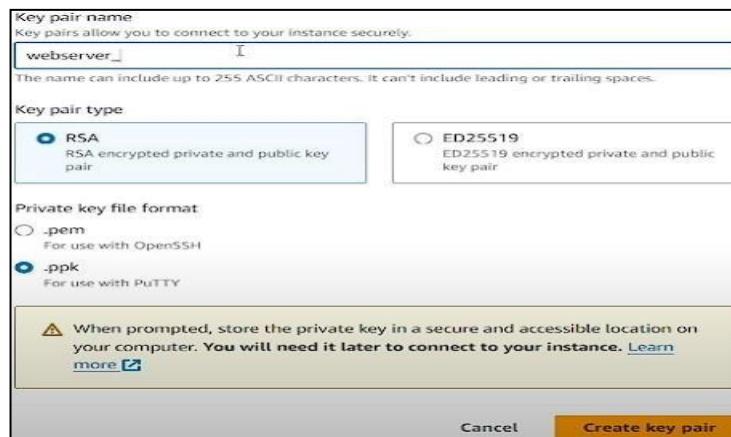
If you are using PuTTY, you need to convert the .pem file to a .ppk file format.

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>

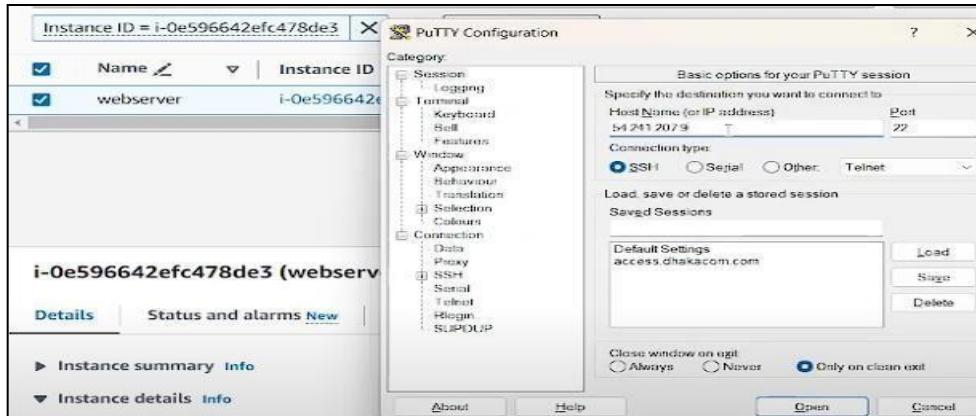
Debian: admin@<Public IP>

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 Mac OS, use the terminal with the SSH command. Make sure you have the .pem file ready.

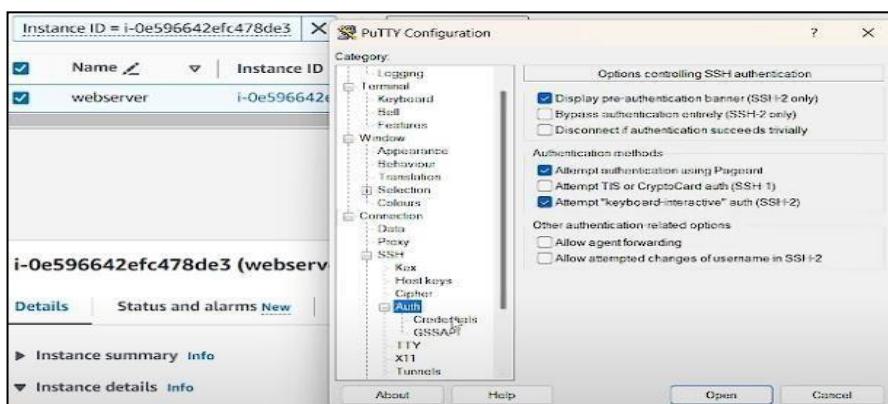
Open a terminal.

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.

A terminal window titled 'root@ip-172-31-5-174:~-' is shown. The user has run the command 'apt-get update'. The output shows the system is verifying and installing packages from the Amazon Linux 2 repository. The 'Installed' section lists several packages such as 'httpd-core-2.4.58-1.amzn2023.x86_64', 'httpd-filesystem-2.4.58-1.amzn2023.noarch', and 'httpd-tools-2.4.58-1.amzn2023.x86_64'. The process is completed successfully.

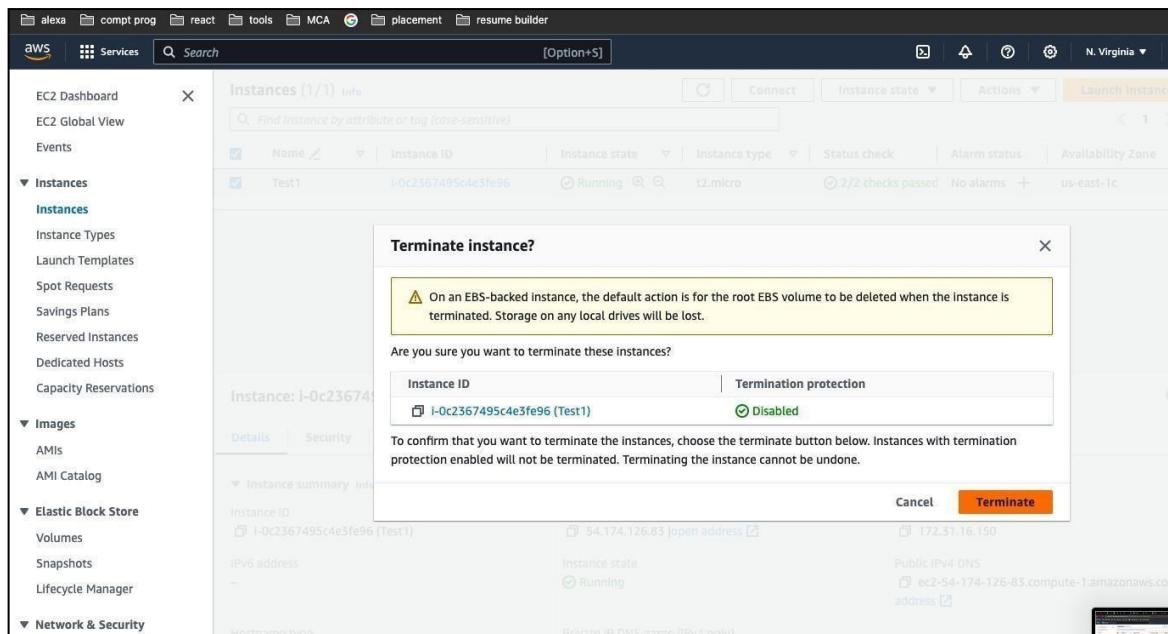
PRACTICAL 8

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

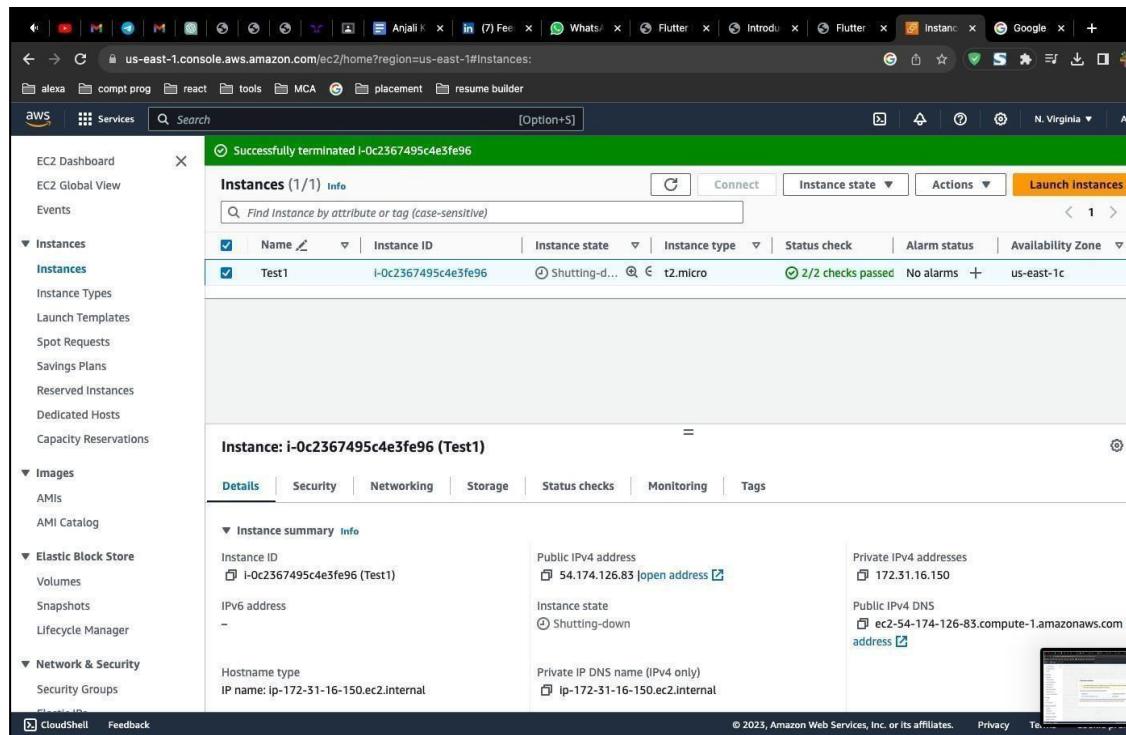
Step 1: In the EC2 dashboard, click on "Instances" in the left navigation pane to view a list of your running instances.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation pane with sections like EC2 Dashboard, EC2 Global View, Events, Instances (which is expanded), Images, Elastic Block Store, Network & Security, and CloudShell. The main area displays a table of instances. One instance is selected, labeled 'Test1' with the ID 'i-0c2367495c4e3fe96'. The instance state is 'Running'. To the right of the table is an 'Actions' dropdown menu with options: Stop instance, Start instance, Reboot instance, Hibernate instance, and Terminate instance. The 'Terminate instance' option is highlighted with a blue border. Below the table, there's a detailed view for the selected instance 'i-0c2367495c4e3fe96 (Test1)'. It shows details like Public IPv4 address (54.174.126.83), Instance state (Running), and Private IPv4 addresses (172.31.16.150). There's also a screenshot of a terminal window showing the Linux desktop environment.

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

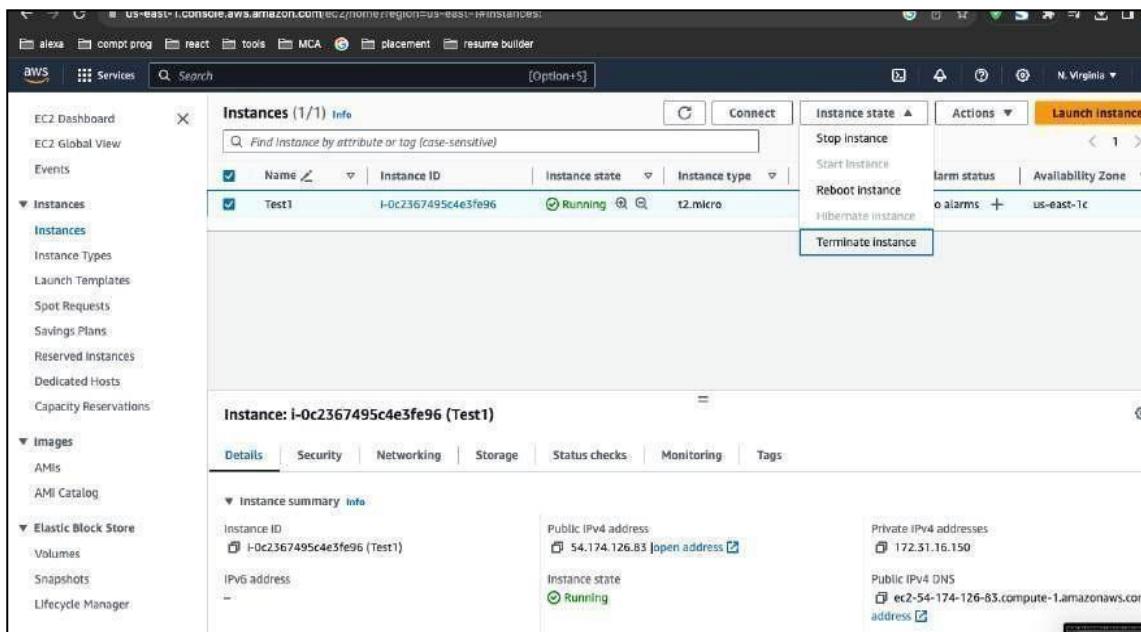
Delete the launched instance.

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:

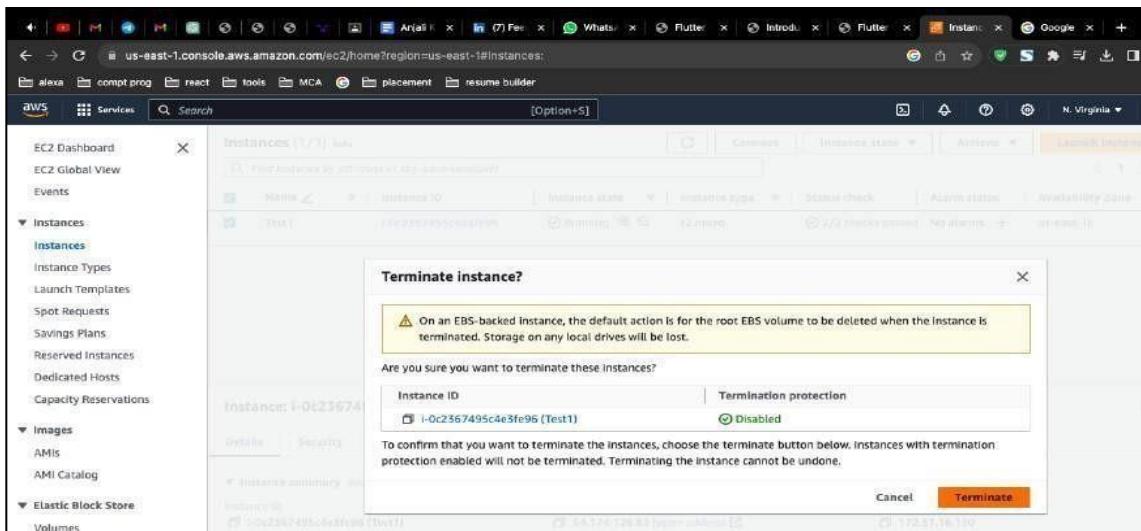
Click the checkbox next to the instance you want to terminate. It will become selected.



The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation pane with options like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots, Lifecycle Manager). The main area displays a table titled 'Instances (1/1) Info'. A single row is listed: 'Test1' (Instance ID: i-0c2367495c4e3fe96, State: Running, Type: t2.micro). To the right of the table are buttons for Stop instance, Start instance, Reboot instance, Hibernate instance, and Launch instances. Below the table, a modal window is open for 'Instance: i-0c2367495c4e3fe96 (Test1)'. It shows details like Public IPv4 address (54.174.126.83), Private IPv4 addresses (172.31.16.150), and Public IPv4 DNS (ec2-54-174-126-83.compute-1.amazonaws.com). The 'Details' tab is selected. At the bottom of the modal, there's a 'Terminate' button.

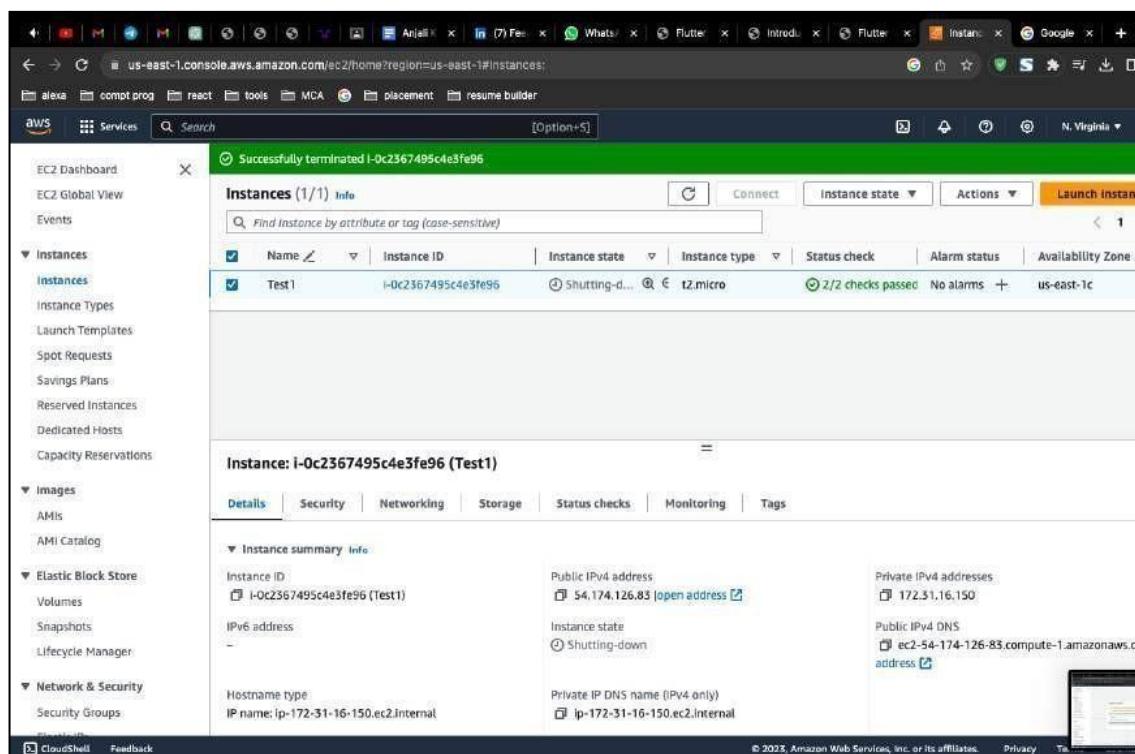
Step 3 :

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 :

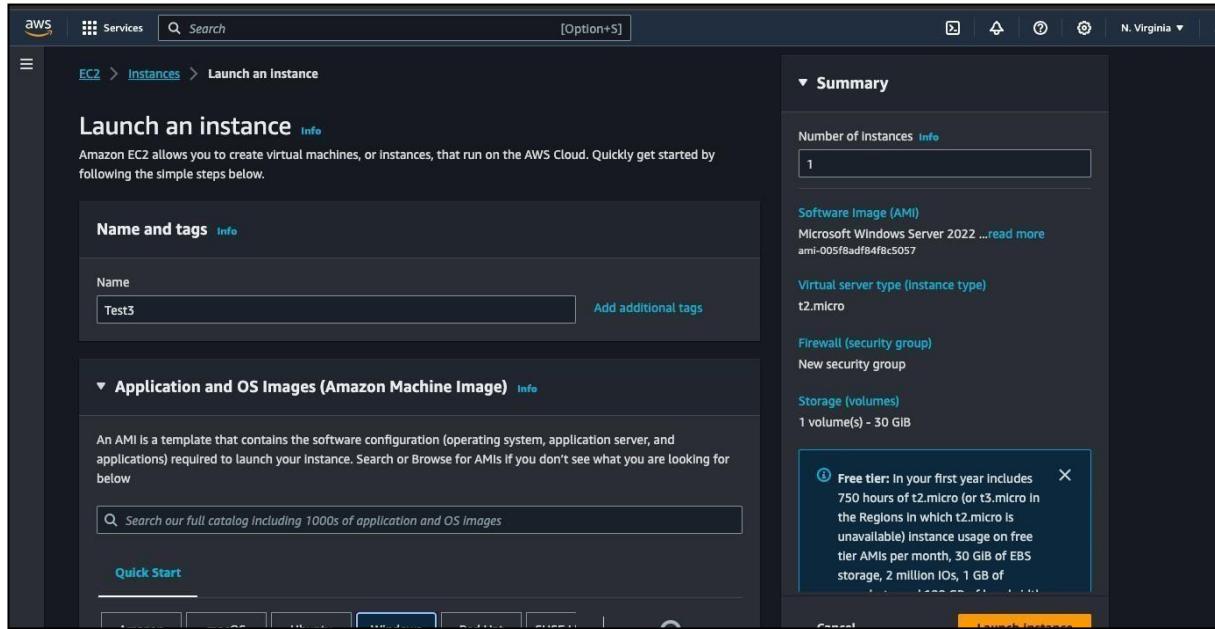
AWS will now initiate the termination process. The instance will be stopped if it was running, and then it will be permanently deleted. This process may take a few minutes.



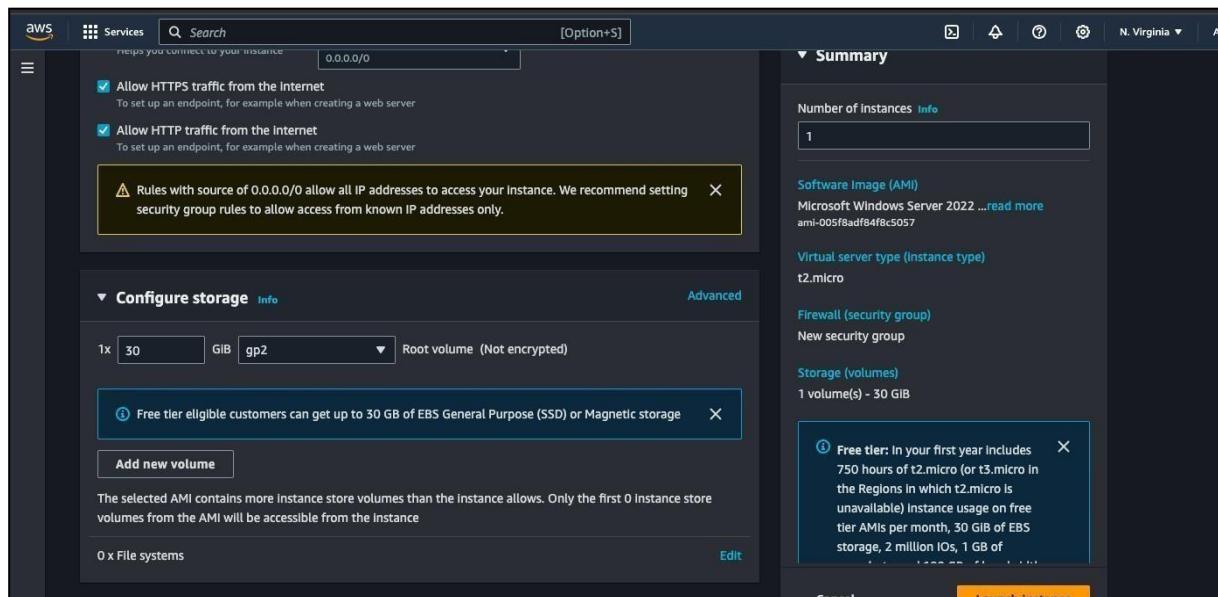
PRACTICAL 10

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

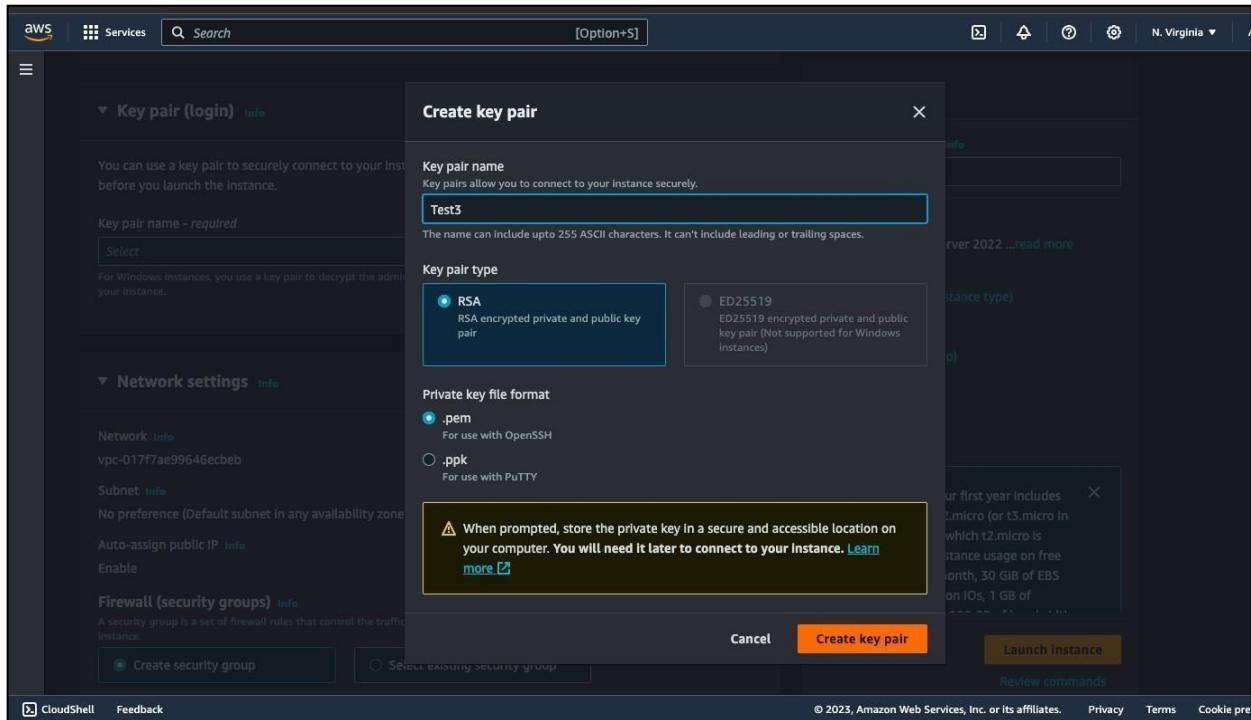
Step 1: Create a Instance



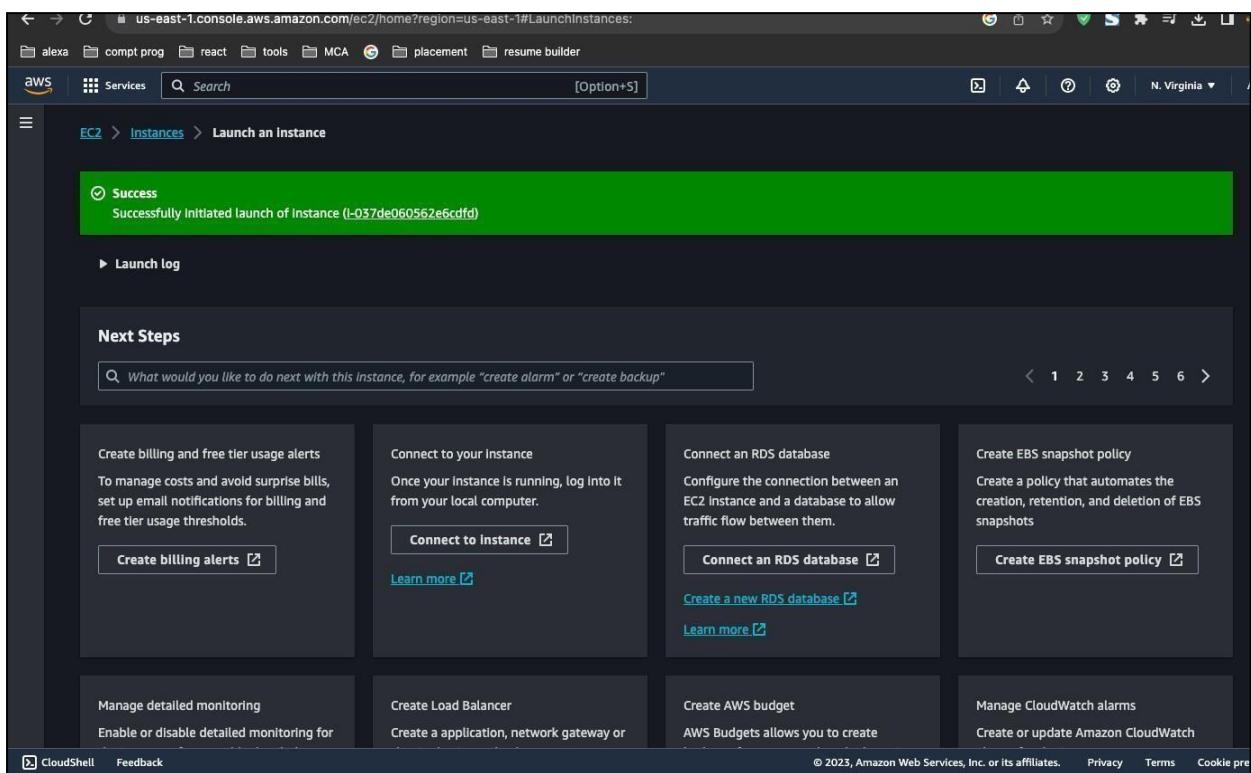
Step 2: In the "Configure Security Group" step, you'll need to configure the security group rules. Security groups act as firewalls to control inbound and outbound traffic to your instance. Ensure that you allow Remote Desktop Protocol (RDP) for Windows instances if you plan to access them remotely.



Step 3: Create a key pair for instance



Step 4: Launch the instance



Step 5: Launch the instance

The screenshot shows the AWS Management Console with the EC2 Instances page open. The left sidebar shows various EC2-related options like EC2 Dashboard, Global View, Events, Instances (selected), Instance Types, Launch Templates, etc. The main pane displays a table with one row for an instance named 'Test3'. The instance details are as follows:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Test3	i-037de060562e6cfdf	Pending	t2.micro	-	No alarms	us-east-1c

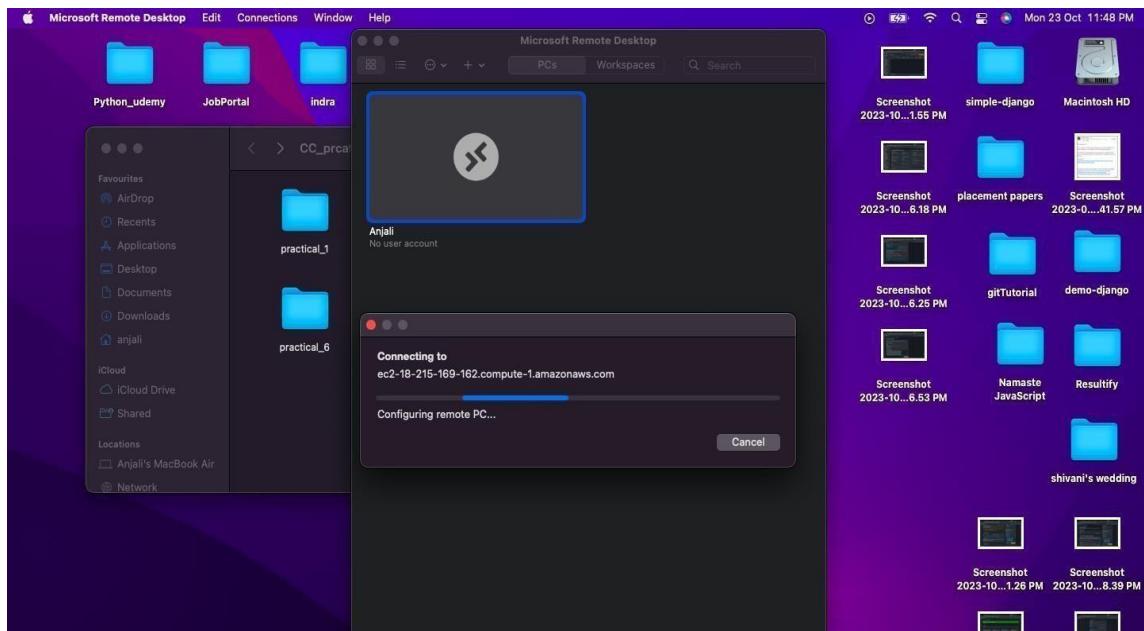
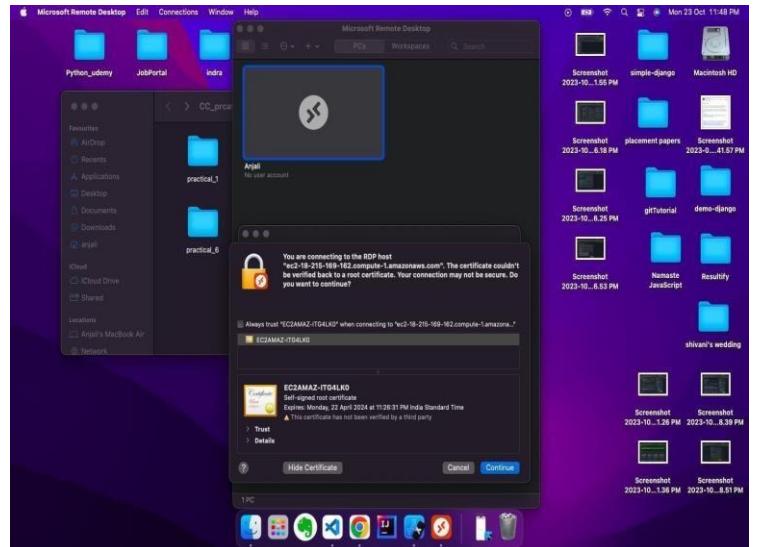
At the bottom of the main pane, there is a message: "Select an instance". The footer of the page includes links for CloudShell, Feedback, and copyright information: "© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

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."

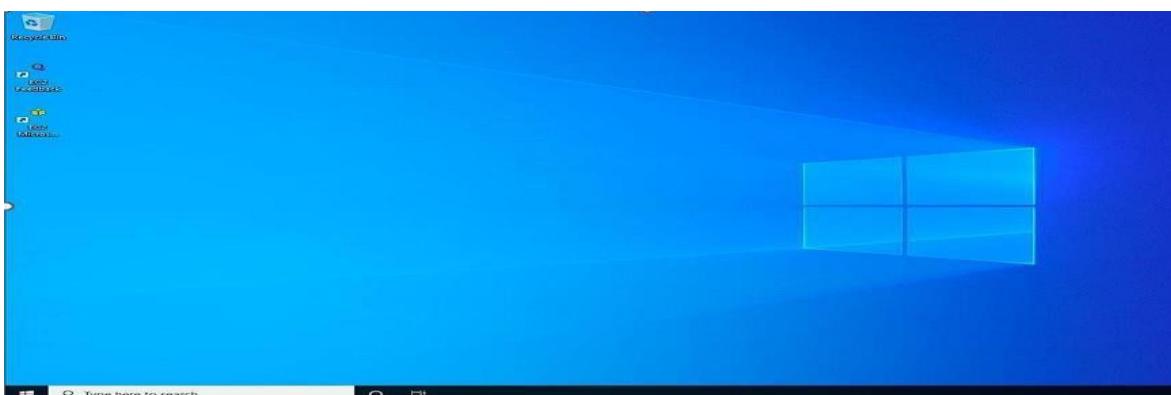
The screenshot shows the AWS Management Console with the connection details for the instance 'Test3'. The top part of the screen shows the instance ID: 'I-037de060562e6cfdf (Test3)'. Below it, the 'Connection Type' section is expanded, showing two options: 'Connect using RDP client' (selected) and 'Connect using Fleet Manager'. A note below says: 'You can connect to your Windows Instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below.' A button labeled 'Download remote desktop file' is present. Further down, it asks for connection details: 'Public DNS' (set to 'ec2-18-215-169-162.compute-1.amazonaws.com') and 'User name' ('Administrator'). There is also a 'Password' field and a 'Get password' link. At the bottom, a note says: 'If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.'

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.

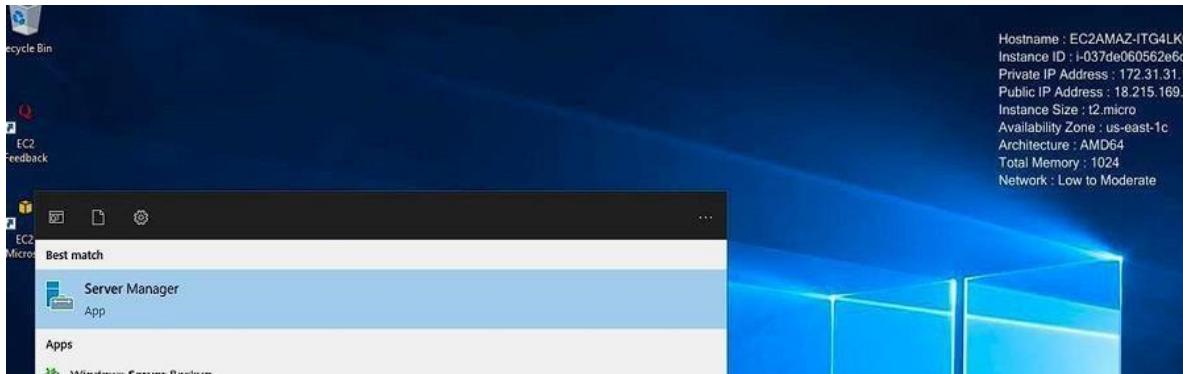
The screenshot shows a terminal window in the AWS CloudShell. The user has run the command `aws ec2 get-windows-password --instance-id i-037de060562e6cdff`. The output displays a private key in a large text block, which is partially obscured. At the bottom of the terminal, there are two buttons: "Cancel" and "Decrypt password".



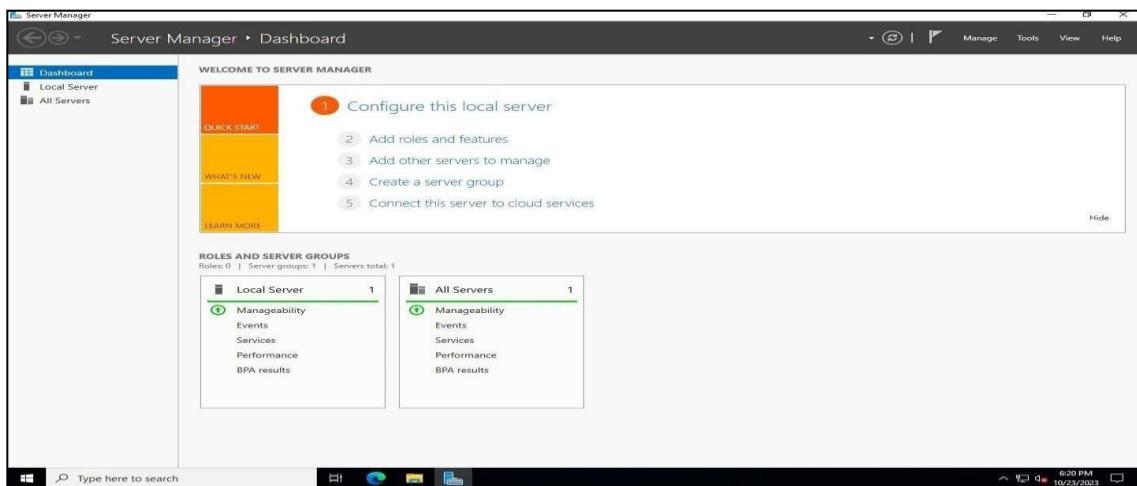
Step 8: To host a website, you'll need web server software. You can use Microsoft Internet Information Services (IIS) as a common choice for hosting websites on Windows Server. Follow these steps to install and configure IIS:



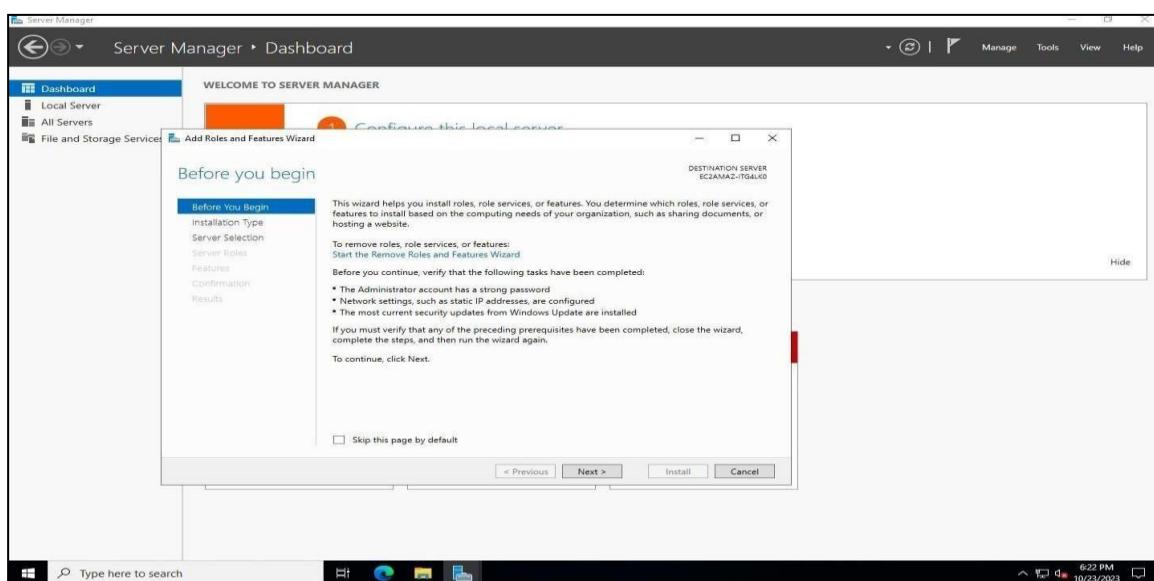
Step 9: In your Windows Server instance, open "Server Manager."



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.

Get Windows password Info

Use your private key to retrieve and decrypt the initial Windows administrator password for this instance.

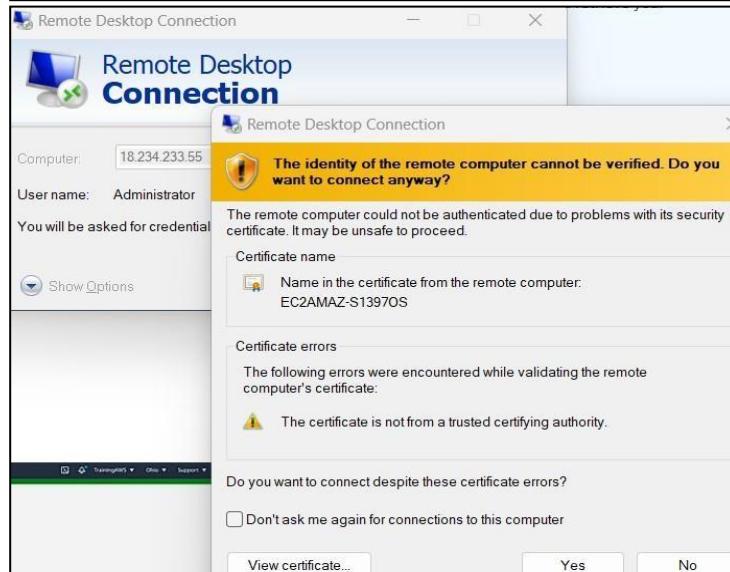
Instance ID
 i-02bdaf7be38bea29 (myserver)

Key pair associated with this instance
 newone

Private key
Either upload your private key file or copy and paste its contents into the field below.

Private key contents - *optional*

```
-----BEGIN RSA PRIVATE KEY-----  
9ZU3+aK/05+qwx4pbfaojNyL/K6SfwLohRiy1a3nlimKuTiYKOsz98Z3xnoWP+A  
B  
VPaQYIUCgYAHmKfhC7ROj3aa1upLYIWUunK6r+rD/tjOUbGgQVrNQmv2f6rO0  
Bs4  
Paf+h1tPy+0Hhp3v6VR3K9ritj6l9oS3gYcGRQX8vkLP9BQgAmGbfDNpDc7zyD  
7  
BlhX99ZRKY6UIKATwYGk+0hW1o8mCUNhy7zEEiSVzeorTvFu6QrQlQ==  
-----END RSA PRIVATE KEY-----
```



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:

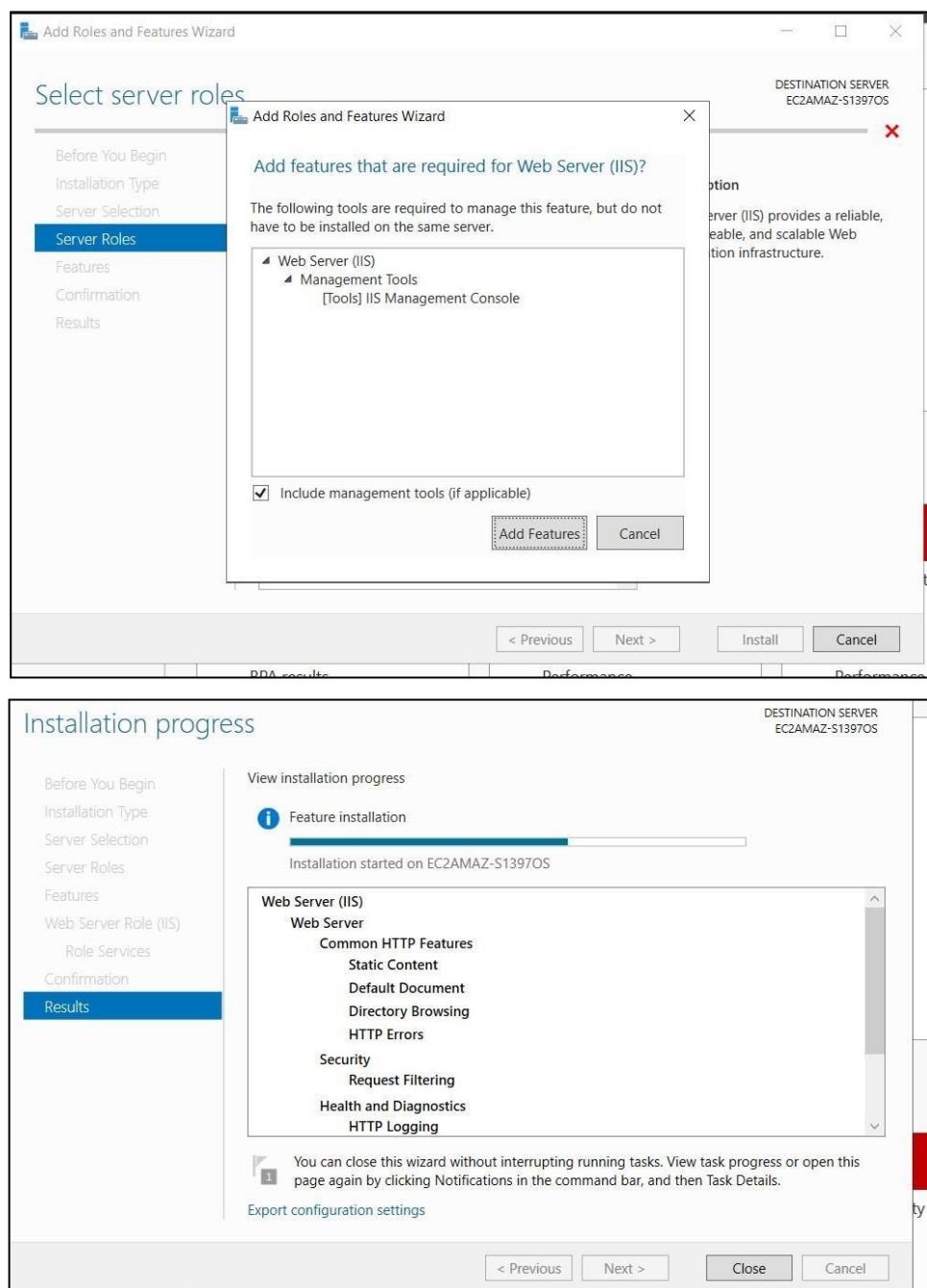
Click on Manage and select Add Roles and Features.

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:

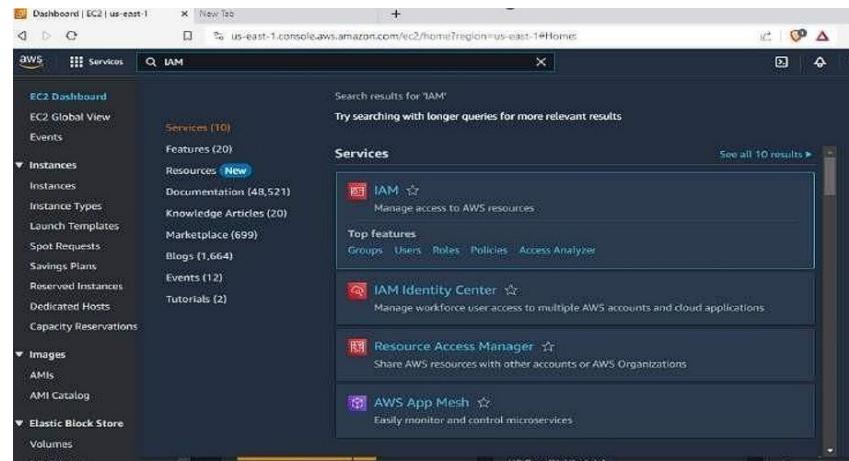
 Enter the public IP address of your EC2 instance in the browser's address bar.



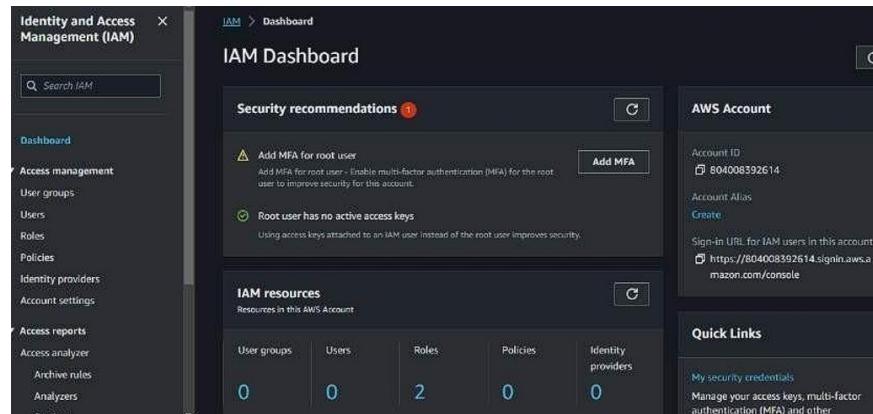
PRACTICAL 12

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

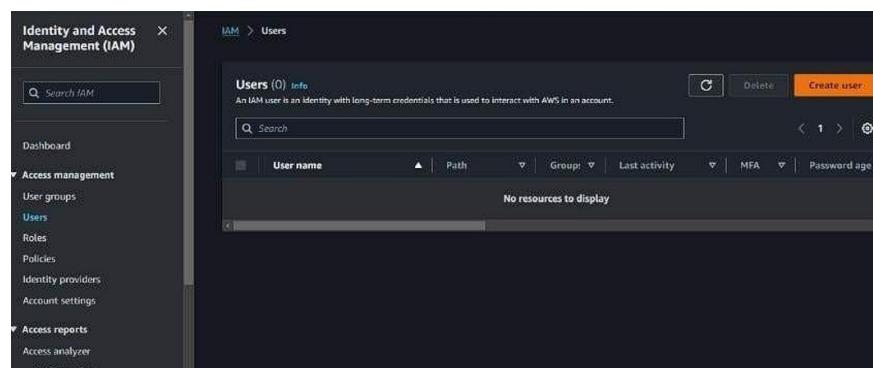
Step 1: Once logged in, navigate to the IAM (Identity and Access Management) Console. You can do this by searching for "IAM" in the AWS Management Console's search bar or by selecting "Security, Identity, & Compliance" and then "IAM" under the "Services" menu.



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.

The top screenshot shows the 'Review and create' step of the IAM user creation process. The user name is set to 'Anjali Kumar'. A note states: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and _ - (hyphen)'. Below this, there's an option to 'Provide user access to the AWS Management Console - optional.' A callout box asks 'Are you providing console access to a person?' with two radio button options: 'Specify a user in Identity Center - Recommended' (selected) and 'I want to create an IAM user'. A note under the second option says: 'We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keypairs, or a backup credential for emergency account access.' At the bottom right of the top window are 'Cancel' and 'Next' buttons.

The bottom screenshot shows the IAM Identity Center landing page. It features a large heading 'IAM Identity Center (successor to AWS Single Sign-On)' and subtext 'Manage workforce access to multiple AWS accounts and cloud applications.' Below this, a note says: 'Use IAM Identity Center to connect an existing directory or use the built-in Identity Center directory to manage user access to AWS accounts and cloud applications.' On the right side, there's a 'Getting started' section with 'Enable IAM Identity Center' (with an 'Enable' button), 'Get started with IAM Identity Center', and 'IAM Identity Center prerequisites'. At the bottom left is a 'Getting started' link.

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.

The left screenshot shows a modal dialog titled 'Enable IAM Identity Center'. It contains a warning: 'IAM Identity Center requires AWS Organizations. We detected that your AWS account does not currently use this service. After you create an organization, you cannot join this account to another organization until you delete its current organization.' Below this, a note says: 'AWS Organizations provides the following benefits: 1. Enables single payer and centralized cost tracking 2. Lets you create and invite other AWS accounts 3. Allows you to apply policy-based controls 4. Helps you simplify organization-wide management of AWS services.' At the bottom are 'Cancel' and 'Create AWS organization' buttons.

The right screenshot shows the 'Dashboard' page of IAM Identity Center. It includes a 'Recommended setup steps' section with 'Step 1: Choose your identity source' (with a note about Identity Center directory) and 'Step 2: Manage access to multiple AWS accounts' (with a note about giving users and groups access to specific AWS accounts). On the right, there's a 'Settings summary' sidebar with tabs for 'Identity source', 'Region', 'AWS access portal URL', and a link to 'https://d-960789.net'.

Step 6: Give limited permission rights by navigating to permissions

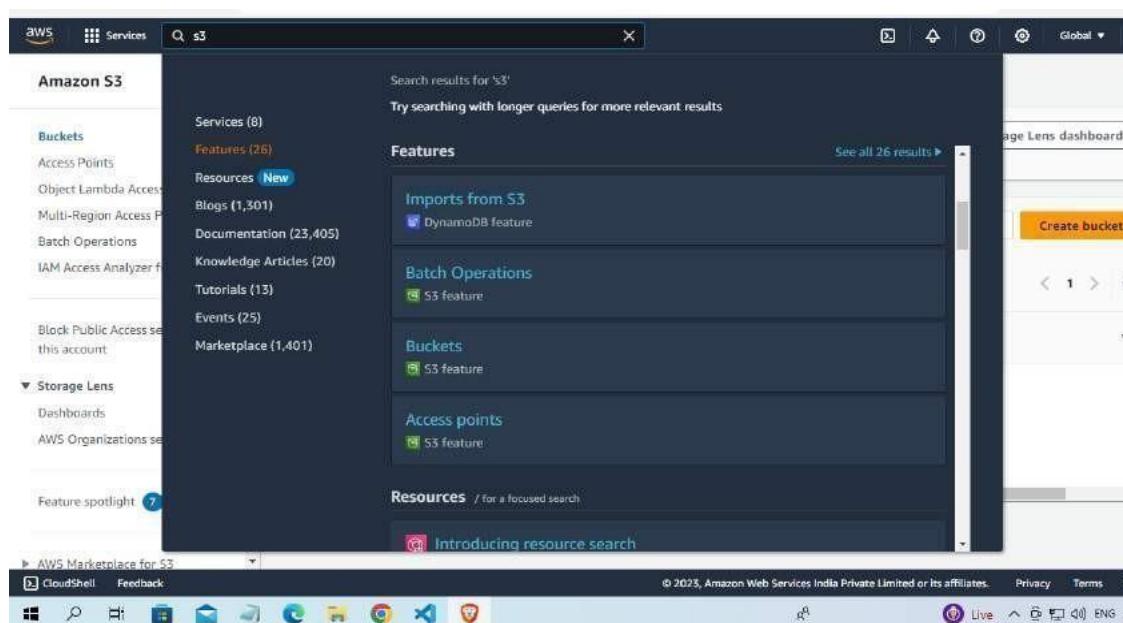
PRACTICAL 13

Create a bucket by using S3 AWS service

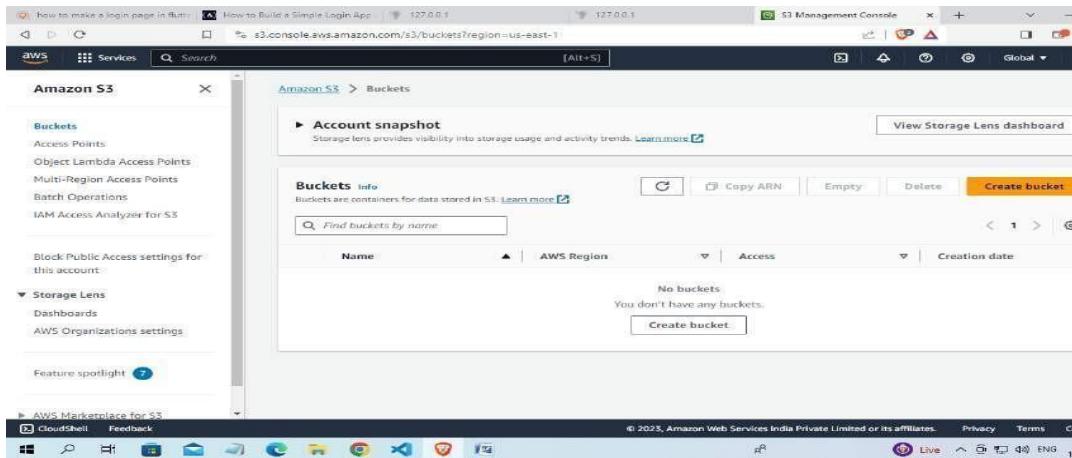
Objective: The primary objective of Amazon S3 (Simple Storage Service) buckets in AWS is to provide a highly scalable, durable, and secure storage solution for a wide range of use cases. S3 buckets serve as containers for storing and managing data, and their key objectives include:

- 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.

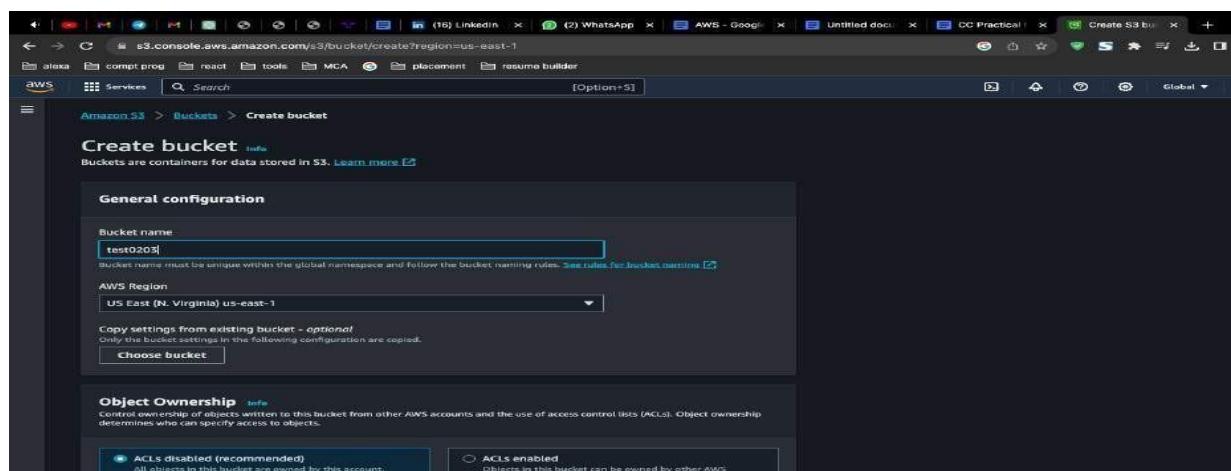
Step 1: Once logged in, navigate to the S3 dashboard. You can do this by searching for "S3" in the AWS Management Console's search bar or by selecting "Storage" and then "S3" under the "Services" menu.



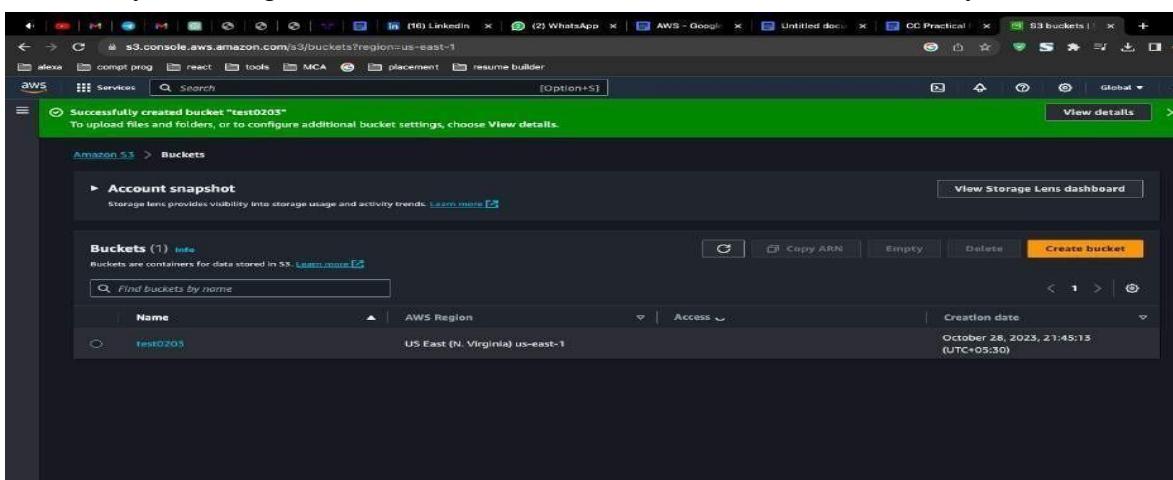
Step 2: Click the "Create bucket" button.



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.



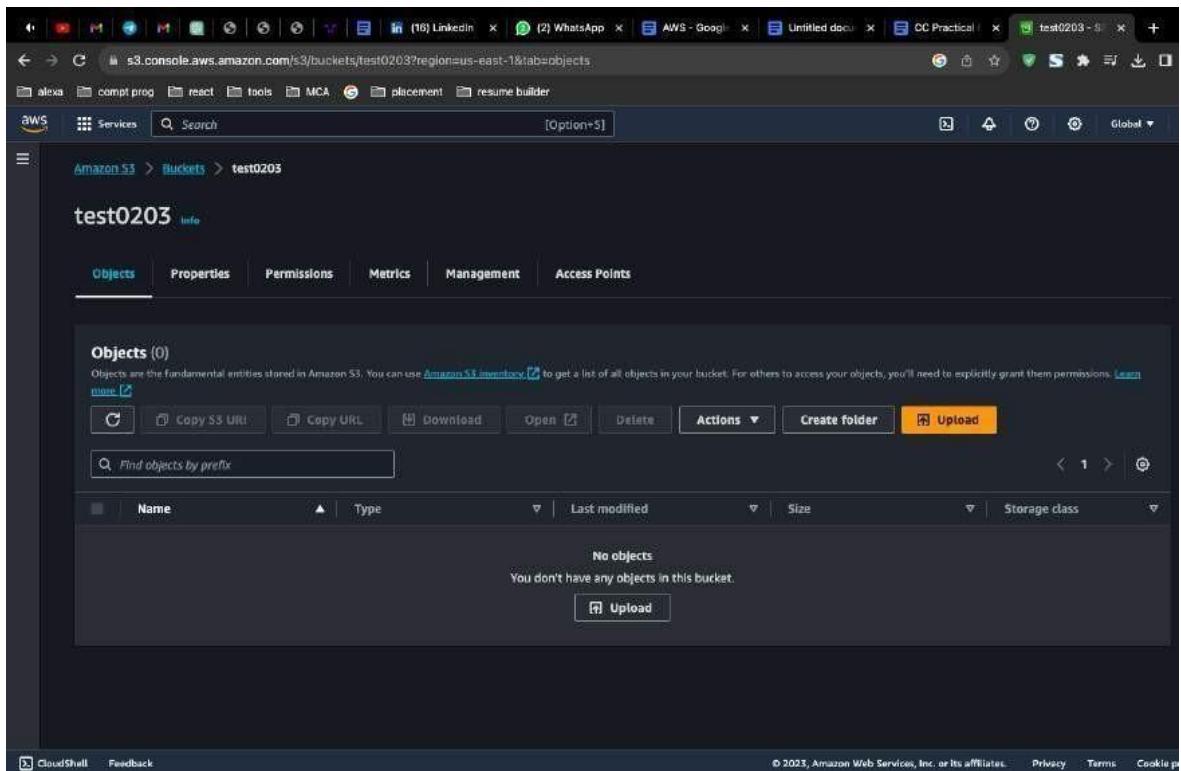
PRACTICAL 14

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"

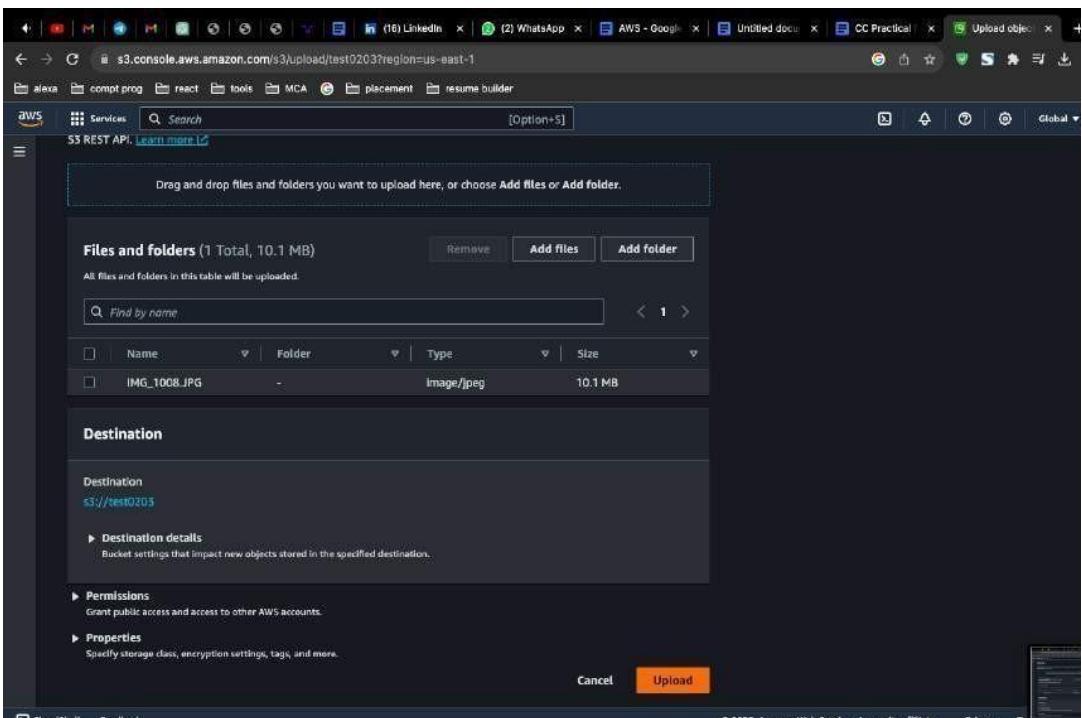
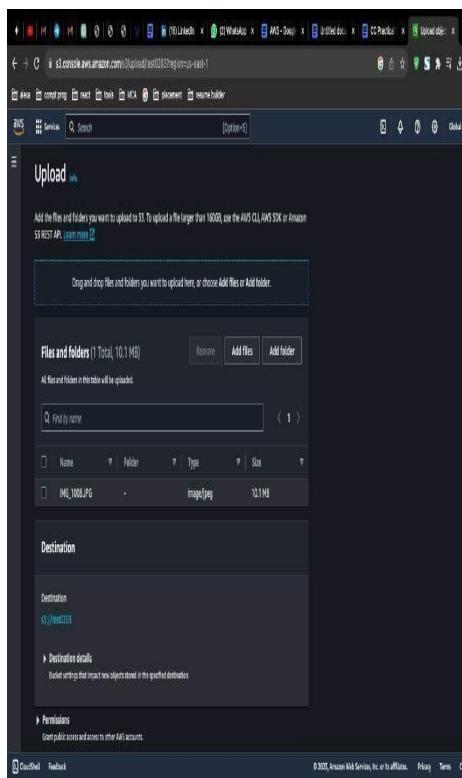
is to demonstrate how to upload files to an Amazon S3 bucket. This can be useful for various purposes, such as hosting static websites, sharing files, or distributing public content.

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.



Step 3: File Uploaded in Bucket

The screenshot shows the AWS S3 console interface. At the top, a progress bar indicates 'Uploading' with '1%' completed. Below it, a message states: 'Total remaining: 1 File: 10.0 MB (99.61%)', 'Estimated time remaining: 4 minutes', and 'Transfer rate: 44.8 KB/s'. The main area is titled 'Upload: status' with a note: 'The information below will no longer be available after you navigate away from this page.' A 'Summary' section shows the destination as 's3://test0203'. Under 'Succeeded', there is 1 file (144.0 KB, 1.39%). Under 'Failed', there are 0 files (0 B, 0%). Below this, there are tabs for 'Files and folders' (selected) and 'Configuration'. The 'Files and folders' section shows a table with one item: 'IMG_1008.JPG' (1 Total, 10.1 MB). The table includes columns for Name, Folder, Type, Size, Status, and Error. The status for the file is 'Succeeded'.

The screenshot shows the AWS S3 console interface. A green banner at the top indicates 'Upload succeeded' with a link to 'View details below.'. Below this, the 'Upload: status' section has a note: 'The information below will no longer be available after you navigate away from this page.' A 'Summary' section shows the destination as 's3://test0203'. Under 'Succeeded', there is 1 file (10.1 MB, 100.00%). Under 'Failed', there are 0 files (0 B, 0%). Below this, there are tabs for 'Files and folders' (selected) and 'Configuration'. The 'Files and folders' section shows a table with one item: 'IMG_1008.JPG' (1 Total, 10.1 MB). The table includes columns for Name, Folder, Type, Size, Status, and Error. The status for the file is 'Succeeded'.

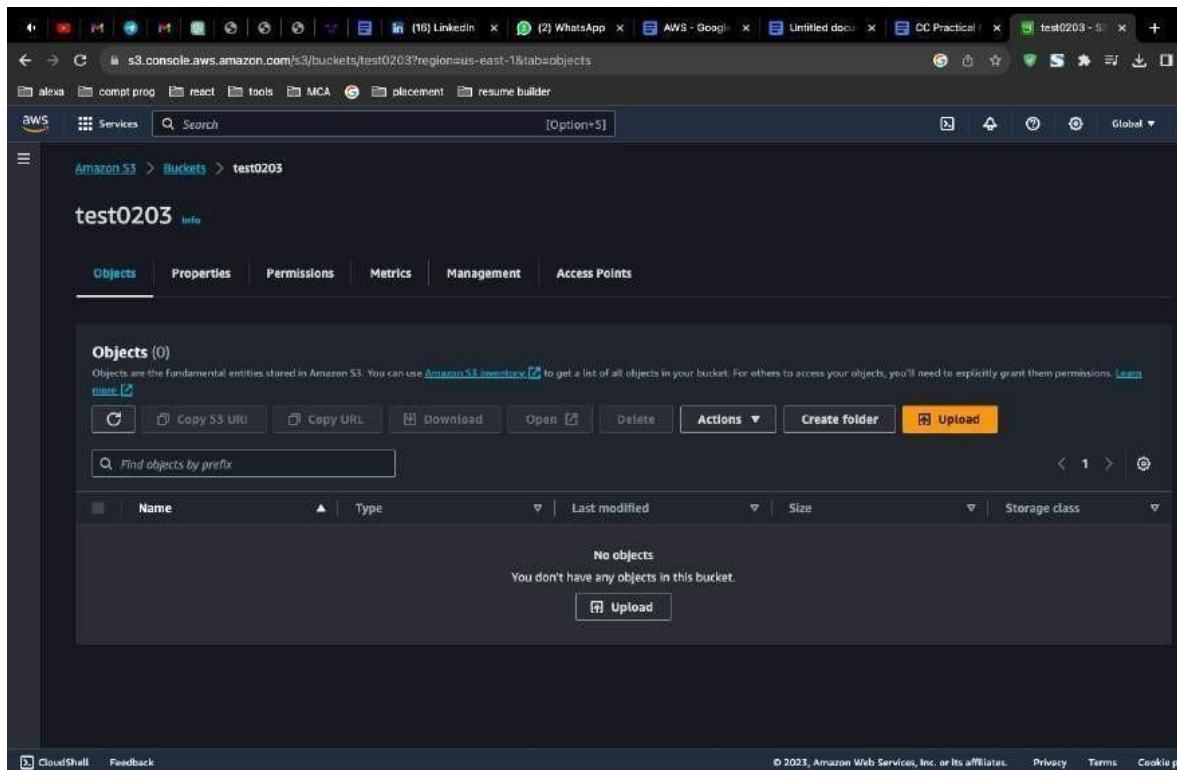
PRACTICAL 15

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.

The left screenshot shows the 'Upload' page in the AWS S3 console. A file named 'IMG_1008.JPG' (10.1 MB) is selected for upload. The right screenshot shows the 'Destination' configuration step, where the file is being uploaded to the 'test0203' bucket.

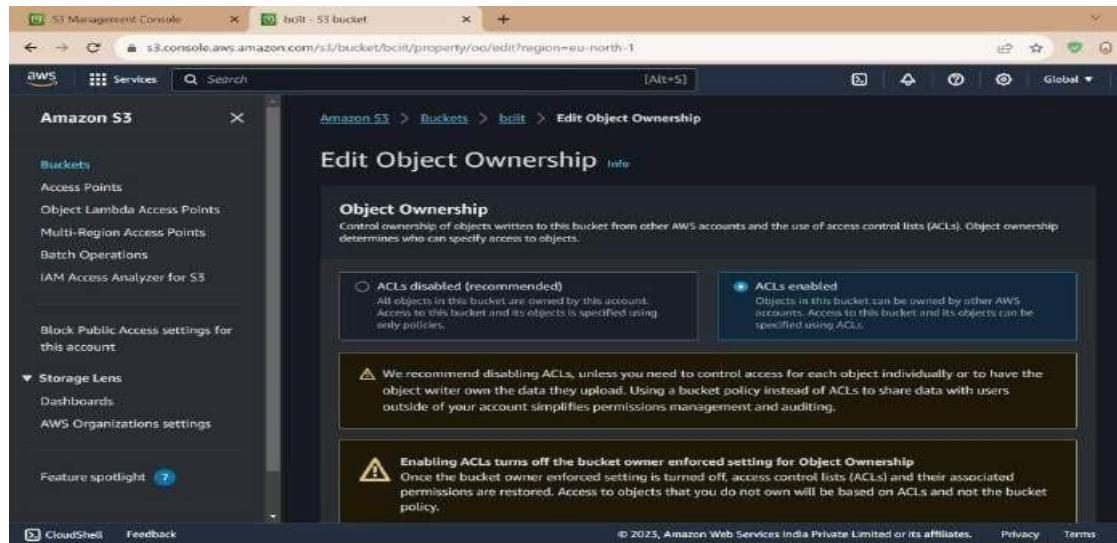
Step 3: File Uploaded in Bucket

The left screenshot shows the 'Uploading' status page with a progress bar at 1%. The right screenshot shows the 'Upload: status' page, which indicates that the upload was successful with 1 file (10.1 MB).

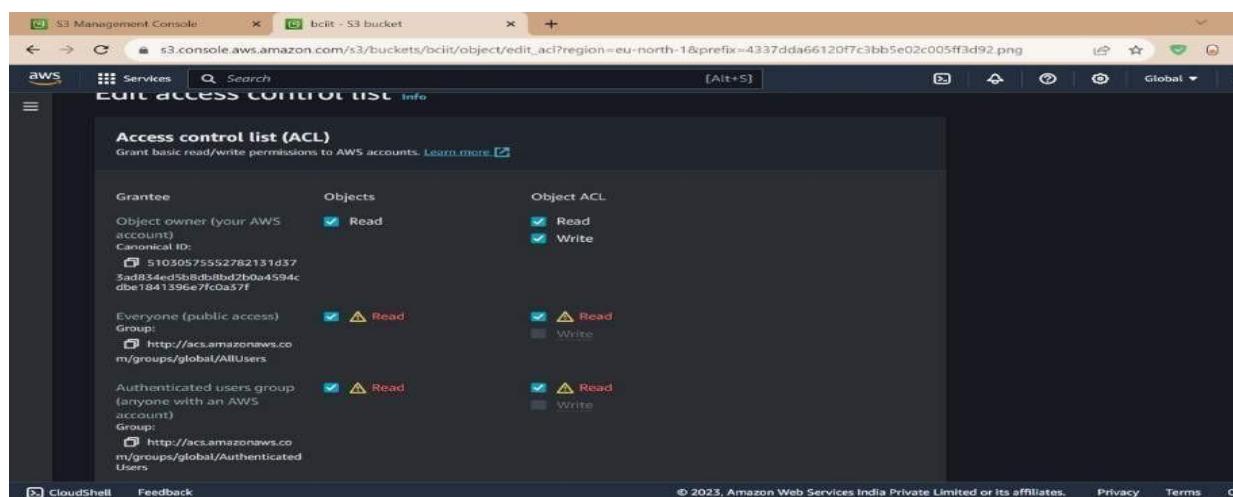
Step 5: Go to permissions and click on bucket owner enforced

The screenshot shows the 'Access control list (ACL)' section of the AWS S3 Management Console. It highlights a note: 'This bucket has the bucket owner enforced setting applied for Object Ownership'. The note explains that when 'bucket_owner_enforced' is applied, use bucket policies to control access.

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

```

[root@ip-172-31-36-199 ec2-user]# wget https://bciiit.s3.eu-north-1.amazonaws.com/4337dda66120f7c3bb5e02c005ff3d92.png
--2023-10-16 05:05:42  https://bciiit.s3.eu-north-1.amazonaws.com/4337dda66120f7c3bb5e02c005ff3d92.png
Resolving bciiit.s3.eu-north-1.amazonaws.com (bciiit.s3.eu-north-1.amazonaws.com) ... 52.95.171.72, 52.95.171.40
Connecting to bciiit.s3.eu-north-1.amazonaws.com (bciiit.s3.eu-north-1.amazonaws.com) |52.95.171.72|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13025 (13K) [image/png]
Saving to: '4337dda66120f7c3bb5e02c005ff3d92.png'

4337dda66120f7c3bb5e02c005ff3d92.png 100%[=====] 12.72K --.-KB/s   in 0.0s

2023-10-16 05:05:43 (96.4 MB/s) - '4337dda66120f7c3bb5e02c005ff3d92.png' saved [13025/13025]

[root@ip-172-31-36-199 ec2-user]# ls
4337dda66120f7c3bb5e02c005ff3d92.png
[root@ip-172-31-36-199 ec2-user]#

```

i-0e9ed085ef8c5e863 (bciiit)
PublicIPs: 13.51.207.215 PrivateIPs: 172.31.36.199

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.

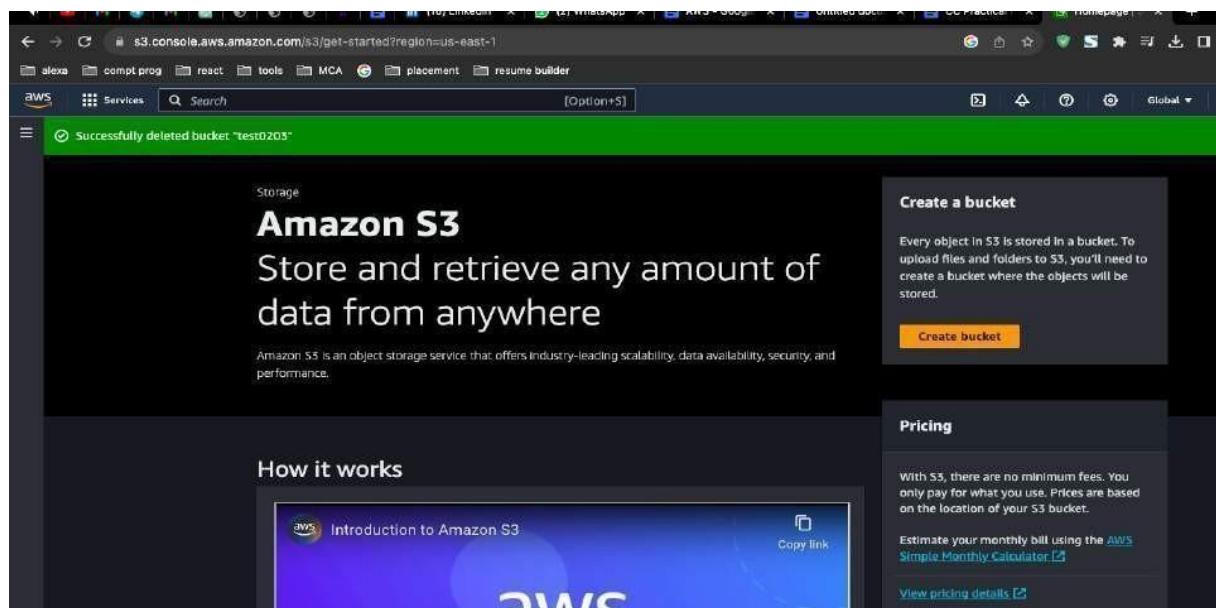
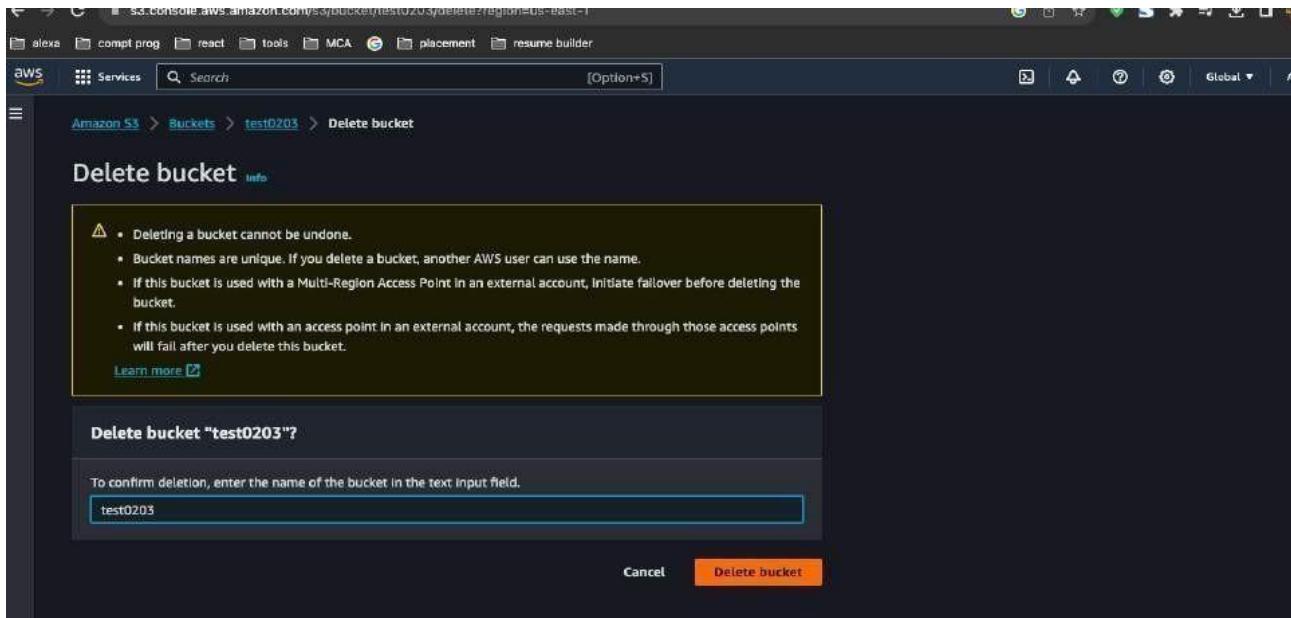
Step 1: In the S3 dashboard, click on the name of the bucket from which you want to delete objects.

The screenshot shows the AWS S3 Buckets page. At the top, there's a header with the AWS logo and a search bar. Below the header, the URL is s3.console.aws.amazon.com/s3/buckets?region=us-east-1®ion=us-east-1. The main content area has a title 'Buckets (1) [Info](#)' and a note 'Buckets are containers for data stored in S3. [Learn more](#)'. There are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. A search bar labeled 'Find buckets by name:' is present. A table lists the bucket details:

Name	AWS Region	Access	Creation date
test0203	US East (N. Virginia) us-east-1	Bucket and objects not public	October 28, 2023, 21:45:13 (UTC-05:30)

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:

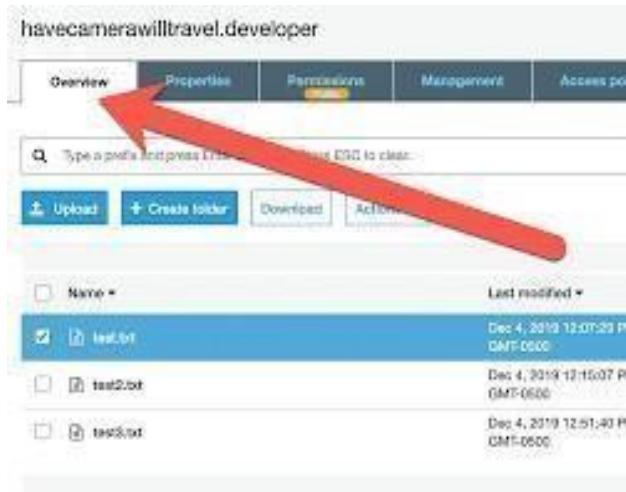
1. Type the following command to check the installed GCC version:

```
gcc -version
```

2. The command should display the installed GCC version.

Screenshots:

1. Object URL in S3 Console:



2. wget command in terminal:

```
adam@adam-Vostro-3460:~$ wget https://en.wikipedia.org/wiki/Wget
--2014-02-09 07:38:43-- https://en.wikipedia.org/w/index.php?title=Wget
Resolving en.wikipedia.org (en.wikipedia.org)... 209.154.224.26
Connecting to en.wikipedia.org (en.wikipedia.org)|209.154.224.26|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'Wget'

[ ==>          ] 91,646      583K/s   in 0.2s

2014-02-09 07:38:43 (583 KB/s) - 'Wget' saved [91646]
```

The screenshot shows a terminal window with a dark background. It displays the command `wget https://en.wikipedia.org/wiki/Wget` being run by a user named adam. The output shows the progress of the download, including the connection details, the HTTP status code 200 OK, the file type as text/html, and the final save location as 'Wget'. The download speed is shown as 583K/s and the time taken is 0.2s. The terminal prompt `adam@adam-Vostro-3460:~$` is visible at the bottom.

3. File listing showing downloaded object:

```
C:\> C:\Users\muralikr... /images/  
[Truncated]_THUMBNAIL.jpg  
7/23/2011 7:17 AM 43 0.39 7.xls  
18/5/2011 22:45 PM 43 251ETPCF001  
1/23/2011 7:17 AM 23744 251ETPCF001  
7/23/2011 7:17 AM 23744 251ETPCF001  
7/23/2011 7:17 AM 500 Activision.xls  
7/23/2011 7:17 AM 23744 Activision.xls  
7/23/2011 7:17 AM 3321 access-american.xls  
7/23/2011 7:17 AM 3321 access-american.xls  
7/23/2011 7:17 AM 3321 access-american.xls  
7/23/2011 7:17 AM 3321 access-left.xls  
8/25/2011 21:09 AM 147 access-right.xls  
8/25/2011 21:09 AM 167 access-right.xls  
8/25/2011 21:09 AM 888 calendarAcronis.xls  
1/23/2011 7:17 AM 43 0.39 7.xls  
7/23/2011 7:17 AM 23744 251ETPCF001  
7/23/2011 7:17 AM 23744 251ETPCF001  
7/23/2011 7:17 AM 191 COMPRESS03.xls  
7/23/2011 7:17 AM 1637 compress03.xls  
1/23/2011 7:17 AM 23744 251ETPCF001  
7/23/2011 7:17 AM 3859 compress03.xls  
7/23/2011 7:17 AM 2633 dokuwiki.xls  
7/23/2011 7:17 AM 2382 dokuwiki.xls  
7/23/2011 7:17 AM 10246 dokuWiki_1201.htm  
5/23/2011 7:17 AM 5379 dokuWiki_1201.htm  
7/23/2011 7:17 AM 24362 dokuWiki_1201.htm  
7/23/2011 7:17 AM 23395 DBRBanner748938.xls  
7/23/2011 7:17 AM 24362 dokuBanner_13143_7.xls  
7/23/2011 7:17 AM 22986 DBRBanner_419-133.xls
```

4. GCC installation command:

```
i: zip 5.0.2libstdc++.archive archive for the file  
i: zlib1g:udeb-1.2.3.1-1df5.udeb compression library  
[sudo] password for muralikrishna:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Package gcc is not available, but is referred to by another package.  
This may mean that the package is missing, has been obsoleted, or  
is only available from another source  
E: Package 'gcc' has no installation candidate  
muralikrishna@muralikrishna-Inspiron-5570:~/programs$ cd programs  
muralikrishna@muralikrishna-Inspiron-5570:~/programs$ gcc welcome  
Command 'gcc' not found, but can be installed with:  
sudo apt install gcc  
muralikrishna@muralikrishna-Inspiron-5570:~/programs$ sudo apt in  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Package gcc is not available, but is referred to by another package.  
This may mean that the package is missing, has been obsoleted, or  
is only available from another source
```

5. GCC version verification:

```
F:\>type wikipedija.c  
#include <stdio.h>  
  
int main (void) {  
    printf("Pozdravljeni na Wikipediji!\n");  
}  
  
F:\>gcc --version  
gcc (GCC) 3.4.2 (mingw-special)  
Copyright (C) 2006 Free Software Foundation, Inc.  
This is free software; see the source for copying conditions. The  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR  
  
F:\>gcc wikipedija.c -o wikipedija  
F:\>wikipedija  
Pozdravljeni na Wikipediji!  
F:\>
```

PRACTICAL 18

Create VPC and implement EC2 services on it.

The screenshot shows the AWS VPC Dashboard with the 'Resources' tab selected. It displays the following information:

- Service Health:**
 - Amazon VPC - US East (N. Virginia) - Service is operating normally
 - Amazon EC2 - US East (N. Virginia) - Service is operating normally
- Additional Information:**
 - VPC Documentation
 - All VPC Resources
 - Topics
 - Report an issue

The screenshot shows the AWS VPC Dashboard with the 'Create VPC' wizard step 1. It displays the following information:

- Virtual Private Cloud:**
 - New
 - VPC ID: vpc-d6fb1
 - Name: myVPC
 - Status: available
 - Region: us-east-1
 - DHCP options set: vpc-d6fb1
 - Route tables: None
 - Internet Gateways: None
 - Egress Only Internet Gateways: None
 - DHCP Options Sets: None
 - Classic IP: None
 - Endpoints: None
 - NAT Gateways: None
 - Peering Connections: None
 - Security: myVPC

The screenshot shows the AWS VPC Dashboard with the 'Create Customer Gateway' dialog open. It displays the following fields:

Name tag	myCG
Type	Static
IP address	192.168.1.1

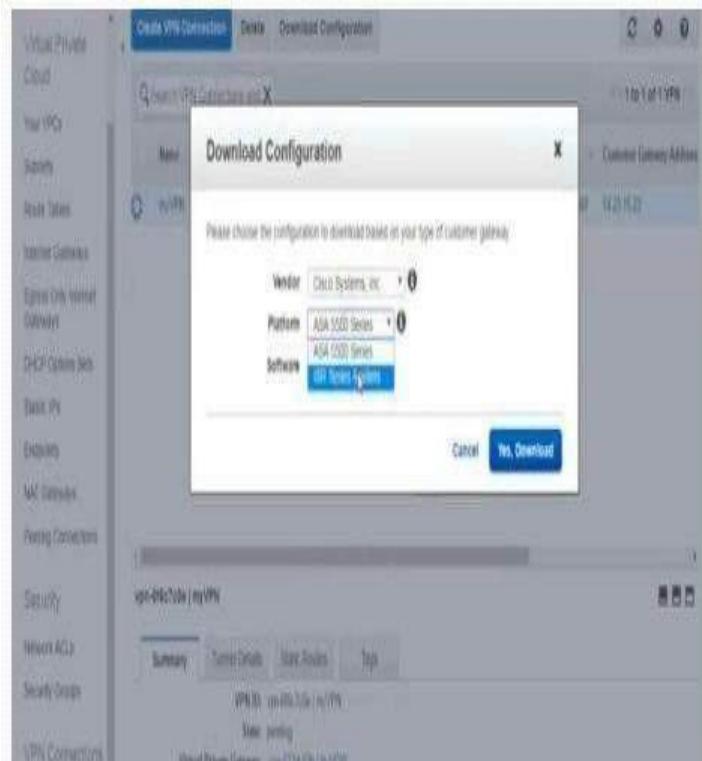
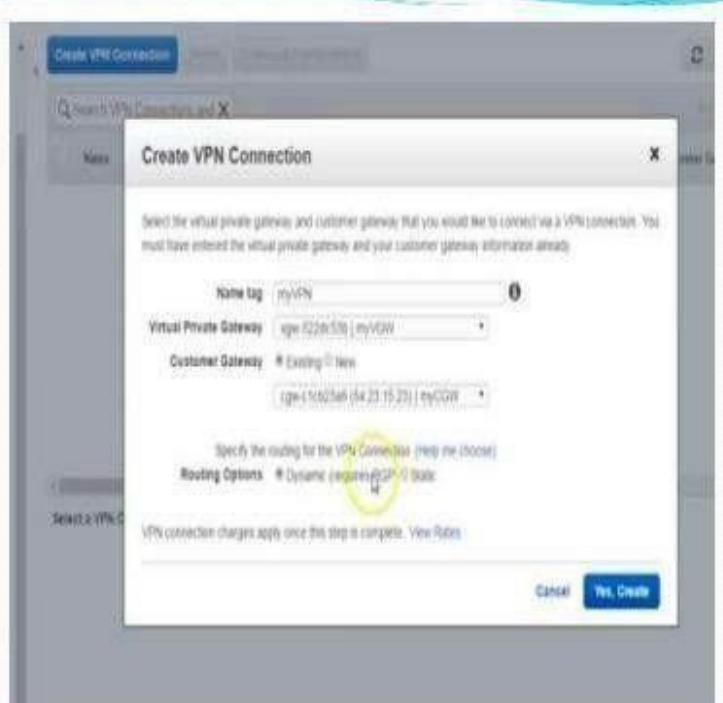
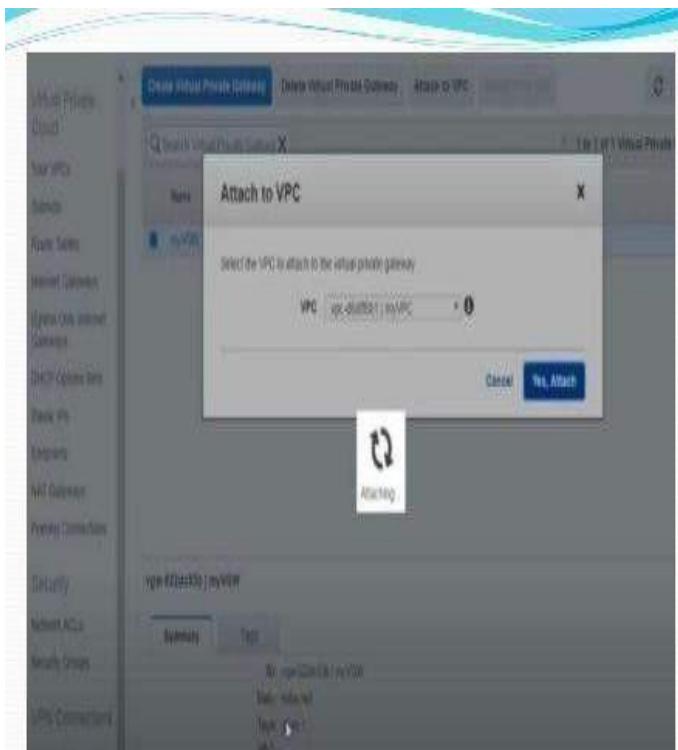
Below the dialog, there is a note: "CREATE A CUSTOMER GATEWAY INSTANCE".

The screenshot shows the AWS VPC Dashboard with the 'Create Virtual Private Gateway' dialog open. It displays the following information:

- Virtual Private Cloud:**
 - New
 - ID: ipg-524e3b
 - Name: myVGW
 - Type: static
 - VPC: myVPC

The 'Summary' tab is selected, showing the following details:

ID	ipg-524e3b myVGW
Name	myVGW
Type	static
VPC	myVPC



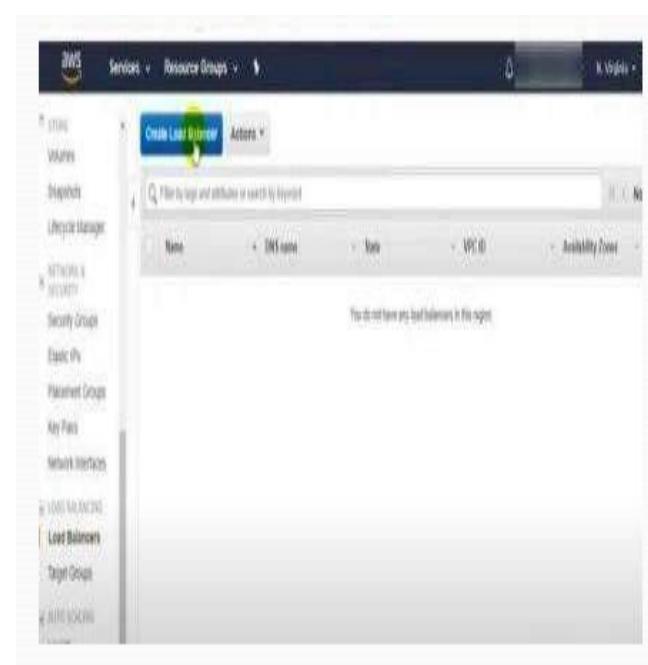
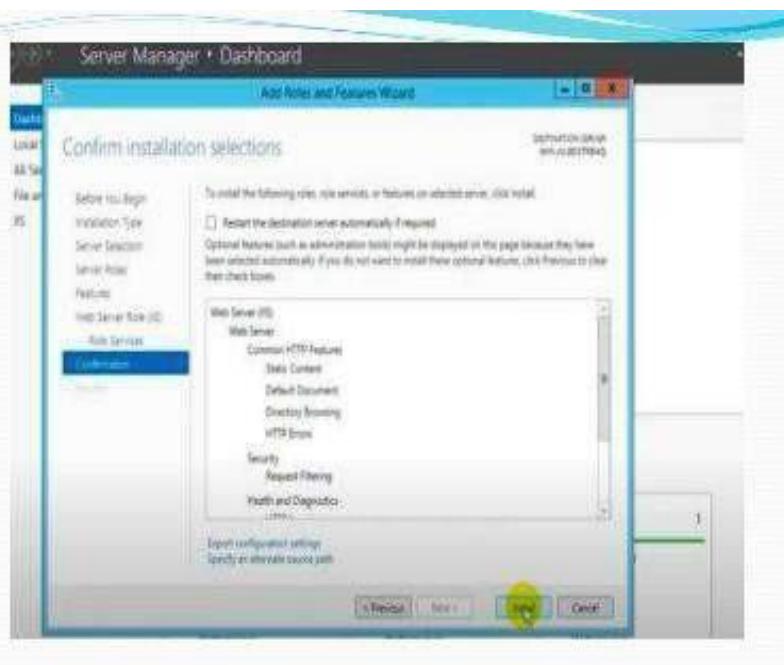
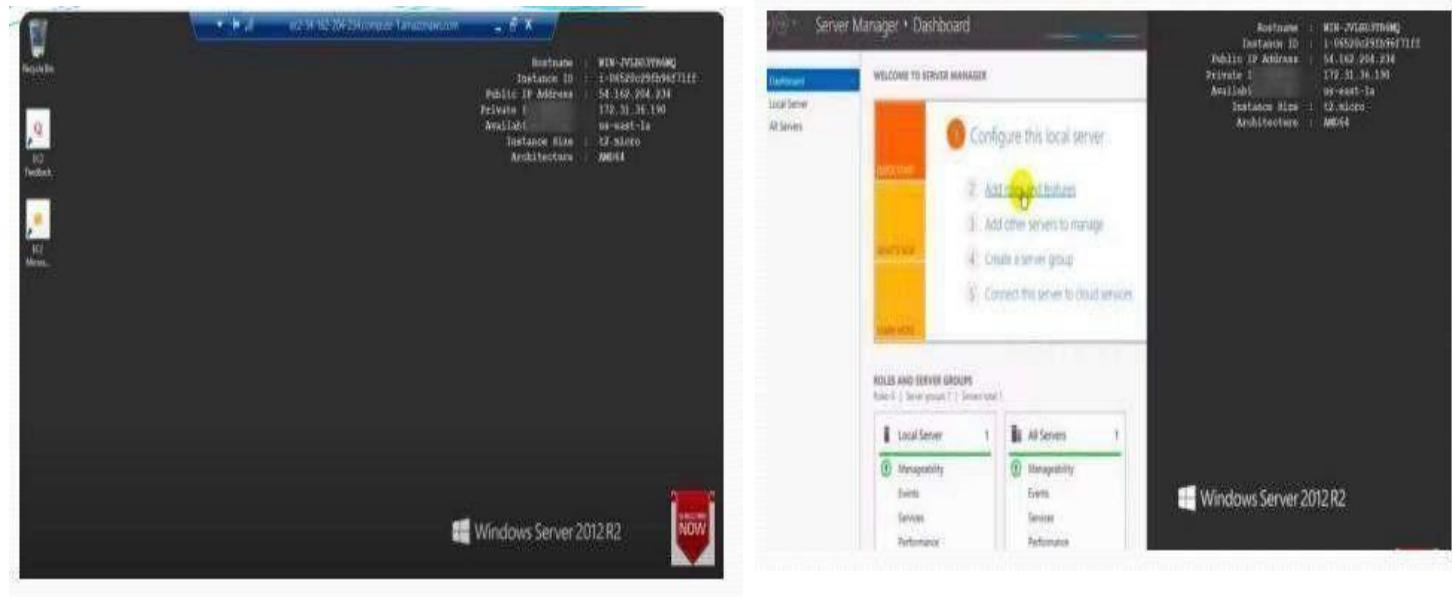
```

Line 1: 1 Amazon Web Services
Line 2: 2 Virtual Private Cloud
Line 3: 3
Line 4: 4 AWS utilizes unique identifiers to manipulate the configuration of
Line 5: 5 a VPN Connection. Each VPN Connection is assigned an identifier and is
Line 6: 6 associated with two other identifiers, namely the
Line 7: 7 Customer Gateway Identifier and Virtual Private Gateway Identifier.
Line 8: 8
Line 9: 9
Line 10: 10 Your VPN Connection ID : vpn-6f6c7c9e
Line 11: 11 Your Virtual Private Gateway ID : vpg-522a53b
Line 12: 12 Your Customer Gateway ID : cgw-11023af (ex-23.15.23) myCGW
Line 13: 13
Line 14: 14 This configuration consists of two tunnels. Both tunnels must be
Line 15: 15 configured on your Customer Gateway. Only a single tunnel will be up at a
Line 16: 16 time to the VGW.
Line 17: 17 You may need to populate these values throughout the config based on your setup:
Line 18: 18 <outside_interface> - External interface of the ASA
Line 19: 19 <outside_access_in> - Inbound ACL on the external interface
Line 20: 20 <asn_vpn_map> - Outside crypto map
Line 21: 21 <vpn_subnet> and <vpn_subnet_mask> - VPC address range
Line 22: 22 <local_subnet> and <local_subnet_mask> - Local subnet address range
Line 23: 23 <sla_monitor_address> - Target address that is part of acl-sla to run SLA monitoring
Line 24: 24
Line 25: 25
Line 26: 26
Line 27: 27
Line 28: 28
Line 29: 29 #1: Internet Key Exchange (IKE) Configuration

```

PRACTICAL 19

Implement & Configure load balancing with all necessary steps.



Select load balancer type

Amazon Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers (new), and Classic Load Balancers. Choose the load balancer type that meets your needs.

Learn more about which load balancer is right for you.

Application Load Balancer

Choose an Application Load Balancer when you need a flexible feature set for your web applications via HTTP and HTTPS traffic. Operating at the request layer, Application Load Balancers provide advanced routing and security features, targeted at application architectures, including microservices.

[Create](#)

Network Load Balancer

Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP and static IP addresses for your application. Operating at the connection layer, Network Load Balancers are capable of handling millions of requests per second.

[Create](#)

Classic Load Balancer

PREVIOUS GENERATION
for HTTP, HTTPS, and TCP

Choose a Classic Load Balancer when you have an application running in the EC2 Classic network.

[Learn more >](#)

[Create](#)

Step 1: Configure Load Balancer

Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or many listeners, and select a network. The default configuration is an internet listener that receives HTTP traffic on port 80.

Name:

Scheme: internet-facing
 internal

IP address type:

Listeners

A Listener is a process that listens for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
HTTP	80

Step 3: Configure Security Groups

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new group or select an existing one.

Assign a security group:

- Create a new security group
- Select an existing security group

[Create New Security Group](#)

Security Group ID	Name	Description
sg-1234567890abcdef	default	Default VPC security group
sg-098765432109876543	test_sg	Test security group created 2019-11-09T12:30:21Z (1d 10h)

Step 4: Configure Routing

Name:

Target type: instance
 ip
 Lambda function

Protocol:

Port:

Health checks

Protocol:
Port:

Advanced health check settings

[Cancel](#) [Previous](#) [Next](#)

Step 5: Register Targets

Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the targets as soon as the registration process completes and passes the initial health checks.

Registered targets

To deregister instances, select one or more registered instances and then click Remove.

Instance	Name	Port	Status	Security groups	Zone
1-08101-000000000000	ELB SERVER 0	80	Green	ELBSG1	us-east-1a
1-08101-000000000001	ELB SERVER 1	80	Green	ELBSG1	us-east-1a

Instances

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

Next Step

Step 6: Review

Tags

Security groups

Security group: sg-01103e16c70c26

Routing

Target group: New target group

Target group name: T01

Port: 80

Target type: Instance

Protocol: HTTP

Health check protocol: HTTP

Path: /

Health check port: Traffic port

Healthy threshold: 3

Unhealthy threshold: 3

Next Step

Create target group

Actions

Filter by tags and attributes or search by keyword:

Name	Port	Protocol	Target type	Load Balancer	VPC ID	Region
T01	80	HTTP	Instance	Application	vpca29930	us-east-1

Registered targets

Instance ID	Name	Port	Availability Zone	Status	Description
1-08101-000000000000	ELB SERVER 0	80	us-east-1a	Initial	Target registration is in progress
1-08101-000000000001	ELB SERVER 1	80	us-east-1a	Initial	Target registration is in progress

Availability Zones

Availability Zone	Target count	Healthy?
us-east-1a	1	No (Availability Zone contains no healthy targets)
us-east-1b	1	No (Availability Zone contains no healthy targets)

Load Balancer (EC2 Manager) applicationnlb-1123994945.us-east-1.elb.amazonaws.com

SERVER 1 HELLO AVAILABILITY ZONE A

applicationnlb-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.

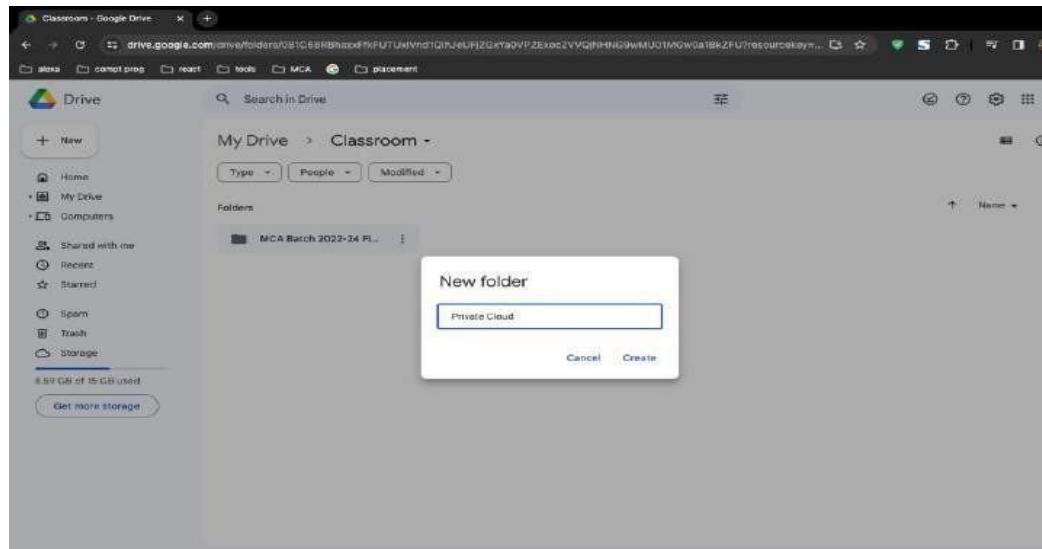
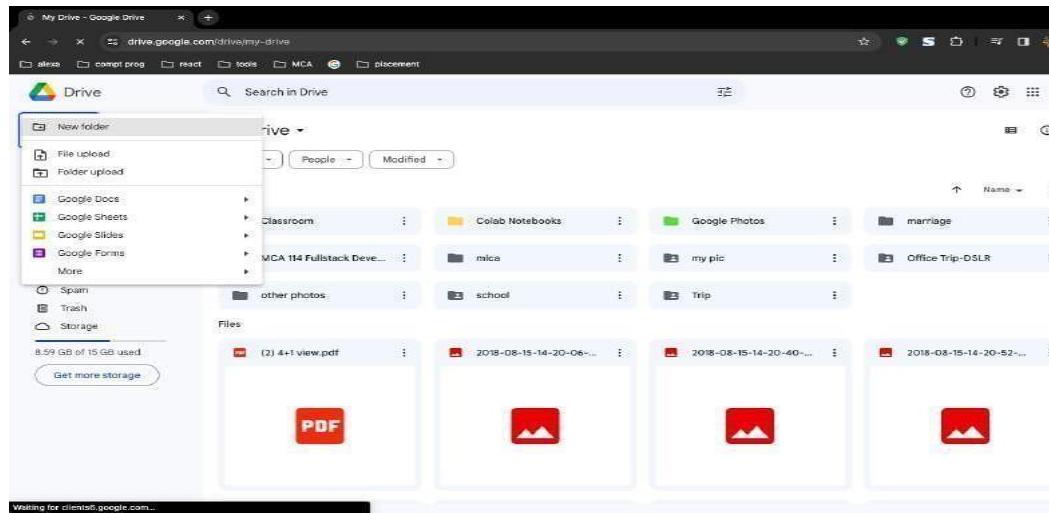
Sign in with your Google account or create one if you don't have it.

2. Create a New Folder:

Click on the "+ New" button on the left side.

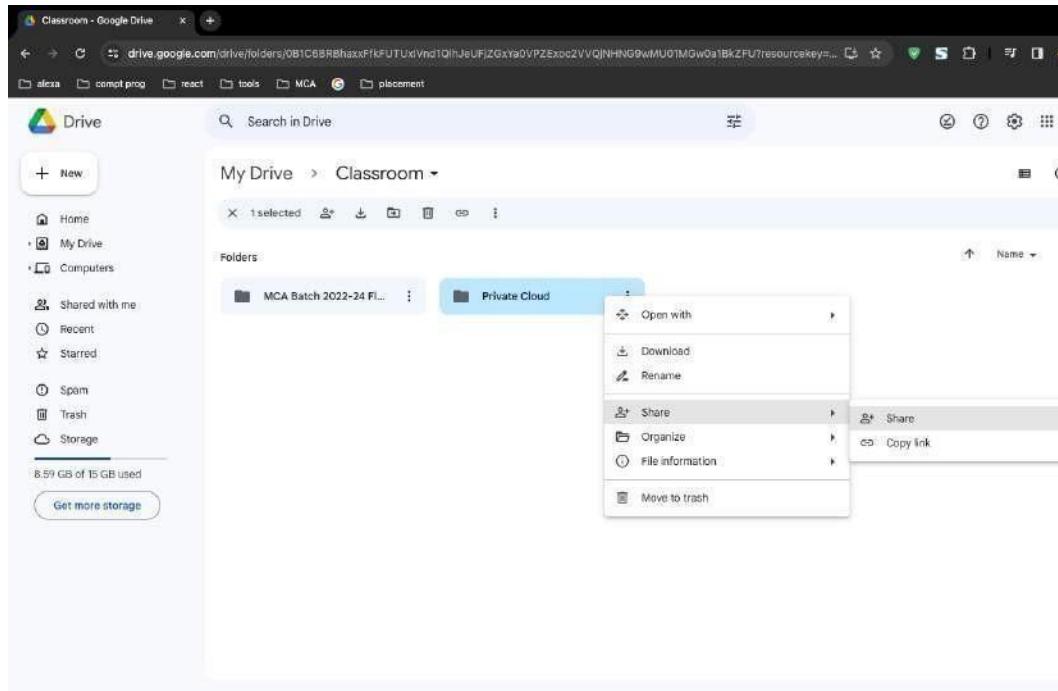
Choose "Folder" to create a new folder.

Name the folder appropriately, e.g., "Private Cloud."



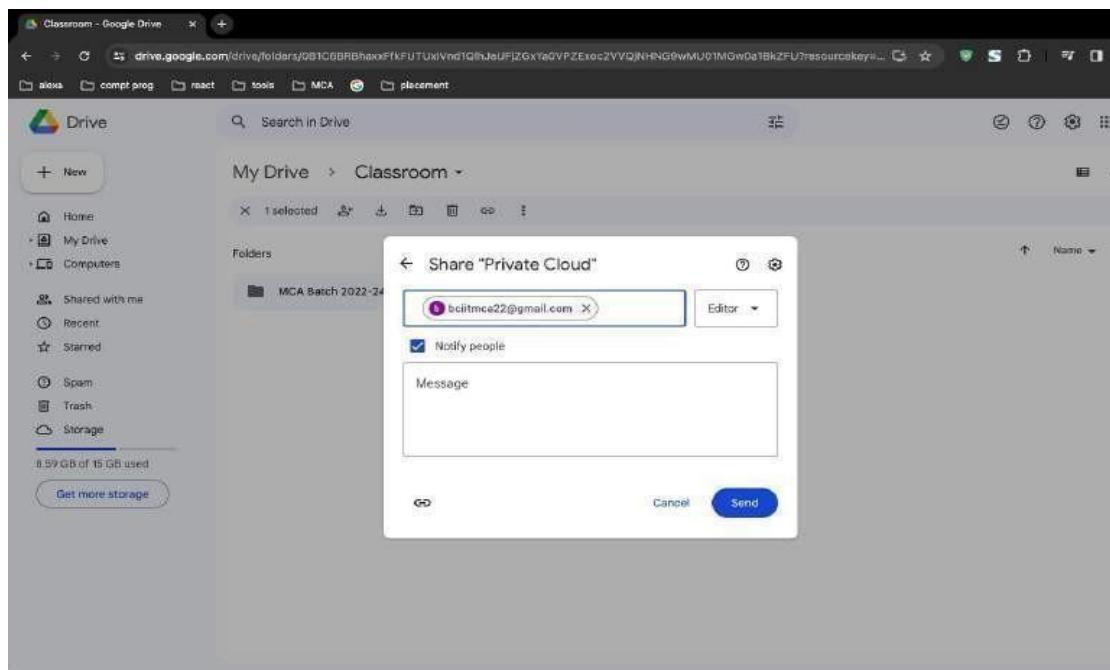
3. Share the Folder:

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



4. Add Users:

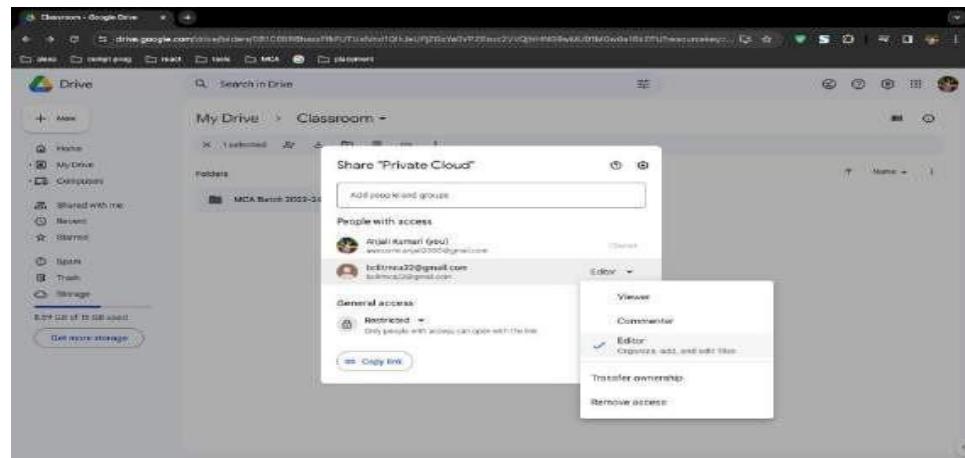
In the sharing dialog, enter the email addresses of the users you want to grant access to.
Choose the appropriate access level (e.g., Viewer, Commenter, Editor) based on the level of access you want to provide.



5. Configure Advanced Settings (Optional):

Click on "Advanced" in the sharing dialog.

Adjust settings like link sharing, preventing editors from changing access, etc.



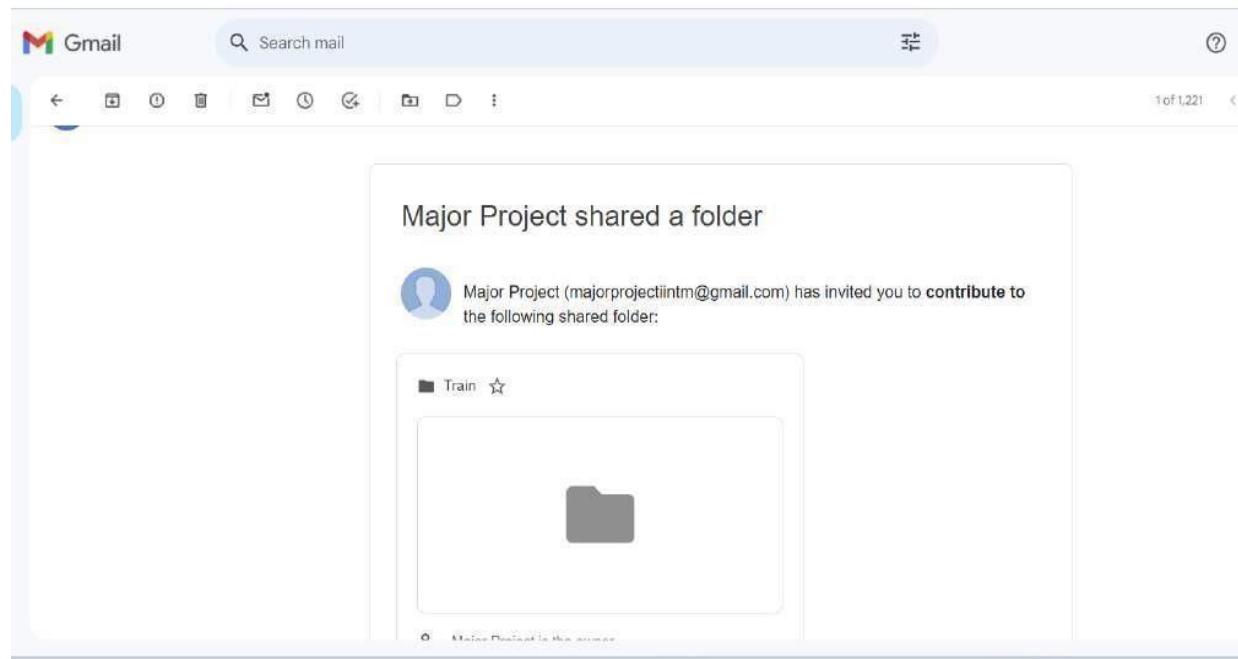
6. Send Invitations:

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:

A screenshot of the AWS IAM Policy Editor. The title bar says 'Specify permissions Info'. Below it is a text area with the heading 'Policy editor'. It shows a JSON configuration for a new policy. The JSON code is as follows:

```
1 ▼ {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "ec2:CreateVpnGateway",
8         "ec2:AttachVpnGateway",
9         "ec2:CreateVpnConnection",
10        "ec2:CreateCustomerGateway",
11        "ec2:DescribeVpnConnections",
12        "ec2:DeleteVpnConnection",
13        "ec2:DescribeVpnGateways",
14        "ec2:DeleteVpnGateway"
15      ],
16      "Resource": "*"
17    }
18  ]
19 }
```

The 'Visual' tab is selected at the top right. To the right of the JSON code, there is a sidebar with the text 'Edit statement' and 'Select an example'.

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.

The screenshot shows the 'Details' page for a newly created VPC. At the top, there's a success message: 'You successfully created vpc-042e91cd222ccf2a2 / new'. Below it, the VPC ID is listed as 'vpc-042e91cd222ccf2a2'. The 'Tenancy' is set to 'Default'. The 'State' is 'Available'. Other details include 'Network Address Usage metrics' (Disabled), 'DHCP option set' (dopt-004cd3d40f234b663), and 'IPv4 CIDR' (10.0.0.0/16). There's also an 'Actions' dropdown menu.

Details	
VPC ID	vpc-042e91cd222ccf2a2
Tenancy	Default
Default VPC	No
Network Address Usage metrics	Disabled
State	Available
DHCP option set	dopt-004cd3d40f234b663
IPv4 CIDR	10.0.0.0/16

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](#)

Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block

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.

Create virtual private gateway Info

A virtual private gateway is the VPN concentrator on the Amazon side of the site-to-site VPN connection.

Details

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

vpg_new

Value must be 256 characters or less in length.

Autonomous System Number (ASN)

Amazon default ASN
 Custom ASN

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs. Name tag helps you track your resources more easily. We recommend adding Name tag.

Key	Value - optional
<input type="text"/> Name	<input type="text"/> vpg_new

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.

VPC > [Virtual private gateways](#) > [vgw-0d0418587101aa0dd](#) > [Attach to VPC](#)

Attach to VPC Info

Details

Virtual private gateway ID
 vgw-0d0418587101aa0dd

Available VPCs
Attach the virtual private gateway to this VPC.

vpc-042e91cd222ccf2a2 / new

Cancel **Attach to VPC**

Step 4: Set Up Customer Gateway (On-premises Configuration)

Go to Customer Gateways:

In the VPC Console, go to Customer Gateways > Create Customer Gateway.

Create customer gateway Info

A customer gateway is a resource that you create in AWS that represents the customer gateway device in your on-premises network.

Details

Name tag - *optional*

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

`new_gateway`

Value must be 256 characters or less in length.

BGP ASN Info

The ASN of your customer gateway device.

`65000`

Value must be in 1 - 4294967294 range.

IP address Info

Specify the IP address for your customer gateway device's external interface.

`106.215.88.233`

Certificate ARN - *optional*

The ARN of a private certificate provisioned in AWS Certificate Manager (ACM).

`Select certificate ARN`

Device - *optional*

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.

Create VPN connection [Info](#)

Select the resources and additional configuration options that you want to use for the site-to-site VPN connection.

Details

Name tag - *optional*

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

vpn_new

Value must be 256 characters or less in length.

Target gateway type [Info](#)

- Virtual private gateway
- Transit gateway
- Not associated

Virtual private gateway

vgw-0d0418587101aa0dd



Customer gateway [Info](#)

- Existing
- New

Customer gateway ID

cgw-049df5edc40bc90a1



Routing options [Info](#)

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.

Download configuration X

Choose the sample configuration you wish to download based on your customer gateway. Please note these are samples, and will need modification to use Advanced Algorithms, Certificates, and/or IPv6.

Vendor

The manufacturer of the customer gateway device (for example, Cisco Systems, Inc.).

Barracuda



Platform

The class of the customer gateway device (for example, J-Series).

NextGen Firewall F-Series



Software

The operating system running on the customer gateway device (for example, ScreenOS).

6.2+



IKE version

The IKE version you are using for your VPN connection.

ikev1



Cancel

Download

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.

The screenshot shows the AWS VPC console interface. At the top, there's a table with columns 'Name' and 'VPN ID'. One row is selected, showing 'vpn_new' and 'vpn-032f83f37fc9694a7'. Below this, a modal window is open for the 'vpn_new' connection. The title bar says 'VPN connection vpn-032f83f37fc9694a7 / vpn_new'. Inside, the 'Tunnel state' section lists two tunnels:

Tunnel number	Outside IP address	Inside IPv4 CIDR
Tunnel 1	35.170.46.91	169.254.47.144/3
Tunnel 2	52.202.195.58	169.254.130.176/32

At the bottom of the modal, there are two links: 'Tunnel 1 options' and 'Tunnel 2 options', each followed by an 'Info' link.

```
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
```