

MINI PROJECT-I

AWS

LAB-1 [I AM Hand-on]

- Setup the MFA (multi factor authentication) for root user

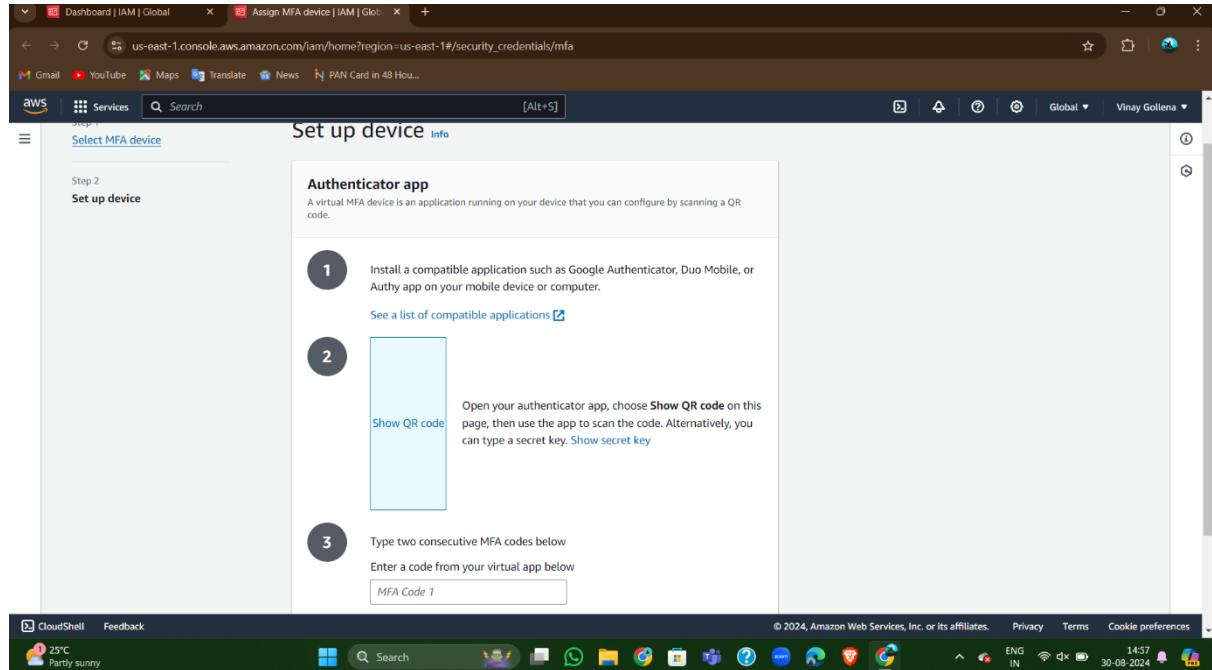
Open the root account and open the I AM service.

The screenshot shows the AWS IAM Dashboard. On the left, there's a sidebar with 'Identity and Access Management (IAM)' and sections for 'Access management' (User groups, Users, Roles, Policies), 'Identity providers', and 'Access reports'. The main area has two sections: 'Security recommendations' and 'IAM resources'. Under 'Security recommendations', there are two items: 'Add MFA for root user' (with a warning icon) and 'Root user has no active access keys' (with a green checkmark). Under 'IAM resources', there are tables for User groups (1), Users (2), Roles (13), Policies (1), and Identity providers (0). To the right, there's a sidebar for 'AWS Account' showing the Account ID (851725625974), Account Alias (vinay-gollena), and Sign-in URL (https://vinay-gollena.signin.aws.amazon.com/console). Below that is a 'Quick Links' section for 'My security credentials'.

Click on the Add MFA for root user

The screenshot shows the 'Assign MFA device' wizard. It's step 1, titled 'Device options'. It says, 'In addition to username and password, you will use this device to authenticate into your account.' There are three options: 'Passkey or security key' (using fingerprint, face, or screen lock), 'Authenticator app' (using a code generated by an app like Google Authenticator), and 'Hardware TOTP token' (using a physical hardware device like a Yubikey). The 'Authenticator app' option is selected. At the bottom are 'Cancel' and 'Next' buttons.

Download the authentication app in your mobile device and scan the QR code by clicking show QR code



Setup the MFA and observe that Root user has MFA.

A screenshot of the AWS IAM Dashboard. The URL is us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/home. The dashboard shows security recommendations and IAM resources. Under 'Security recommendations', it says 'Root user has MFA' and 'Root user has no active access keys'. Under 'IAM resources', it shows 1 User group, 2 Users, 13 Roles, 1 Policy, and 0 Identity providers. On the right side, there is a sidebar titled 'AWS Account' with information about the account ID (851725625974) and sign-in URL (https://vinay-gollena.signin.aws.amazon.com/console). There is also a 'Quick Links' section with a link to 'My security credentials'. The browser's address bar shows 'us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/home'. The bottom of the screen shows a Windows taskbar with various icons.

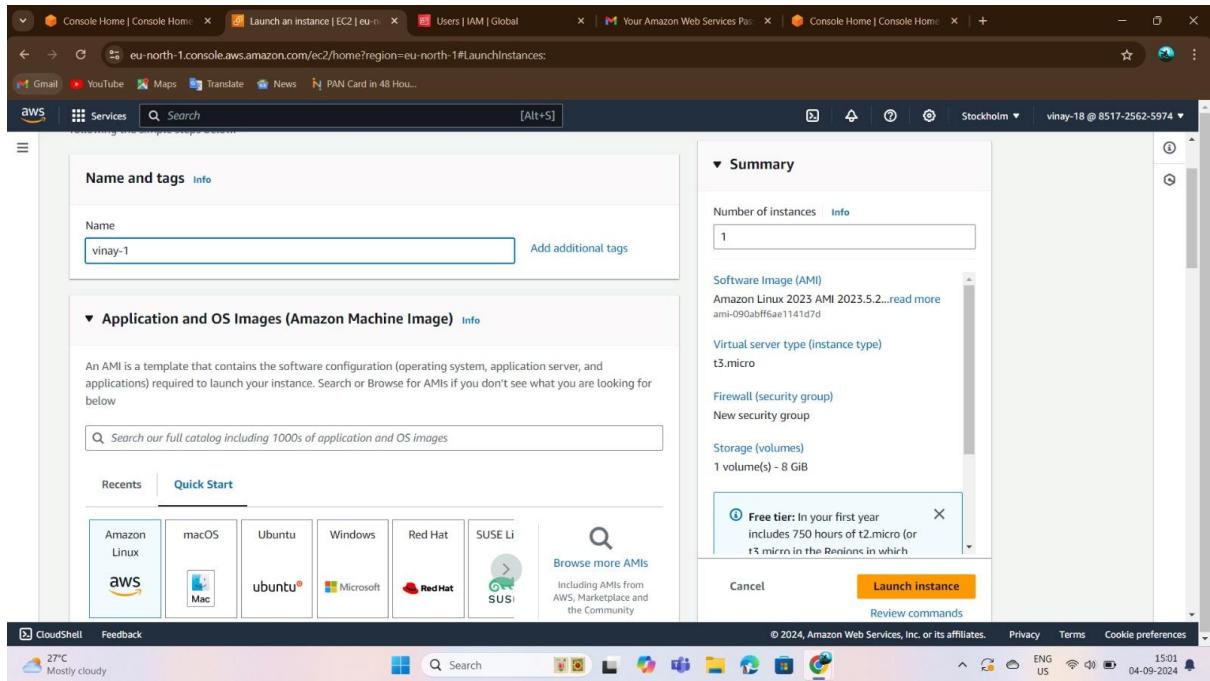
➤ Creating a new user with console access and give the only EC2 full permissions.

The screenshot shows the AWS Identity and Access Management (IAM) console. The left sidebar is titled 'Identity and Access Management (IAM)' and includes options like 'Dashboard', 'User groups', 'Users', 'Roles', 'Policies', 'Identity providers', and 'Account settings'. Under 'Access management', there are sections for 'User groups', 'Users', 'Roles', 'Policies', 'Identity providers', and 'Account settings'. Below these are sections for 'Access reports', 'Access Analyzer', 'External access', 'Unused access', and 'Analyzer settings'. A 'Credential report' section is also present. The main content area is titled 'Permissions policies (1/1)' and shows a table with one row. The row contains a checkbox, the policy name 'AmazonEC2FullAccess', its type 'AWS managed', and the attachment method 'Directly'. There are buttons for 'Search IAM', 'Add permissions', and 'Remove'. Below the table are sections for 'Permissions boundary (not set)' and 'Generate policy based on CloudTrail events', which includes a 'Generate policy' button and a note stating 'No requests to generate a policy in the past 7 days.'

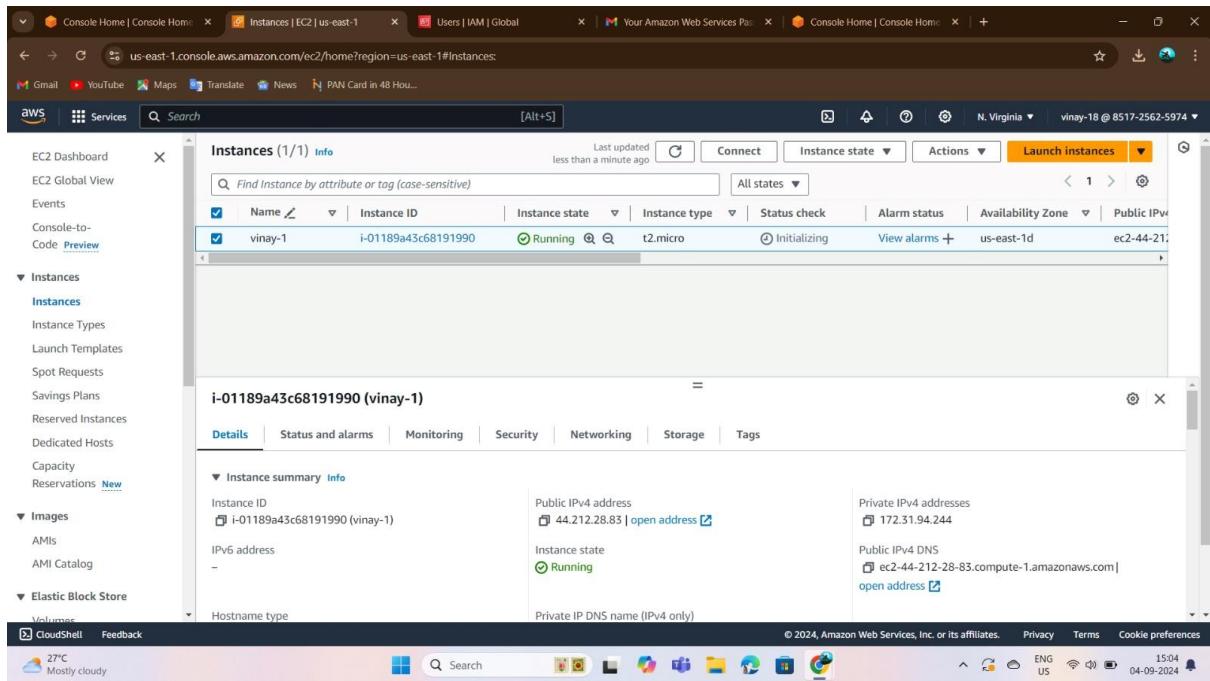
➤ Open the IAM account and check the instances are running or not.

The screenshot shows the AWS EC2 Instances page. The left sidebar includes 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Instances' (selected), 'Images', 'Elastic Block Store', and 'CloudShell'. Under 'Instances', there are sub-options for 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity', and 'Reservations'. The main content area is titled 'Instances Info' and shows a table with columns for 'Name', 'Instance ID', 'Instance state', 'Status check', 'Alarm status', and 'Availability Zone'. A search bar at the top allows filtering by 'Find Instance by attribute or tag (case-sensitive)'. A note at the bottom states 'No instances' and 'You do not have any instances in this region'. A 'Launch instances' button is available. A modal window titled 'Select an instance' is open. The bottom of the screen shows the Windows taskbar with various pinned icons and system status indicators.

➤ Launch the instance in the IAM account.



➤ The instance is running successfully by providing the EC2 full access now.



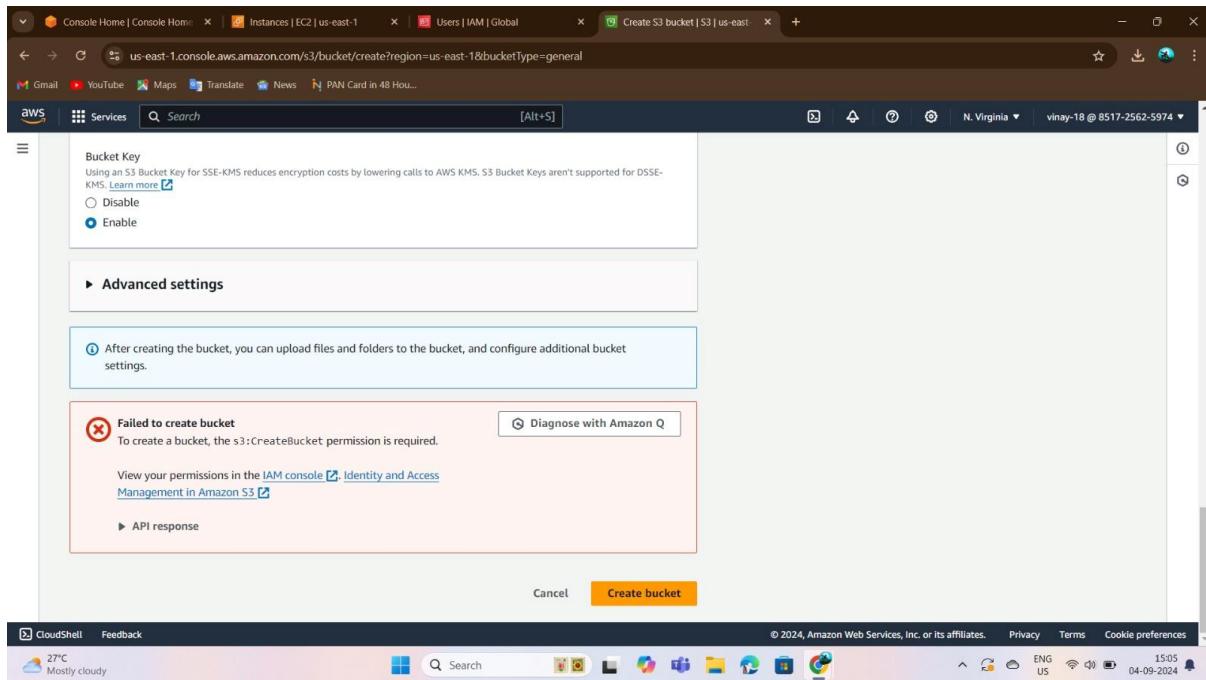
➤ Now checking the other services with this user account can work or not.

Now I am trying to create the S3 bucket by using the user with EC2 full access permissions.

The screenshot shows the Amazon S3 landing page. At the top right, there is a call-to-action box titled "Create a bucket". It contains the text: "Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored." Below this is a large orange "Create bucket" button. The main content area features the heading "Amazon S3" and the sub-headline "Store and retrieve any amount of data from anywhere". A brief description follows: "Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance." Below this, there is a section titled "How it works" with a small thumbnail image showing the AWS logo and the text "Introduction to Amazon S3".

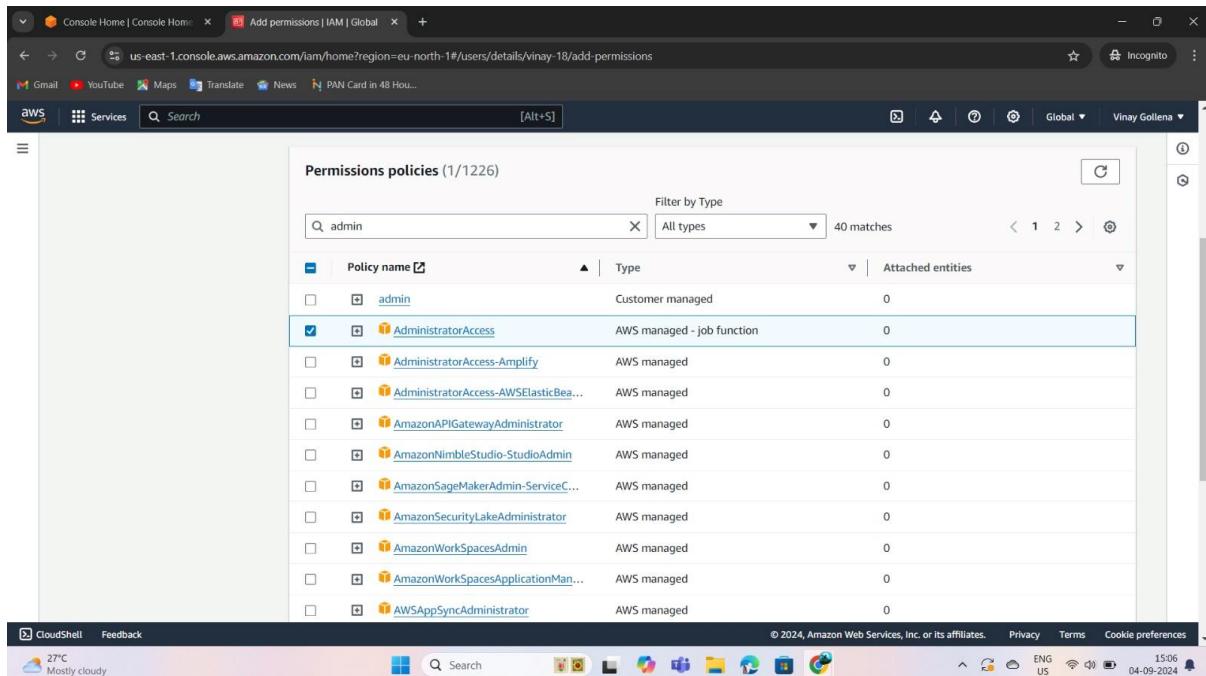
Click on the create bucket and provide a unique name for the bucket.

The screenshot shows the "Create S3 bucket" wizard. The first step, "Bucket Type", is displayed. It includes two options: "General purpose buckets" (selected) and "Low-latency use cases". The "General purpose buckets" section states: "Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones." The "Low-latency use cases" section states: "Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone." Below this, there is a "Bucket name" input field containing "vinay-18". A note below the input field says: "Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming." There is also a "Choose bucket" button and a note: "Copy settings from existing bucket - optional Only the bucket settings in the following configuration are copied."



In the above image it shows that “Failed to create bucket” because I have provided only the EC2fullaccess permissions only, So other services will not work.

➤ Now provide the Administrative permissions.



➤ Now try to create any service
I am again creating the S3 bucket

The screenshot shows the AWS S3 'Create S3 bucket' wizard. In the 'General configuration' tab, the 'Bucket type' is set to 'General purpose'. The bucket name is 'vinay-18'. In the 'Object Ownership' tab, it states: 'Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.'

General configuration

AWS Region: US East (N. Virginia) us-east-1

Bucket type: **General purpose** (Info)

Bucket name: **vinay-18** (Info)

Copy settings from existing bucket - optional

Object Ownership (Info)

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

SUCCESSFUL BUCKET CREATION

Successfully created bucket "vinay-18". To upload files and folders, or to configure additional bucket settings, choose View details.

Buckets Overview

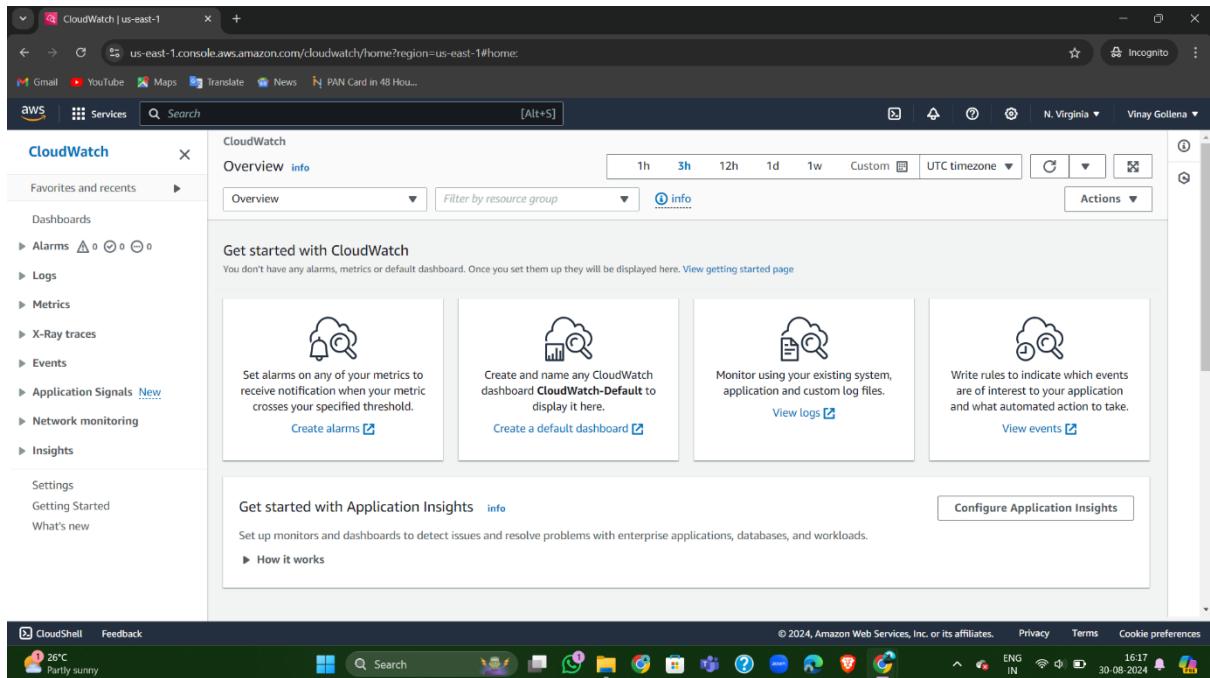
General purpose buckets (3) (Info) All AWS Regions

Name	AWS Region	IAM Access Analyzer	Creation date
cf-templates-4opwclb4wydv-us-east-1	US East (N. Virginia) us-east-1	View analyzer for us-east-1	June 16, 2024, 19:07:03 (UTC+05:30)
elasticbeanstalk-us-east-1-851725625974	US East (N. Virginia) us-east-1	View analyzer for us-east-1	June 16, 2024, 22:13:48 (UTC+05:30)
vinay-18	US East (N. Virginia) us-east-1	View analyzer for us-east-1	September 4, 2024, 15:07:45 (UTC+05:30)

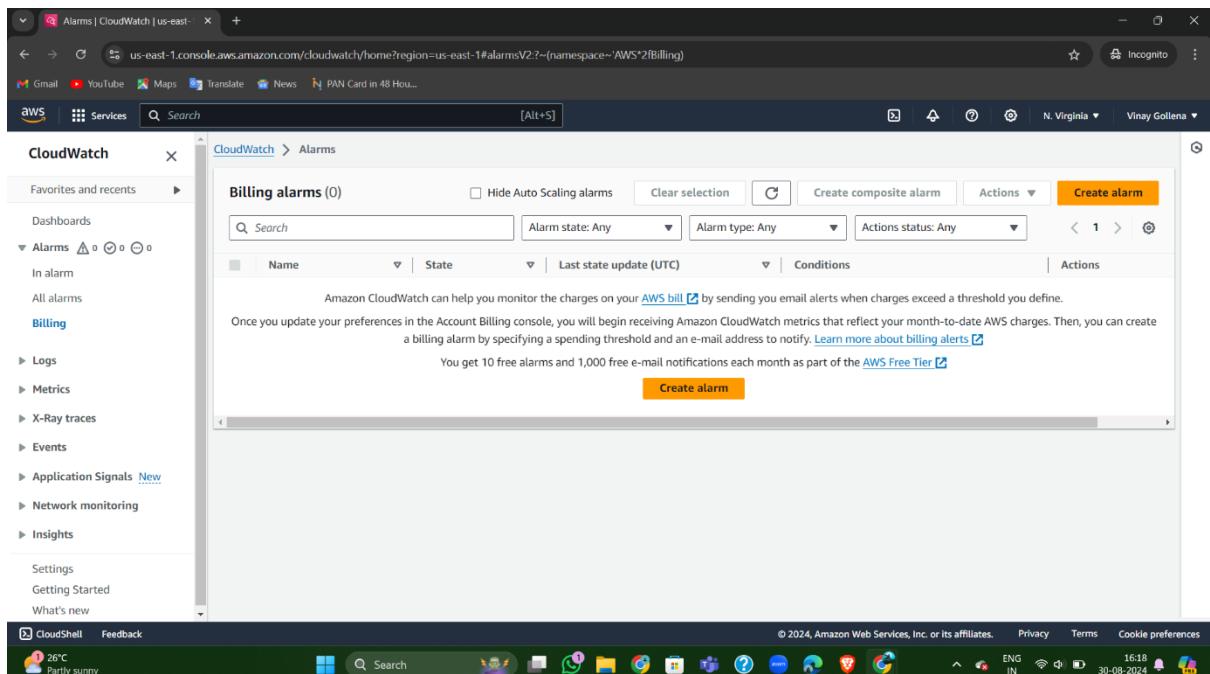
- Now S3 bucket has been created after providing the Admin access or else if we want only S3 bucket access we can create by providing the only S3bucketfullaccess permission.

LAB-2[Billing Alarm]

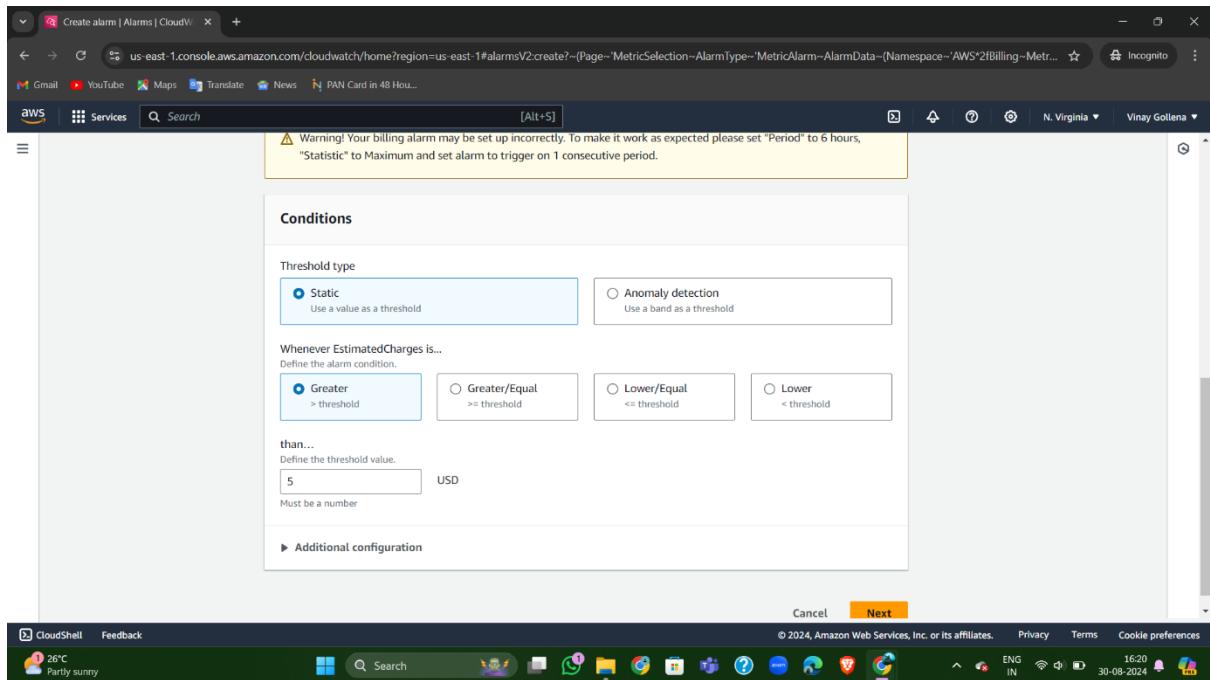
- Open cloud watch service in the Aws console



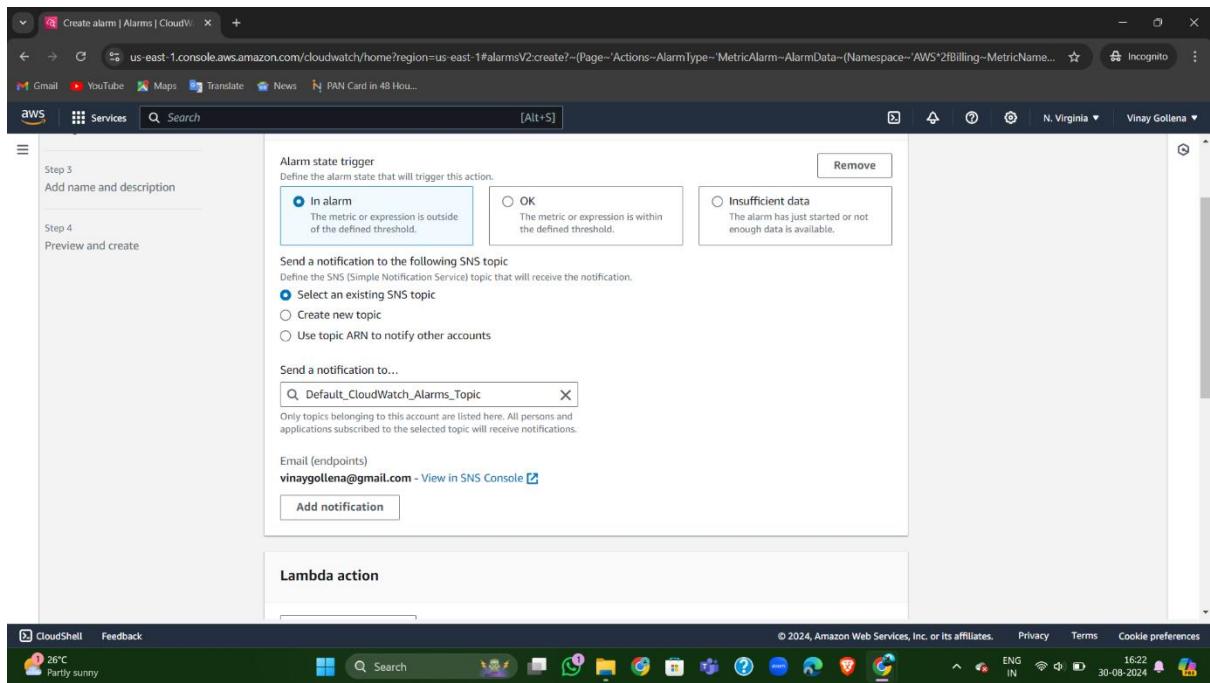
- Click on the Alarms and click on billing it provide the below page



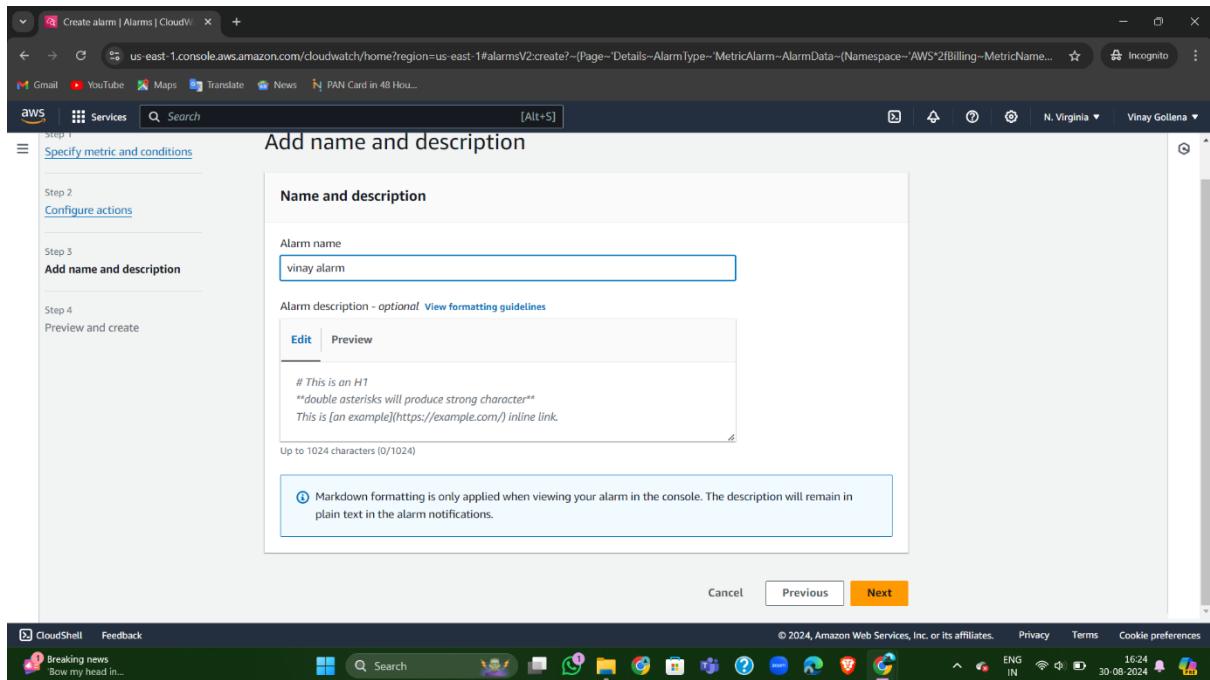
- Create and Alarm by using create alarm and select the conditons.
And click the next.



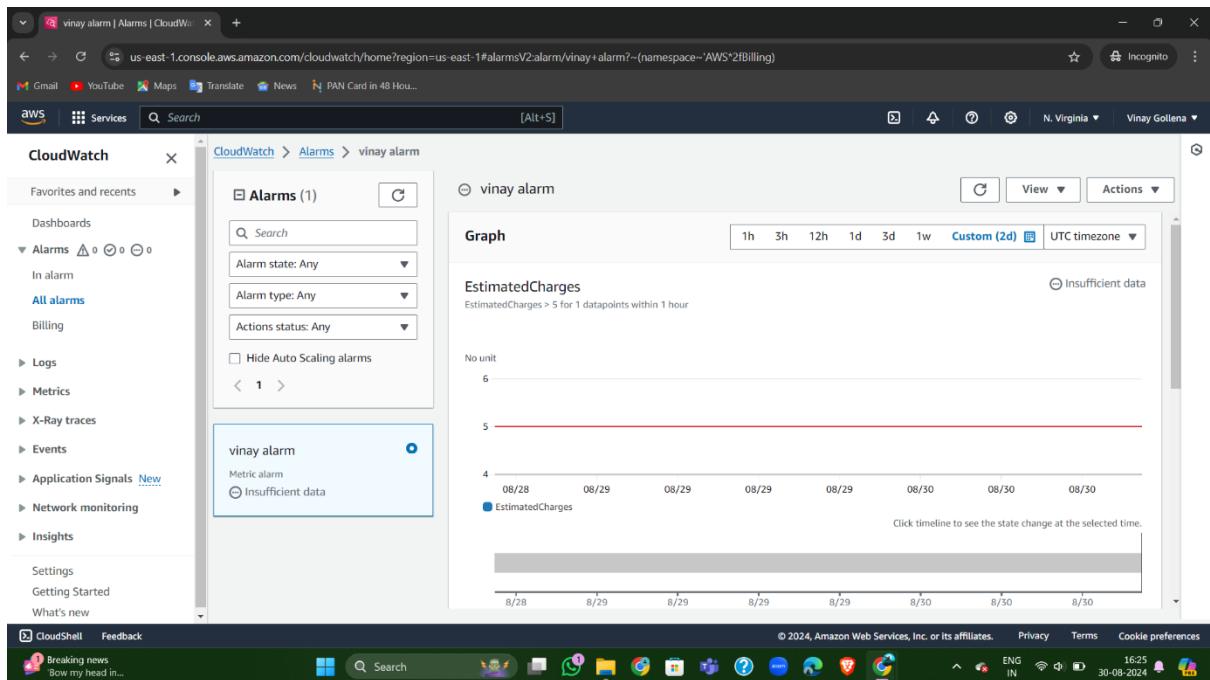
➤ Than select the SNS(simple notification service) by providing the Email,because when the threshold billing will reach the limit than it sends an email to the respective mail Id.



➤ Next give the Alarm name and click on next.



- Alarm has been created and whenever our billing threshold has reached near to the given range than we get a notification by SNS to the Mail.



LAB-3[S3 bucket]

Step-1:

- Create the bucket and give the unique name

The screenshot shows the 'Create bucket' wizard on the AWS S3 service. The 'General configuration' step is selected. The 'AWS Region' dropdown is set to 'Europe (Stockholm) eu-north-1'. Under 'Bucket type', the 'General purpose' option is selected, with a note explaining it's recommended for most use cases and access patterns. The 'Bucket name' field contains 'vinay-7'. A note below states: 'Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming.' There is also a 'Copy settings from existing bucket - optional' section with a 'Choose bucket' button.

- Bucket has been created.

The screenshot shows the 'Buckets' page on the AWS S3 service. A green banner at the top says 'Successfully created bucket "vinay-7"'. Below it, there is an 'Account snapshot' section and a 'View Storage Lens dashboard' button. The main area shows a list of 'General purpose buckets'. One bucket, 'vinay-7', is highlighted. The table includes columns for Name, AWS Region, IAM Access Analyzer, and Creation date. The 'vinay-7' bucket was created on August 30, 2024, at 16:15:37 UTC+05:30. The status bar at the bottom shows the date as 30-08-2024 and the time as 16:34.

Name	AWS Region	IAM Access Analyzer	Creation date
cf-templates-4opwclbwydv-us-east-1	US East (N. Virginia) us-east-1	View analyzer for us-east-1	June 16, 2024, 19:07:03 (UTC+05:30)
elasticbeanstalk-us-east-1-851725625974	US East (N. Virginia) us-east-1	View analyzer for us-east-1	June 16, 2024, 22:13:48 (UTC+05:30)
vin22	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	August 30, 2024, 16:15:37 (UTC+05:30)
vinay-7	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	August 30, 2024, 16:35:21 (UTC+05:30)

- Upload the file in the bucket.

The screenshot shows the AWS S3 console interface for uploading objects. The URL in the address bar is `eu-north-1.console.aws.amazon.com/s3/upload/vinay-7?region=eu-north-1&bucketType=general`. The navigation path is `Amazon S3 > Buckets > vinay-7 > Upload`. The main area is titled "Upload" with an "Info" link. A large box allows users to "Drag and drop files and folders you want to upload here, or choose Add files or Add folder." Below this, a table lists "Files and folders (1 Total, 59.0 KB)". The table has columns for Name, Folder, and Type. One item, "shiva1.jpg", is listed with a type of "image/jpeg". At the bottom, the "Destination" is set to "s3://vinay-7". The status bar at the bottom shows "CloudShell Feedback" and system icons.

Upload objects - S3 bucket vinay-7

eu-north-1.console.aws.amazon.com/s3/upload/vinay-7?region=eu-north-1&bucketType=general

Gmail YouTube Maps Translate News PAN Card in 48 Hou...

aws Services Search [Alt+S]

Upload succeeded
View details below.

The information below will no longer be available after you navigate away from this page.

Summary

Destination	Succeeded	Failed
s3://vinay-7	1 file, 59.0 KB (100.00%)	0 files, 0 B (0%)

Files and folders Configuration

Files and folders (1 Total, 59.0 KB)

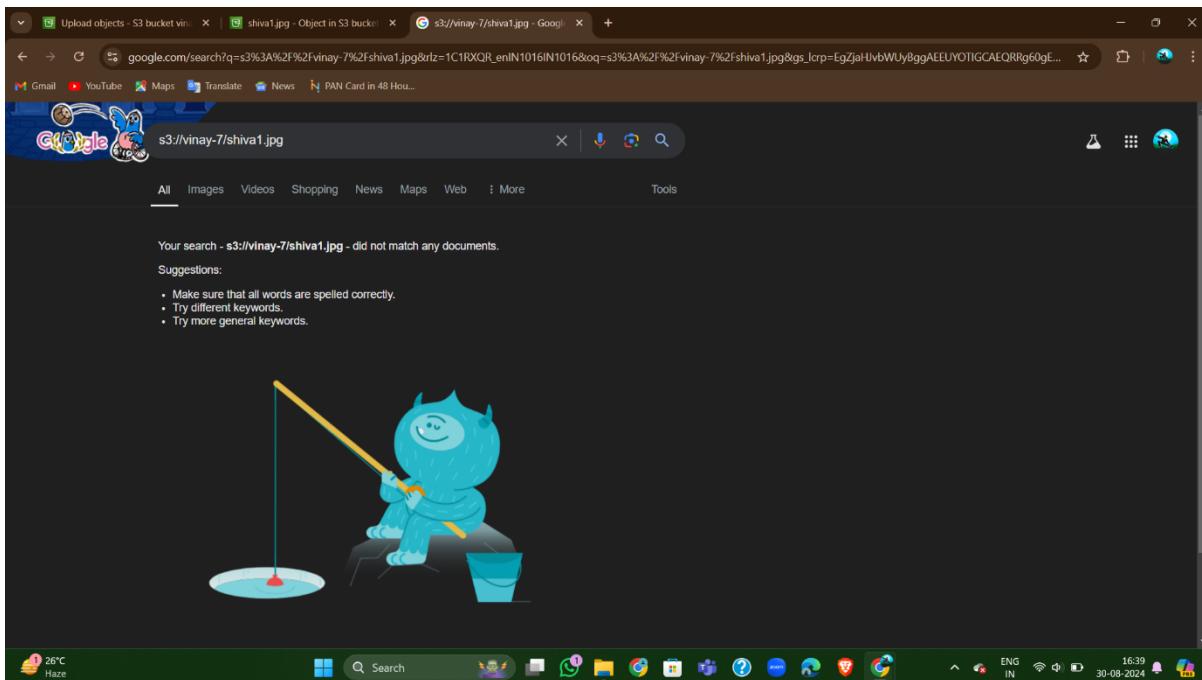
Find by name

Name	Folder	Type	Size	Status	Error
shiva1.jpg	-	image/jpeg	59.0 KB	Succeeded	-

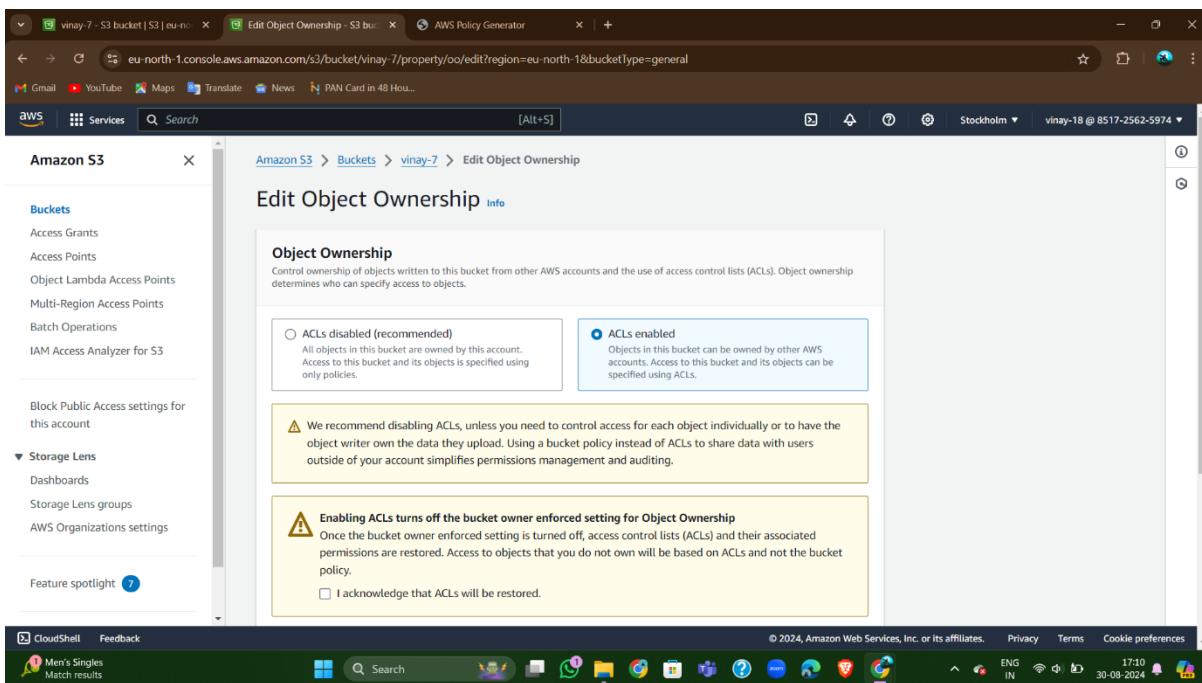
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

26°C Haze ENG IN WiFi 16:37 30-08-2024

- Try to access the file over the browser using its URL, it gave an error.



- Check the permissions of the file and the bucket.
➤ Edit the ACLs enabled



➤ Successfully edited access control list.

The screenshot shows the AWS S3 Bucket ACL editor. It displays two groups: 'Authenticated users group' and 'S3 log delivery group'. Both groups have 'List', 'Read', and 'Write' permissions checked. A yellow warning box states: 'When you grant access to the Everyone or Authenticated users group grantees, anyone in the world can access the objects in this bucket.' Below it is a link to 'Learn more' and a checkbox to 'I understand the effects of these changes on my objects and buckets.' At the bottom right of the dialog is a 'Save changes' button. The background shows the AWS navigation bar and a taskbar at the bottom.

➤ Successfully edited public access to the bucket.

The screenshot shows the 'Make public: status' page. A green banner at the top says 'Successfully edited public access'. Below it is a summary table:

Source	Successfully edited public access	Failed to edit public access
s3://vinay-7	1 object, 59.0 KB	0 objects

Below the summary is a section titled 'Failed to edit public access (0)'. A search bar and a table header are visible. The table has columns: Name, Folder, Type, Last modified, Size, and Error. The taskbar at the bottom includes icons for CloudShell, Feedback, and various system notifications.

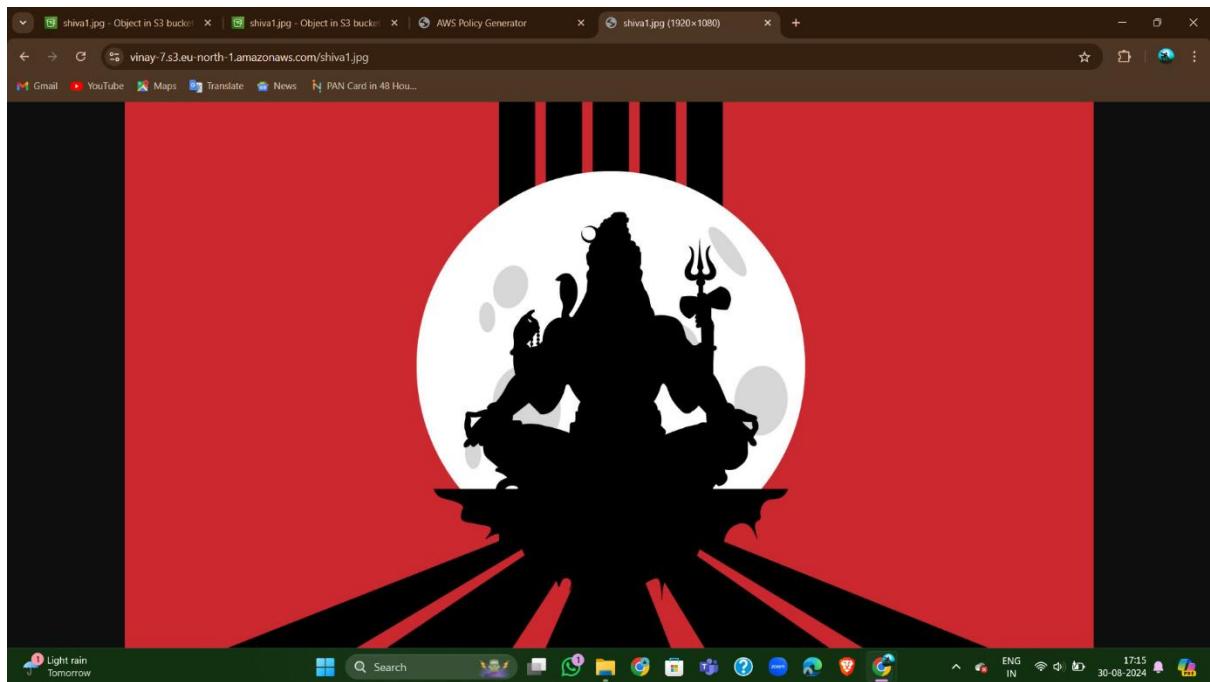
➤ Now copy the end url of the bucket and try to browse.

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with options like Buckets, Storage Lens, and Feature spotlight. The main area is titled 'shiva1.jpg - Object in S3 bucket'. It has tabs for Properties, Permissions, and Versions. Under Properties, the 'Object overview' section contains the following information:

- Owner: 487adcd6e685711e613970d469d94aded17f42b859ee97f9166f55ce00fa0961
- AWS Region: Europe (Stockholm) eu-north-1
- Last modified: August 30, 2024, 16:37:53 (UTC+05:30)
- Size: 59.0 KB
- Type: jpg
- Key: shiva1.jpg

On the right, there are links for S3 URI (s3://vinay-7/shiva1.jpg), Amazon Resource Name (ARN) (arn:aws:s3:::vinay-7/shiva1.jpg), and Entity tag (Etag) (4821ae1c607f3105a0c0556ab390c0da). A tooltip 'Object URL Copied' is shown near the S3 URI link. At the bottom, there are buttons for Copy S3 URI, Download, Open, and Object actions.

- The uploaded image can be browse and access by the url to the public from anywhere.



Step-2:

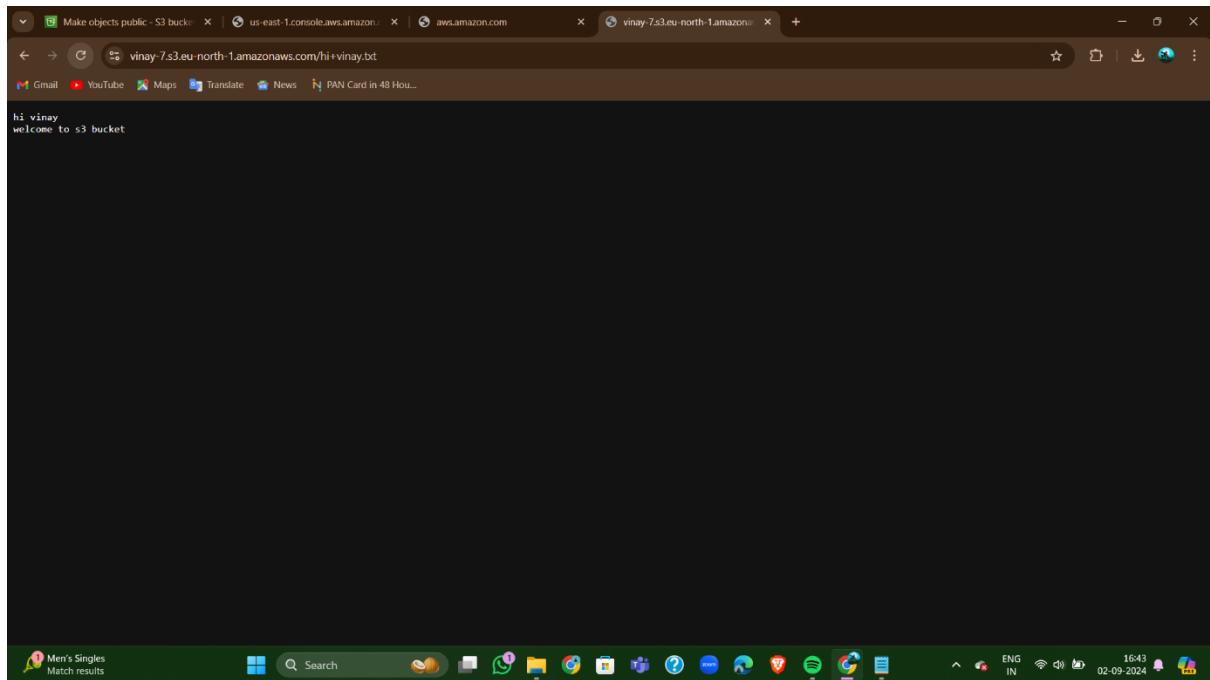
- Enable the versioning

The screenshot shows the AWS S3 console for a bucket named 'vinay-7'. A green success message at the top states: 'Successfully edited Bucket Versioning' and 'To transition, archive, or delete older object versions, configure lifecycle rules for this bucket.' Below this, the 'Bucket overview' section displays basic information: AWS Region (Europe (Stockholm) eu-north-1), Amazon Resource Name (ARN) (arn:aws:s3:::vinay-7), and Creation date (August 30, 2024, 16:35:21 (UTC+05:30)). The 'Bucket Versioning' section shows that versioning is 'Enabled'. The navigation bar includes tabs for Objects, Properties, Permissions, Metrics, Management, and Access Points.

- Create a text file and upload the file in the bucket.

The screenshot shows the AWS S3 console after a file upload. A green success message at the top says 'Upload succeeded' and 'View details below.' It includes a note: 'The information below will no longer be available after you navigate away from this page.' Below this, the 'Summary' section shows the destination as 's3://vinay-7' with a status of 'Succeeded' (1 file, 32.0 B (100.00%)) and 'Failed' (0 files, 0 B (0%)). The 'Files and folders' tab is selected, showing a table with one item: 'hi vinay.txt' (text/plain, 32.0 B, Succeeded). The navigation bar includes tabs for Files and folders and Configuration.

- After enabling the access control list and providing the public access.
- Copy the url of the file and browse.



- Update the uploaded file and reupload the file again.

A screenshot of the AWS S3 console. The top navigation bar shows multiple tabs like 'Upload objects - S3 bucket vinay-' and 'Object in S3 bucket'. The main content area displays a green success message: 'Upload succeeded' with a link to 'View details below.' Below this, there's a summary table:

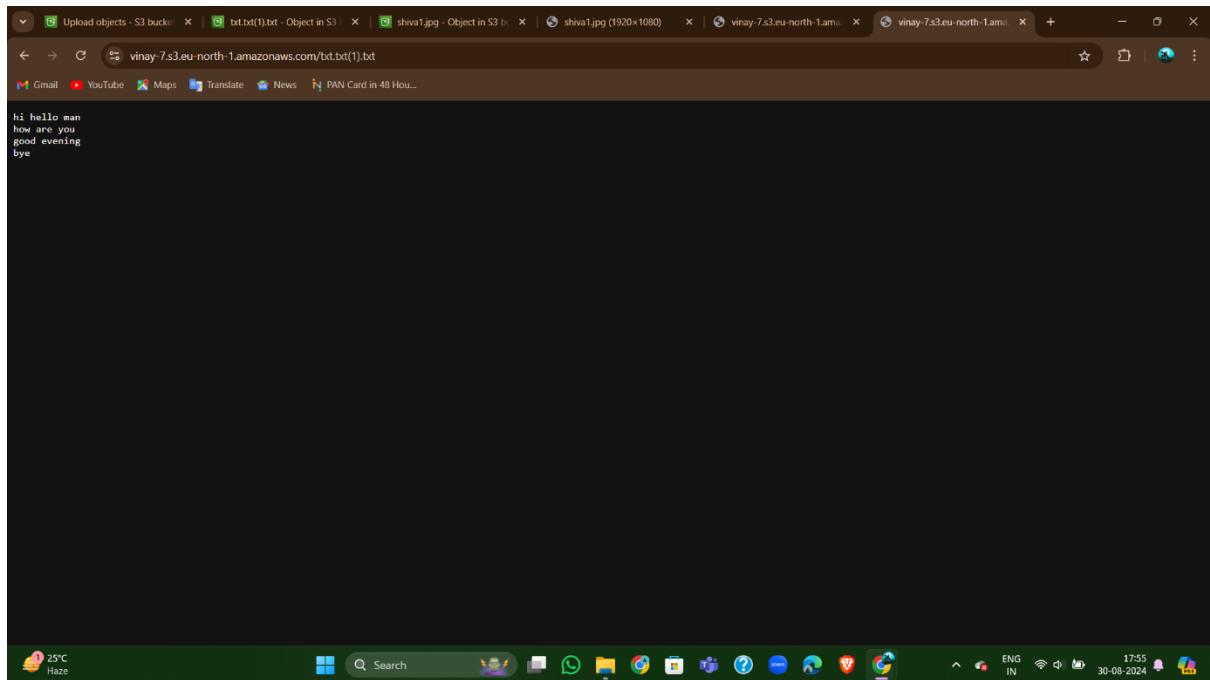
Destination	Succeeded	Failed
s3://vinay-7	1 file, 45.0 B (100.00%)	0 files, 0 B (0%)

Underneath, there are tabs for 'Files and folders' and 'Configuration'. The 'Files and folders' tab shows a table with one item:

Files and folders (1 Total, 45.0 B)						
Name	Folder	Type	Size	Status	Error	Actions
txt.txt(1).txt	-	text/plain	45.0 B	Succeeded	-	

The bottom of the screen shows the AWS navigation bar with links for CloudShell, Feedback, and various services like Lambda, CloudWatch, and CloudFront, along with system status icons.

- Again edit the access control list and provide public access.
- Then browse the url.



- Here we can observe that while we are uploading the updated text file after uploaded the old file will replace the new file and shows the new content with old one.
- The old file will not be display until on the show versions

Name	Type	Last modified	Size	Storage class
shiva1.jpg	jpg	August 30, 2024, 16:37:33 (UTC+05:30)	59.0 KB	Standard
txt.txt(1)	txt	August 30, 2024, 17:47:31 (UTC+05:30)	28.0 B	Standard

- After clicking the show versions than we are able to see the old file as well as updated file.

Objects (3) Info

Name	Type	Version ID	Last modified	Size	Storage class
shiva1.jpg	jpg	null	August 30, 2024, 16:37:33 (UTC+05:30)	59.0 KB	Standard
txt.txt(1).txt	txt	DRFLtO.Mop MILkfBa6USa MfbLPRkakw	August 30, 2024, 17:51:12 (UTC+05:30)	45.0 B	Standard
txt.txt(1).txt	txt	ufopRF8n9zK R3Vt8rOobdJ dthX_qvF1	August 30, 2024, 17:43:31 (UTC+05:30)	28.0 B	Standard

Step-3:

- Upload the file.
- Delete the text file from the bucket.

Upload succeeded

The information below will no longer be available after you navigate away from this page.

Summary

Destination	Succeeded	Failed
s3://vinay-7	1 file, 32.0 B (100.0%)	0 files, 0 B (0%)

Files and folders

Name	Folder	Type	Size	Status	Error
hi vinay.txt	-	text/plain	32.0 B	Succeeded	-

➤ Delete the text file from the file.

The screenshot shows the AWS S3 console interface. A green banner at the top indicates "Successfully deleted objects". Below it, a summary table shows one object successfully deleted from the source "s3://vinay-7". The "Failed to delete" tab shows zero failed objects. The Windows taskbar at the bottom includes icons for CloudShell, Feedback, Search, and various applications like File Explorer, Google Chrome, and Microsoft Edge.

The screenshot shows the AWS S3 console with a warning message in a yellow box: "If a folder is selected for deletion, all objects in the folder will be deleted, and any new objects added while the delete action is in progress might also be deleted. If an object is selected for deletion, any new objects with the same name that are uploaded before the delete action is completed will also be deleted." Below this, a "Specified objects" table lists a single file named "hi vinay.txt". A "Permanently delete objects?" dialog box is open, asking for confirmation to type "permanently delete" into a text input field. The Windows taskbar at the bottom is visible.

Delete the file permanently

- On the show versions and observe the deleted file.

The screenshot shows the AWS S3 console with the 'Objects' tab selected. A delete marker named 'hi vinay.txt' is listed, showing its version ID and last modified date. Other files like 'hi vinay.txt' and 'vinay.c' are also present.

Name	Type	Version ID	Last modified	Size	Storage class
hi vinay.txt	Delete marker	gRM3NdSRq BMUViwZU_3Vu4TzLRoyC tD	September 2, 2024, 16:43:42 (UTC+05:30)	0 B	-
hi vinay.txt	txt	IC5jCy4Pu8jI qgnWvnVrd xg3NIvVOQIKG	September 2, 2024, 16:41:49 (UTC+05:30)	32.0 B	Standard
vinay.c	c	Cb_9l2Ttj4kd2rO9YRbxuOh h35yNiyg18	September 2, 2024, 16:18:57 (UTC+05:30)	20.0 B	Standard

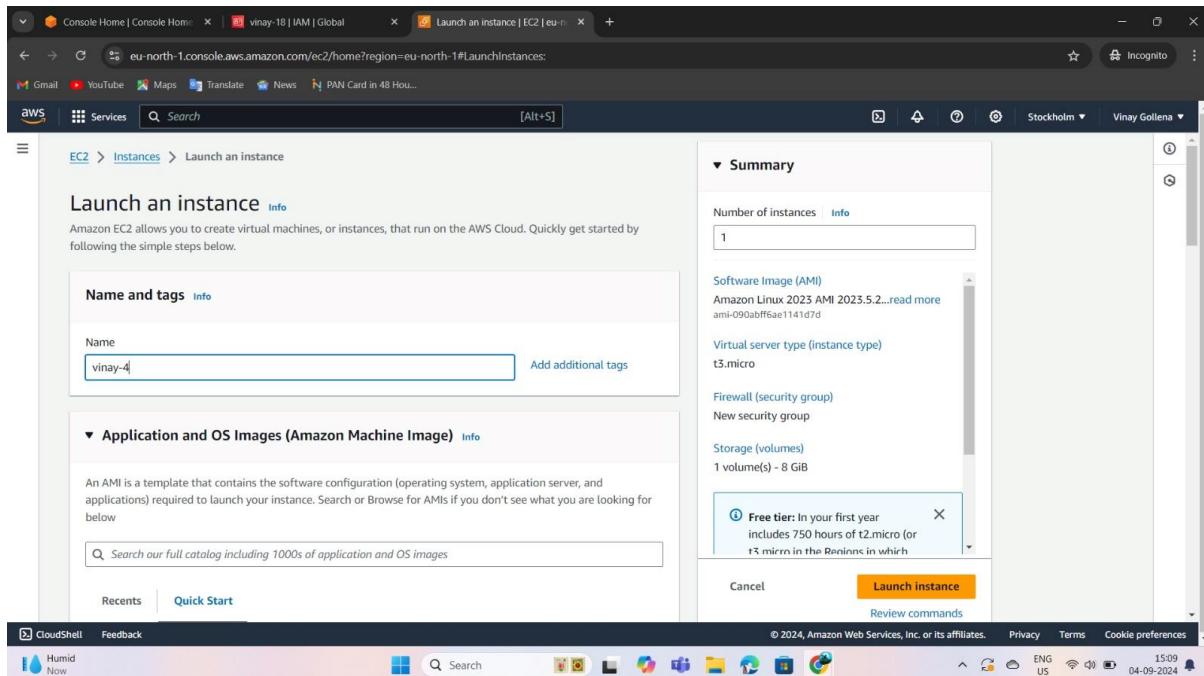
- The deleted file has been recovered from the versioning.
- If we do not enable the versioning than if we delete the file than it will be deleted permanently and no backup for that.

The screenshot shows the AWS S3 console with the 'Objects' tab selected. The file 'hi vinay.txt' is no longer listed in the table, indicating it has been permanently deleted without versioning.

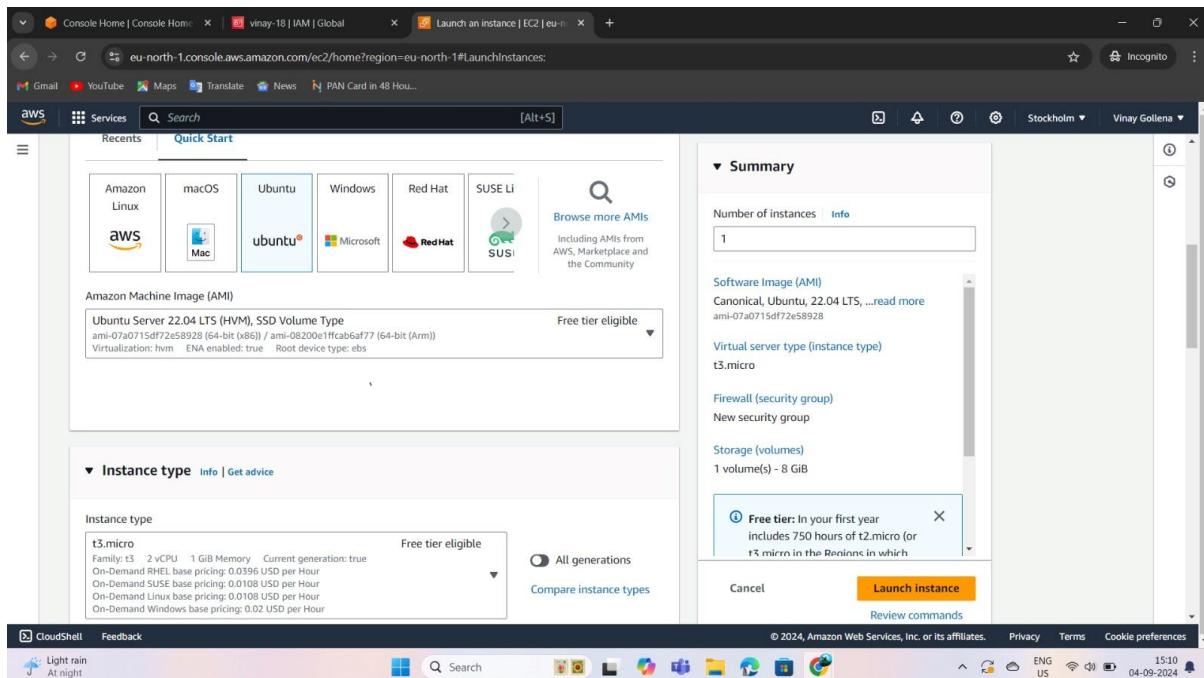
Name	Type	Version ID	Last modified	Size	Storage class
hi vinay.txt	txt	IC5jCy4Pu8jIqgnWvnVrd xg3NIvVOQIKG	September 2, 2024, 16:41:49 (UTC+05:30)	32.0 B	Standard
vinay.c	c	Cb_9l2Ttj4kd2rO9YRbxuOh h35yNiyg18	September 2, 2024, 16:18:57 (UTC+05:30)	20.0 B	Standard

LAB-4[EC2 instance]

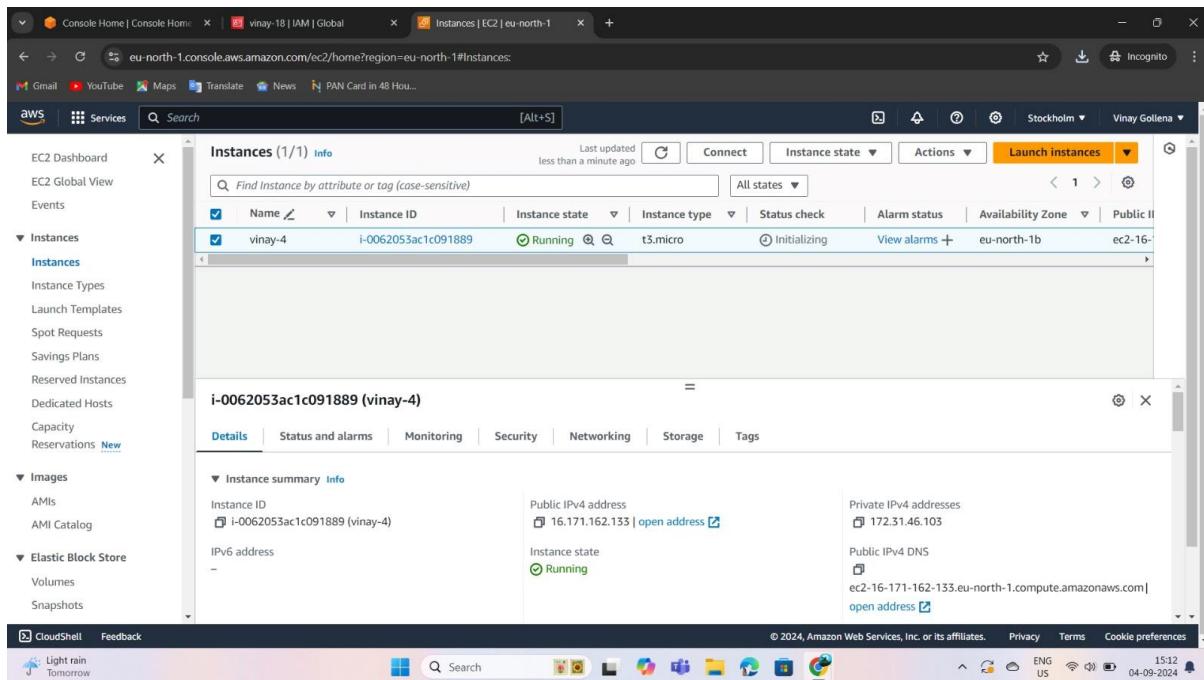
- Launching the instance and connecting it with the using putty.
- Launch the instance with t2.micro or in other regions t3.micro



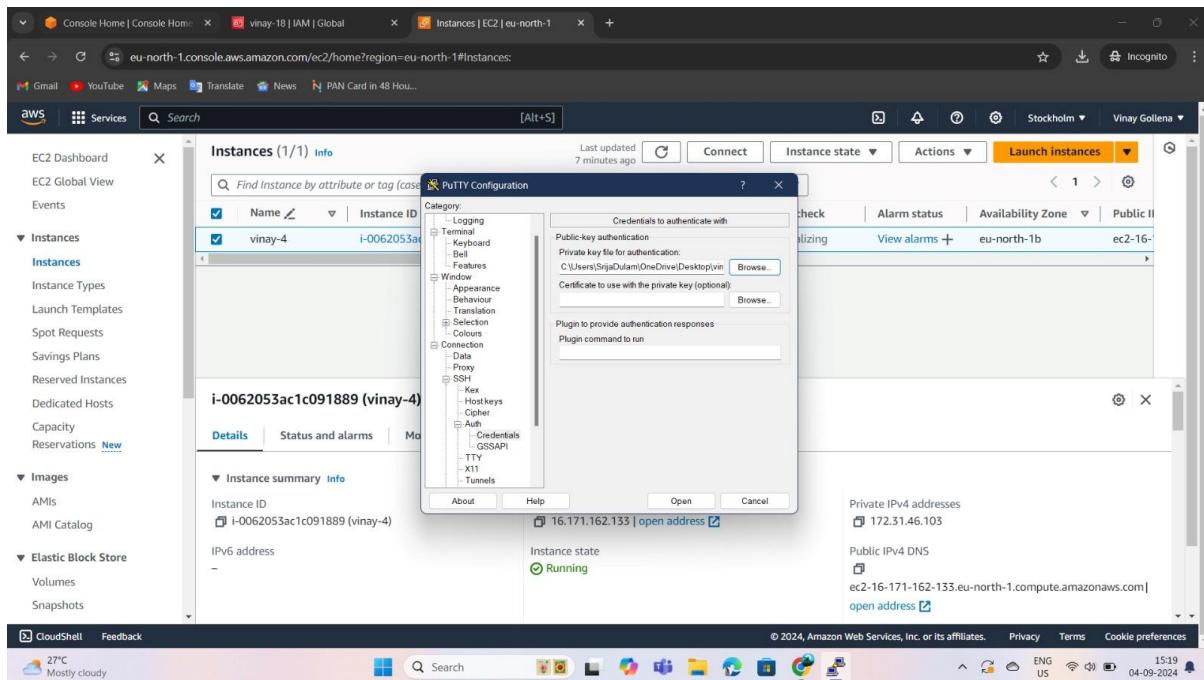
- Select the ubuntu and use the ubuntu server 22.04 version which is in free tier eligible.



- Then lunch the instance.

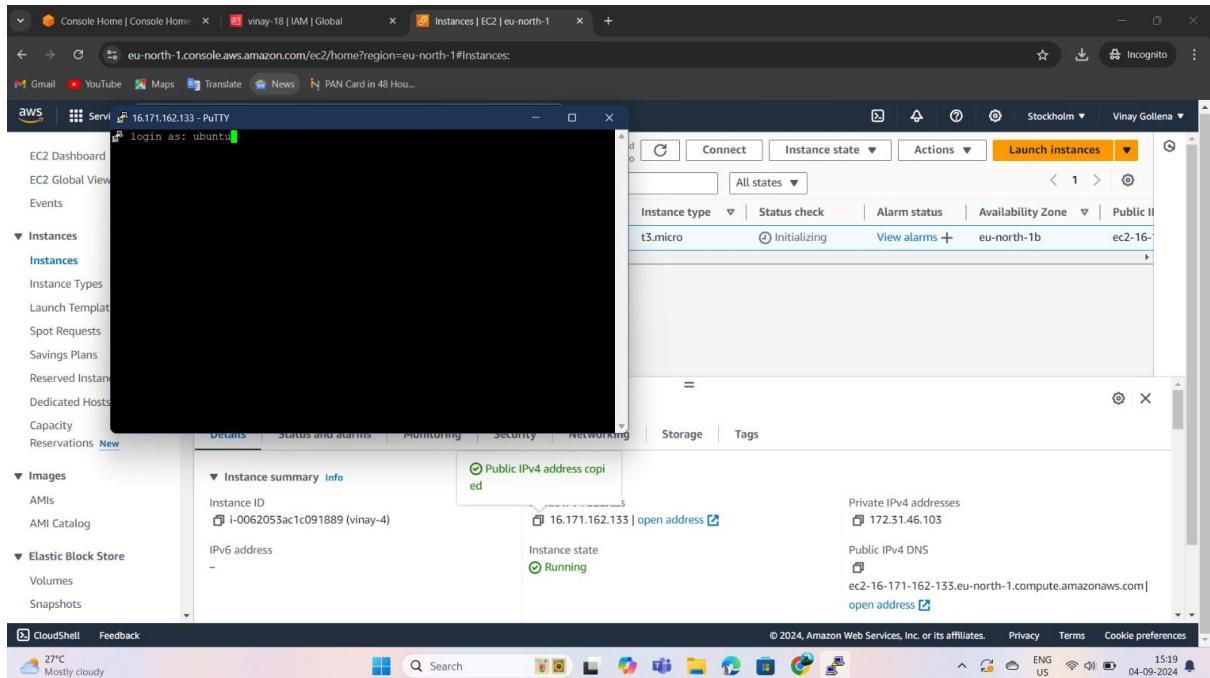


- Download the putty software by using google and install in our systems.
- Than convert the pem file to the ppk file (putty private key) key pair, by using the putty generater.



Here the pem file is converted to the ppk file by using the putty gen.

➤ Then access the EC2 instance by using the putty



➤ Connect the terminal by using ssh command

A screenshot of a Windows terminal window titled 'ubuntu@ip-172-31-46-103:~'. The session starts with a successful SSH login as 'ubuntu'. The user then runs several commands to check system status: 'top' (showing 102 processes), 'free -m' (memory usage), 'df -h' (disk space), and 'apt update' (which indicates no updates are available). The terminal window has a standard Windows taskbar at the bottom.

LAB-5[Security Group]

- Launch instance and wait for running the instance.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, and Reservations. The main content area displays a table of instances. One instance, 'vinay-4' (ID: i-0062053ac1c091889), is listed as 'Running'. Below the table, a detailed view for 'i-0062053ac1c091889 (vinay-4)' is shown, including tabs for Details, Status and alarms, Monitoring, Security (selected), Networking, Storage, and Tags. Under the Security tab, it shows the IAM Role (none) and Owner ID (851725625974). The Security groups section lists 'sg-07ed2919d66e26bb5 (launch-wizard-8)'. At the bottom, there's an 'Inbound rules' section which is currently collapsed.

- Create the security group and name it.

The screenshot shows the 'Create security group' wizard. The top navigation bar includes links for Console Home, IAM, Instances, and CreateSecurityGroup. The main content area has a breadcrumb trail: EC2 > Security Groups > Create security group. The title is 'Create security group' with an 'Info' link. A descriptive text states: 'A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the following steps.' Below this, there's a 'Basic details' section with three fields: 'Security group name' (containing 'vinaynew-SG'), 'Description' (containing 'attach this new sg to your existing ec2 instance'), and 'VPC' (with an 'Info' link).

- Edit the inbound rules or provide the port 80 and 22 from our IP address by selecting a range IP/28.
- So that any changes in your dynamic IP will not impact the rules.

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
-	SSH	TCP	22	Custom	16.171.162.133/28 16.171.162.128/28
-	HTTP	TCP	80	Custom	16.171.162.133/28 16.171.162.128/28

Add rule Cancel Preview changes Save rules

- Then save the rules.
- Attach this to the Instance and access the server.

```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SrijaDulam>cd downloads
C:\Users\SrijaDulam\Downloads>ssh -i "vinay-22.pem" ubuntu@ec2-16-171-162-133.eu-north-1.compute.amazonaws.com

the AMI owner has changed the default AMI username.
```

- While accessing the server it is not connecting the server .

➤ Than SSH port 22 with anywhere IPV4 .

The screenshot shows the 'Inbound rules' section of a security group configuration. It lists three rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-096c86aa7e7ab9116	SSH	TCP	22	Custom	16.171.162.128/28
sgr-0a8693ca95f9685f6	HTTP	TCP	80	Custom	16.171.162.128/28
-	SSH	TCP	22	Anyw...	0.0.0.0/0

A warning message at the bottom states: "Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." There are 'Save rules' and 'Cancel' buttons at the bottom right.

➤ Attach it to the instance

The screenshot shows the details of a security group named 'vinaynew-SG'. It includes the VPC ID 'vpc-07d6f4cbfb185adff'. The 'Inbound rules' tab is selected, showing three rules:

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-096c86aa7e7ab9116	IPv4	SSH	TCP	22
-	sgr-0a8693ca95f9685f6	IPv4	HTTP	TCP	80
-	sgr-080bf575072a041...	IPv4	SSH	TCP	22

➤ Now try to access the server by using putty/bash/command prompt.

```
ubuntu@ip-172-31-46-103: ~
```

```
Connect to Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.
```

```
EC2 Instance Connect
```

```
C:\Users\SrijaDulam>cd downloads
```

```
C:\Users\SrijaDulam\Downloads>ssh -i "vinay-22.pem" ubuntu@ec2-16-171-162-133.eu-north-1.compute.amazonaws.com
The authenticity of host 'ec2-16-171-162-133.eu-north-1.compute.amazonaws.com (16.171.162.133)' can't be established.
ED25519 key fingerprint is SHA256:gtC2WGtnaC2eEJYR5890dvAjBpPzTFbVoZTe7atkjLc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
1. Warning: Permanently added 'ec2-16-171-162-133.eu-north-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
2. Local file 'known_hosts' is up-to-date.
3. Remote Documentation: https://help.ubuntu.com
   * Management: https://landscape.canonical.com
   * Support: https://ubuntu.com/pro
4. Continue

System information as of Wed Sep  4 10:06:52 UTC 2024
CloudShell Feedback
CloudShell 27°C Mostly cloudy
System load: 0.0      Processes: 100
Usage of /: 21.1% of 7.57GB  Users logged in: 0
Memory usage: 22%
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
```

CloudShell Feedback

CloudShell 27°C Mostly cloudy

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

ENG US 15:36 04-09-2024

➤ After providing the SSH port 22 with anywhereIPV4 than it gets connected to the server.

LAB-6[volumes and snapshots]

Create one 5gb volume

The screenshot shows the 'Create volume' page in the AWS EC2 console. The 'Volume type' is set to 'General Purpose SSD (gp3)'. The 'Size (GiB)' is set to 5. The 'IOPS' value is 3000. The 'Throughput (MiB/s)' is not specified. A note indicates that gp3 is now the default selection, providing up to 20% lower cost per GB than gp2.

The screenshot shows the 'Volumes' page in the AWS EC2 console. It displays two newly created volumes: 'vol-04de26e4d92063b79' (8 GiB, gp3, IOPS 3000, Throughput 125, Created 2024-09-04 15:58 GMT+5...) and 'vol-01ac8f4dc6dced300' (5 GiB, gp3, IOPS 3000, Throughput 125, Created 2024-09-04 16:01 GMT+5...). A success message at the top states 'Successfully created volume vol-01ac8f4dc6dced300.'

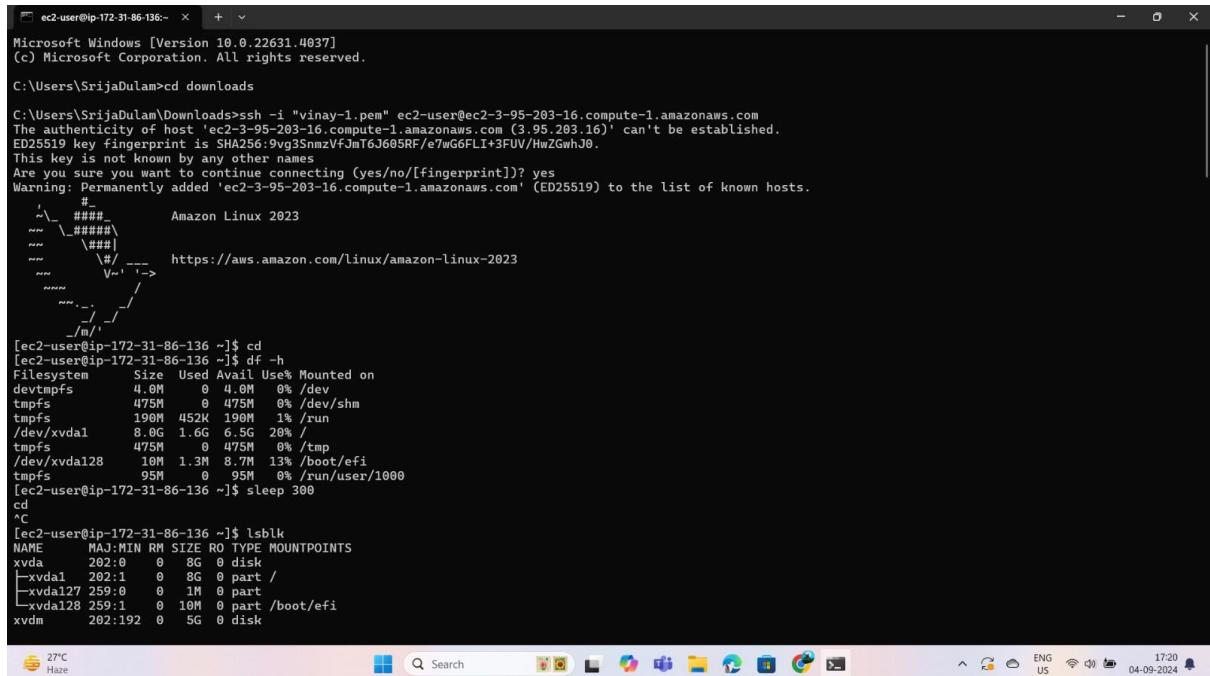
The screenshot shows two screenshots of the AWS Cloud Console side-by-side.

Top Screenshot: The title bar shows three tabs: "Console Home | Console Home", "Attach volume | EC2 | us-east-1", and "Instances | EC2 | us-east-1". The main content is titled "Attach volume" under "EC2 > Volumes > vol-0a4ba4cc97a24d425 > Attach volume". It displays "Basic details" for a volume with Volume ID "vol-0a4ba4cc97a24d425" and Availability Zone "us-east-1". An instance dropdown shows "i-0024731d994ce85a3" selected. A note says "Only instances in the same Availability Zone as the selected volume are displayed." Below it, a device name dropdown says "Select a device name" with a note: "Recommended device names for Linux: /dev/xvda for root volume, /dev/sdf[-p] for data volumes." A warning message in a box says: "Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sd[ef] through /dev/sd[p]."

Bottom Screenshot: The title bar shows three tabs: "Console Home | Console Home", "Volumes | EC2 | us-east-1", and "Instances | EC2 | us-east-1". The main content is titled "Instances (1/2) | Instances". The left sidebar shows navigation options like "EC2 Dashboard", "EC2 Global View", "Events", "Console-to-Code", "Instances", "Images", "AMIs", "AMI Catalog", and "Elastic Block Store". The main table lists one instance: "vinay-6 vol" (Instance ID: i-0024731d994ce85a3, State: Running, Type: t2.micro, Status: 2/2 checks passed, AZ: us-east-1d, Public IP: ec2-3-9). Below the table, the instance details for "i-0024731d994ce85a3 (vinay-6 vol)" are shown, including its attached volumes: "vol-04de26e4d92065b79" (/dev/xvda, 8 GiB, Attached, 2024/09/04 15:58 GMT+5:30) and "vol-0a4ba4cc97a24d425" (/dev/sdm, 5 GiB, Attached, 2024/09/04 16:11 GMT+5:30).

Here the volume is attached to the EC2 instance.

Here this volume is available in the server



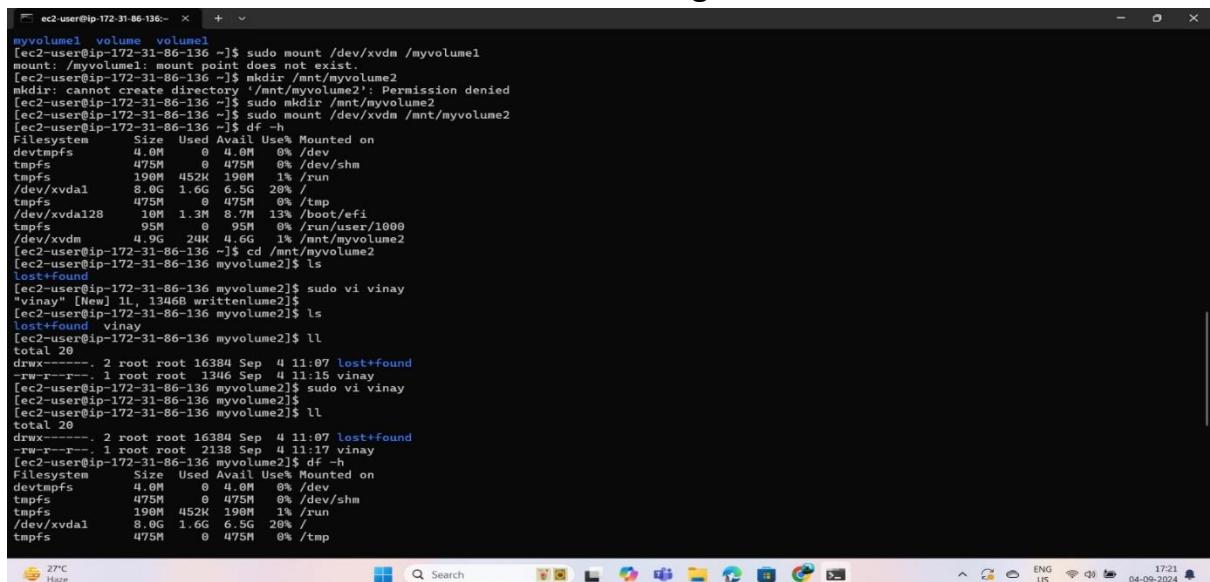
```
ec2-user@ip-172-31-86-136: ~ + 
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SrijaDulam>cd downloads

C:\Users\SrijaDulam\Downloads>sh -i "vinay-1.pem" ec2-user@ec2-3-95-203-16.compute-1.amazonaws.com
The authenticity of host 'ec2-3-95-203-16.compute-1.amazonaws.com (3.95.203.16)' can't be established.
ED25519 key fingerprint is SHA256:9vg3NmzVfJmT6J695RF/e7wG6FLI+3UV/HwZGwhJ0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-95-203-16.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_
  \_###_          Amazon Linux 2023
  \_###_\
  \_###_|
  \_###_| https://aws.amazon.com/linux/amazon-linux-2023
  \_###_|
  \_###_|
  \_###_|
  \_###_|
  \_###_|
[ec2-user@ip-172-31-86-136 ~]$ cd
[ec2-user@ip-172-31-86-136 ~]$ df -h
Filesystem      Size   Used  Avail Use% Mounted on
/devtmpfs        4.0M    0M   4.0M  0% /dev
tmpfs           475M    0M  475M  0% /dev/shm
tmpfs          190M  452K 190M  1% /run
/dev/xvda1       8.0G  1.6G  6.5G  20% /
tmpfs           475M    0M  475M  0% /tmp
/dev/xvda128     10M  1.3M  8.7M  13% /boot/efi
tmpfs           95M    0M  95M  0% /run/user/1000
[ec2-user@ip-172-31-86-136 ~]$ sleep 300
cd
^C
[ec2-user@ip-172-31-86-136 ~]$ lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
xvda   202:0    0   8G  0 disk
└─xvda1  202:1    0   8G  0 part /
xvda127 259:0    0   1M  0 part
└─xvda128 259:1    0  10M  0 part /boot/efi
xvdm   202:192   0   5G  0 disk
27°C
Haze
Q Search
ENG US 17:20 04-09-2024
```

Put some data in this volume like some testing files



```
myvolume1 volume volume1
[ec2-user@ip-172-31-86-136 ~]$ sudo mount /dev/xvdm /myvolume1
mount: /myvolume1: mount point does not exist.
[ec2-user@ip-172-31-86-136 ~]$ mkdir /mnt/myvolume2
mkdir: cannot create directory '/mnt/myvolume2': Permission denied
[ec2-user@ip-172-31-86-136 ~]$ sudo mkdir /mnt/myvolume2
[ec2-user@ip-172-31-86-136 ~]$ sudo mount /dev/xvdm /mnt/myvolume2
[ec2-user@ip-172-31-86-136 ~]$ df -h
Filesystem      Size   Used  Avail Use% Mounted on
/devtmpfs        4.0M    0M   4.0M  0% /dev
tmpfs           475M    0M  475M  0% /dev/shm
tmpfs          190M  452K 190M  1% /run
/dev/xvda1       8.0G  1.6G  6.5G  20% /
tmpfs           475M    0M  475M  0% /tmp
/dev/xvda128     10M  1.3M  8.7M  13% /boot/efi
tmpfs           95M    0M  95M  0% /run/user/1000
/dev/xvdm       4.9G  24K  4.6G  1% /mnt/myvolume2
[ec2-user@ip-172-31-86-136 myvolume2]$ ls
lost+found
[ec2-user@ip-172-31-86-136 myvolume2]$ sudo vi vinay
"vinay" [New] ll, 1346B writtenlum2$ 
[ec2-user@ip-172-31-86-136 myvolume2]$ ls
lost+found vinay
[ec2-user@ip-172-31-86-136 myvolume2]$ ll
total 20
drwxr-xr--  2 root root 16384 Sep  4 11:07 lost+found
-rw-r--r--  1 root root  1346 Sep  4 11:15 vinay
[ec2-user@ip-172-31-86-136 myvolume2]$ sudo vi vinay
[ec2-user@ip-172-31-86-136 myvolume2]$ 
[ec2-user@ip-172-31-86-136 myvolume2]$ ll
total 20
drwxr-xr--  2 root root 16384 Sep  4 11:07 lost+found
-rw-r--r--  1 root root  2138 Sep  4 11:17 vinay
[ec2-user@ip-172-31-86-136 myvolume2]$ df -h
Filesystem      Size   Used  Avail Use% Mounted on
/devtmpfs        4.0M    0M   4.0M  0% /dev
tmpfs           475M    0M  475M  0% /dev/shm
tmpfs          190M  452K 190M  1% /run
/dev/xvda1       8.0G  1.6G  6.5G  20% /
tmpfs           475M    0M  475M  0% /tmp
27°C
Haze
Q Search
ENG US 17:21 04-09-2024
```

Increase the size of this volume to 7gb and check it in the instance.

The screenshot shows the AWS CloudShell interface with two tabs open: "Console Home | Console Home" and "Volumes | EC2 | us-east-1".

The "Volumes | EC2 | us-east-1" tab displays the "Modify volume | EC2 | us-east-1" page. The volume details are as follows:

- Volume ID: vol-0a4ba4cc97a24d425
- Volume type: General Purpose SSD (gp3)
- Size (GiB): 7
- IOPS: 3000
- Throughput (MiB/s): 125

The "Modify" button is highlighted in orange at the bottom right of the form.

The status bar at the bottom of the browser window shows the date and time as 04-09-2024 16:49.

The screenshot shows the AWS CloudShell interface with the same tabs open. The "Volumes | EC2 | us-east-1" tab now displays a confirmation message: "Requested volume modification for volume vol-0a4ba4cc97a24d425. The volume is being modified." Below this, the "Volumes (2) Info" table shows the updated volume information:

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID	Created
-	vol-0a4ba4cc97a24d425	gp3	7 GiB	3000	125	-	2024/09/04 16:07 GMT+5:30...
-	vol-04de26e4d92063b79	gp3	8 GiB	3000	125	snap-0cf83ce6...	2024/09/04 15:58 GMT+5:30...

The status bar at the bottom of the browser window shows the date and time as 04-09-2024 16:49.

The screenshot shows the AWS CloudShell interface with the same tabs open. The "Volumes | EC2 | us-east-1" tab now displays a success message: "Modification successful. The volume has been modified." Below this, the "Volumes (2) Info" table shows the updated volume information.

The status bar at the bottom of the browser window shows the date and time as 04-09-2024 16:50.

```

total 20
drwxr--r--. 2 root root 16384 Sep  4 11:07 lost+found
-rw-r--r--. 1 root root  2138 Sep  4 11:17 vinay
[ec2-user@ip-172-31-86-136 myvolume2]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M   0  4.0M  0% /dev
tmpfs          475M   0  475M  0% /dev/shm
tmpfs          190M  452K 190M  1% /run
/dev/xvda1      8.0G  1.6G  6.5G  20% /
tmpfs          475M   0  475M  0% /tmp
/dev/xvda128    10M  1.3M  8.7M  13% /boot/efi
tmpfs          95M   0  95M  0% /run/user/1000
/dev/xvdm       4.9G  28K  4.6G  1% /mnt/myvolume2
[ec2-user@ip-172-31-86-136 myvolume2]$ cd
[ec2-user@ip-172-31-86-136 ~]$ sudo remount /dev/xvdm /mnt/myvolume2
sudo: remount: command not found
[ec2-user@ip-172-31-86-136 ~]$ sudo mount -o remount /mnt/myvolume2
[ec2-user@ip-172-31-86-136 ~]$ ll
total 0
drwxr-xr-x. 2 ec2-user ec2-user 6 Sep  4 11:08 myvolume1
drwxr-xr-x. 2 root  root  6 Sep  4 11:02 volume
drwxr-xr-x. 2 ec2-user ec2-user 6 Sep  4 11:05 volume1
[ec2-user@ip-172-31-86-136 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M   0  4.0M  0% /dev
tmpfs          475M   0  475M  0% /dev/shm
tmpfs          190M  452K 190M  1% /run
/dev/xvda1      8.0G  1.6G  6.5G  20% /
tmpfs          475M   0  475M  0% /tmp
/dev/xvda128    10M  1.3M  8.7M  13% /boot/efi
tmpfs          95M   0  95M  0% /run/user/1000
/dev/xvdm       4.9G  28K  4.6G  1% /mnt/myvolume2
[ec2-user@ip-172-31-86-136 ~]$ lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
xvda   202:0   0   8G  0 disk
└─xvda1  202:1   0   8G  0 part /
xvda127 259:0   0   1M  0 part
└─xvda128 259:1   0  10M  0 part /boot/efi
xvdm   202:192  0   7G  0 disk /mnt/myvolume2
[ec2-user@ip-172-31-86-136 ~]$ 

```

Extend the size of this volume inside the linux machine.

Create snapshot Info

Create a point-in-time snapshot of an EBS volume and use it as a baseline for new volumes or for data backup. You can create snapshots from an individual volume, or you can create multi-volume snapshots from all of the volumes attached to an instance.

Source

Resource type: [Info](#)

Volume
Create a snapshot from a specific volume.

Instance
Create multi-volume snapshots from an instance.

Volume ID
The volume from which to create the snapshot.
vol-04ba4cc97a24d425
us-east-1d

Snapshot details

Description
Add a description for your snapshot.

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

```

cd
^C
[ec2-user@ip-172-31-86-136 ~]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
xvda    202:0   0   8G  0 disk
└─xvda1  202:1   0   8G  0 part /
└─xvda127 259:0   0   1M  0 part
└─xvda128 259:1   0 10M  0 part /boot/efi
xvdm    202:192  0   5G  0 disk
[ec2-user@ip-172-31-86-136 ~]$ sudo mkdir /mnt/myvolume
[ec2-user@ip-172-31-86-136 ~]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
xvda    202:0   0   8G  0 disk
└─xvda1  202:1   0   8G  0 part /
└─xvda127 259:0   0   1M  0 part
└─xvda128 259:1   0 10M  0 part /boot/efi
xvdm    202:192  0   5G  0 disk
[ec2-user@ip-172-31-86-136 ~]$ sudo mkfs-t ext4/dev/xvdm
sudo: mkfs-t: command not found
[ec2-user@ip-172-31-86-136 ~]$ sudo mkfs -t ext4/dev/xvdm
mkfs: no device specified
Try 'mkfs --help' for more information.
[ec2-user@ip-172-31-86-136 ~]$ sudo mount /dev/xvdm /mnt/myvolume
mount: /mnt/myvolume: wrong fs type, bad option, bad superblock on /dev/xvdm, missing codepage or helper program, or other error.
[ec2-user@ip-172-31-86-136 ~]$ sudo mkdir volume
[ec2-user@ip-172-31-86-136 ~]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
xvda    202:0   0   8G  0 disk
└─xvda1  202:1   0   8G  0 part /
└─xvda127 259:0   0   1M  0 part
└─xvda128 259:1   0 10M  0 part /boot/efi
xvdm    202:192  0   5G  0 disk
[ec2-user@ip-172-31-86-136 ~]$ cd volume
[ec2-user@ip-172-31-86-136 volume]$ cd volume
-bash: cd: volume: No such file or directory
[ec2-user@ip-172-31-86-136 volume]$ cd volume/
-bash: cd: volume/: No such file or directory
[ec2-user@ip-172-31-86-136 volume]$ sudo mnt /dev/xvdm /volume
sudo: mnt: command not found
[ec2-user@ip-172-31-86-136 volume]$ sudo mount /dev/xvdm /volume
mount: /volume: mount point does not exist.

```

Here created a new volume with the snapshot and attach it with the server just like the step1.

The screenshot shows the AWS CloudShell interface with the following terminal session:

```

SuccessFully created snapshot snap-0ba9ac15626edbaf.

```

The terminal then lists the snapshots:

Name	Snapshot ID	Volume size	Description	Storage tier	Snapshot status	Started
-	snap-0ba9ac15626edbaf	7 GiB	-	Standard	Completed	2024/09/04 16:58

Below the table, a message says "Select a snapshot above."

LAB-7[AMI]

- Creating an AMI(Amazon machine image) of our running instance.
- Select the running instance and go to actions.
- Select the image and templates.

The screenshot shows the AWS Management Console with the EC2 Instances page open. A single instance, 'vinay-4' (Instance ID: i-0062053ac1c091889), is listed as 'Running' with the instance type 't3.micro'. The 'Actions' menu is expanded, and the 'Image and templates' option is highlighted. Below the main table, a detailed view for 'i-0062053ac1c091889 (vinay-4)' is shown, including its public and private IP addresses, instance state, and network information.

- Give image name.

The screenshot shows the 'Create Image' page for the instance 'i-0062053ac1c091889 (vinay-4)'. The 'Image name' field is populated with 'vinay-img'. Other fields include 'Image description - optional' (set to 'my image') and a checked 'Reboot instance' checkbox. The 'Instance volumes' section lists attached volumes with their storage type, device, snapshot, size, volume type, IOPS, throughput, delete on termination, and encrypted status. The AWS navigation bar and footer are visible at the bottom.

- Select the required option for the image and create image.

The screenshot shows the 'Create Image' wizard in the AWS EC2 console. The current step is 'Configure volumes'. A single volume is being added, set to EBS General Purpose (SSD) with a size of 5 GiB and an IOPS of 100. The 'Delete on termination' and 'Encrypted' checkboxes are selected. Below this, there's a note about snapshot creation and two radio button options for tagging: 'Tag image and snapshots together' (selected) and 'Tag image and snapshots separately'. At the bottom, there's a 'Create new tag' button and a note about adding up to 50 more tags. The 'Create image' button is at the bottom right.

➤ Here the created AMI with the name vinay-img.

The screenshot shows the 'Images' section in the AWS EC2 console. It lists a single AMI named 'vinay-img' with the ID 'ami-0ccc7a648d5ef1368'. The details page for this AMI is open, showing its configuration. Key details include:

AMI ID	Image type	Platform details	Root device type
ami-0ccc7a648d5ef1368	machine	Linux/UNIX	EBS
AMI name	Owner account ID	Architecture	Usage operation
vinay-img	851725625974	x86_64	RunInstances
Root device name	Status	Source	Virtualization type
/dev/sda1	Pending	851725625974/vinay-img	hvm
Boot mode	State reason	Creation date	Kernel ID

Here we can check the created AMI in My AMIs.

The screenshot shows the AWS IAM console with the URL eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#launchInstances:. The search bar at the top contains "ami Catalog". Below the search bar, there is a message: "you can select one of your own AMIs." A "Selected AMI" section shows "(ami-090abff6ae1141d7d) (Quick Start AMIs)". A search bar below it says "Search for an AMI by entering a search term e.g. "Windows"".

The main interface shows four tabs: "Quick Start AMIs (45)", "My AMIs (1)", "AWS Marketplace AMIs (0)", and "Community AMIs (500)". The "My AMIs" tab is selected, showing "Created by me".

A sidebar on the left titled "Refine results" includes filters for "Owner" (selected "Owned by me"), "OS category" (selected "All Linux/Unix"), and "Publish date range".

The main content area displays "All products (1 filtered, 1 unfiltered)" with a single item listed: "vinay-img" (ami-0ccc7a648d5ef1368). The item details are: OwnerAlias - Platform: Other Linux Architecture: x86_64 Owner: 851725625974 Publish date: 2024-09-04 Root device type: ebs Virtualization: hvm ENA enabled: Yes. A yellow "Select" button is next to the item.

The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray indicating the date and time as 04-09-2024.

- Like this we can create the AMIs as per our required.

LAB-8[Load balancer]

- Launch the two ec2 instances.

The screenshot shows the AWS CloudWatch Metrics console. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code, Instances, Images, AMIs, and Elastic Block Store. The main area displays a table titled 'Instances (2) Info' with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. Two instances are listed: 'vinay-8-2' (Instance ID: i-0ffdc74f50b3ca120, State: Running, Type: t2.micro, Status: Initializing, AZ: us-east-1d, IP: ec2-54-17-188-114) and 'vinay-8-1' (Instance ID: i-0f539751dafb71509, State: Running, Type: t2.micro, Status: Initializing, AZ: us-east-1d, IP: ec2-3-84-4-217). Below the table, a modal window titled 'Select an instance' is open, showing the same two instances. The bottom of the screen shows the Windows taskbar with various icons and system status.

- Open the terminal and connect the server 1.

The screenshot shows a Microsoft Windows terminal window with two tabs: 'vinay-8-1' and 'vinay-8-2'. The 'vinay-8-1' tab is active and shows a command-line session. The user runs 'cd downloads' and then 'ssh -i "vinay-1.pem" ec2-user@ec2-3-84-45-217.compute-1.amazonaws.com'. The terminal prompts for a password, which is entered. A warning message about host key fingerprinting is displayed, followed by a confirmation message asking if the user wants to continue connecting. The user responds 'yes'. The terminal then shows the Amazon Linux 2023 logo and a directory listing for '/etc/ssh'. The session ends with '[ec2-user@ip-172-31-84-106 ~]\$'. The bottom of the screen shows the Windows taskbar with various icons and system status.

- Open the another terminal and connect the server 2.

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\SrijaDulam> cd downloads
PS C:\Users\SrijaDulam\downloads> ssh -i "vinay-1.pem" ec2-user@ec2-54-175-146-121.compute-1.amazonaws.com
The authenticity of host 'ec2-54-175-146-121.compute-1.amazonaws.com (64:ff9b::36af:9279)' can't be established.
ED25519 key fingerprint is SHA256:TwJ0h3C1RK79iEAtyTGAf8cRUy1XijINgHcwukvjkTk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-175-146-121.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

# 
Amazon Linux 2023
\###_ 
\###_ 
\###_ 
\###_ 
https://aws.amazon.com/linux/amazon-linux-2023
/ 
/ 
/ 
/ 
/ 
[ec2-user@ip-172-31-85-223 ~]$
```

- Go to the security groups and edit the inbound rules.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-044c92fffc020e2fb	SSH	TCP	22	Custom	<input type="text"/> 0.0.0.0/0 <button>Delete</button>
-	HTTPS	TCP	443	Anywh...	<input type="text"/> 0.0.0.0/0 <button>Delete</button>
-	HTTP	TCP	80	Anywh...	<input type="text"/> 0.0.0.0/0 <button>Delete</button>

[Add rule](#)

- Allow HTTPS and HTTP to the both servers along with the SSH default port.
- Attach that security groups to the both instances.

The screenshot shows the AWS Management Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroup:group-id=sg-0084c3aeb3e0ef651>. The main content area displays the security group details for 'launch-wizard-27' (Security group ID: sg-0084c3aeb3e0ef651). It shows 3 inbound rules and 1 outbound rule. The 'Inbound rules' tab is active, listing the following rules:

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-077090aa6479f106	IPv4	HTTP	TCP	80
-	sgr-044c92fffc020e2fb	IPv4	SSH	TCP	22
-	sgr-0436d3188bcb84...	IPv4	HTTPS	TCP	443

- In instance 1 install the HTTPD or APACHE by using “`sudo yum install httpd -y`”.

```

vinay-8-1 ~
vinay-8-2 ~
Cloning into '/var/www/html'...
/va/www/html/.git: Permission denied
[ec2-user@ip-172-31-84-106 ~]$ git clone https://github.com/Gouserabbani44/food.git
Cloning into 'food'...
remote: Enumerating objects: 71, done.
remote: Counting objects: 100% (77/77), done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 71 (delta 2), reused 1 (delta 1), pack-reused 64 (from 1)
Receiving objects: 100% (71/71), 2.29 MiB | 33.07 MiB/s, done.
Resolving deltas: 100% (2/2), done.
^Z
[1]+  Stopped                  git clone https://github.com/Gouserabbani44/food.git
[ec2-user@ip-172-31-84-106 ~]$ git clone https://github.com/Gouserabbani44/food.git
fatal: destination path 'food' already exists and is not an empty directory.
[ec2-user@ip-172-31-84-106 ~]$ ls
burger-king.zip food vinay
[ec2-user@ip-172-31-84-106 ~]$ sudo mv food /var/www/html/
[ec2-user@ip-172-31-84-106 ~]$ ls
burger-king.zip vinay
[ec2-user@ip-172-31-84-106 ~]$ cd /var/www/html/
[ec2-user@ip-172-31-84-106 html]$ ls
food
[ec2-user@ip-172-31-84-106 html]$ cd
[ec2-user@ip-172-31-84-106 ~]$ sudo rm -rf /var/www/html
[ec2-user@ip-172-31-84-106 ~]$ ls
burger-king.zip vinay
[ec2-user@ip-172-31-84-106 ~]$ git clone https://github.com/Gouserabbani44/ecomm.git /var/www/html/
fatal: could not create work tree dir '/var/www/html': Permission denied
[ec2-user@ip-172-31-84-106 ~]$ sudo git clone https://github.com/Gouserabbani44/ecomm.git /var/www/html/
Cloning into '/var/www/html'...
remote: Enumerating objects: 210, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 210 (delta 27), reused 23 (delta 23), pack-reused 177 (from 1)
Receiving objects: 100% (210/210), 4.60 MiB | 34.90 MiB/s, done.
Resolving deltas: 100% (48/48), done.
[ec2-user@ip-172-31-84-106 ~]$ cd /var/www/html/
[ec2-user@ip-172-31-84-106 html]$ ls
README.md checkout.html css img js regular-page.html shop.html single-product-details.html test.txt
blog.html contact.html fonts index.html oldfile scss single-blog.html style.css
[ec2-user@ip-172-31-84-106 html]$ |

```

- After installing that check the status of the httpd by using the command “`sudo systemctl status httpd`”.
- It shows the inactive
- Than use the command “`sudo systemctl start httpd`” to start
- Than enable the httpd by using “`sudo systemctl enable httpd`”

- Than clone the repository from the github platform by using the “git clone <url>”.
- Than cloned repository moved to the html “var/www/html/” by using “sudo mv repo name * /var/www/html/”

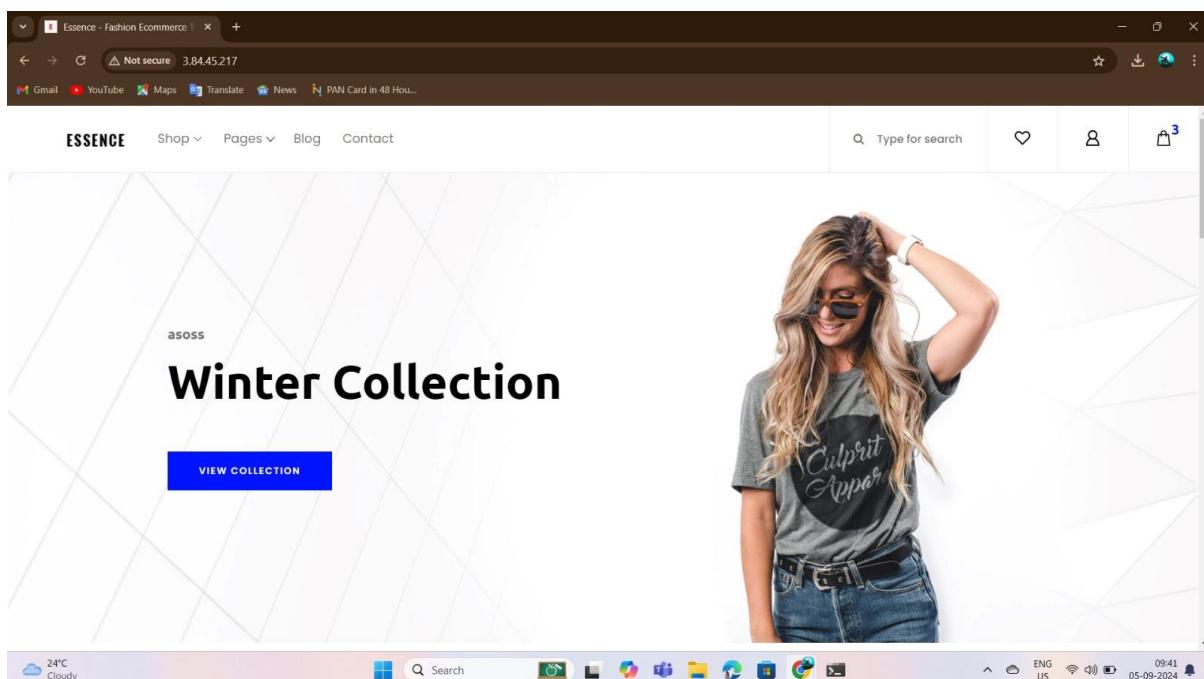
```

vinay-8-1      x  vinay-8-2      x  +  v
Cloning into '/var/www/html'...
/va/www/html/.git: Permission denied
[ec2-user@ip-172-31-84-106 ~]$ git clone https://github.com/Gouserabbani44/food.git
Cloning into 'food'.
remote: Enumerating objects: 71, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 71 (delta 2), reused 1 (delta 1), pack-reused 64 (from 1)
Receiving objects: 100% (71/71), 2.29 MiB | 33.07 MiB/s, done.
Resolving deltas: 100% (2/2), done.
```
[1]+ Stopped git clone https://github.com/Gouserabbani44/food.git
[ec2-user@ip-172-31-84-106 ~]$ git clone https://github.com/Gouserabbani44/food.git
fatal: destination path 'food' already exists and is not an empty directory.
[ec2-user@ip-172-31-84-106 ~]$ ls
burger-king.zip food vinay
[ec2-user@ip-172-31-84-106 ~]$ sudo mv food /var/www/html/
[ec2-user@ip-172-31-84-106 ~]$ ls
burger-king.zip vinay
[ec2-user@ip-172-31-84-106 ~]$ cd /var/www/html/
[ec2-user@ip-172-31-84-106 html]$ ls
food
[ec2-user@ip-172-31-84-106 html]$ cd
[ec2-user@ip-172-31-84-106 ~]$ sudo rm -rf /var/www/html
[ec2-user@ip-172-31-84-106 ~]$ ls
burger-king.zip vinay
[ec2-user@ip-172-31-84-106 ~]$ git clone https://github.com/Gouserabbani44/ecomm.git /var/www/html/
fatal: could not create work tree dir '/var/www/html': Permission denied
[ec2-user@ip-172-31-84-106 ~]$ sudo git clone https://github.com/Gouserabbani44/ecomm.git /var/www/html/
Cloning into '/var/www/html'...
remote: Enumerating objects: 210, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 210 (delta 27), reused 23 (delta 23), pack-reused 177 (from 1)
Receiving objects: 100% (210/210), 4.60 MiB | 34.90 MiB/s, done.
Resolving deltas: 100% (48/48), done.
[ec2-user@ip-172-31-84-106 ~]$ cd /var/www/html/
[ec2-user@ip-172-31-84-106 html]$ ls
README.md checkout.html css img js regular-page.html shop.html single-product-details.html test.txt
blog.html contact.html fonts index.html oldfile scss single-blog.html style.css
[ec2-user@ip-172-31-84-106 html]$ |

```

Cloudy 24°C ENG US 05-09-2024 09:41

- After moving the path copy the public IP of the instance 1 and search in the google, we can get the below image which I cloned from the github.



- From the instance 2 connect the terminal by using ssh command.

```

Sep 05 03:37:45 ip-172-31-85-223.ec2.internal nginx[26004]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Sep 05 03:37:45 ip-172-31-85-223.ec2.internal nginx[26004]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Sep 05 03:37:45 ip-172-31-85-223.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-172-31-85-223 ~]$ ls
techmax
[ec2-user@ip-172-31-85-223 ~]$ sudo mv * /usr/share/nginx/html
[ec2-user@ip-172-31-85-223 ~]$ cd /usr/share/nginx/html
[ec2-user@ip-172-31-85-223 html]$ ls
404.html 50x.html icons index.html nginx-logo.png poweredby.png techmax
[ec2-user@ip-172-31-85-223 html]$ ls
404.html 50x.html icons index.html nginx-logo.png poweredby.png techmax
[ec2-user@ip-172-31-85-223 html]$ ls
404.html 50x.html icons index.html nginx-logo.png poweredby.png techmax
[ec2-user@ip-172-31-85-223 html]$ sleep 400
^C
[ec2-user@ip-172-31-85-223 html]$ cd
[ec2-user@ip-172-31-85-223 ~]$ sudo git clone https://github.com/Gouserabbani44/ecomm.git /usr/share/nginx/html
fatal: destination path '/usr/share/nginx/html' already exists and is not an empty directory.
[ec2-user@ip-172-31-85-223 ~]$ git clone https://github.com/Gouserabbani44/ecomm.git
Cloning into 'ecomm'...
remote: Enumerating objects: 210, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 210 (delta 27), reused 23 (delta 23), pack-reused 177 (from i)
Receiving objects: 100% (210/210), 4.60 MiB | 37.99 MiB/s, done.
Resolving deltas: 100% (48/48), done.
[ec2-user@ip-172-31-85-223 ~]$ cd /usr/share/nginx/html
[ec2-user@ip-172-31-85-223 html]$ ls
404.html 50x.html icons index.html nginx-logo.png poweredby.png techmax
[ec2-user@ip-172-31-85-223 html]$ sudo rm -rf techmax
[ec2-user@ip-172-31-85-223 html]$ ls
404.html 50x.html icons index.html nginx-logo.png poweredby.png
[ec2-user@ip-172-31-85-223 html]$ cd
[ec2-user@ip-172-31-85-223 ~]$ ls
ecomm
[ec2-user@ip-172-31-85-223 ~]$ sudo mv * /usr/share/nginx/html
[ec2-user@ip-172-31-85-223 ~]$ ls
[ec2-user@ip-172-31-85-223 ~]$ cd /usr/share/nginx/html
[ec2-user@ip-172-31-85-223 html]$ ls
404.html 50x.html ecomm icons index.html nginx-logo.png poweredby.png
[ec2-user@ip-172-31-85-223 html]$
```

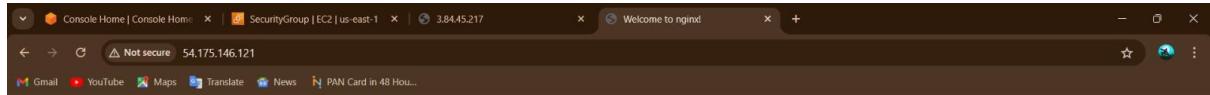
- Install the nginx by using “sudo yum install nginx -y”

```

Sep 05 03:45 ip-172-31-85-223.ec2.internal nginx[26004]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Sep 05 03:45 ip-172-31-85-223.ec2.internal nginx[26004]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Sep 05 03:45 ip-172-31-85-223.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-172-31-85-223 ~]$ ls
techmax
[ec2-user@ip-172-31-85-223 ~]$ sudo mv * /usr/share/nginx/html
[ec2-user@ip-172-31-85-223 ~]$ cd /usr/share/nginx/html
[ec2-user@ip-172-31-85-223 html]$ ls
404.html 50x.html icons index.html nginx-logo.png poweredby.png techmax
[ec2-user@ip-172-31-85-223 html]$ ls
404.html 50x.html icons index.html nginx-logo.png poweredby.png techmax
[ec2-user@ip-172-31-85-223 html]$ ls
404.html 50x.html icons index.html nginx-logo.png poweredby.png techmax
[ec2-user@ip-172-31-85-223 html]$ sleep 400
^C
[ec2-user@ip-172-31-85-223 html]$ cd
[ec2-user@ip-172-31-85-223 ~]$ sudo git clone https://github.com/Gouserabbani44/ecomm.git /usr/share/nginx/html
fatal: destination path '/usr/share/nginx/html' already exists and is not an empty directory.
[ec2-user@ip-172-31-85-223 ~]$ git clone https://github.com/Gouserabbani44/ecomm.git
Cloning into 'ecomm'...
remote: Enumerating objects: 210, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 210 (delta 27), reused 23 (delta 23), pack-reused 177 (from i)
Receiving objects: 100% (210/210), 4.60 MiB | 37.99 MiB/s, done.
Resolving deltas: 100% (48/48), done.
[ec2-user@ip-172-31-85-223 ~]$ cd /usr/share/nginx/html
[ec2-user@ip-172-31-85-223 html]$ ls
404.html 50x.html icons index.html nginx-logo.png poweredby.png techmax
[ec2-user@ip-172-31-85-223 html]$ sudo rm -rf techmax
[ec2-user@ip-172-31-85-223 html]$ ls
404.html 50x.html icons index.html nginx-logo.png poweredby.png
[ec2-user@ip-172-31-85-223 html]$ cd
[ec2-user@ip-172-31-85-223 ~]$ ls
ecomm
[ec2-user@ip-172-31-85-223 ~]$ sudo mv * /usr/share/nginx/html
[ec2-user@ip-172-31-85-223 ~]$ ls
[ec2-user@ip-172-31-85-223 ~]$ cd /usr/share/nginx/html
[ec2-user@ip-172-31-85-223 html]$ ls
404.html 50x.html ecomm icons index.html nginx-logo.png poweredby.png
[ec2-user@ip-172-31-85-223 html]$
```

- Start the nginx “sudo systemctl start nginx”
- Enable the nginx “sudo systemctl enable nginx”
- Check the status of the nginx “sudo systemctl status nginx”
- Then go to the instance2 and copy the public IP of the instance.

➤ Search the Public Ip in the google and we get below image.



### Welcome to nginx!

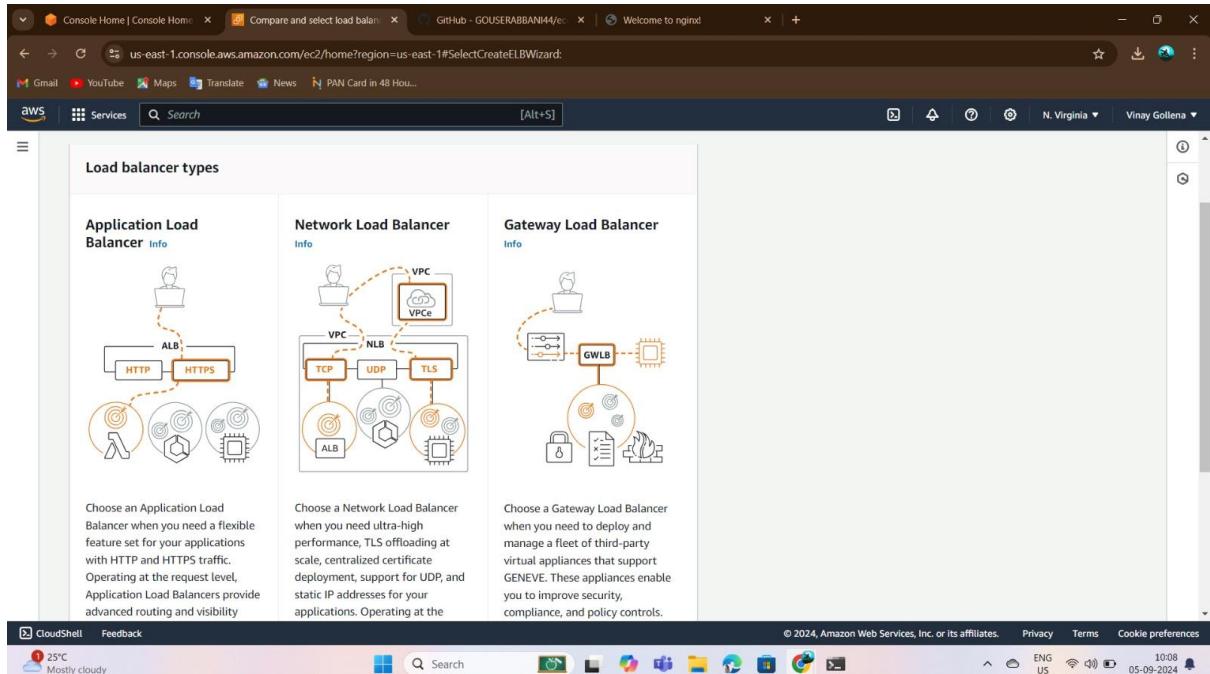
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*



➤ Creating the Loadbalancer.



The screenshot shows the AWS CloudFront console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateALBWizard>. The page displays information about Application Load Balancers, specifically how they distribute incoming HTTP and HTTPS traffic across multiple targets. It includes sections for basic configuration, such as load balancer name (set to 'vinay-lb'), scheme (set to 'Internet-facing'), and IP address type (set to 'IPv4'). The interface is clean with a white background and standard AWS branding.

## ➤ Give the all availability zones.

The screenshot shows the AWS CloudFront console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateALBWizard>. The page displays the 'Availability zones' section where four zones are selected: 'us-east-1a (use1-az4)', 'us-east-1b (use1-az6)', 'us-east-1c (use1-az1)', and 'us-east-1d (use1-az2)'. Each selected zone has its corresponding subnet listed below it. The interface is clean with a white background and standard AWS branding.

The screenshot shows the 'Create application load balancer' wizard on the AWS CloudFront service. The user is selecting target subnets for their load balancer. Three subnets are listed:

- us-east-1e (use1-az3)**: Subnet subnet-0bb45835ee4b71d6a (IPv4 subnet CIDR: 172.31.80.0/20)
- us-east-1f (use1-az5)**: Subnet subnet-05189e934e16d5a9a (IPv4 subnet CIDR: 172.31.64.0/20)

Below the subnets, there is a section for **Security groups** with a note: "A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can create a new security group." A link to 'Info' is also present.

## ➤ Create the target groups by using instances.

The screenshot shows the 'Step 1: Create target group' wizard on the AWS EC2 service. The user is selecting a target type for their target group. The 'Basic configuration' section is visible, stating: "Your load balancer routes requests to the targets in a target group and performs health checks on the targets." The 'Choose a target type' section contains three options:

- Instances** (selected):
  - Supports load balancing to instances within a specific VPC.
  - Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.
- IP addresses**:
  - Supports load balancing to VPC and on-premises resources.
  - Facilitates routing to multiple IP addresses and network interfaces on the same instance.
  - Offers flexibility with microservice based architectures, simplifying inter-application communication.
  - Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.
- Lambda function**:
  - Facilitates routing to a single Lambda function.
  - Accessible to Application Load Balancers only.

Screenshot of the AWS CloudFront console showing the creation of a target group. The target group is named "vinay-target" and is of type "Application Load Balancer". The protocol is set to "HTTP" on port "80". The IP address type is selected as "IPv4".

**Target group name:** vinay-target

**Protocol : Port:** HTTP : 80

**IP address type:** IPv4

**Register targets:**

| Instance ID         | Name      | State   | Security groups  |
|---------------------|-----------|---------|------------------|
| i-0ffdc74f50b3ca120 | vinay-8-2 | Running | launch-wizard-28 |
| i-0f539751dafb71509 | vinay-8-1 | Running | launch-wizard-27 |

**Ports for the selected instances:** 80  
1-65535 (separate multiple ports with commas)

The screenshot shows the AWS EC2 Target Groups page. On the left, a sidebar navigation includes EC2 Dashboard, EC2 Global View, Events, Console-to-Code Preview, Instances (selected), Images, AMIs, AMI Catalog, and Elastic Block Store. The main content area displays a summary of target status: 2 Total targets, 0 Healthy, 0 Unhealthy, 2 Unused, 0 Initial, and 0 Draining. Below this is a table titled "Registered targets (2) Info". The table has columns for Instance ID, Name, Port, Zone, Health status, and Health status details. Two targets are listed: "i-0ffd74f50b5ca120" (Name: vinay-8-2, Port: 80, Zone: us-east-1d, Health status: Unused, Details: Target group is not co...) and "i-0f539751dafb71509" (Name: vinay-8-1, Port: 80, Zone: us-east-1d, Health status: Unused, Details: Target group is not co...).

- Allow port 80.
- Attach the target group.

The screenshot shows the AWS Load Balancers CreateALBWizard page. In the "Security groups" section, "default" is selected. In the "Listeners and routing" section, a new listener for "HTTP:80" is being configured. The protocol is set to "HTTP" and the port is "80". The "Default action" is "Forward to" the "vinay-target" target group, which is defined as "Target type: Instance, IPv4". There is also a "Create target group" button. Below the listener configuration, there is a section for "Listener tags - optional" with a "Add listener tag" button and a note about adding up to 50 more tags.

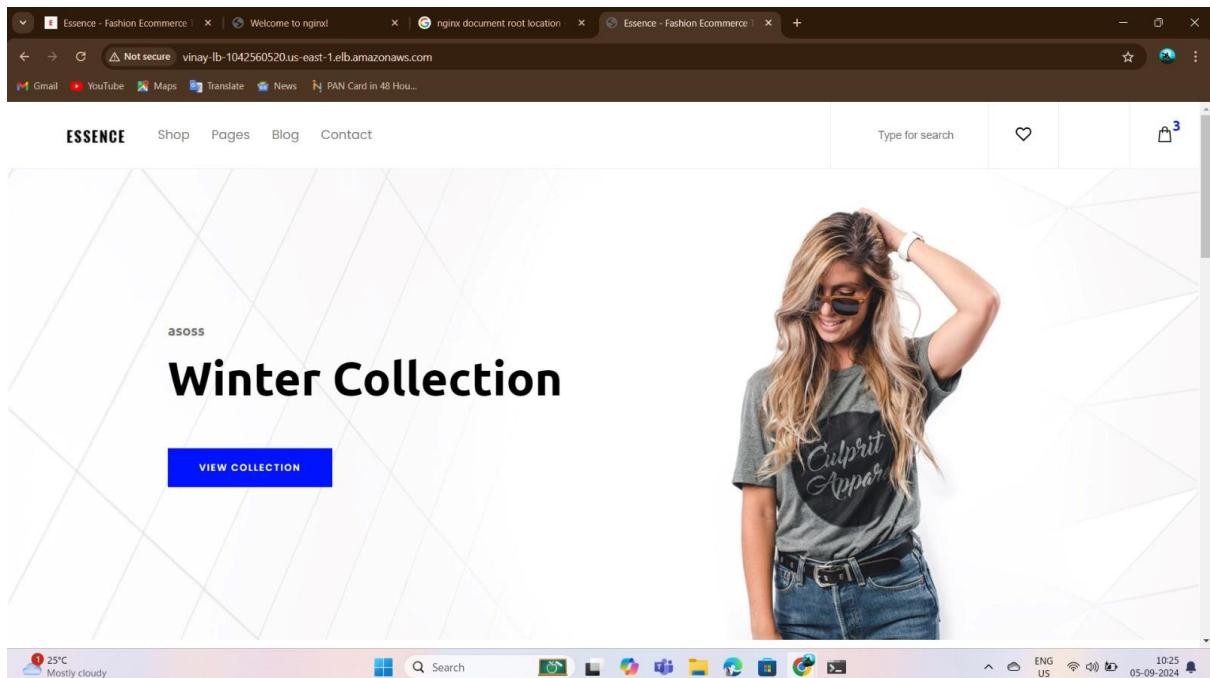
The screenshot shows the AWS CloudWatch Metrics interface. On the left, there's a sidebar with navigation links like 'AWS Dashboard', 'Services', 'Search', and 'Metrics'. The main area displays a single log stream titled 'HelloWorld'. The log entries show timestamped messages from the Lambda function, such as 'HelloWorld - Hello world!' and 'HelloWorld - Hello world!'. The log stream has a 'CloudWatch Metrics' icon next to it.

➤ The load balancer has created.

The screenshot shows the AWS CloudWatch Metrics interface. On the left, there's a sidebar with navigation links like 'AWS Dashboard', 'Services', 'Search', and 'Metrics'. The main area displays a single log stream titled 'HelloWorld'. The log entries show timestamped messages from the Lambda function, such as 'HelloWorld - Hello world!' and 'HelloWorld - Hello world!'. The log stream has a 'CloudWatch Metrics' icon next to it.

- Copy the end point of the load balancer and search in the google.
- Refresh the page and observe the changes.

➤ First one.



➤ Second one after refreshing.



# LAB-9[ASG & LT]

- Create one launch templete.

The screenshot shows the AWS EC2 Launch Templates landing page. The URL is [us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchTemplates](https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchTemplates). The page title is "standardize instance launches". On the left, there's a sidebar with navigation links like EC2 Dashboard, Instances, Launch Templates (which is selected), and API reference. The main content area has a section titled "Benefits and features" with three items: "Streamline provisioning", "Simplify permissions", and "Governance". A prominent button at the top right says "Create launch template". To the right, there's a "Documentation" section with links to "Documentation" and "API reference". The bottom right corner shows the AWS logo and the date "05-09-2024".

The screenshot shows the "Create launch template" wizard, step 1: "Launch template name and description". The URL is [us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTemplate](https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTemplate). The page title is "Create launch template". The left sidebar shows the user is in the EC2 > Launch templates section. The main form has fields for "Launch template name - required" (containing "vinay-LT") and "Template version description" (containing "A prod webserver for MyApp"). There's also a "Auto Scaling guidance" section with a checkbox for "Provide guidance to help me set up a template that I can use with EC2 Auto Scaling". A tooltip for "Free tier" is visible, explaining the benefits of using t2.micro instances. At the bottom right is a "Create launch template" button.

The screenshot shows the AWS Lambda console interface. At the top, there are tabs for 'Console Home' and 'Create launch template'. Below the tabs, there's a search bar and a navigation menu with 'Services' selected. The main area is titled 'Create new function' and contains fields for 'Function name' (set to 'lambda'), 'Runtime' (set to 'Node.js 18.x'), and 'Handler' (set to 'index.handler'). A large 'Create Function' button is at the bottom right. The status bar at the bottom indicates 'CloudWatch Metrics' and 'CloudWatch Metrics'.

## ➤ Creating a launch template with ubuntu server.

The screenshot shows the AWS Lambda console interface. At the top, there are tabs for 'Console Home' and 'Create launch template'. Below the tabs, there's a search bar and a navigation menu with 'Services' selected. The main area is titled 'Create new function' and contains fields for 'Function name' (set to 'lambda'), 'Runtime' (set to 'Node.js 18.x'), and 'Handler' (set to 'index.handler'). A large 'Create Function' button is at the bottom right. The status bar at the bottom indicates 'CloudWatch Metrics' and 'CloudWatch Metrics'.

➤ Templatized has launched.

The screenshot shows the AWS EC2 Launch Templates page. The left sidebar is collapsed. The main area displays a table titled "Launch Templates (1/1) Info". A single row is selected, showing the following details:

| Launch Template ID   | Launch Template Name | Default Version | Latest Version | Create Time              | Created By       |
|----------------------|----------------------|-----------------|----------------|--------------------------|------------------|
| lt-01c85e8a70108beb9 | vinay-LT             | 1               | 1              | 2024-09-05T09:00:15.000Z | arn:aws:iam::... |

Below the table, a modal window titled "vinay-LT (lt-01c85e8a70108beb9)" is open, showing "Launch template details". The details are as follows:

| Launch template ID   | Launch template name | Default version | Owner                         |
|----------------------|----------------------|-----------------|-------------------------------|
| lt-01c85e8a70108beb9 | vinay-LT             | 1               | arn:aws:iam:851725625974:root |

At the bottom of the page, there are tabs for "Details", "Versions", and "Template tags". The status bar at the bottom right shows "ENG US" and the date "05-09-2024".

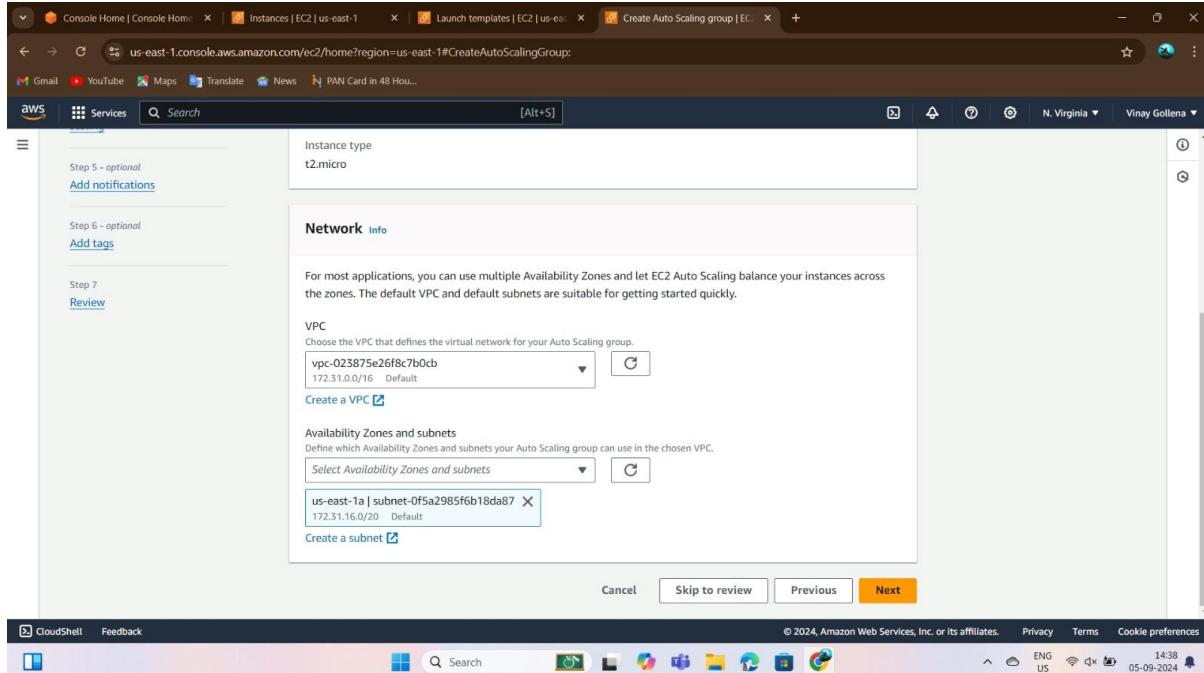
➤ Create an Auto scaling group.

The screenshot shows the "Create Auto Scaling group" wizard, currently on Step 2: "Choose instance launch options". The left sidebar lists steps from Step 2 to Step 7. The main form is titled "Name" and contains the following fields:

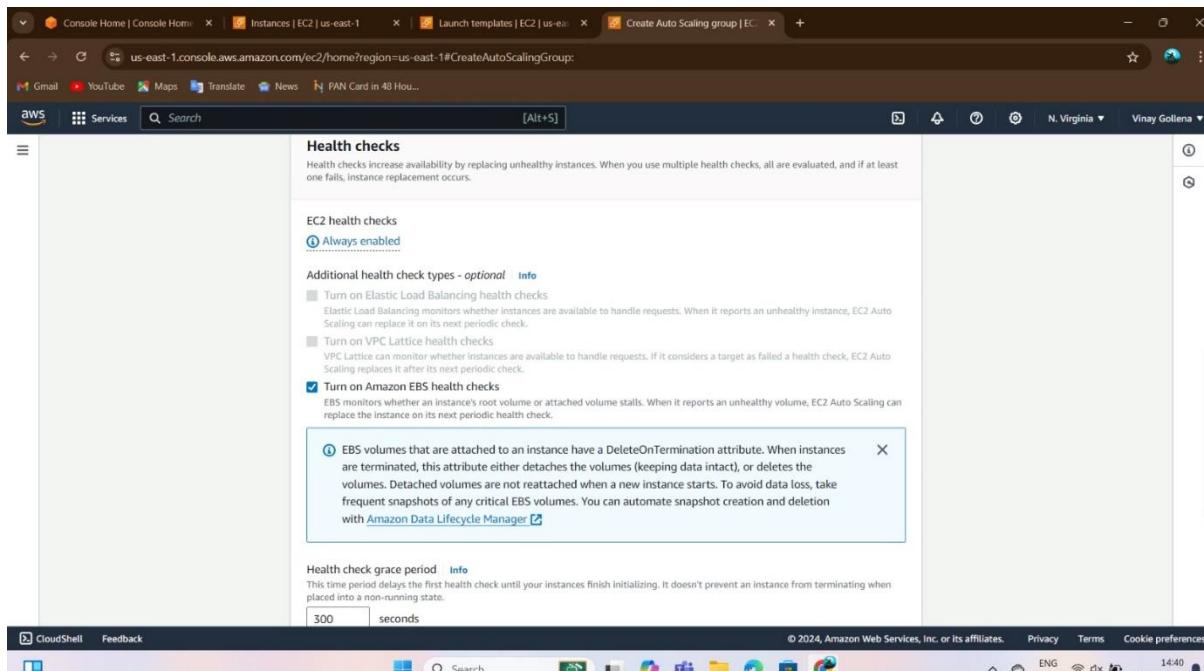
- "Auto Scaling group name": A text input field containing "vinay-ASG". A note below it states: "Must be unique to this account in the current Region and no more than 255 characters."
- "Launch template": A dropdown menu set to "vinay-LT". Below it is a link "Create a launch template".
- "Version": A dropdown menu set to "Default (1)". Below it is a link "Create a launch template version".

At the bottom of the form, there is a note: "For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023." The status bar at the bottom right shows "ENG US" and the date "05-09-2024".

- Attach the availability zones and give default vpc



- Turn on Amazon EBS health checks.



- Set the desired capacity=2,minimum desired capacity=1 and maximum desired capacity=3

Configure group size and scaling

Step 5 - optional

[Add notifications](#)

Step 6 - optional

[Add tags](#)

Step 7

[Review](#)

Desired capacity

Specify your group size.

Scaling Info

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity      Max desired capacity

1      3

Equal or less than desired capacity      Equal or greater than desired capacity

Automatic scaling - optional

Choose whether to use a target tracking policy [Info](#)

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

No scaling policies  
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

Target tracking scaling policy  
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

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

CloudShell Feedback

ENG US 14:41 05-09-2024

➤ Auto scaling group has been created.

Auto Scaling groups (1/1) [Info](#)

Search your Auto Scaling groups

| Name                      | Launch template/configuration              | Instances | Status | Desired capacity | Min | Max | Available  |
|---------------------------|--------------------------------------------|-----------|--------|------------------|-----|-----|------------|
| <a href="#">vinay-ASG</a> | <a href="#">vinay-LT</a>   Version Default | 2         | -      | 2                | 1   | 3   | us-east-1a |

Auto Scaling group: vinay-ASG

Details    Activity    Automatic scaling    Instance management    Monitoring    Instance refresh

Group details

|                                                                         |                       |                                                      |                                                                                                                                                                               |
|-------------------------------------------------------------------------|-----------------------|------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Auto Scaling group name<br>vinay-ASG                                    | Desired capacity<br>2 | Desired capacity type<br>Units (number of instances) | Amazon Resource Name (ARN)<br><a href="#">arn:aws:autoscaling:us-east-1:851725625974:autoScalingGroup:17400d0e-d997-4af8-8ec5-4ae76bee41b9:autoScalingGroupName/vinay-ASG</a> |
| Date created<br>Thu Sep 05 2024 14:42:49 GMT+0530 (India Standard Time) | Minimum capacity<br>1 | Status<br>-                                          |                                                                                                                                                                               |
|                                                                         | Maximum capacity      |                                                      |                                                                                                                                                                               |

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

CloudShell Feedback

ENG US 14:45 05-09-2024

Desired instances are launched.

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main table displays two instances:

| Name | Instance ID         | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IP |
|------|---------------------|----------------|---------------|--------------|--------------|-------------------|-----------|
|      | i-02569f165bb4f4527 | Running        | t2.micro      | Initializing | View alarms  | us-east-1a        | ec2-34-   |
|      | i-0393315f3331cdc22 | Running        | t2.micro      | Initializing | View alarms  | us-east-1a        | ec2-34-   |

A modal window titled "Select an instance" is open at the bottom of the page.

- Terminate the instances.
- Increased the max desired capacity to 4.

The screenshot shows the "Edit Auto Scaling group" page for the vinay-ASG group. The "Desired capacity type" section is expanded, showing the unit of measurement for the desired capacity value. The "Desired capacity" field is set to 2. The "Scaling limits" section shows a range from 1 to 4. The "Launch template" section contains a note about launch configurations being available via the CLI and API until December 31, 2023.

- Another 2 instances are launched according to the desired capacity.

The screenshot shows the AWS EC2 Instances dashboard. On the left, a sidebar lists various services like EC2 Dashboard, EC2 Global View, Events, and Instances. The Instances section is expanded, showing Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, and Reservations. Below this, there's a section for Images (AMIs, AMI Catalog) and Elastic Block Store (Volumes).

The main content area displays a table titled "Instances (2/4) Info". It lists four instances:

| Name                                | Instance ID          | Instance state | Instance type | Status check      | Alarm status  | Availability Zone | Public IP |
|-------------------------------------|----------------------|----------------|---------------|-------------------|---------------|-------------------|-----------|
| <input checked="" type="checkbox"/> | i-0d696bd8e01516c6a  | Running        | t2.micro      | 2/2 checks passed | View alarms + | us-east-1a        | ec2-34-   |
| <input checked="" type="checkbox"/> | i-00347fda0eedad5f71 | Running        | t2.micro      | 2/2 checks passed | View alarms + | us-east-1a        | ec2-107   |
| <input type="checkbox"/>            | i-02569ff165bb4f4527 | Terminated     | t2.micro      | -                 | View alarms + | us-east-1a        | -         |
|                                     | i-0202215f2331ad-22  | Terminated     | t2.micro      | -                 | View alarms + | us-east-1a        | -         |

Below the table, it says "2 instances selected". Under the "Monitoring" tab, there are four charts for CPU utilization (%), Network in (bytes), Network out (bytes), and Network packets in (count). Each chart shows a single data series with no data available.

When we changes the maximum capacity new instances has been created.

# LAB-10[RDS]

## ➤ Create a database.

The screenshot shows the AWS RDS Management console with the 'Databases' page open. The left sidebar includes links for Dashboard, Databases (which is selected), Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, CloudShell, and Feedback. The main area displays a table titled 'Databases (0)' with columns for DB identifier, Status, Region &..., Size, Recommendations, and Current act. A search bar at the top allows filtering by database name. A prominent orange button labeled 'Create database' is located at the top right of the table area. The status bar at the bottom indicates the user is in the N. Virginia region and shows the date as 05-09-2024.

## ➤ Select MySQL.

The screenshot shows the 'Create database - RDS Management' wizard. The first step, 'Choose a database creation method', offers two options: 'Standard create' (selected) and 'Easy create'. The 'Standard create' option is described as allowing configuration for availability, security, backups, and maintenance. The 'Easy create' option is described as using recommended best-practice configurations. Below this, the 'MySQL' section provides information about MySQL being the most popular open source database and details its features, including support for up to 64 TiB, various instance classes, automated backup, and up to 15 read replicas. The 'Engine options' section lists four engine types: Aurora (MySQL Compatible), Aurora (PostgreSQL Compatible), MySQL (selected), and MariaDB. Each engine type has a corresponding icon. The status bar at the bottom indicates the user is in the N. Virginia region and shows the date as 05-09-2024.

## ➤ Select the free tier.

The screenshot shows the 'Create database' step in the AWS RDS Manager. In the 'Templates' section, the 'Free tier' option is selected, highlighted with a blue border. The 'MySQL' sidebar on the right provides a brief overview of MySQL's popularity and features, including support for up to 64 TiB and various instance classes. The deployment options section lists 'Multi-AZ DB Cluster', 'Multi-AZ DB instance', and 'Single DB instance', with 'Multi-AZ DB instance' being the selected choice.

## ➤ make the master username as admin and keep an master password.

The screenshot shows the 'Create database' step in the AWS RDS Manager. Under 'Master password', the 'Self managed' option is selected. A strong password is entered in the 'Master password' field. The 'MySQL' sidebar on the right reiterates the database's features and capabilities. The 'Instance configuration' section at the bottom is visible but contains no specific configuration details.

➤ choose existing Vpc and security group.

The screenshot shows the 'Create database' step in the AWS RDS Management console. The configuration includes:

- DB subnet group:** default
- Public access:** Yes
- VPC security group (firewall):** Choose existing (selected), choosing 'Choose existing VPC security groups'
- Existing VPC security groups:** default
- MySQL details:** MySQL is described as the most popular open source database. It supports database sizes up to 64 TiB, General Purpose, Memory Optimized, and Burstable Performance instance classes, automated backup, point-in-time recovery, and up to 15 Read Replicas per instance.

➤ database has been created.

The screenshot shows the 'Databases' page in the AWS RDS Management console. A message at the top indicates 'Creating database database-1'. The database table shows:

| DB identifier | Status   | Role     | Engine          | Region & ... | Size        | Recommendations |
|---------------|----------|----------|-----------------|--------------|-------------|-----------------|
| database-1    | Creating | Instance | MySQL Community | us-east-1a   | db.t3.micro | -               |

- go to the security group and edit the inbound rules , add MYSQL/Aurora with port number 3306 and save rule.

The screenshot shows the 'ModifyInboundSecurityGroupRules' page in the AWS EC2 console. It displays two inbound rules for a security group:

- sg-0b1c35b6f55936c96**: Type: All traffic, Protocol: All, Port range: All, Source: Custom, Description: sg-01fad5a4a865b204a.
- : Type: MySQL/Aurora, Protocol: TCP, Port range: 3306, Source: Anywhere, Description: 0.0.0.0/0.

Buttons at the bottom include 'Add rule', 'Cancel', 'Preview changes', and a prominent orange 'Save rules' button.

The screenshot shows the 'Launch instance' page in the AWS RDS console. The storage configuration section includes:

- Root volume: 1x 8 GiB gp3 (Not encrypted).
- A message indicates that free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage.
- An info box states: "Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Region in which you launch)."

On the right, the 'Summary' section shows 1 instance, the software image (Amazon Linux 2023 AMI 2023.5.2), and the virtual server type (t2.micro). A 'Launch instance' button is prominently displayed.

- Add the security groups and edit inbound rules, provide mysql port.

The screenshot shows the AWS Management Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroup:group-id=sg-01fad5a4a865b204a>. The left sidebar shows navigation options like EC2 Dashboard, Instances, Images, and Elastic Block Store. The main pane displays the 'Details' section for a security group named 'default'. It shows the security group ID (sg-01fad5a4a865b204a), owner (851725625974), and various rule counts. Below this is the 'Inbound rules' table, which lists three entries:

| Name                   | Security group rule type | IP version | Type | Protocol | Port range |
|------------------------|--------------------------|------------|------|----------|------------|
| sgr-0b1c35b6f55936c96  | All traffic              | -          | All  | All      |            |
| sgr-0dc12b2d84a2c97... | MySQL/Aurora             | IPv4       | TCP  | 3306     |            |
| sgr-05e0e744b6c2443... | SSH                      | IPv4       | TCP  | 22       |            |

- Connect the instance to the terminal.

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\SrijaDulam> cd downloads
PS C:\Users\SrijaDulam\downloads> ssh -i "vinay-1.pem" ec2-user@ec2-98-81-162-75.compute-1.amazonaws.com
The authenticity of host 'ec2-98-81-162-75.compute-1.amazonaws.com (98.81.162.75)' can't be established.
ED25519 key fingerprint is SHA256:sM0vflvhkxEhpdeDEsqBnpsil3EuvpAVAs+MZG/e4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? YES
Warning: Permanently added 'ec2-98-81-162-75.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

 ####_
 ##_ \###_
 ##_ \|##| AL2 End of Life is 2025-06-30.
 ##_ |#| /`-->
 ##_ | /`-->
 ##_ /`--> A newer version of Amazon Linux is available!
 ##_ /`-->
 ##_ /`--> Amazon Linux 2023, GA and supported until 2028-03-15.
 ##_ /`-->
 ##_ /`--> https://aws.amazon.com/linux/amazon-linux-2023/
 ##_ /`-->

2 package(s) needed for security, out of 2 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-29-106 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package kernel.x86_64 0:5.10.224-212.876.amzn2 will be installed
--> Package kernel-tools.x86_64 0:5.10.223-212.873.amzn2 will be updated
--> Package kernel-tools.x86_64 0:5.10.224-212.876.amzn2 will be an update
--> Finished Dependency Resolution
Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
kernel x86_64 5.10.224-212.876.amzn2 amzn2extra-kernel-5.10 34 M

[ec2-user@ip-172-31-29-106 ~]$
```

- Install the Mysql by using command “sudo yum install mysql -y”

```

Verifying : kernel-tools-5.10.224-212.876.amzn2.x86_64 2/3
Verifying : kernel-tools-5.10.223-212.873.amzn2.x86_64 3/3

Installed:
 kernel.x86_64 0:5.10.224-212.876.amzn2

Updated:
 kernel-tools.x86_64 0:5.10.224-212.876.amzn2

Complete!
[ec2-user@ip-172-31-29-106 ~]$ sudo yum install mysql -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Existing lock /var/run/yum.pid: another copy is running as pid 6299.
Another app is currently holding the yum lock; waiting for it to exit...
The other application is: yum
 Memory : 126 M RSS (346 MB VSZ)
 Started: Thu Sep 5 10:22:36 2024 - 00:15 ago
 State : Running, pid: 6299
Another app is currently holding the yum lock; waiting for it to exit...
The other application is: yum
 Memory : 136 M RSS (356 MB VSZ)
 Started: Thu Sep 5 10:22:36 2024 - 00:17 ago
 State : Running, pid: 6299
Another app is currently holding the yum lock; waiting for it to exit...
The other application is: yum
 Memory : 155 M RSS (375 MB VSZ)
 Started: Thu Sep 5 10:22:36 2024 - 00:19 ago
 State : Uninterruptible, pid: 6299
Another app is currently holding the yum lock; waiting for it to exit...
The other application is: yum
 Memory : 175 M RSS (395 MB VSZ)
 Started: Thu Sep 5 10:22:36 2024 - 00:21 ago
 State : Running, pid: 6299
Another app is currently holding the yum lock; waiting for it to exit...
The other application is: yum
 Memory : 188 M RSS (407 MB VSZ)
 Started: Thu Sep 5 10:22:36 2024 - 00:23 ago
 State : Running, pid: 6299
Another app is currently holding the yum lock; waiting for it to exit...

```

- Use command “mysql -h endpoint url of database -u admin -p” to connect to the RDS mysql.
- Provide password and enter.
- Observe that mysql has been connected.

```

--> Package mariadb.x86_64 1:5.5.68-1.amzn2.0.1 will be installed
--> Finished Dependency Resolution
Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
mariadb x86_64 1:5.5.68-1.amzn2.0.1 amzn2-core 8.8 M

Transaction Summary
=====
Install 1 Package

Total download size: 8.8 M
Installed size: 49 M
Downloading packages:
mariadb-5.5.68-1.amzn2.0.1.x86_64.rpm
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
 Installing : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64
 Verifying : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64
 1/1
 1/1

Installed:
 mariadb.x86_64 1:5.5.68-1.amzn2.0.1

Complete!
[ec2-user@ip-172-31-29-106 ~]$ mysql -h database-1.cfayqi0o8ulz.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor Commands end with ; or \g.
Your MySQL connection id is 32
Server version: 8.0.35 Source distribution

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

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

MySQL [(none)]>

```