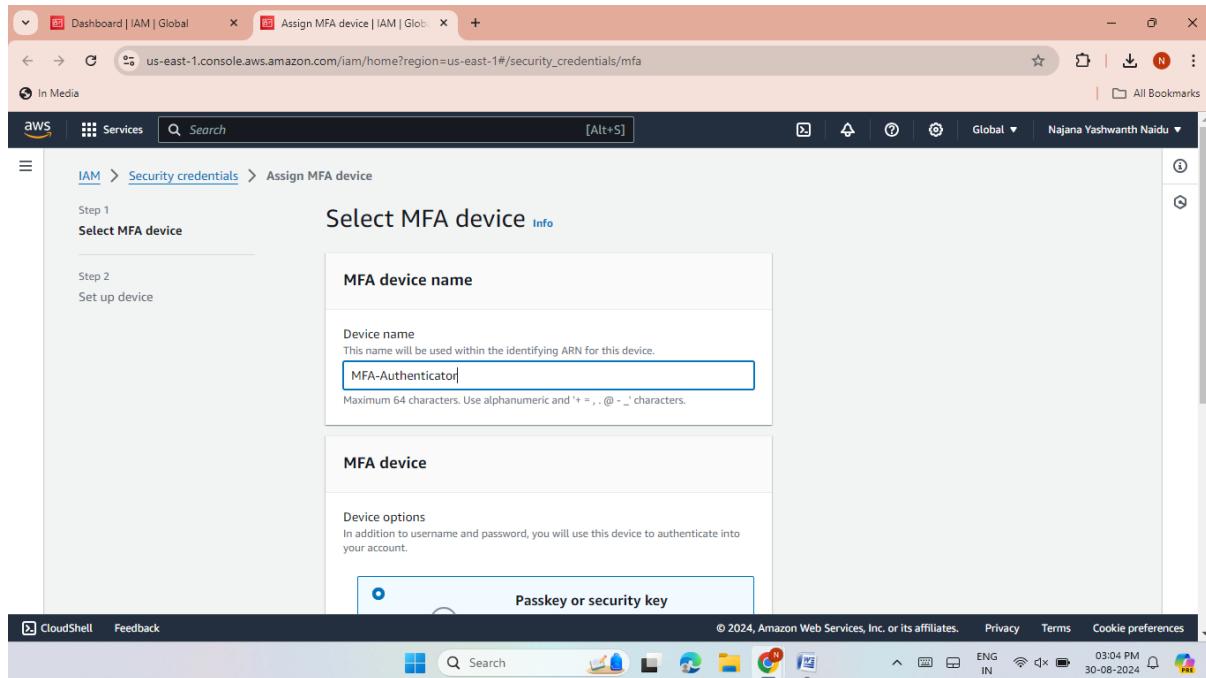


AWS MINI PROJECT

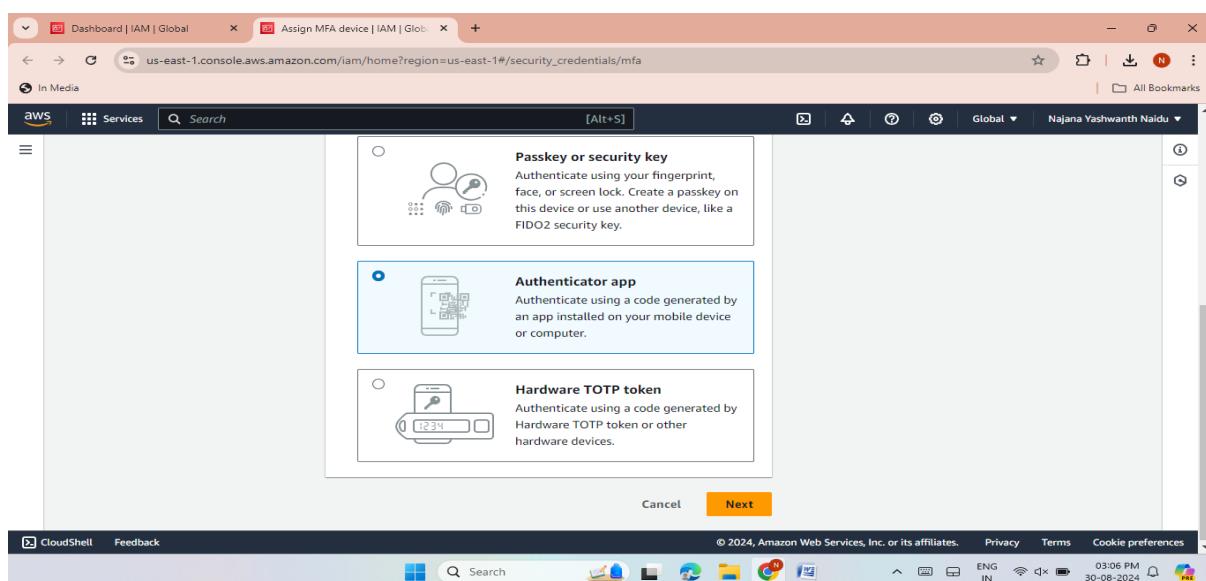
N.Yashwanth Naidu

LAB-1 : IAM HANDS-ON

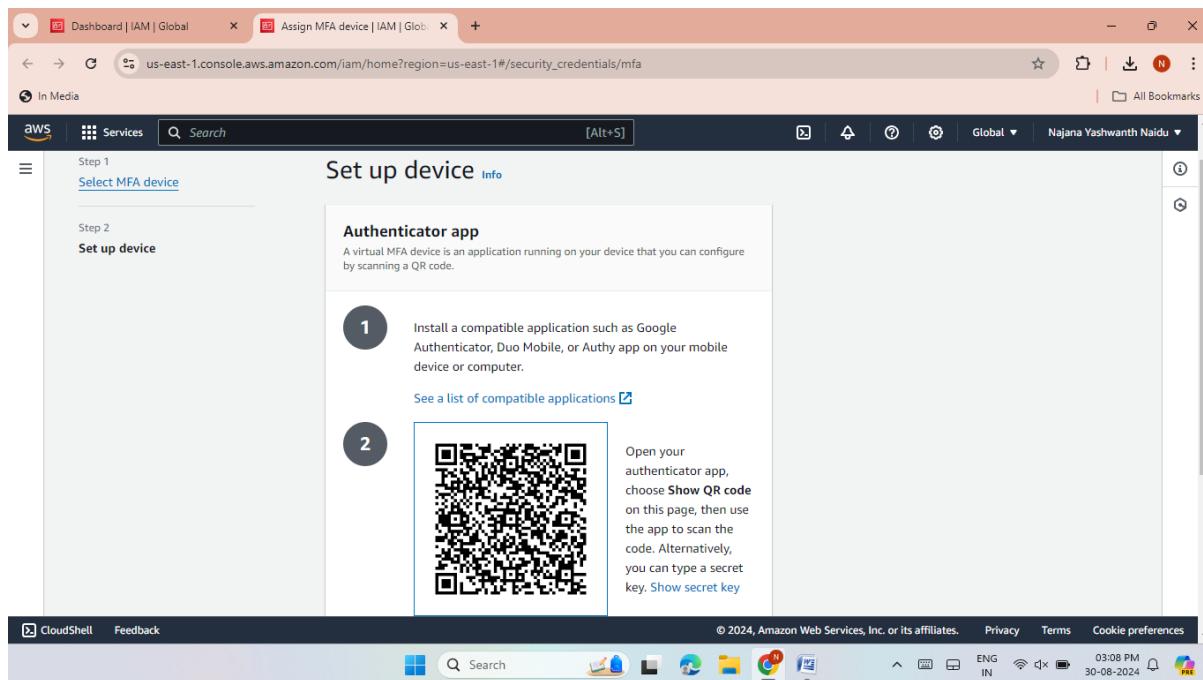
- Set the MFA for root user.
- Choose the device name.



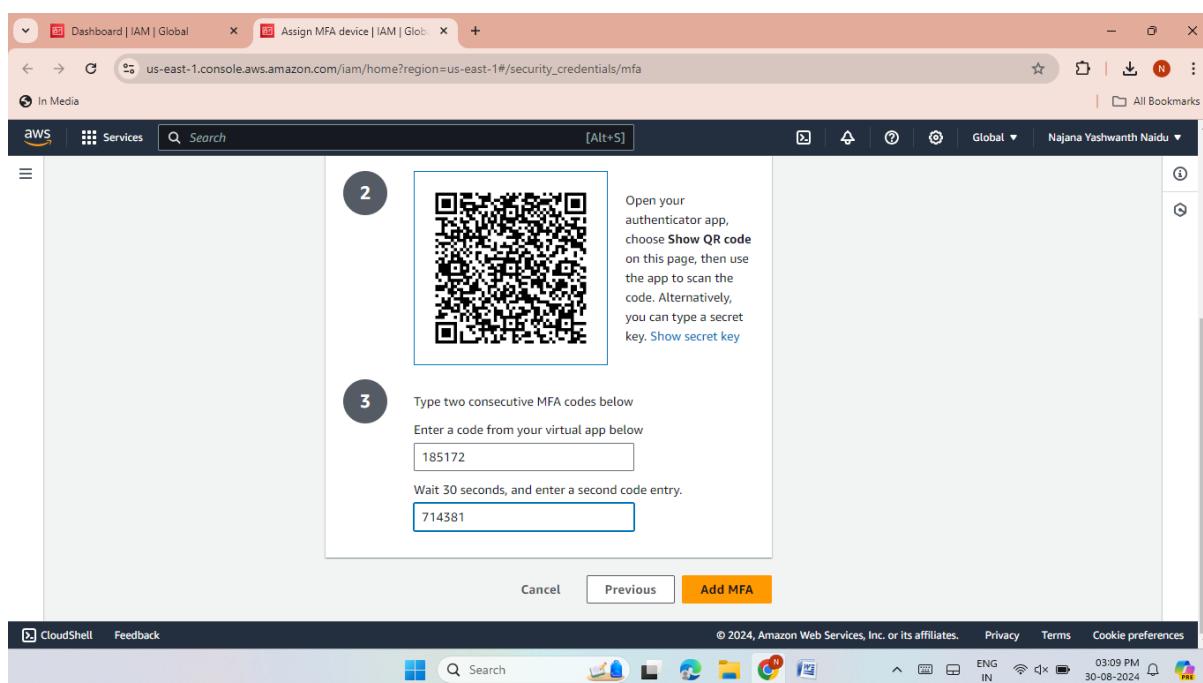
- Select the device option. I took the authenticator app.



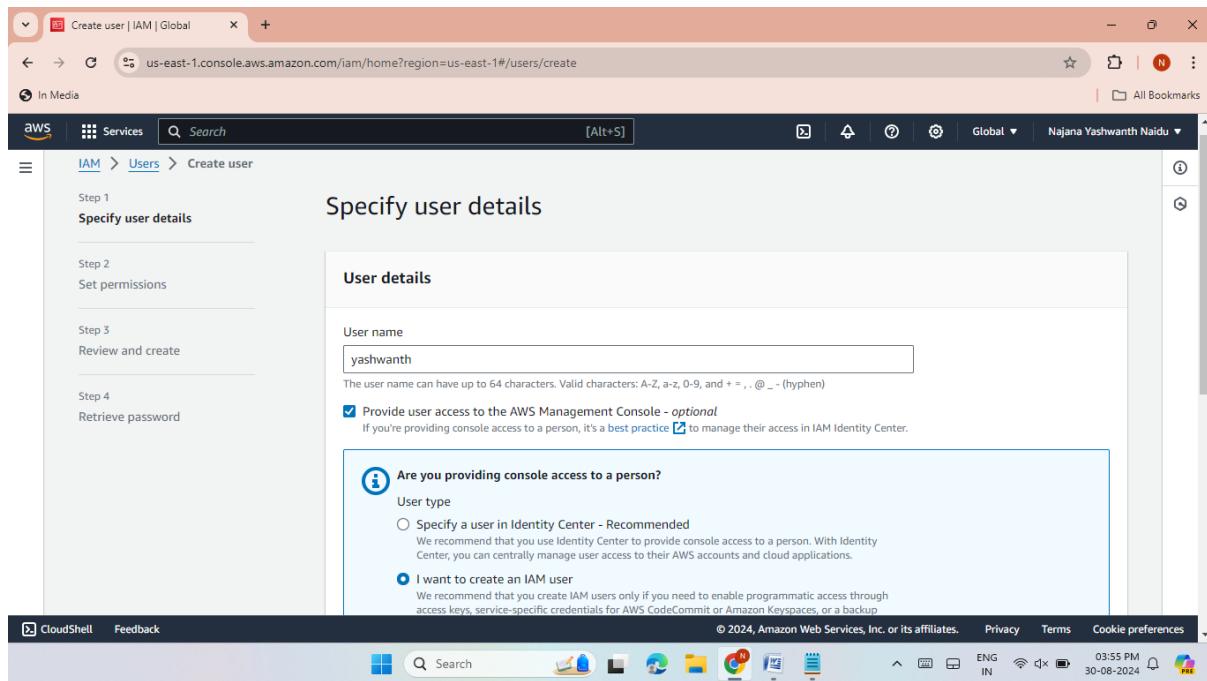
- Scan the QR code from your mobile app like Google authenticator,Duo mobile or Authy app.



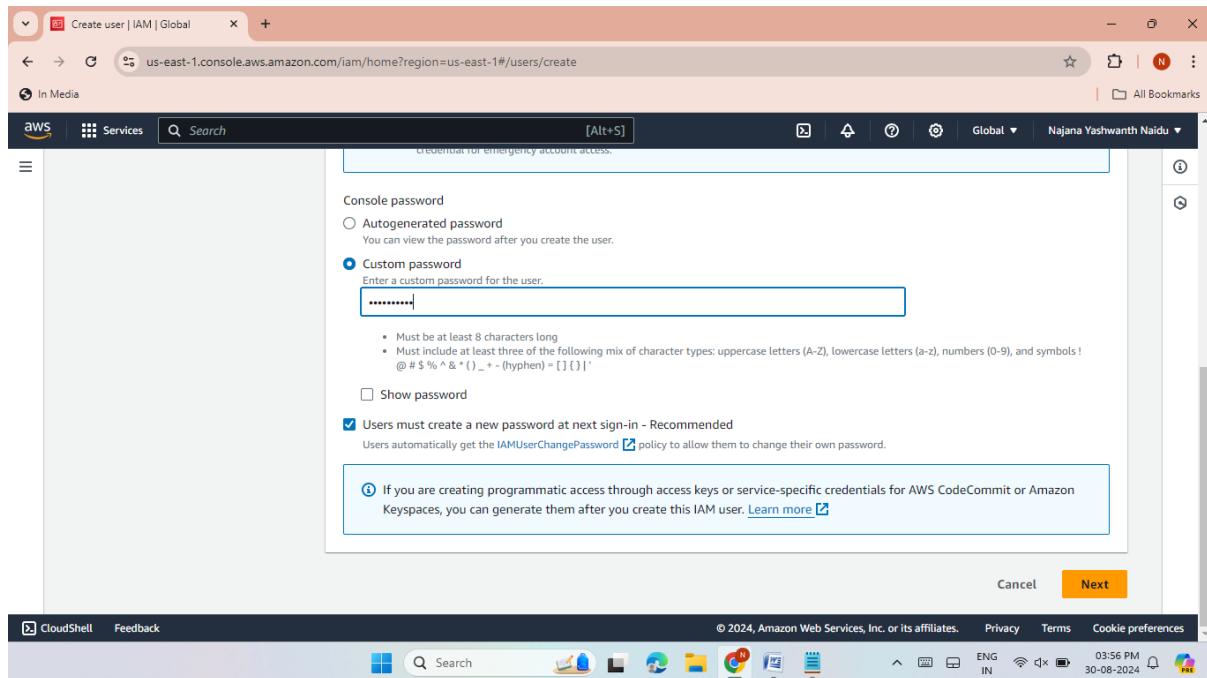
- First enter the code from your mobile app then after 30 sec enter the second code.
- Then click add MFA, Then the MFA is successfully created.



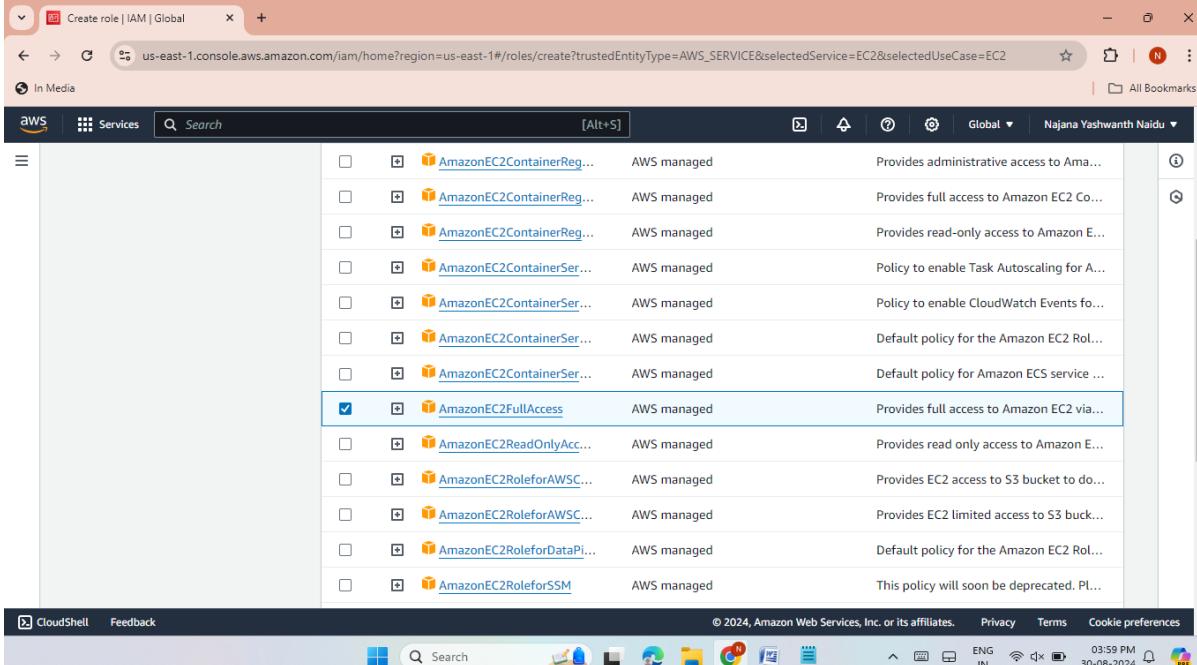
- Create a new user with console access and check its default permissions.



- Create a custom password for your IAM login purpose.

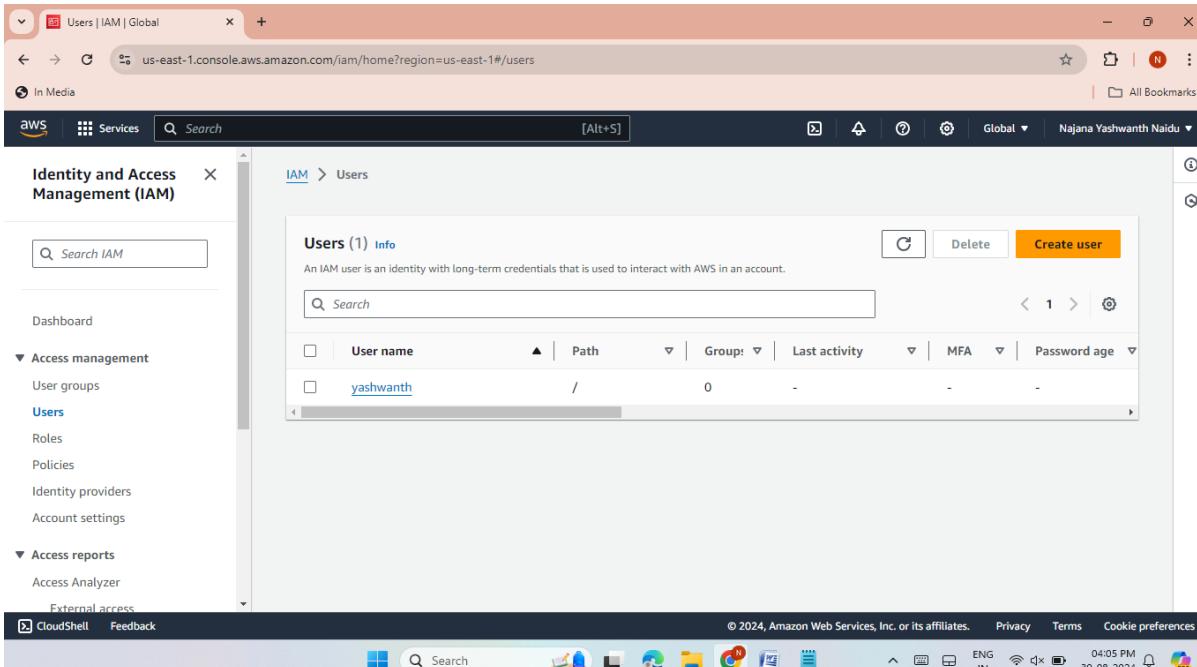


- Assign only ec2 permissions to this user and navigate to different services with this user access.



The screenshot shows the AWS IAM Roles page. A specific policy, 'AmazonEC2FullAccess', is highlighted with a blue selection bar. The policy details are visible: 'Provides full access to Amazon EC2 via...'. Other policies listed include 'AmazonEC2ReadOnlyAccess', 'AmazonEC2RoleforAWS...', 'AmazonEC2RoleforAWSC...', 'AmazonEC2RoleforDataPi...', and 'AmazonEC2RoleforSSM'.

- Here the IAM user was created.



The screenshot shows the AWS IAM Users page. A single user, 'yashwanth', is listed in the 'Users (1) Info' section. The user details show they have no groups, 0 last activity, and no MFA or password age information. The left sidebar shows the navigation menu for IAM, including 'Dashboard', 'Access management' (with 'Users' selected), 'Policies', 'Identity providers', 'Account settings', and 'Access reports'.

- Instance was launched with the name yash-instance.

The screenshot shows the AWS EC2 Instances page. The left sidebar has sections for EC2 Dashboard, EC2 Global View, Events, Console-to-Code Preview, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, and Reservations. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
yash-instance	i-089db0a1fdd16d5ee	Running	t2.micro	Initializing	View alarms	us-east-1d
new-ec2	i-0c9ba77b4b0de1391	Terminated	t2.micro	-	View alarms	us-east-1d

A modal window titled "Select an instance" is open at the bottom, listing the same two instances.

- While launching the S3 bucket it shows fail to create bucket.
- We Can't work with any other services except ec2.

The screenshot shows the AWS S3 Create Bucket page. The left sidebar has sections for Services (with "Enable" selected), Advanced settings, and a note about configuring additional bucket settings. A prominent error message in a red-bordered box states:

Failed to create bucket
To create a bucket, the s3:CreateBucket permission is required.

Below the message are links to "View your permissions in the IAM console" and "Identity and Access Management in Amazon S3". There is also a "Diagnose with Amazon Q" button. At the bottom are "Cancel" and "Create bucket" buttons.

- Provide then Administrator Access to this user.

The screenshot shows the 'Add permissions' step in the AWS IAM console. In the 'Permissions options' section, the 'Attach policies directly' option is selected. Below it, the 'Permissions policies' table lists one policy: 'AdministratorAccess' (AWS managed - job function). The table includes columns for Policy name, Type, and Attached entities.

Permissions policies (1/1226)		
	Type	Attached entities
<input checked="" type="checkbox"/> AdministratorAccess	AWS managed - job function	2

- Click on add permissions.

The screenshot shows the 'Review' step in the IAM 'Add permissions' wizard. It displays the 'User details' (User name: yashwanth) and 'Permissions summary' (1 policy: 'AdministratorAccess'). At the bottom right, the 'Add permissions' button is highlighted in orange.

- Here the remaining services also work with the Admin permissions.

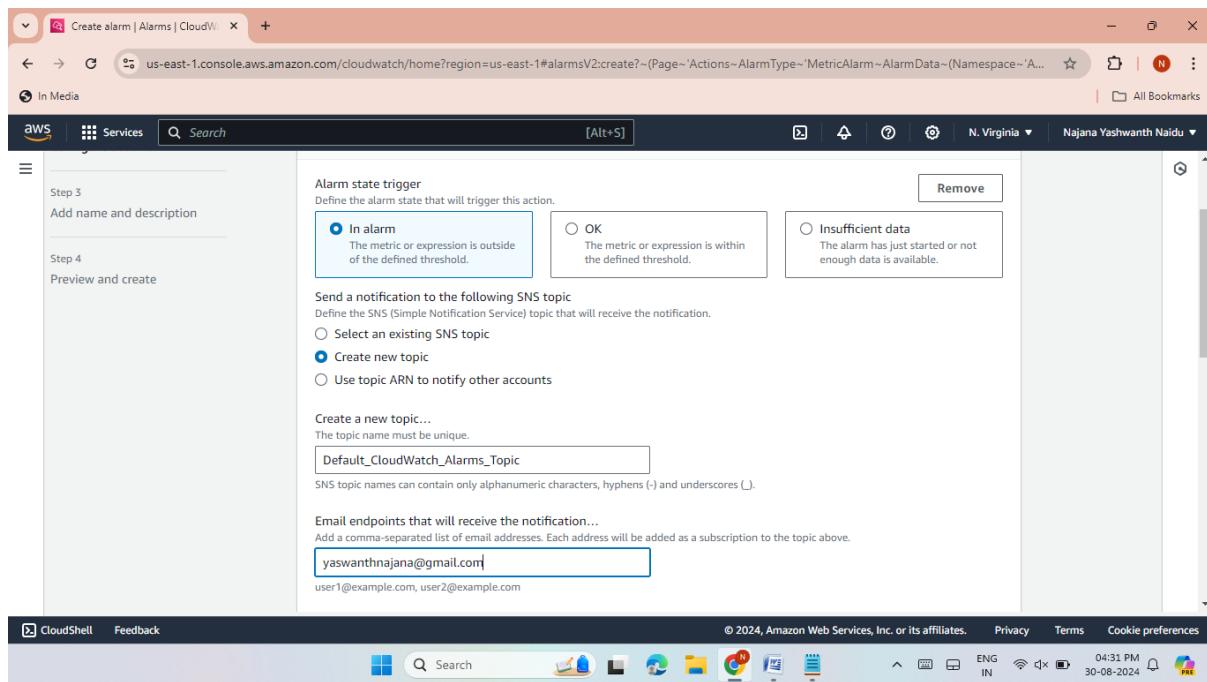
The screenshot shows the AWS S3 buckets page. A green success banner at the top states: "Successfully created bucket 'newbucket-user'. To upload files and folders, or to configure additional bucket settings, choose View details." Below the banner, there's an "Account snapshot - updated every 24 hours" section with a "View Storage Lens dashboard" button. The main table lists two buckets: "elasticbeanstalk-us-east-1-637423323663" and "newbucket-user". Both buckets are in the "General purpose buckets" category, located in the "US East (N. Virginia) us-east-1" region. The "Create bucket" button is visible at the top right of the table area.

LAB-2 : BILLING ALARM

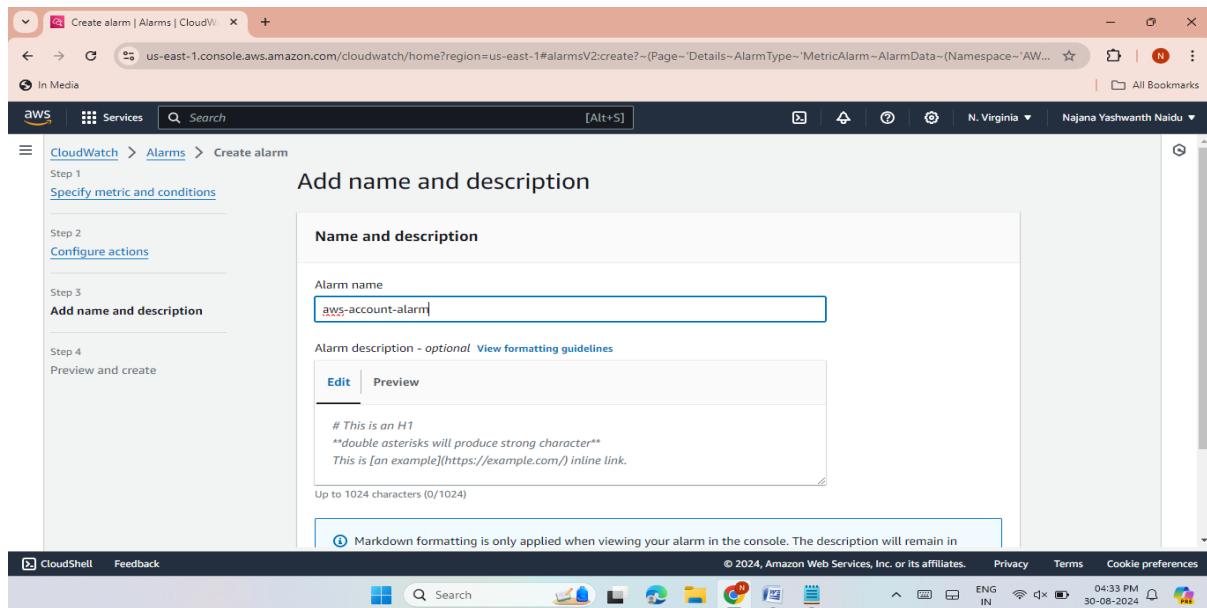
- Setup the billing alarm for your account to get a notification whenever you cross the billing threshold.

The screenshot shows the "Create alarm" wizard in the AWS CloudWatch Metrics console. The left sidebar shows steps: Step 2 (Configure actions), Step 3 (Add name and description), and Step 4 (Preview and create). The main area is titled "Metric" and displays a graph for "EstimatedCharges" over the period from 08/28 to 08/30. The graph shows a blue line for "EstimatedCharges" and a red horizontal line at the value "5". A legend indicates "EstimatedCharges". On the right, configuration fields are shown: Namespace "AWS/Billing", Metric name "EstimatedCharges", Currency "USD", Statistic "Maximum", and Period "1 hour". The status bar at the bottom shows the date as 30-08-2024 and time as 04:30 PM.

- Select the SNS(simple notification service).
- Enter the email for receiving the notifications.



➤ Enter the name of Alarm.



- Here the Billing Alarm is successfully created.

The screenshot shows the AWS CloudWatch Alarms interface. A green banner at the top indicates "Successfully created alarm aws-account-alarm." The main table lists one alarm:

Name	State	Last state update (UTC)	Conditions
aws-account-alarm	Insufficient data	2024-08-30 11:04:04	EstimatedCharges > 5 for 1 datapoints within 1 hour

LAB-3 : S3 bucket

Step 1

- Create an S3 bucket make sure to give it a unique name.
- Upload some test files/folder in the bucket.

The screenshot shows the AWS S3 Upload objects interface. A message at the top says "Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more." Below is a drag-and-drop area and a table of uploaded files:

Name	Folder	Type
EC2 Basics.pptx	-	application/vnd.openxmlformats-officedocument.presentationml.presentation
Screenshot 2024-08-30 145957.png	-	image/png

- Successfully uploaded.

The screenshot shows a browser window titled "Upload objects - S3 bucket yashwanth" with the URL "us-east-1.console.aws.amazon.com/s3/upload/yashwanth-bucket?region=us-east-1&bucketType=general". The main content area has a green header bar with the message "Upload succeeded" and "View details below." Below this, there is a summary table:

Destination	Succeeded	Failed
s3://yashwanth-bucket	2 files, 277.4 KB (100.00%)	0 files, 0 B (0%)

Below the table, there are two tabs: "Files and folders" (which is selected) and "Configuration". Under "Files and folders", there is a table titled "Files and folders (2 Total, 277.4 KB)" showing the uploaded files:

Name	Folder	Type	Size	Status	Error
EC2 Basics.p...	-	application/v...	114.8 KB	Succeeded	-
Screenshot 2...	-	image/png	162.6 KB	Succeeded	-

At the bottom of the browser window, there is a toolbar with icons for CloudShell, Feedback, and other browser controls. The status bar at the bottom right shows the date and time as 04:45 PM 30-08-2024.

- Access the file over the browser using its URL but It should give error.

The screenshot shows a browser window with the URL "yashwanth-buckets3.amazonaws.com/Screenshot+2024-08-30+145957.png". The page content is as follows:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<Error>
<Code>AccessDenied</Code>
<Message>Access Denied</Message>
<RequestId>T92356FYSATMH10</RequestId>
<HostId>haJlbSeK9yCOeYgzm19vN916Bxr+TS1vgNlyy077L/XILYD3CQECABeIwg1/VquTvyY2uKuOTbI=</HostId>
</Error>
```

At the bottom of the browser window, there is a toolbar with icons for CloudShell, Feedback, and other browser controls. The status bar at the bottom right shows the date and time as 04:49 PM 30-08-2024.

- Check the permissions of the file and the bucket
- First give permissions to the bucket. In permissions tab disabled block public access.

The screenshot shows the AWS S3 Bucket Permissions Overview page for the 'yashwanth-bucket'. A green success message at the top states 'Successfully edited Object Ownership.' Below it, the 'Permissions' tab is selected. Under the 'Permissions overview' section, there is a note about access findings and a link to 'View analyzer for us-east-1'. At the bottom of this section is a 'Block public access (bucket settings)' button with an 'Edit' link. The browser's address bar shows the URL: 'us-east-1.console.aws.amazon.com/s3/buckets/yashwanth-bucket?region=us-east-1&bucketType=general&tab=permissions'.

- Then click edit and give public access in Access control list (ACLs).

The screenshot shows the AWS S3 Bucket Permissions Overview page for the 'yashwanth-bucket'. A green success message at the top states 'Successfully edited access control list.' Below it, the 'Permissions' tab is selected. Under the 'Permissions overview' section, there is a note about access findings and a link to 'View analyzer for us-east-1'. At the bottom of this section is a 'Block public access (bucket settings)' button with an 'Edit' link. The browser's address bar shows the URL: 'us-east-1.console.aws.amazon.com/s3/buckets/yashwanth-bucket?region=us-east-1&bucketType=general&tab=permissions'.

- Make the object public and access it again over the browser.

The make public action enables public read access in the object access control list (ACL) settings. [Learn more](#)

Specified objects

Name	Type	Last modified	Size
EC2 Basics.pptx	pptx	August 30, 2024, 16:45:24 (UTC+05:30)	114.8 KB
Screenshot 2024-08-30 145957.png	png	August 30, 2024, 16:45:26 (UTC+05:30)	162.6 KB

Cancel **Make public**

- Here Successfully edited public access. Now we can access the file.

Successfully edited public access

View details below.

Make public: status

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

Source	Successfully edited public access	Failed to edit public access
s3://yashwanth-bucket	2 objects, 277.4 KB	0 objects

Failed to edit public access (0)

https://us-east-1.console.aws.amazon.com/console/home?region=us-east-1

Step 2

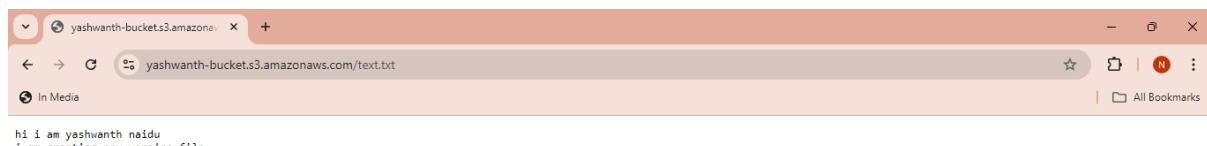
- Enable versioning of the excited bucket.

The screenshot shows the AWS S3 Bucket Overview page for the 'yashwanth-bucket'. 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 (US East (N. Virginia) us-east-1), Amazon Resource Name (ARN) (arn:aws:s3:::yashwanth-bucket), and Creation date (August 30, 2024, 16:41:49 (UTC+05:30)). The 'Bucket Versioning' section shows that it is currently 'Enabled'. There is a note about Multi-factor authentication (MFA) delete, stating that an additional layer of security is required for changing Bucket Versioning settings and permanently deleting object versions. The AWS navigation bar at the bottom includes CloudShell, Feedback, and various icons for search and other services.

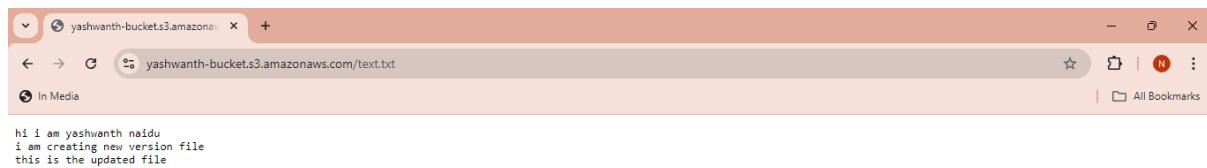
- Create a text file and upload the updated version to the bucket.

The screenshot shows the AWS S3 'Upload objects' page for the 'yashwanth-bucket'. A green success message at the top says 'Upload succeeded' and 'View details below.' Below this, the 'Summary' section shows the destination 's3://yashwanth-bucket' with one file uploaded successfully (1 file, 55.0 B (100.00%)). The 'Files and folders' section lists a single file named 'text.txt' with a size of 55.0 B and a status of 'Succeeded'. The AWS navigation bar at the bottom includes CloudShell, Feedback, and various icons for search and other services.

- Check the content of the current file.



- Here the updated version of the text file.



Step 3

- Delete the text file from the bucket.

The screenshot shows the AWS S3 console with a green success message: "Successfully deleted objects". It details one object deleted successfully (81.0 B) and zero objects failed to delete. Below this, there's a "Failed to delete" section which is currently empty. The browser status bar at the bottom indicates it's from August 30, 2024, at 05:30 PM.

- In objects enable show versions then Check the deleted file.
- Delete the delete marker file.

The screenshot shows the AWS S3 console with the "Show versions" option selected. It lists three objects: a PNG file named "145957.png" and two "text.txt" files. One "text.txt" file is a "Delete marker" (version ID: yM6XopZPb...), while the other two are standard text files (version IDs: qoVx4za08Z... and l3kuudok045...). The browser status bar at the bottom indicates it's from August 30, 2024, at 05:38 PM.

- It recovers the deleted file by versioning.
- It will restore the last updated file.

The screenshot shows the AWS S3 console interface. The URL in the address bar is us-east-1.console.aws.amazon.com/s3/buckets/yashwanth-bucket?region=us-east-1&bucketType=general&tab=objects. The page displays the 'yashwanth-bucket' details, including its creation date (August 30, 2024) and a single object named 'text.txt' which was last modified on the same day at 17:28:55 (UTC+05:30). The object is 81.0 B in size and has a storage class of Standard.

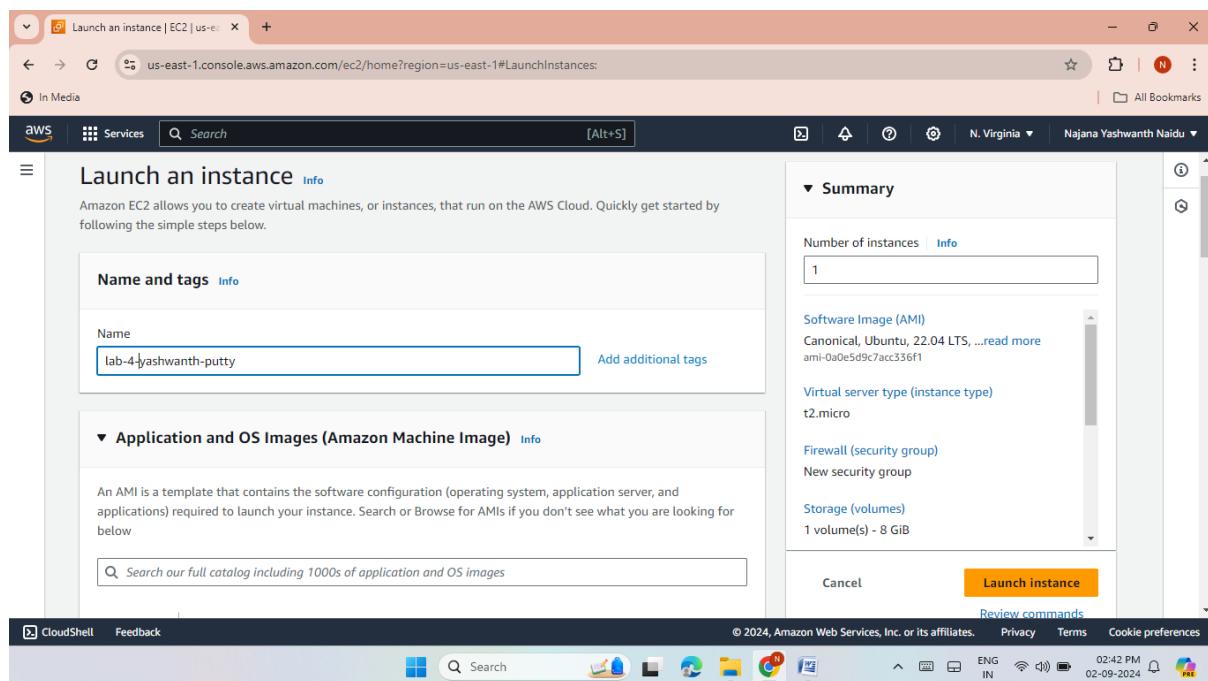
- Then browse the URL it will display the content of the file.

The screenshot shows a web browser window displaying the contents of the 'text.txt' file from the 'yashwanth-bucket'. The URL in the address bar is yashwanth-bucket.s3.amazonaws.com/text.txt. The page content is as follows:

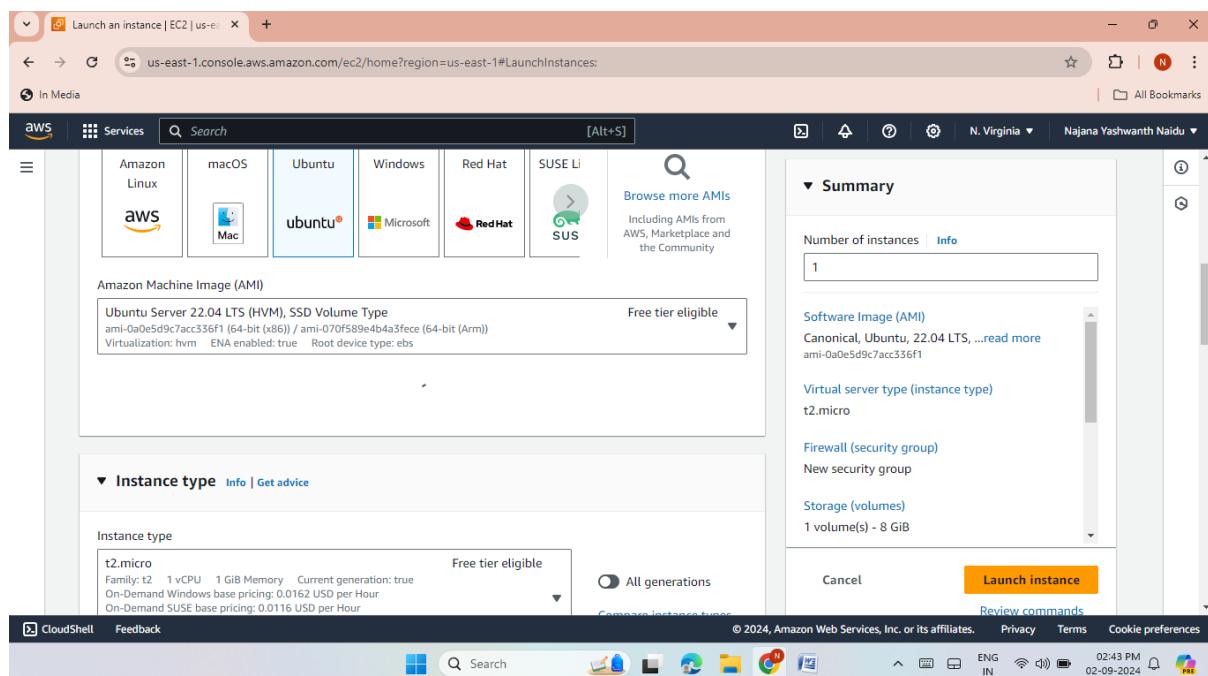
```
hi i am yashwanth naidu
i am creating new version file
currently this is the updated file
```

Lab 4 : EC2 INSTANCE

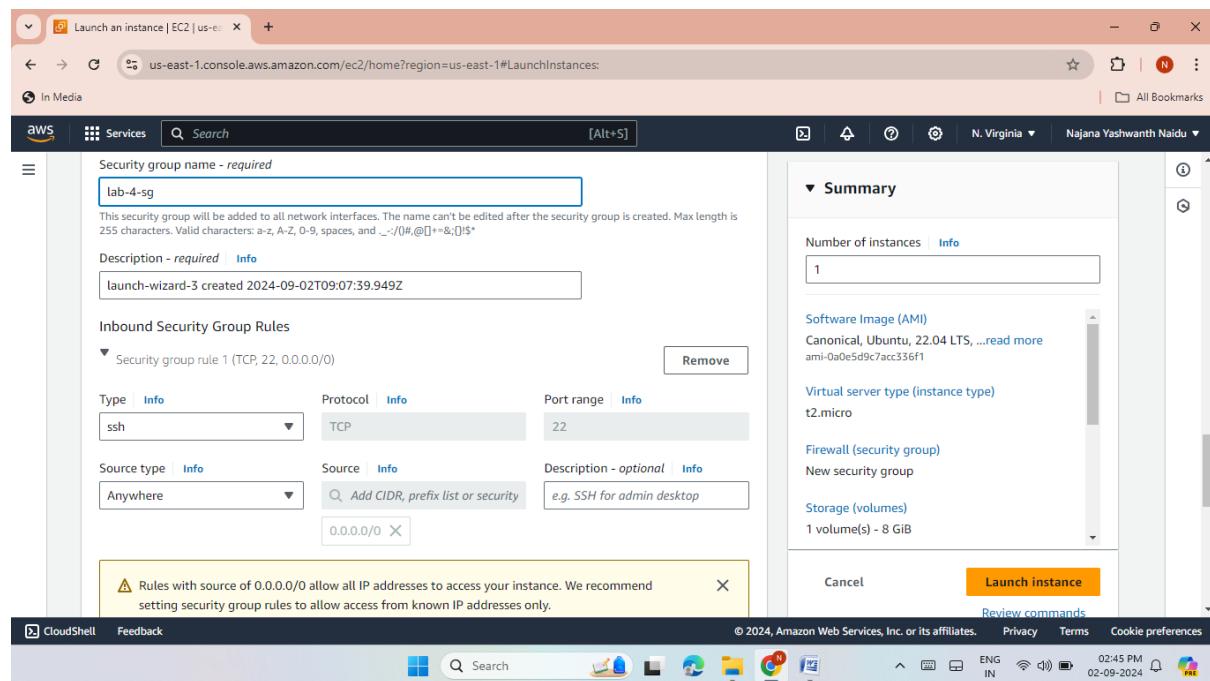
- Go to EC2 then Launch the EC-2 instance.



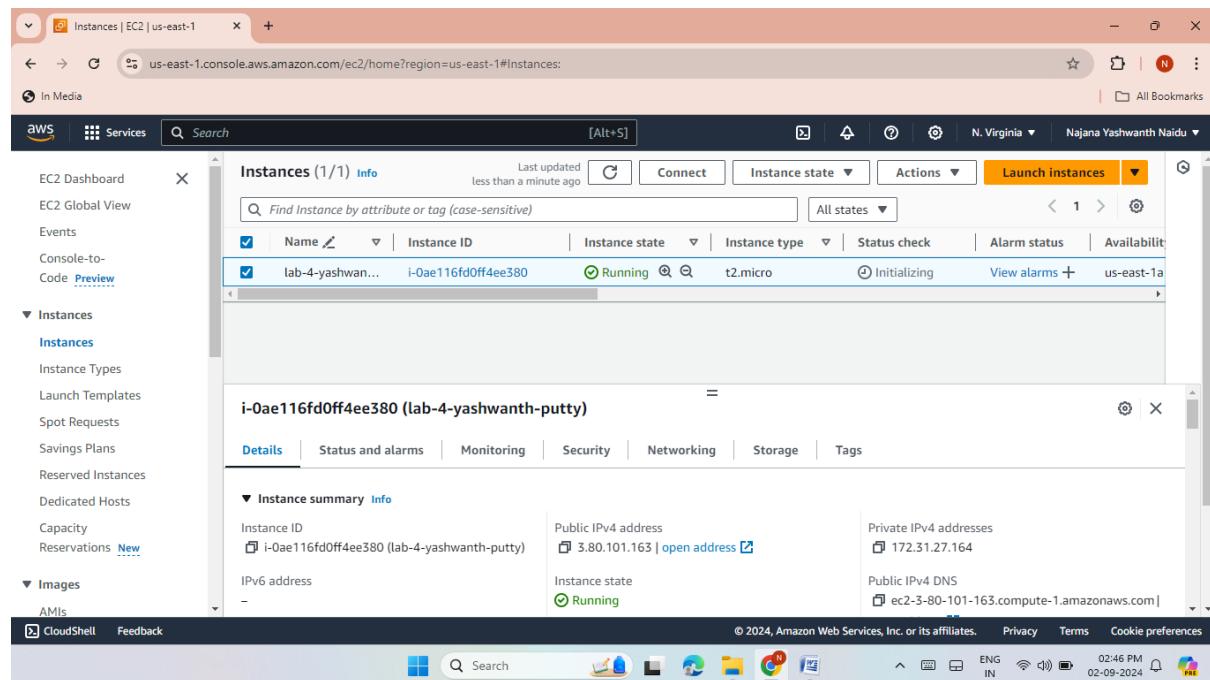
- Give the instance type t2.micro with the Ubuntu server.(free tier)



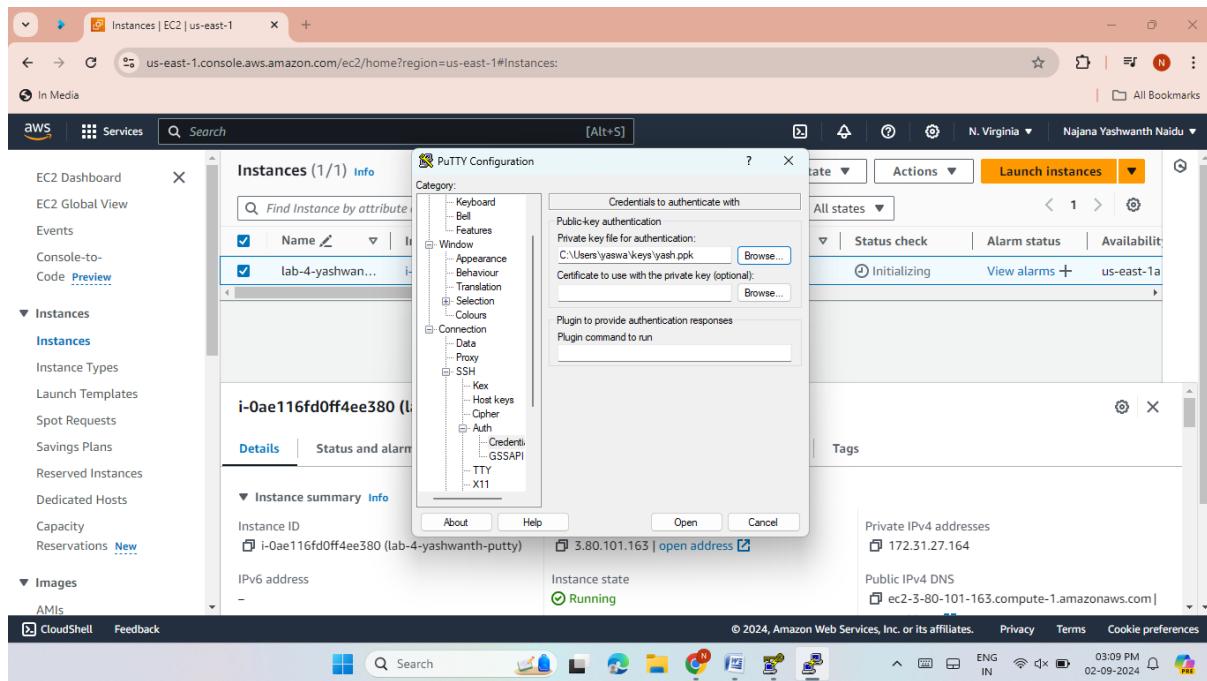
- Allow the SSH port in the security group so that you can access it from GitBash or putty etc.



- Here the instance was launched.



- Access this instance from my machine using putty with help of SSH.



- Now I have access instance using PPK file in putty software.

```
ubuntu@ip-172-31-27-164:~$ 
ubuntu@ip-172-31-27-164:~$ login as: ubuntu
ubuntu@ip-172-31-27-164:~$ Authenticating with public key "imported-openssh-key"
ubuntu@ip-172-31-27-164:~$ Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1022-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Mon Sep 2 09:39:49 UTC 2024

System load: 0.0          Processes:      97
Usage of /: 20.7% of 7.87GB   Users logged in: 0
Memory usage: 20%           IPv4 address for eth0: 172.31.27.164
Swap usage: 0k

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-27-164:~$ 
```

Lab 5 : SECURITY GROUPS

- Create a new Security Group and name it as “mynewsg”.

The screenshot shows the 'Create security group' page in the AWS EC2 console. The 'Basic details' section is visible, containing fields for the security group name ('mvnewsg'), description ('Allows SSH access to developers'), and VPC ('vpc-05f2f9c35f844e371'). The browser address bar shows the URL: `us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateSecurityGroup`.

- Check the default rules in this Security Group.

The screenshot shows the 'Inbound rules' and 'Outbound rules' sections for the security group 'mvnewsg'. Both sections are currently empty, indicating no rules have been defined. The browser address bar shows the URL: `us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateSecurityGroup`.

- Allowing inbound Port HTTP-80 and SSH-22 from my IP address.
- Select a range (IP/28) so that any changes in my dynamic IP will not impact the rule.

The screenshot shows the AWS EC2 console with the 'CreateSecurityGroup' page open. On the left, the navigation pane is visible with sections like EC2 Dashboard, EC2 Global View, Events, and Instances. The main area displays two inbound rules:

- Inbound rule 1:** Type: SSH, Protocol: TCP, Port range: 22. Source type: Custom, Source: 54.86.218.55/28.
- Inbound rule 2:** Type: HTTP, Protocol: TCP, Port range: 80. Source type: Custom, Source: 54.86.218.55/28.

- The Security Group was created successfully.

The screenshot shows the AWS EC2 console with the 'SecurityGroup' page open. The security group 'sg-02c8b8b0cb0135bc6 - mvnewsg' has been successfully created. The details section shows:

- Security group name: mvnewsg
- Security group ID: sg-02c8b8b0cb0135bc6
- Description: it is for ec2-instance
- VPC ID: vpc-05f2f9c35f844e371
- Owner: 637423323663
- Inbound rules count: 2 Permission entries
- Outbound rules count: 0 Permission entries

- Attach this new Security Group to my existing EC2 Instance.
- Go to EC2 Dashboard.
- Click Instances and select the instance you want to modify.
- Under Actions, select Networking → Change Security Groups.

The screenshot shows the 'Change security groups' interface for an EC2 instance. In the 'Associated security groups' section, a search bar contains 'sg-02c8b8b0cb0135bc6'. An 'Add security group' button is visible. Below the search bar, it says 'Security groups associated with the network interface (eni-05de34acf5a9f0fae)'. The interface includes tabs for 'CloudShell' and 'Feedback' at the bottom.

- Attached to this instance.

The screenshot shows the EC2 Instances dashboard. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, and Instances. The Instances section is expanded, showing sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, and Reservations. The main pane displays 'Instances (1/2)' with one item listed: 'lab-5' (i-0ad19de154c2c79fe), which is 'Running'. Below the instance list, a detailed view for 'i-0ad19de154c2c79fe (lab-5)' shows its security groups: 'sg-02c8b8b0cb0135bc6 (mvnewsg)' and 'sg-0abde9ca120a125b1 (launch-wizard-1)'. The interface includes tabs for 'CloudShell' and 'Feedback' at the bottom.

- Try to access the server with help of SSH.
- It successfully accessed.

```

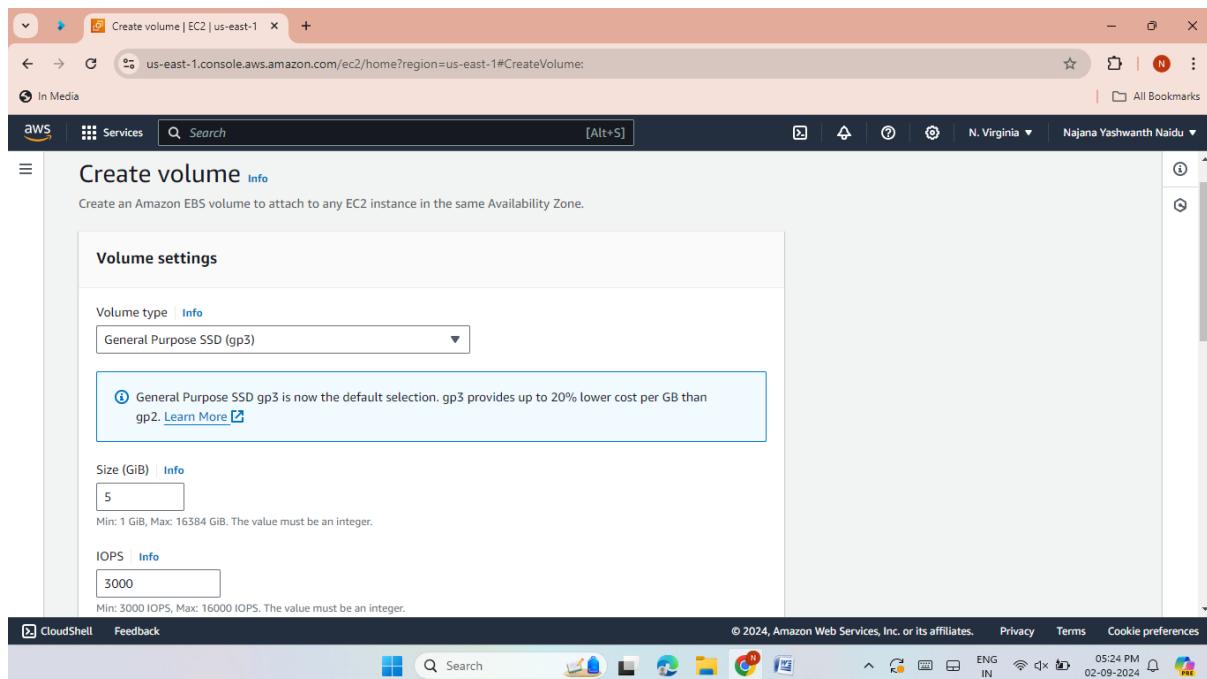
ec2-user@ip-172-31-94-255:~$ login as: ec2-user
Authenticating with public key "imported-openssh-key"
Amazon Linux 2
AL2 End of Life is 2025-06-30.
A newer version of Amazon Linux is available!
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-172-31-94-255 ~]$ ls
[ec2-user@ip-172-31-94-255 ~]$ 

```

The screenshot shows a Windows taskbar at the bottom with various icons. The CloudShell window has a title bar with the AWS logo and a status bar at the bottom right showing the date and time.

LAB 6 : VOLUMES AND SNAPSHOTS

- Create one 5GB volume.
- Select the same Availability Zone as your running EC2 instance.



- Here the volume is Successfully created.

The screenshot shows the AWS EC2 Volumes page. A green banner at the top says "Successfully created volume vol-0f6799f24f6650978." Below it, a table lists two volumes:

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID	Create
lab-6-volume	vol-0f6799f24f6650978	gp3	5 GiB	3000	125	-	2024/
-	vol-0c541967f7a1f064b	gp2	8 GiB	100	-	snap-07e55c5...	2024/

Below the table, a detailed view for the first volume is shown:

Volume ID: vol-0f6799f24f6650978 (lab-6-volume)

Volume ID vol-0f6799f24f6650978 (lab-6-volume)	Size 5 GiB	Type gp3	Volume status Okay
AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more	Volume state Available	IOPS 3000	Throughput 125

- Attach it with the running ec2 instance.
- Select the volume from the list and click Actions → Attach Volume.

The screenshot shows the "Attach volume" page for the volume "vol-0f6799f24f6650978". The URL is "us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AttachVolume:volumeld=vol-0f6799f24f6650978".

The page displays the "Basic details" section:

- Volume ID: vol-0f6799f24f6650978 (lab-6-volume)
- Availability Zone: us-east-1d
- Instance: i-03e66d2e4b05c1747
- Device name: /dev/sdf

Below the form, a note says: "Only instances in the same Availability Zone as the selected volume are displayed."

- Put some data in this volume like some testing files
- Connect the ec2 instance in the terminal using ssh command.
- Run the following command to check if the volume is attached OR not.
- Lsblk
- Create directory with the mkdir command (/mnt/myvolume)
- Then mount the path with the command Sudo mount /dev/xvdf /mnt/myvolume.

```

ec2-user@ip-172-31-93-193:~$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda 202:0 0 8G 0 disk
└─xvda1 202:1 0 8G 0 part /
xvdf 202:80 0 3G 0 disk
[ec2-user@ip-172-31-93-193 ~]$ sudo file -s /dev/xvdf
/dev/xvdf: data
[ec2-user@ip-172-31-93-193 ~]$ sudo mkdir /mnt/myvolume
[ec2-user@ip-172-31-93-193 ~]$ sudo mkfs -t ext4 /dev/xvdf
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=4096, Stripe width=0 blocks
32768 inodes, 1310720 blocks
65536 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1342177280
40 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736
Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[ec2-user@ip-172-31-93-193 ~]$ sudo mount /dev/xvdf /mnt/myvolume
[ec2-user@ip-172-31-93-193 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs          4.6G   47M  4.5G  1% /dev/shm
tmpfs          4.77M  0  4.77M  0% /run
tmpfs          4.77M  464K  4.76M  1% /run/pts
tmpfs          4.77M  0  4.77M  0% /sys/fs/cgroup
/dev/xvda1     8.0G  1.8G  6.3G  23% /
tmpfs          96M   0  96M  0% /run/user/1000
/dev/xvdf      4.8G  24K  4.66G  1% /mnt/myvolume
tmpfs          96M   0  96M  0% /run/user/0
[ec2-user@ip-172-31-93-193 ~]$ |

```

The screenshot shows a terminal window on an Amazon Linux 2023 desktop environment. The terminal output details the creation of a new ext4 file system on /dev/xvdf and its subsequent mounting at /mnt/myvolume. The desktop interface includes a taskbar with icons for various applications like File Explorer, Google Chrome, and Microsoft Word, along with system status indicators for battery, signal, and time.

- Put Some Data in the myvolume directoery.
- Create a some text files using touch command.

```

ec2-user@ip-172-31-93-193:/mnt/myvolume$ cd /mnt/myvolume
[ec2-user@ip-172-31-93-193 myvolume]$ touch file1.txt file2.txt
touch: cannot touch 'file1.txt': Permission denied
touch: cannot touch 'file2.txt': Permission denied
[ec2-user@ip-172-31-93-193 myvolume]$ sudo touch file1.txt file2.txt
[ec2-user@ip-172-31-93-193 myvolume]$ ls
file1.txt  file2.txt  lost+found
[ec2-user@ip-172-31-93-193 myvolume]$ |

```

This terminal session demonstrates attempting to create files directly from a non-root user, which fails due to permission denial. However, using sudo grants the necessary permissions, allowing the creation of 'file1.txt' and 'file2.txt'. The final command, indicated by '|', suggests piping the output of the 'ls' command to another command or application.

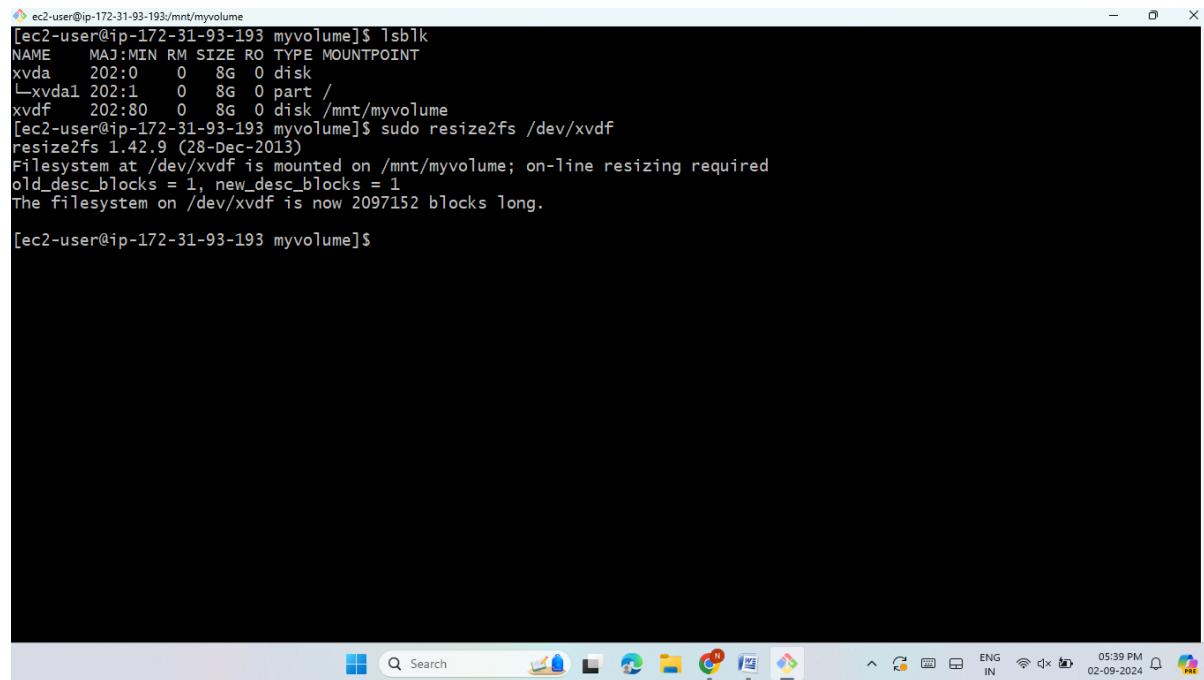
- Increase the size of the this volume it 8GB.
- Select the volume created (5 GiB) and click Actions → Modify Volume.
- Change the size to 8 GiB and click Modify.

The screenshot shows the AWS EC2 Modify Volume interface. The volume ID is vol-0f6799f24f6650978. The volume type is General Purpose SSD (gp3). The current size is 5 and is being changed to 8 GiB. The IOPS value is set to 3000. The page includes standard AWS navigation and status bars at the top and bottom.

- Inside the EC2 instance, run the following command to ensure the volume size is updated →lsblk.

```
ec2-user@ip-172-31-93-193:/mnt/myvolume
[ec2-user@ip-172-31-93-193 myvolume]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   8G  0 disk
└─xvda1 202:1    0   8G  0 part /
xvdf    202:80   0   8G  0 disk /mnt/myvolume
[ec2-user@ip-172-31-93-193 myvolume]$
```

- Extend the size of this volume inside the Linux machine
- The following command is sudo resize2fs /dev/xvdf.



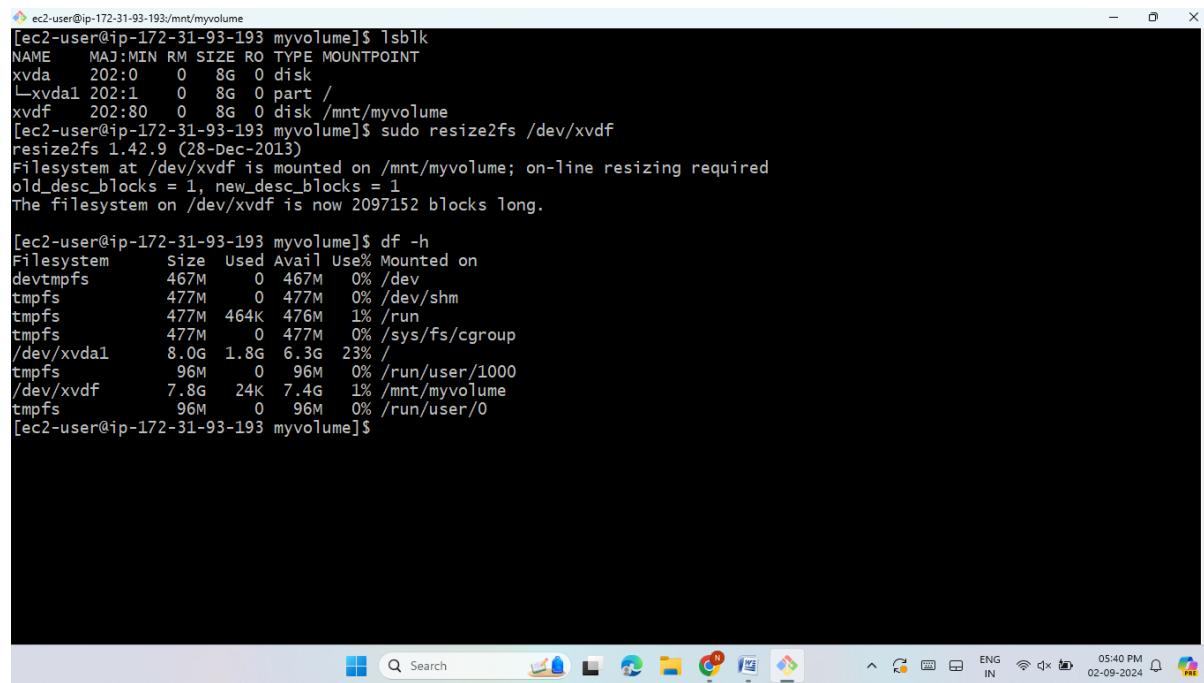
```

[ec2-user@ip-172-31-93-193:/mnt/myvolume]$ lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda   202:0    0 8G  0 disk 
└─xvda1 202:1    0 8G  0 part /
xvdf   202:80   0 8G  0 disk /mnt/myvolume
[ec2-user@ip-172-31-93-193 myvolume]$ sudo resize2fs /dev/xvdf
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/xvdf is mounted on /mnt/myvolume; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/xvdf is now 2097152 blocks long.

[ec2-user@ip-172-31-93-193 myvolume]$

```

- Take a screen shot of this volume and delete it.



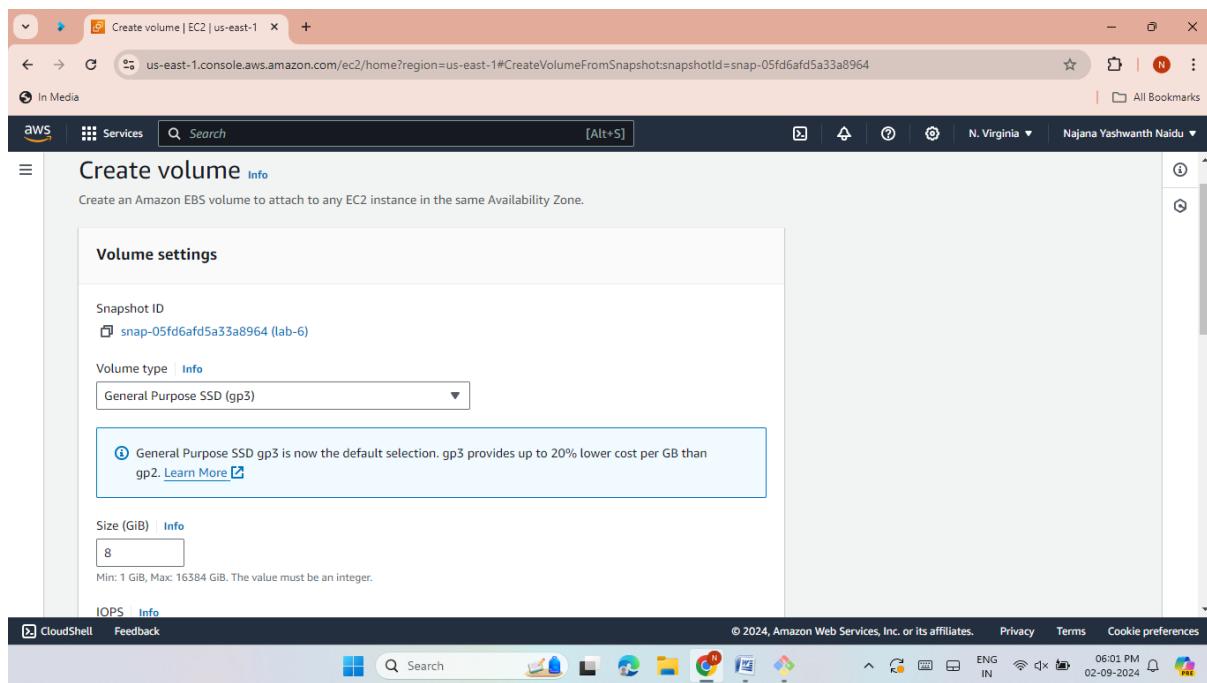
```

[ec2-user@ip-172-31-93-193:/mnt/myvolume]$ lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda   202:0    0 8G  0 disk 
└─xvda1 202:1    0 8G  0 part /
xvdf   202:80   0 8G  0 disk /mnt/myvolume
[ec2-user@ip-172-31-93-193 myvolume]$ sudo resize2fs /dev/xvdf
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/xvdf is mounted on /mnt/myvolume; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/xvdf is now 2097152 blocks long.

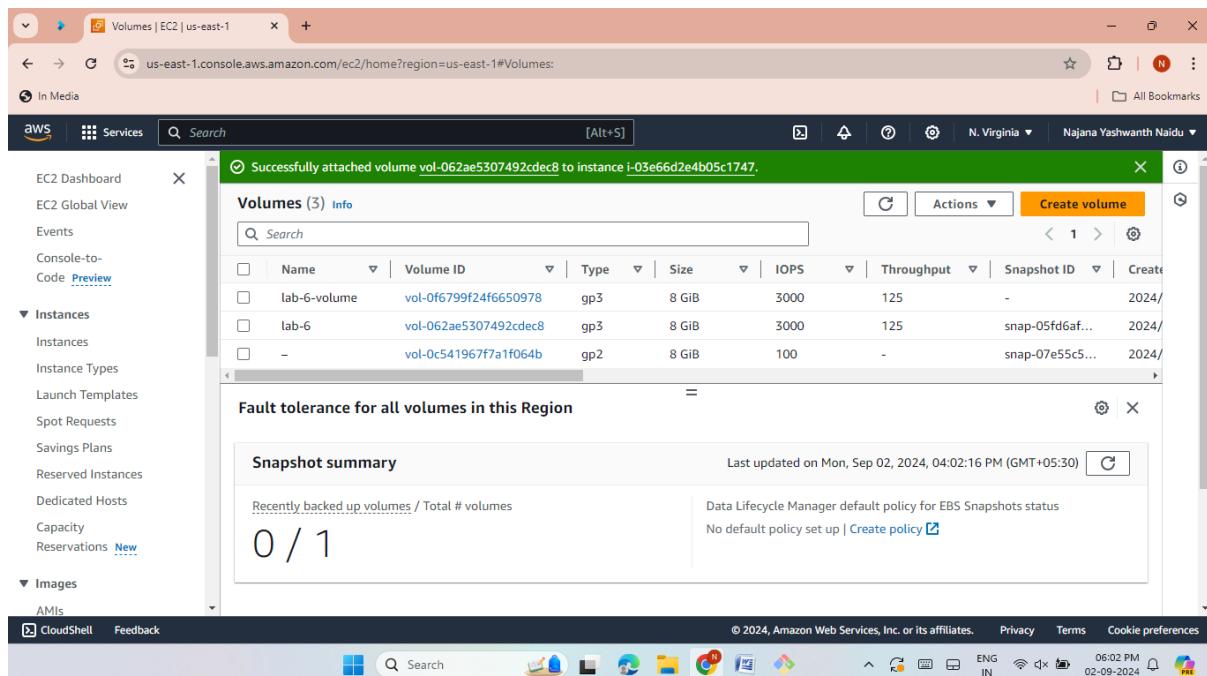
[ec2-user@ip-172-31-93-193 myvolume]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        467M   0  467M  0% /dev
tmpfs          477M   0  477M  0% /dev/shm
tmpfs          477M  464K  476M  1% /run
tmpfs          477M   0  477M  0% /sys/fs/cgroup
/dev/xvda1     8.0G  1.8G  6.3G 23% /
tmpfs          96M   0   96M  0% /run/user/1000
/dev/xvdf      7.8G  24K  7.4G  1% /mnt/myvolume
tmpfs          96M   0   96M  0% /run/user/0
[ec2-user@ip-172-31-93-193 myvolume]$

```

- Create a new volume with the snapshot and attach it to the server.



- The new volume is attached to the instance successfully.



LAB 7 : AMI

- Create an AMI of your running instance.
- Go to Actions → Image and templates → click on create image.

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, Console-to-Code, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, and Reservations. The main area displays two instances: 'lab-7' (instance ID i-03e66d2e4b05c1747, running, t2.micro) and 'lab-6' (instance ID i-0260dfcd92d24be08, terminated, t2.micro). A context menu is open over 'lab-7', with 'Create image' highlighted. Other options in the menu include 'Create template from instance' and 'Launch more like this'. The top navigation bar shows the URL as us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:.

- Give the name of image.

The screenshot shows the 'Create image' wizard. Step 1: 'Create image'. It shows the instance ID 'i-03e66d2e4b05c1747 (lab-7)' and the 'Image name' field containing 'lab-7-ami'. Below it, there's a note: 'Maximum 127 characters. Can't be modified after creation.' There's also an 'Image description - optional' field with 'Image description' and a note: 'Maximum 255 characters'. A checked checkbox 'Reboot instance' has a note: 'When selected, Amazon EC2 reboots the instance so that data is at rest when snapshots of the attached volumes are taken. This ensures data consistency.' At the bottom, there's an 'Instance volumes' section and a progress bar indicating 'Step 1 of 3'.

- Here the image was created and attached to the exited instance.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, and Instances. Under Instances, there are sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations, and Images. The main content area displays two instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
lab-7	i-03e66d2e4b05c1747	Running	t2.micro	2/2 checks passed	View alarms	us-east-1d
lab-6	i-0260dfcd92d24be08	Terminated	t2.micro	-	View alarms	us-east-1d

A modal window is open in the center, stating: "Currently creating AMI ami-0ae4dd89214483883 from instance i-03e66d2e4b05c1747. Check that the AMI status is 'Available' before deleting the instance or carrying out other actions related to this AMI." Below the modal, the instance details for lab-7 are shown:

Details

Instance ID	Public IPv4 address	Private IPv4 addresses
i-03e66d2e4b05c1747 (lab-7)	44.211.248.221 open address	172.31.93.193
IPv6 address	Instance state	Public IPv4 DNS
-	Running	-

- This is the created AMI.

The screenshot shows the AWS Images page. On the left, there's a sidebar with navigation links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store. Under Images, there are sub-links for AMIs, AMI Catalog, Volumes, Snapshots, and Lifecycle Manager. The main content area displays the created AMI:

Name	AMI name	AMI ID	Source	Owner
lab-7-ami	ami-0ae4dd89214483883	637423323663/lab-7-ami	637423323663	637423323663

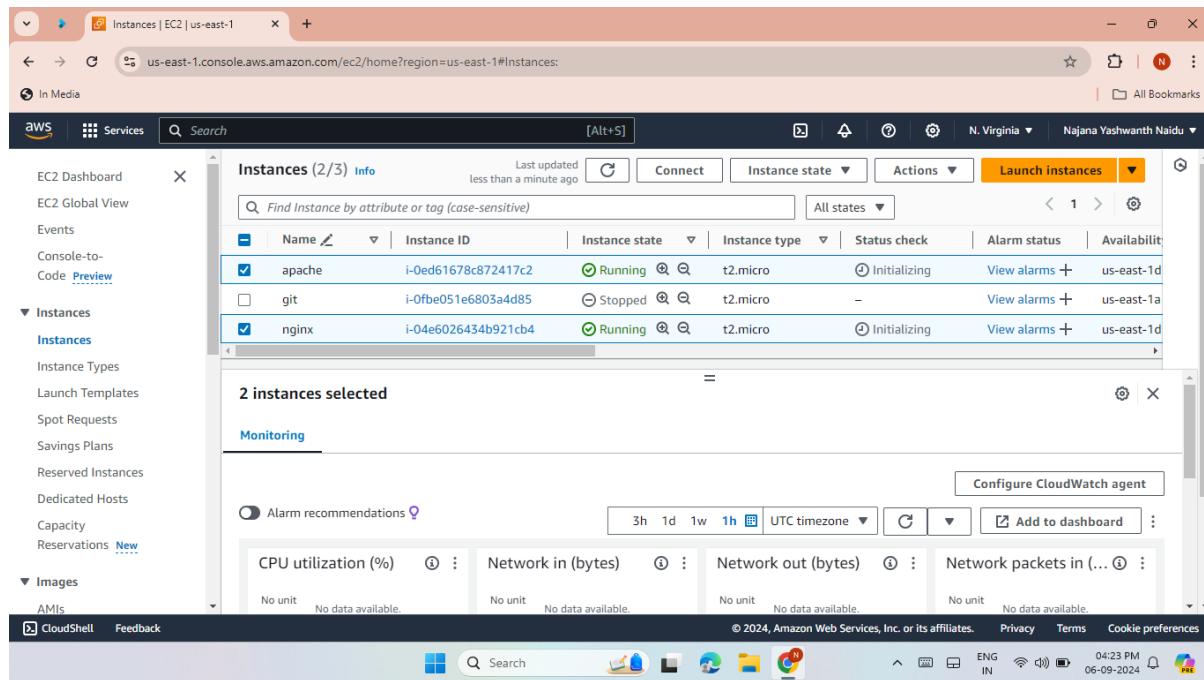
A modal window is open below the table, showing the AMI details:

AMI ID: ami-0ae4dd89214483883

AMI ID	Image type	Platform details	Root device type
ami-0ae4dd89214483883	machine	Linux/UNIX	EBS
AMI name	Owner account ID	Architecture	Usage operation
lab-7-ami	637423323663	x86_64	RunInstances
Root device name	Status	Source	Virtualization type
-	-	-	-

LAB 8 : LAOD BALANCERS

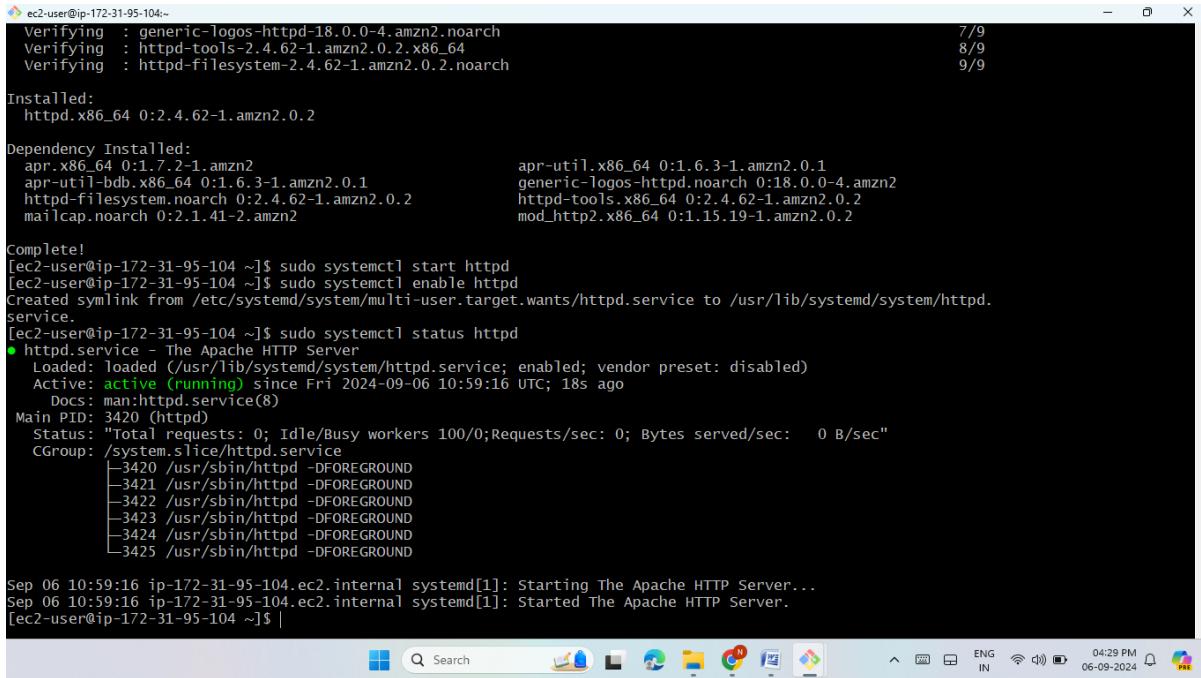
- Create two EC2 Instances.



- install nginx server on one machine.
 - Using Amazon-Linux-extras install nginx1 command.

```
ec2-user@ip-172-31-89-69:~$ 52 tomcat9 available [ =stable ]  
53 unbound1.13 available [ =stable ]  
54 mariadb10.5 available [ =stable ]  
55 kernel-5.10=latest enabled [ =stable ]  
56 redis6 available [ =stable ]  
58 postgresql12 available [ =stable ]  
59 postgresql13 available [ =stable ]  
60 mock2 available [ =stable ]  
61 dnsmasq2.85 available [ =stable ]  
62 kernel-5.15 available [ =stable ]  
63 postgresql14 available [ =stable ]  
64 firefox available [ =stable ]  
65 lustre available [ =stable ]  
66 fphp8.1 available [ =stable ]  
67 awscli1 available [ =stable ]  
68 fphp8.2 available [ =stable ]  
69 dnsmasq available [ =stable ]  
70 unbound1.17 available [ =stable ]  
72 collectd-python3 available [ =stable ]  
i Note on end-of-support. Use 'info' subcommand.  
[ec2-user@ip-172-31-89-69 ~]$ sudo systemctl start nginx  
[ec2-user@ip-172-31-89-69 ~]$ sudo systemctl enable nginx  
Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.service to /usr/lib/systemd/system/nginx.service.  
[ec2-user@ip-172-31-89-69 ~]$ sudo systemctl status nginx  
● nginx.service - The nginx HTTP and reverse proxy server  
  Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)  
  Active: active (running) since Fri 2024-09-06 10:56:29 UTC; 23s ago  
    Main PID: 3447 (nginx)  
      CGroup: /system.slice/nginx.service  
           └─3447 nginx: master process /usr/sbin/nginx  
              ├─3448 nginx: worker process  
  
Sep 06 10:56:29 ip-172-31-89-69.ec2.internal systemd[1]: Starting The nginx HTTP and reverse proxy server...  
Sep 06 10:56:29 ip-172-31-89-69.ec2.internal nginx[3441]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
Sep 06 10:56:29 ip-172-31-89-69.ec2.internal nginx[3441]: nginx: configuration file /etc/nginx/nginx.conf test is s...sful  
Sep 06 10:56:29 ip-172-31-89-69.ec2.internal systemd[1]: Started The nginx HTTP and reverse proxy server.  
Hint: Some lines were ellipsized, use -l to show in full.  
[ec2-user@ip-172-31-89-69 ~]$
```

- Install Apache HTTP server on server2.
- Using sudo yum -y install httpd command.



```

ec2-user@ip-172-31-95-104:~ Verifying : generic-Togos-httpd-18.0.0-4.amzn2.noarch
ec2-user@ip-172-31-95-104:~ Verifying : httpd-tools-2.4.62-1.amzn2.0.2.x86_64
ec2-user@ip-172-31-95-104:~ Verifying : httpd-filesystem-2.4.62-1.amzn2.0.2.noarch
7/9
8/9
9/9

Installed:
httpd.x86_64 0:2.4.62-1.amzn2.0.2

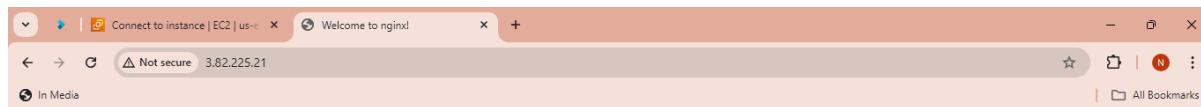
Dependency Installed:
apr.x86_64 0:1.7.2-1.amzn2
apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1
httpd-filesystem.noarch 0:2.4.62-1.amzn2.0.2
mailcap.noarch 0:2.1.41-2.amzn2
mod_http2.x86_64 0:1.15.19-1.amzn2.0.2

Complete!
[ec2-user@ip-172-31-95-104 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-95-104 ~]$ sudo systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-95-104 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
     Active: active (running) since Fri 2024-09-06 10:59:16 UTC; 18s ago
       Docs: man:httpd.service(8)
 Main PID: 3420 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
   CGroup: /system.slice/httpd.service
           └─3420 /usr/sbin/httpd -DFOREGROUND
               ├─3421 /usr/sbin/httpd -DFOREGROUND
               ├─3422 /usr/sbin/httpd -DFOREGROUND
               ├─3423 /usr/sbin/httpd -DFOREGROUND
               ├─3424 /usr/sbin/httpd -DFOREGROUND
               └─3425 /usr/sbin/httpd -DFOREGROUND

Sep 06 10:59:16 ip-172-31-95-104.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Sep 06 10:59:16 ip-172-31-95-104.ec2.internal systemd[1]: Started The Apache HTTP Server.
[ec2-user@ip-172-31-95-104 ~]$ |

```

- Access both the servers over browser and check if their web page is visible or not.



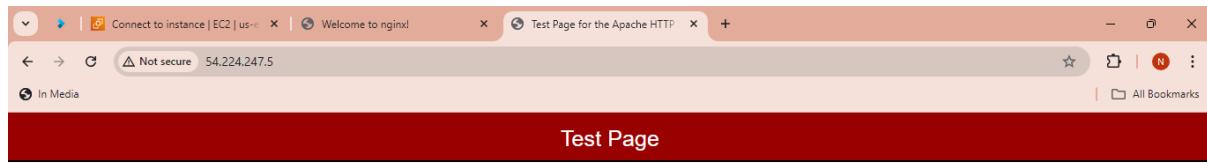
Welcome to nginx!

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. Commercial support is available at nginx.com.

Thank you for using nginx.





This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

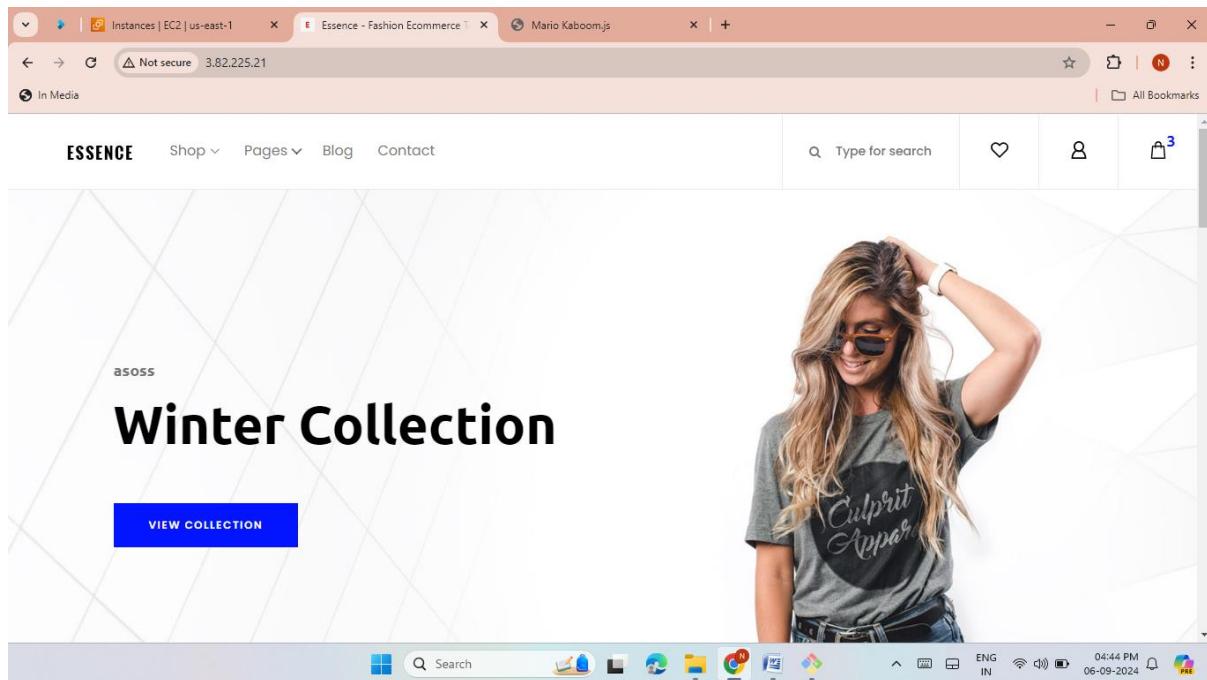
If you are the website administrator:

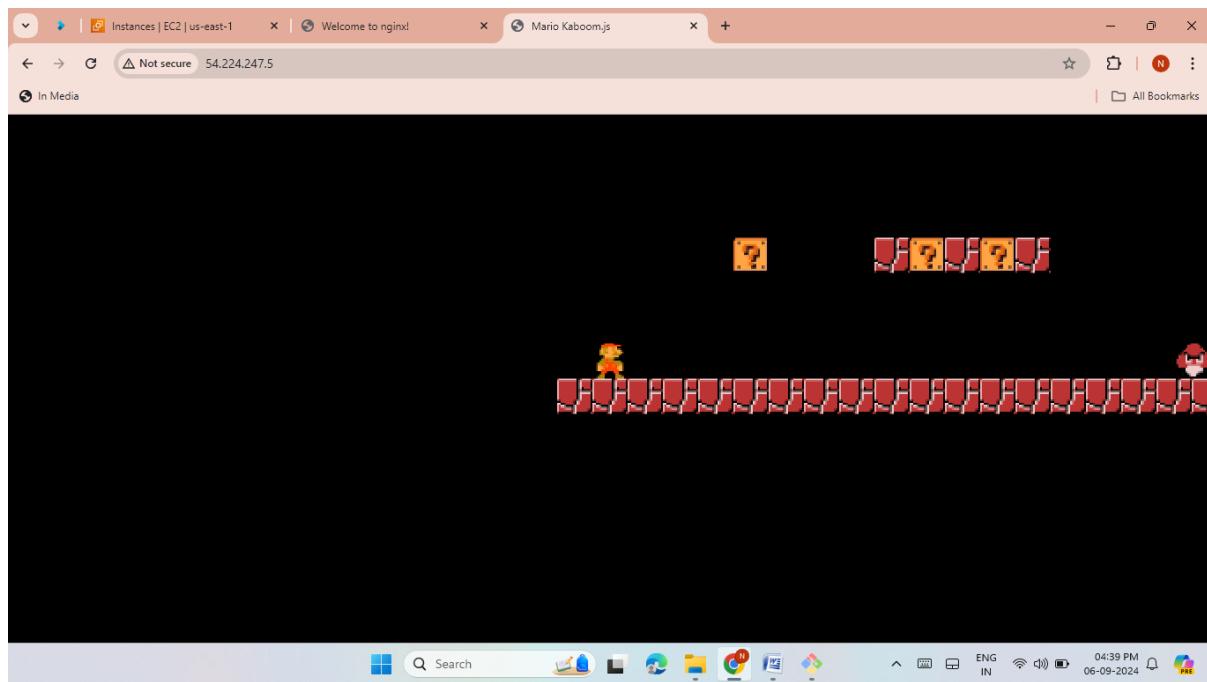
You may now add content to the directory /var/www/html/. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file /etc/httpd/conf.d/welcome.conf.

You are free to use the image below on web sites powered by the Apache HTTP Server.

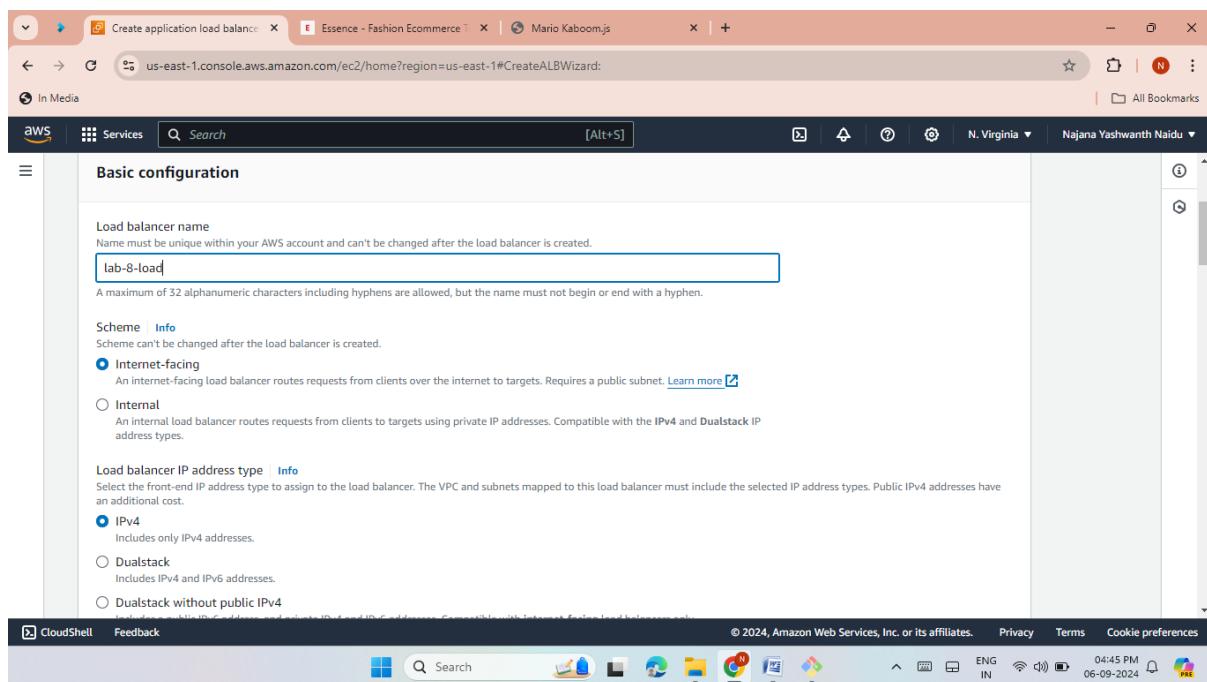


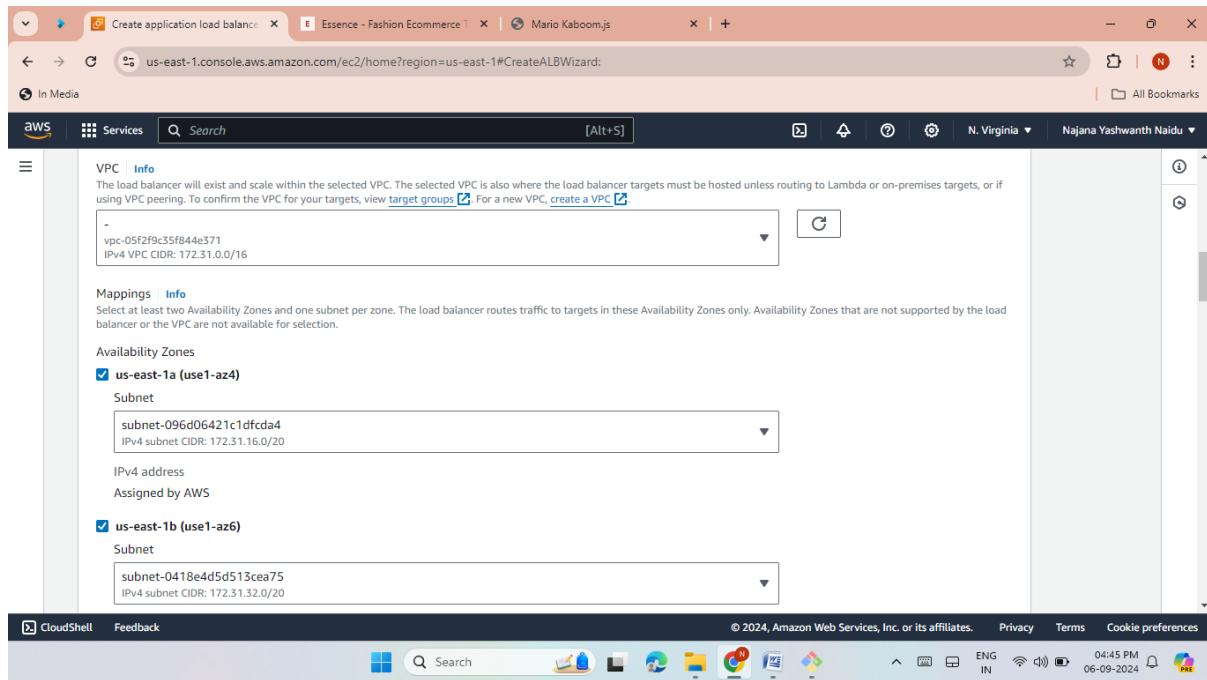
- Install git then clone the different repo on two servers.
- Path of nginx → /usr/share/nginx/html/.
- Path of HTTP → /var/www/html/.



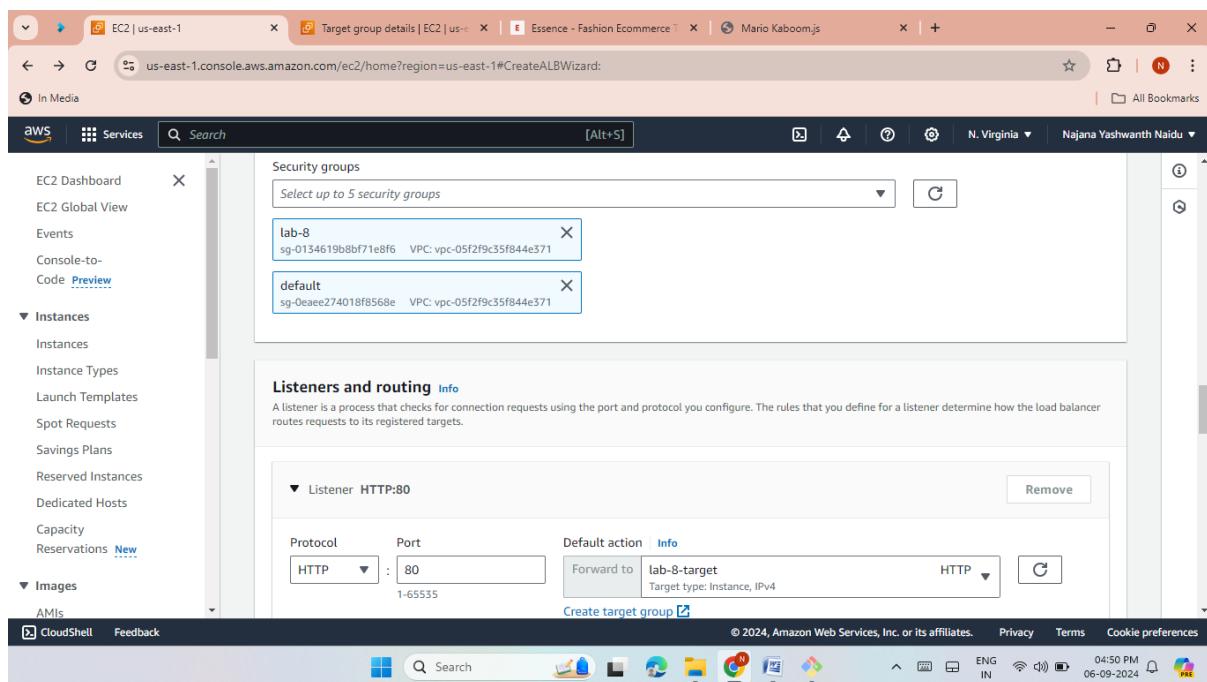


- Create one load balancers and attach both the instances with the load balancer.
- Select VPC and Availability zones.





- Allow only port 80 in Load Balancer security groups and also make the Security group of your instances to receive request from LB only on port 80.



- Here the load balancer was created Successfully.

Successfully created load balancer: lab-8-load

It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

lab-8-load

Details	
Load balancer type	Application
Status	Provisioning
VPC	vpc-05f2f9c35f844e371
Load balancer IP address type	IPv4
Scheme	Internet-facing
Hosted zone	Z35SXDOTRQ7X7K
Availability Zones	subnet-0418e4d5d513cea75 us-east-1b (use1-az6) subnet-0b9cd5a0e3ae871a us-east-1d (use1-az2)
Date created	September 6, 2024, 16:51 (UTC+05:30)

- Create a target group and attach to the load balancer.
- Same VPC and subnet.

Step 1
Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section can't be changed after the target group is created.

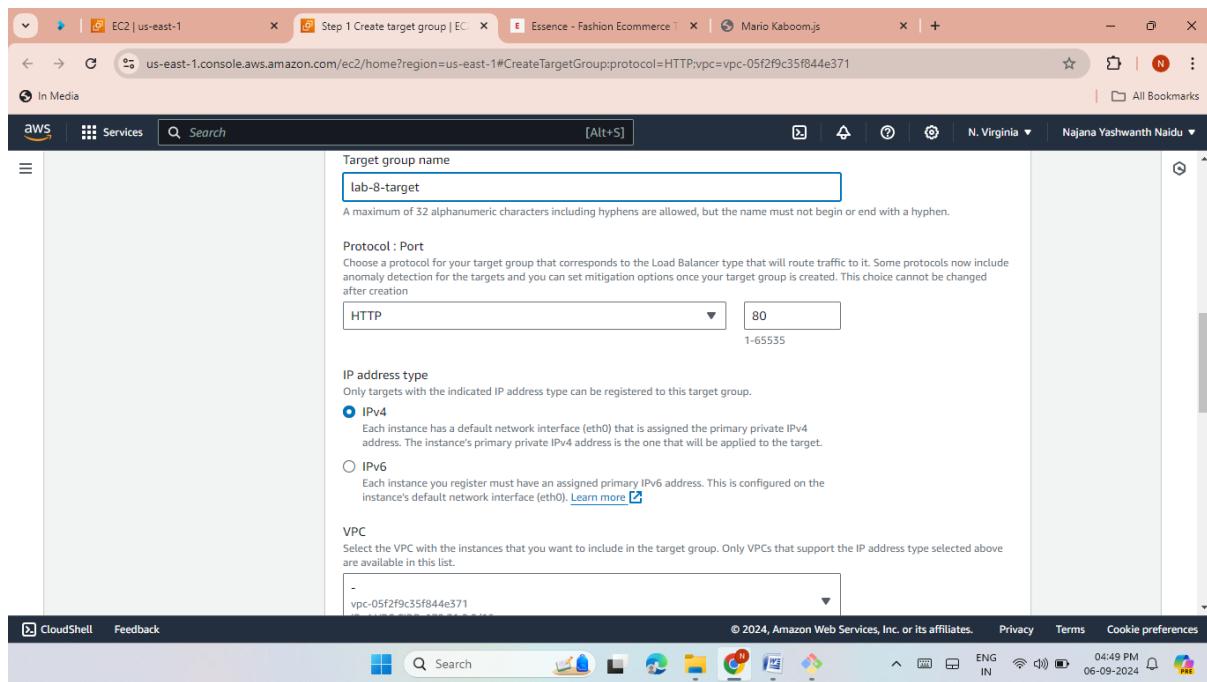
Choose a target type

Instances

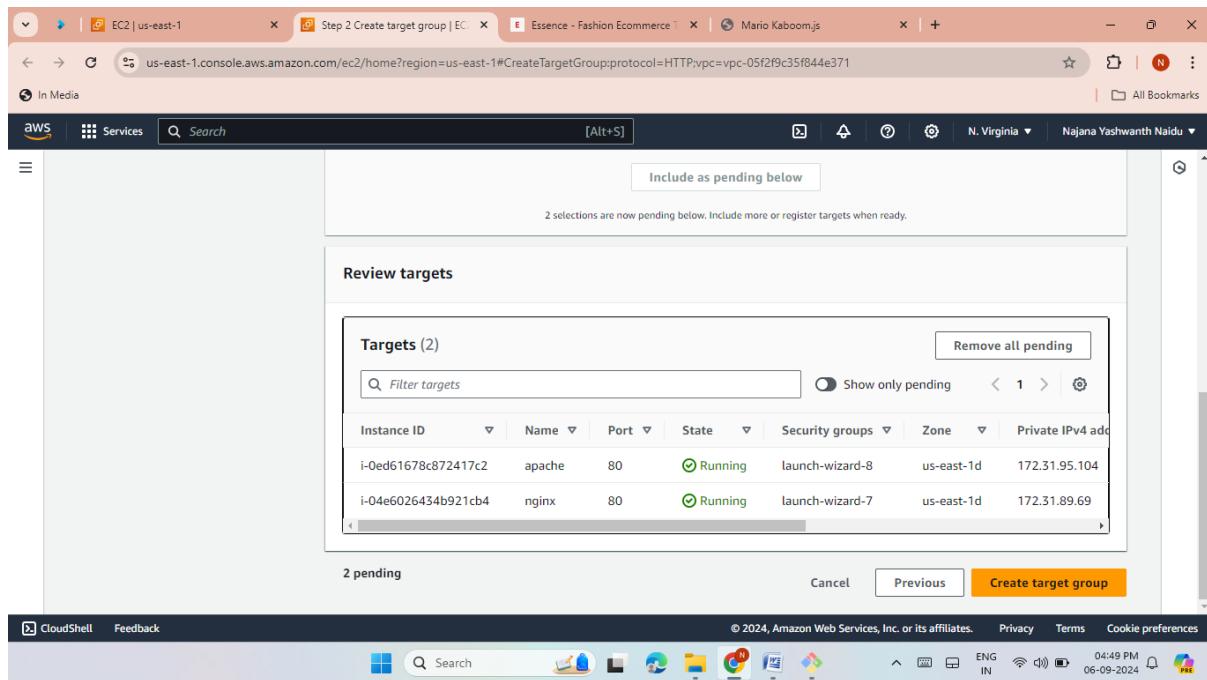
- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

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.



➤ Register the instances in the target group.



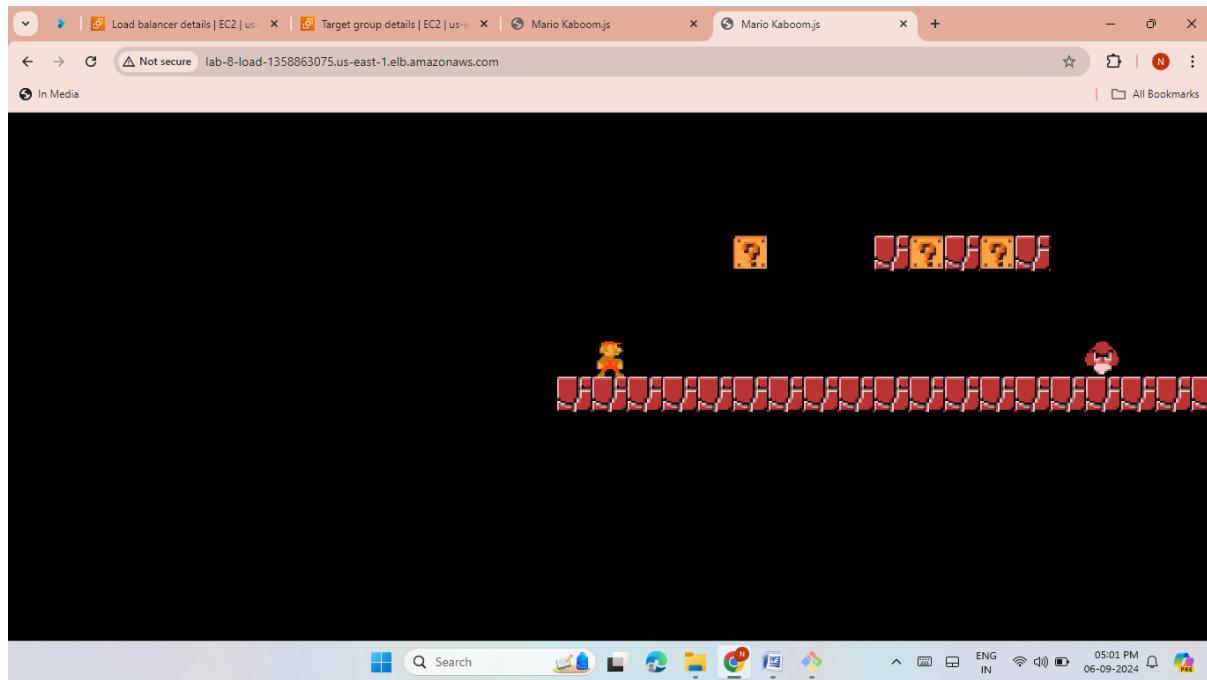
- Access the load balancer DNS link over the browser and hit it a couple of times. Check if the Webpage (Nginx/Apache HTTP are visible alternatively).

The screenshot shows the AWS CloudShell interface with the following details:

- Services Menu:** Shows options like Images, AMIs, AMI Catalog, Elastic Block Store, Network & Security, Load Balancing, and Target Groups.
- Load balancer ARN:** arn:aws:elasticloadbalancing:us-east-1:637423323663:loadbalance/r/app/lab-8-load/81f6123d68487462
- DNS name Info:** lab-8-load-1358863075.us-east-1.elb.amazonaws.com (A Record)
- Listeners and rules (1) Info:** A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.
- CloudShell Feedback:** Includes a search bar and various system icons.

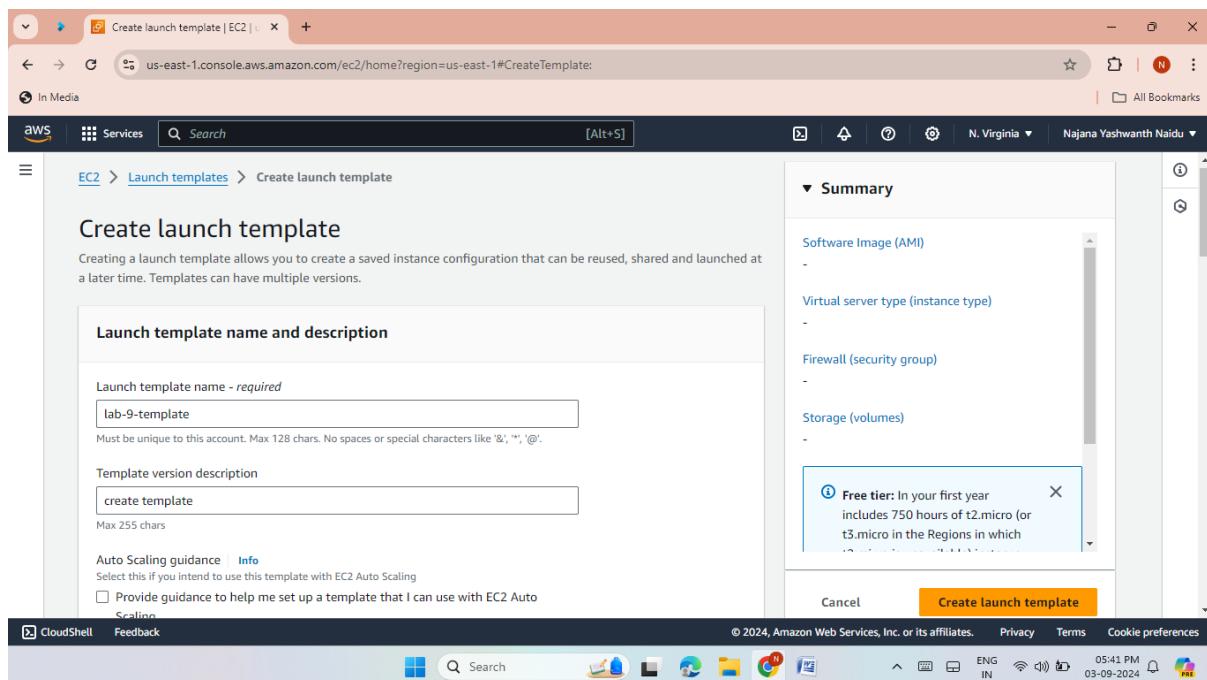
The screenshot shows a web browser window with the following details:

- Address Bar:** Not secure lab-8-load-1358863075.us-east-1.elb.amazonaws.com
- Content Area:** A navigation menu under "Shop" listing categories: Women's Collection, Dresses, Blouses & Shirts, T-shirts, Rompers, Bras & Panties, Men's Collection, T-Shirts, Polo, Shirts, Jackets, Trench, Kid's Collection, Dresses, Shirts, T-shirts, Jackets, Trench. Below the menu is a large image of a brown leather handbag.
- System Icons:** Standard Windows taskbar icons for search, file explorer, and other applications.



LAB 9 : AUTO SCALING GROUP AND LAUNCH TEMPLATE

- Create one launch template with ubuntu server.



Screenshot of the AWS Cloud Console showing the 'Create launch template | EC2' page. The 'Summary' section is displayed, showing the selected AMI (Ubuntu Server 22.04 LTS (HVM), SSD Volume Type), instance type (t2.micro), and storage (1 volume(s) - 8 GiB). A tooltip for the 'Free tier' is visible, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which you're using it)'.

Screenshot of the AWS Cloud Console showing the 'Create launch template | EC2' page. The 'Summary' section is displayed, showing the selected AMI (Ubuntu Server 22.04 LTS (HVM), SSD Volume Type), instance type (t2.micro), and storage (1 volume(s) - 8 GiB). A tooltip for the 'Free tier' is visible, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which you're using it)'. The 'Security group' dropdown is set to 'Create security group'.

- Create an auto scaling group and attach the above created Template.

Choose launch template

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name

Auto Scaling group name
Enter a name to identify the group.
lab-9-ASG

Must be unique to this account in the current Region and no more than 255 characters.

Launch template

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.

Launch template

Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

lab-9-template

Create a launch template

Description: create template

AMI ID

Launch template: lab-9-template
lt-014de9b0b159a766c

Instance type: t2.micro

Security groups

Request Spot Instances

Launch template

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.

Launch template

Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

lab-9-template

Create a launch template

Description: create template

AMI ID

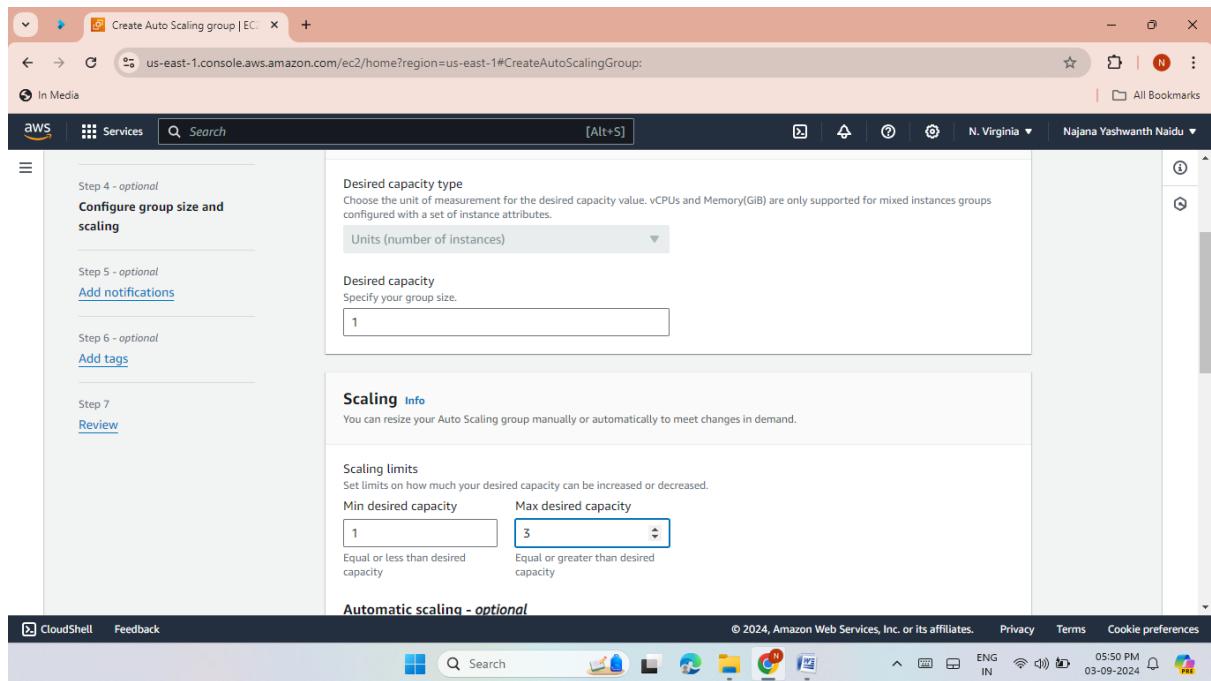
Launch template: lab-9-template
lt-014de9b0b159a766c

Instance type: t2.micro

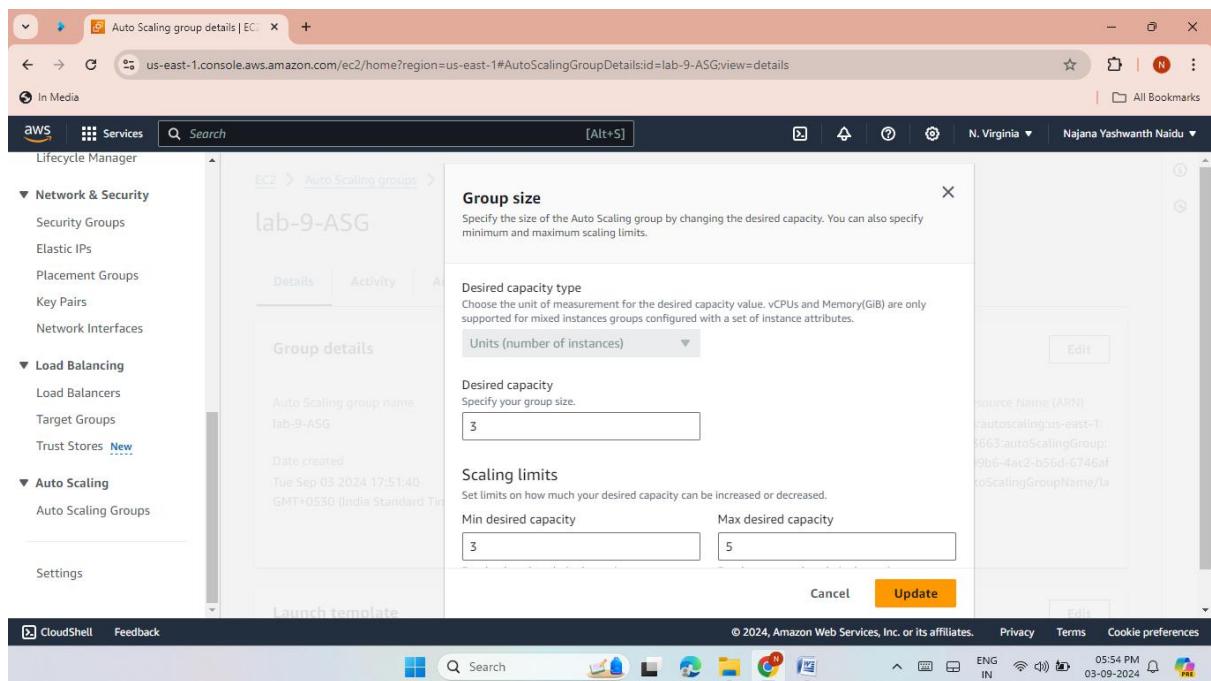
Security groups

Request Spot Instances

- Keep the size of instances as Min = 1, Max = 3.



- Try to change the max capacity and see the new instance should get created.
- Here the Min 3 and Max 5.



- Here the Min 3 Instances was launched. If we delete any instance it automatically created another one (max 5).

The screenshot shows the AWS EC2 Instances page. The left sidebar is expanded to show 'Instances' under 'EC2 Dashboard'. The main area displays a table of 5 instances. The first three instances are listed as 'Running' t2.micro type. A modal window titled 'Select an instance' is open over the table, listing the same three instances.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
	i-0e9127fc63858447	Running	t2.micro	Initializing	View alarms	us-east-1a
	i-03fb6b447e359edf4	Running	t2.micro	Initializing	View alarms	us-east-1a
	i-0e4ee0a6a982d9ae8	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a

LAB 10 : RDS (RELATIONAL DATABASE)

- Provision an RDS instance.
- Choose Standard creation, Select MYSQL, Free tier.
- Select self managed then Create master password.

The screenshot shows the 'Choose a database creation method' step of the RDS instance creation wizard. The 'Standard create' option is selected. To the right, a 'MySQL' info panel provides details about the MySQL database, including its popularity and features like automated backups and up to 15 read replicas. Below this, engine options for MySQL, Aurora (MySQL Compatible), and Aurora (PostgreSQL Compatible) are shown, with MySQL selected.

RDS | us-east-1

us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:

In Media

Services Search [Alt+S]

Templates
Choose a sample template to meet your use case.

- Production Use defaults for high availability and fast, consistent performance.
- Dev/Test This instance is intended for development use outside of a production environment.
- Free tier Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas across Regions.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 02:45 PM 04-09-2024

RDS | us-east-1

us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:

In Media

Services Search [Alt+S]

Credentials management
You can use AWS Secrets Manager or manage your master user credentials.

- Managed in AWS Secrets Manager - most secure RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.
- Self managed Create your own password or have RDS create a password that you manage.
- Auto generate password Amazon RDS can generate a password for you, or you can specify your own password.

Master password Info

Password strength Strong

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / \ * @

Confirm master password Info

Instance configuration
The DB instance configuration options below are limited to those supported by the engine that you selected above.

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas across Regions.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 02:47 PM 04-09-2024

RDS | us-east-1

us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:

In Media

Services Search [Alt+S]

Virtual private cloud (VPC) Info
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Default VPC (vpc-05f2f9c35f844e371)
6 Subnets, 6 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change its VPC.

DB subnet group Info
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

default

Public access Info

Yes RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas across Regions.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 02:50 PM 04-09-2024

- Here RDS (database) was created successfully.

The screenshot shows the AWS RDS console in the us-east-1 region. A green success banner at the top states "Successfully created database database-1". Below it, a message says "You can use settings from database-1 to simplify configuration of suggested database add-ons while we finish creating your DB for you." On the left sidebar, the "Databases" section is selected. The main content area shows a table titled "Databases (1)" with one row for "database-1". The table includes columns for DB identifier, Status, Role, Engine, Region &..., Size, and Recommendations. The "Status" column shows "Backing-up". The "Actions" dropdown menu for the database includes options like "Modify", "Restore from S3", and "Create database". At the bottom of the page, there are links for "CloudShell" and "Feedback".

- Open MYSQL port in the connected Security Group.
- Give MYSQL port (3306).

The screenshot shows the "Create security group" wizard in the AWS EC2 console. The first step, "Basic details", is completed. The security group name is "lab-10" and the description is "allow 3306 port". The VPC is set to "vpc-05f2f9c35f844e371". The second step, "Inbound rules", is shown below. It lists a single rule: "Allow traffic from 0.0.0.0/0 on port 3306 (tcp)". The "Inbound rules" section also includes a link to "Info" and a note: "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 fields below." At the bottom of the page, there are links for "CloudShell" and "Feedback".

The screenshot shows the 'Edit inbound rules' section of the AWS RDS ModifyInboundSecurityGroupRules page. It displays two rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
-	SSH	TCP	22	Any... <input type="button" value="Delete"/>	0.0.0.0/0 <input type="button" value="Delete"/>
-	Custom TCP	TCP	3306	Any... <input type="button" value="Delete"/>	0.0.0.0/0 <input type="button" value="Delete"/>

A button labeled 'Add rule' is located at the bottom left. 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.' A 'CloudShell' tab is visible at the bottom left.

➤ Launched instance.

The screenshot shows the 'Instances (1/1)' page in the AWS EC2 Instances section. A single instance is listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
lab-10-instance	i-001055d6abeb98a62	Running	t2.micro	Initializing	View alarms	us-east-1d

The instance details page for 'i-001055d6abeb98a62 (lab-10-instance)' is shown below. The 'Details' tab is selected, displaying information such as Instance ID, Public IPv4 address (18.207.188.252), Private IPv4 address (172.31.87.110), Public IPv4 DNS (ec2-18-207-188-252.compute-1.amazonaws.com), and Instance state (Running).

- Access this RDS from your EC2 instance.
 - Install MYSQL on server using sudo yum install -y MYSQL command.
 - Connect the MYSQL Using MYSQL -h (<end point address>) -u admin -p command.

```
ec2-user@ip-172-31-87-110:~  
Warning: Permanently added 'ec2-18-207-188-252.compute-1.amazonaws.com' (ED25519) to the list of known hosts.  
#  
Amazon Linux 2  
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/  
[ec2-user@ip-172-31-87-110 ~]$ sudo yum -y install mysql  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd  
Resolving Dependencies  
--> Running transaction check  
--> 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 | 8.8 MB 00:00:00  
ENG IN WiFi 03:13 PM 04-09-2024
```