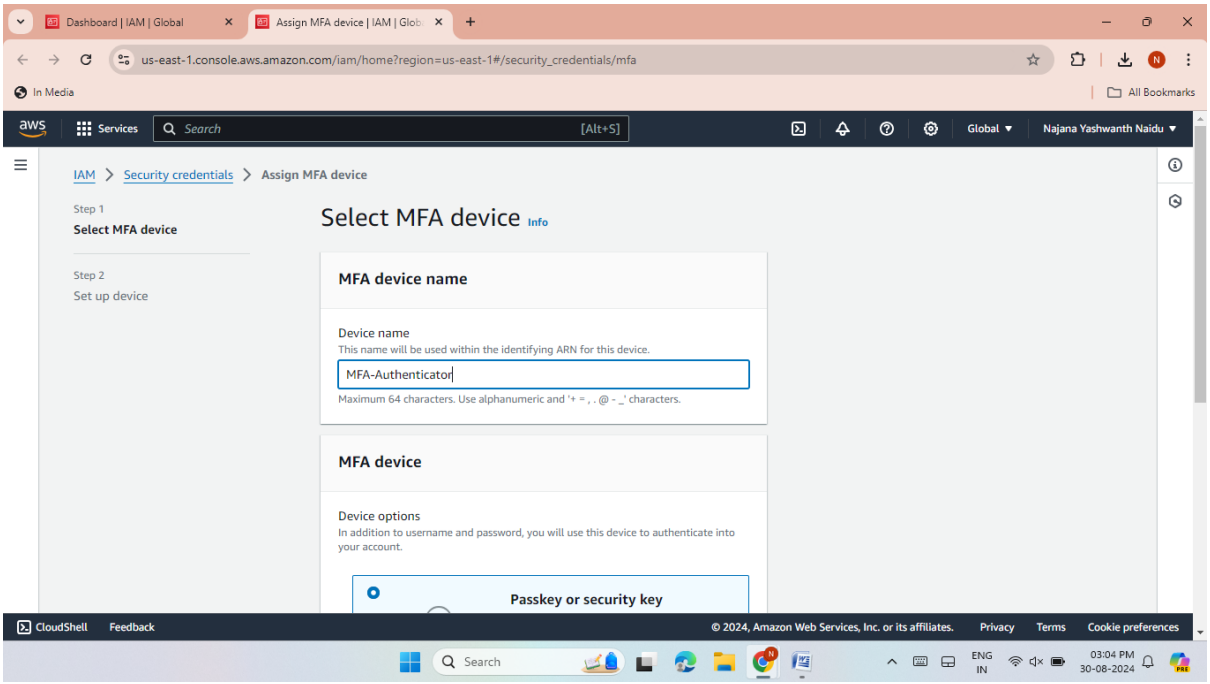
AWS MINI PROJECT

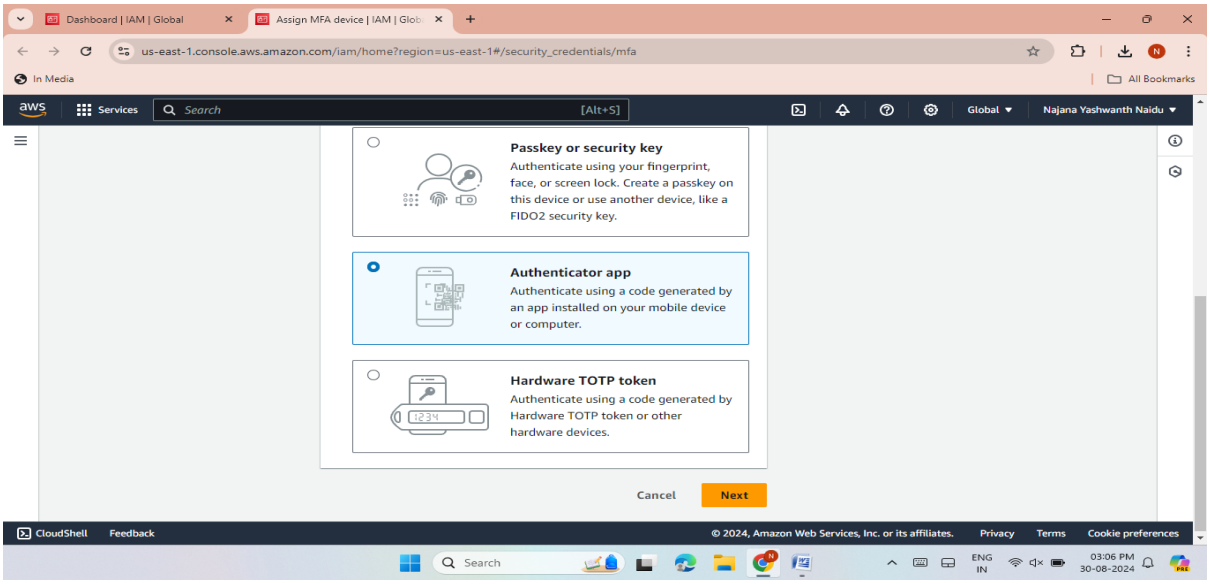
N.Yashwanth Naidu

LAB-1 : IAM HANDS-ON

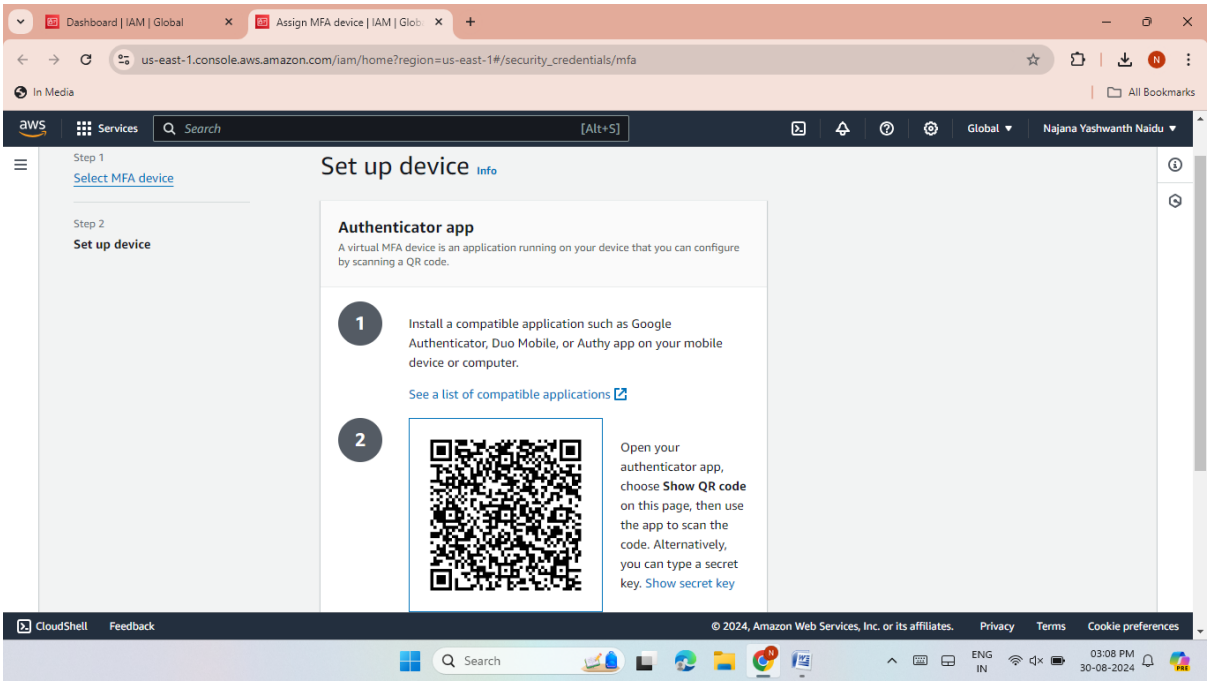
- Set the MFA for root user.
- Choose the device name.



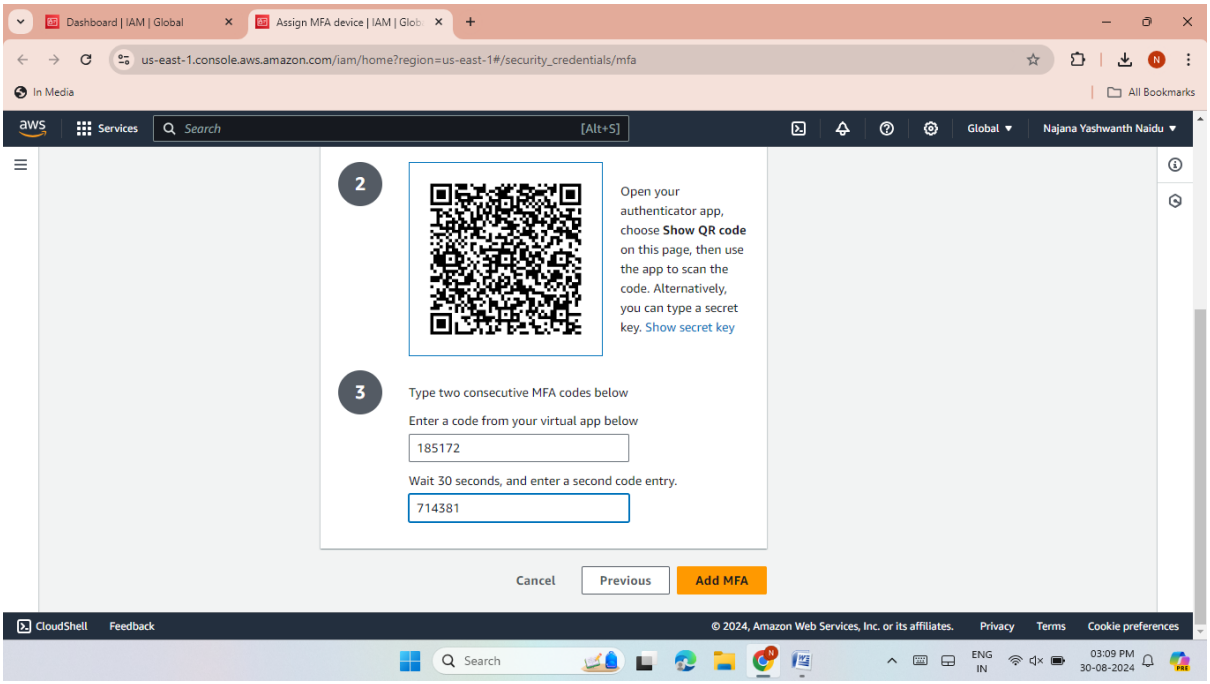
- Select the device option. I took the authenticator app.



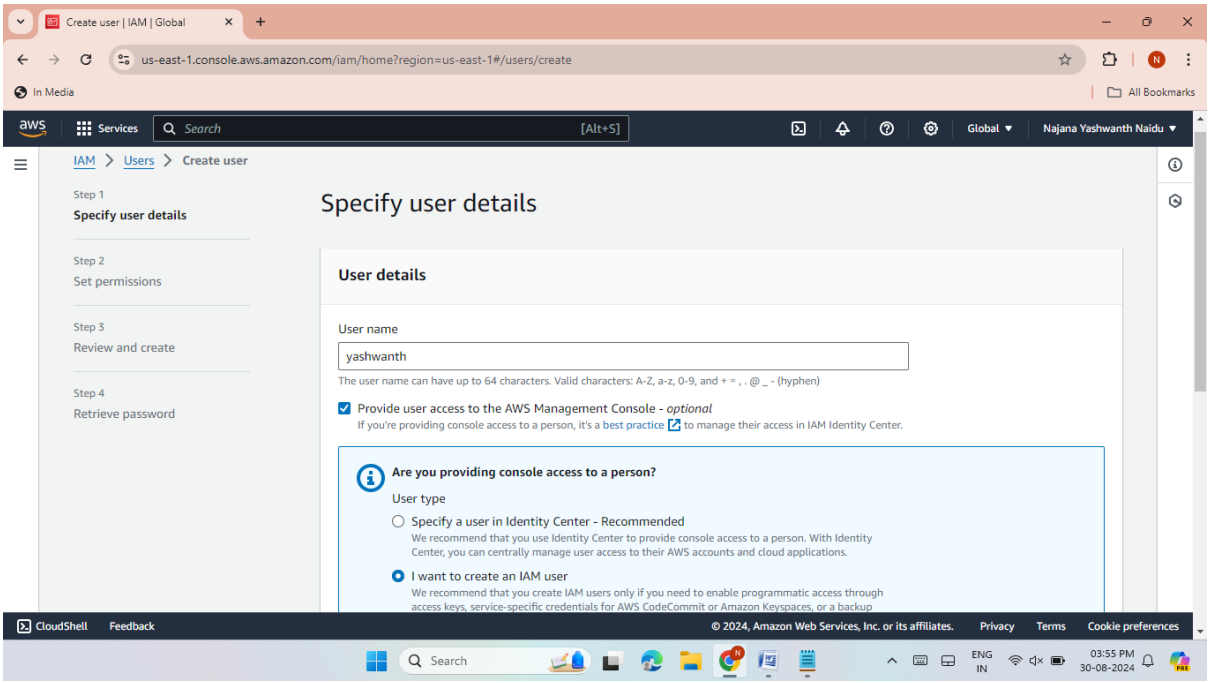
- Scane the QR code from your mobile app like Google authenticator,Duo mobile or Authy app.



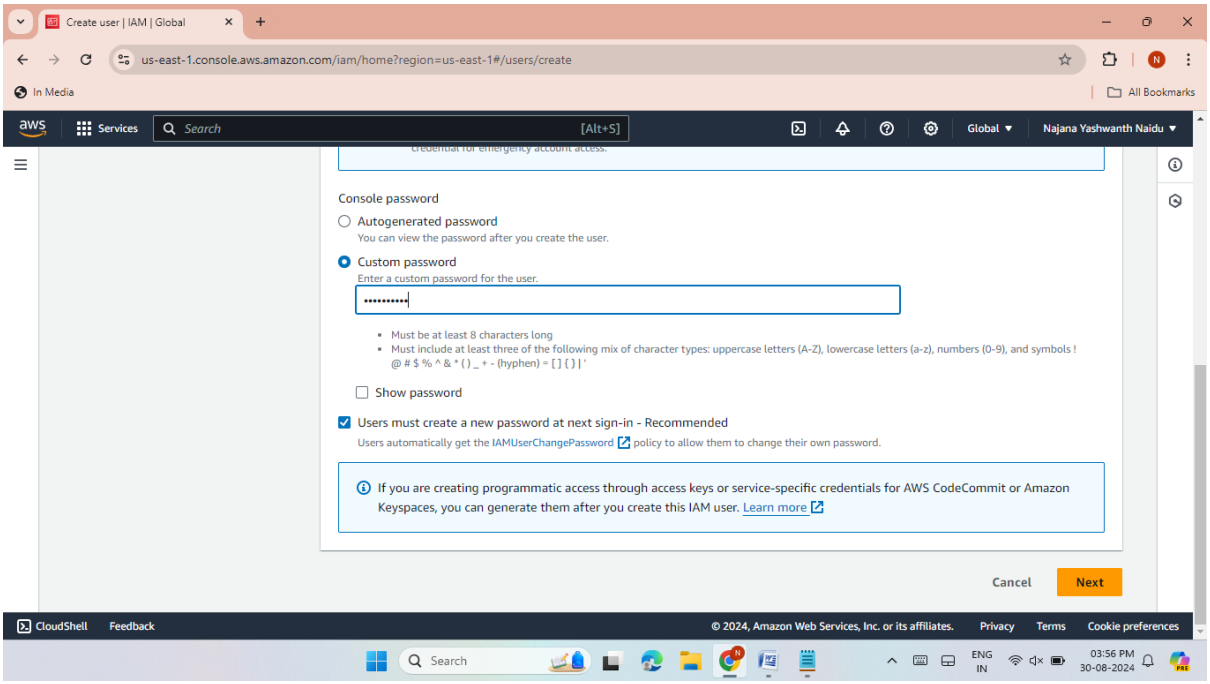
- First enter the code from your mobile app then after 30 sec enter the second code.
- Then click add MFA, Then the MFA is succesfully created.



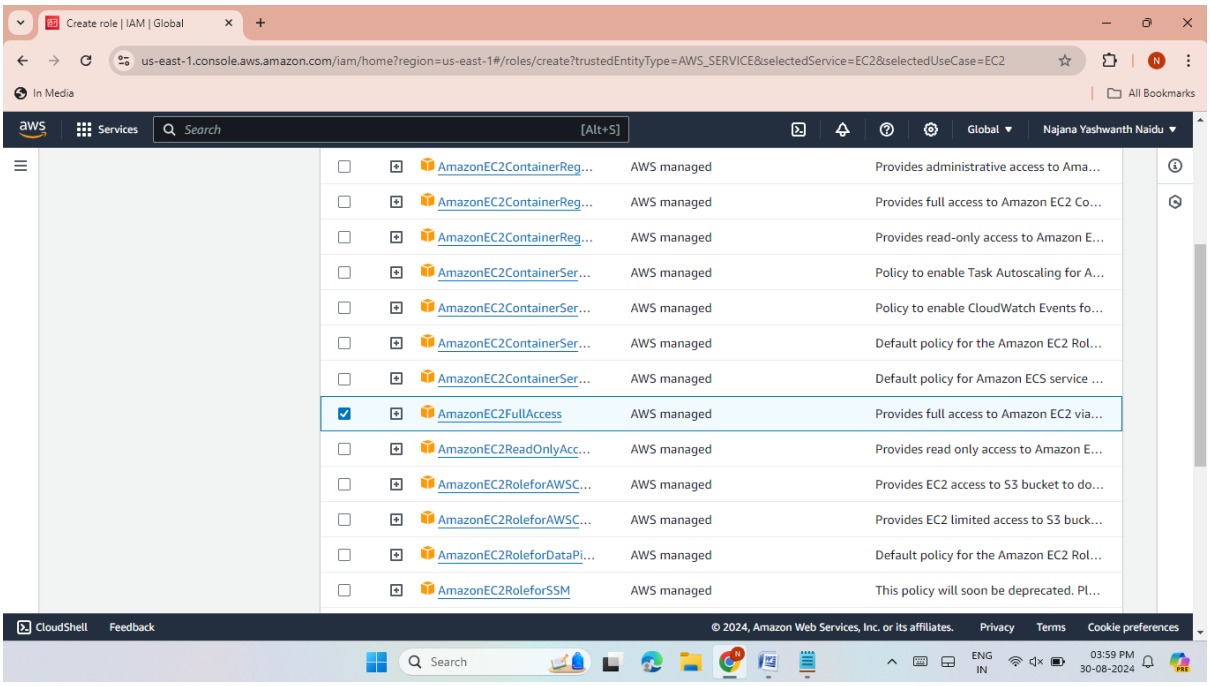
- Create a new user with console access and check its default permissions.



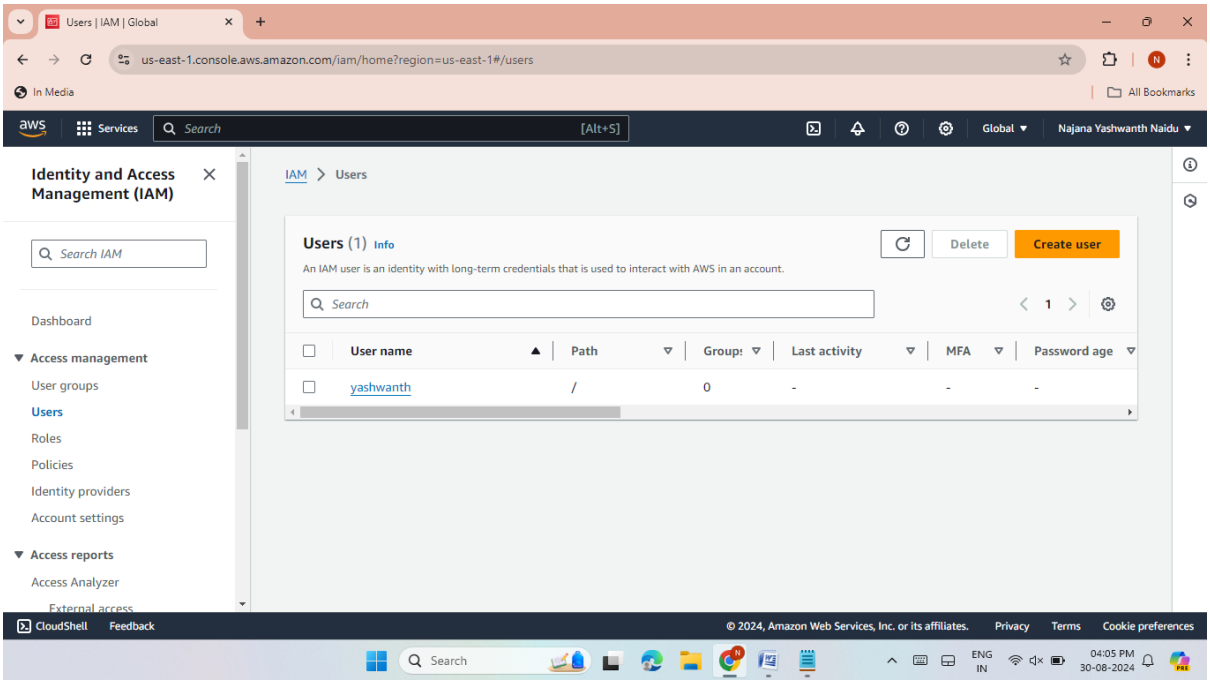
- Create a custom password for your IAM login purpose.



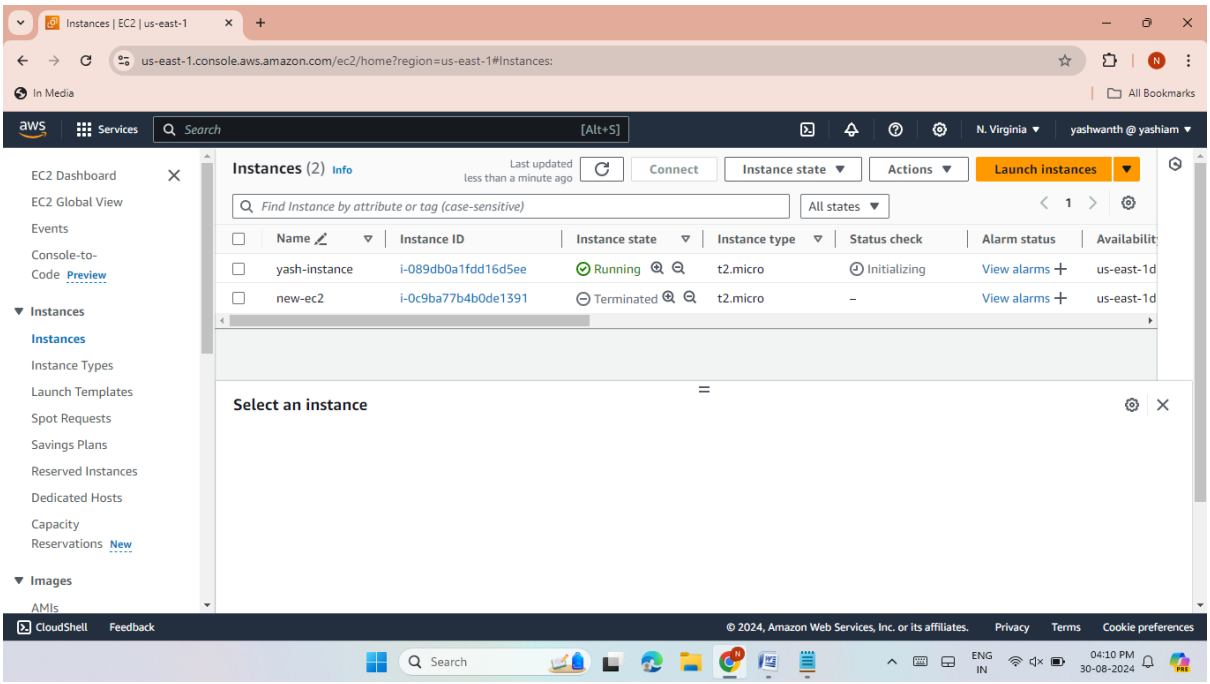
- Assign only ec2 permissions to this user and navigate to different services with this user access.



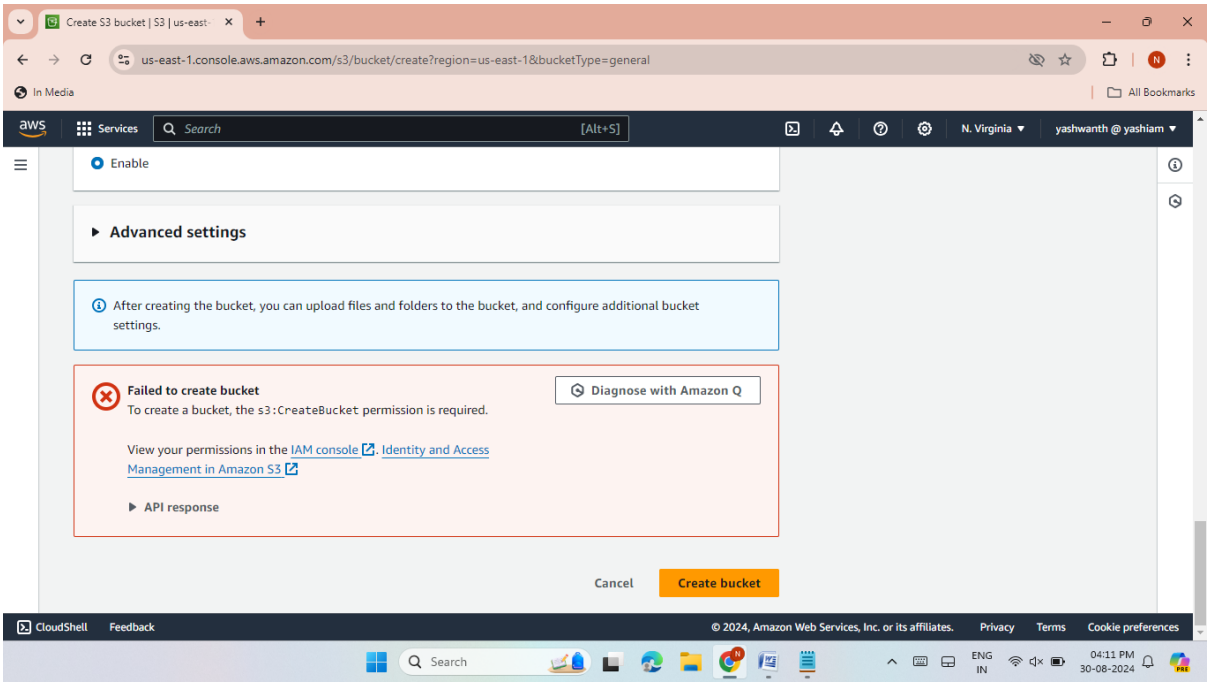
- Here the IAM user was created.



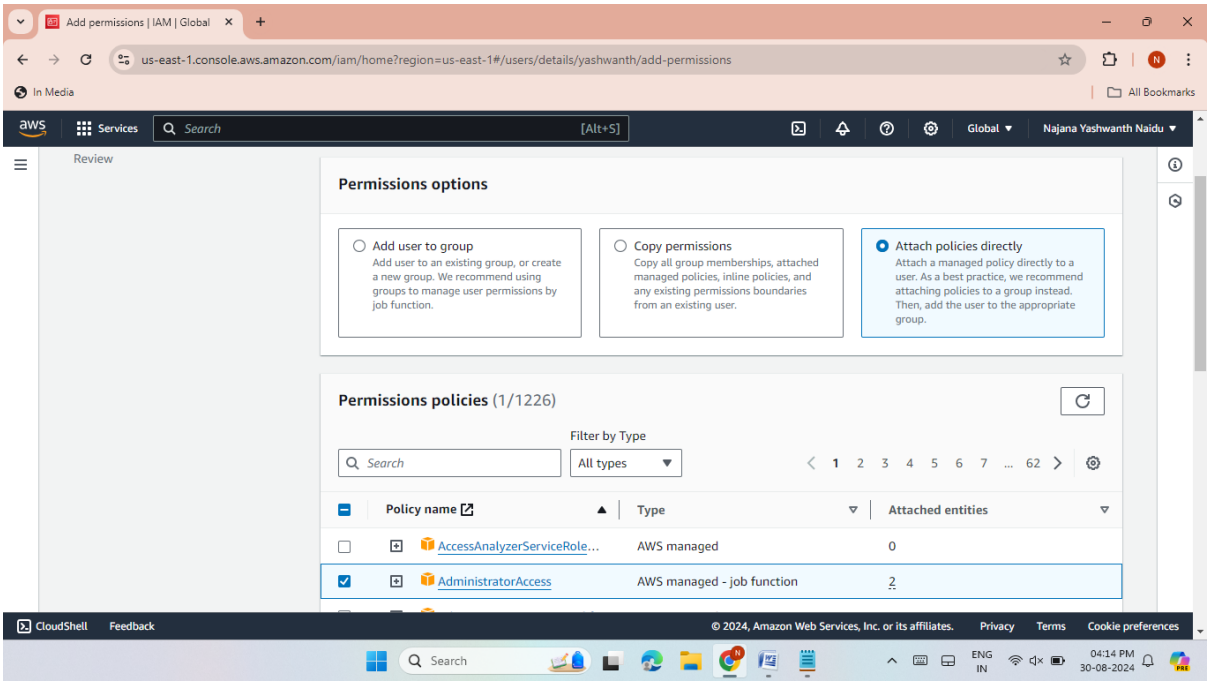
- Instance was launched with the name yash-instance.



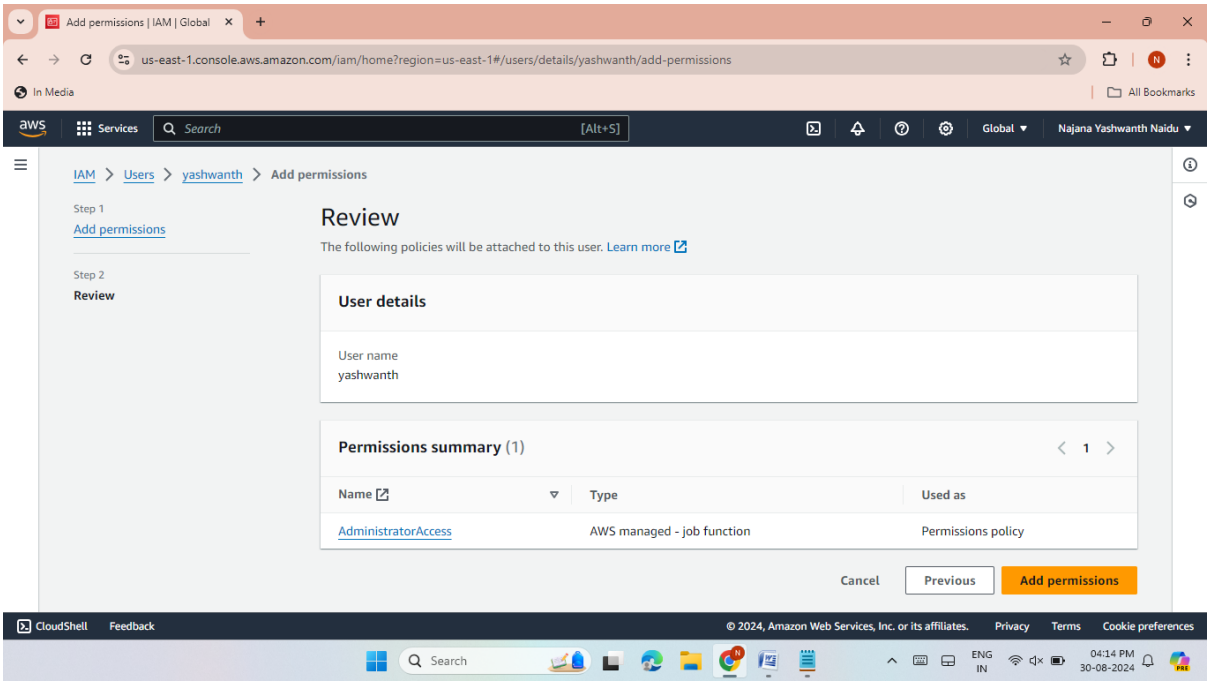
- While launching the S3 bucket it shows fail to create bucket.
- We Can't work with any other services except ec2.



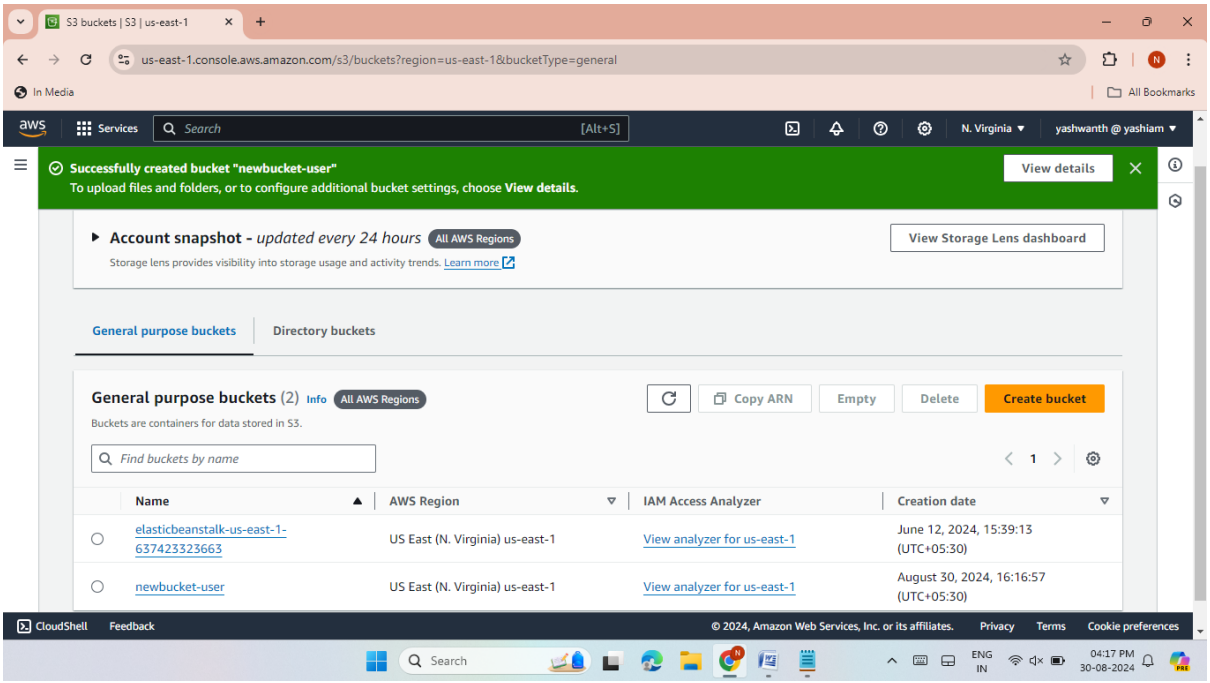
➢ Provide then Administrator Access to this user.



➢ Click on add permissions.

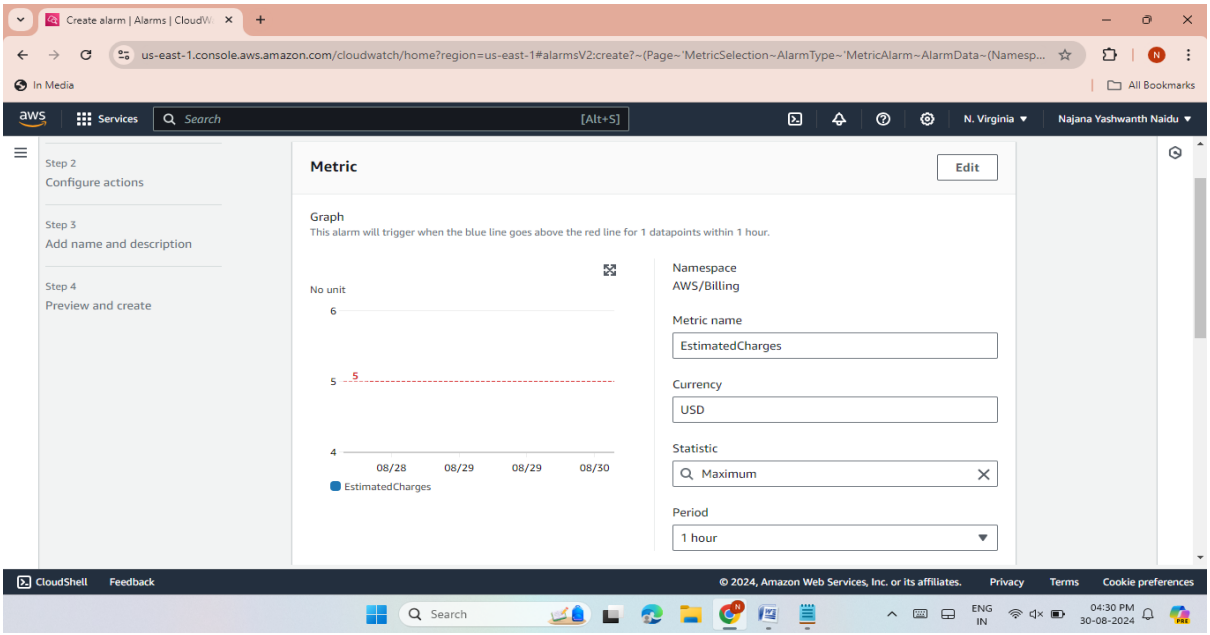


- Here the remaining services also work with the Admin permissions.

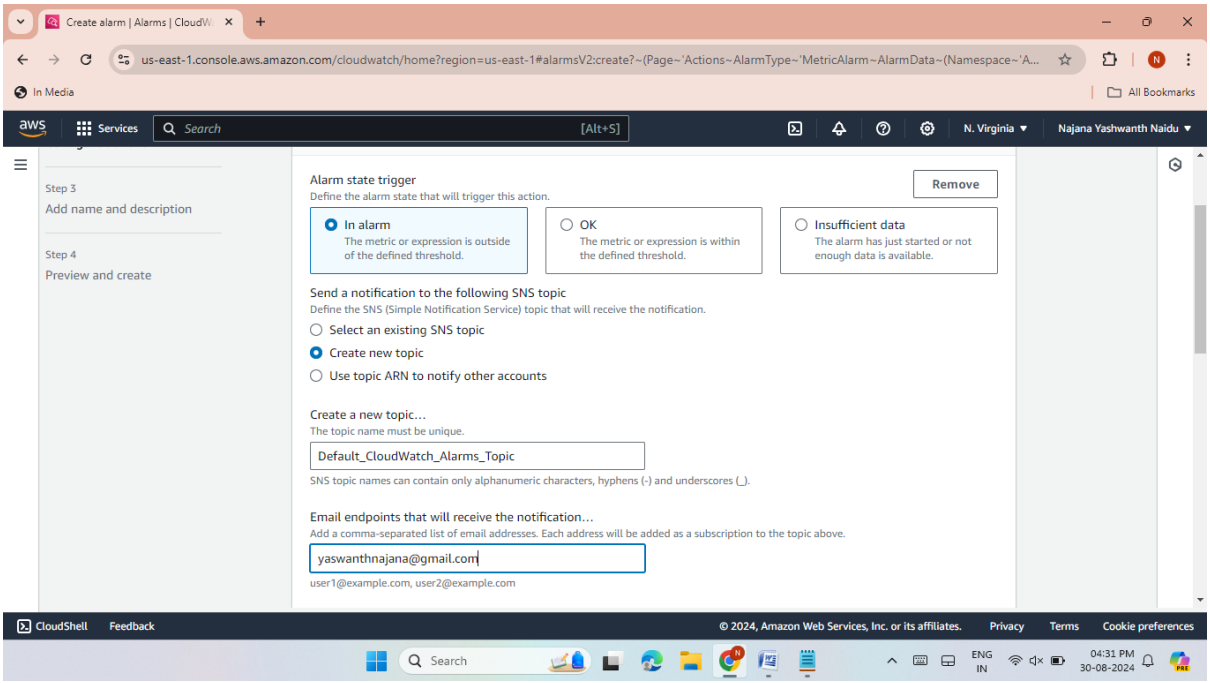


LAB-2 : BILLING ALARM

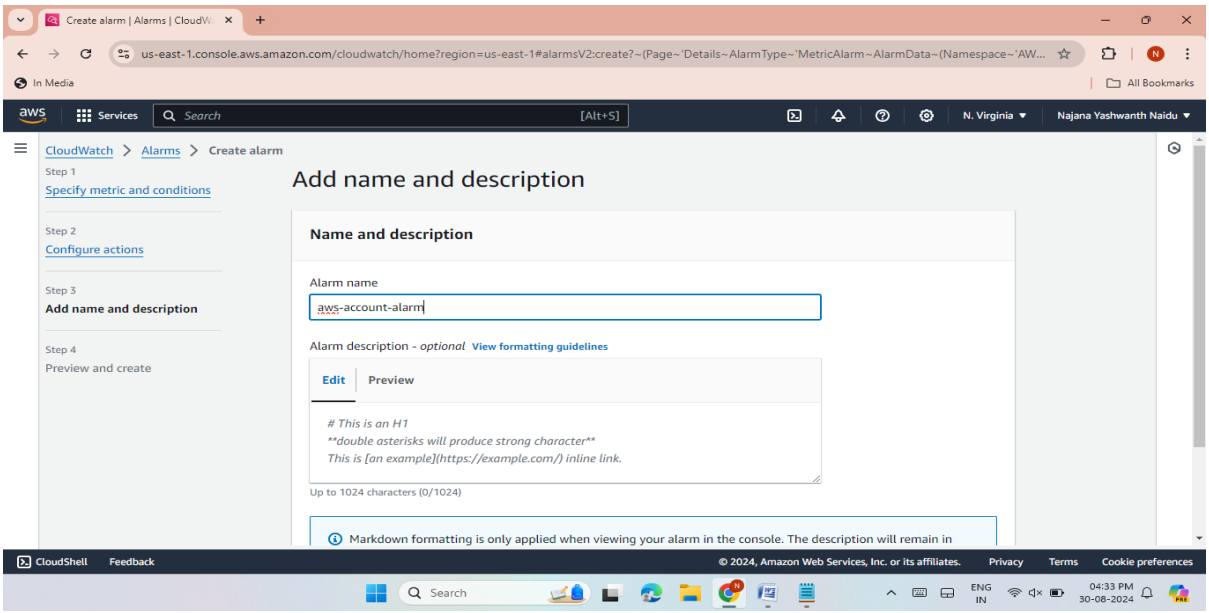
- Setup the billing alarm for your account to get a notification whenever you cross the billing threshold.



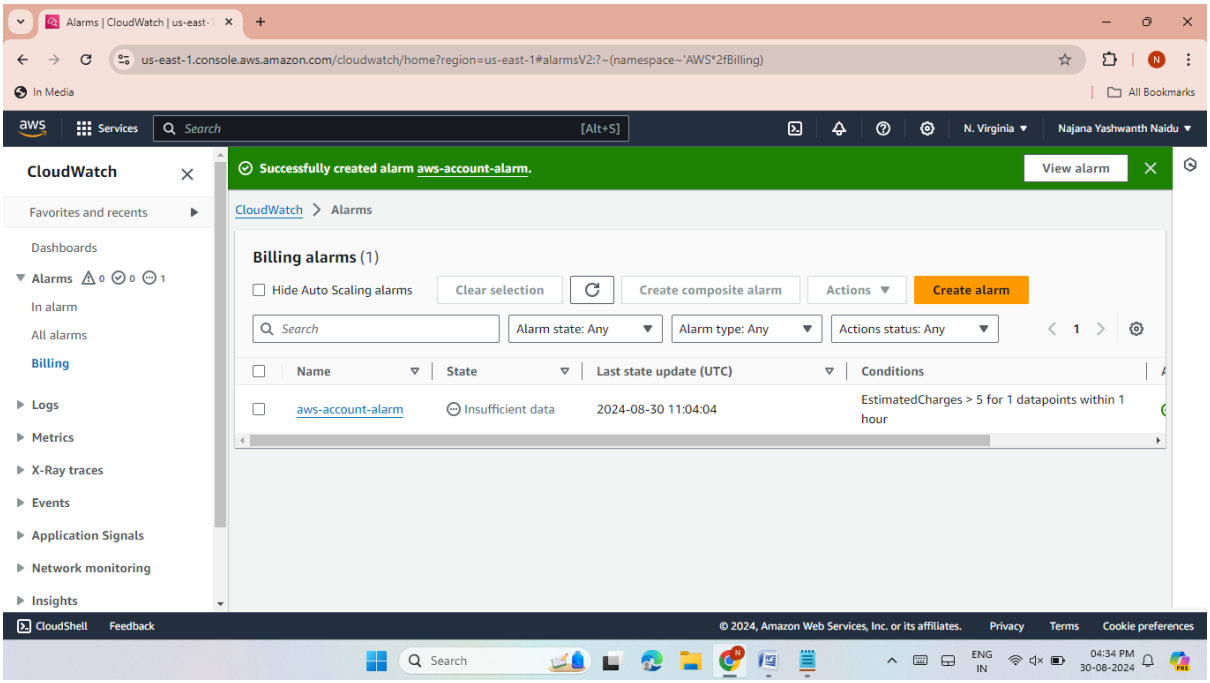
- Select the SNS(simple notification service).
- Enter the email for receiving the notifications.



➢ Enter the name of Alarm.



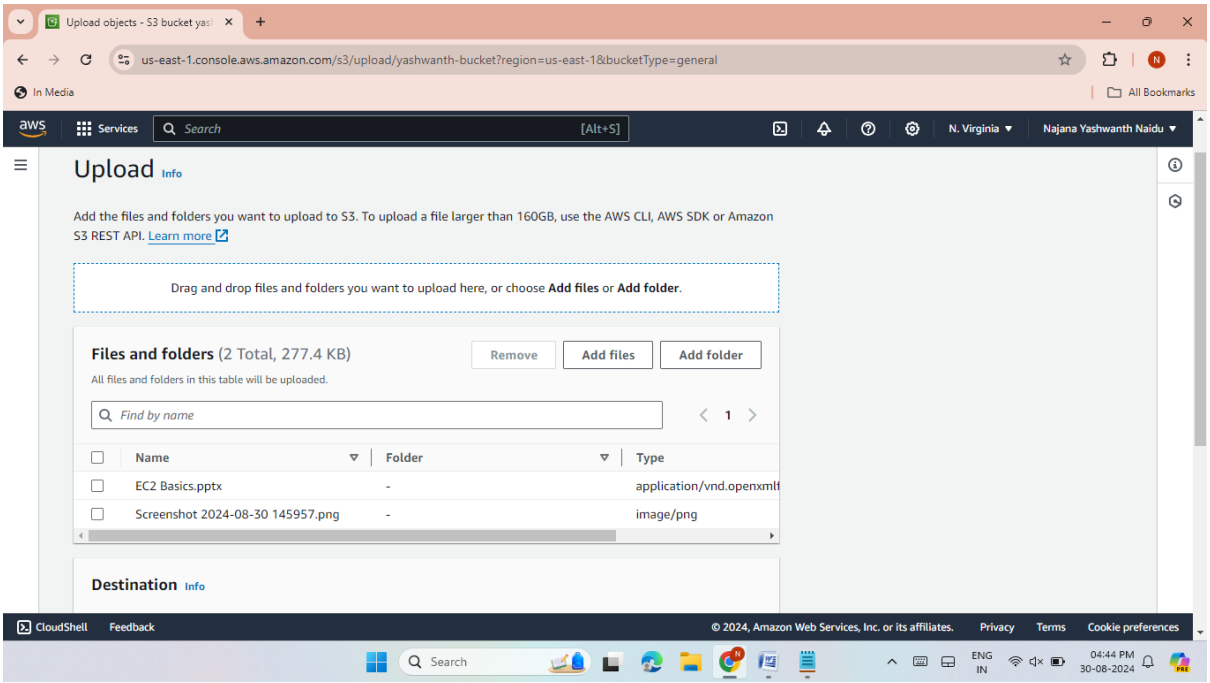
- Here the Billing Alarm is successfully created.



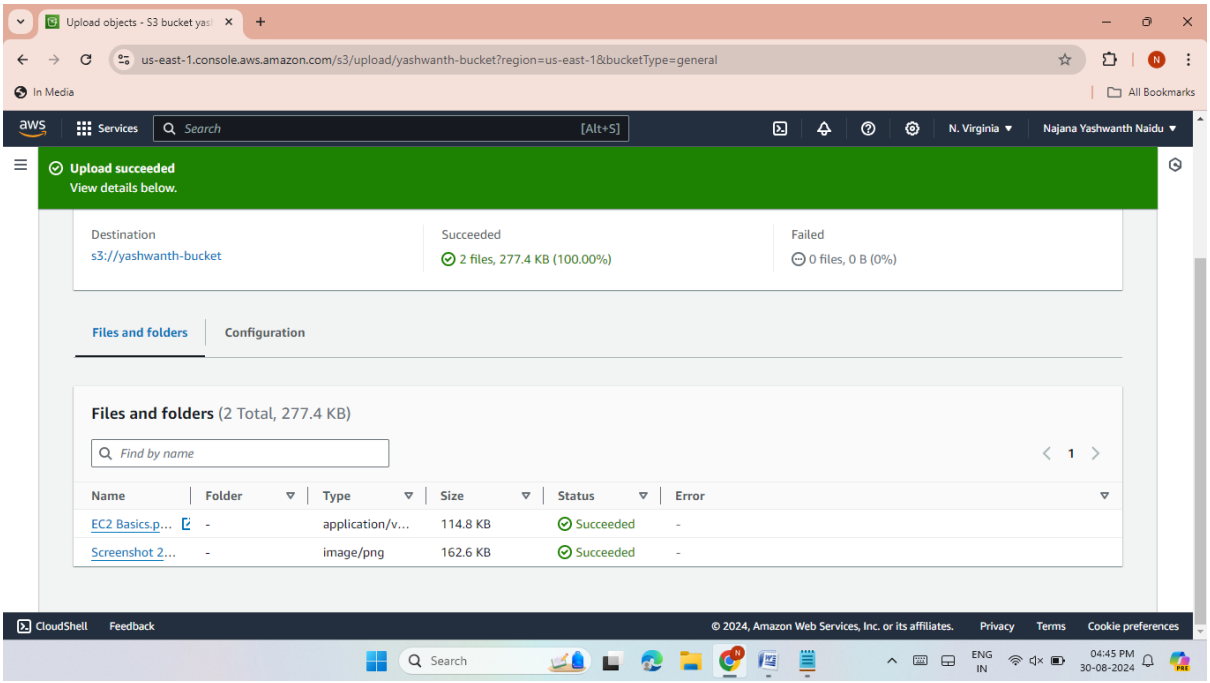
LAB-3 : S3 bucket

Step 1

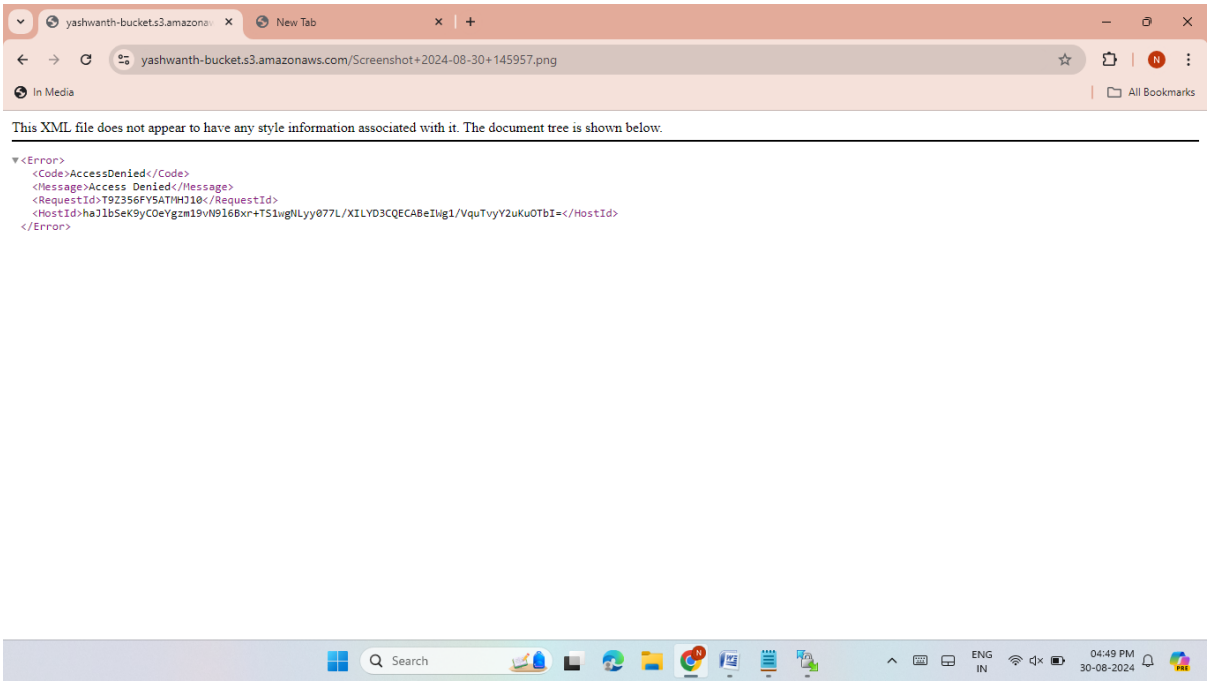
- Create an S3 bucket make sure to give it a unique name.
- Upload some test files/folder in the bucket.



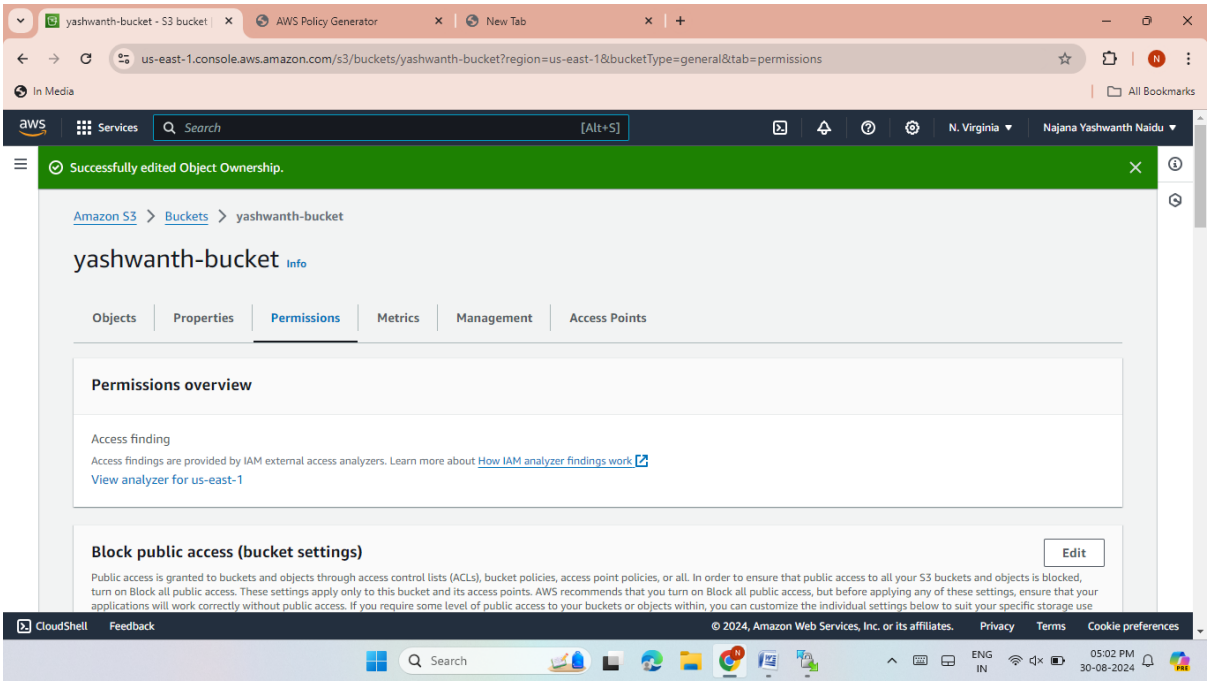
- Successfully uploaded.



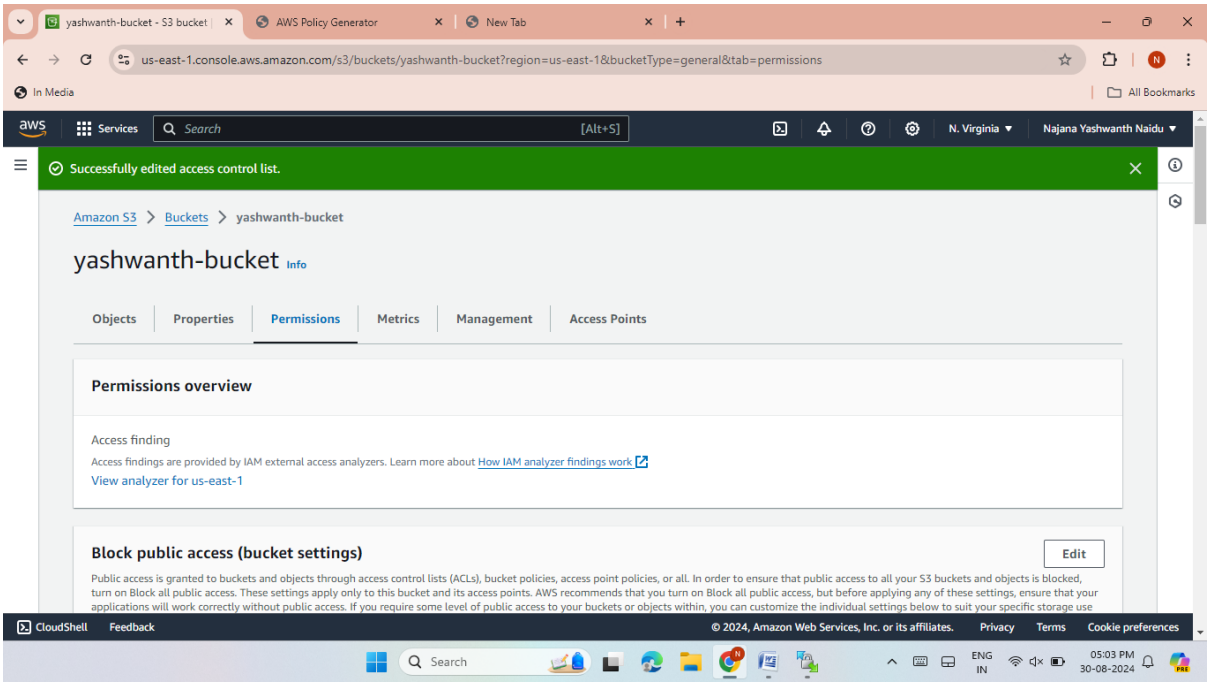
- Access the file over the browser using its URL but It should give error.



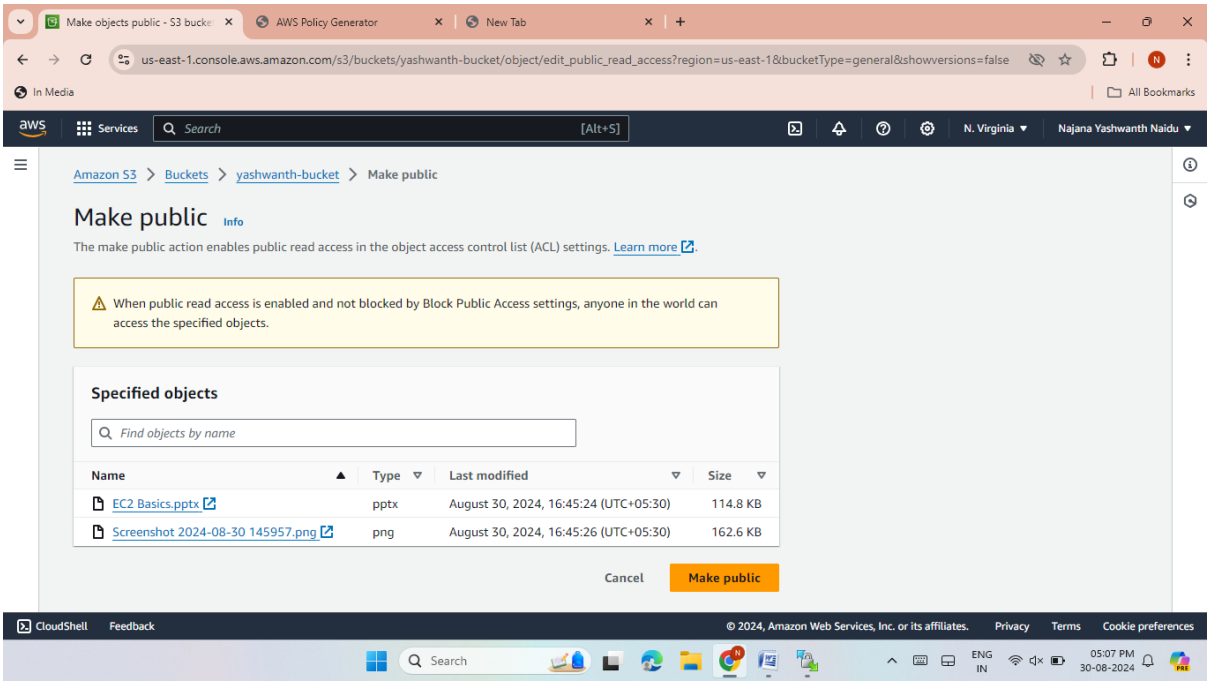
- Check the permissions of the file and the bucket
- First give permissions to the bucket. In permissions tab disabled block public access.



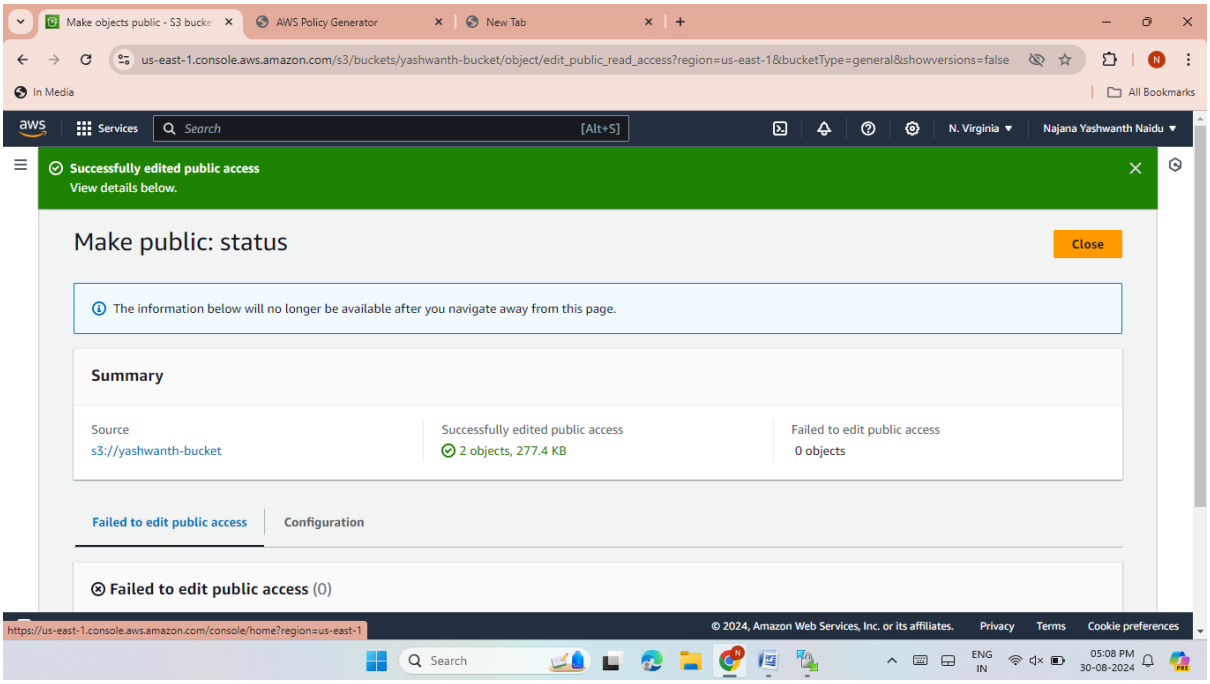
- Then click edit and give public access in Access control list (ACLs).



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

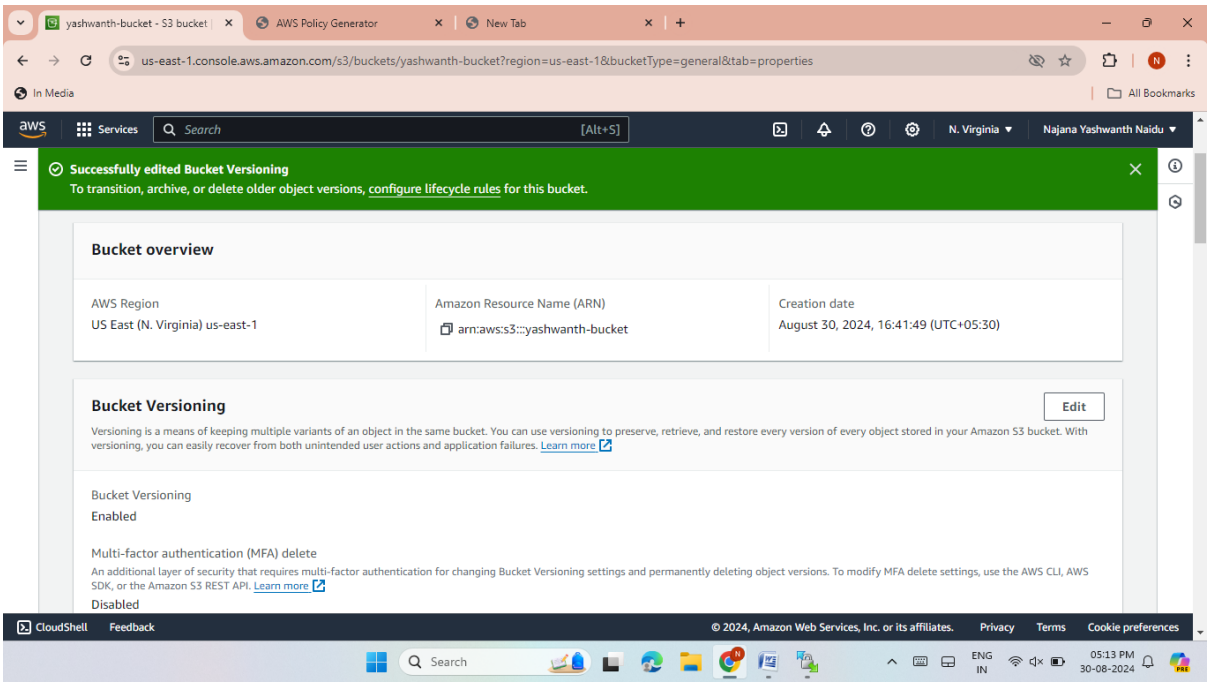


- Here Successfully edited public access. Now we can access the file.

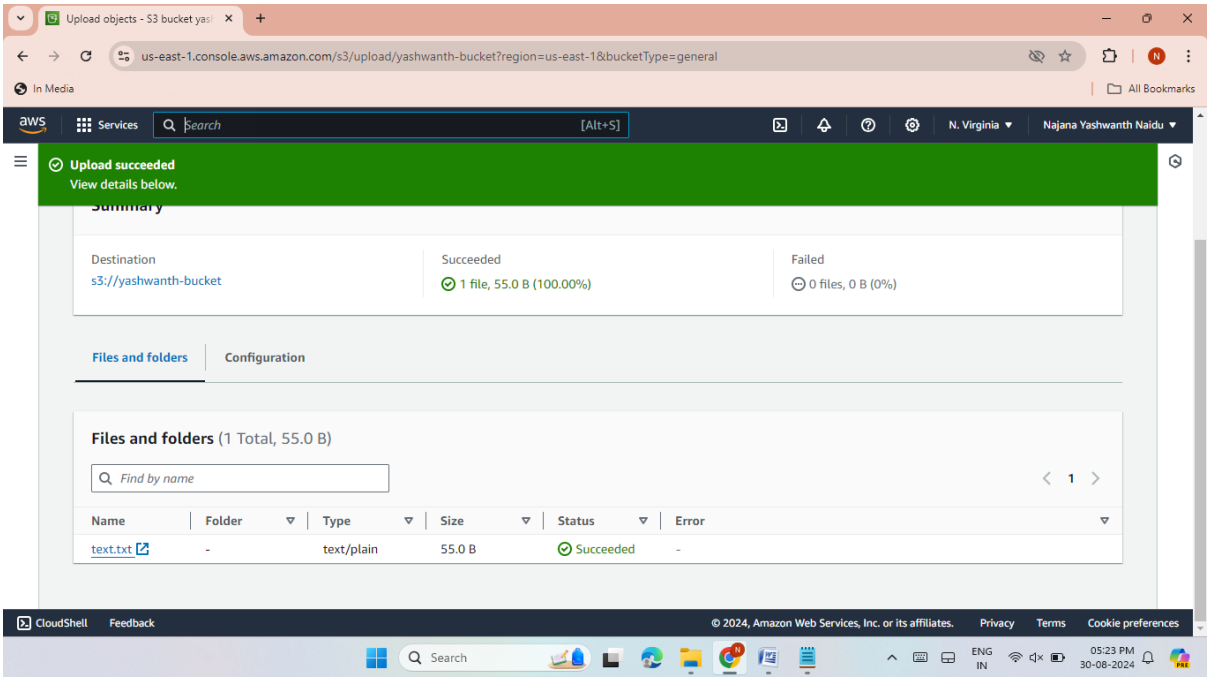


Step 2

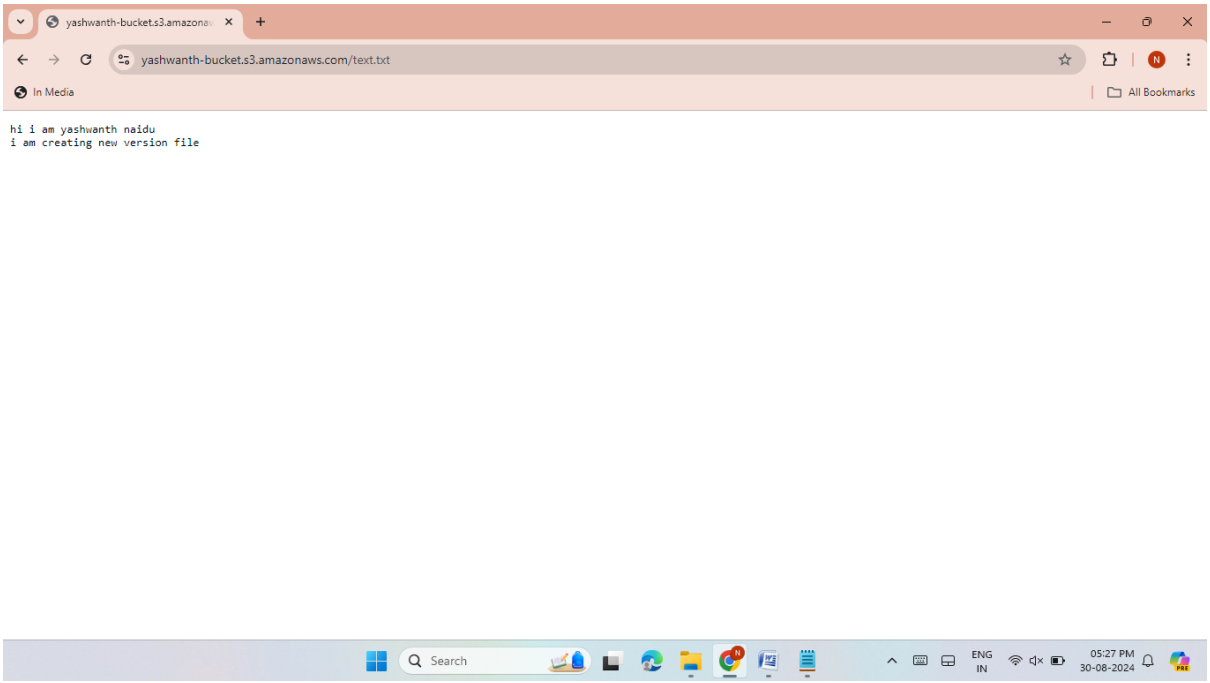
- Enable versioning of the excited bucket.



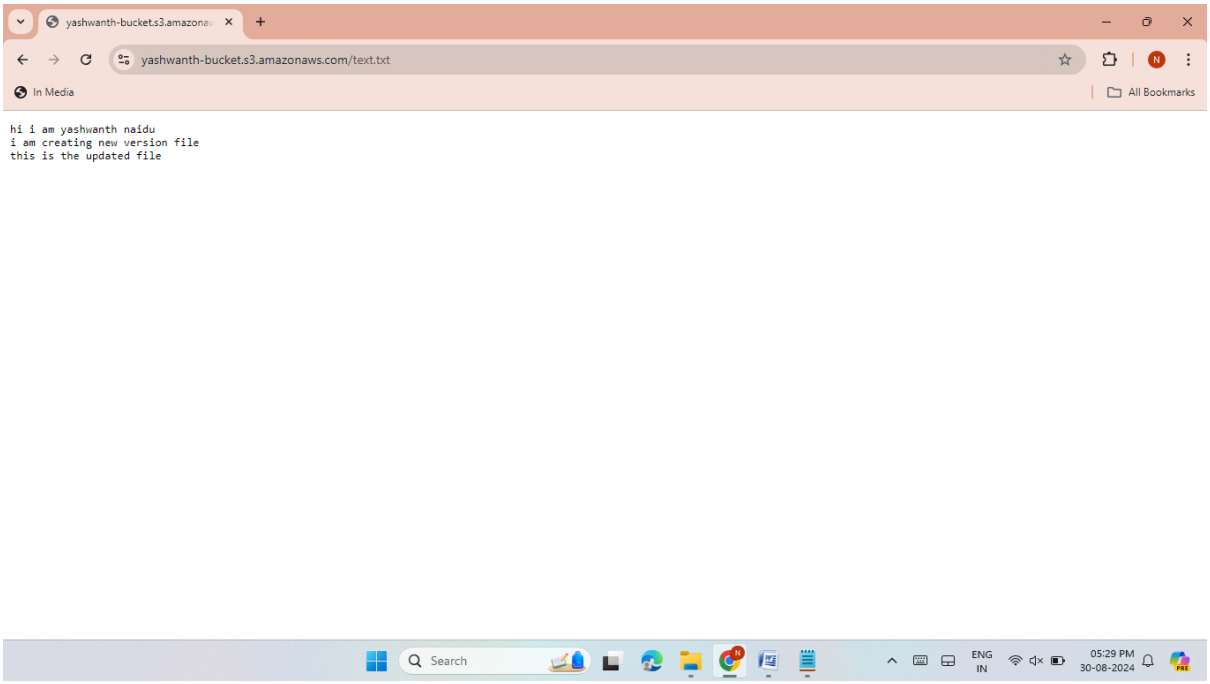
- Create a text file and upload the updated version to the bucket.



- Check the content of the current file.

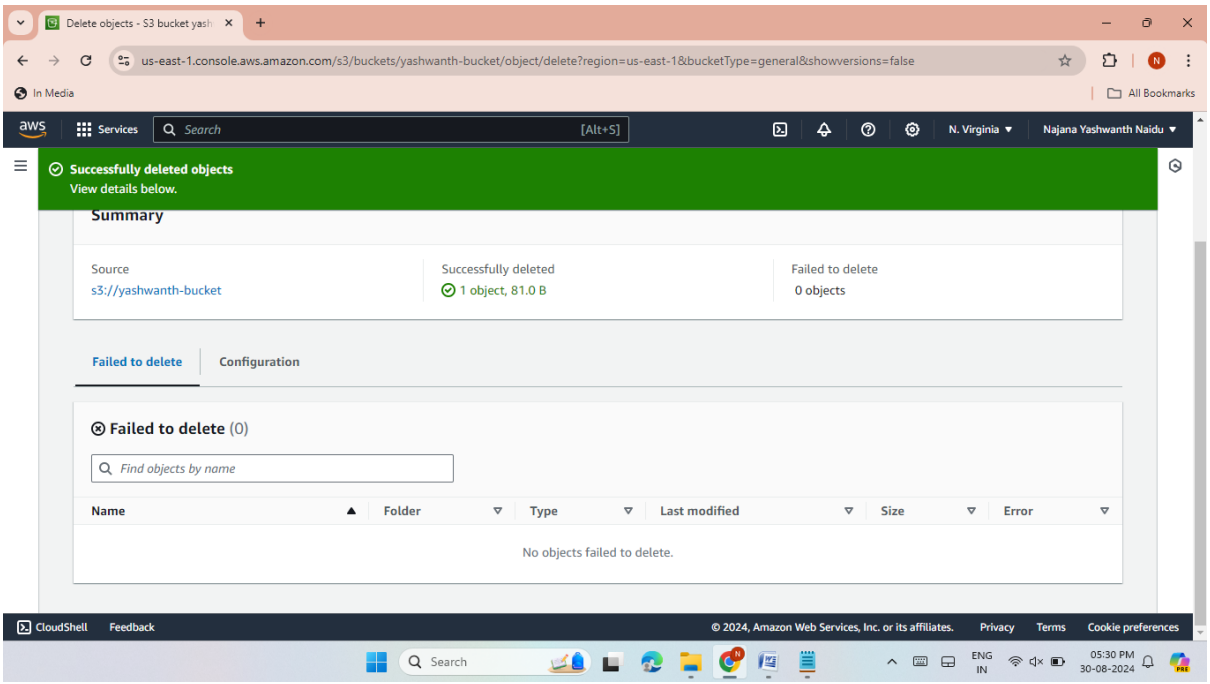


- Here the updated version of the text file.

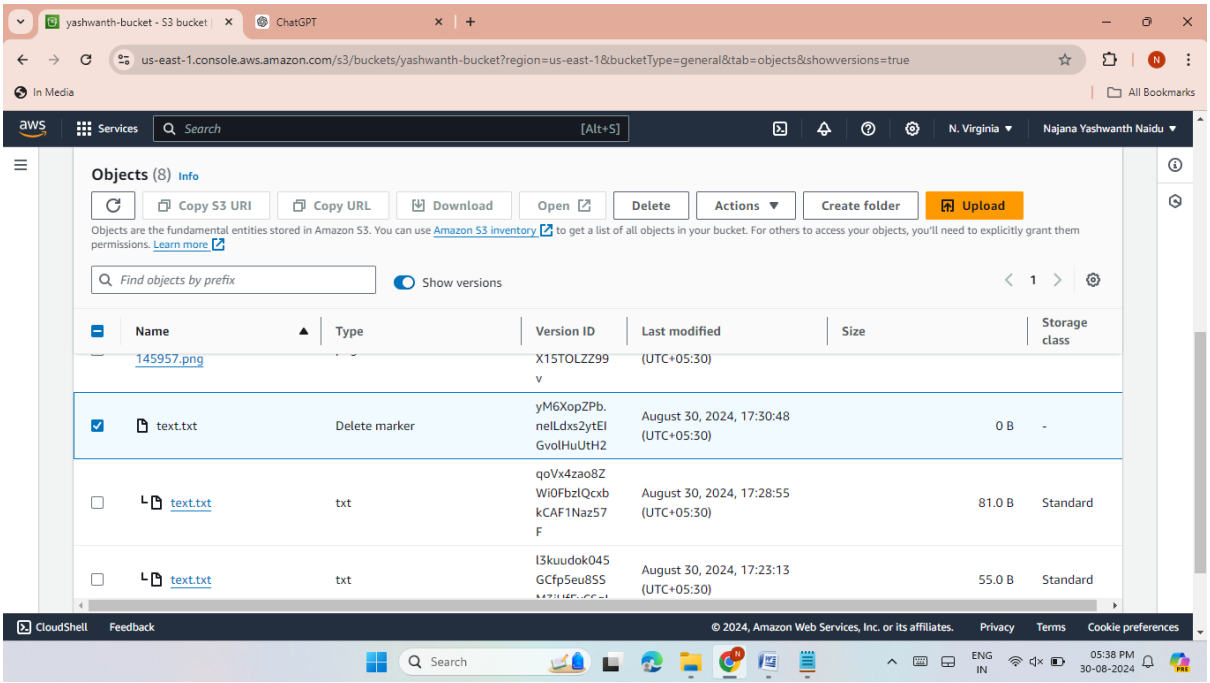


Step 3

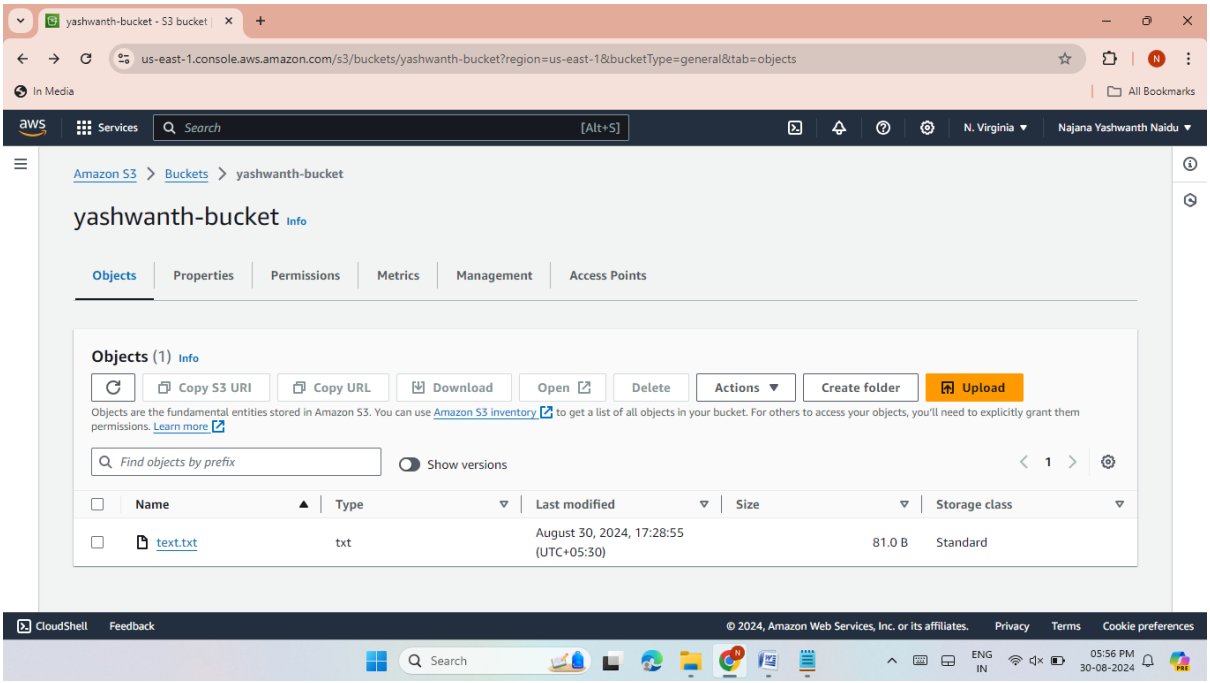
- Delete the text file from the bucket.



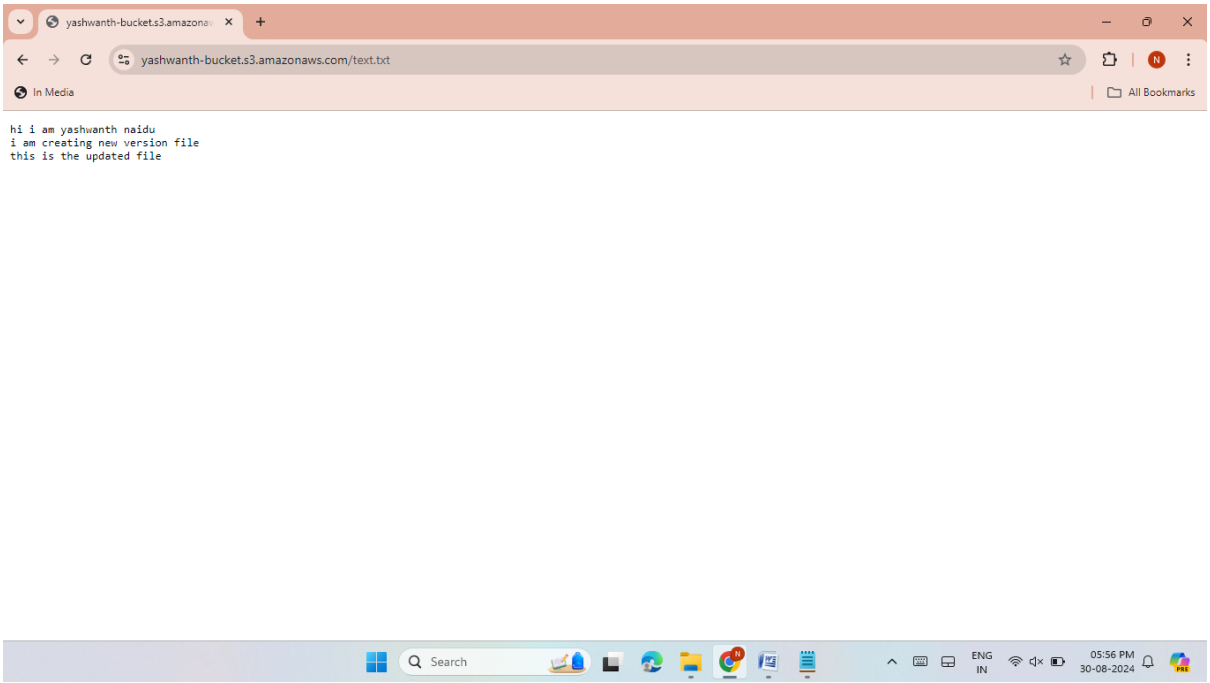
- In objects enable show versions then Check the deleted file.
- Delete the delete marker file.



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

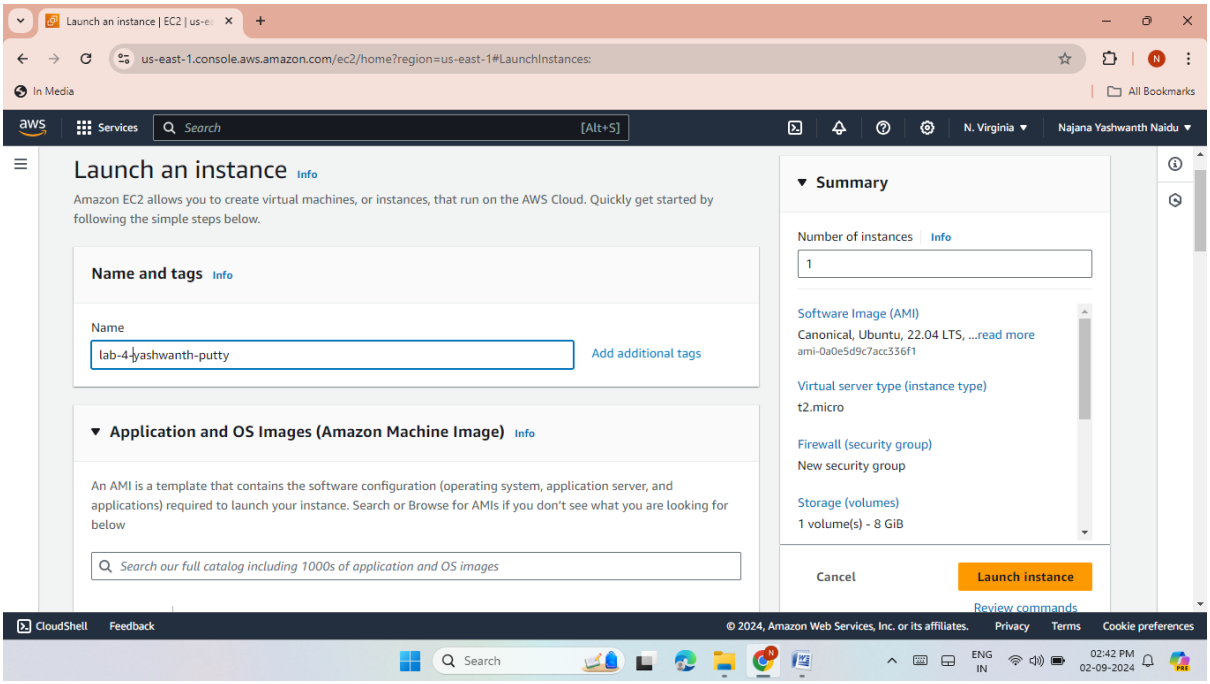


- Then browse the URL it will display the content of the file.

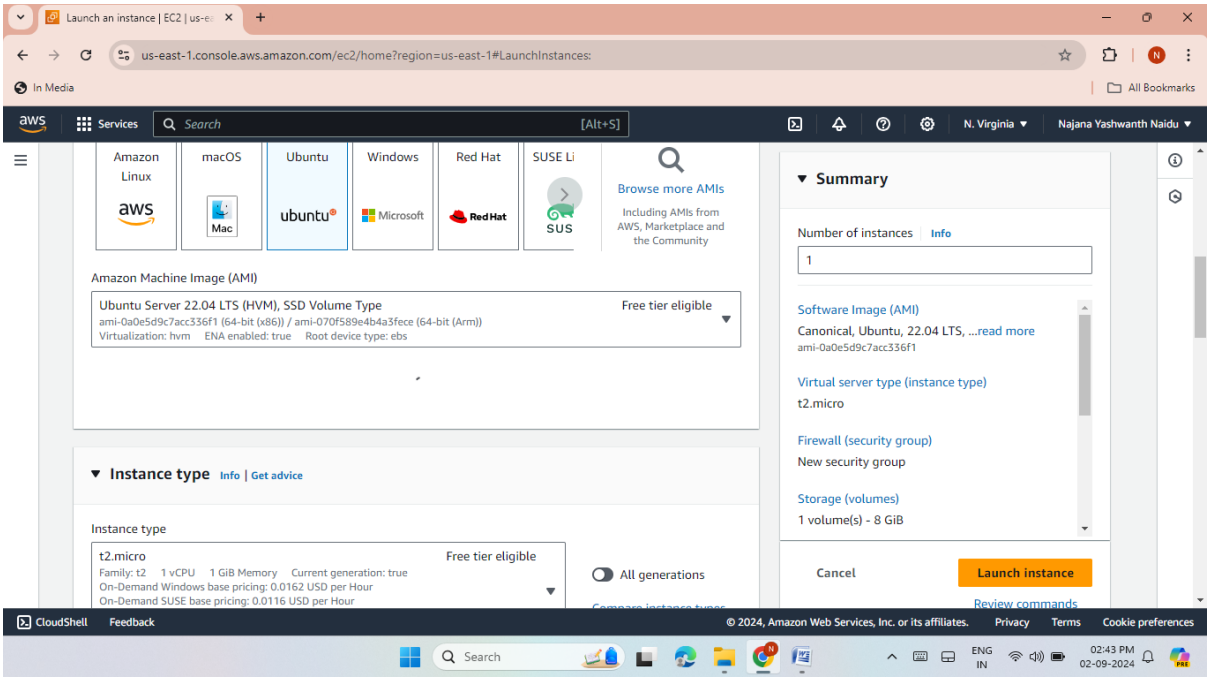


Lab 4 : EC2 INSTANCE

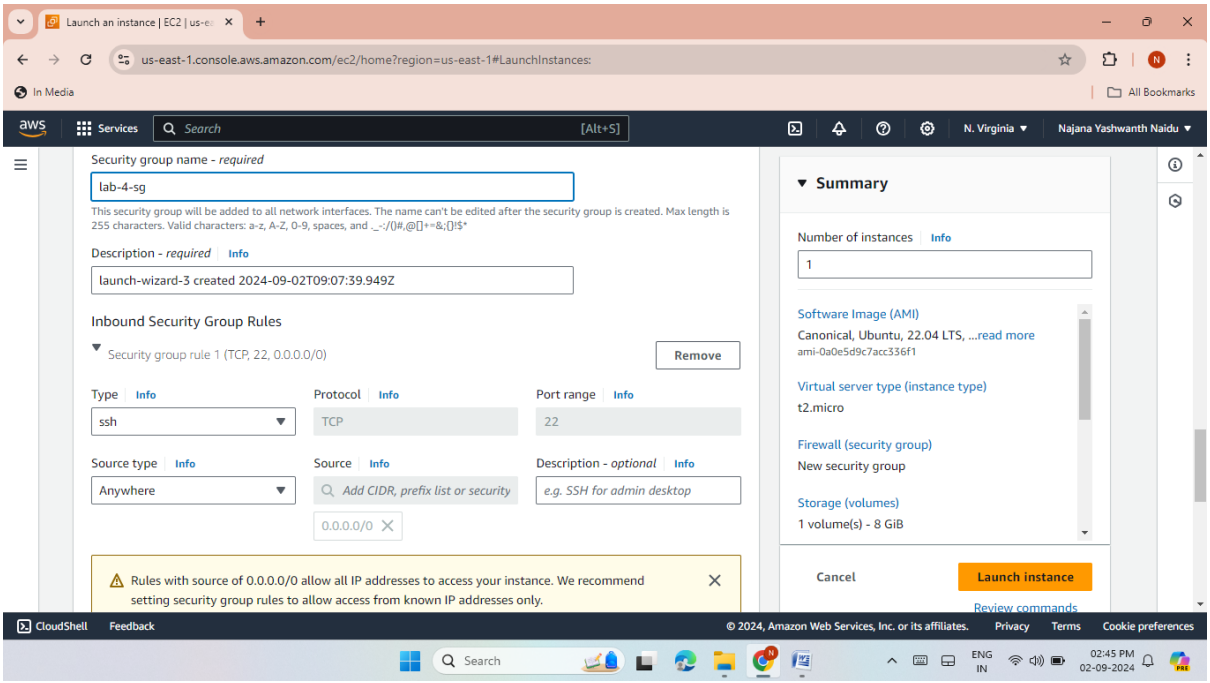
- Go to EC2 then Launch the EC-2 instance.



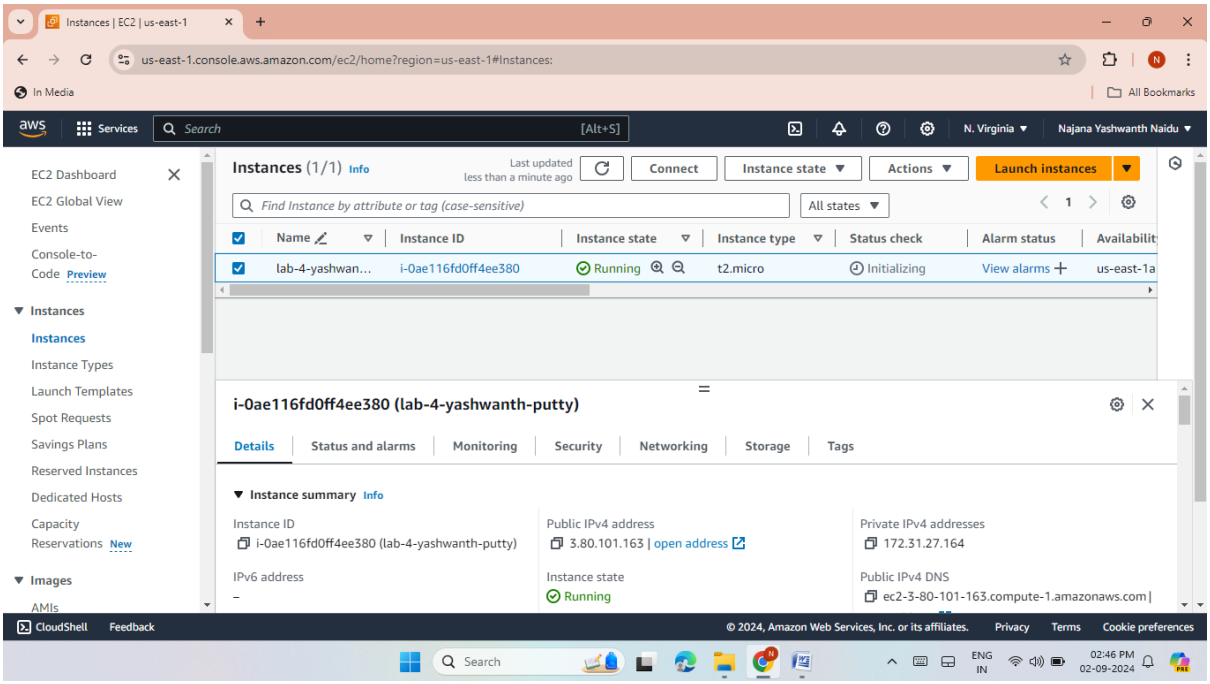
- Give the instance type t2.micro with the Ubuntu server.(free tier)



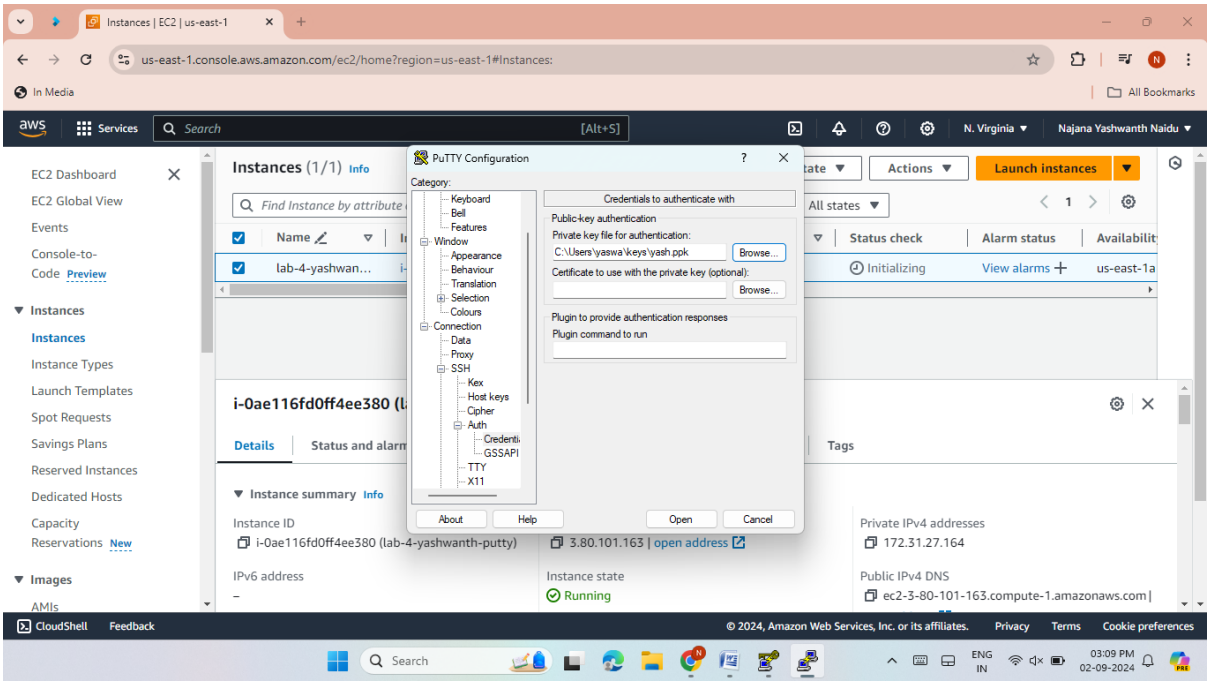
➢ Allow the SSH port in the security group so that you can access it from GitBash or putty etc.



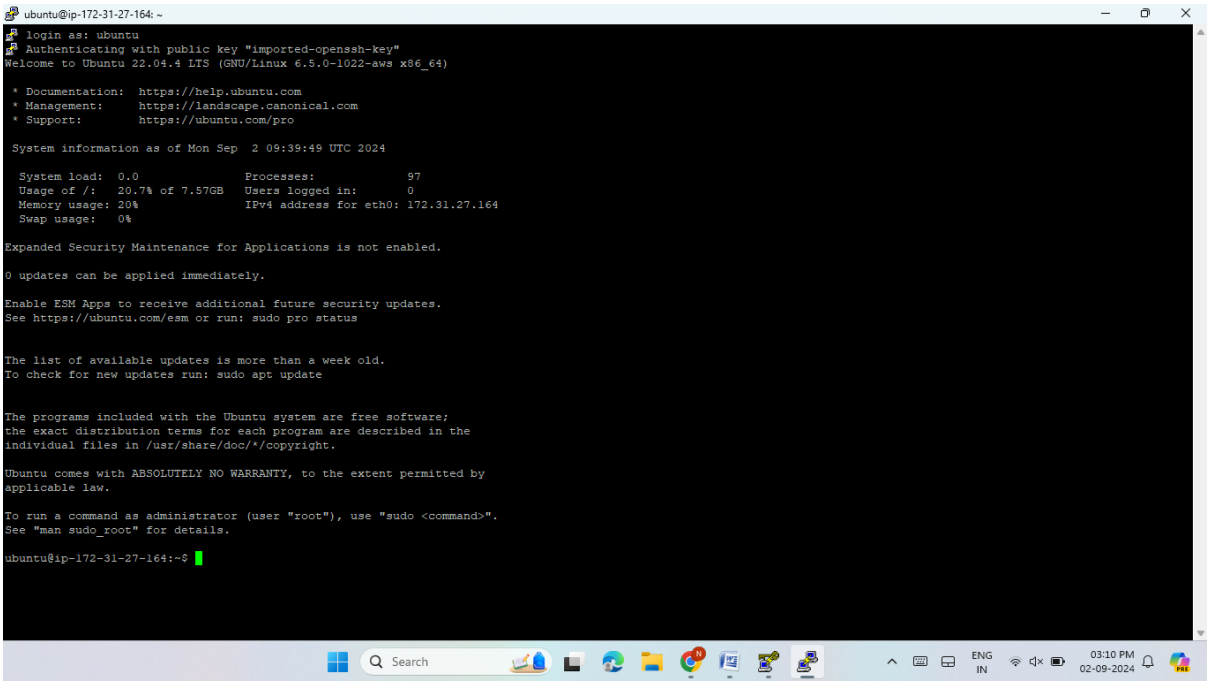
➢ Here the instance was launched.



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

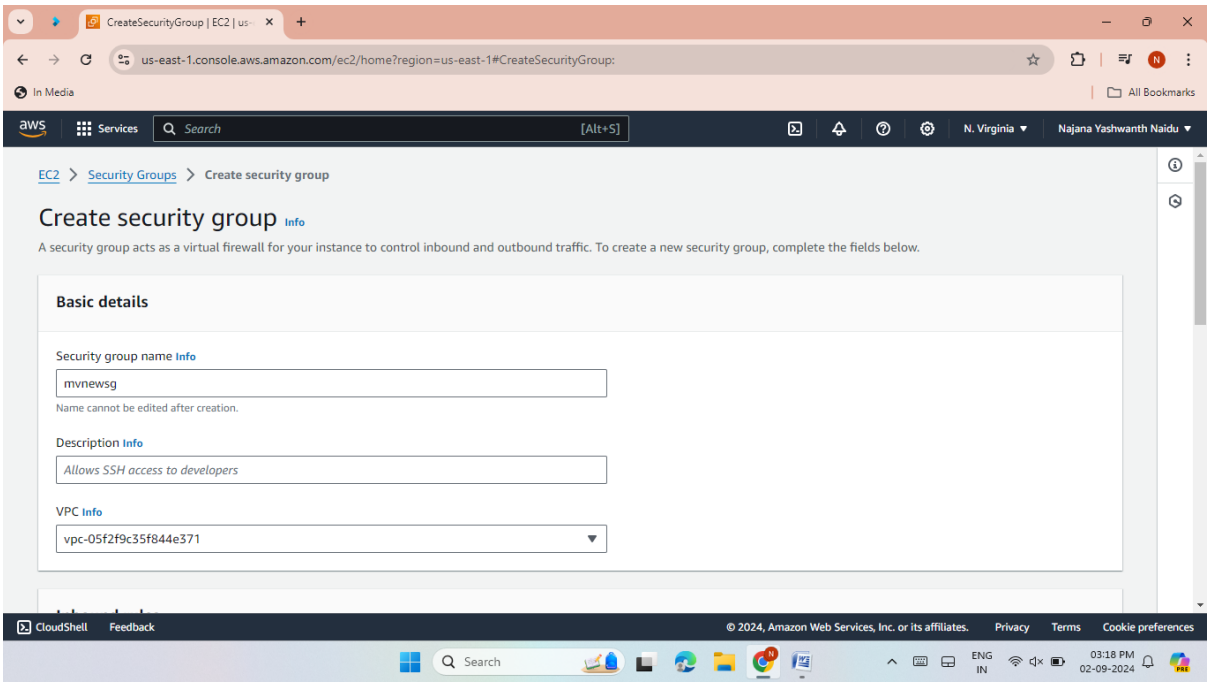


- Now I have access instance using PPK file in putty software.

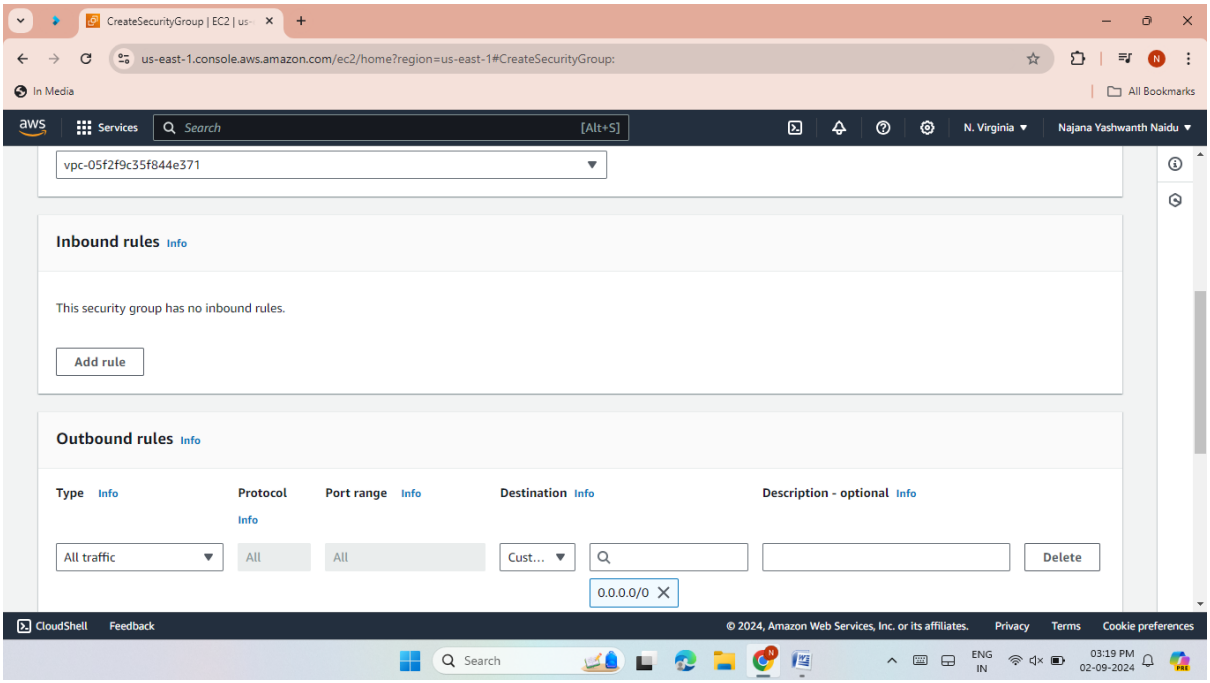


Lab 5 : SECURITY GROUPS

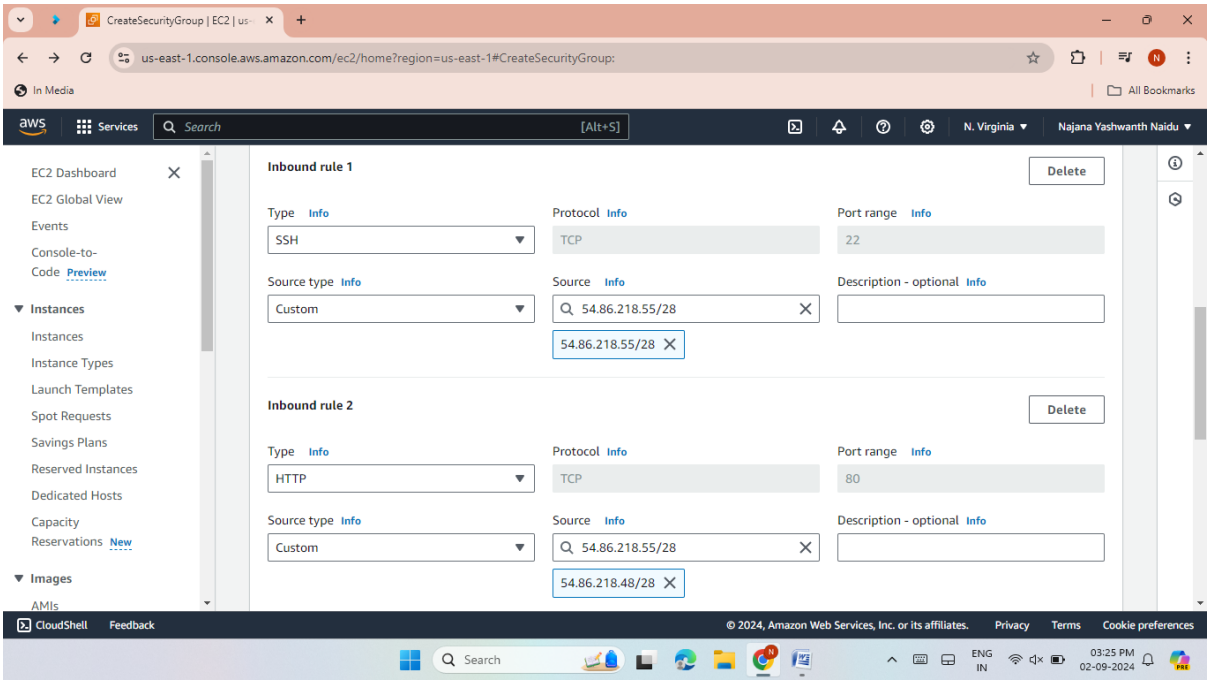
- Create a new Security Group and name it as “mynewsg”.



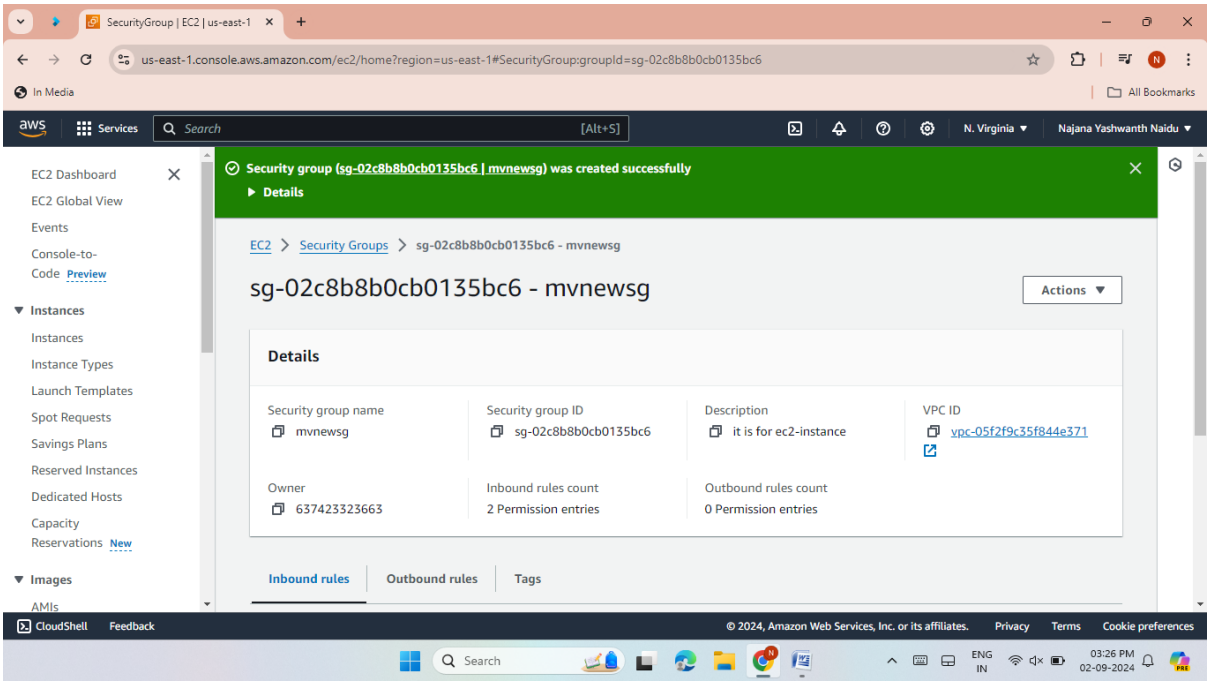
- Check the default rules in this Security Group.



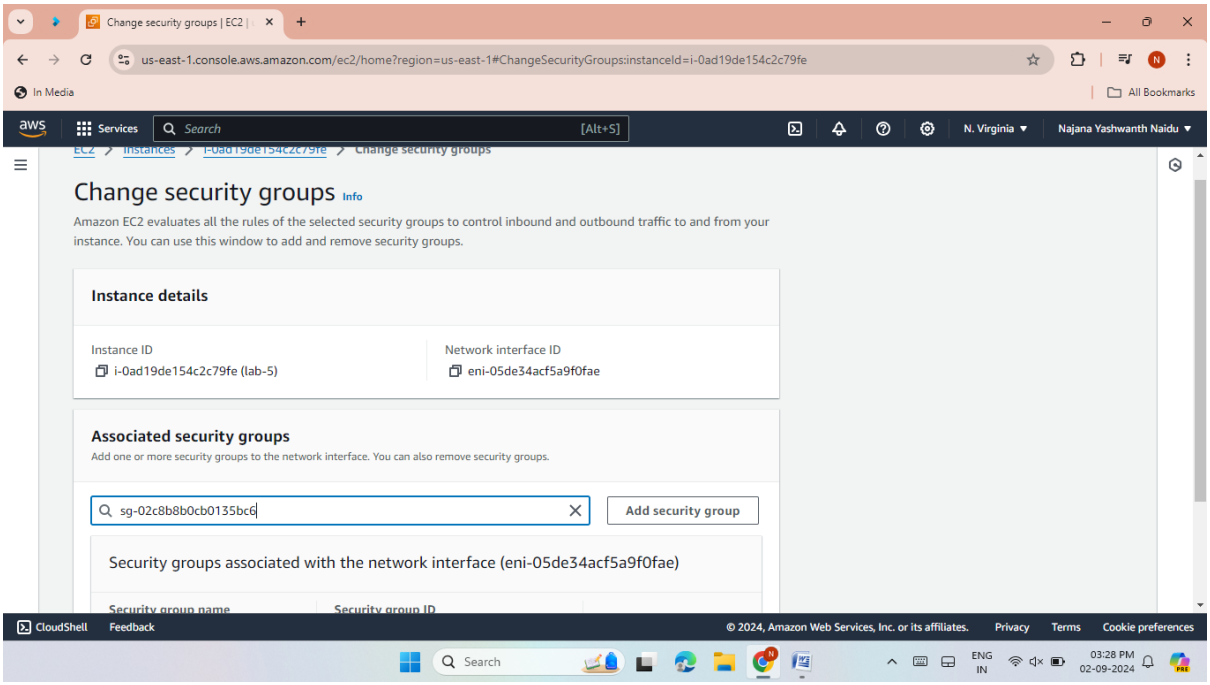
- Allowing inbound Port HTTP-80 and SSH-22 from my IP address.
- Select a range (IP/28) so that any changes in my dynamic IP will not impact the rule.



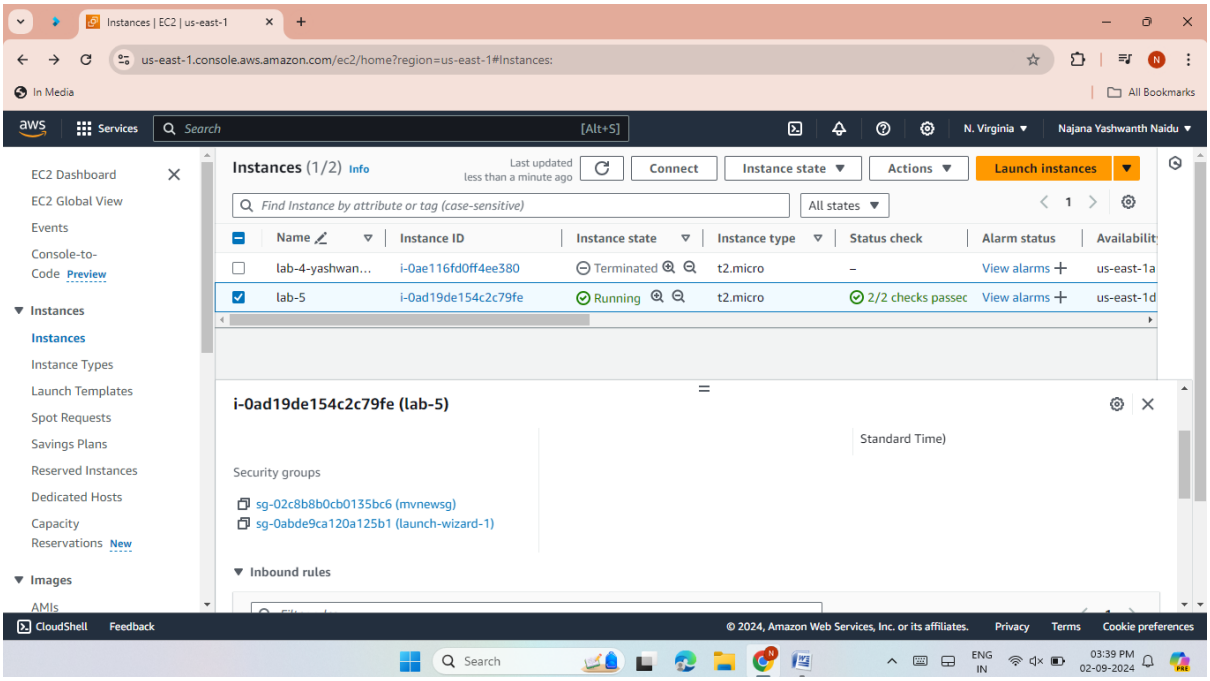
- The Security Group was created successfully.



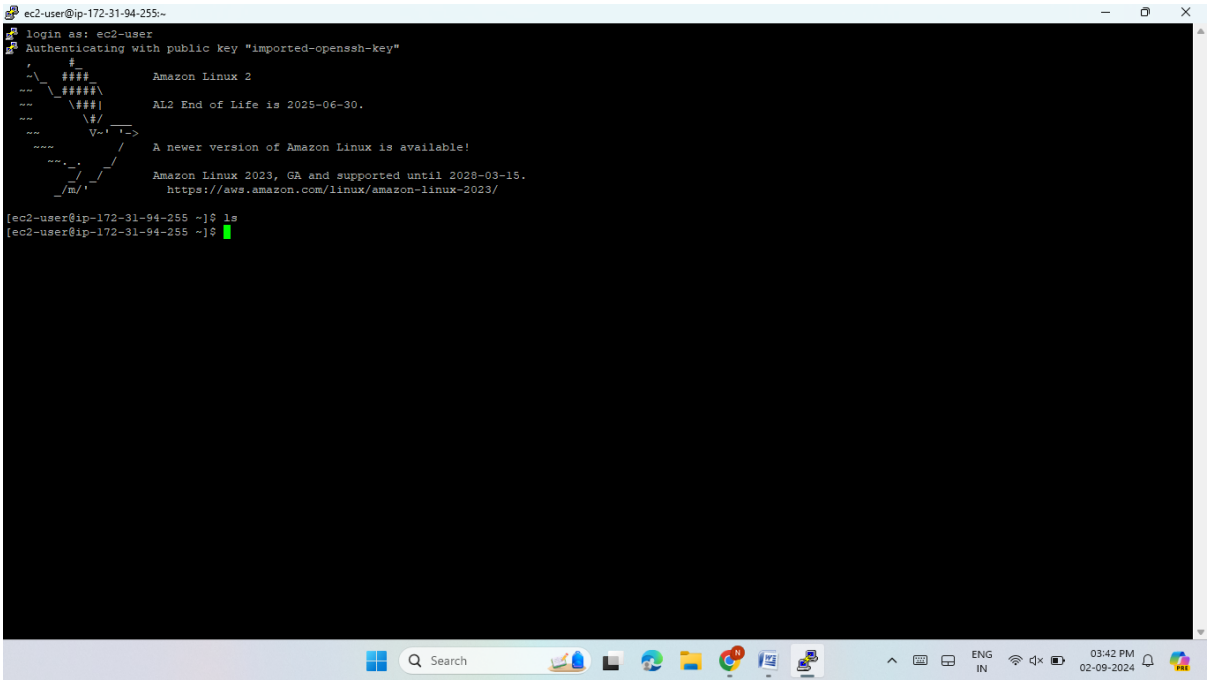
- Attach this new Security Group to my existing EC2 Instance.
- Go to EC2 Dashboard.
- Click Instances and select the instance you want to modify.
- Under Actions, select Networking → Change Security Groups.



- Attached to this instance.

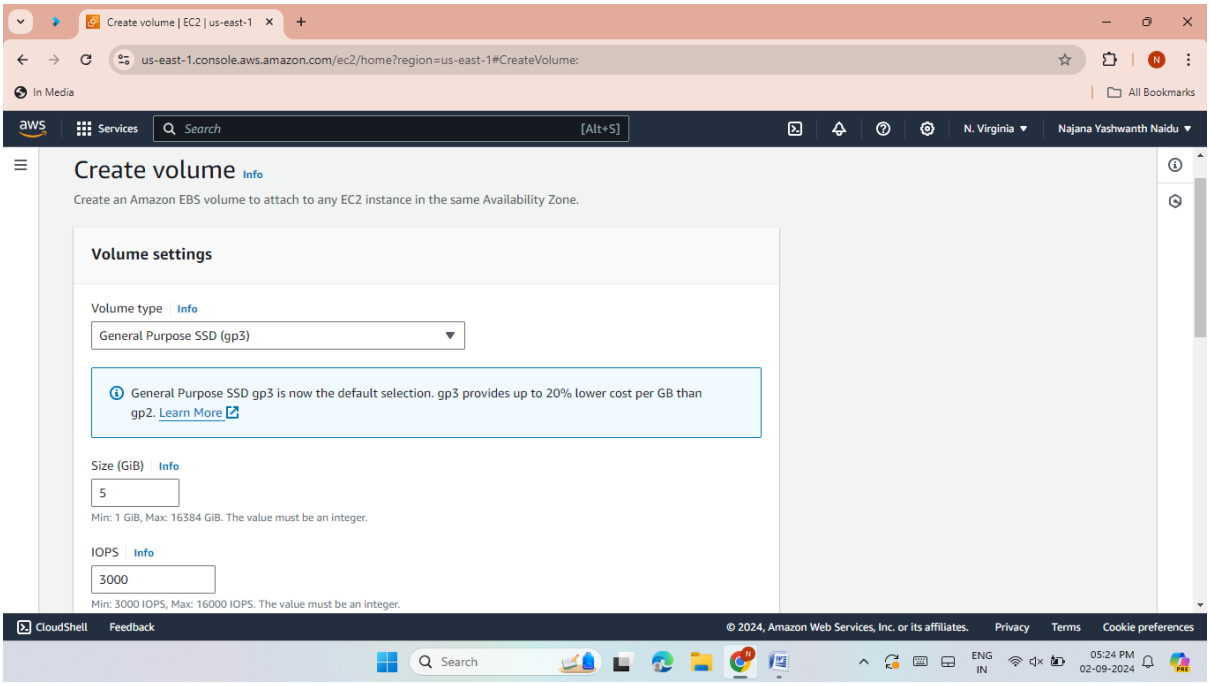


- Try to access the server with help of SSH.
- It successfully accessed.

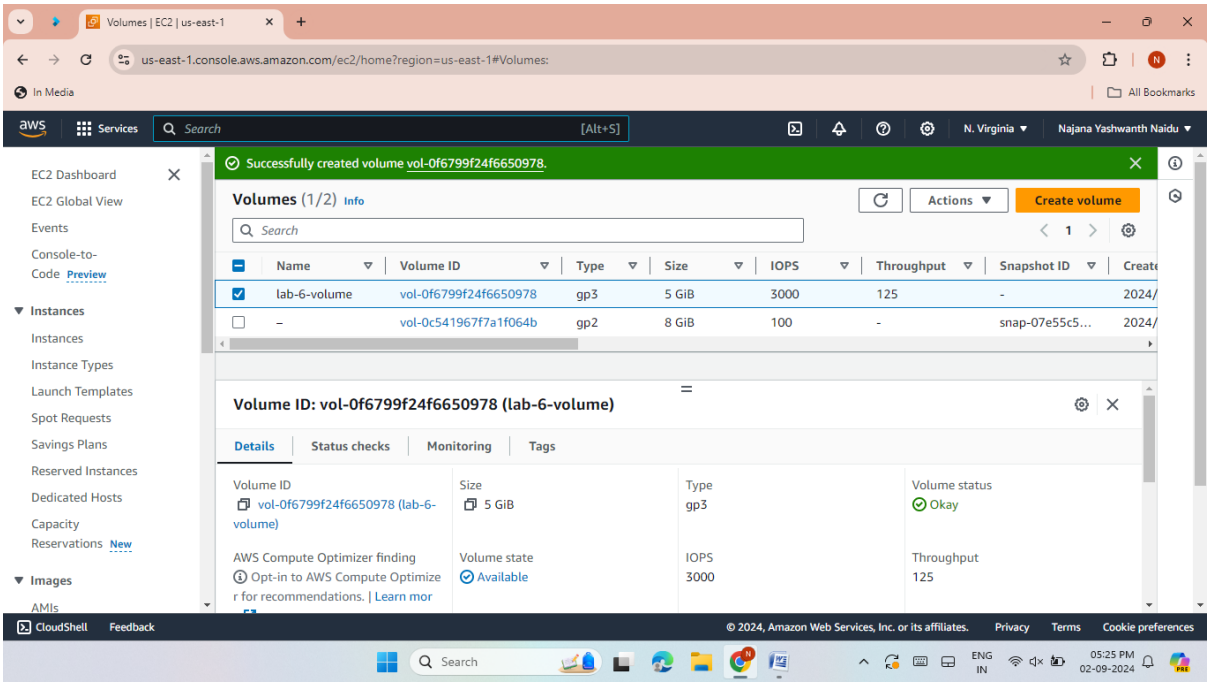


LAB 6 : VOLUMES AND SNAPSHOTS

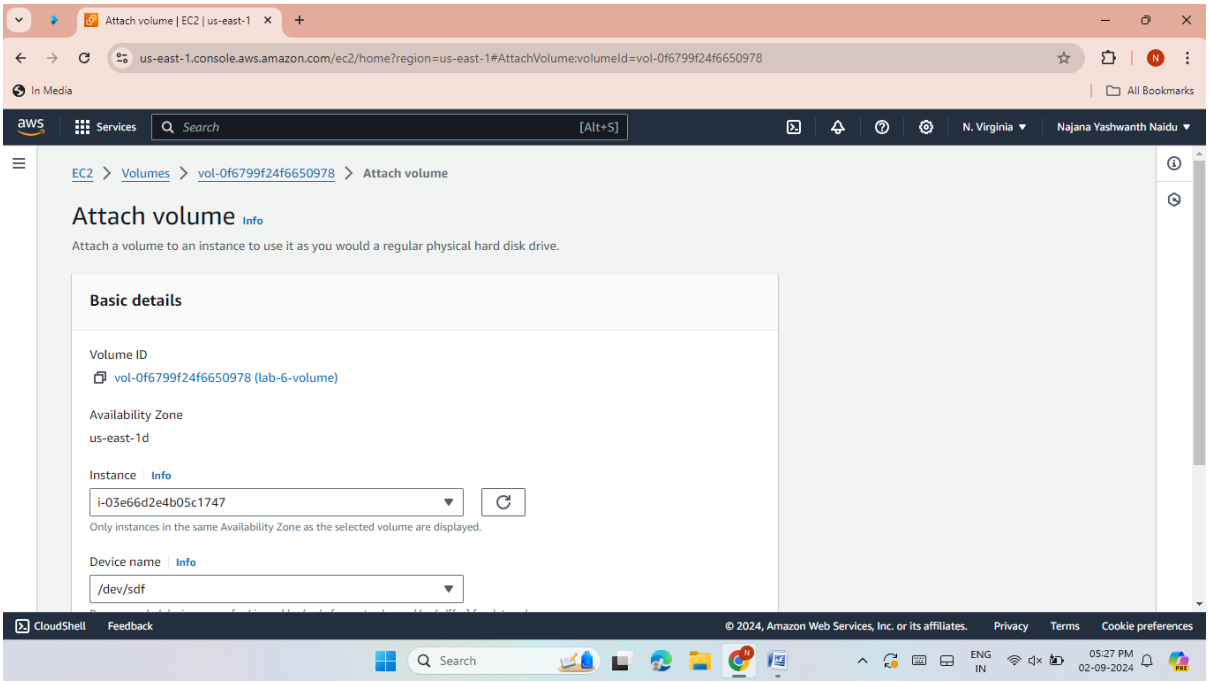
- Create one 5GB volume.
- Select the same Availability Zone as your running EC2 instance.



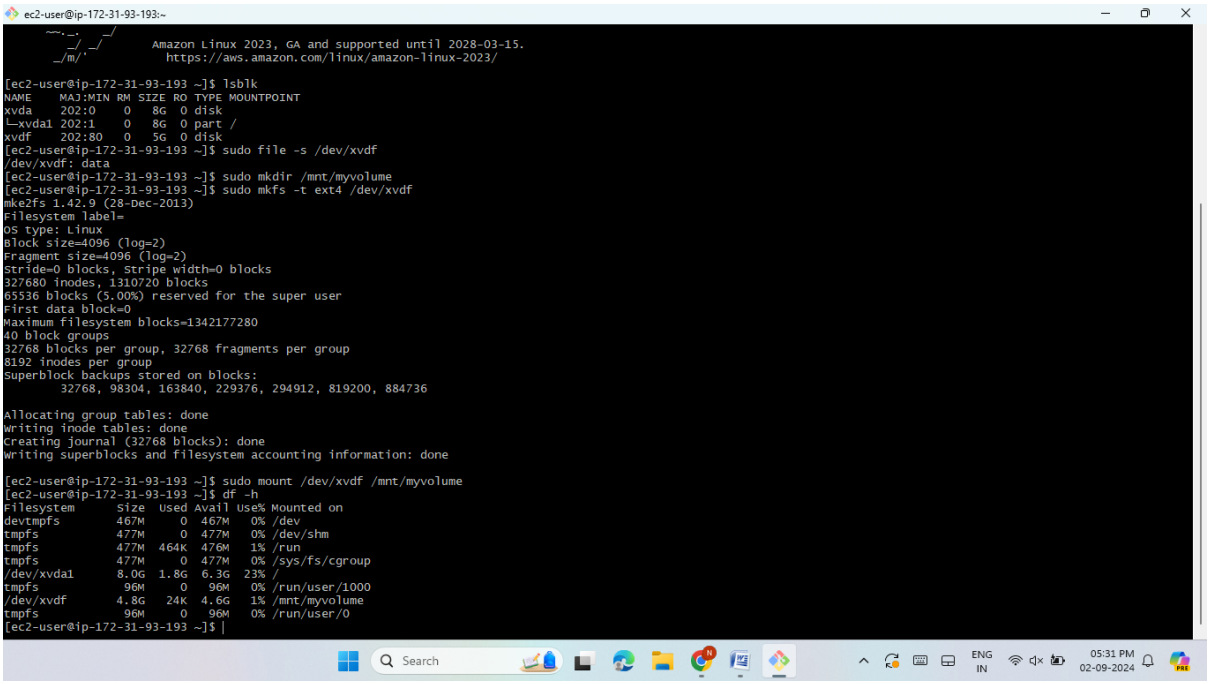
- Here the volume is Successfully created.



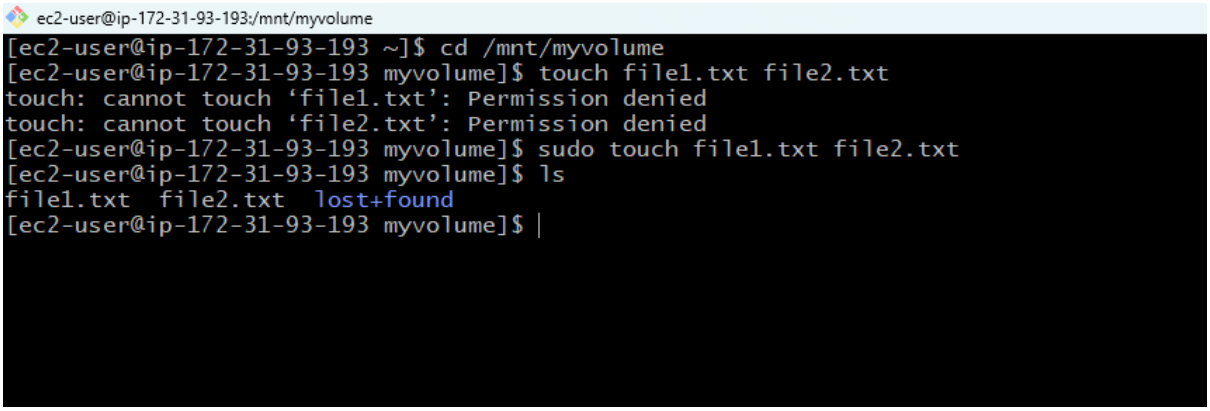
- Attach it with the running ec2 instance.
- Select the volume from the list and click Actions → Attach Volume.



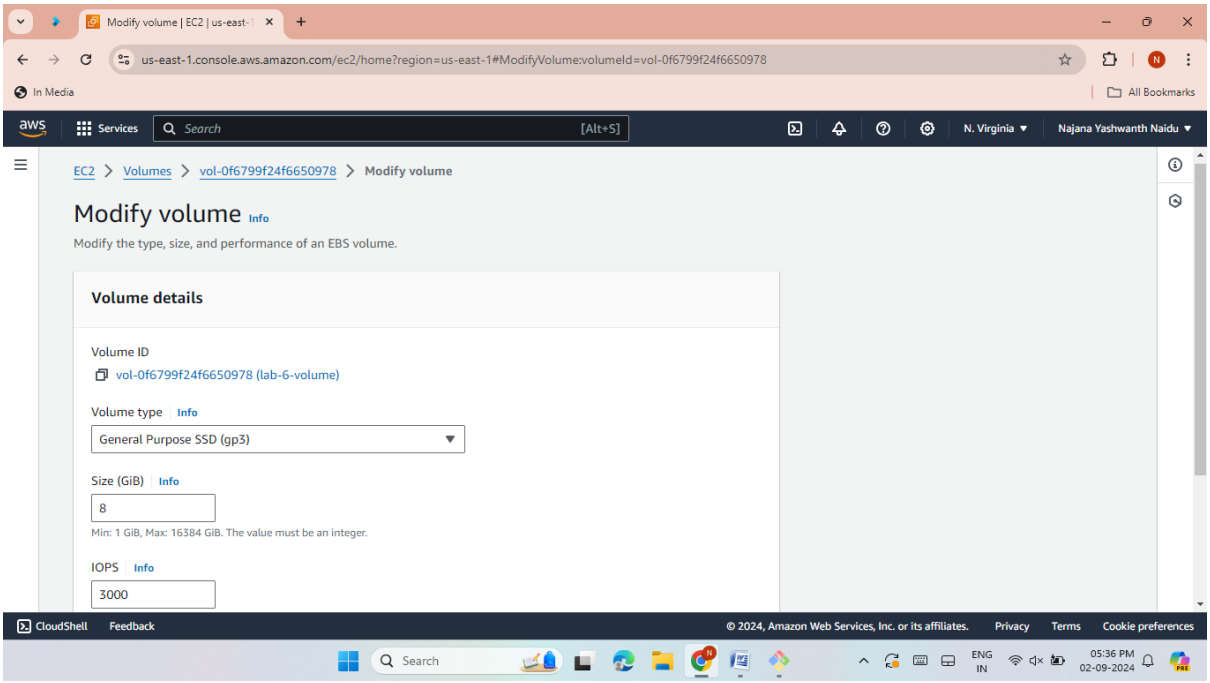
- Put some data in this volume like some testing files
- Connect the ec2 instance in the terminal using ssh command.
- Run the following command to check if the volume is attached OR not.
- Lsblk
- Create directory with the mkdir command (/mnt/myvolume)
- Then mount the path with the command Sudo mount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



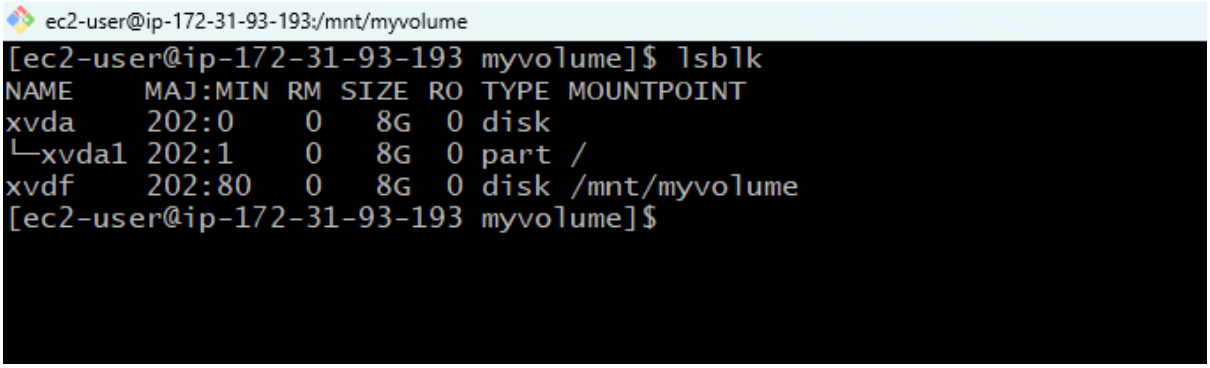
- Put Some Data in the myvolume directoery.
- Create a some text files using touch command.



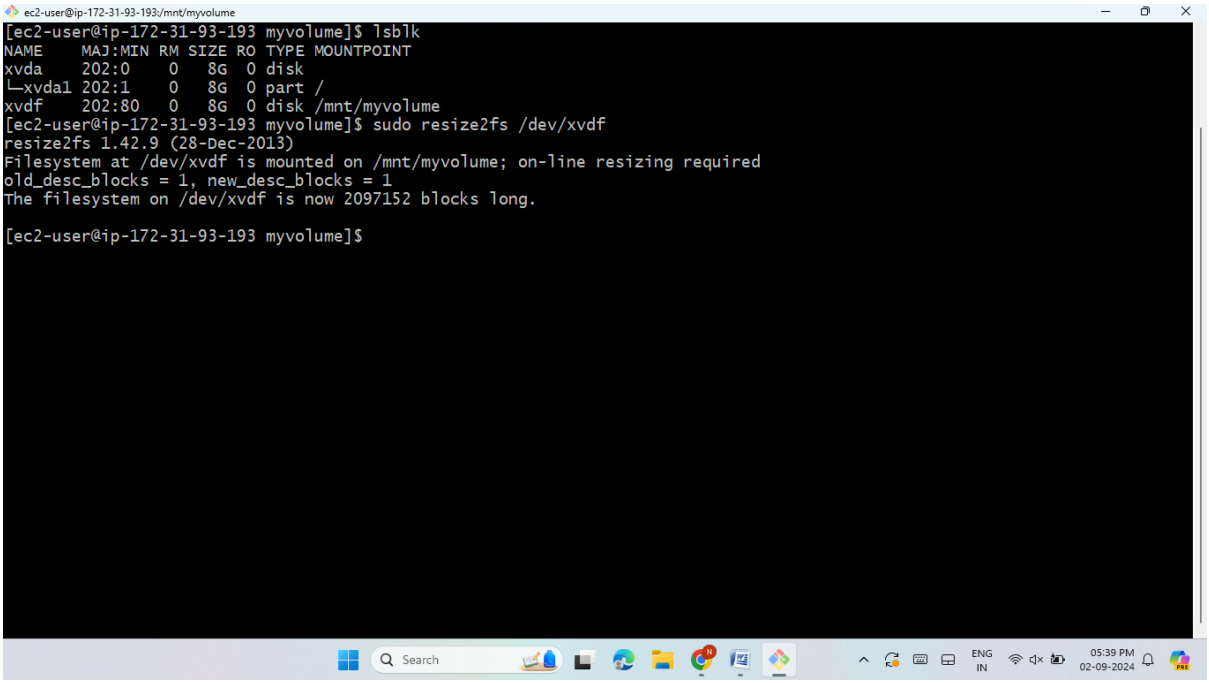
- Increase the size of the this volume it 8GB.
- Select the volume created (5 GiB) and click Actions → Modify Volume.
- Change the size to 8 GiB and click Modify.



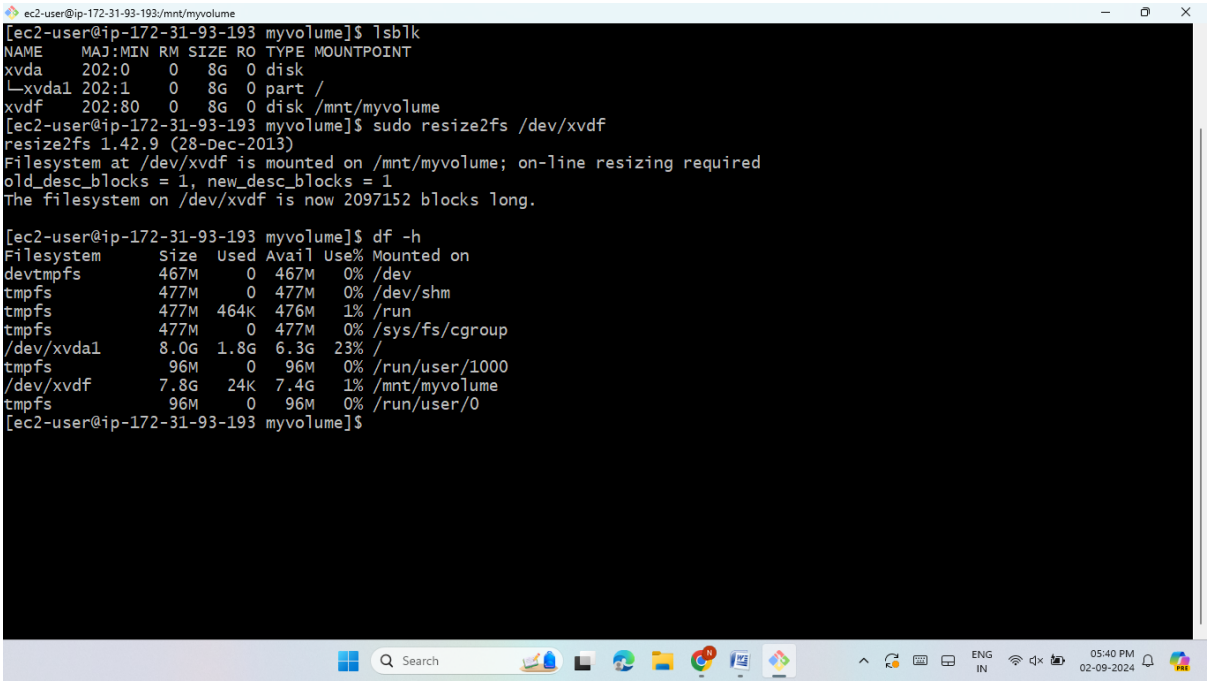
- Inside the EC2 instance, run the following command to ensure the volume size is updated →lsblk.



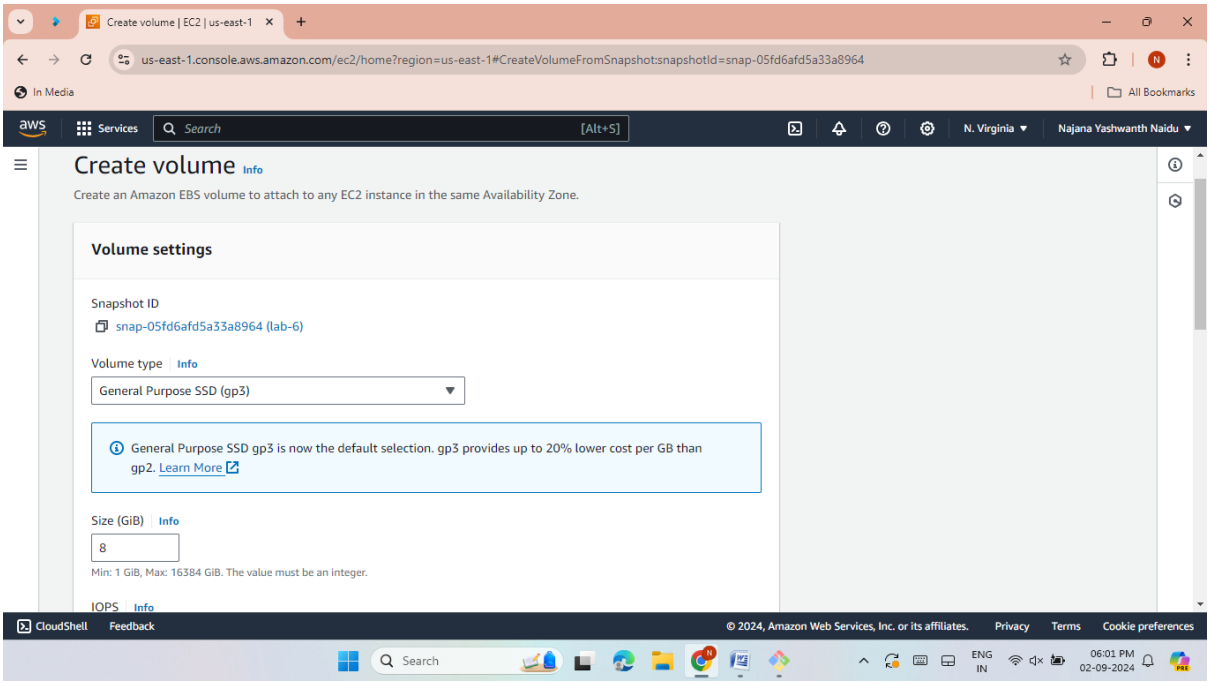
- Extend the size of this of this volume inside the Linux machine
- The following command is sudo resize2fs /dev/xvdf.



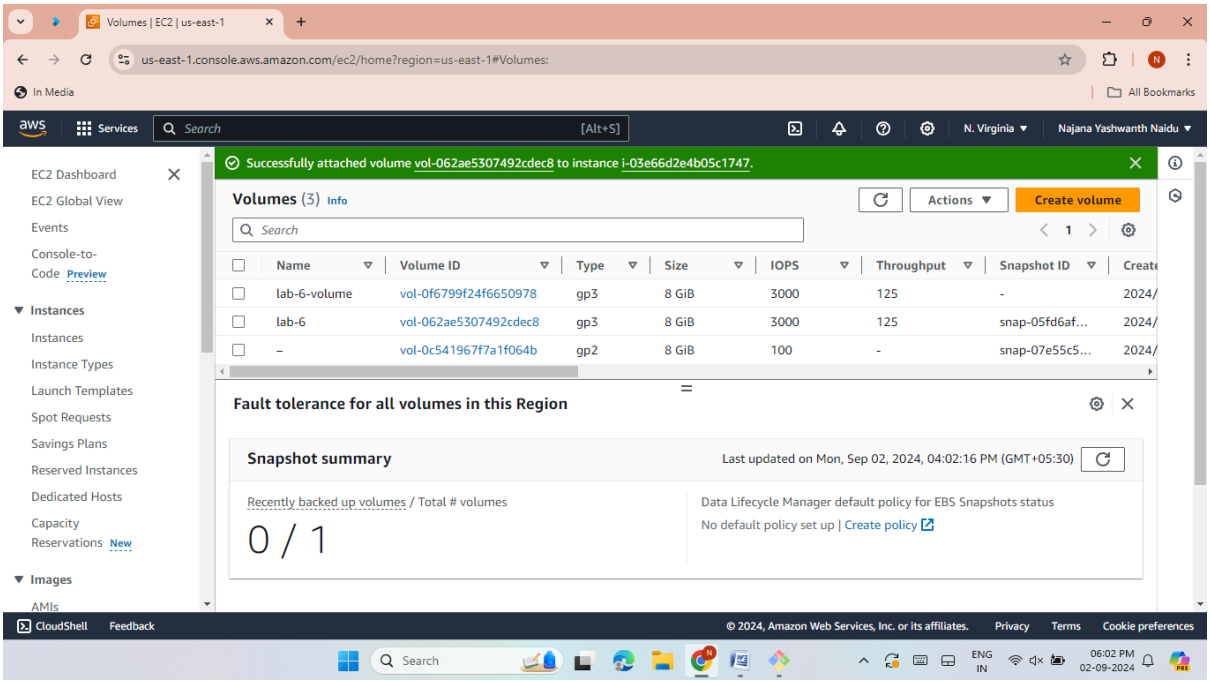
- Take a screen shot of this volume and delete it.



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

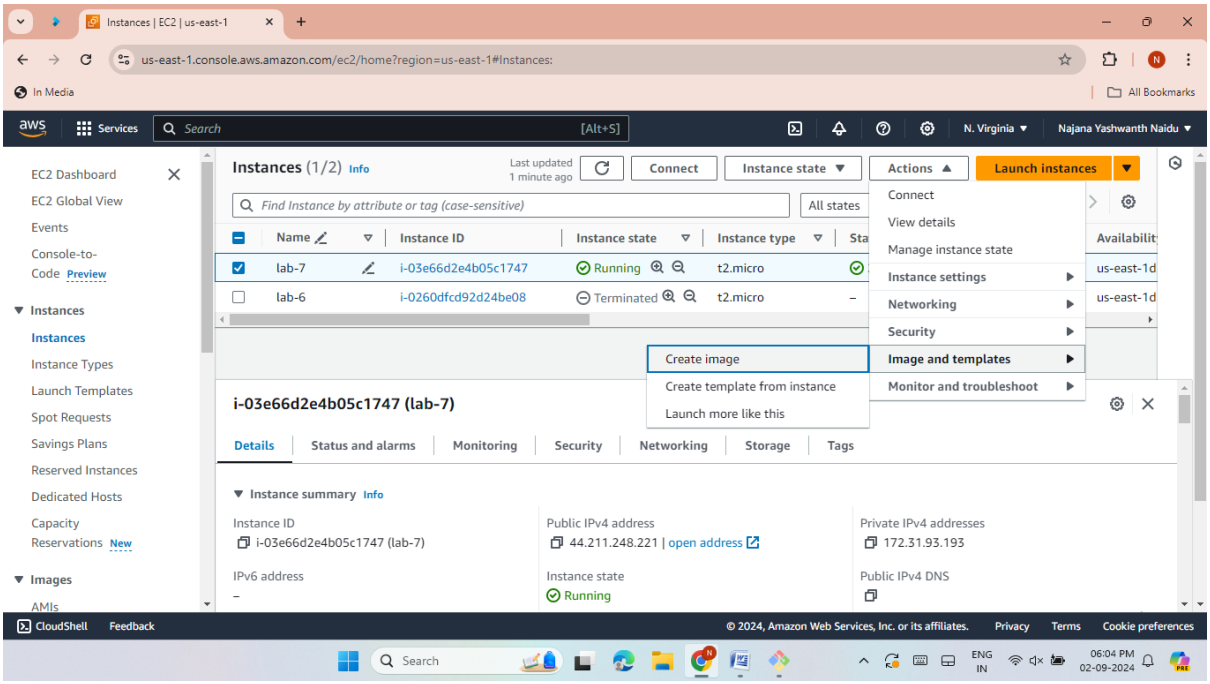


➢ The new volume is attached to the instance successfully.

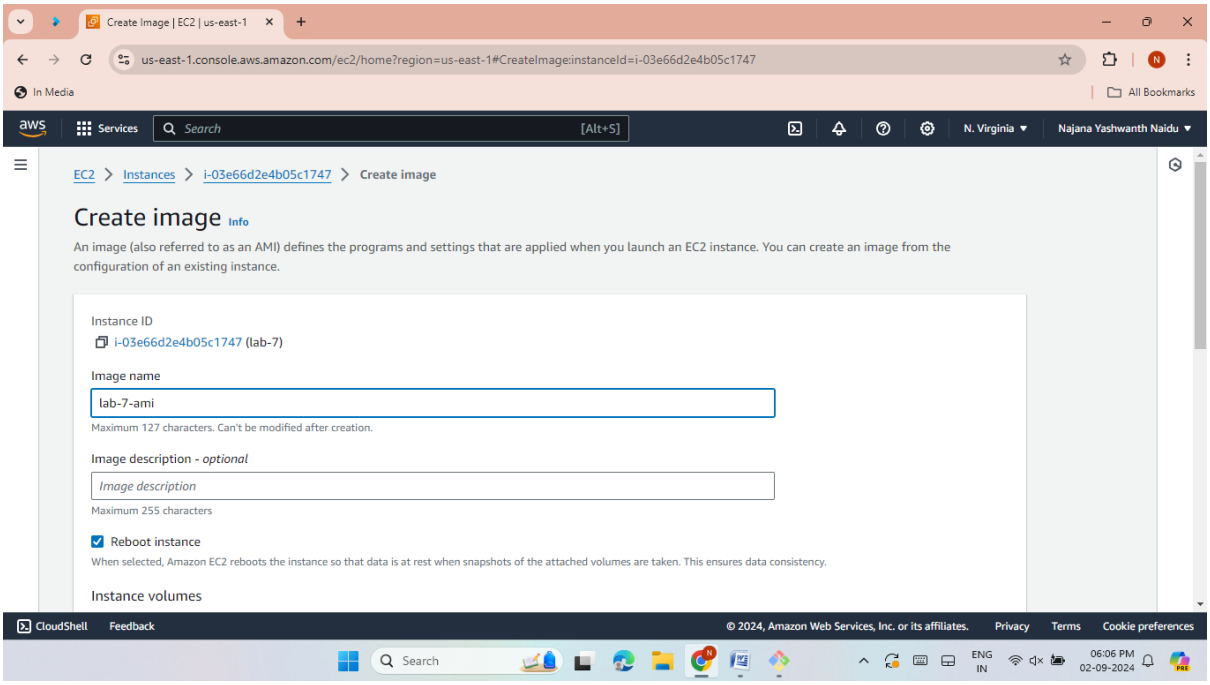


LAB 7 : AMI

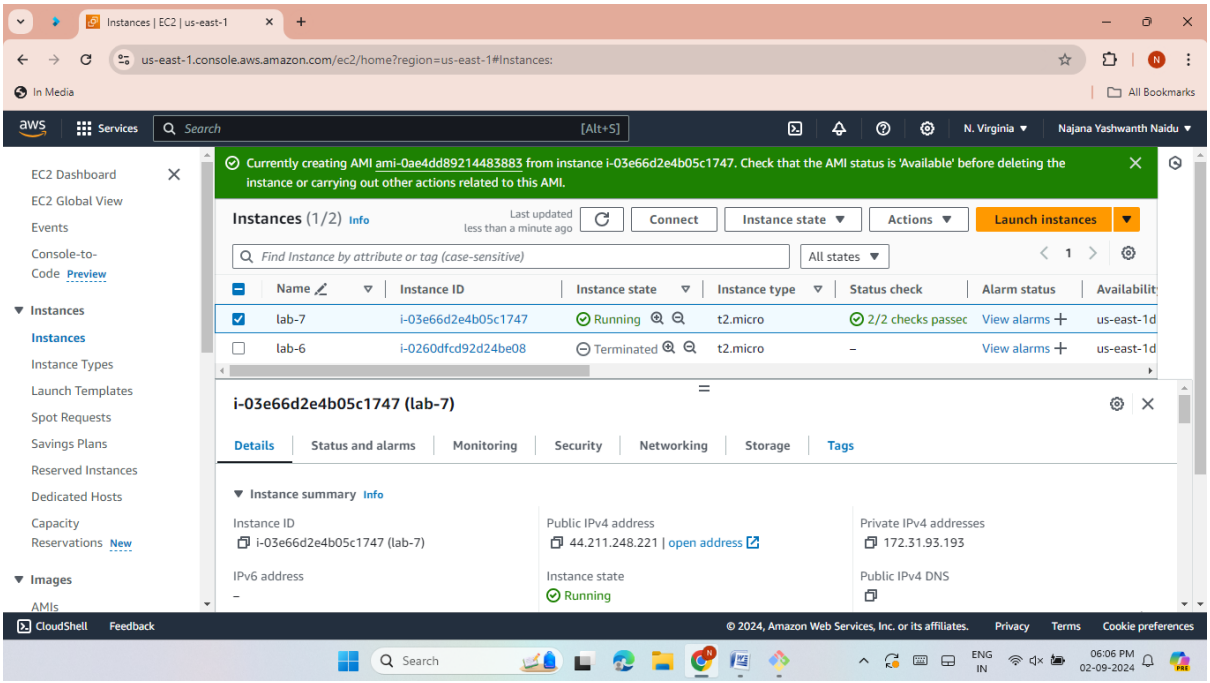
- Create an AMI of your running instance.
- Go to Actions → Image and templates → click on create image.



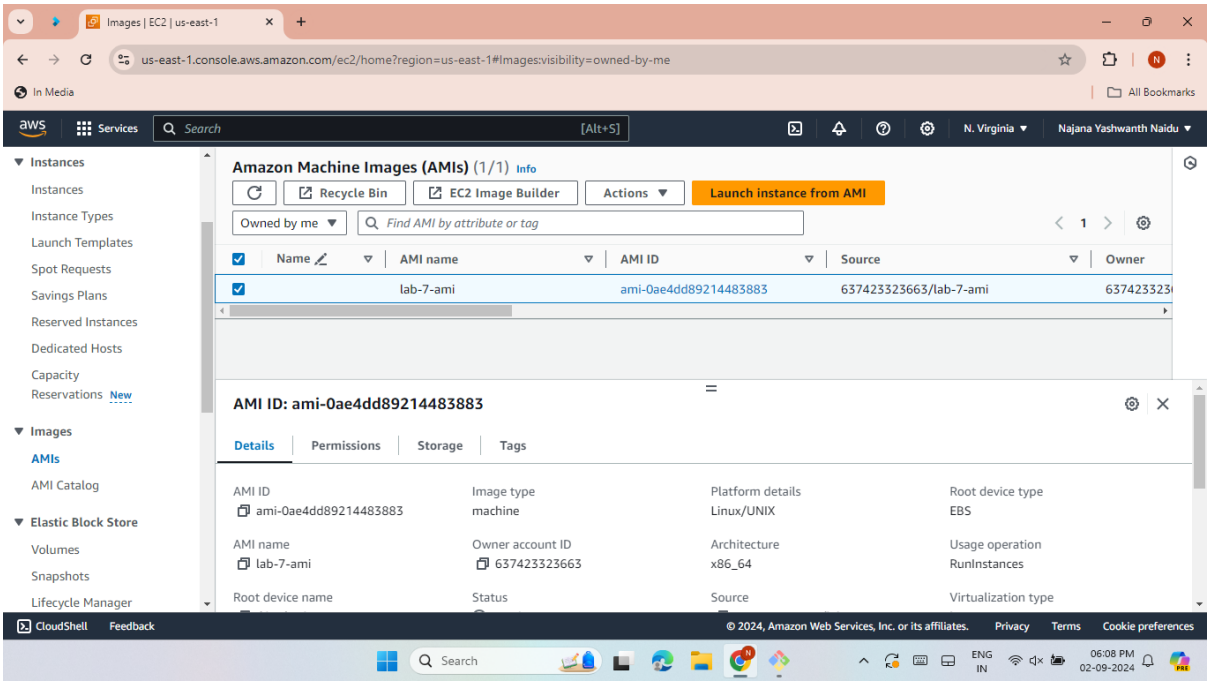
- Give the name of image.



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

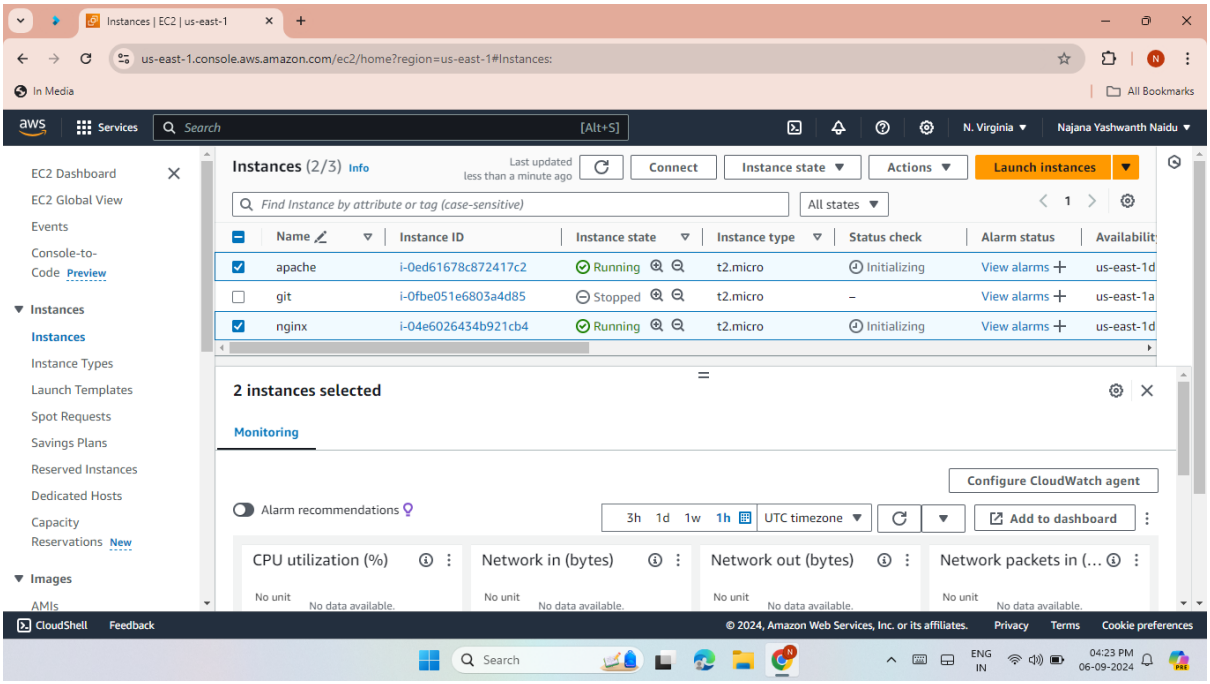


➢ This is the created AMI.

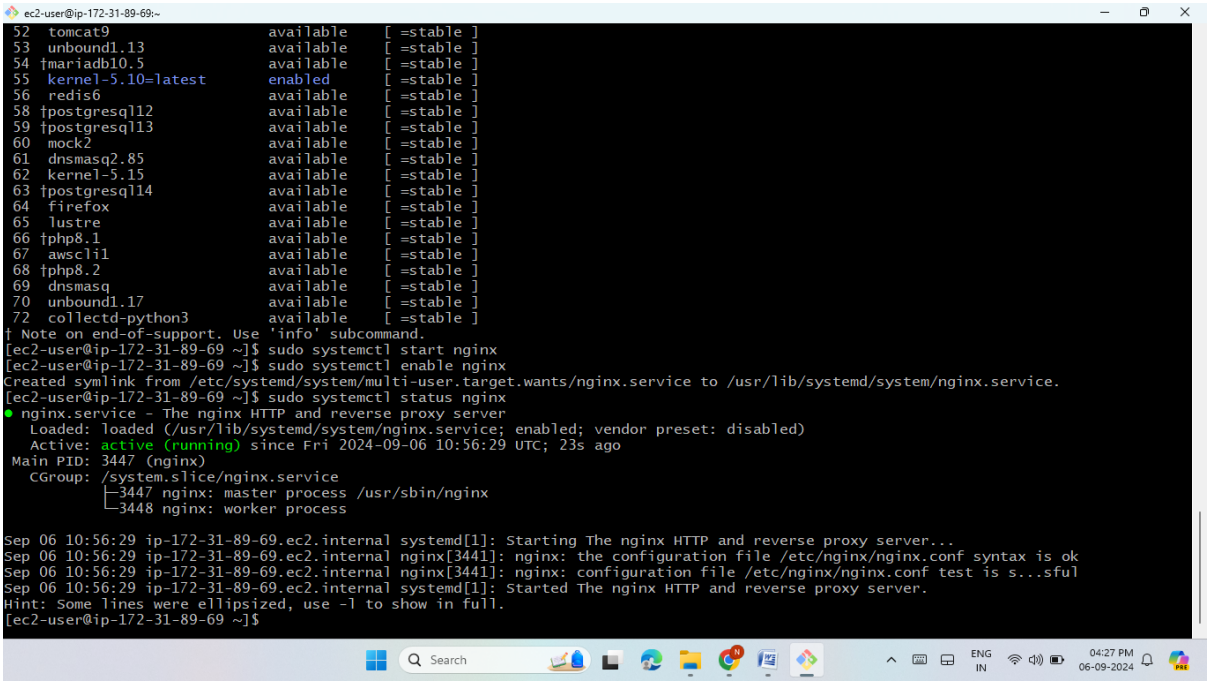


LAB 8 : LAOD BALANCERS

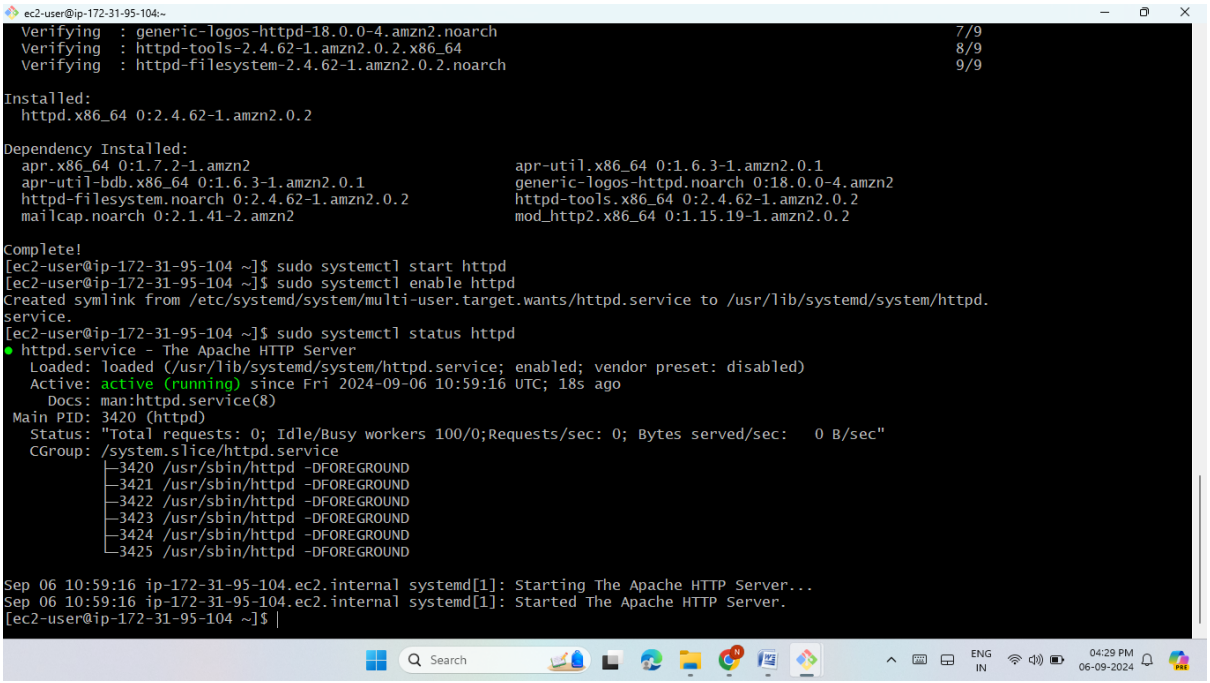
- Create two EC2 Instances.



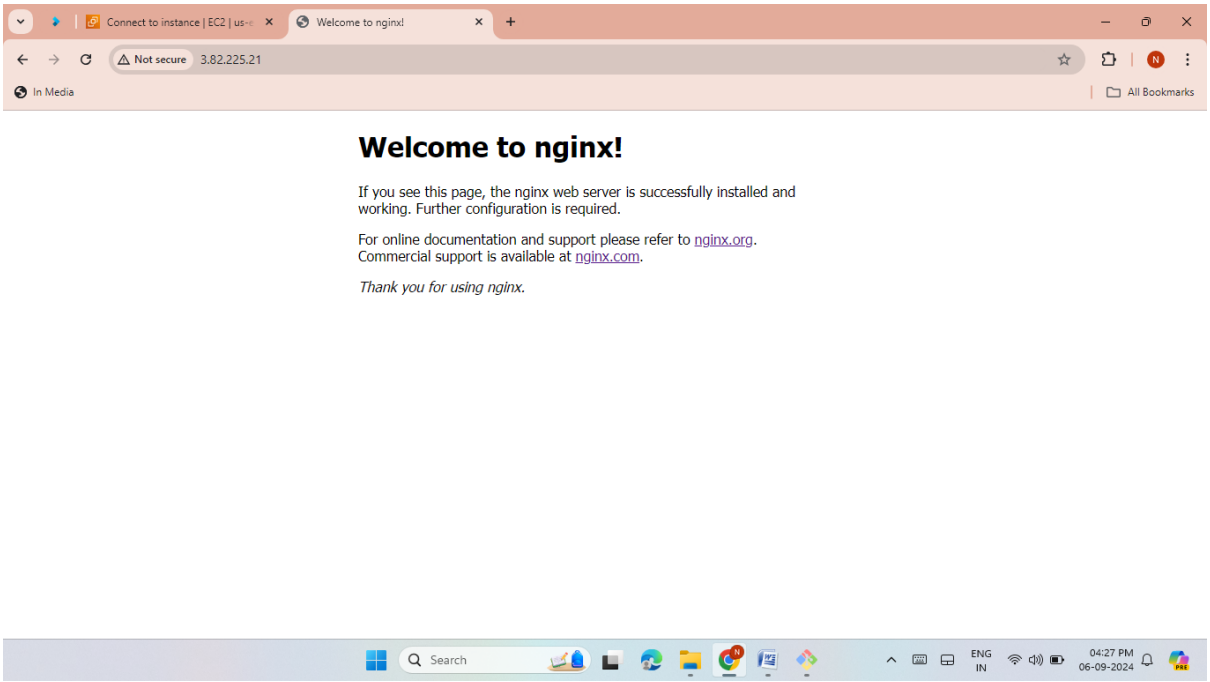
- install nginx server on one machine.
- Using Amazon-Linux-extras install nginx1 command.

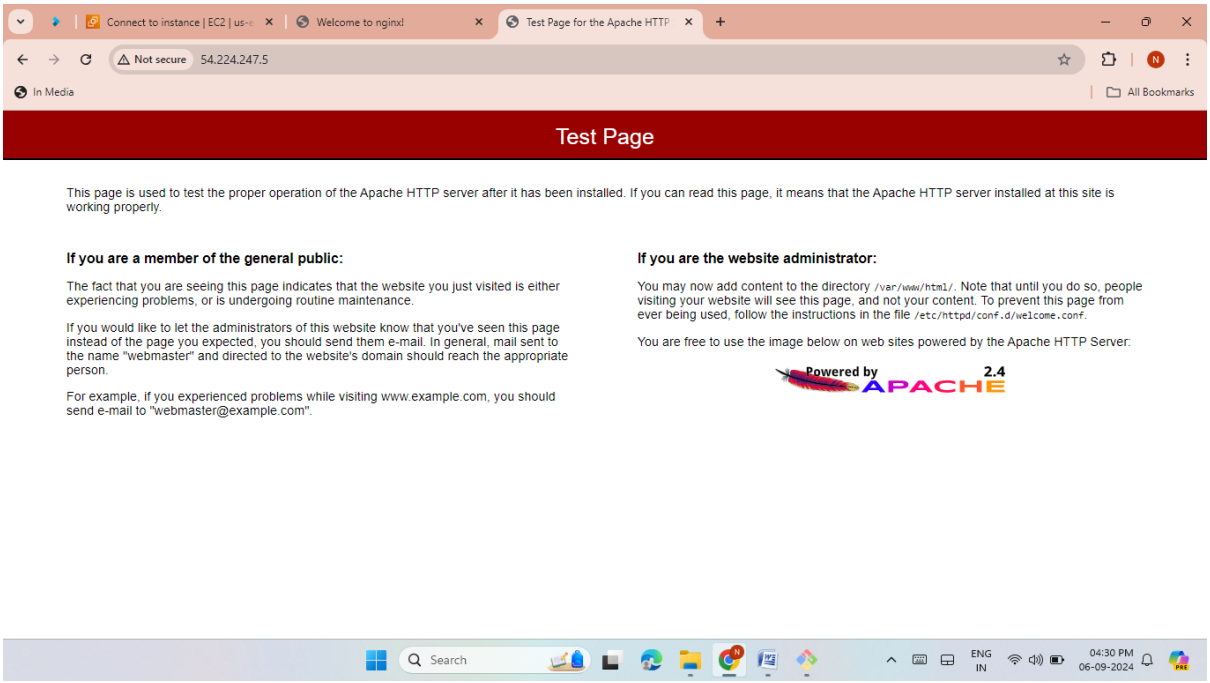


- Install Apache HTTP server on server2.
- Using sudo yum –y install httpd command.

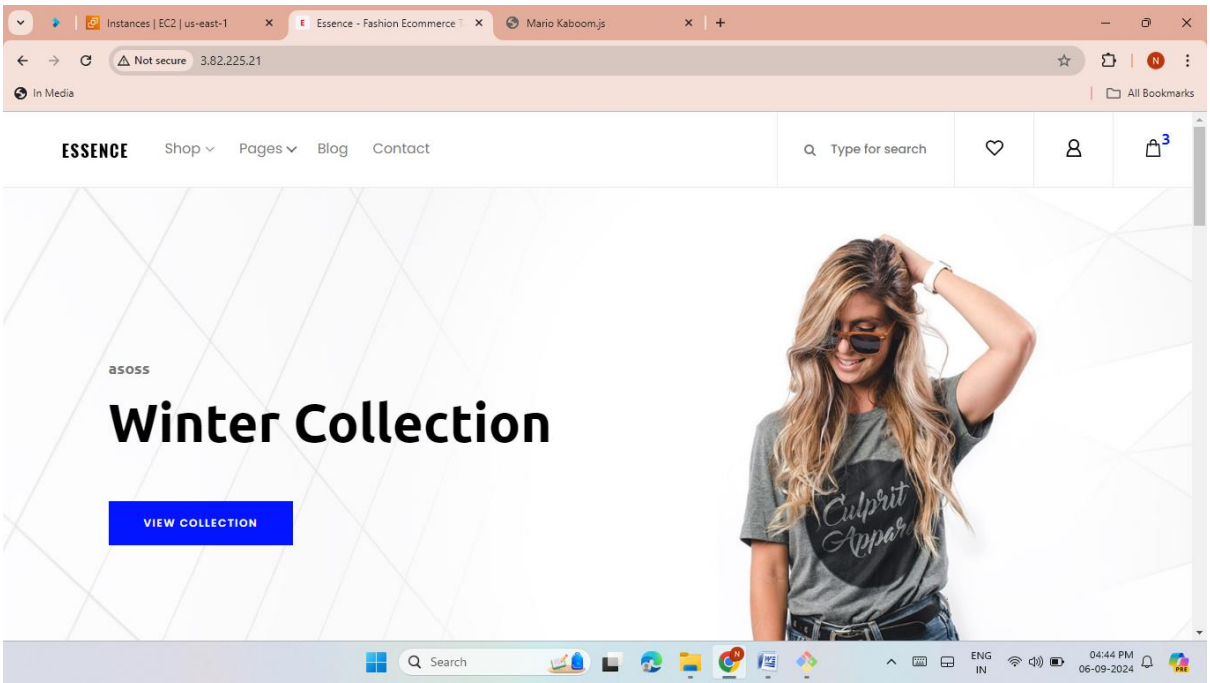


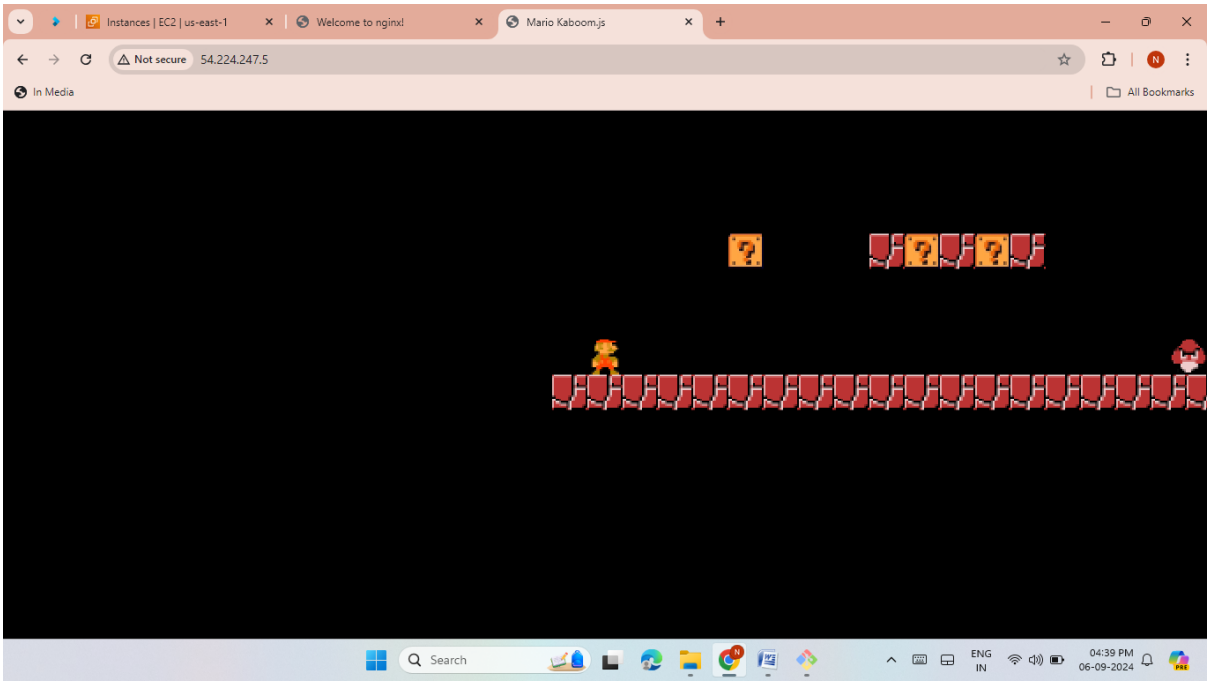
- Access both the servers over browser and check if their web page is visible or not.



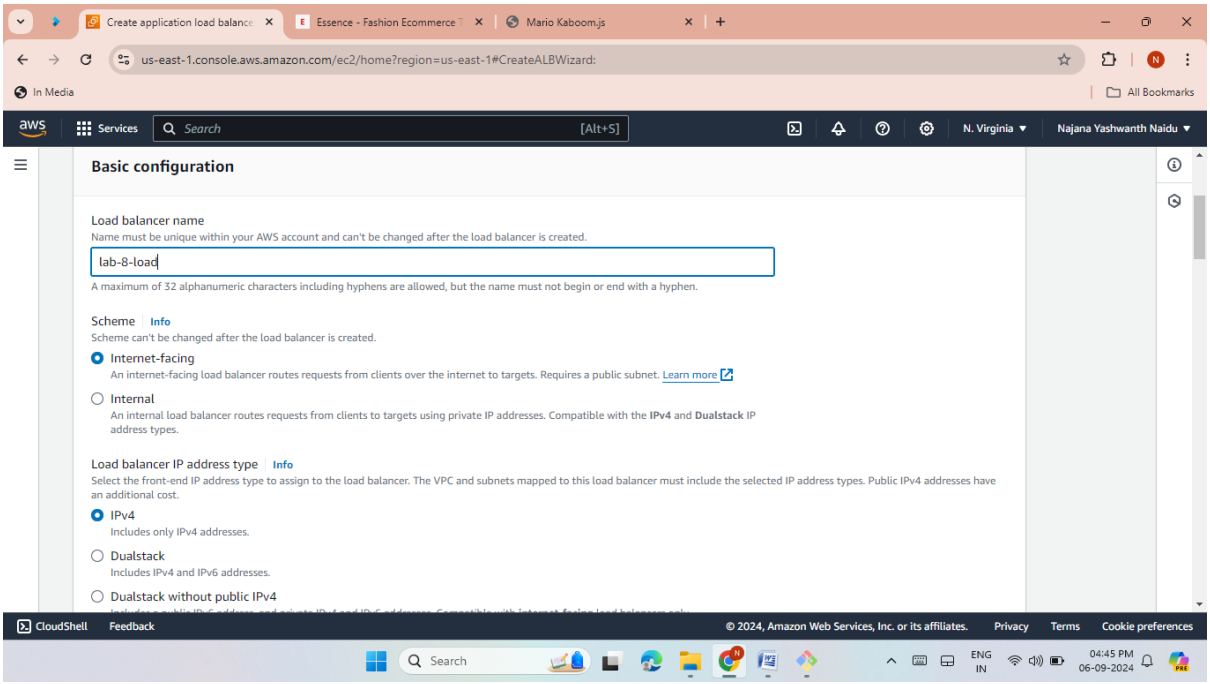


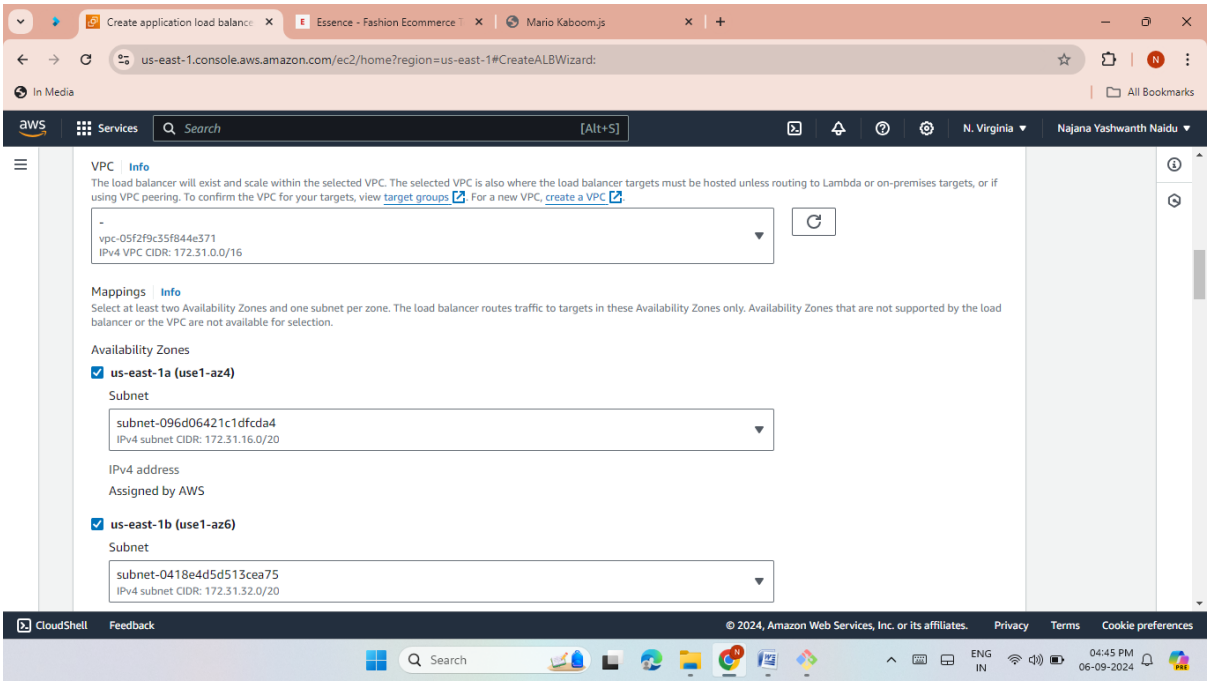
- Install git then clone the different repo on two servers.
- Path of nginx → /usr/share/nginx/html/.
- Path of HTTP → /var/www/html/.



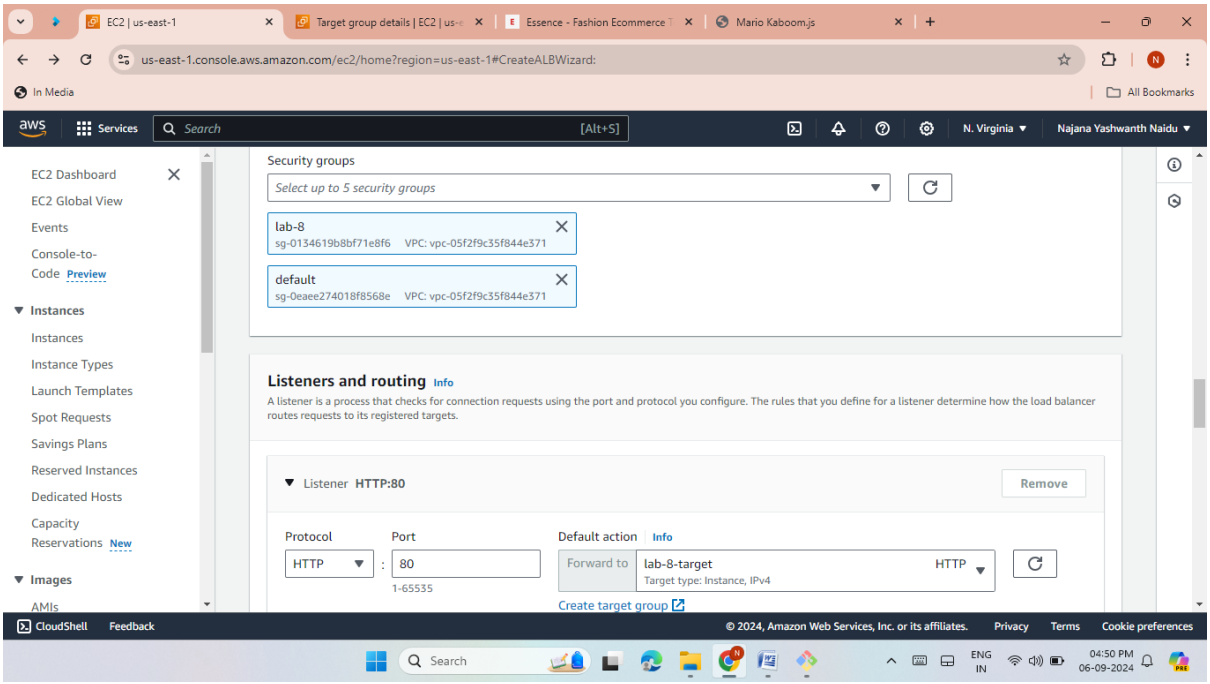


- Create one load balancers and attach both the instances with the load balancer.
- Select VPC and Availability zones.

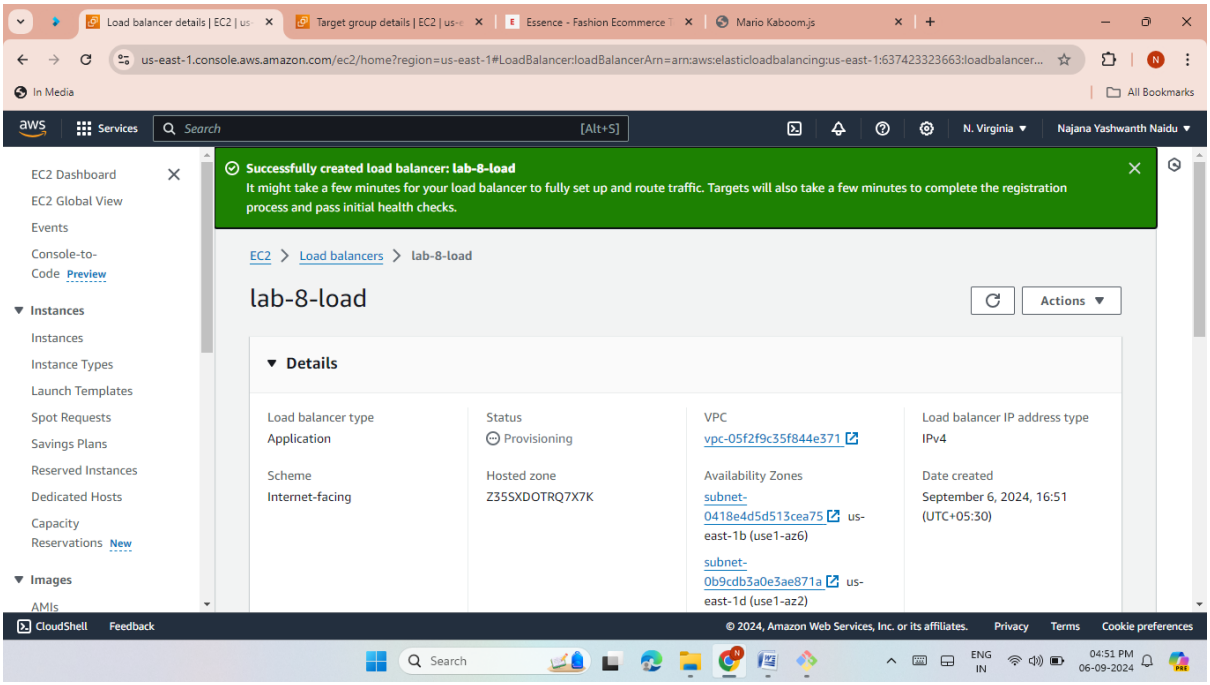




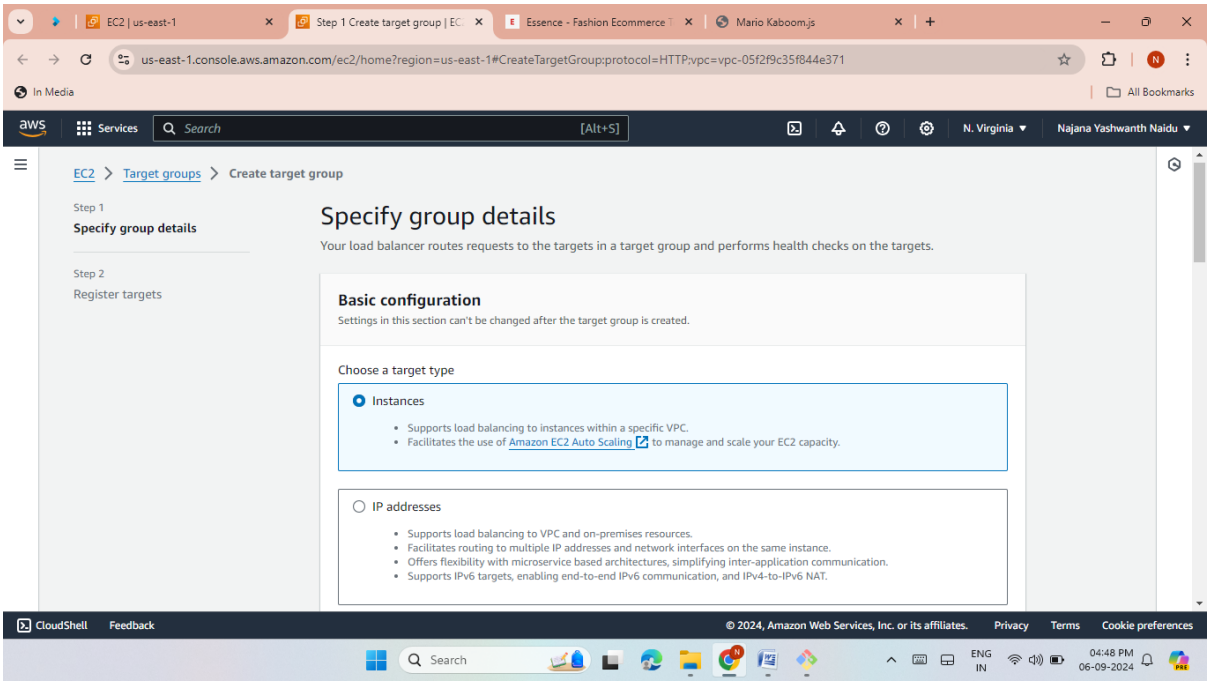
- Allow only port 80 in Load Balancer security groups and also make the Security group of your instances to receive request from LB only on port 80.

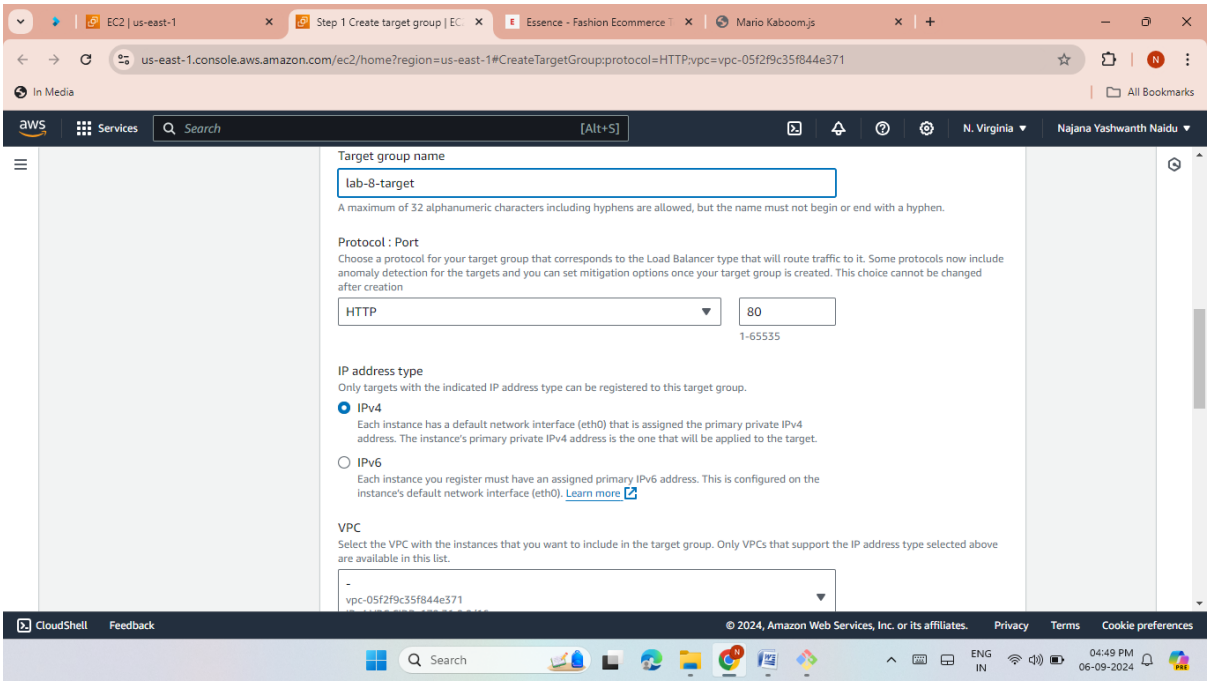


- Here the load balancer was created Successfully.

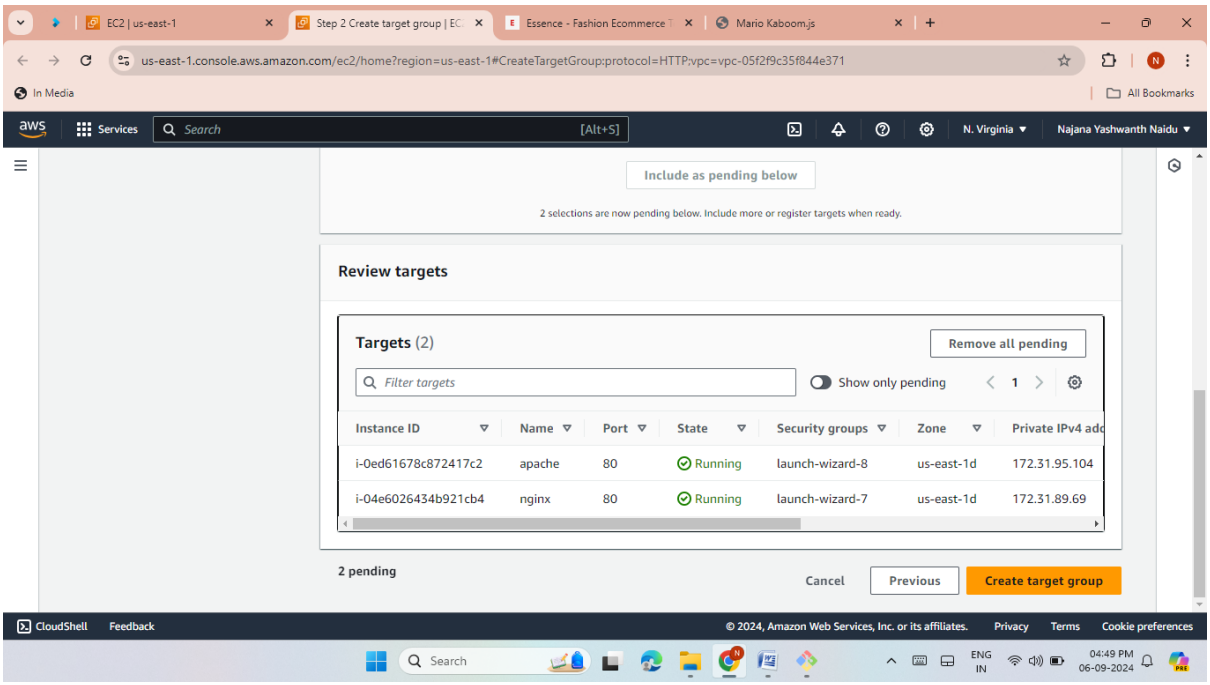


- Create a target group and attach to the load balancer.
- Same VPC and subnet.

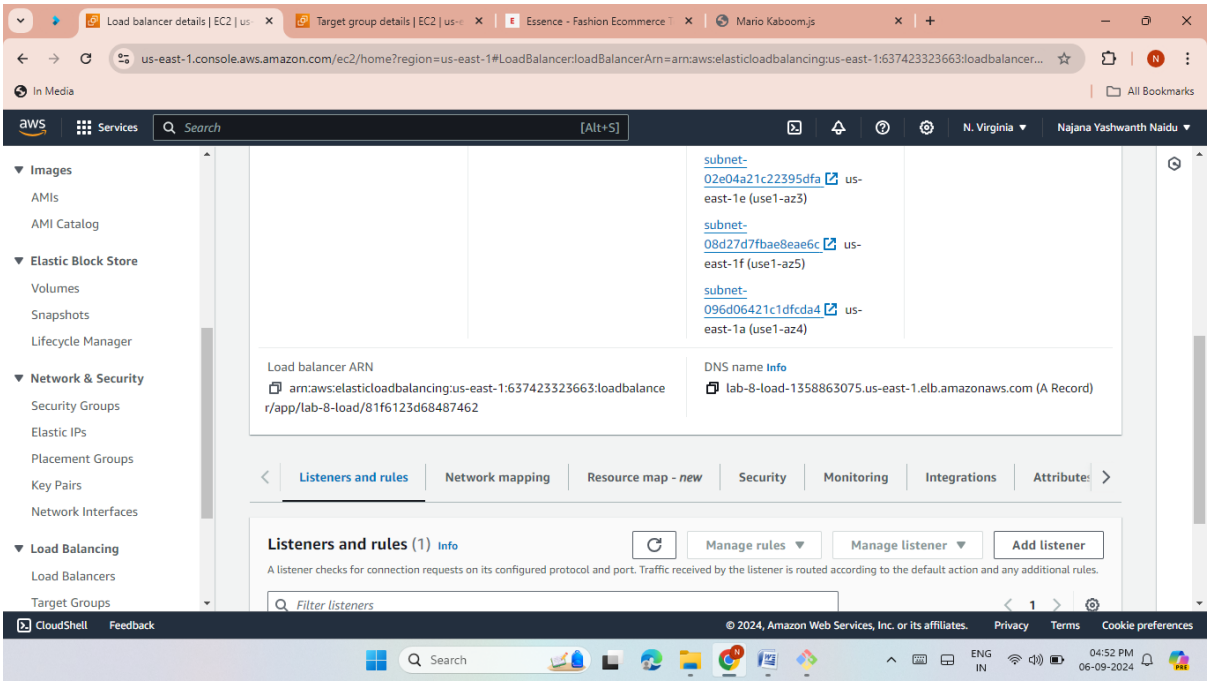


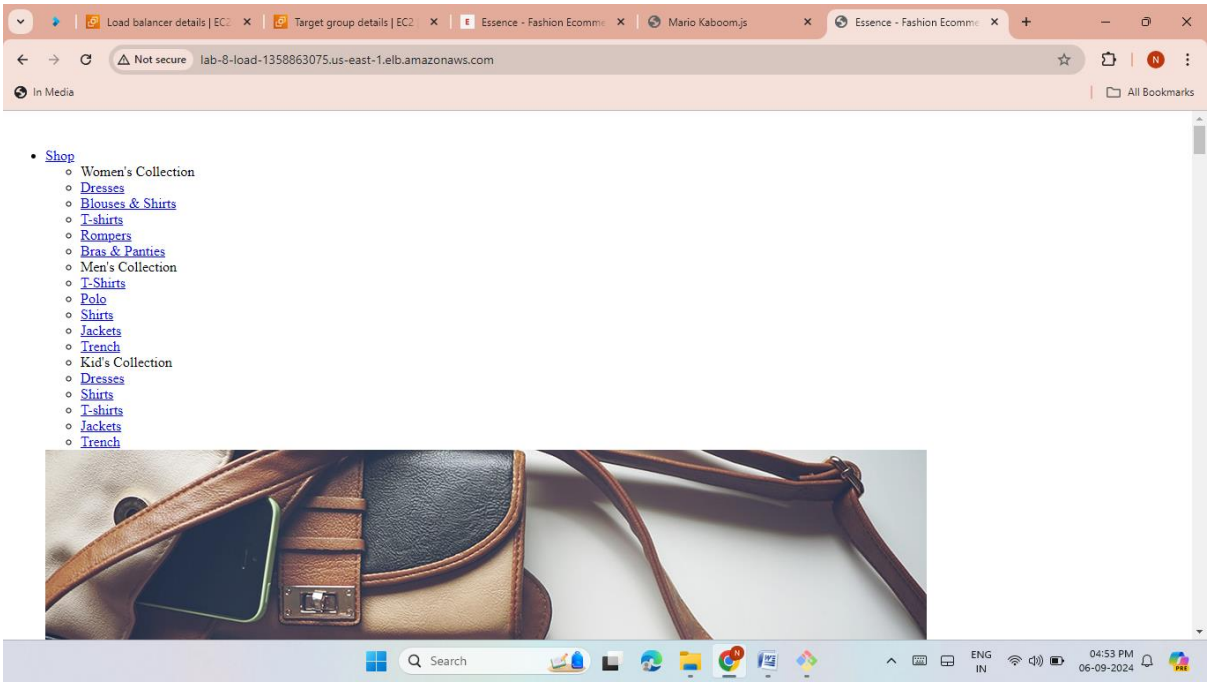


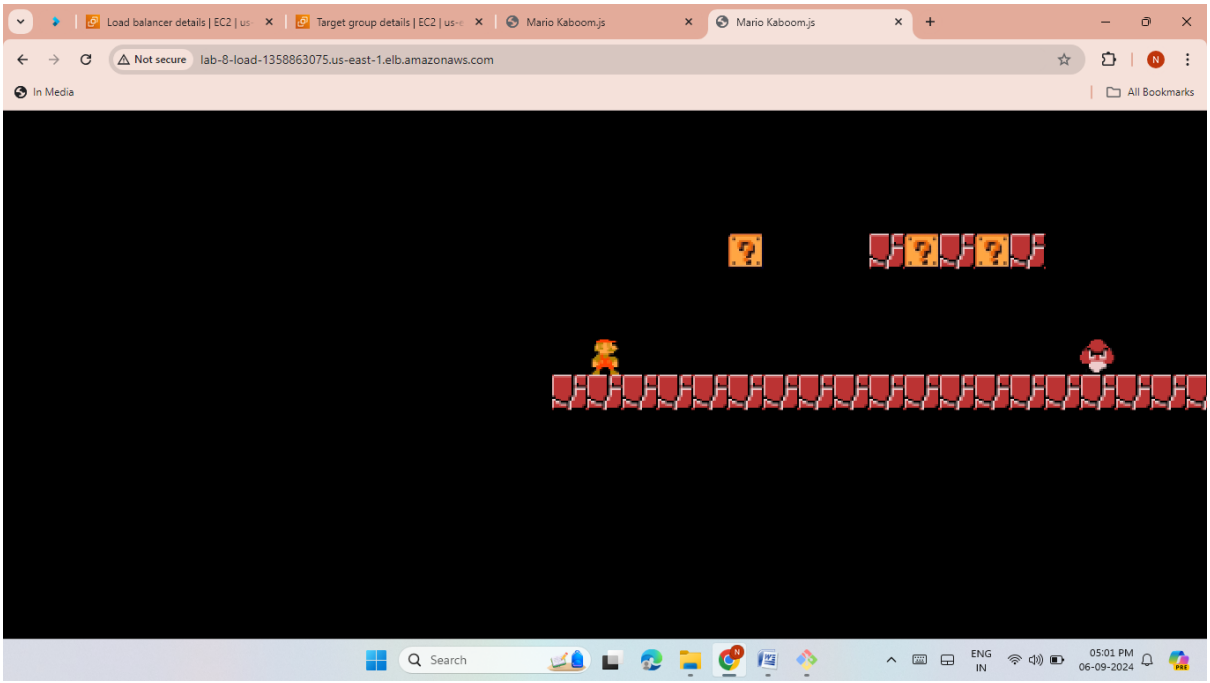
➢ Register the instances in the target group.



- 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).

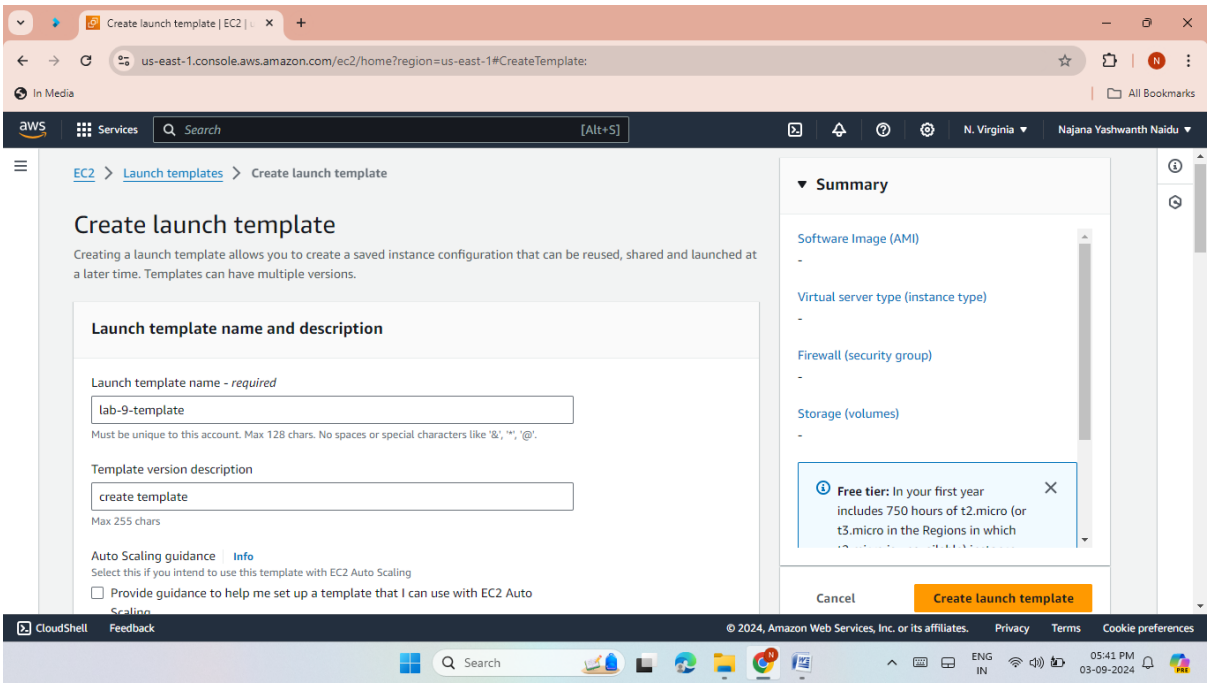


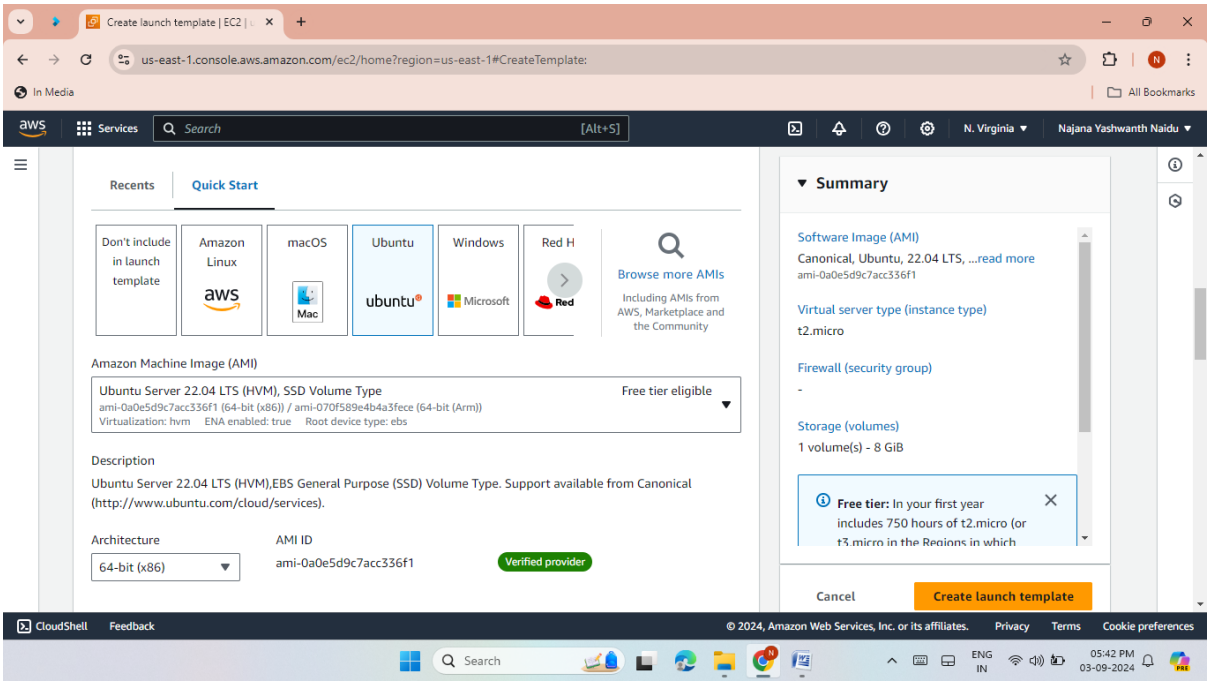


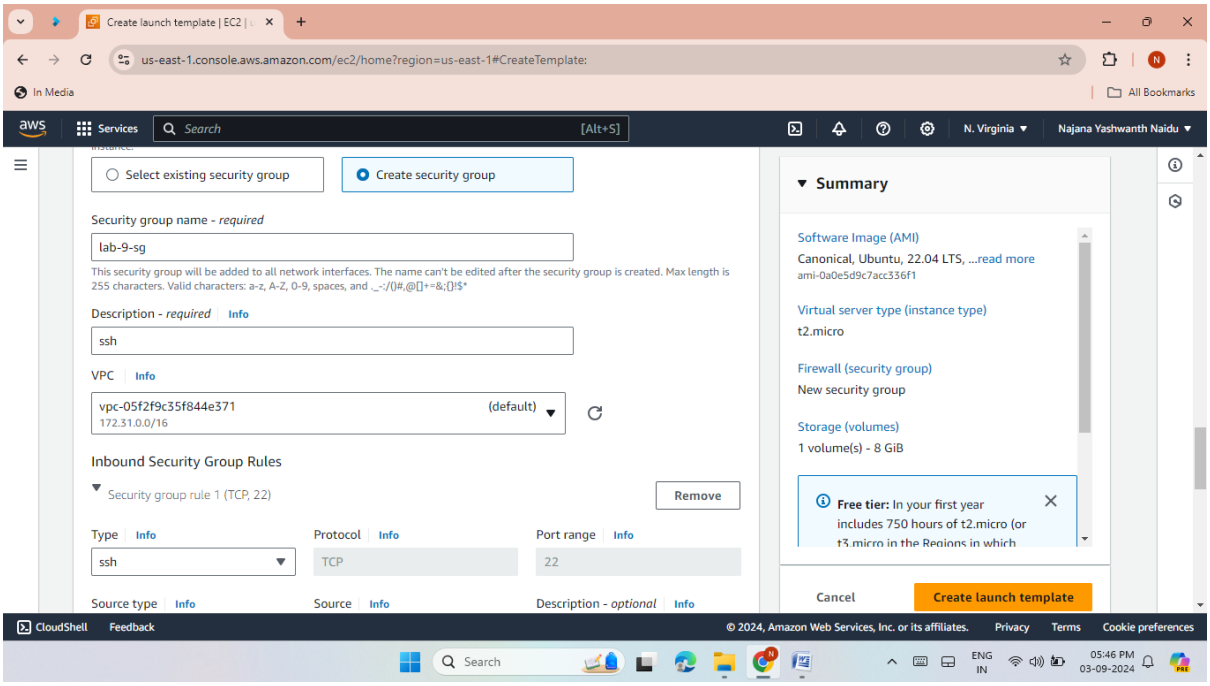


LAB 9 : AUTO SCALING GROUP AND LAUNCH TEMPLATE

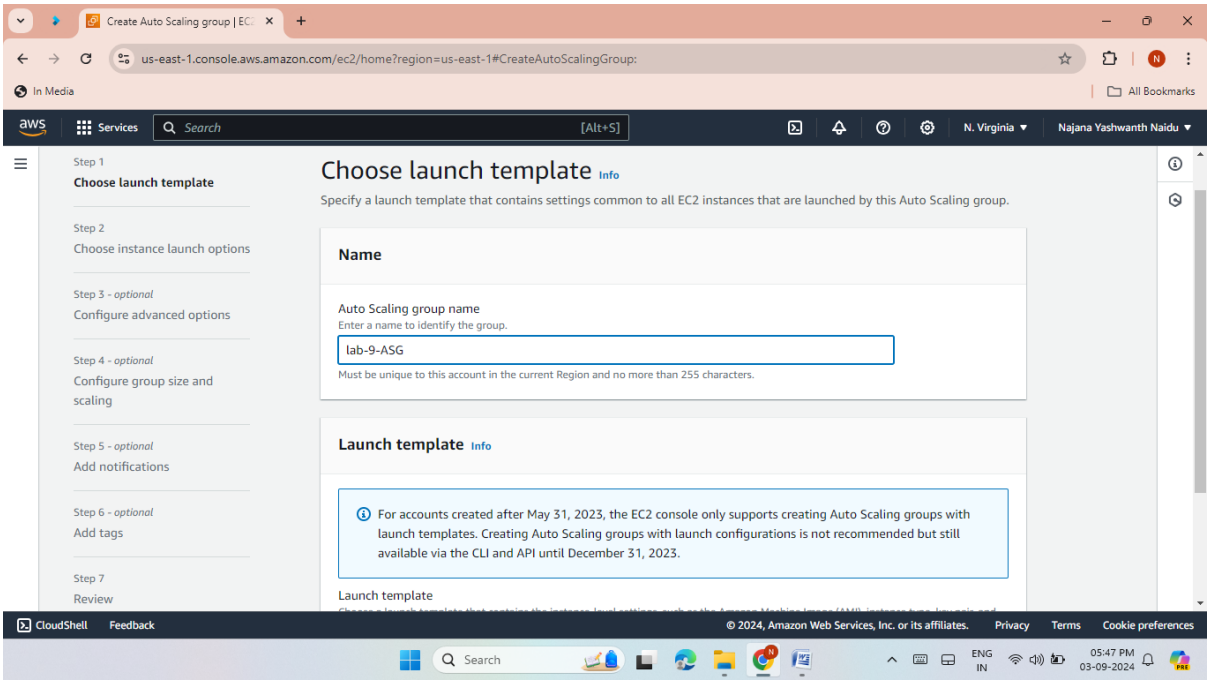
- Create one launch template with ubuntu server.

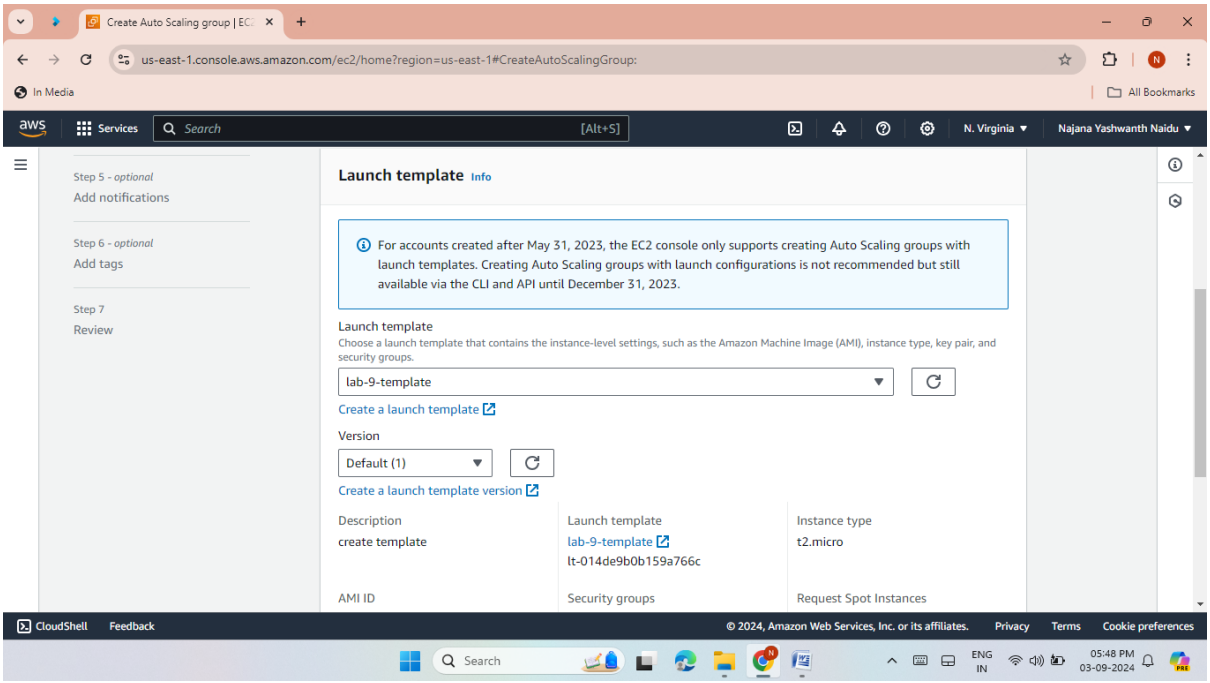



Create launch template | EC2
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTemplate:
Services
Search
[Alt+S]
N. Virginia
Najana Yashwanth Naidu
Recents
Quick Start
Don't include in launch template
Amazon Linux
macOS
Ubuntu
Windows
Red H
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
Free tier eligible
ami-0a0e5d9c7acc336f1 (64-bit (x86)) / ami-070f589e4b4a3fece (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs
Description
Ubuntu Server 22.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Architecture
64-bit (x86)
AMI ID
ami-0a0e5d9c7acc336f1
Verified provider
Summary
Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-0a0e5d9c7acc336f1
Virtual server type (instance type)
t2.micro
Firewall (security group)
-
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or
Cancel
Create launch template
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
05:42 PM
03-09-2024

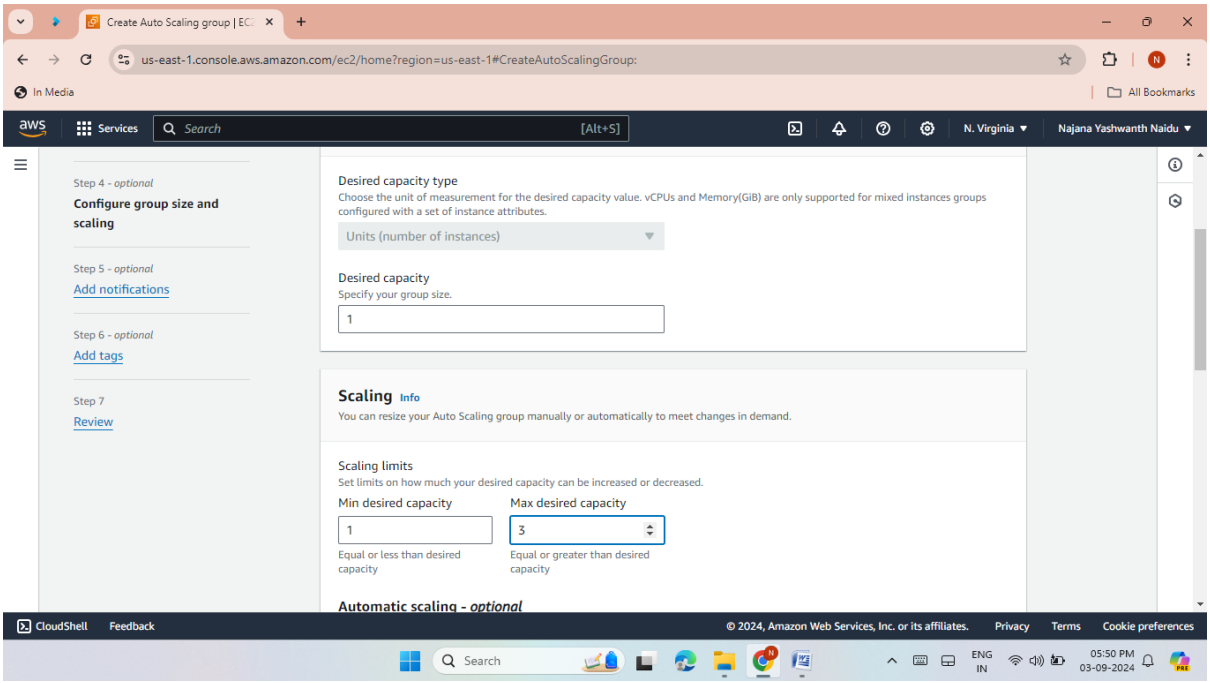

Create launch template | EC2
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTemplate:
Services
Search
[Alt+S]
N. Virginia
Najana Yashwanth Naidu
Select existing security group
Create security group
Security group name - required
lab-9-sg
This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*
Description - required Info
ssh
VPC Info
vpc-05f2f9c35f844e371 (default)
172.31.0.0/16
Inbound Security Group Rules
Security group rule 1 (TCP, 22)
Remove
Type Info
ssh
Protocol Info
TCP
Port range Info
22
Source type Info
Source Info
Description - optional Info
Summary
Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-0a0e5d9c7acc336f1
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or
Cancel
Create launch template
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
05:46 PM
03-09-2024

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

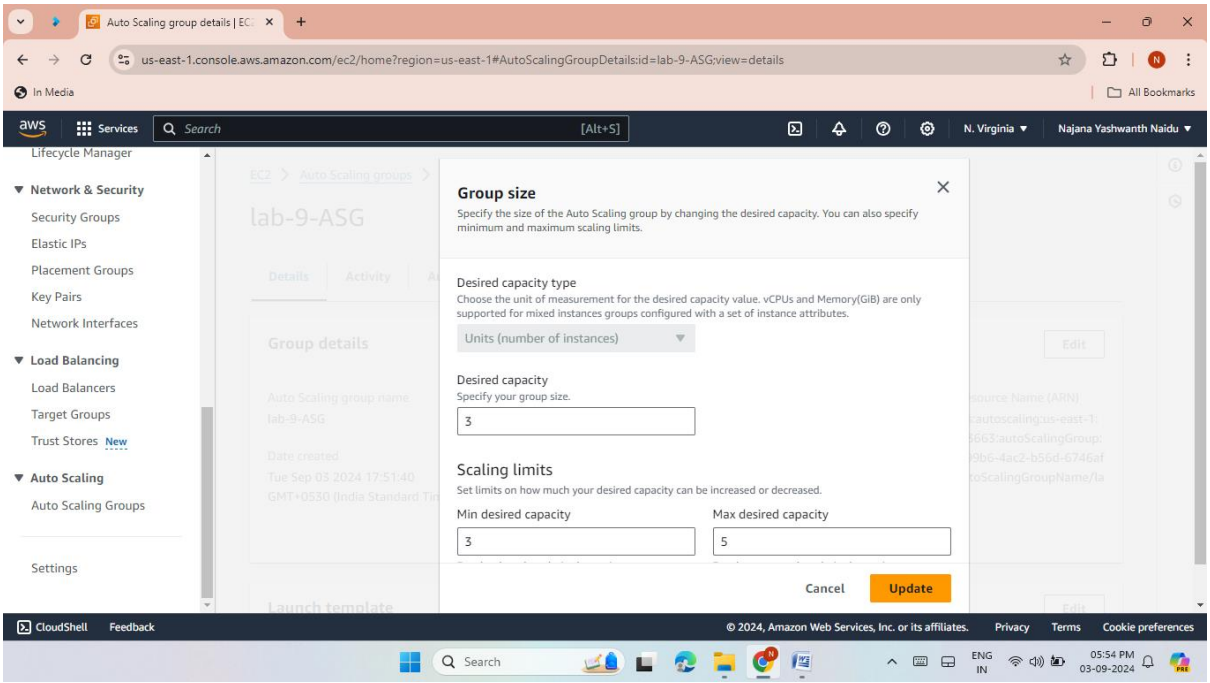




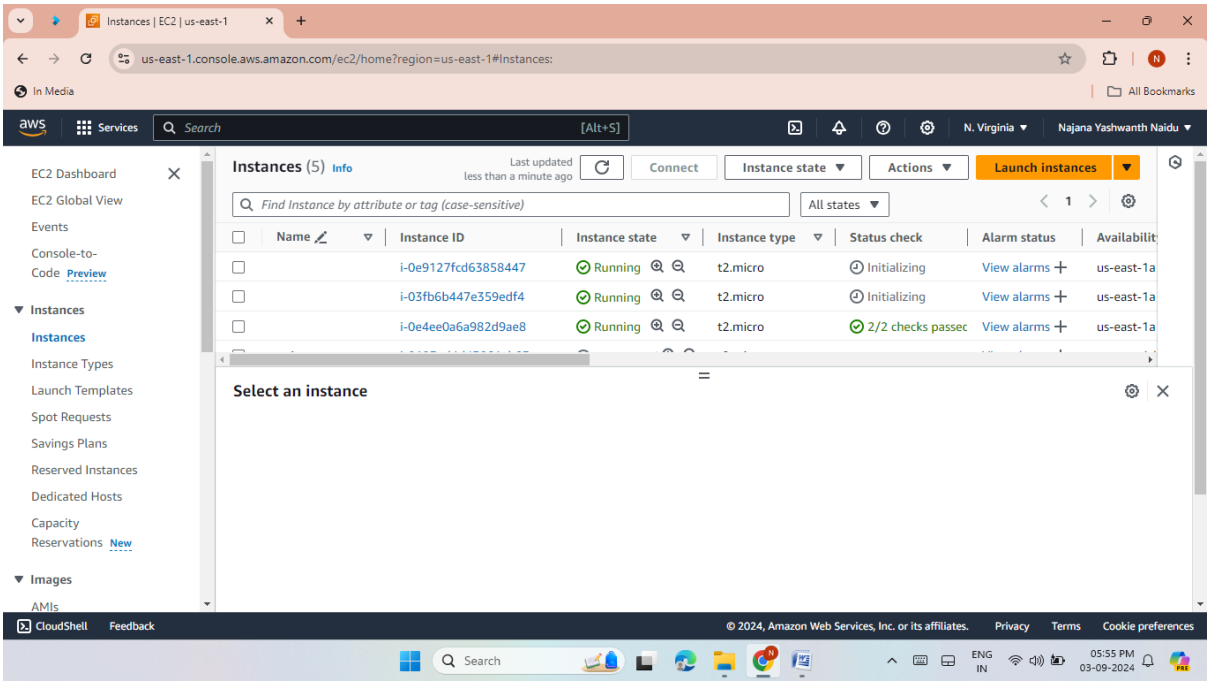
- Keep the size of instances as Min = 1, Max = 3.



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

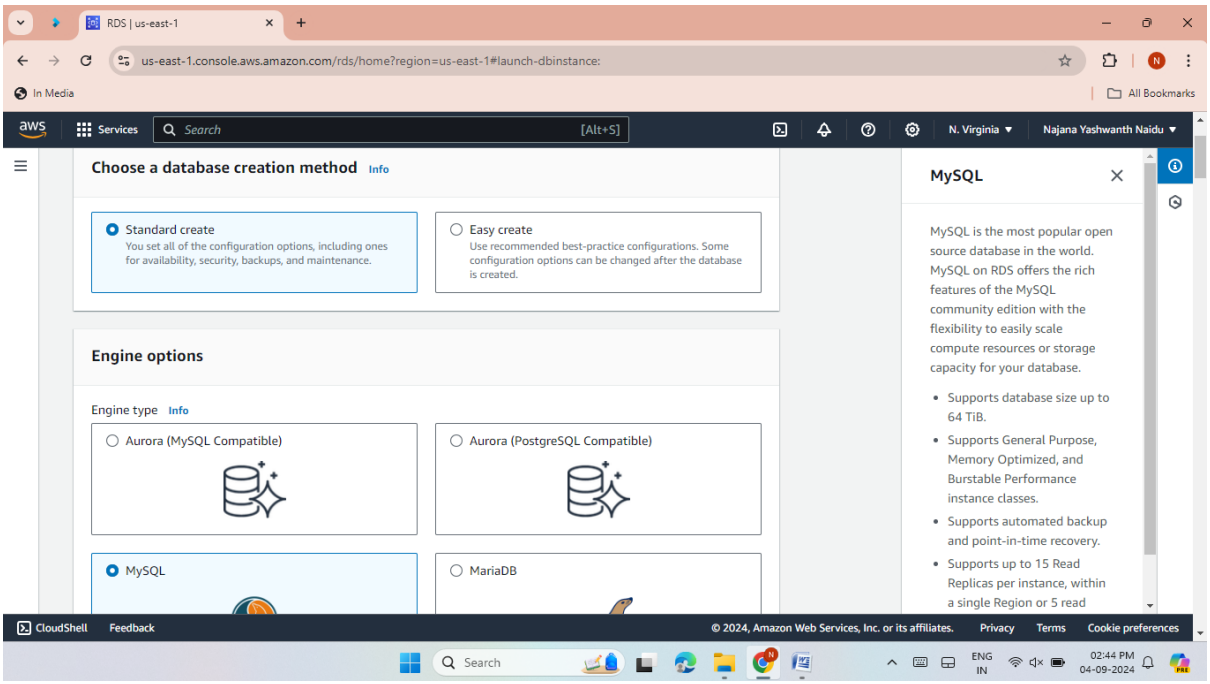


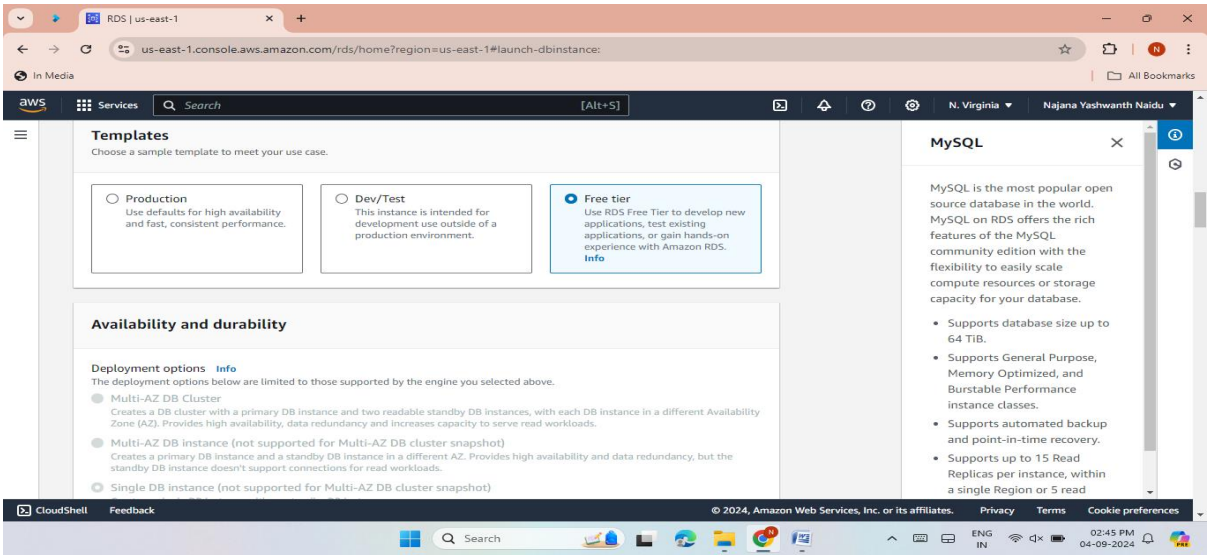
- Here the Min 3 Instances was launched. If we delete any instance it automatically created another one (max 5).

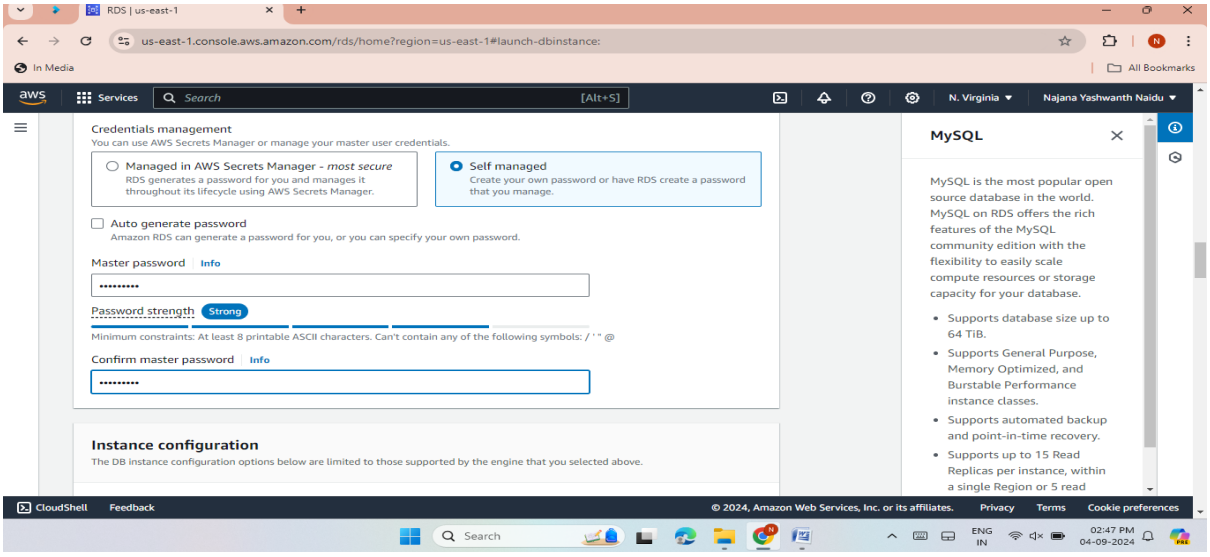


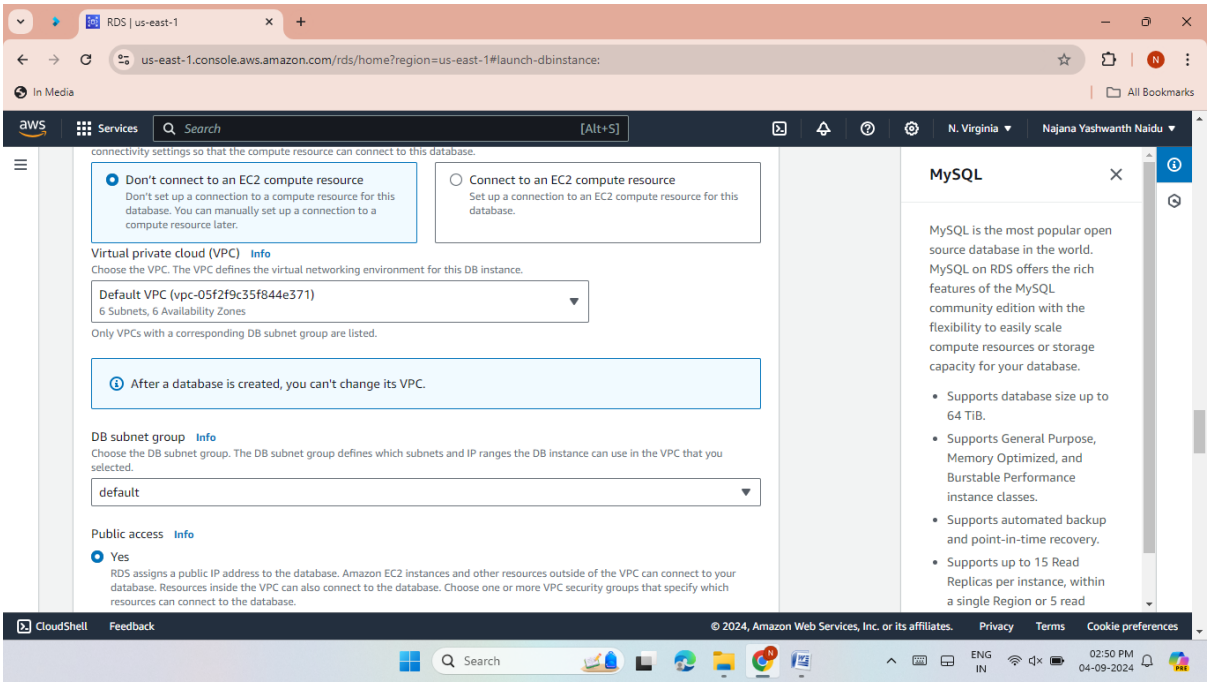
LAB 10 : RDS (RELATIONAL DATABASE)

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

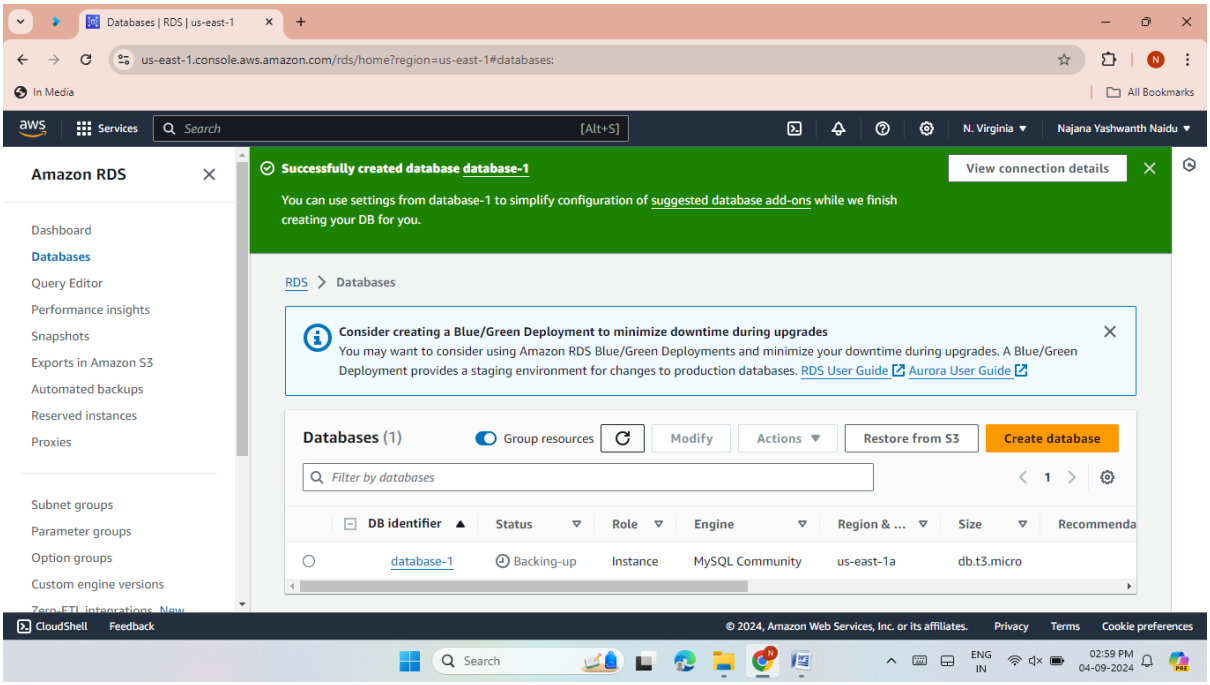



RDS | us-east-1
us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:
Templates
Choose a sample template to meet your use case.
Production
Use defaults for high availability and fast, consistent performance.
Dev/Test
This instance is intended for development use outside of a production environment.
Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
Info
Availability and durability
Deployment options Info
The deployment options below are limited to those supported by the engine you selected above.
Multi-AZ DB Cluster
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.
Multi-AZ DB instance (not supported for Multi-AZ DB cluster snapshot)
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.
Single DB instance (not supported for Multi-AZ DB cluster snapshot)
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
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

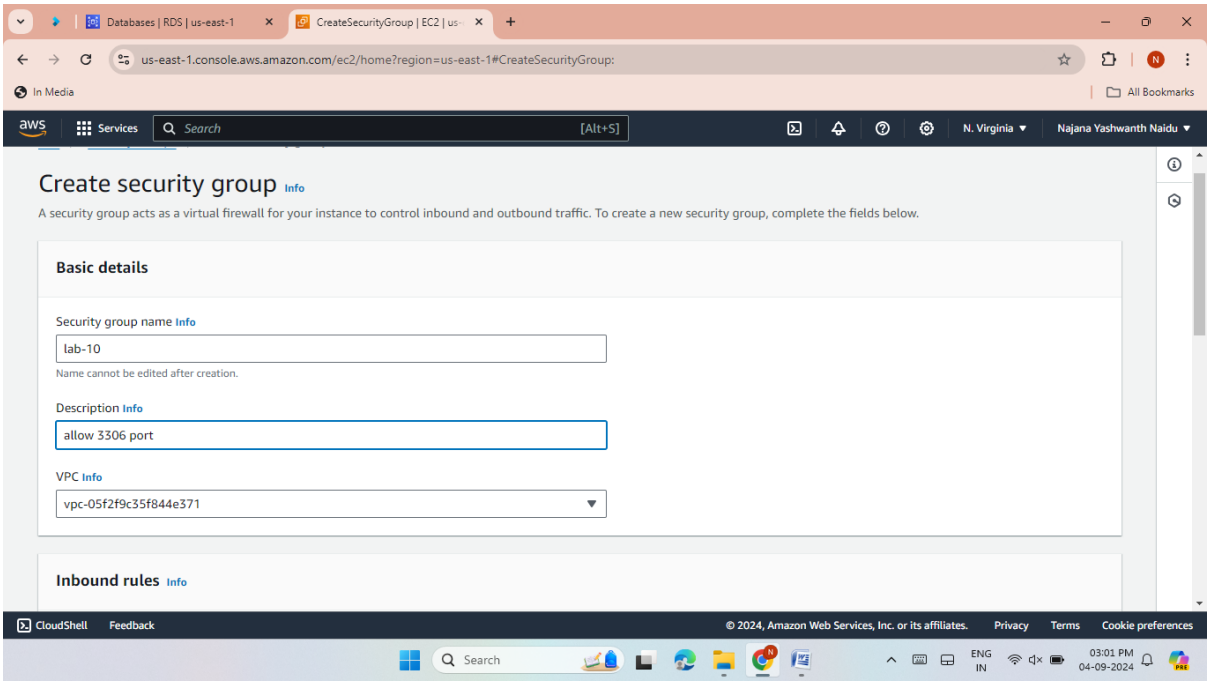

RDS | us-east-1
us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:
Credentials management
You can use AWS Secrets Manager or manage your master user credentials.
Managed in AWS Secrets Manager - most secure
RDS generates a password for you and manages it throughout its lifecycle 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
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

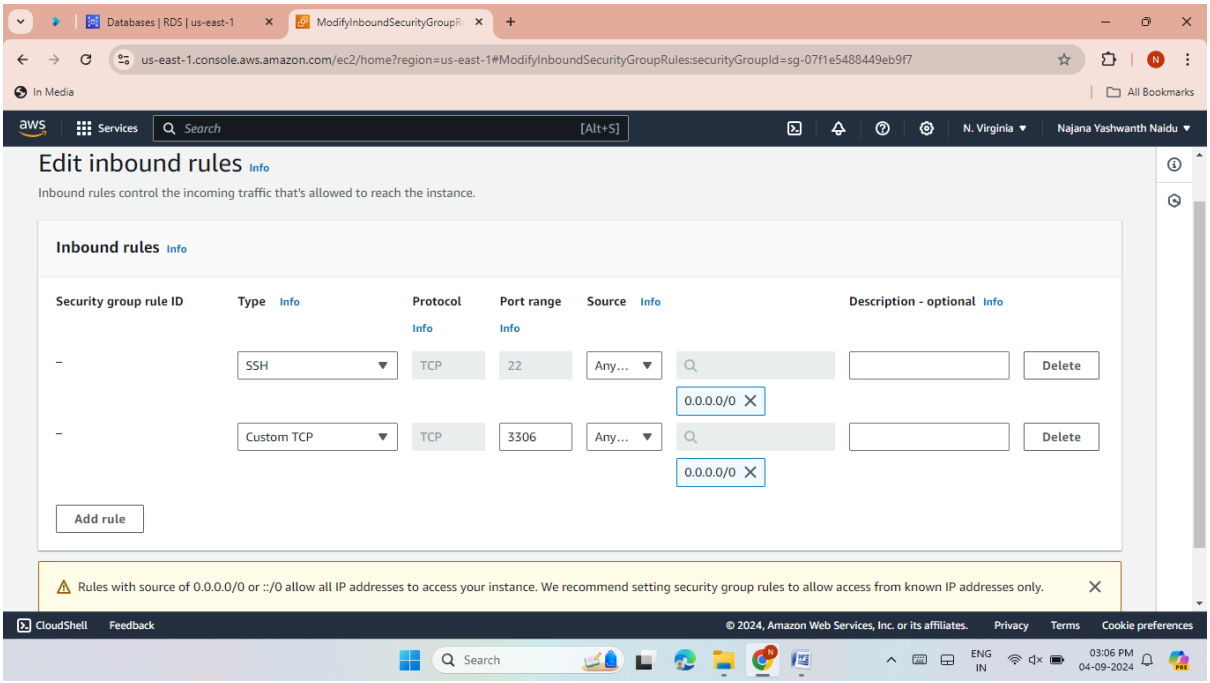

RDS | us-east-1
us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:
connectivity settings so that the compute resource can connect to this database.
Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.
Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.
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
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

- Here RDS (database) was created successfully.

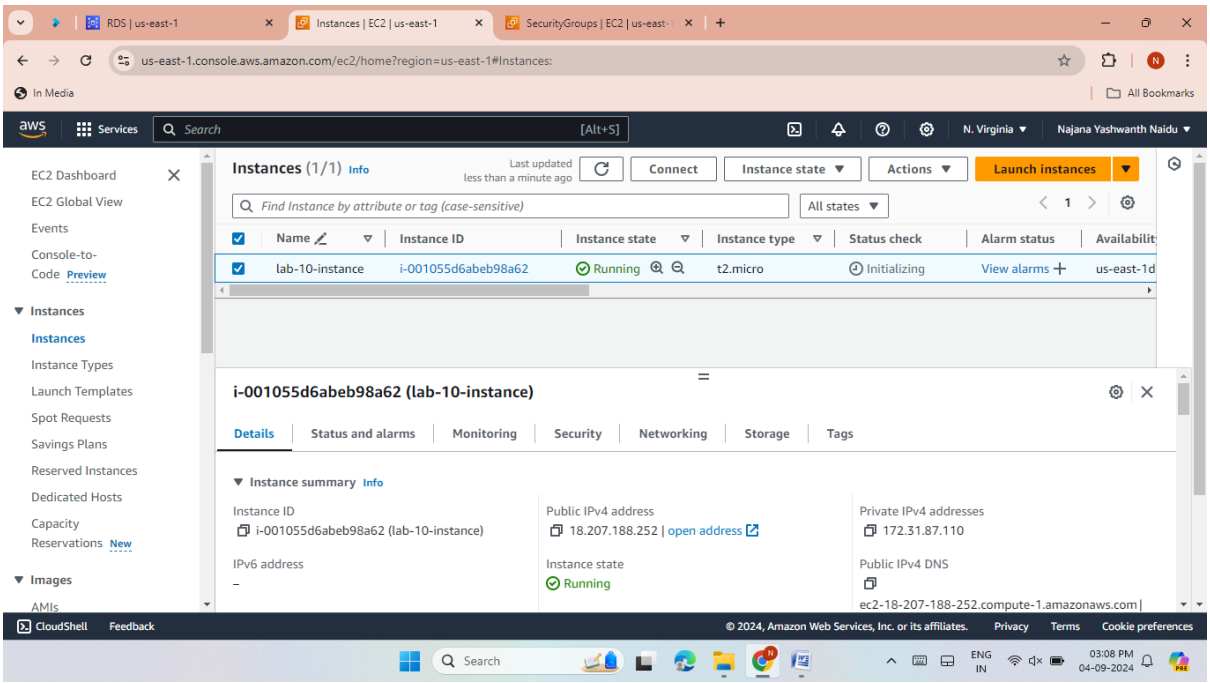


- Open MYSQL port in the connected Security Group.
- Give MYSQL port (3306).





➢ Launched instance.



- 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.