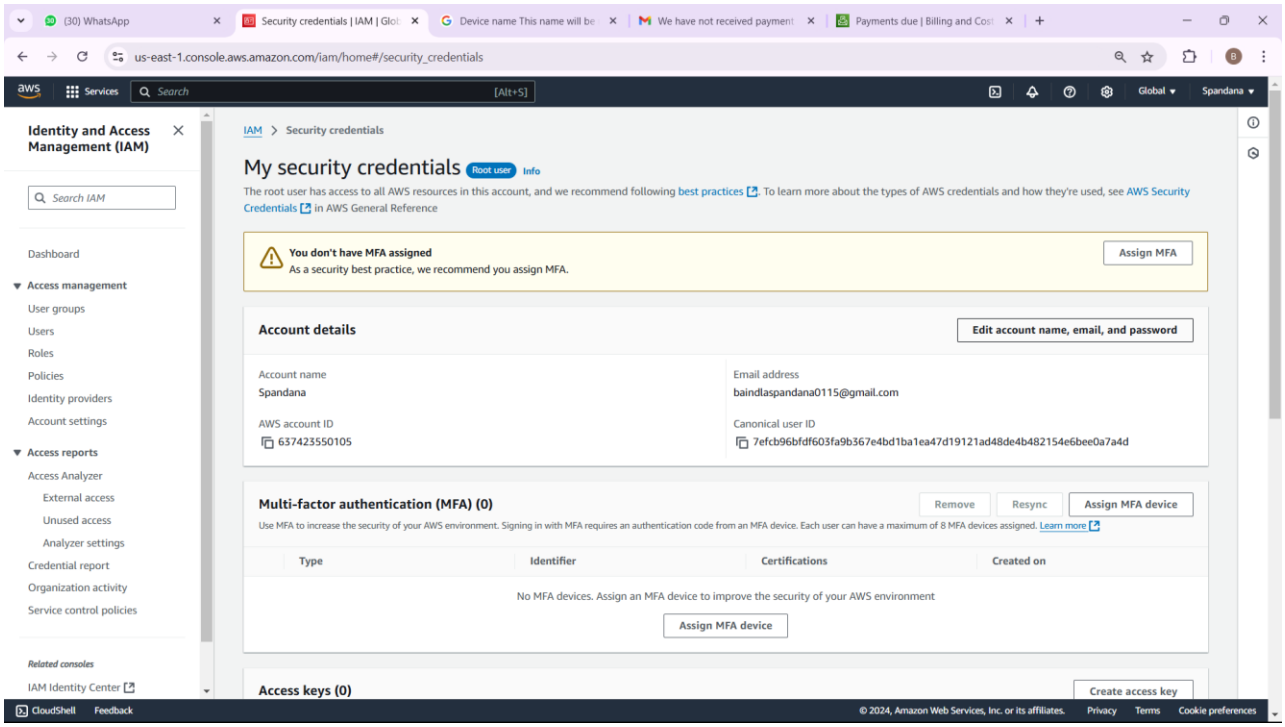
AWS MINI PROJECT 1

LAB1: IAM HANDS-ON

Log in to aws management console and sign in to the aws management console as the root user.

On the right hand side corner we have our account details in that we have security and credentials click on that the below tab will be open.



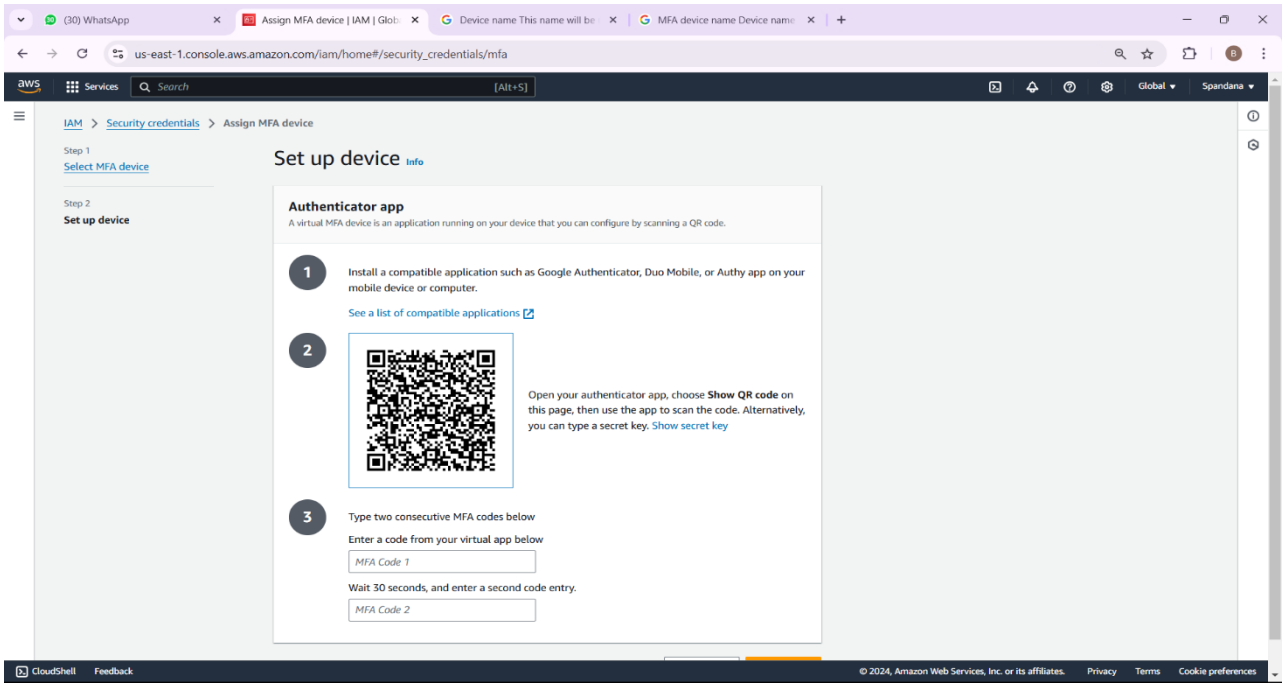
After that we have to click on multi-factor authentication (MFA) Click on Assign MFA.

There are three types of Authentication methods for AWS console management account

1. Authentication app
2. Security Key
3. Hardware TOTP token

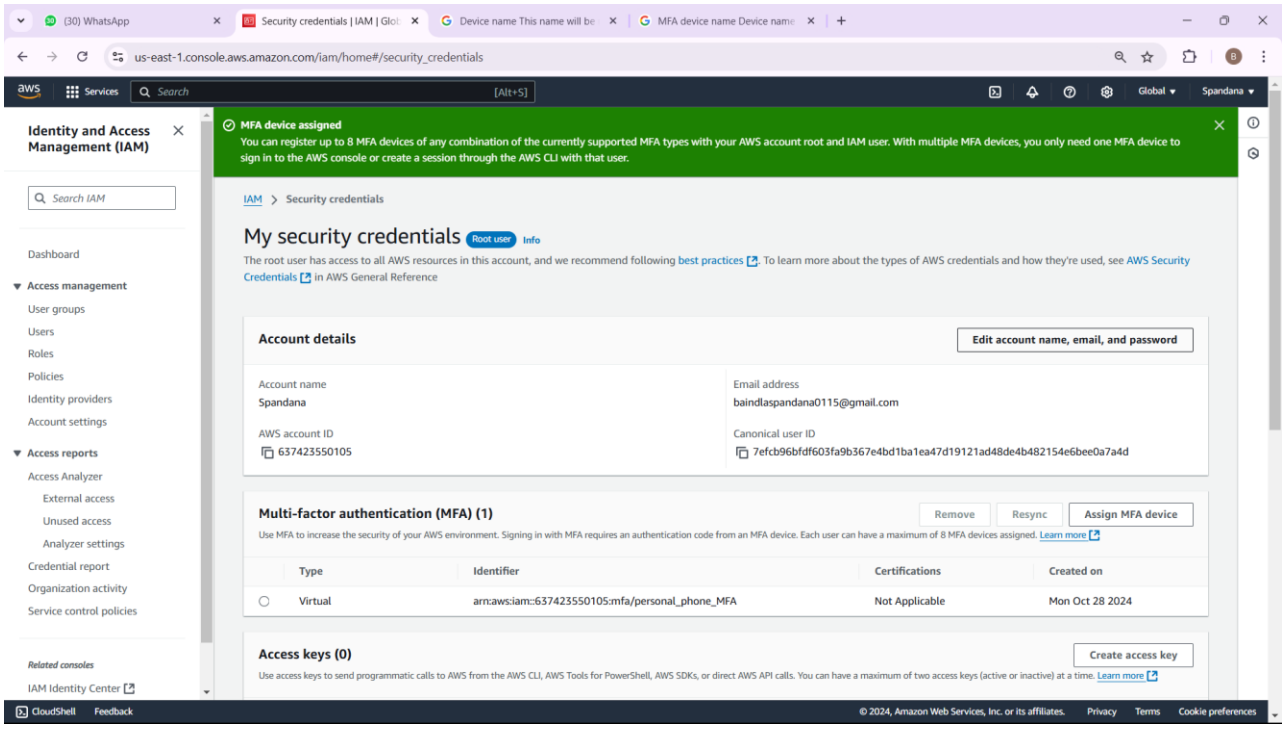
I have selected Authentication app for MFA , First we need to install Authy app on our mobiles so that we can able to do the further process. We need to give a name for MFA, and in the AWS console, a QR code will be displayed. Use the authentication app for the scan this QR code.

If our app does not support for the scanning we need to enter the provided secret key manually. The app will start generating the 6 digit codes. Enter the first code in MFA Code 1 and wait for the second code and enter in the place of MFA Code 2



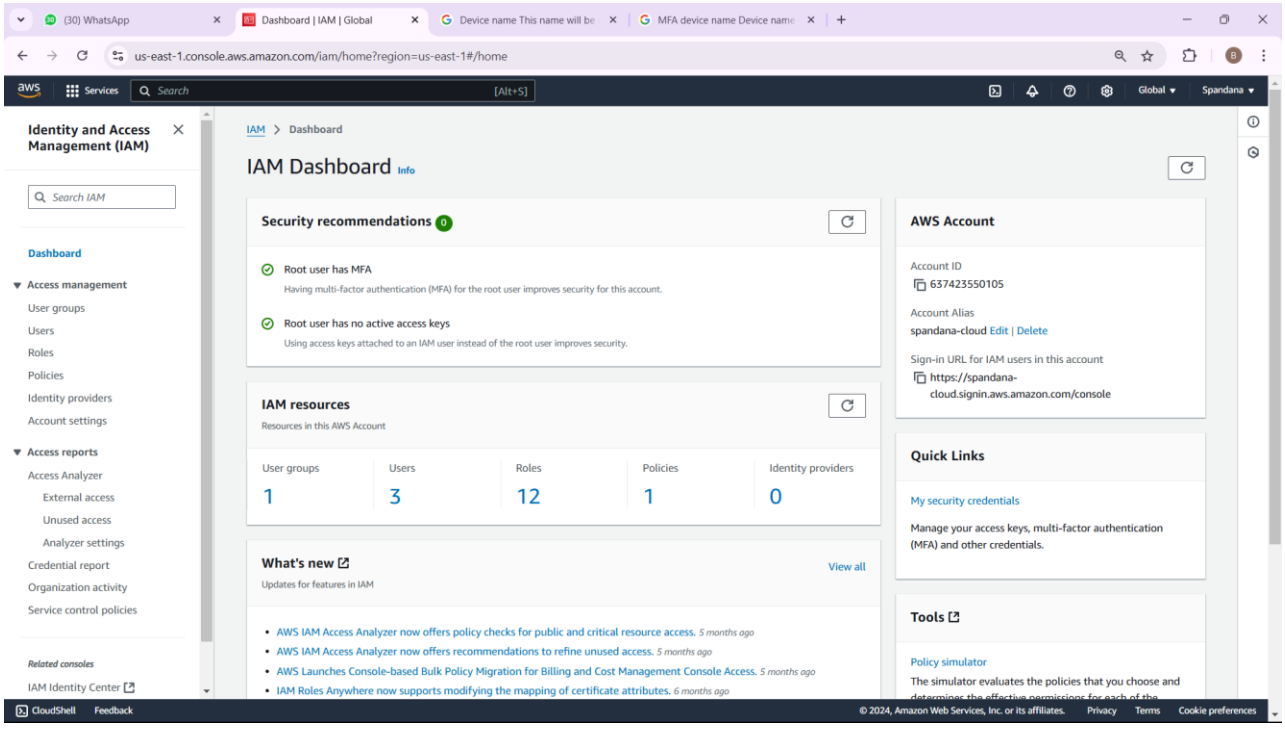
After Entering all the Information we need to confirmation so we will see the conformation message indicating that the MFA device assigned.

We will see the one change in the MFA and if we want to check that we have to logout our AWS management console and we need to login again it will ask for the MFA Code for those who choose Authentication app.

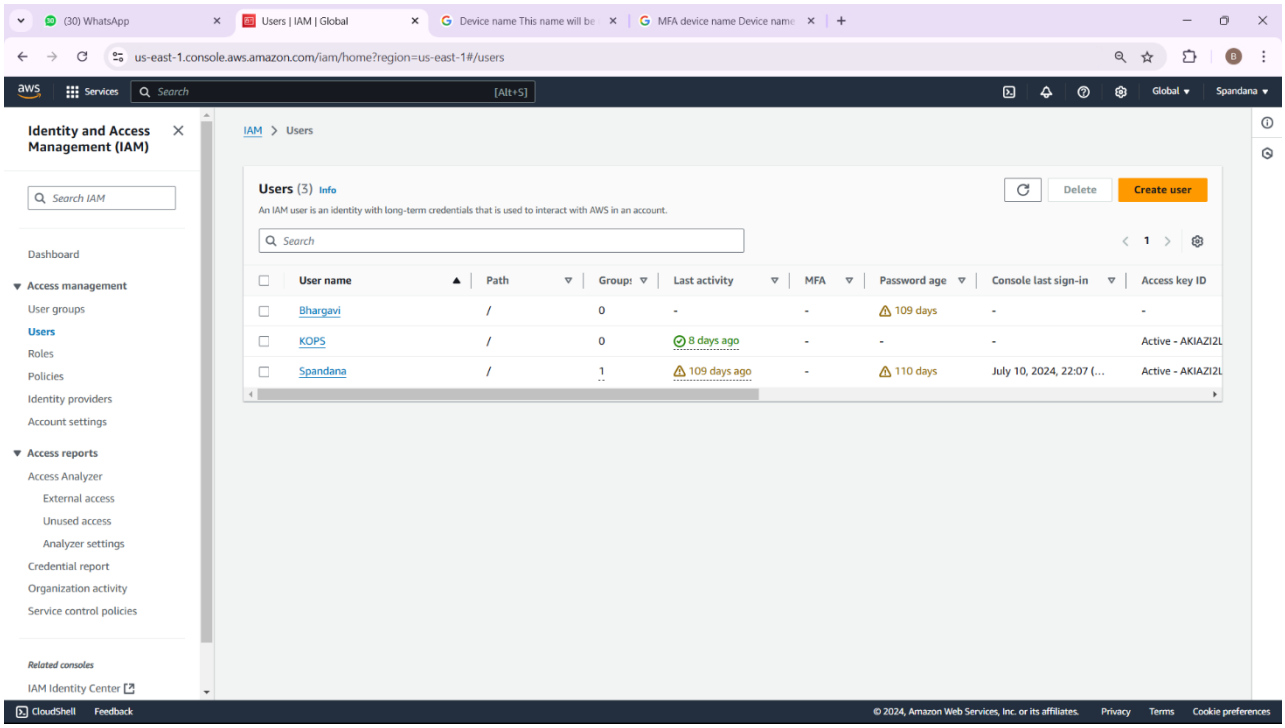


2.Creating a New user with access and check its default permissions.

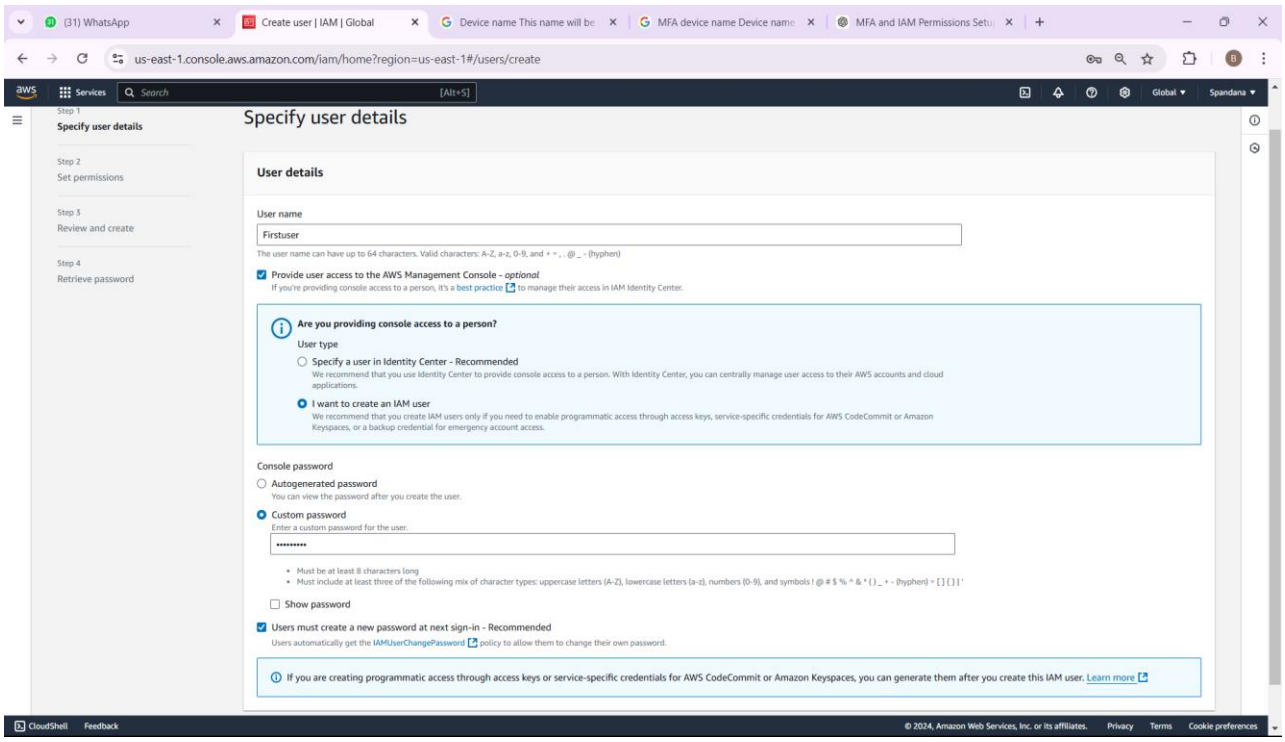
In the IAM dashboard we will see users in the left panel.



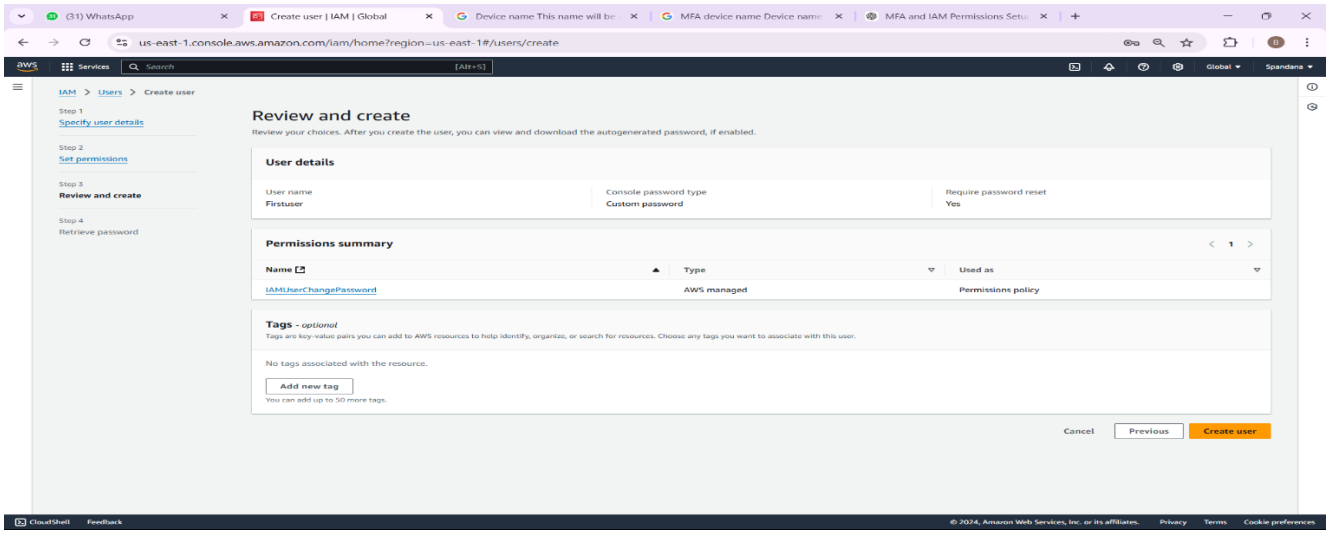
Click on the Users which is there below the user groups.

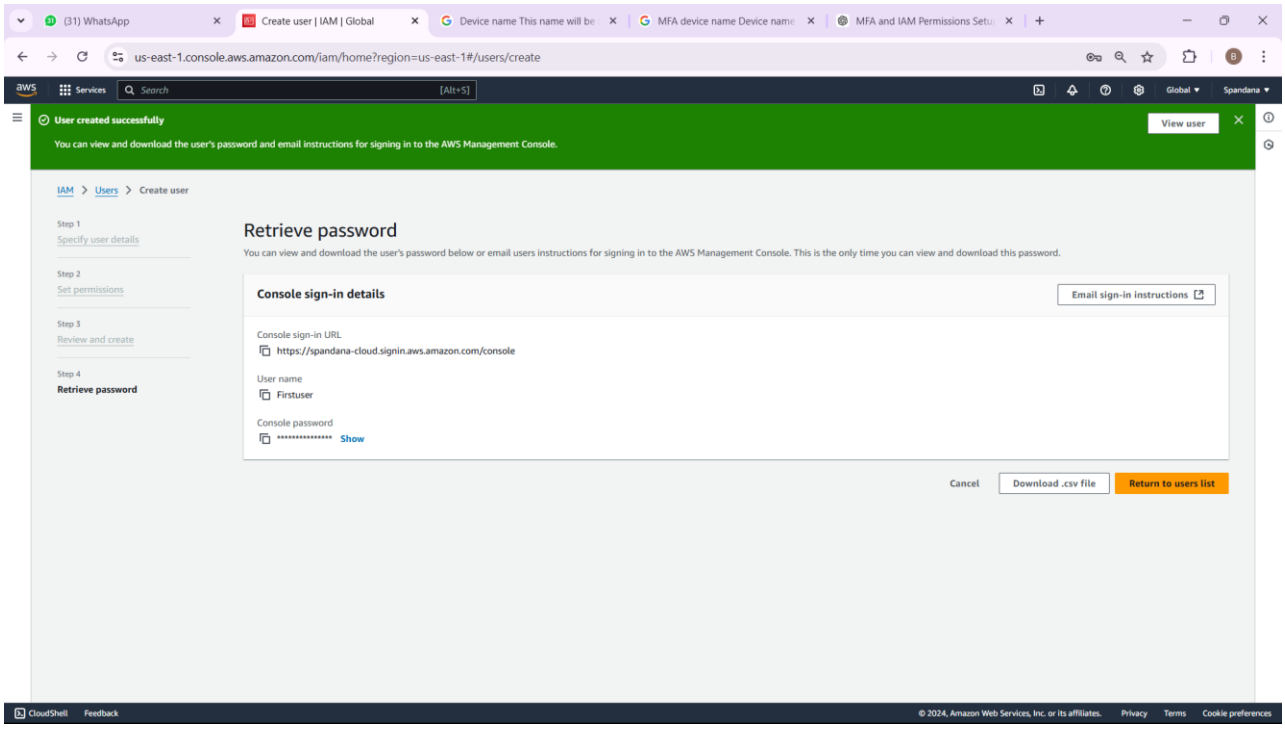


When we click on the Create user this page willl be open in this page we need to Enter our Username for new user and select AWS management console access and set a custome password for security purpose. If we select the Autogenerated password we can able to see that password after creation of user. So we will choose custom password for privacy and security purpose.



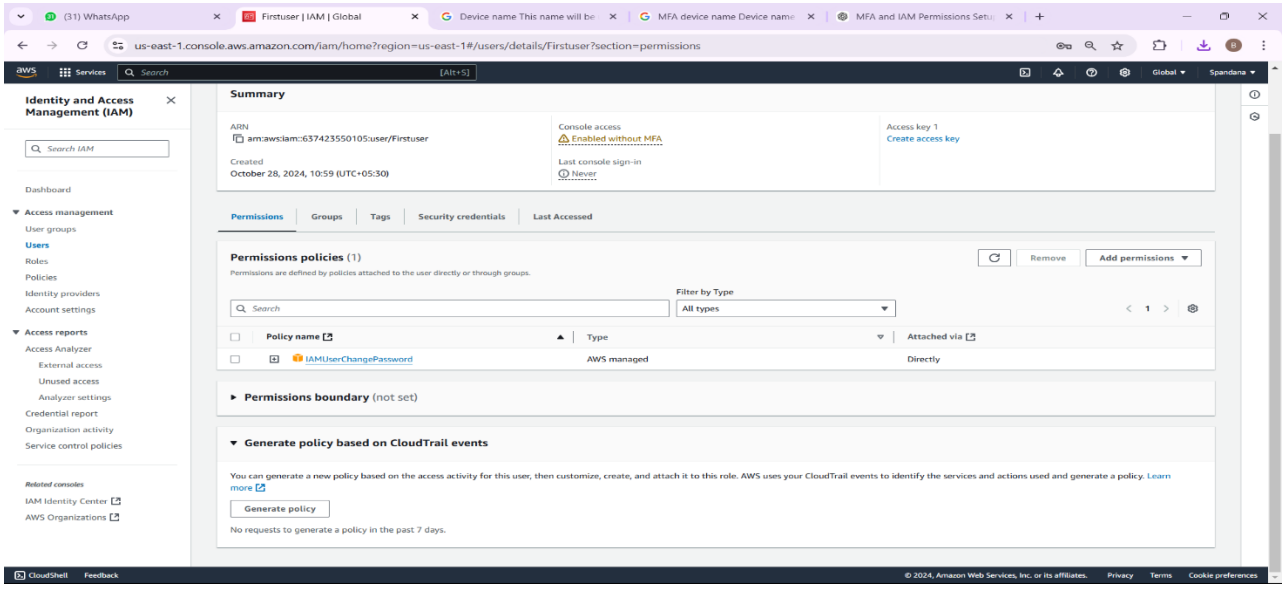
We will click on the check box below show password that is Users must create a new password at next sign in it is recommended for the security purpose. Ater that review and then Create user after creation we will see user's access key and secrect access key save these credentials securely.





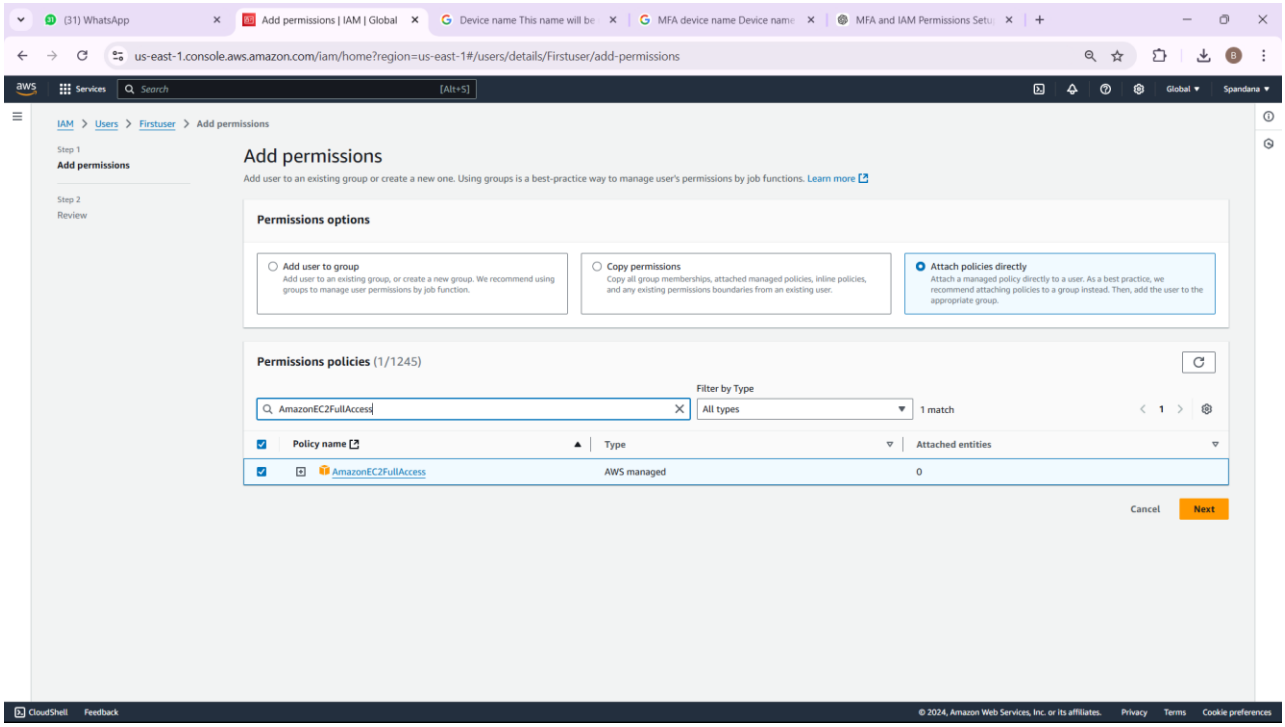
Check Default Permissions:

After creation of user click the user of newly created IAM user for which we want to check the default permissions. We will move to the user details page, Click on the permissions tab. The permissions tab is nothing but below image. Default permissions for the user is **IAMUserChangePassword** this is the default permission for the user.

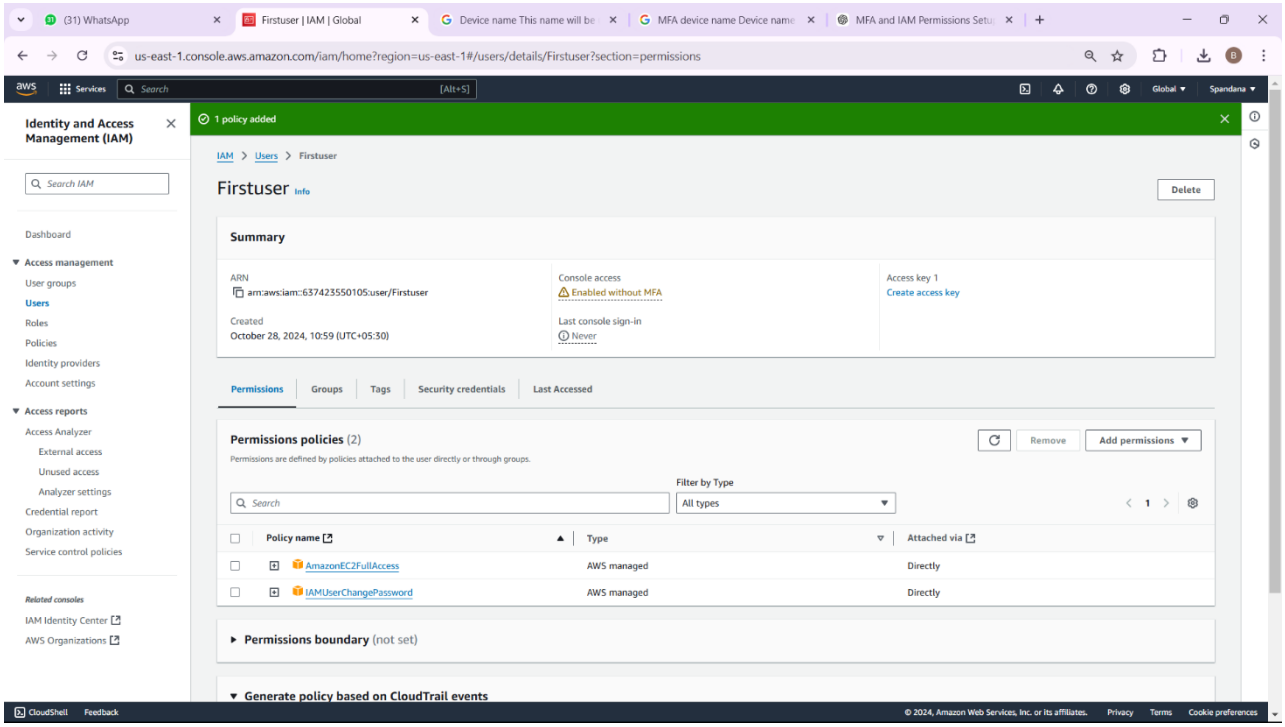


4.Assign full permissions to the user

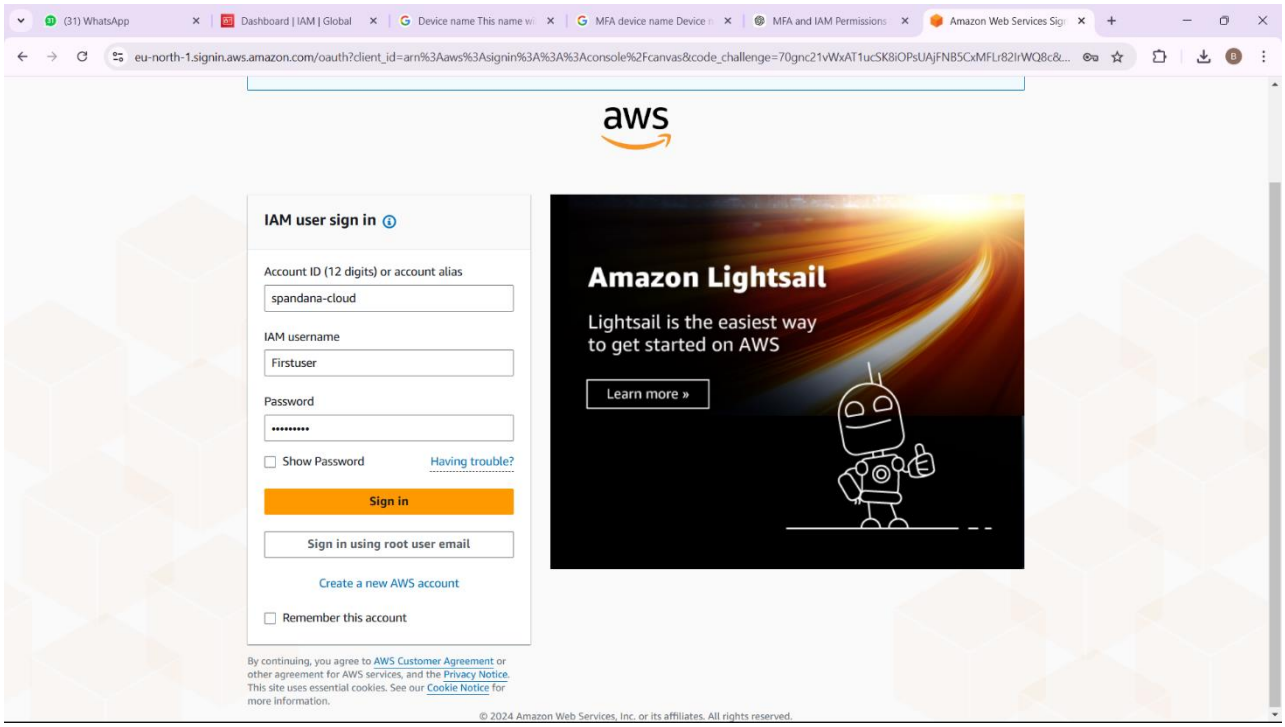
We can able to see the Add Permissions click on Add Permissions > Attach existing policies directly.



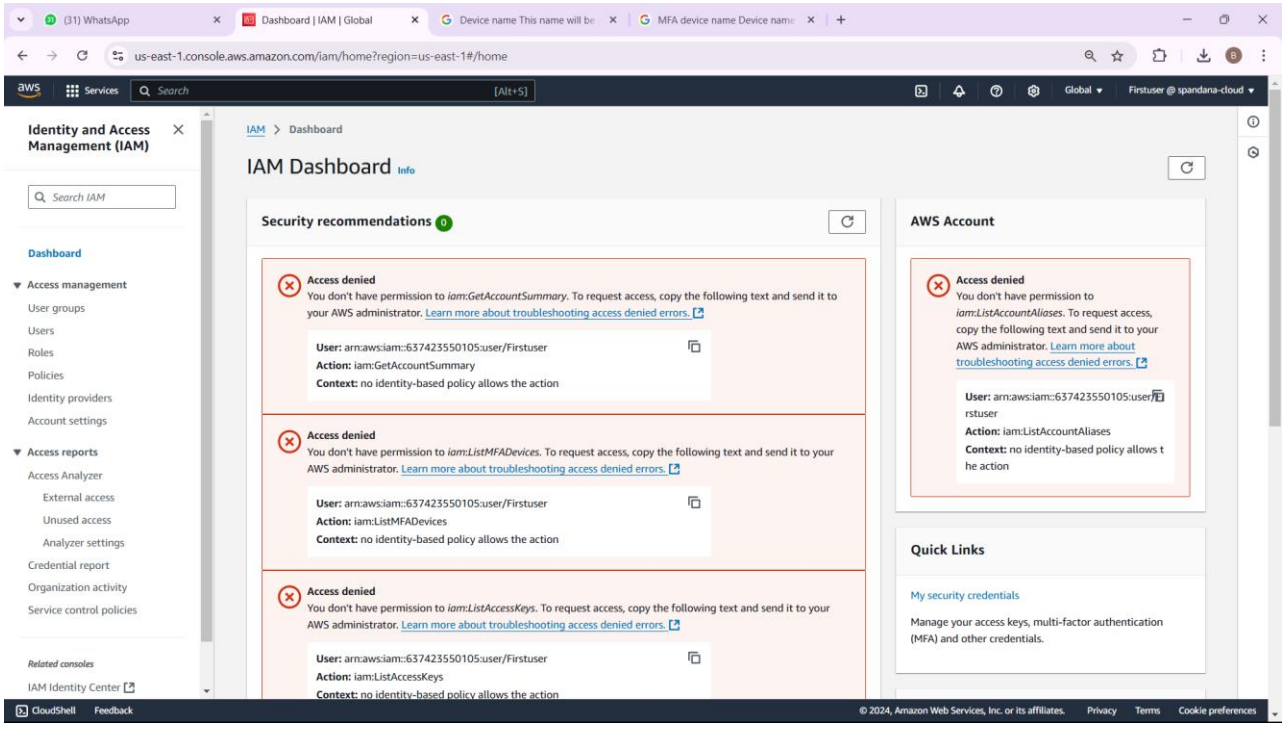
In the search bar, type **AmazonEC2FullAccess** and check the box next to the policy. Click Next and then Add permissions.

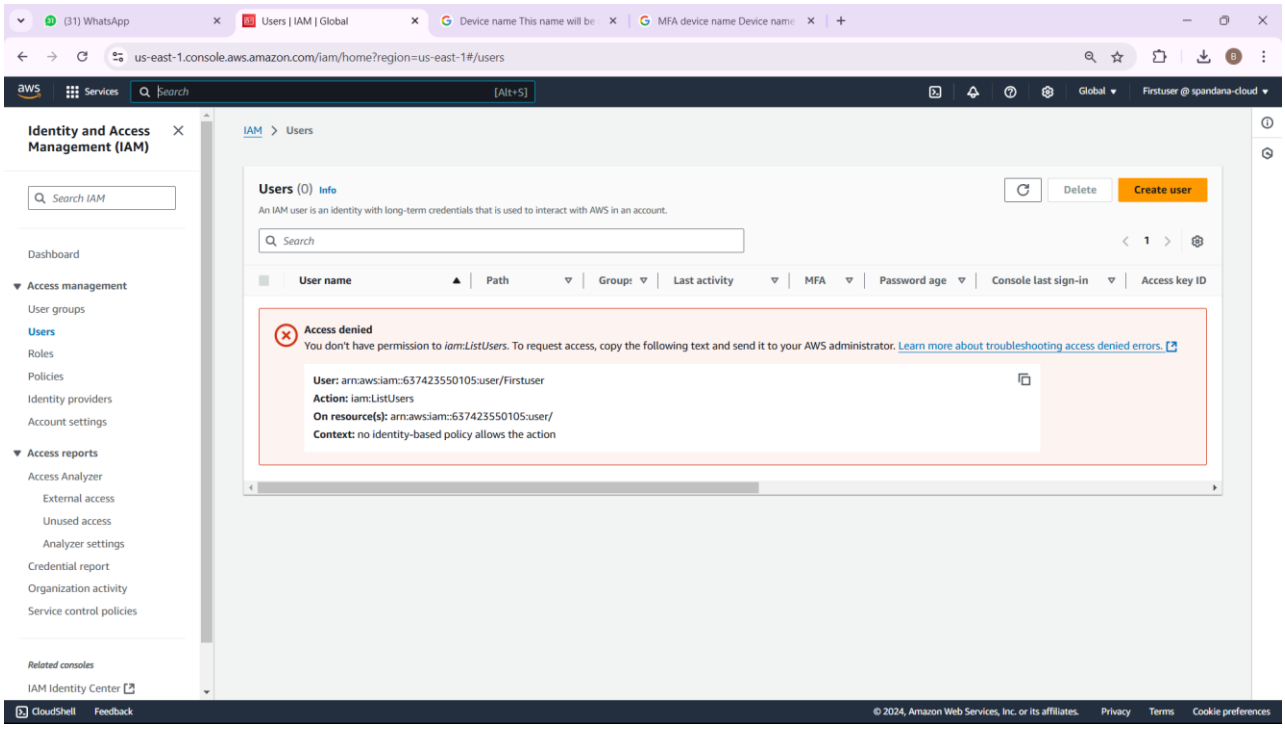


Log in as the new user: Use the credentials of the new user to log in to the AWS Management console.



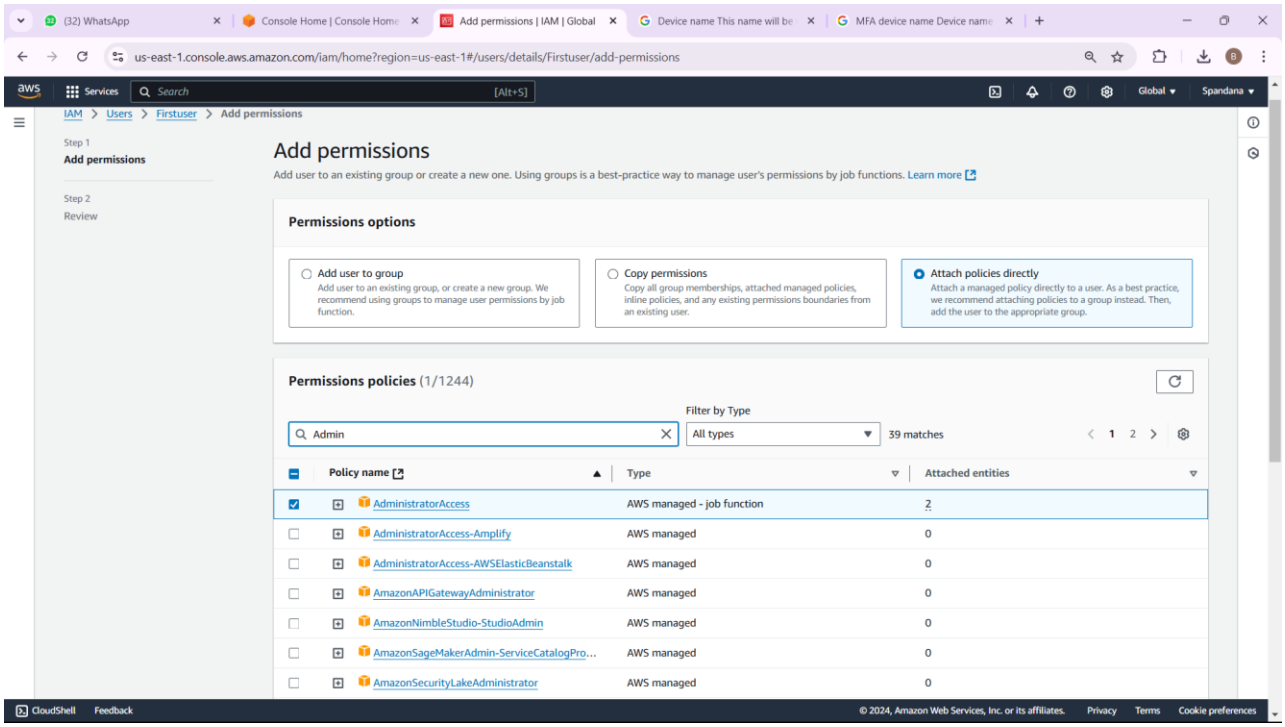
If we check the access to EC2 we can able to perform EC2 related tasks and we have all permissions to access EC2 but when we navigate to other services like IAM, S3 and confirm that we do not have access because EC2 full permissions should restrict access to other services.

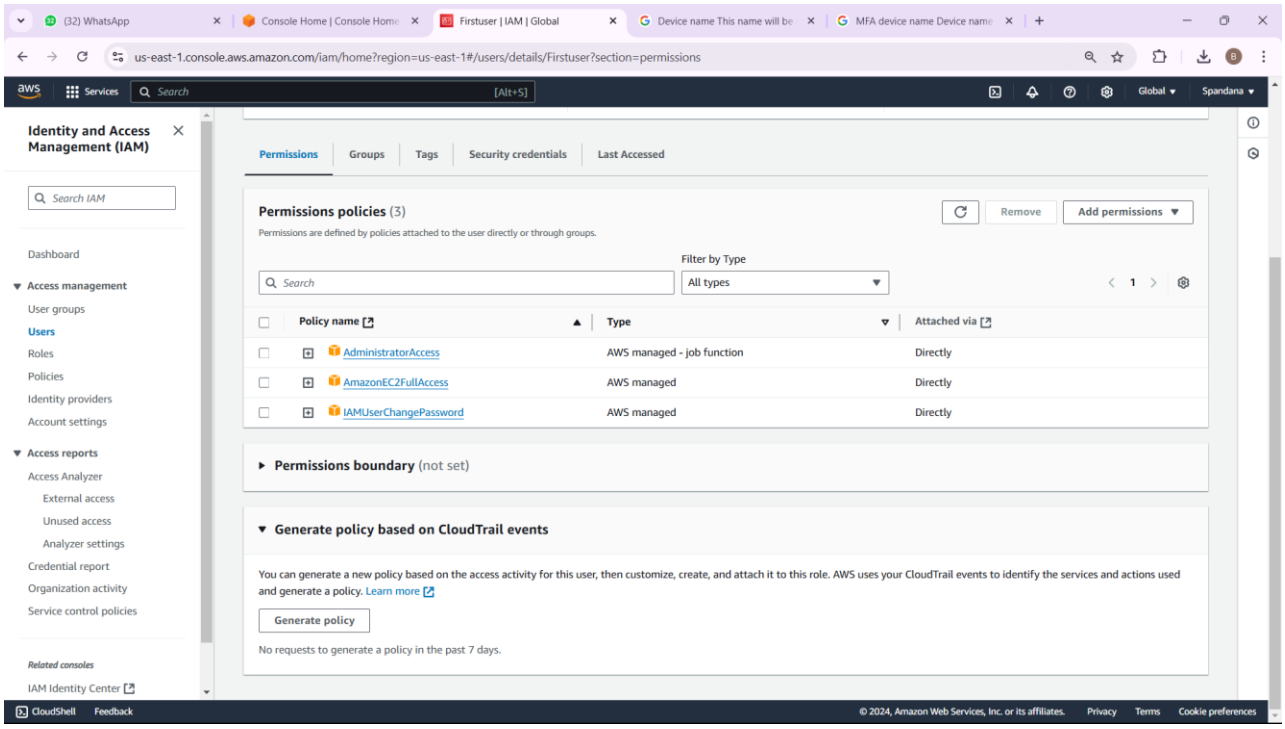


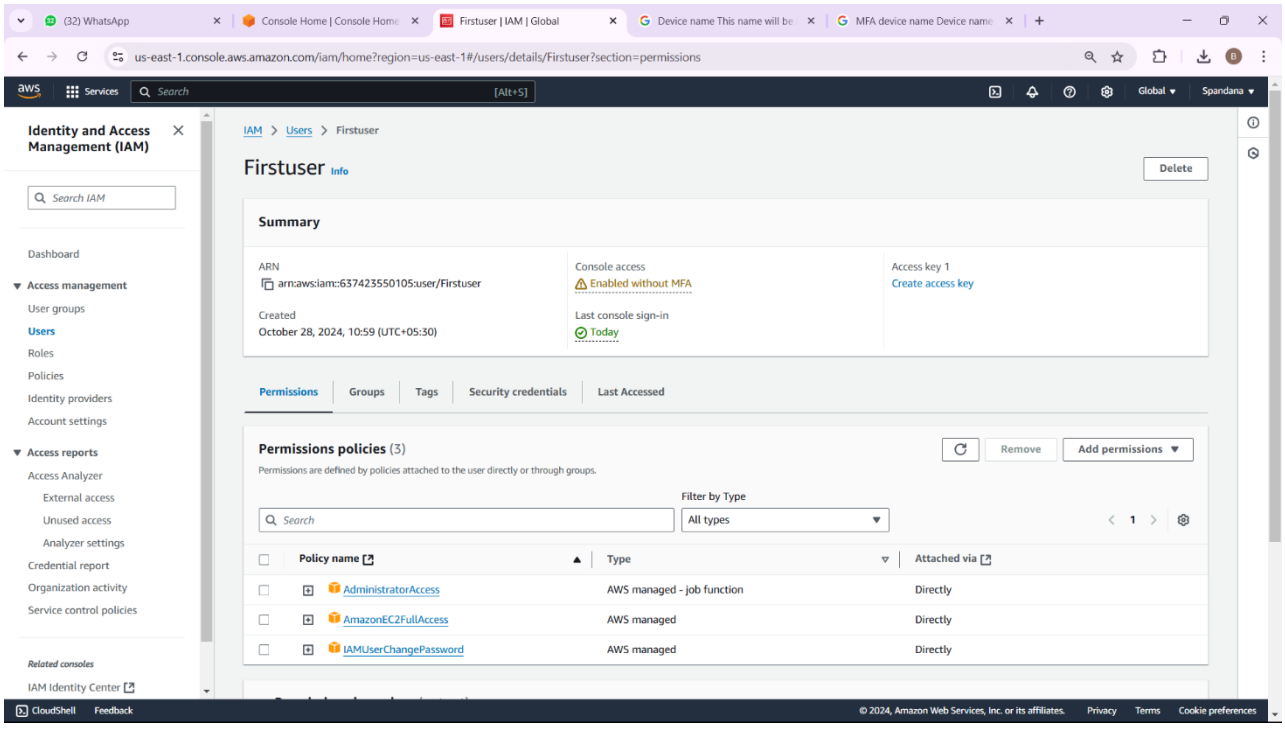


4.PROVIDING ADMINISTRATIVE PERMISSIONS TO THE USER.

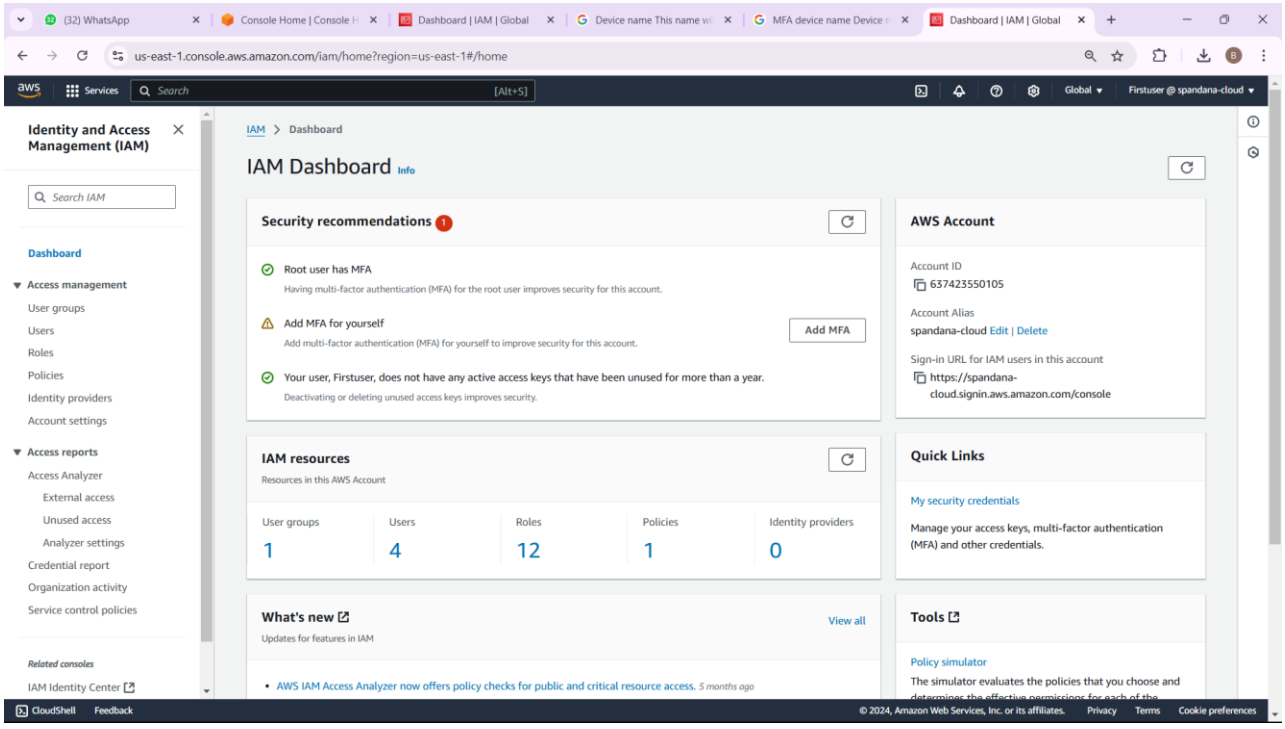
Go back to the IAM dashboard and select the user again click on the Permissions tab click on Add permissions > Attach existing policies directly and search for **AdministratorAccess** and select that policy click next and then Add permissions.




Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access Analyzer
External access
Unused access
Analyzer settings
Credential report
Organization activity
Service control policies
Related consoles
IAM Identity Center
Permissions
Groups
Tags
Security credentials
Last Accessed
Permissions policies (3)
Permissions are defined by policies attached to the user directly or through groups.
Remove
Add permissions
Filter by Type
Search
All types
Policy name
Type
Attached via
AdministratorAccess
AWS managed - job function
Directly
AmazonEC2FullAccess
AWS managed
Directly
IAMUserChangePassword
AWS managed
Directly
Permissions boundary (not set)
Generate policy based on CloudTrail events
You can generate a new policy based on the access activity for this user, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. Learn more
Generate policy
No requests to generate a policy in the past 7 days.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

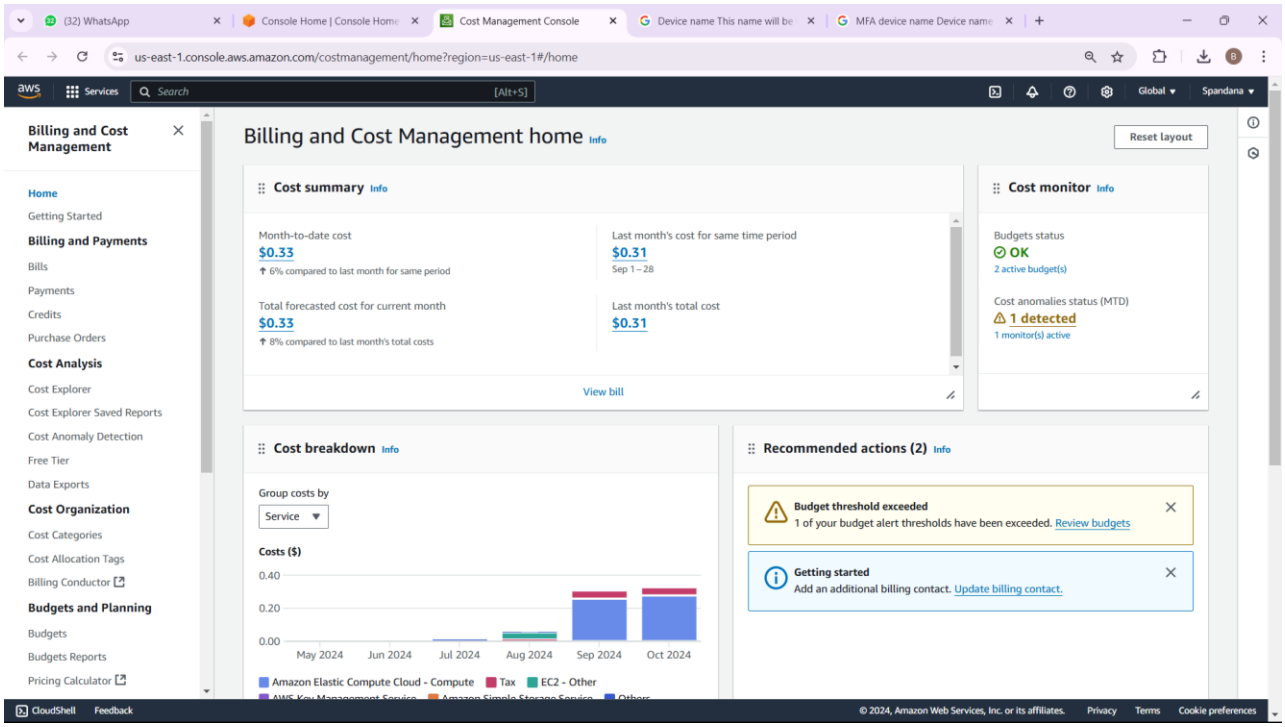

Identity and Access Management (IAM)
IAM > Users > Firstuser
Firstuser Info
Delete
Summary
ARN
arn:aws:iam::637423550105:user/Firstuser
Console access
Enabled without MFA
Access key 1
Create access key
Created
October 28, 2024, 10:59 (UTC+05:30)
Last console sign-in
Today
Permissions
Groups
Tags
Security credentials
Last Accessed
Permissions policies (3)
Permissions are defined by policies attached to the user directly or through groups.
Remove
Add permissions
Filter by Type
Search
All types
Policy name
Type
Attached via
AdministratorAccess
AWS managed - job function
Directly
AmazonEC2FullAccess
AWS managed
Directly
IAMUserChangePassword
AWS managed
Directly
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

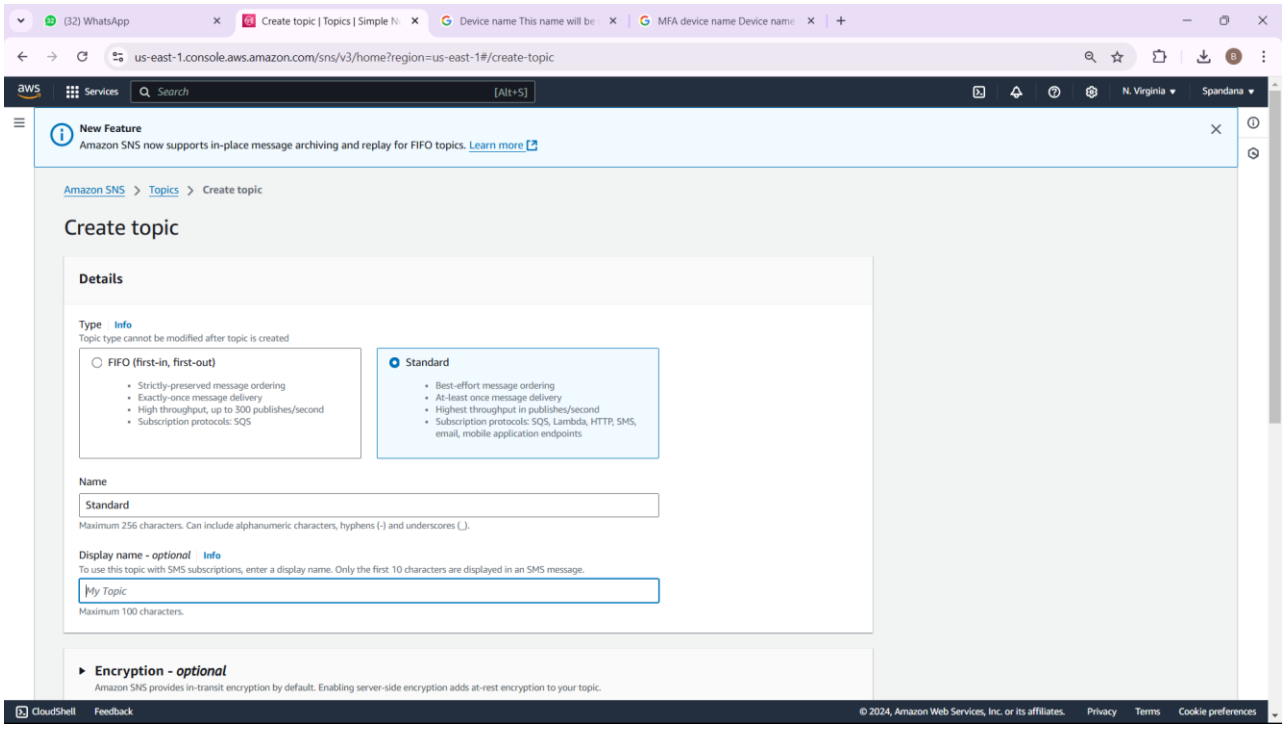
Login to the AWS Management console again as the user and confirm that now we have access to all services including the ability to manage IAM users and roles.



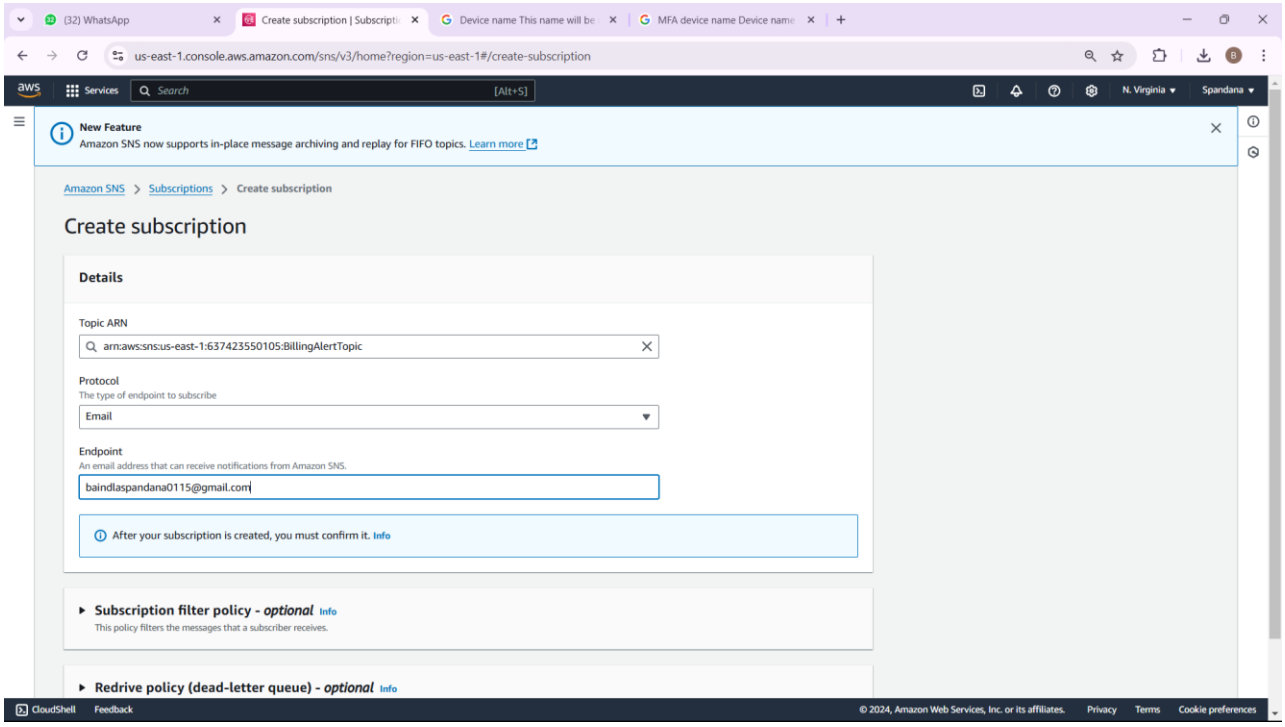
LAB 2-SETUP BILLING ALARAM

Go to the AWS Management Console and log in to your account search for Billing and Cost Management and select it.

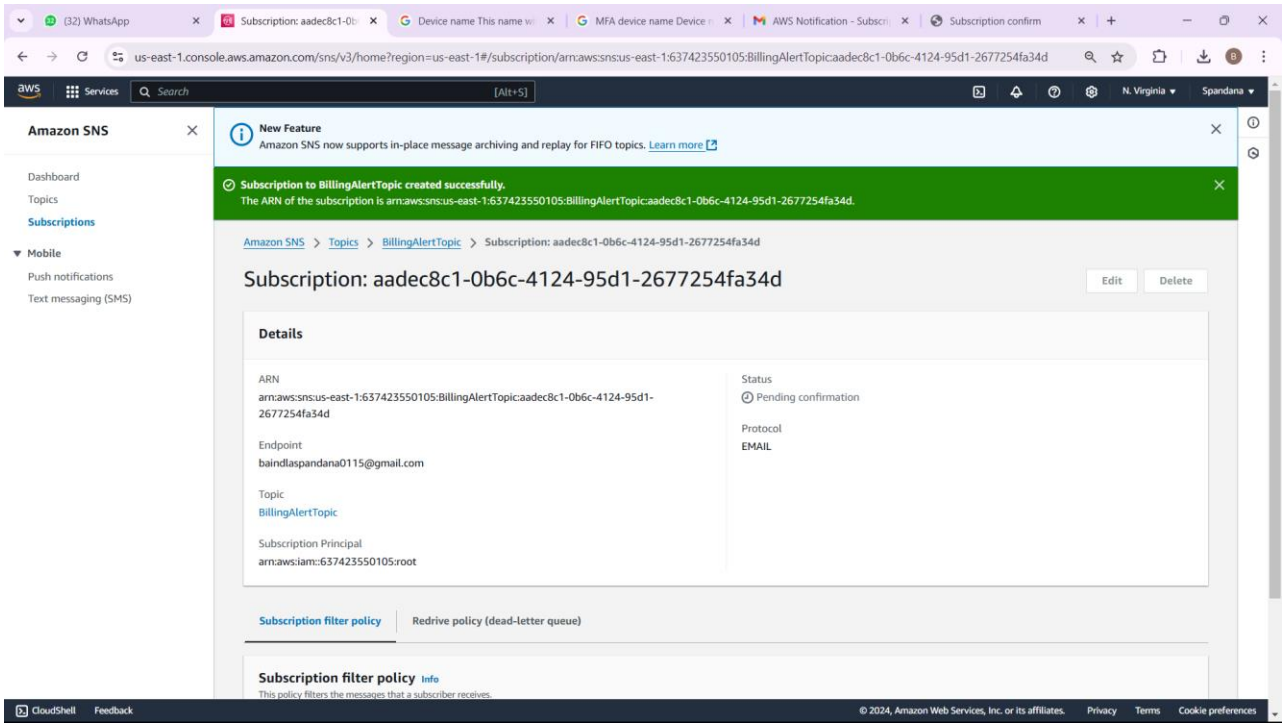




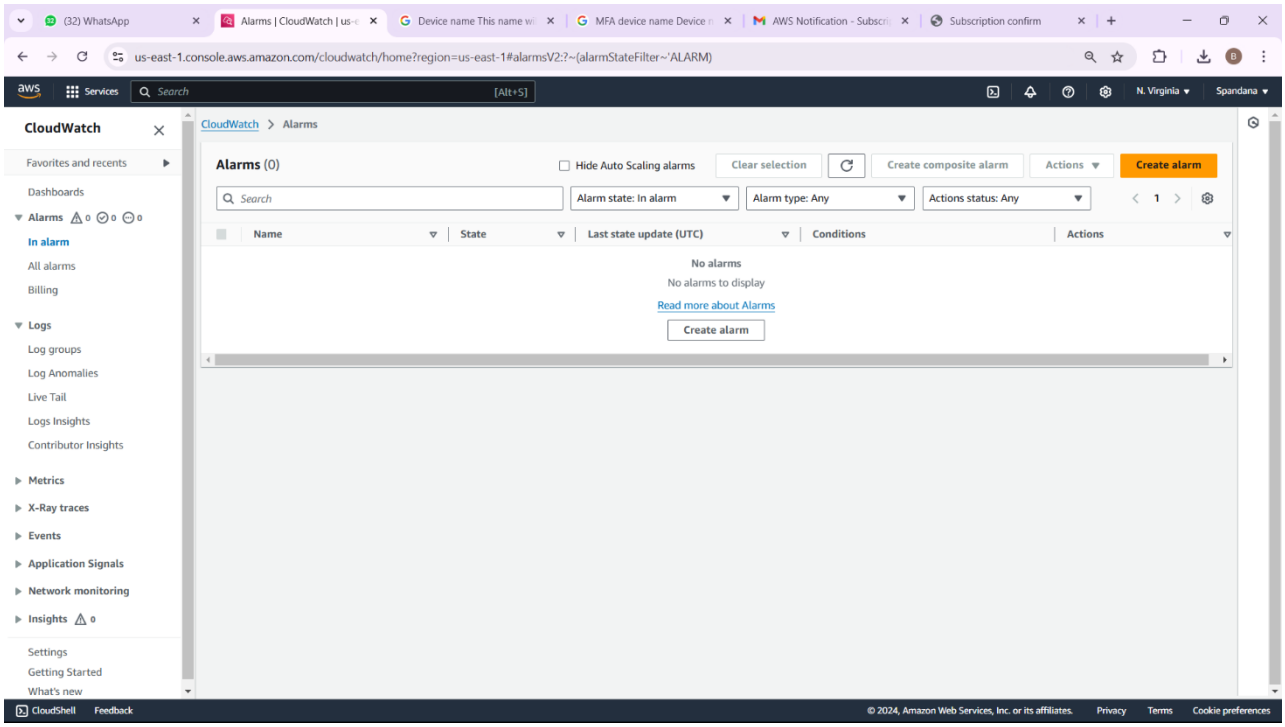
Create a cost budget, your desired threshold amount, and configure email alerts. This setup ensures you receive notifications whenever spending approaches or exceeds the specified budget, helping you monitor and manage costs effectively.



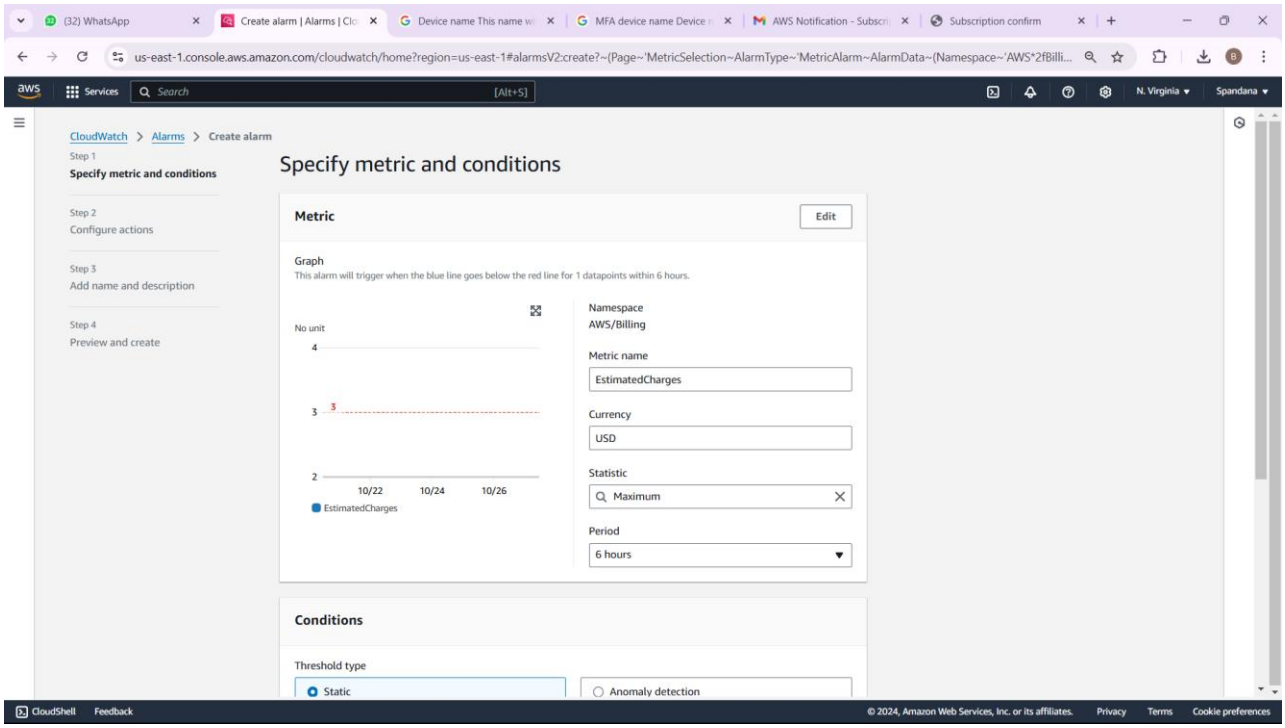
Subscription to BillingAlertTopics created successfully.



We have to go to Cloud Watch click on the alarm in the left side select billing under the metrics click on create alarm .

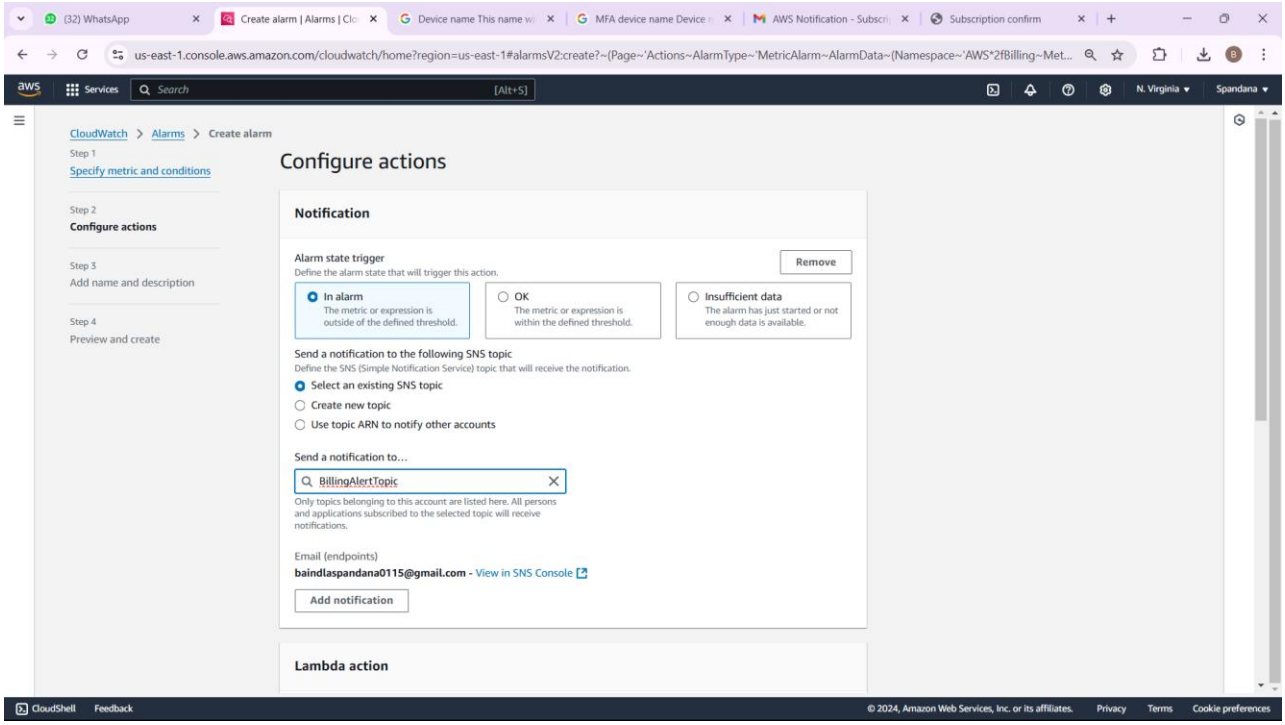


Select metric choose Billing and select Total estimated charge in our preferency currency click on select metric

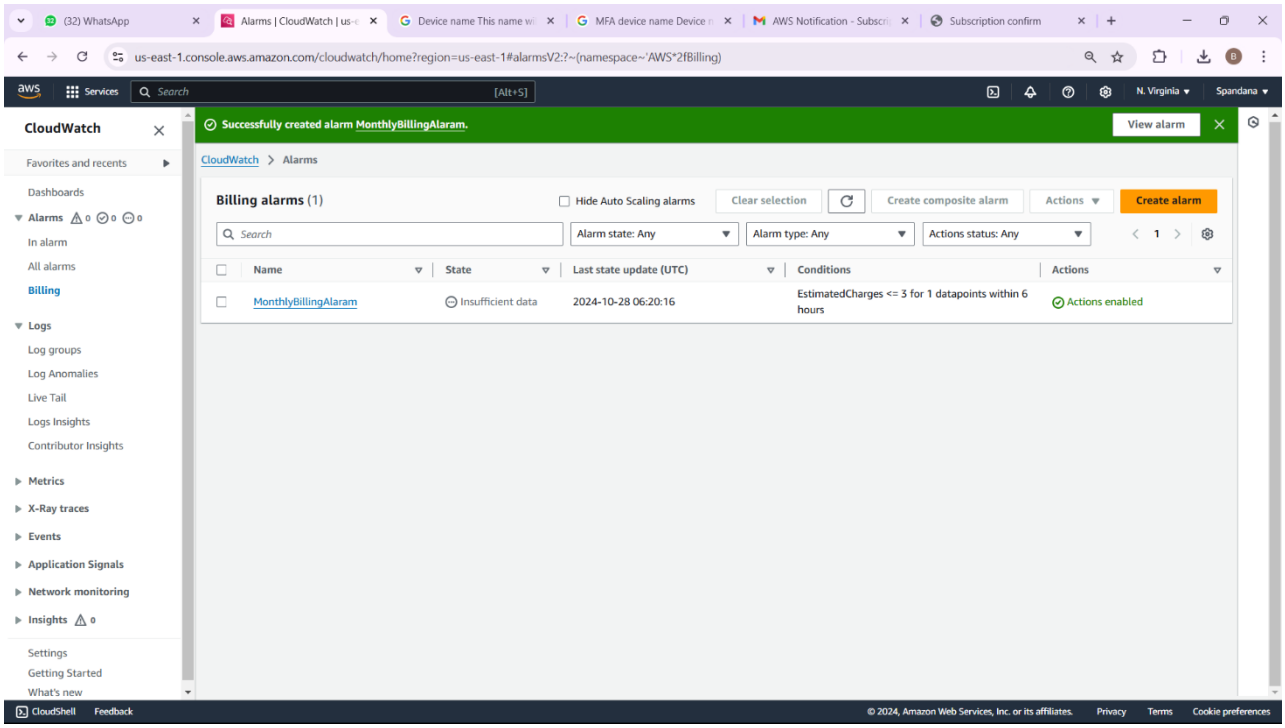


In the configuration actions section under notification choose send notification to the following SNS topic

Select the BillingAlertTopic optionally we can specify additional SNS topics if needed by adding topic ARN. Under the alarm state select the alarm to notify us whenever the alarm thresholds crossed.

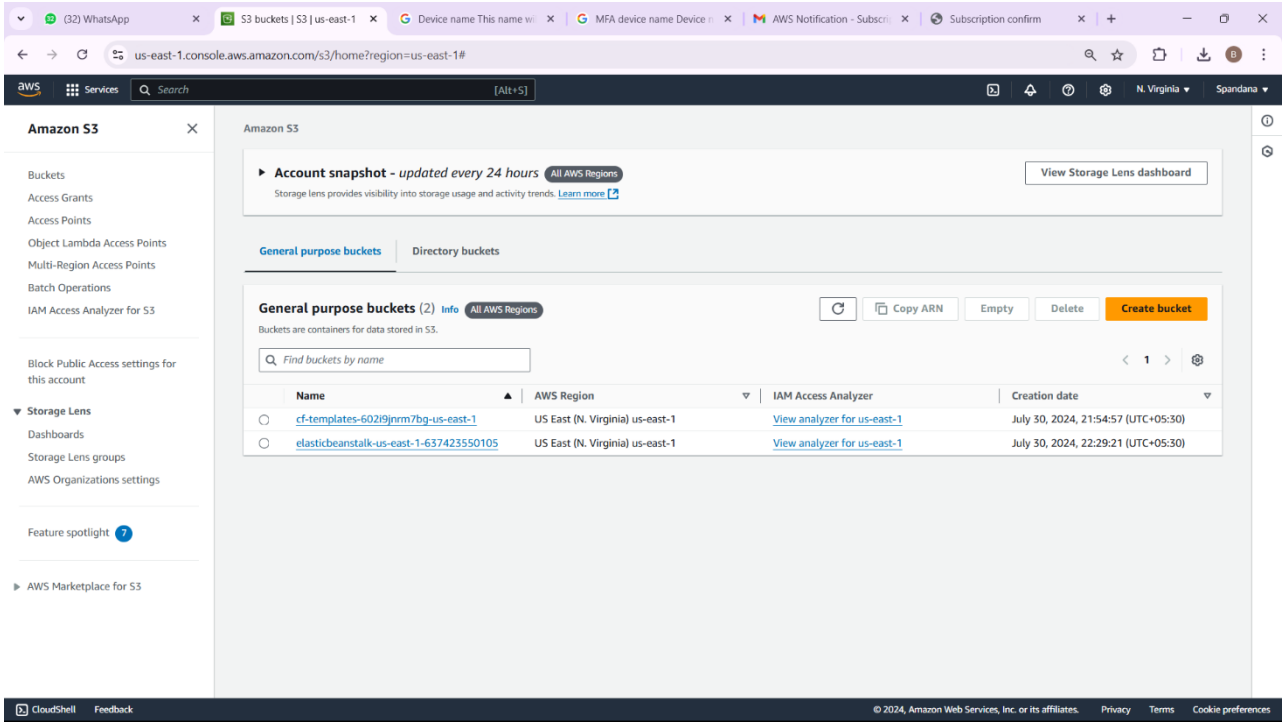


In the below image we will see our Billing alaram setup is done successfully.

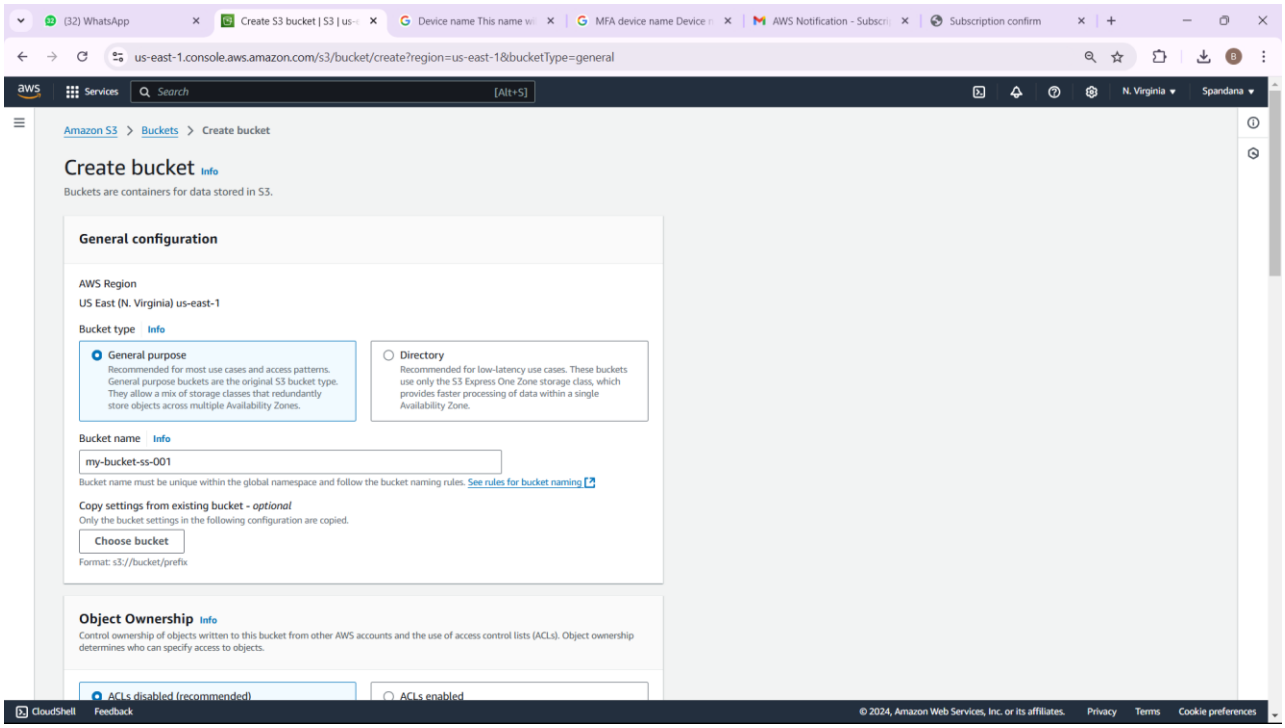


LAB-3 S3 BUCKET

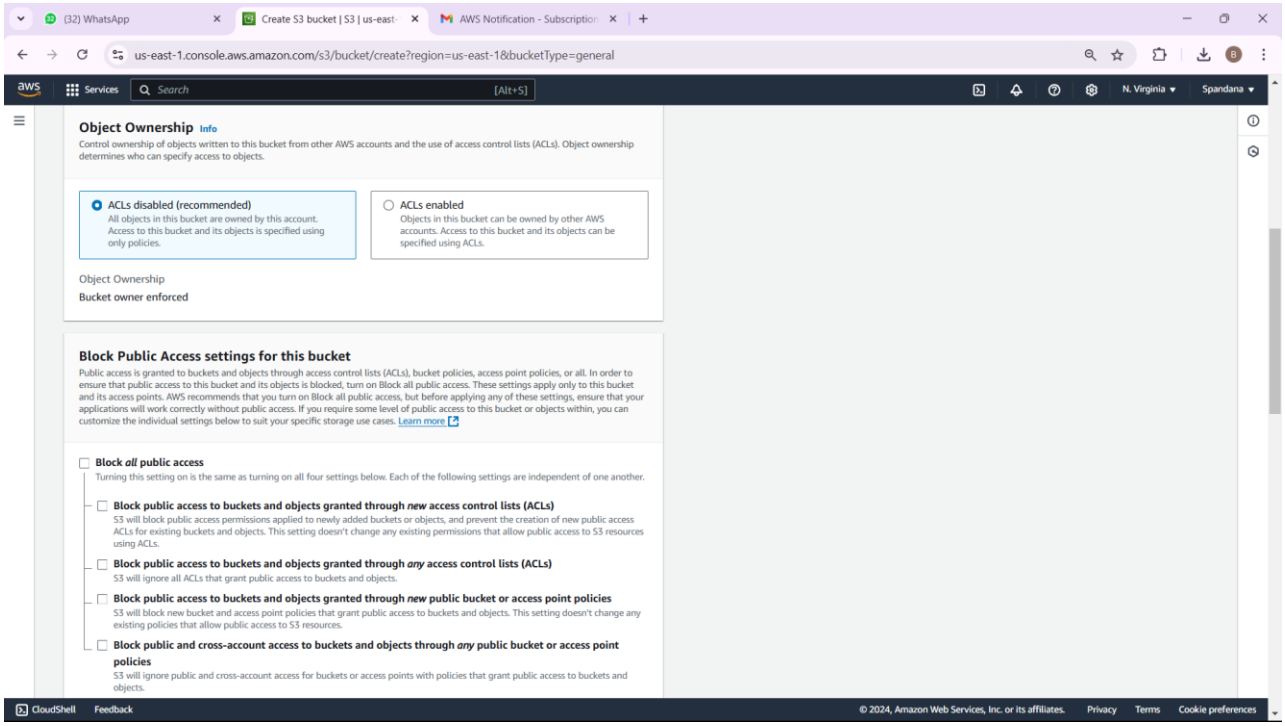
In the console search for S3 and open it we can able to see below image.



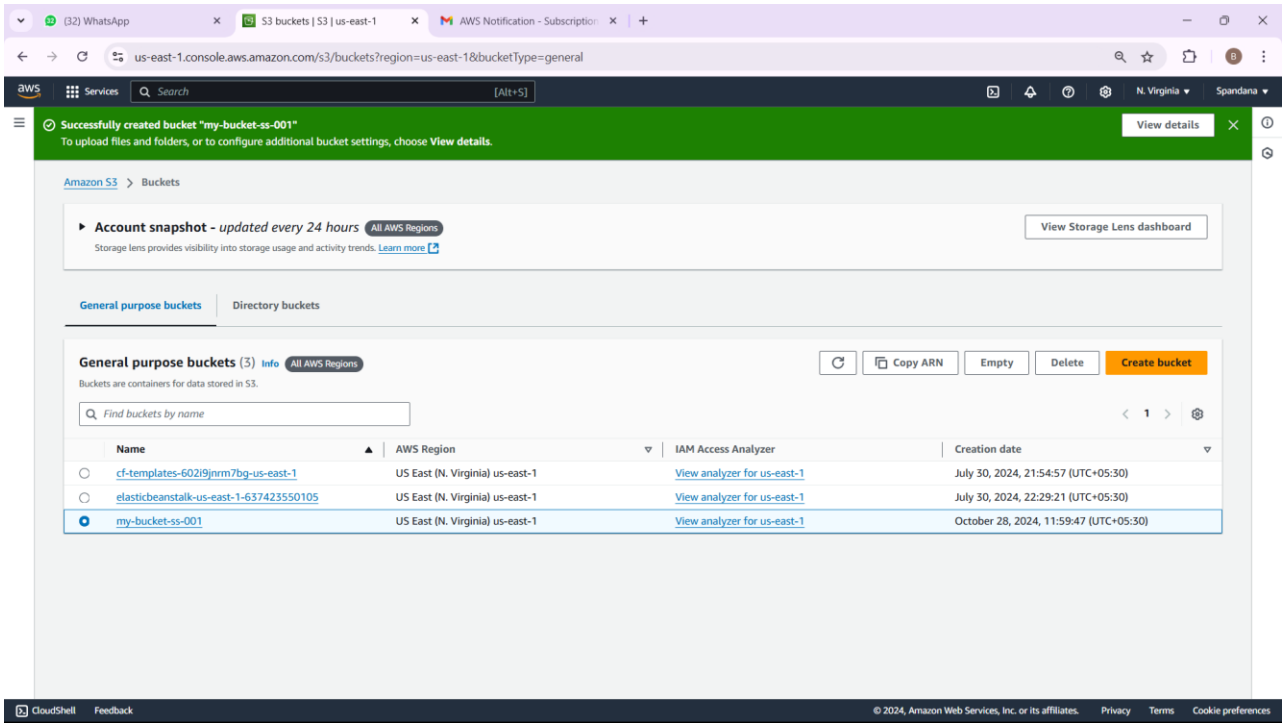
Click on create bucket and we have to give unique name for the bucket.



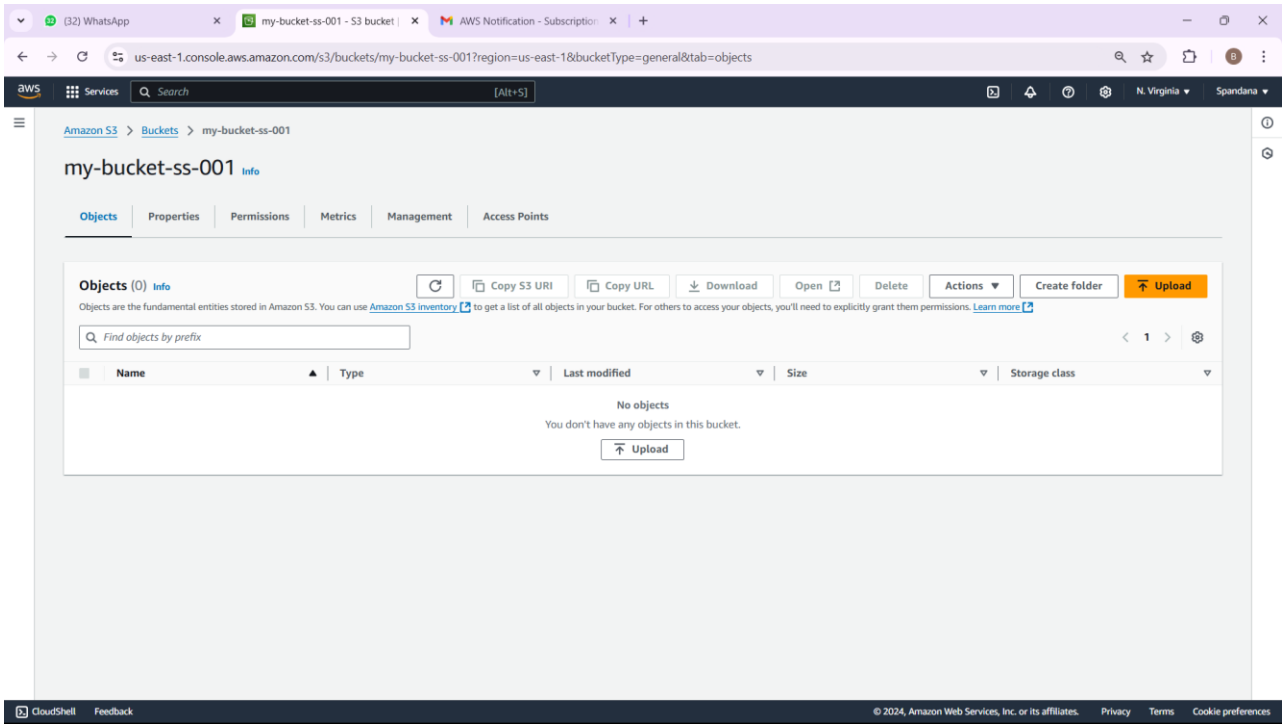
Choose other setting default for now and click on create bucket.



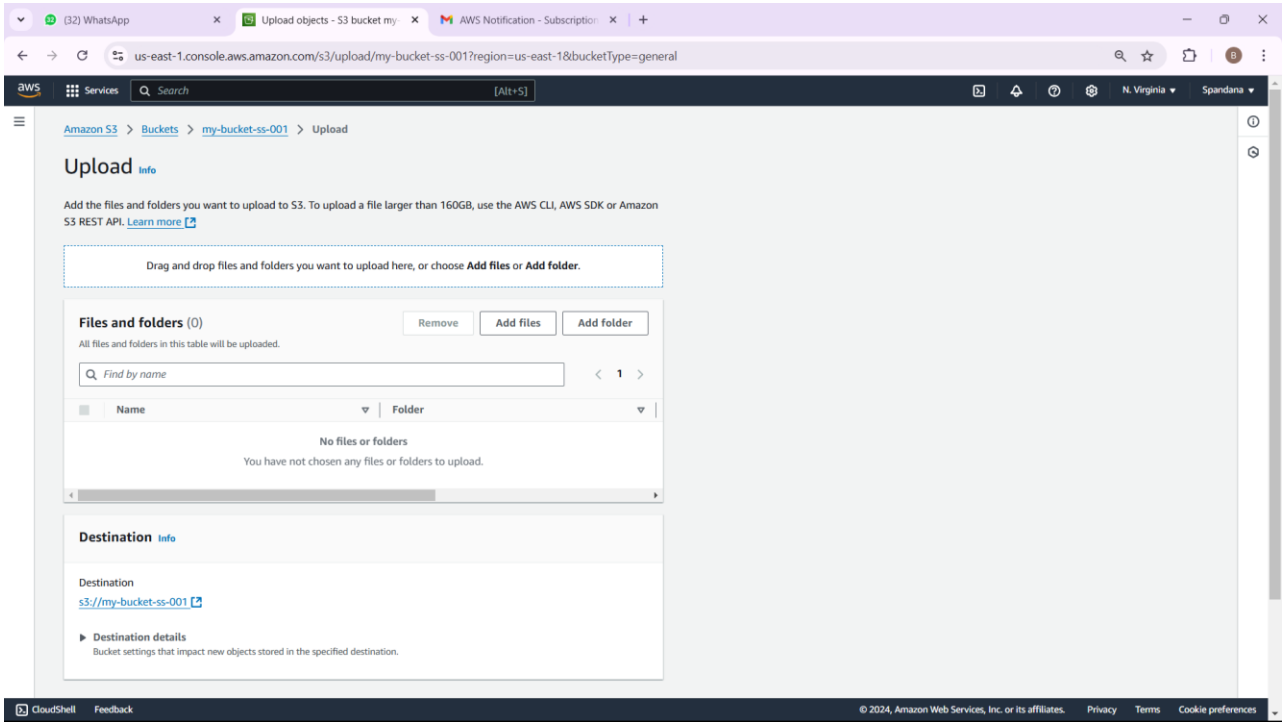
In the below image we can able to see out new bucket has been created successfully.



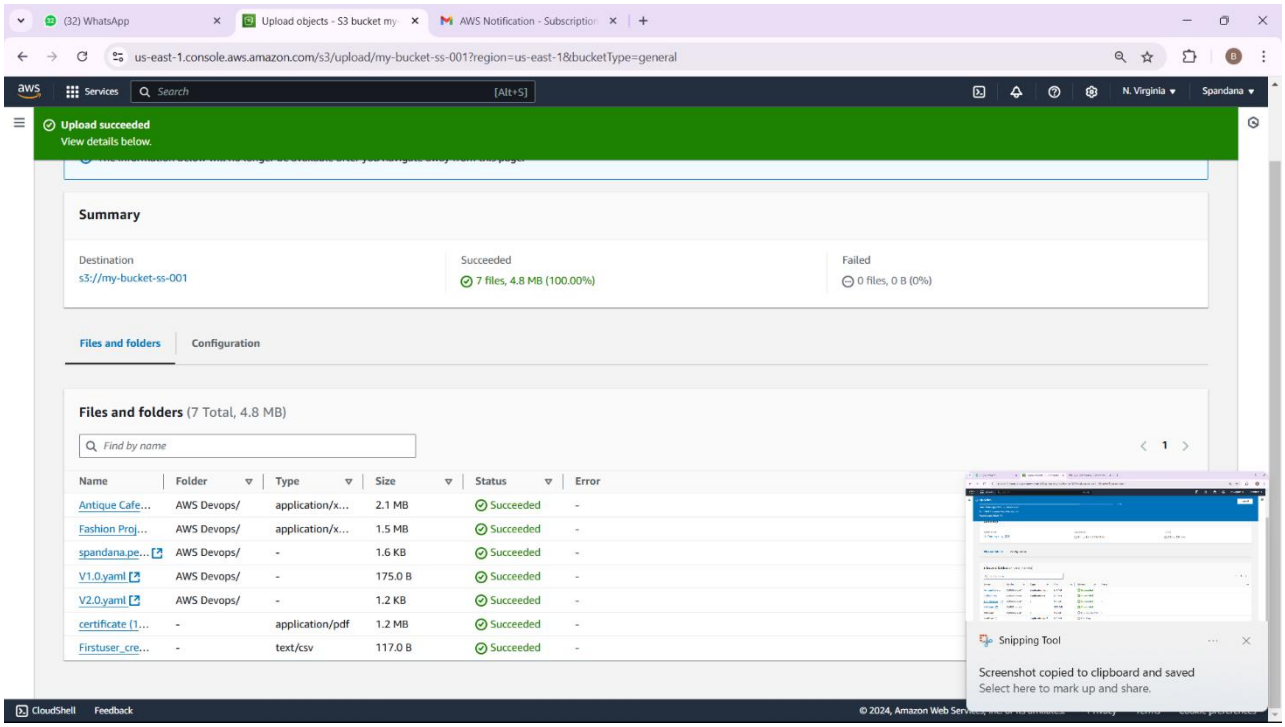
In S3 dashboard click on the name of the bucket we have just created to open it. We can able to see our bucket and there is a upload button on the right corner of the object.



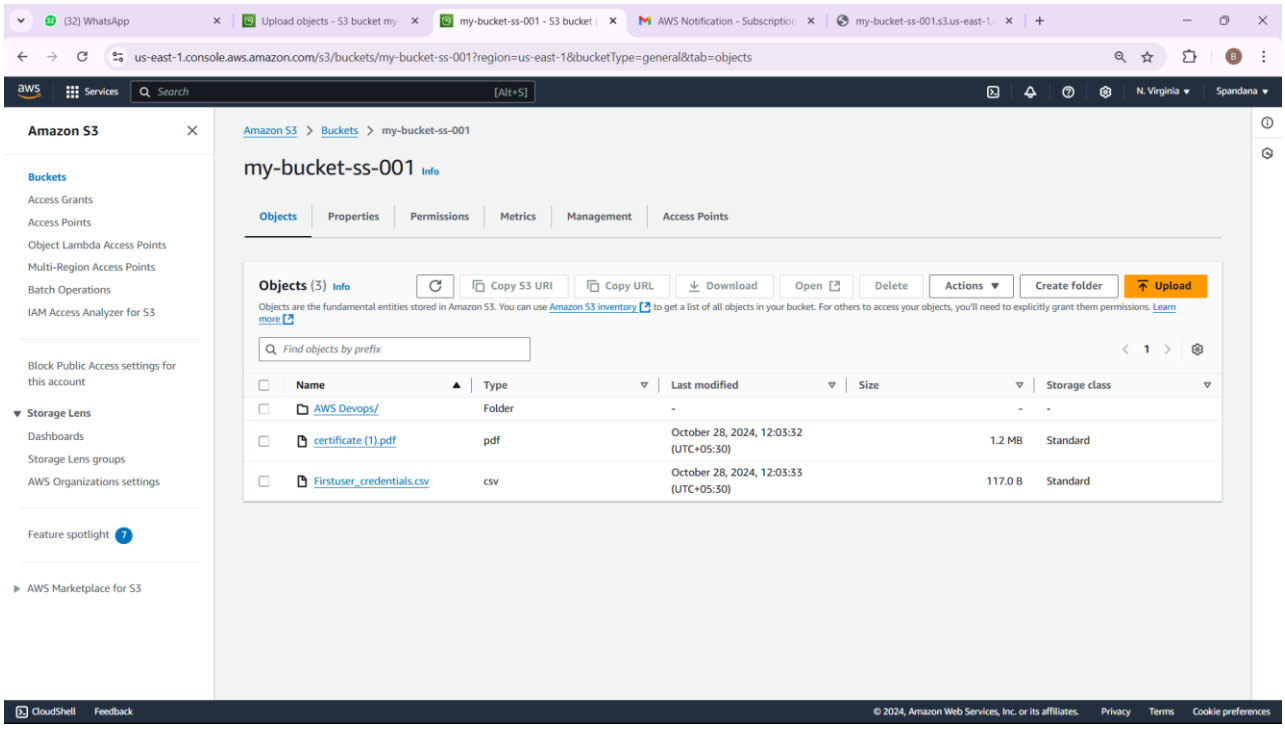
Click on upload choose add files and add folder and select the files and folders we want to upload and click on Upload to confirm.



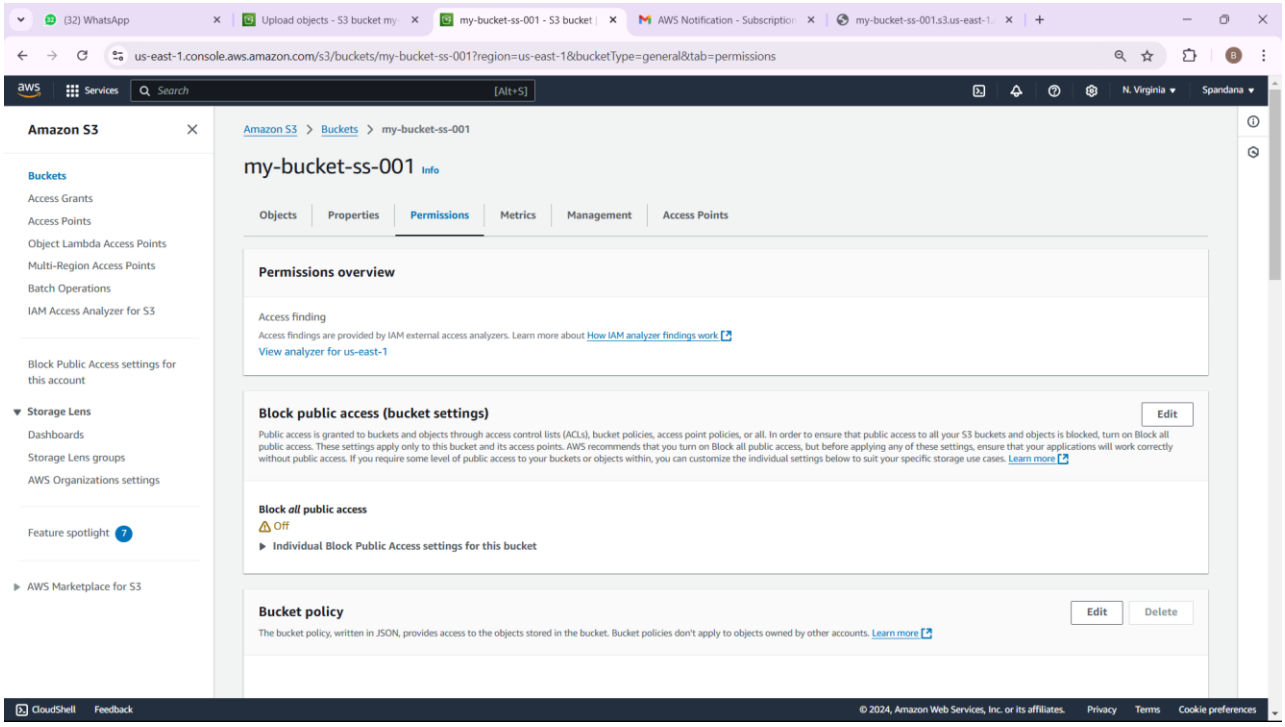
We can able to see our uploaded files and folder in my bucket.



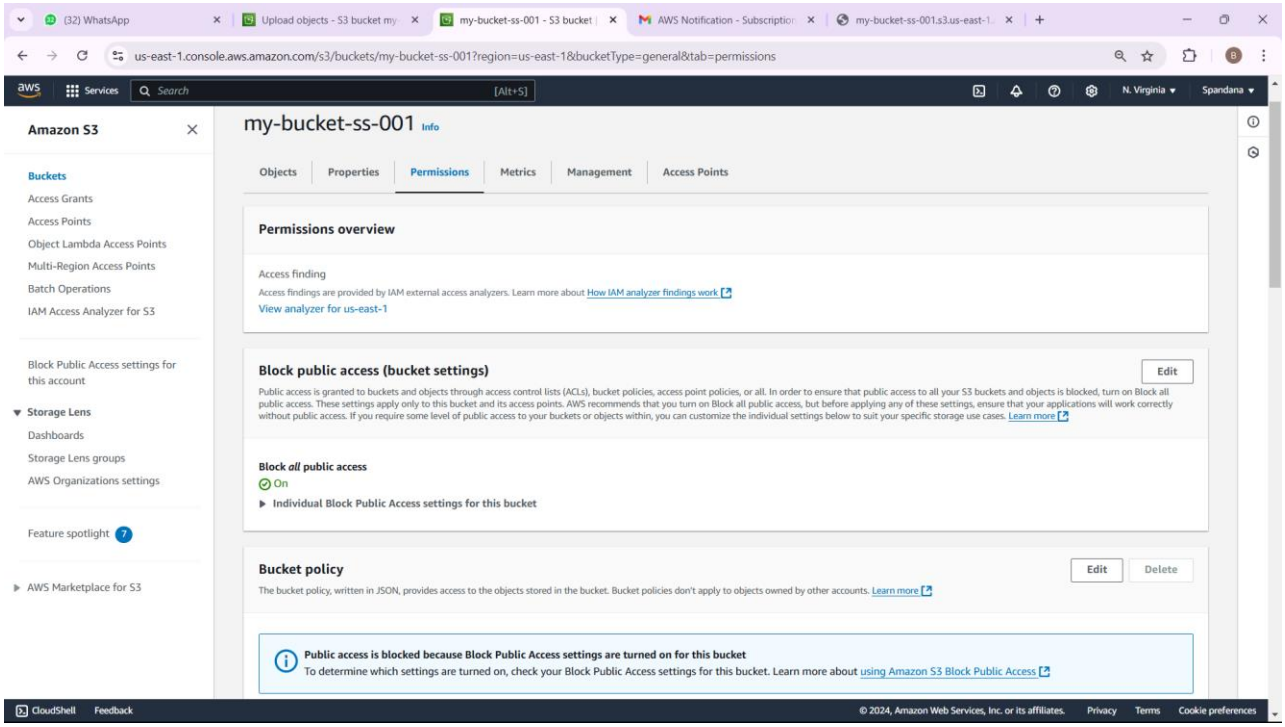
The below files and folder I have uploaded in my S3 Bucket.



Go to bucket and click on the Permission tab. We will notice that Block all public access is usually enable by default this restricts public access to the bucket.



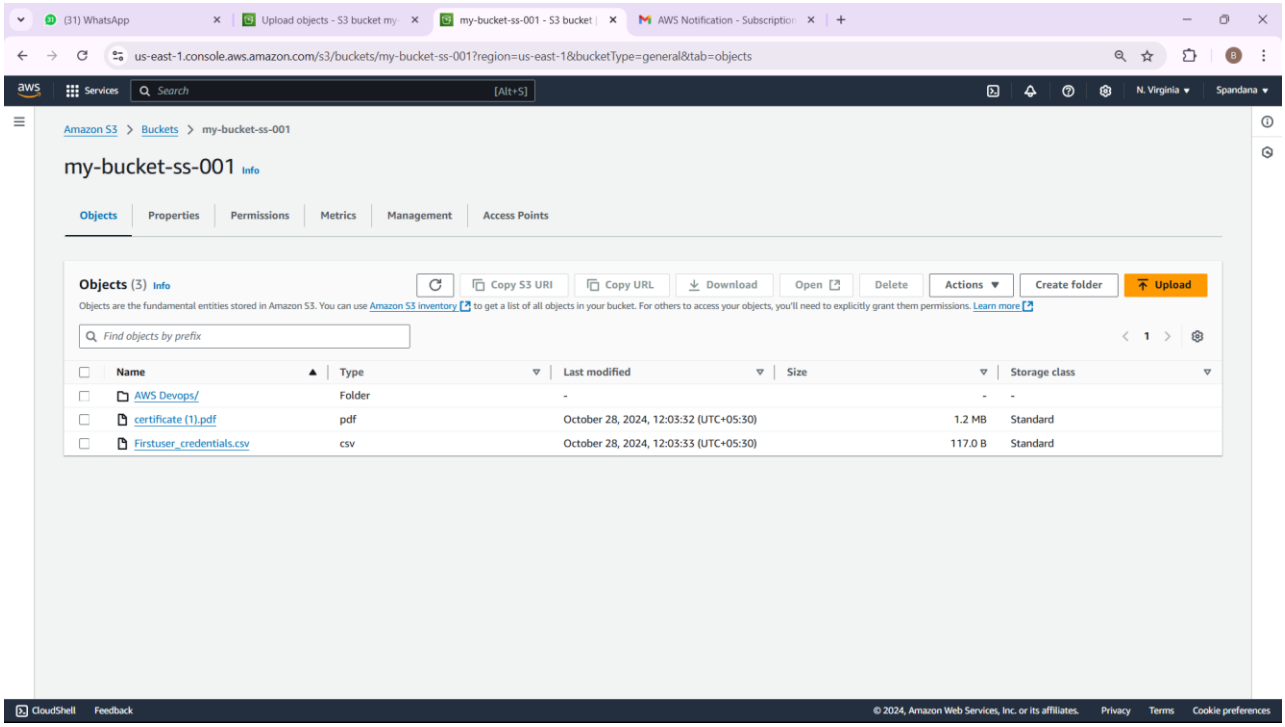
In the permissions tab of the file find the public access settings click edit and choose Grant Public read access to this object confirm by clicking save changes.



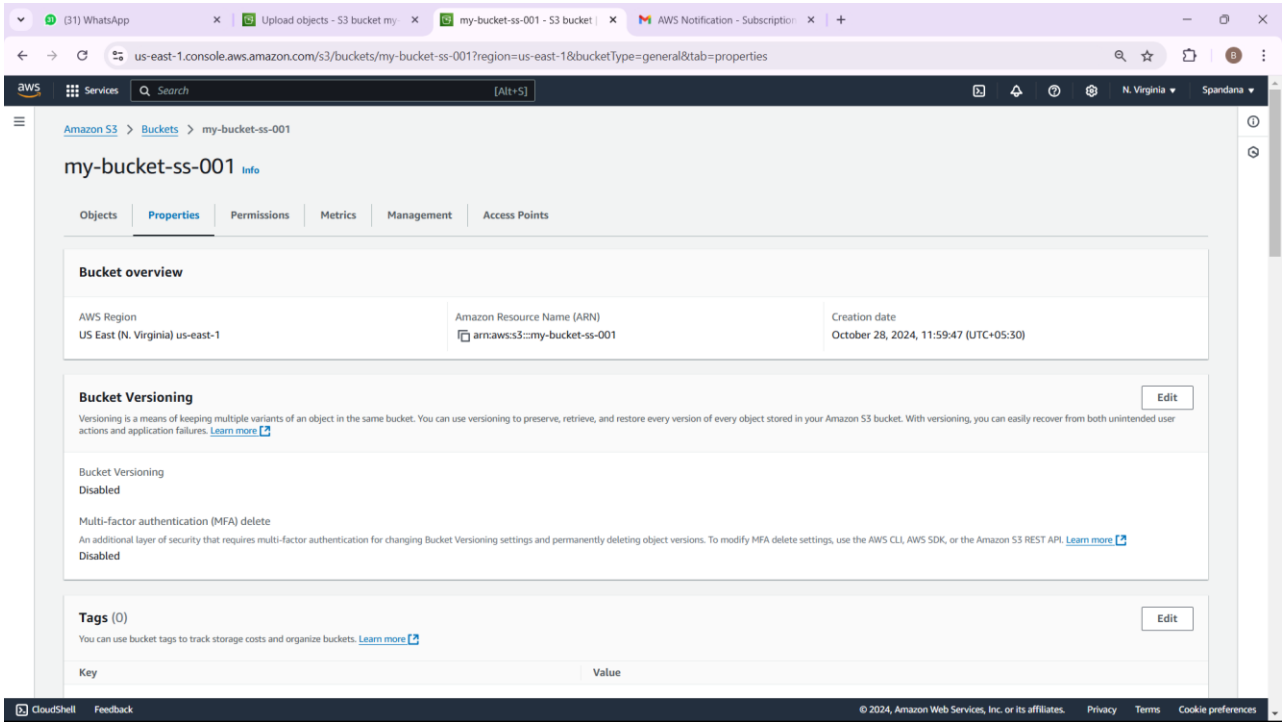
Access the file with browser we can able to see it like this because images are shown like this I have taken an certificate image for the process.

STEP 2

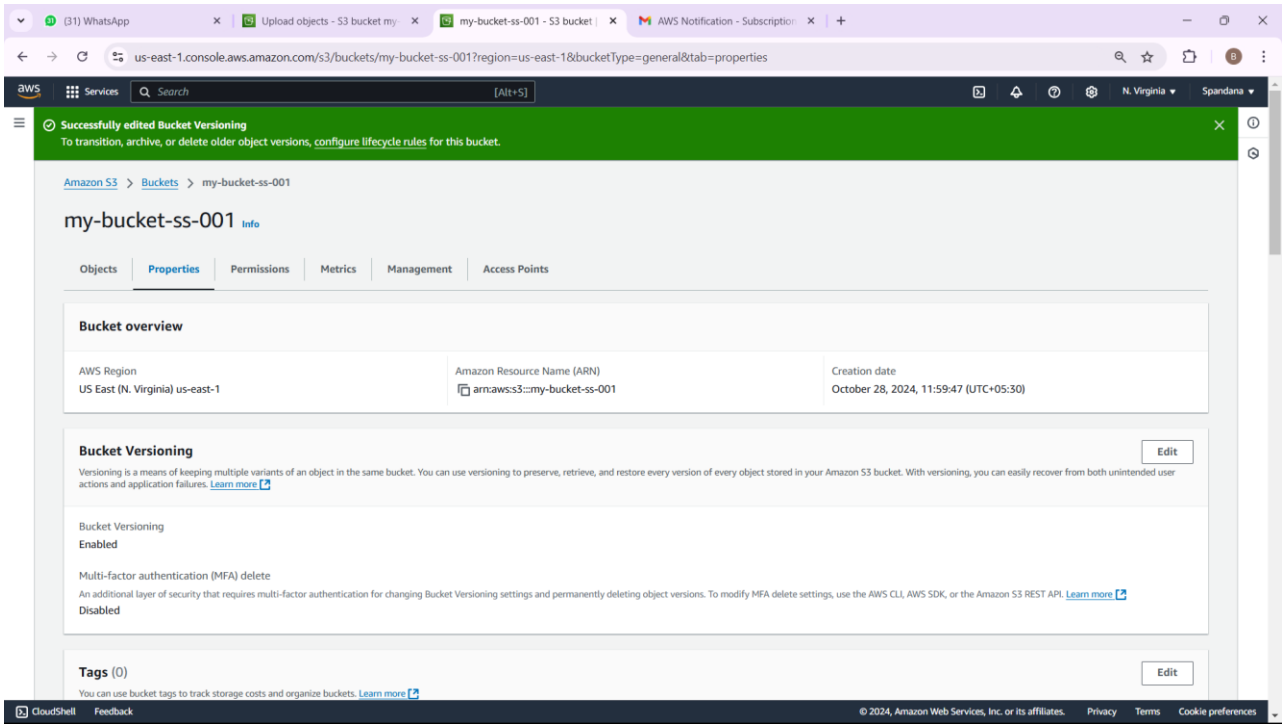
Go to the bucket and go to the Properties tab of the bucket



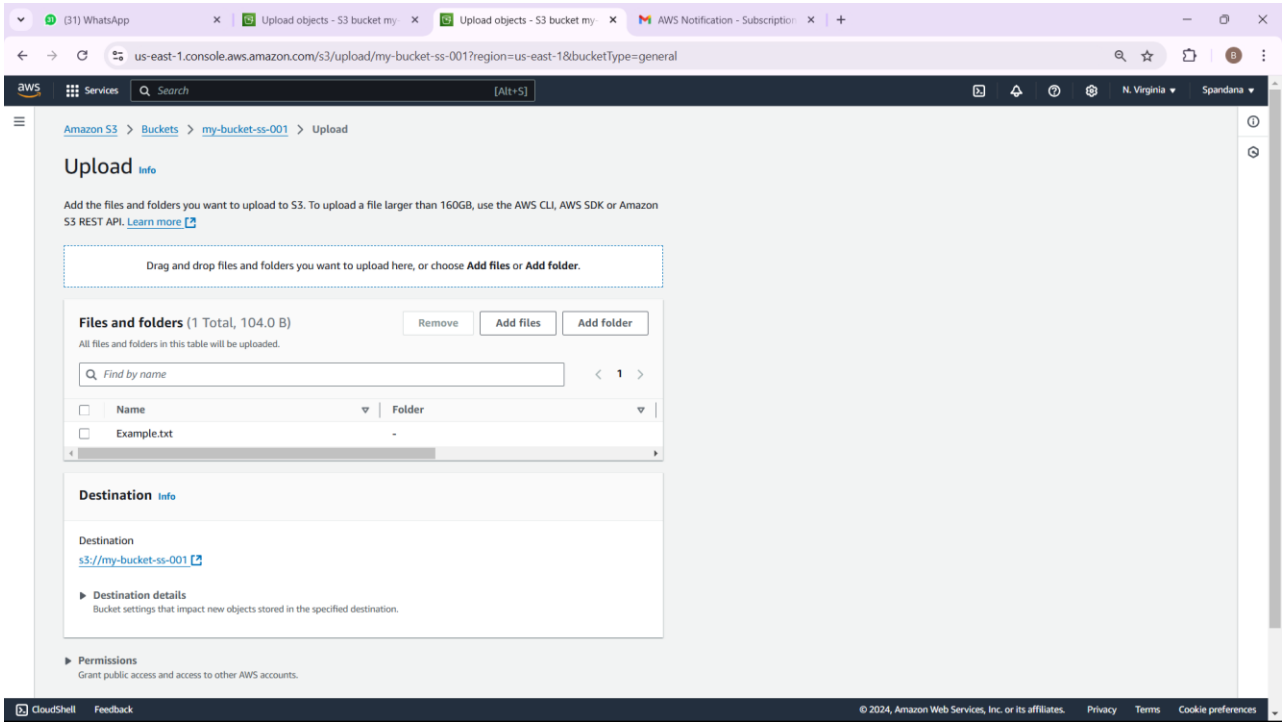
Scroll to the Bucket Versioning we will see Bucket versioning is Disable in the below image.



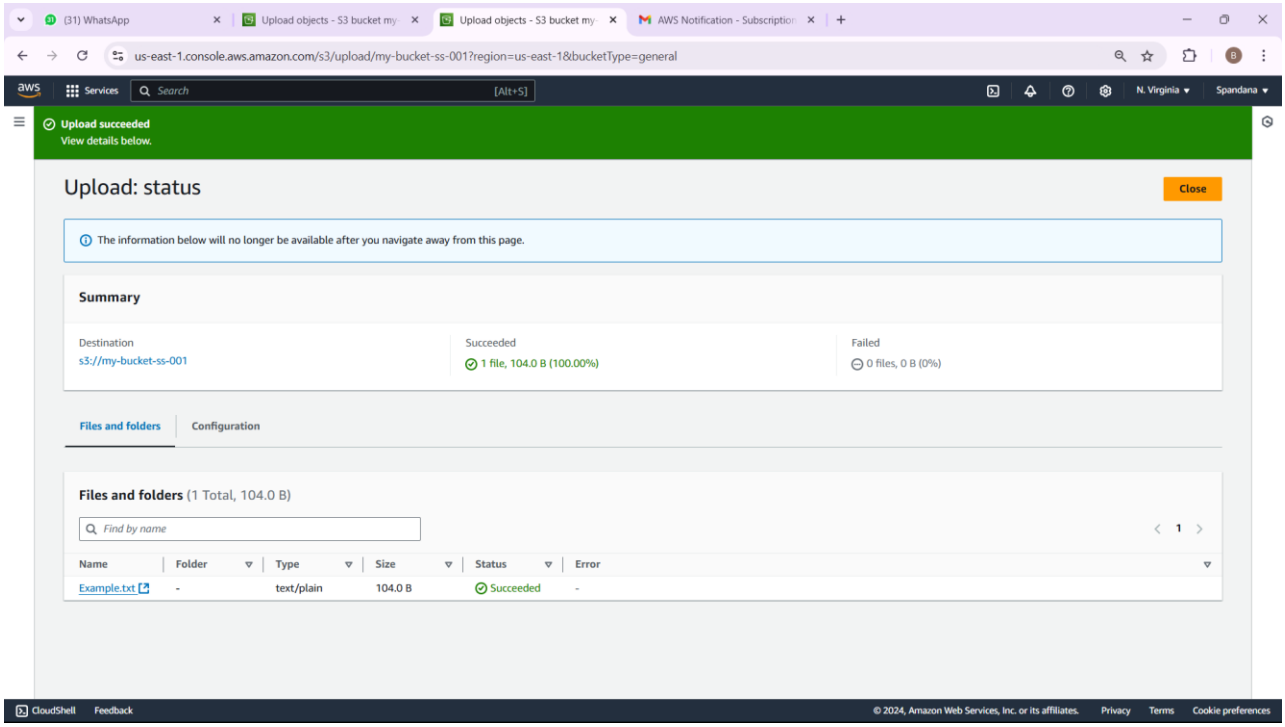
Click on Edit option and enable Bucket Versioning and then click on save changes.



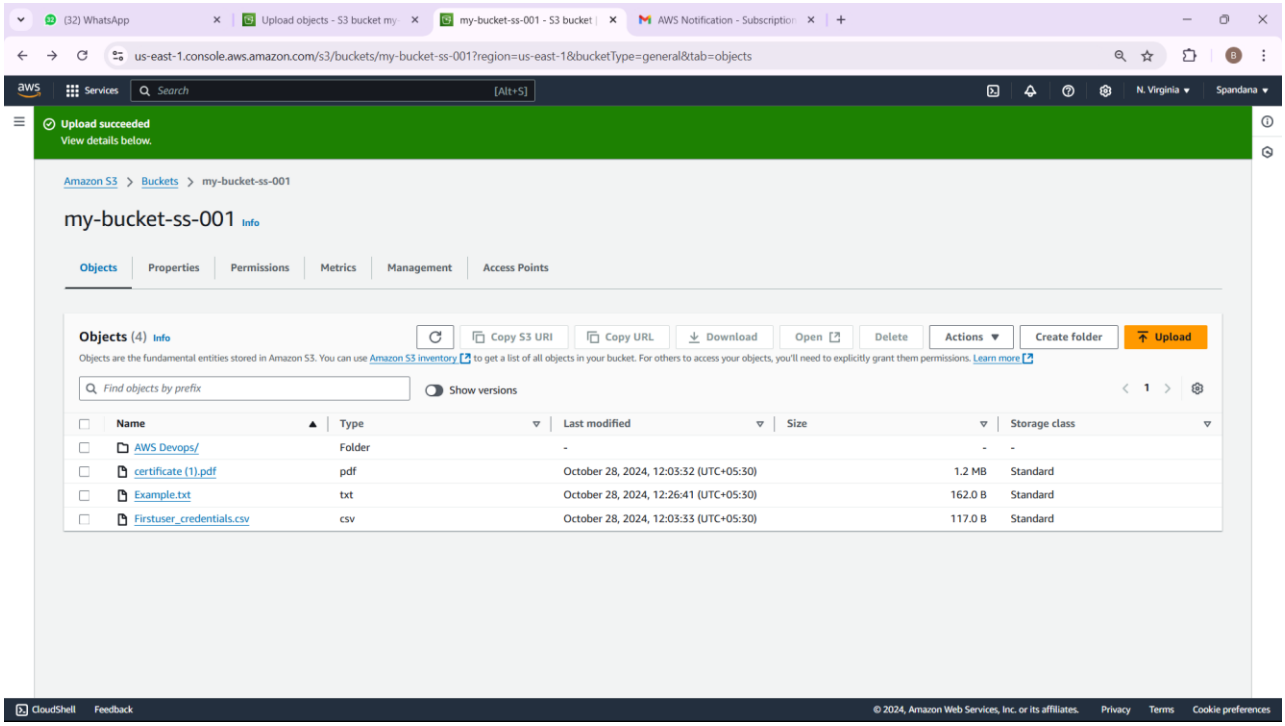
Create one text file on local machine (e.g., Example.txt) with some content. This is original version.



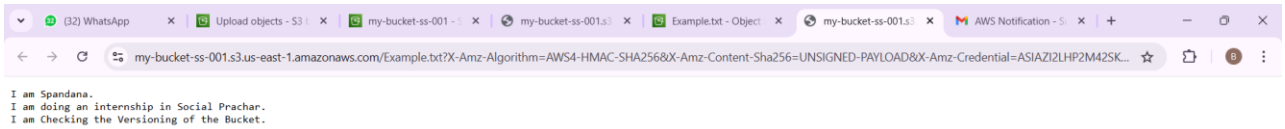
In the click on upload select add files and choose Example.txt Click upload to Confirm and we can able to see that our text file uploaded successfully.



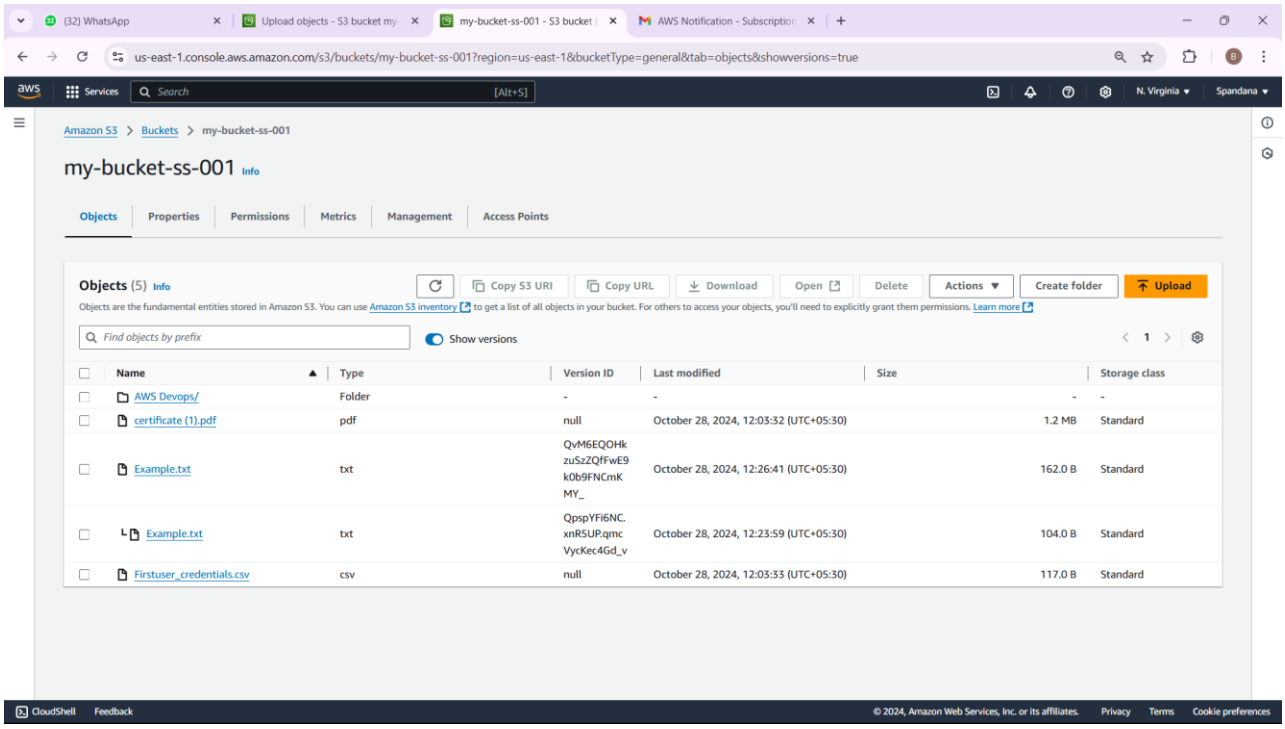
We can able to see Example.txt filein the S3 Bucket.



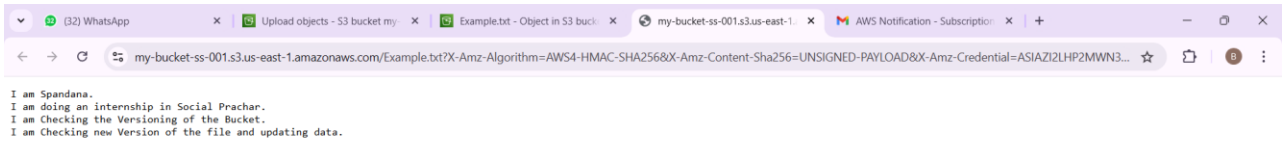
This is the Content of the text file before updating the file.



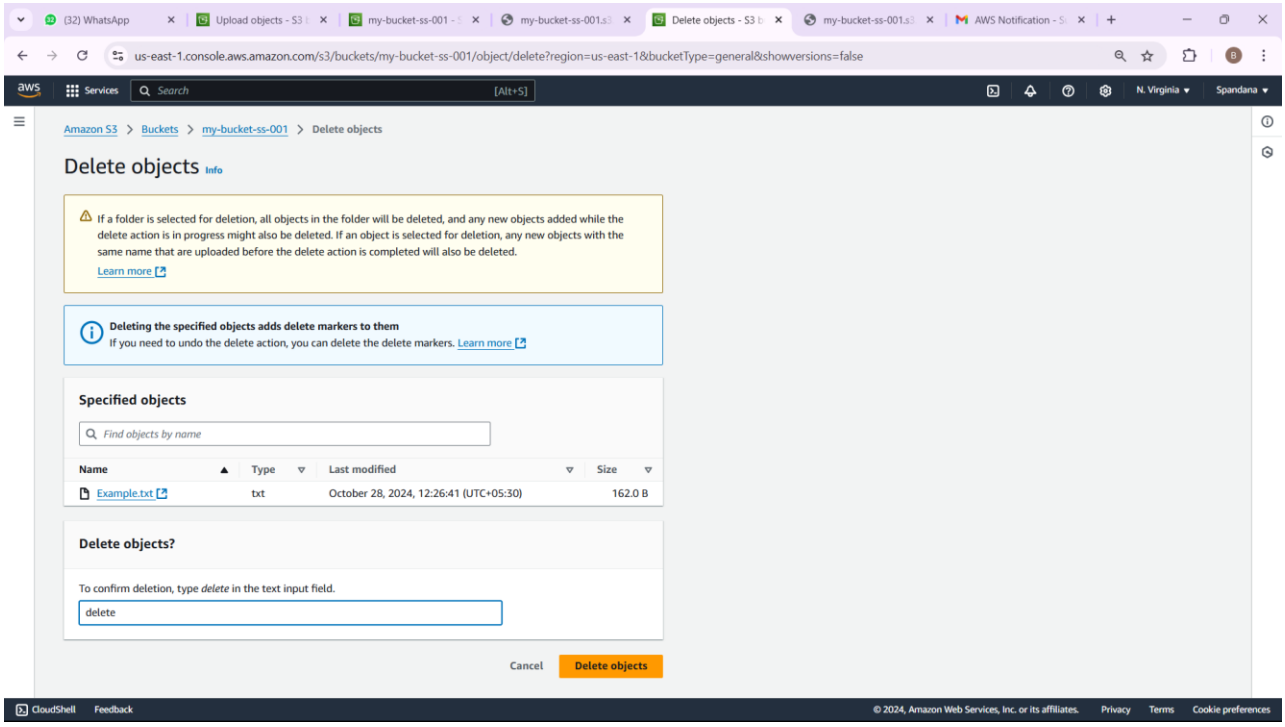
Update the Example.txt file on local machine and modify its content (This is updated version). Save the changes and upload the file and confirm AWS S3 automatically treat this as a new version of the file. We can able to see that there are 2 Example.txt files in S3 Bucket.



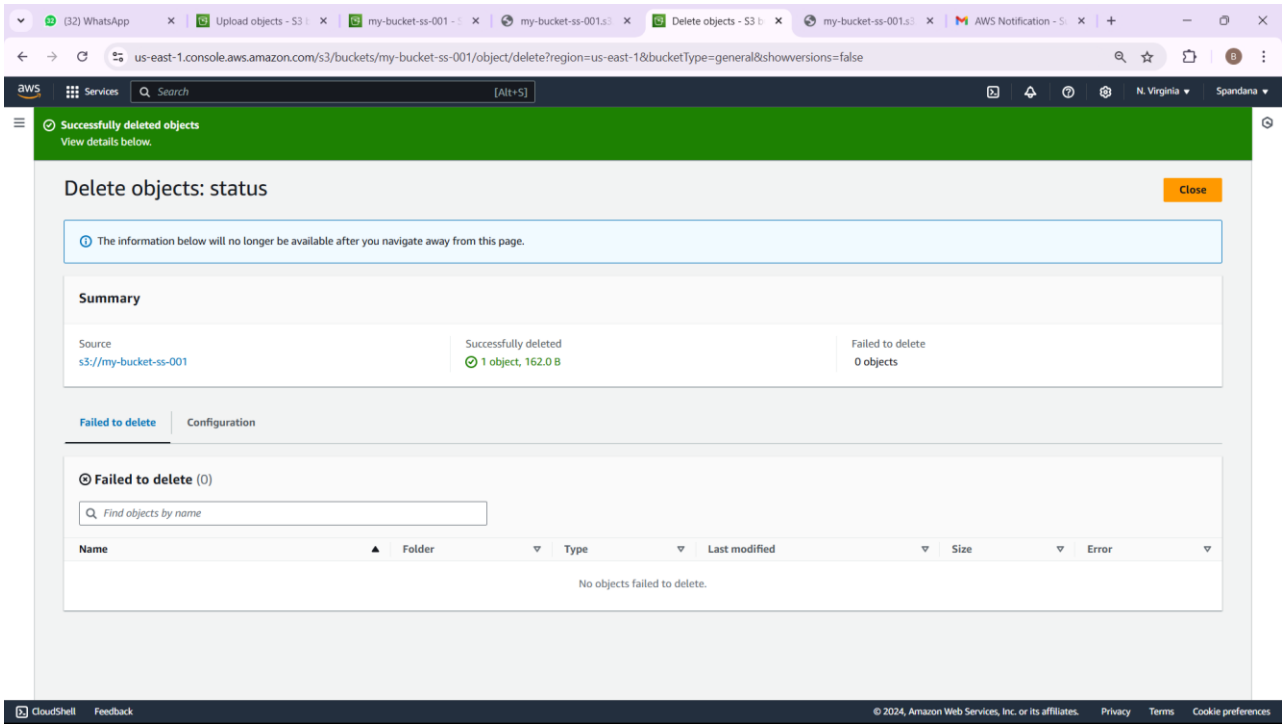
This is the Updated Version of Example.txt file I have updated last line.



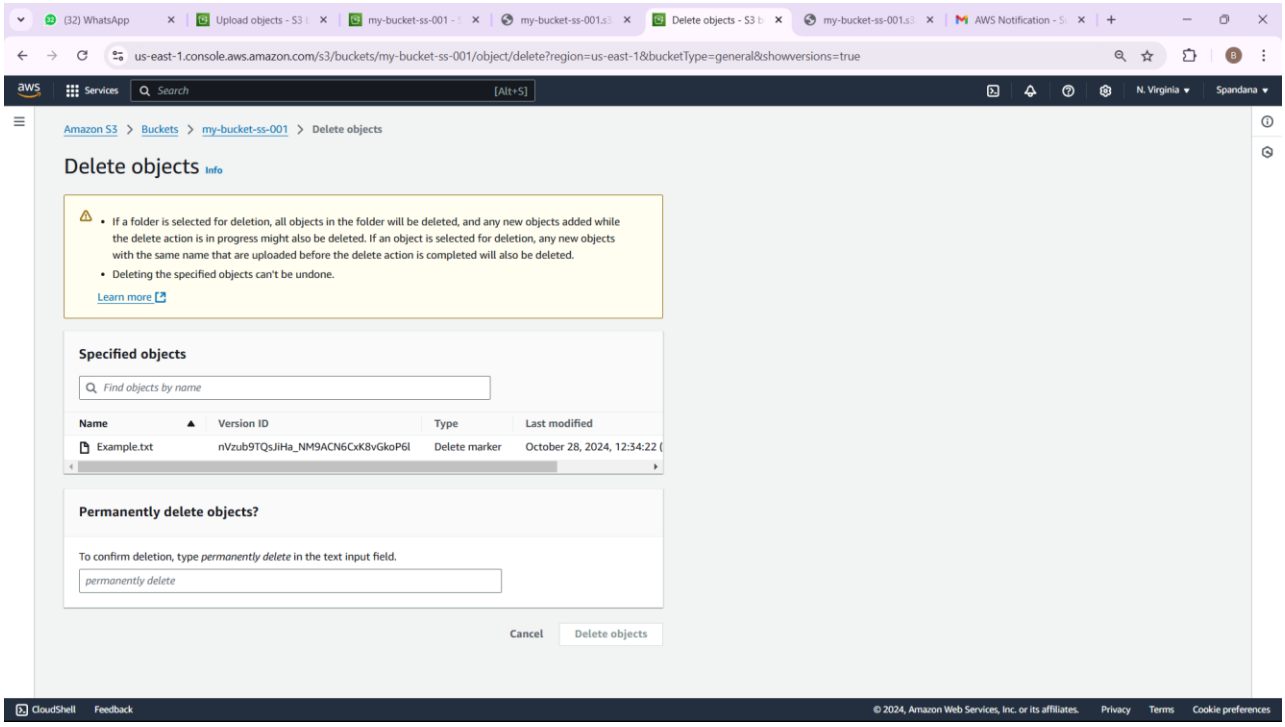
In the Bucket locate Example.txt select it and click on Delete. Confirm the deletion.



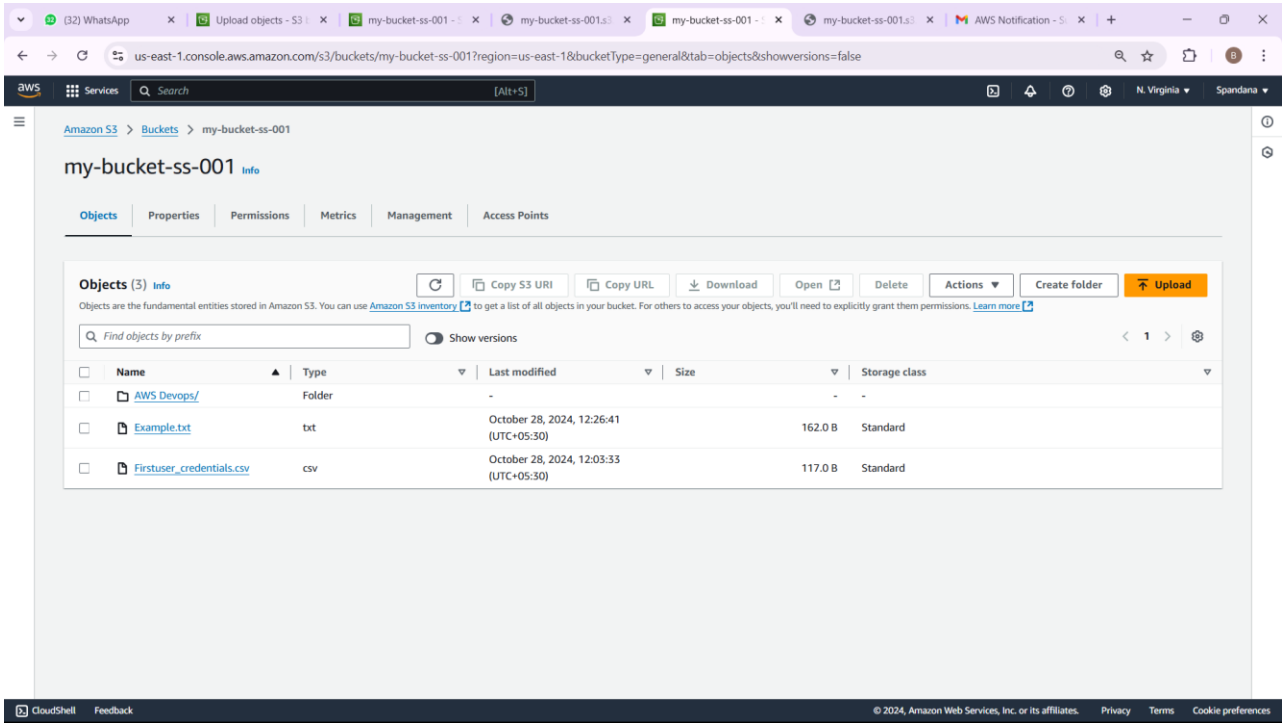
We can able to see that the object is deleted successfully.



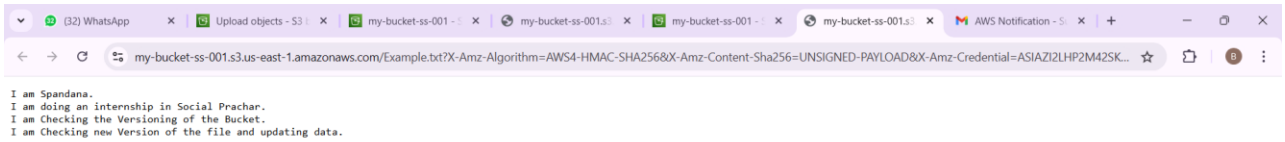
Delete the Example.txt file permanently so that we can able to see the versioning properly.



Go back to the versions of the objects tab. We will find the previous versions, including the one marked as “deleted marked”. Restore the deleted version to recover the file delete the delete marker version of Example.txt once we remove the delete marker the file will be restored to its previous version.

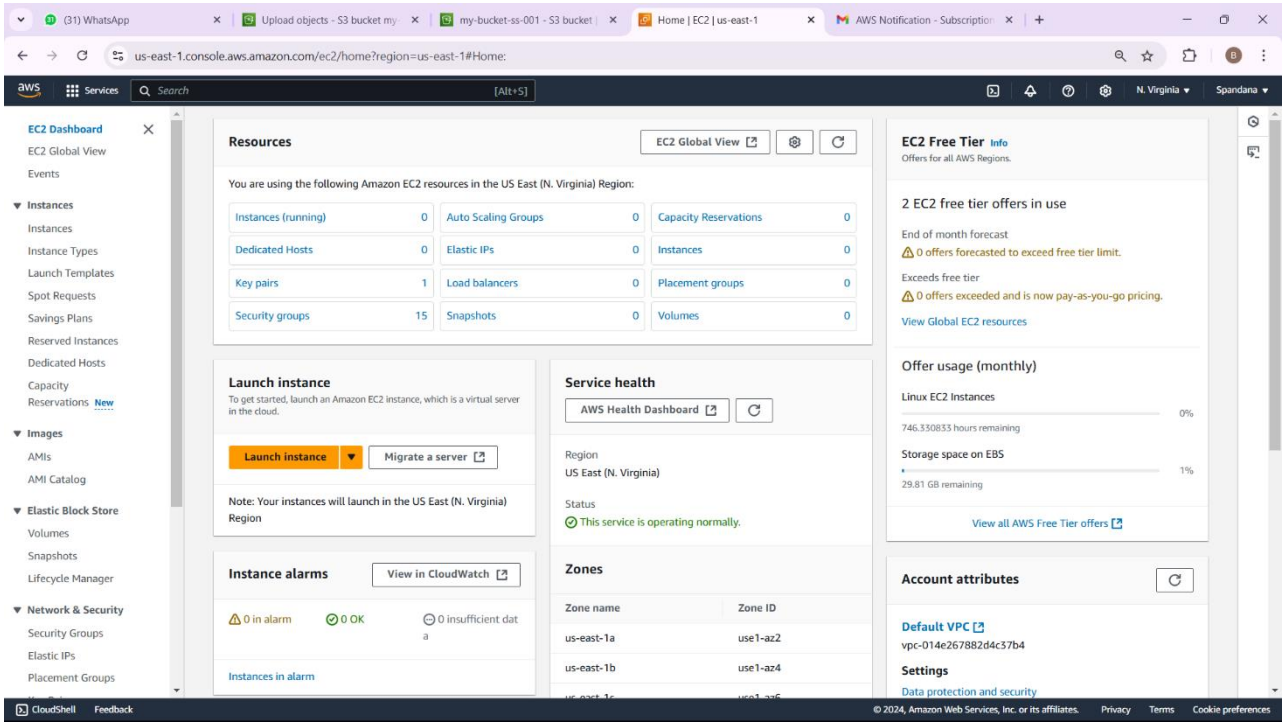


We can able to see the restore of the file ad content of the file.

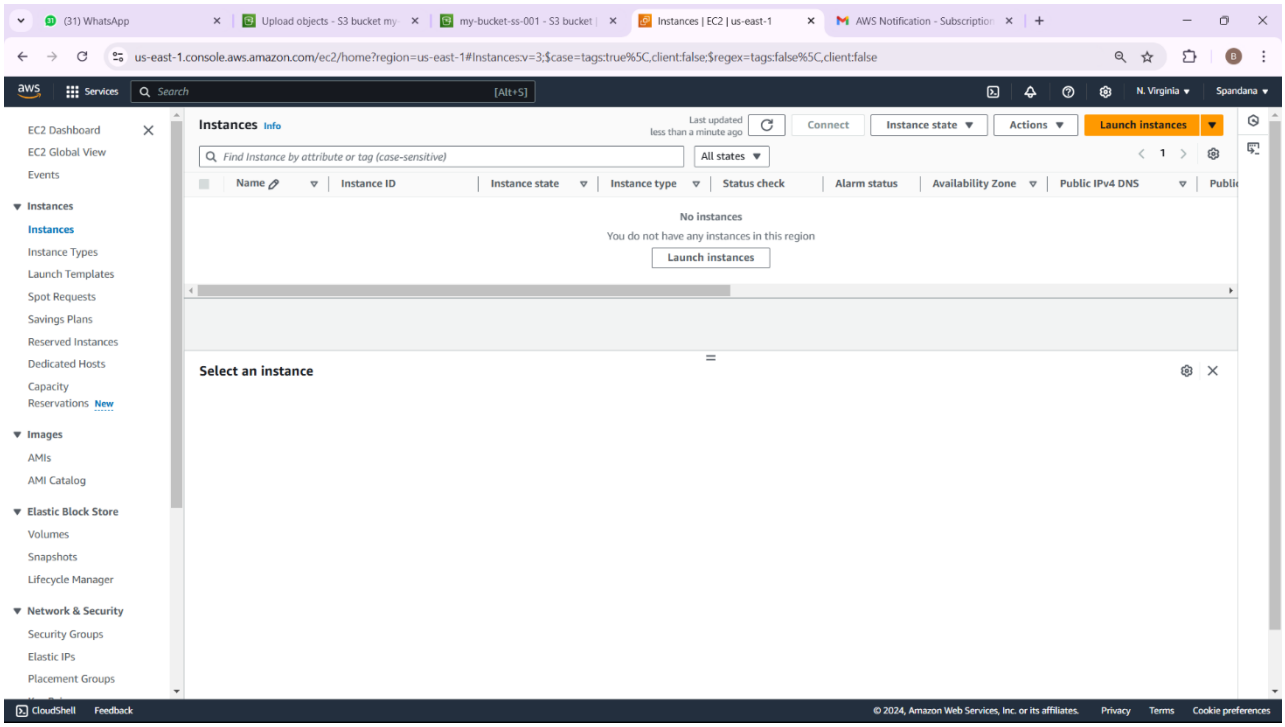


LAB 4 - EC2-INSTANCE

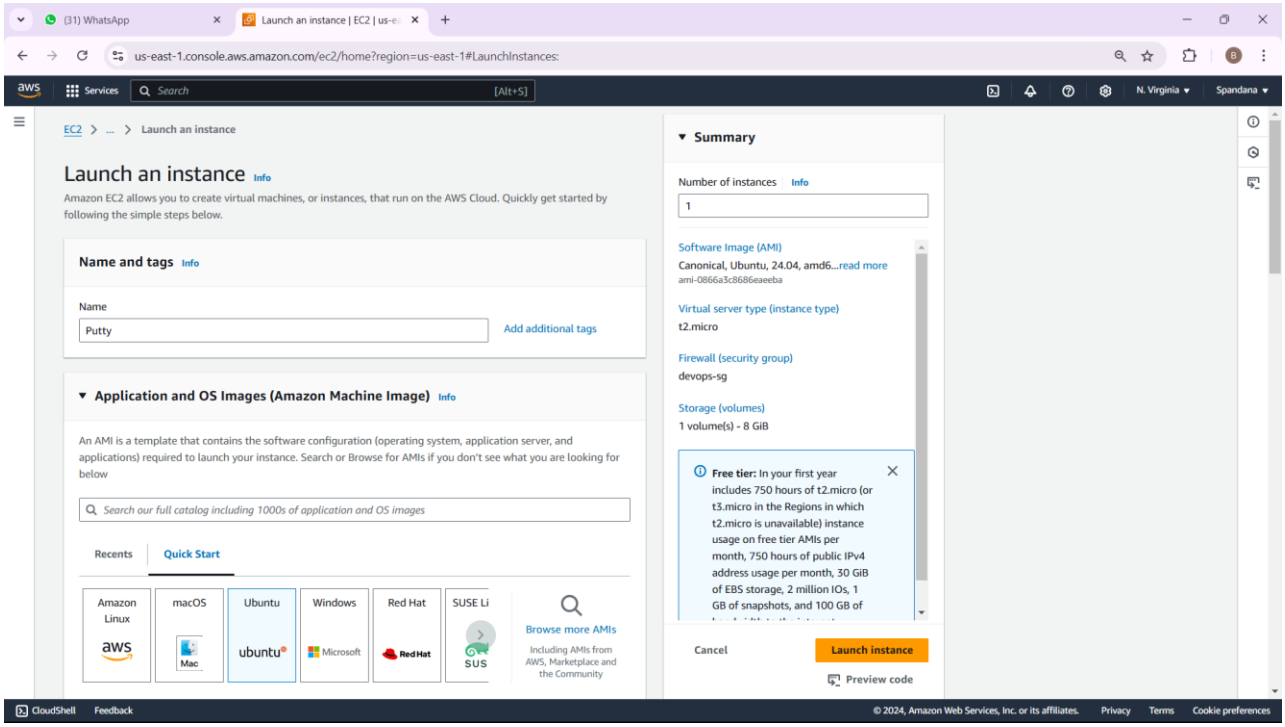
In AWS console search for EC2 and open it.



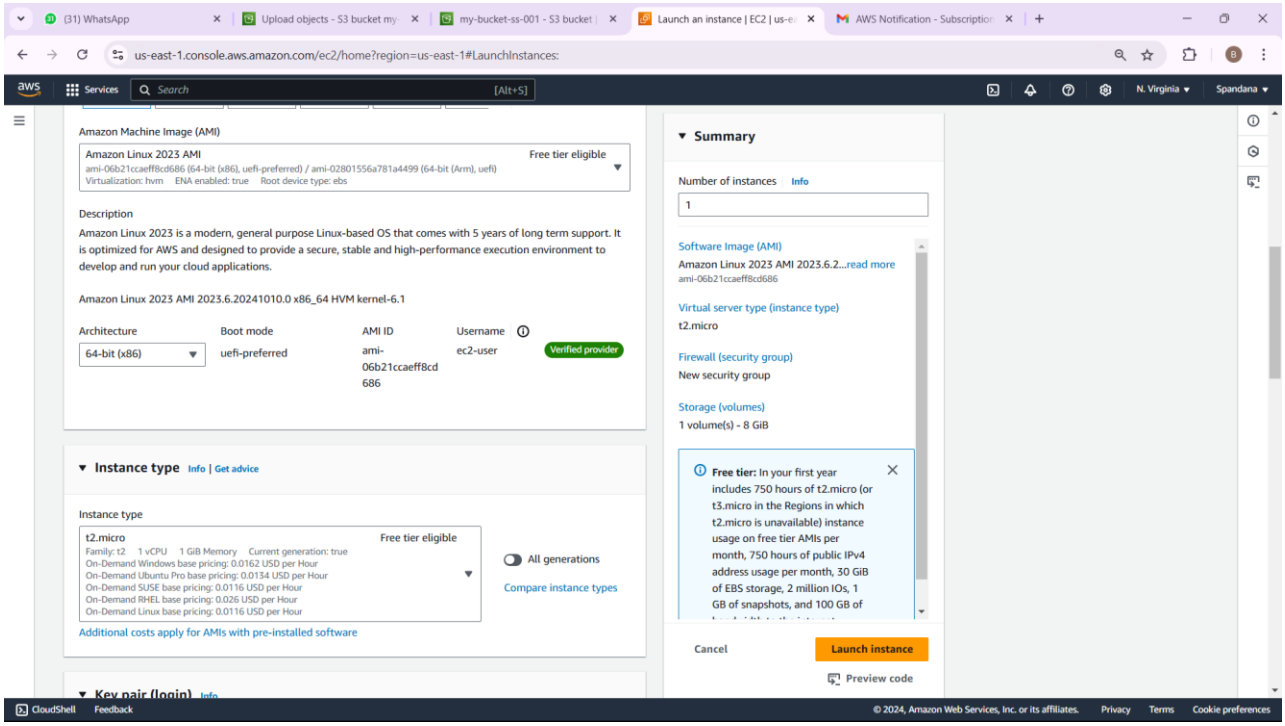
Click on Launch instance that we are seeing in the below image at the right top corner.



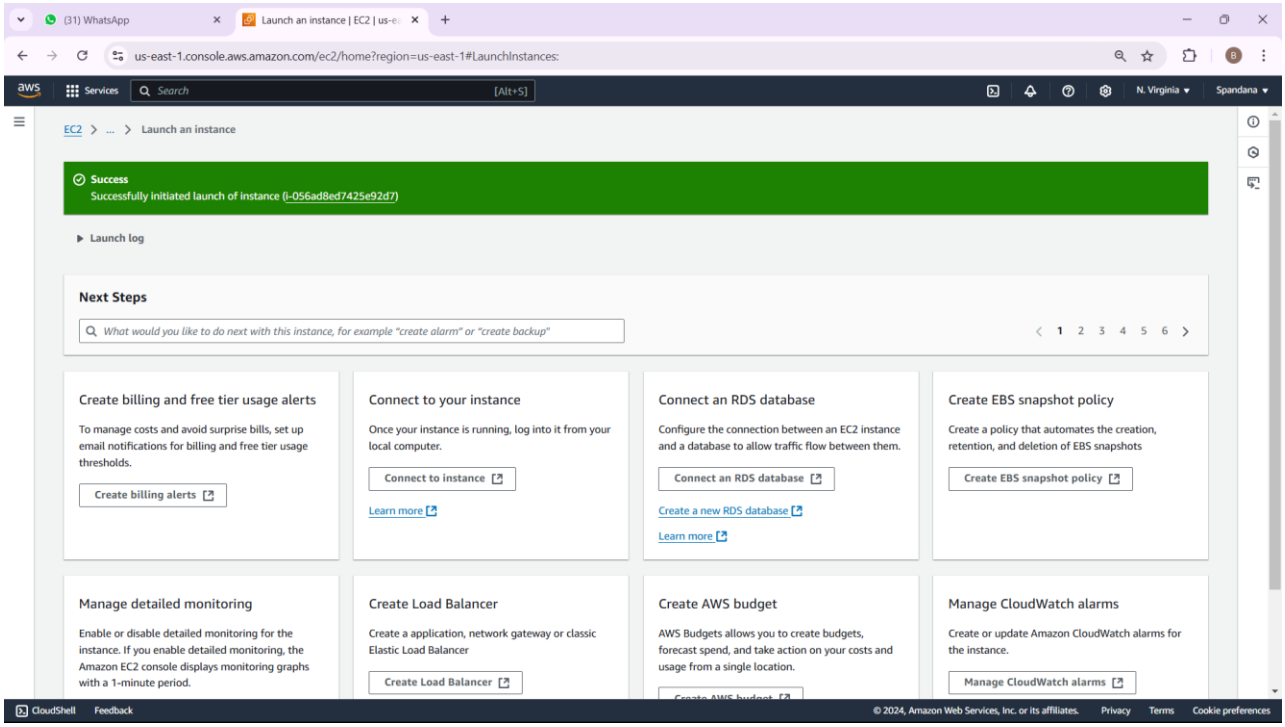
When we click on launch instance new page will be appear that we are seeing in the below image. It is nothing but the configuration of instances firstly we have to give name for the instance and select AMI based on our choice. I have selected ubuntu server AMI.



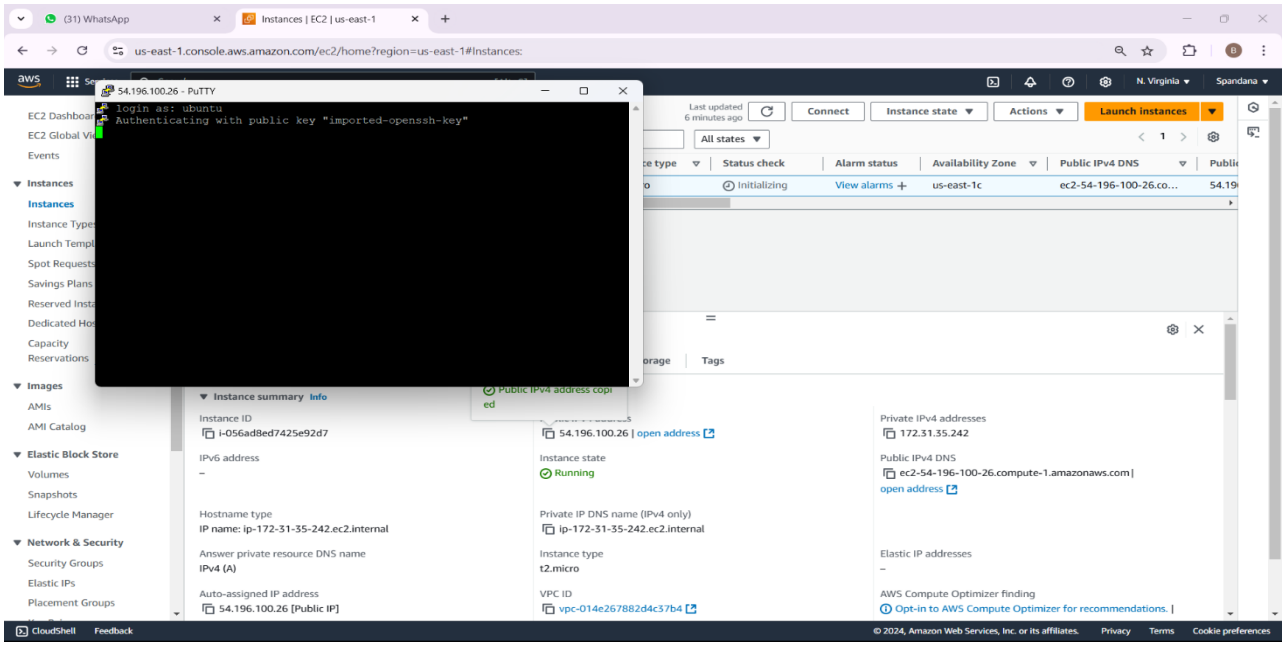
Under key pair select an existing key pair or create new key pair we have to select instance type. I have selected t2.micro which is under free tier eligible.



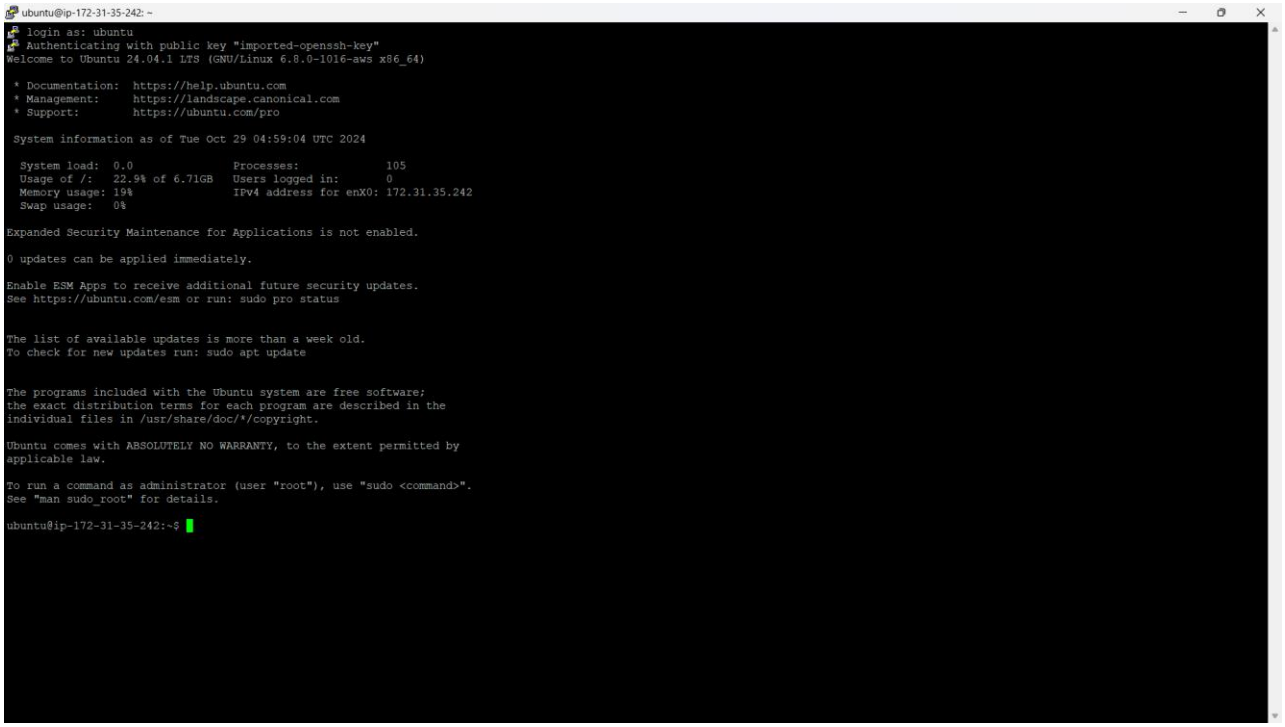
After configuration check click on create instance. We will see a notification that Success so that our instance has been created successfully.



Download PuTTYgen open the PuTTYgen click load and select .pem file once loaded click save private key to save it in .ppk format This file will be used for authentication in PuTTY. Copy the pubic IPv4 address open PuTTY in the host name filed enter public ip and go to SSH >Auth >connection and browse to select our .ppk file. Click on browse to connection and accept the security alert to connect.

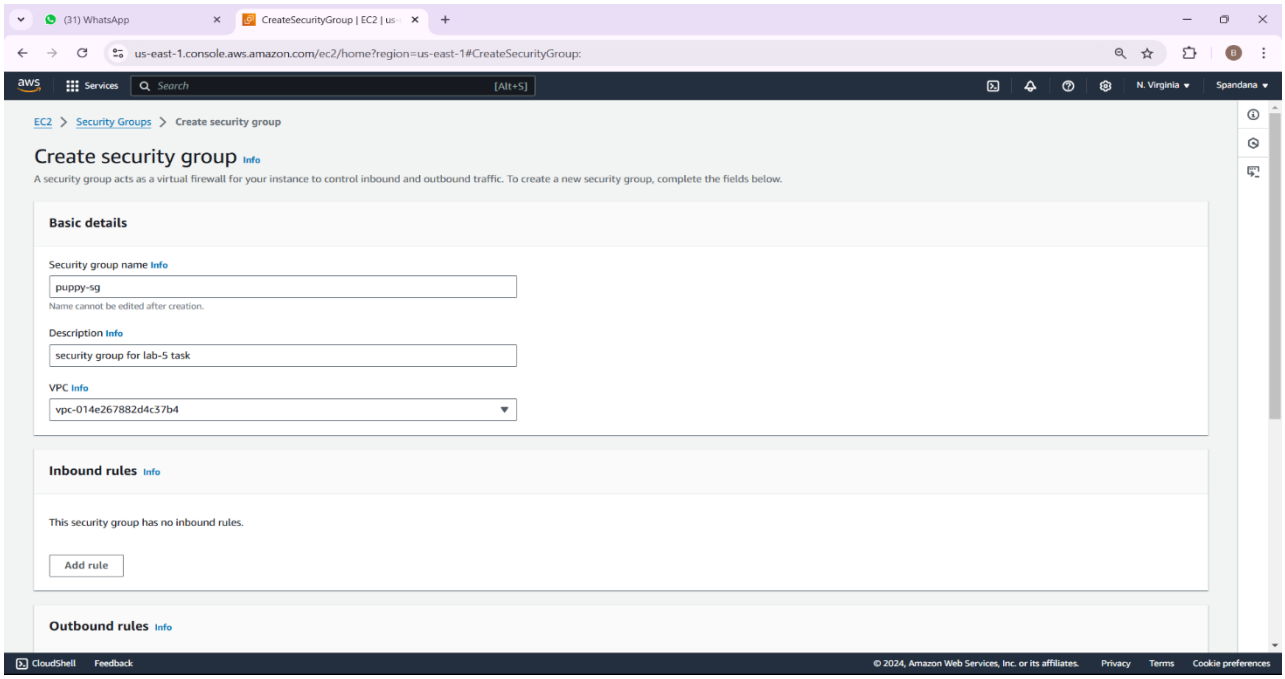


Once the connection is done we have to go to terminal prompt and we have to login as ubuntu and now run commands on your ubuntu instance.

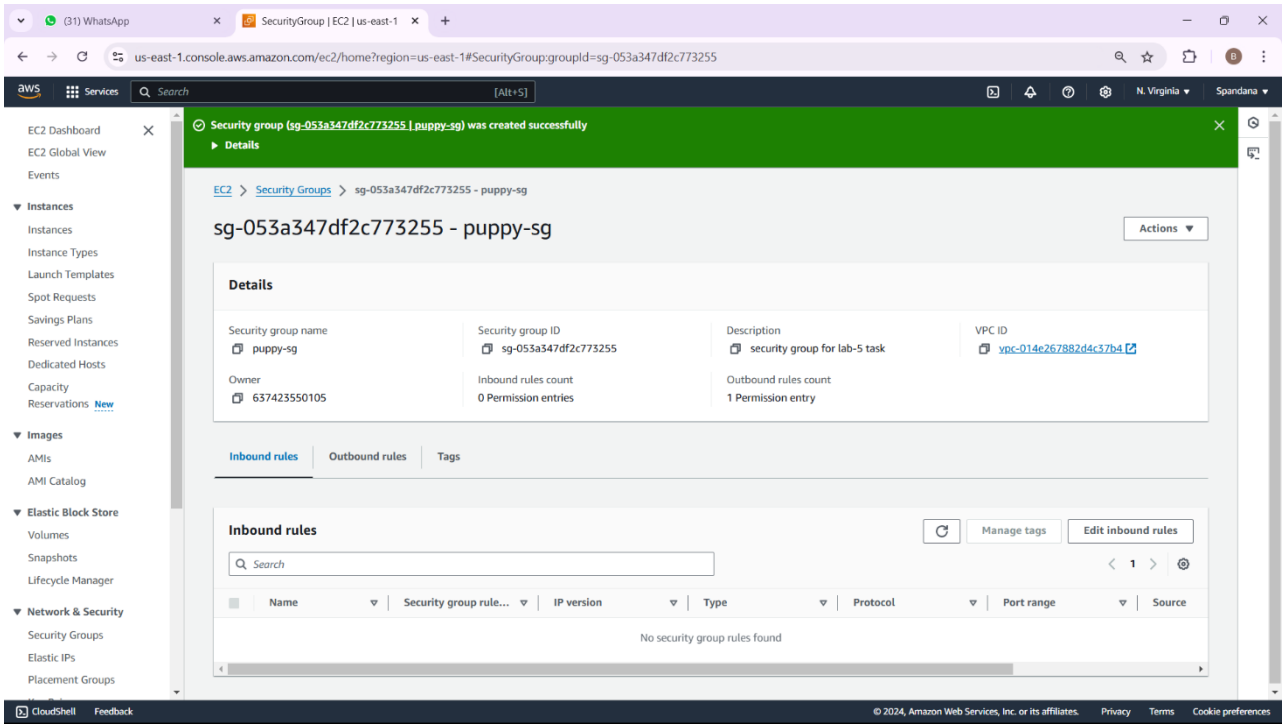


LAB 6 – VOLUMES AND SNAPSHORTS

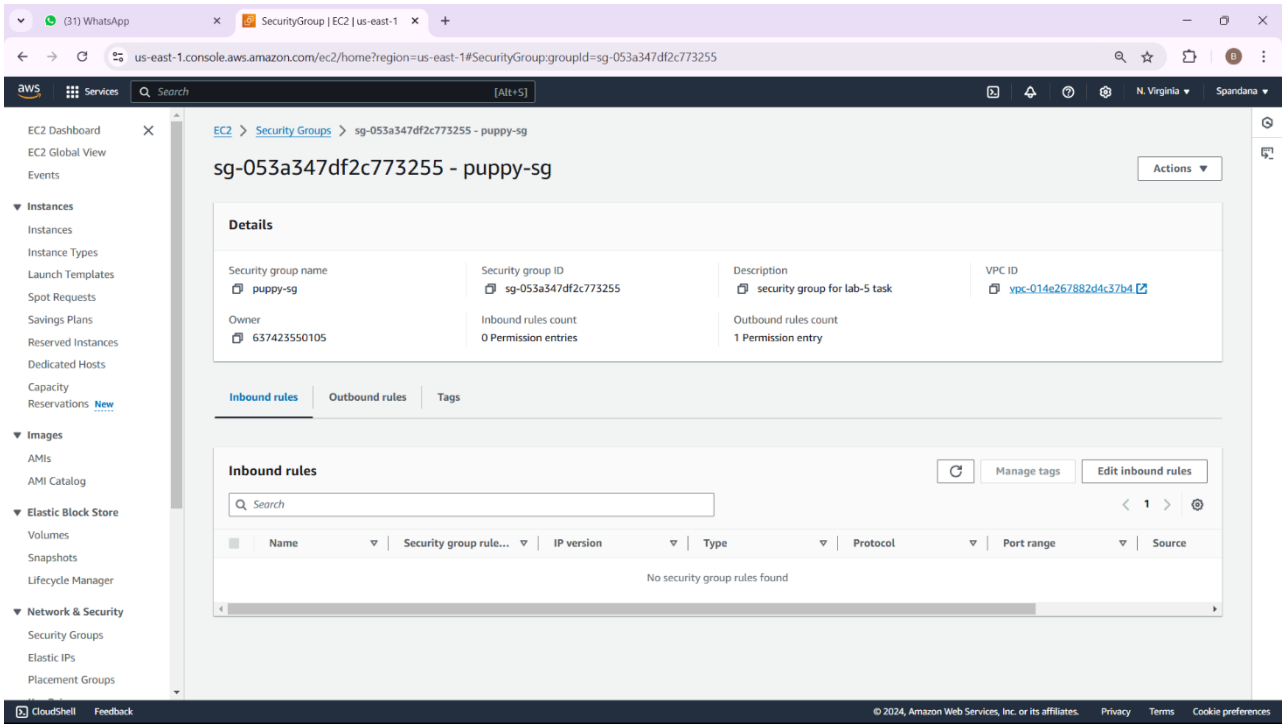
In the left hand menu, under Network & Security , select Security groups click on create security group name security group. Provide a description select your VPC where our EC2 instance.

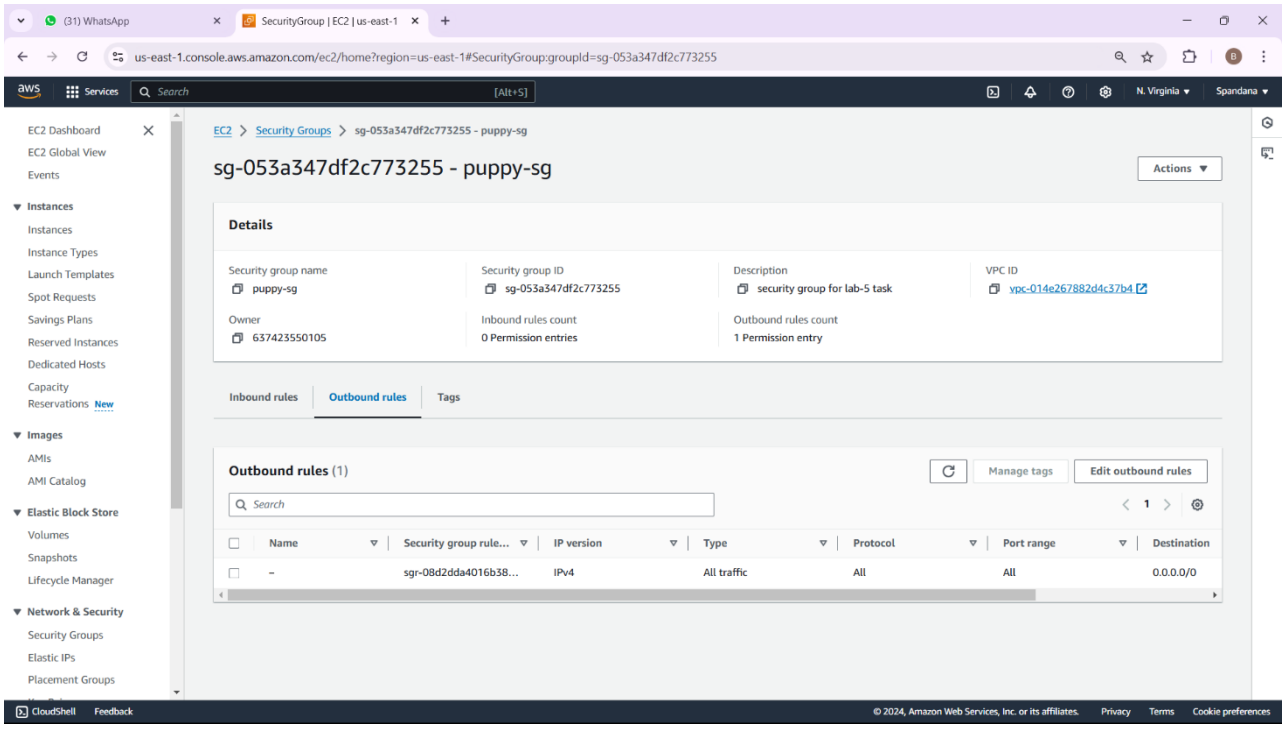


In the below image we can able to see our security group has been created successfully.

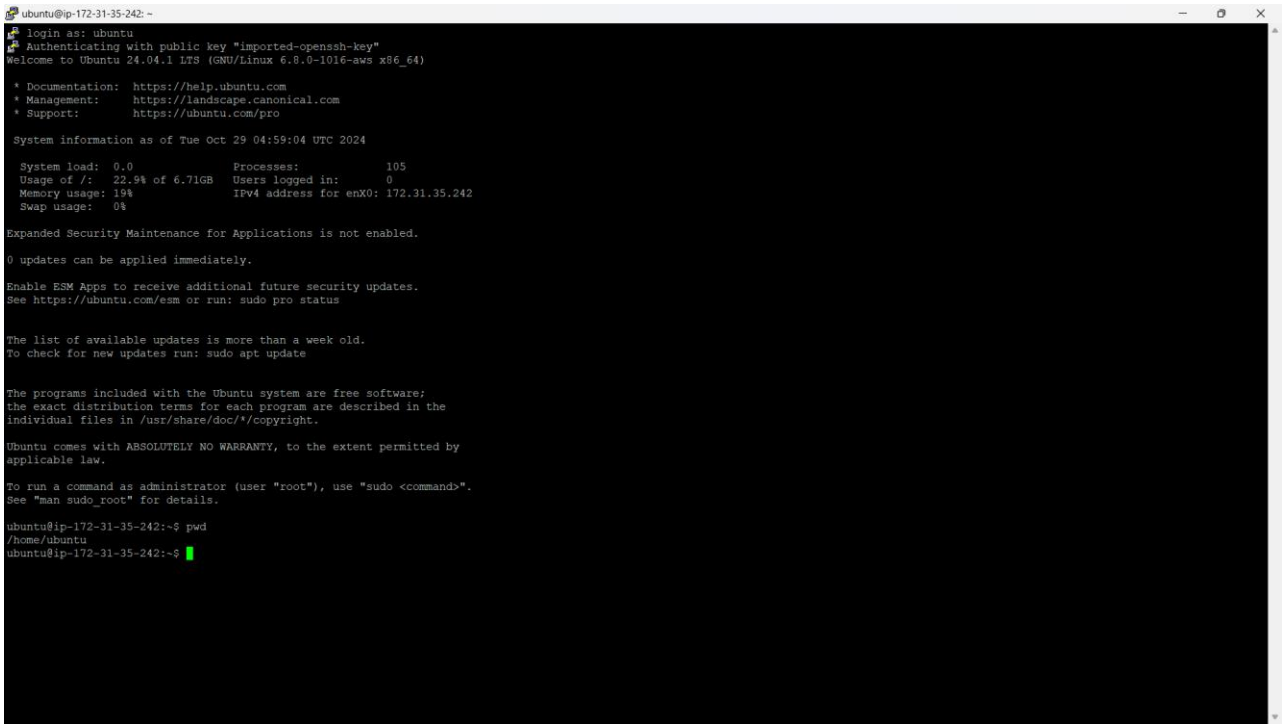


To check default rules in a new security group there are generally no inbound rules in the above image meaning all inbound traffic is denied by default. Outbound traffic typically allow traffic by default.

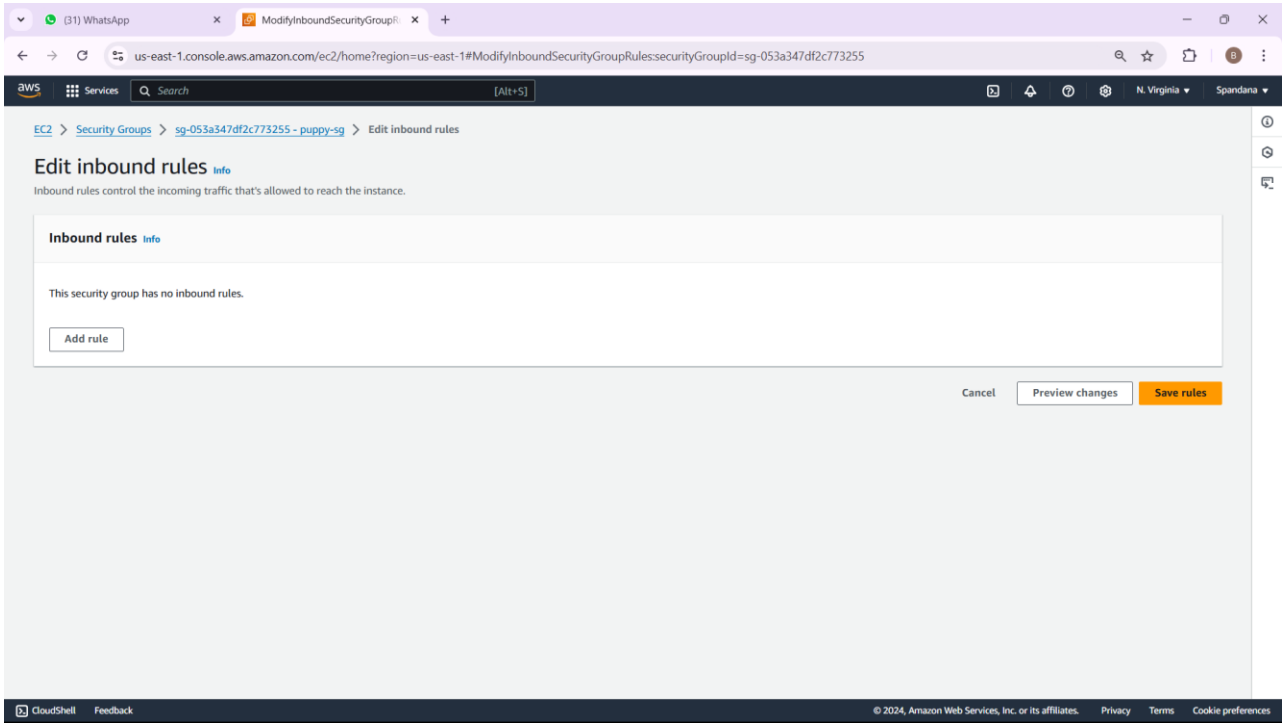




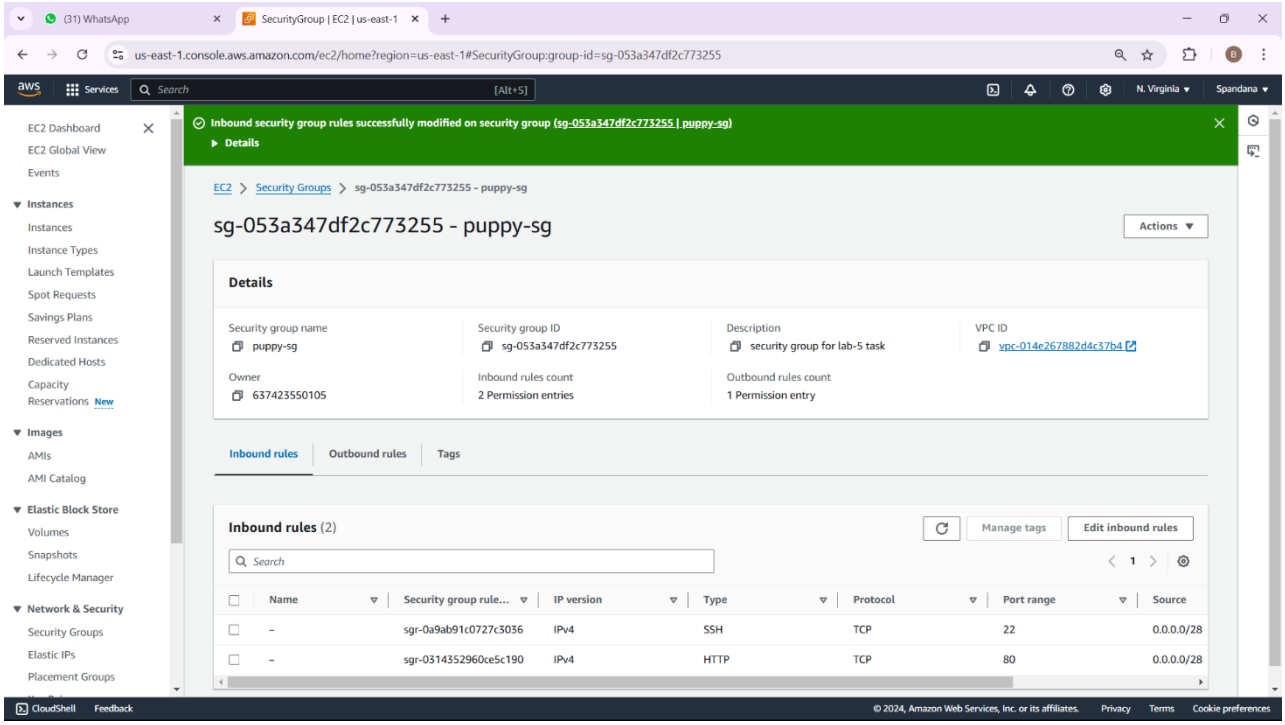
Check on command prompt we can able to run commands successfully in the below image.



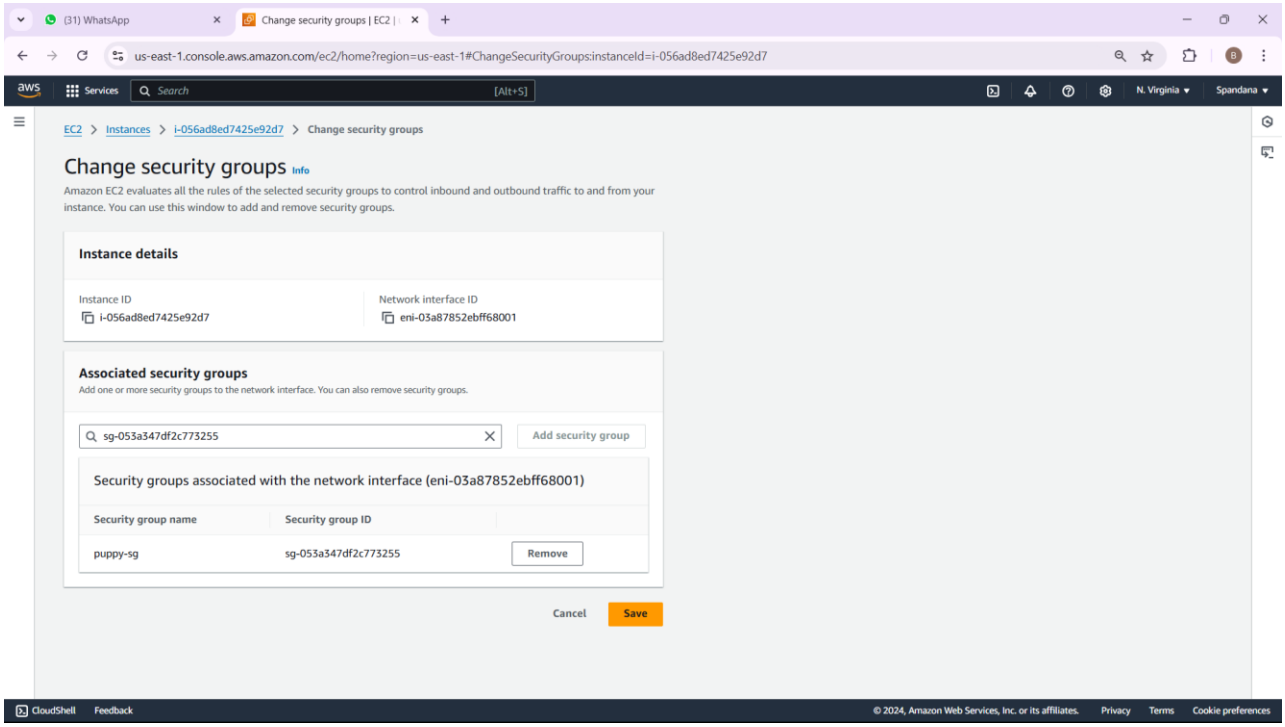
Allowing inbound rules for Ports 80 and 22 in the inbound rules section click on Add rule in that we have to select Type: SSH, Protocol: TCP, Port range: 80, Type: SSH, Protocol: TCP, Port range: 22 Source type my IP/28 click on save changes and click on create security group to finalize.

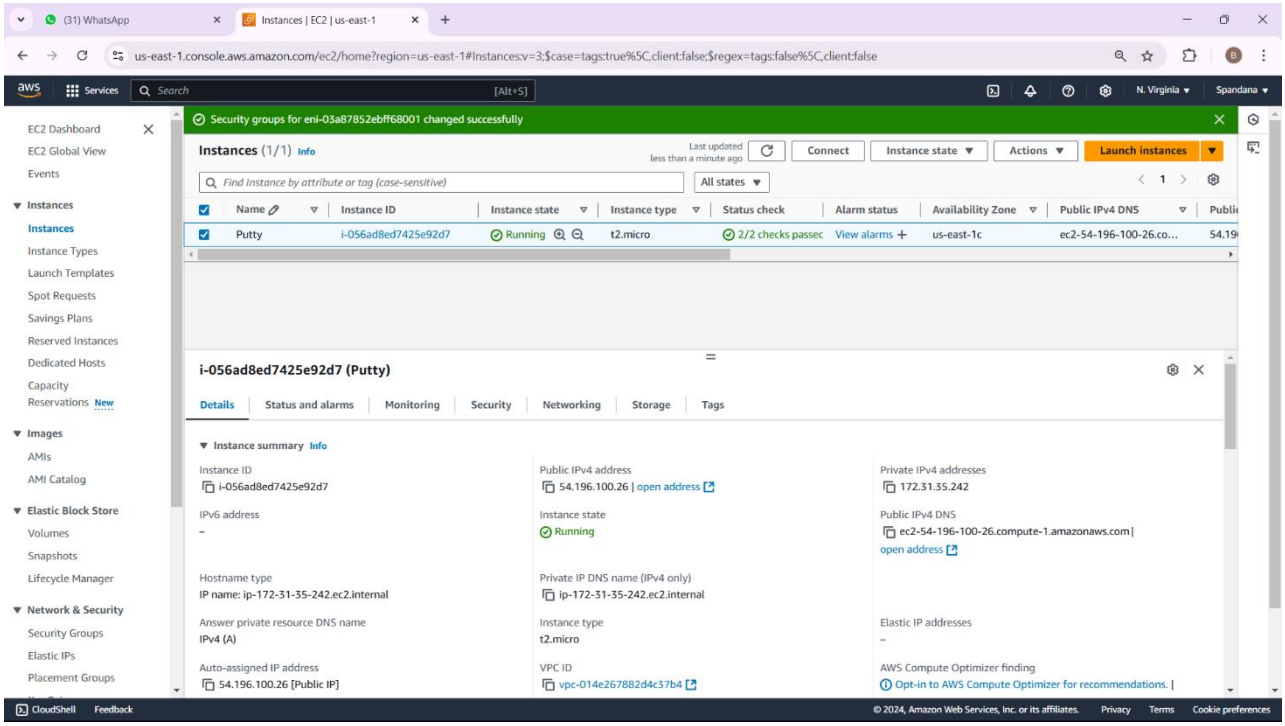


We can able to see in the below image that Inbound security group rules successfully modified on security group.

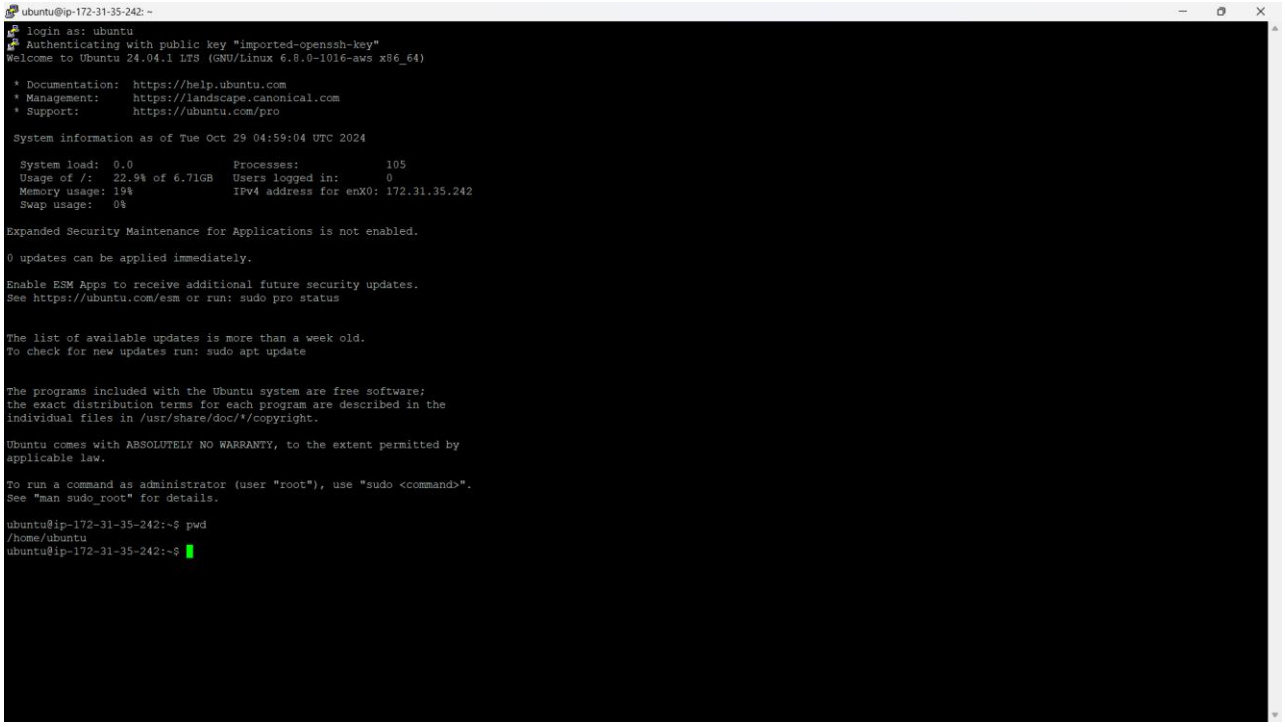


Navigate to instance click on the EC2 instance to which you want to attach this security group with the instance selected click Action > Security > this security group and select the new security group and remove any other security group if not needed after that update security group.



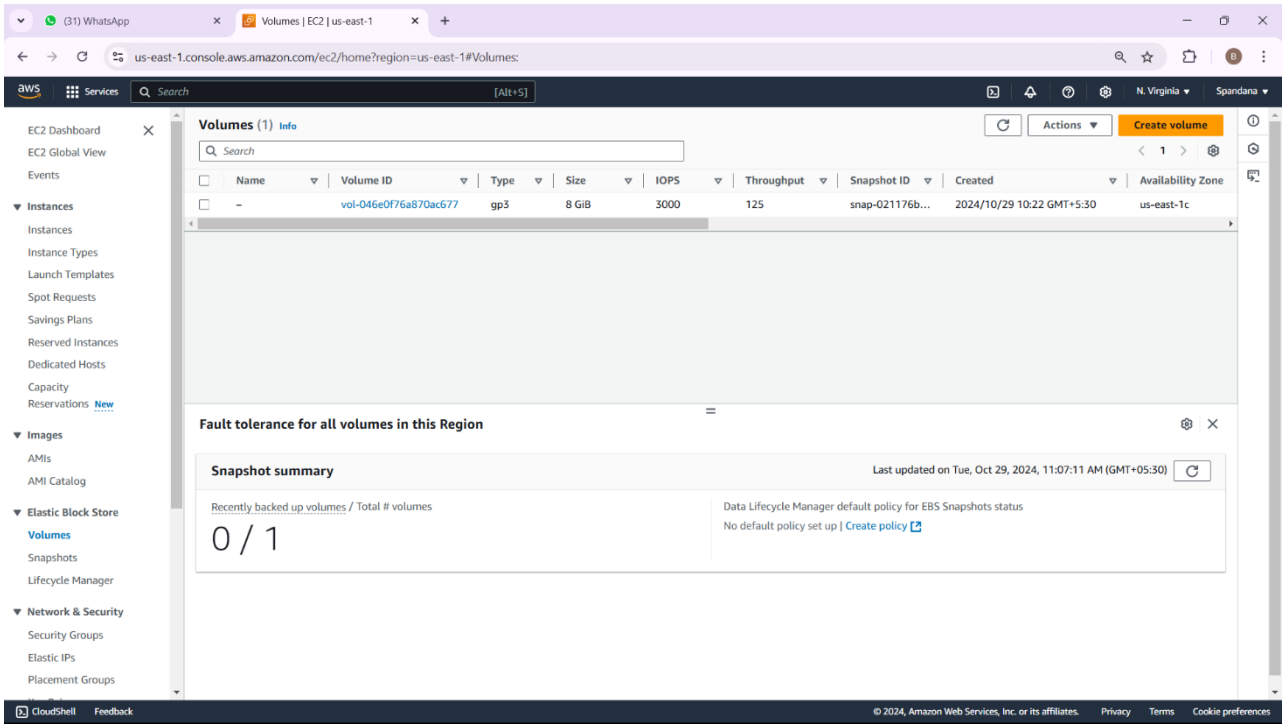


Open the PuTTY and attempt to connect to the instance and we cannot able to access.

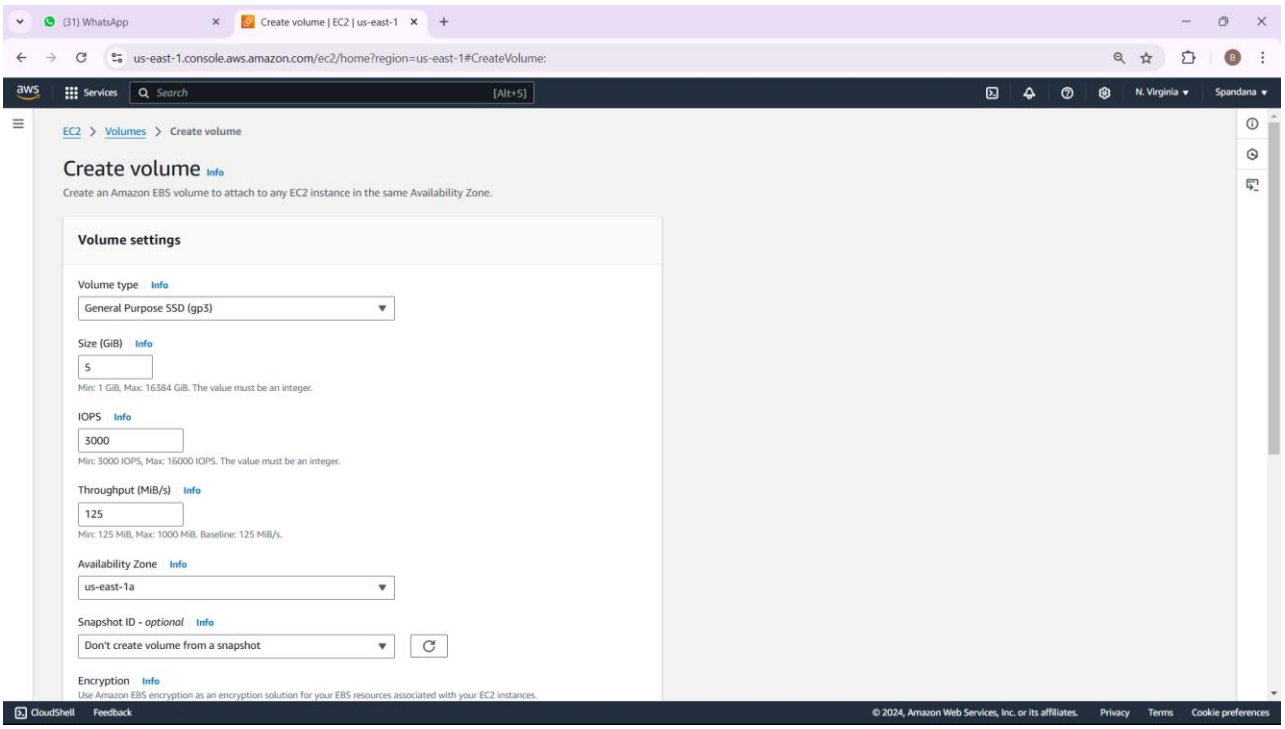


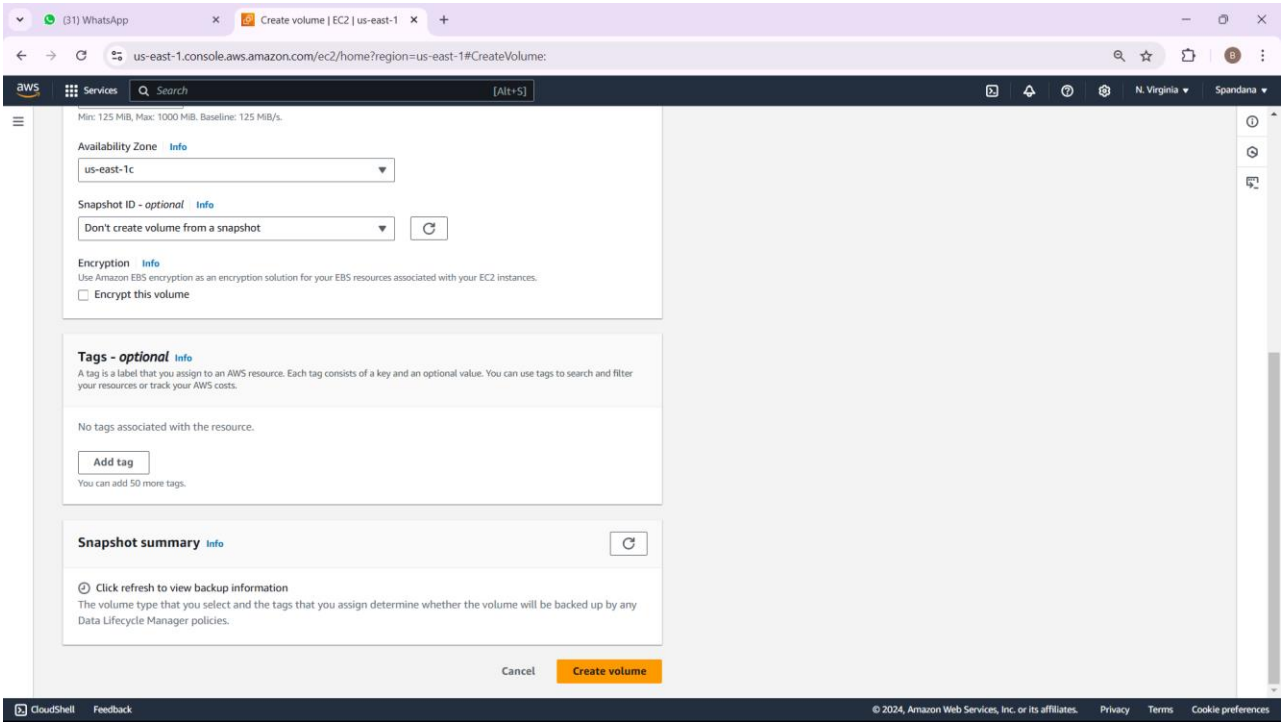
LAB 6 – VOLUMES AND SNAPSHORTS

Create a new volume in the left-hand menu under Elastic Block store select volumes and click on create volume.

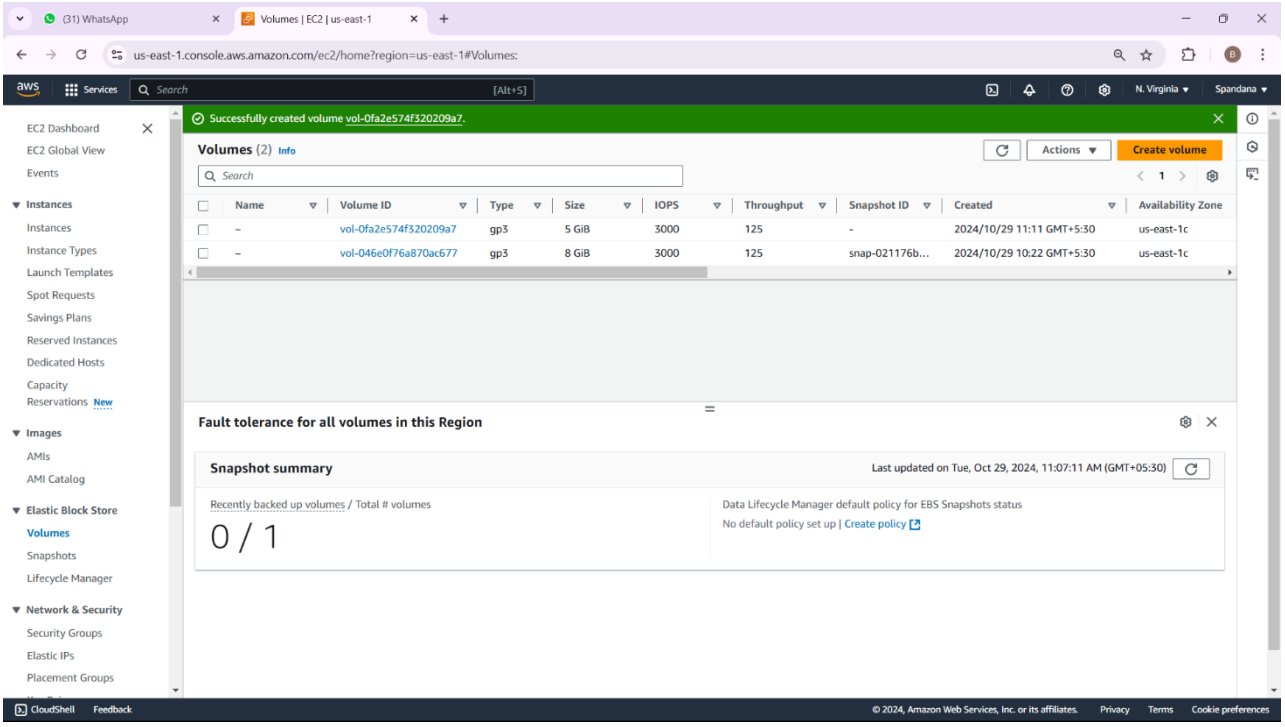


Configure volume settings size 5GiB Availability Zone select the same zone as your Ec2 instance and volume type General Purpose SSD. Click on the create volume to create EBS Volume.

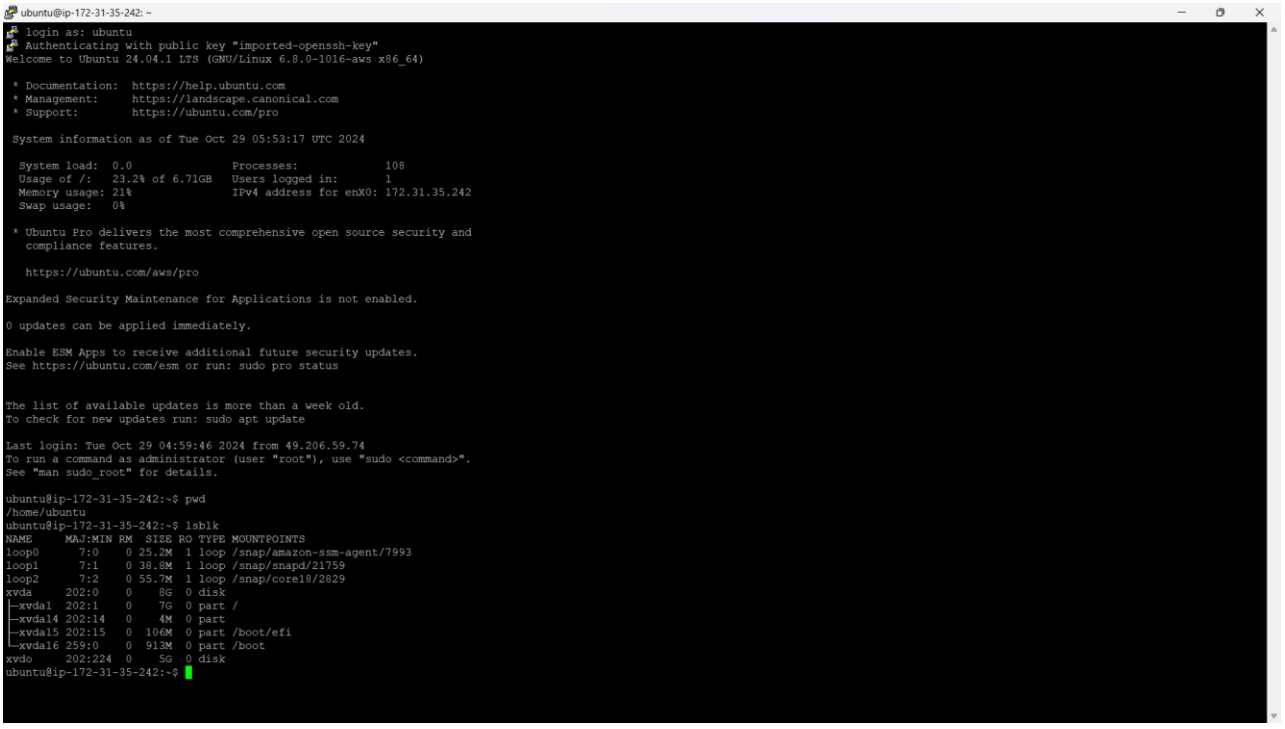




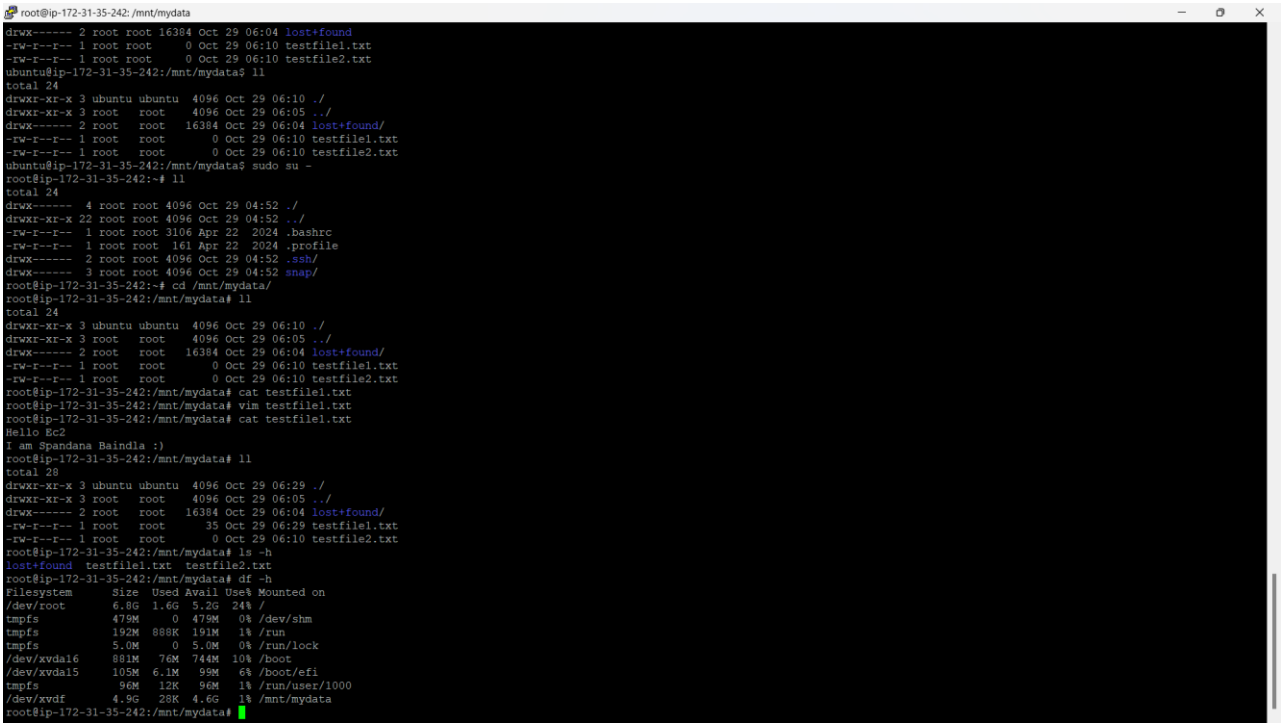
Volume has been created successfully we can able to see in the below image. Once the volume is created select it in the volumes list and click on Action > Attach volume in the instance filed choose our running instance and select /dev/sdf or similar device name. Click on Attach volume.



Connect our PuTTY and list all the attached disks to verify our new volume using lsblk we can see the attached volumes in the below image.

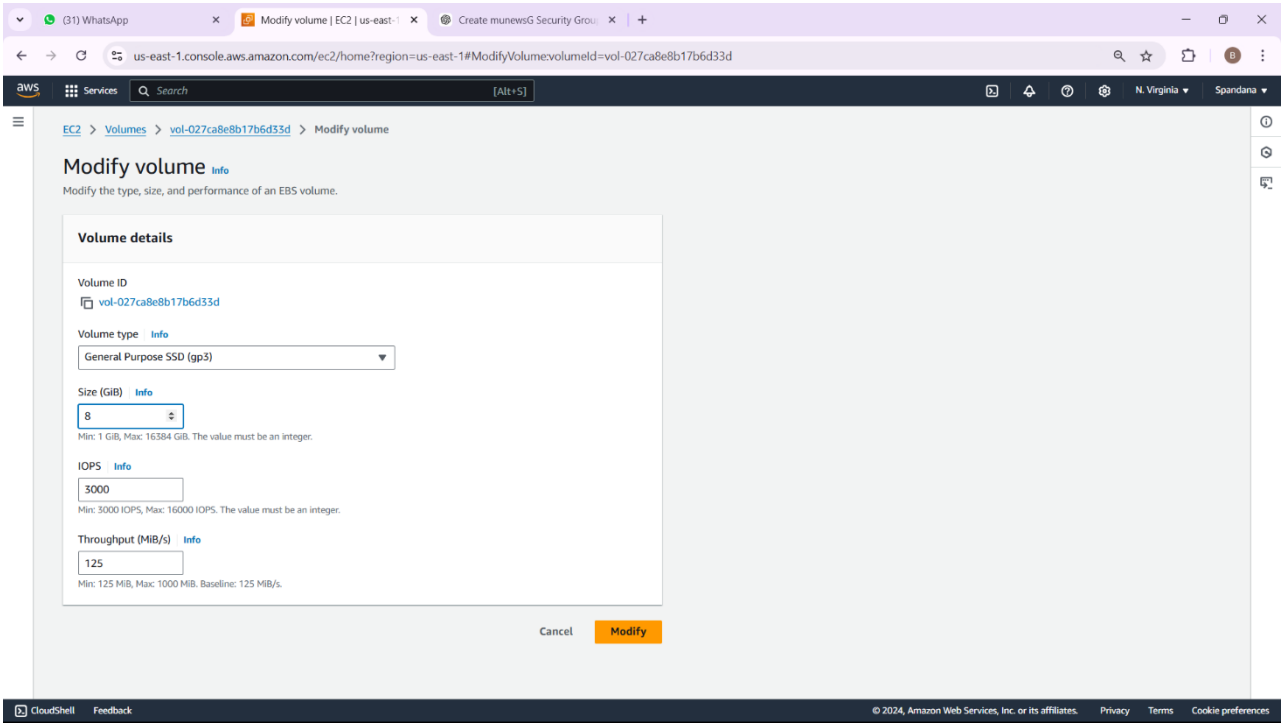


Create Test files using touch command. We can able to see 2 testfiles in the below image.

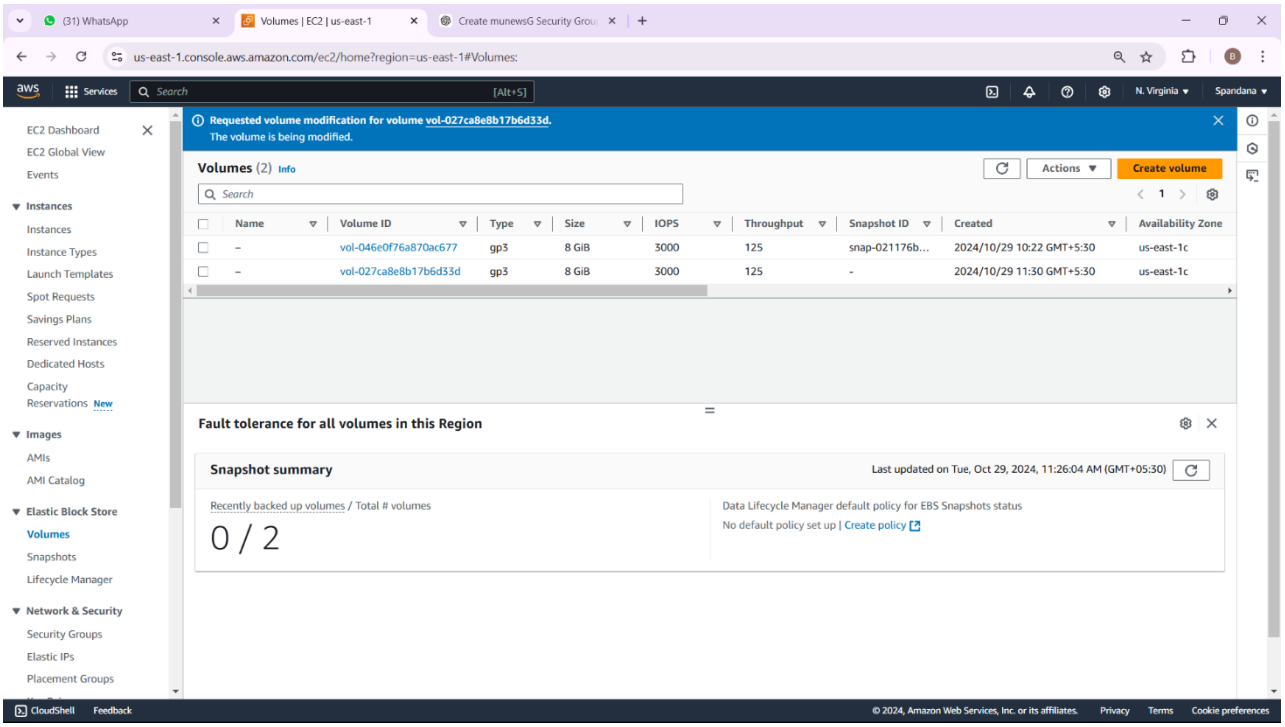



```
buntu@ip-172-31-35-242:~$ sudo mkdir /mnt/mydata
buntu@ip-172-31-35-242:~$ ll
otal 28
rwxr-x--- 4 ubuntu ubuntu 4096 Oct 29 05:58 ./
rwxr-xr-x 3 root   root   4096 Oct 29 04:52 ../
rw-r--r-- 1 ubuntu ubuntu  220 Mar 31  2024 .bash_logout
rw-r--r-- 1 ubuntu ubuntu 3771 Mar 31  2024 .bashrc
rwx------ 2 ubuntu ubuntu 4096 Oct 29 04:59 .cache/
rw-r--r-- 1 ubuntu ubuntu  807 Mar 31  2024 .profile
rwx------ 2 ubuntu ubuntu 4096 Oct 29 04:52 .ssh/
rw-r--r-- 1 ubuntu ubuntu    0 Oct 29 05:58 .sudo_as_admin_successful
buntu@ip-172-31-35-242:~$ sudo mount /dev/xvdf /mnt/data
ount: /mnt/data: mount point does not exist.
      dmesg(1) may have more information after failed mount system call.
buntu@ip-172-31-35-242:~$ sudo mount /dev/xvdf /mnt/mydata
buntu@ip-172-31-35-242:~$ df -h
ilesystem      Size  Used Avail Use% Mounted on
dev/root       6.8G  1.6G  5.2G  24% /
mpfs           479M     0  479M   0% /dev/shm
mpfs           192M  888K  191M   1% /run
mpfs           5.0M     0  5.0M   0% /run/lock
dev/xvda16     881M   76M  744M  10% /boot
dev/xvda15     105M  6.1M   99M   6% /boot/efi
mpfs            96M   12K   96M   1% /run/user/1000
dev/xvdf       4.9G   24K  4.6G   1% /mnt/mydata
buntu@ip-172-31-35-242:~$ cd /mnt/mydata
buntu@ip-172-31-35-242:/mnt/mydata$ ll
otal 24
rwxr-xr-x 3 root root  4096 Oct 29 06:04 ./
rwxr-xr-x 3 root root  4096 Oct 29 06:05 ../
rwx------ 2 root root 16384 Oct 29 06:04 lost+found/
buntu@ip-172-31-35-242:/mnt/mydata$
```

Go to volumes and select our 5Gib volume click on Actions > Modify volume change the size to 8Gib and click on Modify. Confirm the modification. AWS will automatically resize the volume.



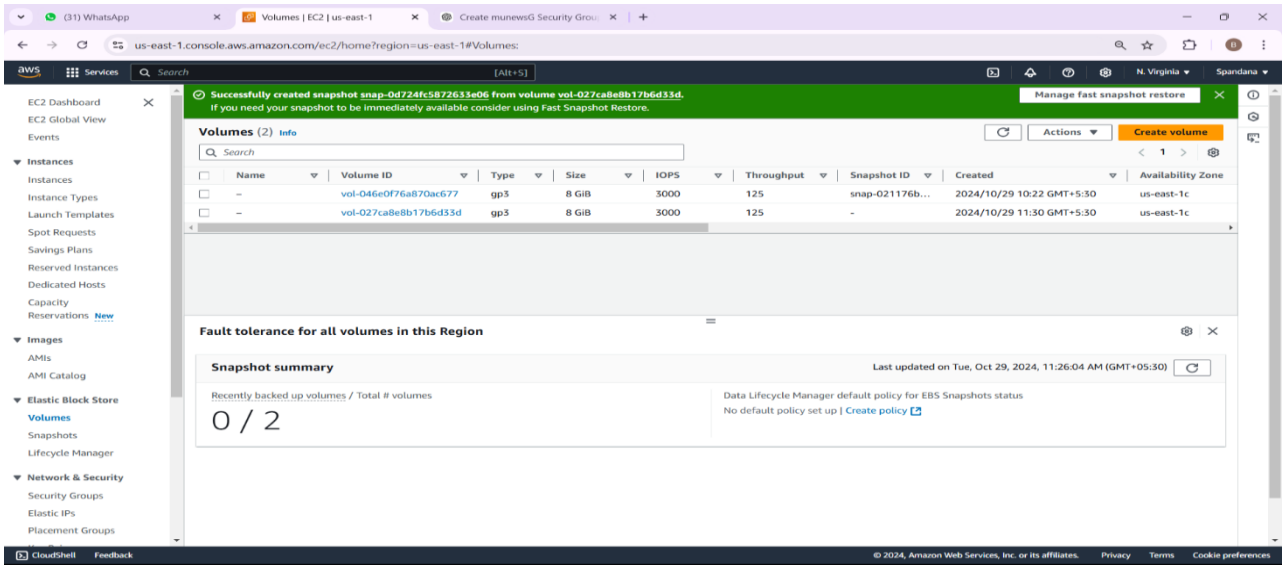
In the below image we can able to see that the volume size has been increased from 5GiB to 8GiB.

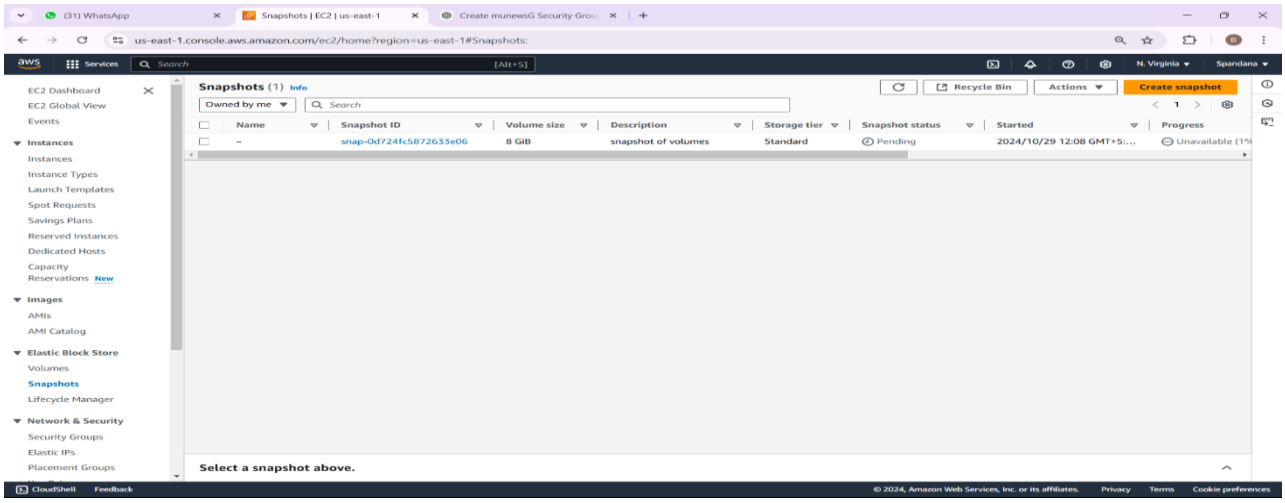


Return to our PuTTY and use grow part to extend the partition to occupy the increased volume space and use resize2fs to extend the filesystem to utilize the full size in the below image.

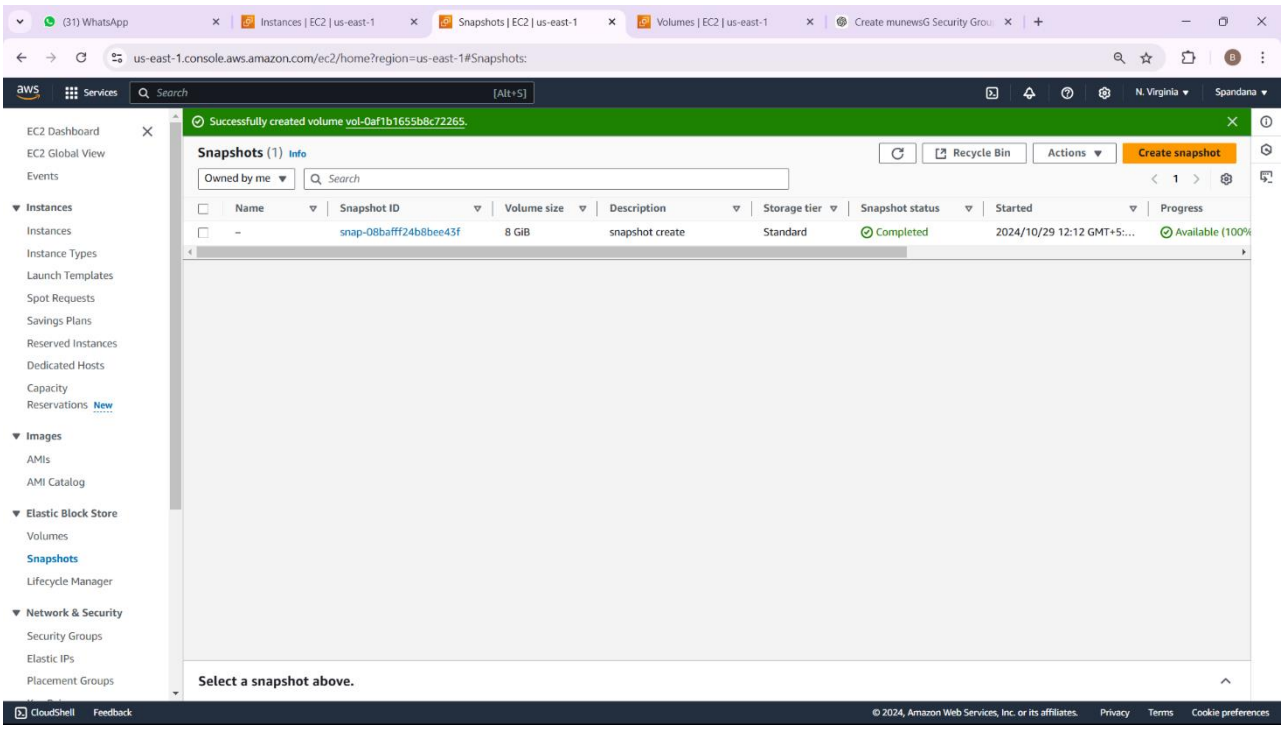
```
lost+found  testfile1.txt  testfile2.txt
root@ip-172-31-35-242:/mnt/mydata# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       6.8G  1.6G  5.2G  24% /
tmpfs           479M     0  479M   0% /dev/shm
tmpfs           192M  888K  191M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
/dev/xvda16     881M   76M  744M  10% /boot
/dev/xvda15     105M  6.1M   99M   6% /boot/efi
tmpfs            96M   12K   96M   1% /run/user/1000
/dev/xvdf       4.9G   28K  4.6G   1% /mnt/mydata
root@ip-172-31-35-242:/mnt/mydata# lsblk
NAME     MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0      7:0    0 25.2M  1 loop /snap/amazon-ssm-agent/7993
loop1      7:1    0 38.8M  1 loop /snap/snapd/21759
loop2      7:2    0 55.7M  1 loop /snap/core18/2829
xvda     202:0    0    8G  0 disk
├─xvda1  202:1    0    7G  0 part /
├─xvda14 202:14   0    4M  0 part
├─xvda15 202:15   0  106M  0 part /boot/efi
└─xvda16 259:0    0  913M  0 part /boot
xvdf     202:80   0    8G  0 disk /mnt/mydata
root@ip-172-31-35-242:/mnt/mydata#
```

Go to volumes in the EC2 select the volume click on Action > Create snapshot enter a description click on create snapshot. Snapshot will appear under the snapshots in the Elastic Block Store section.

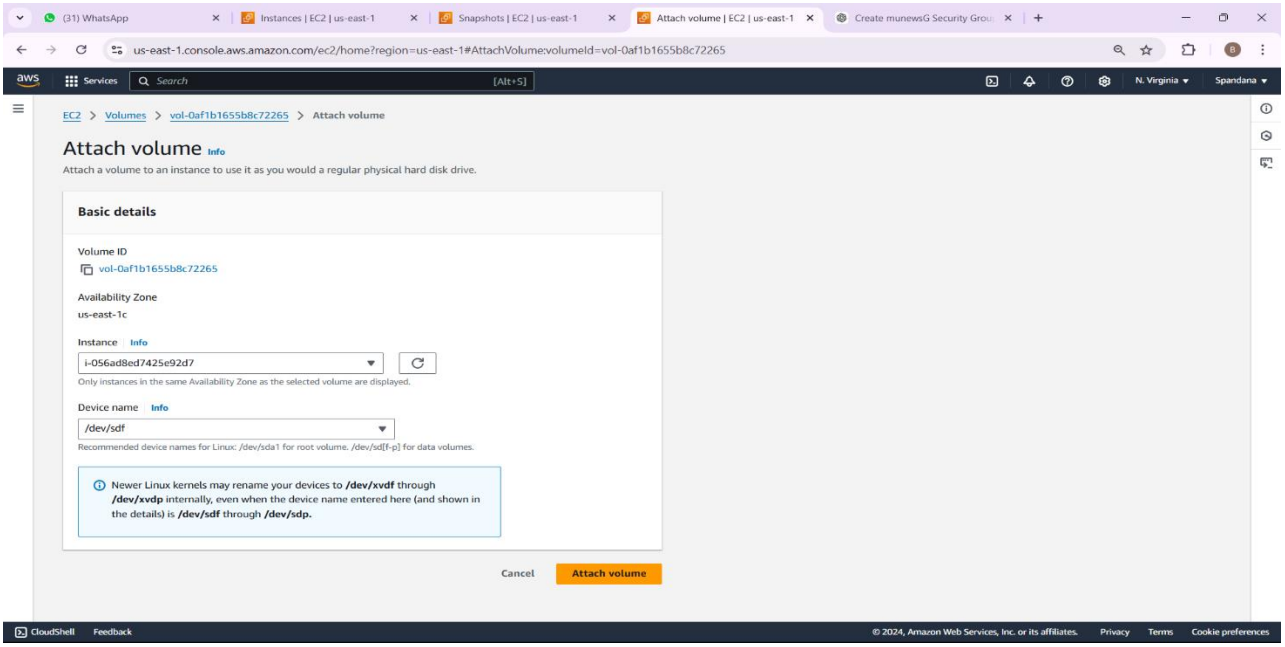




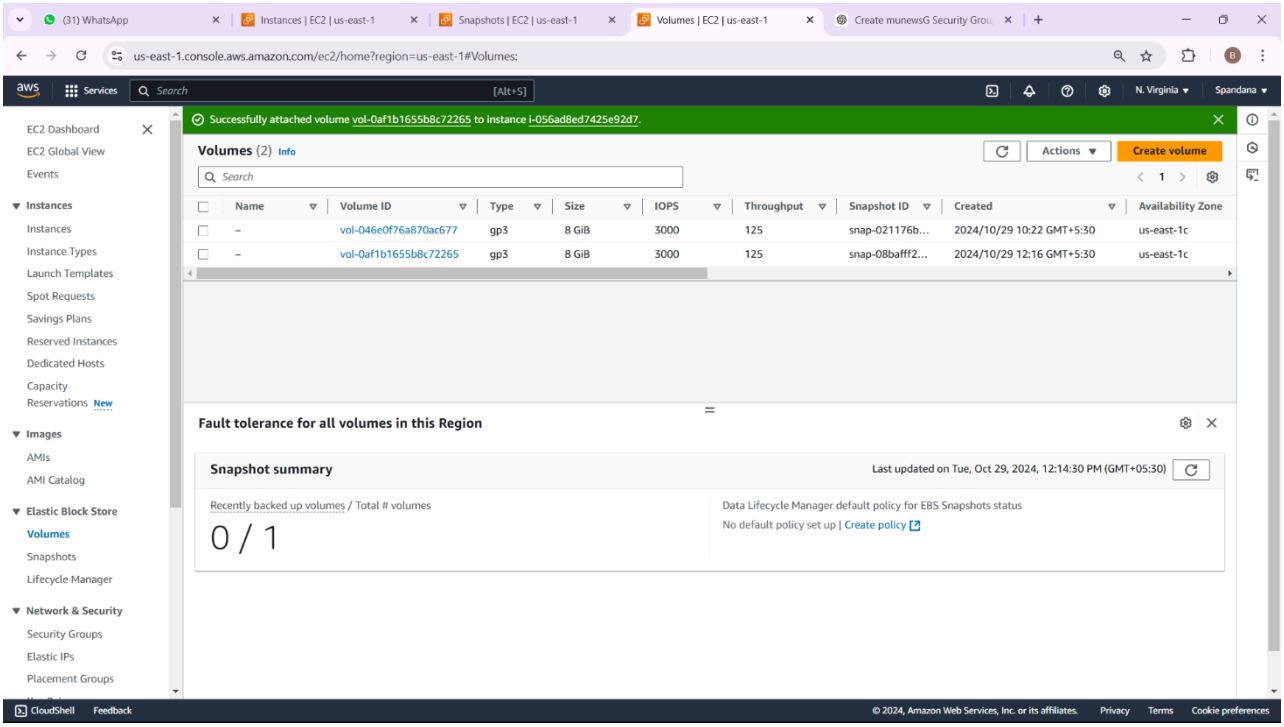
Select the snapshot you created earlier with the snapshot selected click > Action > Create Volume in that Select Availability Zone as your EC2 instance select Volume size and Volume type Choose the desired type Click create Volume to create a new volume from the snapshot.



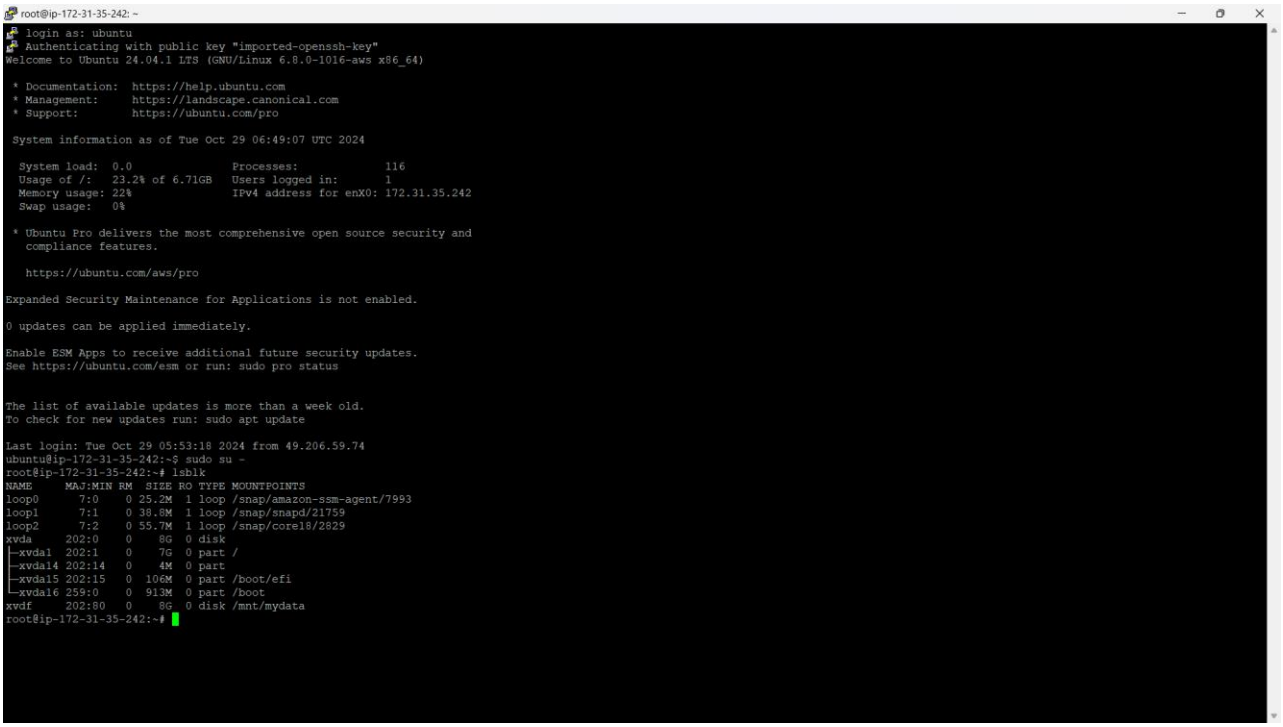
In the Volumes section under Elastic Block Store find the newly created volume click Action > Attach Volume select our EC2 instance and choose a device name Click on Attach Volume to attach it to your instance.



Successfully attached Volume to EC2 instance.

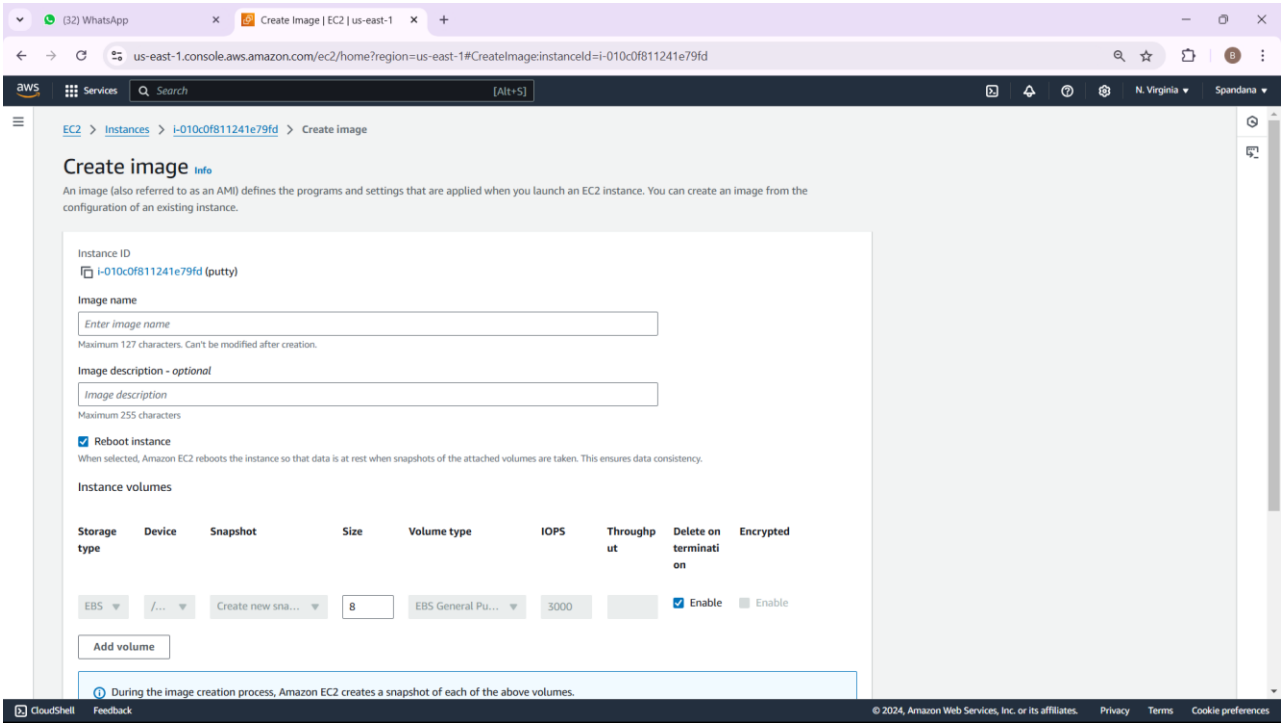


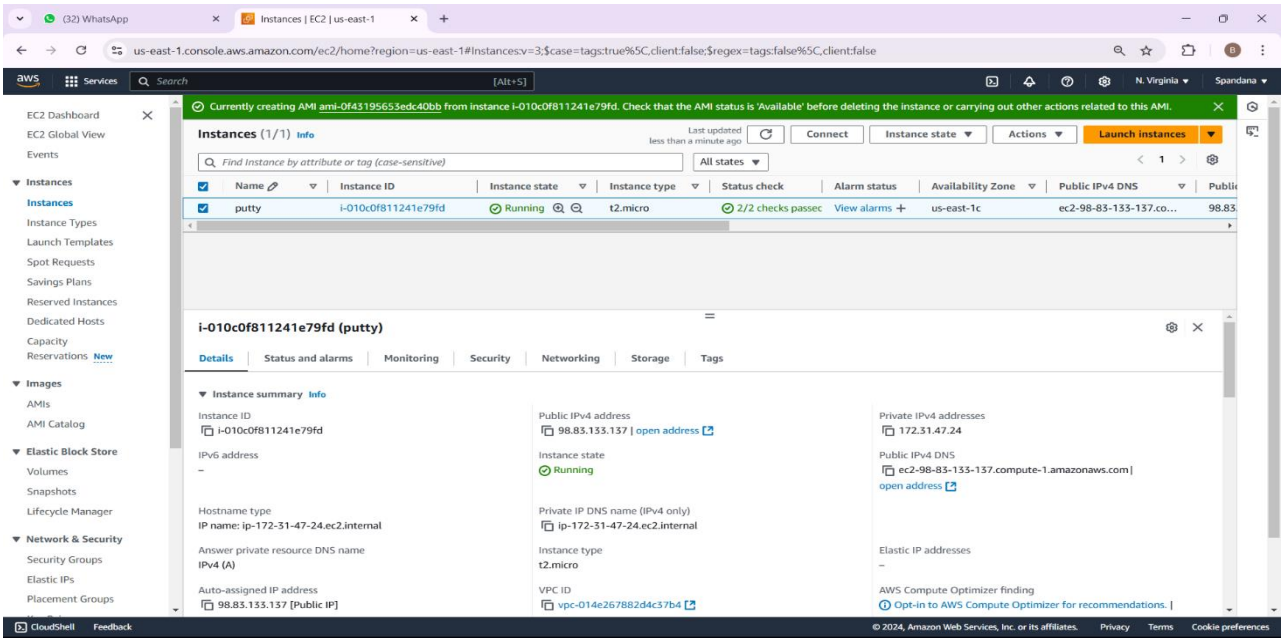
Open PuTTY and connect to our instance run the lsblk command to verify the attached volume. We will see the new volume listed.



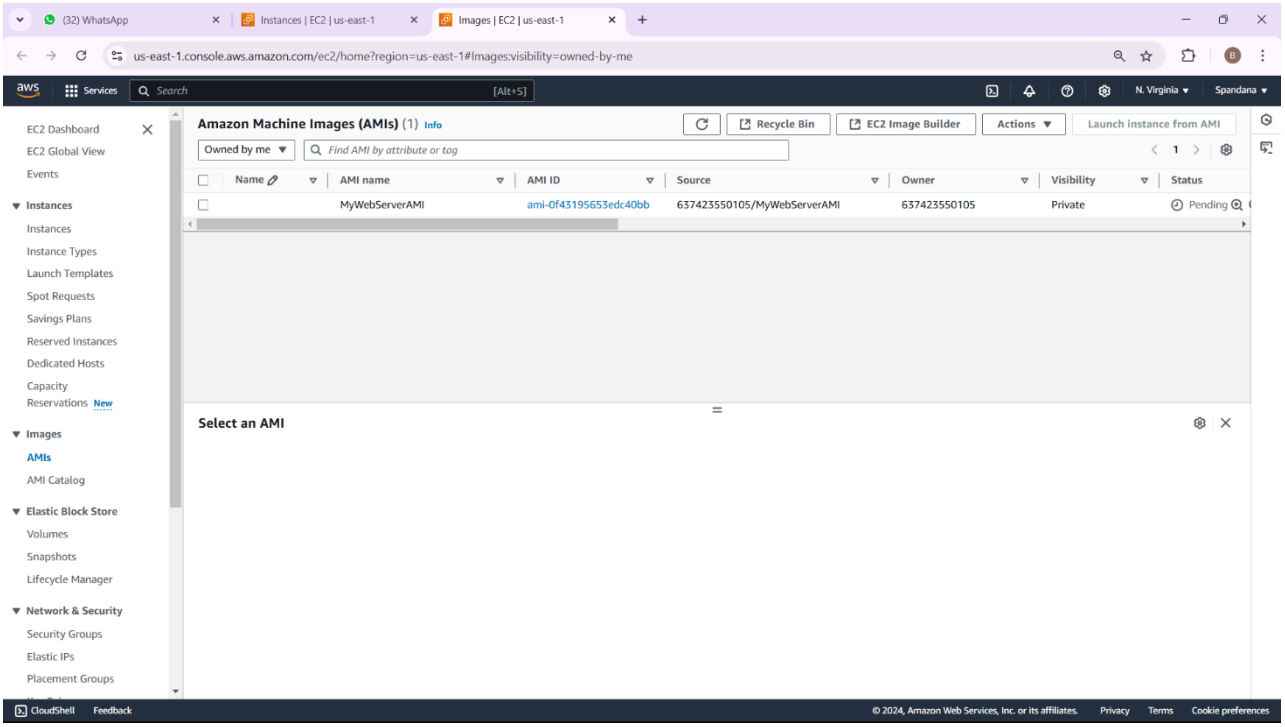
LAB 7- AMI (AMAZON MACHINE IMAGE)

Select the instance and with the instance selected, click Actions > Image and templates > Create image. In the Create Image name image description optionally, add a description instance volumes verify that the root volume and any additional volumes you want are selected. You can adjust the size or add new volumes as needed. No reboot check this option if you want to avoid a reboot during image creation.



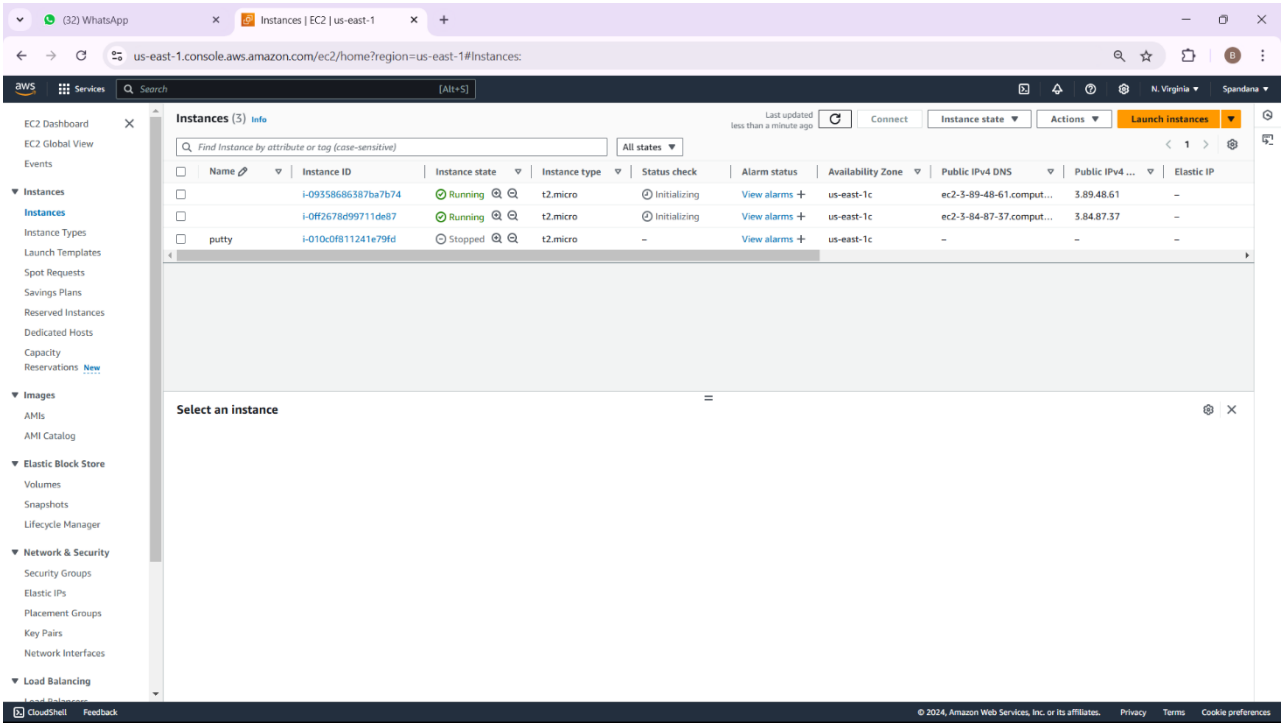


Successfully created Amazon Machine Image we can able to see in the below image.

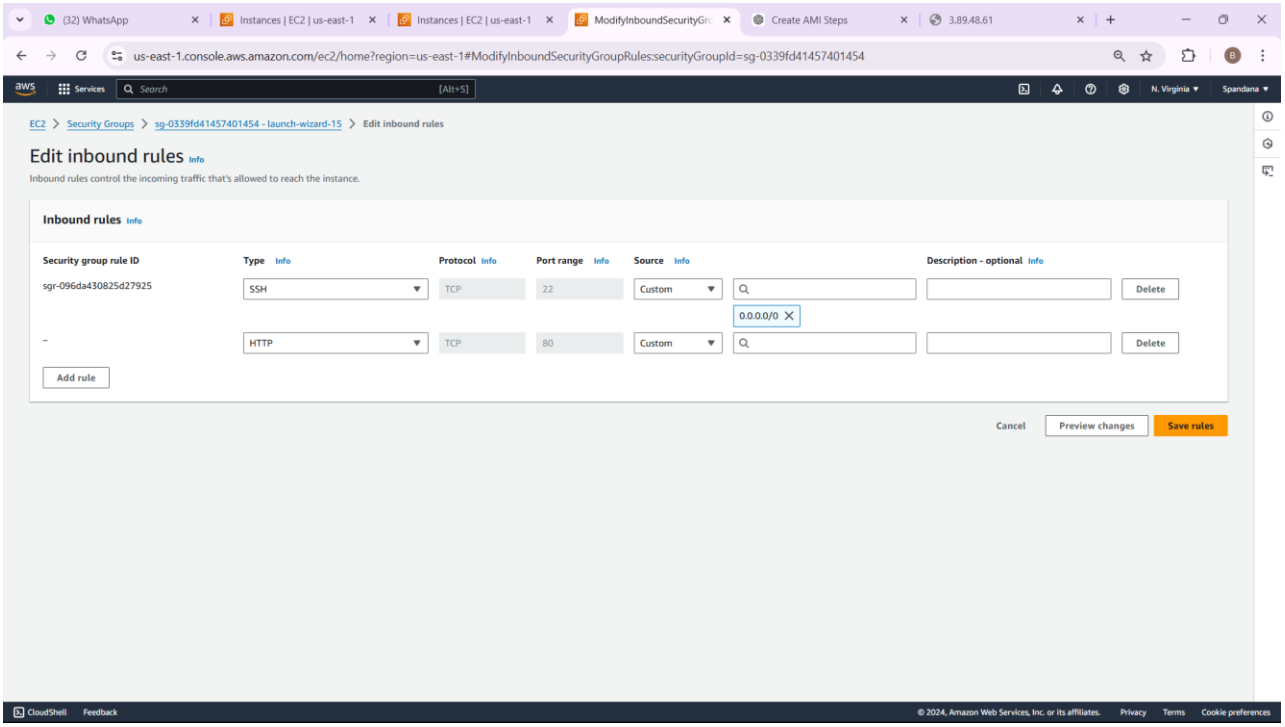


LAB 8 – LOAD BALANCER

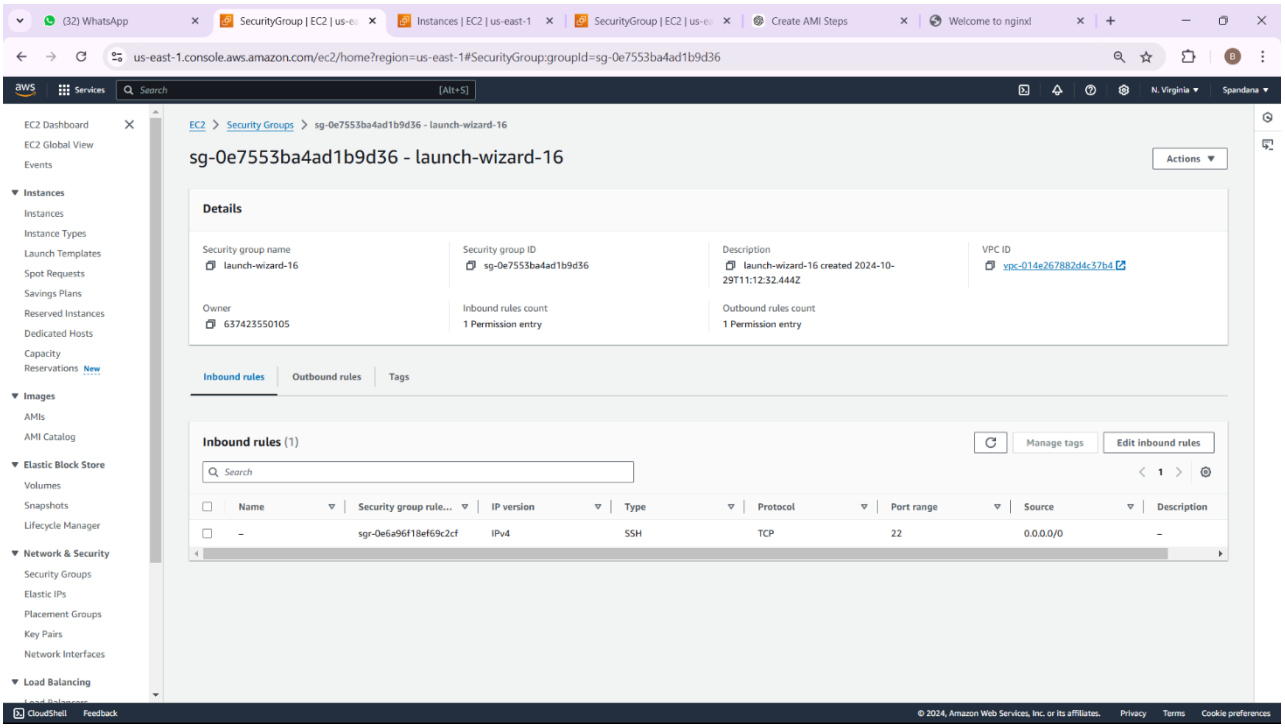
Launch two EC2 instances ensure both instances are in same availability zone for simplicity and assign security groups to the instances that allows only inbound traffic from load balancer on port 80.



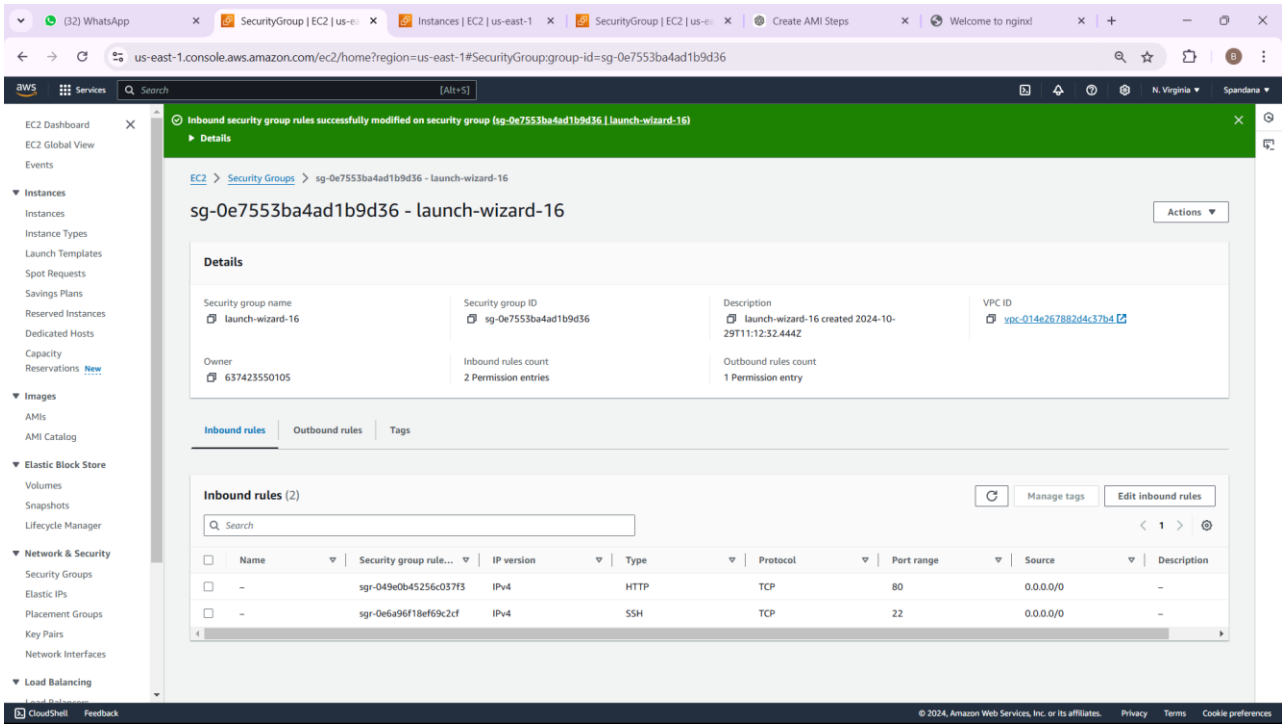
In the below image we can see that inbound traffic on port 80 (http)



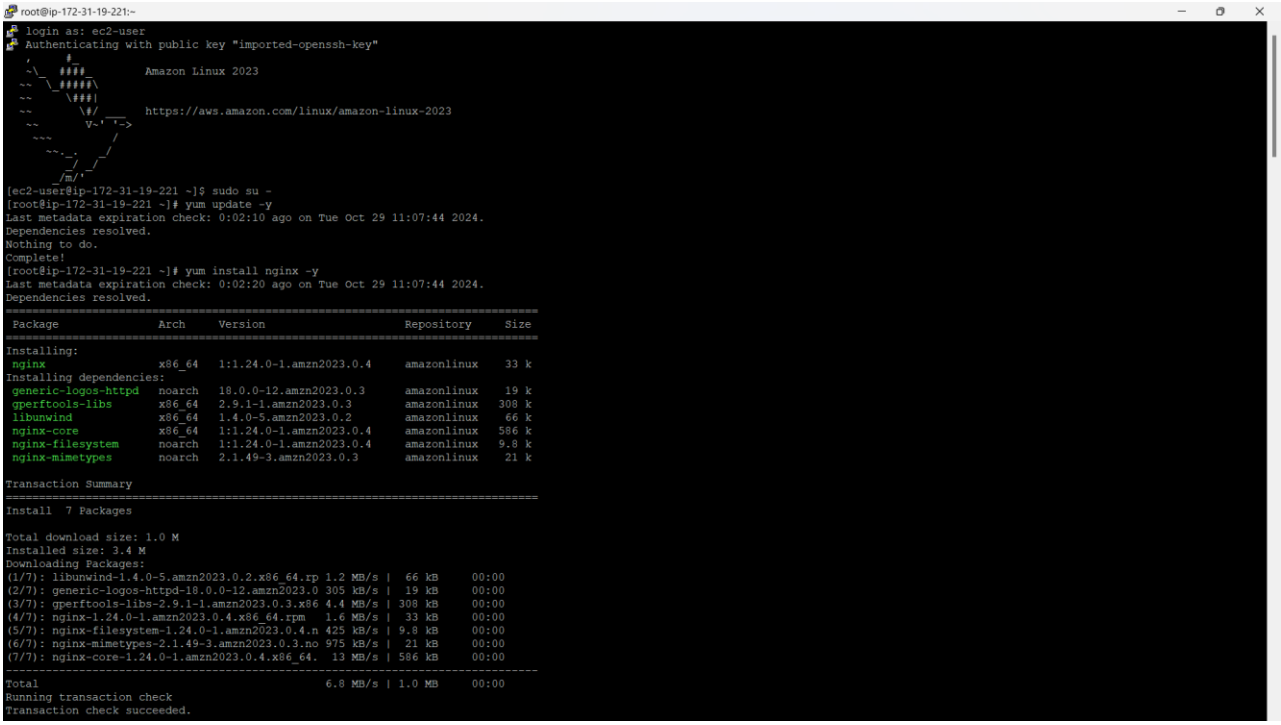
The below image shows that the before modification of inbound rules.



The below image shows that the inbound rules has been successfully updated.

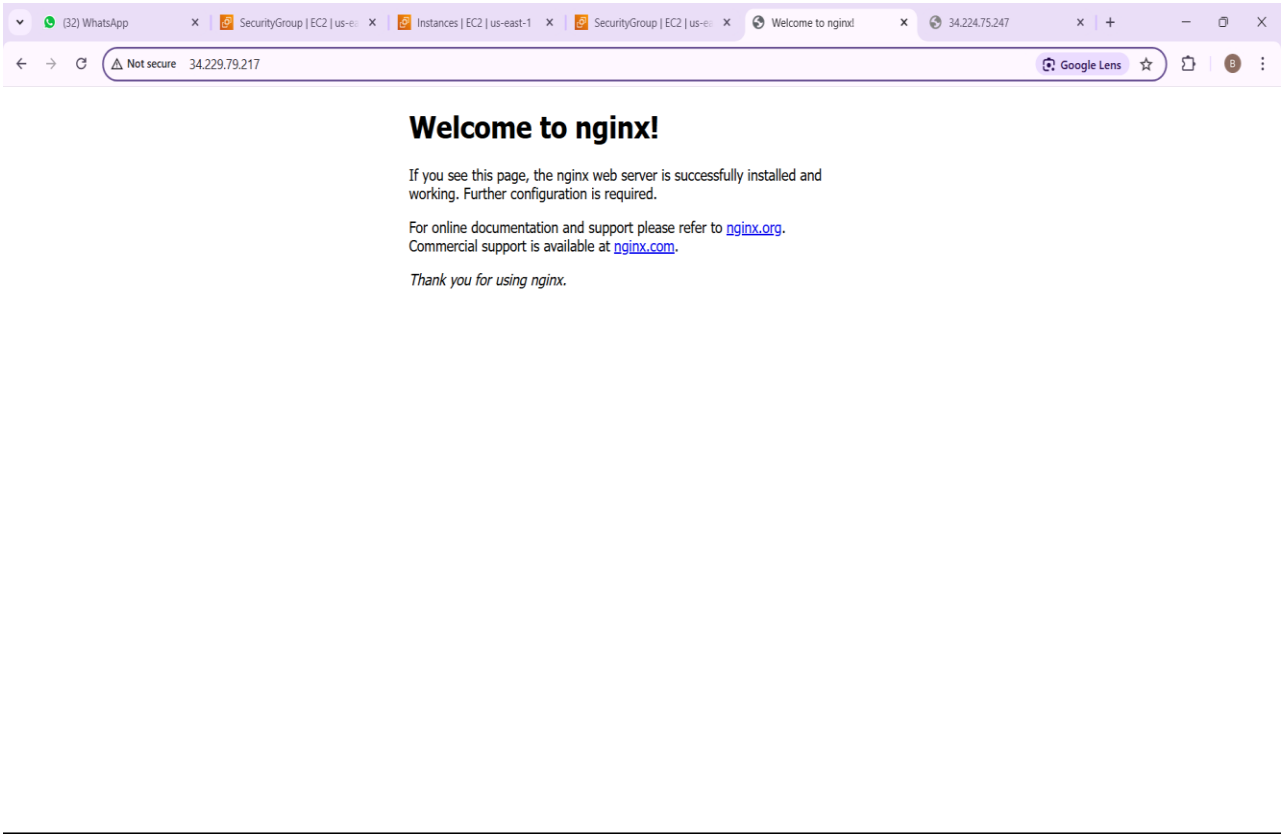


Connect both instances using PuTTY and installing web server using ec2-user for nginx. Switch to root user through **sudo su -,** update the package index using **yum install update -y,** Install nginx using **yum install nginx -y** and start and enable nginx server using **systemctl start nginx && systemctl enable nginx.**

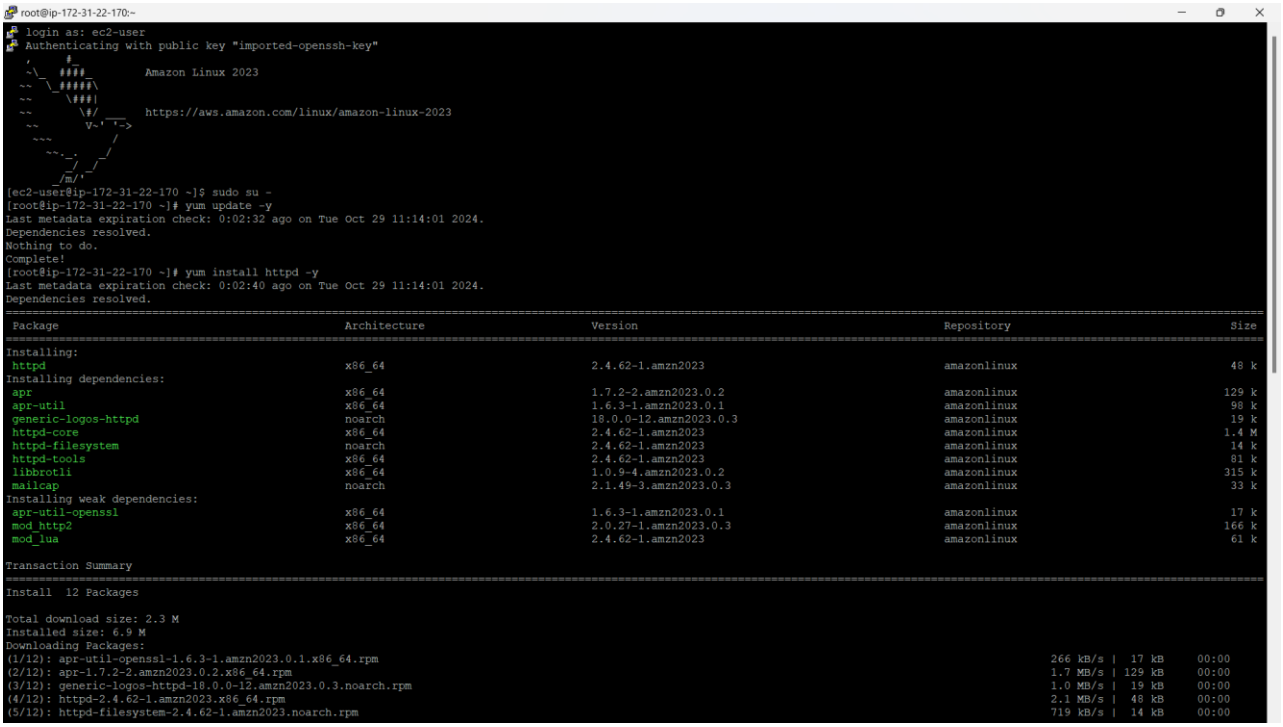


We can able to check the status through systemctl status and we can able to see nginx server is in active state in below image.

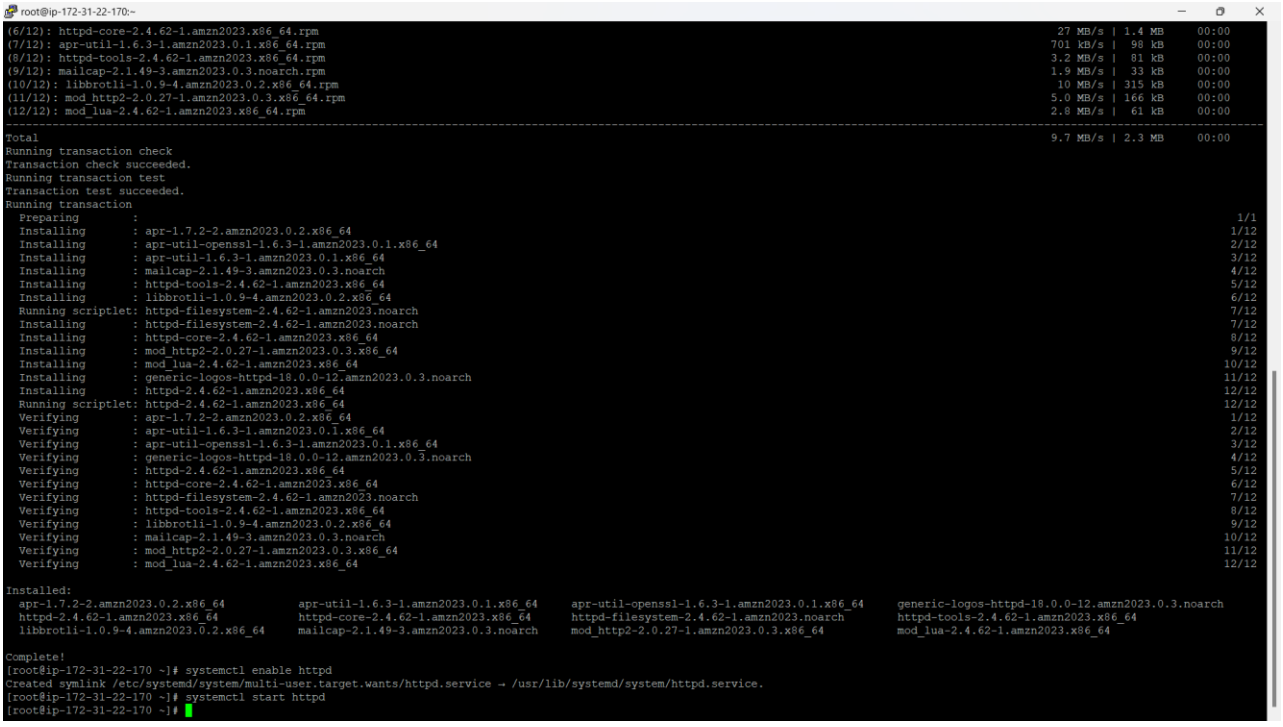
Access nginx web server through copy each instance's public IP address in browser to verify. If the server is properly connected then we can able to see like a below image and we can confirm that nginx server is connected.



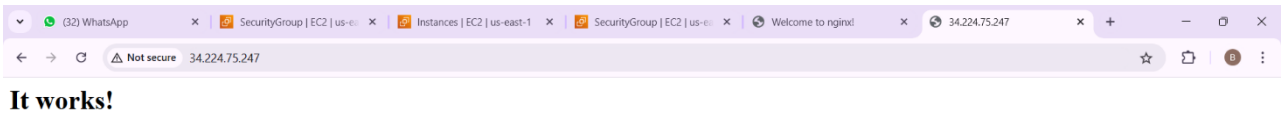
Connect both instances using PuTTY and installing web server using ec2-user for Apache. Switch to root user through **sudo su -,** update the package index using **yum install update -y,** Install Apache using **yum install httpd -y.**



Start and enable httpd server using **systemctl start httpd && systemctl enable httpd.**



Copy the public IP of your instance and paster it in our browser so that we can able to see like a below image if it is like this then the server is connected properly.

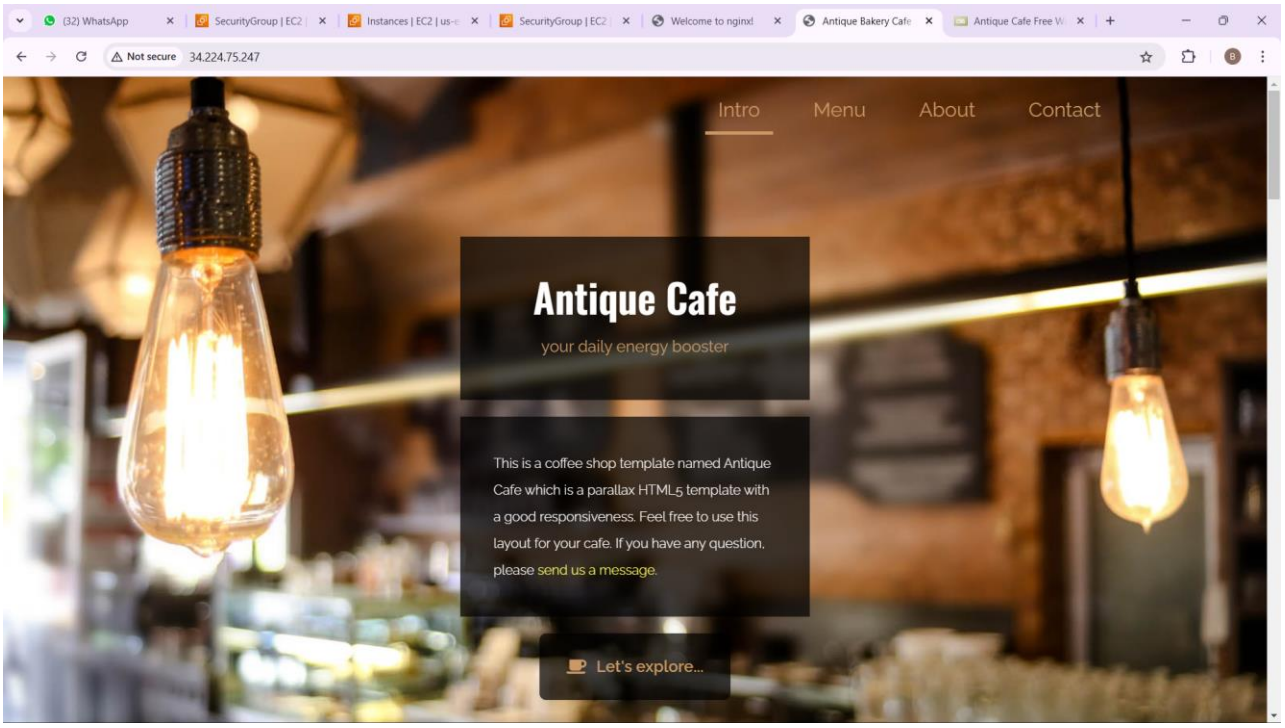


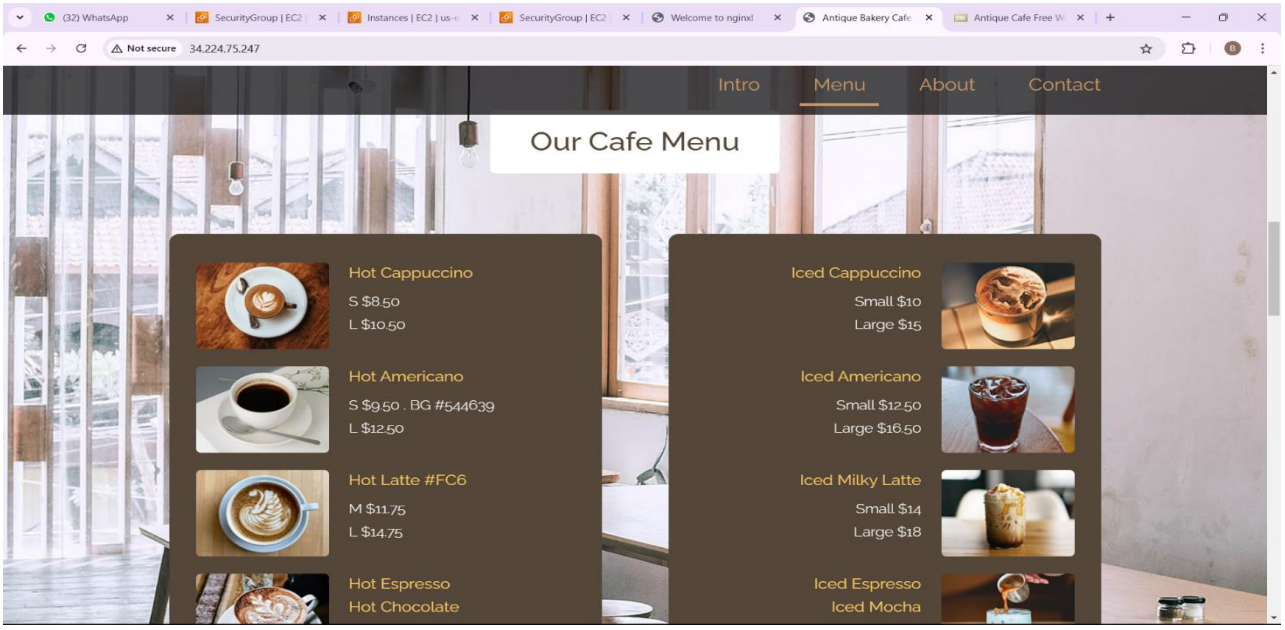
Deploying one application through Apache (httpd):

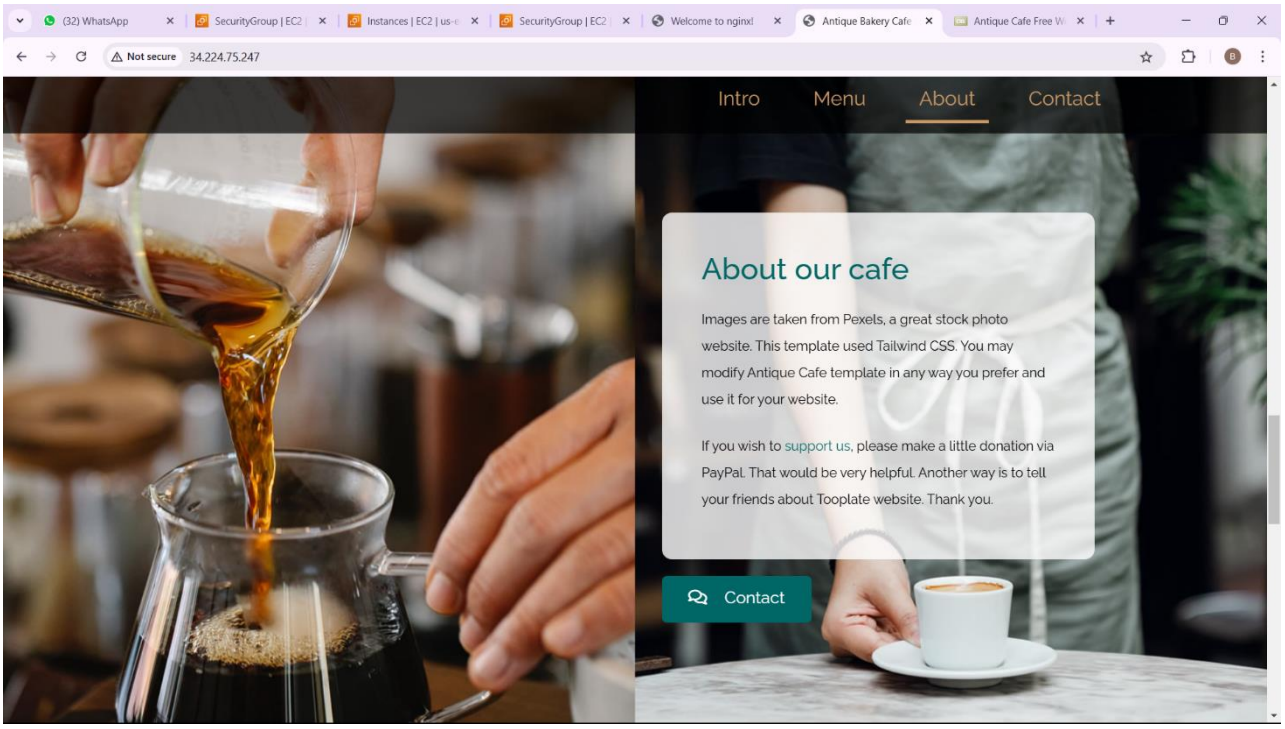
Setting Up the Apache (HTTP) Server with "Antique Cafe" Template Connect to the Apache EC2 instance using SSH (ec2-user). Open Antique Cafe and click and copy the link and go to PuTTY and paste using wget <link> we will get one zip file that is **antique-café.zip** unzip the file using **unzip <filename>** after that we will get another file and go to that file and check through ll we will find some files move all files for the httpd path usually /var/www/html/ this is the default path for httpd. After that copy the public IP and paster it our browser so that we can able to see the application as shown in the below images.

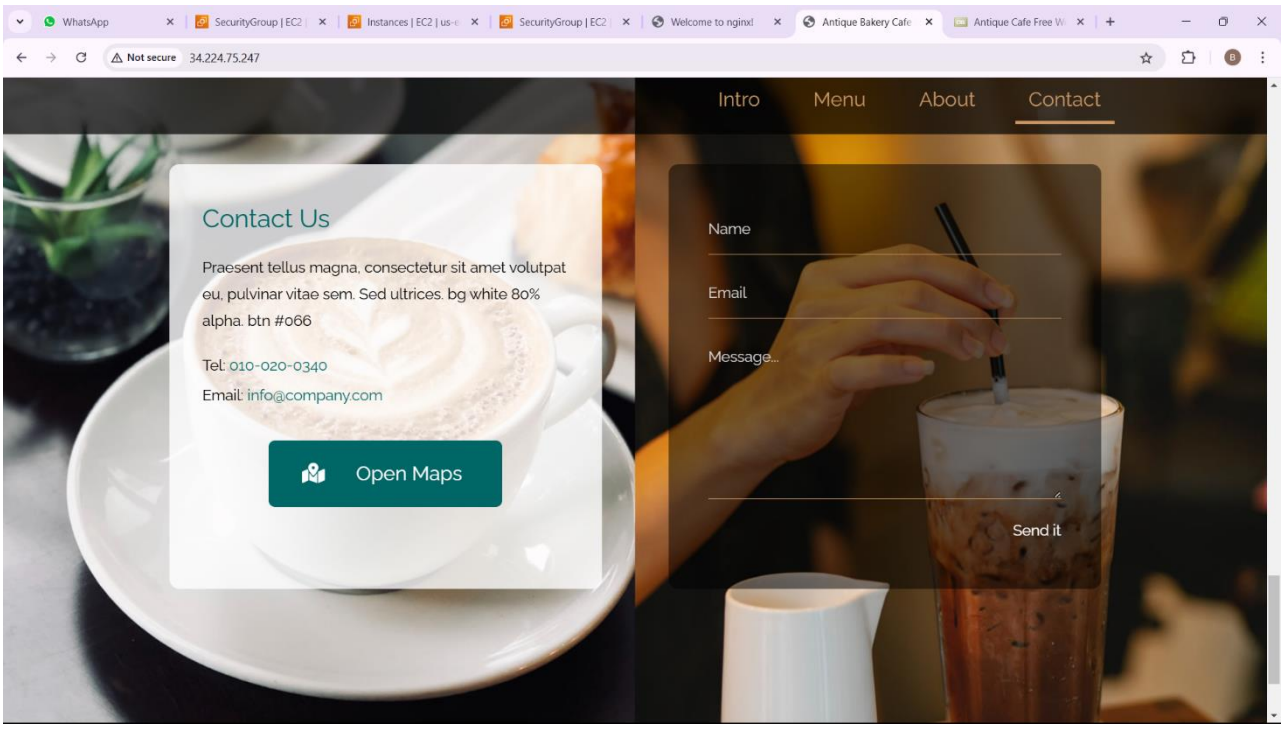
```
[root@ip-172-31-22-170 ~]# ll
total 2168
drwxr-xr-x. 6 root root     103 Sep 30  2021 2126_antique_cafe
-rw-r--r--. 1 root root 2217176 Jan 26  2022 antique-cafe.zip
[root@ip-172-31-22-170 ~]# cd 2126_antique_cafe/
[root@ip-172-31-22-170 2126_antique_cafe]# ll
total 52
-rw-r--r--. 1 root root   510 Jul 30  2019 'ABOUT THIS TEMPLATE.txt'
drwxr-xr-x. 2 root root    78 Sep 30  2021  css
drwxr-xr-x. 2 root root 16384 Sep 29  2021  img
-rw-r--r--. 1 root root 15522 Sep 30  2021  index.html
drwxr-xr-x. 2 root root    91 Sep 29  2021  js
drwxr-xr-x. 2 root root 16384 Sep 29  2021  webfonts
[root@ip-172-31-22-170 2126_antique_cafe]# mv * /var/www/html
[root@ip-172-31-22-170 2126_antique_cafe]# ll
total 0
[root@ip-172-31-22-170 2126_antique_cafe]#
```

This is my Antique Café application using httpd



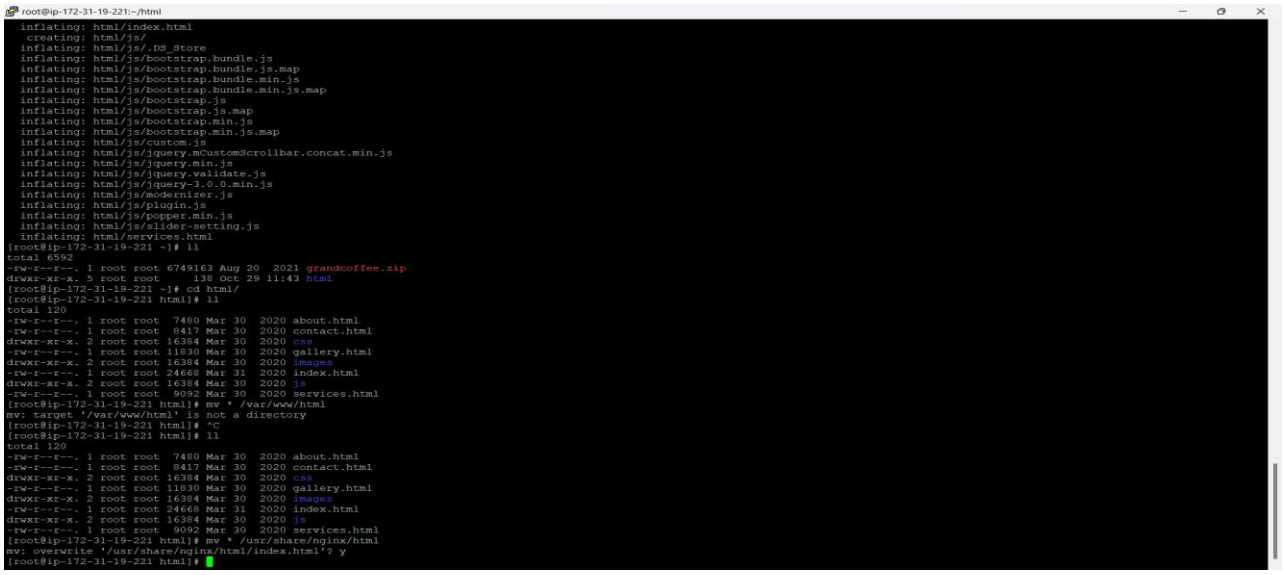


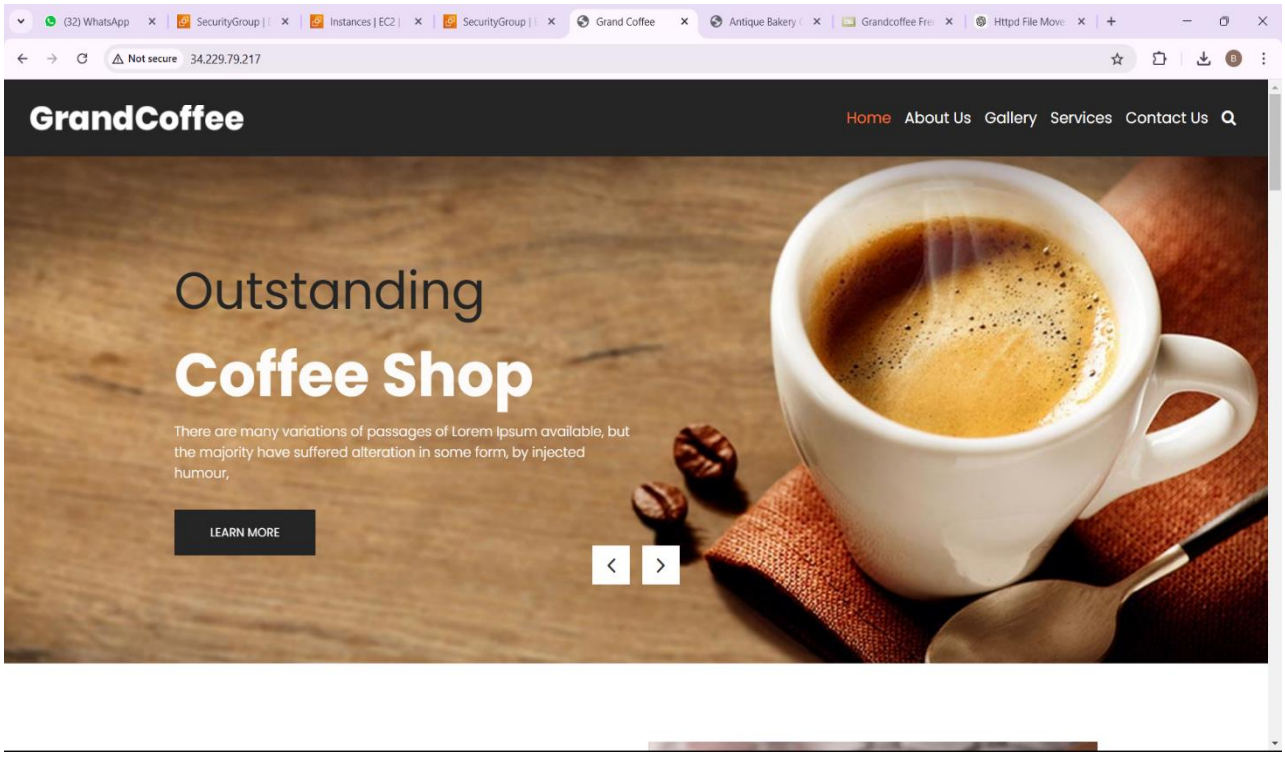

Not secure
34.224.75.247
Intro
Menu
About
Contact
About our cafe
Images are taken from Pexels, a great stock photo website. This template used Tailwind CSS. You may modify Antique Cafe template in any way you prefer and use it for your website.
If you wish to support us, please make a little donation via PayPal. That would be very helpful. Another way is to tell your friends about Tooplate website. Thank you.
Contact

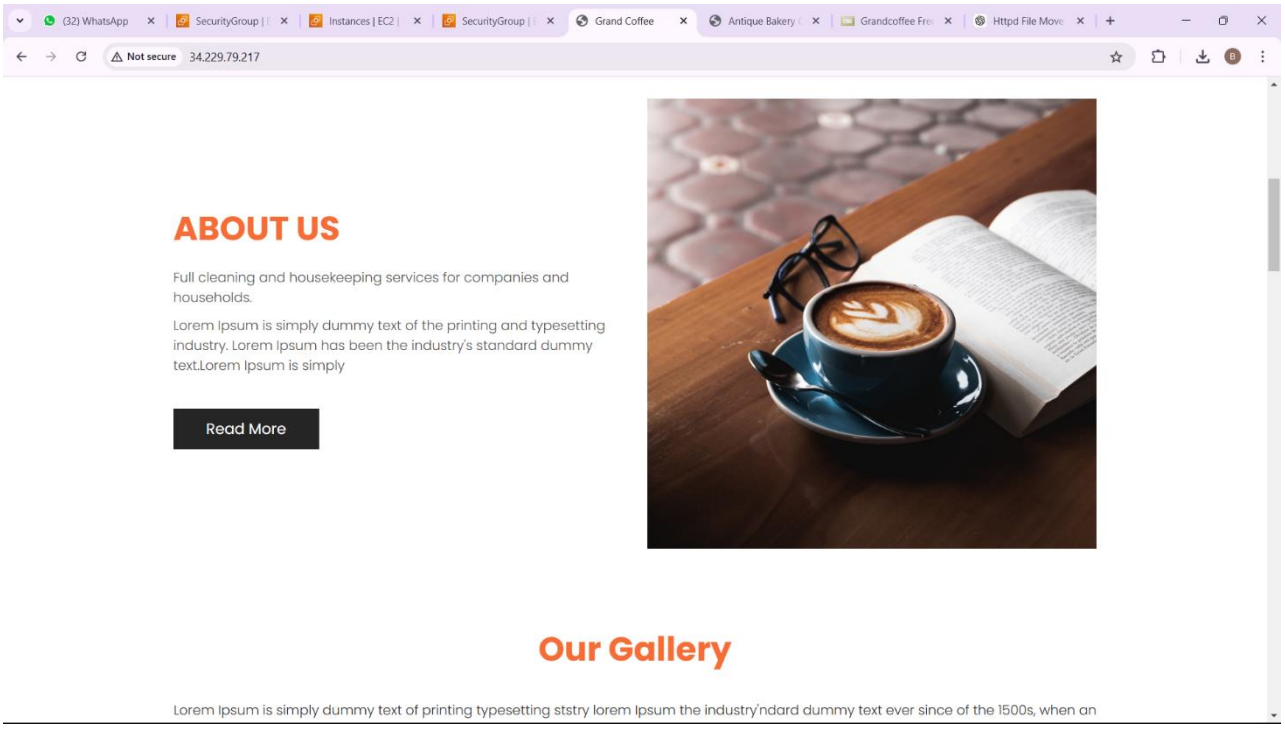

Not secure
34.224.75.247
Intro
Menu
About
Contact
Contact Us
Praesent tellus magna, consectetur sit amet volutpat eu, pulvinar vitae sem. Sed ultrices. bg white 80% alpha. btn #066
Tel: 010-020-0340
Email: info@company.com
Open Maps
Name
Email
Message...
Send it

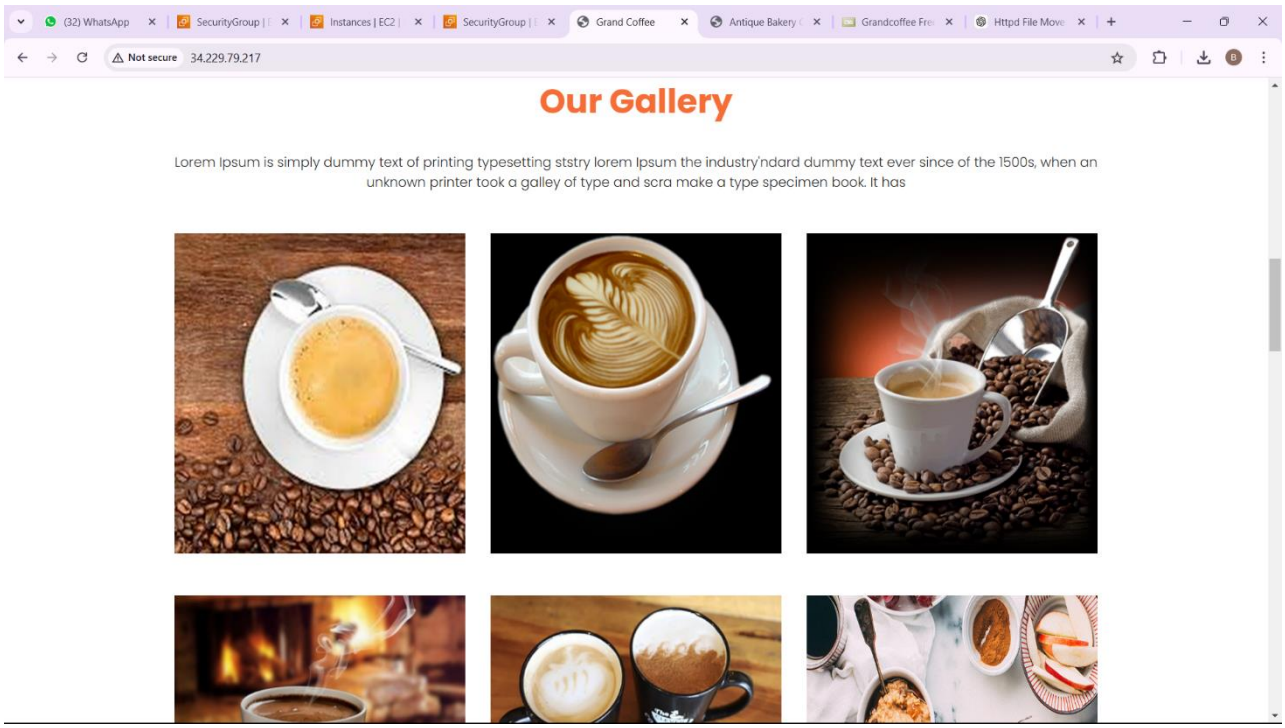
Setting Up the Nginx Server with "Grand Coffee" Application.

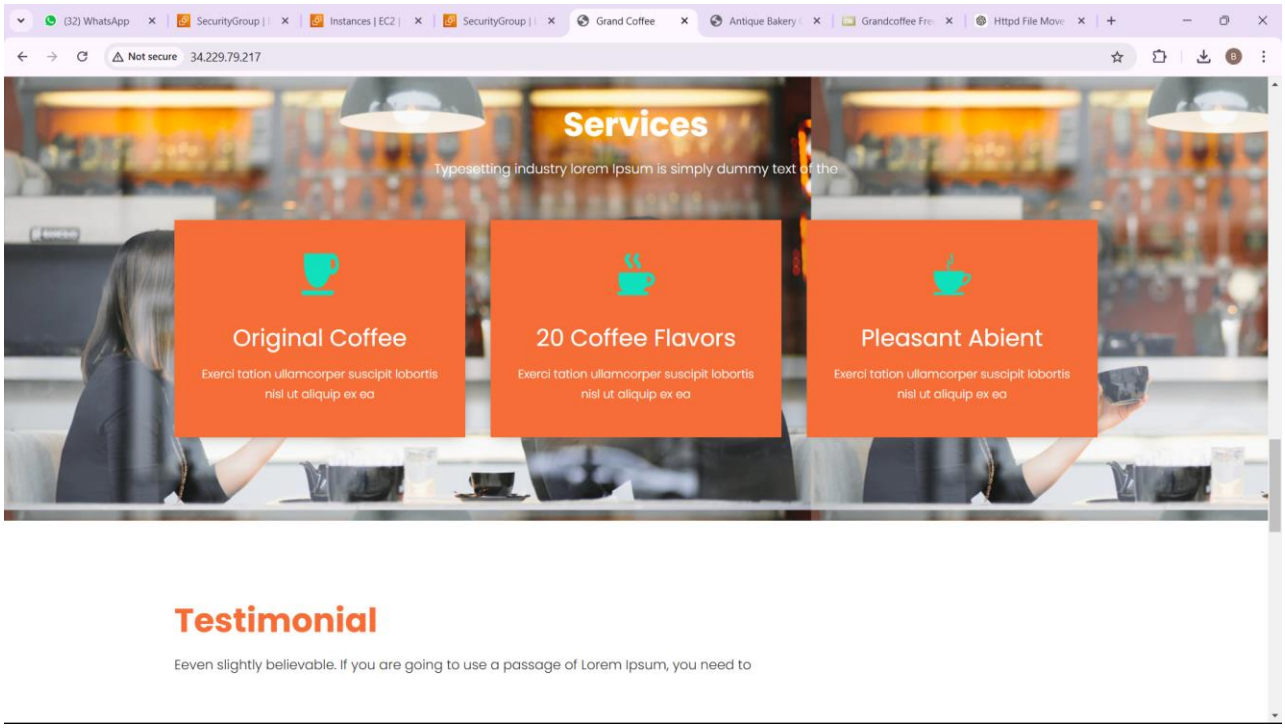
Setting Up the nginx Server with "Grand Coffee" Template Connect to the nginx EC2 instance using SSH (ec2-user). Open Grand Coffee and click and copy the link and go to PuTTY and paste using wget <link> we will get one zip file that is **grandcoffee.zip** unzip the file using **unzip <filename>** after that we will get another file and go to that file and check through ll we will find some files move all files for the nginx path usually /var/www/html/ or /usr/share/nginx/html this is the default path for nginx. After that copy the public IP and paster it our browser so that we can able to see the application as shown in the below images.





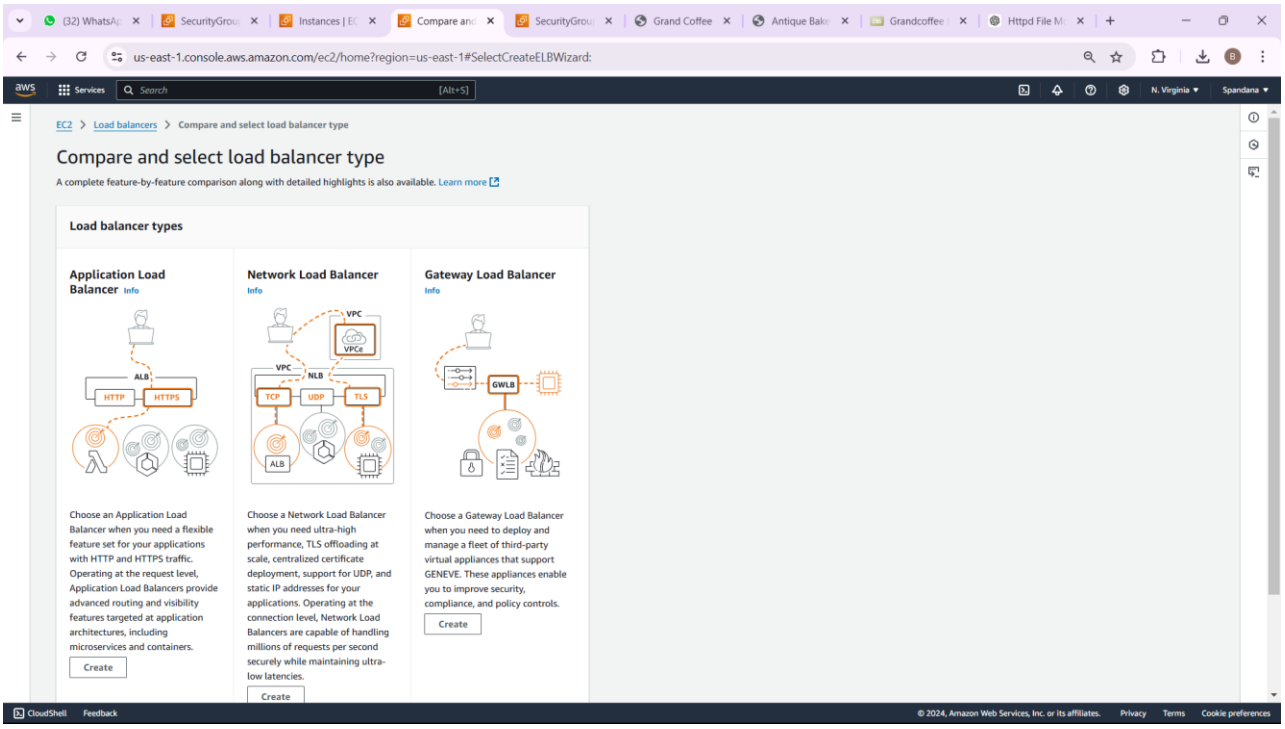

Not secure
34.229.79.217
ABOUT US
Full cleaning and housekeeping services for companies and households.
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text.Lorem Ipsum is simply
Read More
Our Gallery
Lorem Ipsum is simply dummy text of printing typesetting ststry lorem Ipsum the industry'ndard dummy text ever since of the 1500s, when an


Not secure
34.229.79.217
Our Gallery
Lorem Ipsum is simply dummy text of printing typesetting ststry lorem Ipsum the industry'ndard dummy text ever since of the 1500s, when an unknown printer took a galley of type and scra make a type specimen book. It has

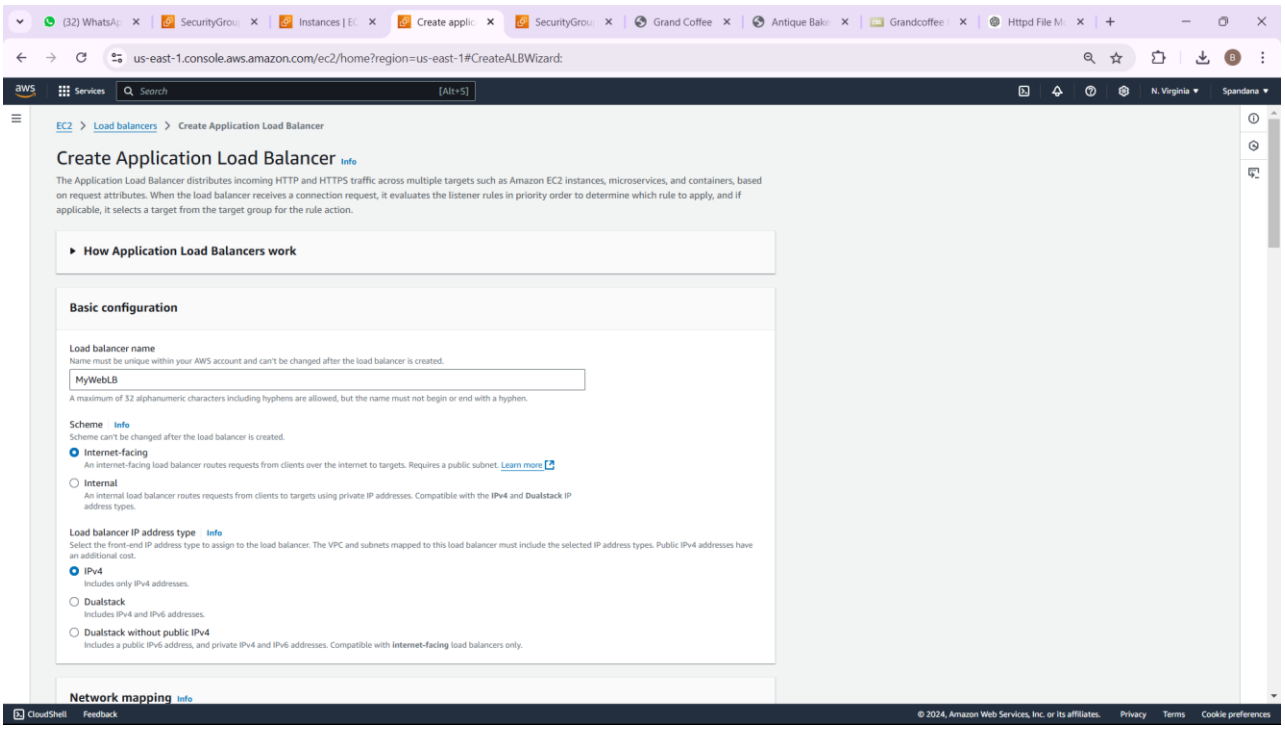


ATACHING LOAD BALACER TO SERVERS:

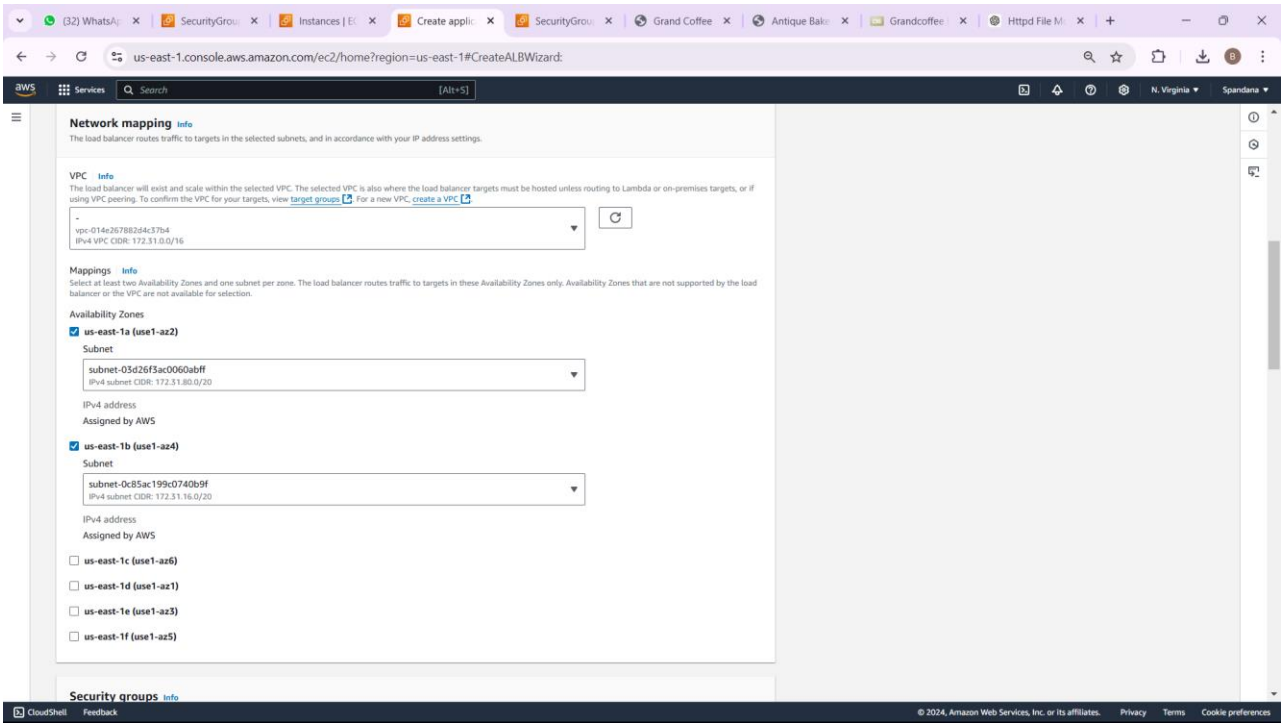
Go to EC2 Dashboard > Load Balancers and click Create Load Balancer. Choose Application Load Balancer or Classic Load Balancer .



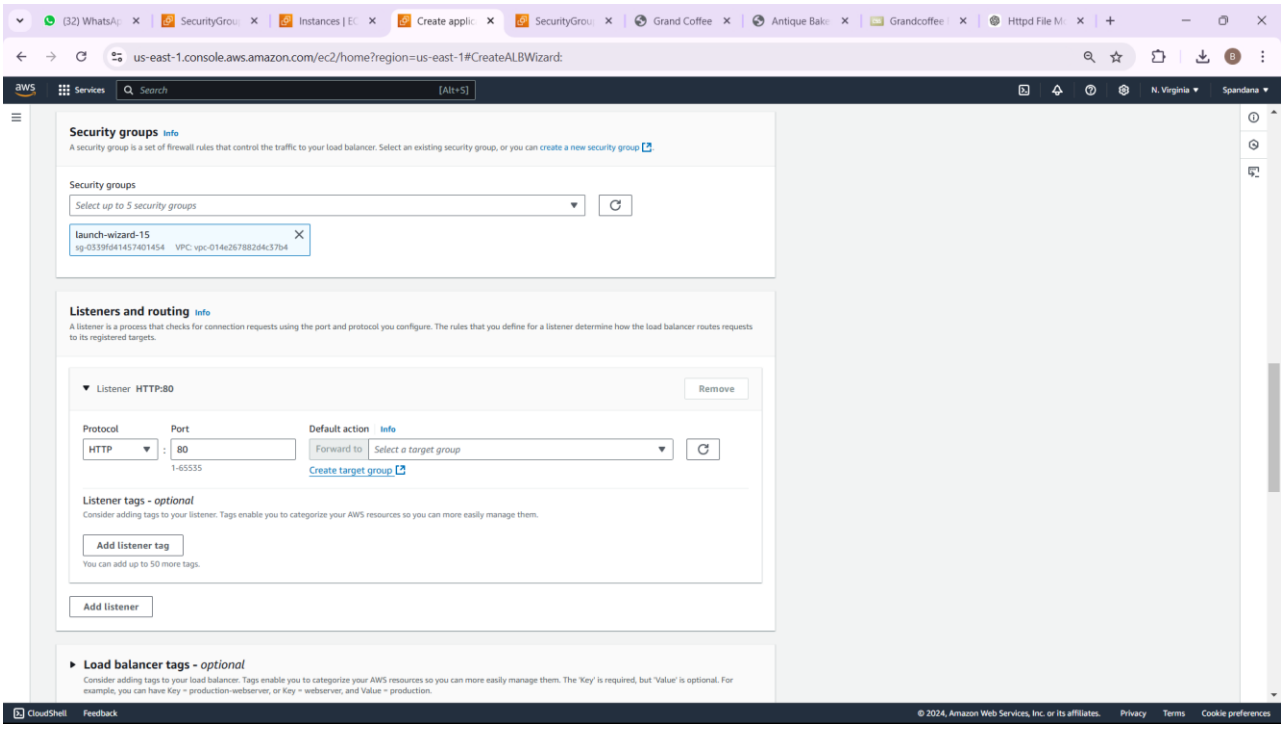
Configure the load balancer give name for the load balancer Scheme we have to select **Internet-facing** IP Address Type is IPv4. Listeners Set the listener protocol to HTTP and port to 80.



Availability Zones Select the availability zone where your EC2 instances are located and select one extra availability zone. If not it will through an error.

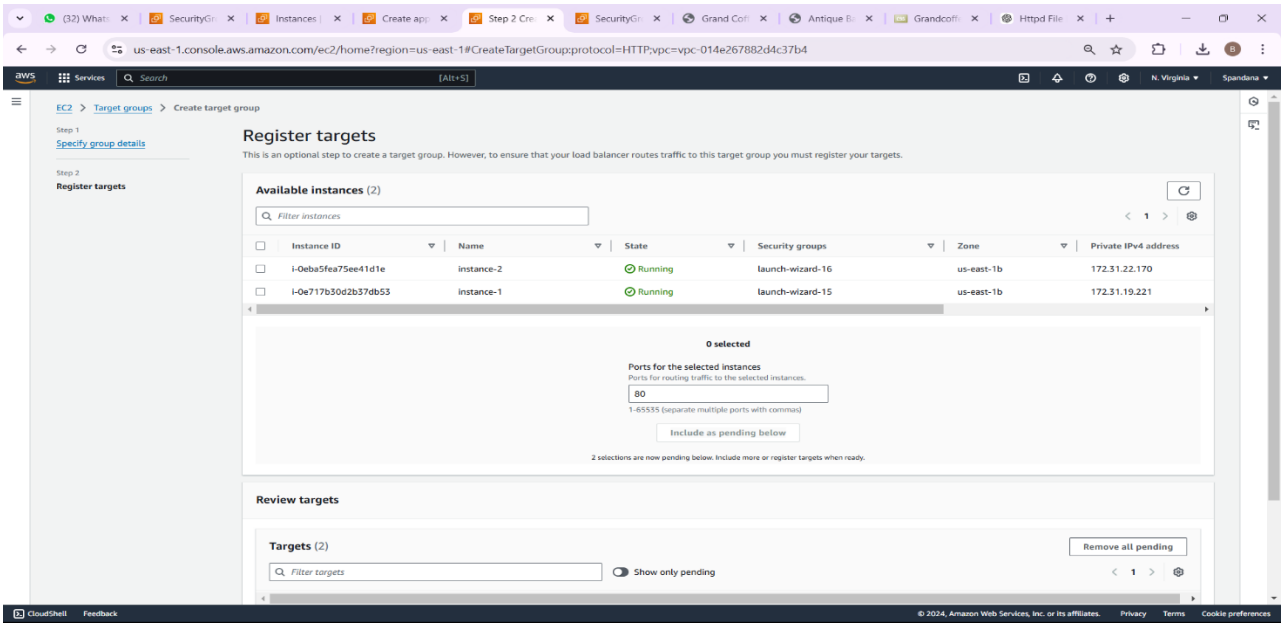


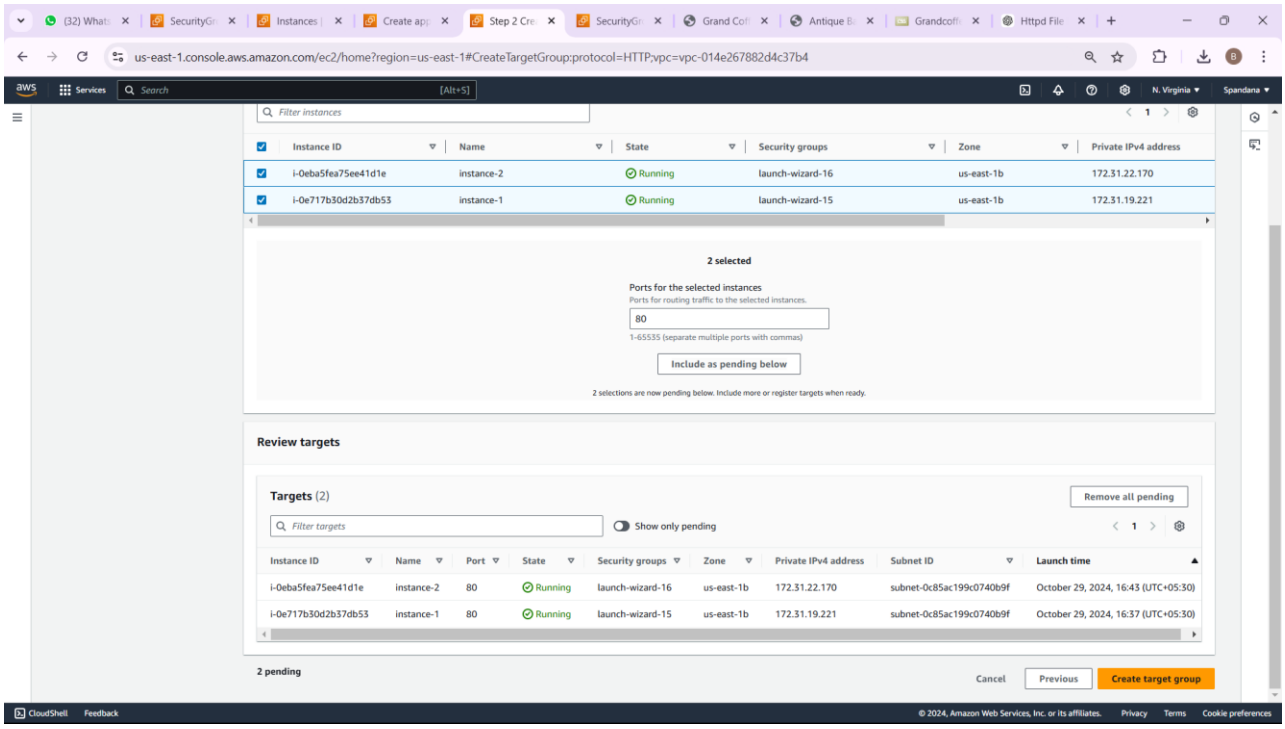
Configure Security Groups for the Load Balancer create a security group for the load balancer, allowing inbound HTTP traffic on port 80. If we have already then we can able to select it. Before creating Load balancer we need to create a target group if not we can able to create a target group directly. I am creating directly from configurations page after creating target group selecting target group.



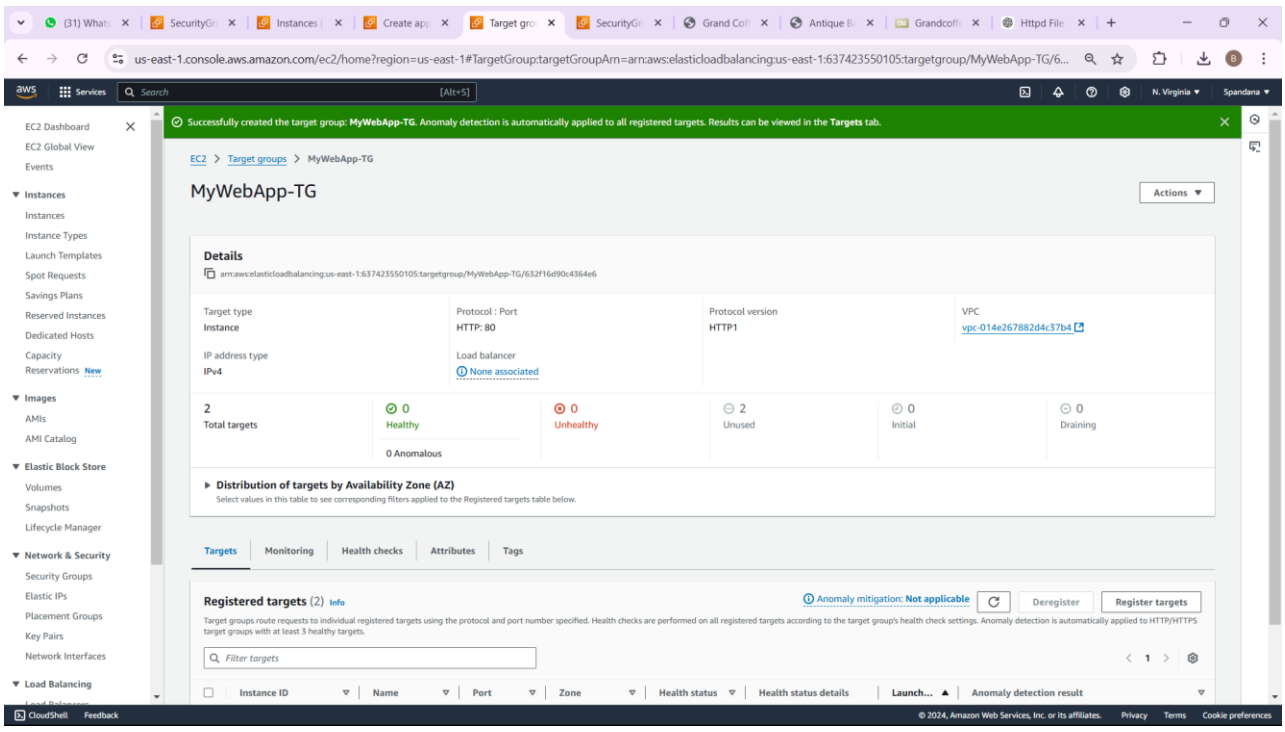
WE NEED TO SELECT THE TARGET GROUP

In the Registry targets we need to select our instances and we need to click on create target group which is shown in below images.

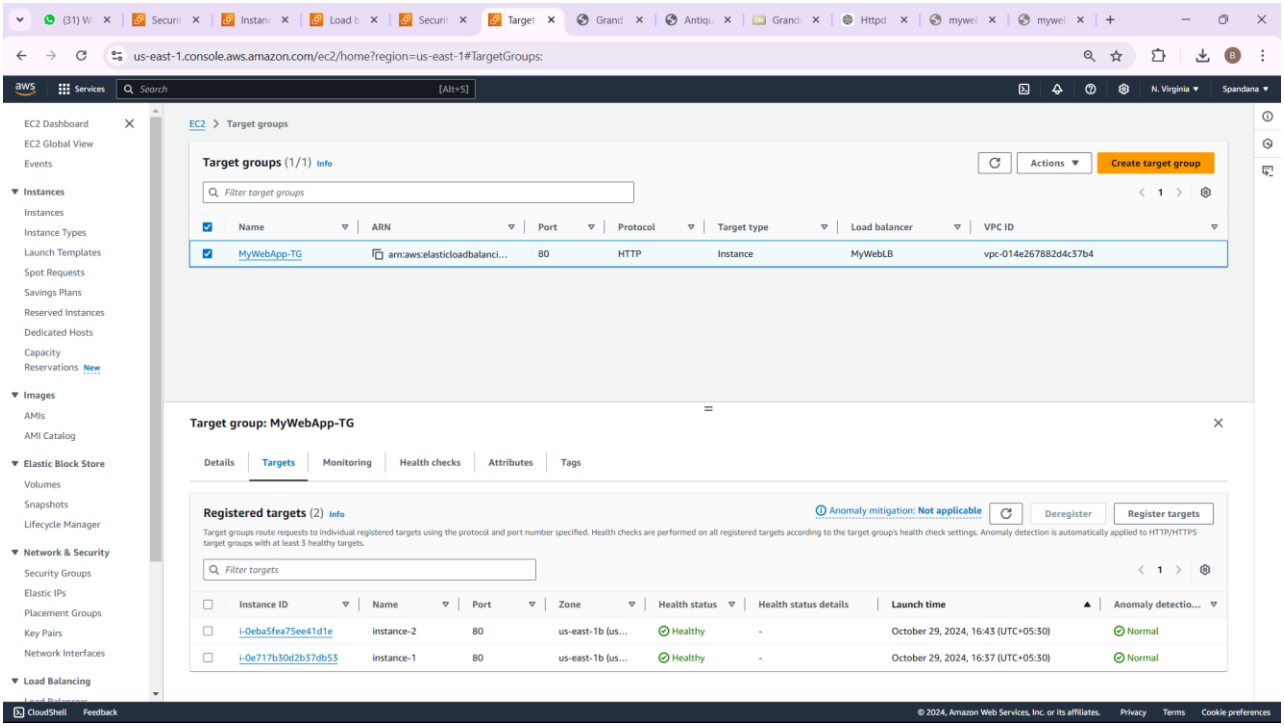




Successfully created Target group we can able to see in the below image.

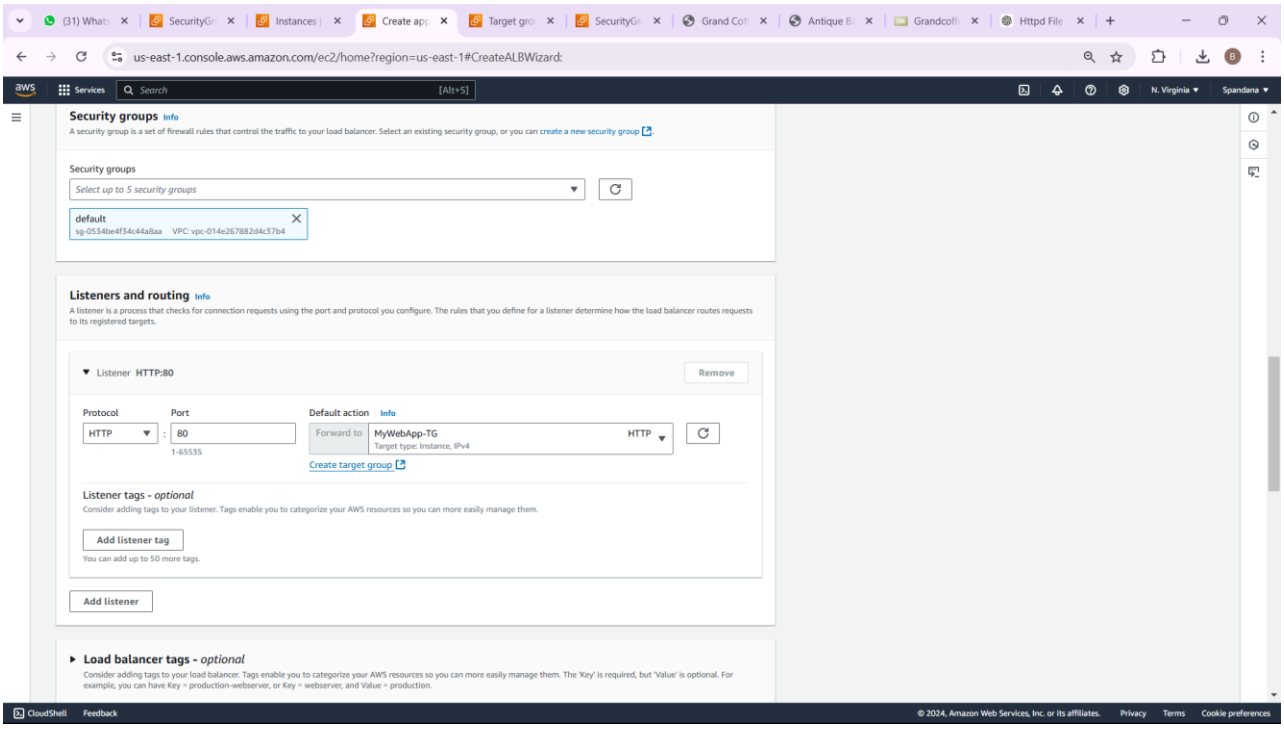


Click on target group go to targets and check our two instances are Healthy so again go to the congiguration page.

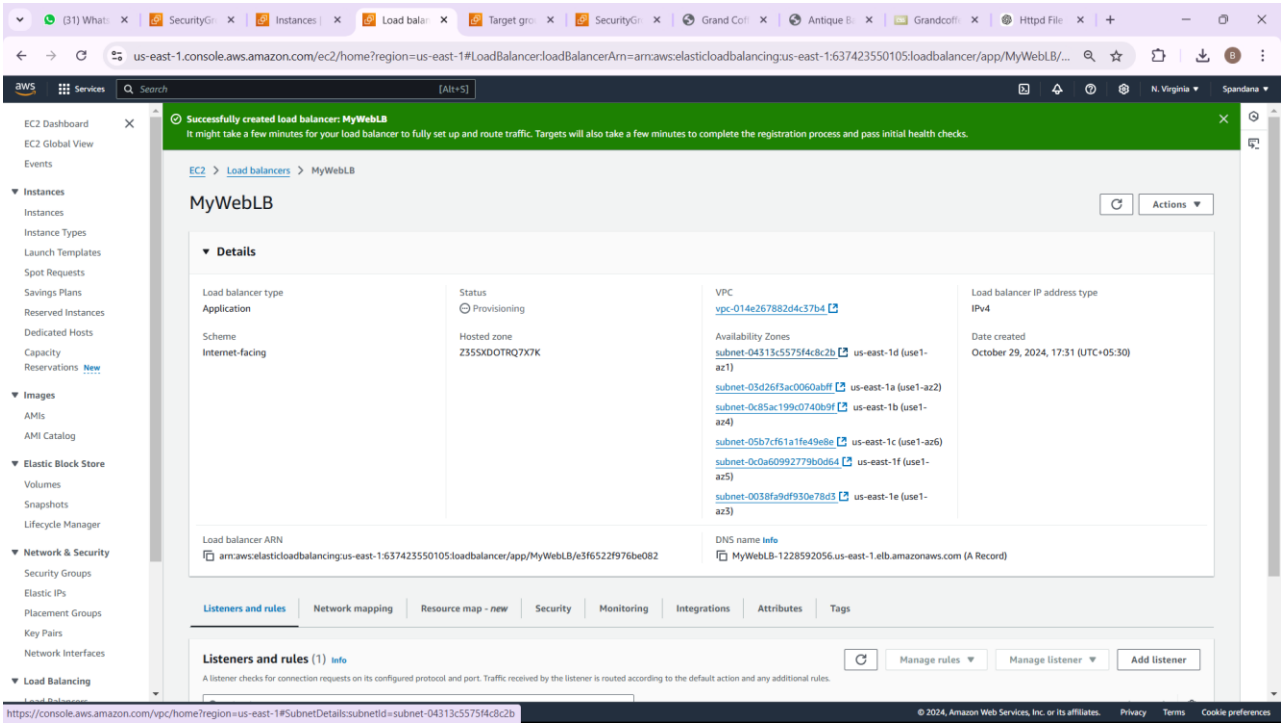


We need to assign a target group at the configuration of load balancer.

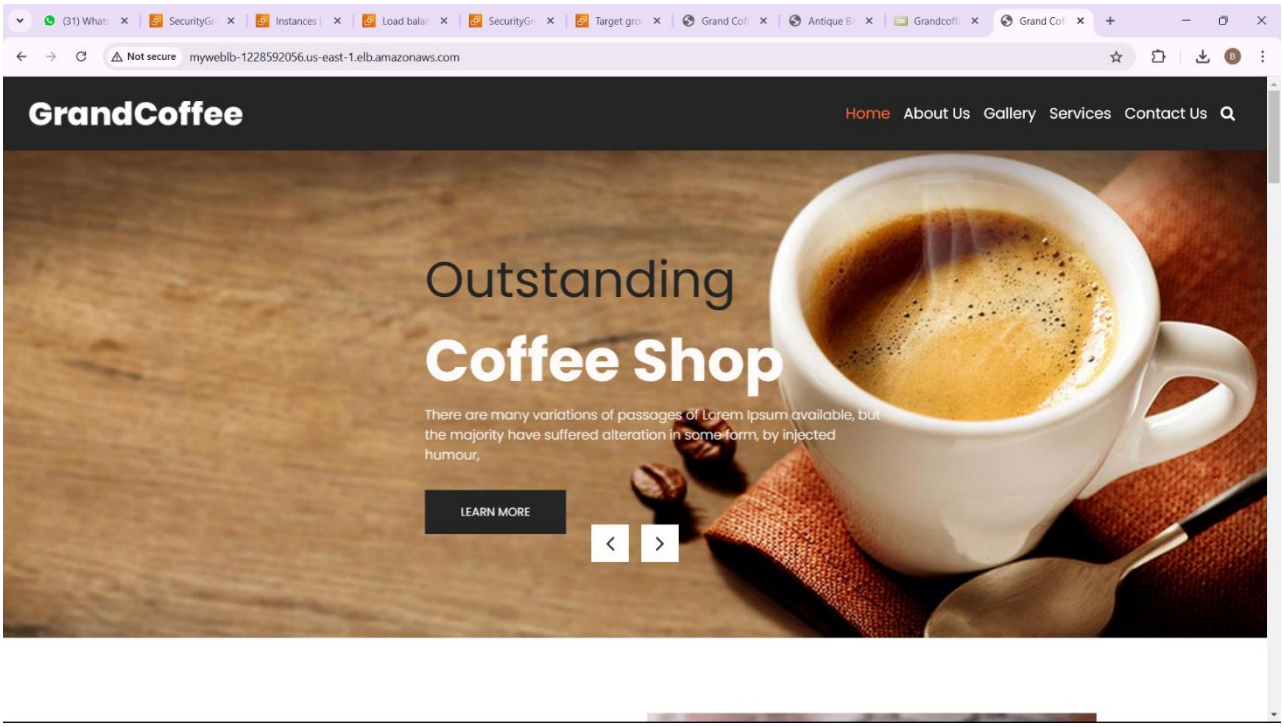
In the below image I have assigned a target group for load balancer. After that click on create Load balancer.



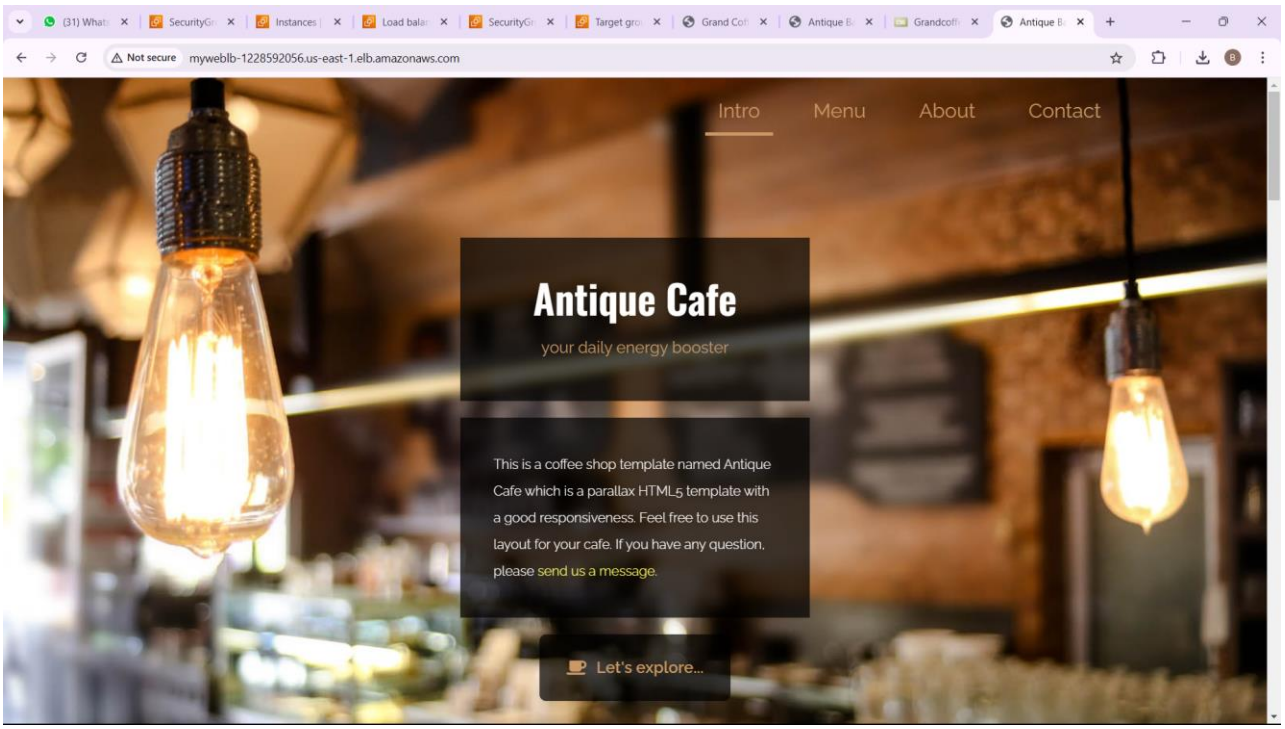
In the below image we can able to see our load balancer has been created Successfully.

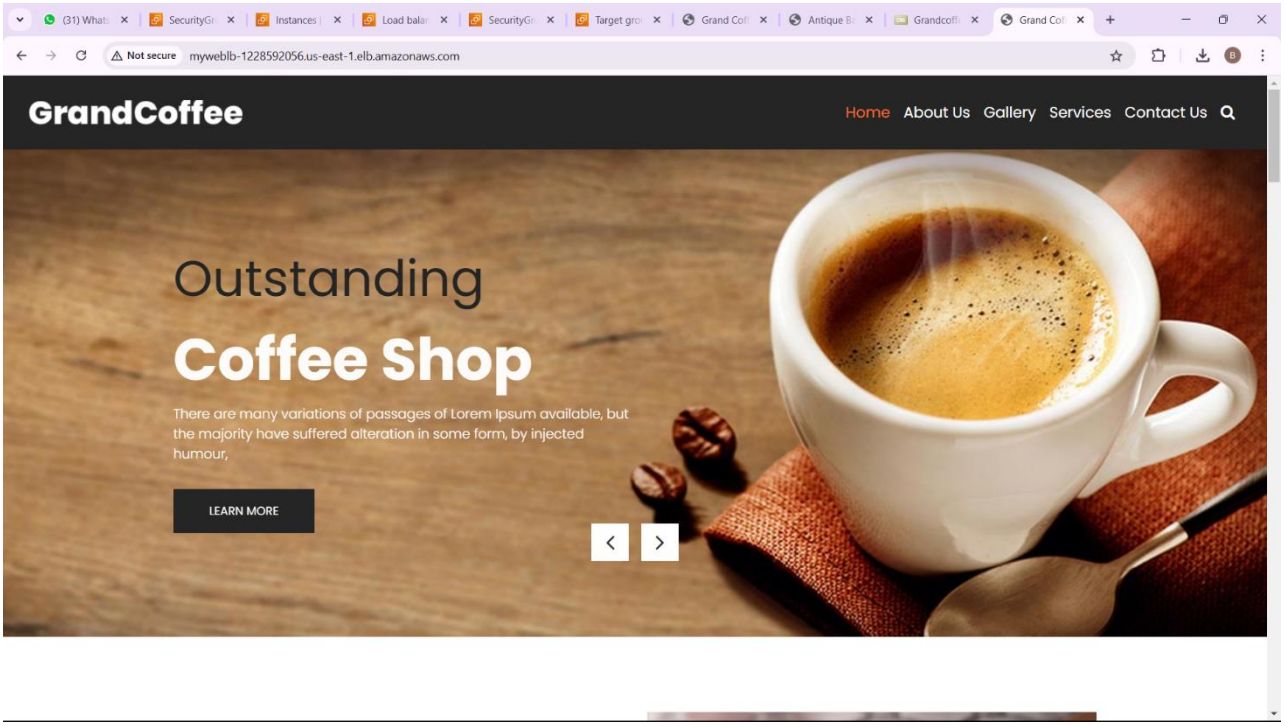


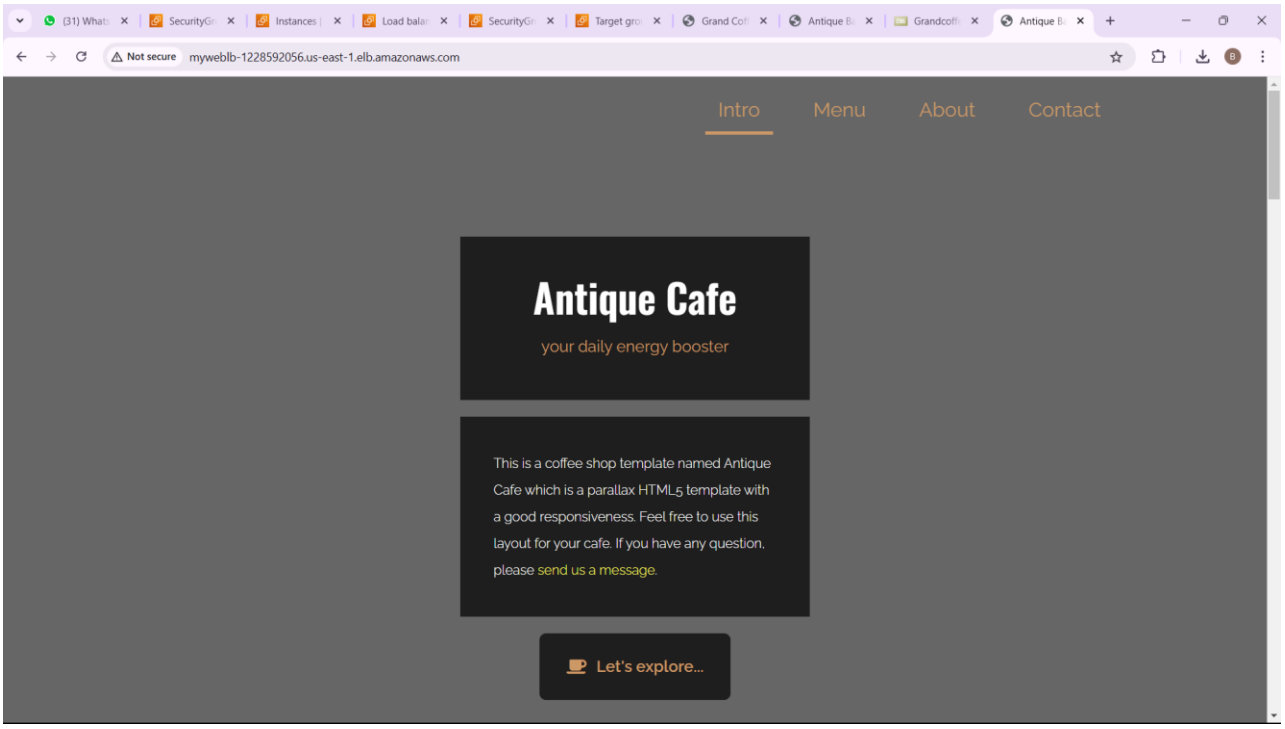
We need to copy DNS url and paste it in browser we can able to see Grand Coffee first and click on reload option provided at the top left corner.

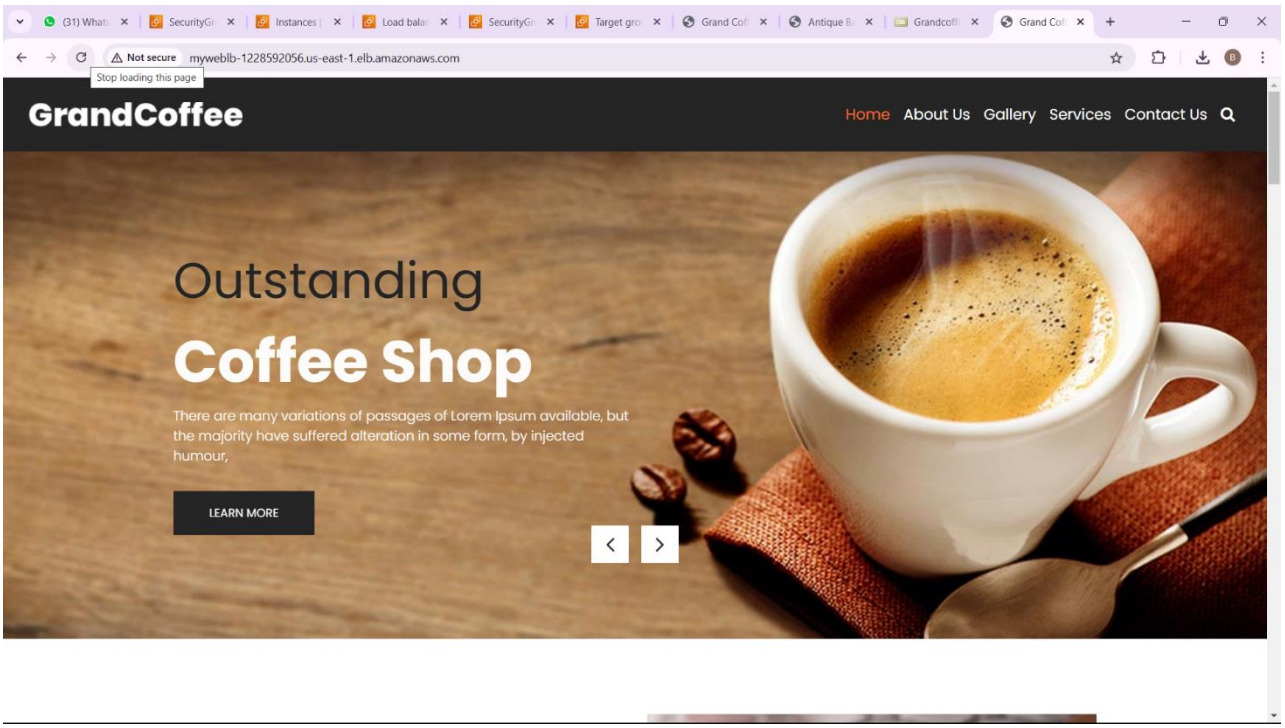


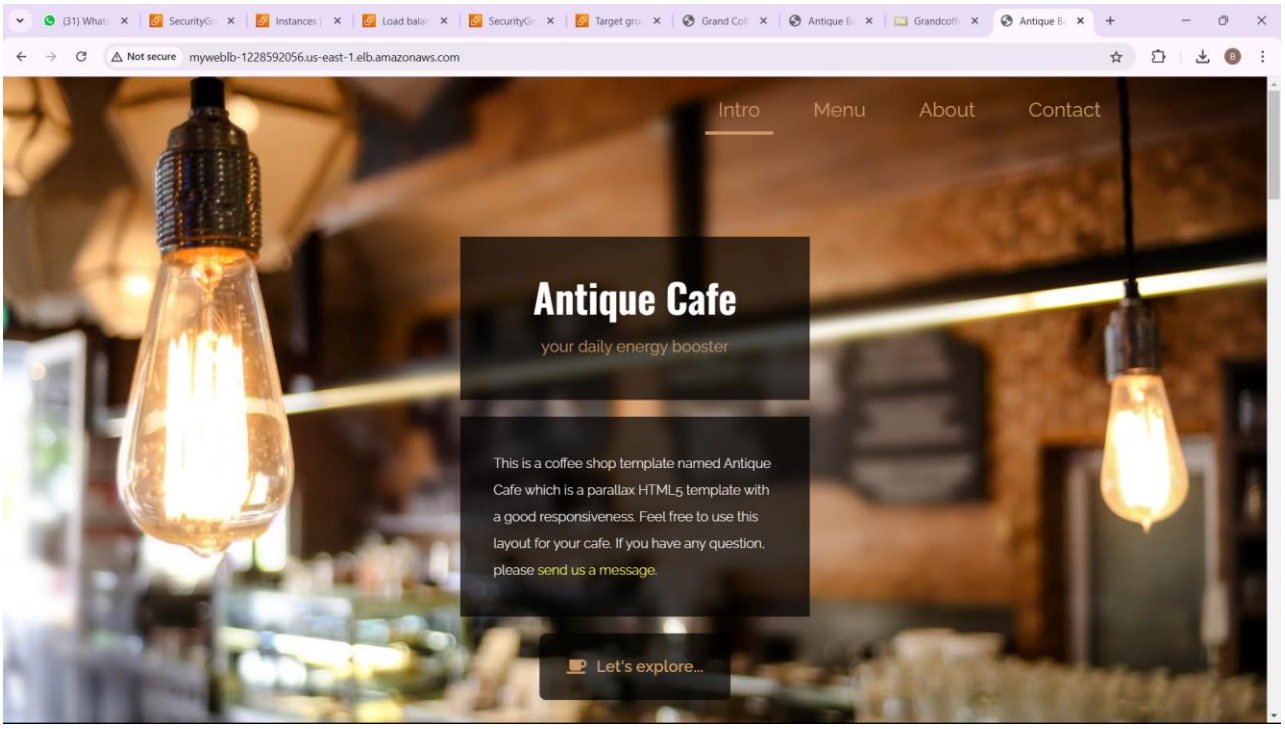
After reloading we can able to see the Antique Cafe so that repeat it two or three types load is fluctuate and we can able to see two applications alternatively. From this we can able to see Load is distributing.





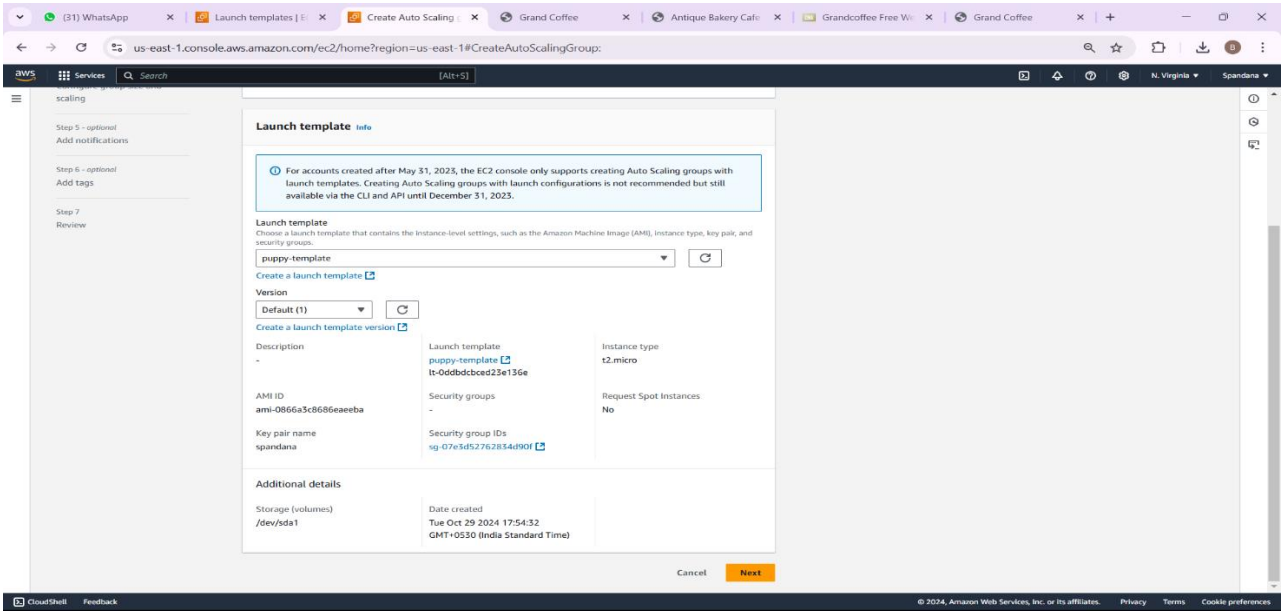

Not secure
myweblb-1228592056.us-east-1.elb.amazonaws.com
Intro
Menu
About
Contact
Antique Cafe
your daily energy booster
This is a coffee shop template named Antique Cafe which is a parallax HTML5 template with a good responsiveness. Feel free to use this layout for your cafe. If you have any question, please send us a message.
Let's explore...


Not secure
myweblb-1228592056.us-east-1.elb.amazonaws.com
Stop loading this page
GrandCoffee
Home
About Us
Gallery
Services
Contact Us
Outstanding
Coffee Shop
There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some form, by injected humour,
LEARN MORE

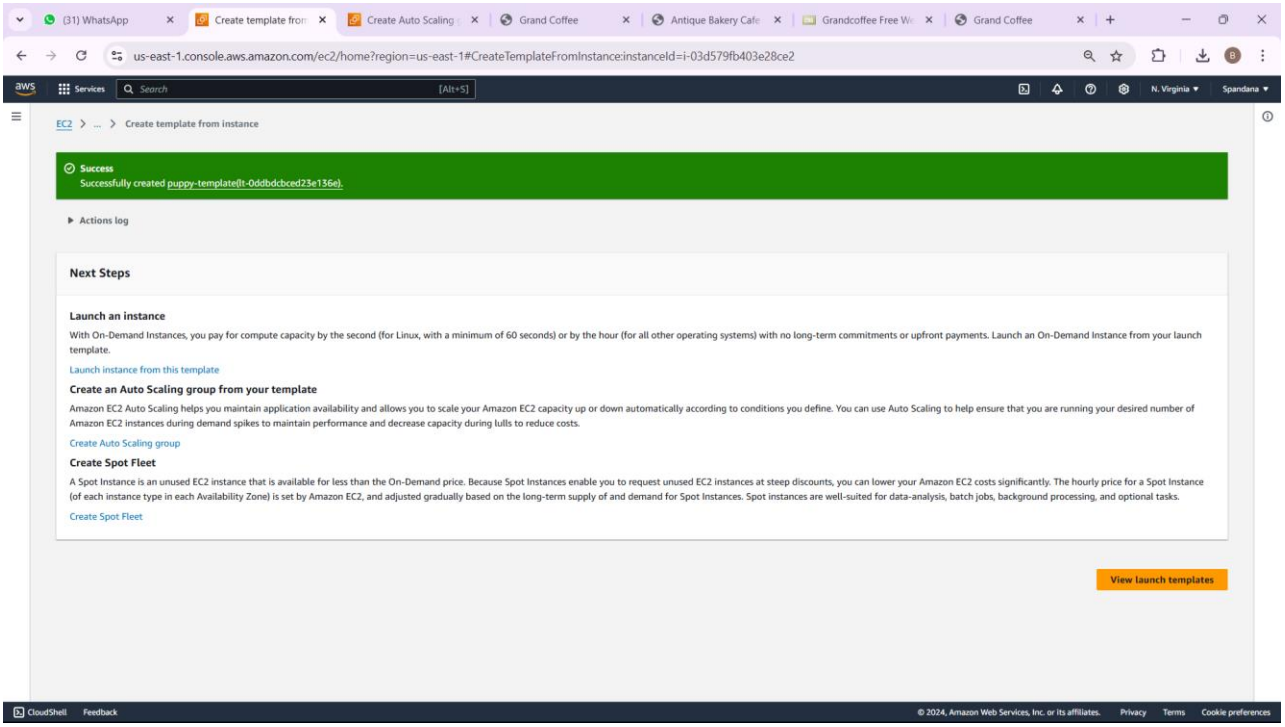


LAB 9 – AUTO SCALING GROUPS (ASG)

Create an Auto Scaling Group (ASG) go to Auto Scaling Groups in the EC2 Dashboard. Click Create Auto Scaling group. Select Launch template and choose the template you created earlier. Auto Scaling group name: Give it a descriptive name, like UbuntuAutoScalingGroup. VPC and Subnet Select the VPC and the subnets where you want the instances to launch.

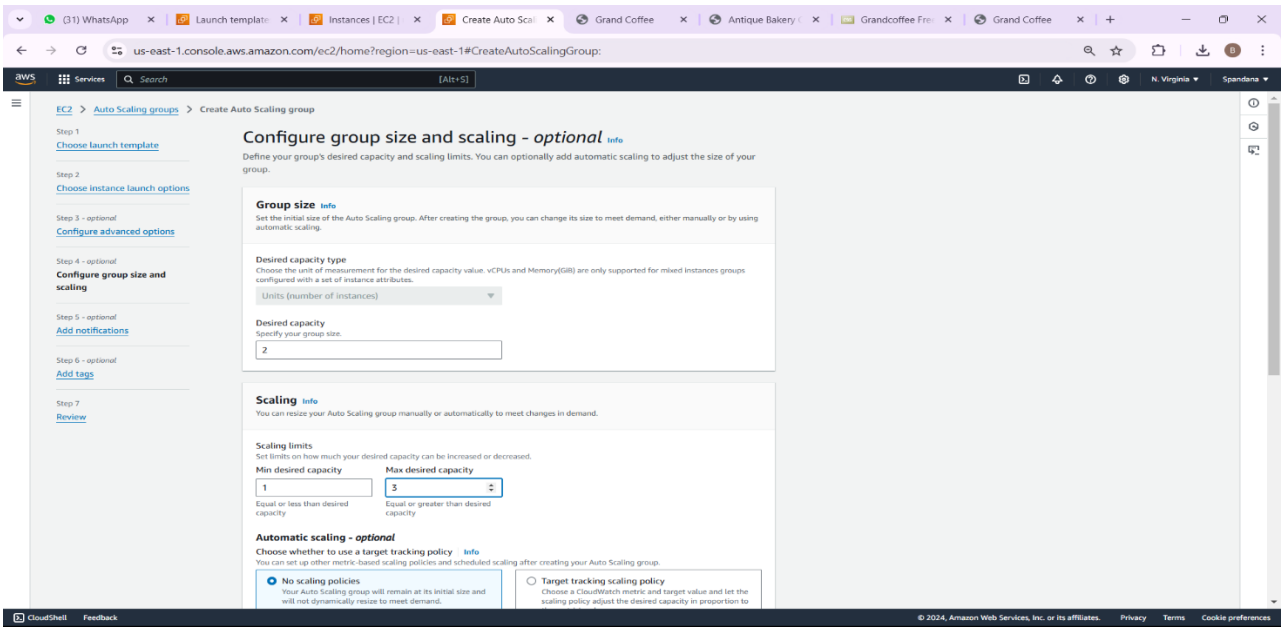


Successfully template has been created.

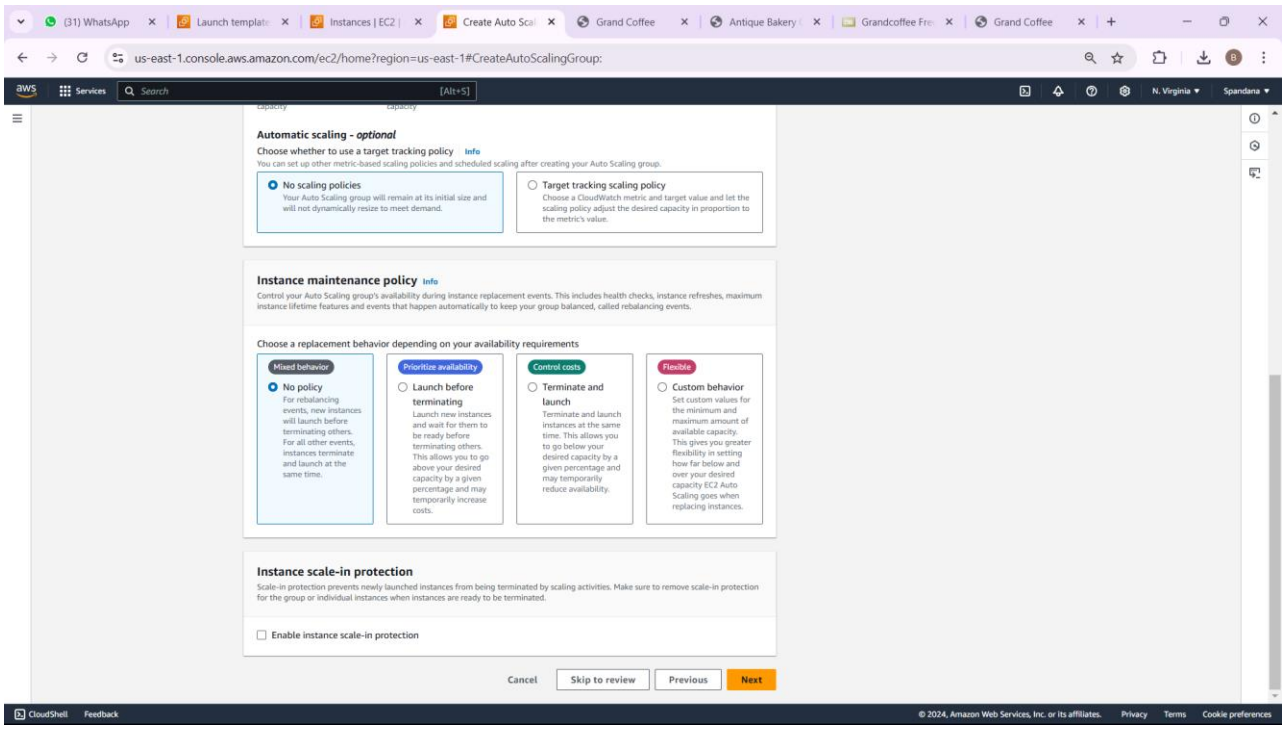


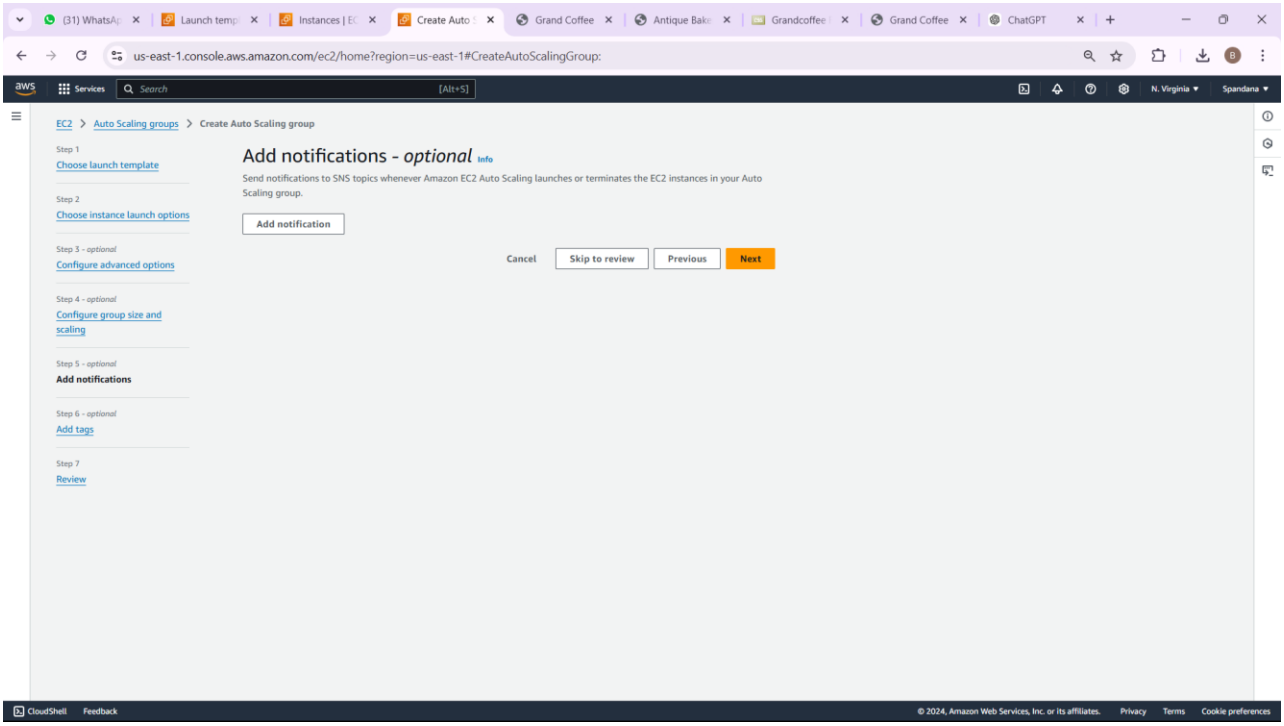
Configure ASG Size and Scaling Policies Set the Desired Capacity:

- Minimum capacity: 1
- Desired capacity: 2
- Maximum capacity: 3

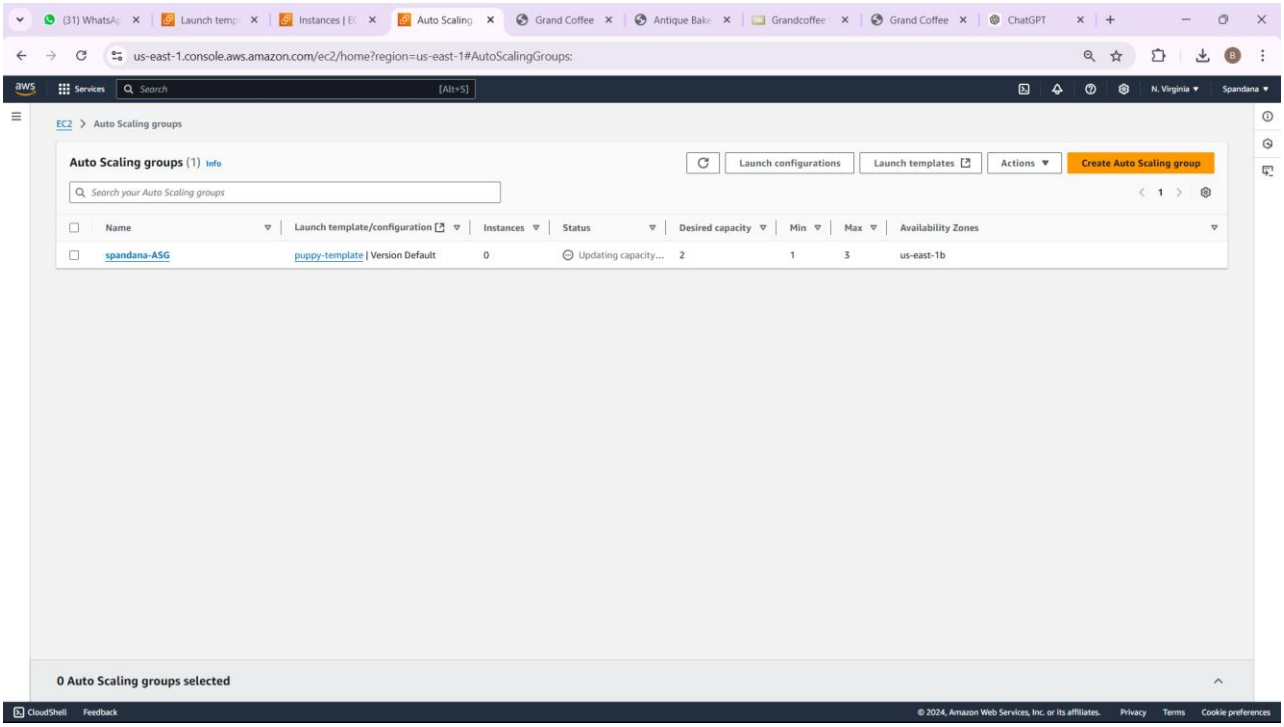


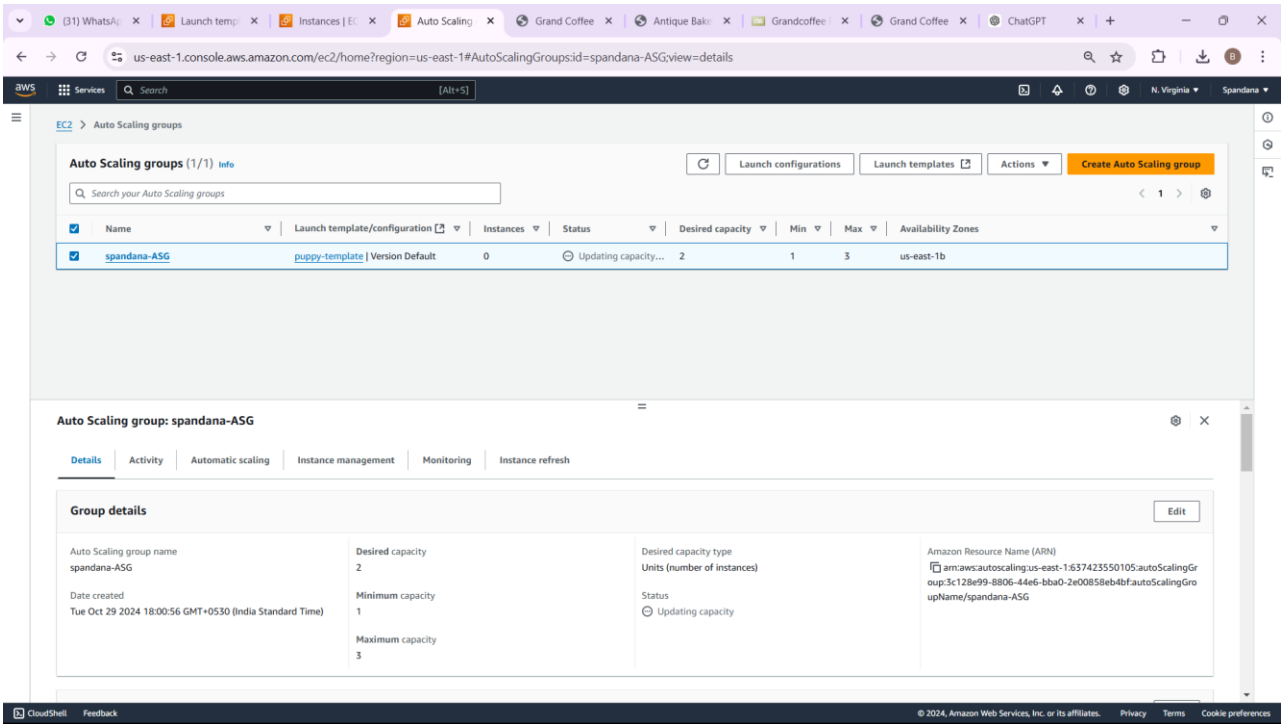
Scaling policies for simplicity, you can use the default setting, which keeps instances at the desired capacity. If you want more control, you can set policies based on CPU utilization or other metrics.



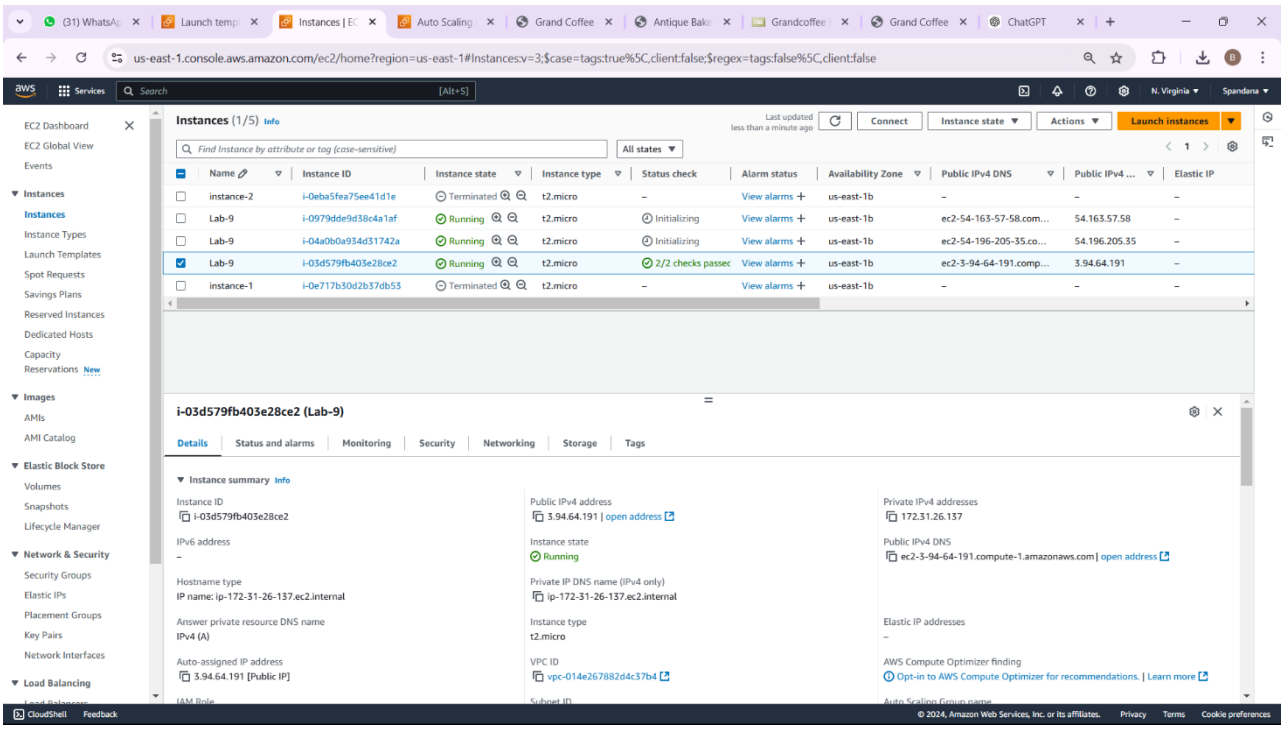


Review and create Auto Scaling Group below we can see that Auto Scaling Group has been created.

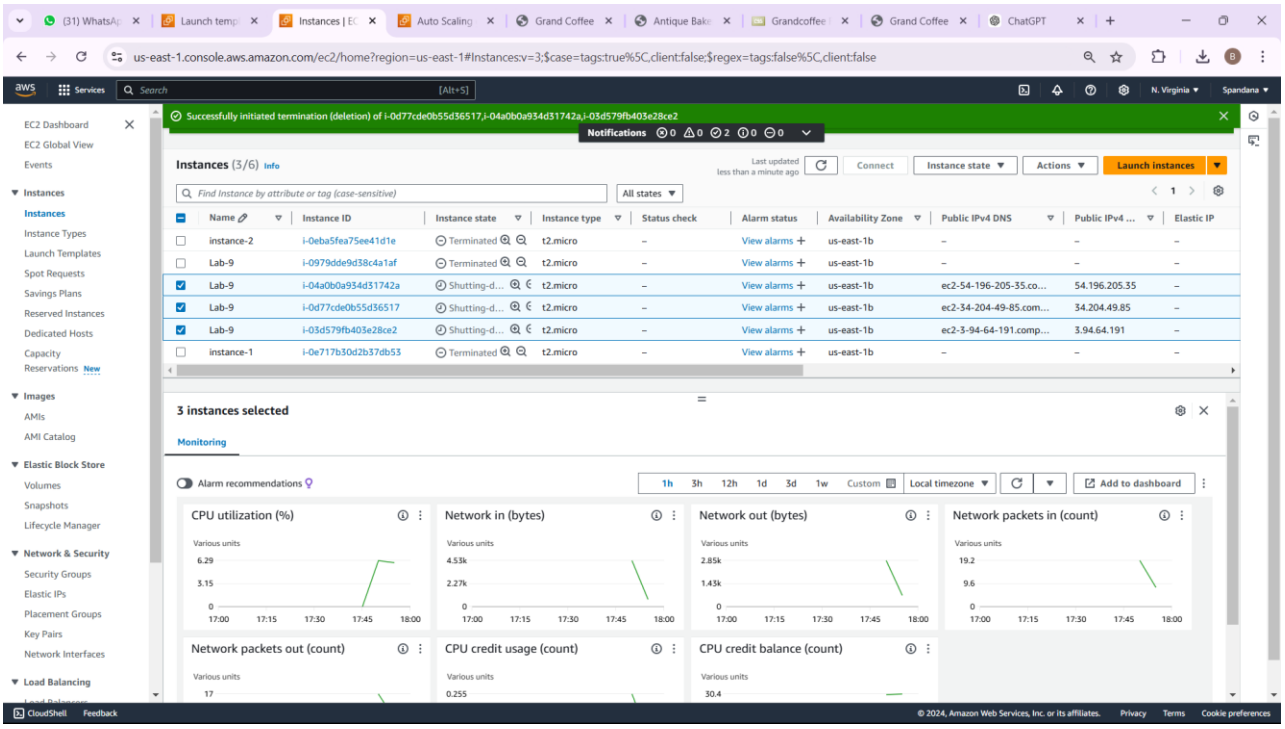




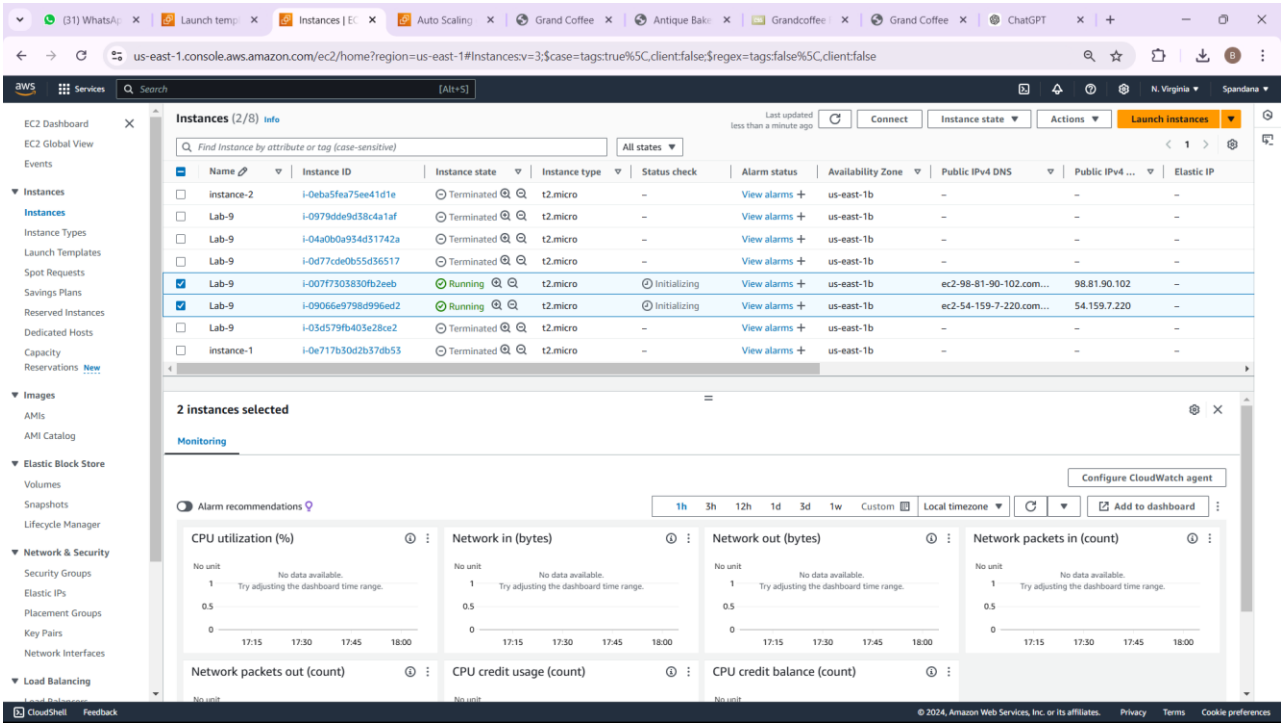
Verify Auto Scaling Behaviour after the ASG is created WS will automatically launch instances based on the desired capacity.



Delete existing instances and monitor.



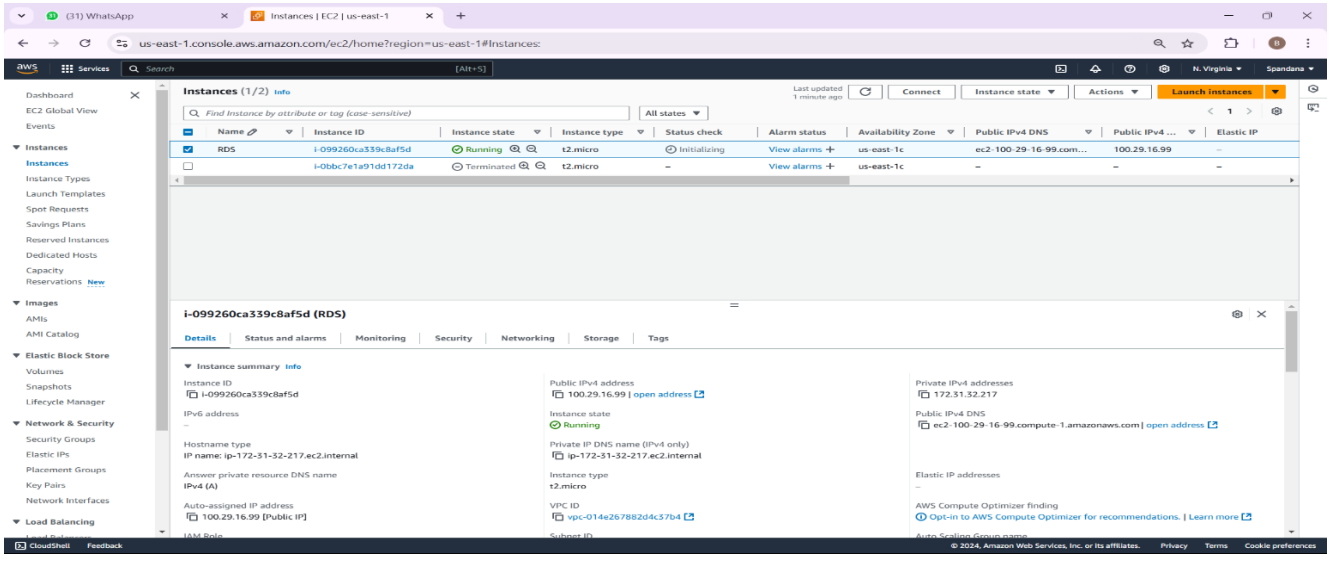
Auto Scaling Group start launching additional instances to reach the capacity.



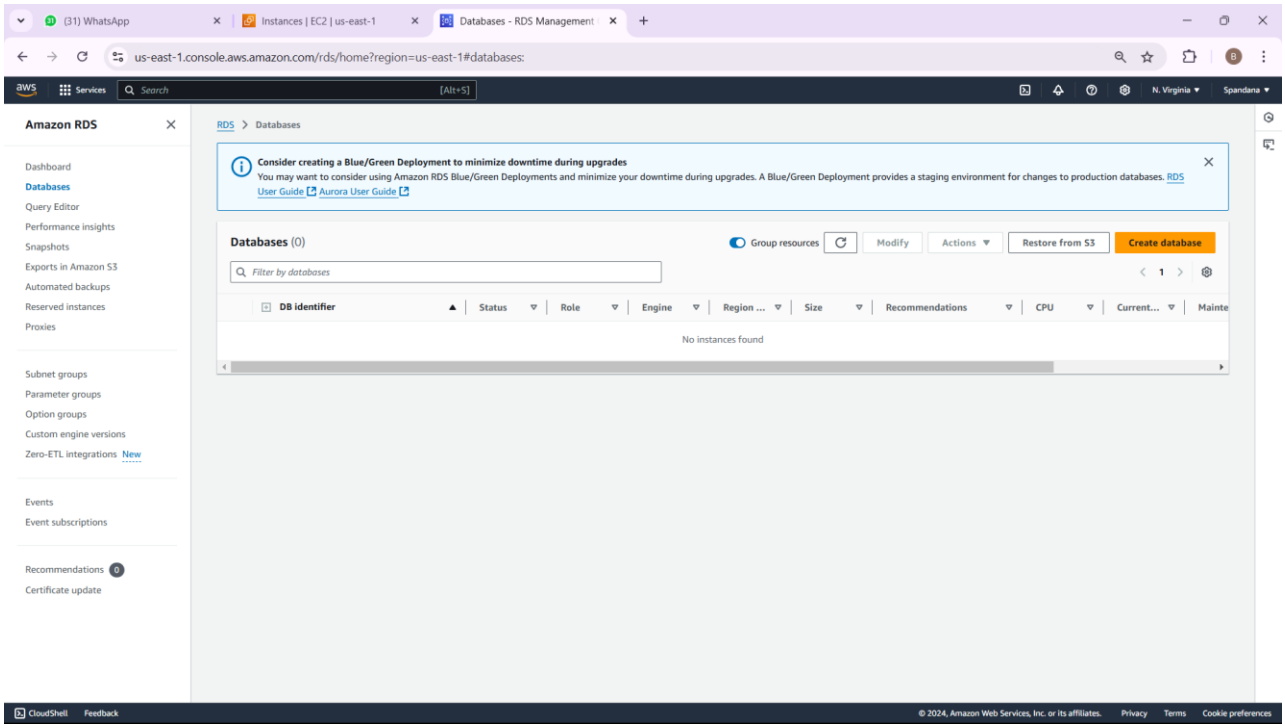
LAB-10 RDS

Amazon RDS is a managed database service supporting multiple engines (e.g., MySQL, PostgreSQL). Itautomates backups, patching, and scaling, while offering high availability with Multi-AZ deployment and robust security features.

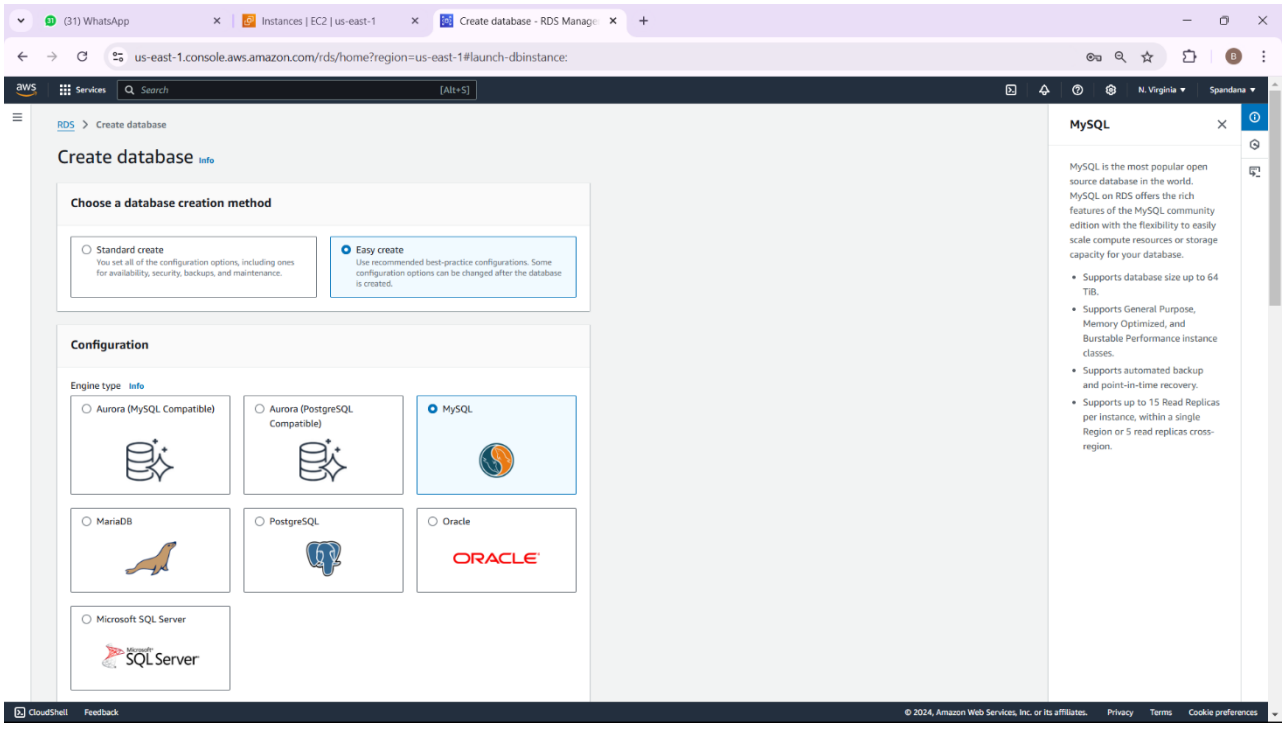
Firstly we have to create one EC2 instance.



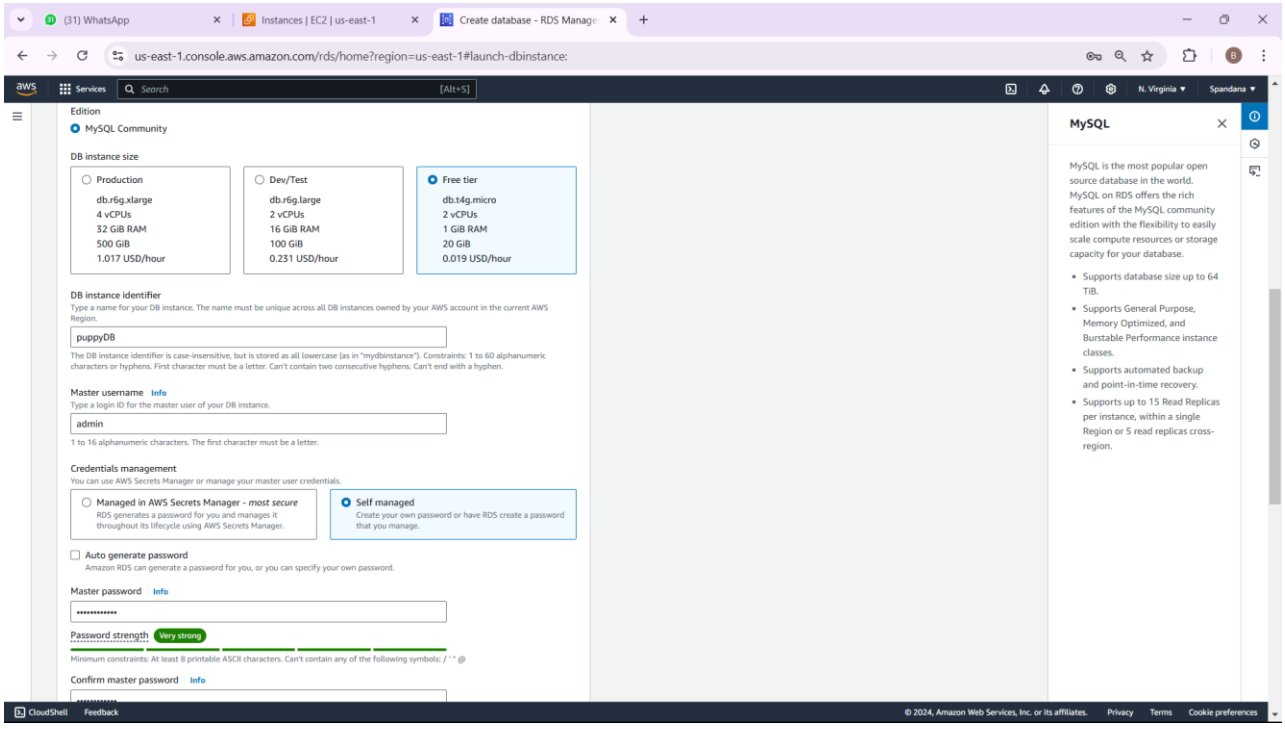
In the console search for RDS service and click on Databases and click on create databases.



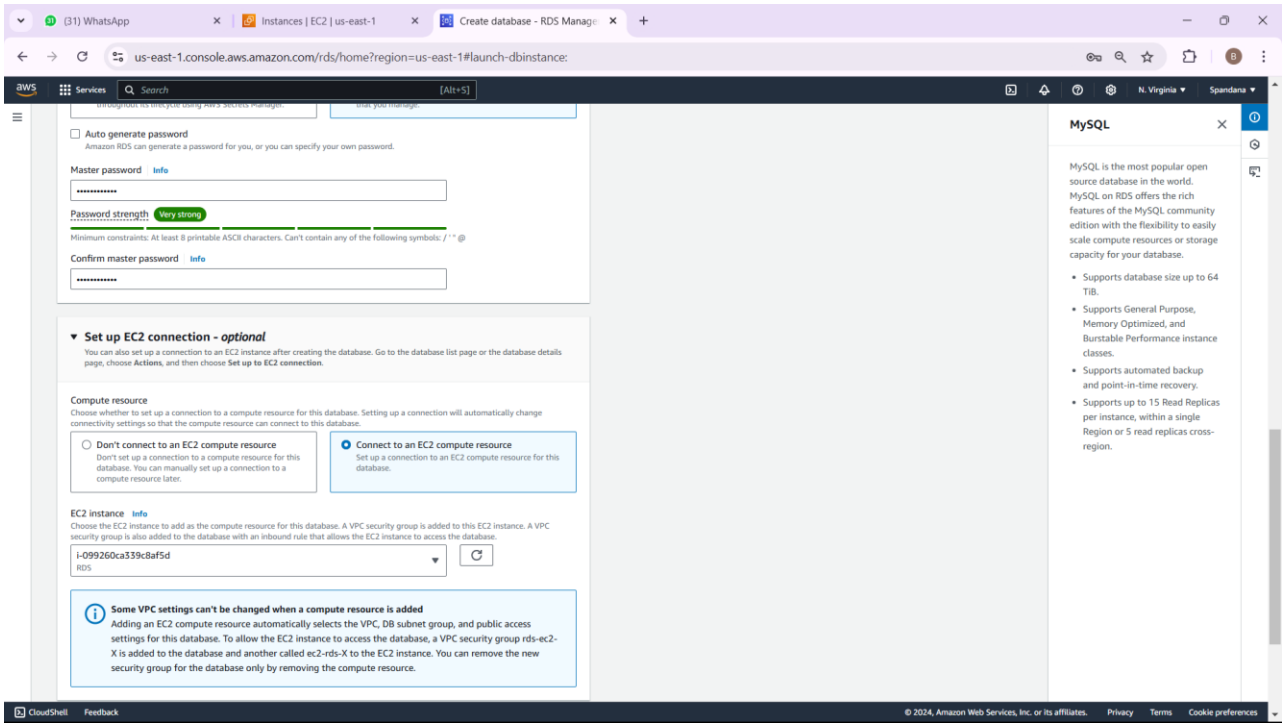
We have to give configurations for databases choose data creation method and choose Database Engine as MYSQL .



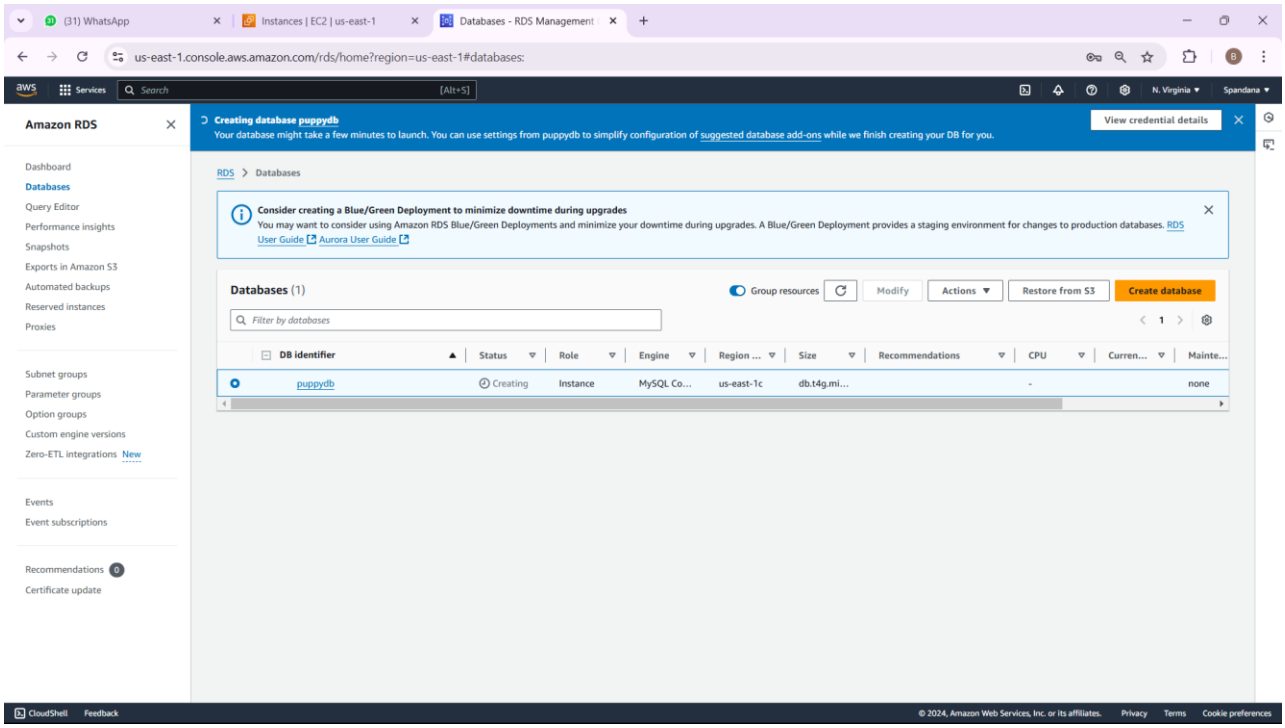
Select DB instance specifications and provide DB instance identifier optional and Provide Master username and Master password for security purpose.



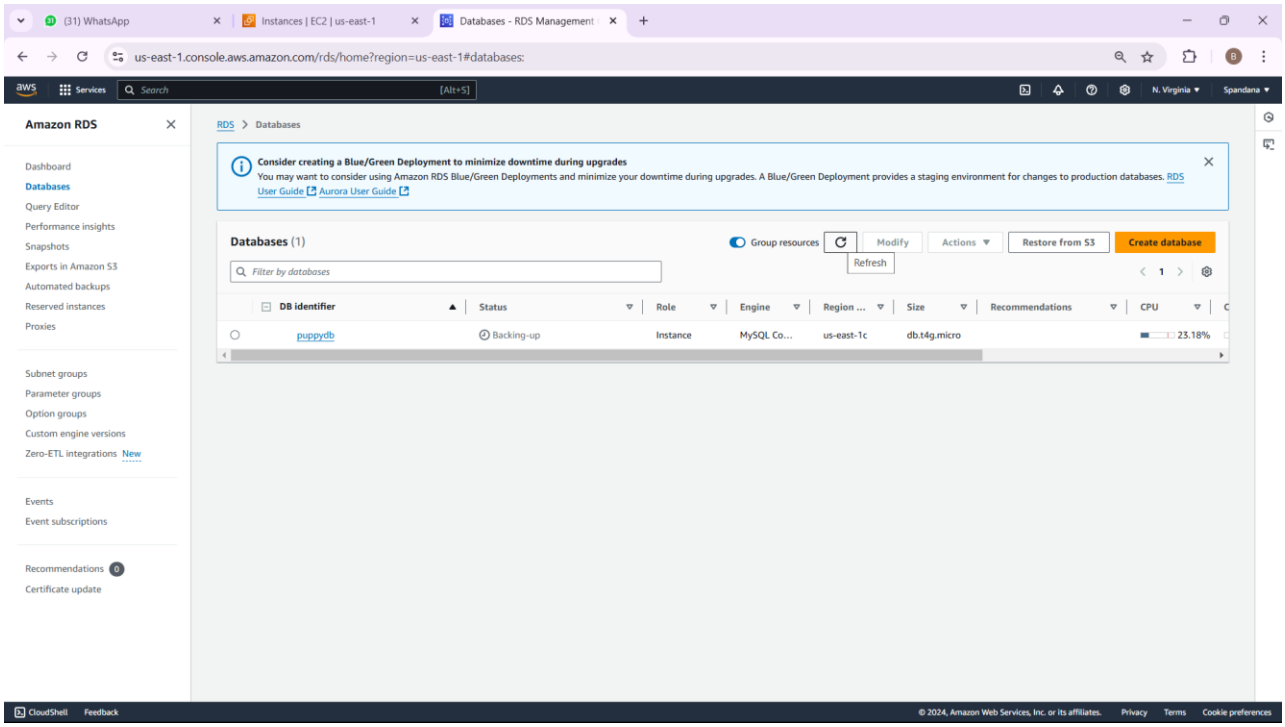
Choose EC2 instance and computer resource as Connect to an EC2 Compute resource after that click on create database.



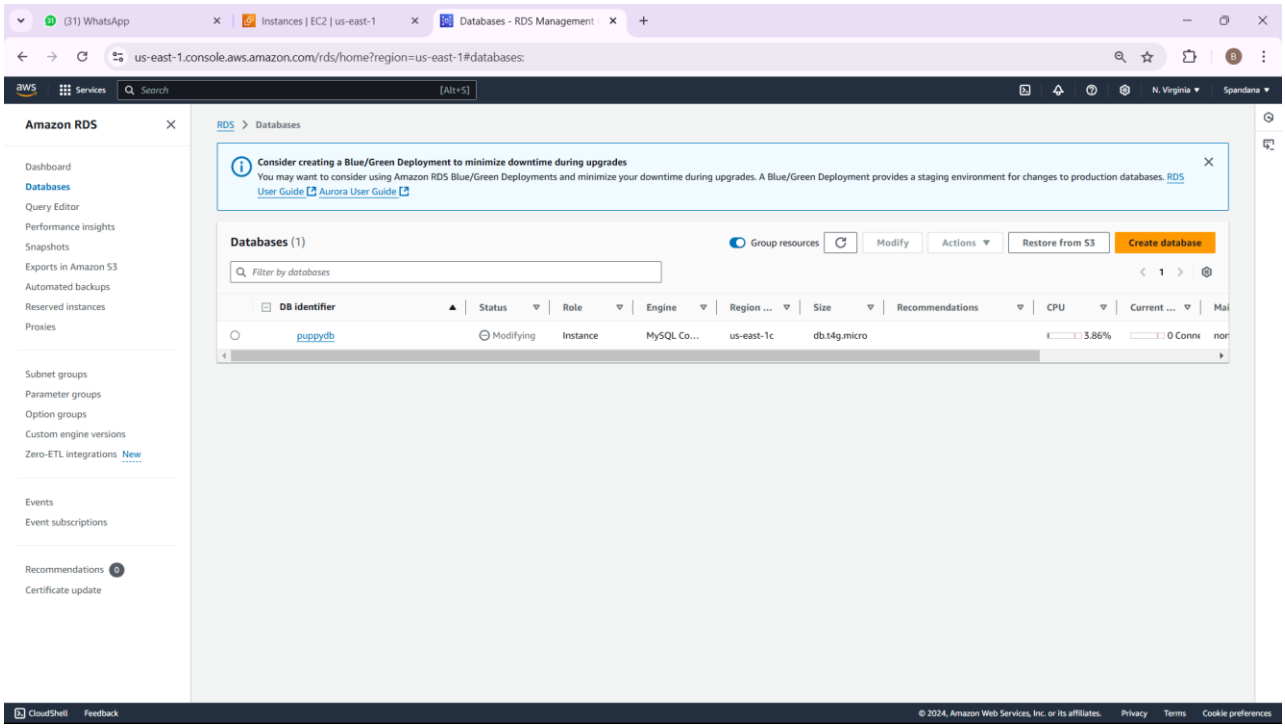
Database has been creating in the below image check the status. The status of the below image is creating.



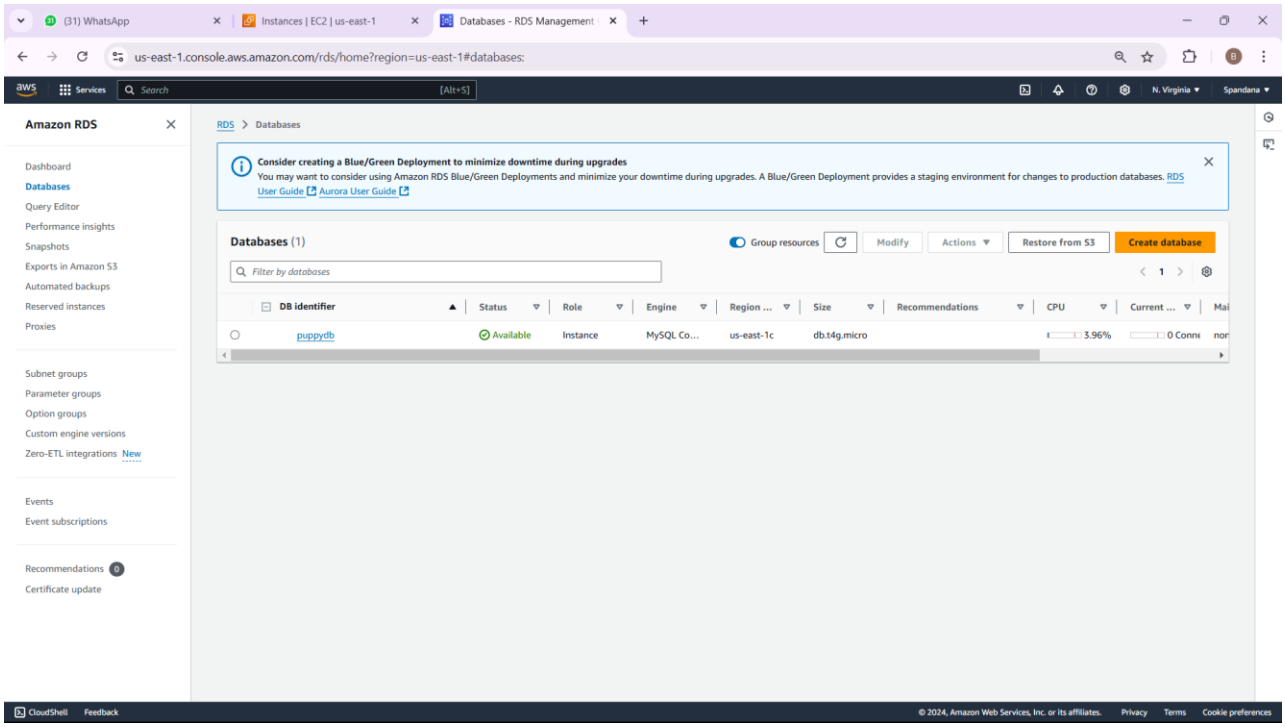
The status of the below image is Backing-up.



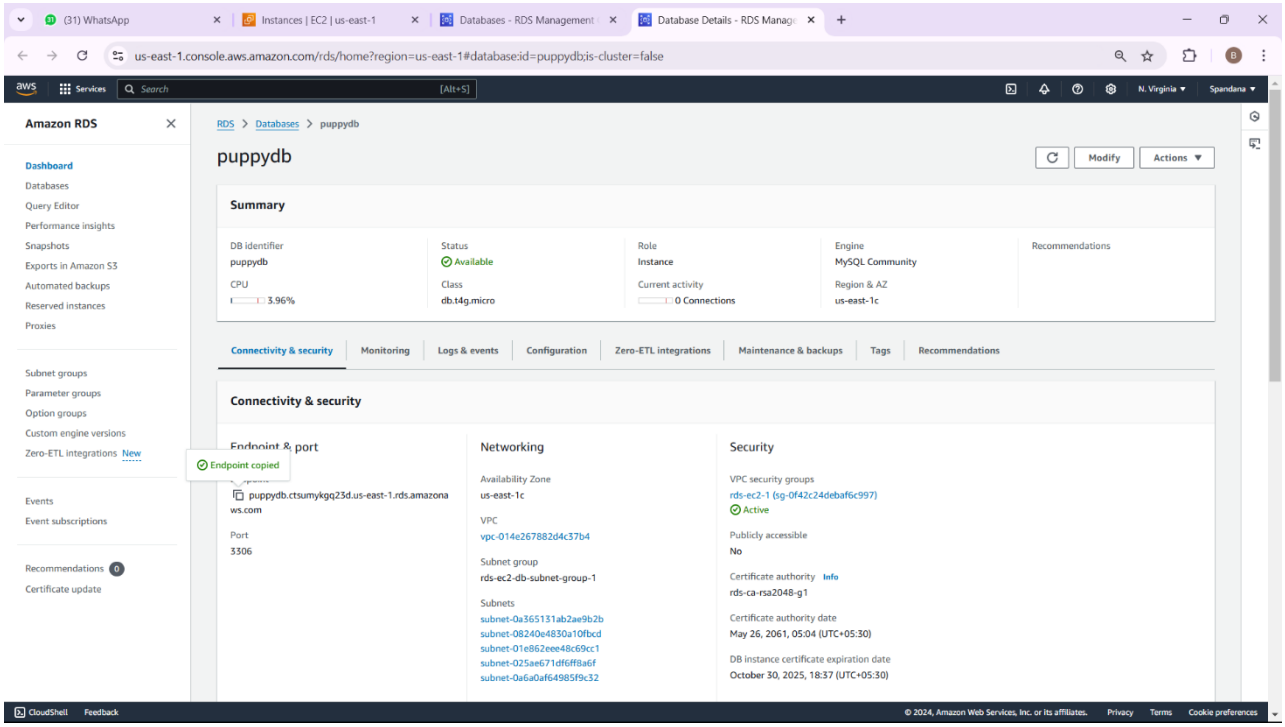
The status of the below image is Modifying.



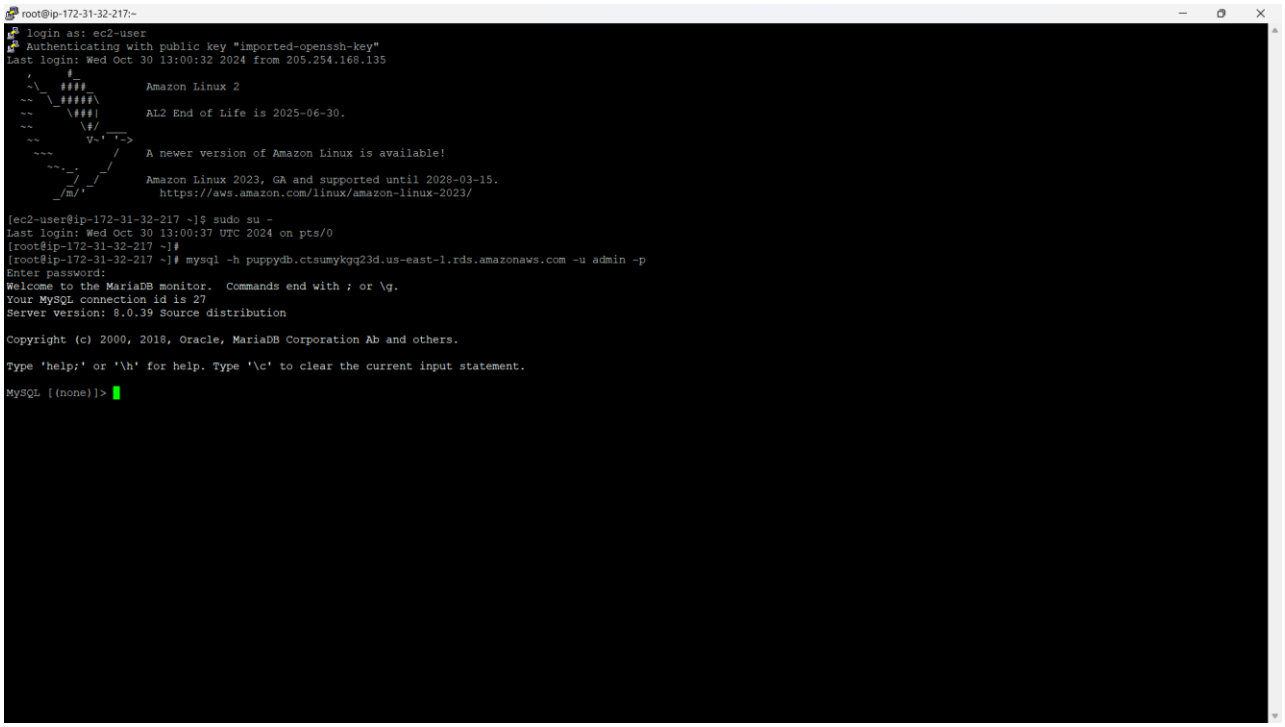
At last the database status is Available.



Click in the database it will open the below page and copy the Endpoint & port and paste it in PuTTY or terminal using command.



In Terminal install MySQL and using mysql -h <RDS-Endpoint> -u admin -P and provide password so we can able to access this RDS database from EC2 instance.

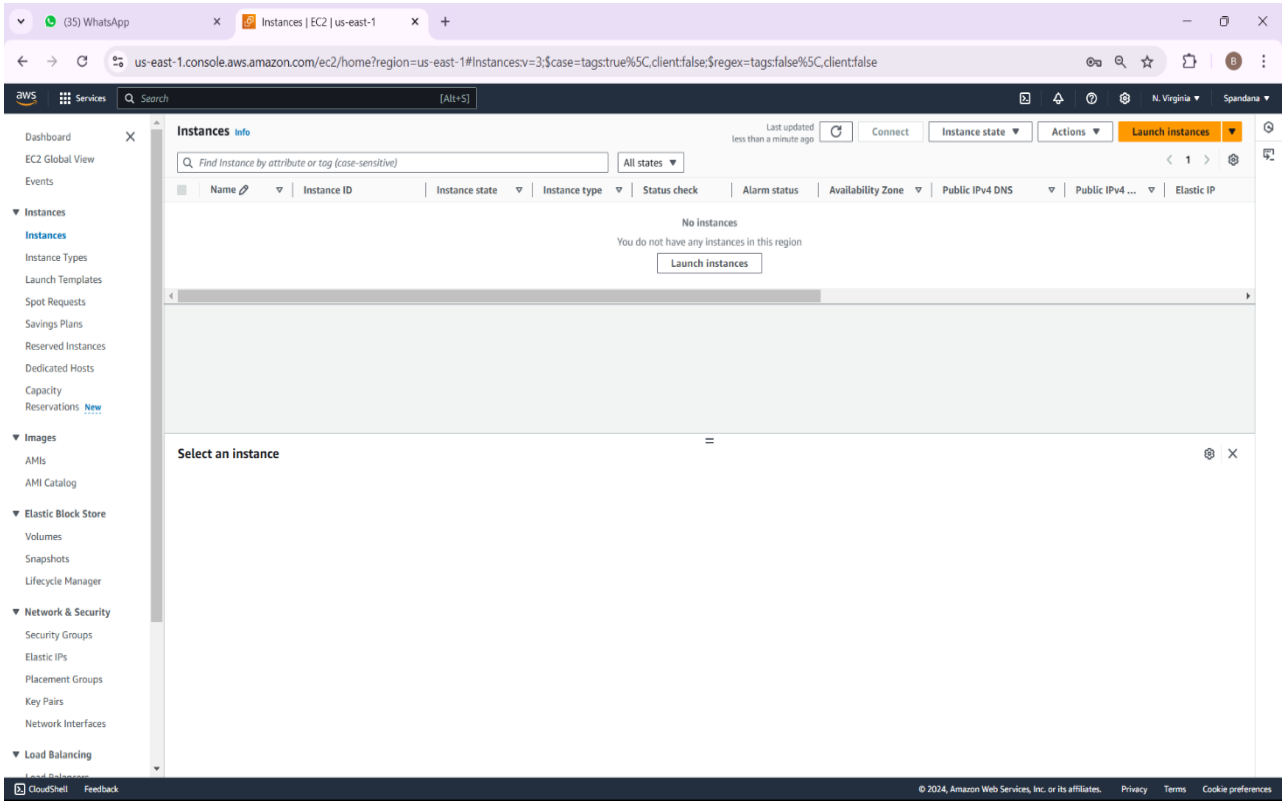


MINI PROJECT – 2

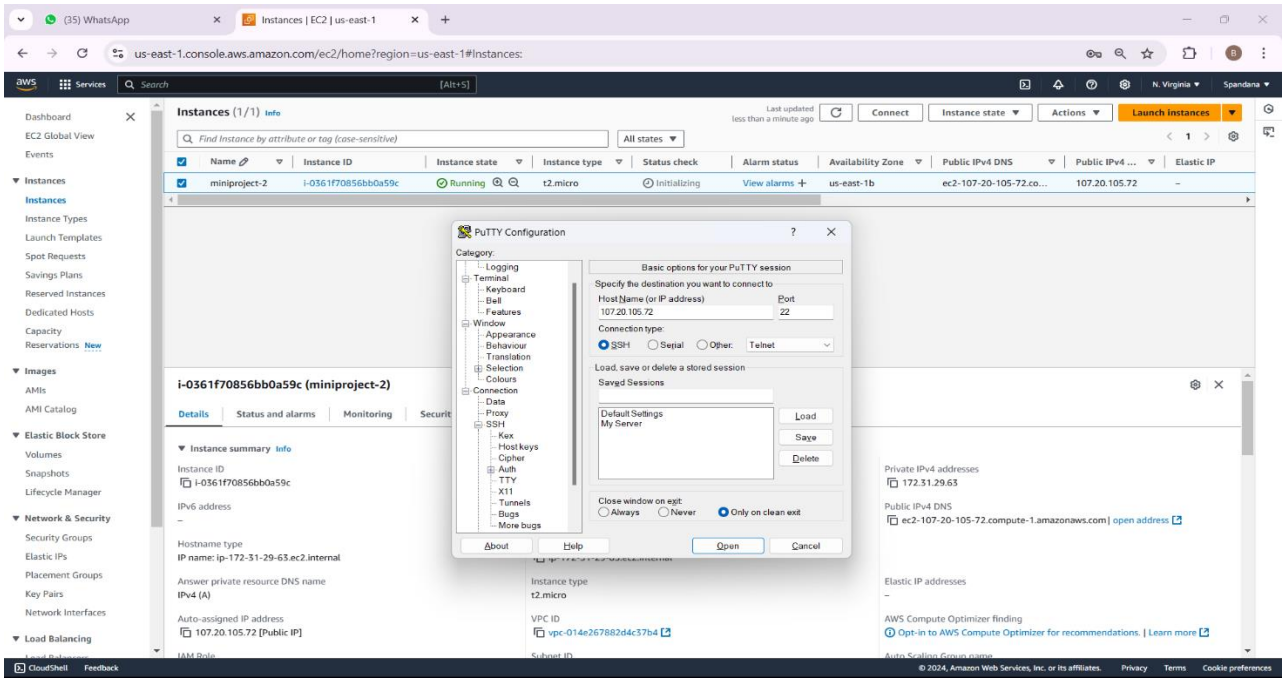
GIT-VERSION CONTROL SYSTEM(VCS) BY USING AMAZON WEB SERVICES

LAB-1 CREATING INSTANCE

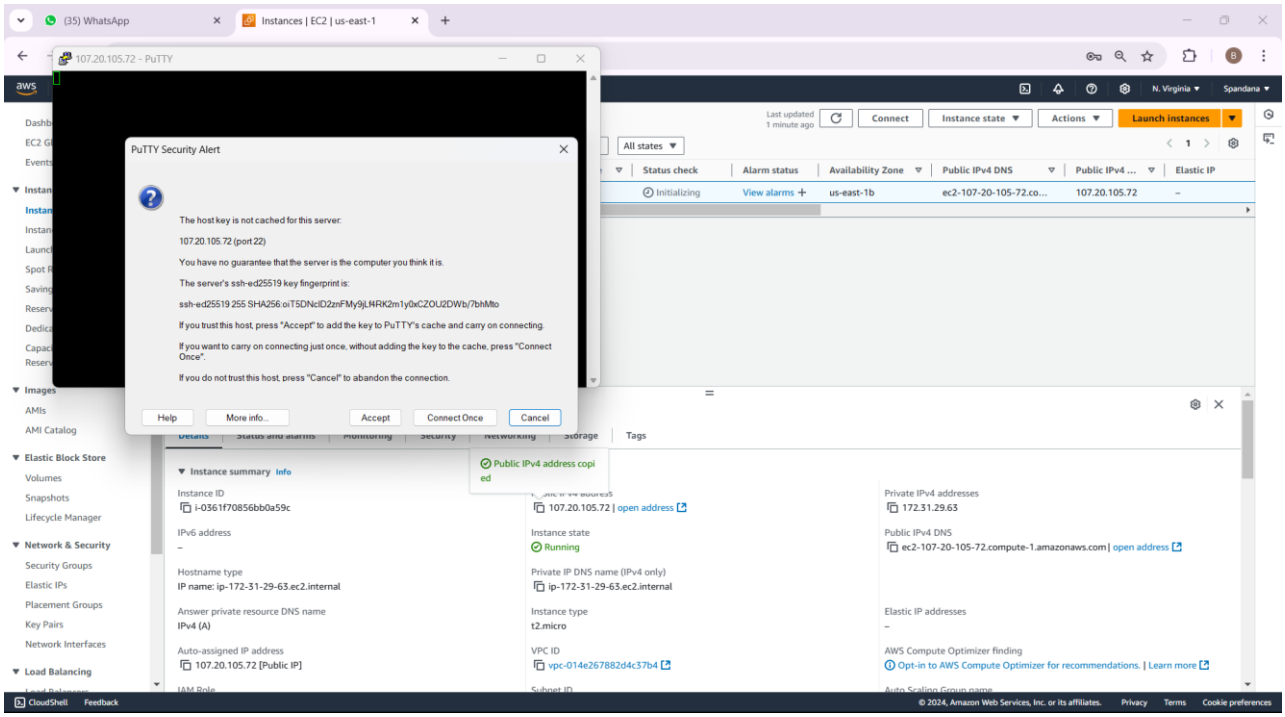
Go to AWS Management console and navigate to EC2 in the AWS console search for EC2 and select it click on launch instance and give instance name select IAM based on preference choose instance type t2.micro key pair security group allow SSH access from IP address once configured click on launch instance.



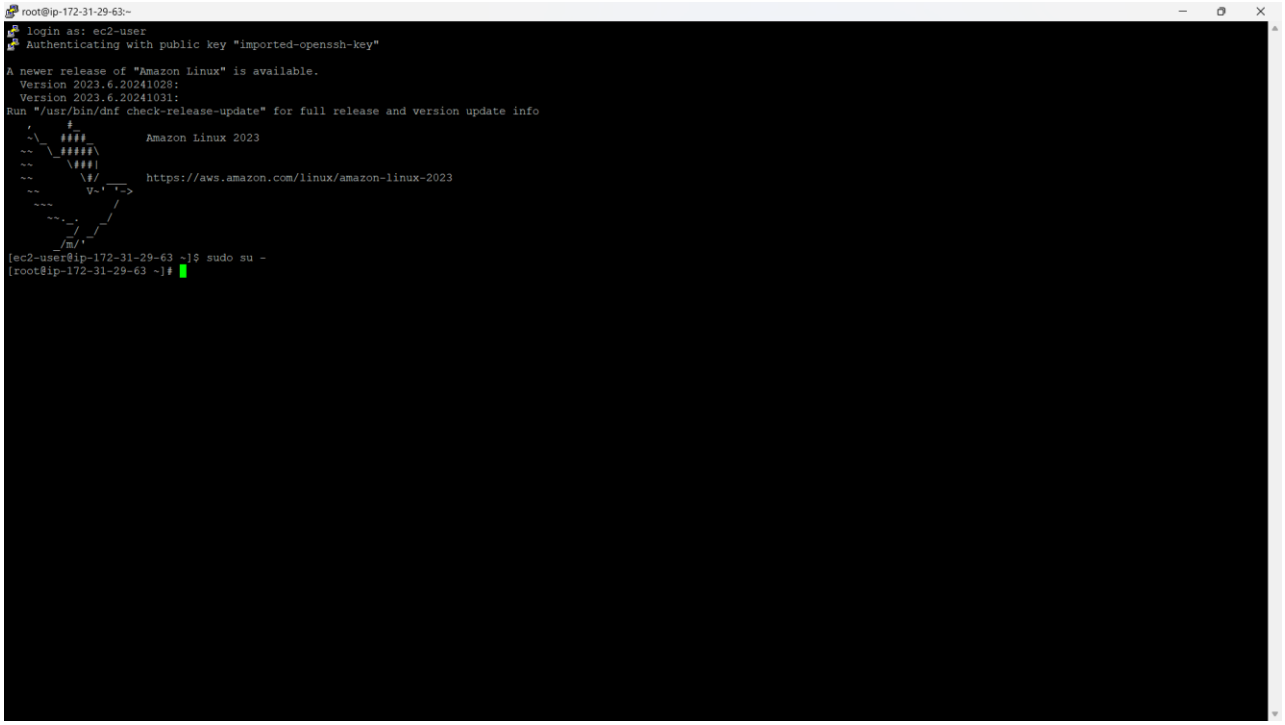
Copy public IP of our instance and open PuTTY and paste public IP and go to SSH > Auth > Connection and browse for .ppk and connect through PuTTY and login as ec2-user because I have taken amazon machine image as Amazon Linux 2. We can able to see this process in below image.



Connection through PuTTY.

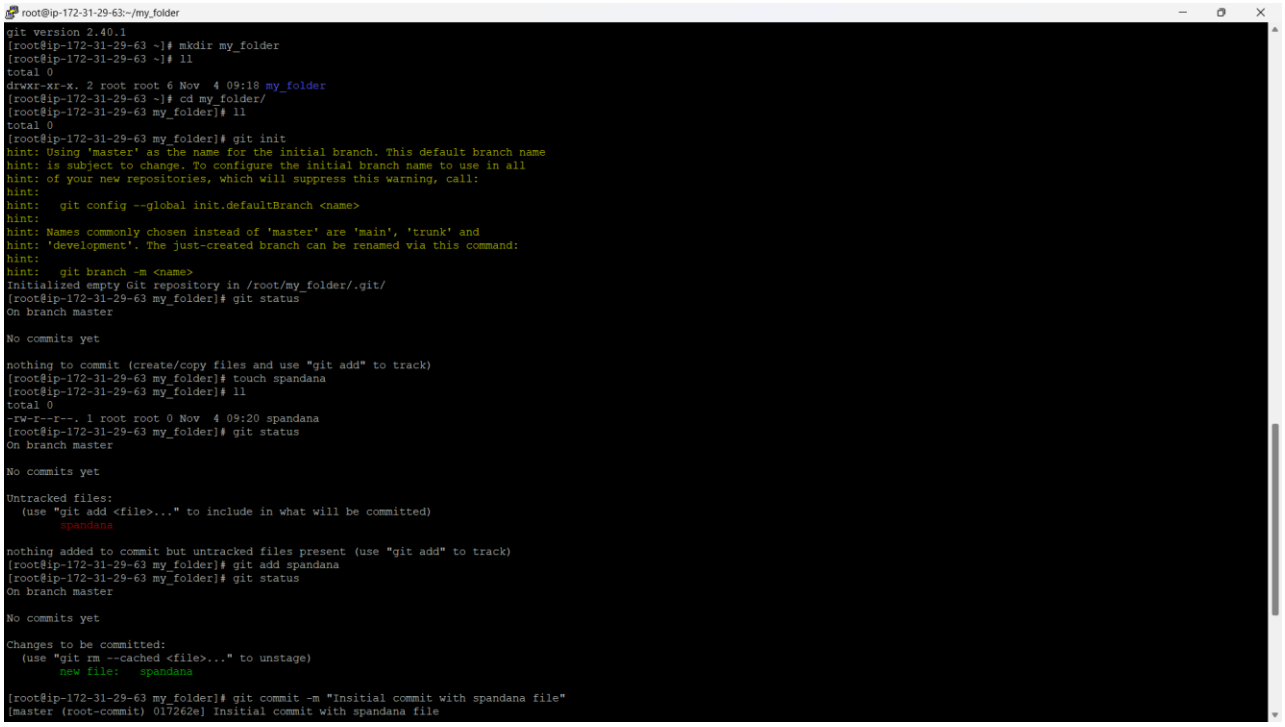


This is the terminal open when we connect through PuTTY.



LAB-2 CREATING REPO IN LOCAL MACHINE

First install git in our local machine using yum install git -y and create one folder on local machine and navigate to that folder inside the folder initialize it using **git init.** This will create a hidden .git folder and set up the directory run the **git status** command to check the status.

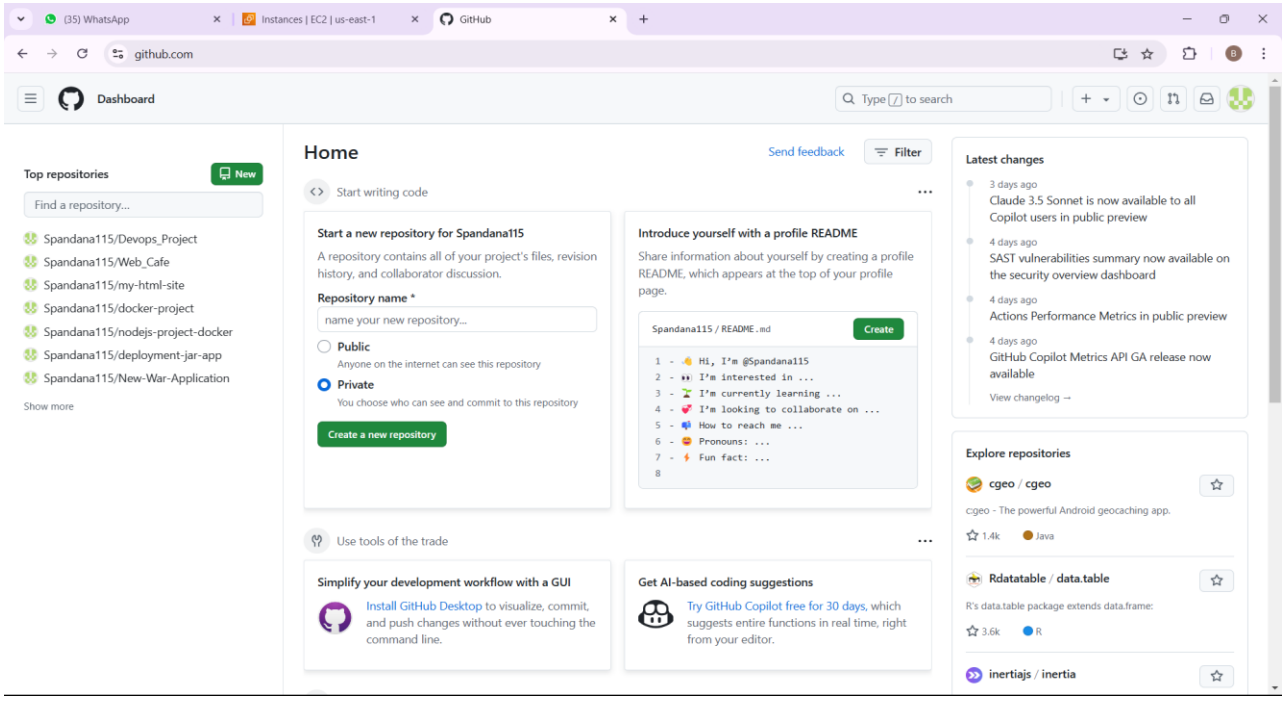


Create one empty file using touch command **touch <filename>** run git status and see we will see that file as a untracked file. Now we need to stage the file for that we need to **git add <filename>** running this command the file will be moved to the staging area and start tracking. Again check the **git status** now the file will be appear in green including it's staged and ready to be committed. Now commit the changes using **git commit -m "message"** to save the changes to local repo and run **git status** it will show working tree is clean. Check **git log** to check the commits have been done or not.

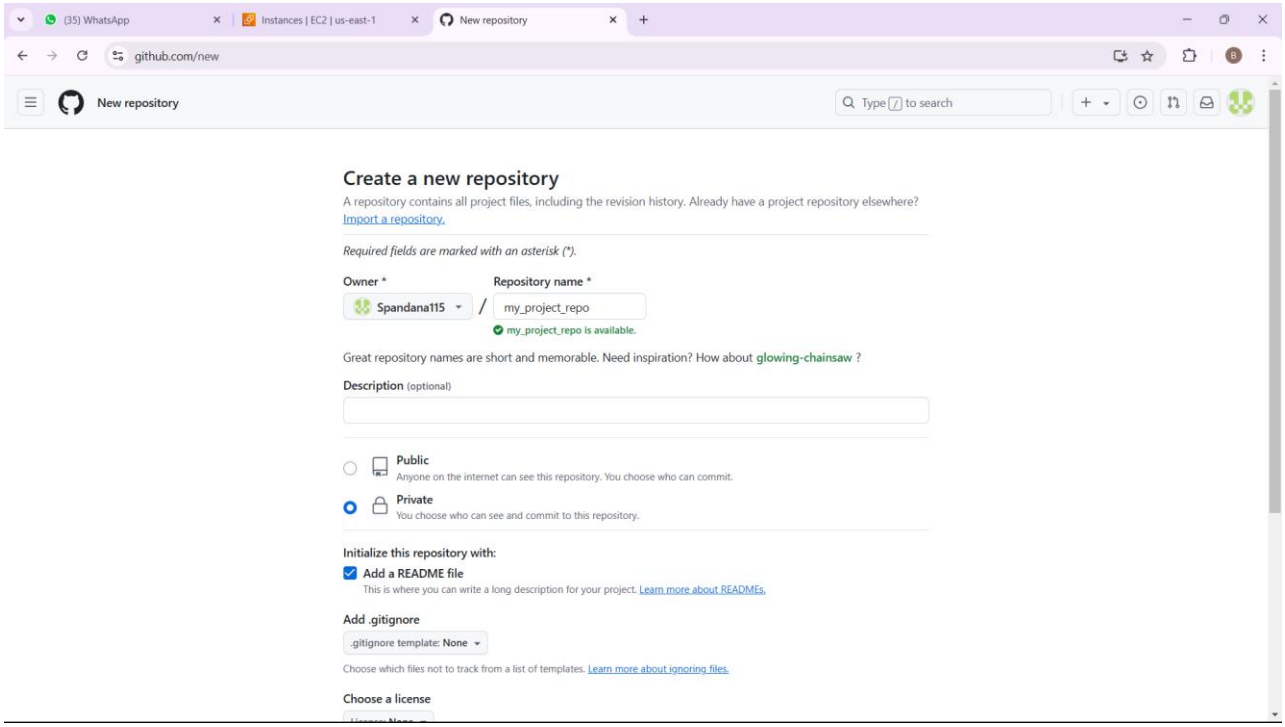
```
nothing added to commit but untracked files present (use "git add" to track)
[root@ip-172-31-29-63 my_folder]# git add spandana
[root@ip-172-31-29-63 my_folder]# git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   spandana

[root@ip-172-31-29-63 my_folder]# git commit -m "Insitial commit with spandana file"
[master (root-commit) 017262e] Insitial commit with spandana file
 Committer: root <root@ip-172-31-29-63.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 spandana
[root@ip-172-31-29-63 my_folder]# git status
On branch master
nothing to commit, working tree clean
[root@ip-172-31-29-63 my_folder]# git log
commit 017262e5059ea92b678806cce390b0724cac2dac (HEAD -> master)
Author: root <root@ip-172-31-29-63.ec2.internal>
Date:   Mon Nov 4 09:22:08 2024 +0000

    Insitial commit with spandana file
[root@ip-172-31-29-63 my_folder]#
```

LAB-3 CREATING REPO IN REMOTE LOCATION- GITHUB

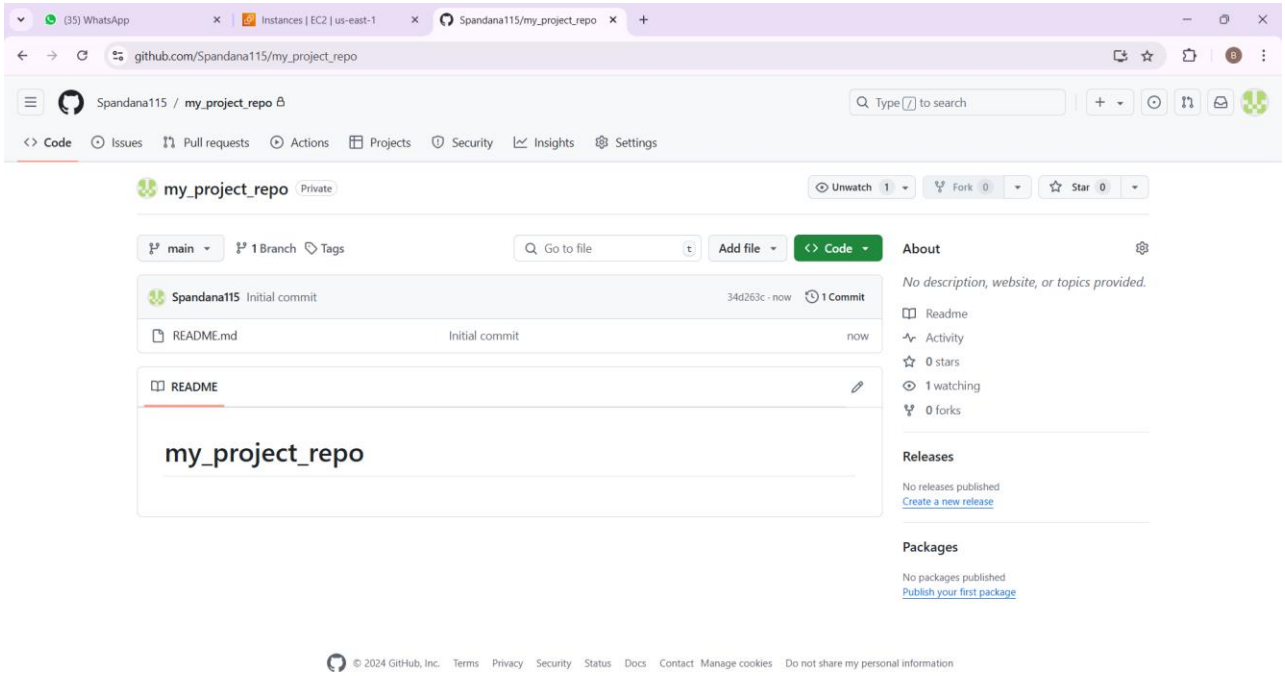
Open GitHub account we can able to see the below page after opening GitHub in the upper left corner we have a new click on that and create new repository.



Setup our new repository enter a name for repository in the repository field name choose wheather the repository should be public or private I have choosen private repository. Initialize it with README.md this file is a place to describe our project and easy to get started and initializing.

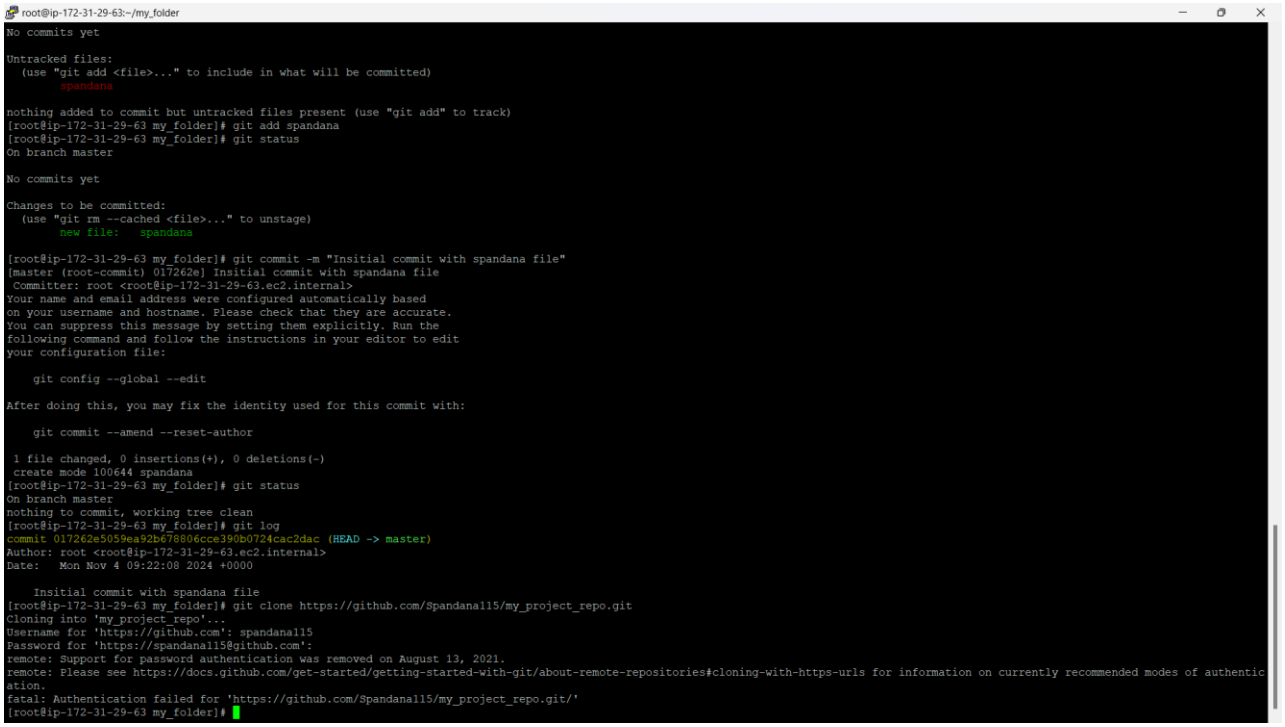


At the last we have Create repository button so click on that button then new repository has been created it will directly taken to the repository main page.

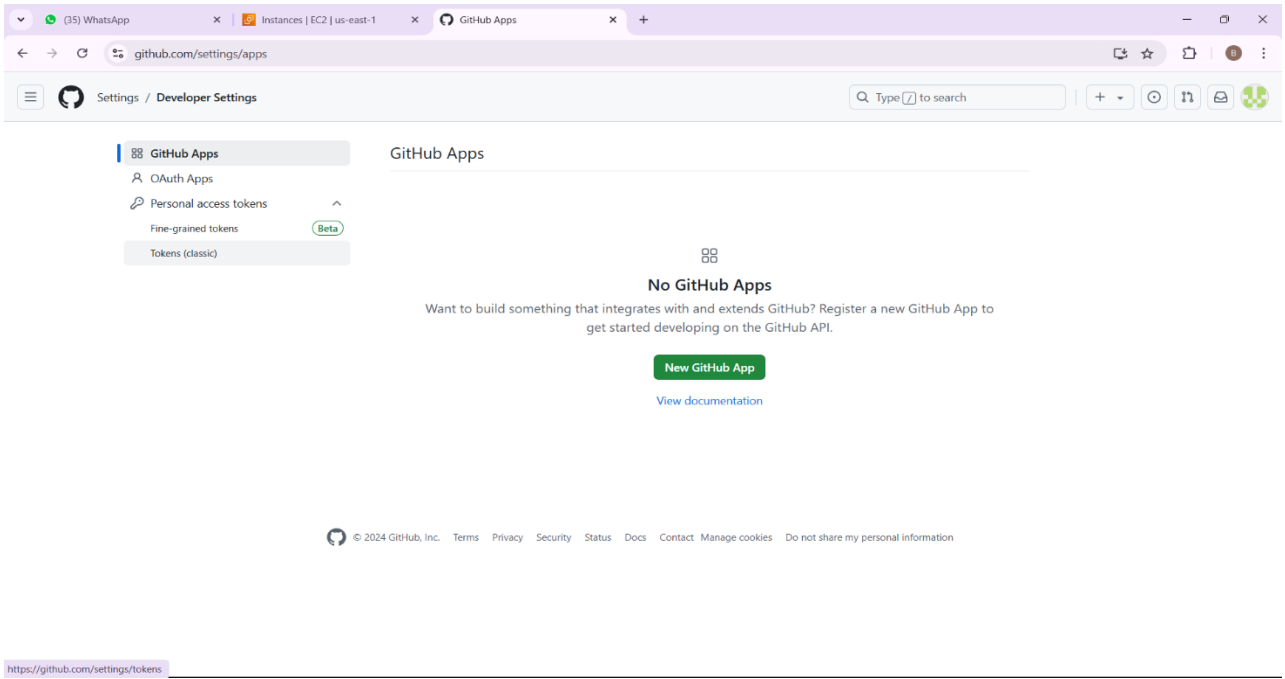


LAB-4 WORKING WITH REMOTE REPOSITORY

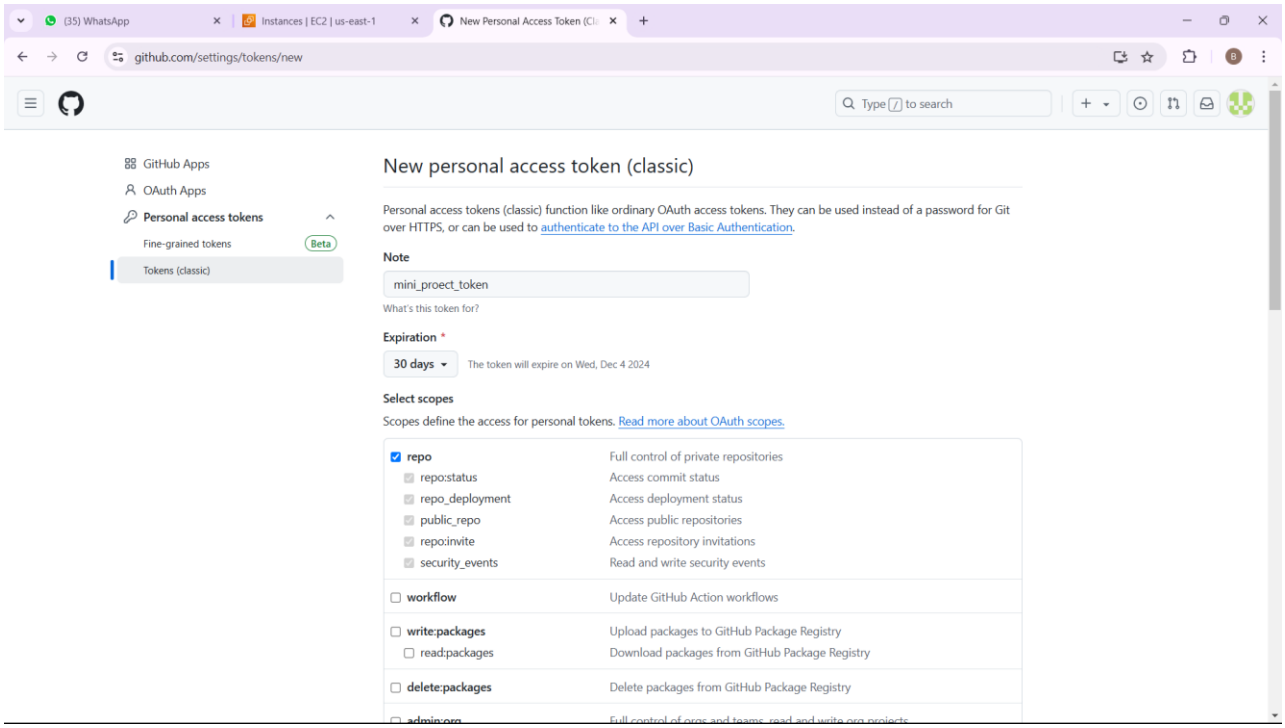
Go to repository page on GitHub and click on code button and copy the repository url use HTTPS. Open the terminal of our local machine run the command **git clone <repo url>** it will ask for username and password in the place of password it will throw an error.so we need to create personal access token for that.



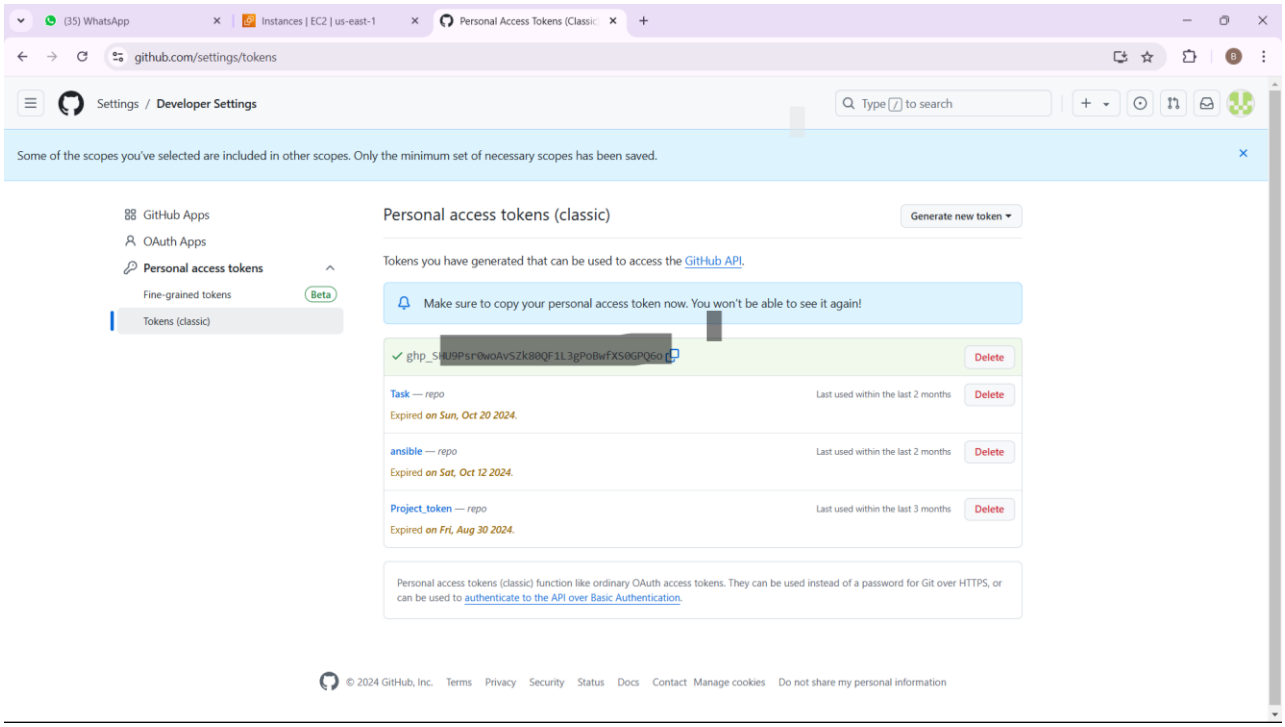
To create personal access token first we need to click on profile and go to settings in that it we have to select Developer settings after that click on personal access token (classic) as shown below image.



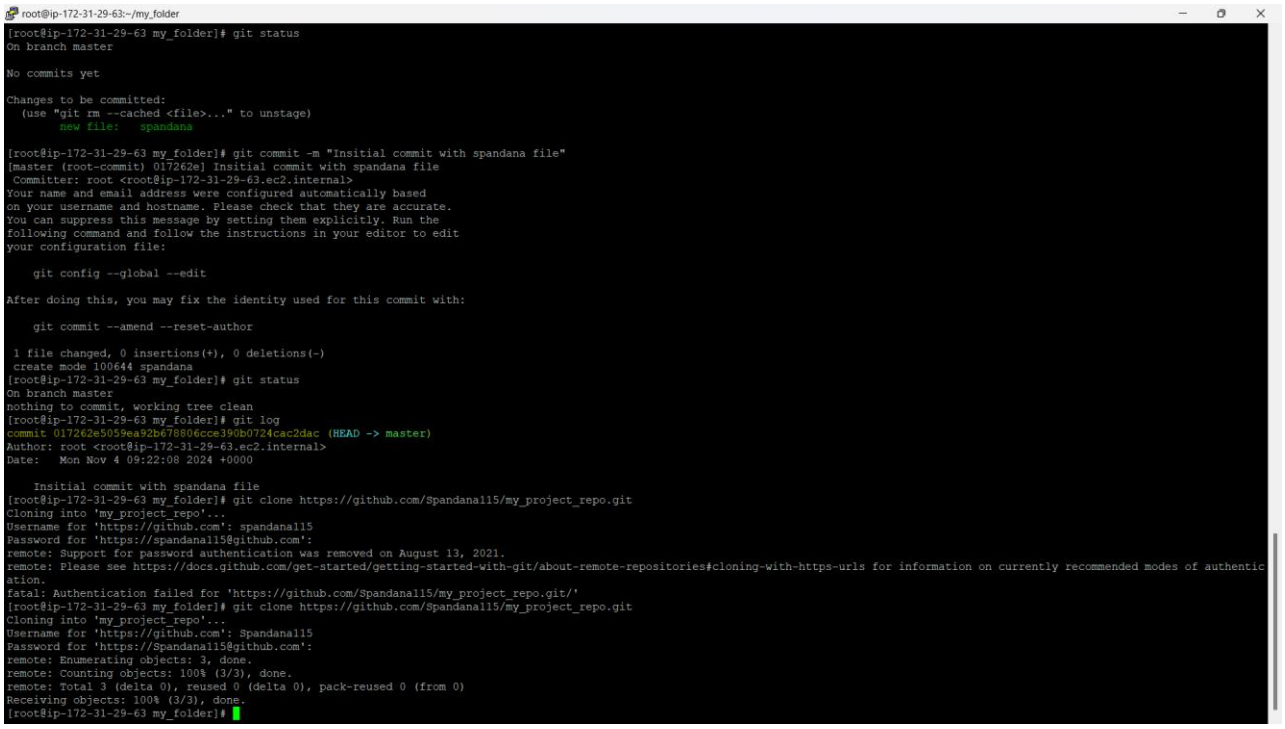
Click on generate new token provide a Note to label token we have to provide expiration date scope select repo check box for full access and then click on generate token.



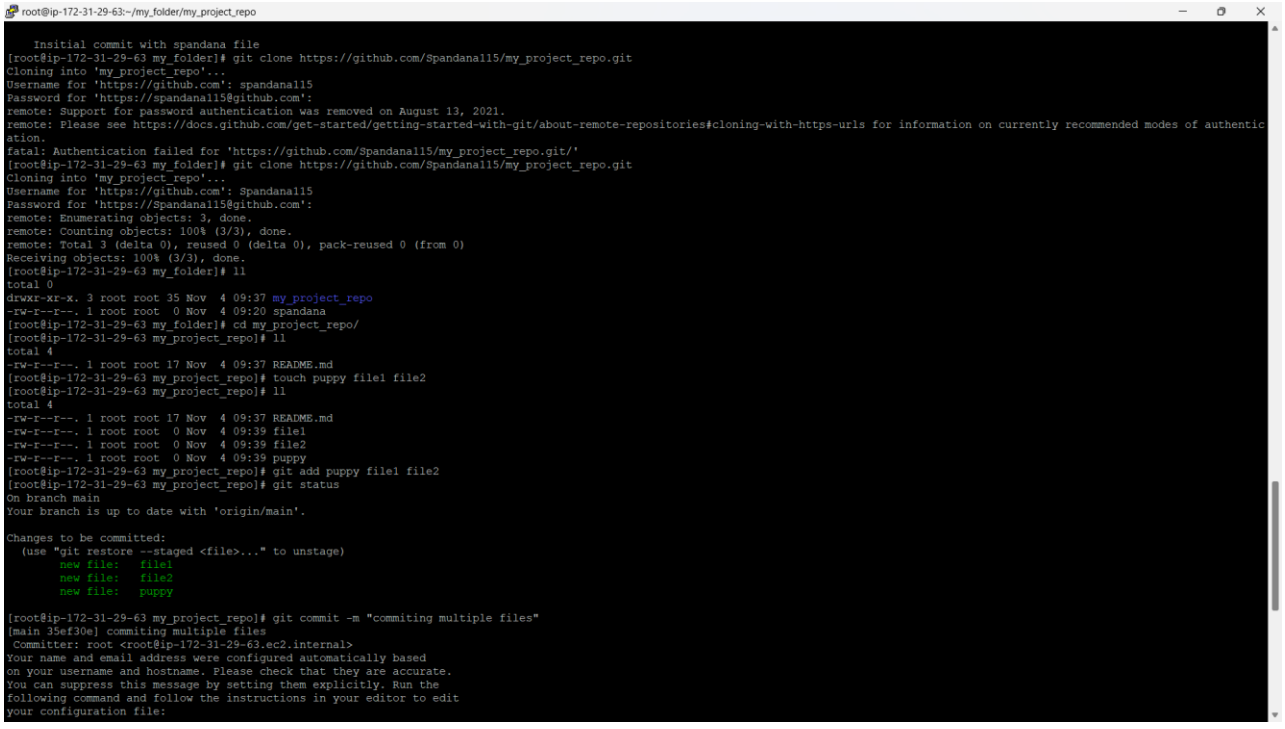
Now the token has been created successfully copy that token immediately and paste it in one place because we won't be able to view it again later.



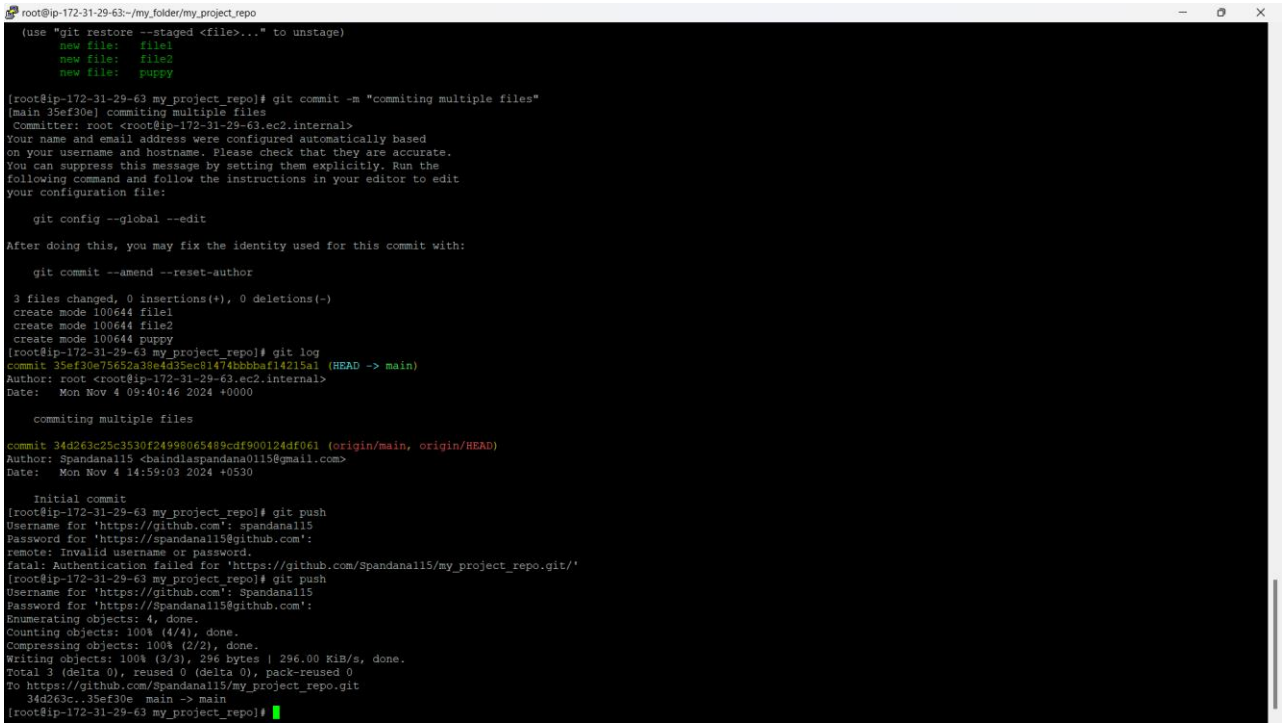
Now we again we need to clone it asks for username and password at the username we have to provide our GitHub account username and at the place of password we have to provide a token that we are generated previously.



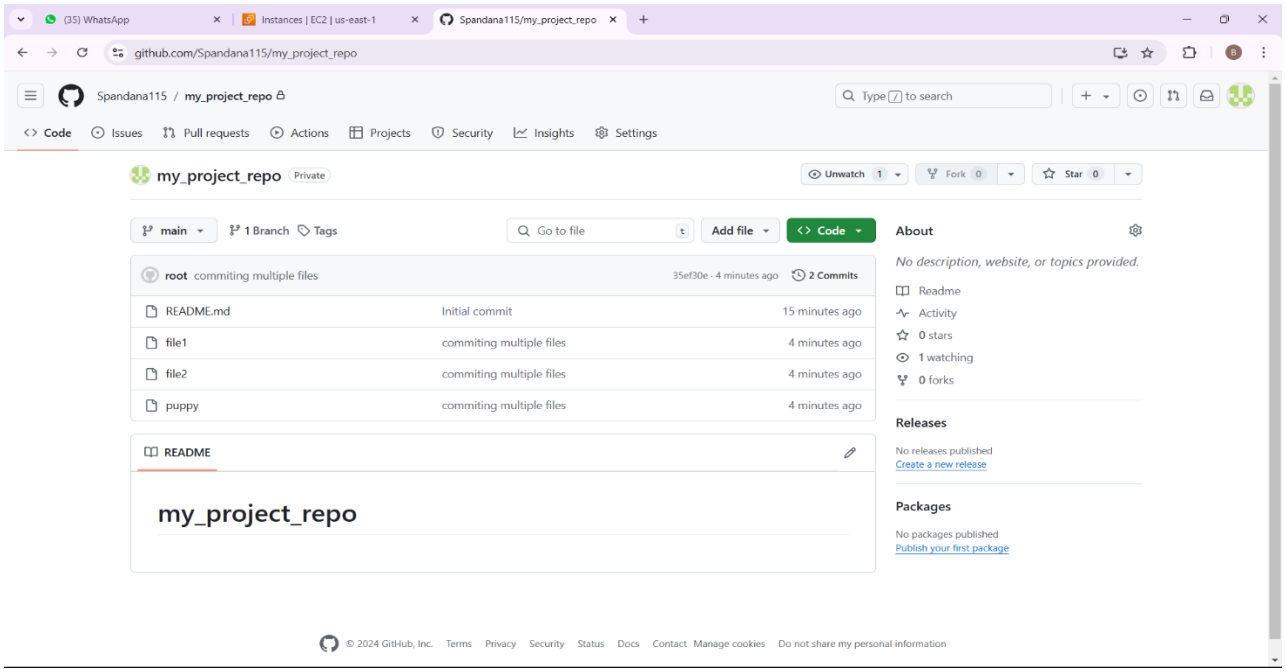
Go to new cloned repository folder use **touch** command and create some empty files. Add the new files to the staging area using **git add <file names>** after staging commit the changes using **git commit -m "message**"



After committing changes we need to push changes to the Remote Repository using **git push** while pusing it asks for username and password so provide username and token in the place of password and push.The new files should be uploaded in remote GitHub repository.

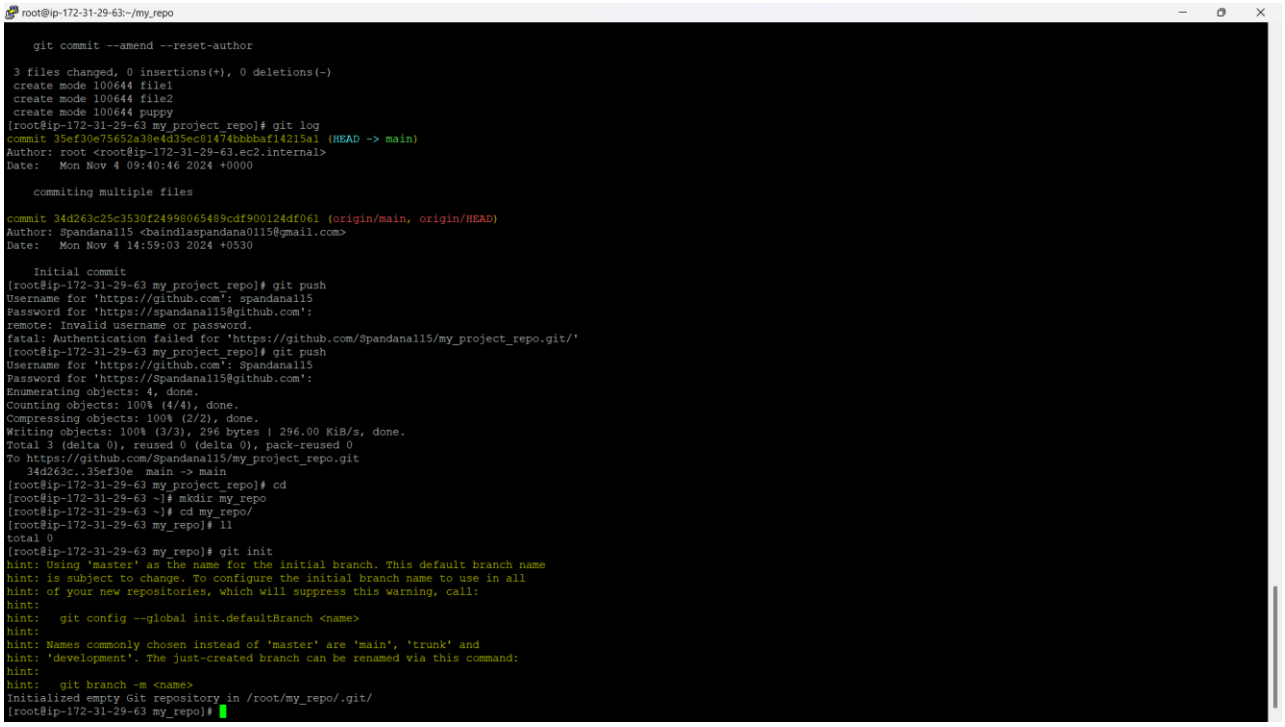


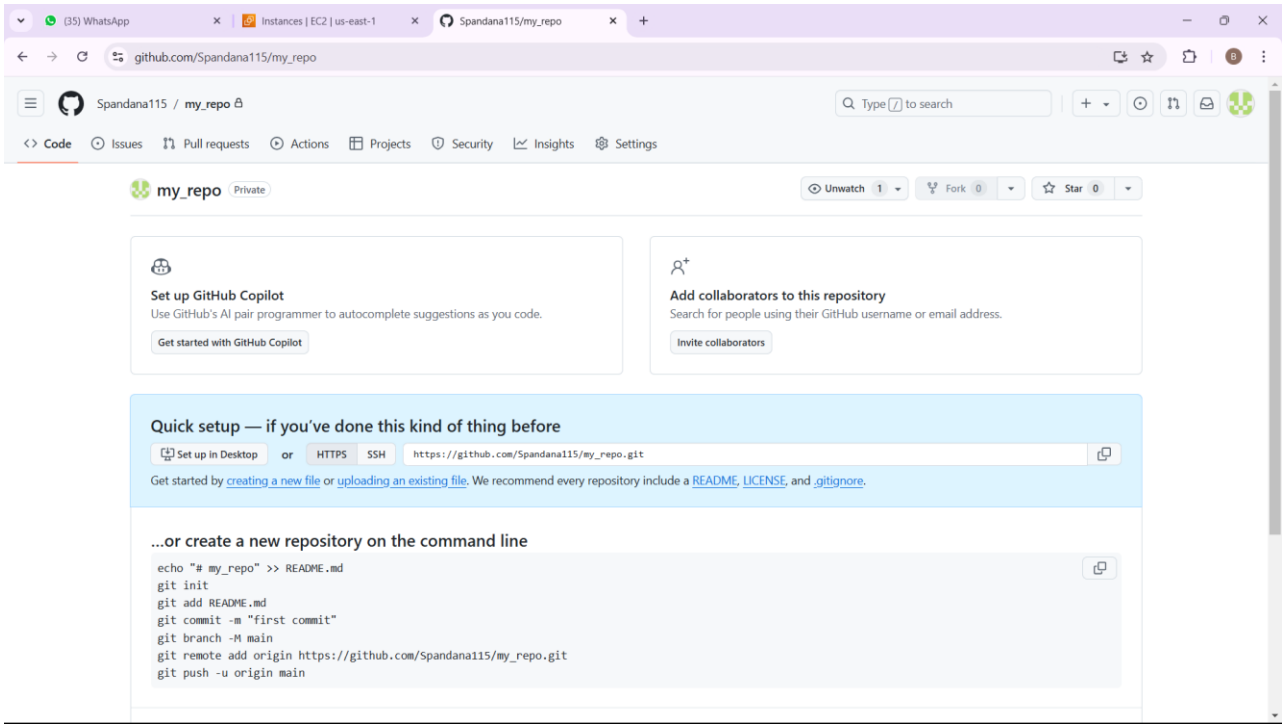
Go to GitHub account and open our newly created repository and check the repository the new files have been pushed to remote server we will find a new changes.



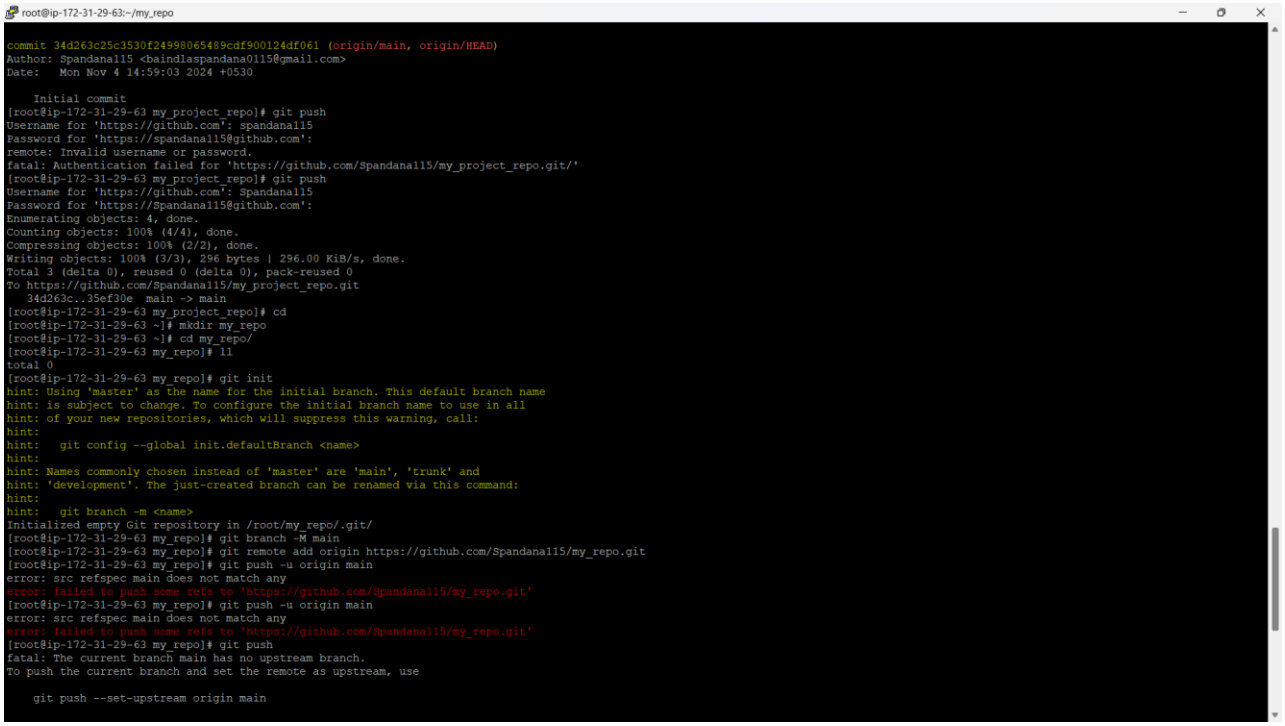
LAB-5 PUSHING A LOCALLY CREATED REPO TO GITHUB

Create a repository in local machine and go to GitHub click on new repository name the repository the same as your local repository we do not initialize this repo with a README.is selected click on create repo.

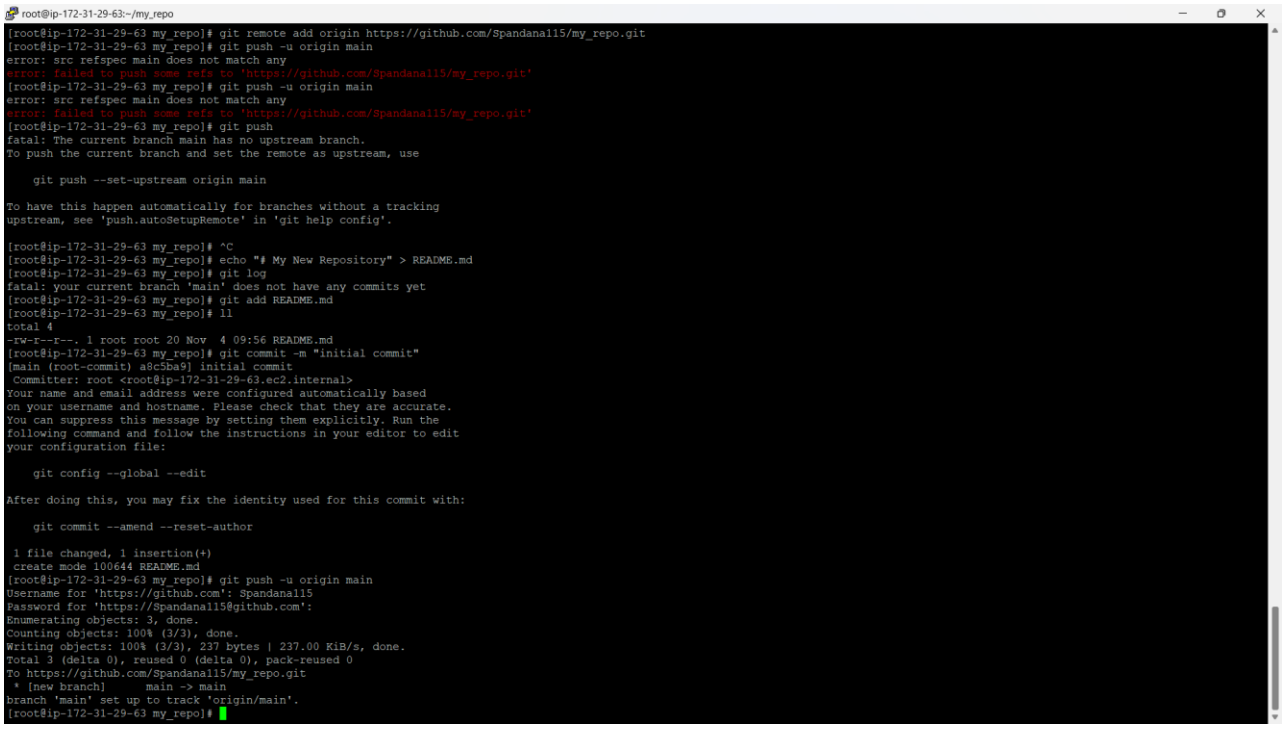




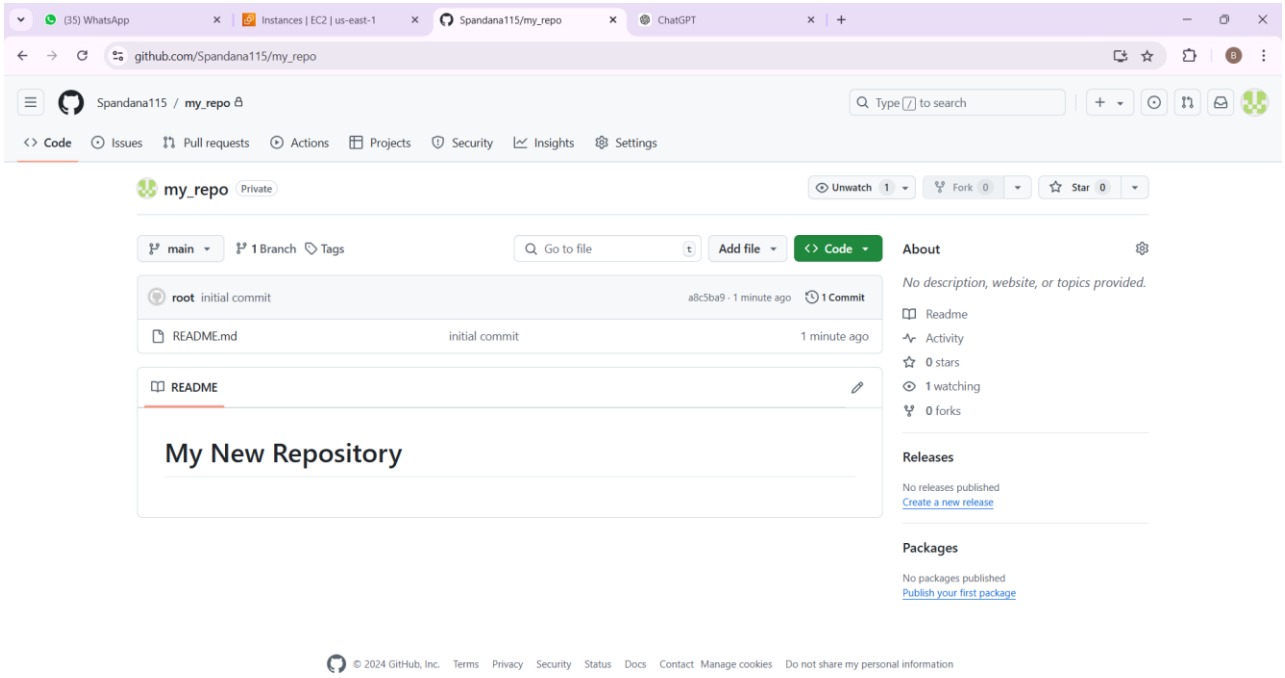
Rename the Default Branch to main **git branch -M main** and use the repository url get from GitHub creating remote repository run the following command to add the GitHub repository as the remote **git remote add origin <remote-repo-url>** we will get an error because we did not committed any files so after adding and commiting any file it will be pushed.



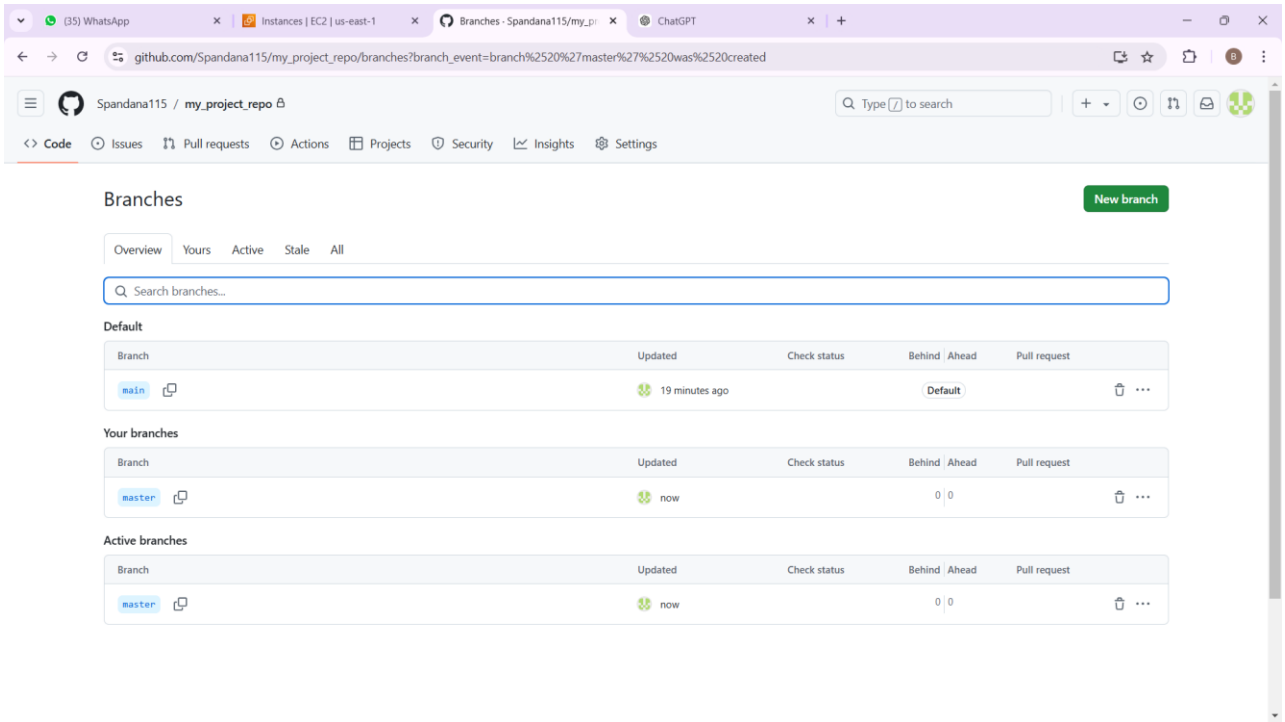
Push your local branch(main) to GitHub git push -u origin main -u flag sets the upstream branch linking your local branch to the remote.



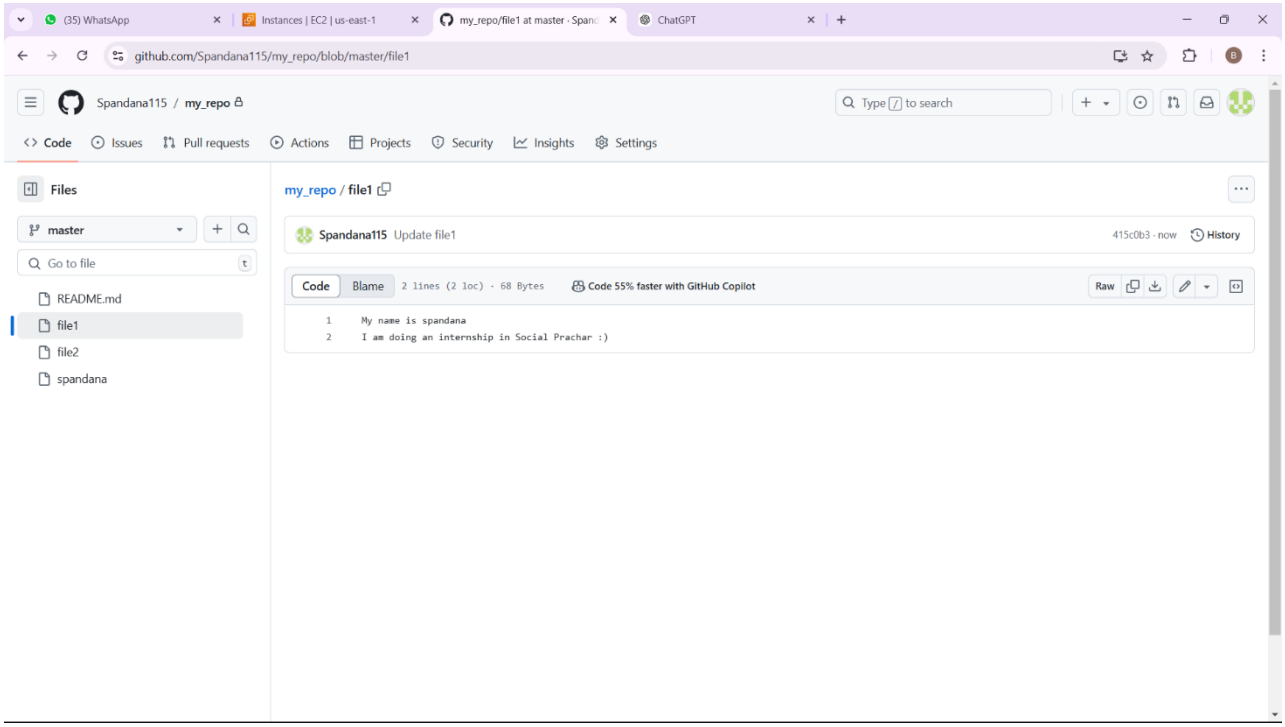
Open GitHub repository click on main branch in the dropdown type the name of new branch select the option and create new branch form main branch.



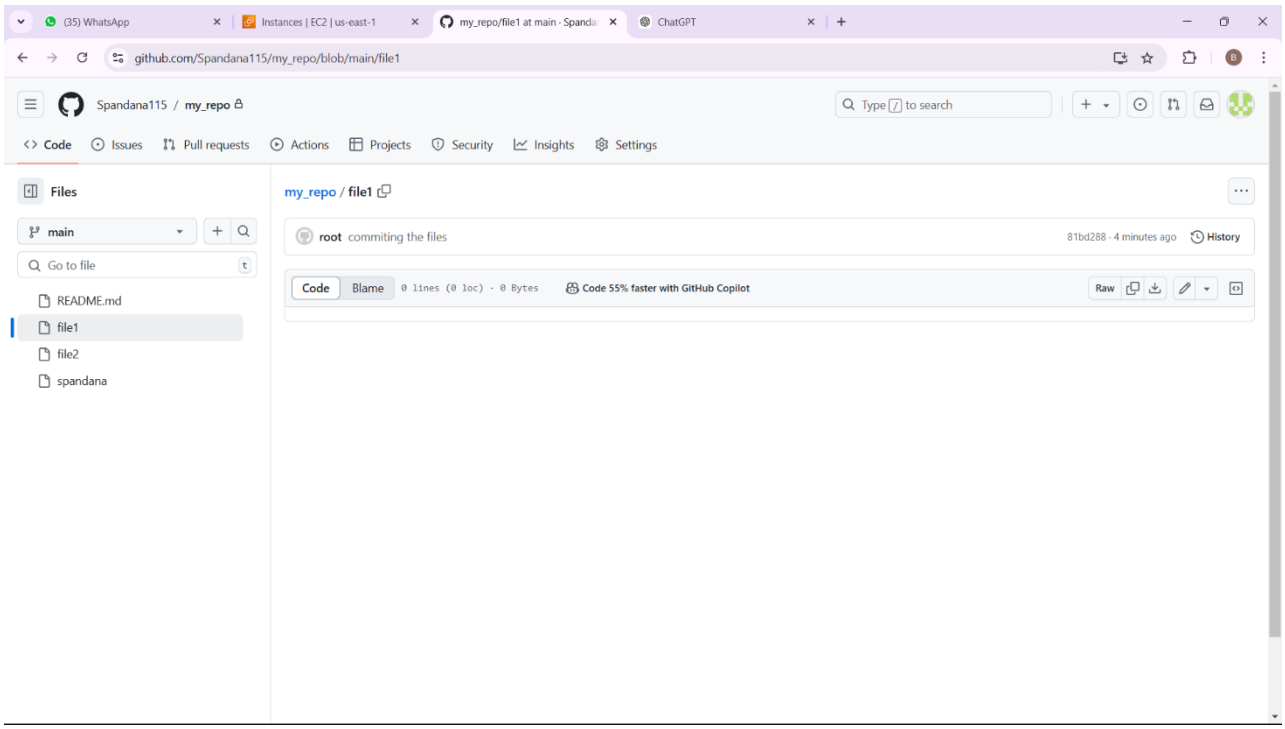
In the below image we can able to see that master branch has been created successfully.



Switch to new branch and go to the any file edit that file or write a content in that file with in new repository make changes and click on commit changes.

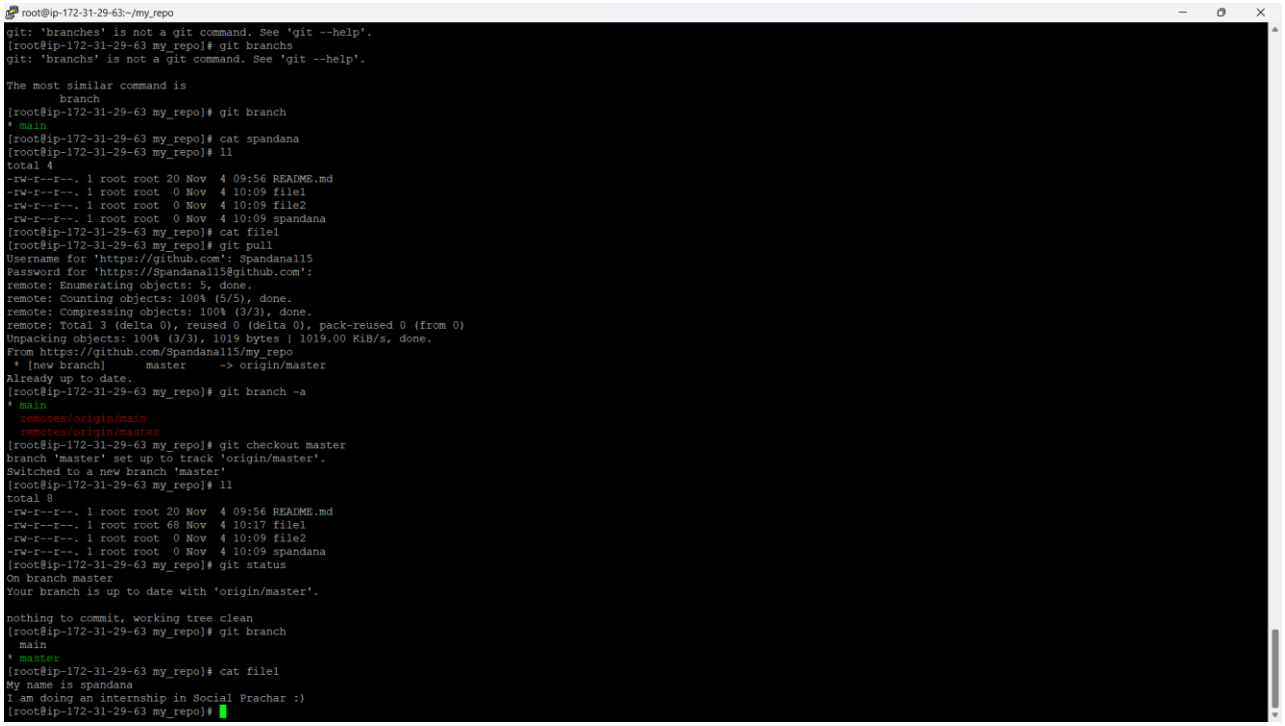


Switch back to main branch and using branch dropdown check the file that has been updated or modified we can see that remains unchanged in file1 in main branch in the below image. Go back to the previous image and see that stores some content in the file1.



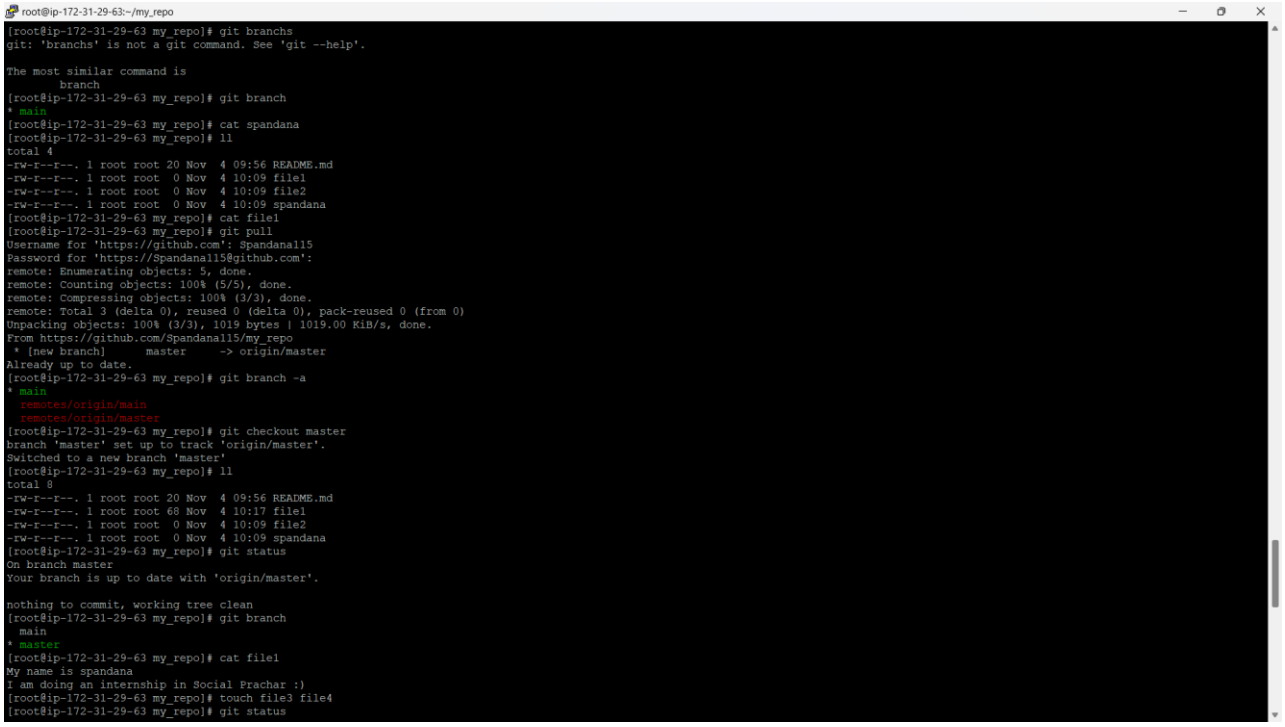
LAB -7 PULL ALL THE BRANCHES IN YOUR LOCAL MACHINES

Open PuTTY we have to go to where git repository located we have to run the following commands to fetch the branches from remote repository **git pull** list all branches using **git branch -a.**



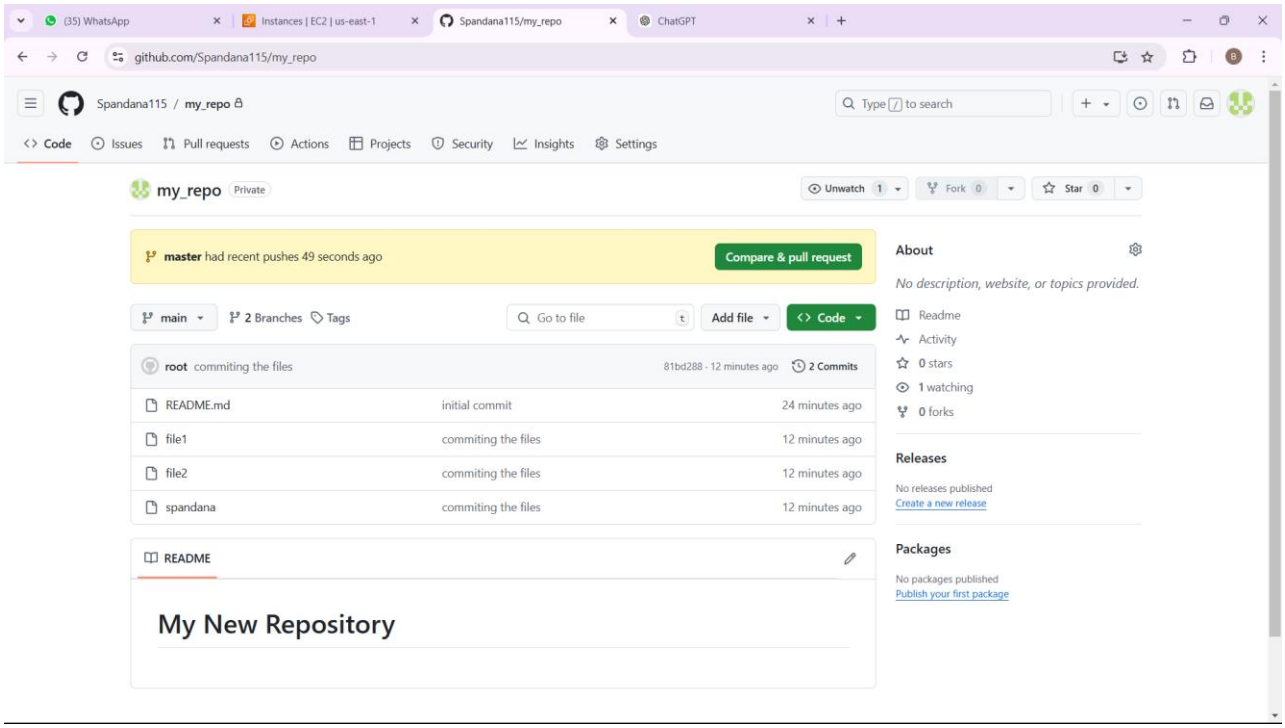
Switch to newly created branch using git checkout <master> confirm that we are on a new branch by using git branch

the active branch is marked with *.

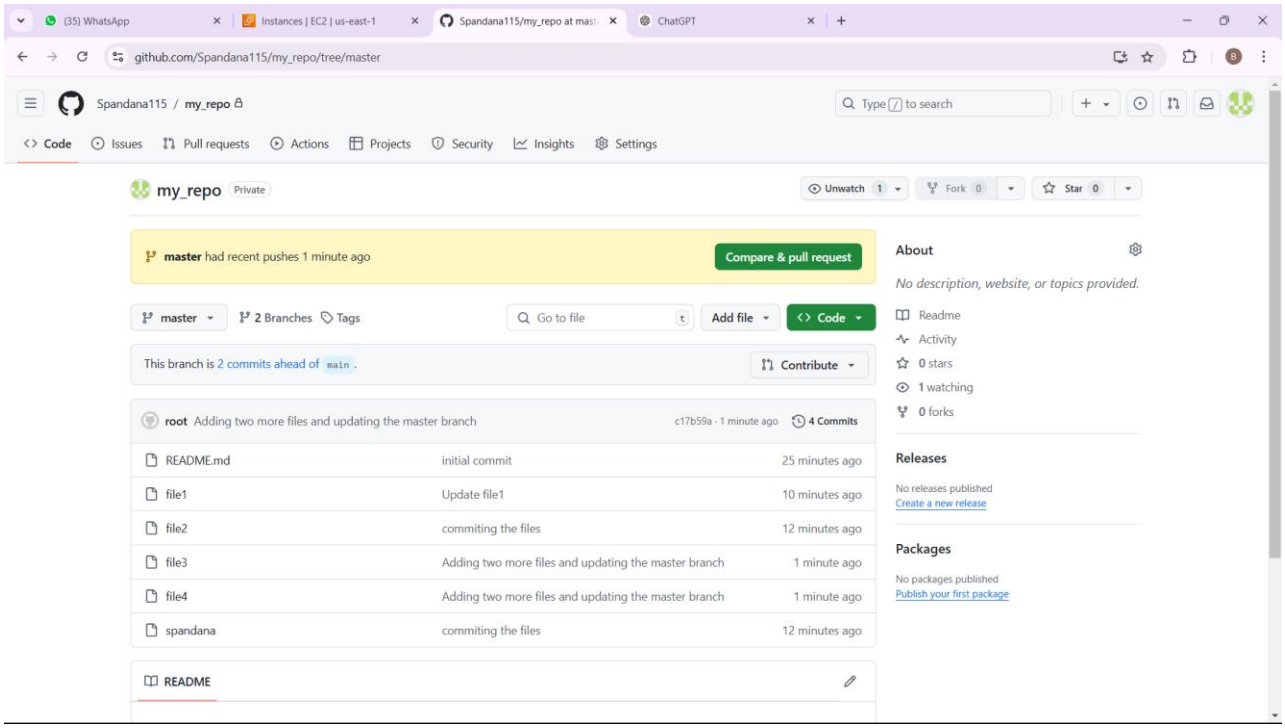


Add some iles to the branch you can create empty files with the touch command **touch file3.txt file4.txt** and stage the new files by **git add <file names>** and commit changes using **git commit -m "message"** after commiting check logs if everythhing correct push the committed changes to remote feature branch.

Confirm no changes in main branch switch back to the main branch in GitHub and verify that the new files are not present in the main branch.

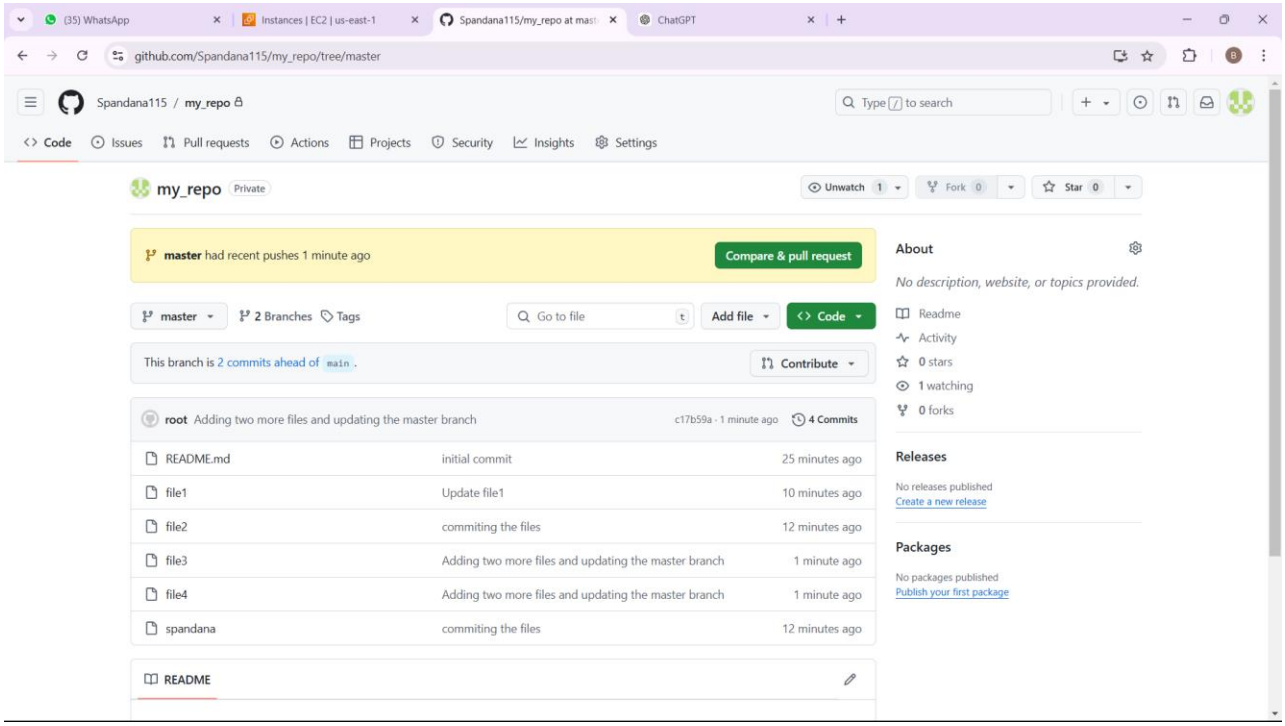


Switch to master branch to verify that the new files are presented.

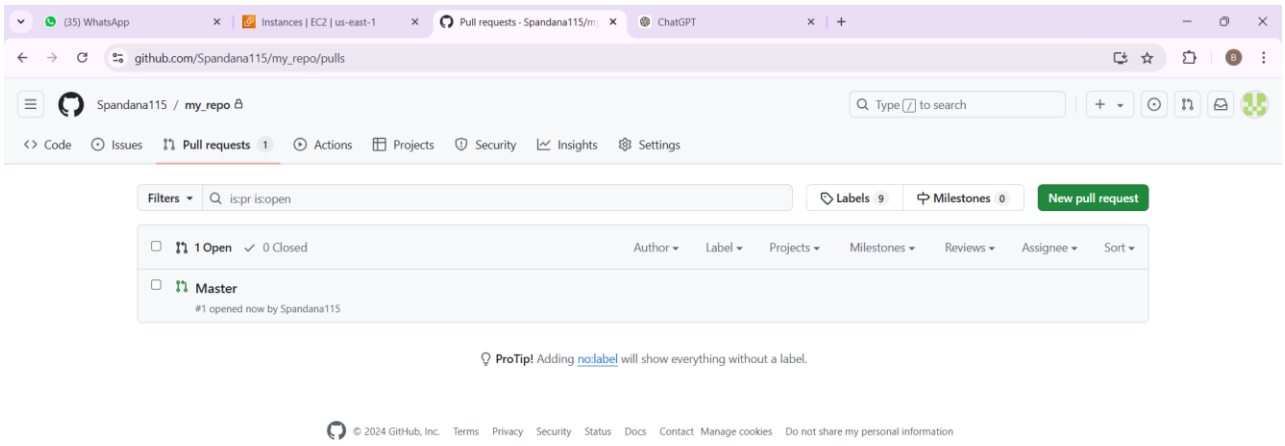


LAB-8 MERGE OUR FEATURE BRANCH WITH OUR MAIN BRANCH

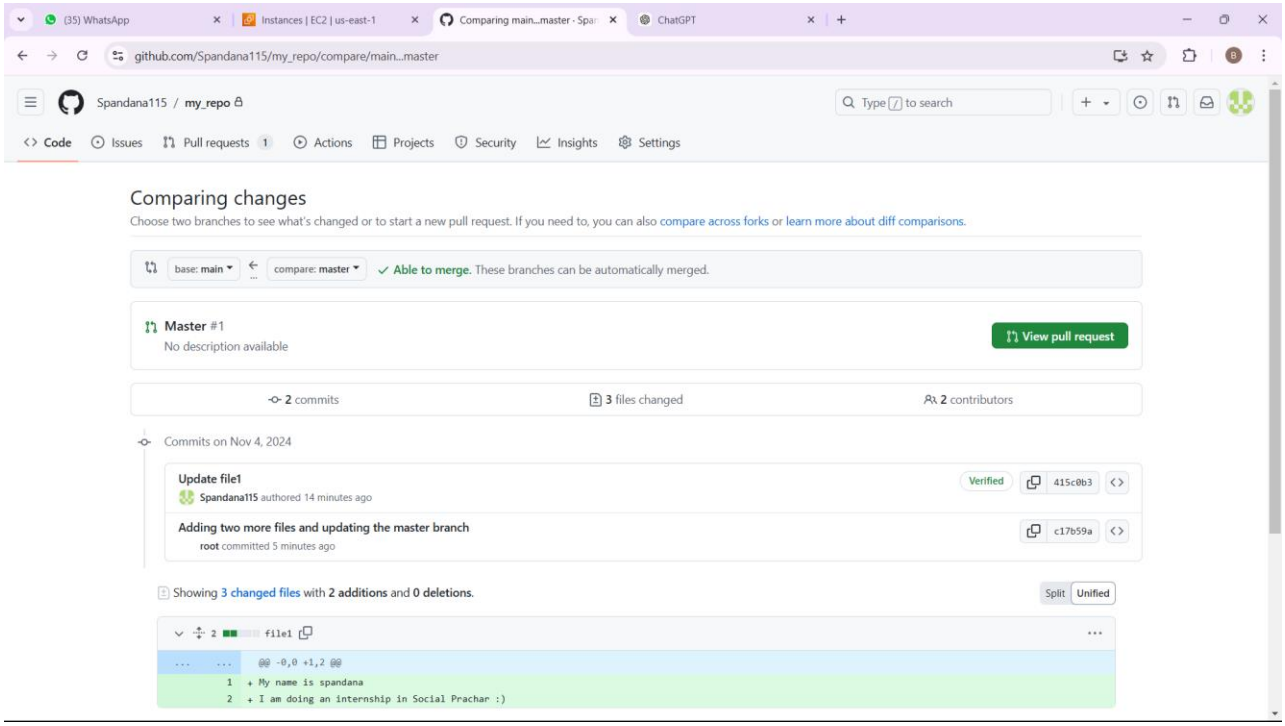
Open your repository in GitHub where you have both the main and master branches. Click on the “Pull request” tab at the top of the repository page.



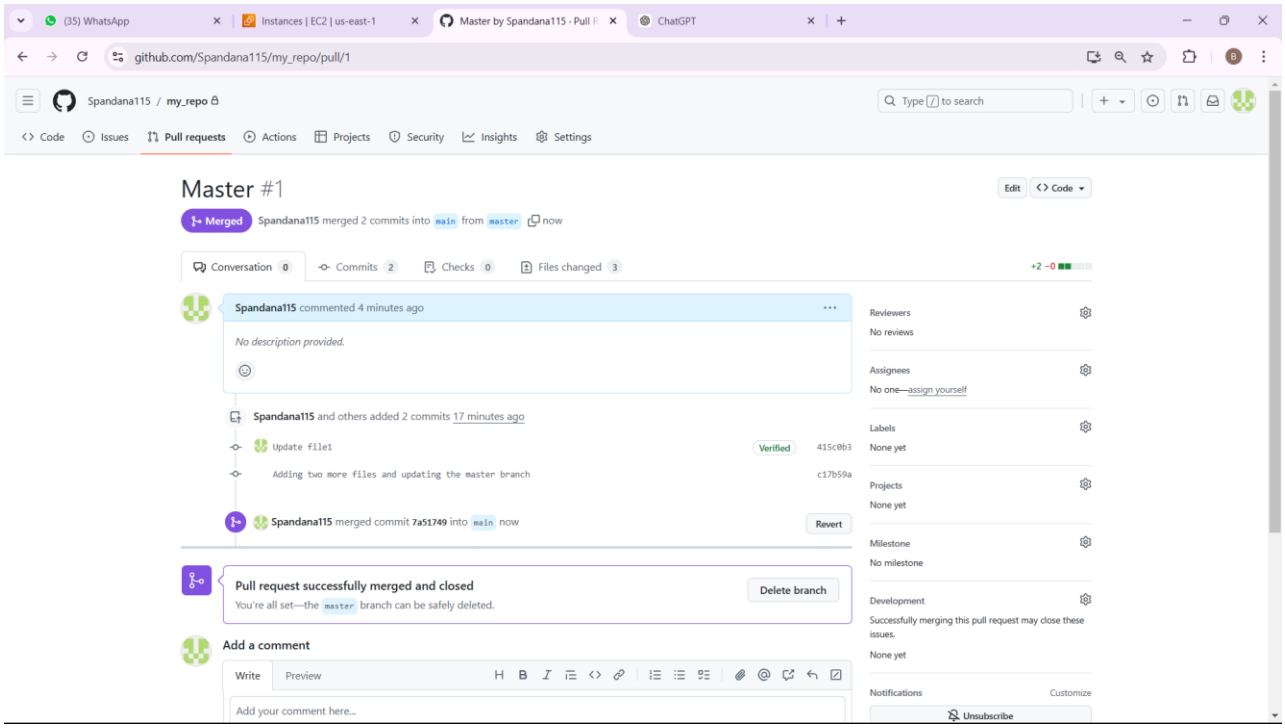
Click on the “New pull request” button to initiate a pull request.



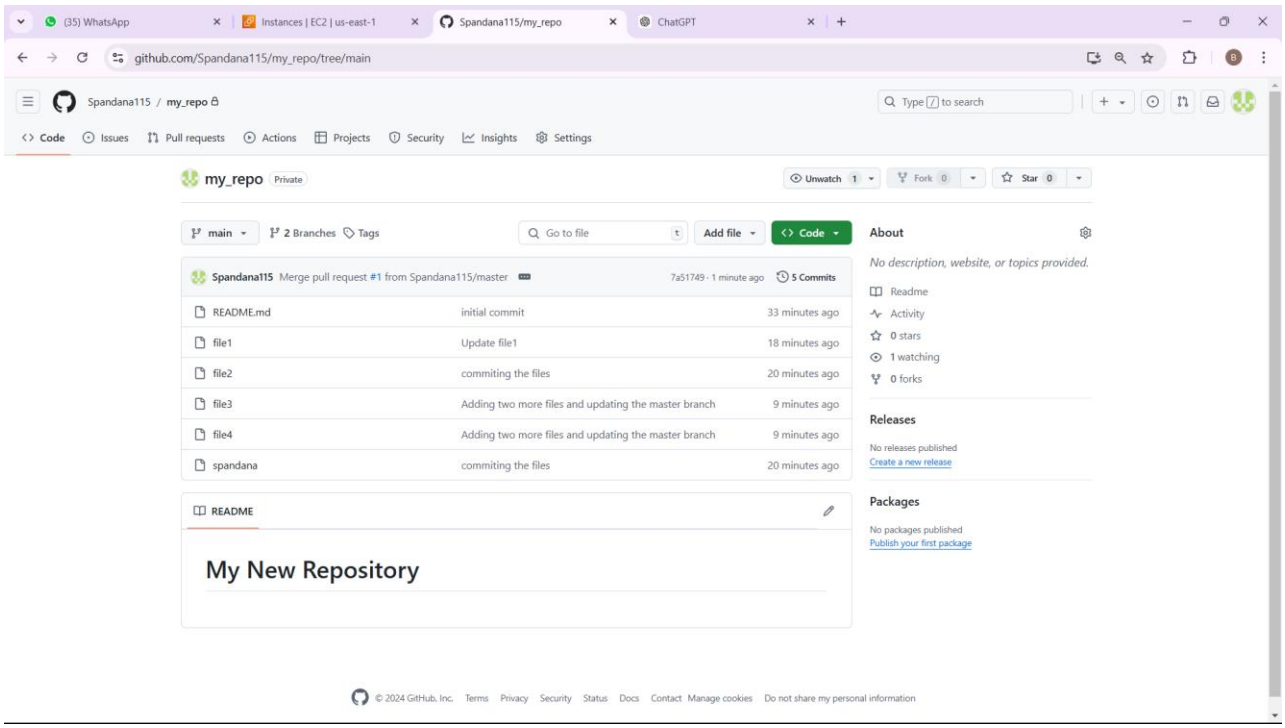
Select branches for the Pull Request in the comparison options set base branch as main and set compare branch as a master click on create pull request and confirm.



In the pull request tab locate a newly created pull request. Click on the pull request to view details. Click on the “Merge pull request” button and confirm the merge by clicking on confirm merge.



We can able to see that the files are merged from master branch to main branch.



LAB-9 GO TO LOCAL MACHINE

Open PuTTY on local machine where you have a cloned version of remote repository. Checkout to the main branch using **git checkout main** this ensures we are working with the main branch before pulling any changes. To pull the latest changes from the remote repository using **git pull** this will fetch and merge any updates from the remote repository into local main branch. After the pull command is executed main branch will be up to date with the remote repository run the following command to check the status and confirm using **git status and git log.**