KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTION)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Narayanguda, Hyderabad – 500029

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB RECORD

CLOUD COMPUTING LAB

B.Tech. III YEAR II SEM (KR21)

ACADEMIC YEAR 2023-24

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTE)
Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad
Department of Computer Science & Engineering

Certificate

This is to certify that following is a Bonafide Record of the workbook task done by

_____bearing Roll No_____of _____

Branch of_____year B.Tech Course in the_____

Subject during the Academic year_____&_____under our supervision.

Number of week tasks completed: _____

Signature of Staff Member Incharge

Signature of Head of the Dept.

Signature of Internal Examiner

Signature of External Examiner

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad
Department of Computer Science & Engineering

Daily Laboratory Assessment Sheet

Name of the Lab: Name of the Student:

Class: HT.No:

S.No.	Name of the Experiment	Date	Observation Marks (3M)	Record Marks (4M)	Viva Voice Marks (3M)	Total Marks (10M)	Signature of Faculty
	TOTAL						

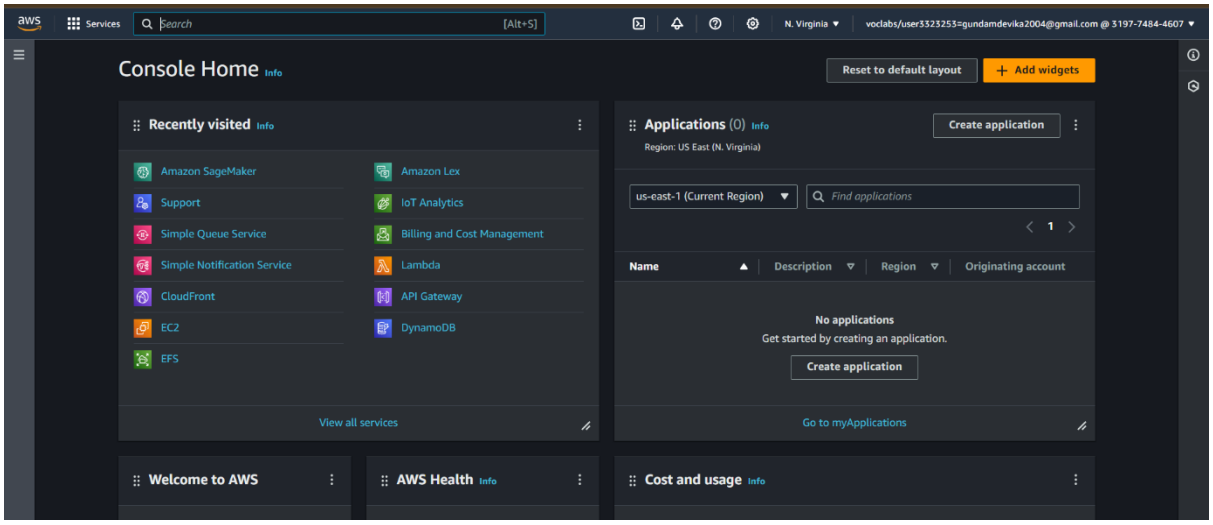
Faculty Incharge

INDEX

Experiment-1

Aim: Establish an AWS account. Use the AWS Management Console to launch an EC2 instance and connect to it.

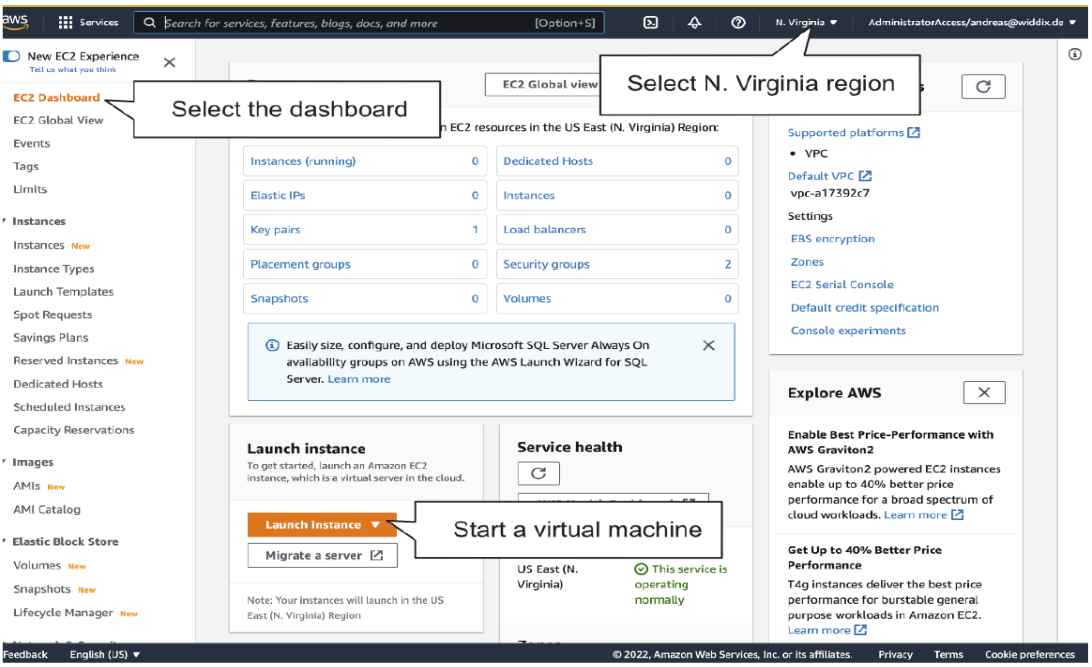
Step 1: Login to the AWS Management Console using Login Credentials



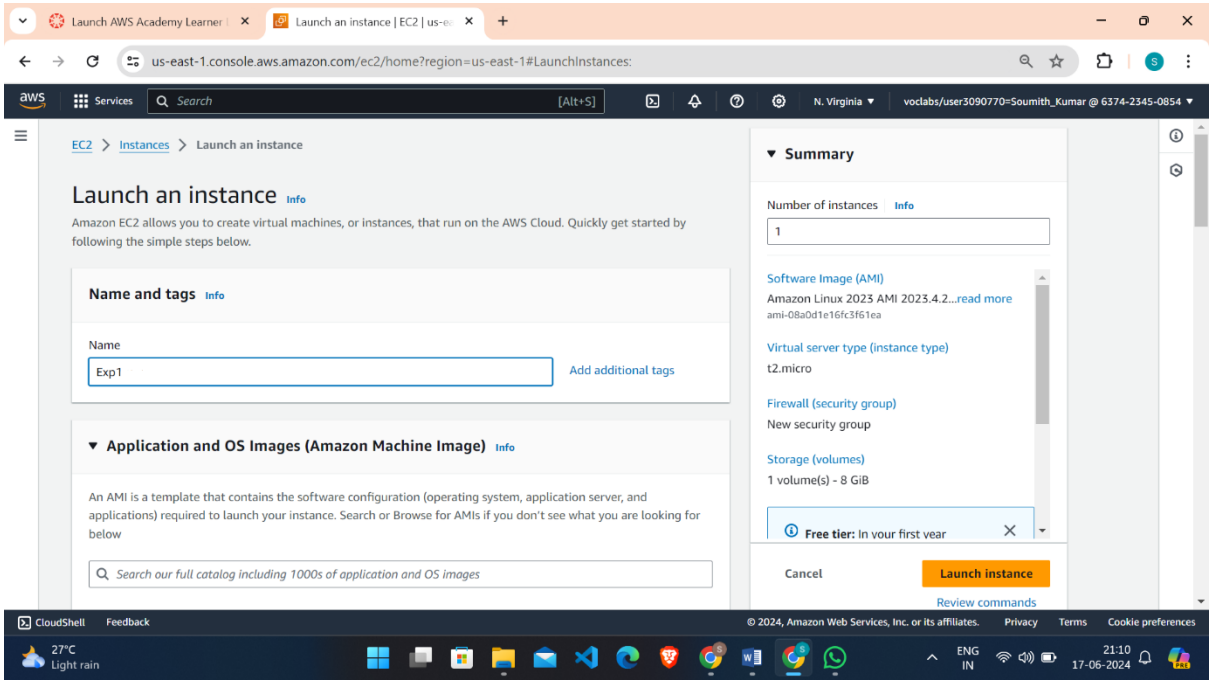
Step 2:Make sure we're in the N. Virginia (US East) region because we optimized our examples for this region.

Step 3:Click on Services and search for EC2.

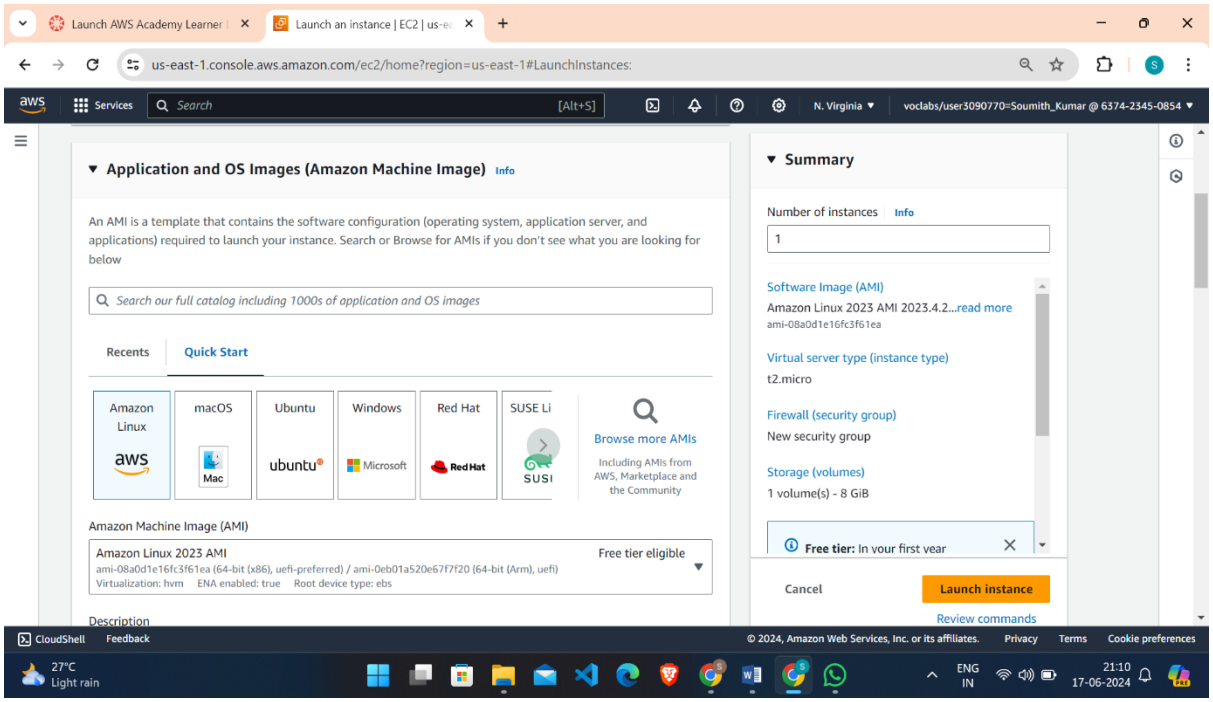
Step 4:Click Launch instance to start the wizard for launching a virtual machine as shown in the figure.



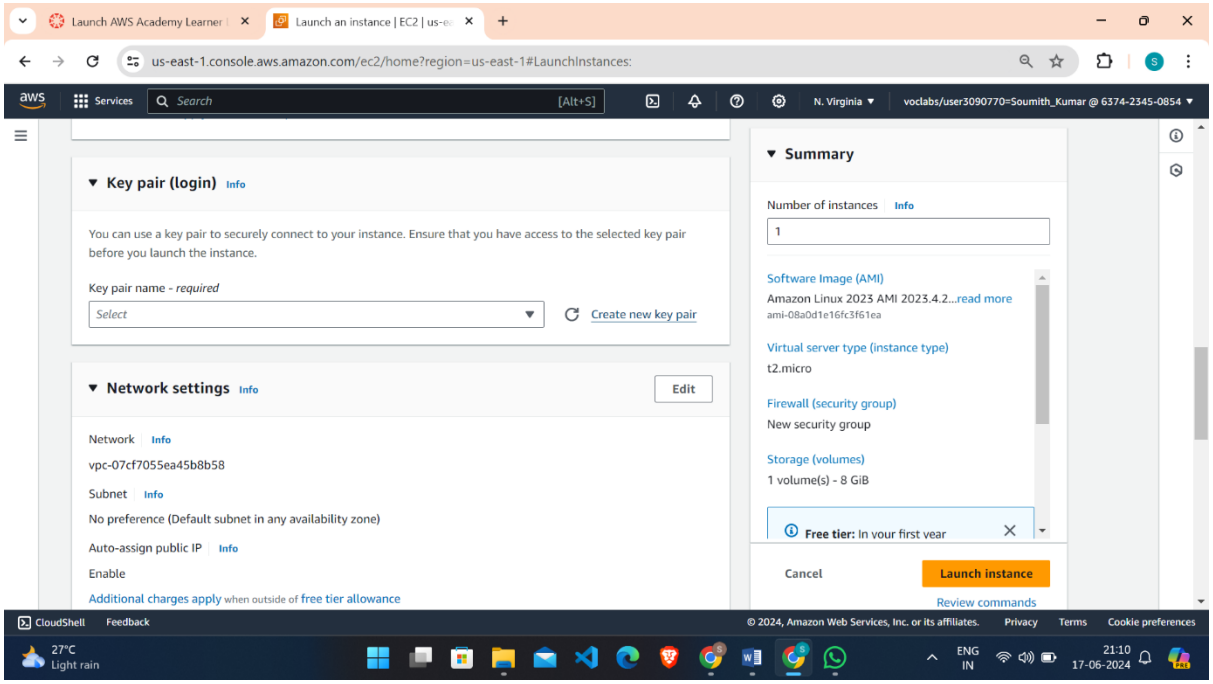
Step 5: Give a name to the instance Exp1



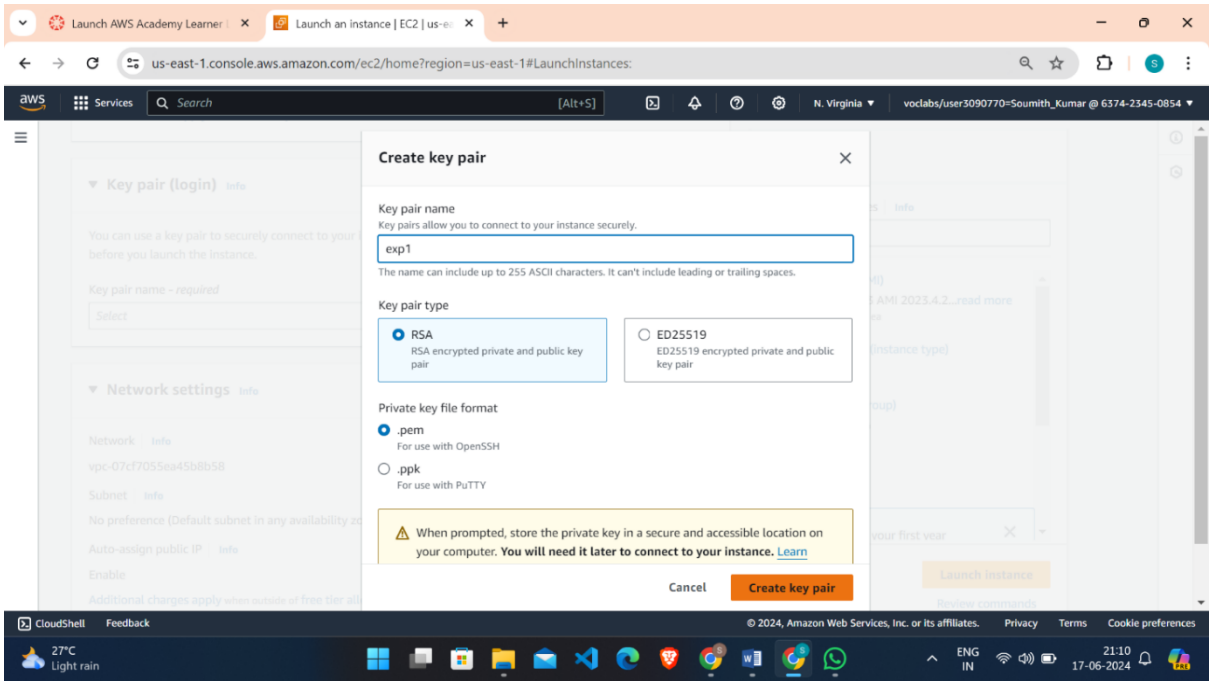
Step 6: Select the Amazon Linux as our Operating System and select instance type as t2.micro



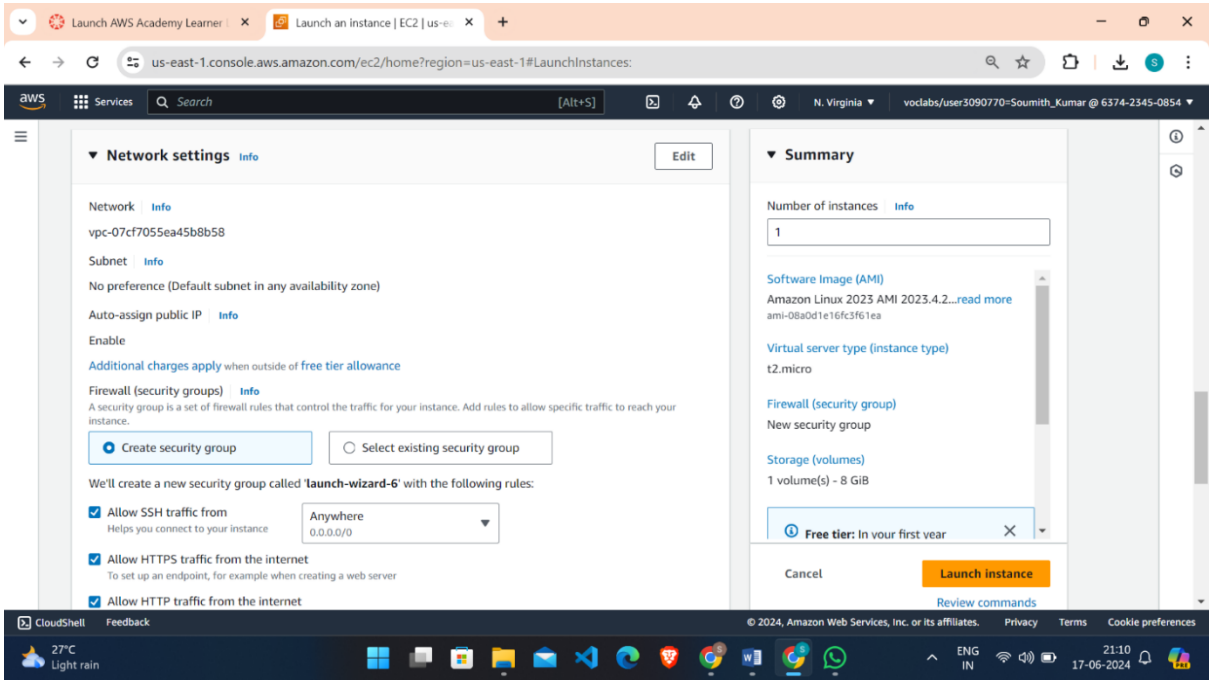
Step 7:Now create a key pair by clicking on the hyperlink



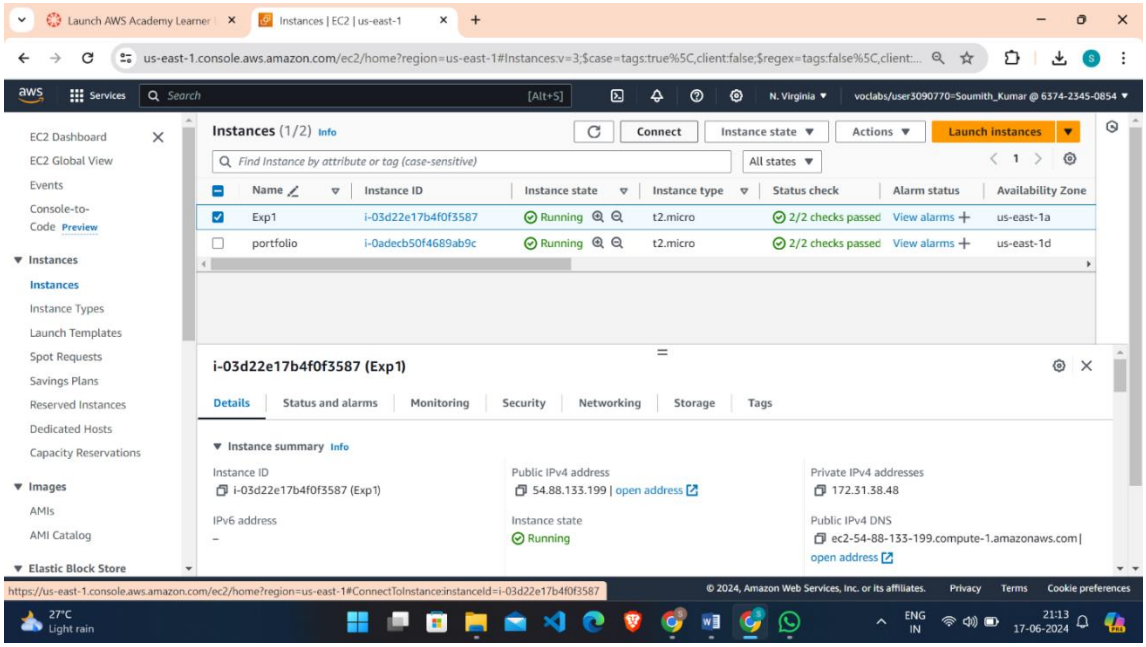
Step 8:Name the key pair exp1 and click on create key pair keeping other fields default.



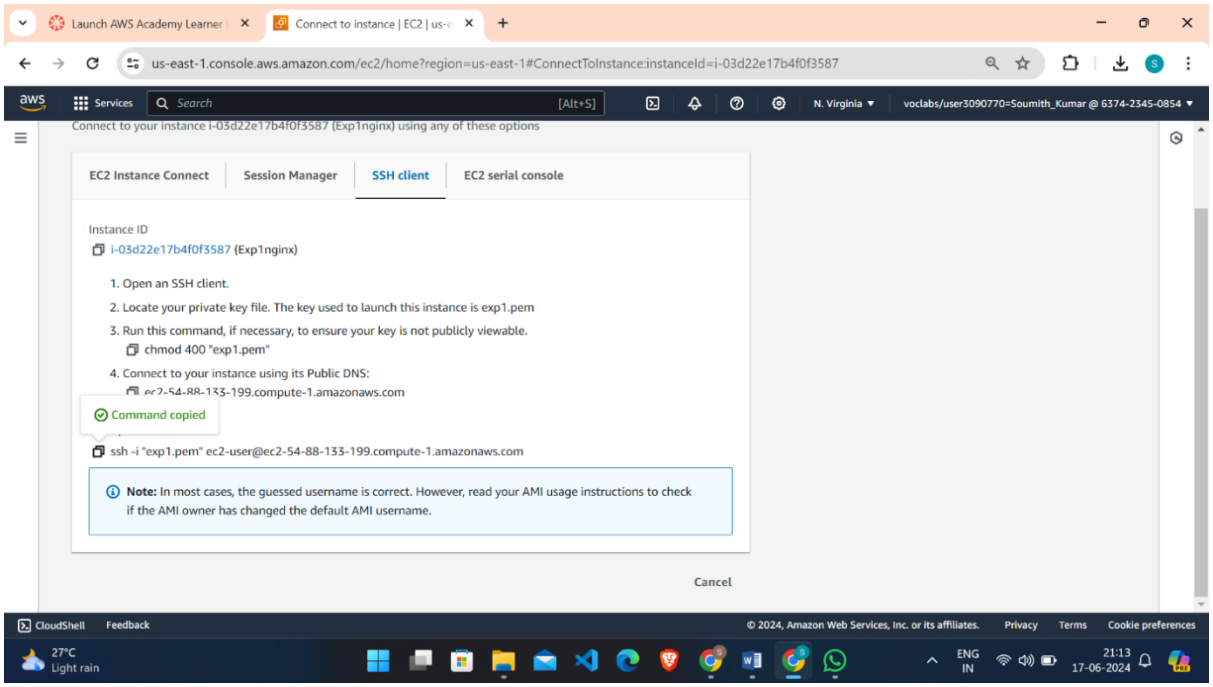
Step 9:Click on the checkboxes Allow HTTPS and HTTP traffic from internet along with SSH under Network settings. Then keeping everything else default click on Launch instance.



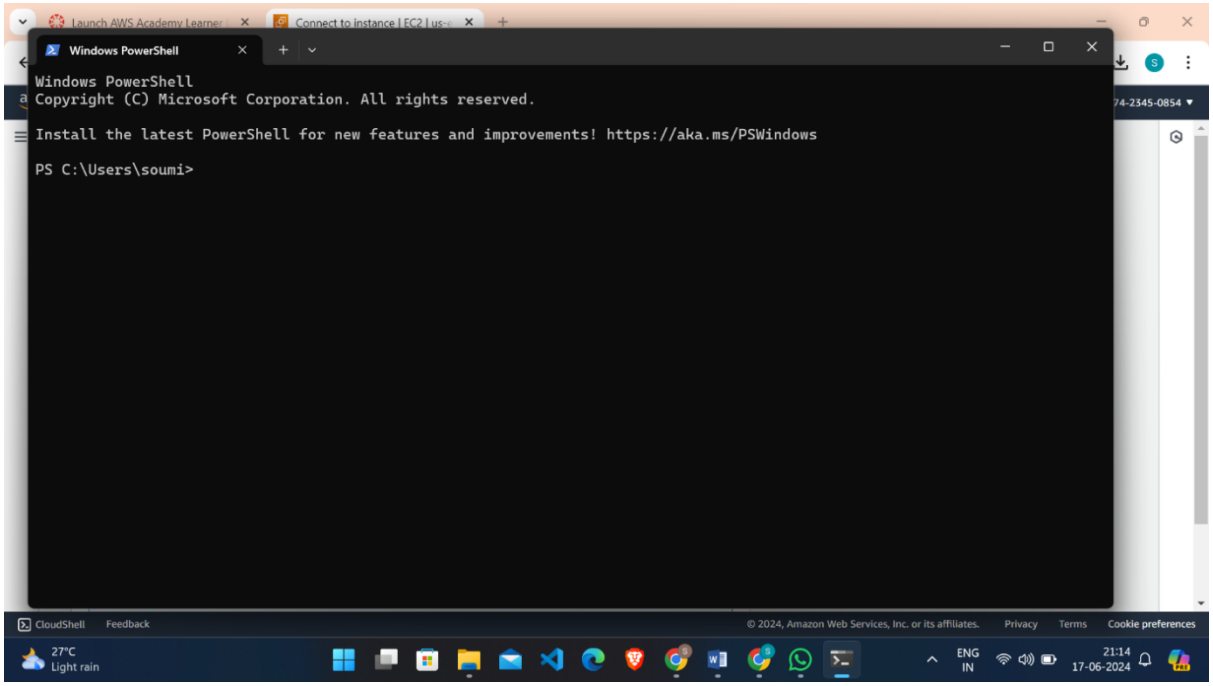
Step 10:Go to instances and select the given instance and click on connect after we see the instance state as running and 2/2 status checks



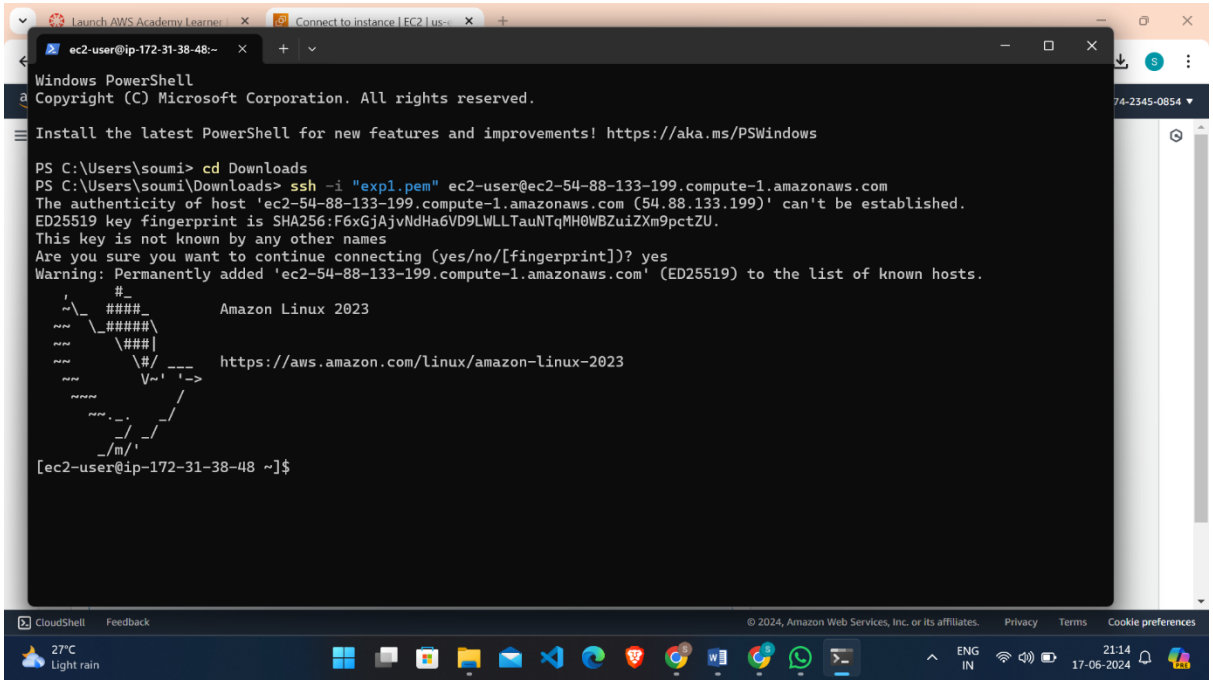
Step 11:Go to SSH client and copy the ssh key



Step 12: Now open PowerShell on desktop



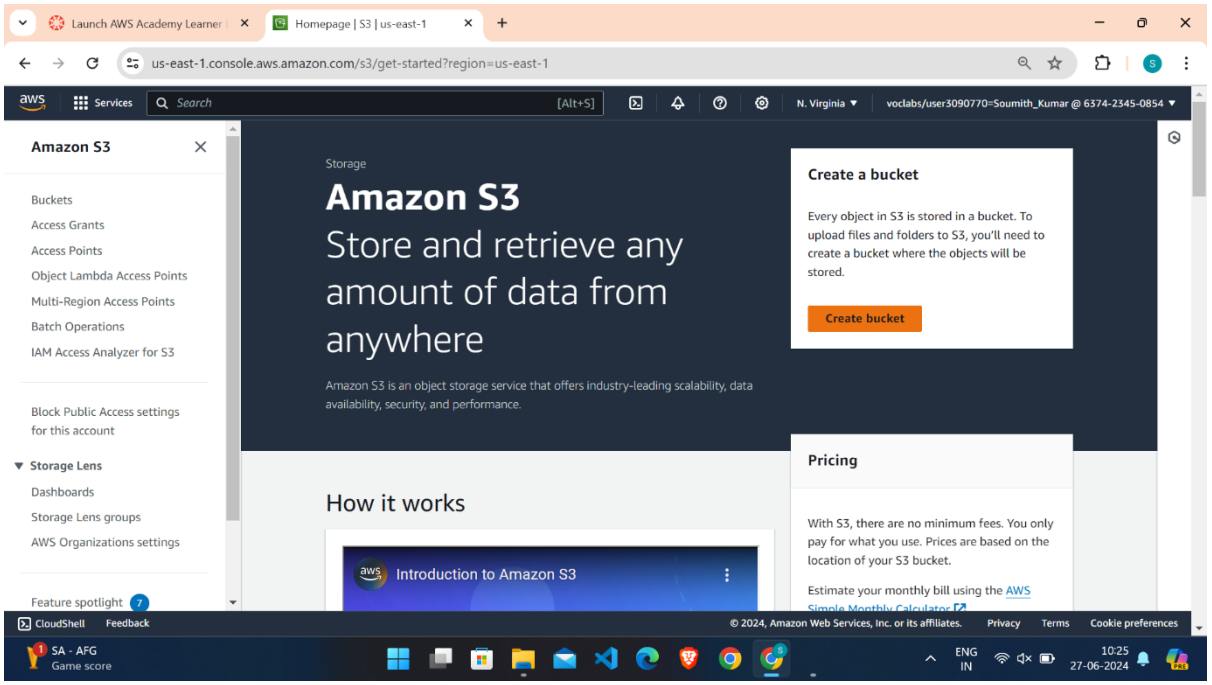
Step13: Change the directory where key.pem file is located by using cd and then paste the ssh key.



Experiment– 2

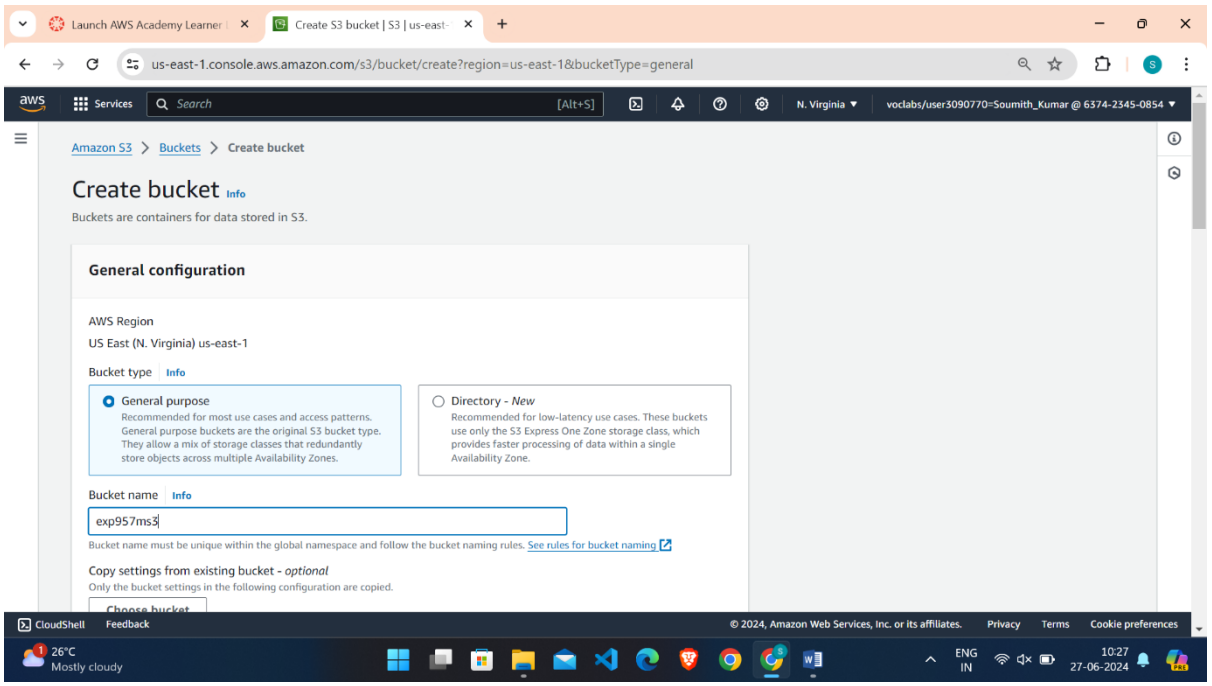
Aim : Create our First AWS S3 Bucket and Upload Content to Bucket and Manage their Access and Create Static Website using AWS S3

Step 1: Navigate to AWS console and search for S3 and click on create bucket.

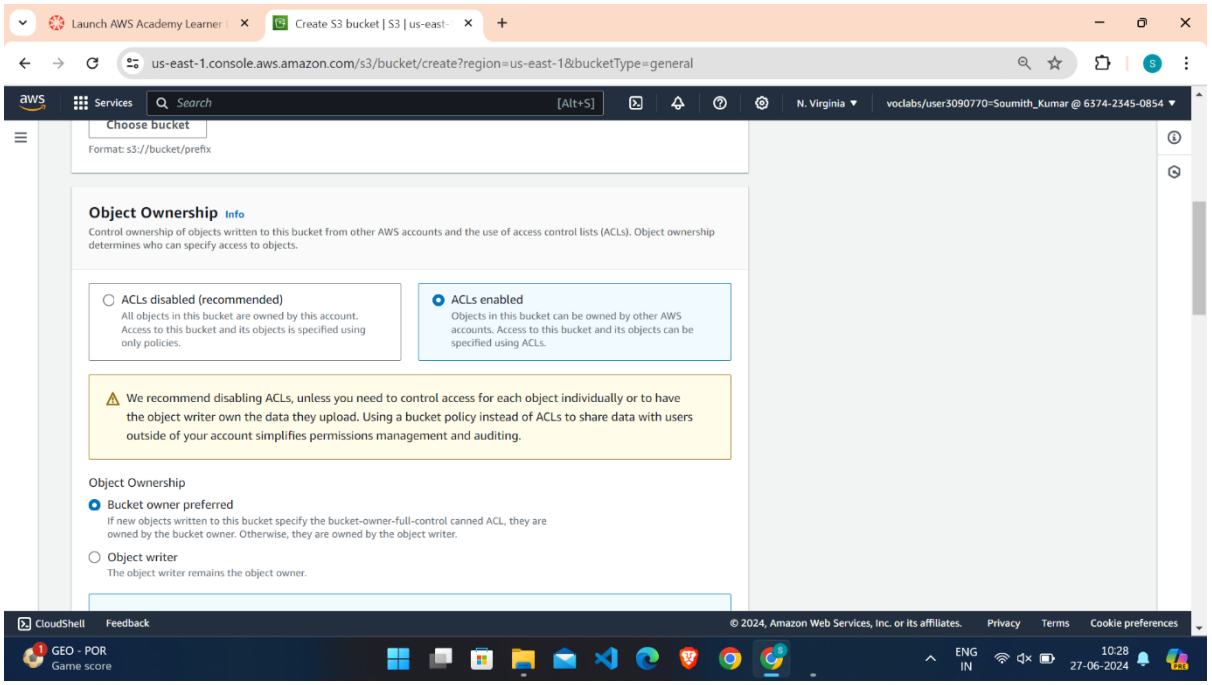


Step 2: Enter the following details

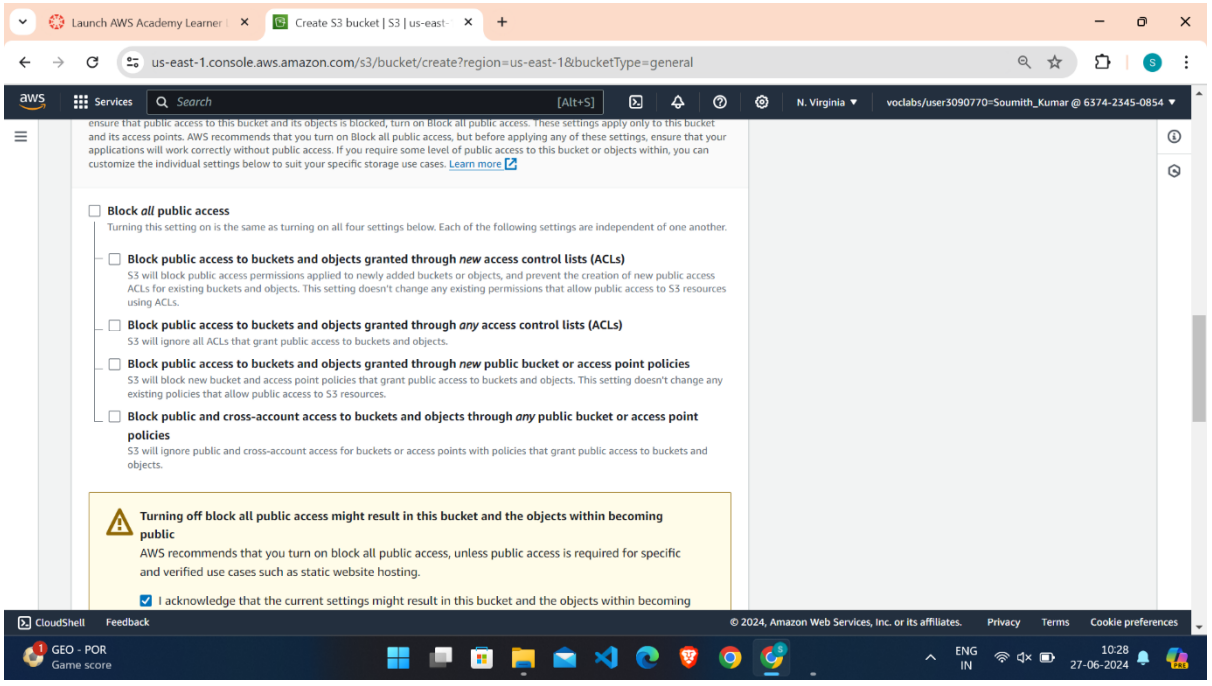
- Bucket type – General purpose
- Bucket name – must be unique within global namespace.



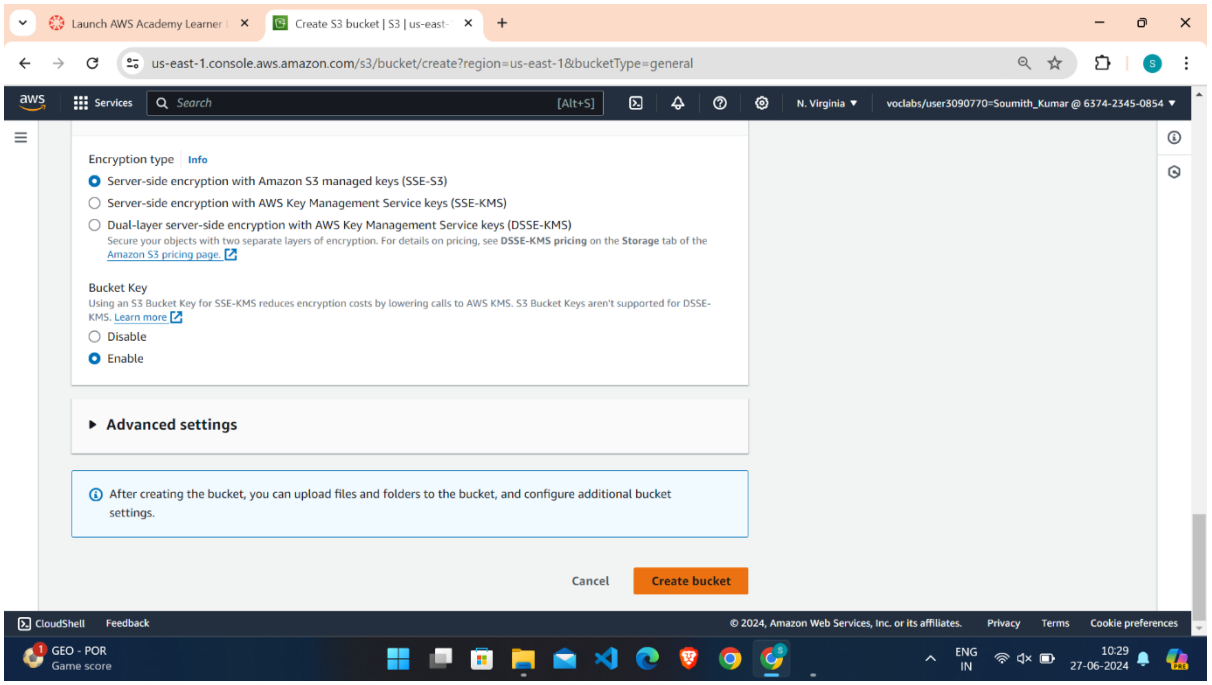
Step 3: Select Object Ownership as ACLs enabled



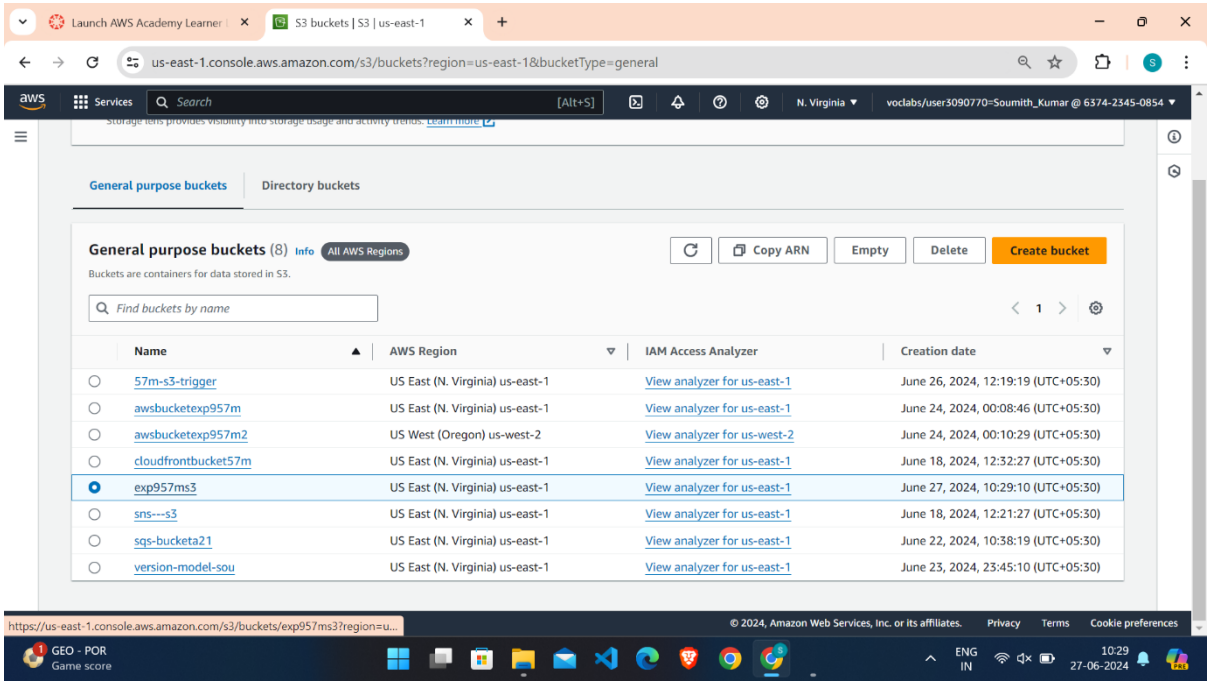
- Allow all public access



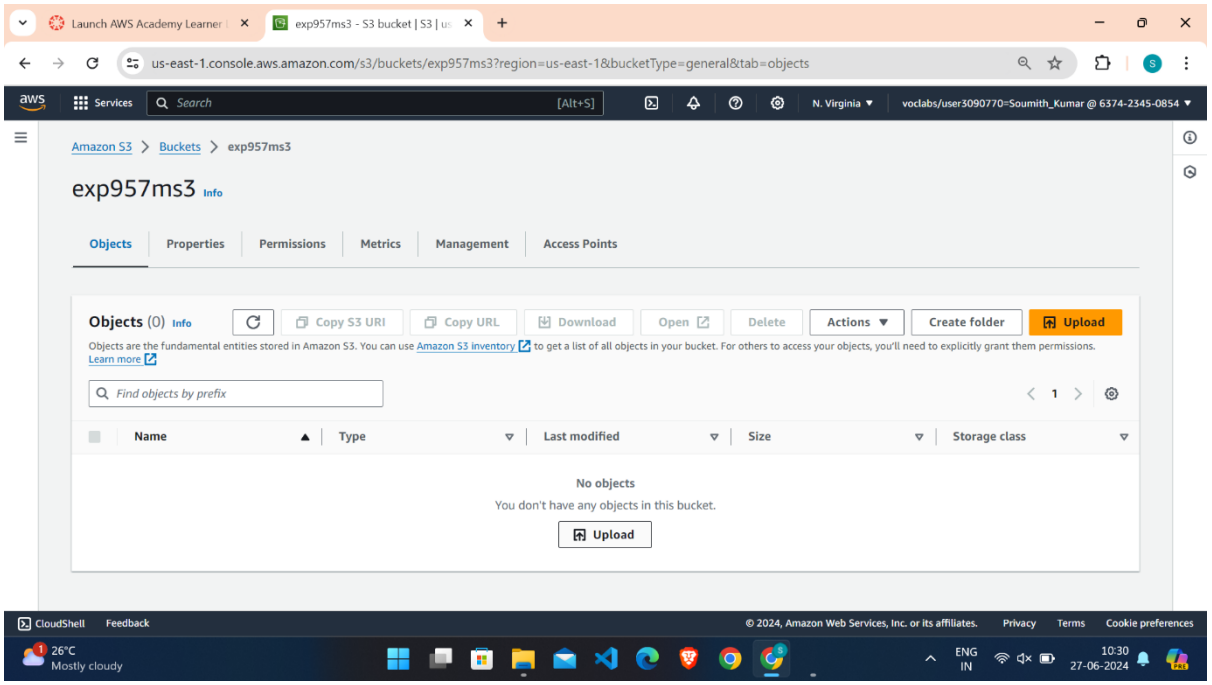
- Click on Create Bucket

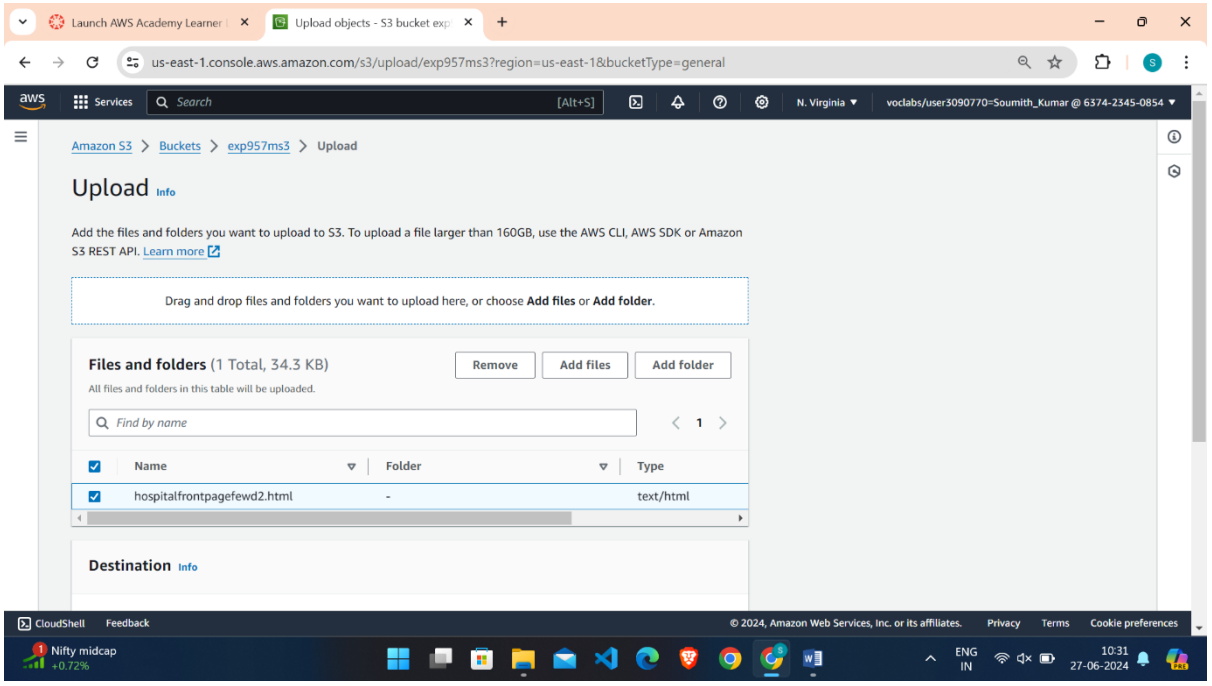


Step 4: Click on bucket

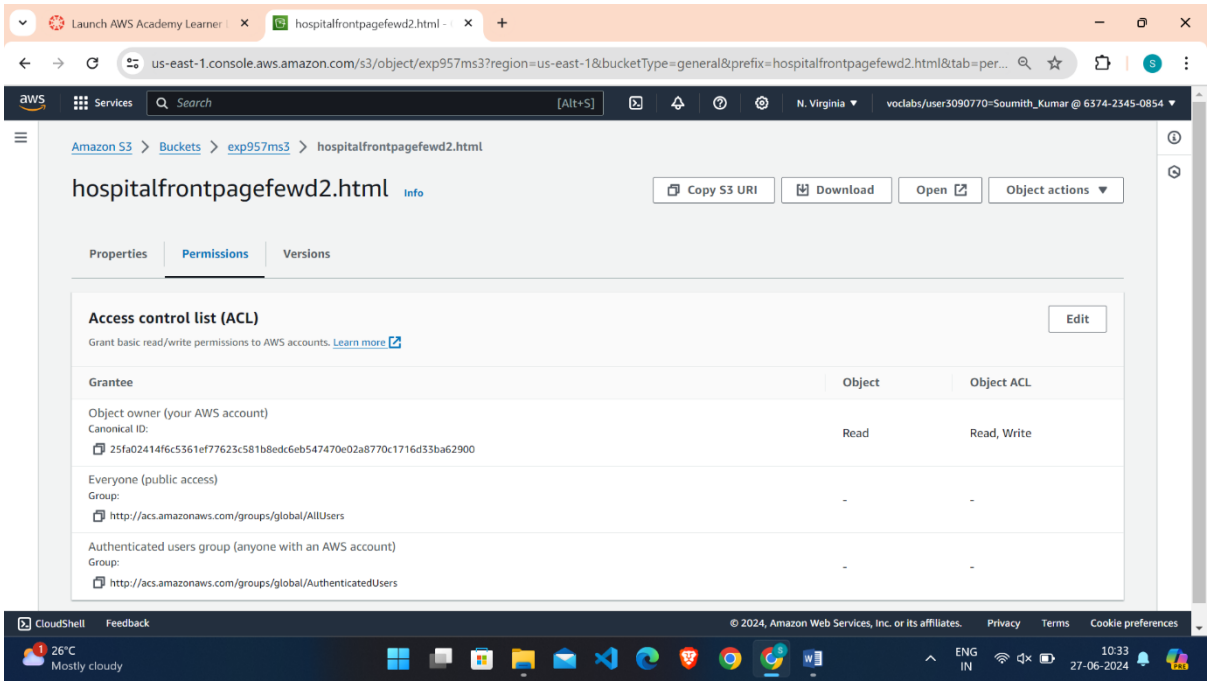


- Click on upload and select file to upload and click on upload.

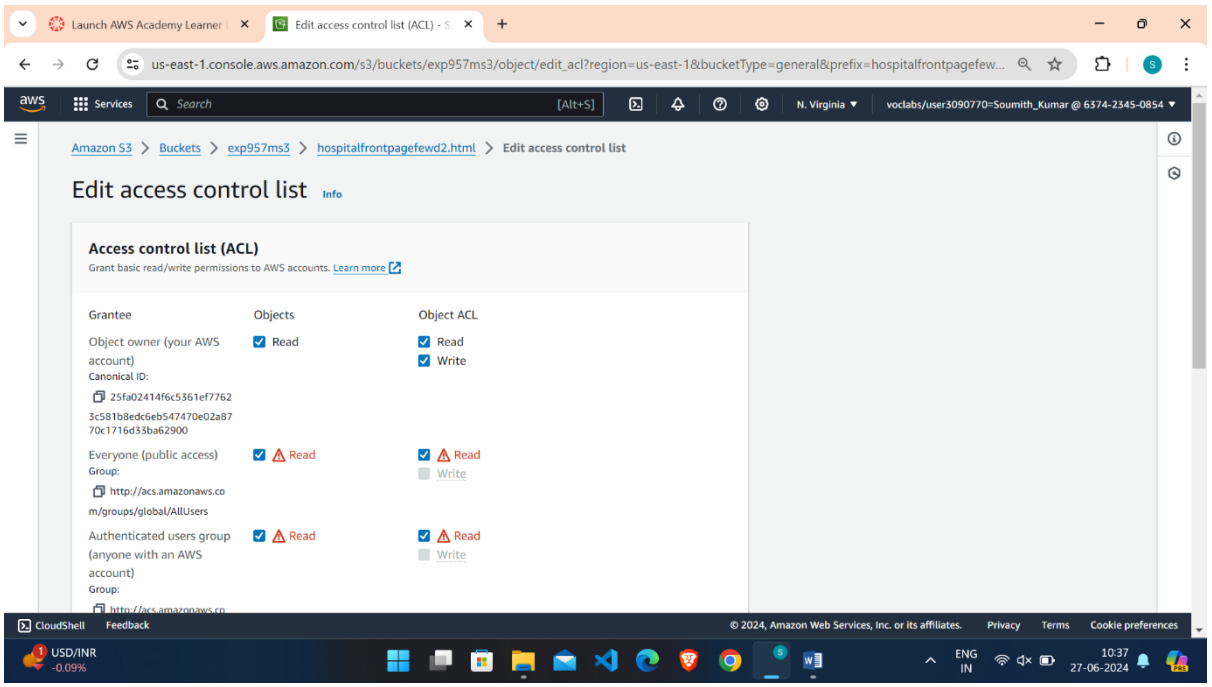

Launch AWS Academy Learner
exp957ms3 - S3 bucket | S3 | us
us-east-1.console.aws.amazon.com/s3/buckets/exp957ms3?region=us-east-1&bucketType=general&tab=objects
Services
Search
[Alt+S]
N. Virginia
voclabs/user3090770=Soumith_Kumar @ 6374-2345-0854
Amazon S3 > Buckets > exp957ms3
exp957ms3 Info
Objects
Properties
Permissions
Metrics
Management
Access Points
Objects (0) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions.
Learn more
Find objects by prefix
Name
Type
Last modified
Size
Storage class
No objects
You don't have any objects in this bucket.
Upload
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
26°C
Mostly cloudy
ENG
IN
10:30
27-06-2024


Launch AWS Academy Learner
Upload objects - S3 bucket exp
us-east-1.console.aws.amazon.com/s3/upload/exp957ms3?region=us-east-1&bucketType=general
Services
Search
[Alt+S]
N. Virginia
voclabs/user3090770=Soumith_Kumar @ 6374-2345-0854
Amazon S3 > Buckets > exp957ms3 > Upload
Upload Info
Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more
Drag and drop files and folders you want to upload here, or choose Add files or Add folder.
Files and folders (1 Total, 34.3 KB)
Remove
Add files
Add folder
All files and folders in this table will be uploaded.
Find by name
Name
Folder
Type
hospitalfrontpagefewd2.html
-
text/html
Destination Info
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Nifty midcap
+0.72%
ENG
IN
10:31
27-06-2024

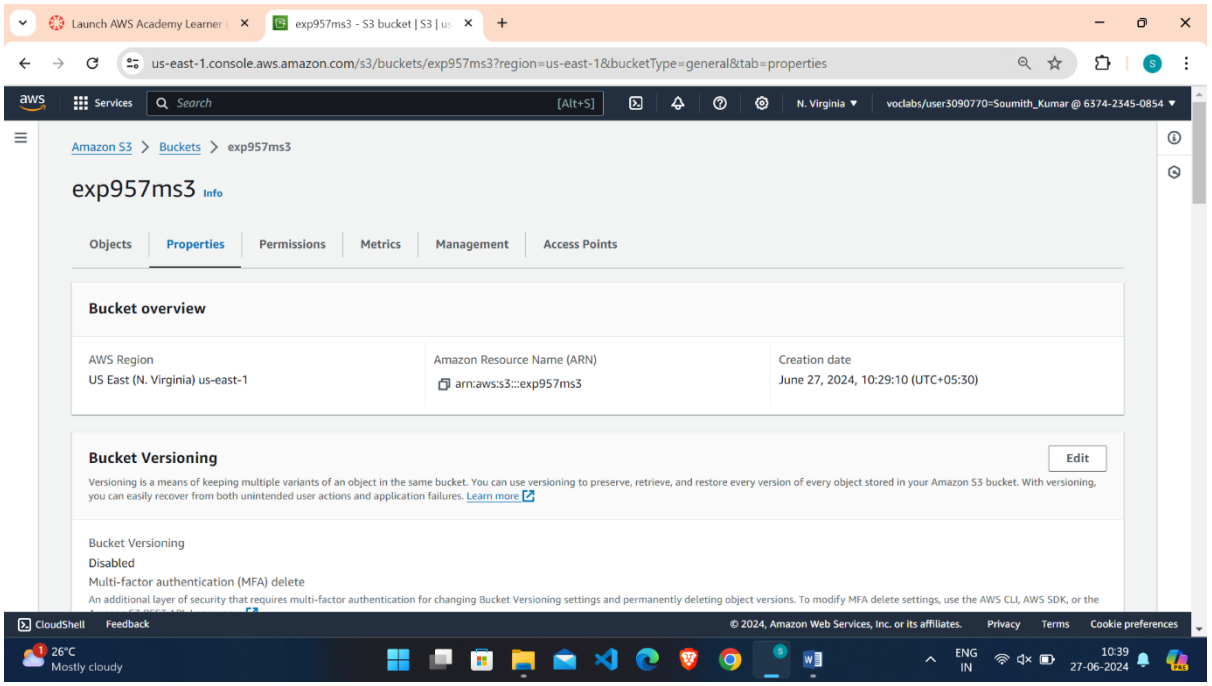
Step 5: Select the file and go to Permissions tab.



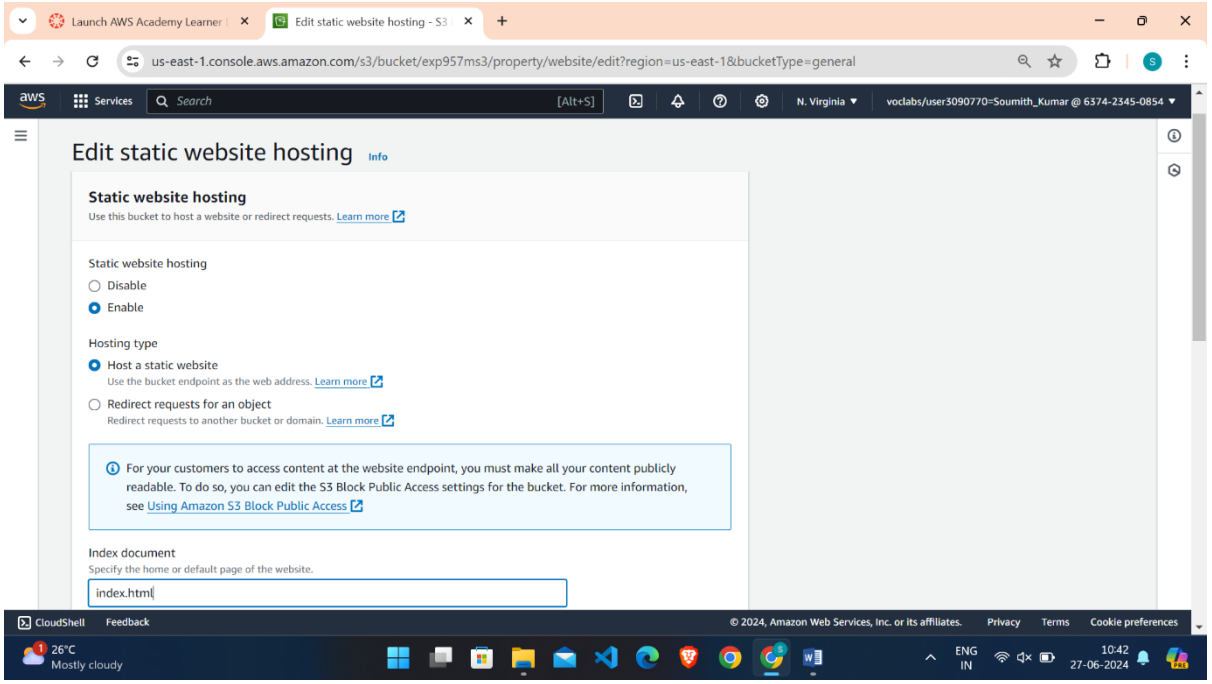
Step 6: Click on edit and allow all the Object and Object ACL permissions and click on save changes



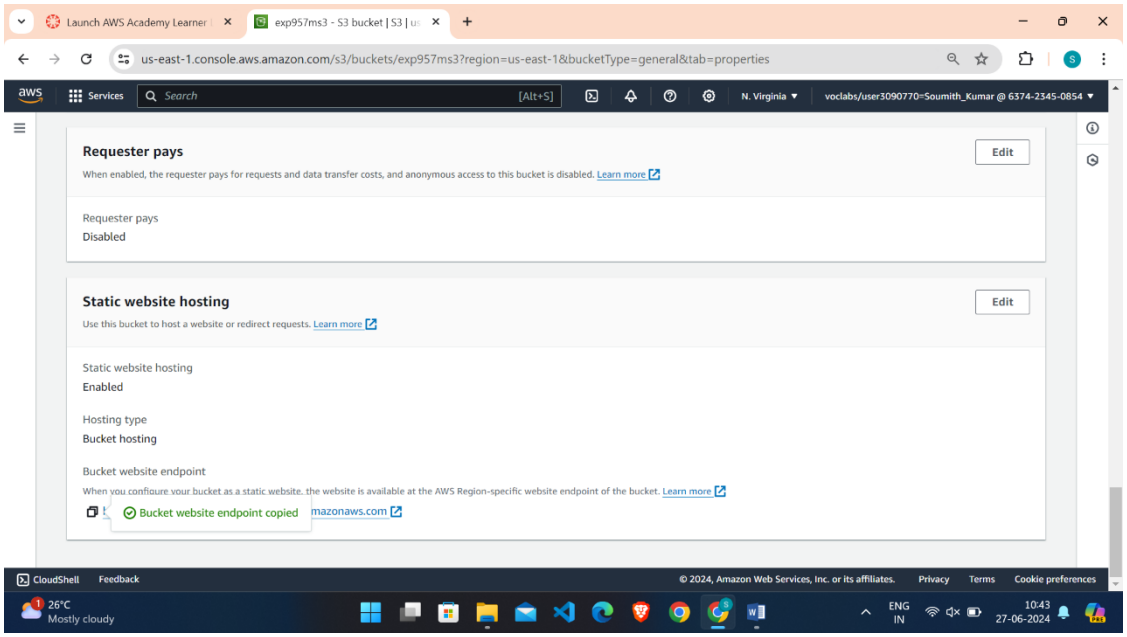
Step 7: Select the bucket and go to properties tab.



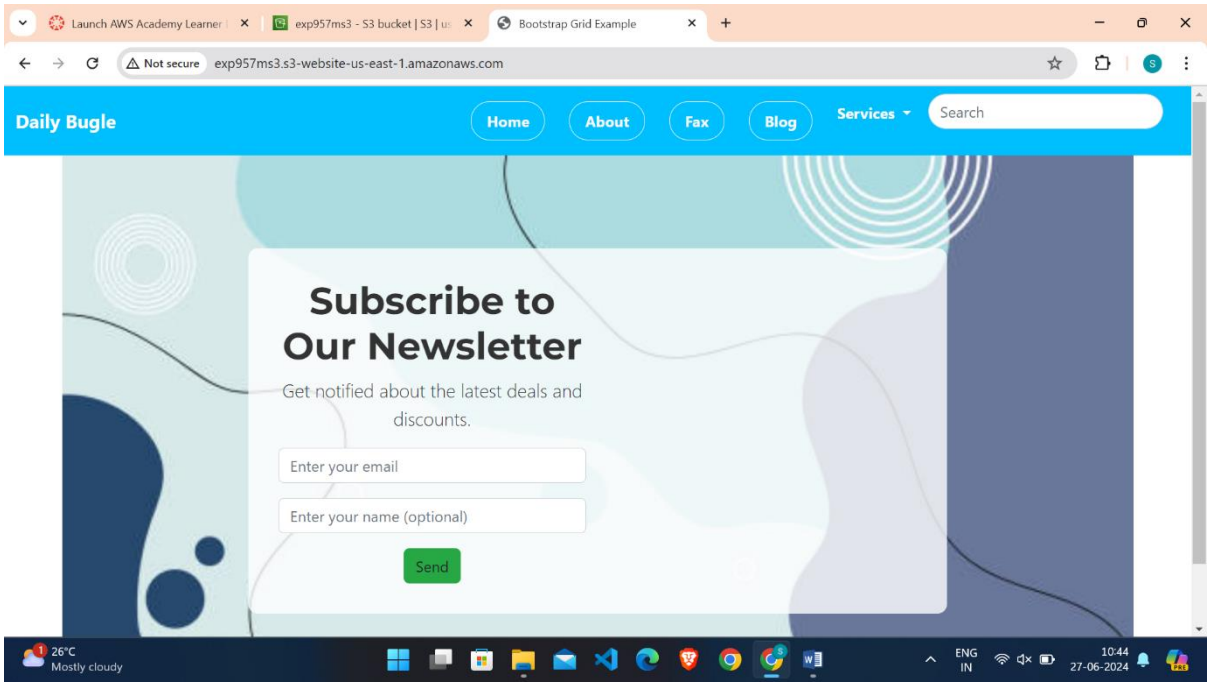
Step 8: Scroll down for Static web hosting and click on edit. Enable the static web hosting and enter the index document name as “index.html”



Step 9: We can find a link of the static website copy it.



Step 10: Paste it in new tab and we can access our file.

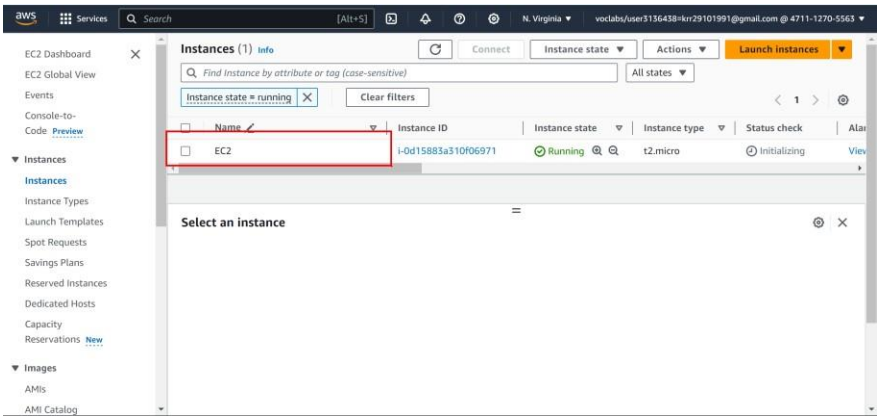


Experiment-3

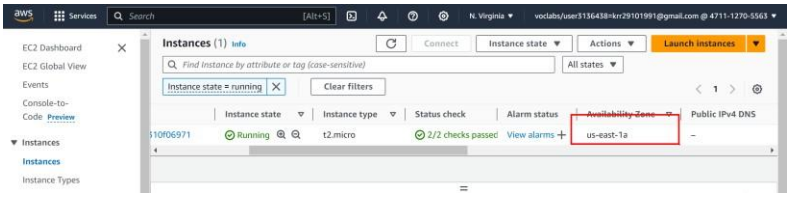
Aim: Exemplify the principle of rapid elasticity through a practical exercise involving the setup of an EC2 Amazon instance and the creation of multiple Elastic Block Store (EBS) volumes to be attached to EC2 instance.

→Attach a new Volume to EC2

Step 1: Launch an EC2 instance

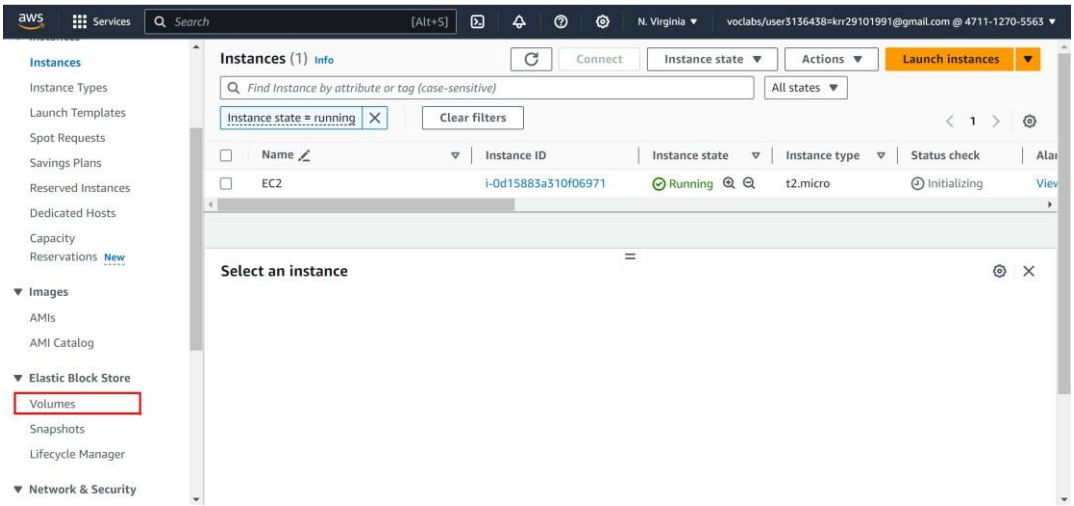


Step 2: Check the Availability Zone === In this case its us-east-1a

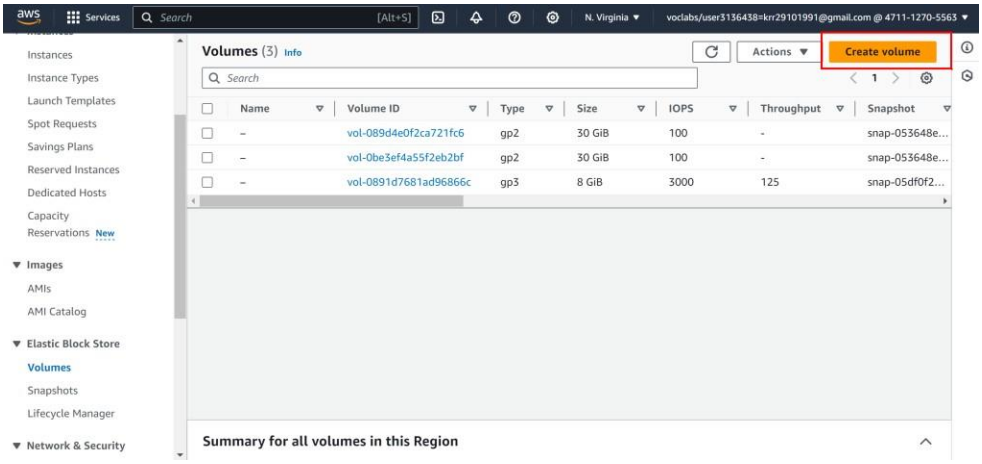


- Create a volume

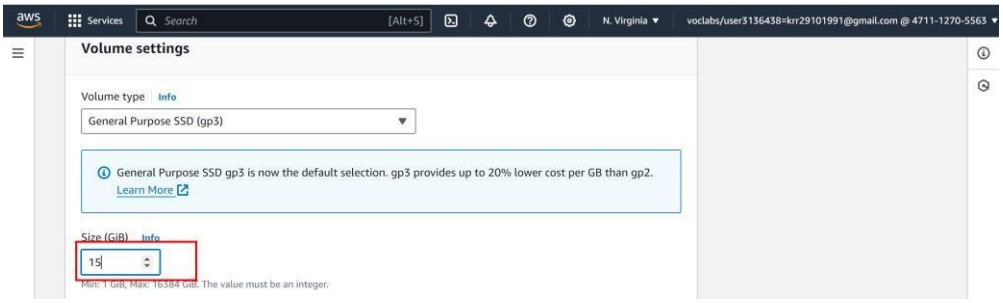
Step 3:Click on volume From Elastic Block Store



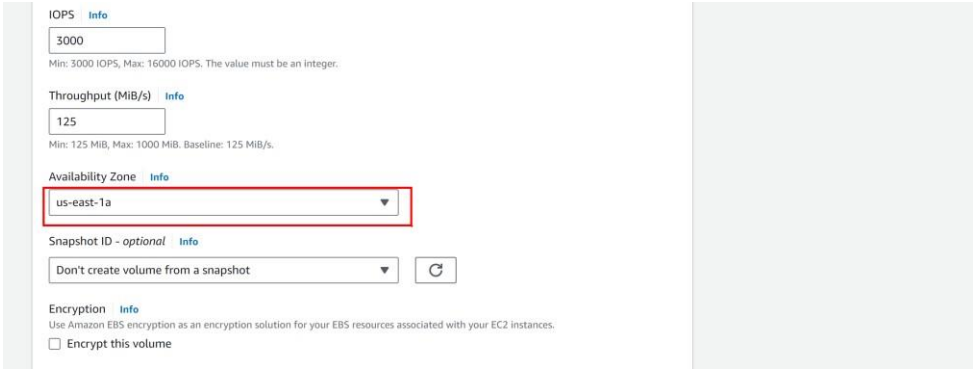
Step 4: Click on create Volume



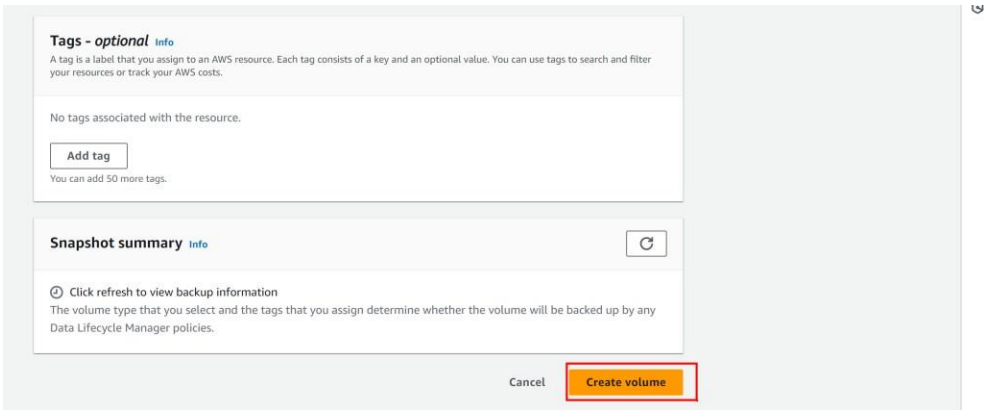
Step 5 :Change the size to required capacity == In this case I am using 15 GB

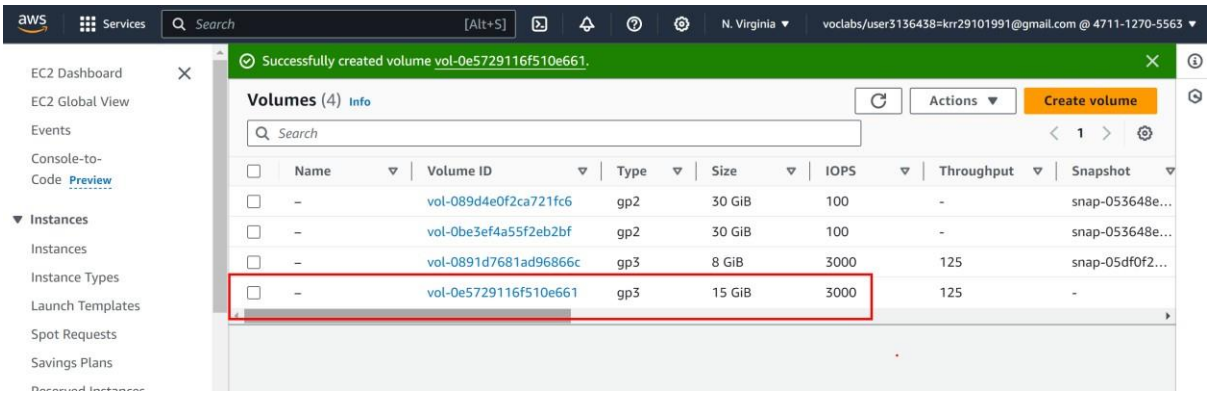


Step 6: Check the Availability Zone == In this case it should be us-east-1a (as EC2 is launched in that AZ)



Step 7: Click on Create Volume

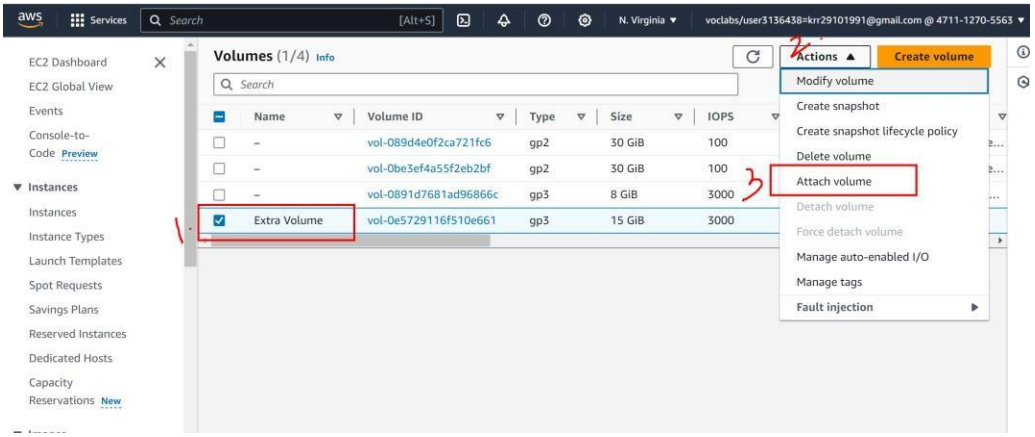




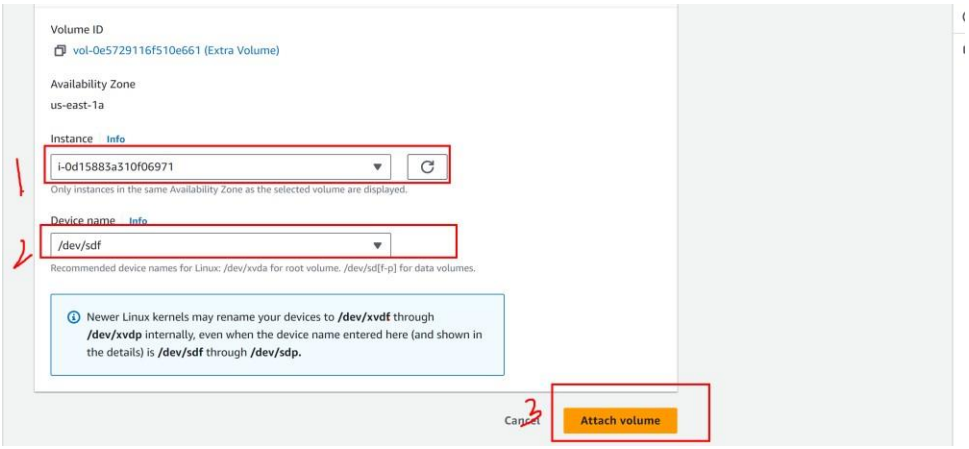
Name the Volume == to easily recognize == IN this case I will name as Extra Volume

Step 10: Now attach the volume to the EC2 Instance

I. Select the Extra Volume
II. Drop down the Actions
III. Click on Attach Volume

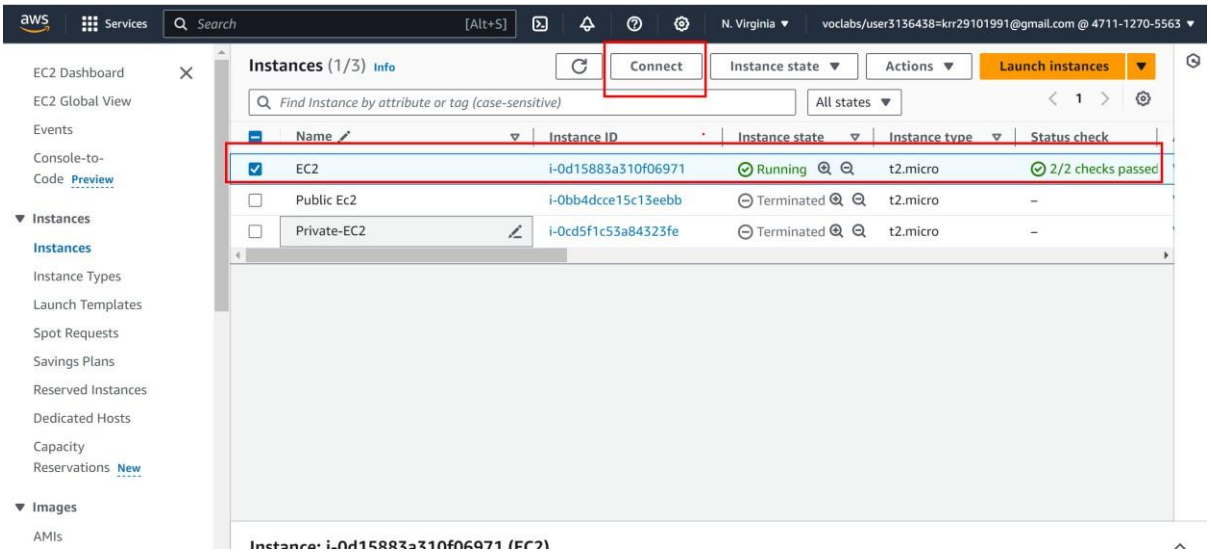


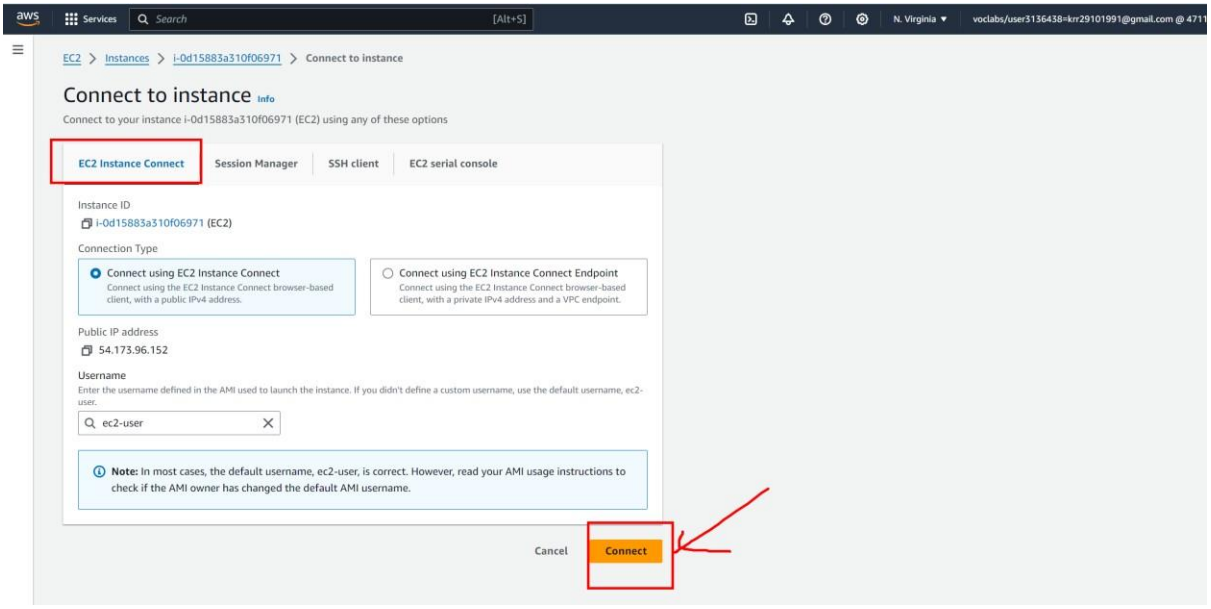
IV. Select your ec2 instance
V. Name the Device Name as /dev/sdf
VI. Click on Attach Volume



Step 11: Now Connect to the EC2 Instance named “EC2”

a) Select the instance
b) Click on connect



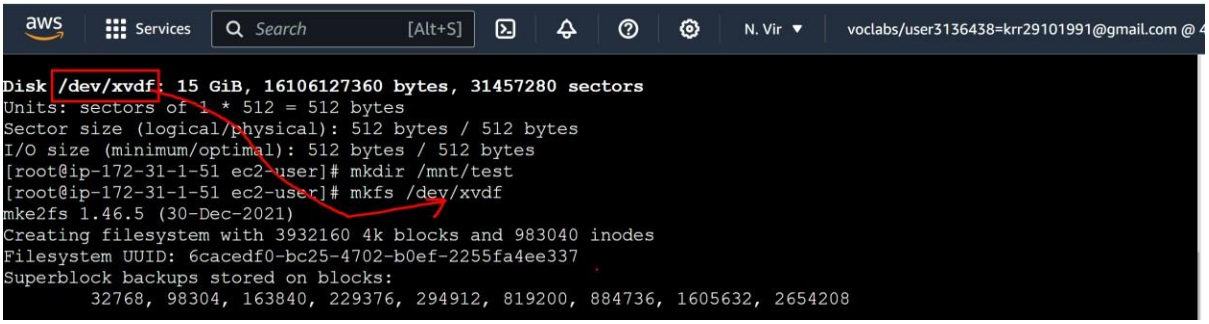


Step 12: Now run few commands to verify the disks attached to the Ec2 Instance

a) Change over to root user ==== sudo su
b) Check the disks ========== fdisk -l

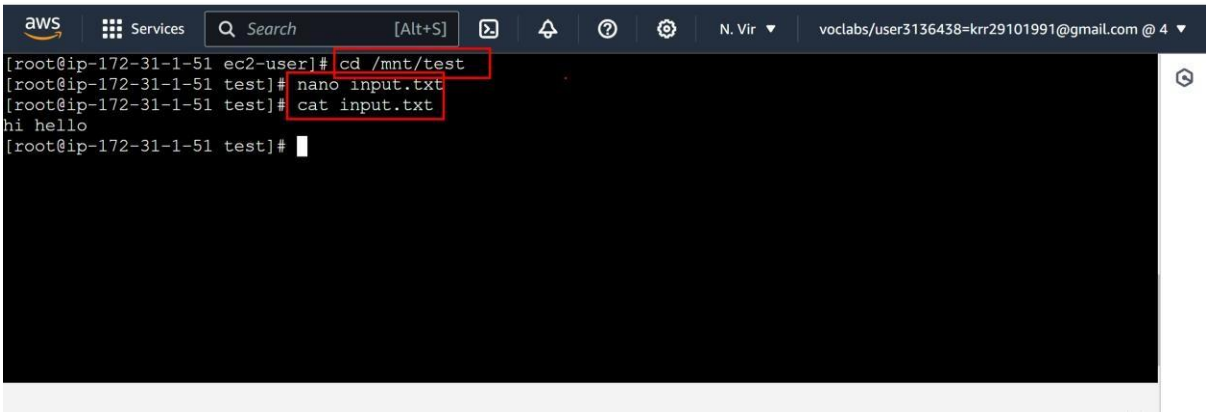
```
[ec2-user@ip-172-31-1-51 ~]$ sudo su
[root@ip-172-31-1-51 ec2-user]# fdisk -l

Disk /dev/xvdf: 15 GiB, 16106127360 bytes, 31457280 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

c) Create a directory ========== mkdir /mnt/test
d) Now format the newly attached disk ========== mkfs /dev/xvdf



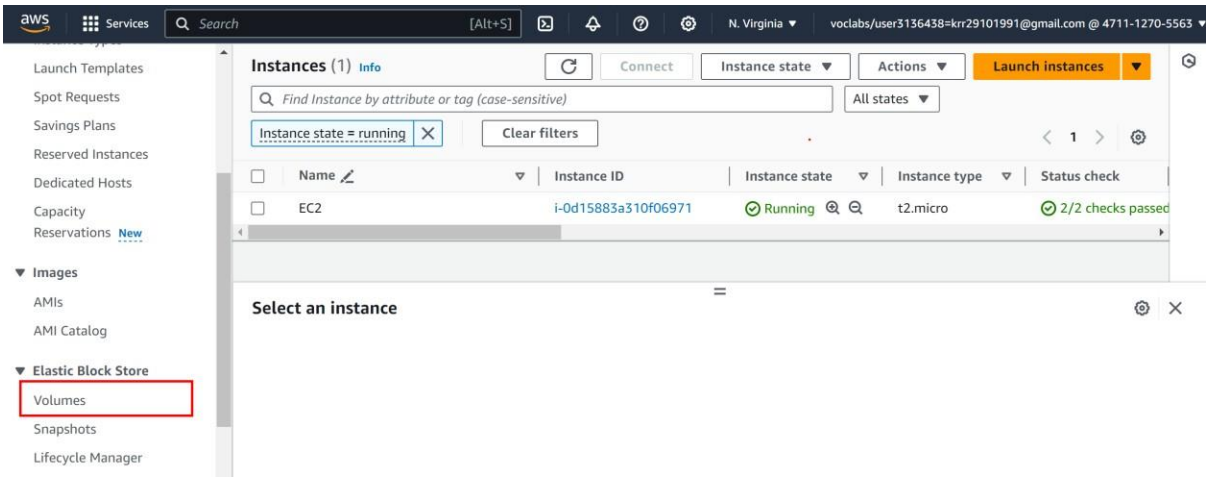
e) Now mount the disk on the directory

f) Now change the directory and create a file

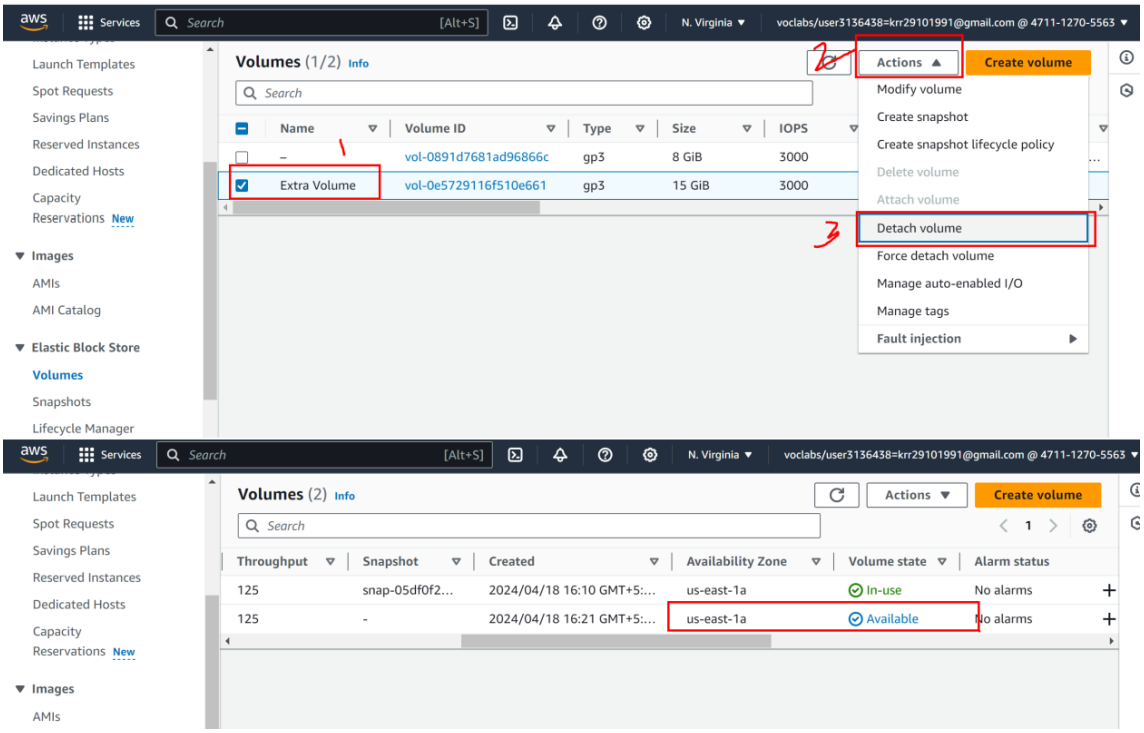


→Detach the new volume and Attach to a another EC2 Instance

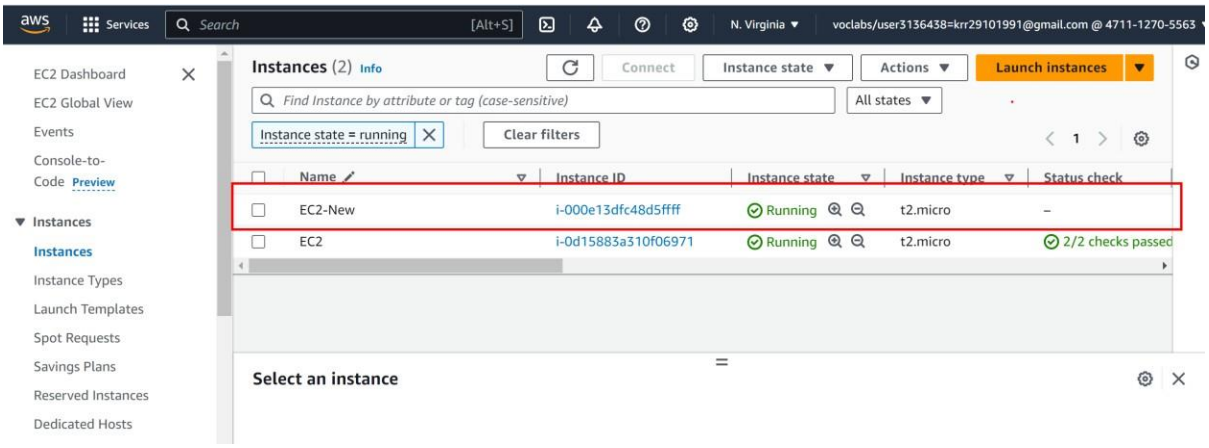
Step 1: Click on volumes



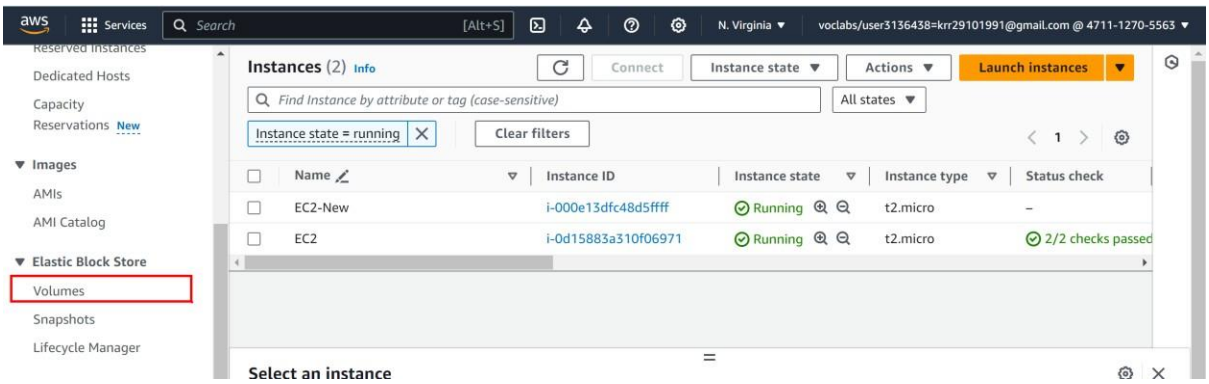
Step 2: Select the volume then drop down the Actions and click on detach volume then click ondetach and volume state needs to change from in-use to available



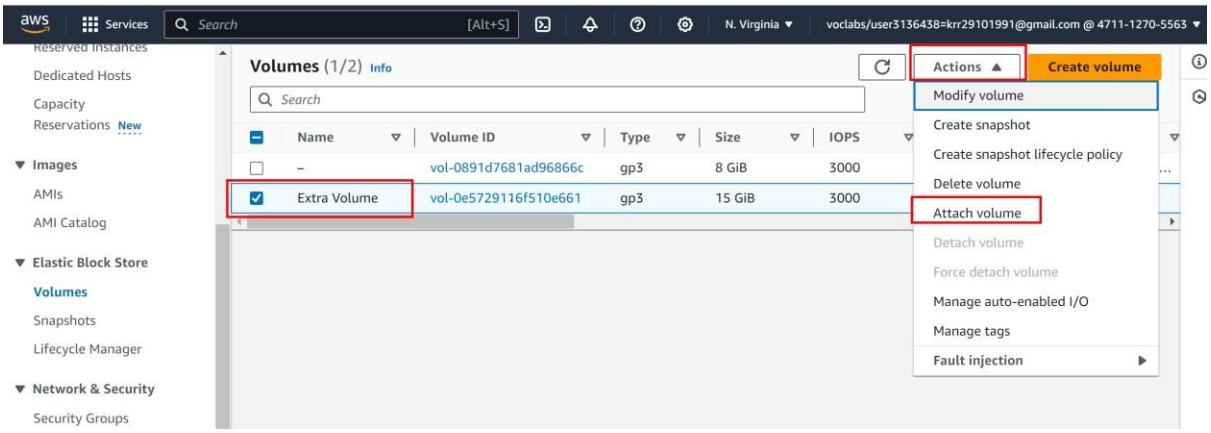
Step 3: Now go to EC2 dashboard and Launch another new instance named as EC2-New



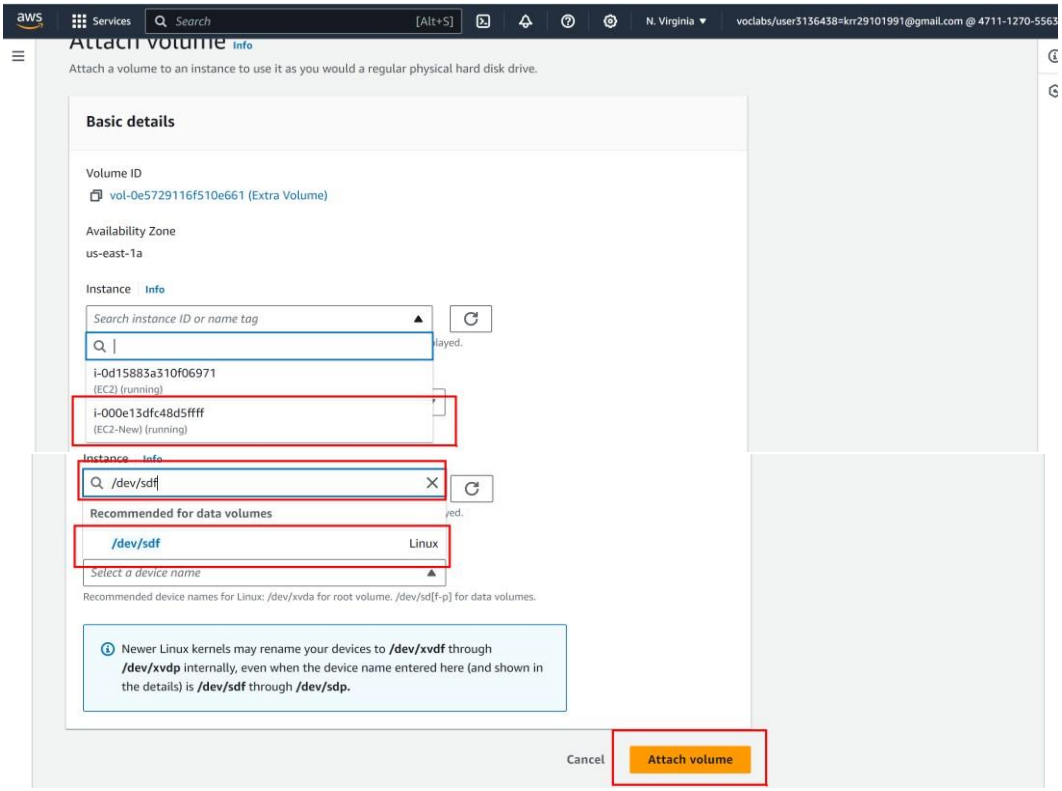
Step 4: Now click again on Volumes



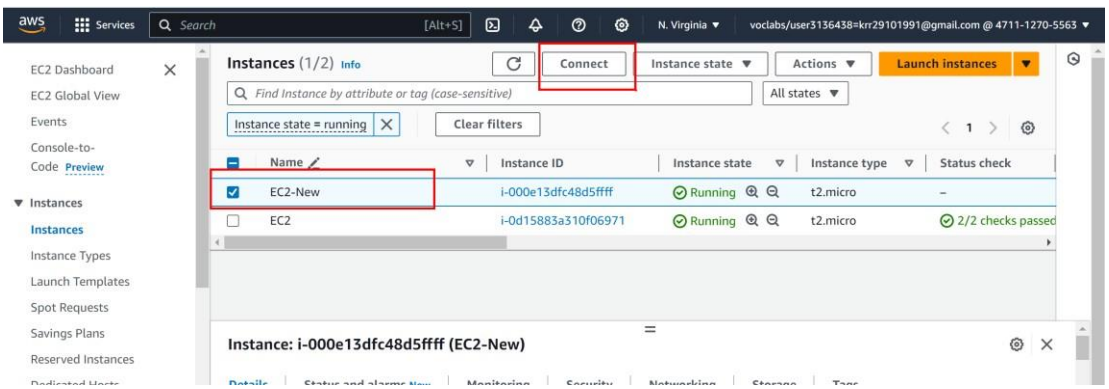
Step 5: Select the Extra Volume then drop down the Actions and click on Attach

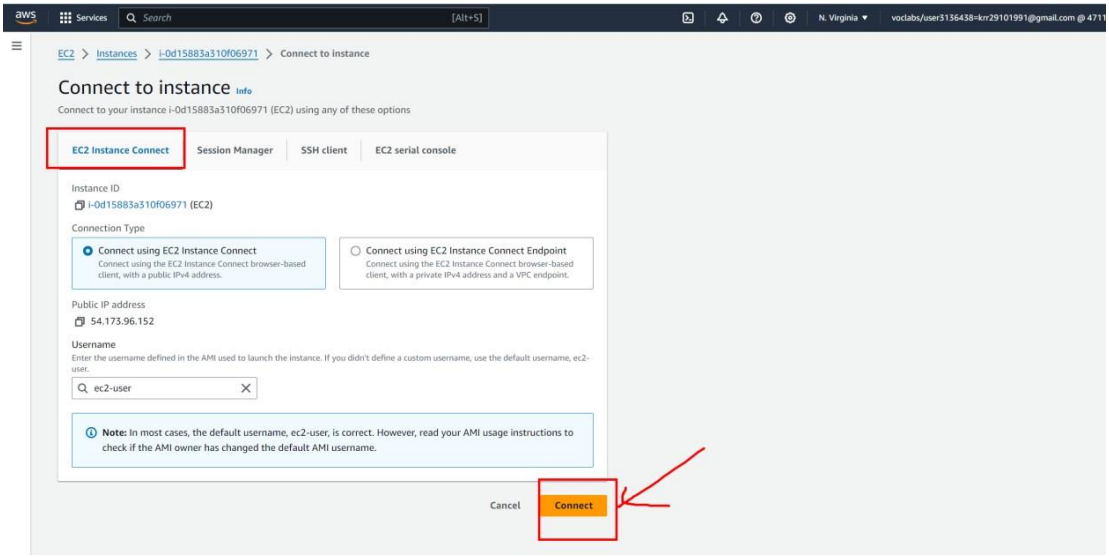


Step 6: Drop down and Select the EC2-New then search for Device name /dev/sdf and select the diskThen click on attach volume

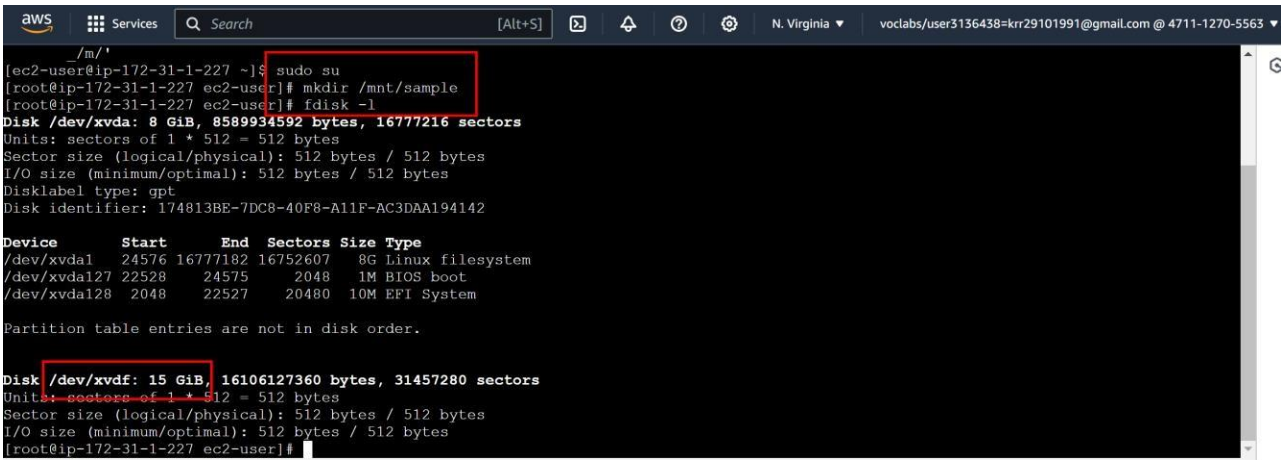


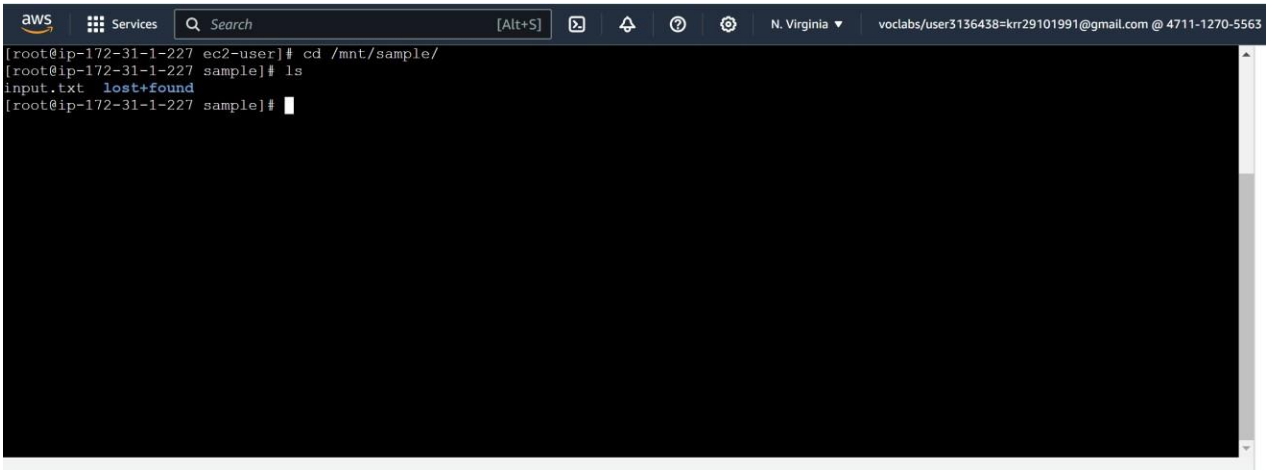
Step 7: Select the EC2-New Instance then Click on Connect





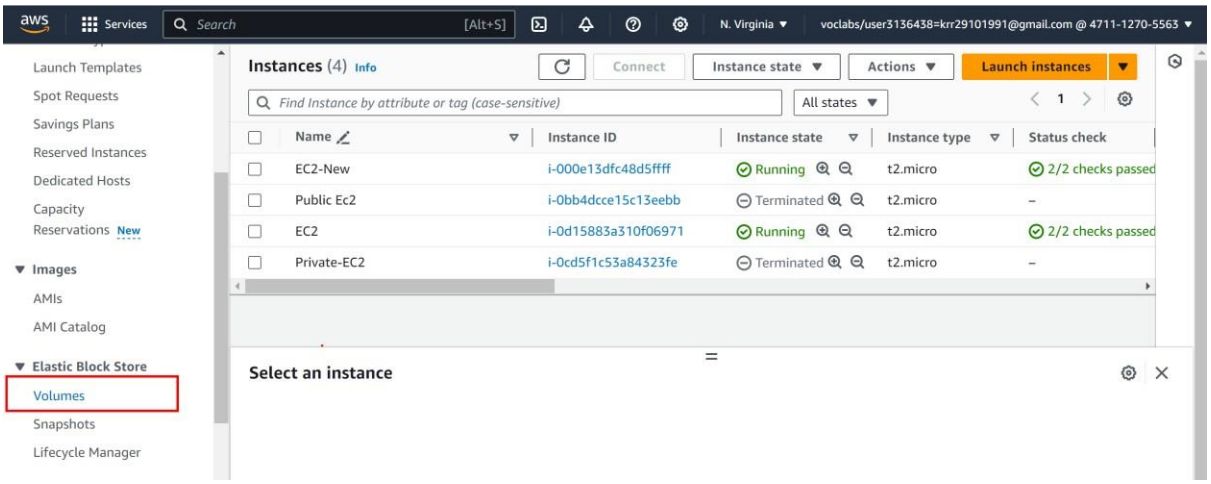
Step 8: Now change user to root then create a directory and verify the disk



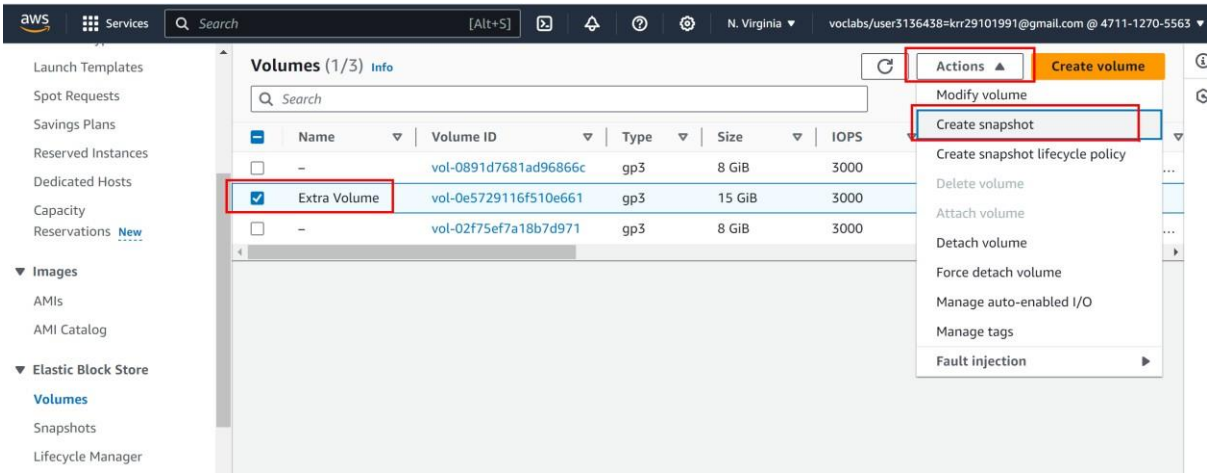


→Create a snapshot and copy to another region

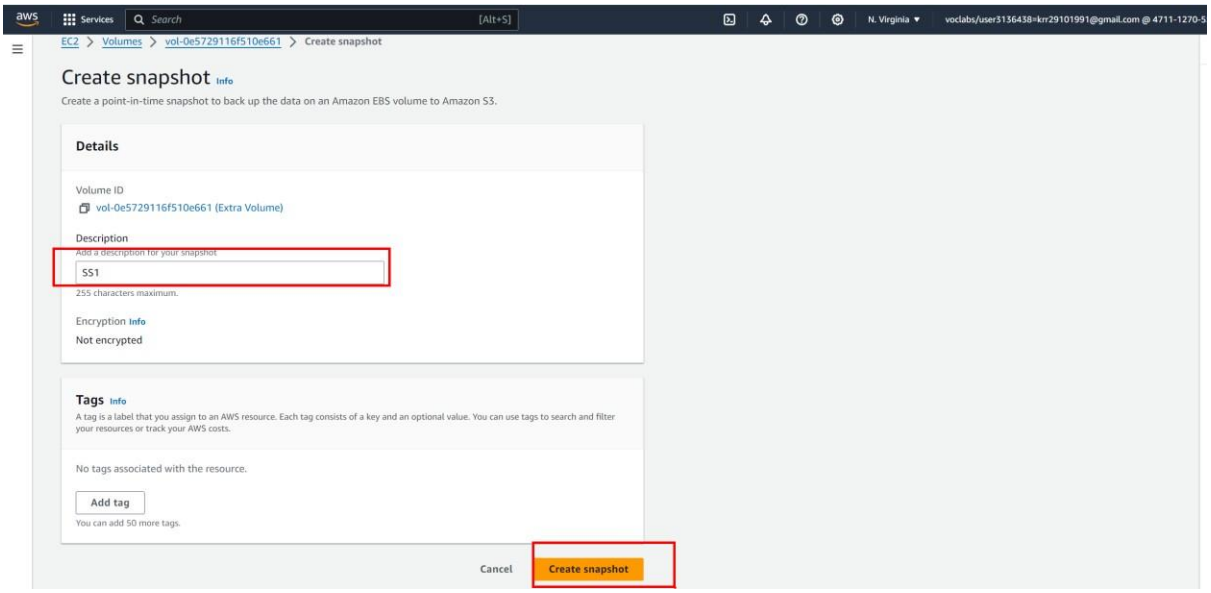
Step 1: Click on Volumes



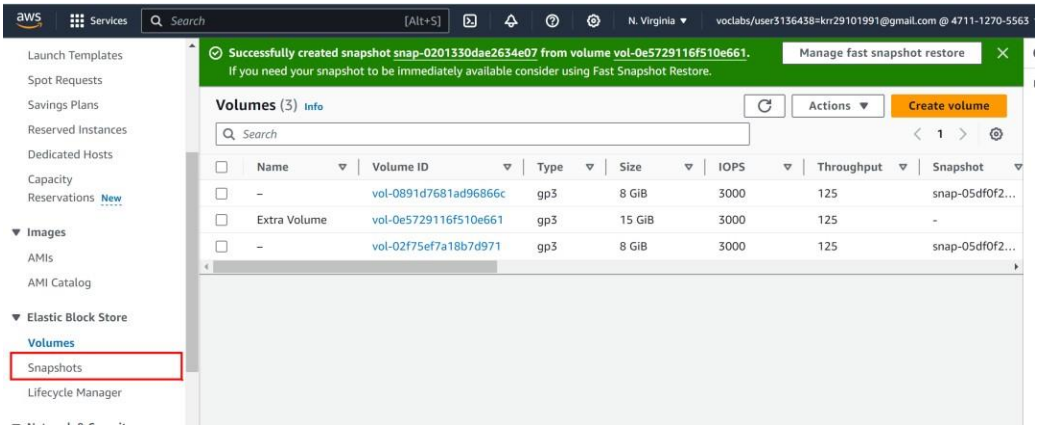
Step 2: Select the Extra Volume then Drop down the Actions and Click on Create Snapshot



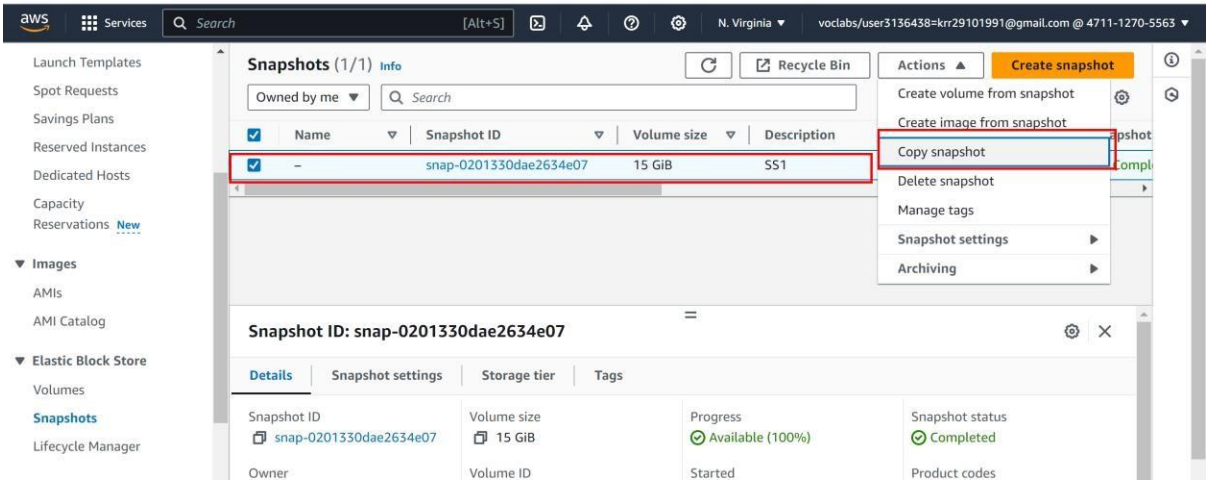
Step 3:Click on Create Snapshot



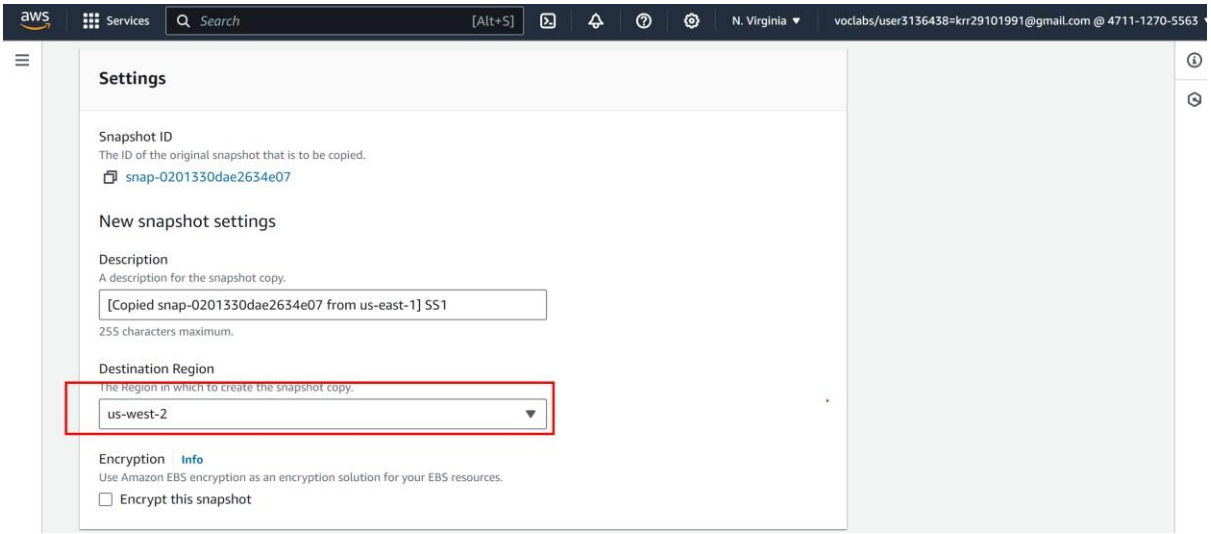
Step 4: Click on Snapshots

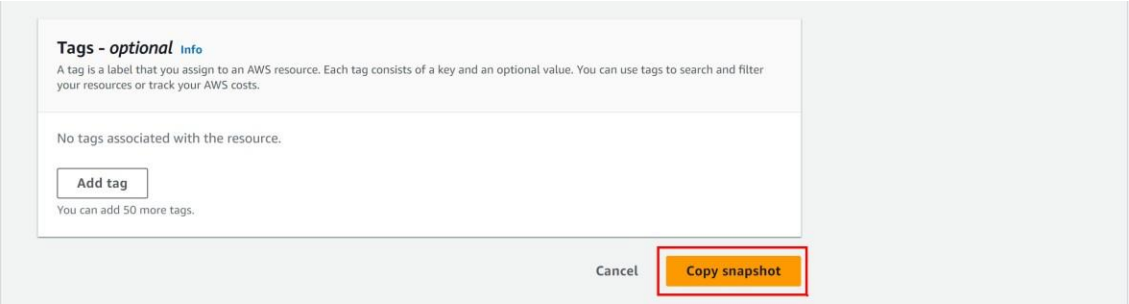


Step 5: Select the Snapshot and then drop-down Actions then click on Copy Snapshot presently weare in N. Virginia Region



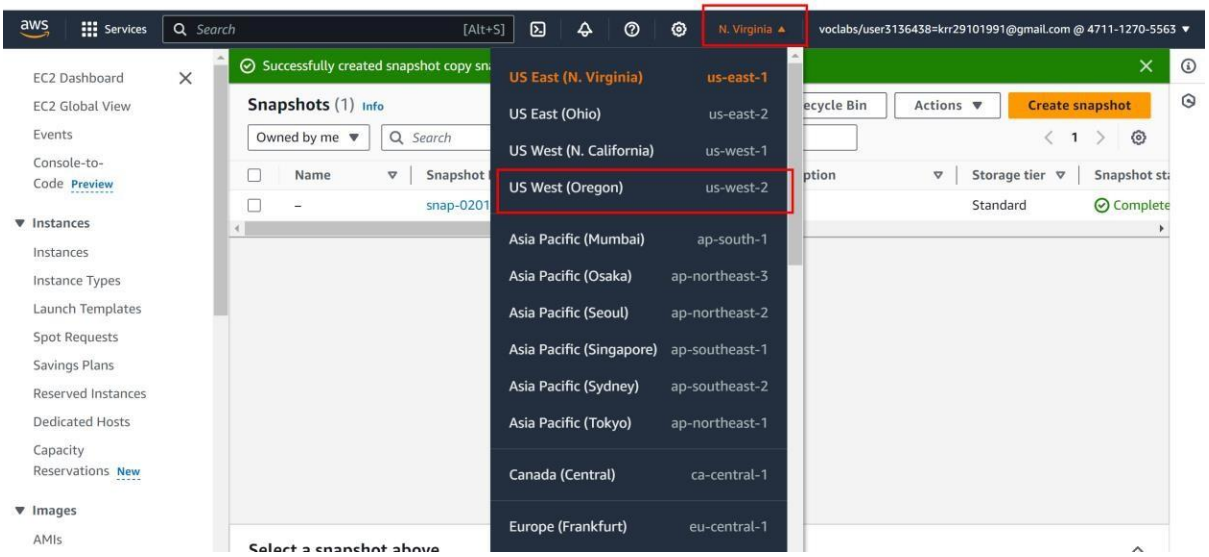
Step 6:Change the Destination only and only to Us-West-2 for AWS Academy users rest can whohave a free tier account can select any region then click Copy Snapshot

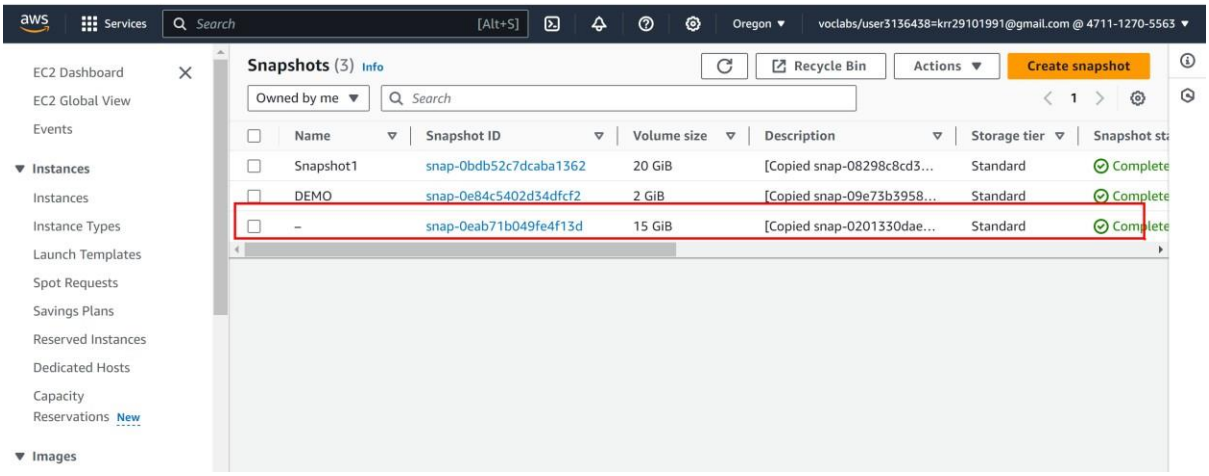




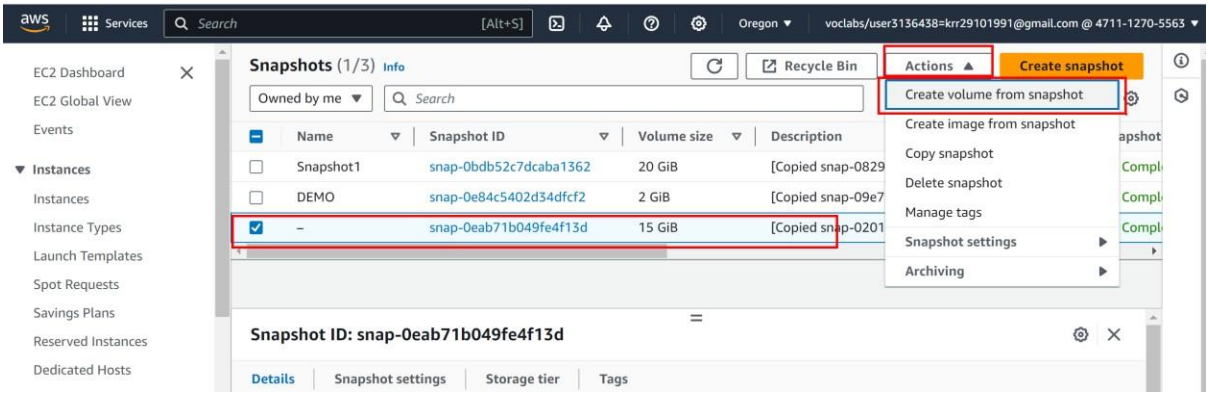
→Create volume from the snapshot and attach to an EC2 Instance

Step 1: Now change the Region from N. Virginia to Us-West-2 (Oregon)

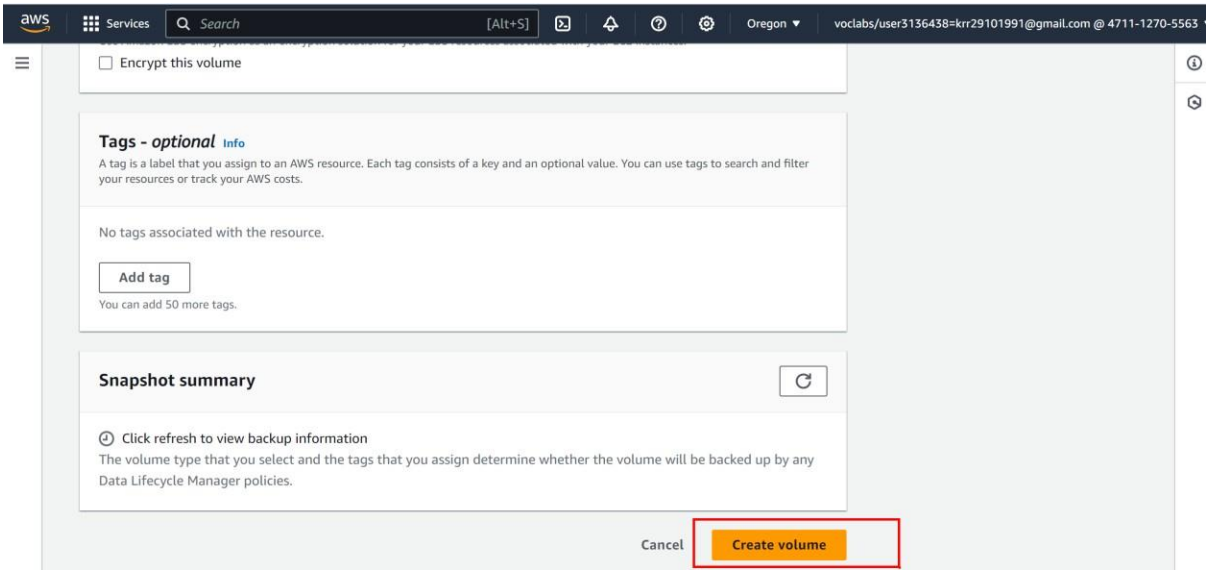




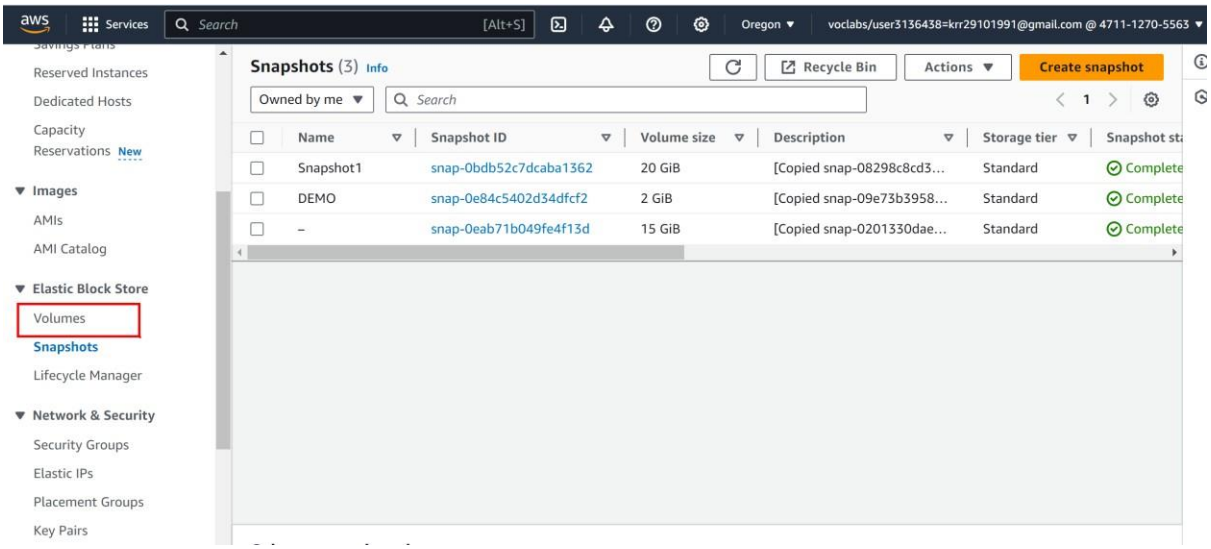
Step 2: Select the Copied Snapshot then drop down the actions and click on Create volume fromsnapshot

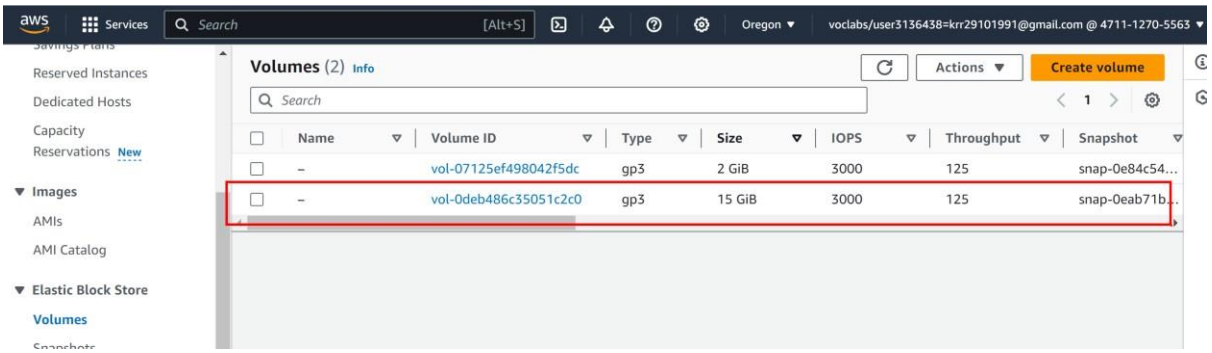


Step 3: Leave everything as default just click on Create Volume

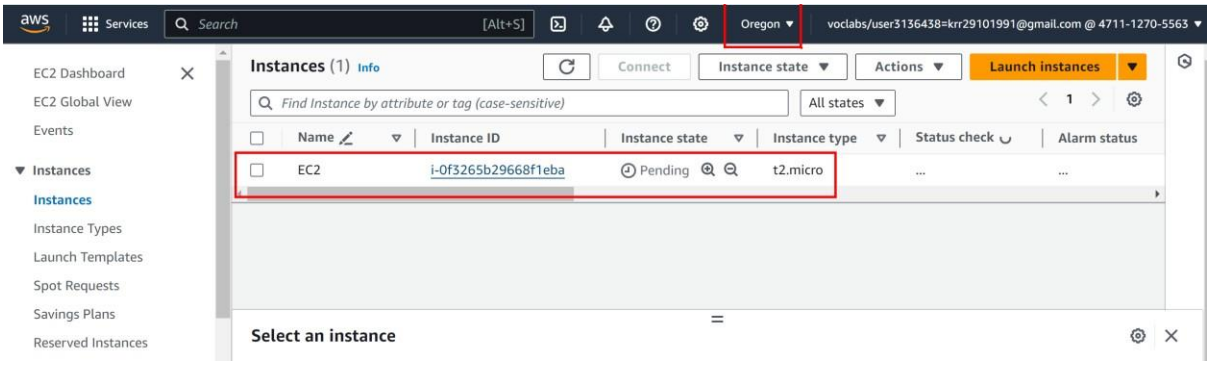


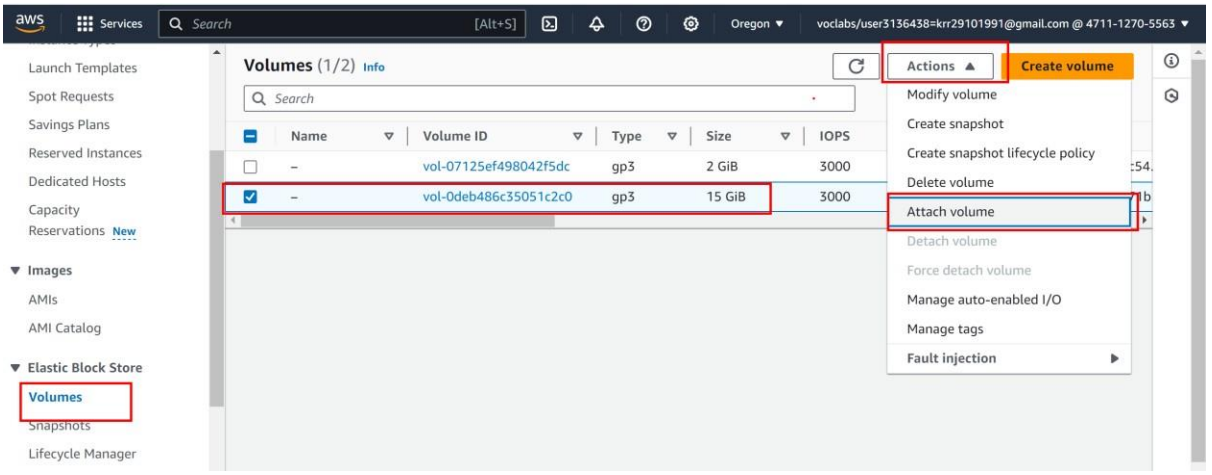
Step 4: Click on Volumes

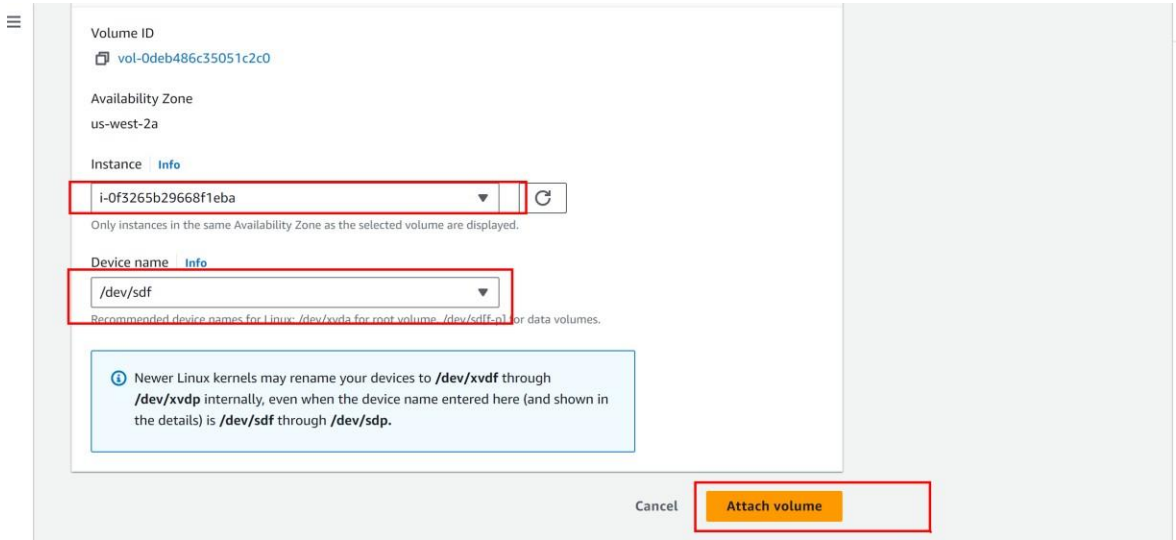


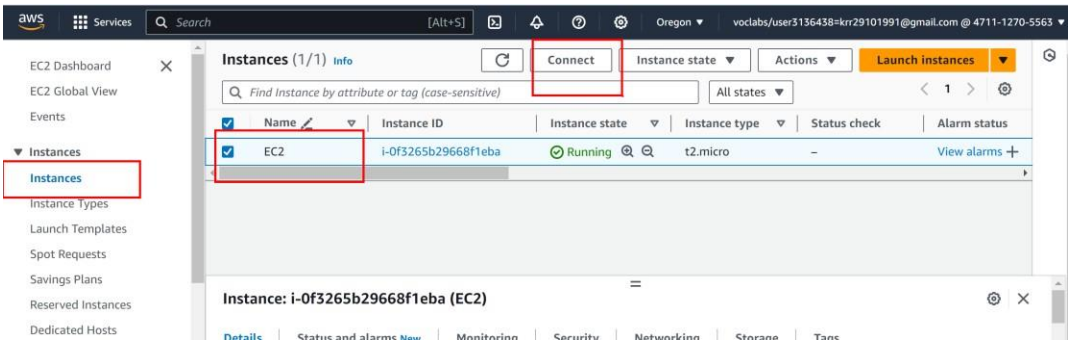


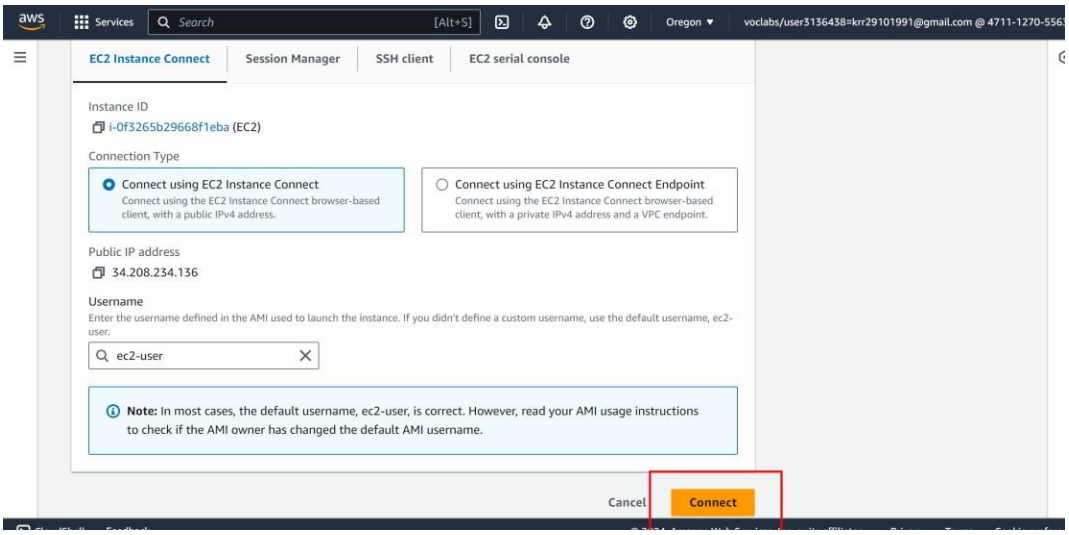
Step 5: Launch an EC2 Instance and attach the volume then connect to Instance

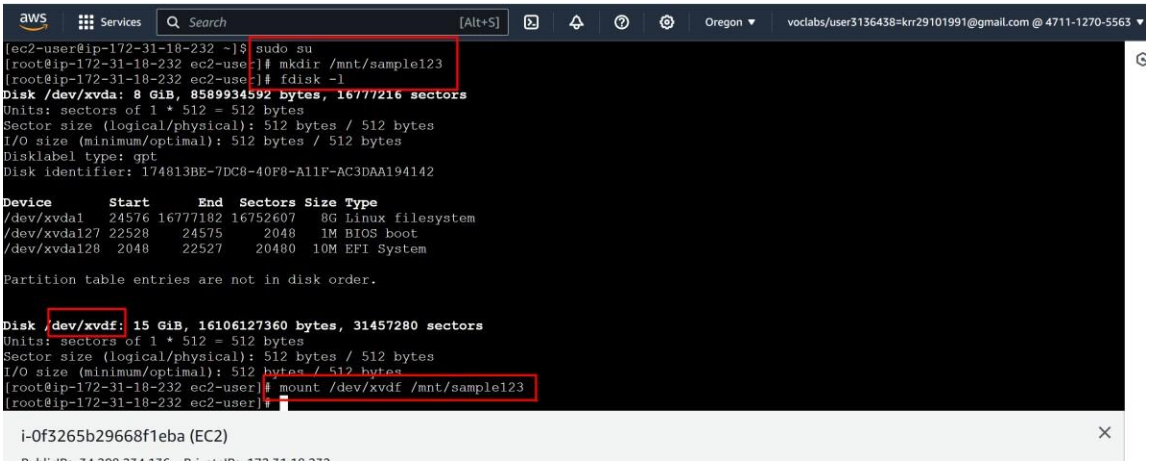


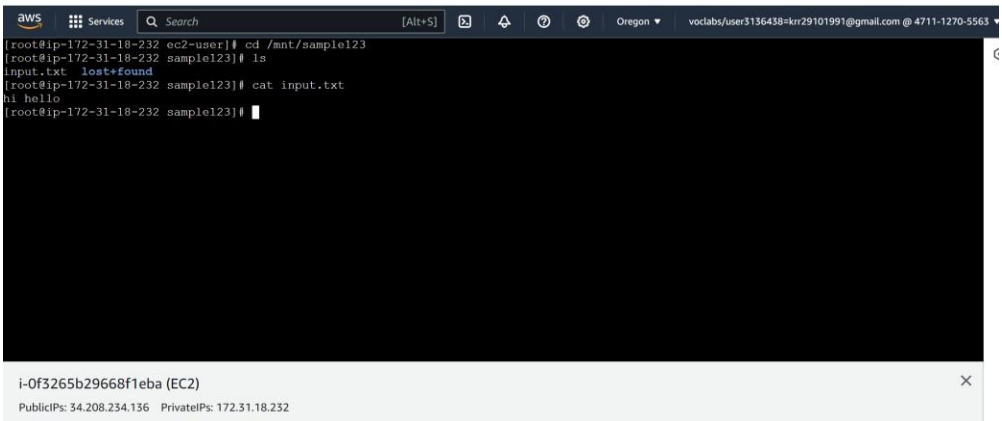





Services
Search
[Alt+S]
Oregon
voclabs/user3136438=krr29101991@gmail.com @ 4711-1270-5563
EC2 Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Instances (1/1) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
EC2
i-0f3265b29668f1eba
Running
t2.micro
View alarms
Instance: i-0f3265b29668f1eba (EC2)
Details
Status and alarms New
Monitoring
Security
Networking
Storage
Tags


Services
Search
[Alt+S]
Oregon
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Instance ID
i-0f3265b29668f1eba (EC2)
Connection Type
Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.
Public IP address
34.208.234.136
Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.
ec2-user
Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.
Cancel
Connect


Services
Search
[Alt+S]
Oregon
voclabs/user3136438=krr29101991@gmail.com @ 4711-1270-5563
[ec2-user@ip-172-31-18-232 ~]$ sudo su
[root@ip-172-31-18-232 ec2-user]# mkdir /mnt/sample123
[root@ip-172-31-18-232 ec2-user]# fdisk -l
Disk /dev/xvda: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 174813BE-7DC8-40F8-A11F-AC3DAA194142
Device Start End Sectors Size Type
/dev/xvda1 24576 16777182 16752607 8G Linux filesystem
/dev/xvda127 22528 24575 2048 1M BIOS boot
/dev/xvda128 2048 22527 20480 10M EFI System
Partition table entries are not in disk order.
Disk /dev/xvdf: 15 GiB, 16106127360 bytes, 31457280 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
[root@ip-172-31-18-232 ec2-user]# mount /dev/xvdf /mnt/sample123
[root@ip-172-31-18-232 ec2-user]#
i-0f3265b29668f1eba (EC2)


Services
Search
[Alt+S]
Oregon
voclabs/user3136438=krr29101991@gmail.com @ 4711-1270-5563
[root@ip-172-31-18-232 ec2-user]# cd /mnt/sample123
[root@ip-172-31-18-232 sample123]# ls
input.txt lost+found
[root@ip-172-31-18-232 sample123]# cat input.txt
hi hello
[root@ip-172-31-18-232 sample123]#
i-0f3265b29668f1eba (EC2)
PublicIPs: 34.208.234.136 PrivateIPs: 172.31.18.232

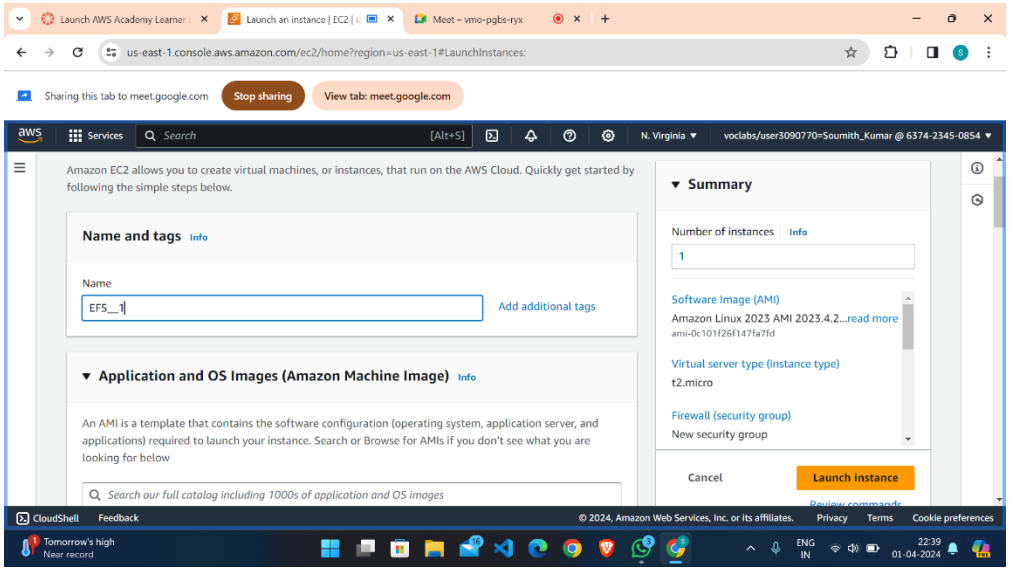
Experiment-4

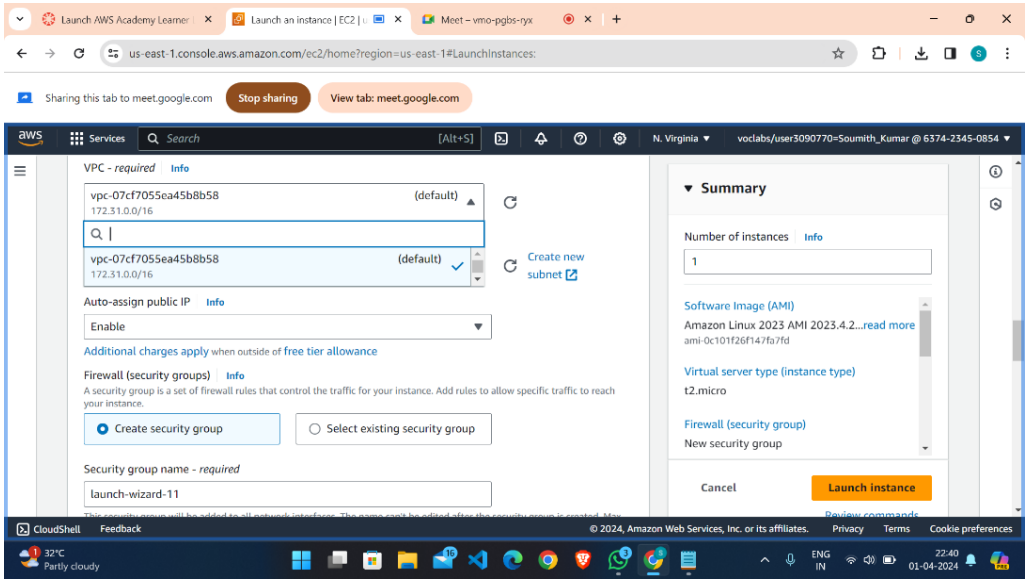
Aim: Provide a step-by-step demonstration of the process for setting up and configuring file sharing mechanisms using EFS.

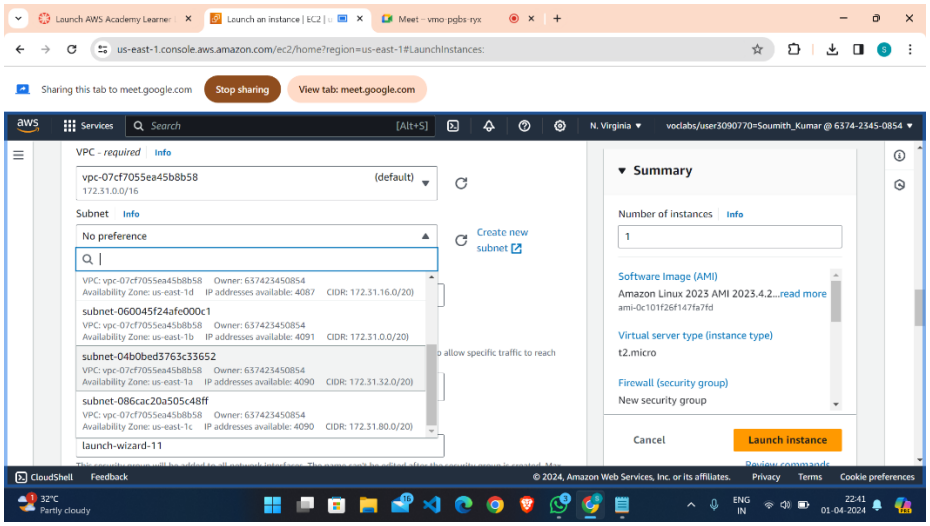
Step 1: Create an instance using following rules-

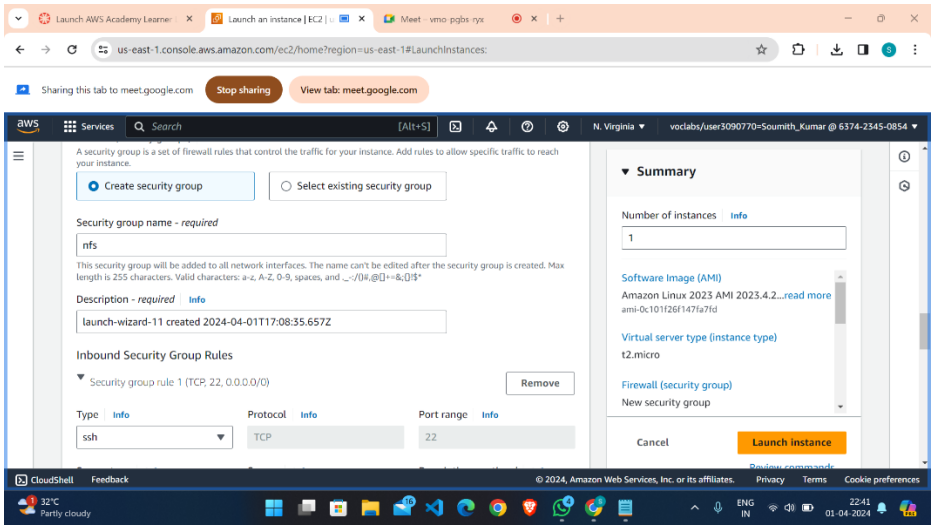
First Instance:

1. Create a Linux Instance.(EFS__1)
2. In the Network settings click on edit
3. Select the subnet as in which region we want to create instance.(us-east-1a)
4. Select the security group as Create New Security Group.
5. Give the name of the security Group.



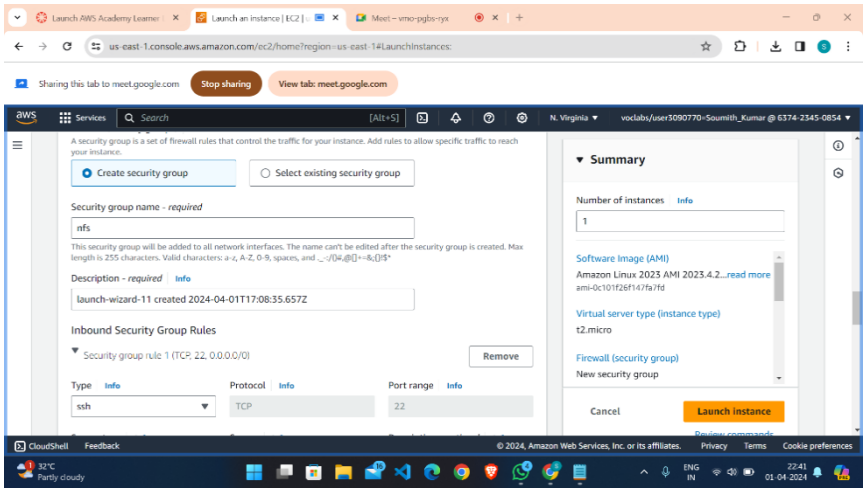


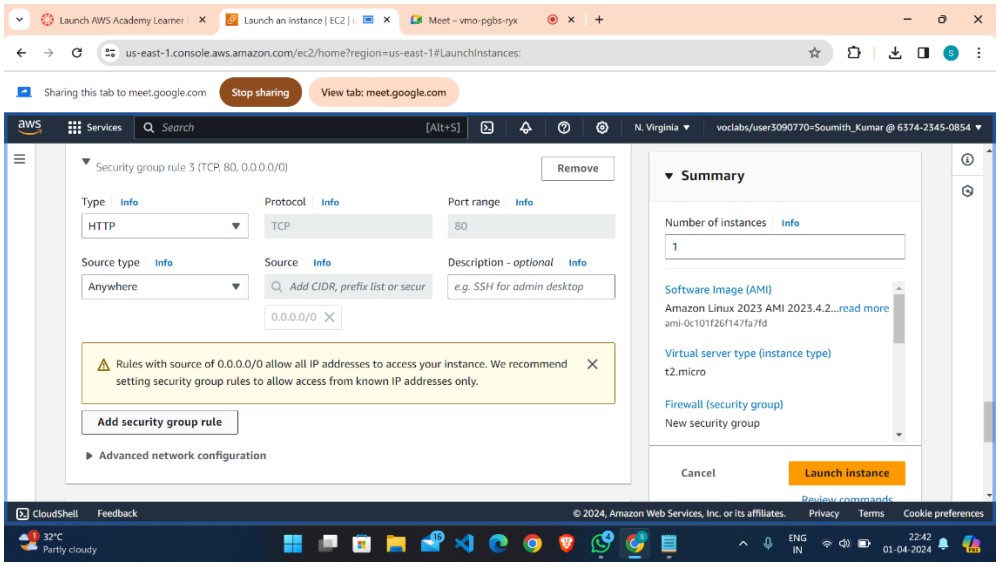


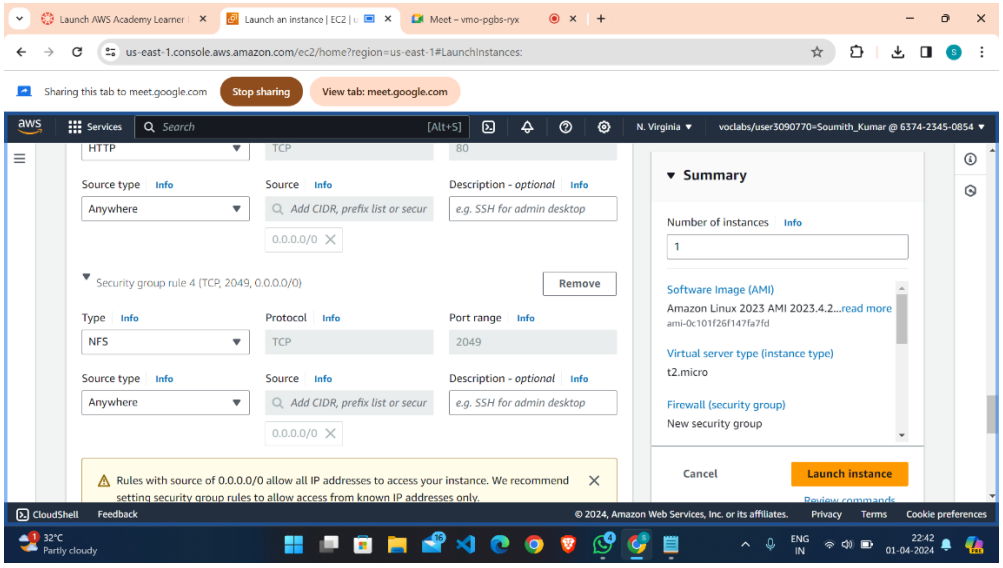


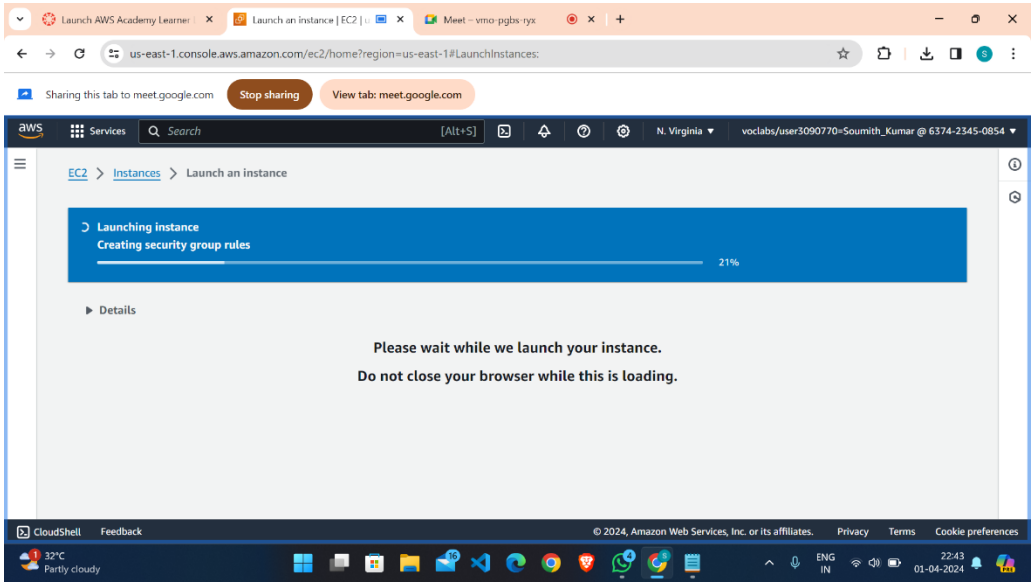
6. Add the security group rule by clicking on it.

7. Set the type as ‘NFS’ and source type as ‘Anywhere’ and click on Launch instance.



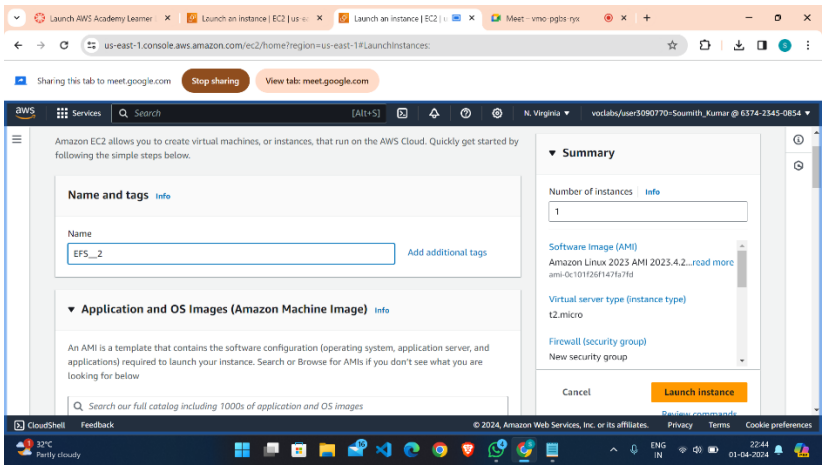


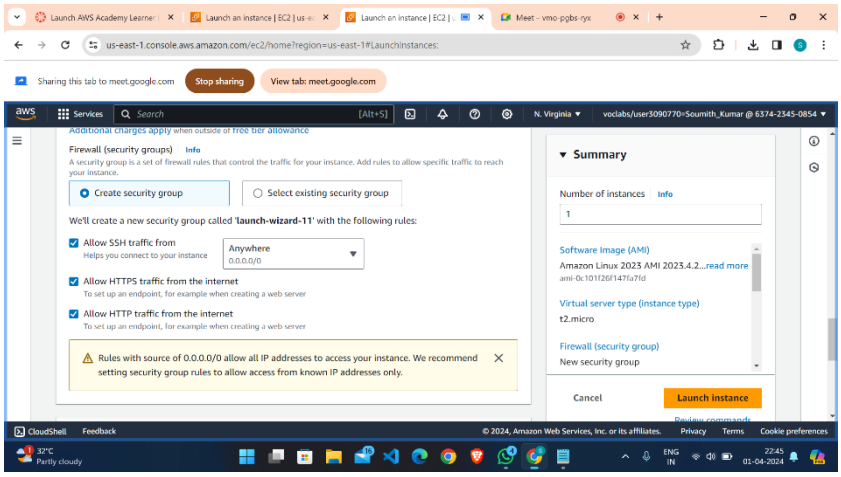


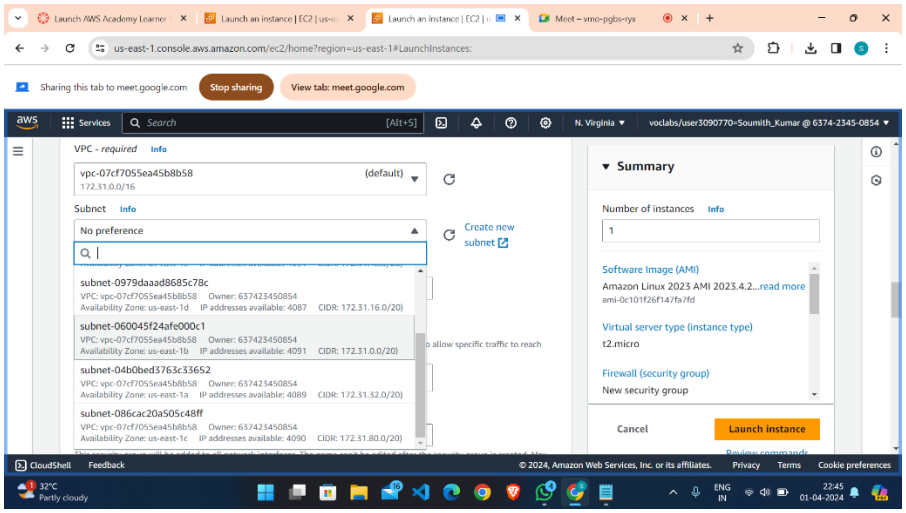


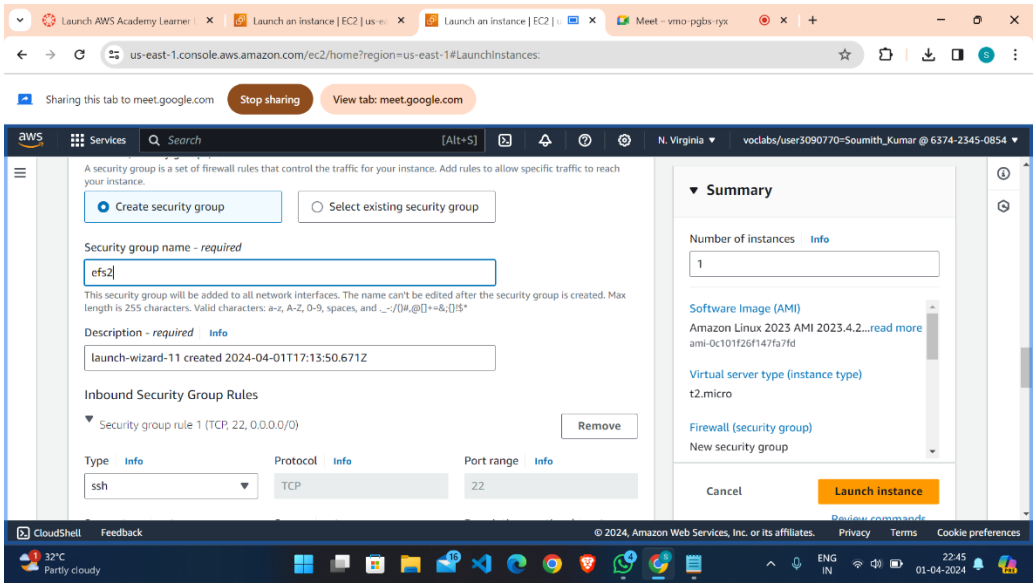
Second Instance:

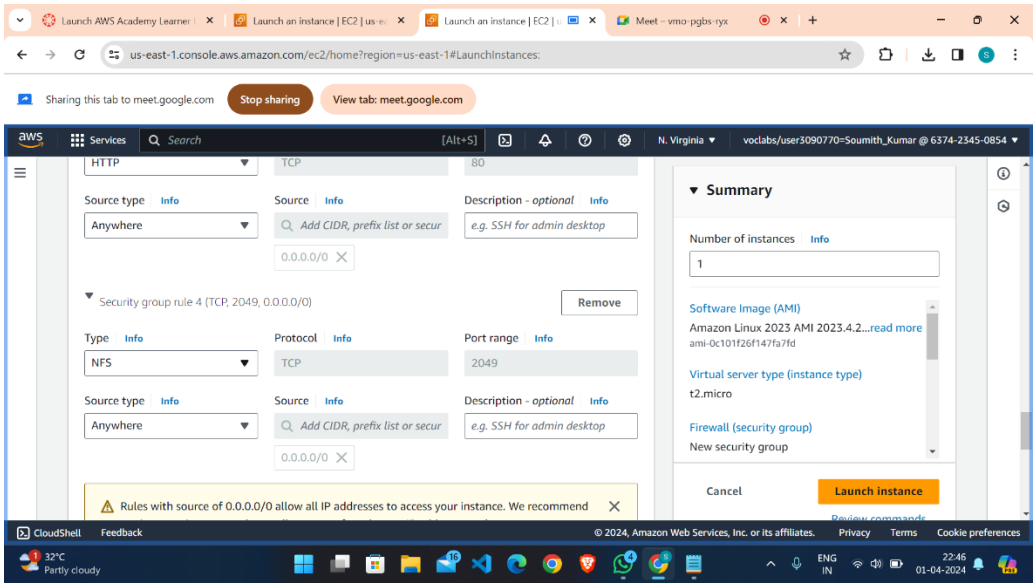
1. Create another instance with same configuration.(EFS_2)
2. In the Network settings click on edit
3. Select the subnet as in which region we want to create instance.(us-east-1b)
4. Select the security group as Select Existing Security Group.
5. Select the Security group which was created in the first instance from the dropdown.

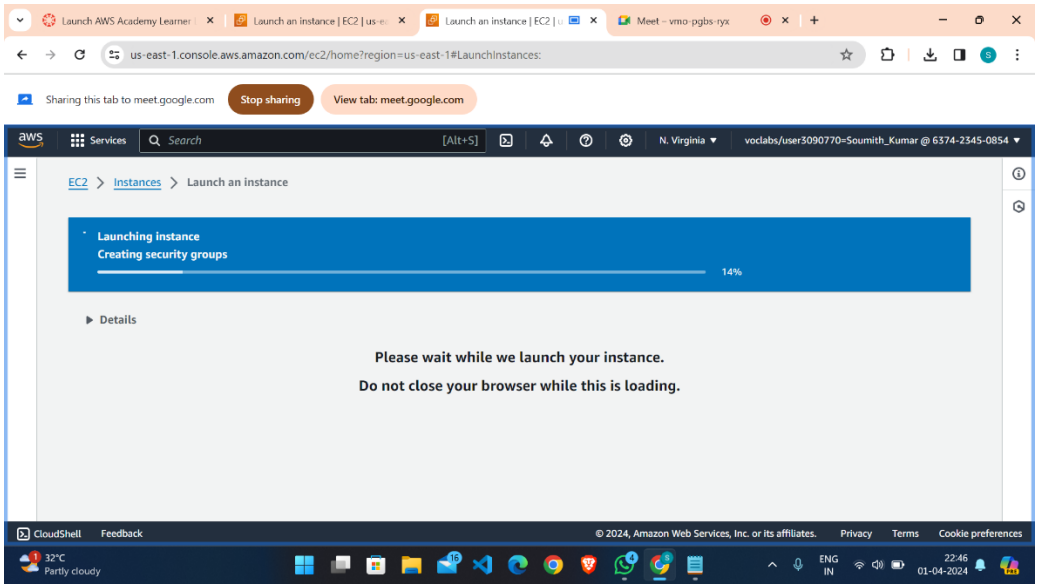




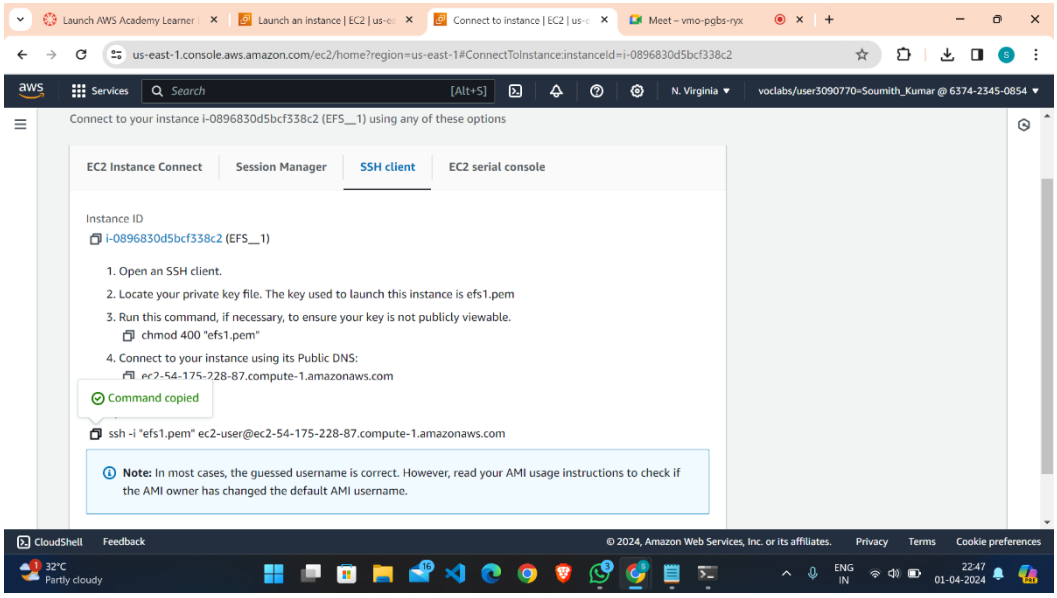



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
Sharing this tab to meet.google.com
Stop sharing
View tab: meet.google.com
Services
Search
[Alt+S]
N. Virginia
voclabs/user3090770=Soumith_Kumar @ 6374-2345-0854
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
Security group name - required
efs2
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
launch-wizard-11 created 2024-04-01T17:13:50.671Z
Inbound Security Group Rules
Security group rule 1 (TCP, 22, 0.0.0.0/0)
Remove
Type Info
ssh
Protocol Info
TCP
Port range Info
22
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.4.2...read more
ami-0c101f26f147fa7fd
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Cancel
Launch instance
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
32°C
Partly cloudy
ENG
IN
22:45
01-04-2024

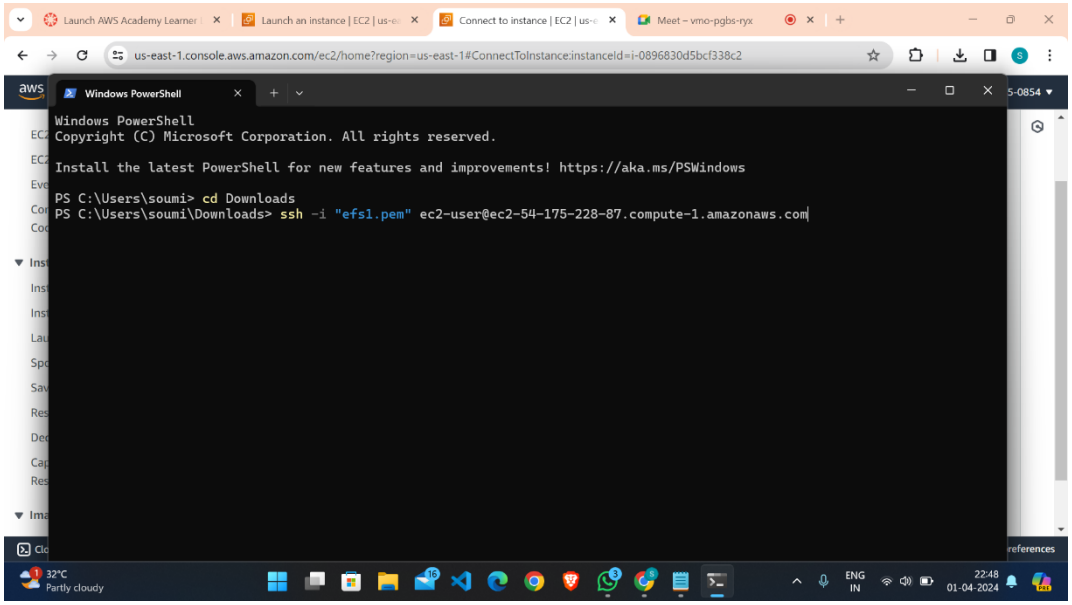

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
Sharing this tab to meet.google.com
Stop sharing
View tab: meet.google.com
Services
Search
[Alt+S]
N. Virginia
voclabs/user3090770=Soumith_Kumar @ 6374-2345-0854
HTTP
TCP
80
Source type Info
Anywhere
Source Info
Add CIDR, prefix list or secur
Description - optional Info
e.g. SSH for admin desktop
0.0.0.0/0
Security group rule 4 (TCP, 2049, 0.0.0.0/0)
Remove
Type Info
NFS
Protocol Info
TCP
Port range Info
2049
Source type Info
Anywhere
Source Info
Add CIDR, prefix list or secur
Description - optional Info
e.g. SSH for admin desktop
0.0.0.0/0
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.4.2...read more
ami-0c101f26f147fa7fd
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Cancel
Launch instance
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
32°C
Partly cloudy
ENG
IN
22:46
01-04-2024


us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
Sharing this tab to meet.google.com
Stop sharing
View tab: meet.google.com
Services
Search
[Alt+S]
N. Virginia
voclabs/user3090770=Soumith_Kumar @ 6374-2345-0854
EC2 > Instances > Launch an instance
Launching instance
Creating security groups
14%
Details
Please wait while we launch your instance.
Do not close your browser while this is loading.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
32°C
Partly cloudy
ENG
IN
22:46
01-04-2024

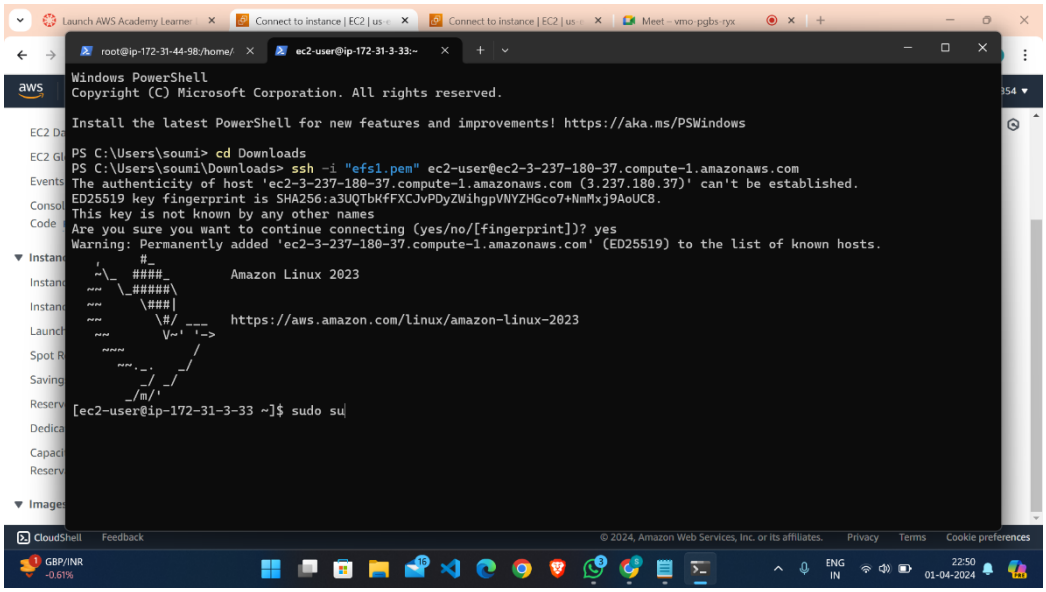
Step 2: Click on instance 1 and click on connect and choose SSH client , copy the command



- Go to Windows PowerShell and then paste the ssh command

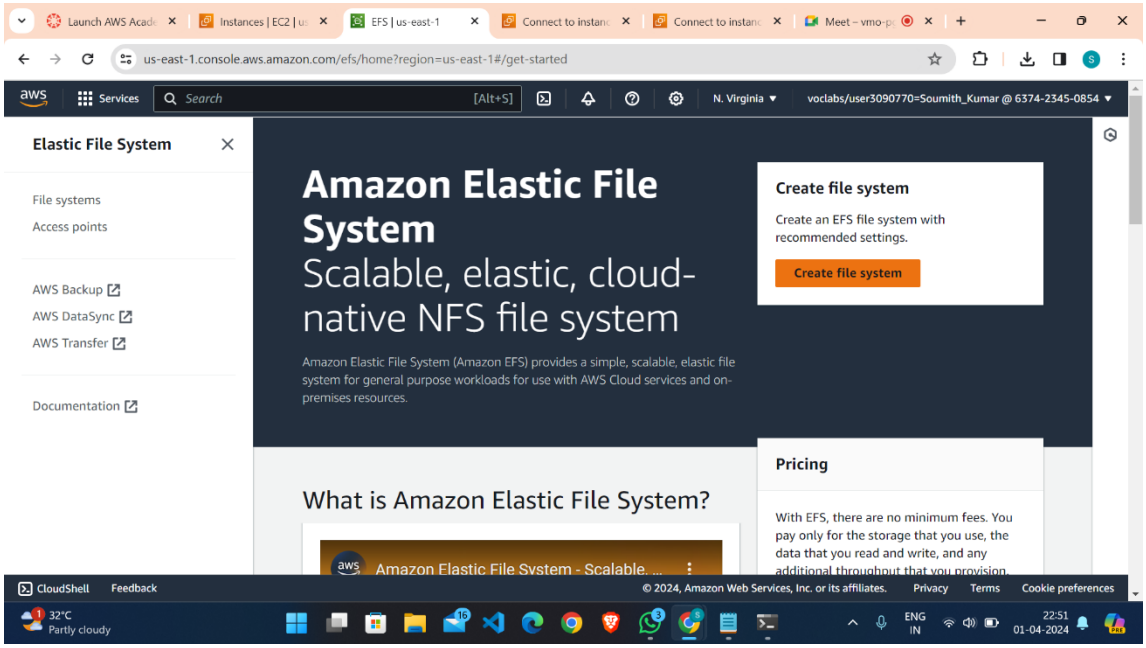


Step 3: Do to the same for instance 2

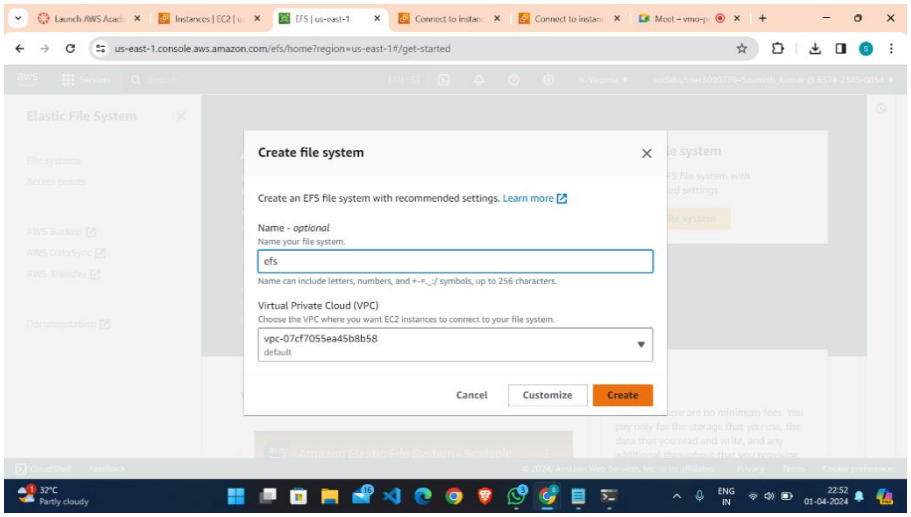


- Enter sudo su commands in both terminals

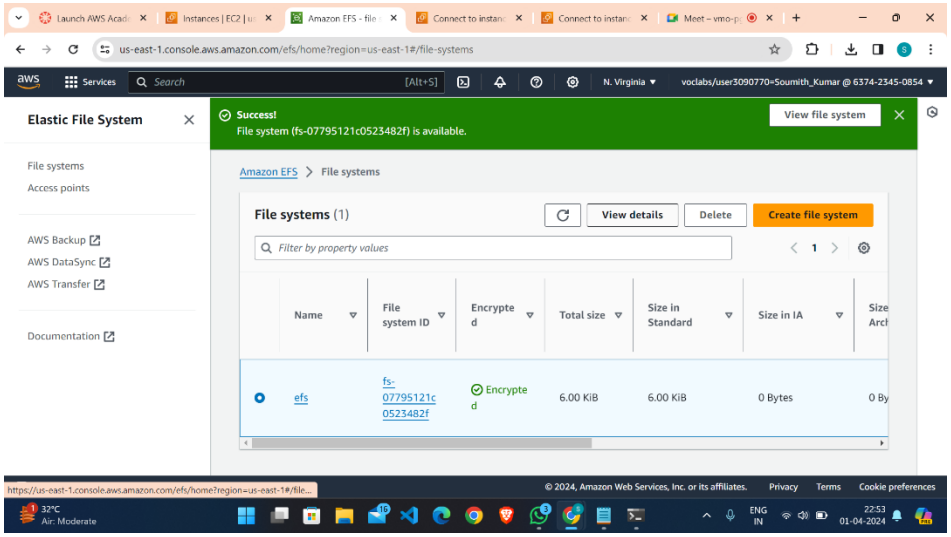
Step 4: Go to AWS and search for EFS in the Services



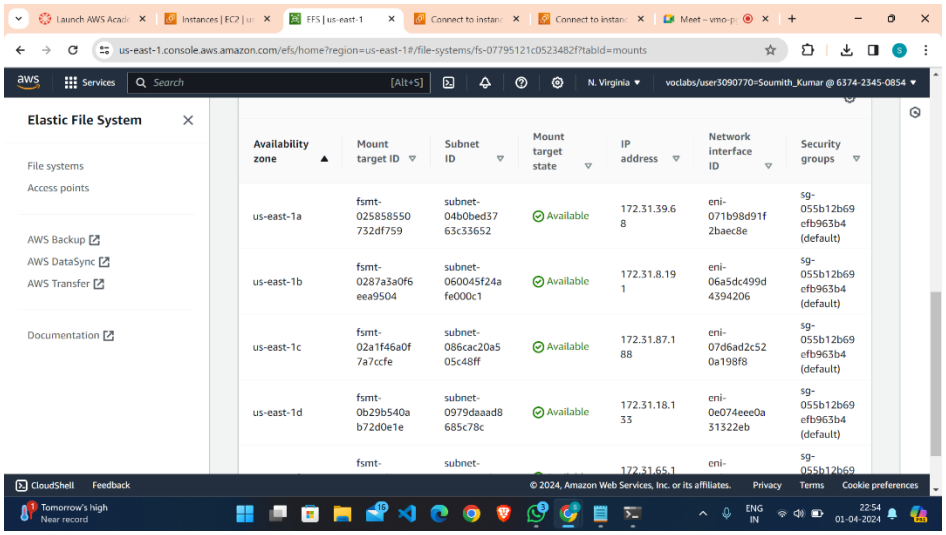
1. Click on Create File System and Enter the name of the file system and VPC as default and click on create.



2. Select the efs and click on the hyperlink at the name of the instance

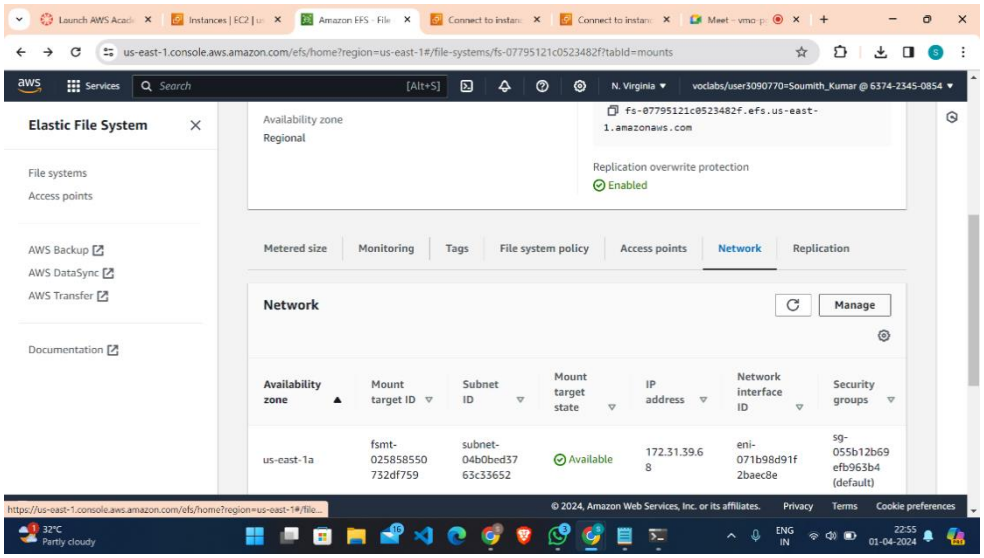


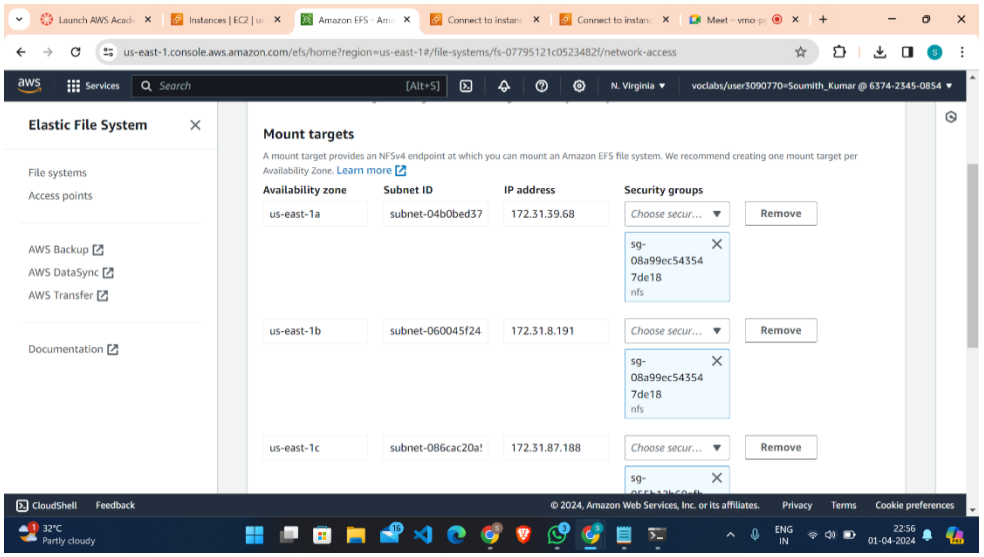
3. Go to Network and wait until all Mount Target State will come to ‘Available’

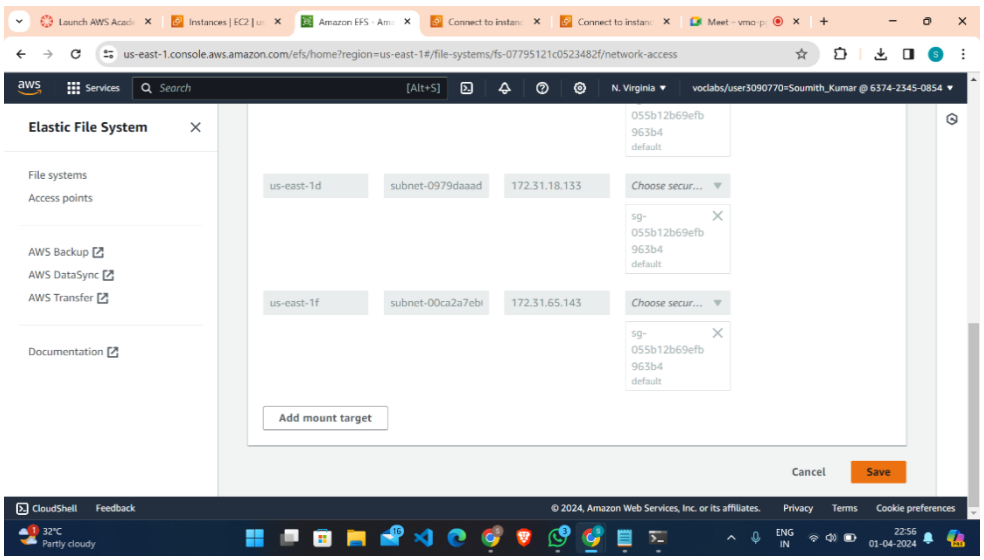


I. Click on Manage and change the mount target Security groups to our security group which is created while creating the instance.

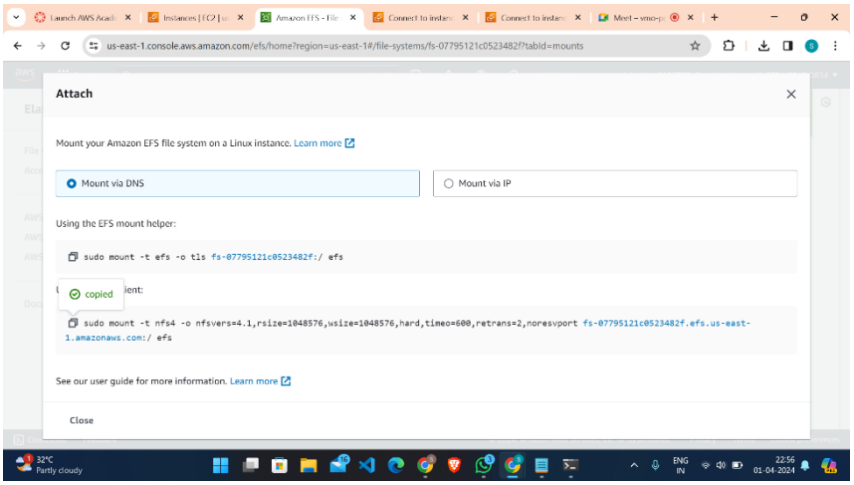
II. Remove the default and select ‘EFS’ for “us-east-1a” and “us-east-1b”





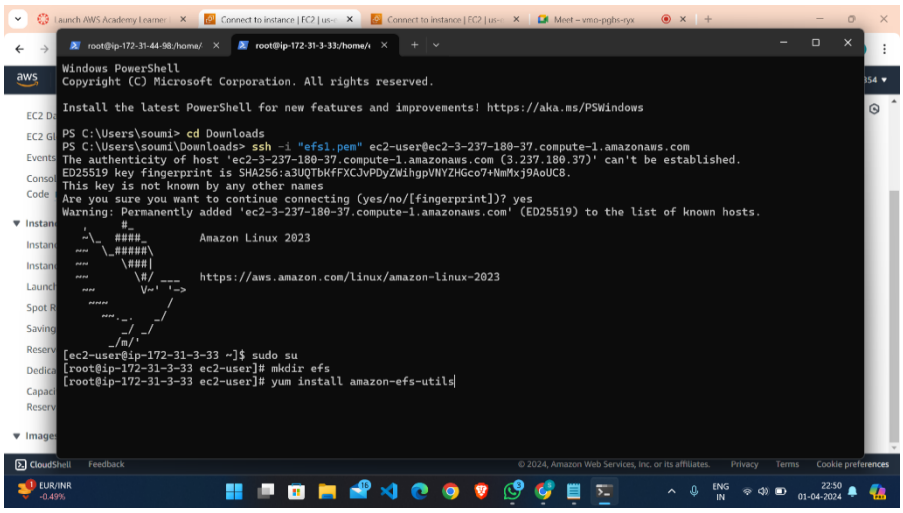


- Click on save.
- Now click on attach
- Copy the NFS client address

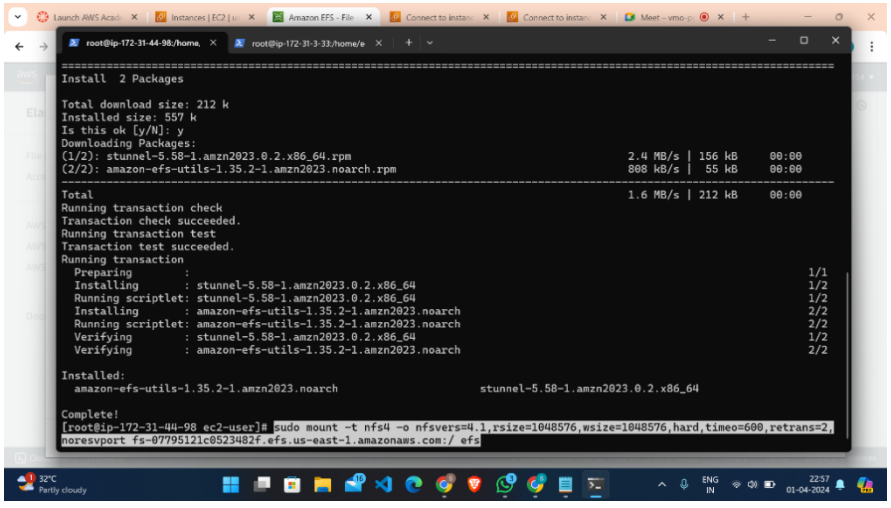


Step 5: Now go to PowerShell and type the commands in both the terminals.

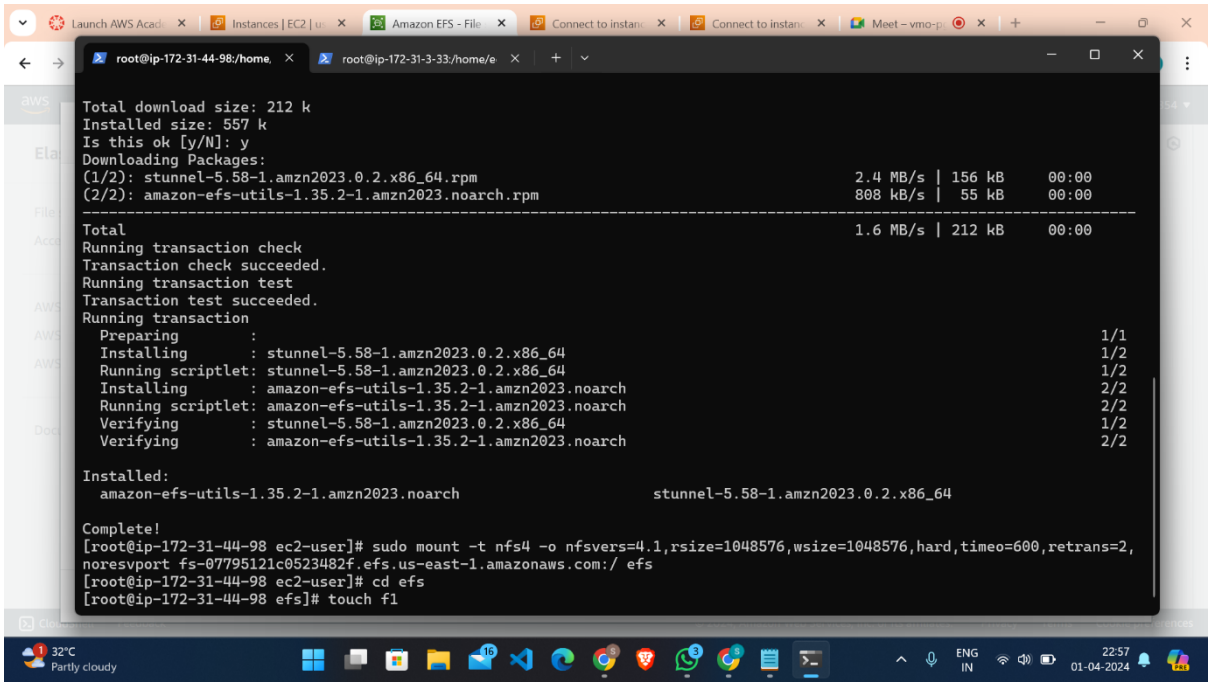
yum install amazon-efs-utils –y



- paste the copied nfs client address and hit enter in both the terminals



1. Create a folder using ‘mkdir efs’ in both terminals.
2. Now go to folder using ‘cd efs’
3. Create a file in any of the one terminal using ‘sudo touch FileName’



4. Now Enter ls in both the terminal and we can see f1 file in both the terminals

```
[root@ip-172-31-44-98 ec2-user]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,
noresvport fs-07795121c0523482f.efs.us-east-1.amazonaws.com:/ efs
[root@ip-172-31-44-98 ec2-user]# cd efs
[root@ip-172-31-44-98 efs]# touch f1
[root@ip-172-31-44-98 efs]# ls
f1
[root@ip-172-31-44-98 efs]#
```

```
[root@ip-172-31-3-33 ec2-user]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,n
oresvport fs-07795121c0523482f.efs.us-east-1.amazonaws.com:/ efs
[root@ip-172-31-3-33 ec2-user]# cd efs
[root@ip-172-31-3-33 efs]# ls
f1
```

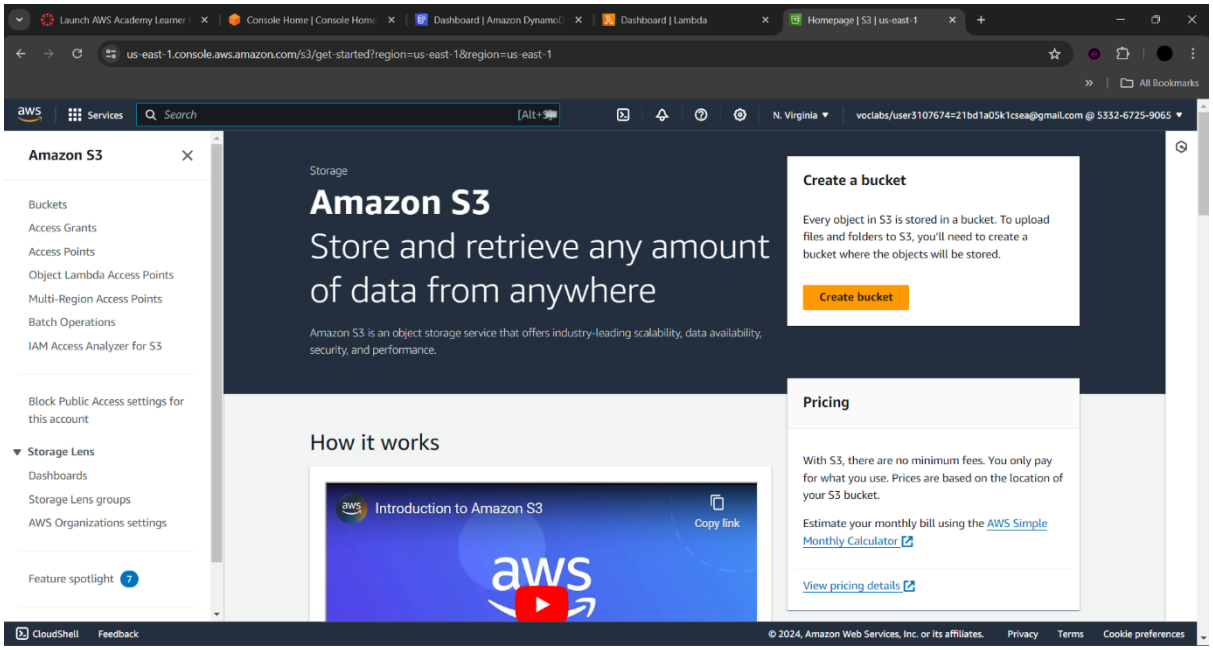
Experiment-5

Aim: Demonstrate the execution of a simple Python program using AWS Lambda functions. Include step-by-step instructions for creating and configuring the Lambda function, list out the languages supported by AWS Lambda.

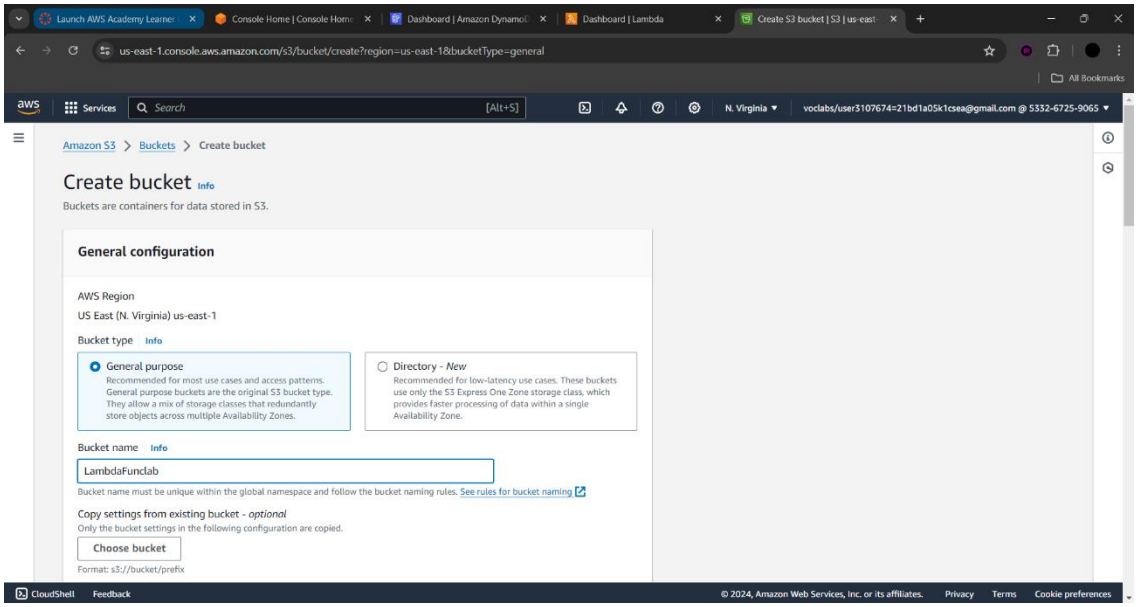
Step 1 : We need to open 3 Services parallelly in 3 different tabs

- S3 Bucket
- Dynamo DB
- Lambda Function

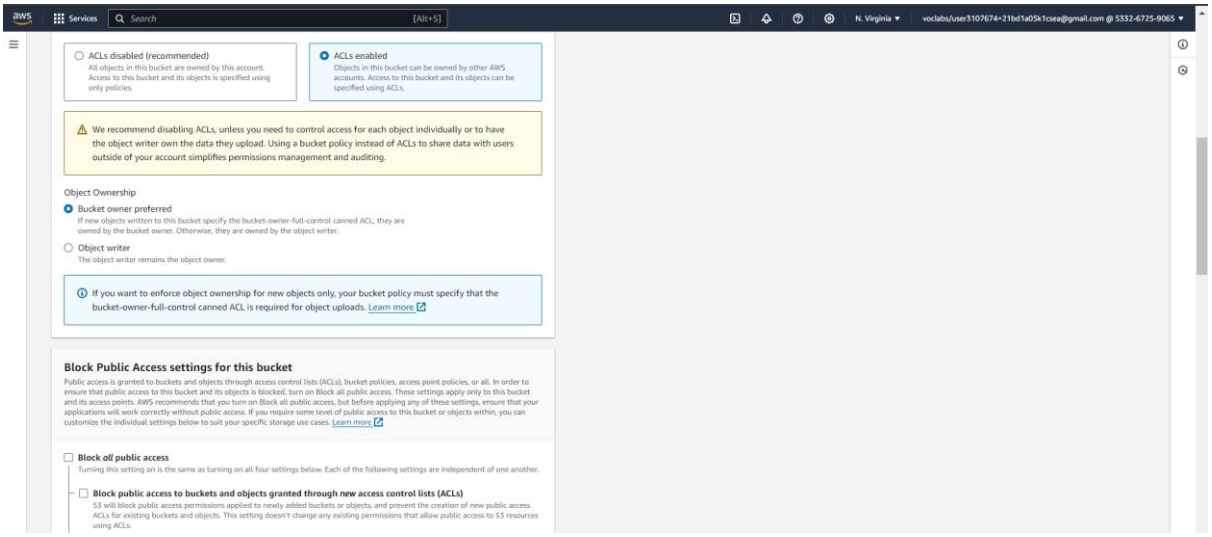
Step 2: Navigate to S3 via AWS Management Console.



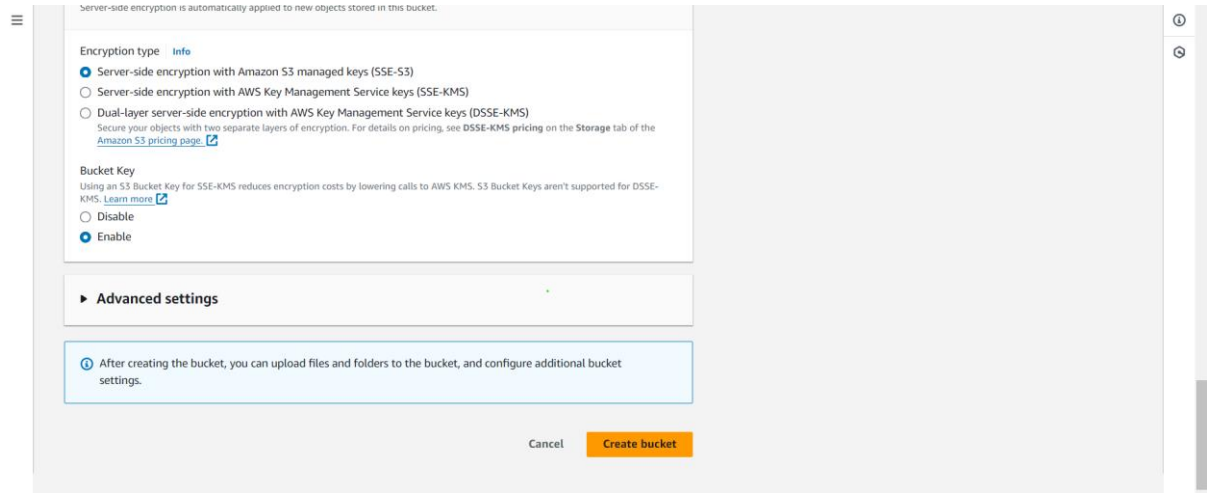
Step 3:Click on Create Bucket , Choose General purpose radio option and proceed to provide a unique name to the bucket.



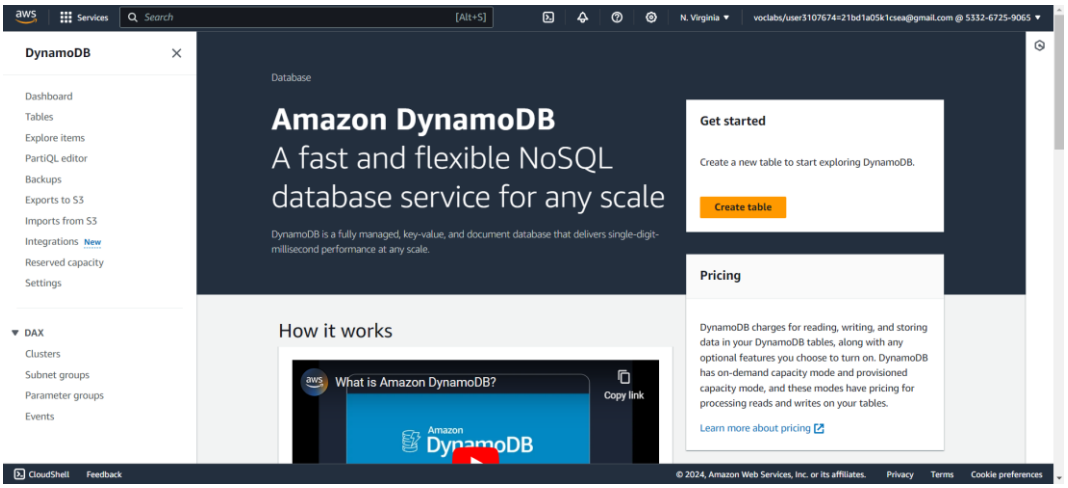
Step 4:Enable ACLs and uncheck the Block public access checkbox and proceed to accept the conditions by checking the “I acknowledge” checkbox.



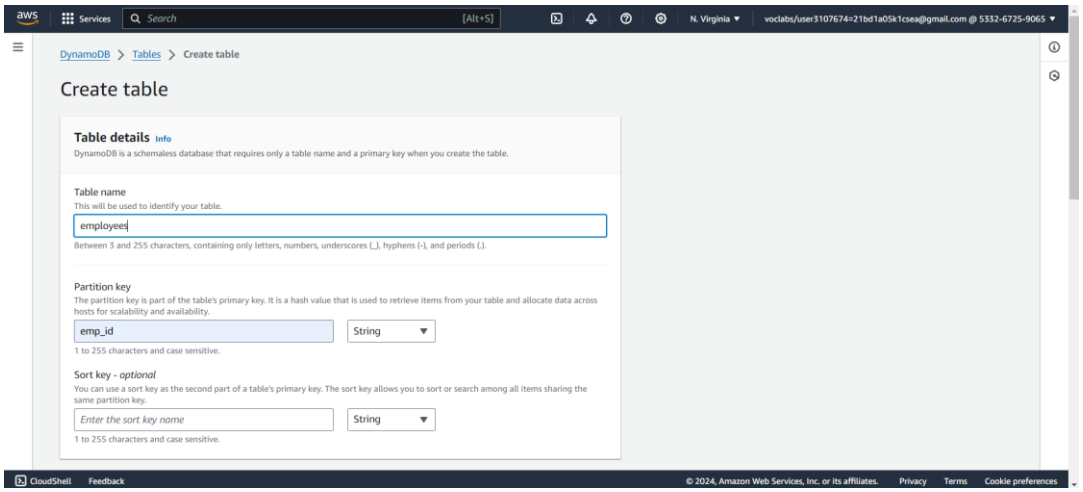
Step 5:Proceed to create the Bucket.



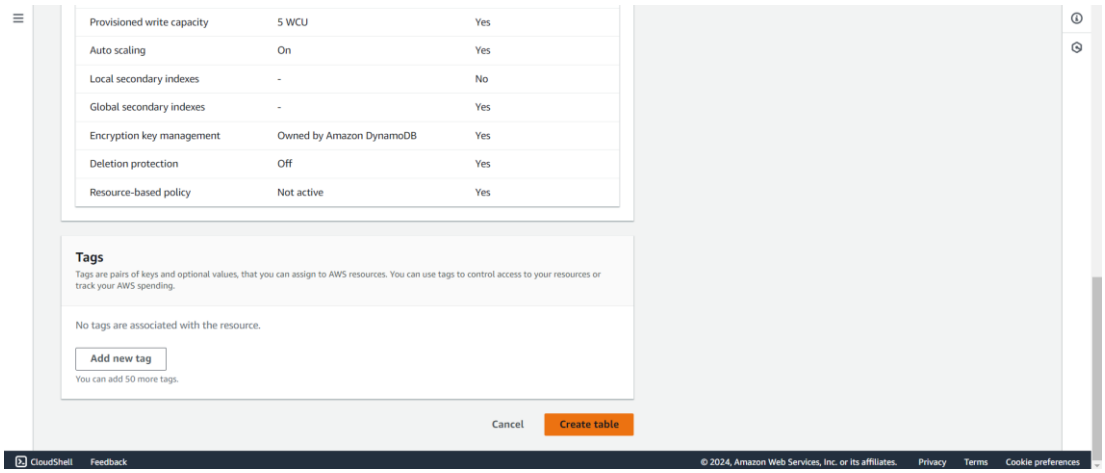
Step 6:Now Navigate to Dynamo DB via Management Console to configure it upon trigger by Lambda function , Proceed to click on create Table.



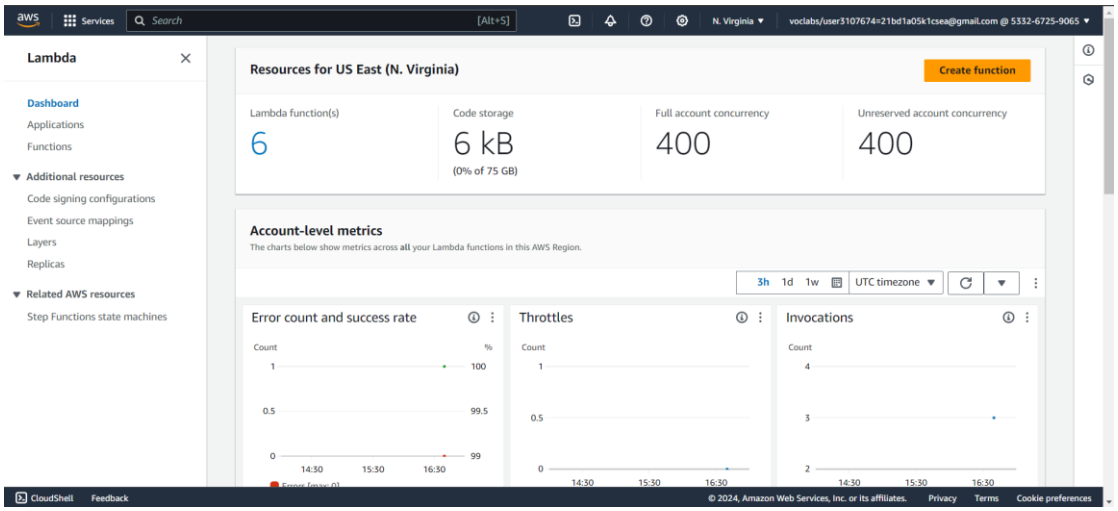
Step 7: Enter the name of the table and Enter the partition key . Make sure that the table name is same as the name mentioned in the python script and the partition key is same as mentioned in the json file that is to be uploaded into the S3 bucket .



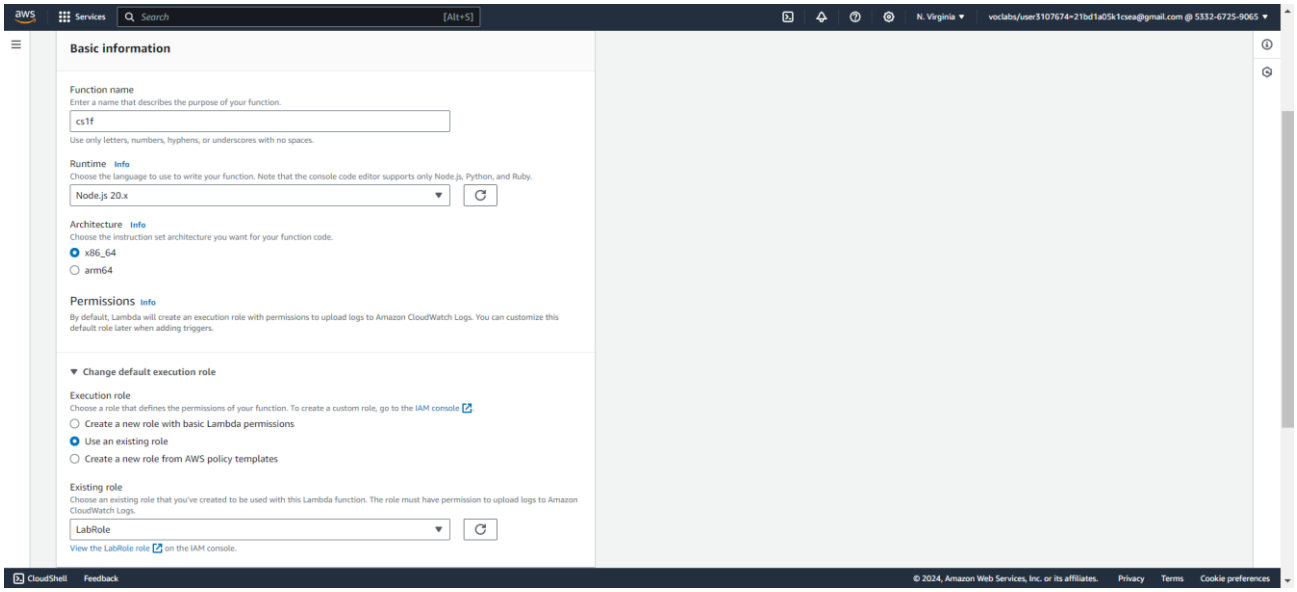
Step 8:Proceed to Click on Create Table .



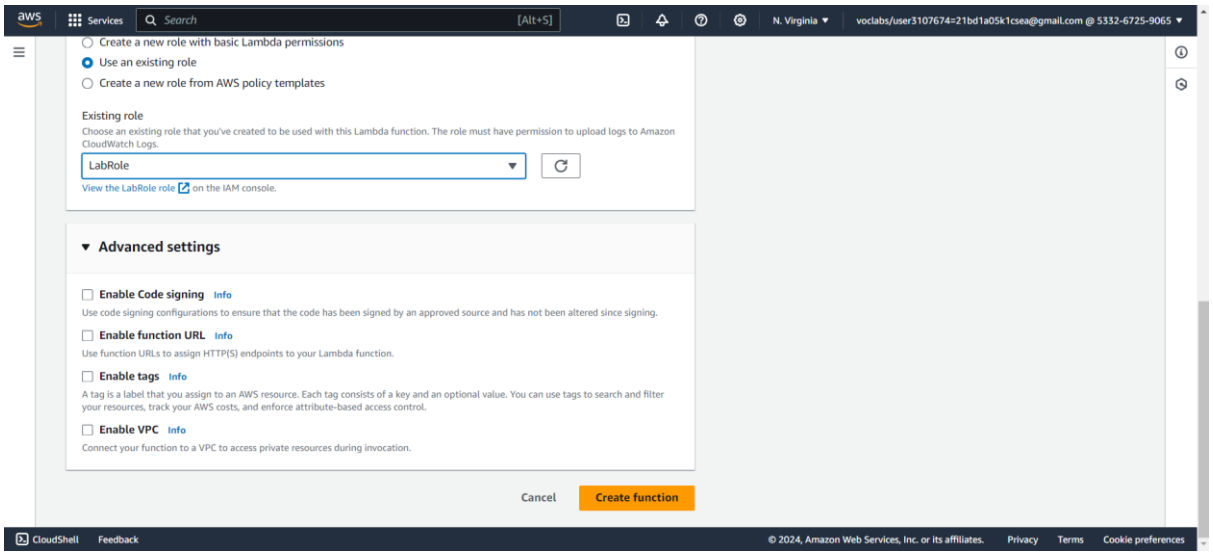
Step 9:Now, Navigate to Lambda function via AWS Management Console and click on Create Function.



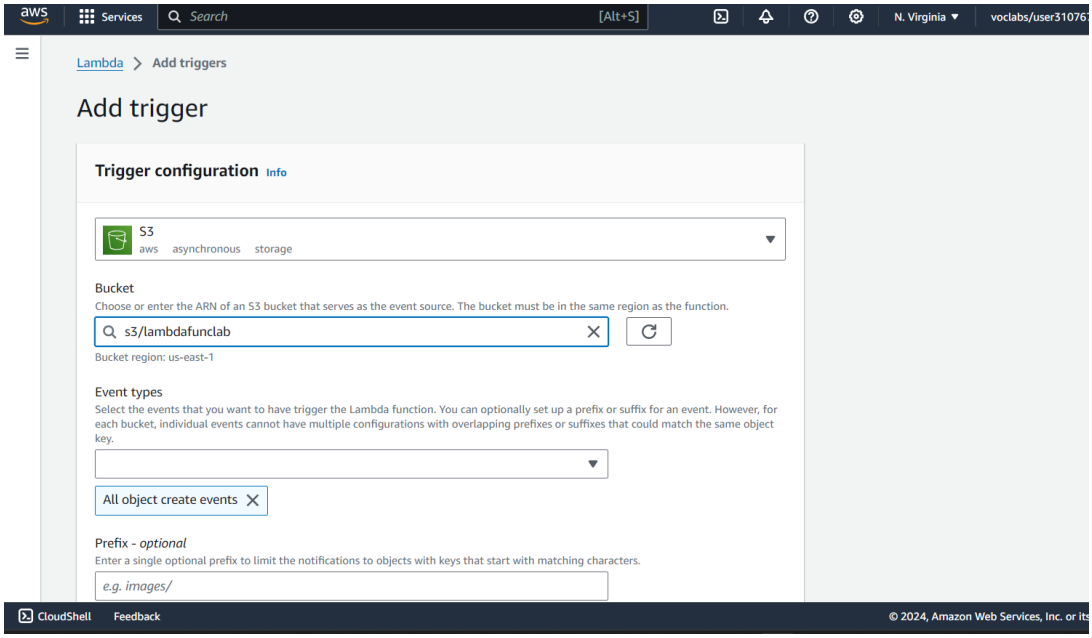
Step 10:Enter a name and click on Change default execution role , choose Use an existing role and proceed to choose LabRole .



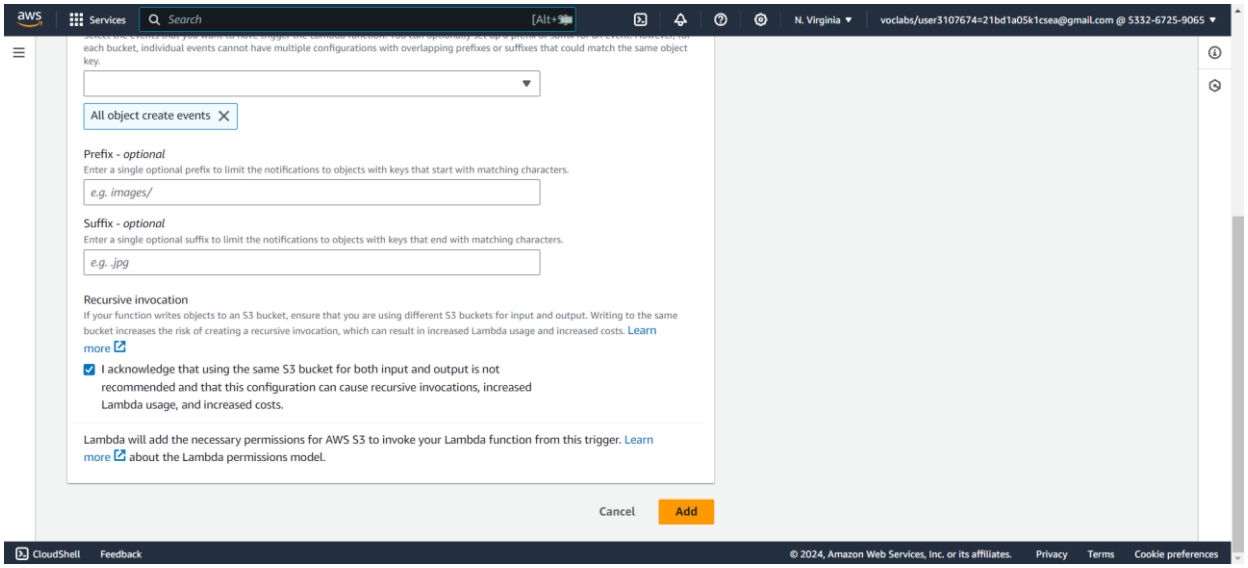
Step 11: Click on Create function.



Step 12: Once the Function is created it will give us a window to create a Source to create a Trigger , so Click on Add Trigger , Choose S3 as AWS service and choose the bucket we have created .



Step 13: Click on Add to add Source Trigger .

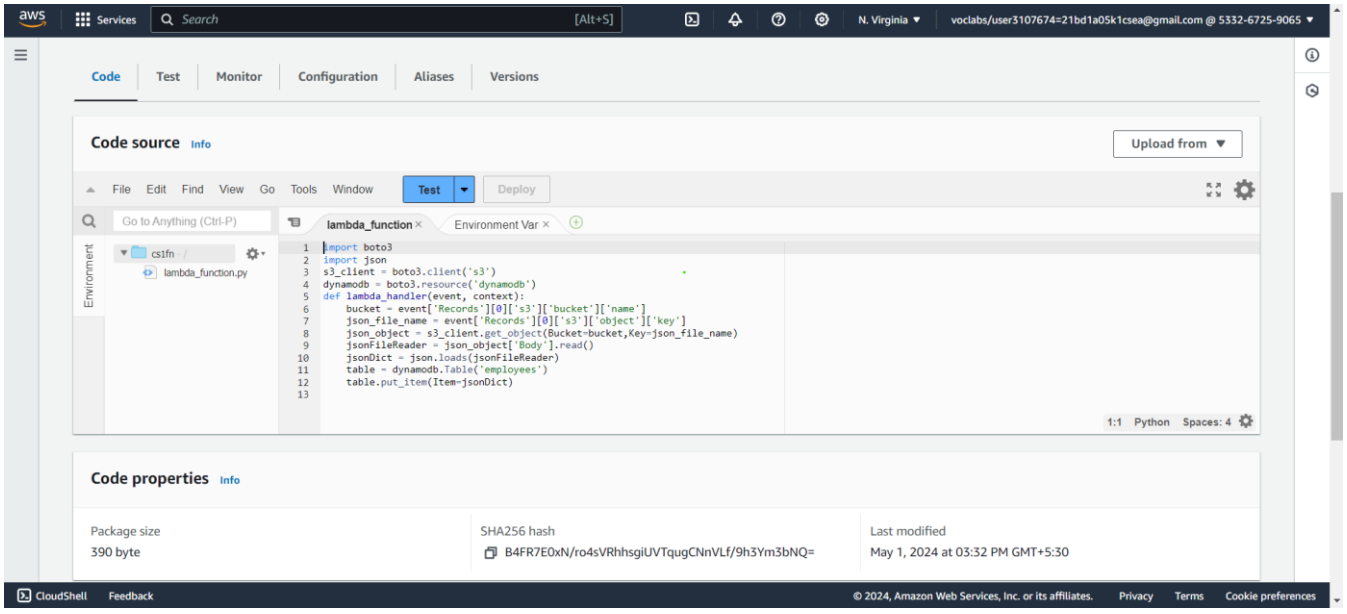


Step 14: Once Created procced to Code option in the same window and paste to following code

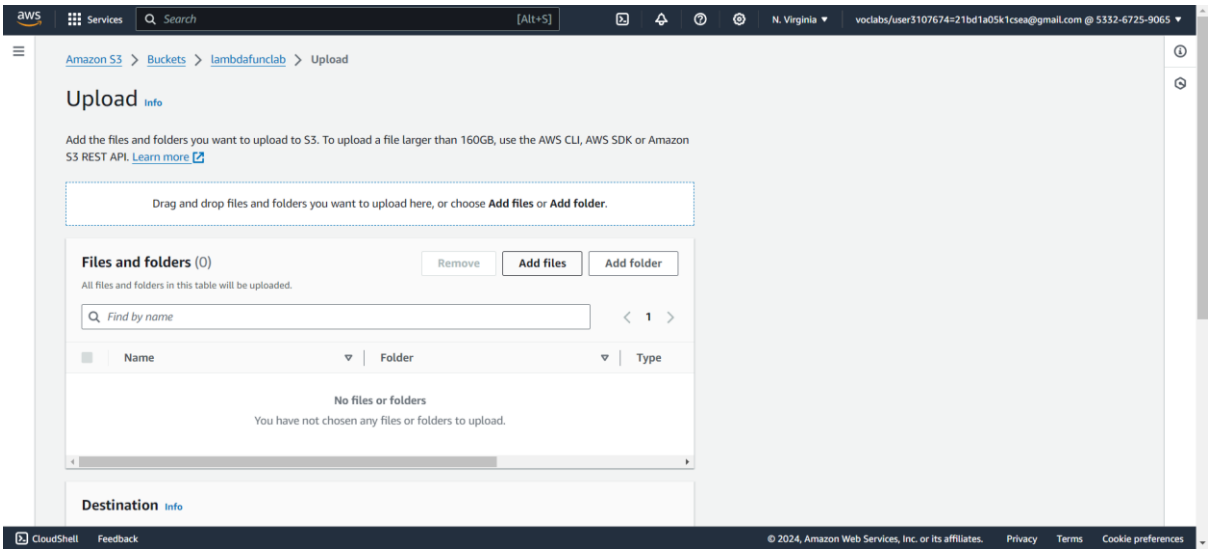
Code:

```
import boto3

import json

s3_client = boto3.client('s3')

dynamodb = boto3.resource('dynamodb')

def lambda_handler(event, context):

    bucket = event['Records'][0]['s3']['bucket']['name']

    json_file_name = event['Records'][0]['s3']['object']['key']

    json_object = s3_client.get_object(Bucket=bucket,Key=json_file_name)

    jsonFileReader = json_object['Body'].read()

    jsonDict = json.loads(jsonFileReader)

    table = dynamodb.Table('employees')

    table.put_item(Item=jsonDict)
```

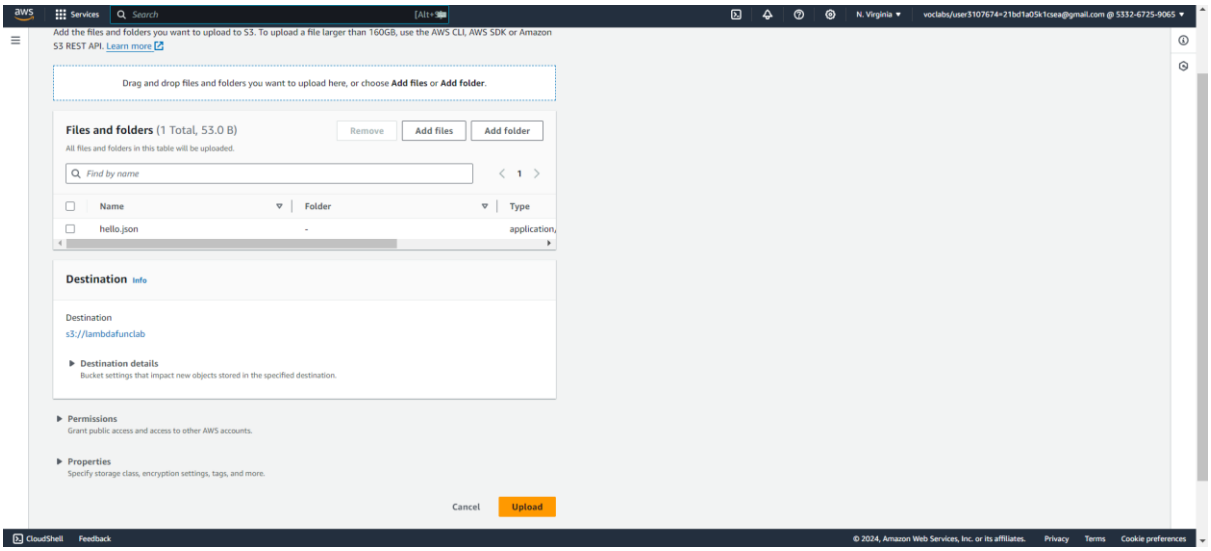
→Further proceed and click on Test and give it a name , An error less code will procced to a successful compilation of the code and then proceed to click on Deploy, If not deployed the trigger will not be effective .



Step 15: Now Go back to S3 Bucket we have created and upload the Json file into it and Click on Upload.

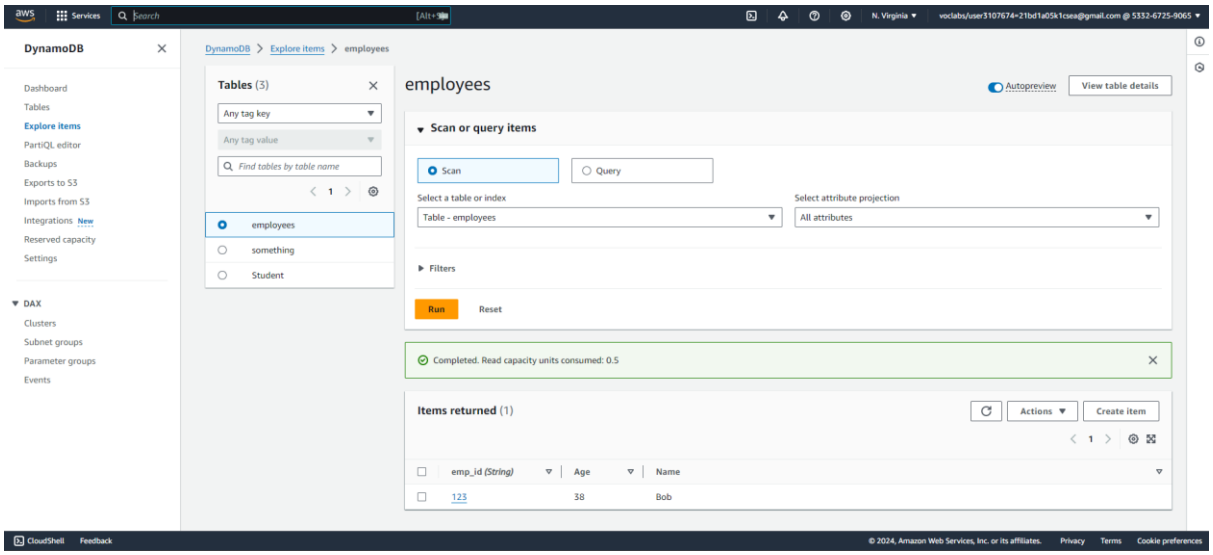


Step 16: Click on Add Files to upload the hello.json file and Click on Upload.



Once we click upload the lambda function is triggered and the result is directed to the Dynamo DB where we have our employees table with emp_id as the partition key.

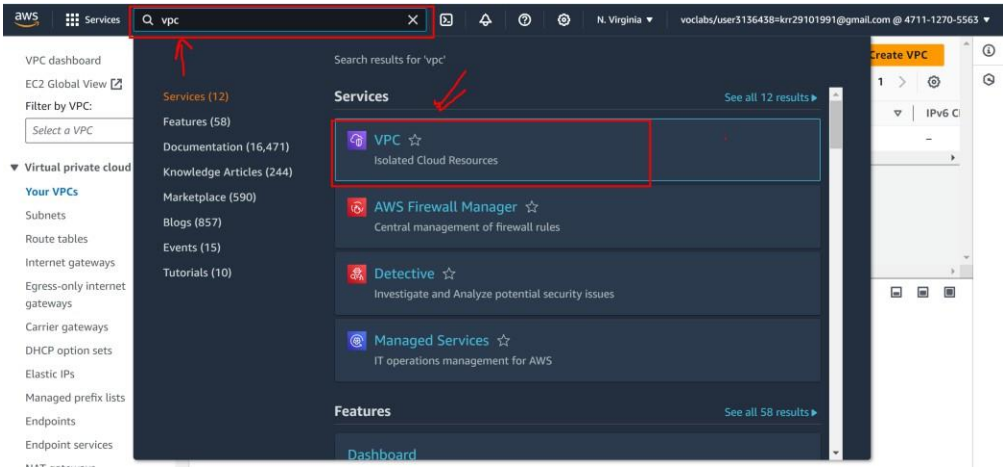
Step 17:Navigate to Dynamo DB and refresh the page to observe items of the table, Click on Explore items , we can see that the contents uploaded in the form of json in the S3 bucket have been updated in Dynamo DB employees table .



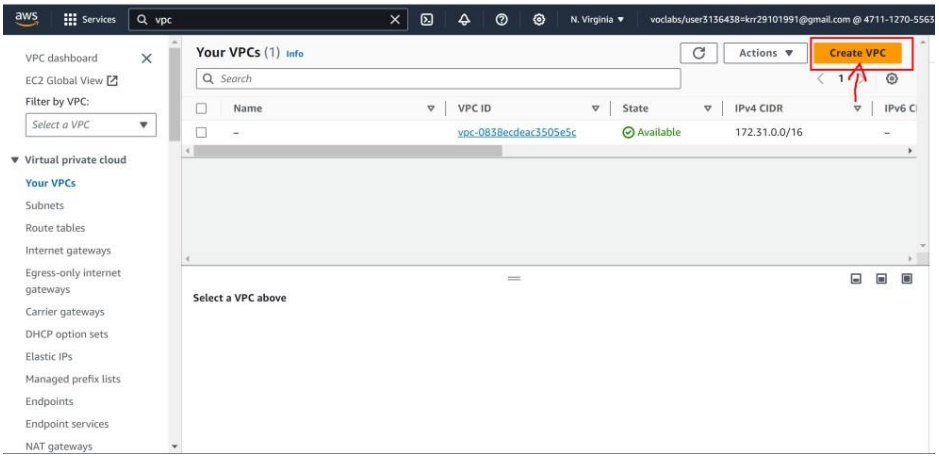
Experiment-6

Aim: Demonstrate the use of VPC using a public and private subnet. Establish the connection with public subnet using Internet Gateway and Route table. As well as launch EC2 instances in each subnet and establish the connection in between two EC2 instances.

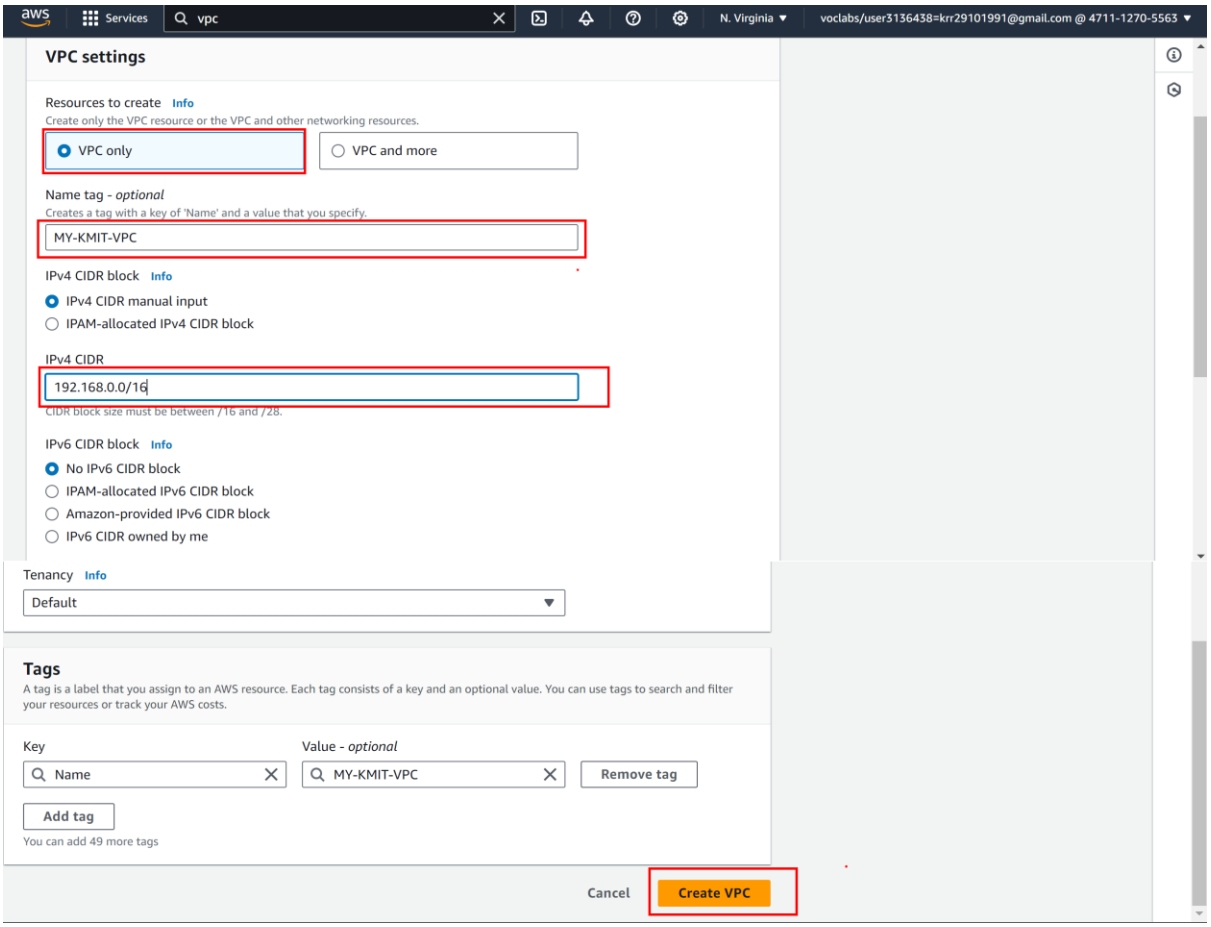
Step 1:Firstly Search for **VPC and click on VPC**



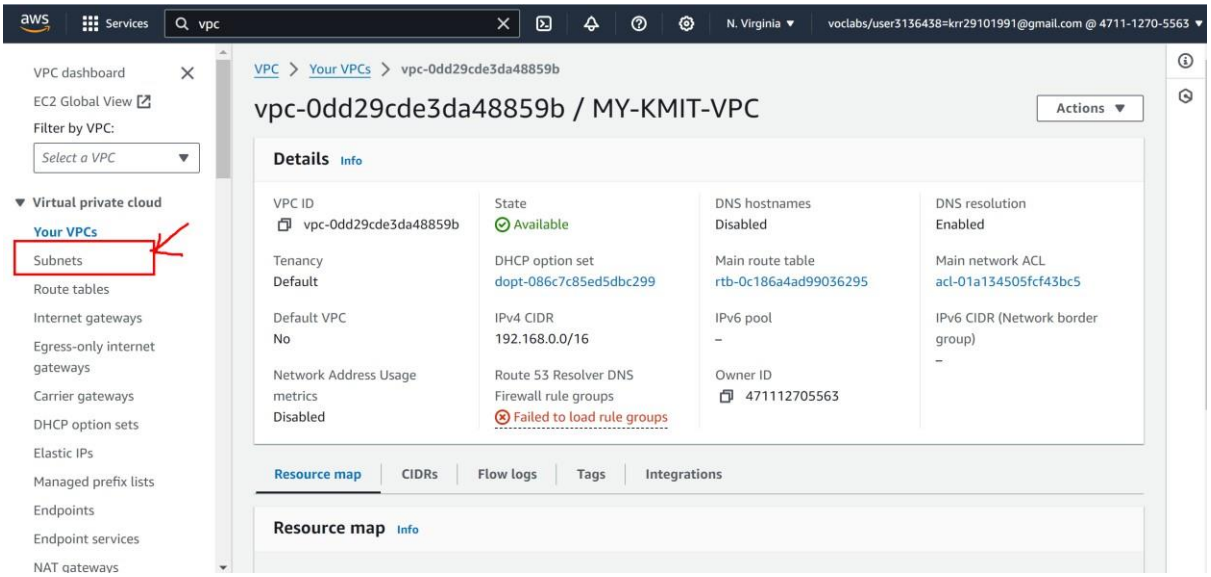
Step 2 :Click on **Create VPC**



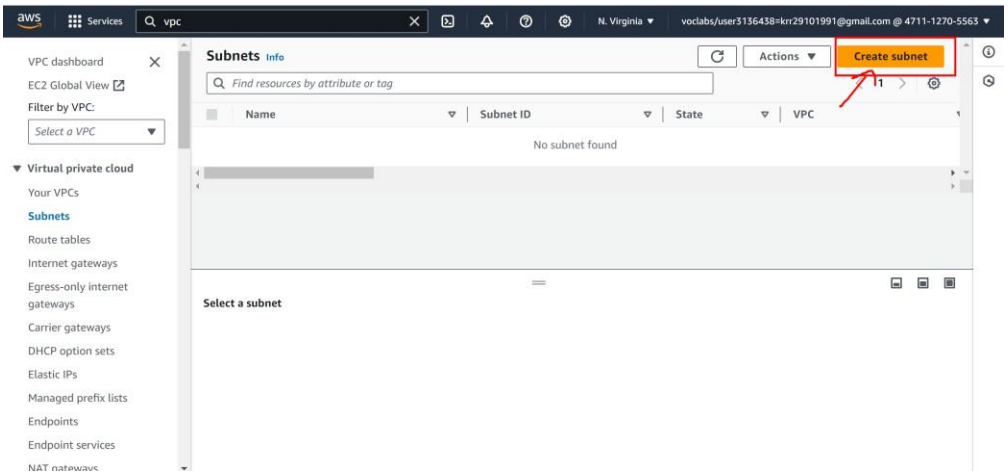
Step 3: Name our **VPC** and Insert **CIDR** and Click on **Create VPC**



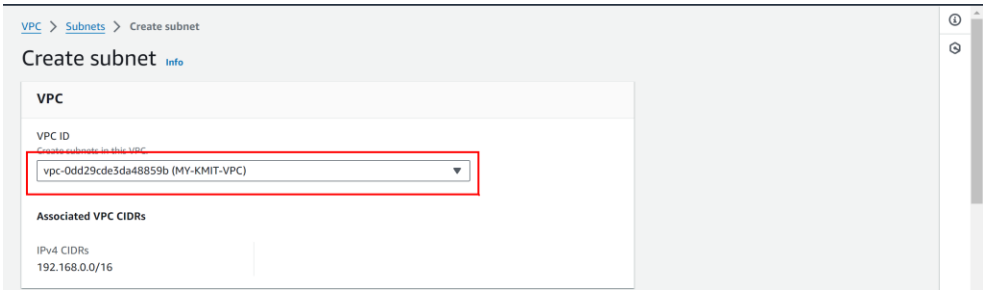
Step 4: Now click on subnets to create two subnets



Step 5: Click on Create Subnet

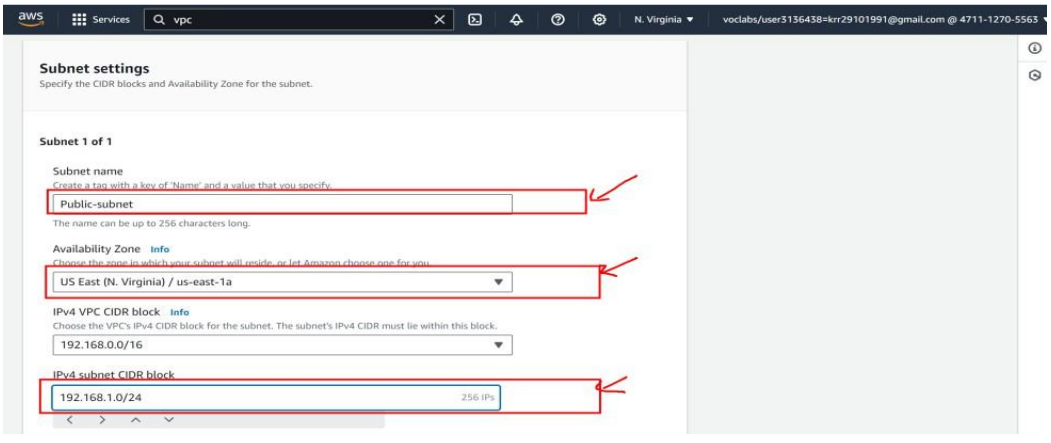


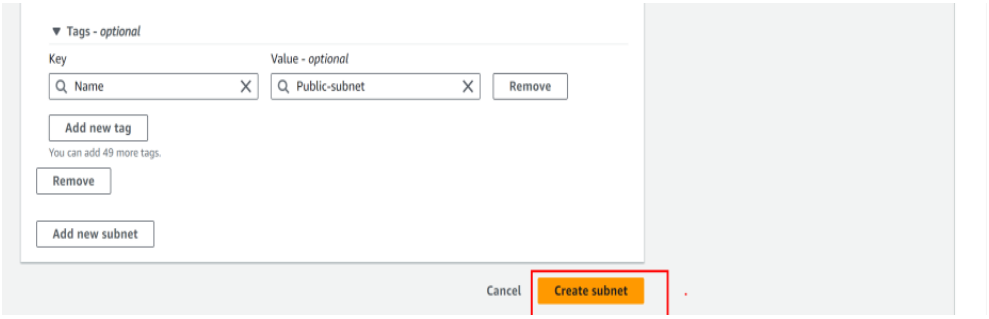
Step 6: Select the VPC which you have created in previous step from drop down



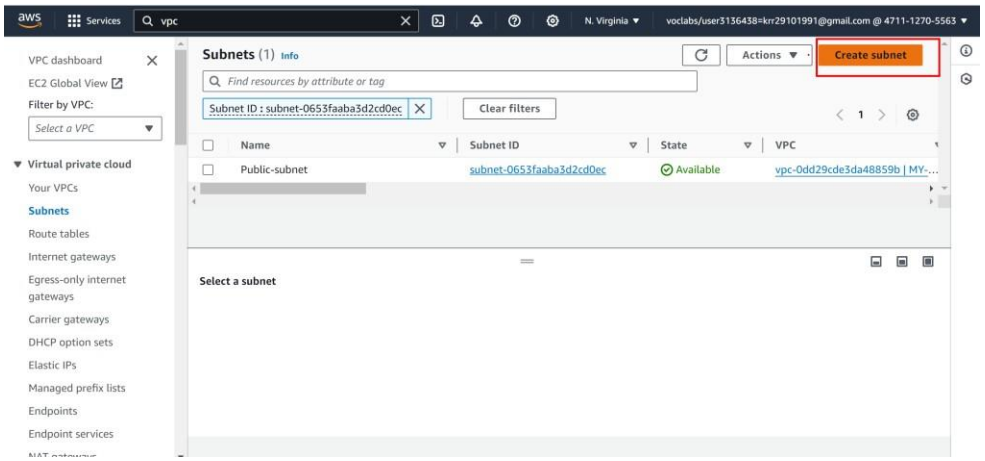
Then

a) Name our subnet
b) Select the Availability Zone
c) Set the IPV4 Subnet CIDR block
d) Click on create subnet



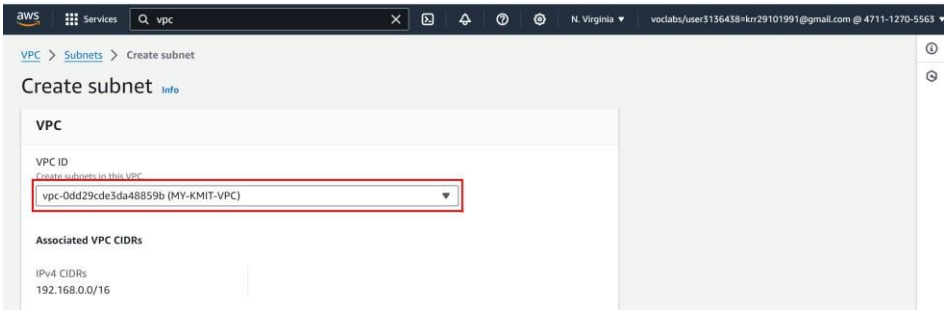


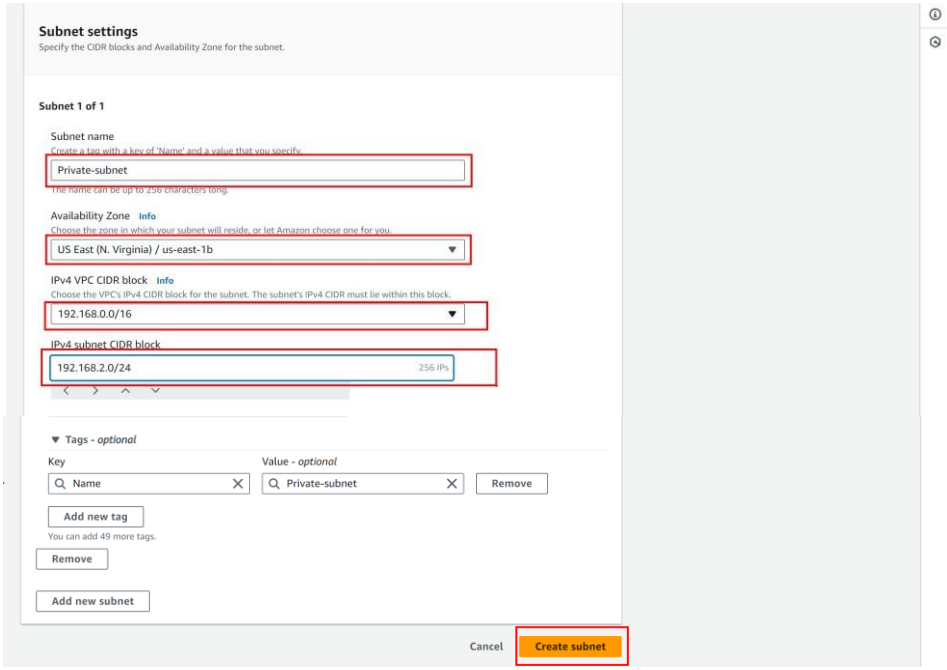
- Again Create one more subnet



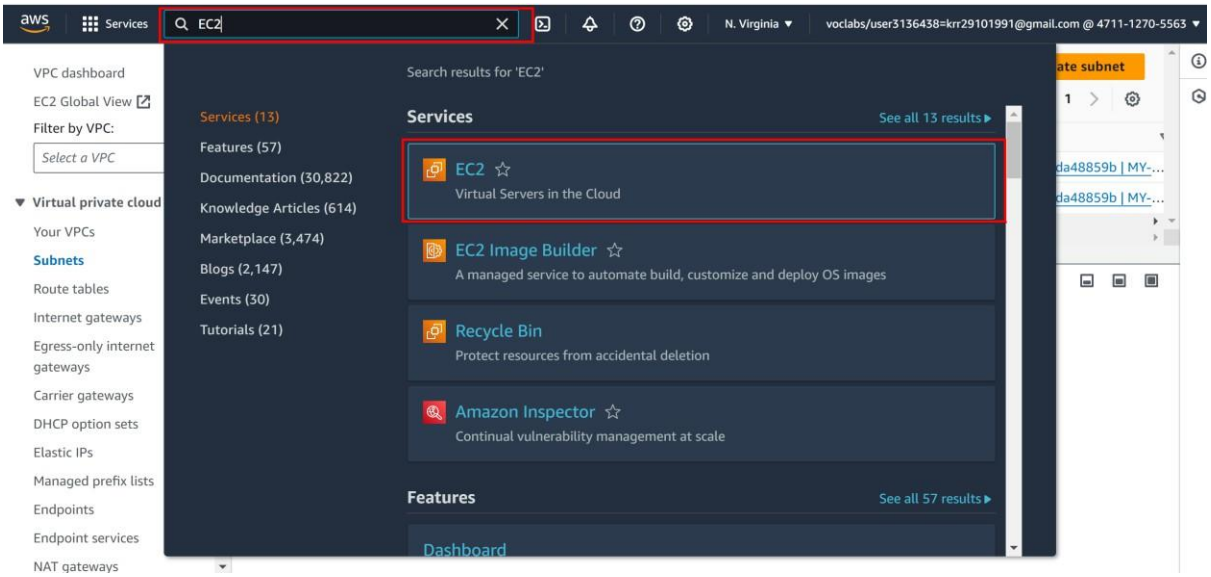
Select our VPC which you have created Then

a) Name our subnet
b) Select the Availability Zone
c) Set the IPV4 Subnet CIDR block
d) Click on create subnet

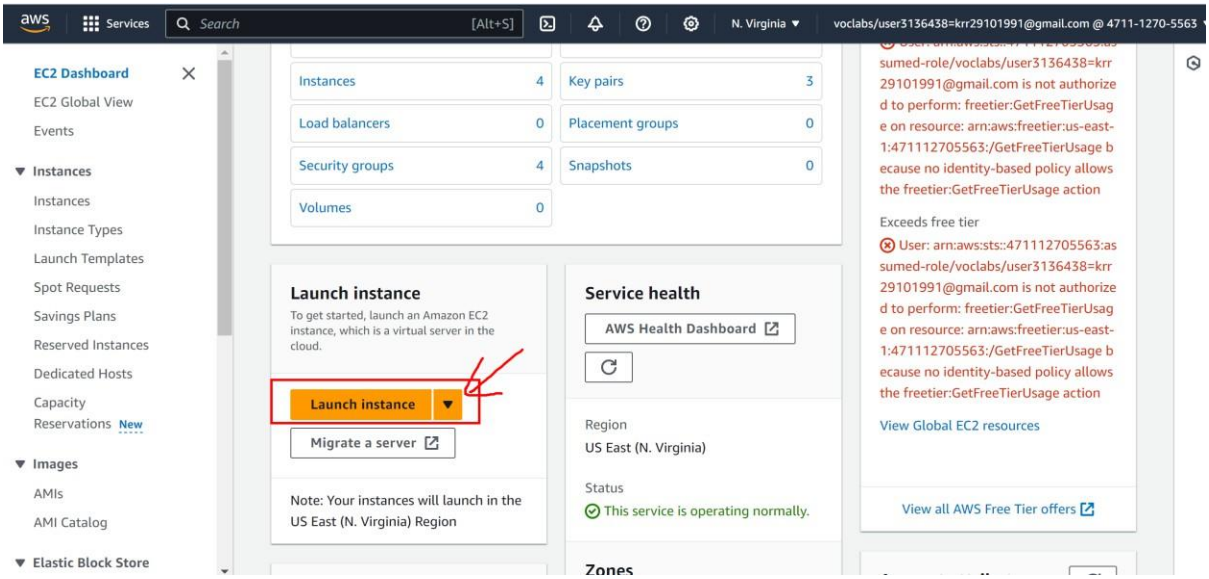




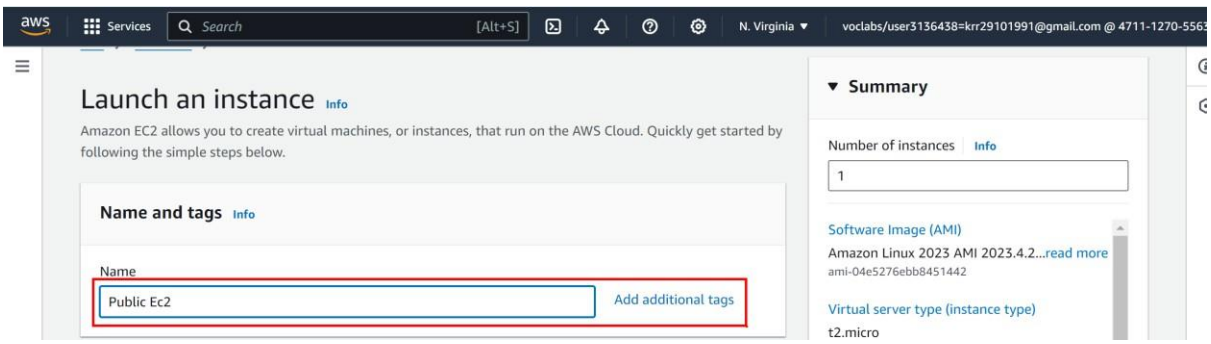
Step 7:Now Create two EC2 instances in two different subnets. Search for EC2 and click on EC2



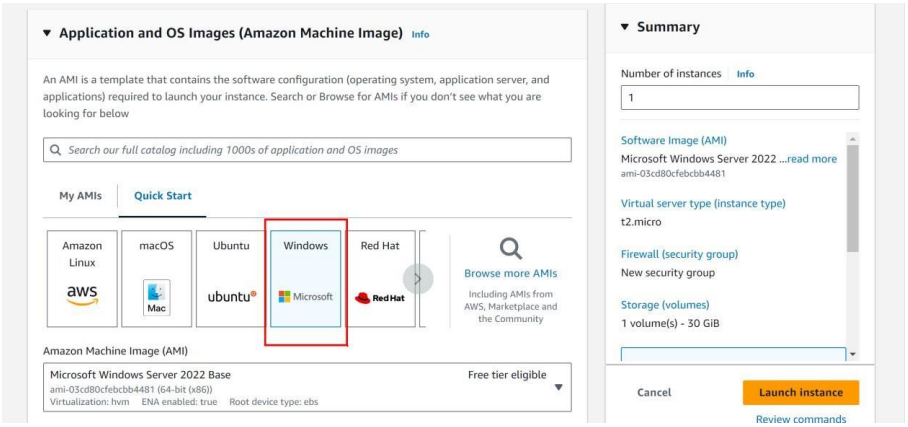
Step 8: Click on Launch Instance



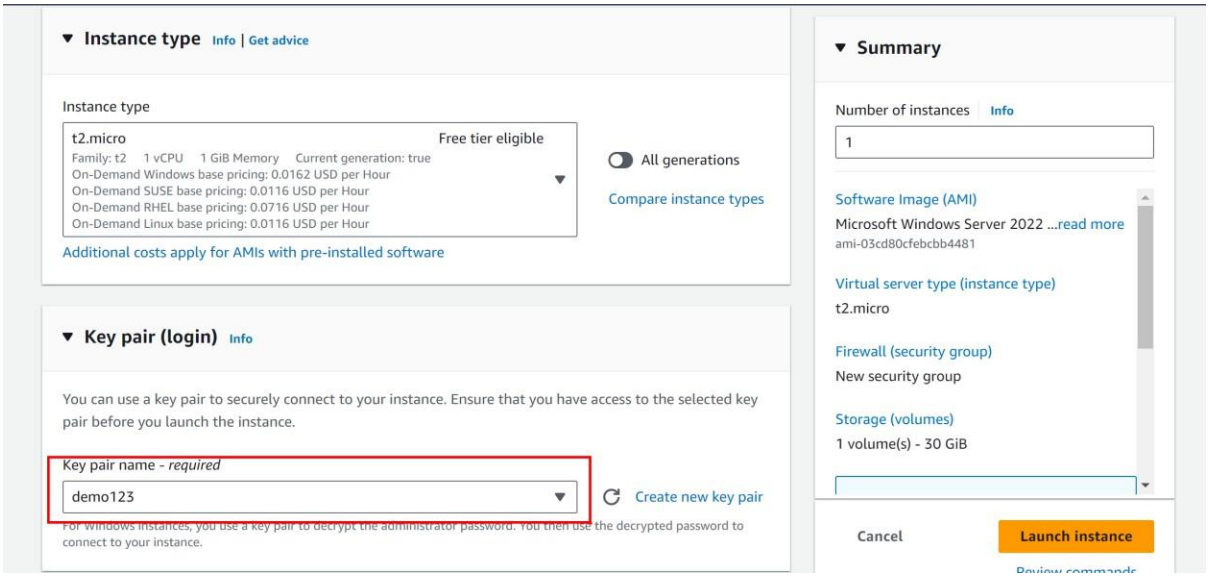
Step 9:Name the Ec2 Instance



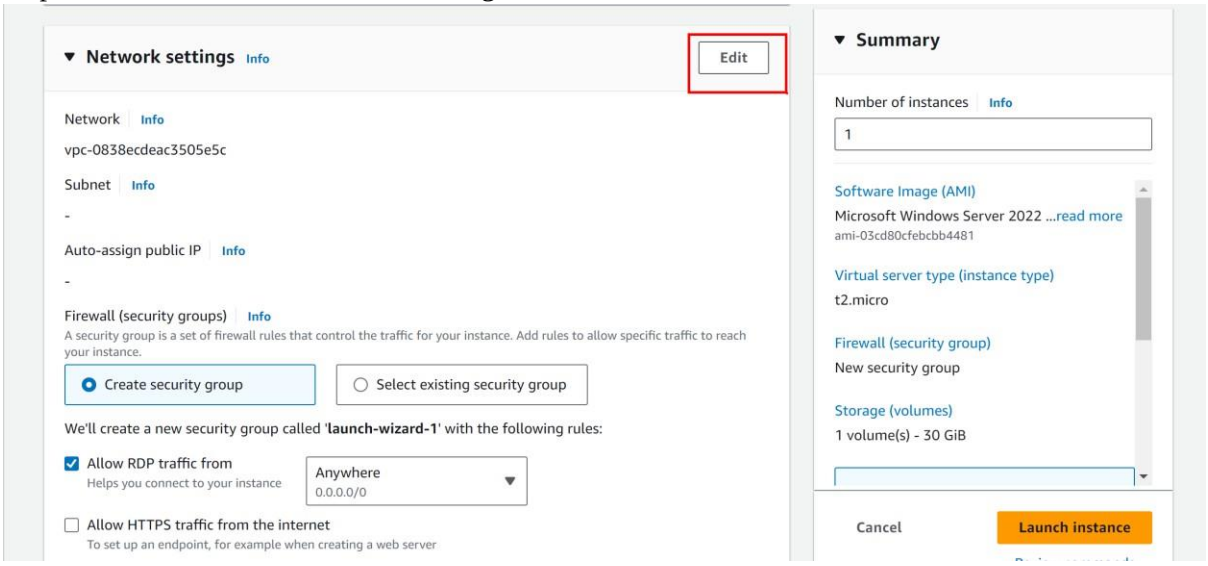
Step 11:Select the AMI



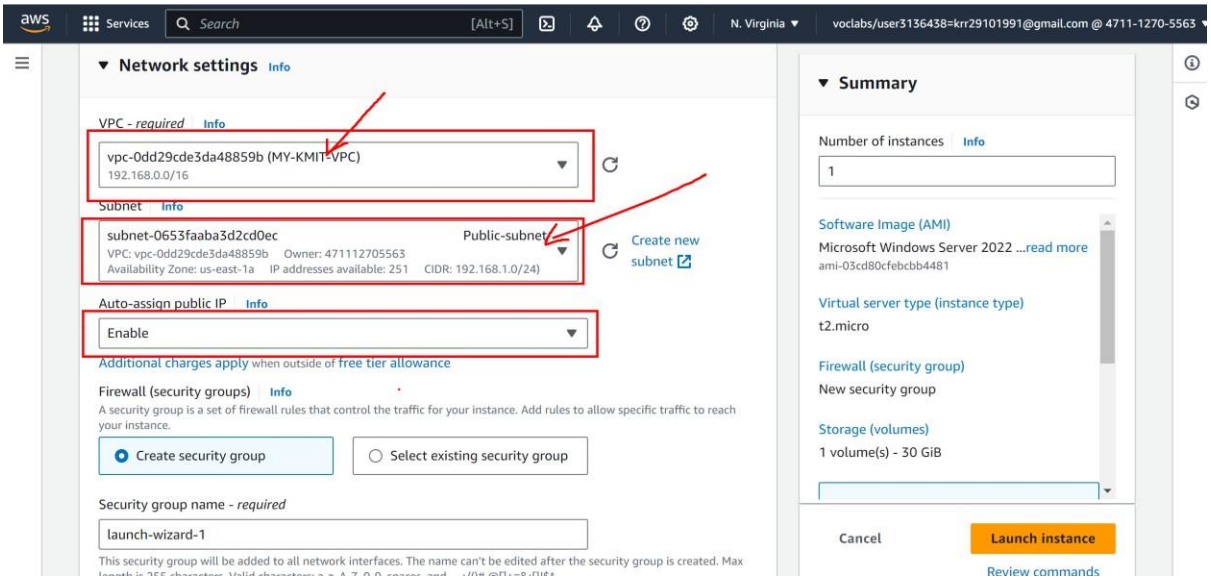
Step 12: Create a Keypair



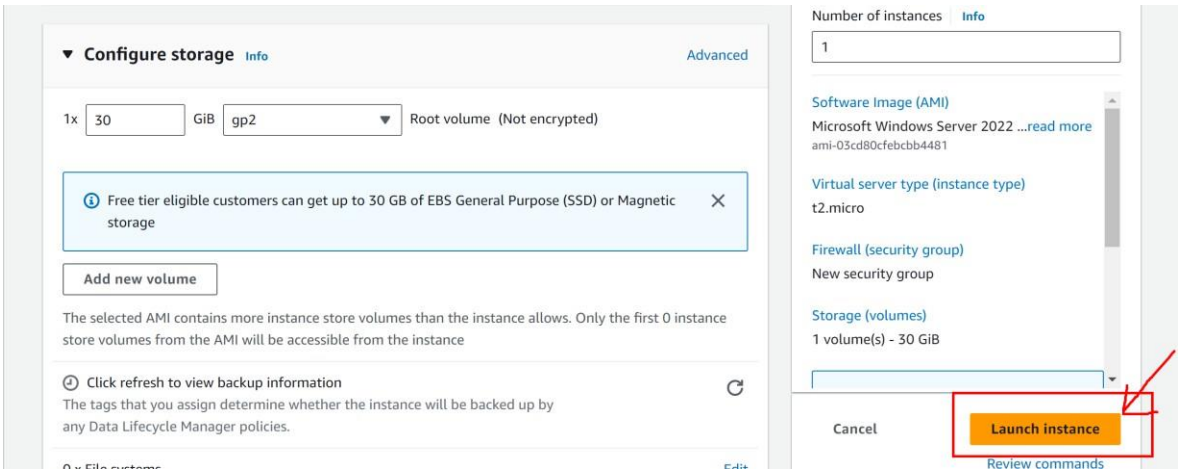
Step 13: Click on Edit in Network Settings



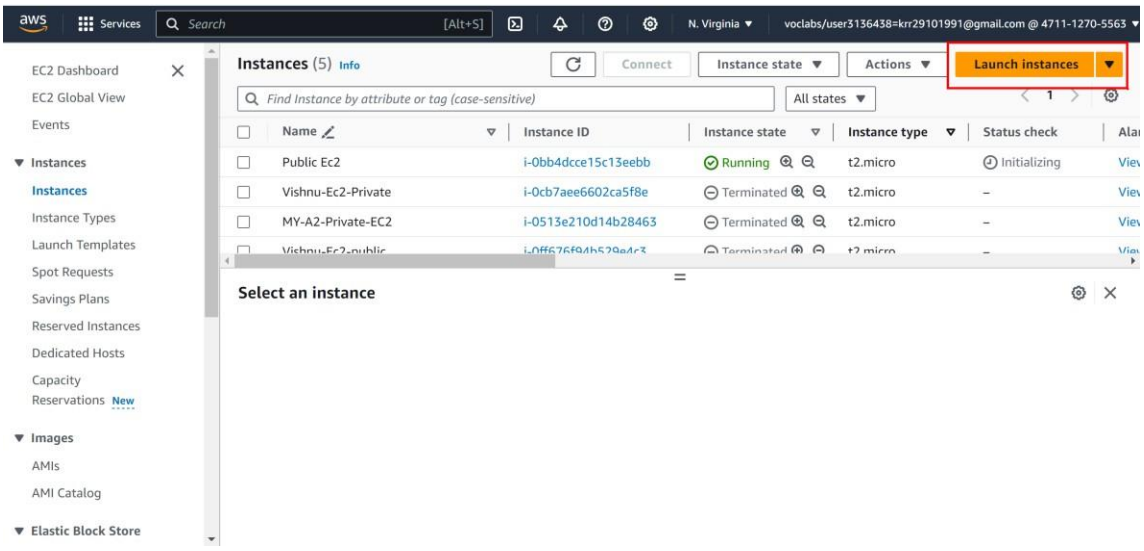
Step 14: Select the VPC which you have created then Select the Public subnet and Enable the Auto assign IpAddress



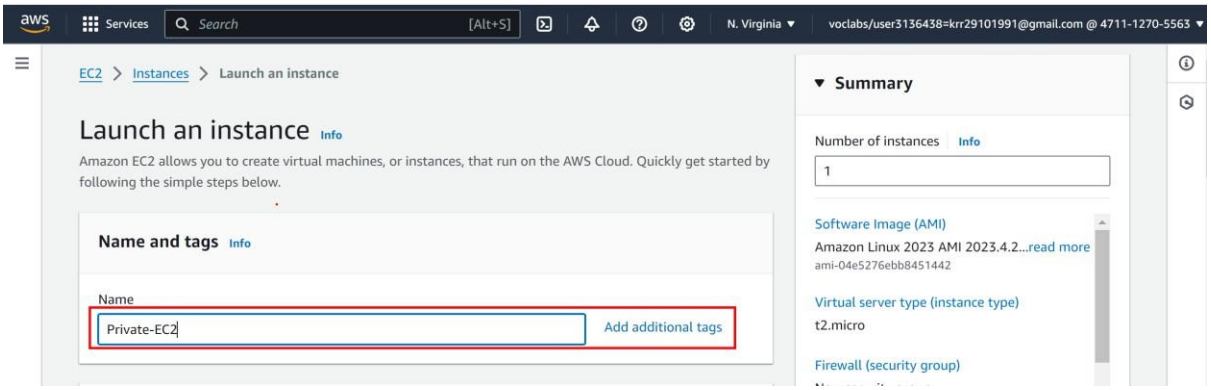
Step 15 :Click on Launch Instance



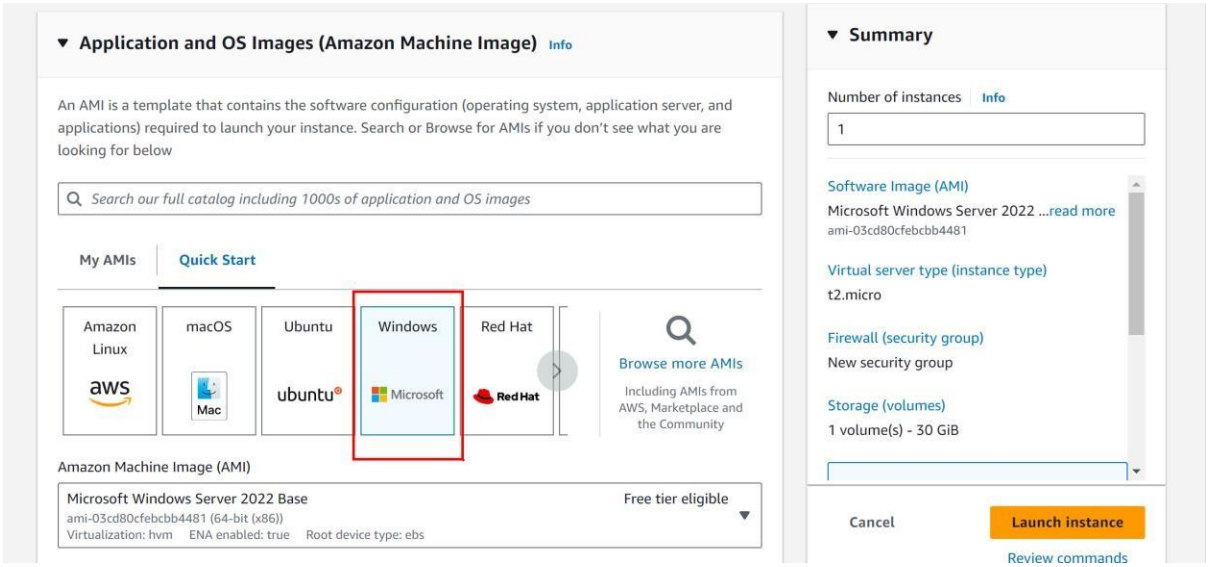
Step 16:Click on Launch Instance



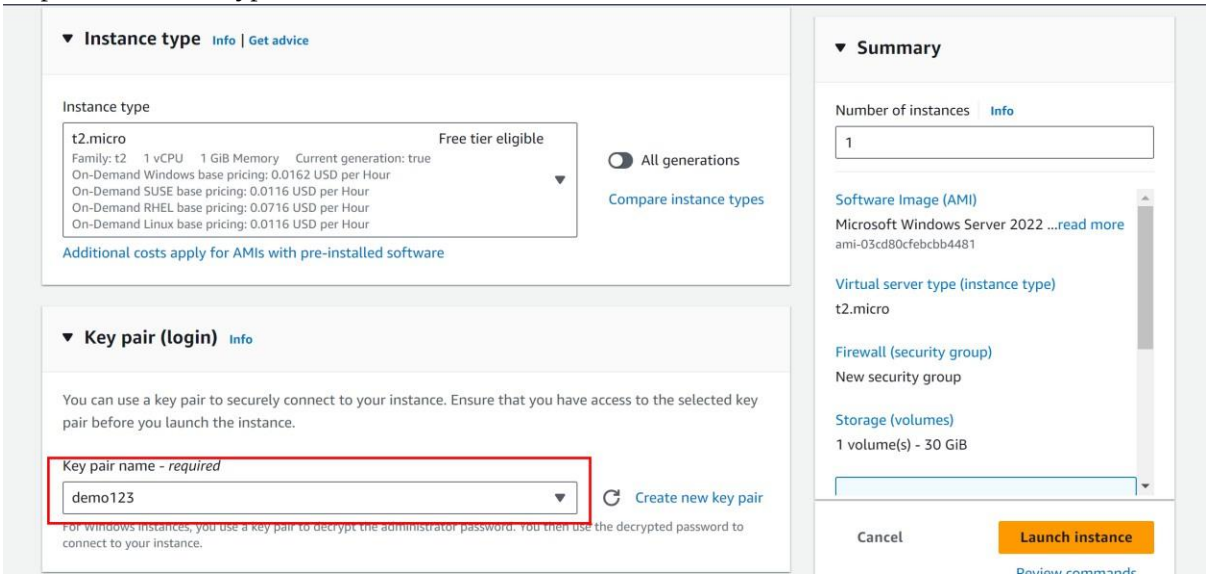
Step 17: Name the Ec2 Instance



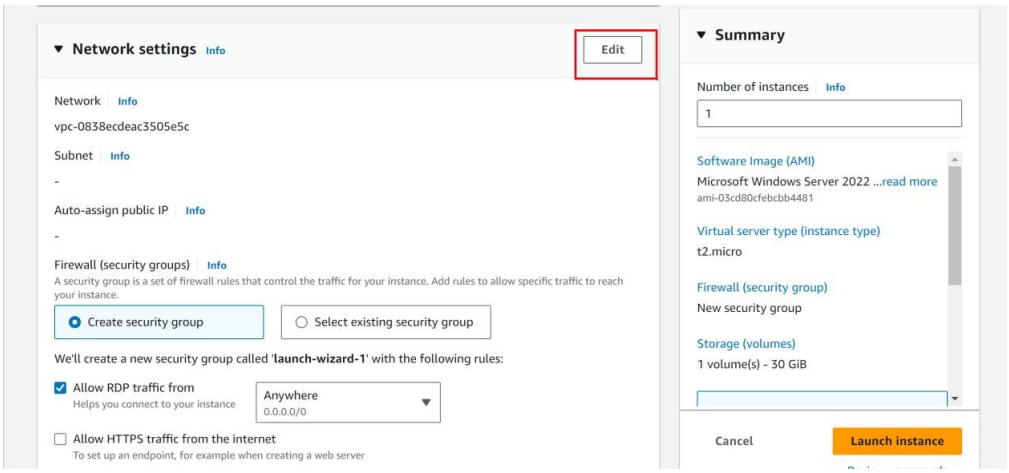
Step 18:Select the AMI



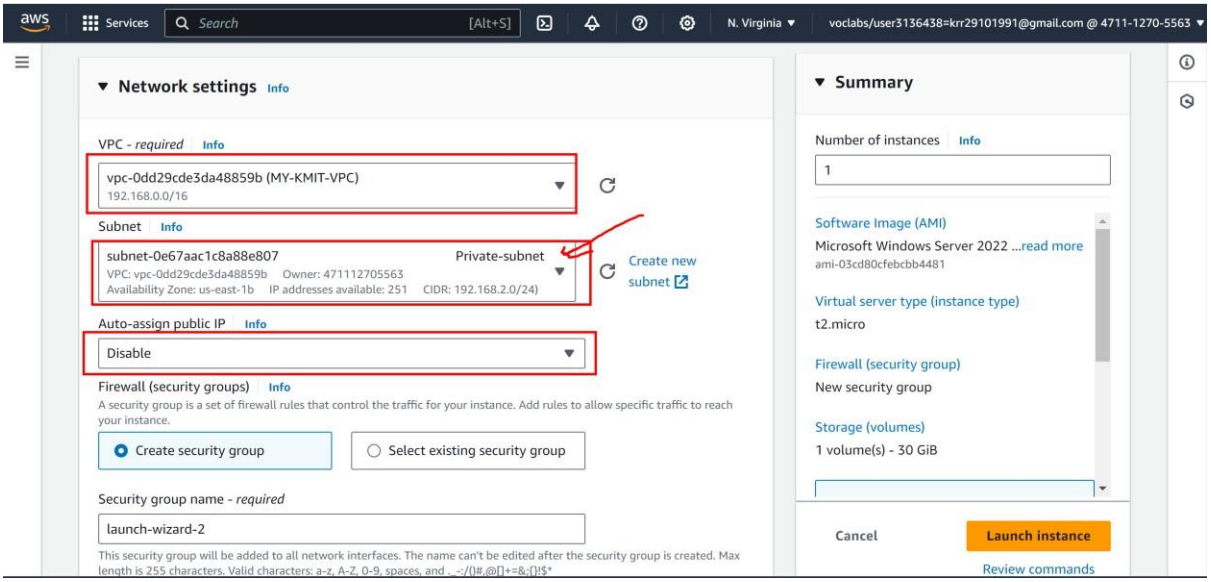
Step 19:Create a Keypair



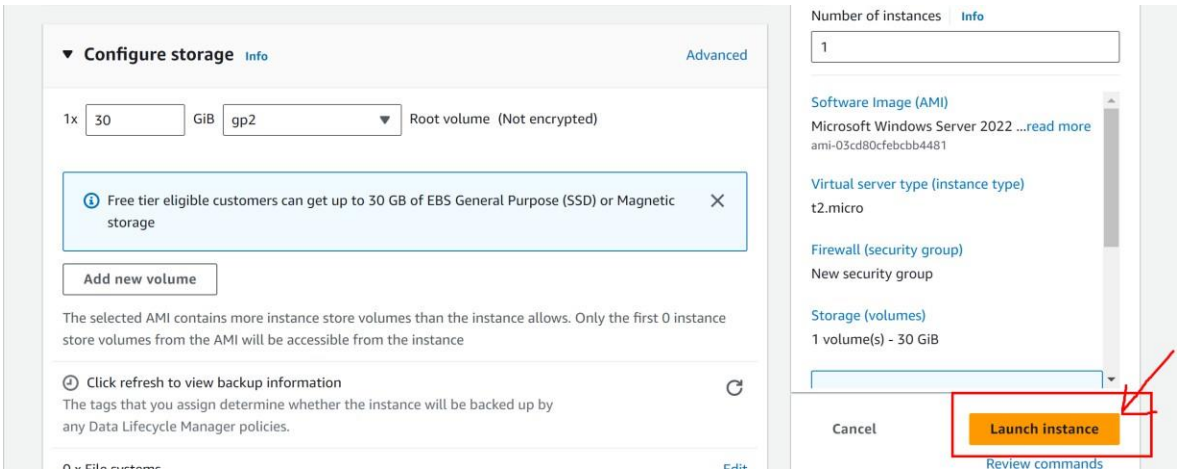
Step 20 : Click on Edit in Network Settings



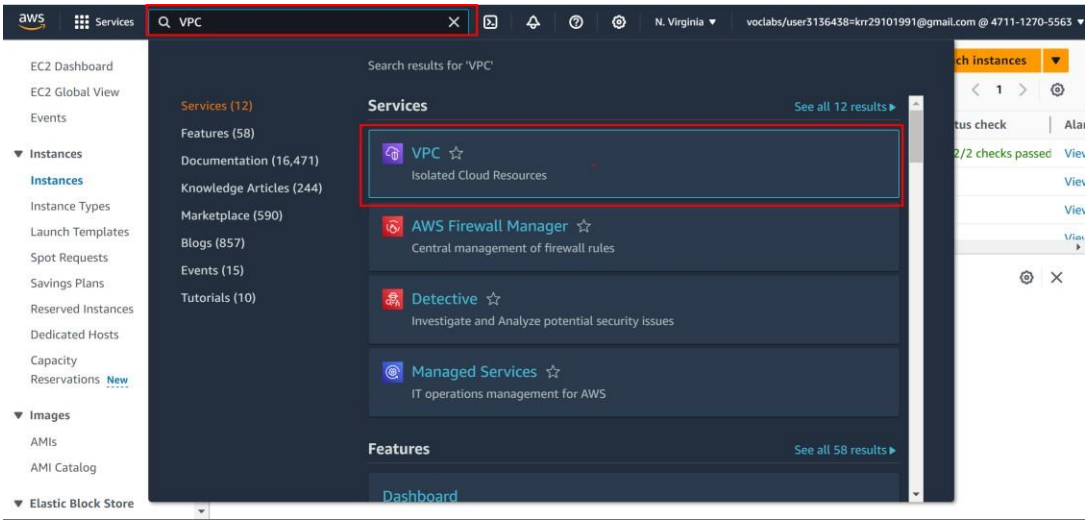
Step 21: Select the VPC which you have created then Select the Private subnet and Disable the Auto assign IP Address



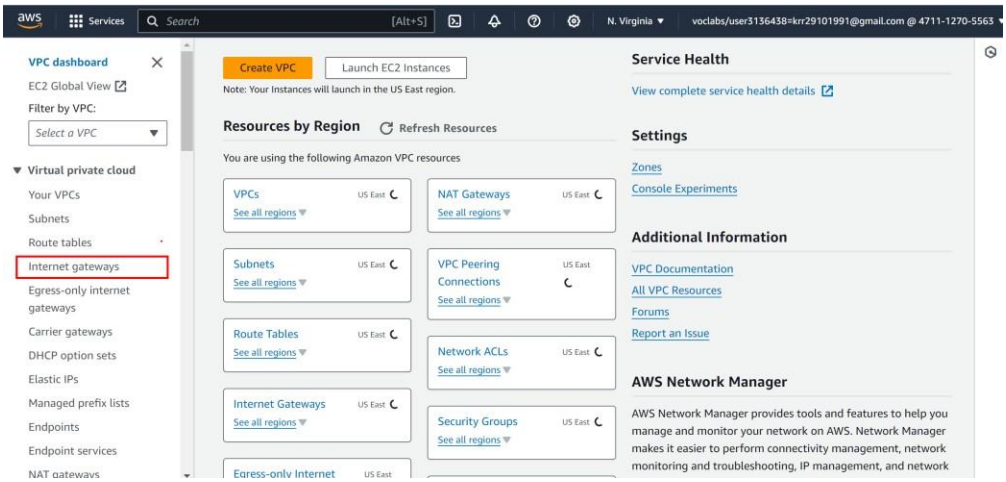
Step 22: Click on Launch Instance



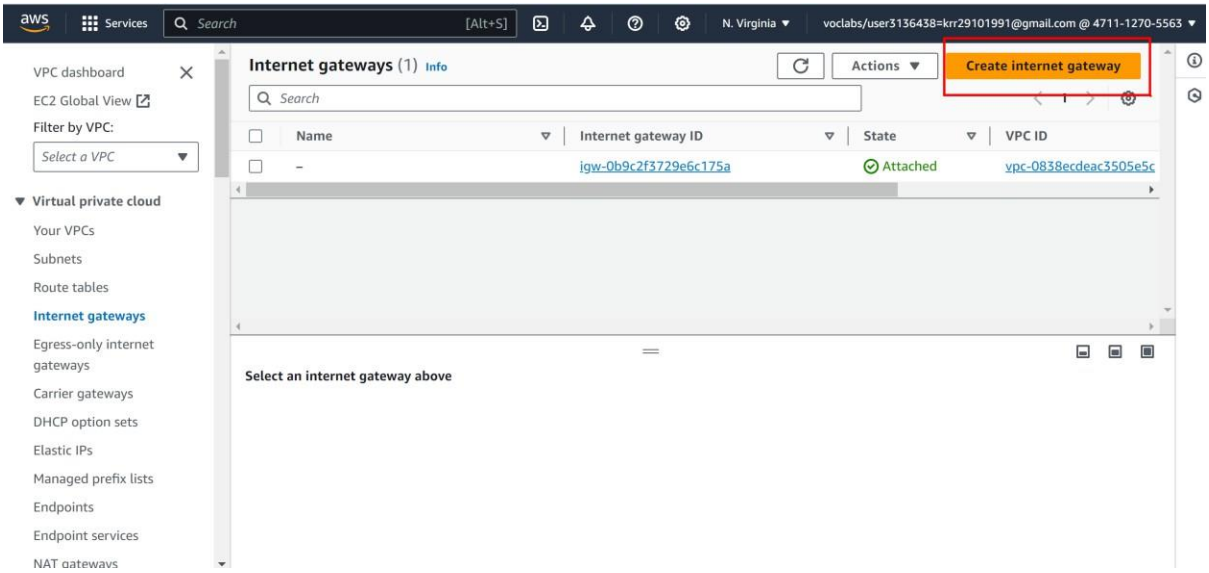
Step 23: Now Search for VPC and Select the VPC



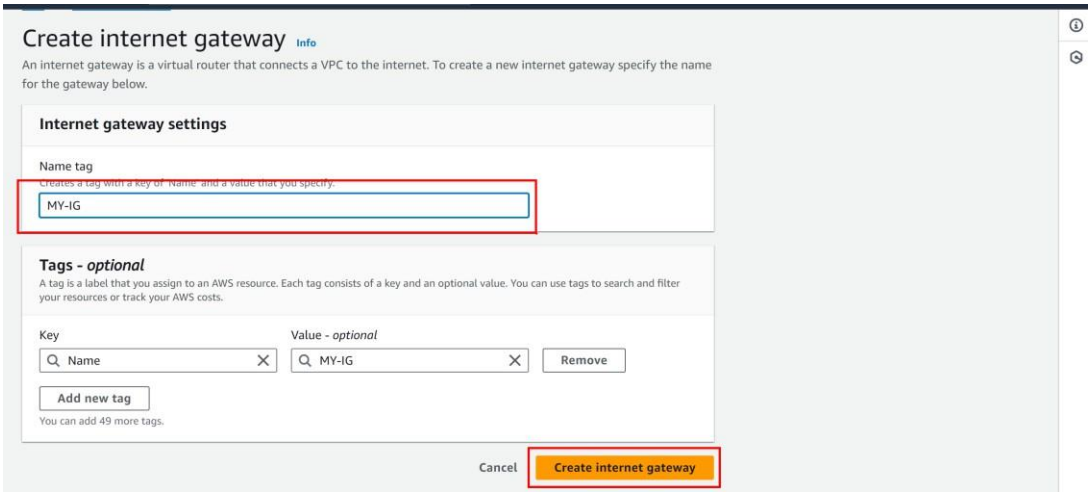
Step 24: Now click on Internet Gateway



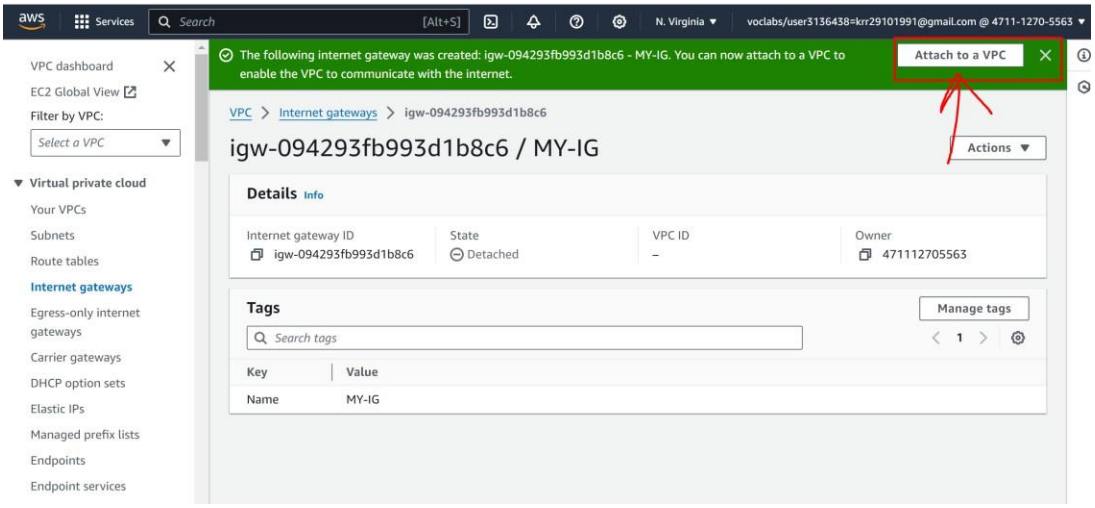
Step 25 :Click on Create Internet Gateways



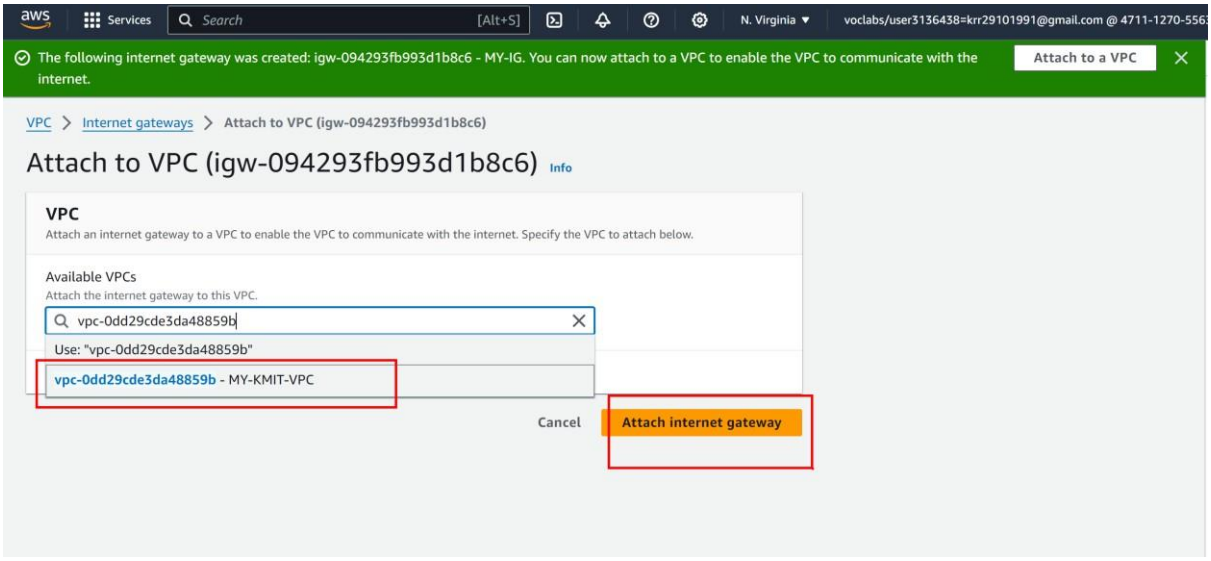
Step 26: Name our Internet gateway and click on Create internet gateway



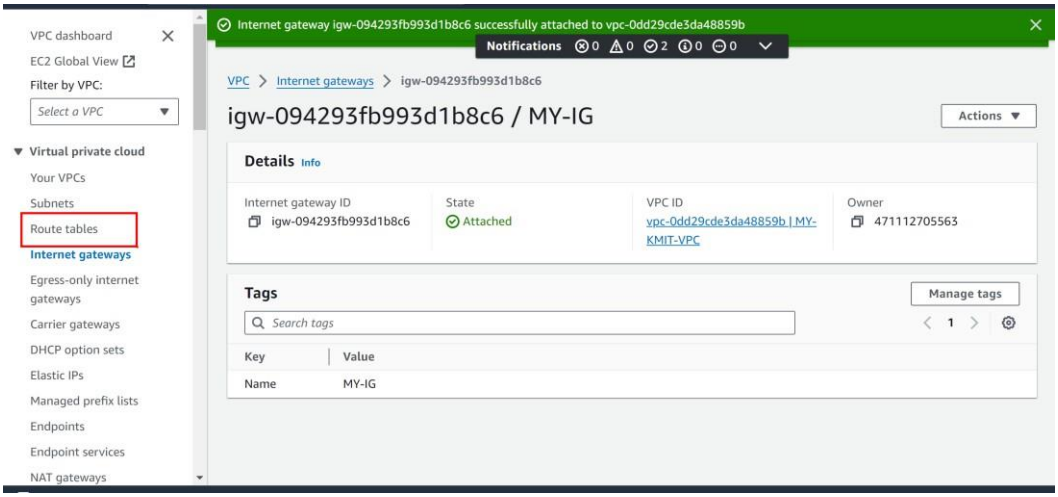
Step 27 : Click on Attach to a VPC



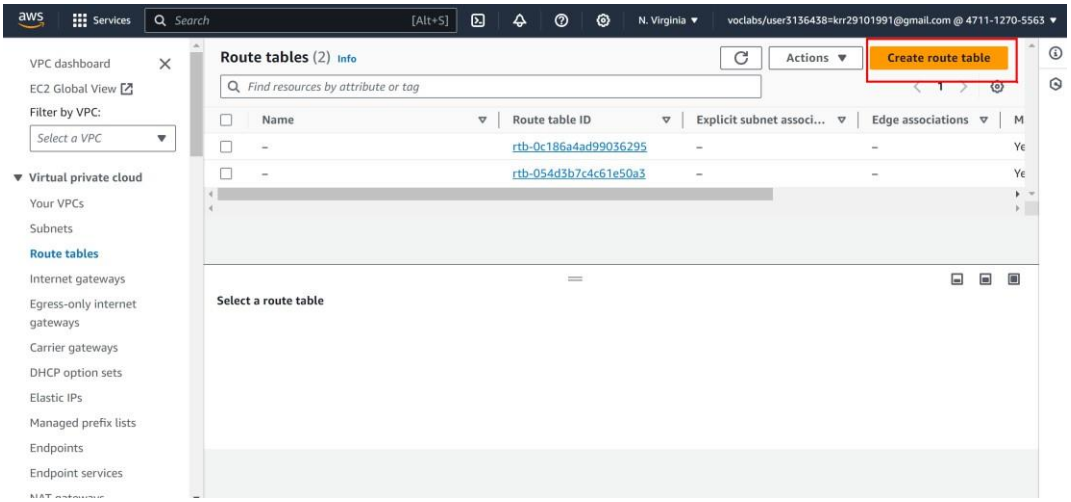
Step 28:Select our VPC and Click On Attach Internet Gateway



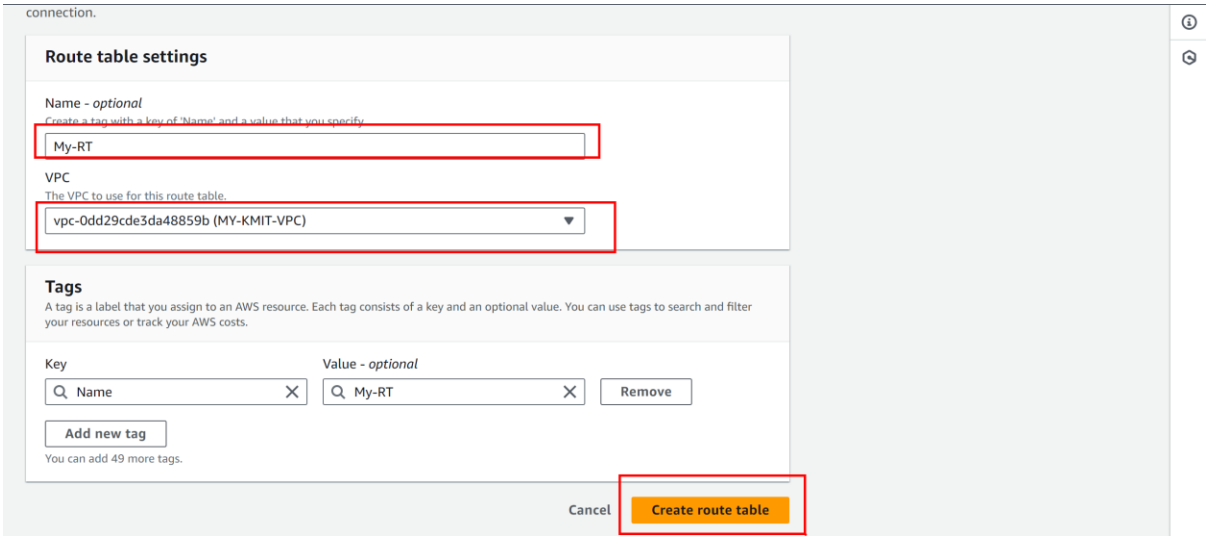
Step 29: Now click on Route Tables



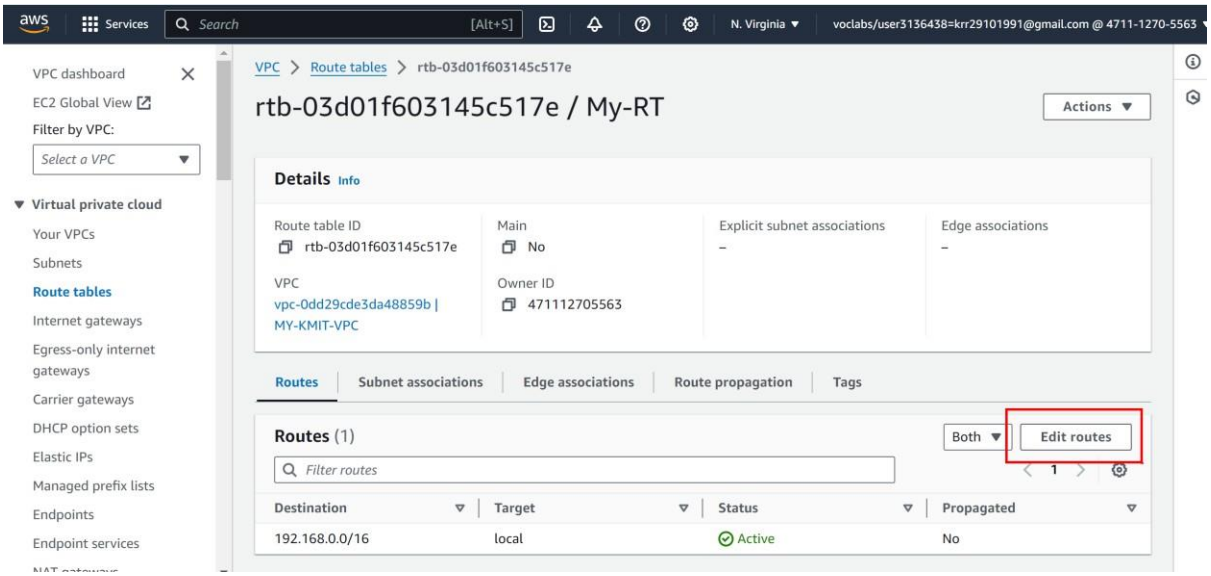
Step 30 :Click On create Route tables Name our Route table



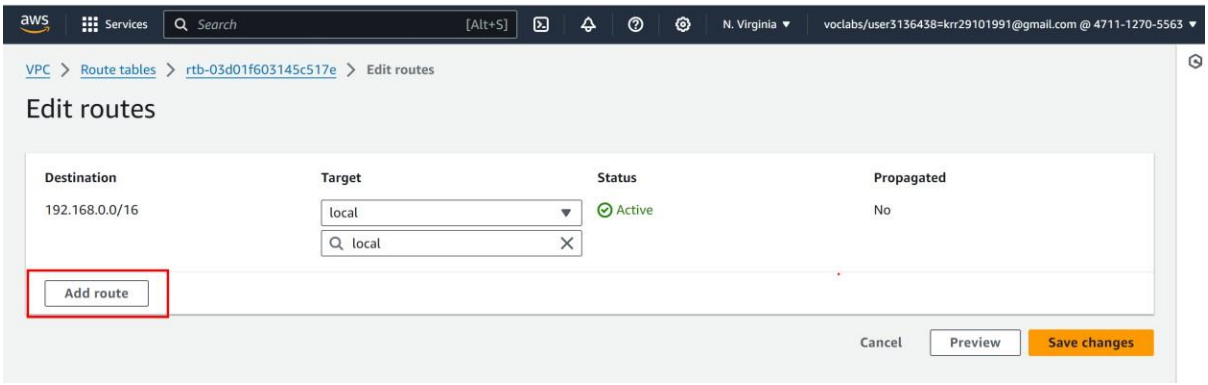
Step 31: Select our VPC and Click Create Route Table



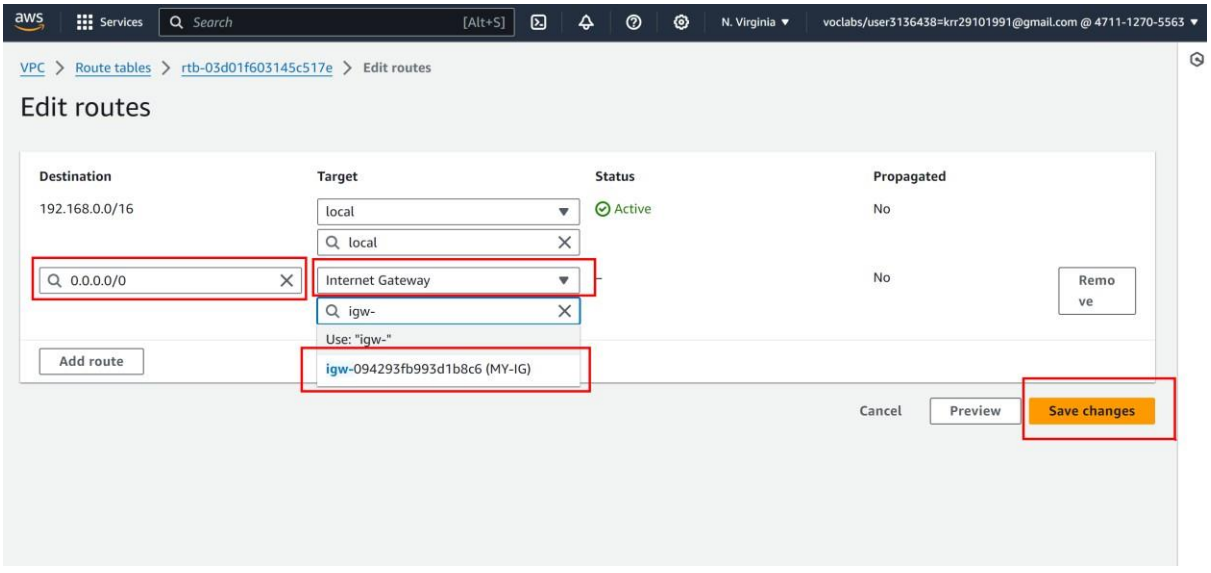
Step 32: Now Click on Edit Routes



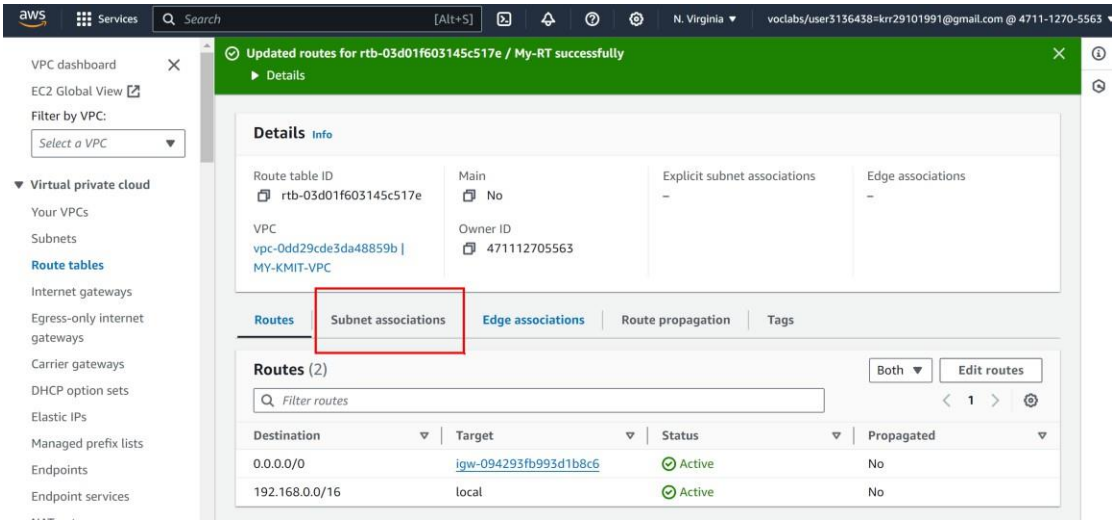
Step 33: Click on Add Route



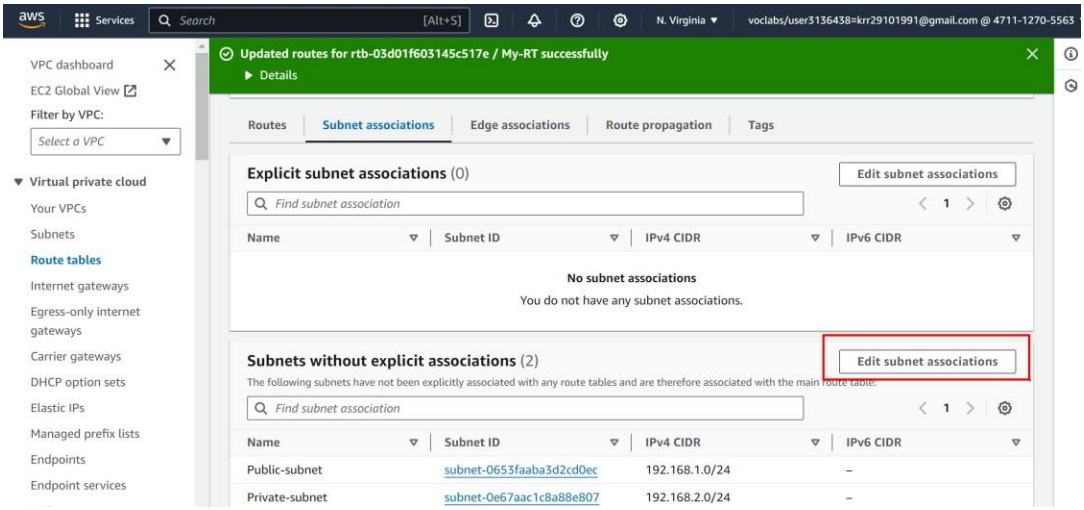
Step 34 :Now add from Any Where IP Address then select the Internet Gateway from the drop down and select our Internet Gateway then click on Save Changes



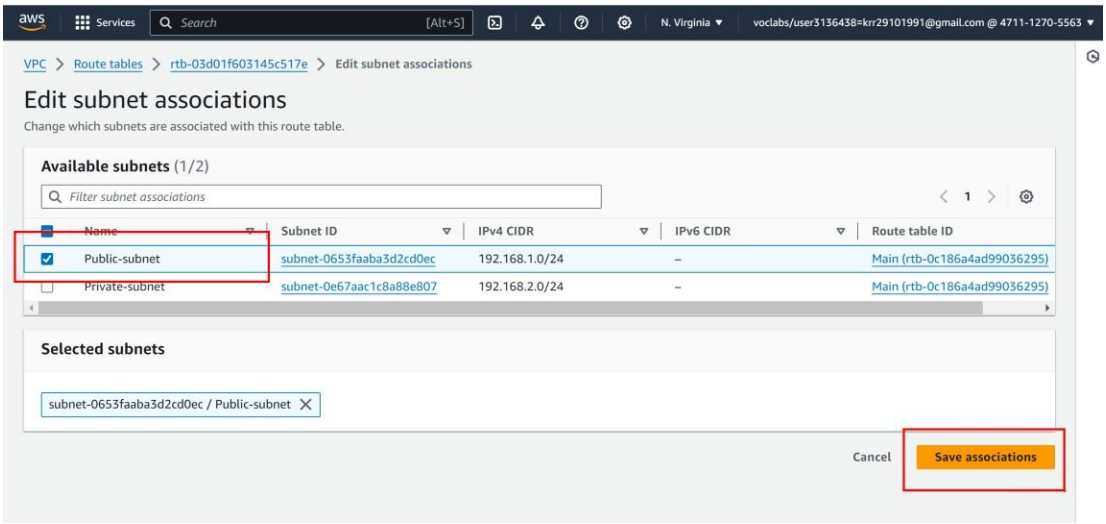
Step 36: Click on Subnet Association



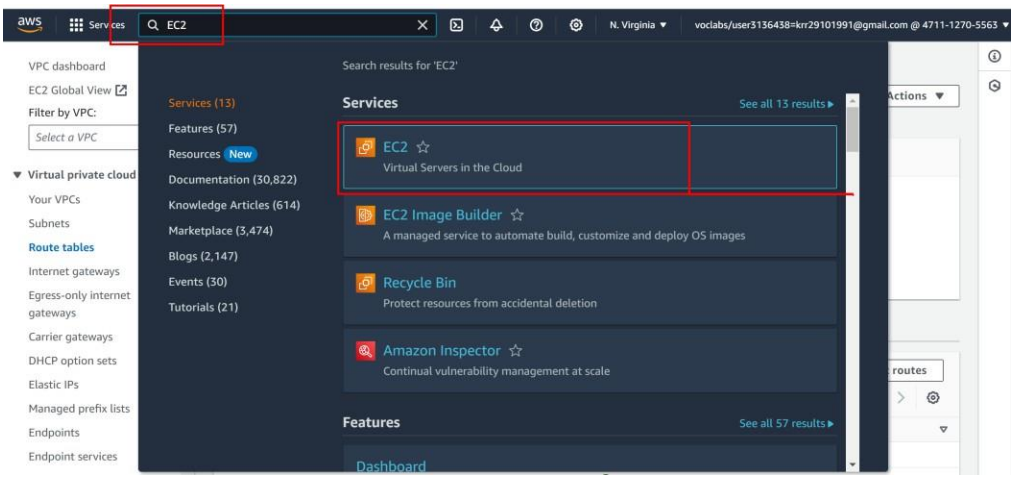
Step 37: Click on Edit Subnet Associations



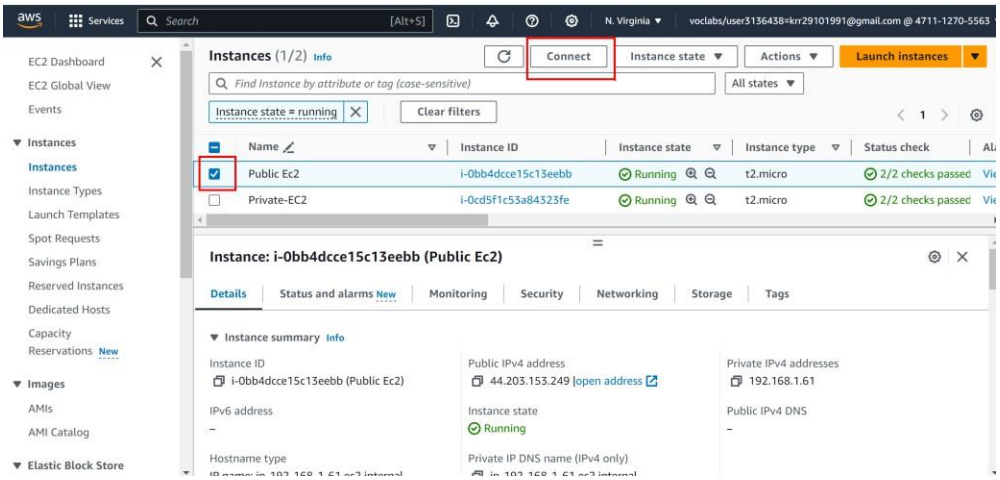
Step 38: Select the Public-subnet and click on save associations



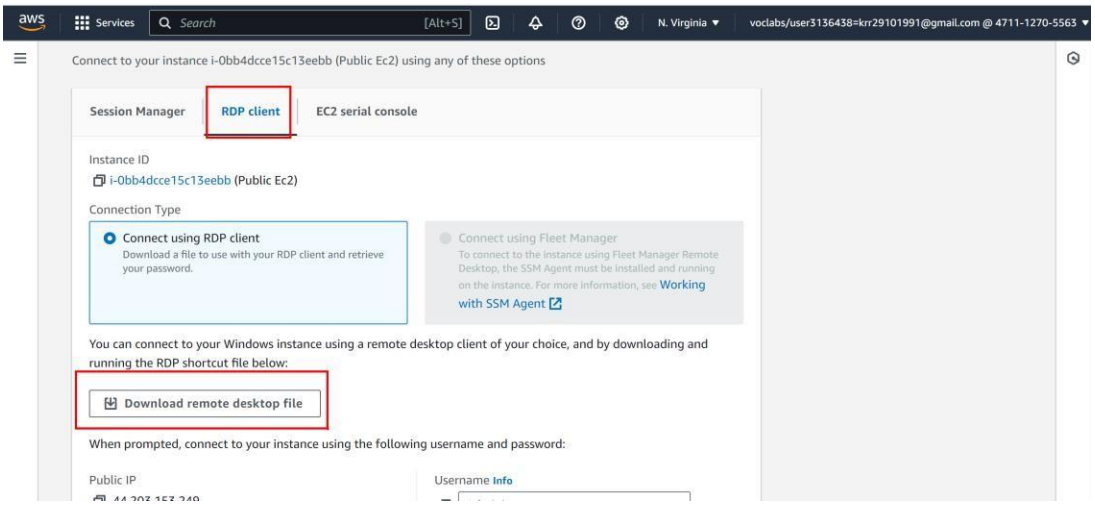
Step 39 : Now search for EC2 And click on the Ec2



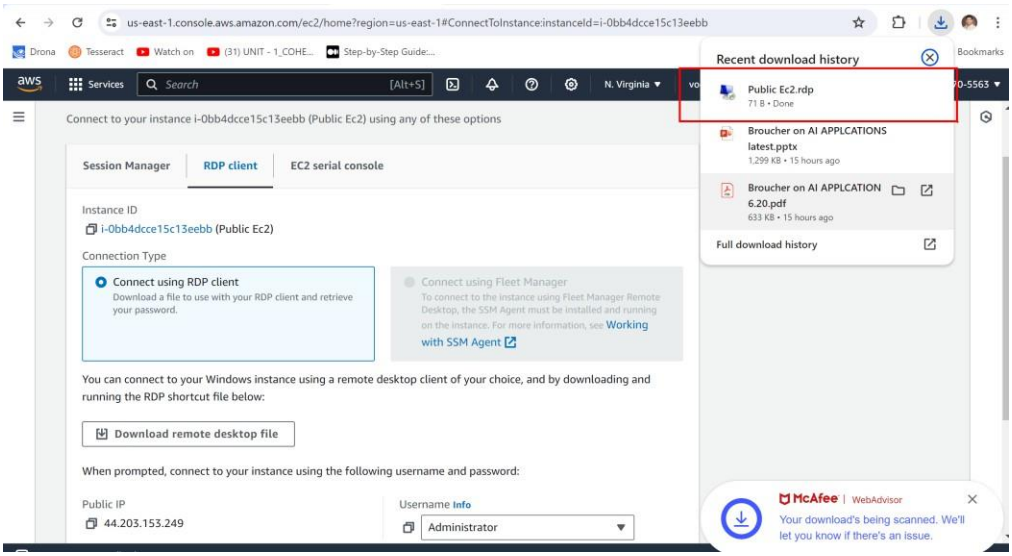
Step 40:Select the EC2 Instance and Click on Connect



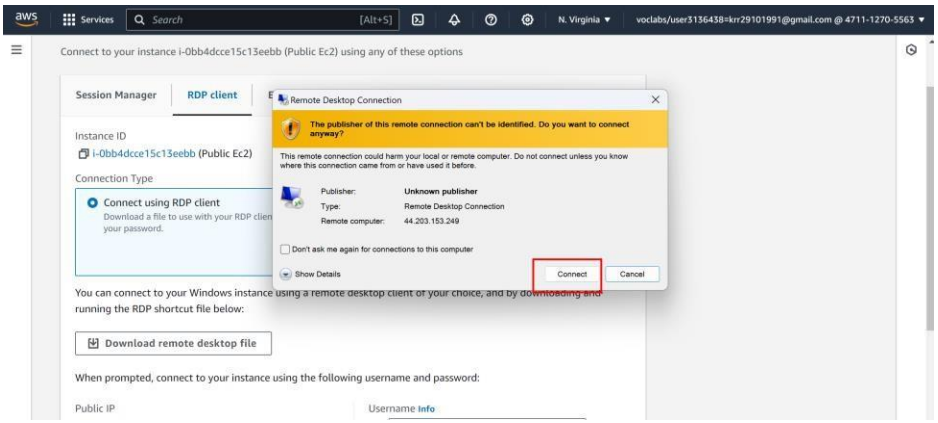
Step 41:Click on RDP Client and Click on Download remote desktop file



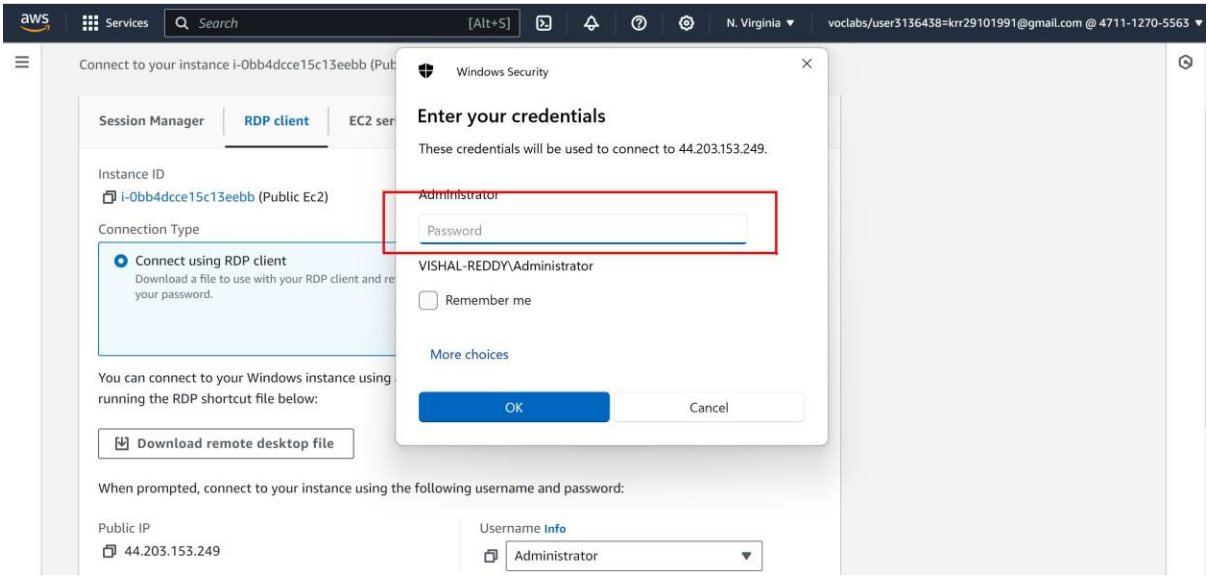
Step 42: Open the downloaded file by clicking on it



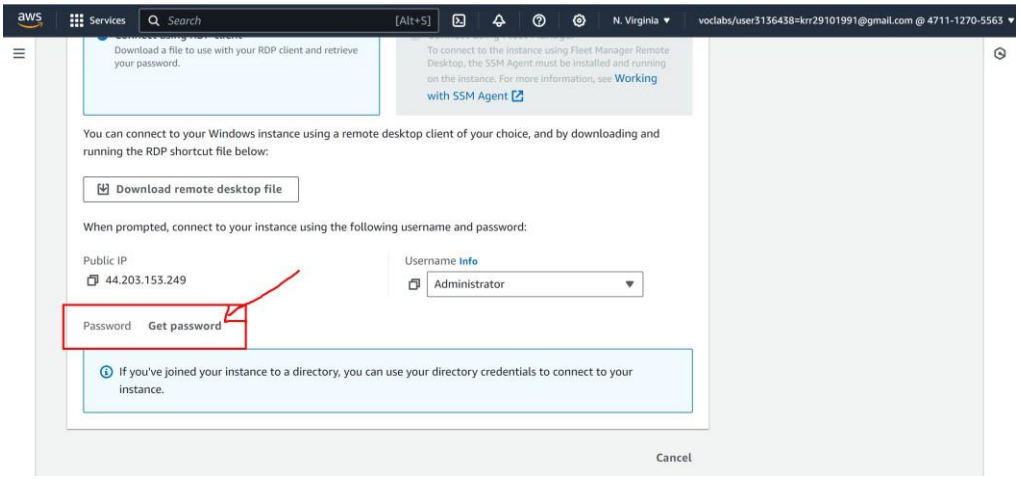
Step 43 :Click on connect



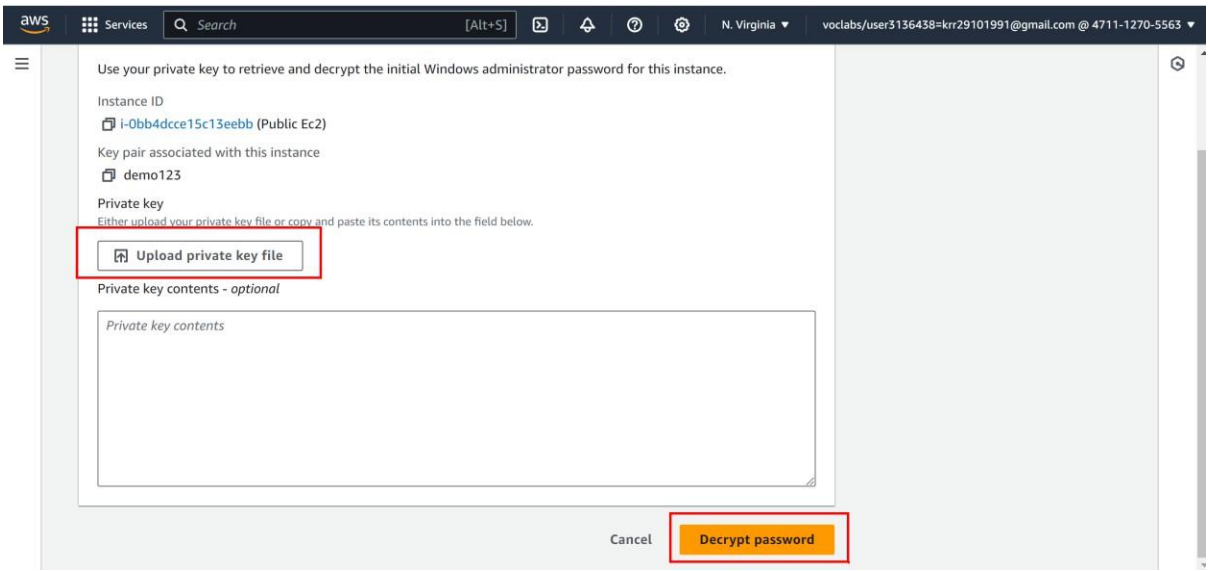
It will prompt for Password



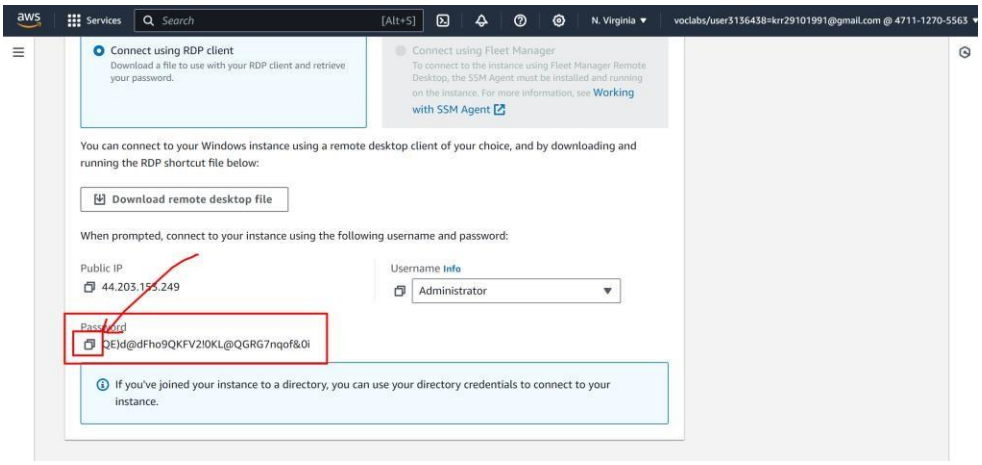
Step 44 : For Password get back to AWS EC2 Browser and click on get password

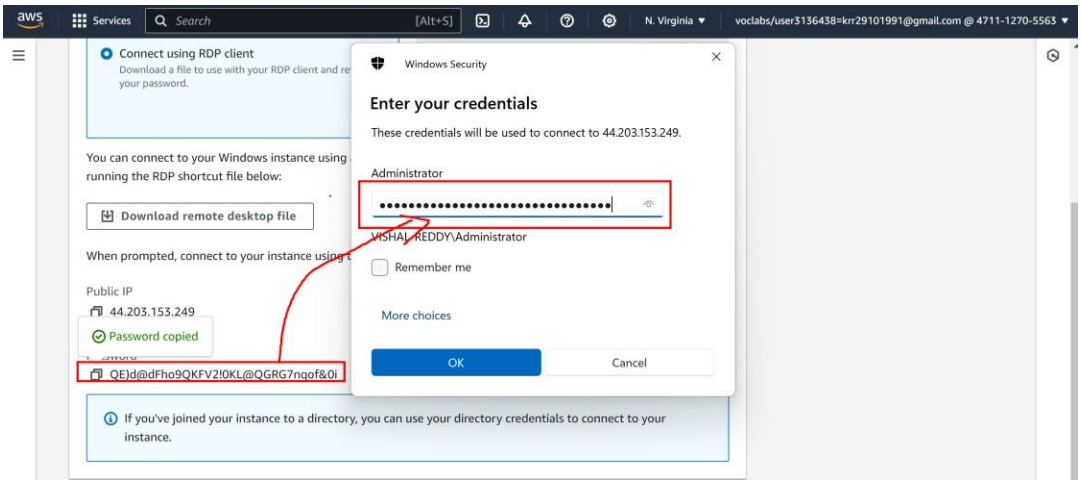


Step 45: Upload our private key and click on decrypt password

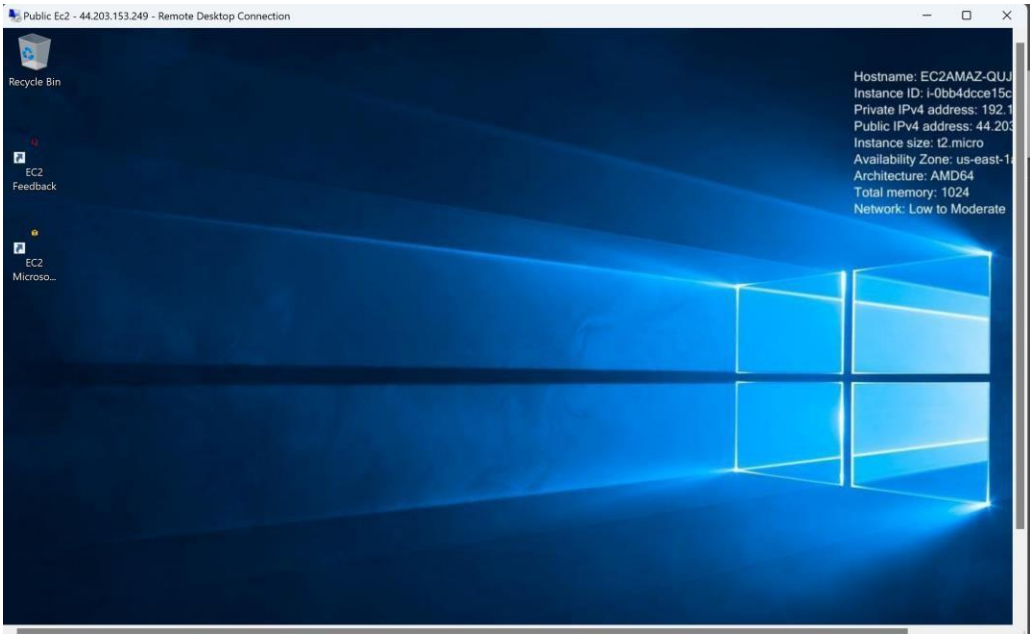


Step 46: Copy the password and paste it on the window and then click on OK and yes

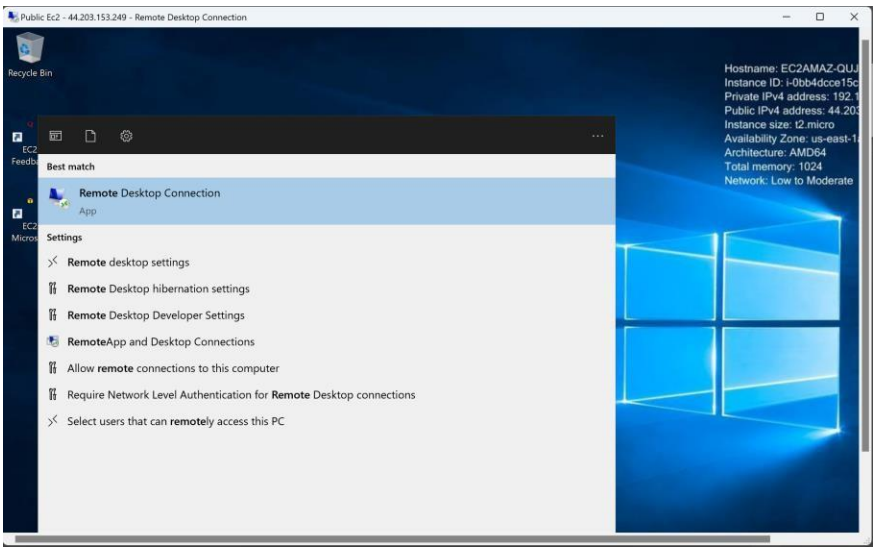


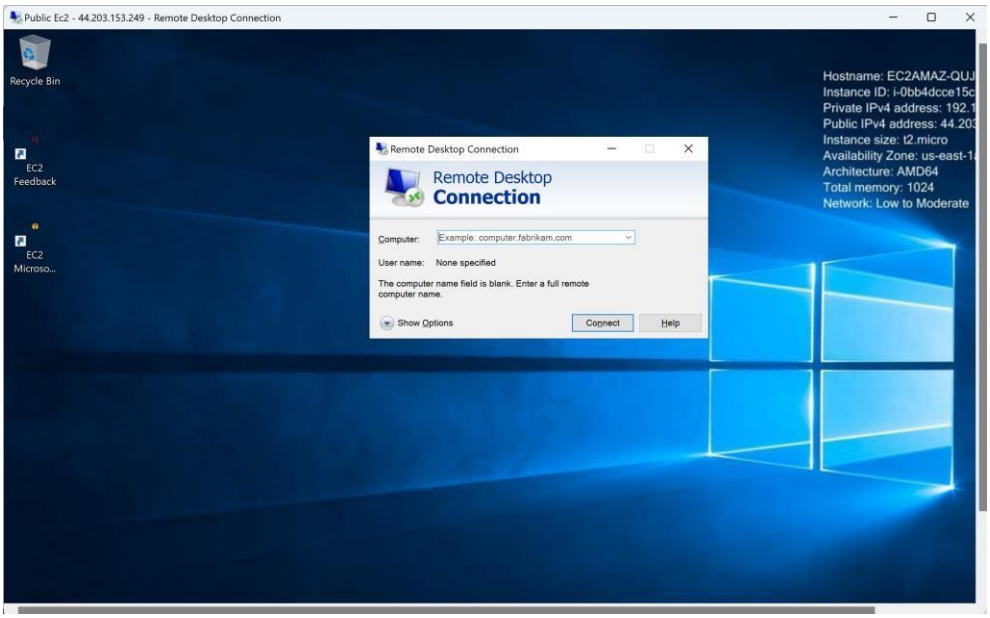


Our EC2 WILL BE OPENED

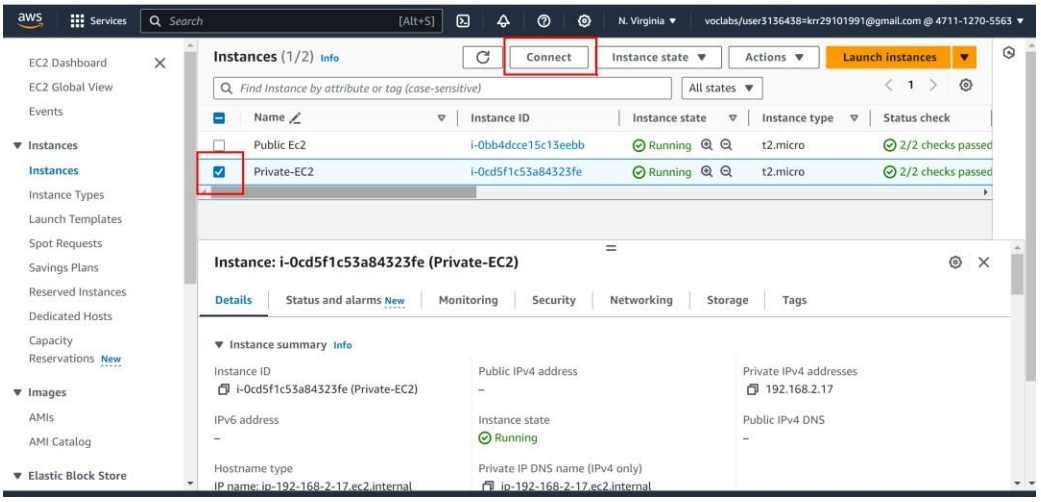


Step 47: Search for remote desktop connection within our EC2 Instance

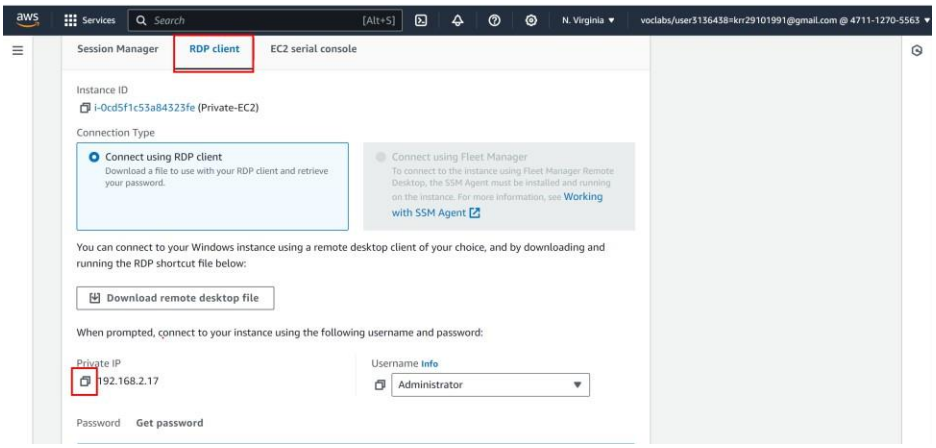




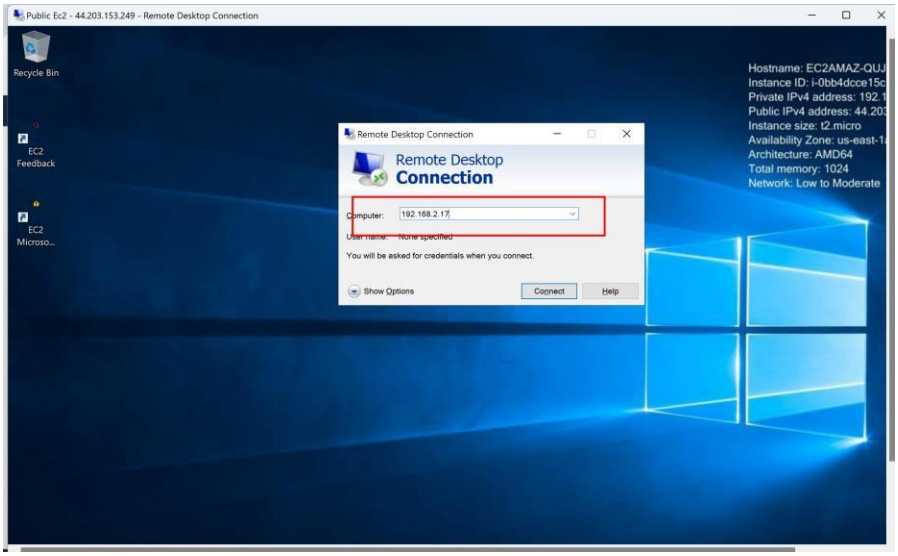
Step 48: Go back to the Browser and Open instances now select the Private EC2 instance and click on connect



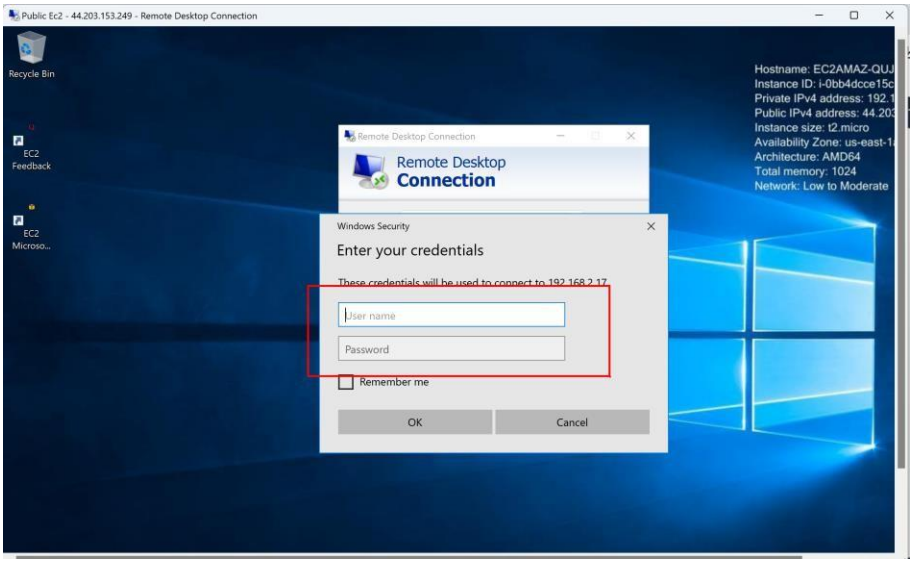
Step 49: Click on RDP Client and Copy the Private IP



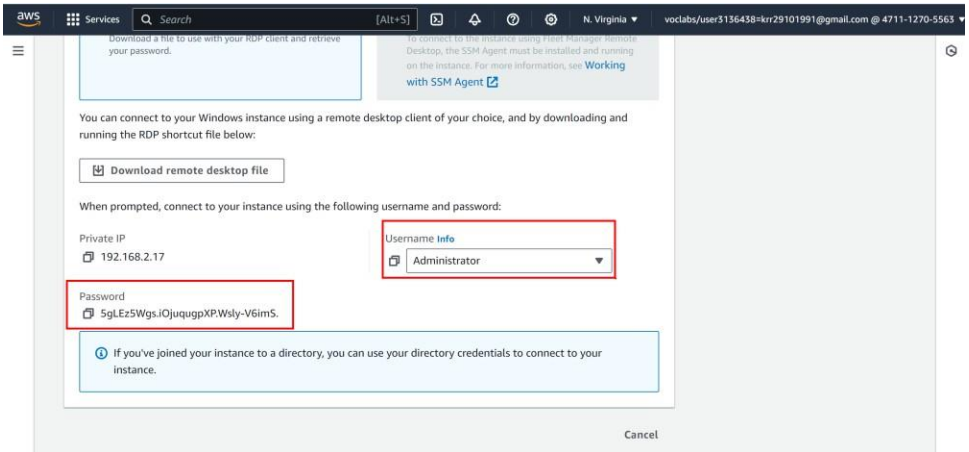
Step 50: Paste the Private Ip on the EC2 Remote Desktop Connection and then click on connect



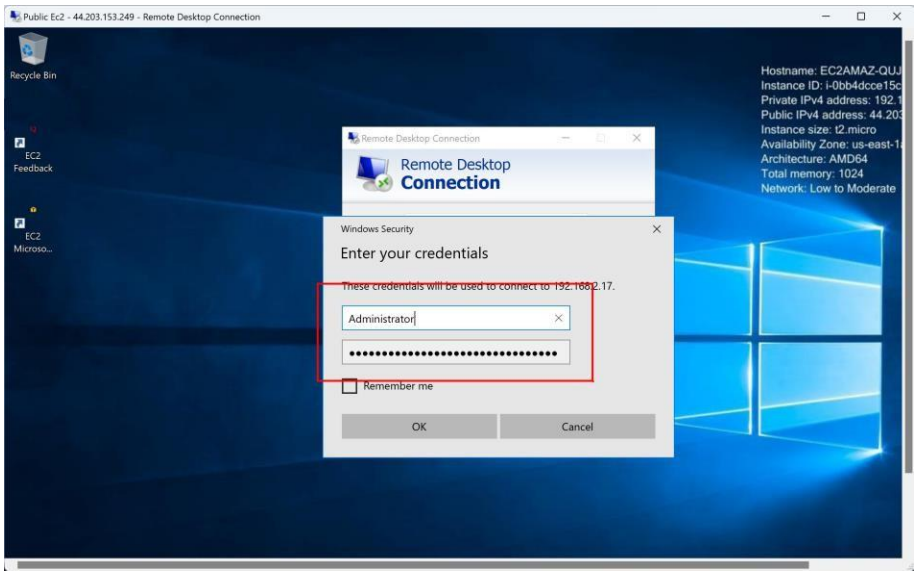
Step 51: You will be prompted for Username and Password

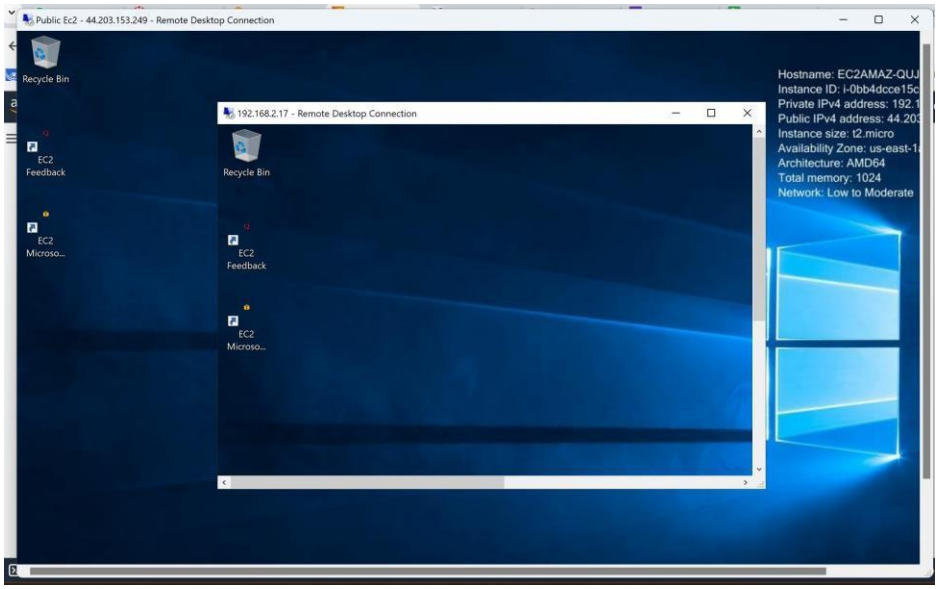


Step 52: Now go back to EC2 Browser and click on the getpassword and upload private key and click on decrypt password copy the password



Step 53: Paste the Username and Password on the EC2 Instance and click on ok and then yes

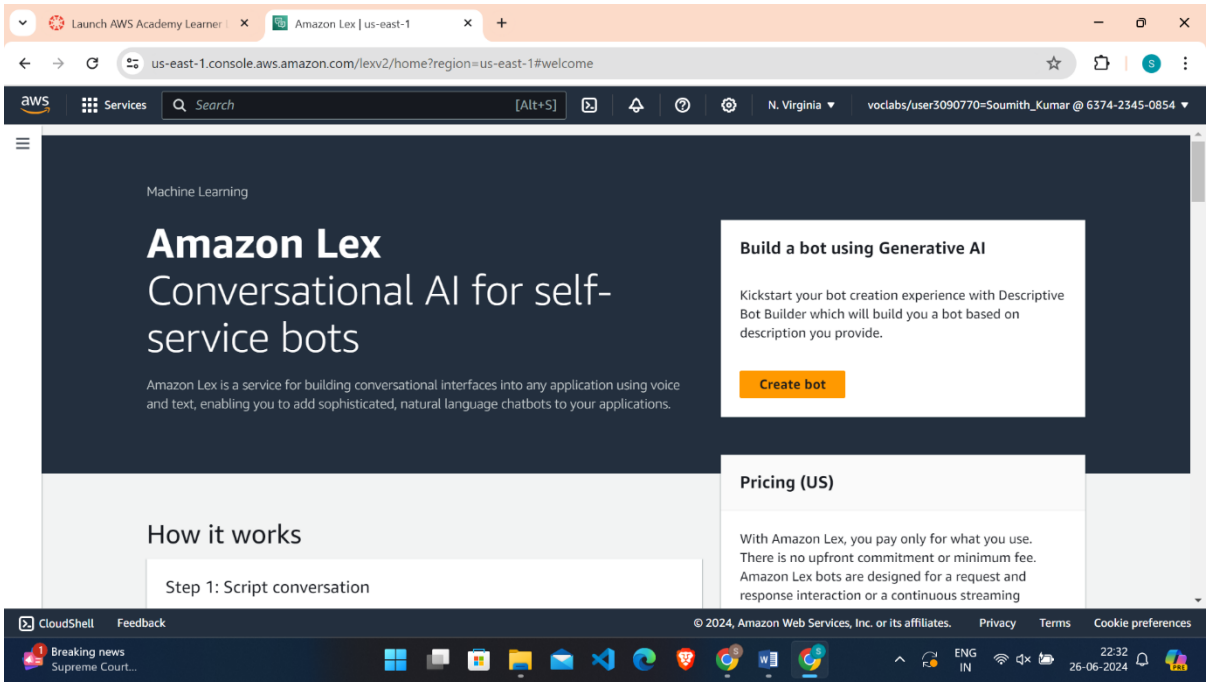




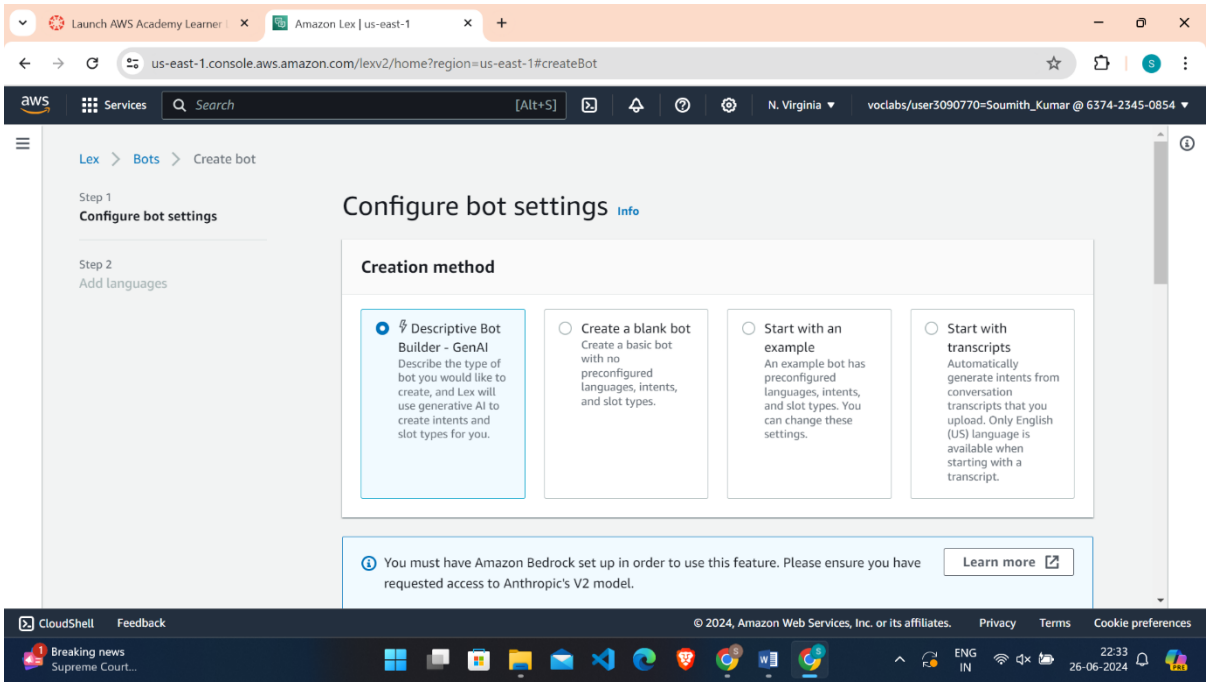
Experiment-7

Aim: Create a help desk assistant integrating Amazon LEX.

Step 1 : Navigate to Amazon Lex service via Amazon Console.

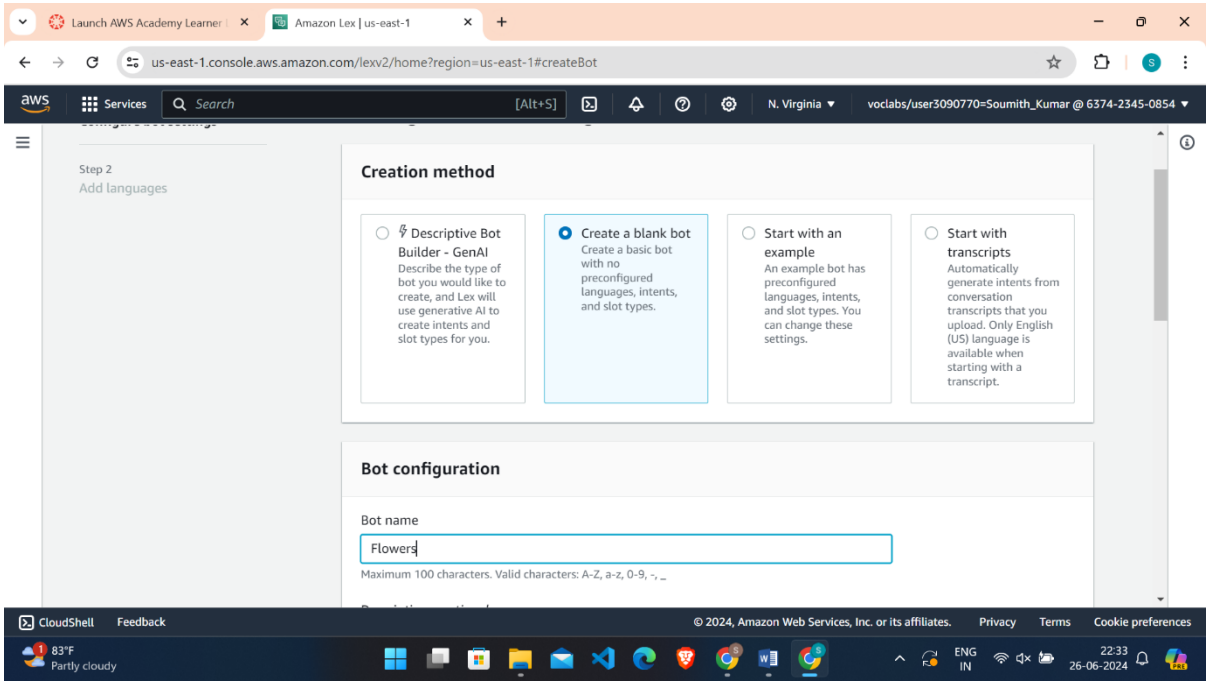


Step 2: Click on Create bot



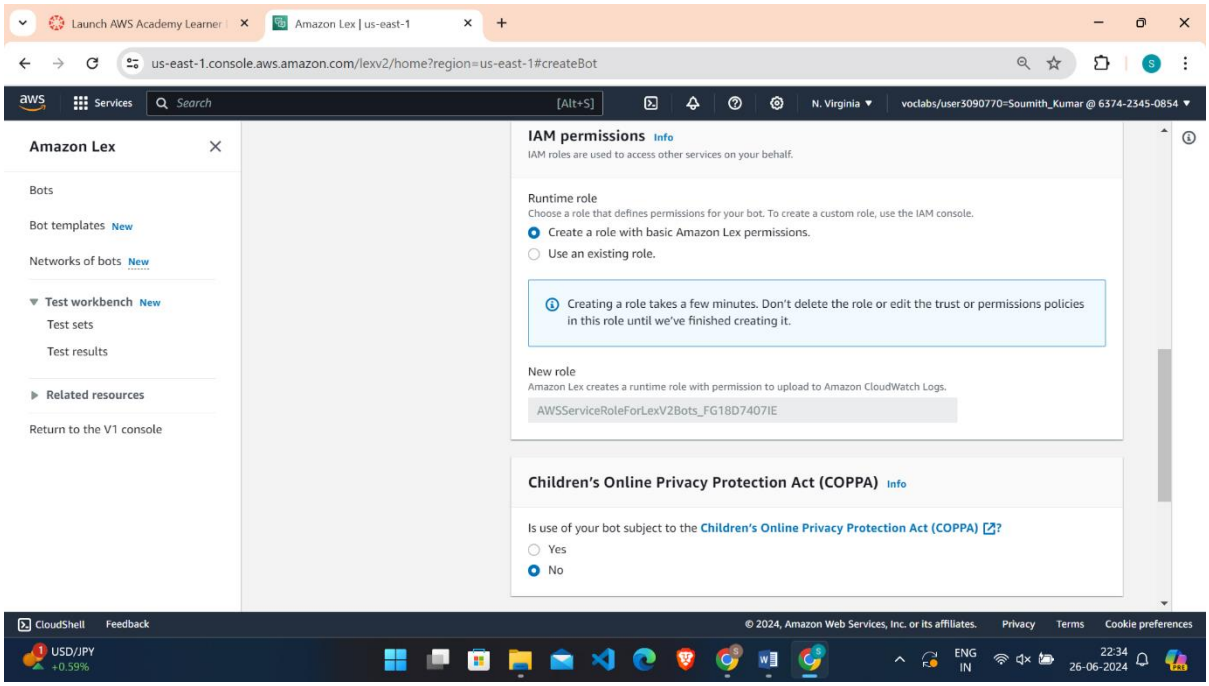
Step 3: Configure the bot settings

- Select Create a blank bot
- Give name for the bot.

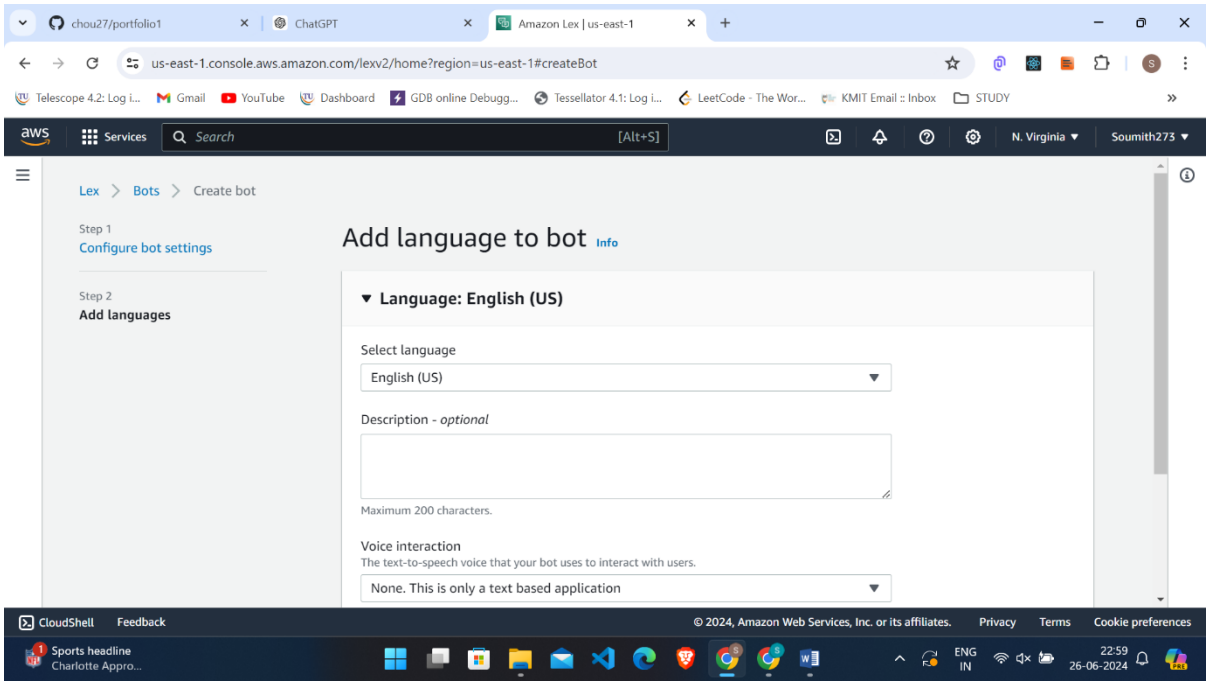


Step 4: In IAM permissions Choose create a role with basic Lex Permissions.

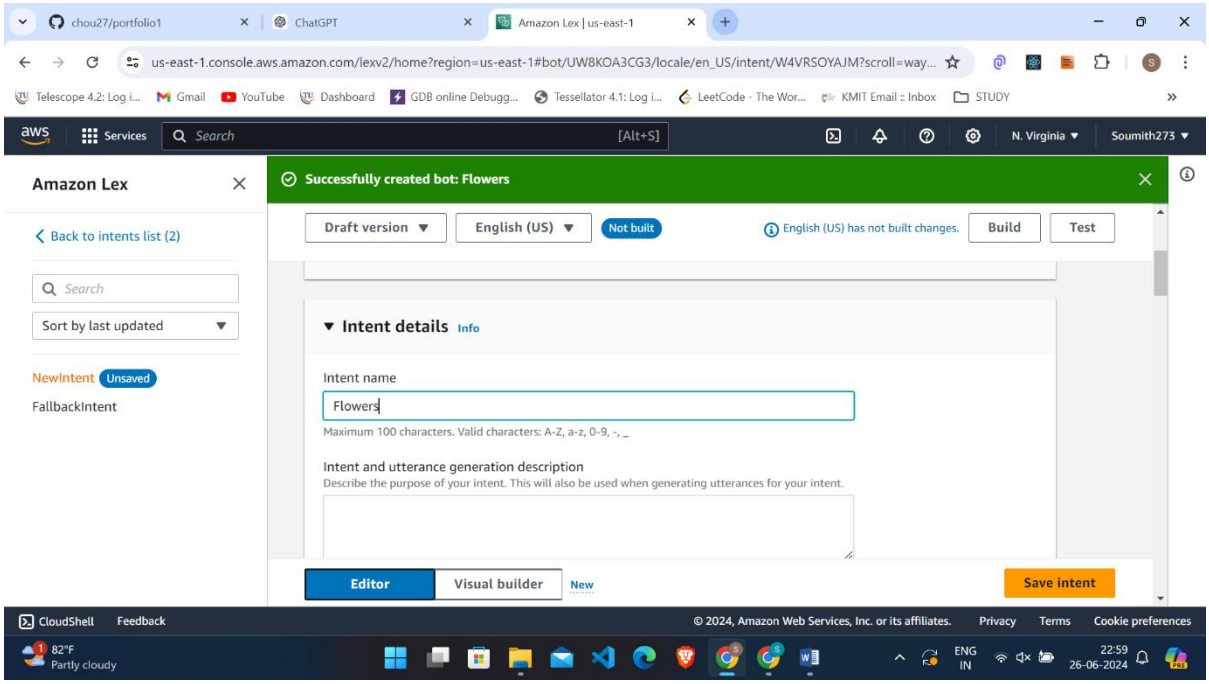
- Check No radio option in the COPPA field.
- Click on next.



Step 5: Click on Done

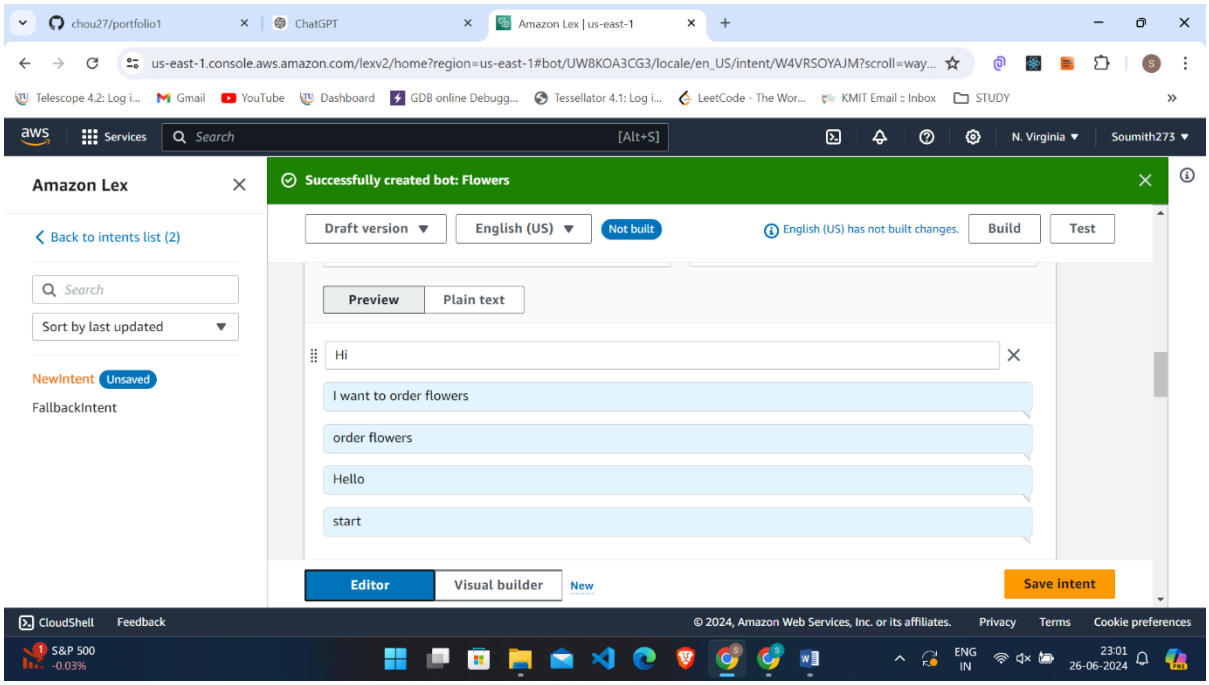


Step 4: Give Intent name as ‘Flowers’



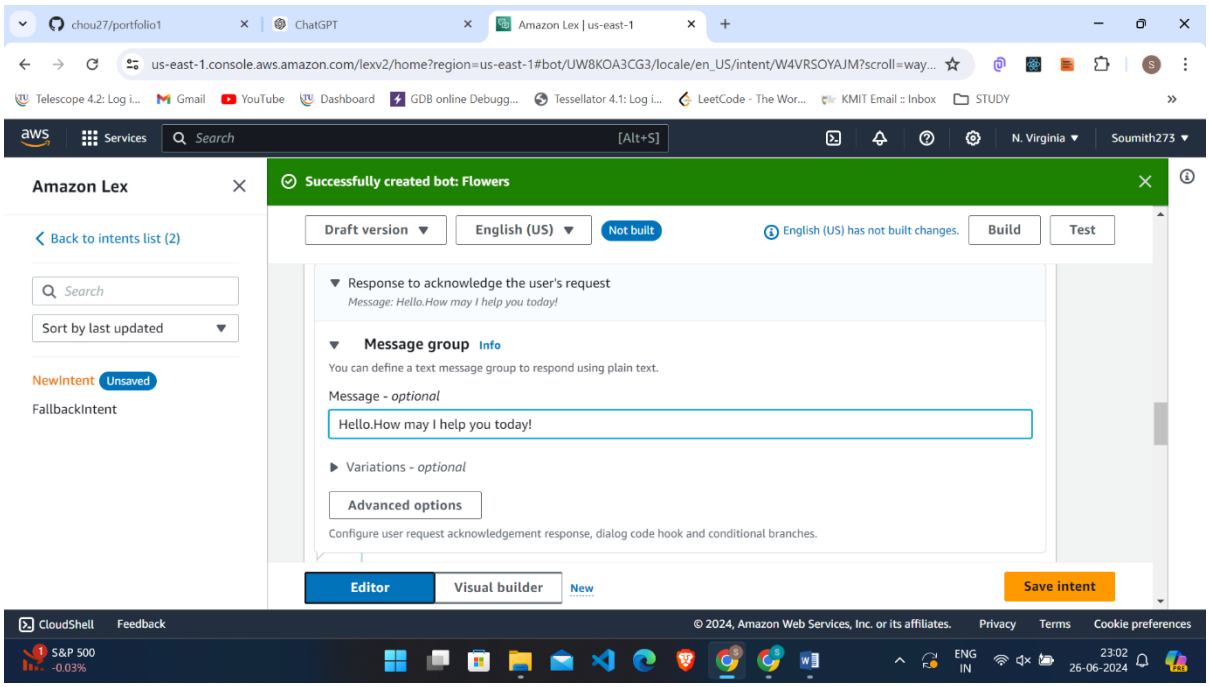
Step 5: Scroll down for Utterances

- Add some utterances like “Hi ”, ”Hello”, ”book order” etc. When these utterances listen by the bot these intent will be invoked.



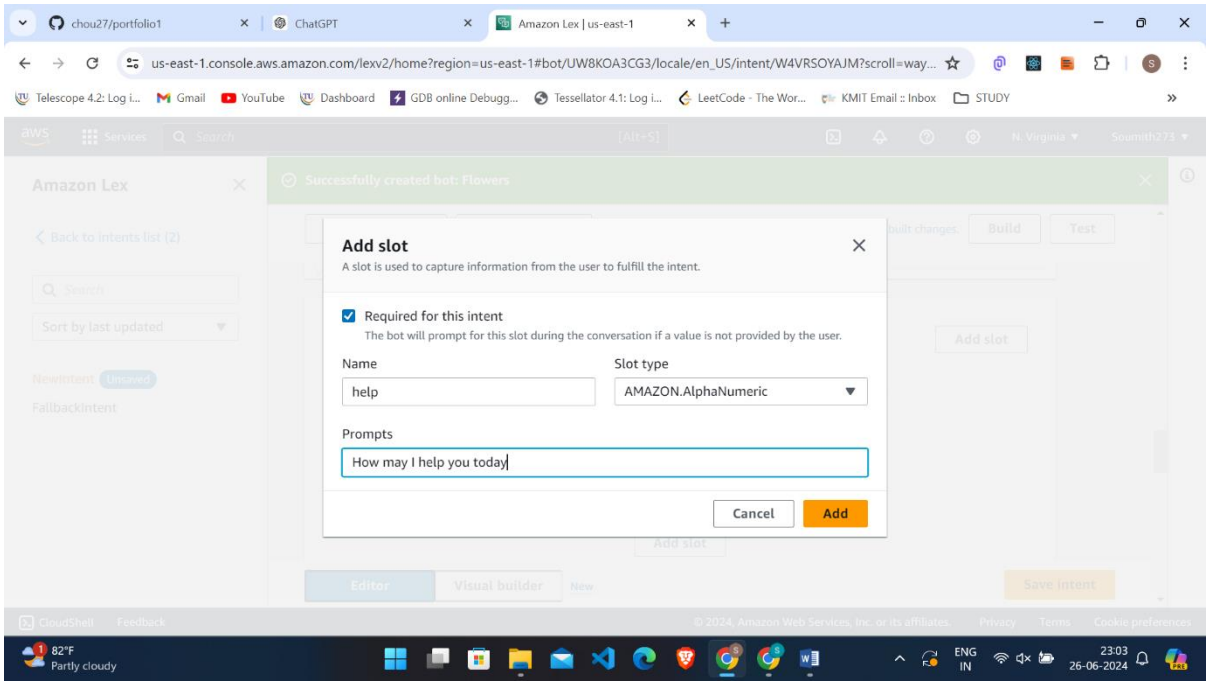
Step 6: Initial Response – when bot listen some input these message will be sent first . Generally , Initial response message is used to greet the user.

- In the message box we can write our initial response (“Hello, How can I help we today”).



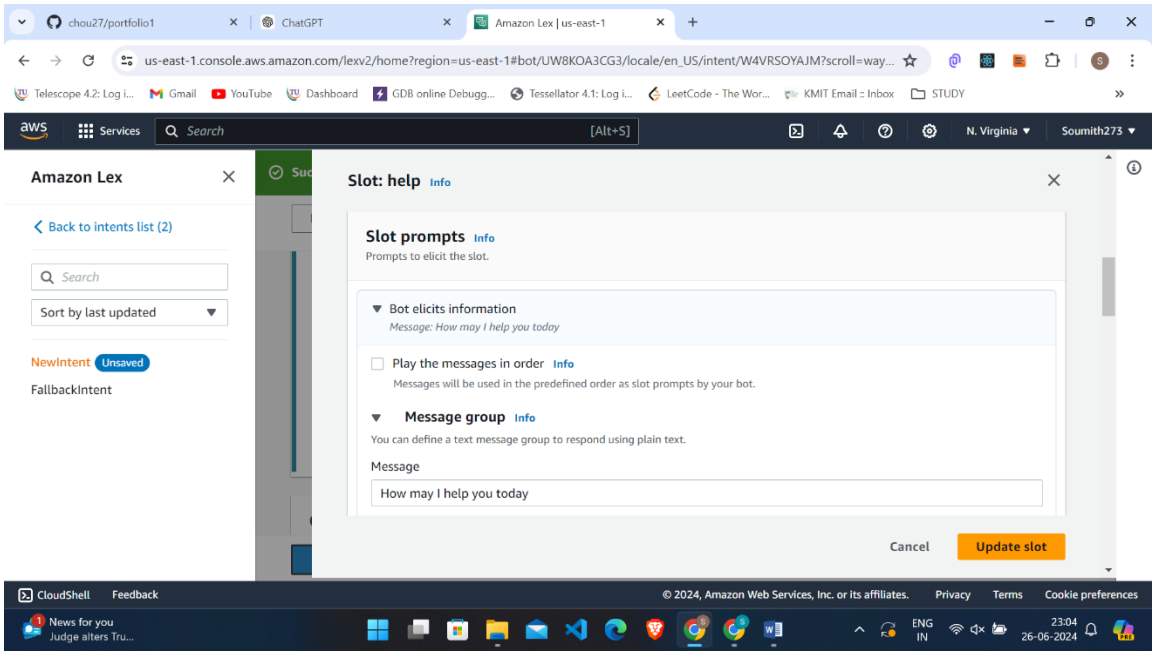
Step 7: Slots - Slots are values provided by the user to fulfill the intent.

- Click on “add slot”
- Give the name of the slot . eg: name ,flower,wishes,etc.
- Select the slot type . there are several types like alphanumeric, date, city .
- Give prompt like how bot should respond .

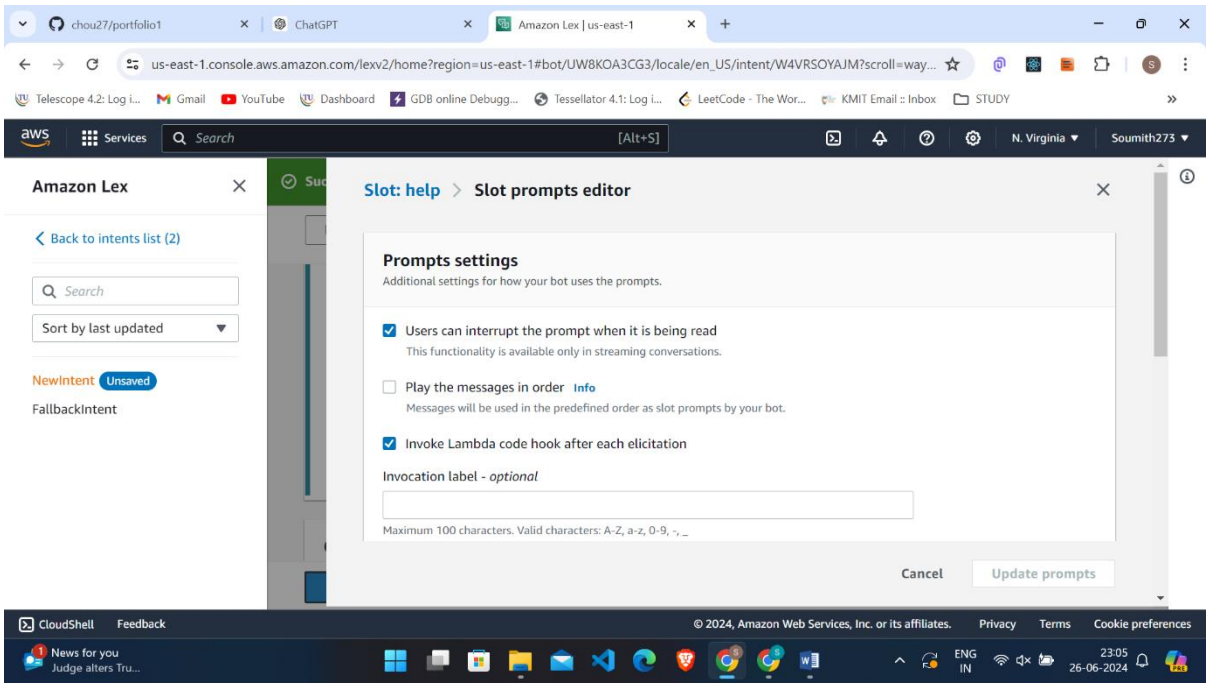


Step 8: To add cards .

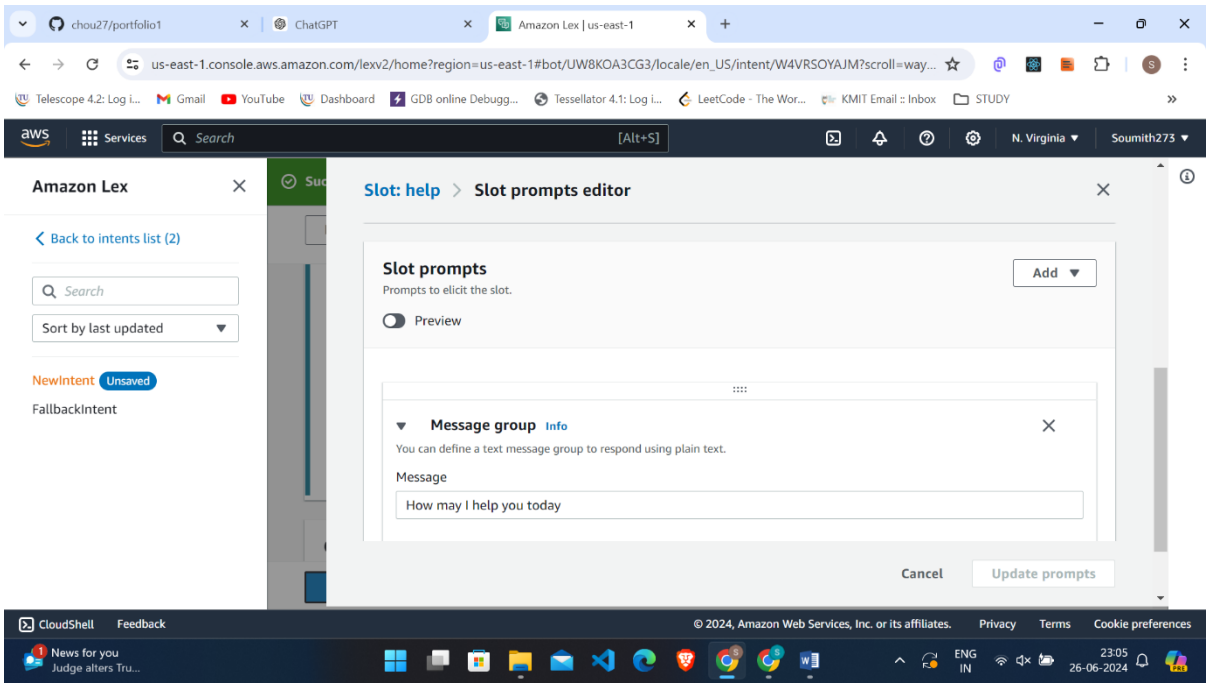
- Select the slot
- Click on Advanced Options
- Under “Slot Prompts” , click on Bot elicits information .



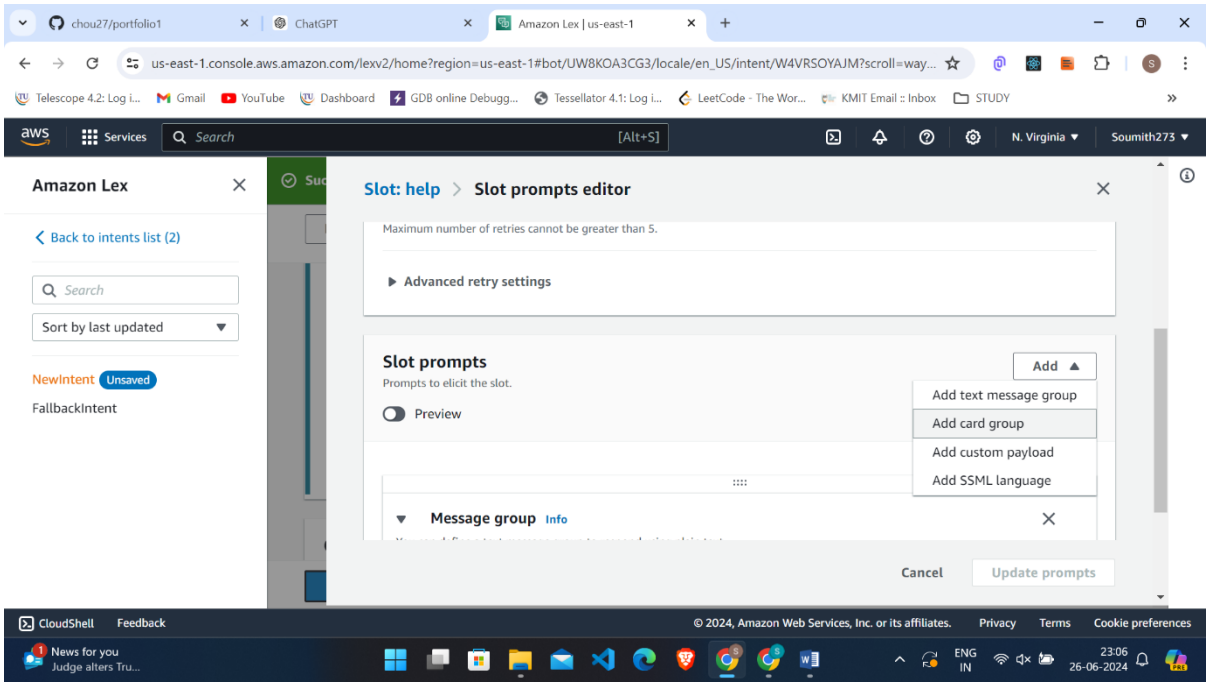
- Click on More prompt options.



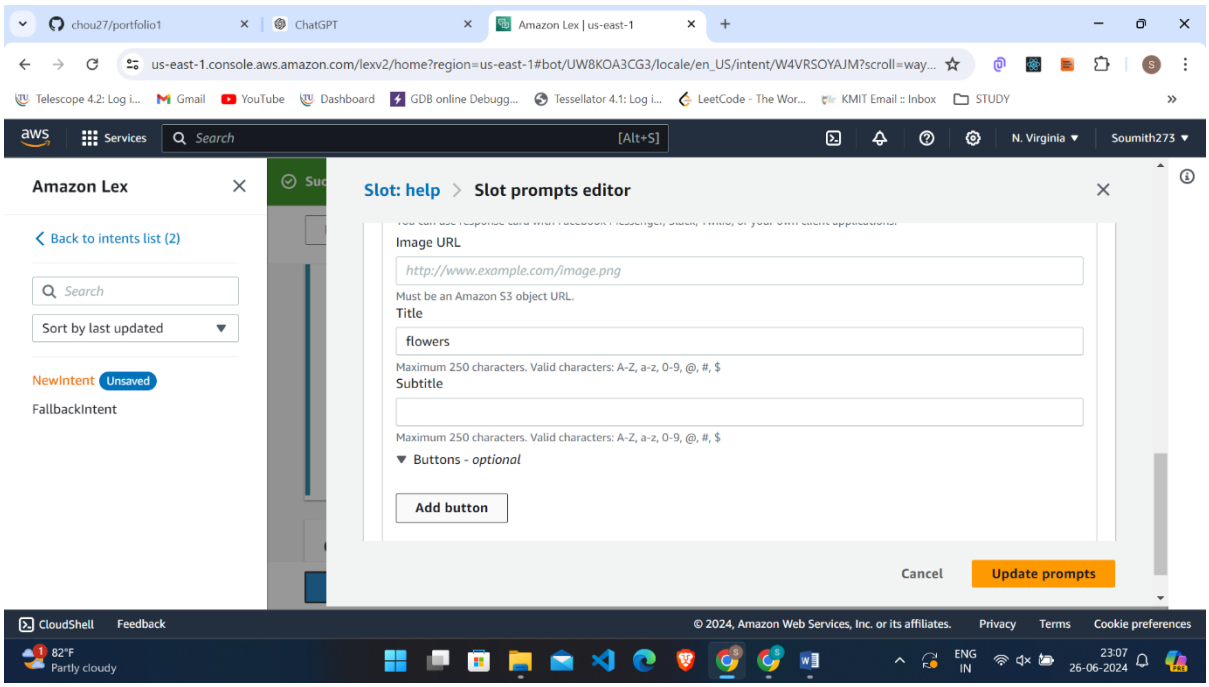
- Scroll down for slot prompts and click on add .



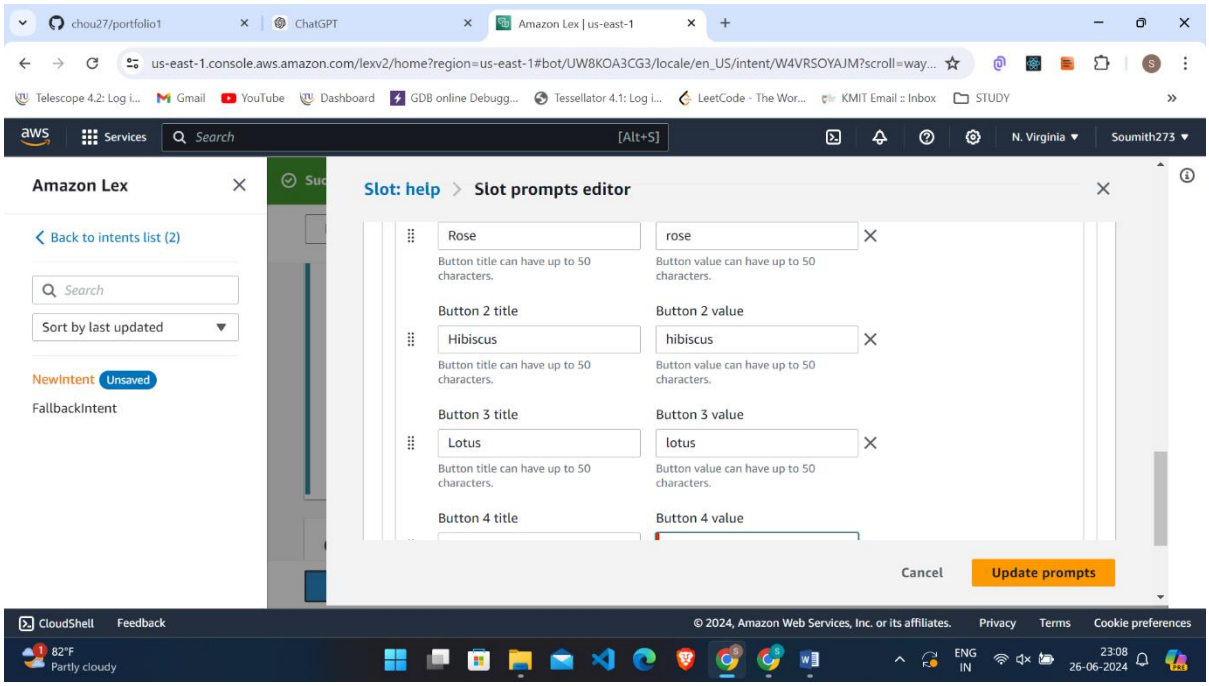
- Select “Add card group” .



- Give the title such as “flowers”
- Click on Buttons
- Add Button

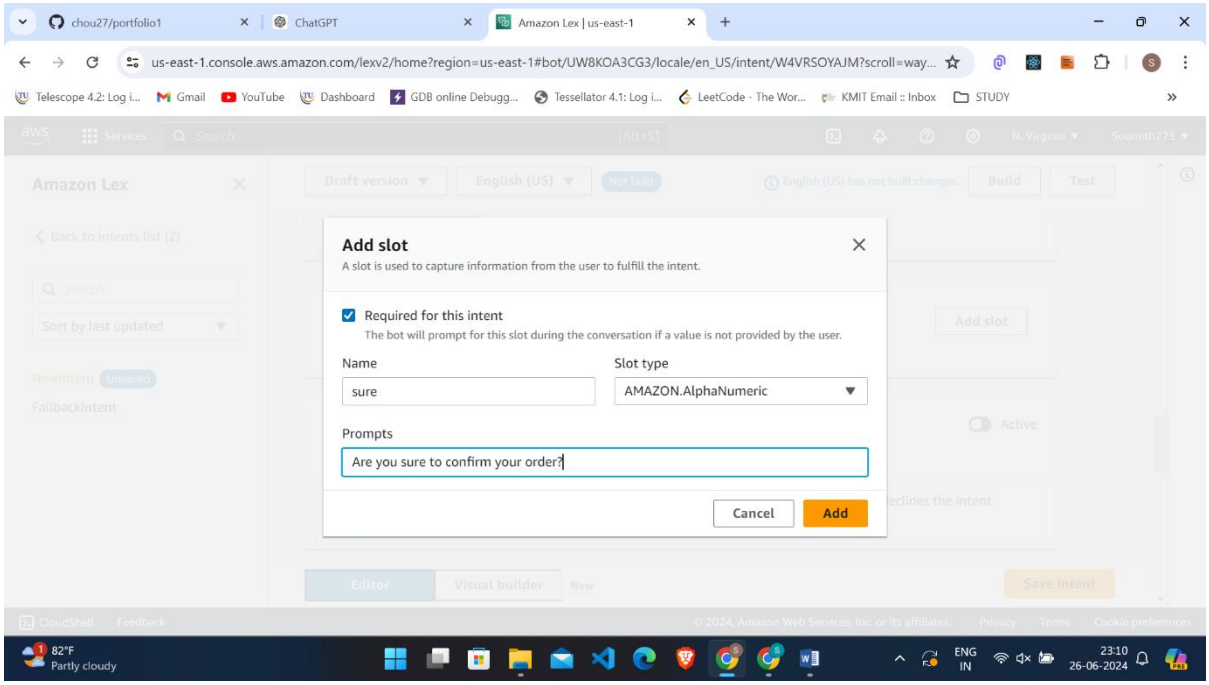


- Add buttons . For each Button , give Button title(which will displayed in the card) and Button value(returned when we select that button).
- Click on update prompts and then update slot.

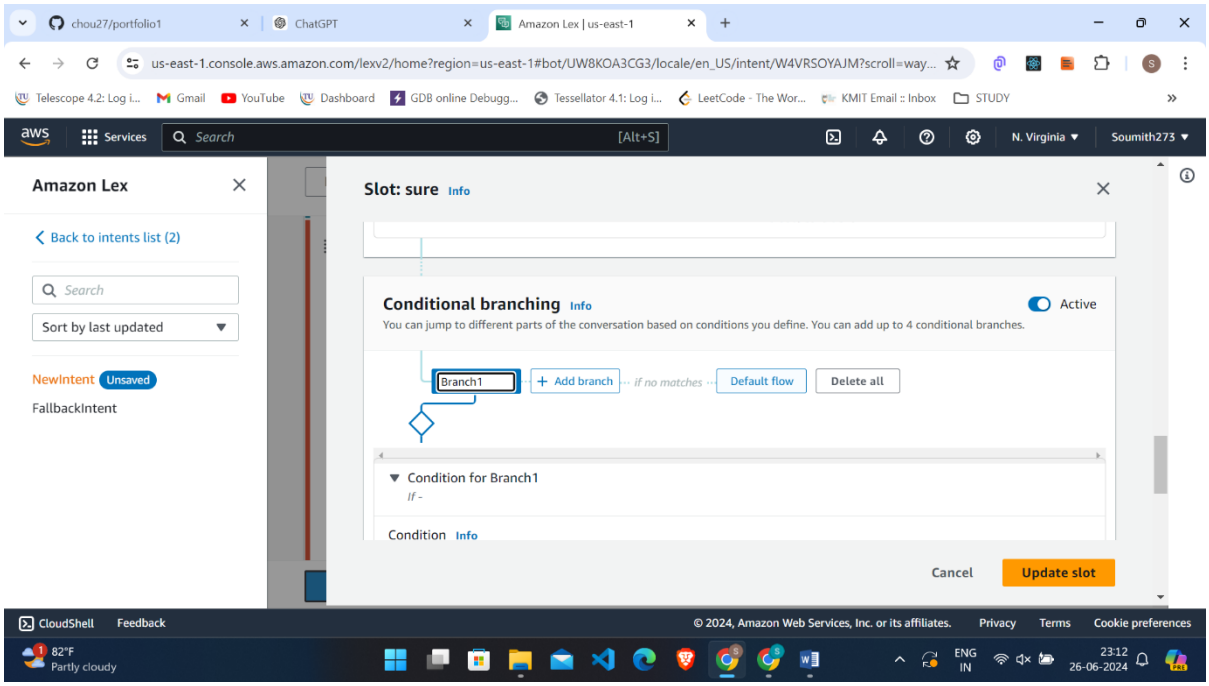


Step 9: To add conditional branching to slot.

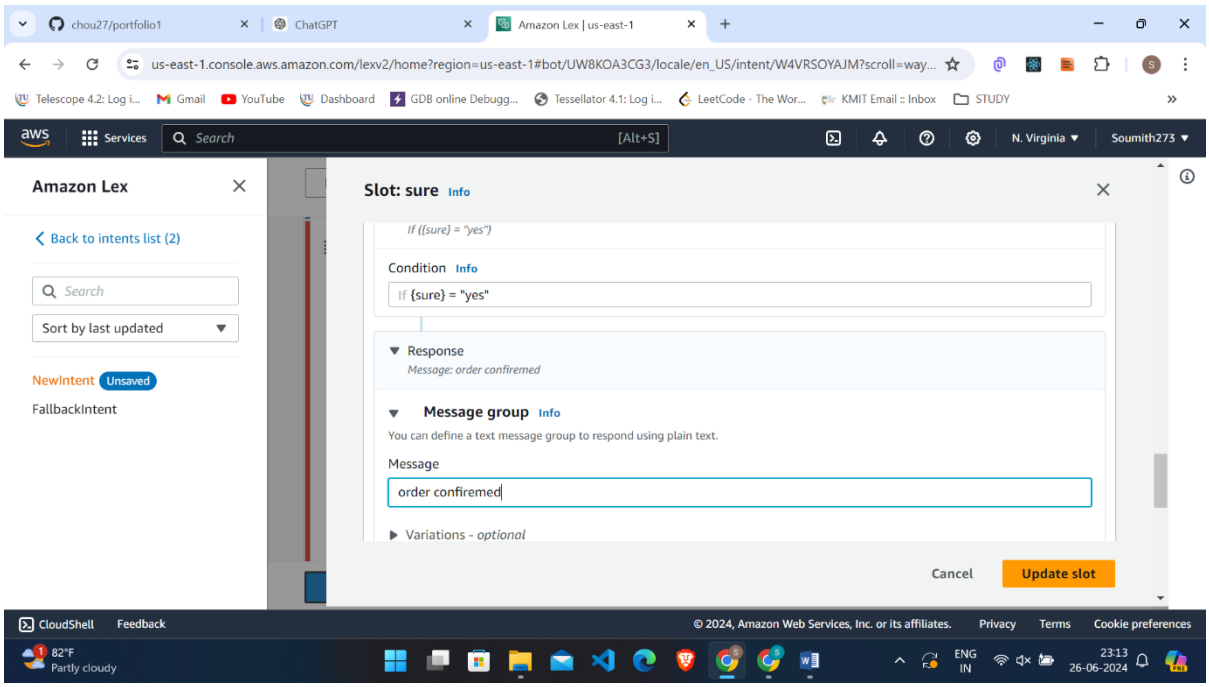
- Create a slot .



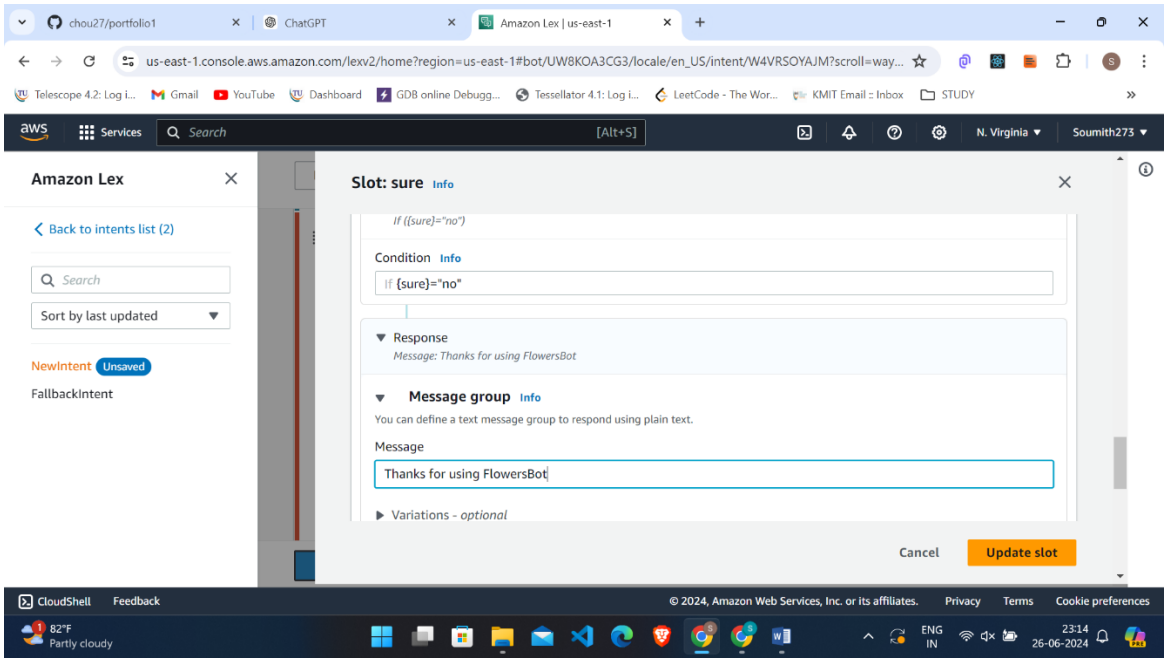
- Click on Advanced options
- Goto slot captures : Success response
- Click on add Conditional branching



- Under Branch1 , write condition for the branch1 and write Message we want to show if success. In my condition , if sure equals yes order confirmation message will be displayed.

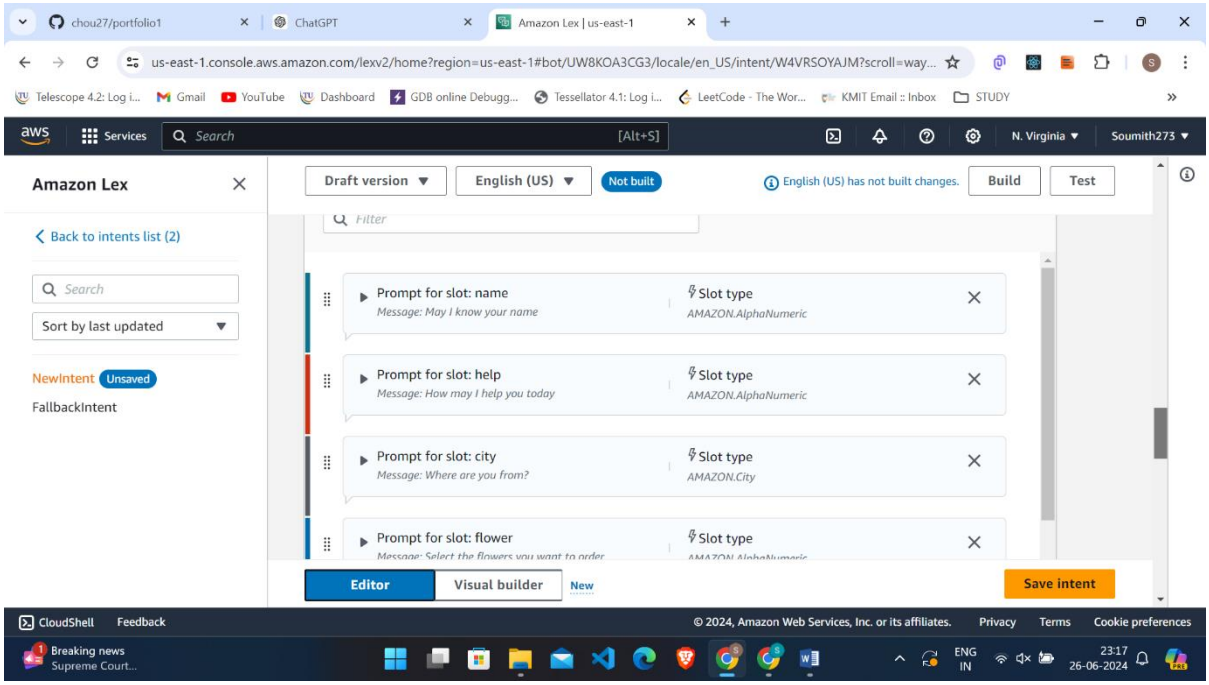


- We can add branches according to usage .
- In my case I added second branch with condition if sure equals no then thanks for using bot

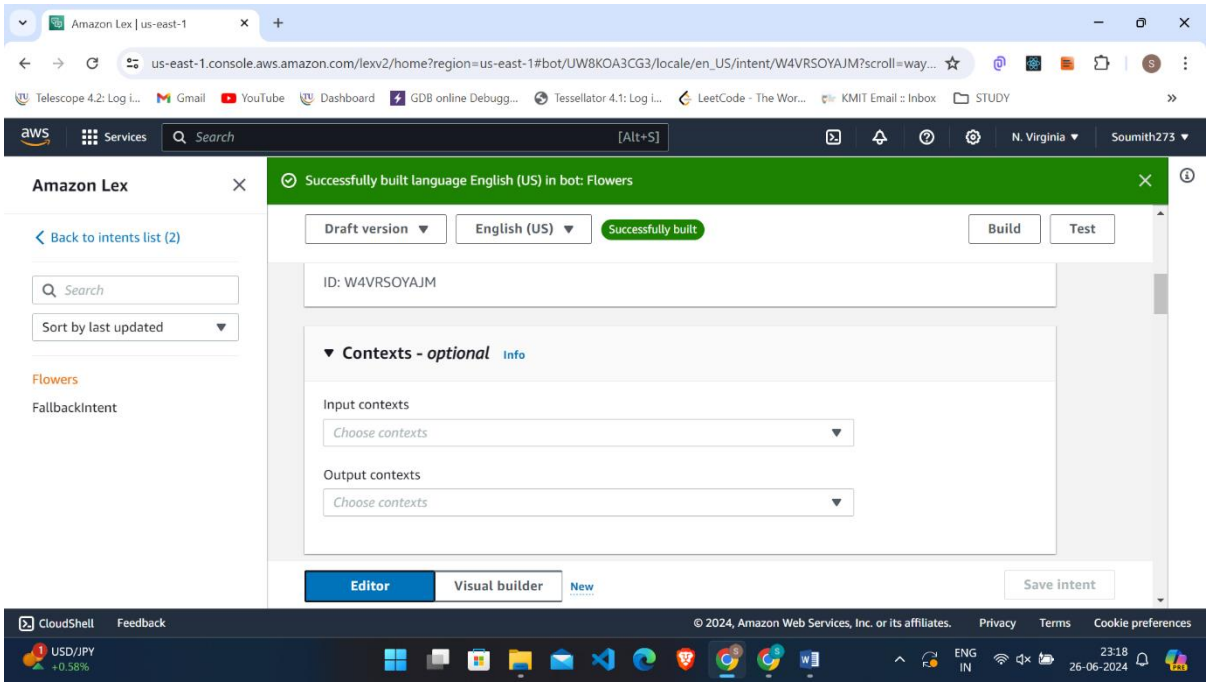


Step 10: In same way , add more slots as many as we needed .

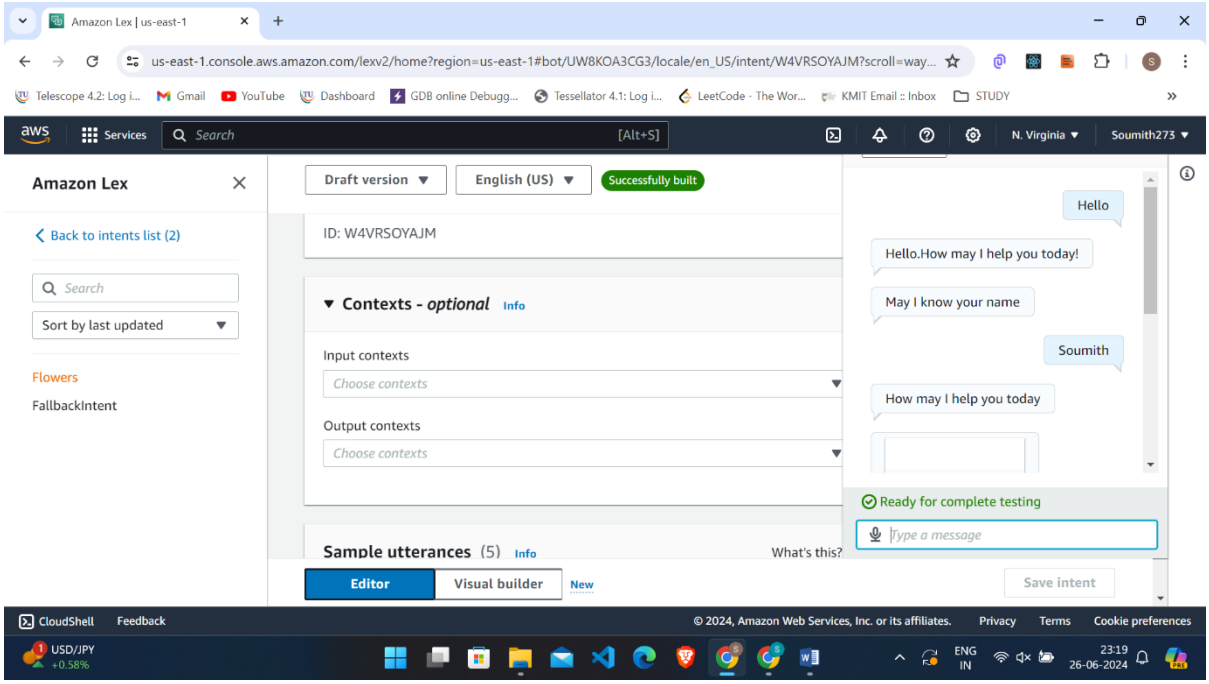
- After adding all slots , Click on Save Intent



Step 11:Click on Build and wait until the bot is build .



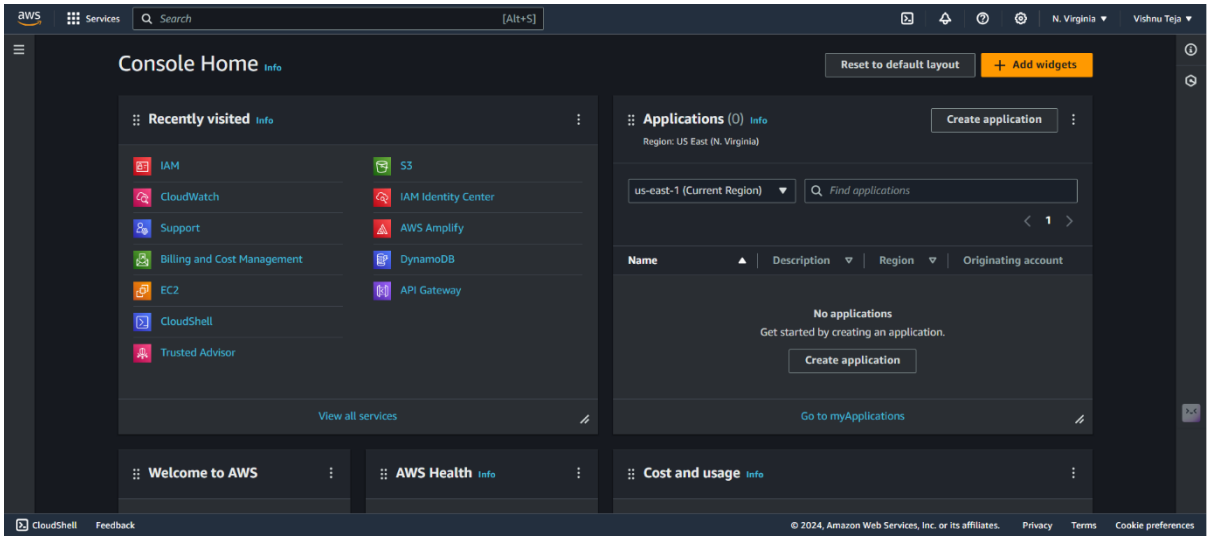
Step 12: Click on Test and check our bot.



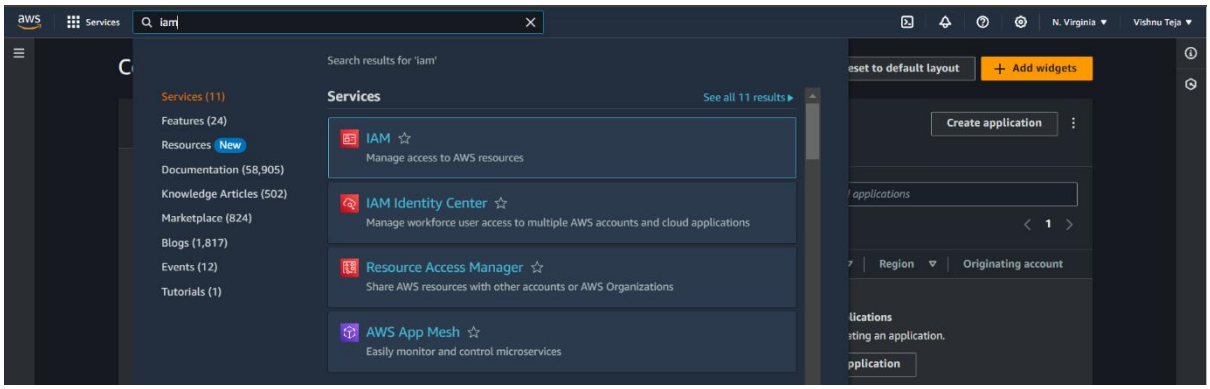
Experiment – 8

Aim: Create an AWS IAM User with attached policy and also Implement a role based access between AWS services as well as Create an group with attached policies

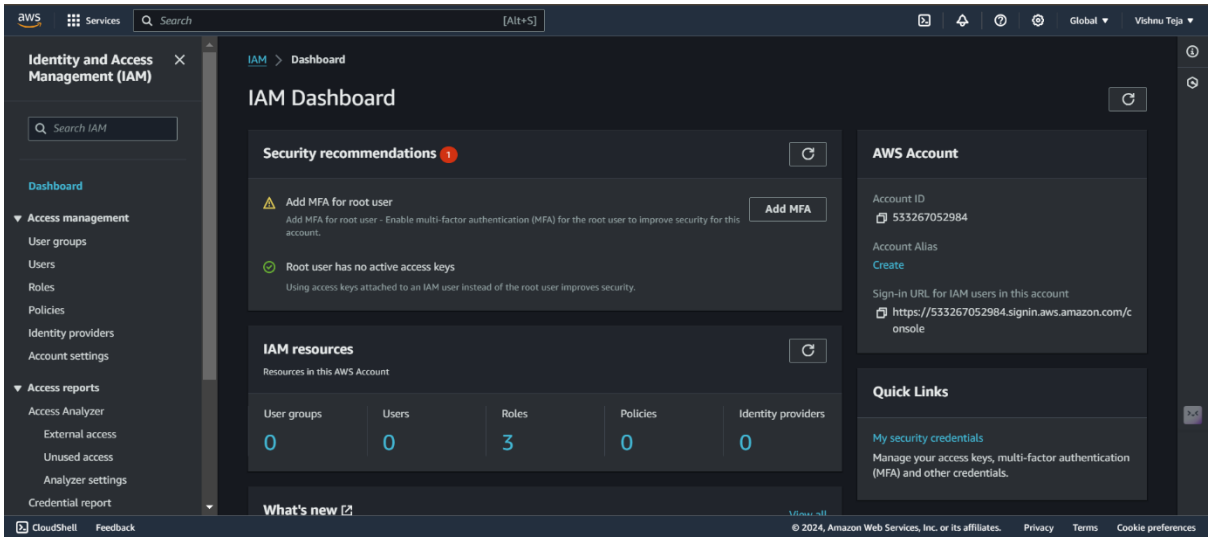
Step 1: Login to AWS Free Tier Account with the Login Credentials.



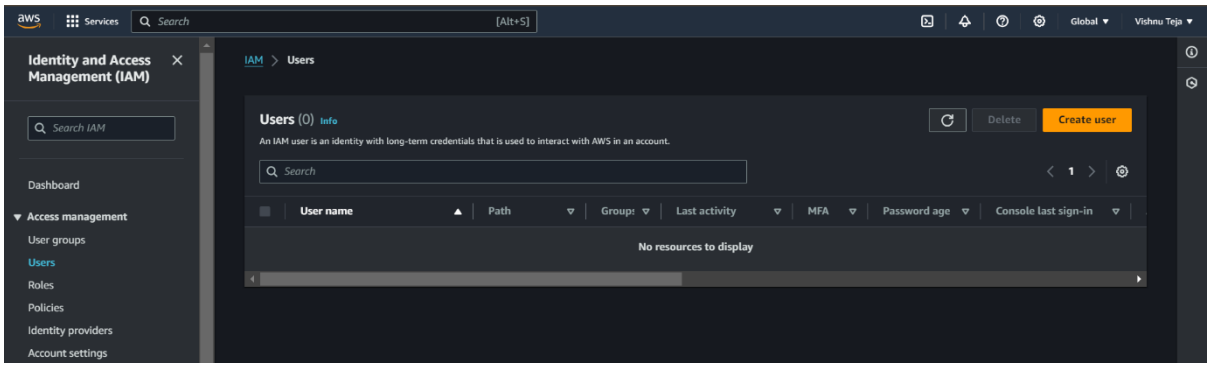
Step 2: Navigate to the Service IAM through the search bar.



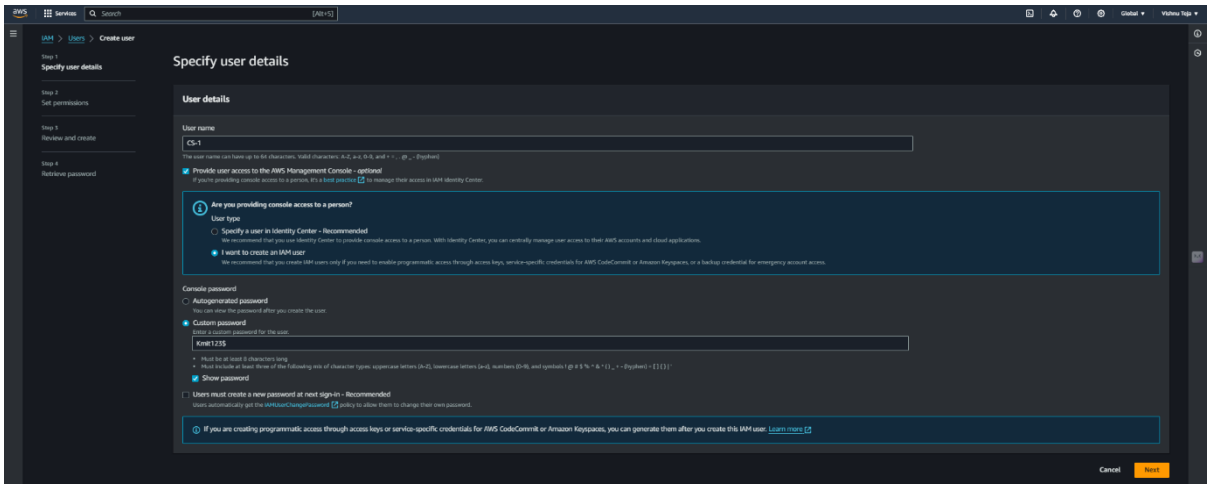
Step 3: In the IAM Dashboard, Left pane Search for Users and Navigate to the user dashboard.



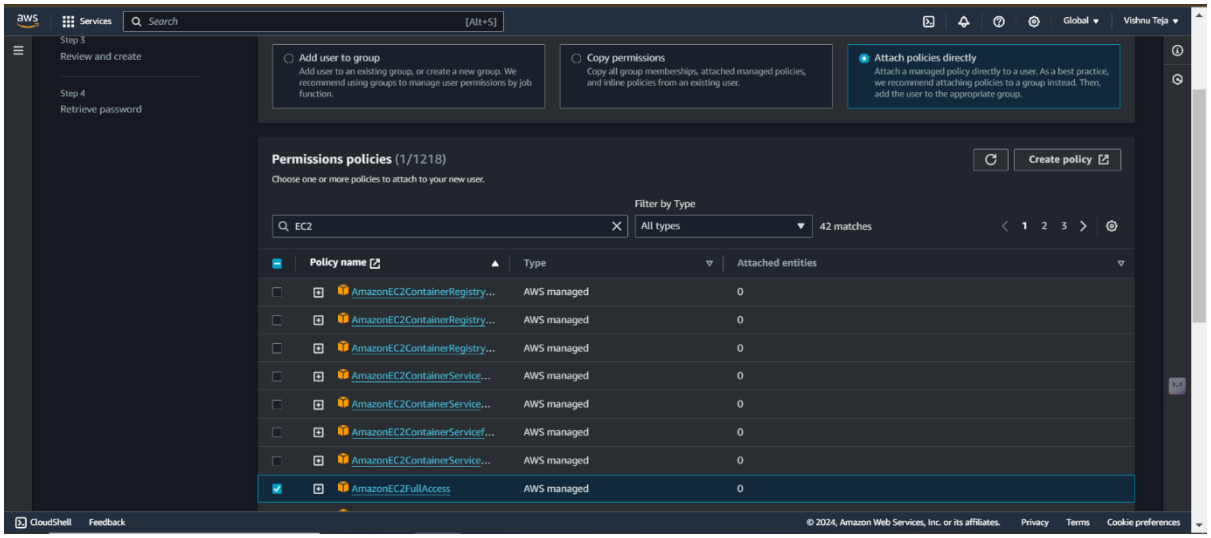
Step 4: To create a user with the IAM policy Click on Create User.



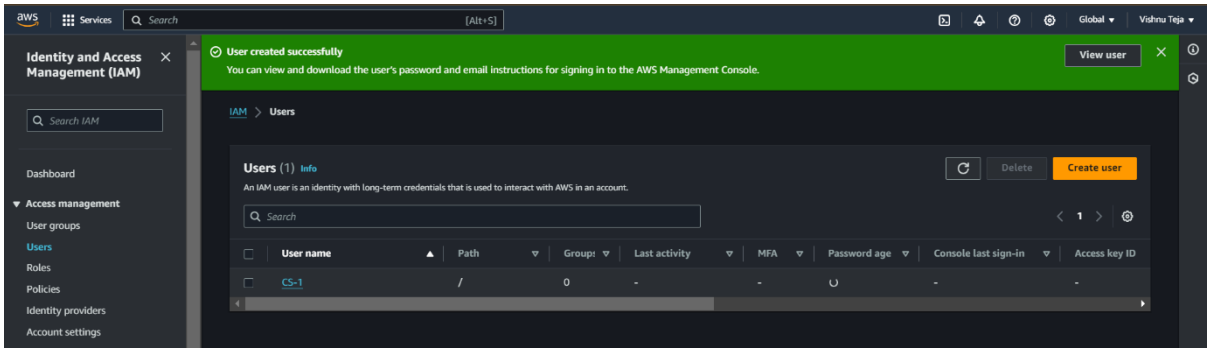
Step 5: Specify the User Name, Provide the access to Management console and create a Custom password and click on next.



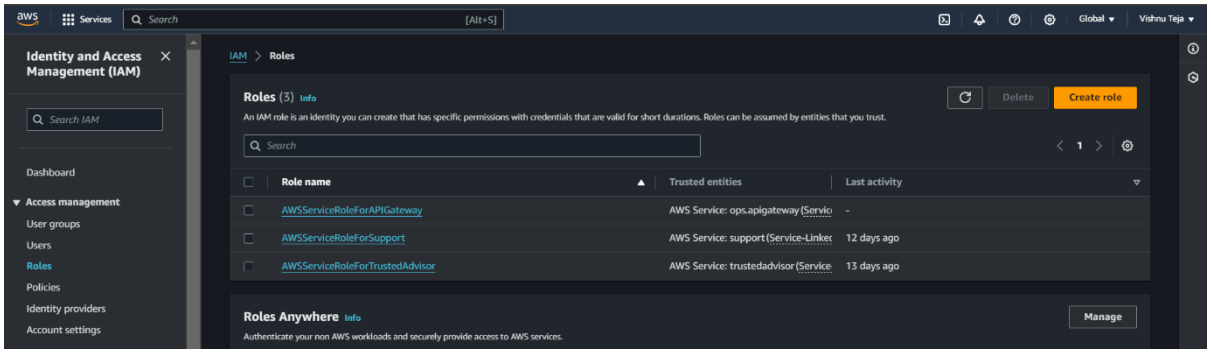
Step 6: Select Attach Policies directly, and search for EC2fullAccess policy and click on next.



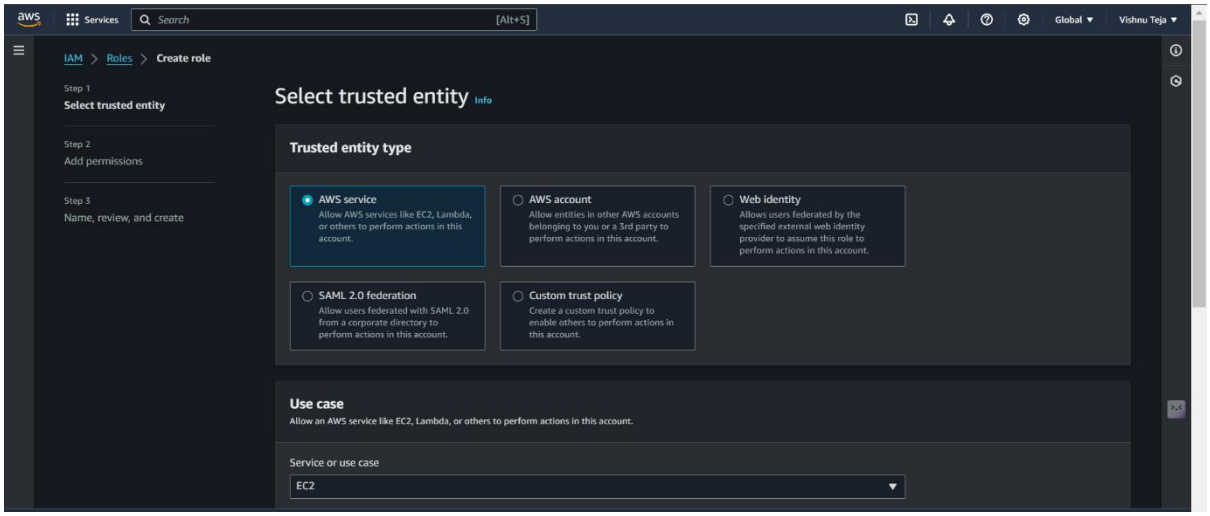
Step 7: Review the policy and create user. Now we have create a User with attach policy



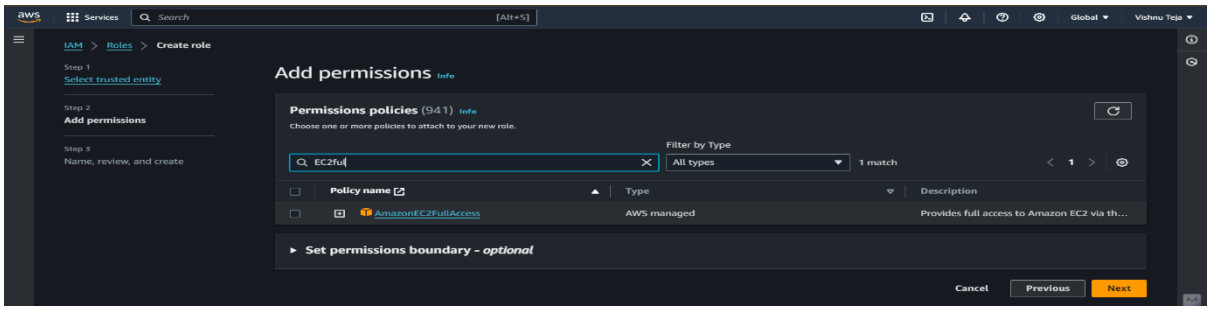
Step 8: Click on Roles from the left pane, and click create role.



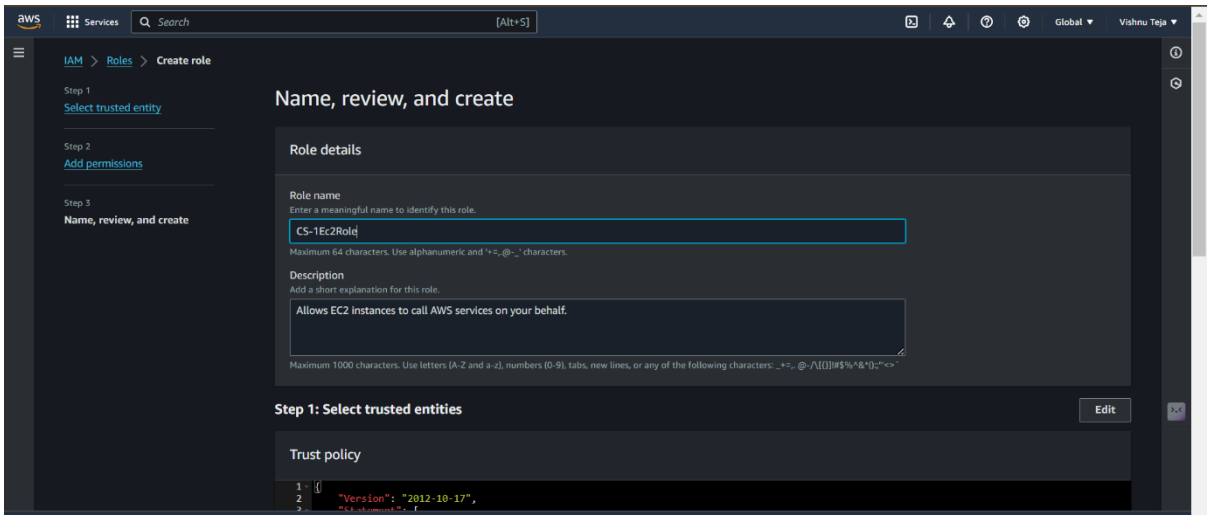
Step 9: Select AWS Service, From the list of use case, select EC2 and click on next.



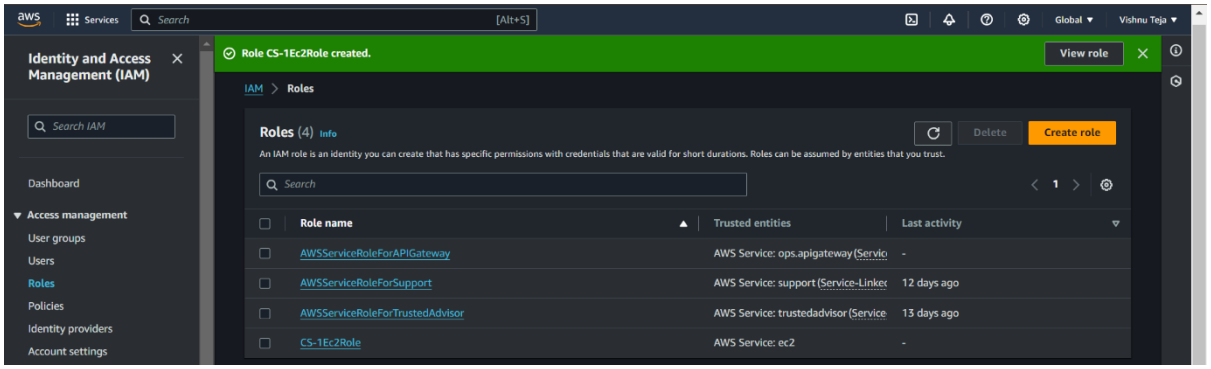
Step 10: Search for the permission EC2FullAccess, Assign it and click on next.



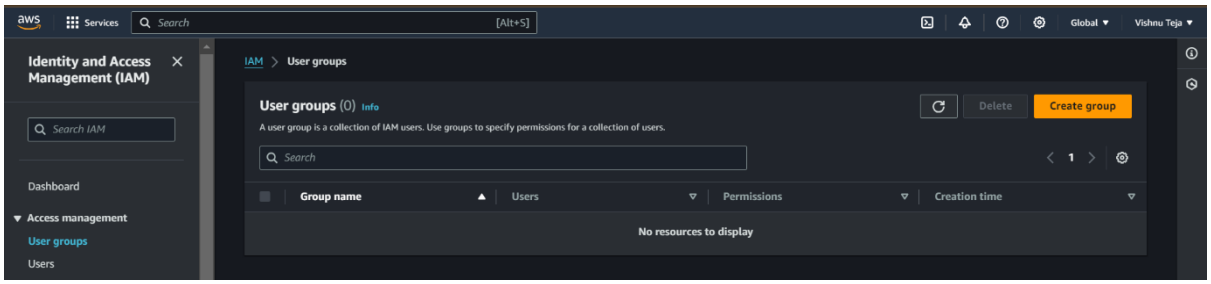
Step 11: Give a Name to the role and review the policy and click on create role.



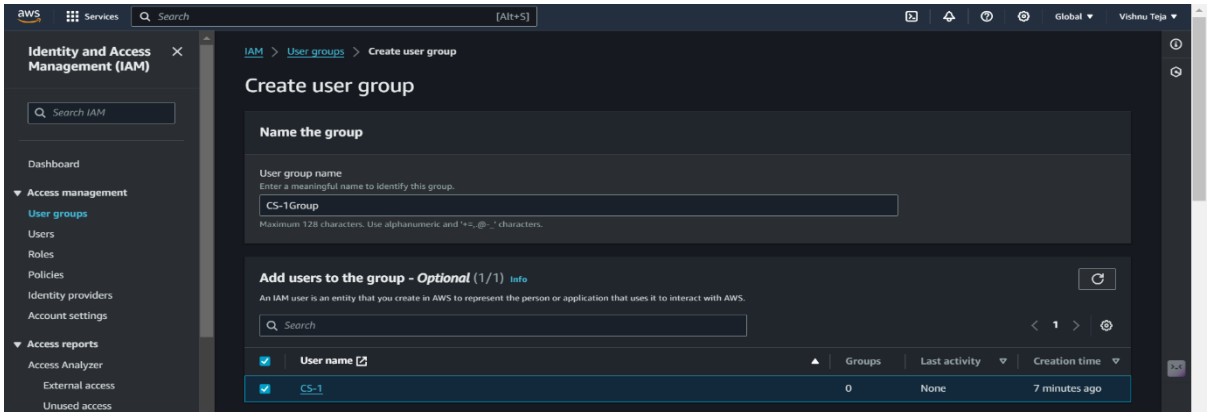
Step 12: Now successfully we have created a role based access between AWS service.



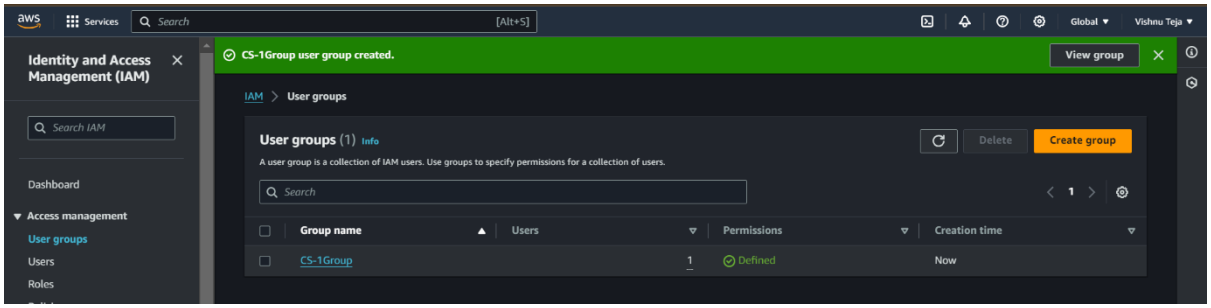
Step 13: To create user group, navigate to the user group dashboard and select create group.



Step 14: Give a name to the Group and select the users from the list to add to the group.



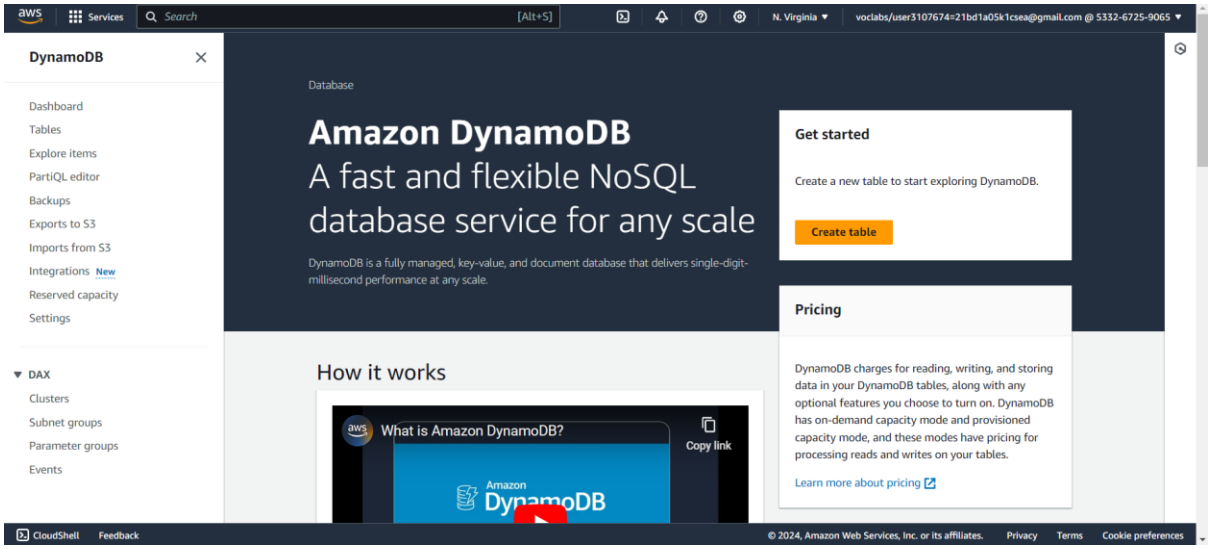
Step 15: Give necessary permissions to the group and click on create group. Now successfully we have created a user group and added a user to that group.



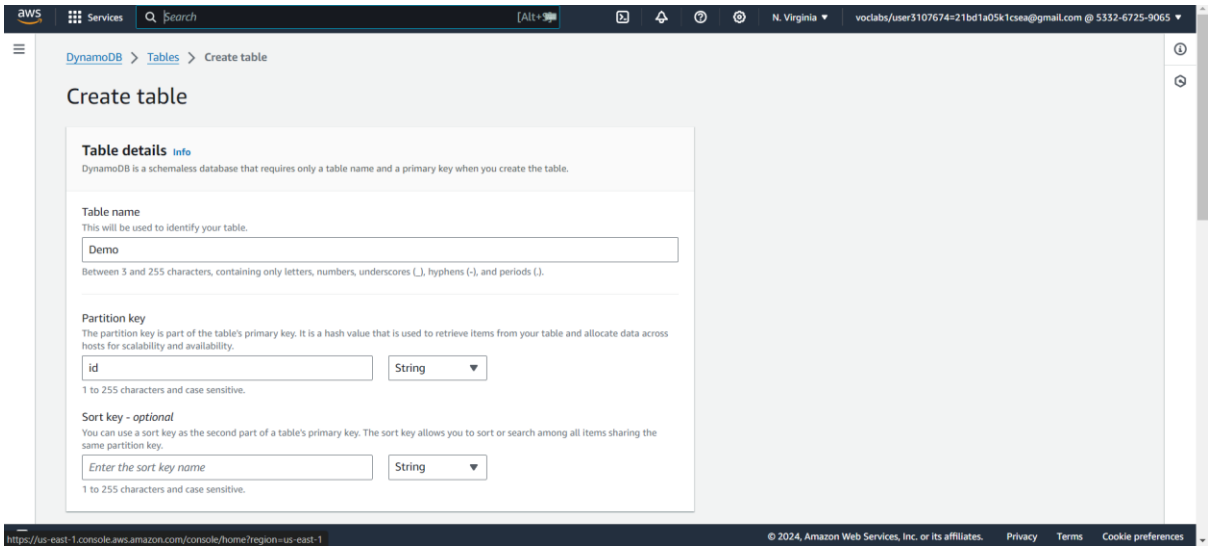
Experiment-9

Aim: Launch a NoSQL database using Amazon DynamoDB.

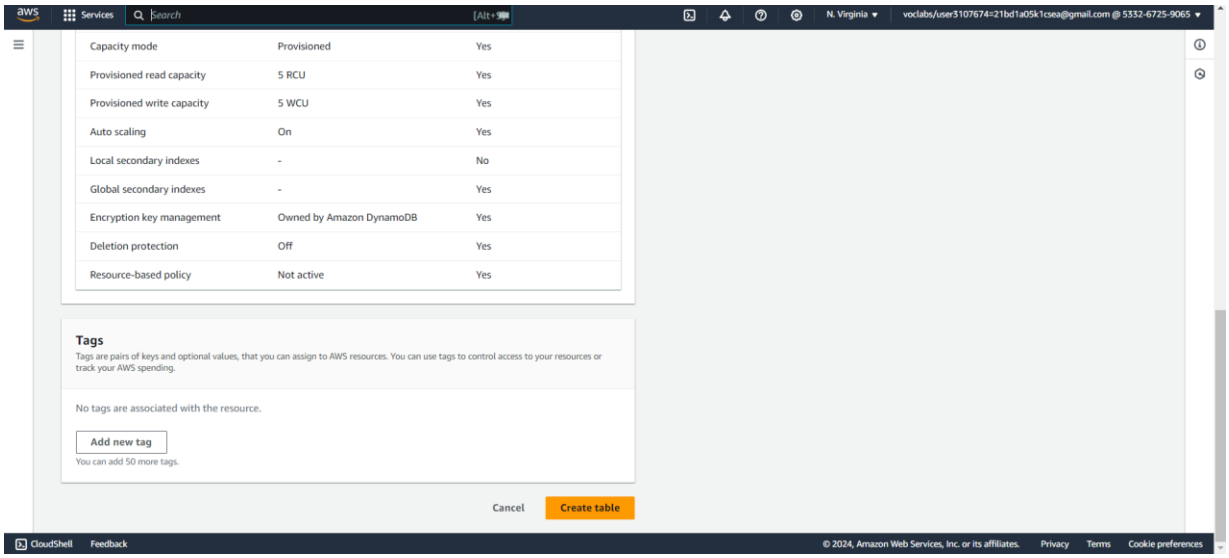
Step 1: Navigate to Dynamo DB Service via AWS Management console and Click on Create Table.



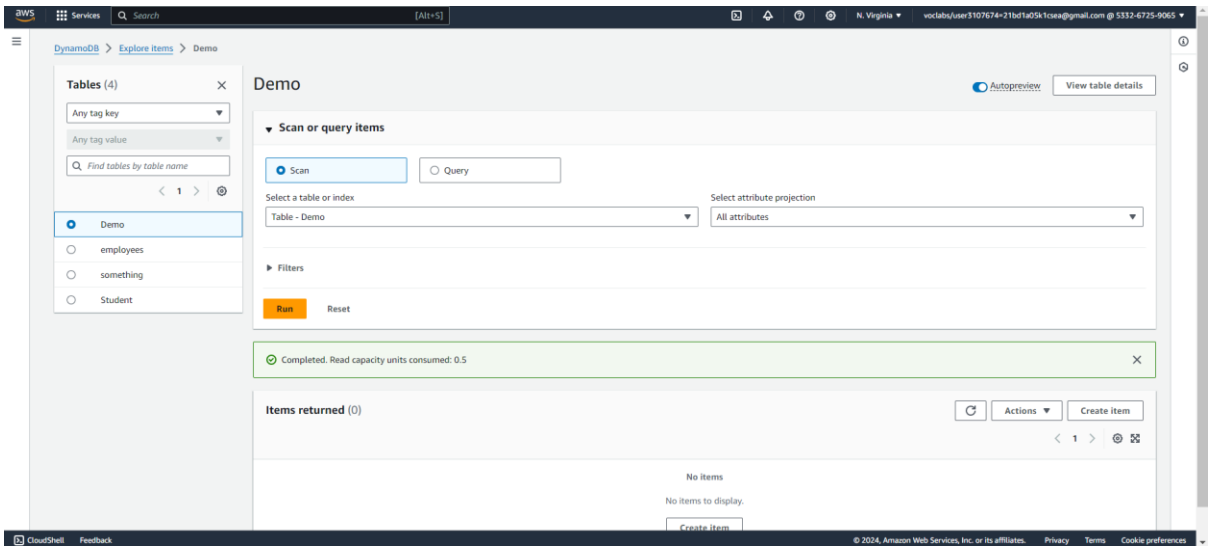
Step 2: Enter the name of the table and enter the partition key which is essentially the primary key of the table which is unique.



Step 3: Leave the other options as default and proceed to Click on Create Table .

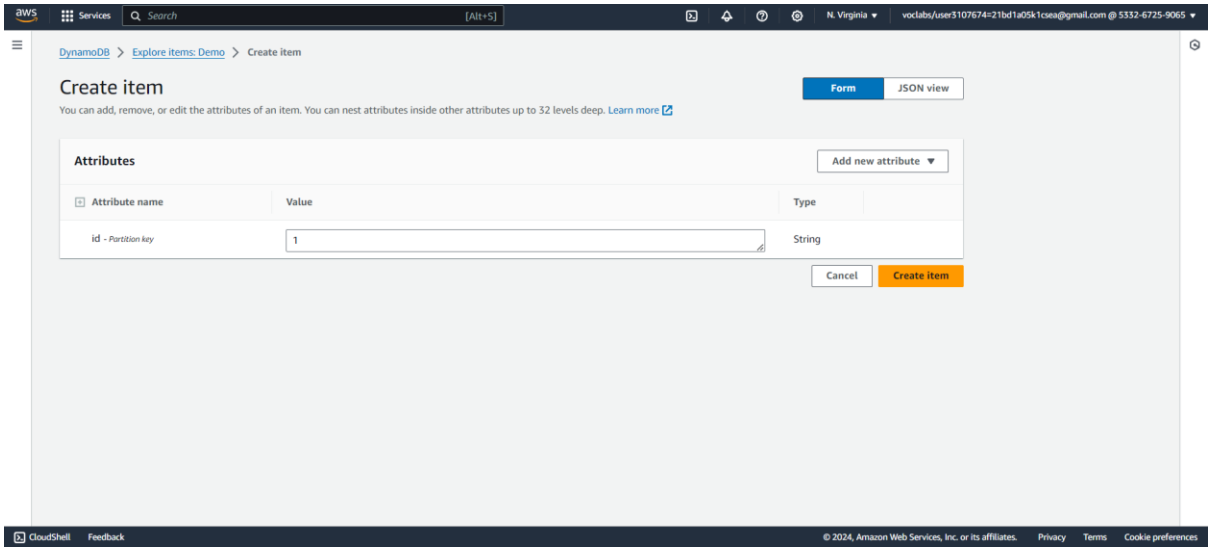


Step 4: Once the table is created and running select the created table from the dashboard and click on Explore items .



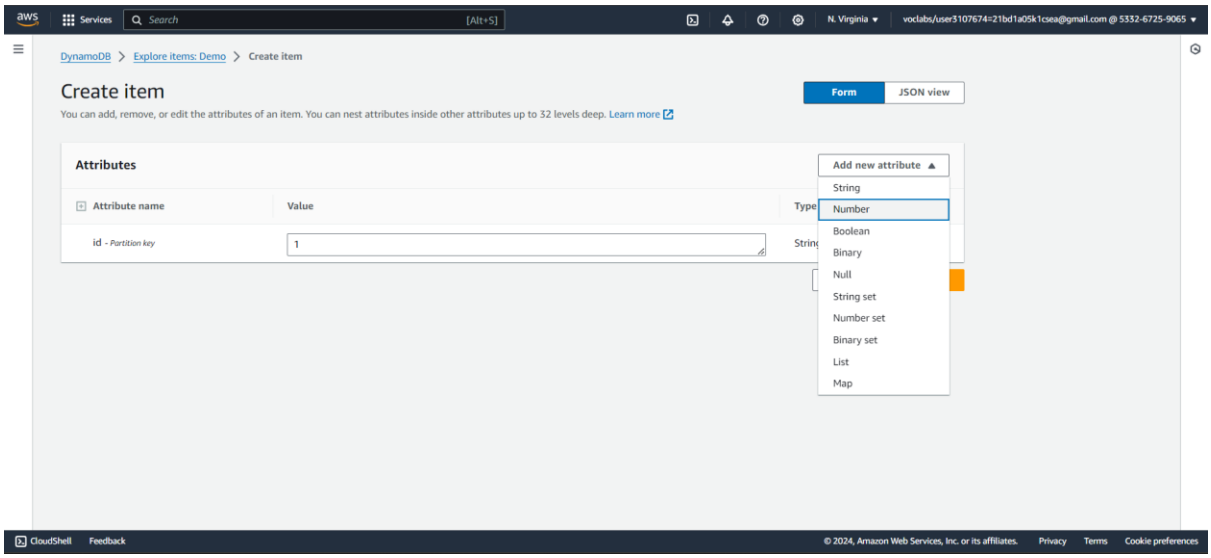
We can see at present there are no items in the table.

Step 5: To create items in the table click on Create item and enter some value for the partition key.

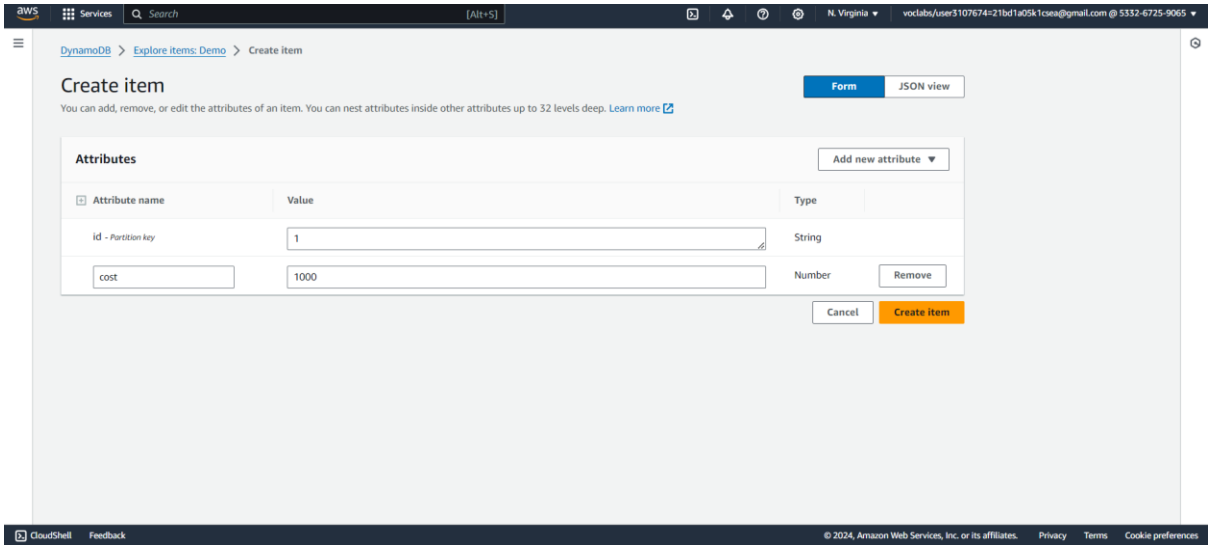


We can also add other fields/Attributes in the table .

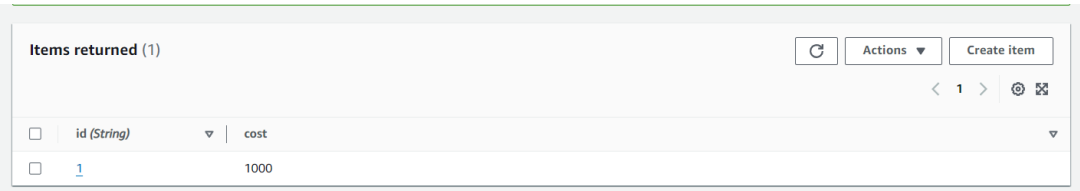
Step 6: To add Additional Attributes in the table click on Add new attribute drop-box and select the data type of the new attribute.



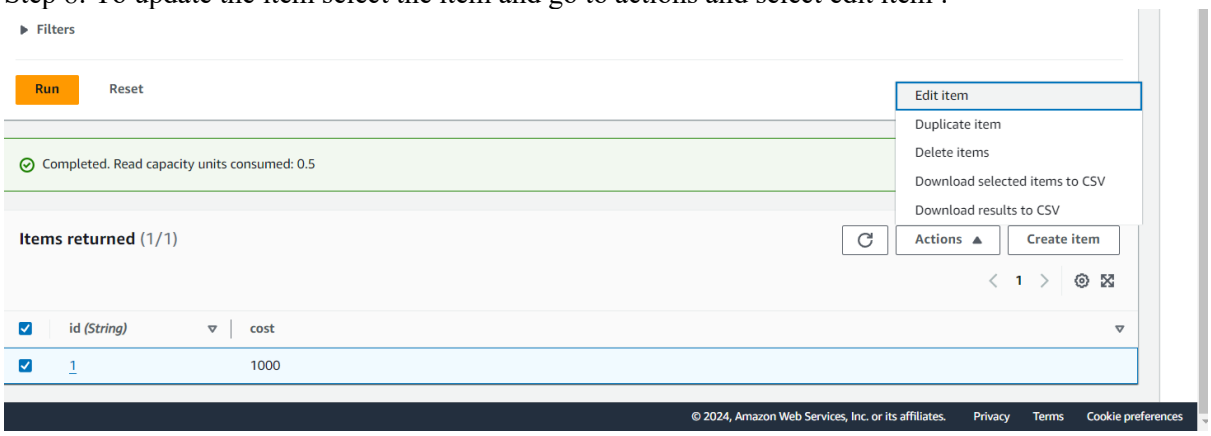
Step 7: Proceed to Enter the name of the Attribute and value of the attribute and click on Create Item.



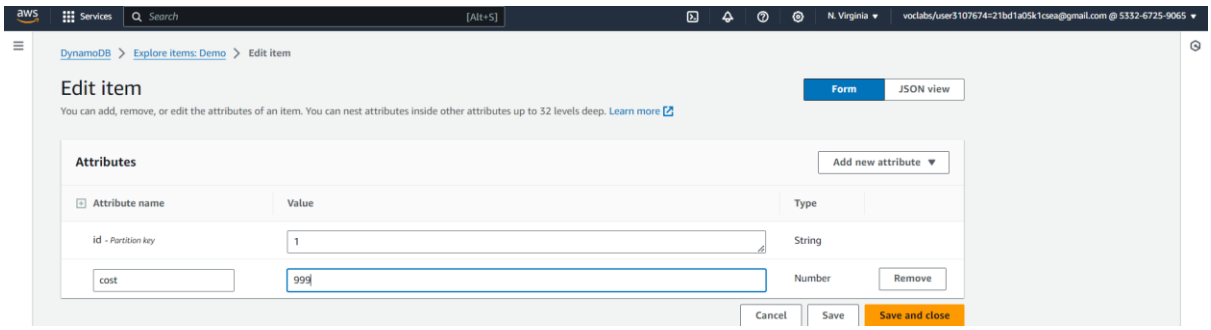
We can see that value is added to the table .



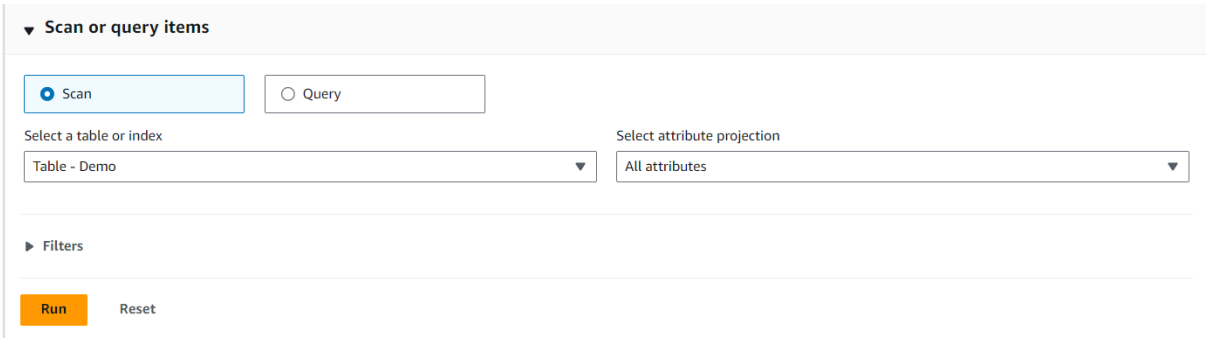
Step 8: To update the item select the item and go to actions and select edit item .



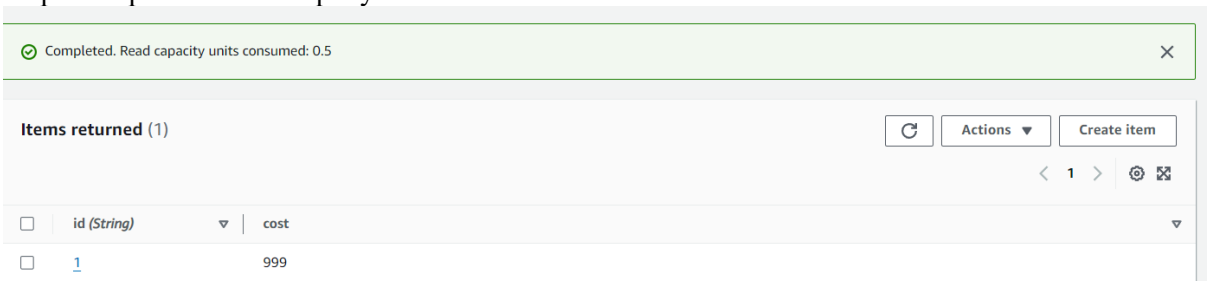
Step 9: Update the value and click on save and close.



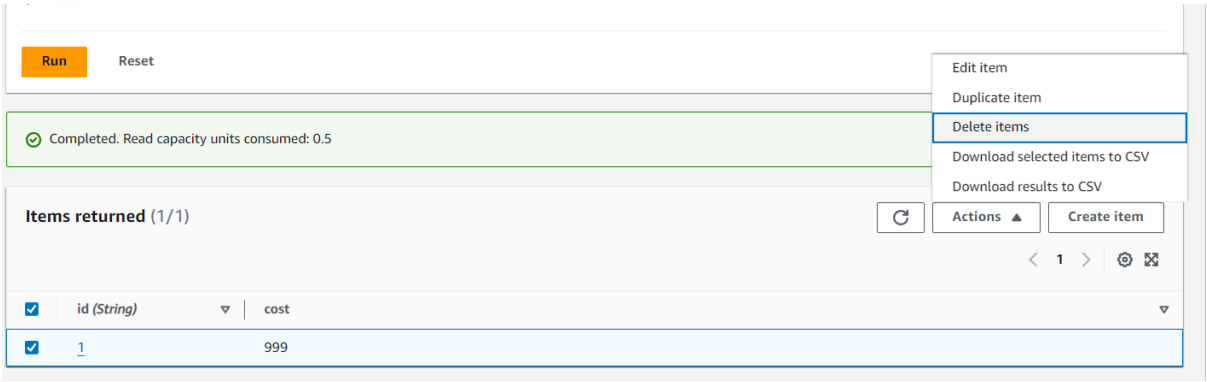
Step 10: To display the items navigate to the scan or query items panel and select all attributes from select attribute projection and click on run.



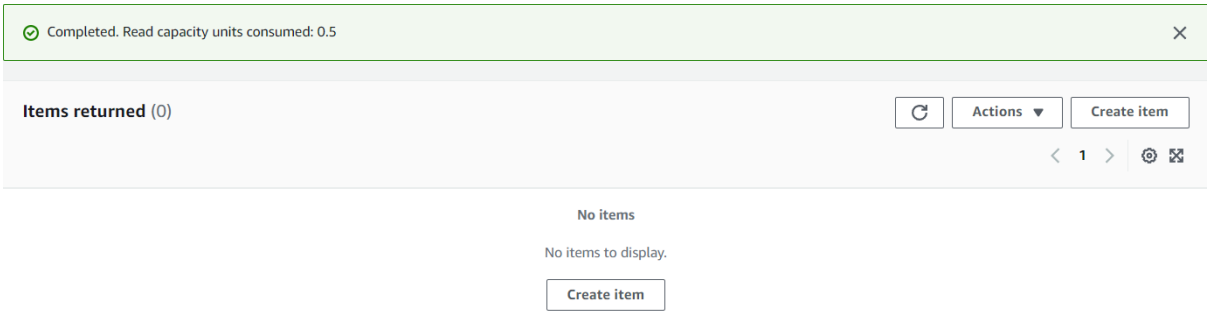
Step 11: Upon successful query we can see all the values of the table .



Step 12: To Delete an item , select the item ,click on Actions Drop-down box and click on Delete Items.



Step 13: Click on Run from the Scan or query Items panel and click on Run to see the updates in the table.

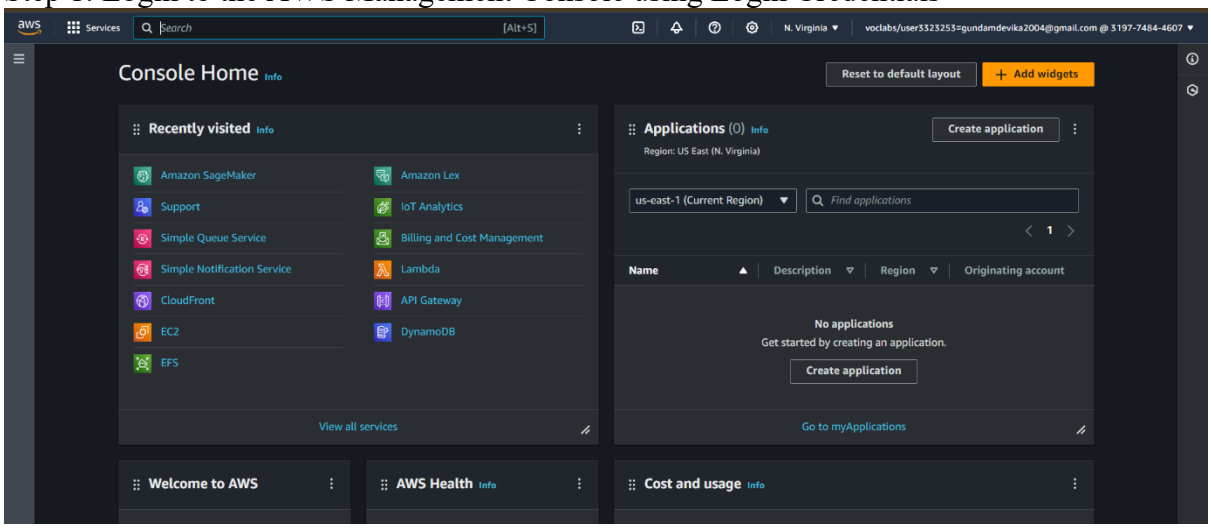


We can see that the table is empty and there are no values in the table.

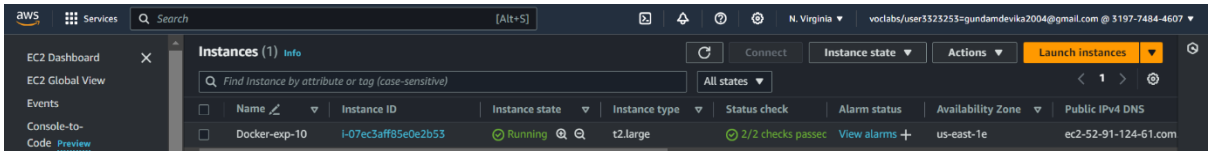
Experiment-10

Aim: Migrate a website from local server to Cloud using Docker.

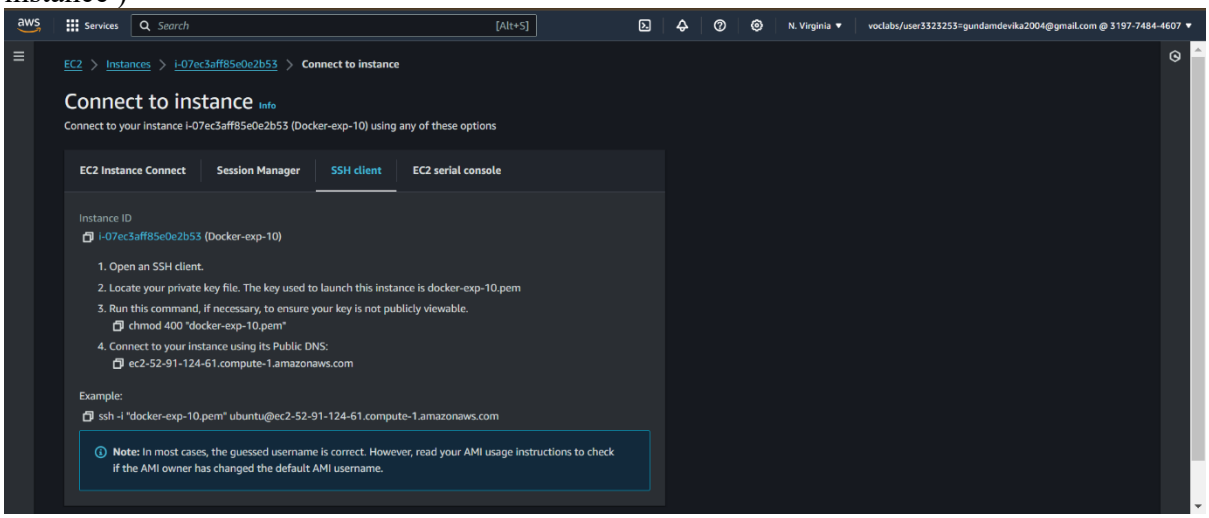
Step 1: Login to the AWS Management Console using Login Credentials



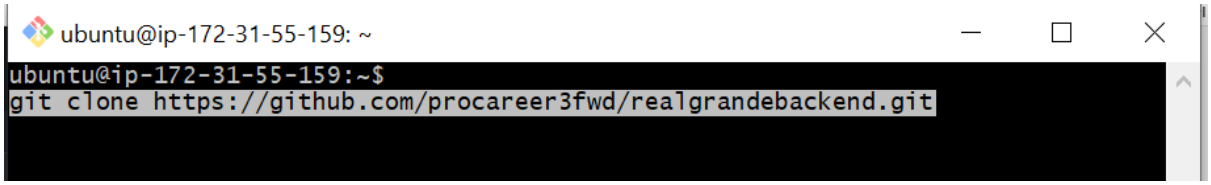
Step 2: Navigate to the EC2 Service by using Search bar and Launch an EC2 instance with required storage (30GB) and type (t2.large) and Launch the instance



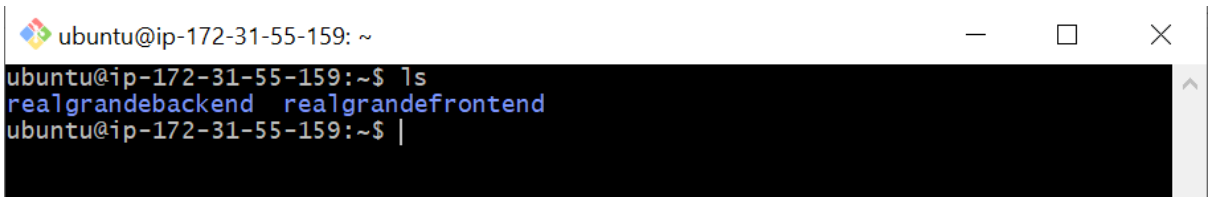
Step 3: Connect to the instance through SSH Client (or through any other way to connect to instance)



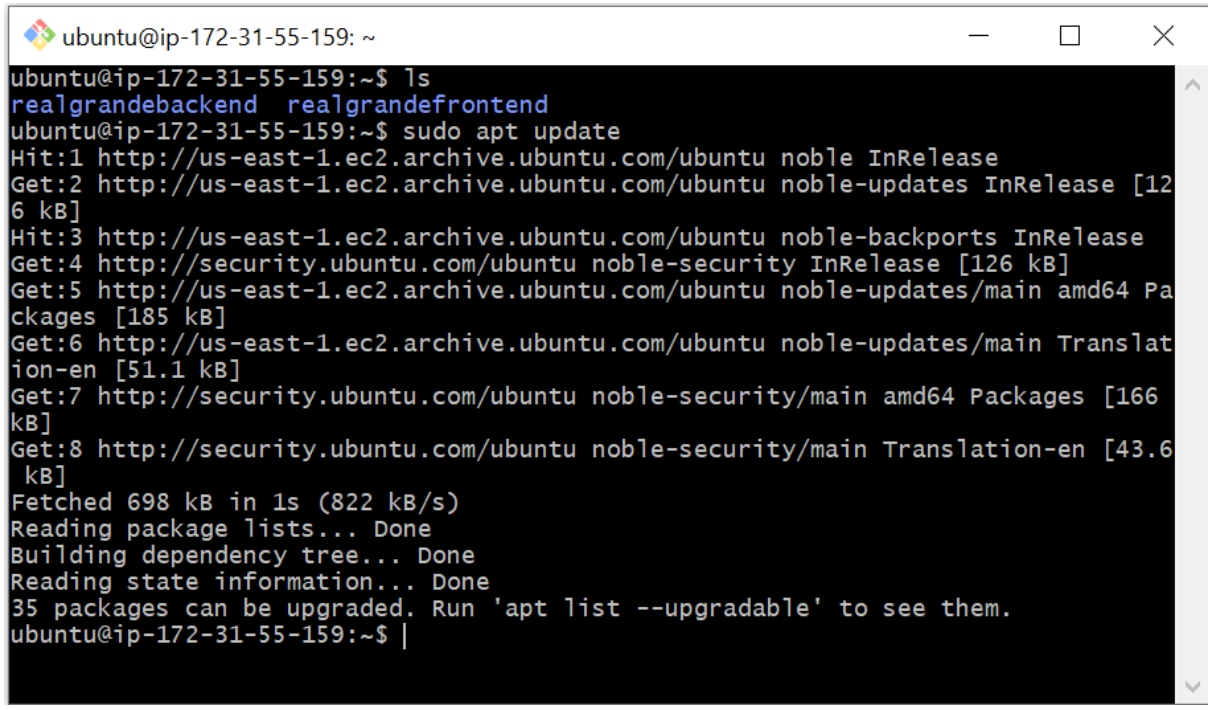
Step 4: Login to the GitHub and get the URL's for the FrontEnd and BackEnd repositories and clone them into the instance connected, such as $git clone https://url-to-frontend.git



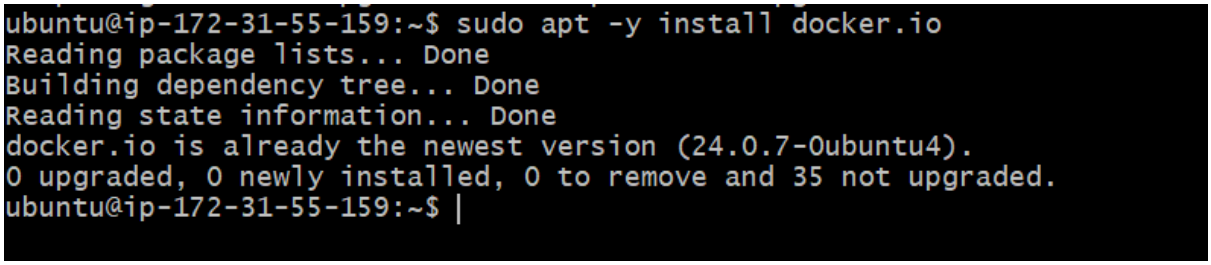
Step 5: Perform same step for the backend repository also. Now we can check the two repositories have cloned to our local server. $ls



Step 6: Perform the update operation on Instance. $sudo apt update, all the required packages and updates will be done to the instance.



Step 7: After updating the instance, now to create the images of our front end and back end we need to install the docker on the local machine. $sudo apt -y install docker.io



Step 8: Now Navigate to the realgrandebackend folder and search for .env file wether it is exists in the folder or not, if not we need to create the .env file

ubuntu@ip-172-31-55-159: ~/realgrandebackend

```
ubuntu@ip-172-31-55-159:~/realgrandebackend$ ls
Dockerfile  models  package-lock.json  package.json  routes  server.js
ubuntu@ip-172-31-55-159:~/realgrandebackend$
```

Step 9: To create .env file, run the command $nano .env and specify the database credentials in this file such as mongodburl, dbusername, dbpassword, in frontenduri specify the public ip of our instance

ubuntu@ip-172-31-55-159: ~/realgrandebackend

```
  GNU nano 7.2                                                                .env
MONGODBURL="mongodb+srv://fsd04.2hxrdca.mongodb.net/realgrande?retryWrites=true&w=majority"
DBUSERNAME=procareer3
DBPASSWORD=ISobjBDohsFqEAqg
FRONTENDURI="http://34.229.184.54"
```

Step 10: Now we need to build the docker image for the backend, run the command to build the docker image. $sudo docker build -t backend_server . and check the docker image $sudo docker ps

```
ubuntu@ip-172-31-55-159:~/realgrandebackend$ sudo docker build -t backend_server .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  179.2kB
Step 1/6 : FROM node
 ---> 3d4b037e6712
Step 2/6 : WORKDIR /app
 ---> Using cache
 ---> ac262bc309f2
Step 3/6 : COPY . /app
 ---> Using cache
 ---> f49f99875b21
Step 4/6 : RUN npm install
 ---> Using cache
 ---> cd5ebdecfc7a
Step 5/6 : EXPOSE 5000
 ---> Using cache
 ---> 88583bf19491
Step 6/6 : CMD ["npm","start"]
 ---> Using cache
 ---> b057381e8a17
Successfully built b057381e8a17
Successfully tagged backend_server:latest
ubuntu@ip-172-31-55-159:~/realgrandebackend$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
ubuntu@ip-172-31-55-159:~/realgrandebackend$
```

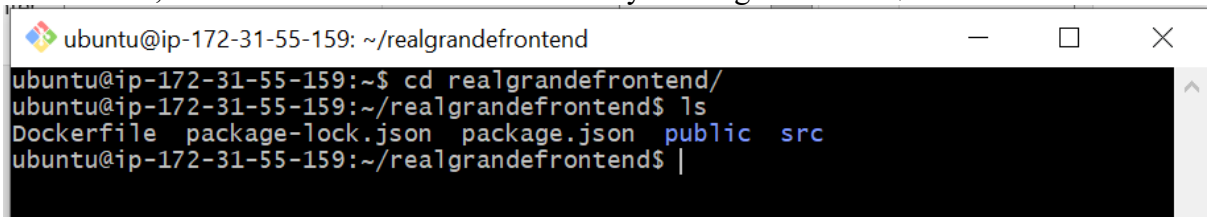
Step 11: Now run the image created for the backend $sudo docker run -d -p 2001:5000 backend_server

```
ubuntu@ip-172-31-55-159:~/realgrandebackend$ sudo docker run -d -p 2001:5000 backend_server
d1548fd58799eeb01b512d773784179e7a5367a3d239c9c77f78de42f753a708
ubuntu@ip-172-31-55-159:~/realgrandebackend$
```

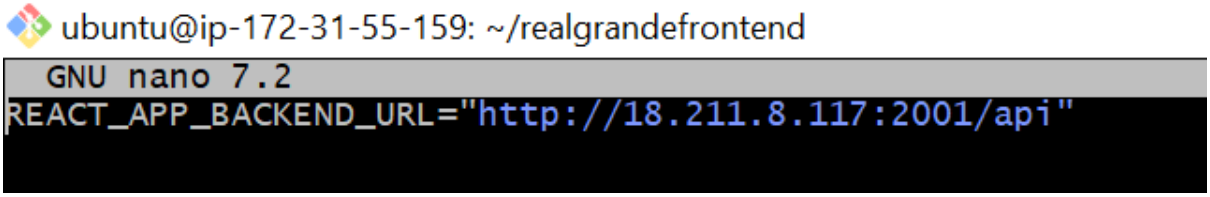
Step 12: Now try to access the data in the browser by <Ec2_ip_address>:2001/api, we need to get the json data


```
[
  {
    "name": "Adeel Solangi",
    "language": "Sindhi",
    "id": "V59OF92YF627HFY0",
    "bio": "Donec lobortis eleifend condimentum. Cras dictum dolor lacinia lectus vehicula rutrum. Maecenas quis nisi nunc. Nam tristique feugiat est vitae mollis. Maecenas quis nisi nunc.",
    "version": 6.1
  },
  {
    "name": "Afzal Ghaffar",
    "language": "Sindhi",
    "id": "ENTOCR13RSCLZ6KU",
    "bio": "Aliquam sollicitudin ante ligula, eget malesuada nibh efficitur et. Pellentesque massa sem, scelerisque sit amet odio id, cursus tempor urna. Etiam congue dignissim volutpat. Vestibulum pharetra libero et velit gravida euismod.",
    "version": 1.88
  },
```

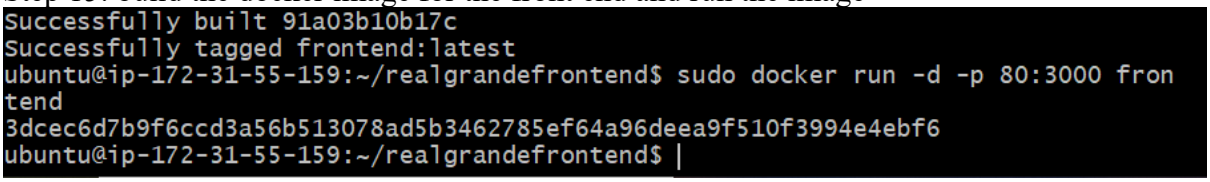

Step 13: Perform the above steps for frontend, navigate to the frontend folder and search for the .env file, if not found create a new .env file by running command $nano .env



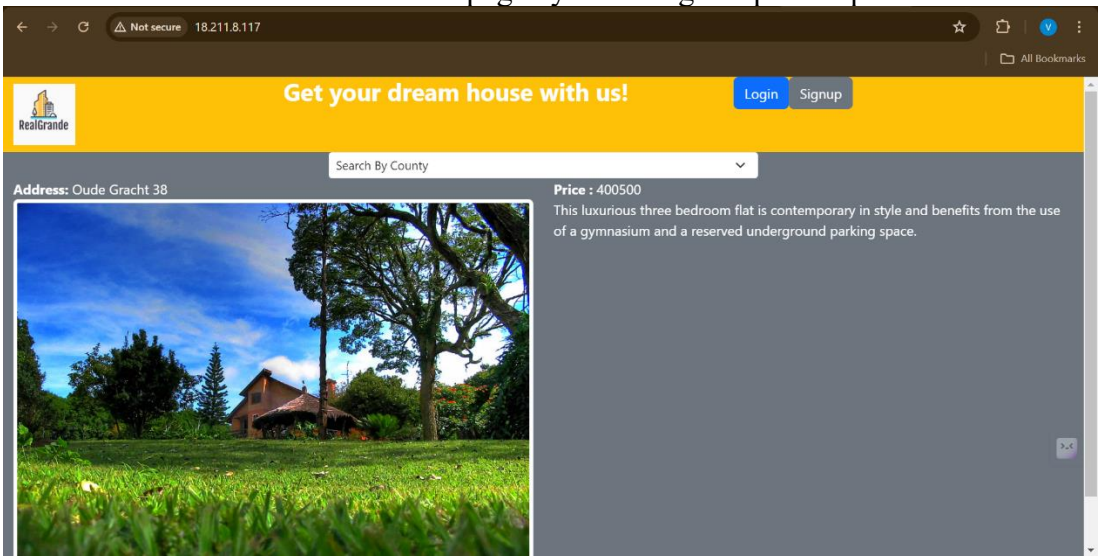
Step 14: Replace our public ip address in the .env file



Step 15: build the docker image for the front end and run the image



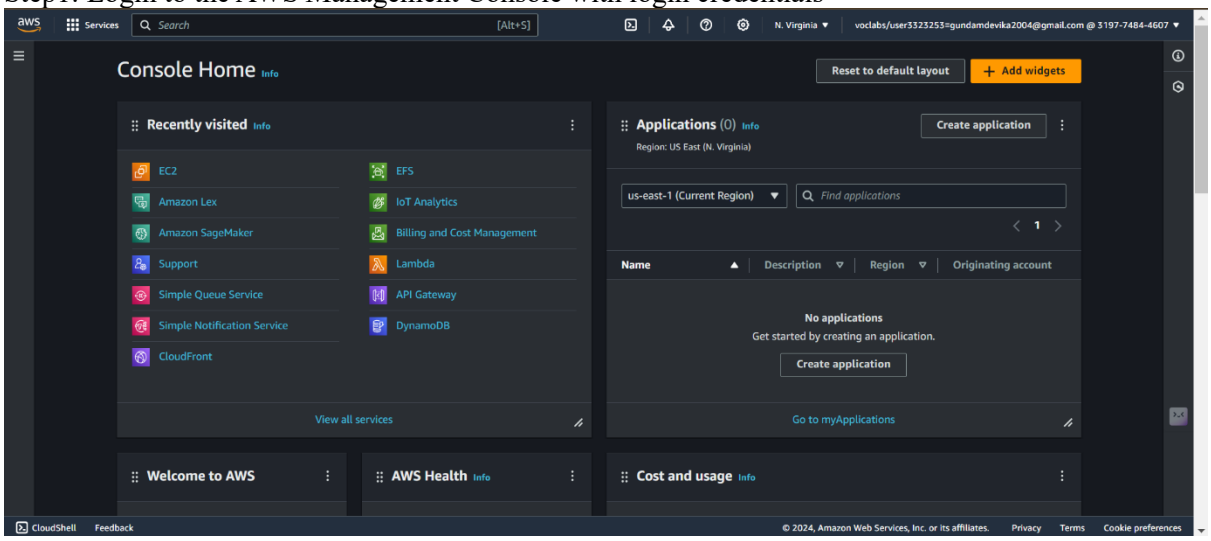
Step 16: Now successfully we created the frontend and backend images by using docker containers and Access the front end page by browsing our public ip address of our instance.



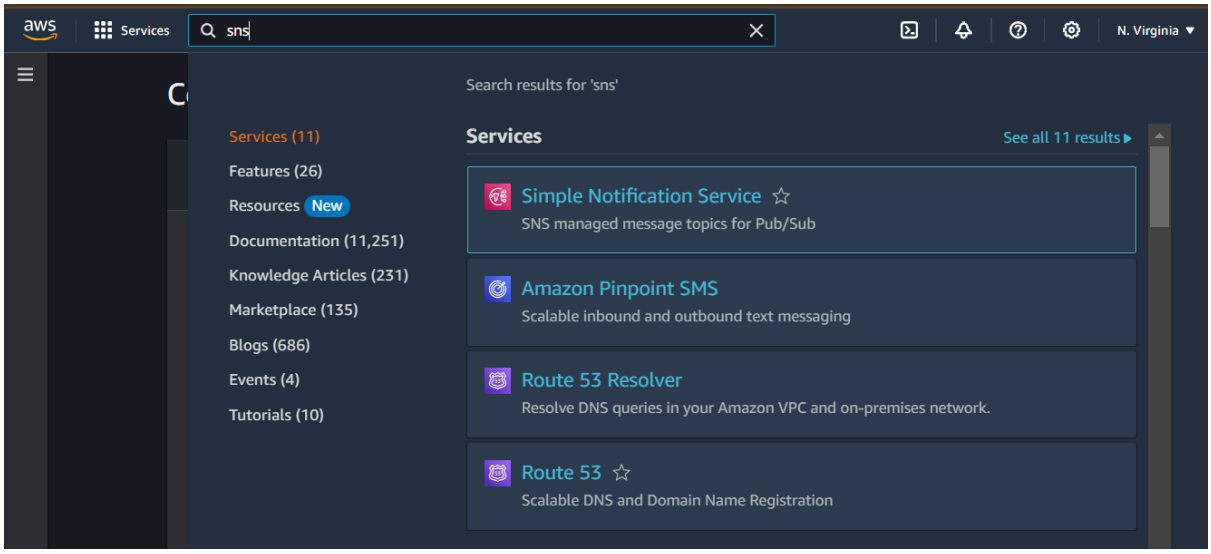
Experiment-11

Aim: Create loosely coupled services with Amazon SQS and Amazon SNS to process data received from the applications.

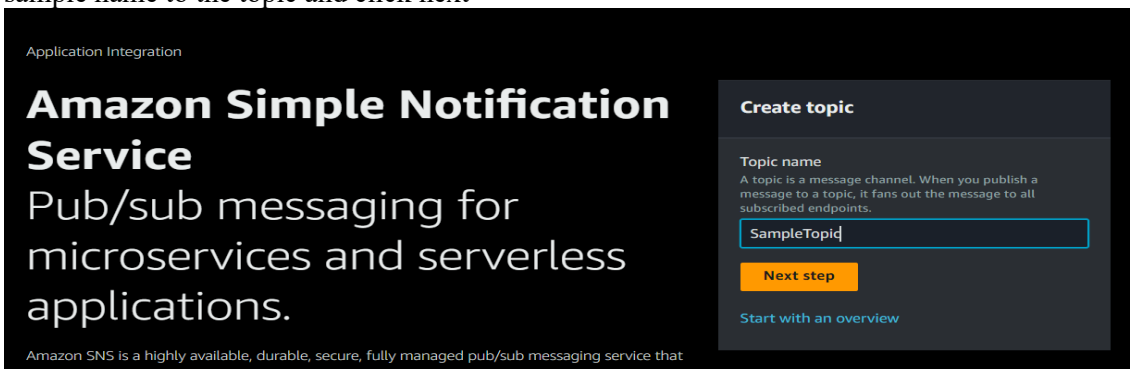
Step1: Login to the AWS Management Console with login credentials



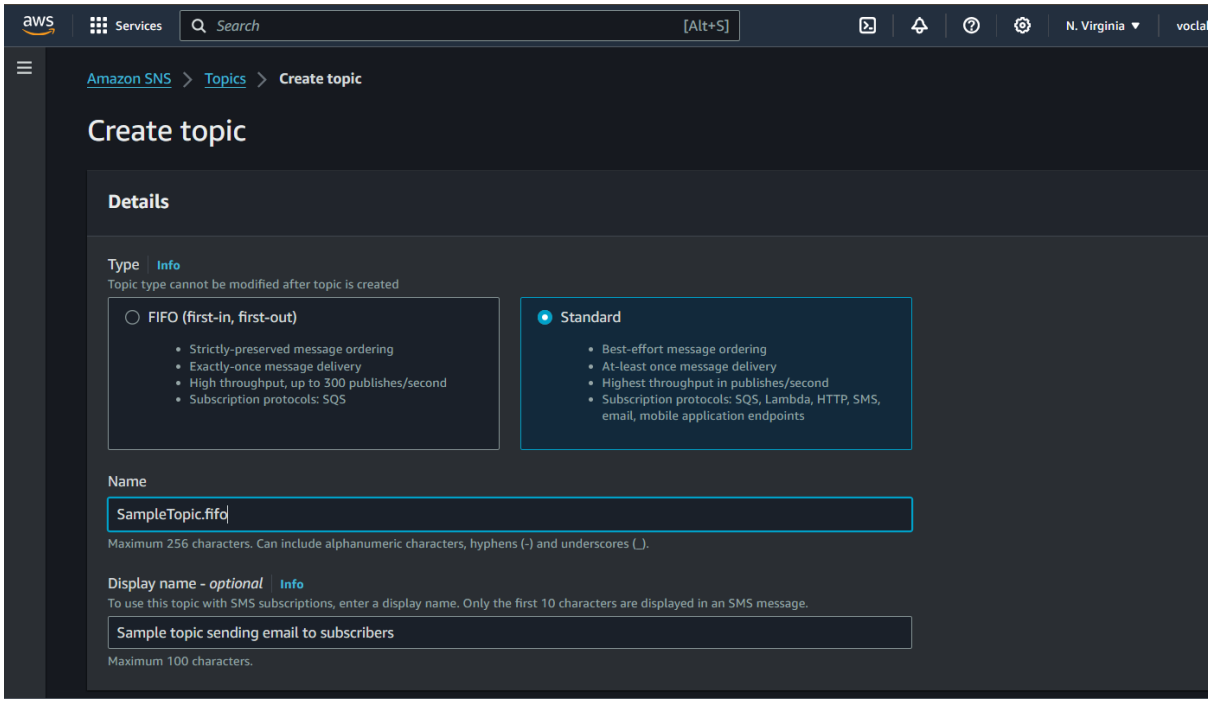
Step 2: Navigate to the search bar and search for the service SNS(Simple Notification Service)



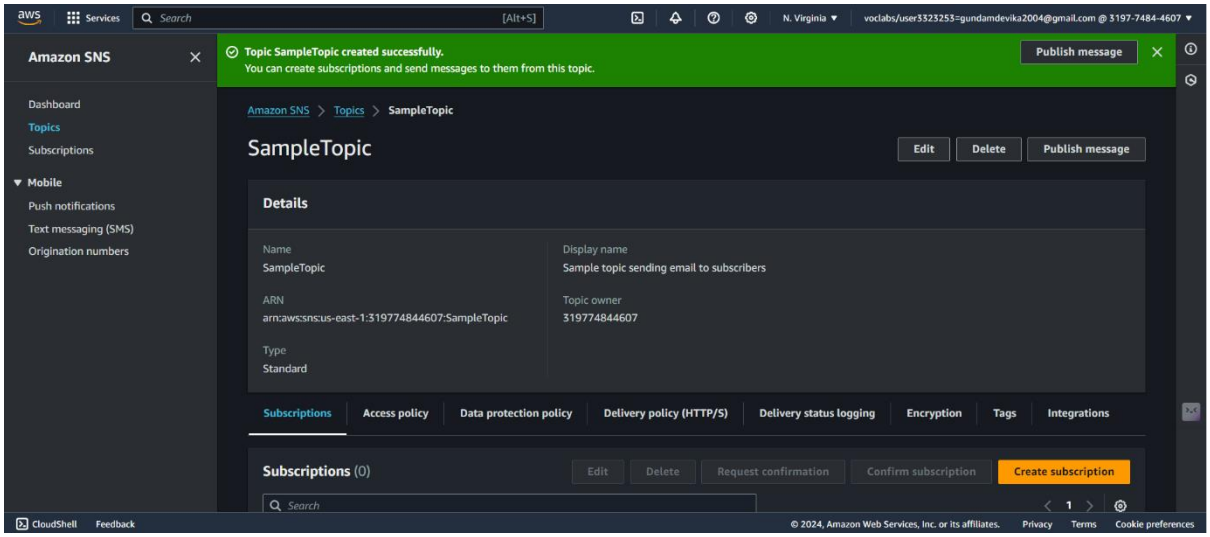
Step 3: After Selecting the SNS service from the search bar, navigate to the service dashboard, give a sample name to the topic and click next



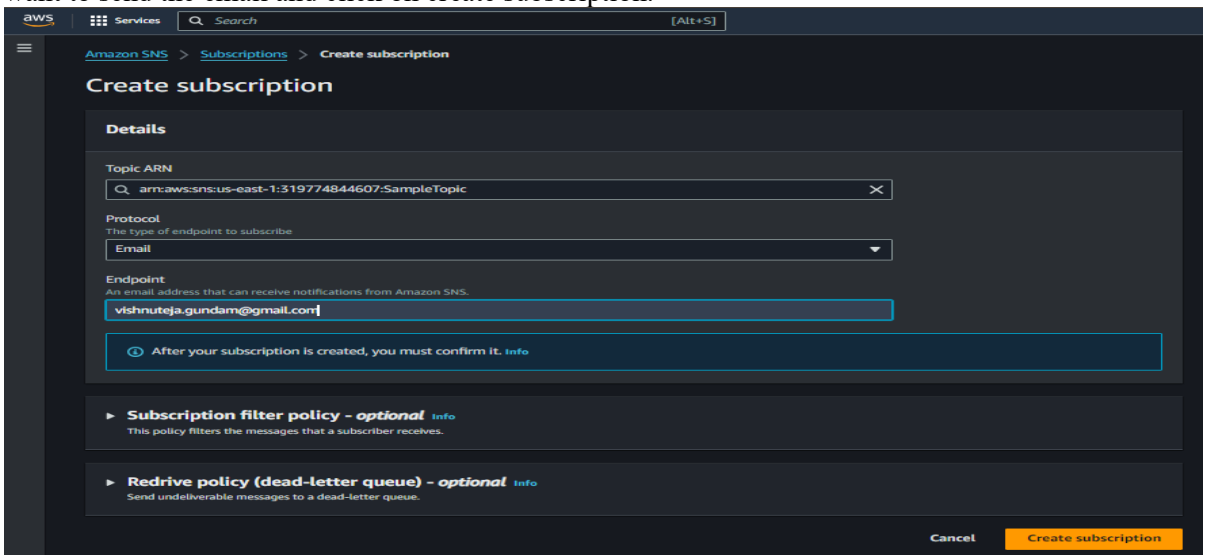
Step 4: Select the topic type to Standard and in the topic name and give some description if needed.



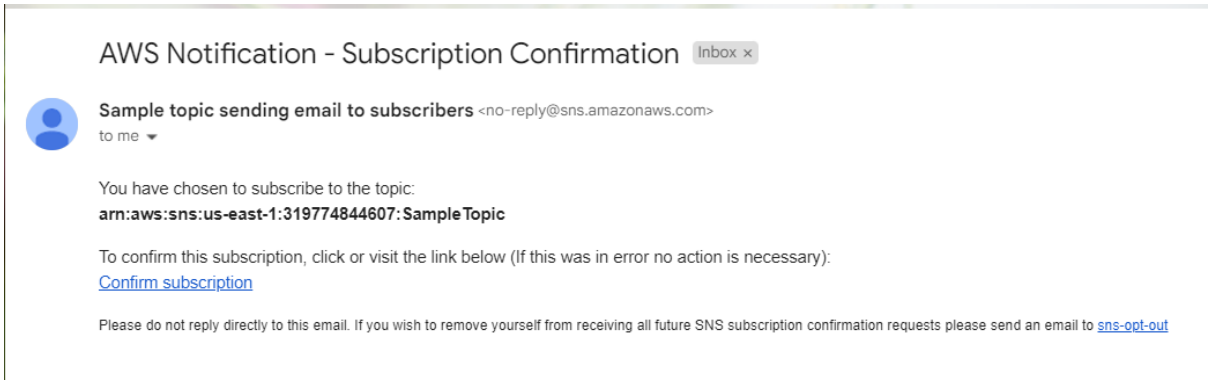
Step 5: Then Leave all the optional settings to default and scroll down to the bottom of the page and click on create topic. After successfully creating the topic then click on create subscription.



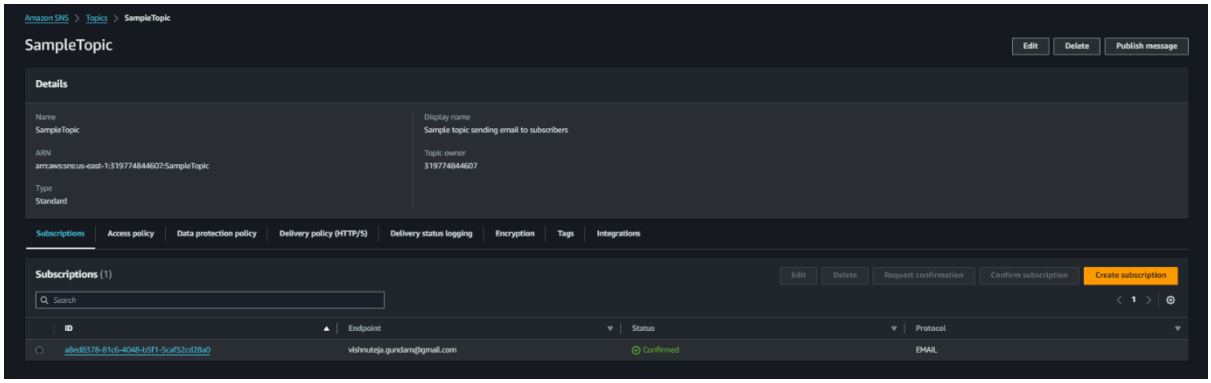
Step 6: Select the protocol as email and specify the Endpoint i.e. the subscriber email ID to which we want to send the email and click on create subscription.



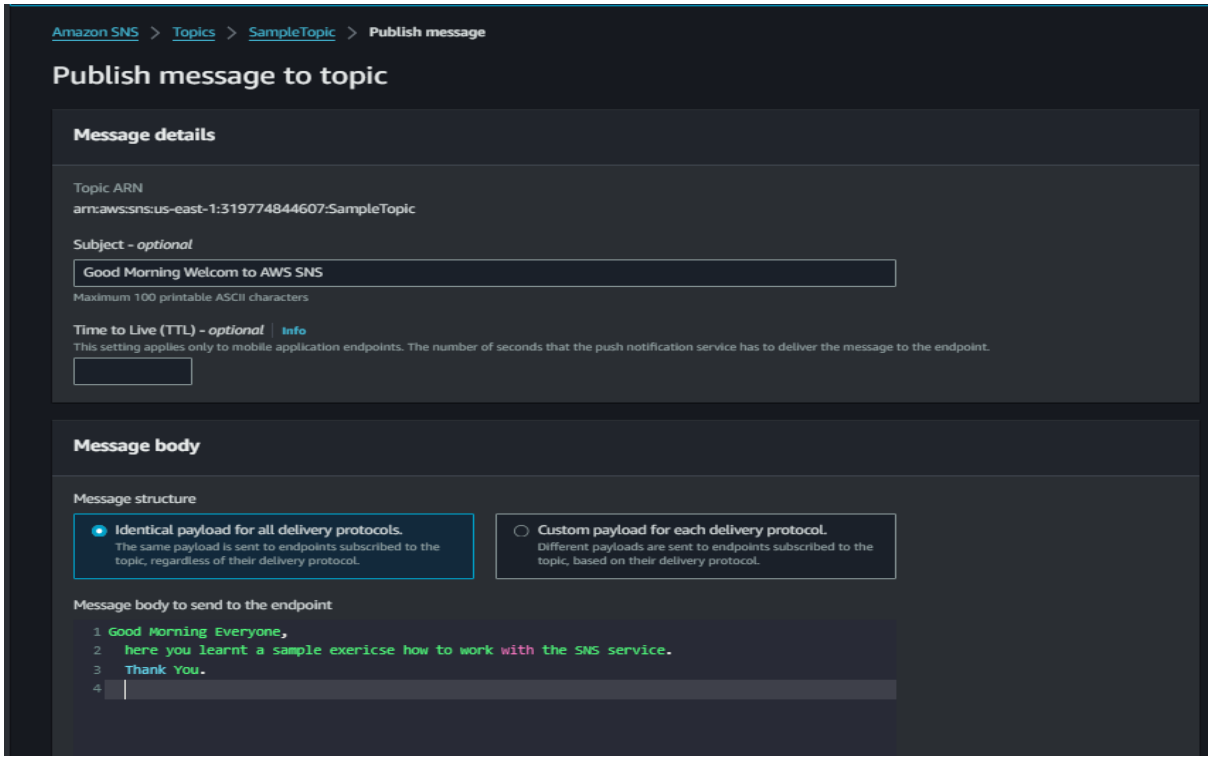
Step 7: After successfully creation of subscription, verify the email send to the subscriber mail id, by clicking confirm subscription



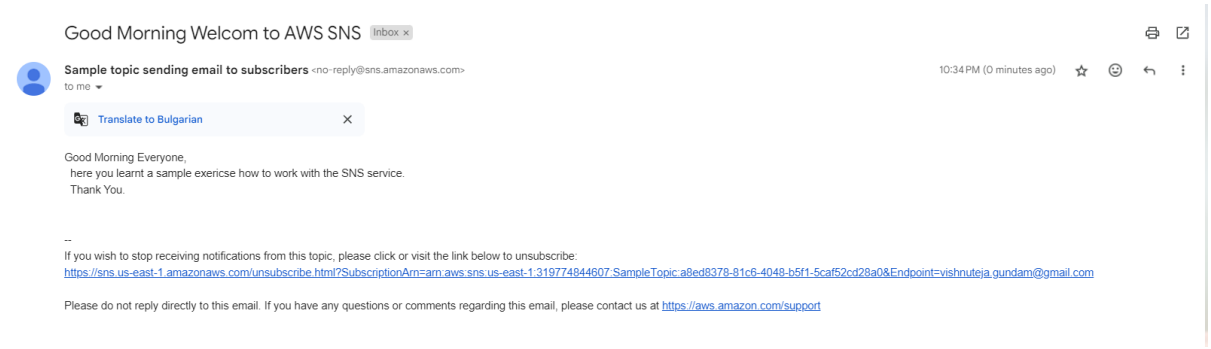
Step 8: After successfully Confirming the Subscription, then click on publish message.



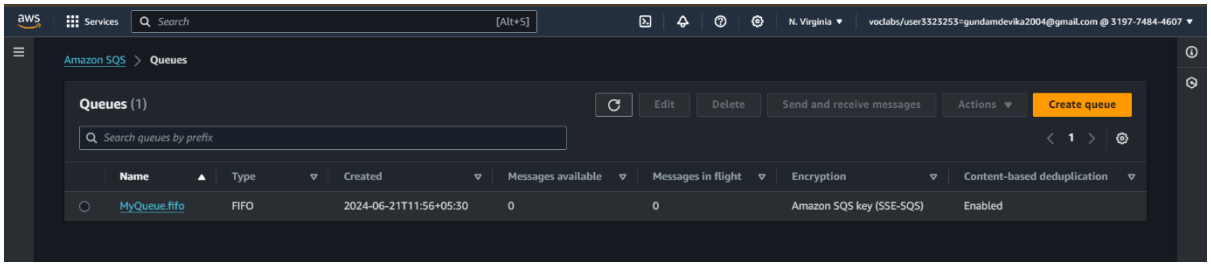
Step 9: Enter the subject and the body of the mail to want to send to the subscriber.



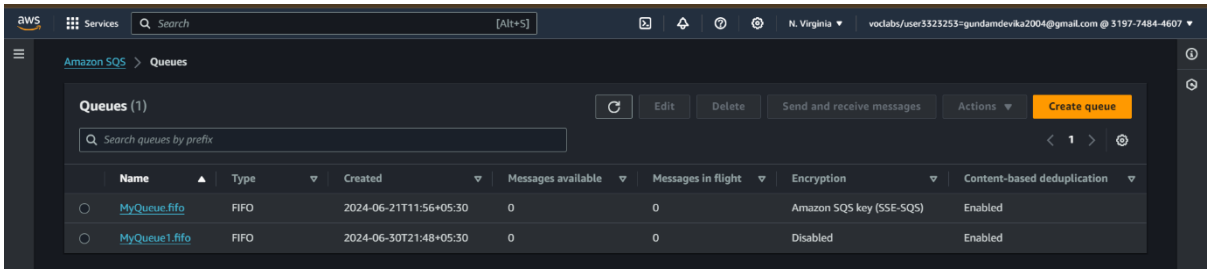
Step 10: Click on Publish Message, Now check the subscriber email and find the mail send by the SNS service.



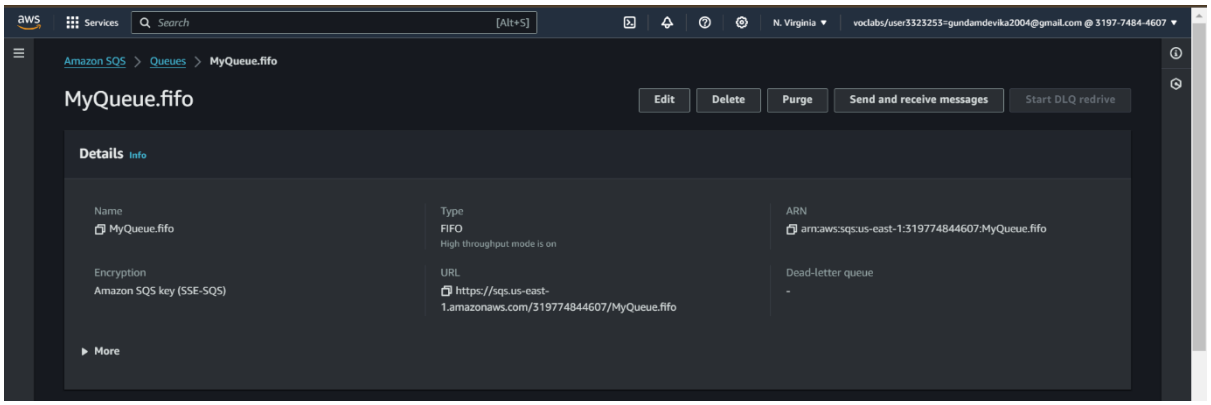
Step 11: Navigate to the SQS service through the search bar. Click on create Queue.



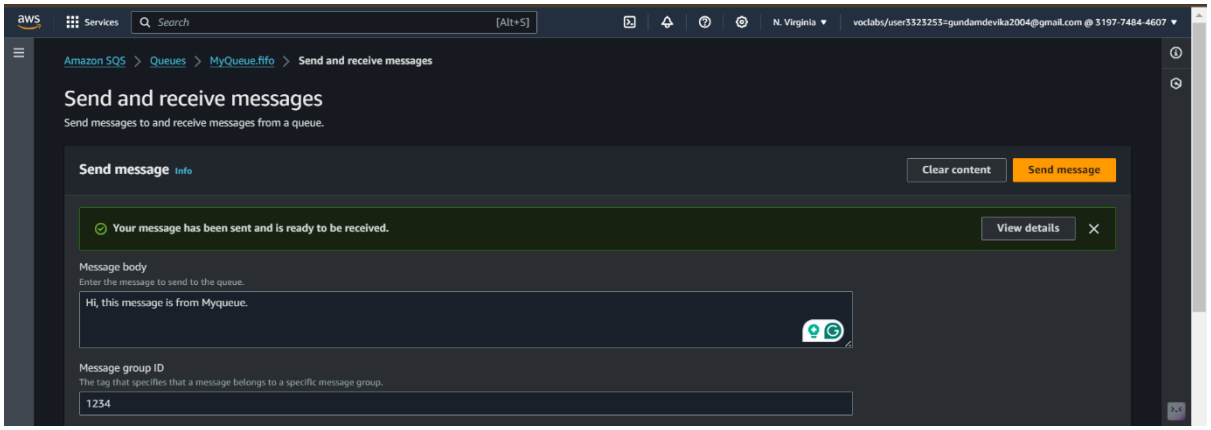
Step 12: Name the Queue with .fifo extension and enable the content-based duplication. And click on create Queue.



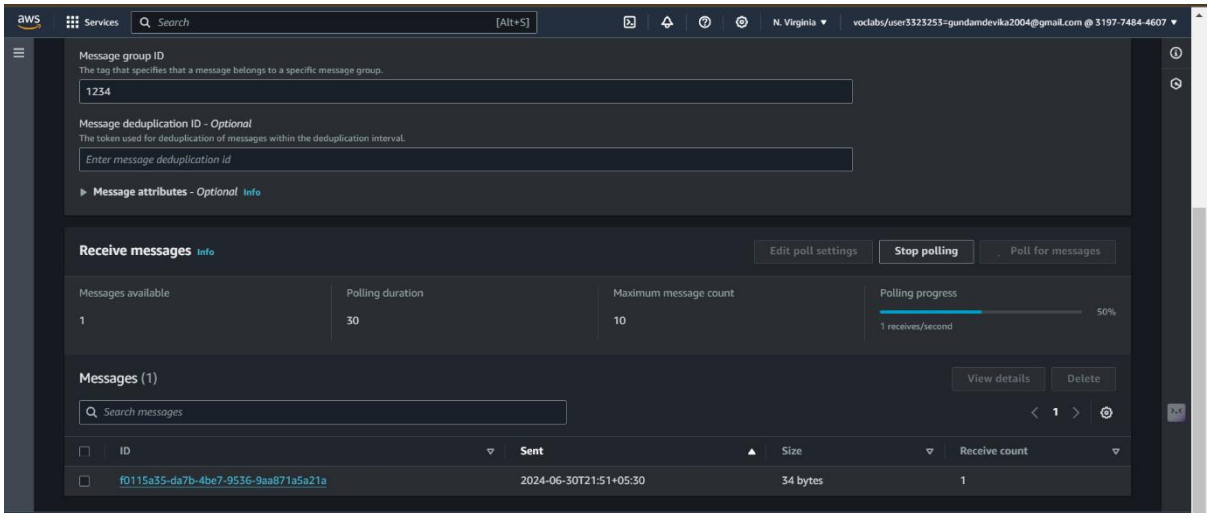
Step 13: click on your queue and and click on send and receive messages in the right side top .



Step 14: Enter the message body, group ID, and click on send message.



Step 15: Scroll down, to the receive message pane, their click for poll messages and select the message sent from the queue.



Step 16: The message from the queue sent will be displayed here.