Bharath Kumar N

PS ID:10843180

Milestone assessment 2 - AWS -Set 1

Linux 1.2

You are the AWS Administrator for LTIMindtree organization and your management has decided to implement an infrastructure with the following configurations.

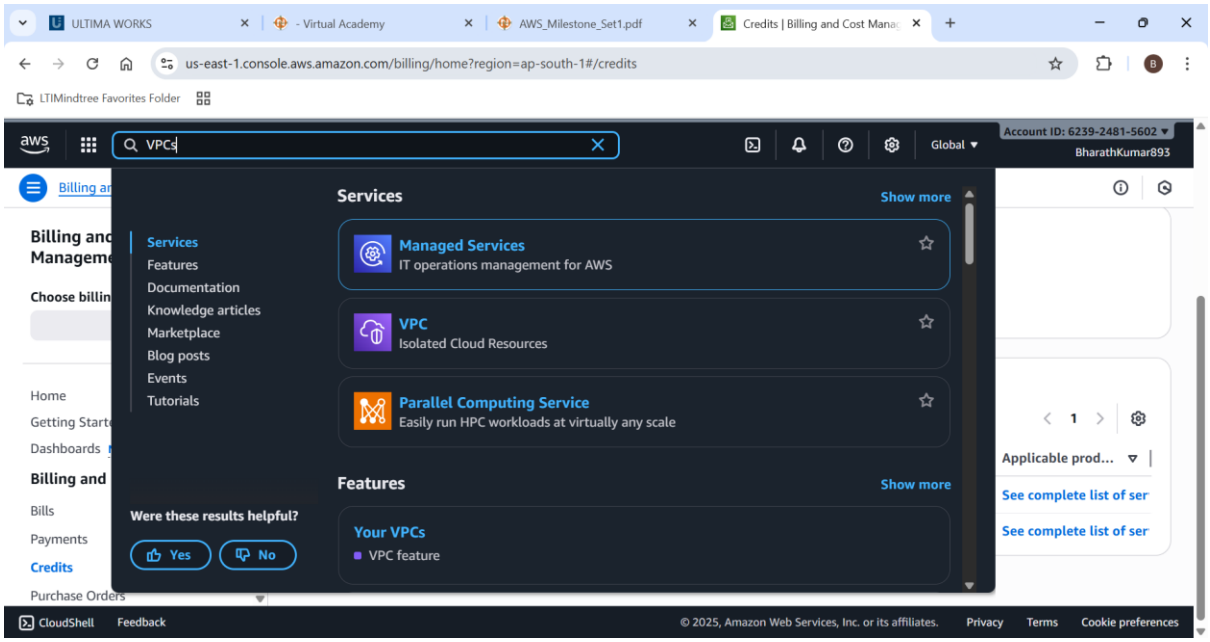
1. **Create a VPC**

   - **VPC Name: DevVPC**
   - **CIDR Block: 10.10.0.0/16**

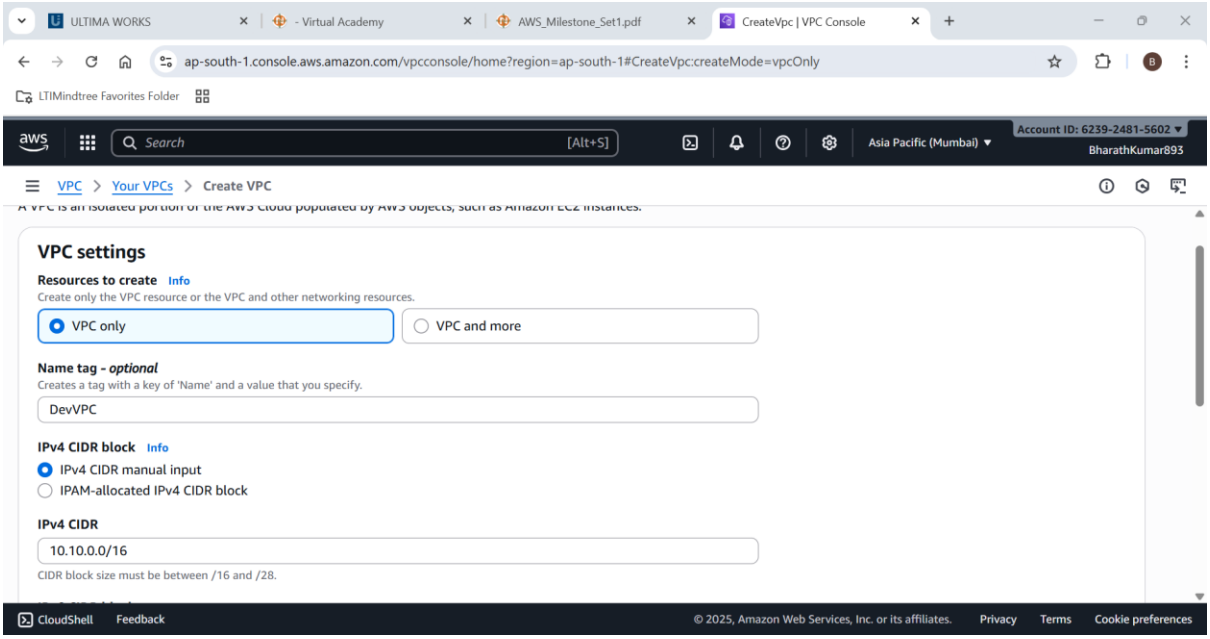
   Create two Subnets inside DevVPC:

   - **Subnet1: 10.10.1.0/24 in AZ1**
   - **Subnet2: 10.10.2.0/24 in AZ2**

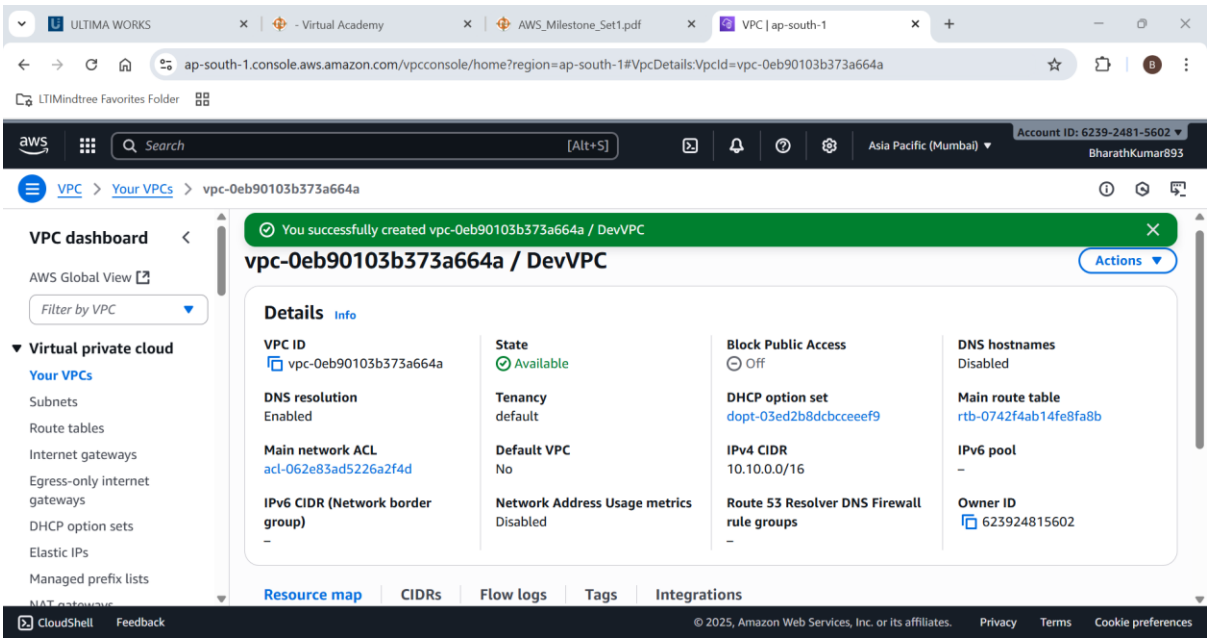
Step 1: Lets open the VPC service



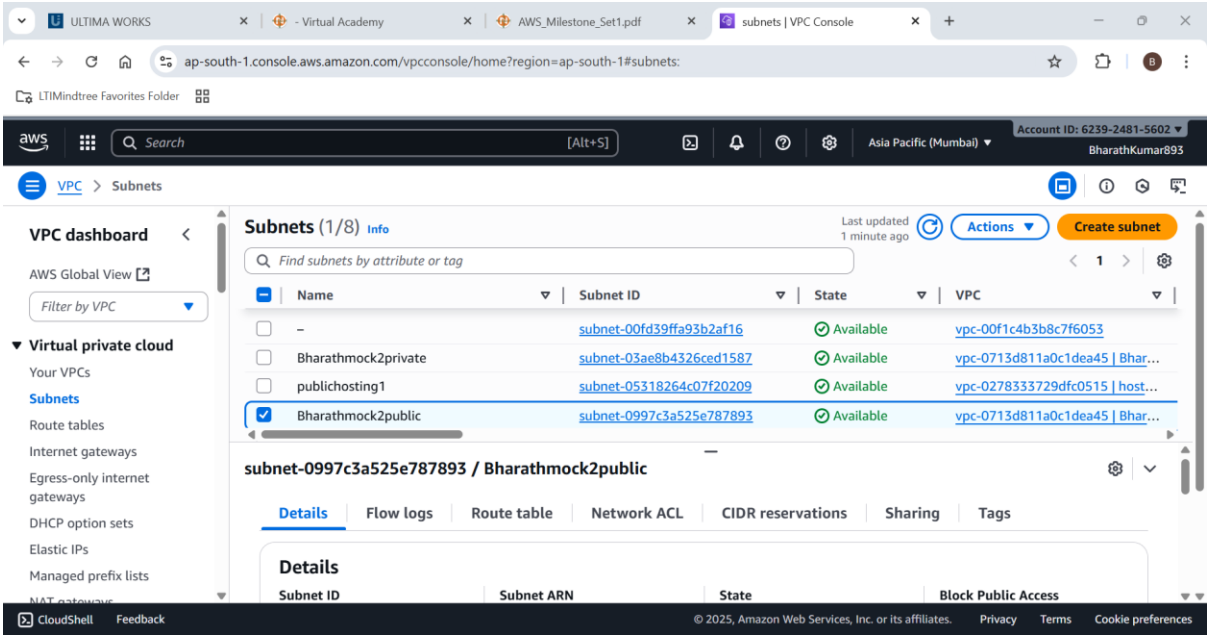
Step 2: Lets create a new VPC with the name “DevVPC” and CIDR “10.10.0.0/16”



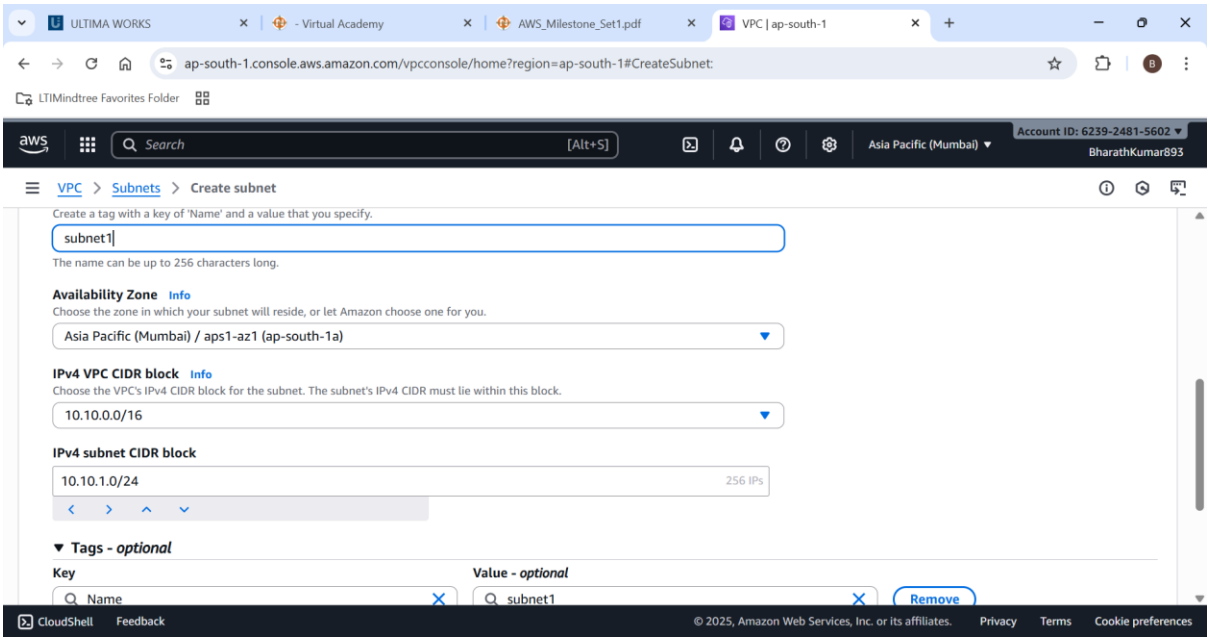
VPC Created successfully



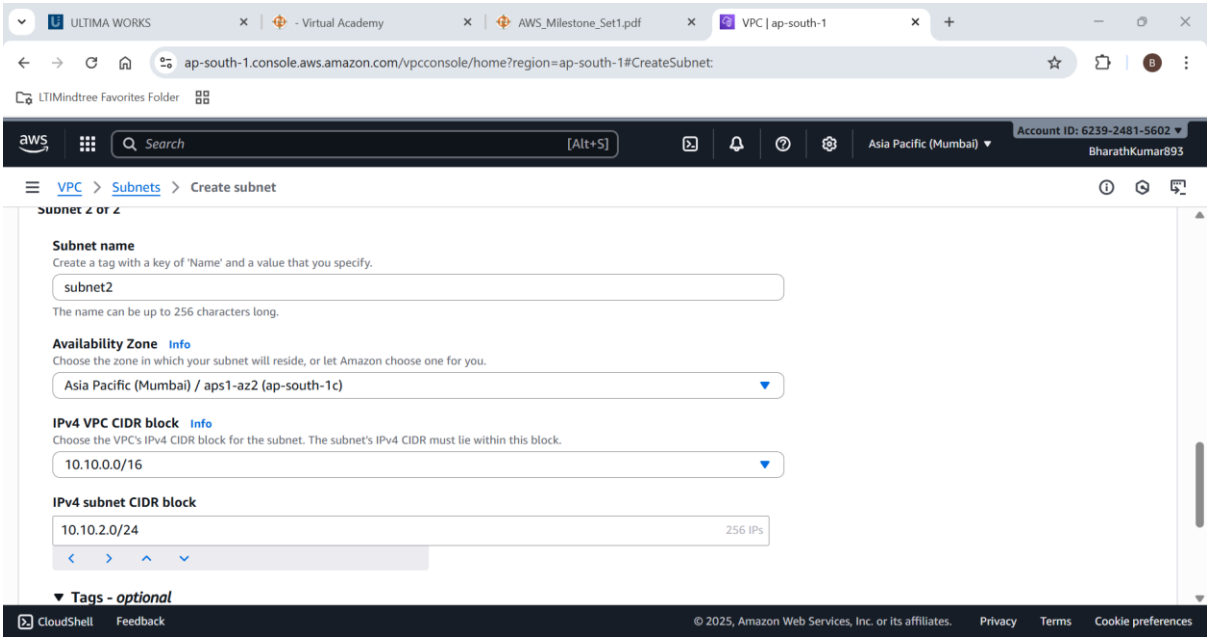
Step 3: Lets create 2 subnets for it



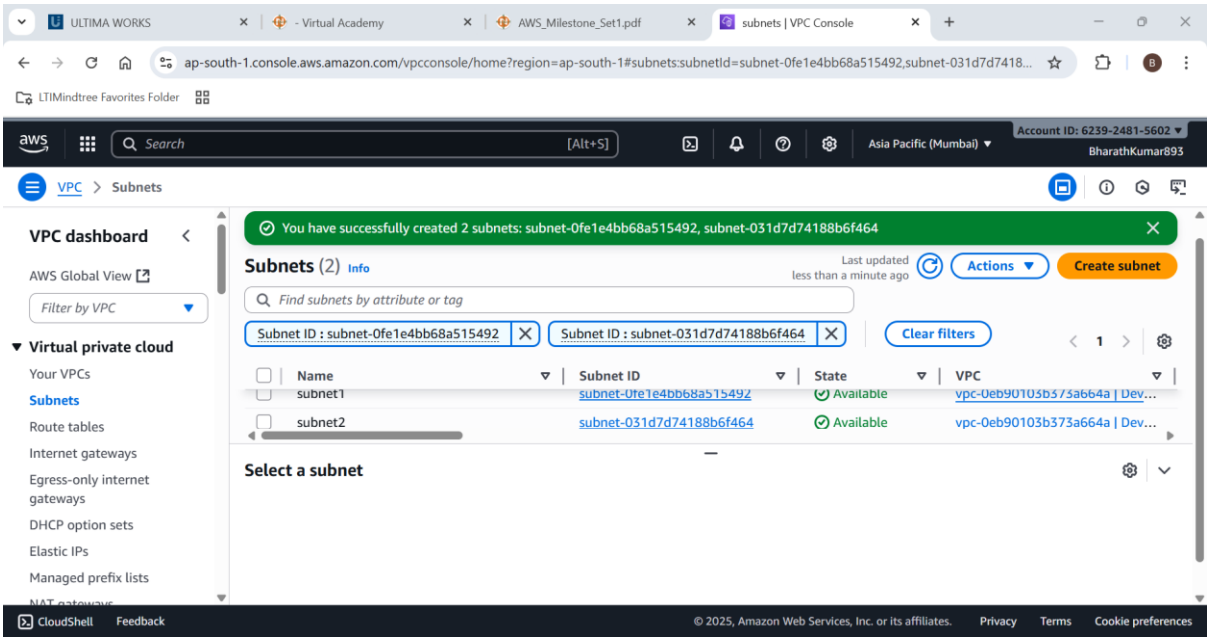
Step 4: creating subnet 1 with Az1 as availability zone and with subnet CIDR:10.10.1.0/24



Step 5: creating subnet 2 with Az2 as availability zone and with subnet CIDR:10.10.2.0/24



Subnet 1 and subnet 2 created successfully

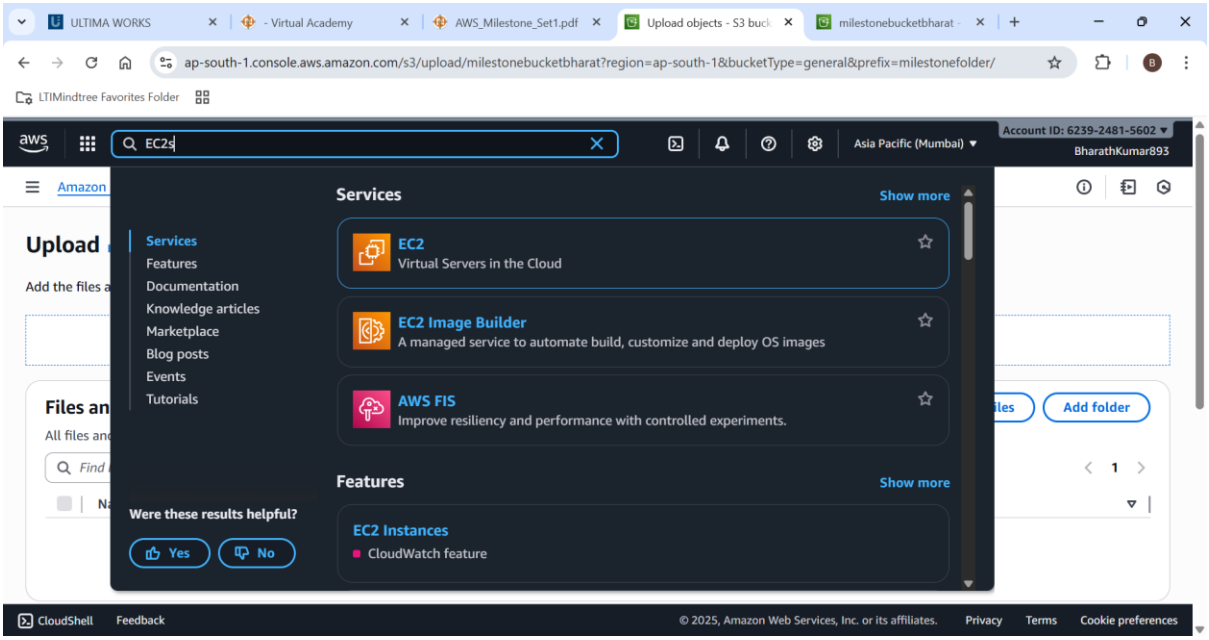


2. **Create an EC2 Instance**
   - **Instance Name: SubnetVM01**
   - **AMI: Windows Server 2019 Base / Amazon Linux 2023 Kernel-6.1 AMI**
   - **Instance Type: t2.micro(Or t3 micro)**
   - **Authentication: Create Key-Pair**
   - **VPC: DevVPC**
   - **Subnet: Subnet2**
   - **Public IP: Enabled**
   - **Security Group: Basic rules(SSH/HTTP/RDP) :**

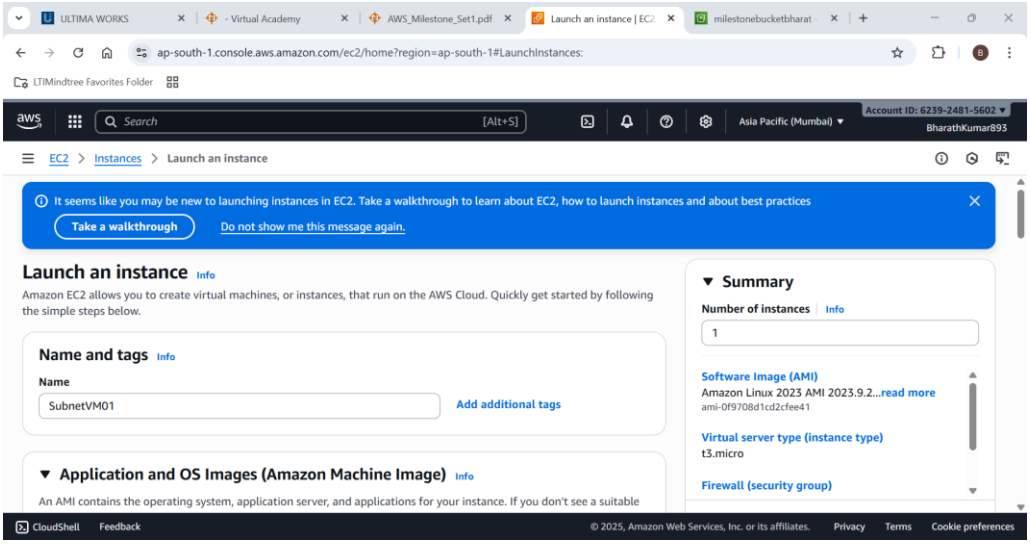
   Follow Question 3 for this •

   Region: us-east-1 (N. Virginia) or Allowed Region

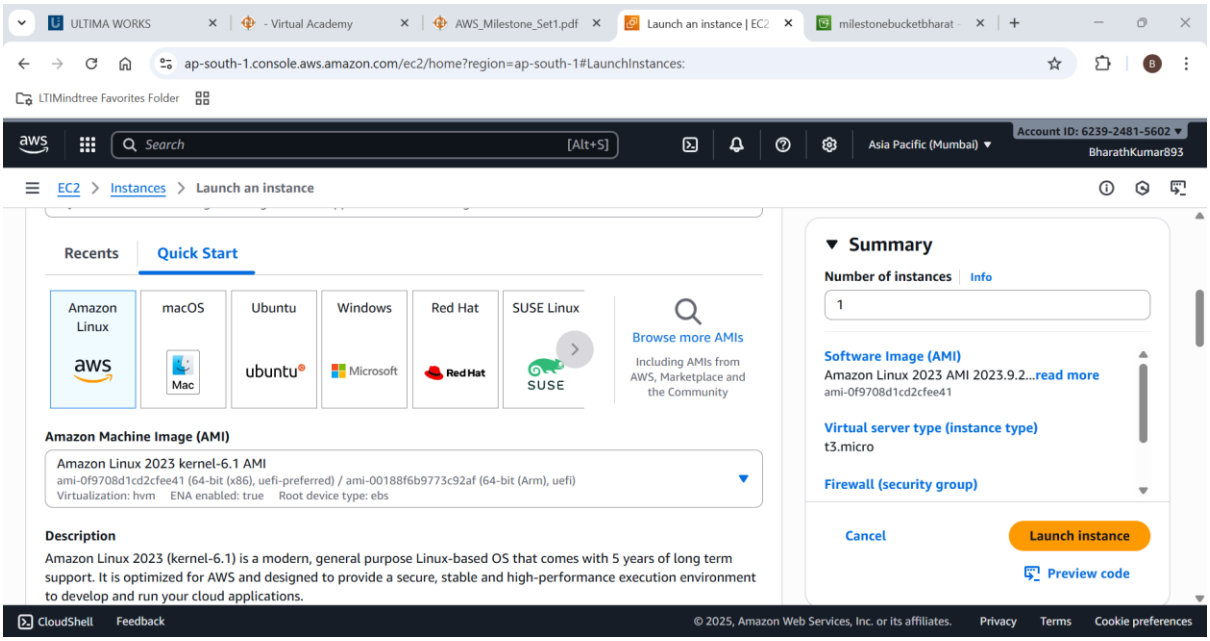
Step 1: Lets open EC2 Service



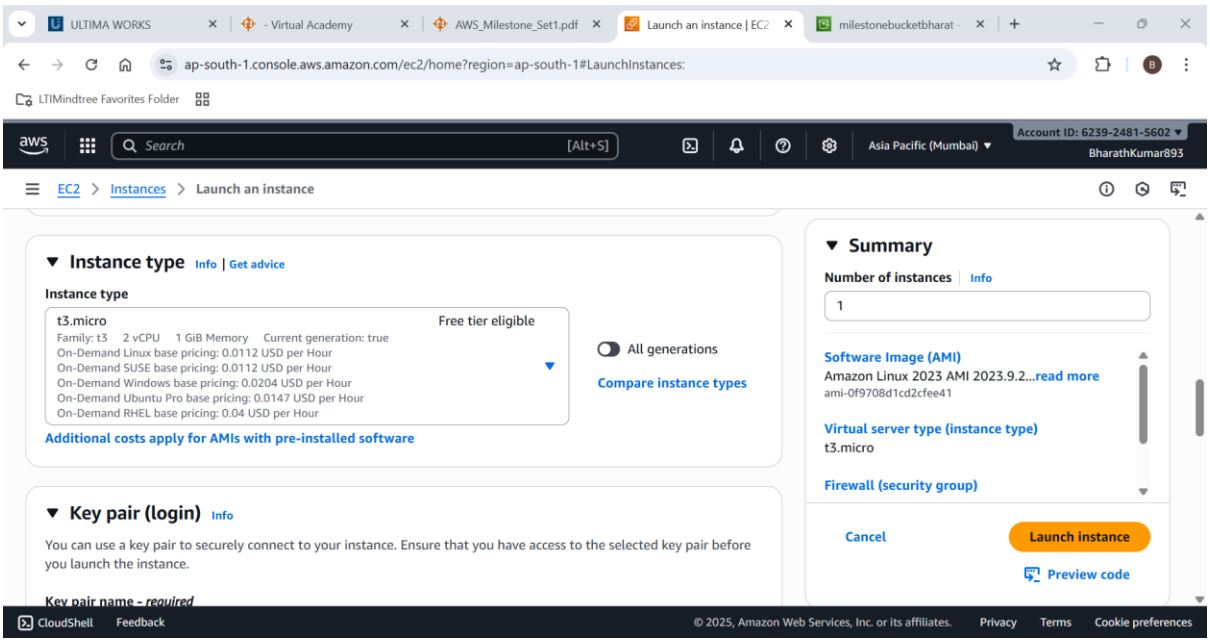
Step 2: Lets create a new EC2 instance with name subnetVm01



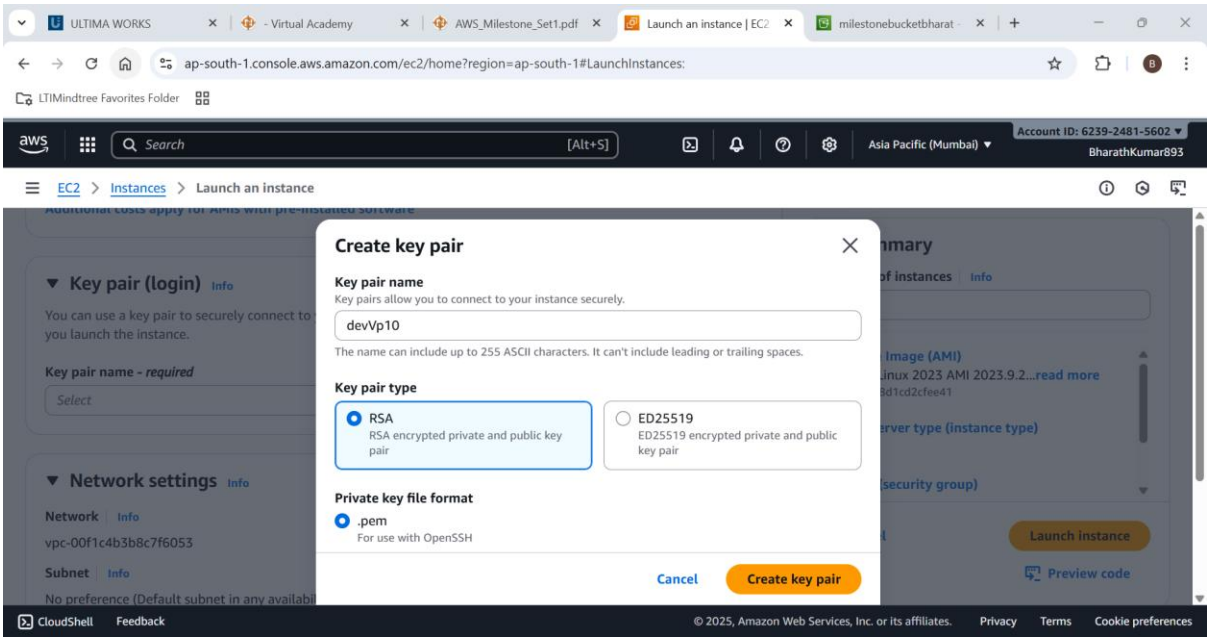
Step 3: Choos Linux / Amazon Linux 2023 Kernel-6.1 AMI as AMI



Step 4: Choose instance type as t3.micro



Create a new key-value pair

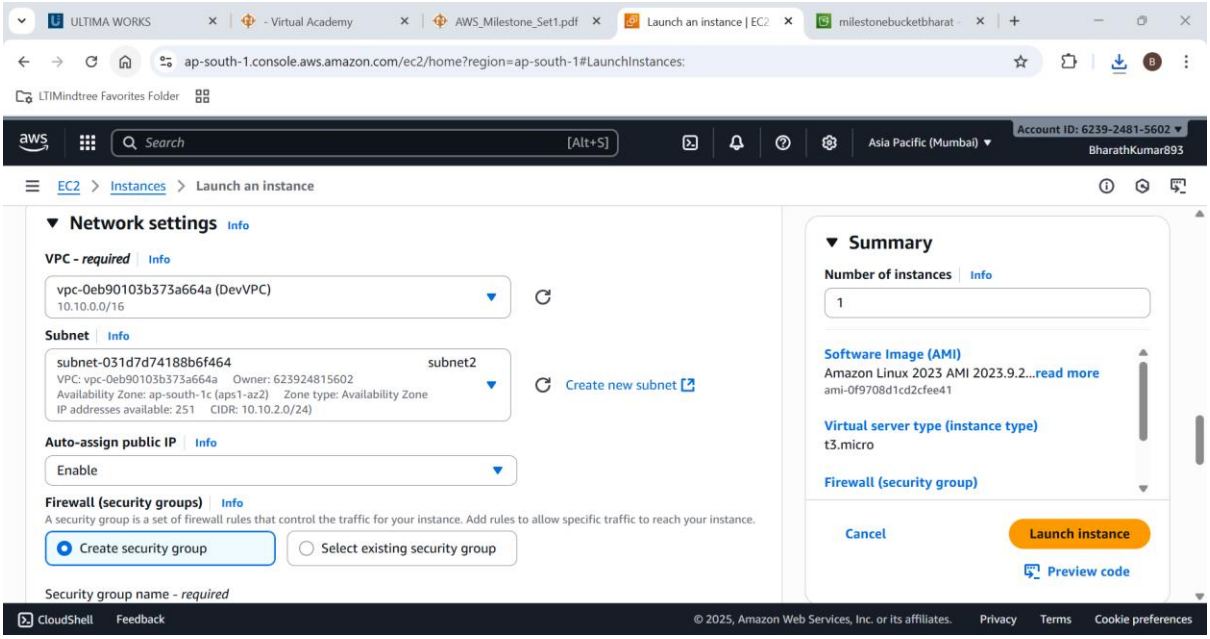


Edit Network settings

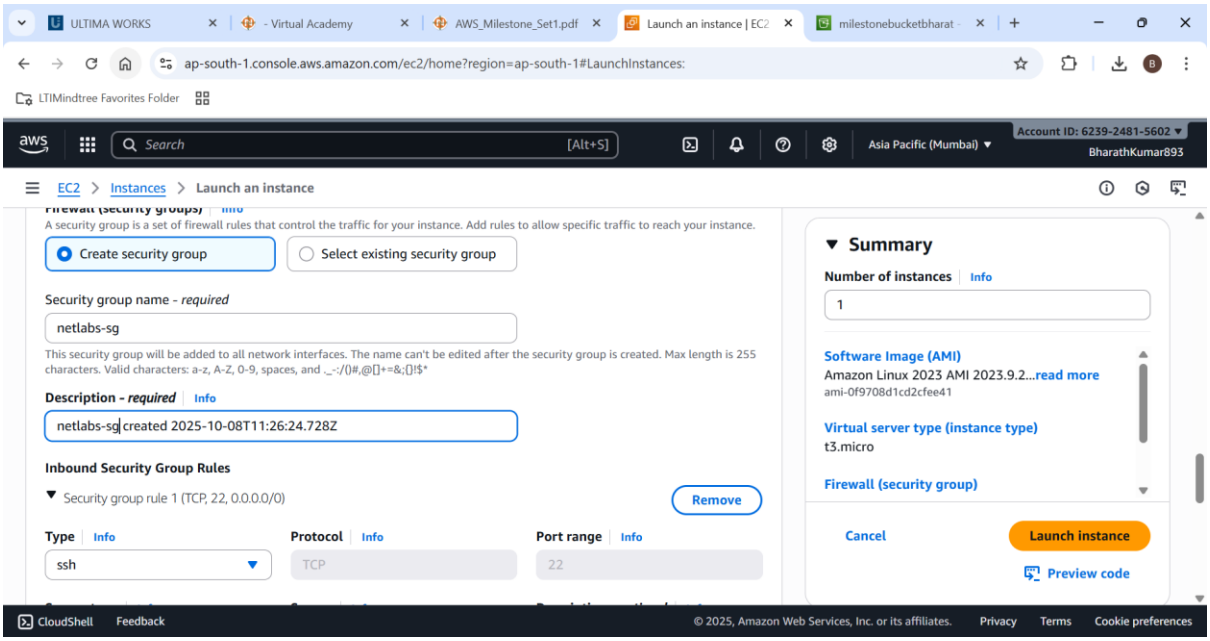
VPC=DevVPC

Subnet=Subnet2

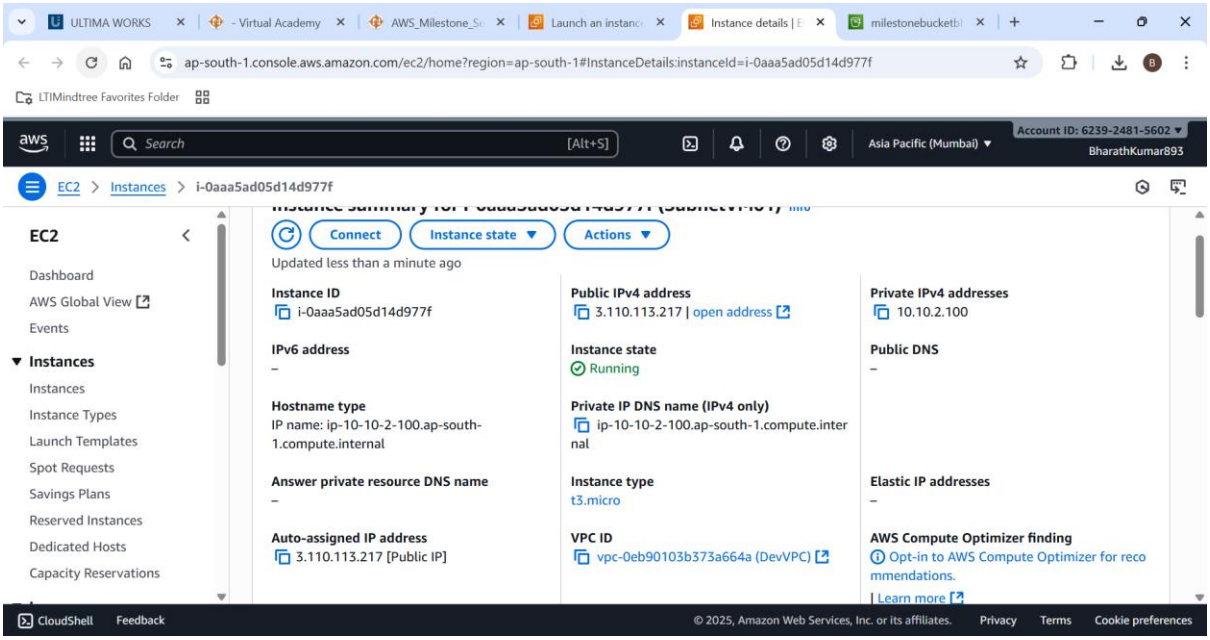
Auto assign Public IP=Enable



Assign New Security group for it



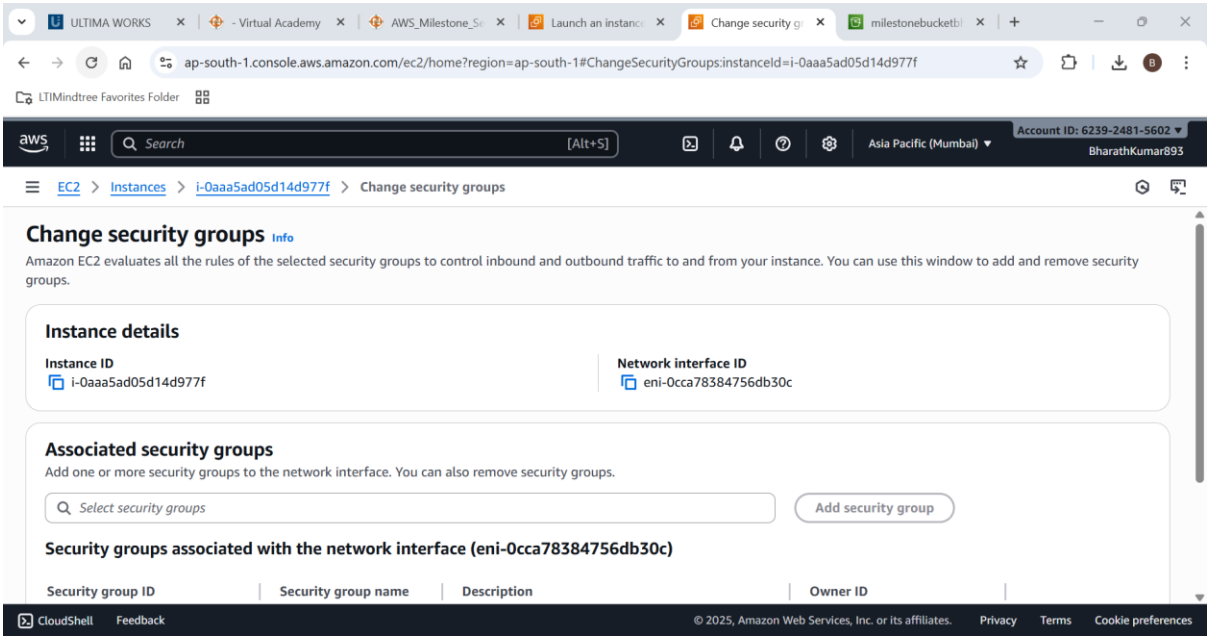
New EC2 intsance is created successfully



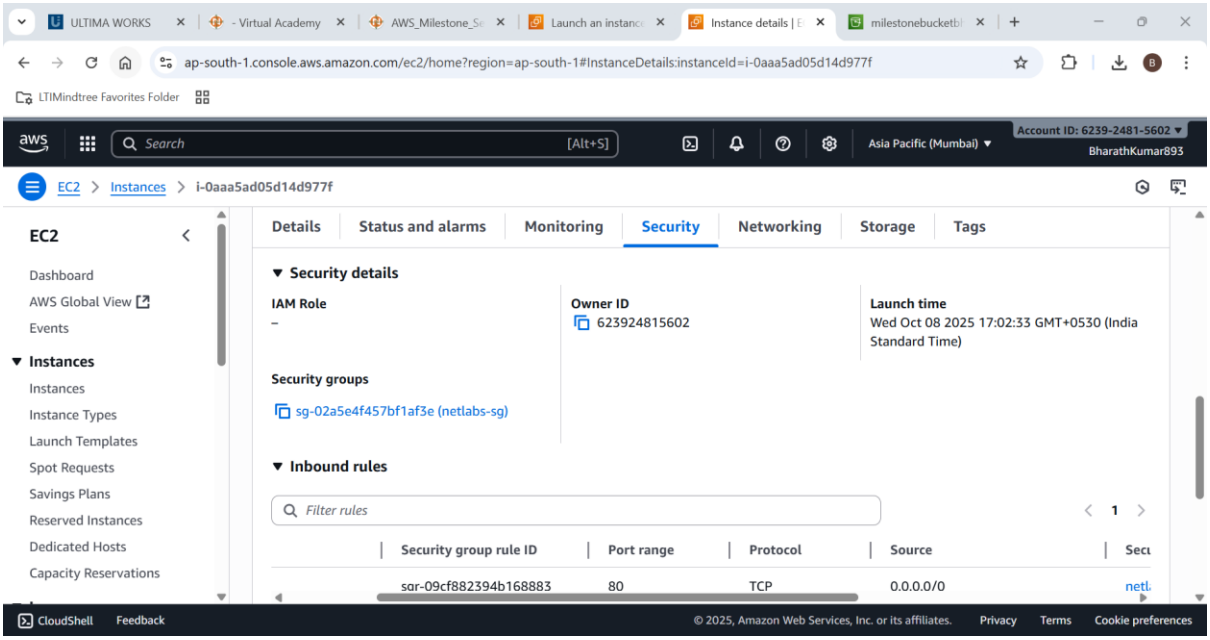
The requested setup has been done successfully.

3. **Create a Security Group**
   - **Name: netlabs-sg**
   - **Define inbound and outbound rules as mentioned below:**
     - **Inbound Rules: Allow RDP (3389), HTTP (80), HTTPS (443)**
     - **Outbound Rules: Deny HTTP (80) and HTTPS (443)**
   - **Associate this Security Group with EC2 in Subnet2 you are creating**

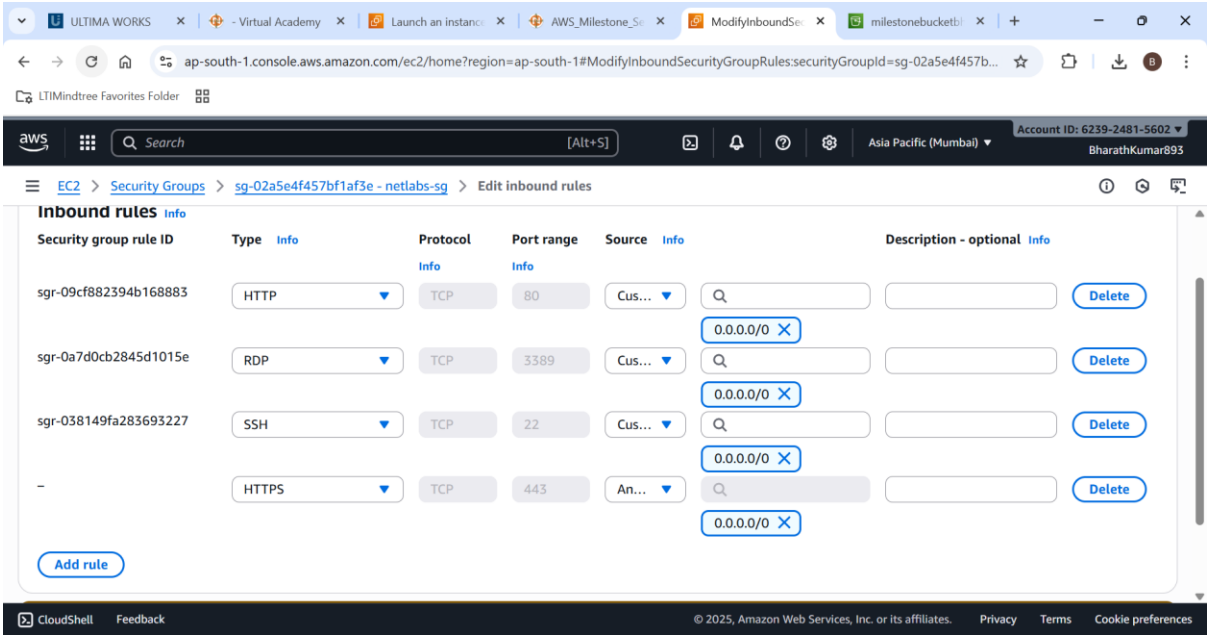
Lets choose the security group we created



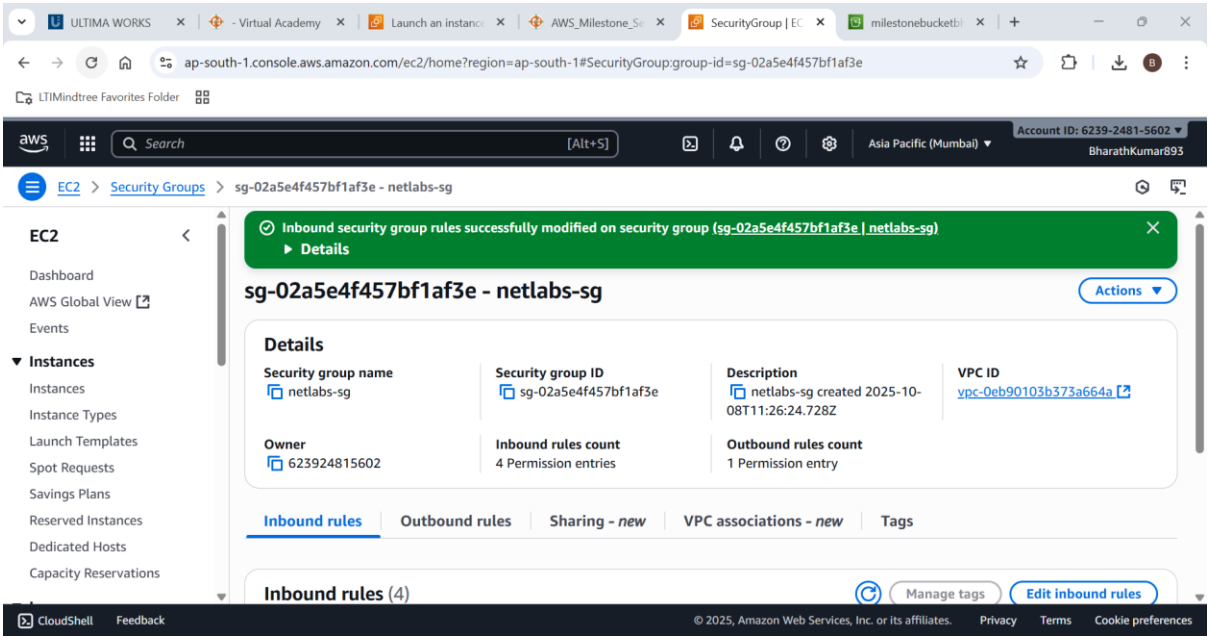
Lets edit the Inbound rules first



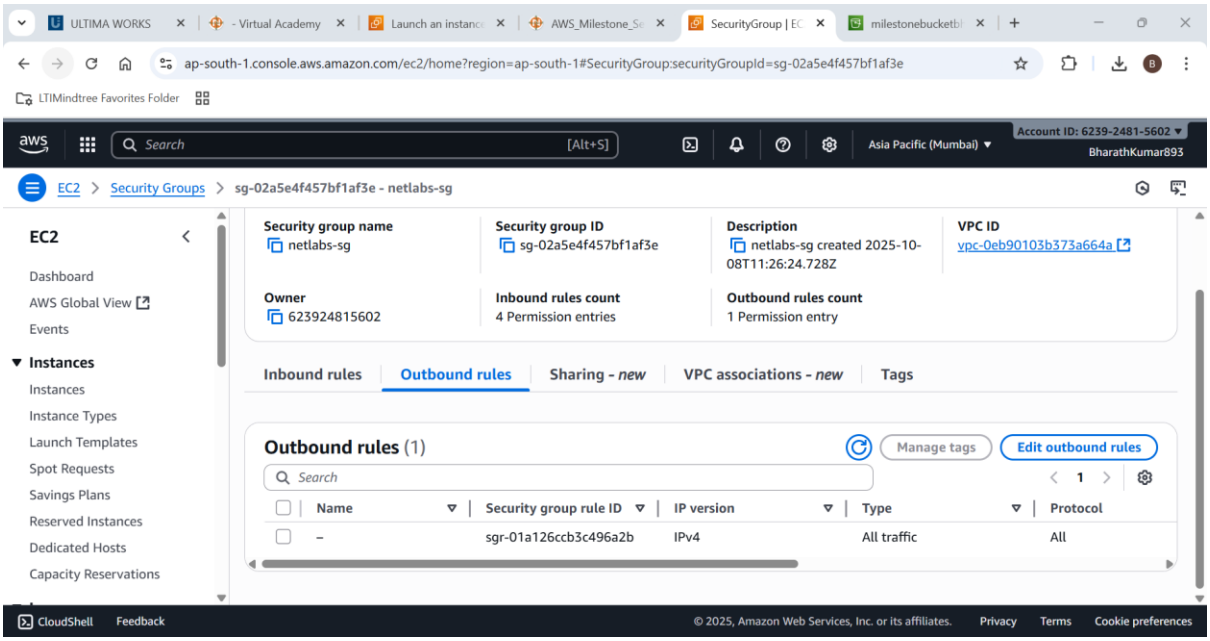
Adding RDP(3389),HTTPS(443),HTTP(80)



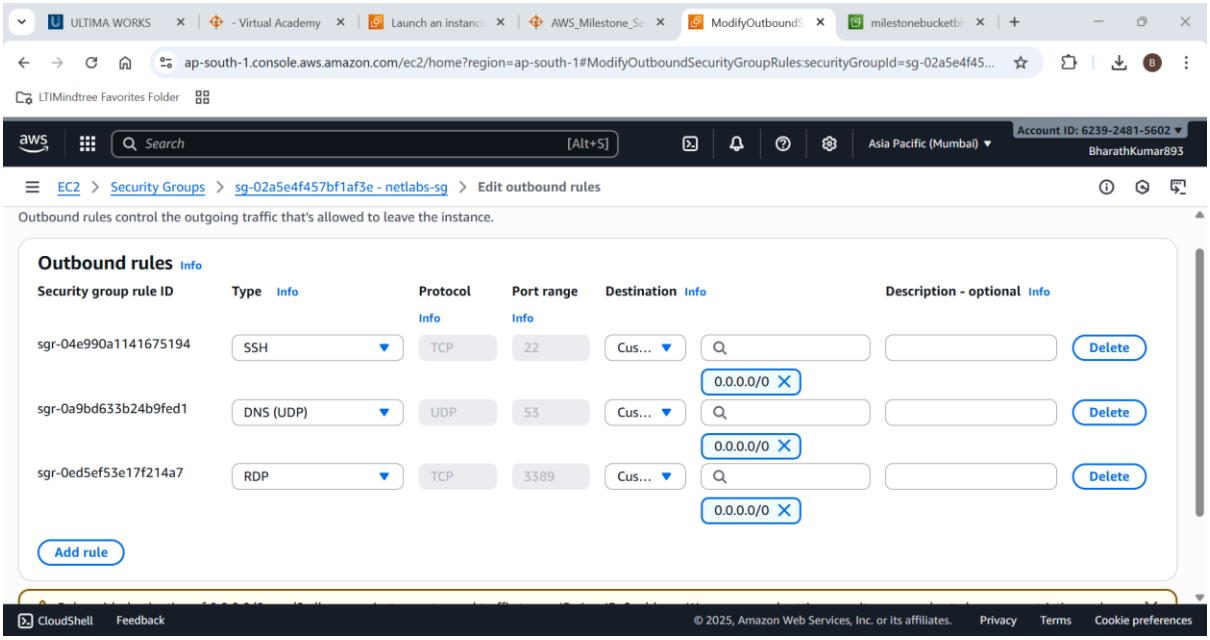
Inbound security is associated to the EC2 successfully



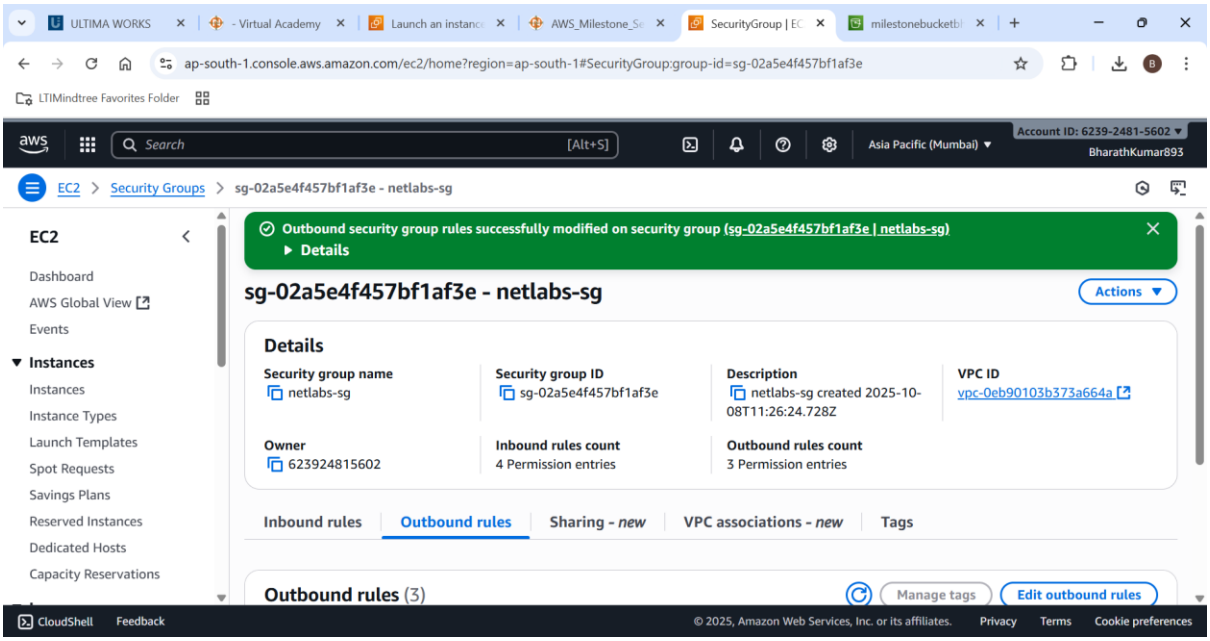
Now lets edit the outbound rules



Deny Http(80) and Https(443) from the EC2 instance by deleting the all traffic and the adding the required access ports except HTTP and HTTPS.

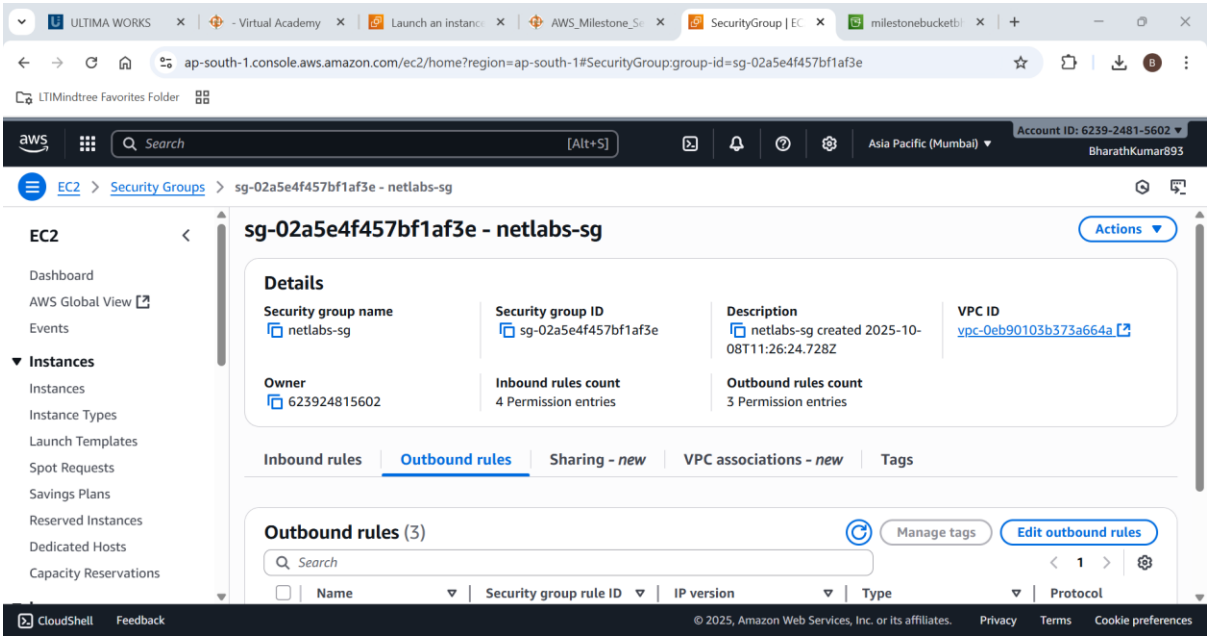


Add it to the EC2 Intsance



Outbound Security has been added to the instance successfully

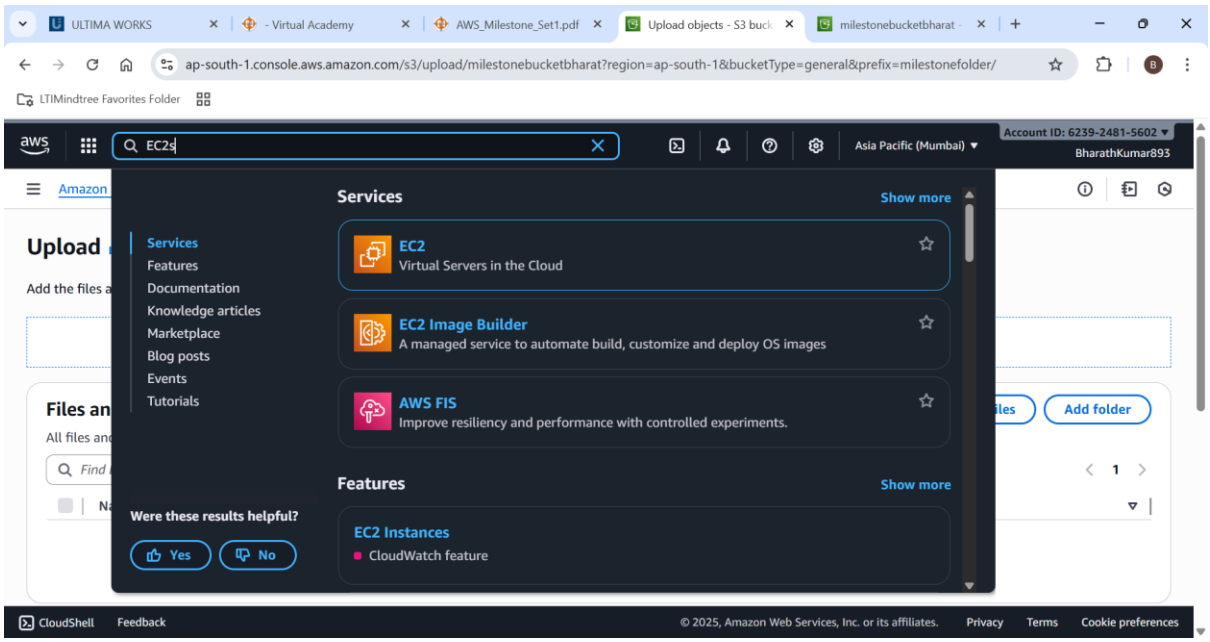
Now lets see in Security group for verification



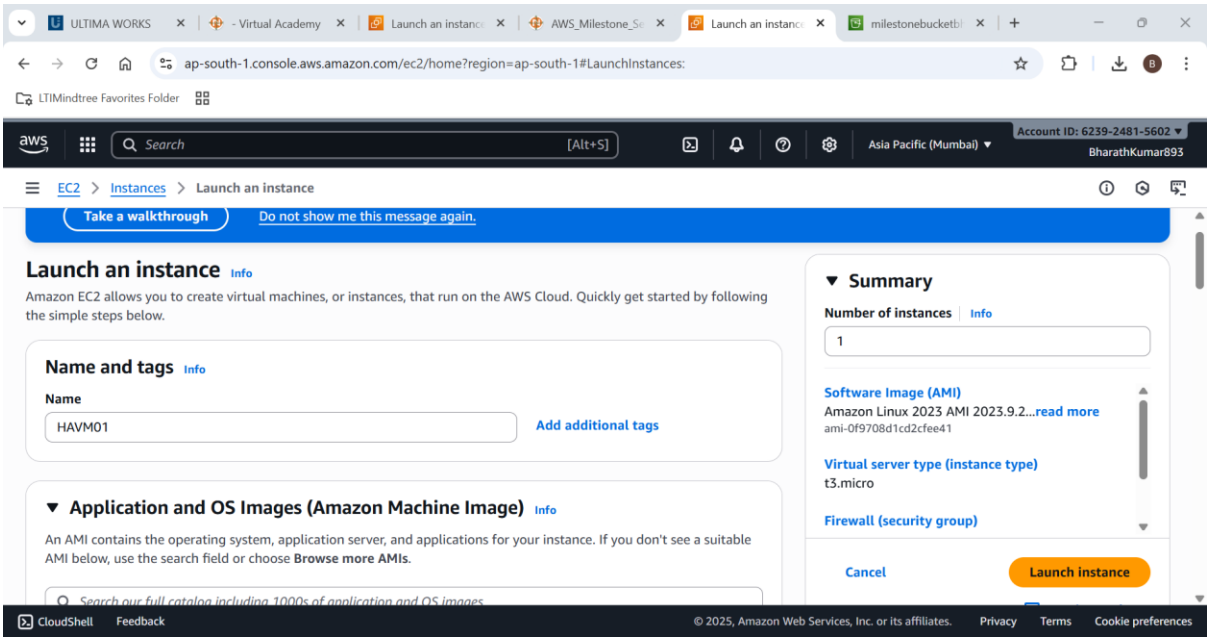
Now you can see there are 4 inbound entries and 4 outbound entries

4. **Create an Availability Zone Setup**
   - **Availability Option: Availability Set equivalent in AWS → Launch multiple instances in different Availability Zones.**
   - **Instance Name: HAVM01**
   - **AMI: Windows Server 2019 Base / Amazon Linux 2023 Kernel-6.1 AMI**
   - **Instance Type: t2.micro(Or t3 micro)**
   - **Authentication: Create Key-Pair**
   - **VPC: DevVPC**
   - **Subnet: Subnet2**
   - **Public IP: Enabled**
   - **Security Group: Basic rules(SSH/HTTP/RDP) : Follow Question 3 for this**
   - **Region: us-east-1 (N. Virginia) or Allowed Region**
   - **Placement: Choose two different Availability Zones in us-east-1 i.e. Subnet1 of selected region for high availability.**

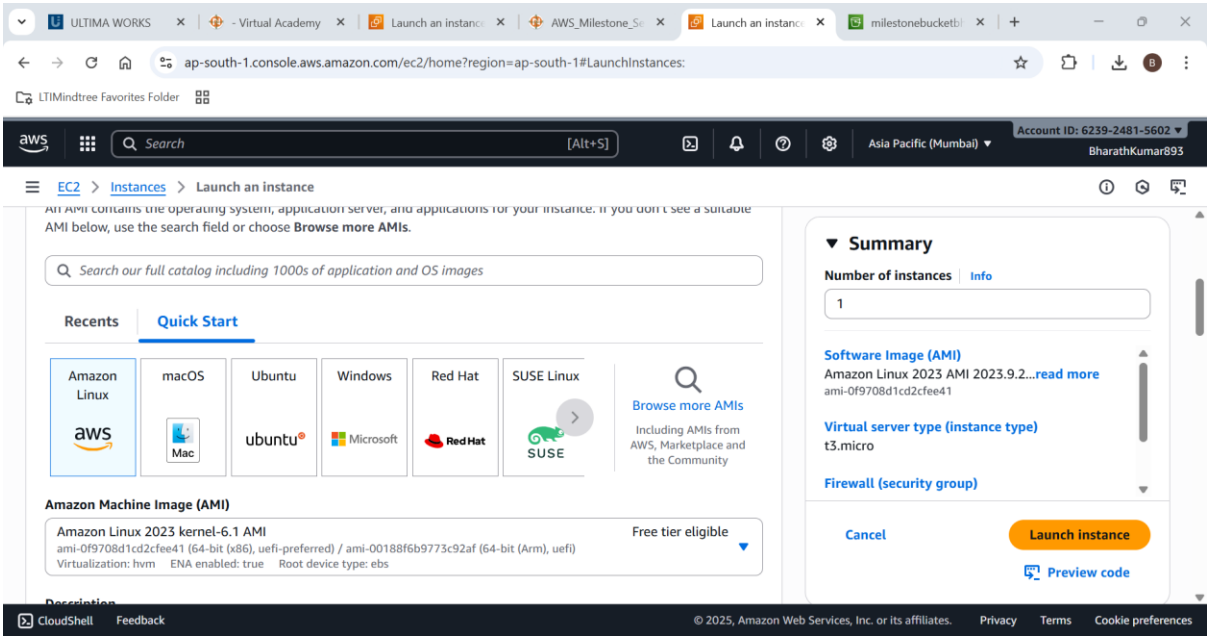
Step 1: lets open EC2 service



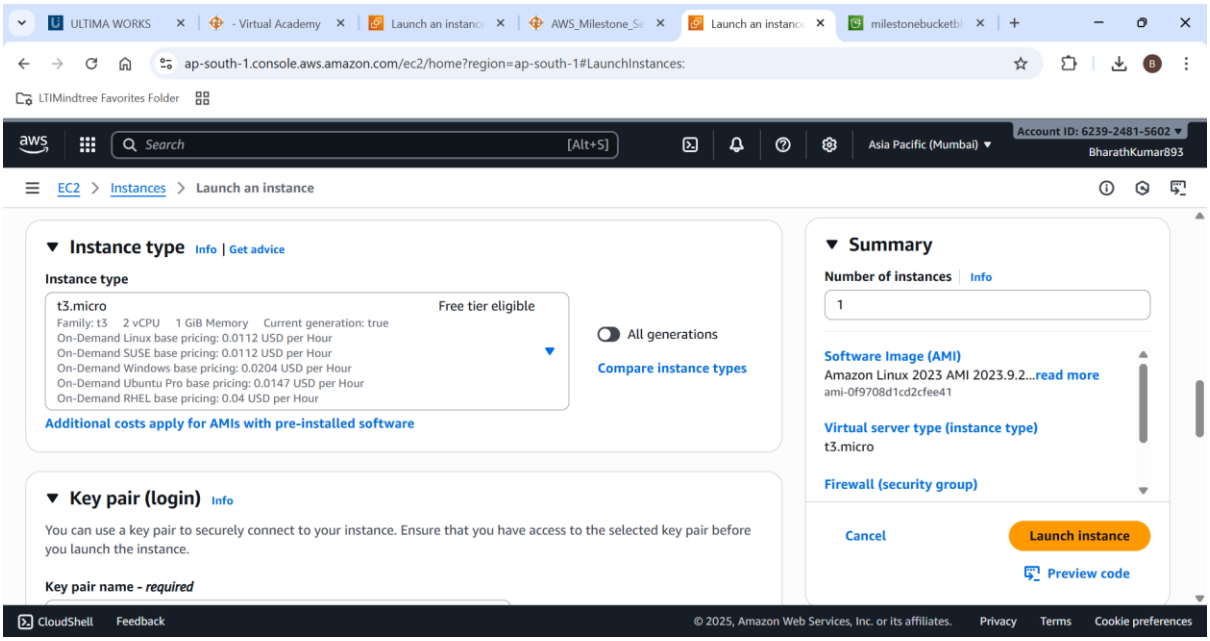
Lets create a EC2 intance with the name HAVM01



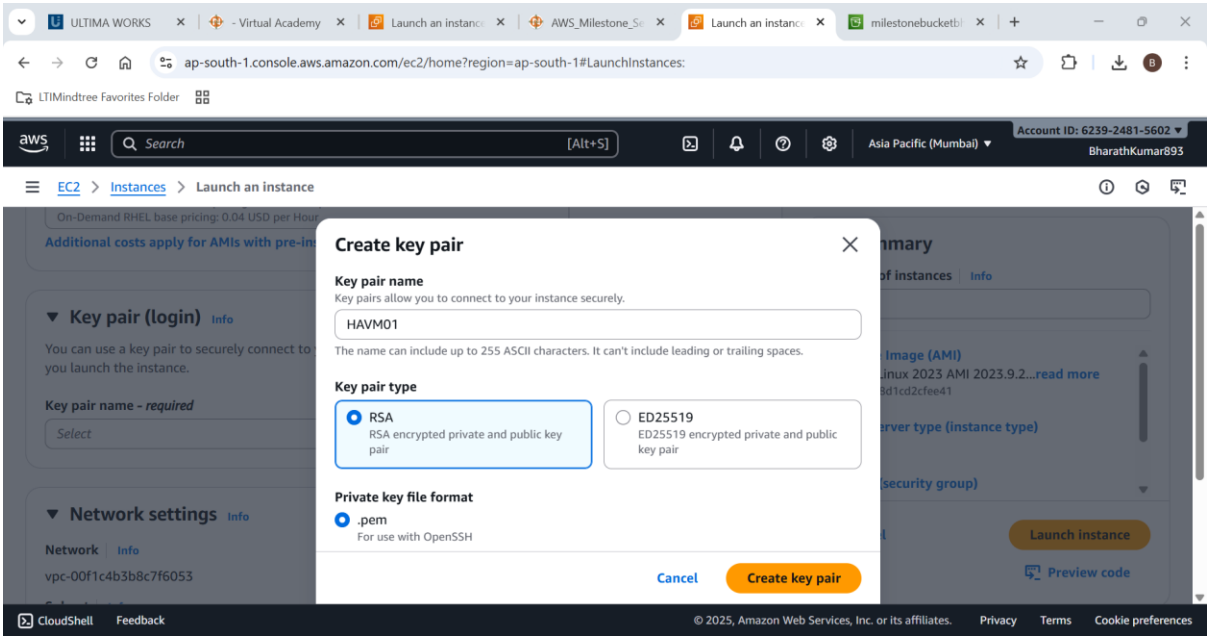
Lets Choose Amazon linux-2023 kernel-6.1 AMI



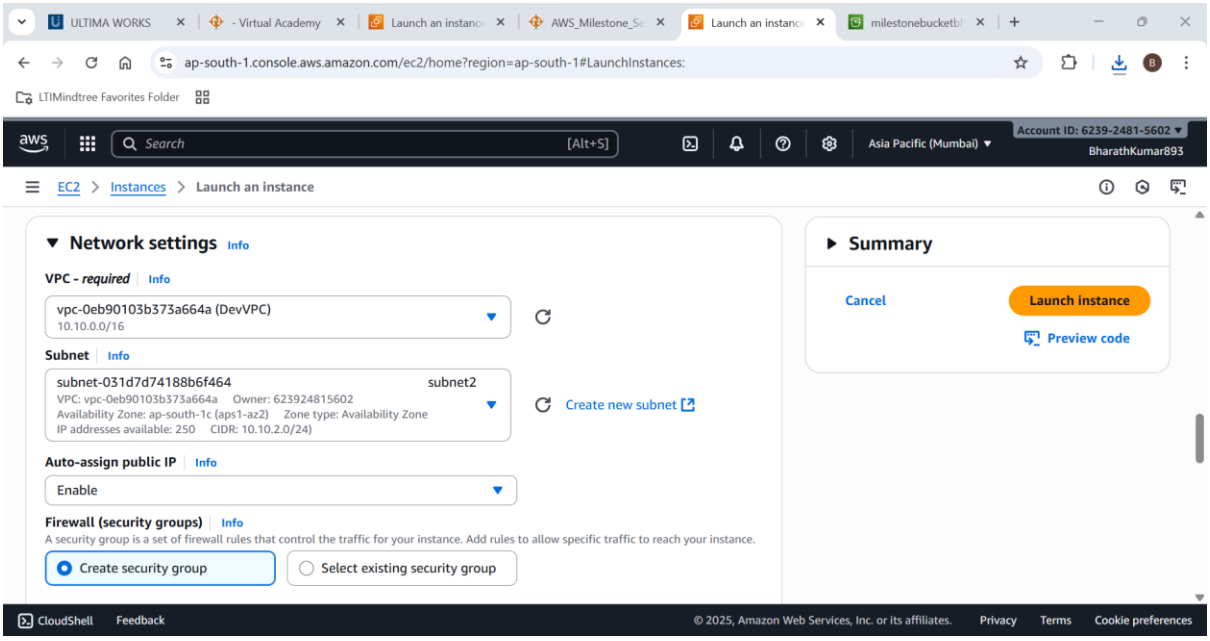
Lets choose t3.micro as instance type



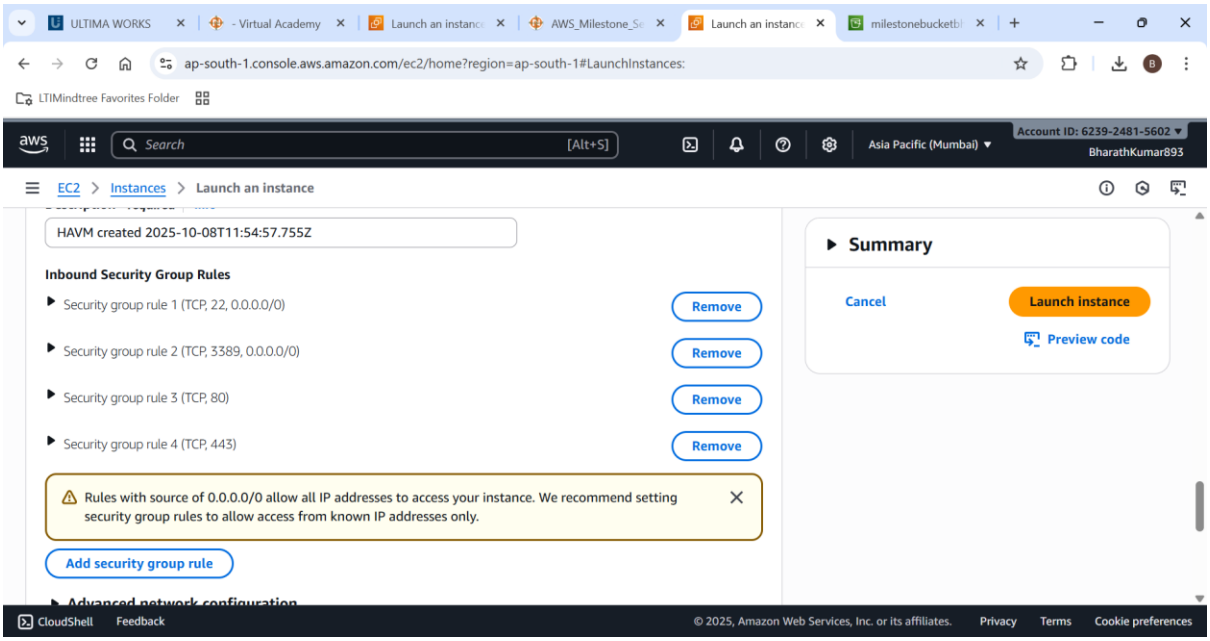
Create a new key value pair.



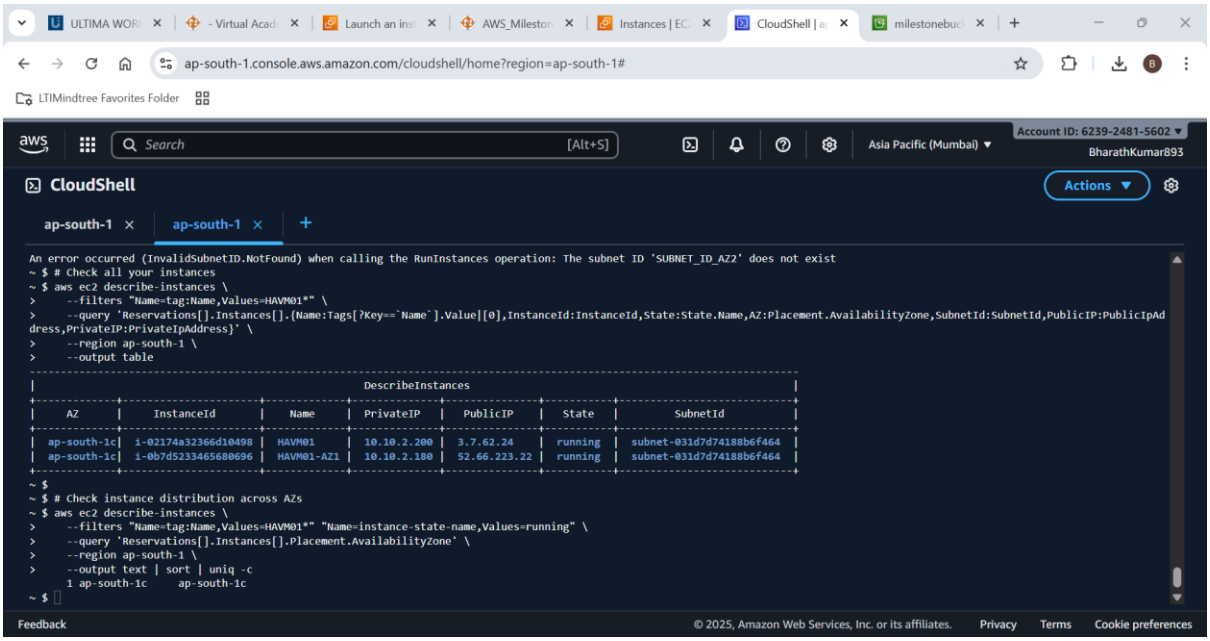
Edit the network settings as
VPC=devVPC
Subet=subnet2
And also enable auto assign public IP



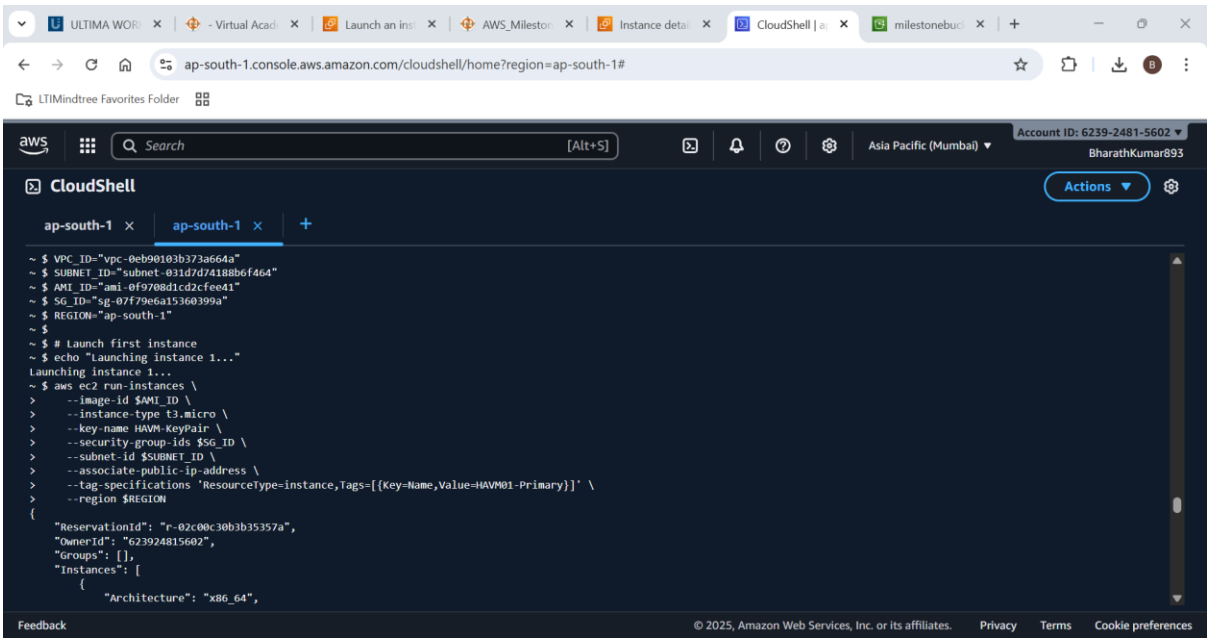
Create a new security adding rdp(3389),http(80),https (443)



Use Cloudshell to create multiple Availabilty zones



Instances is deployed at multiple zones successfully
Thus now lets verify instances connection



The requested has been done and verified successfuly

5. Create an S3 Bucket and Generate Pre-Signed URL

• Create a private S3 bucket (Do not make it Public).

• Enable ACL

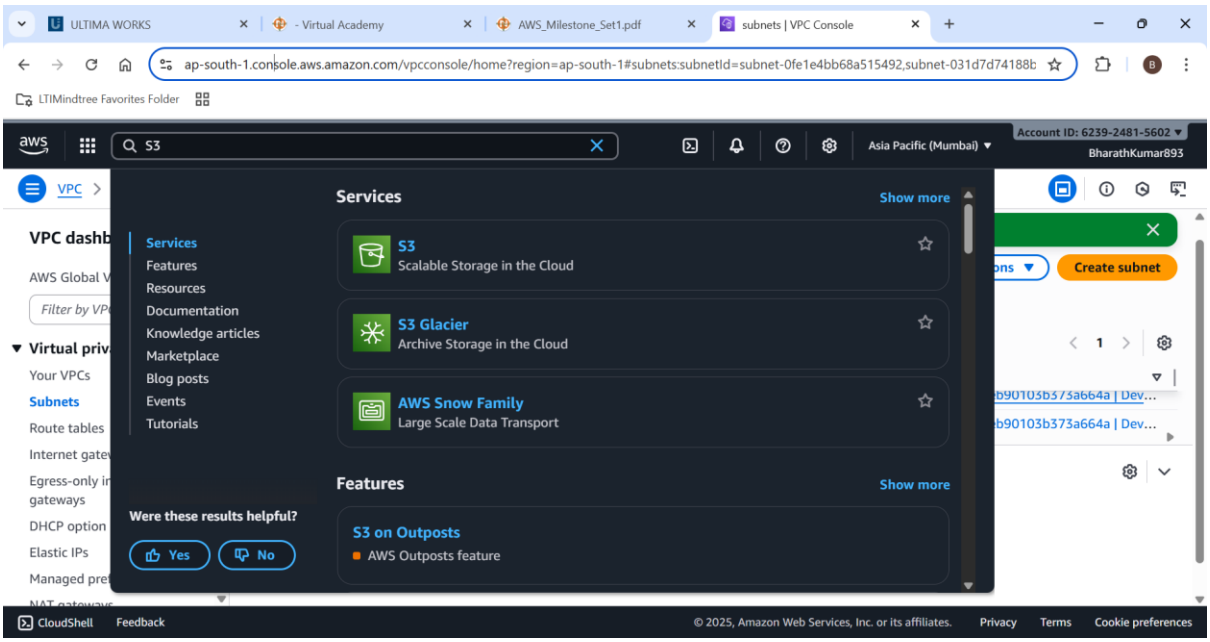
• Create Folder name it container

• Upload a file into a container (folder) inside the bucket.

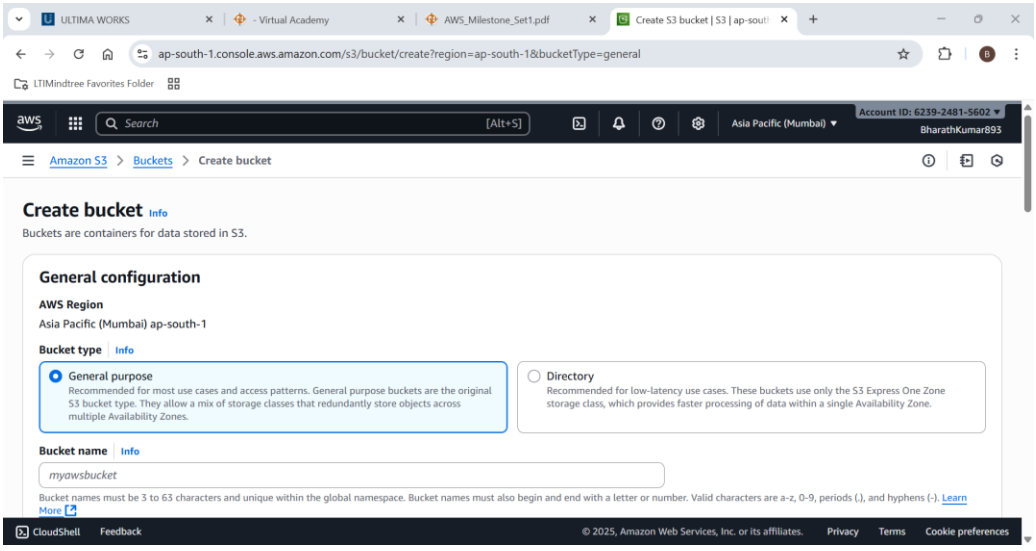
• Use Object Url and check whether you can see it or not, if No. How can you see with object ACL enablement.

• Ensure that the bucket and objects are private by default.

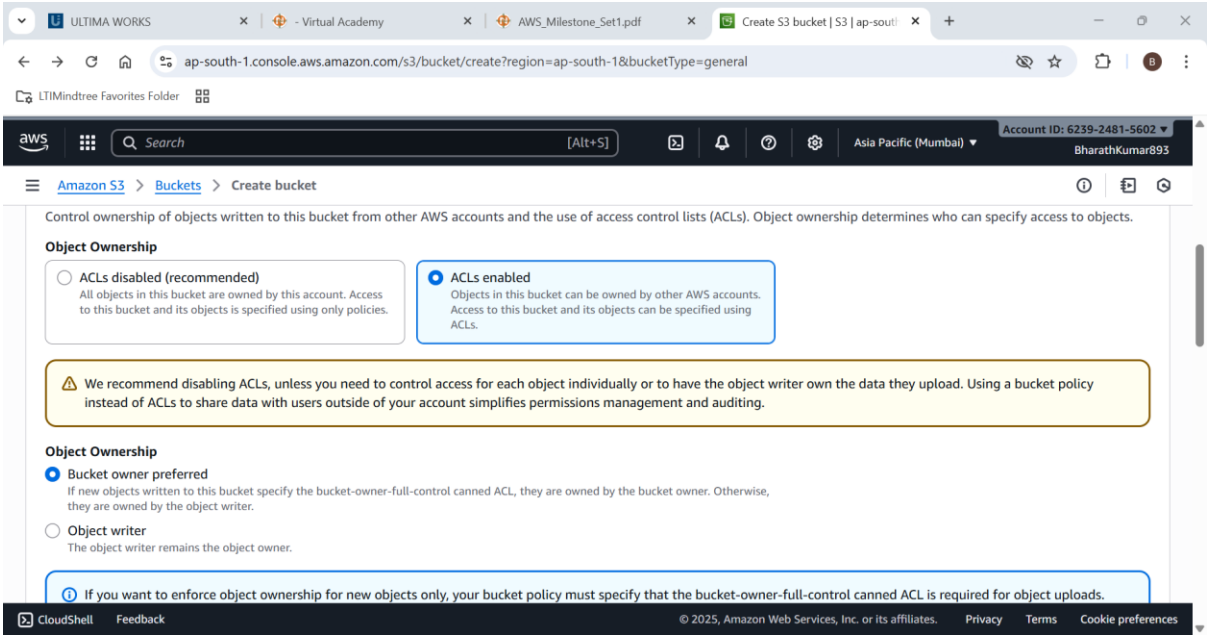
Step 1:Lets open the S3 bucket service



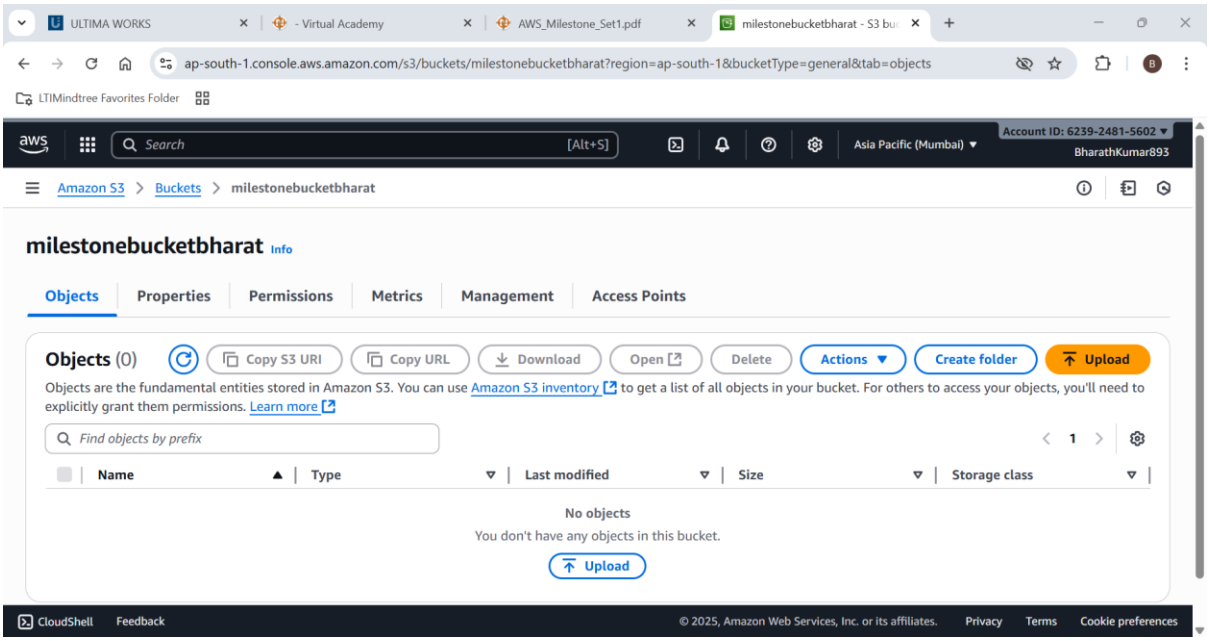
Create a new bucket



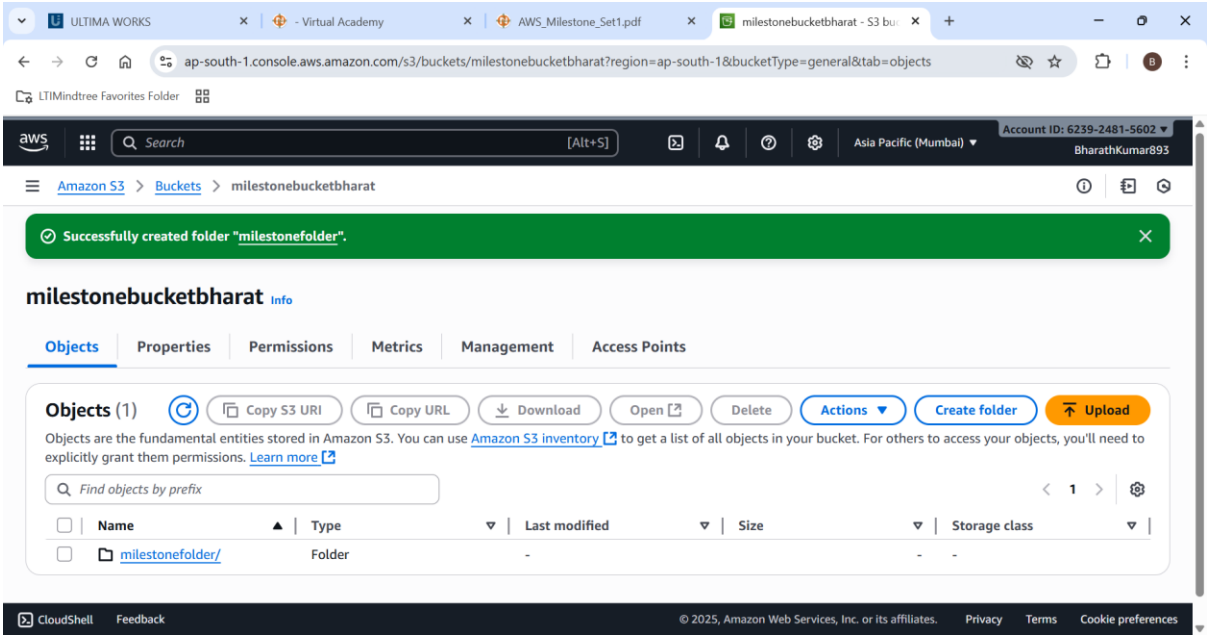
Enabling the ACL



The new bucket is created successfully

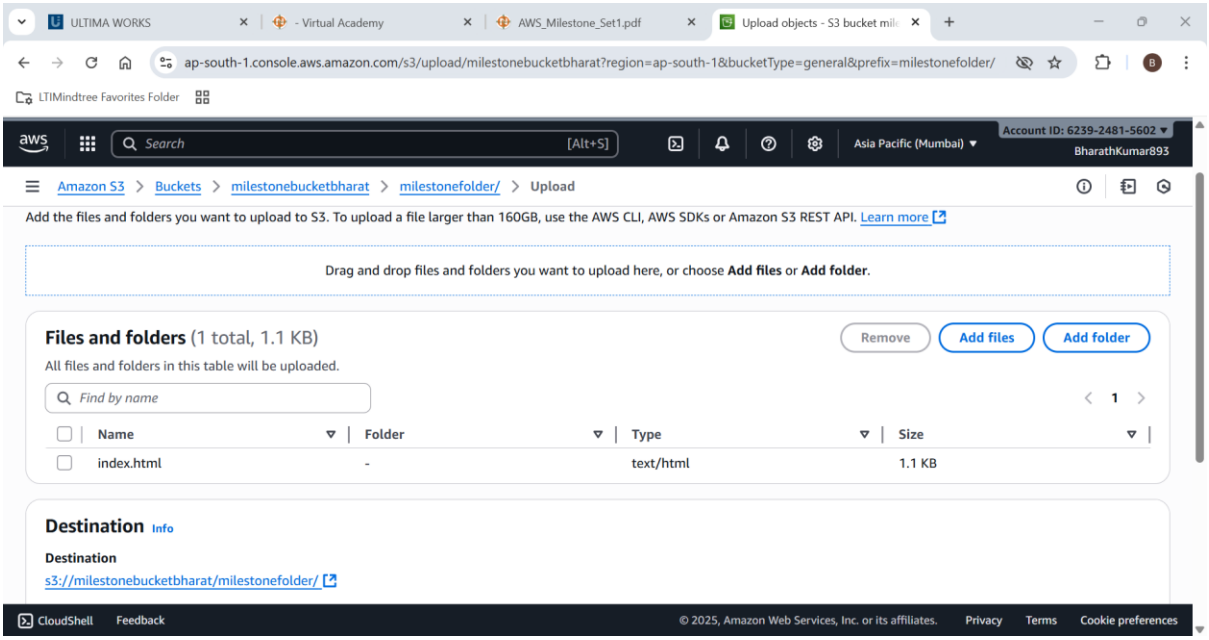


Lets create a folder in it

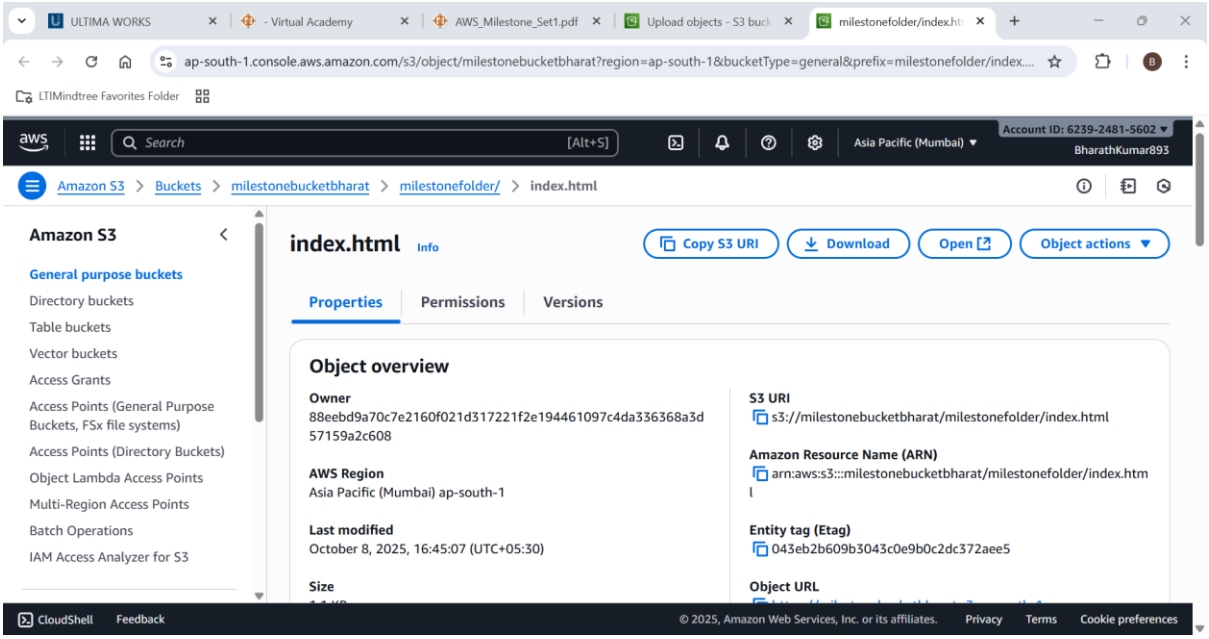


New folder created successfully

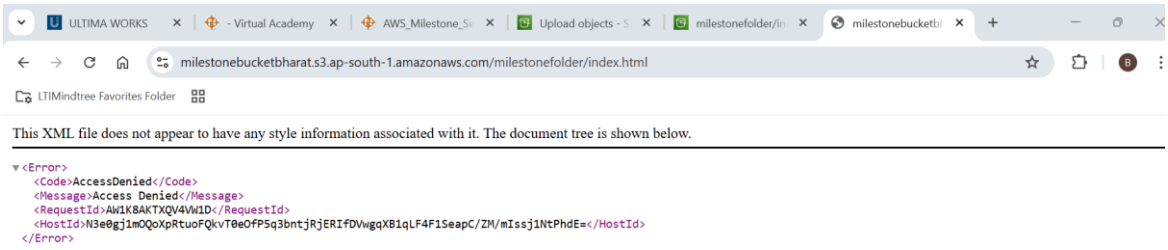
Lets upload a file (index.html)



Lets check whether the the object is accessible to public or not

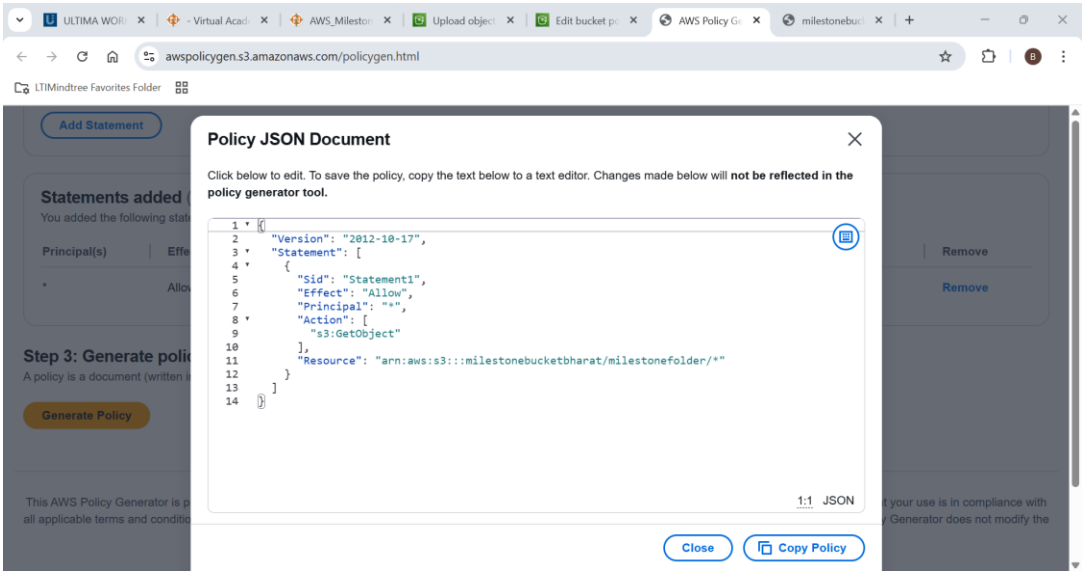


The Object is not accessible to public

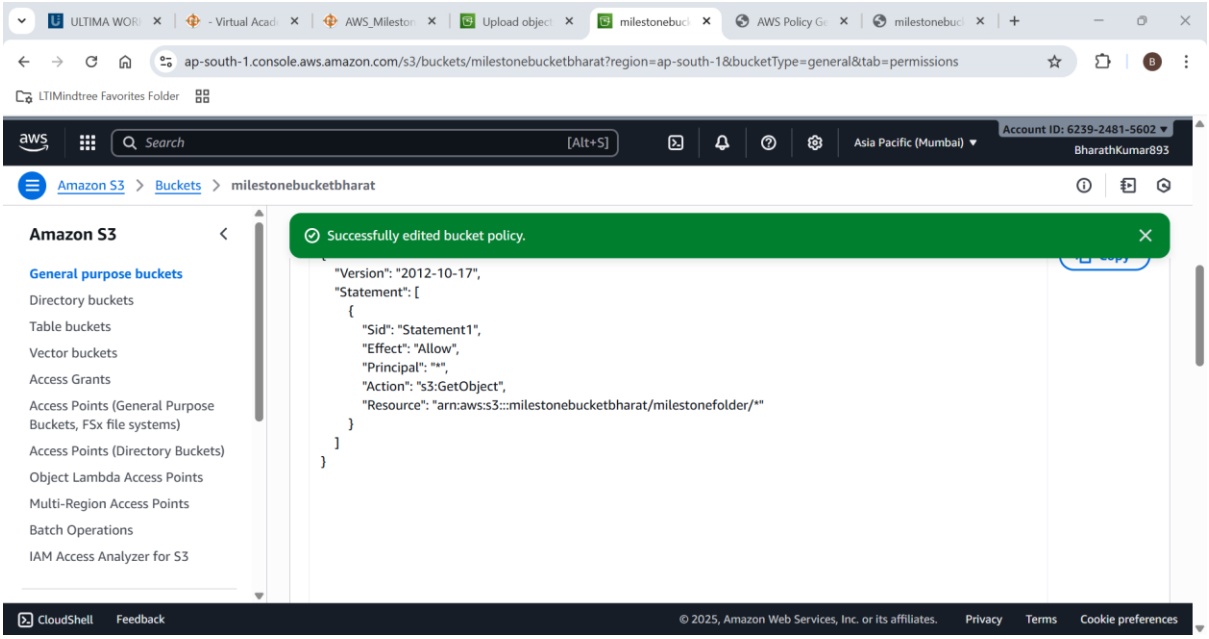


To make it public we need to edit the bucket policy

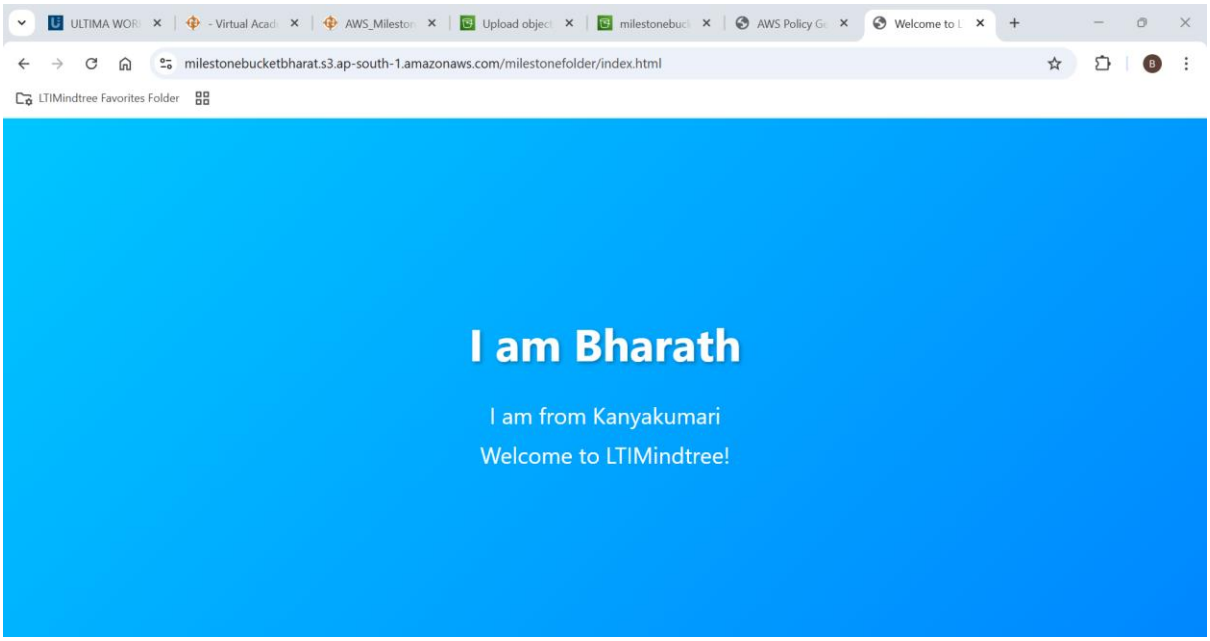
Genarate a Bucket policy



Bucket policy added successfuly



Now lets see whether the object is access to public or not



Now the Object is visible from public source