

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.

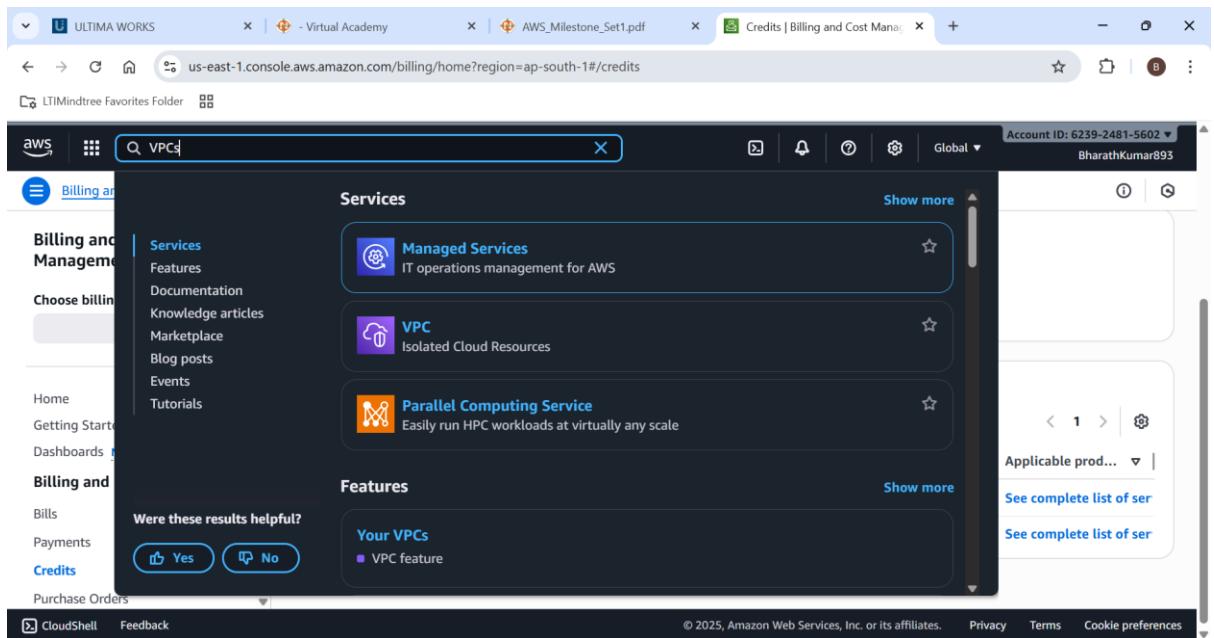
1. Create a VPC

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

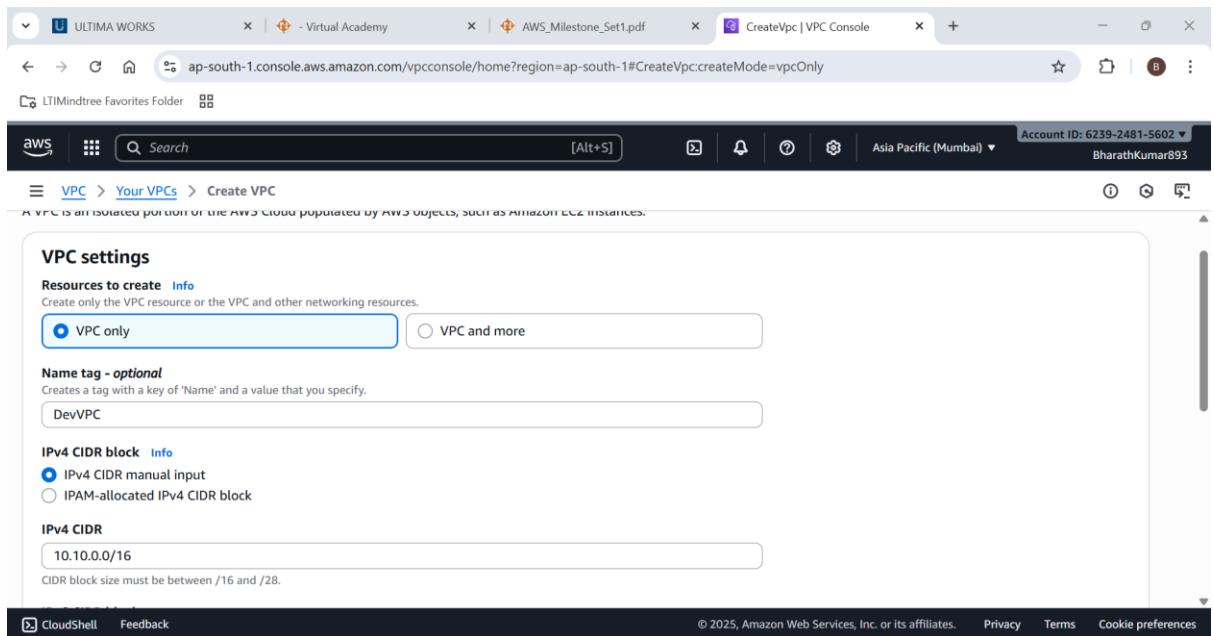
Create two Subnets inside DevVPC:

- **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 settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

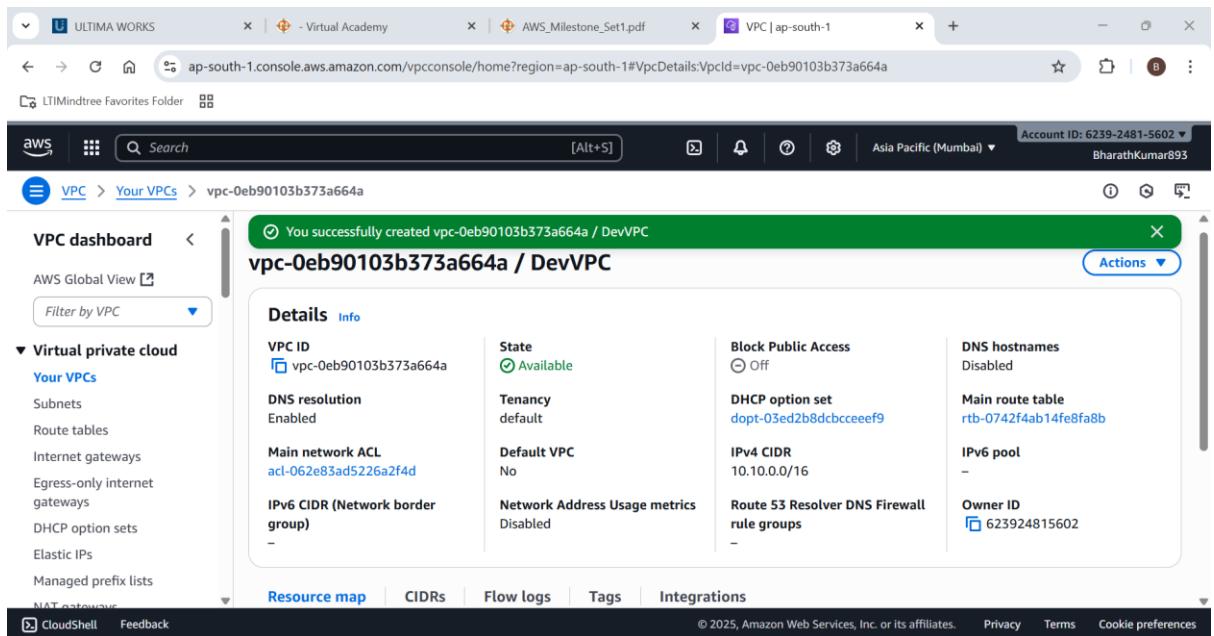
Name tag - **optional**
Creates a tag with a key of 'Name' and a value that you specify.
DevVPC

IPv4 CIDR block [Info](#)
 IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.10.0.0/16

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

VPC Created successfully



You successfully created vpc-0eb90103b373a664a / DevVPC

vpc-0eb90103b373a664a / DevVPC

[Actions](#)

Details Info	
VPC ID	vpc-0eb90103b373a664a
State	Available
Block Public Access	<input type="radio"/> Off
DNS hostnames	Disabled
DNS resolution	Enabled
Tenancy	default
DHCP option set	dopt-03ed2b8dcbeceef9
Main network ACL	acl-062e83ad5226a2f4d
Default VPC	No
IPv4 CIDR	10.10.0.0/16
IPv6 CIDR (Network border group)	-
Network Address Usage metrics	Disabled
Route 53 Resolver DNS Firewall rule groups	-
IPv6 pool	-
Owner ID	623924815602

Resource map | CIDRs | Flow logs | Tags | Integrations

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 3: Lets create 2 subnets for it

Name	Subnet ID	State	VPC
-	subnet-00fd39ffa93b2af16	Available	vpc-00f1c4b3b8c7f6053
Bharathmock2private	subnet-03ae8b4326ced1587	Available	vpc-0713d811a0c1dea45 Bhar...
publichosting1	subnet-05318264c07f20209	Available	vpc-0278333729dfc0515 host...
Bharathmock2public	subnet-0997c3a525e787893	Available	vpc-0713d811a0c1dea45 Bhar...

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

Create a tag with a key of 'Name' and a value that you specify.

subnet1

The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

Asia Pacific (Mumbai) / aps1-az1 (ap-south-1a)

IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

10.10.0.0/16

IPv4 subnet CIDR block
10.10.1.0/24

Tags - optional

Key	Value - optional
Name	subnet1

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

Subnet name: subnet2

Availability Zone: Asia Pacific (Mumbai) / aps1-az2 (ap-south-1c)

IPv4 CIDR block: 10.10.0.0/16

IPv4 subnet CIDR block: 10.10.2.0/24

Subnet 1 and subnet 2 created successfully

You have successfully created 2 subnets: subnet-0fe1e4bb68a515492, subnet-031d7d74188b6f464

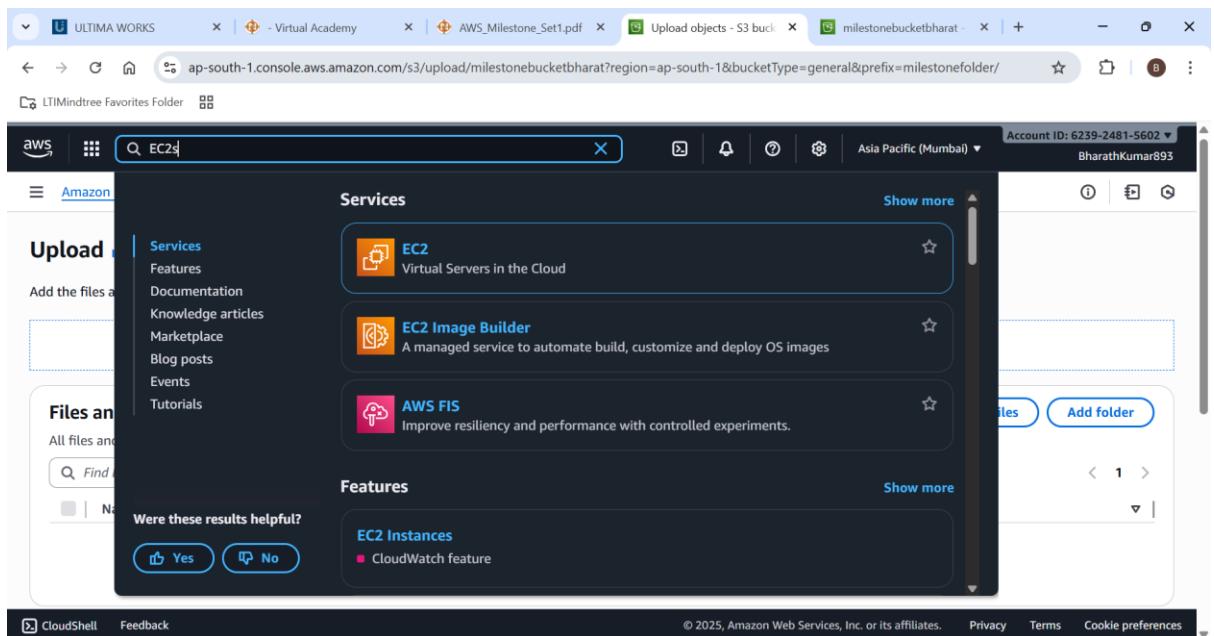
Subnets (2) Info

Name	Subnet ID	State	VPC
subnet1	subnet-0fe1e4bb68a515492	Available	vpc-0eb90103b373a664a Dev...
subnet2	subnet-031d7d74188b6f464	Available	vpc-0eb90103b373a664a Dev...

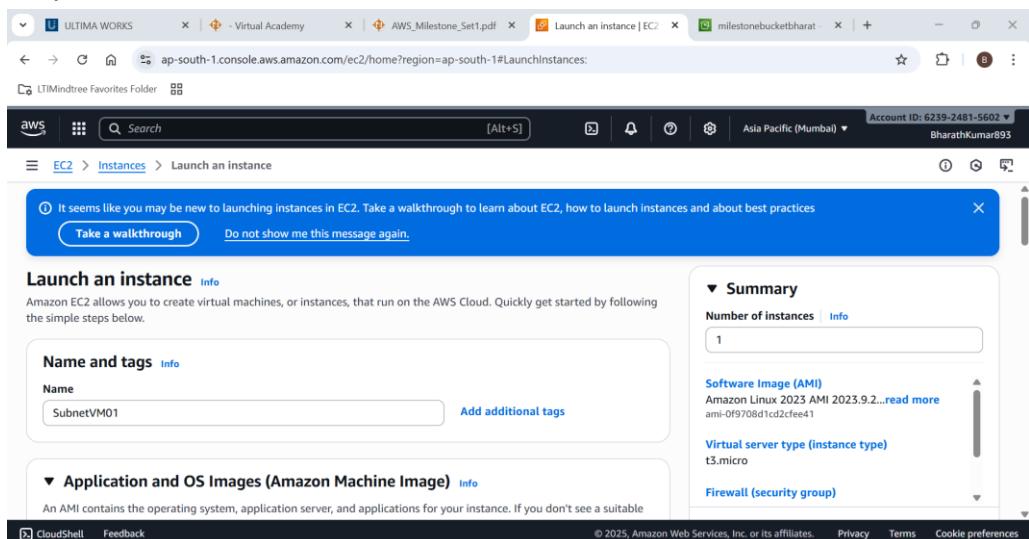
2. Create an EC2 Instance

- **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**

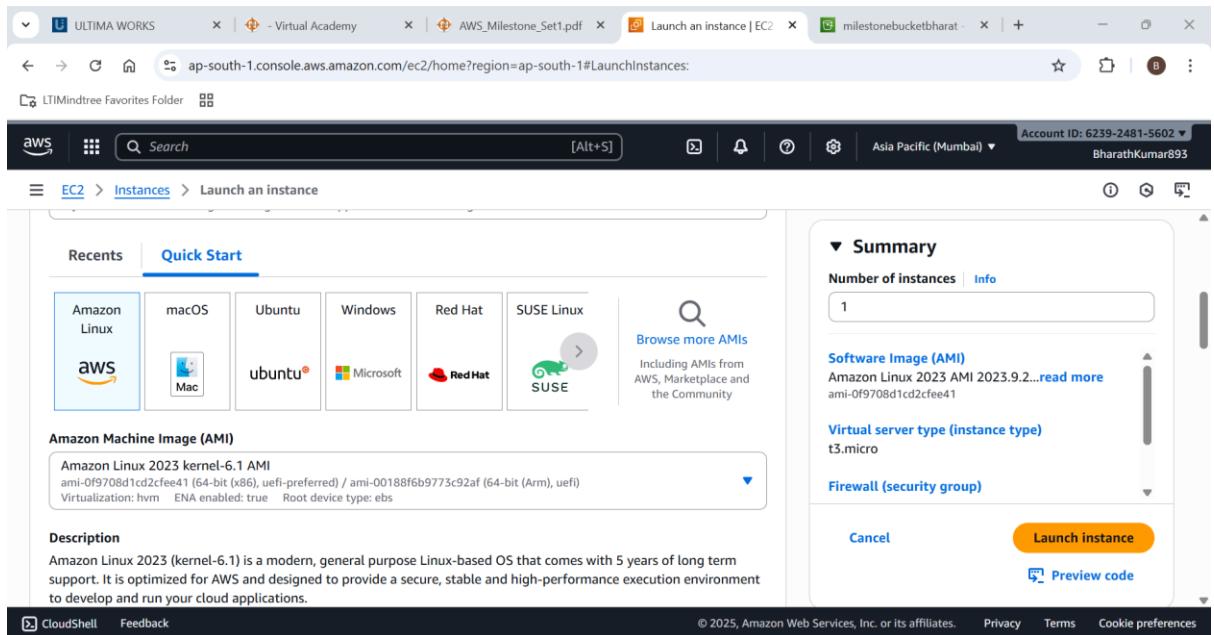
Step 1: Lets open EC2 Service



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

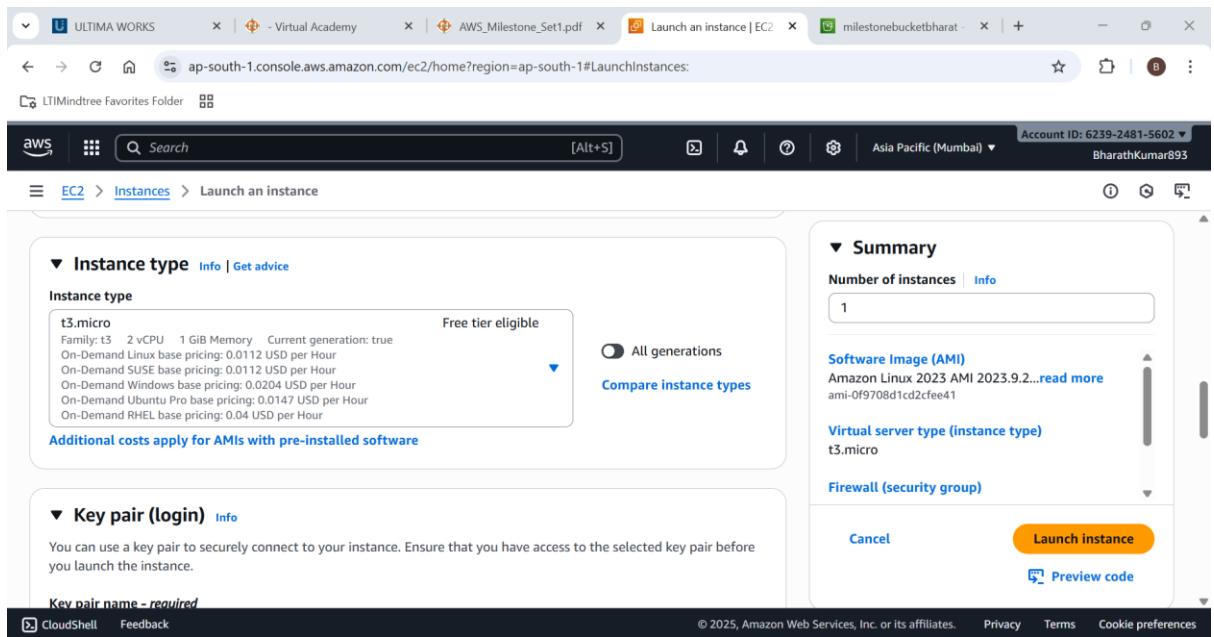


Step 3: Choose Linux / Amazon Linux 2023 Kernel-6.1 AMI as AMI



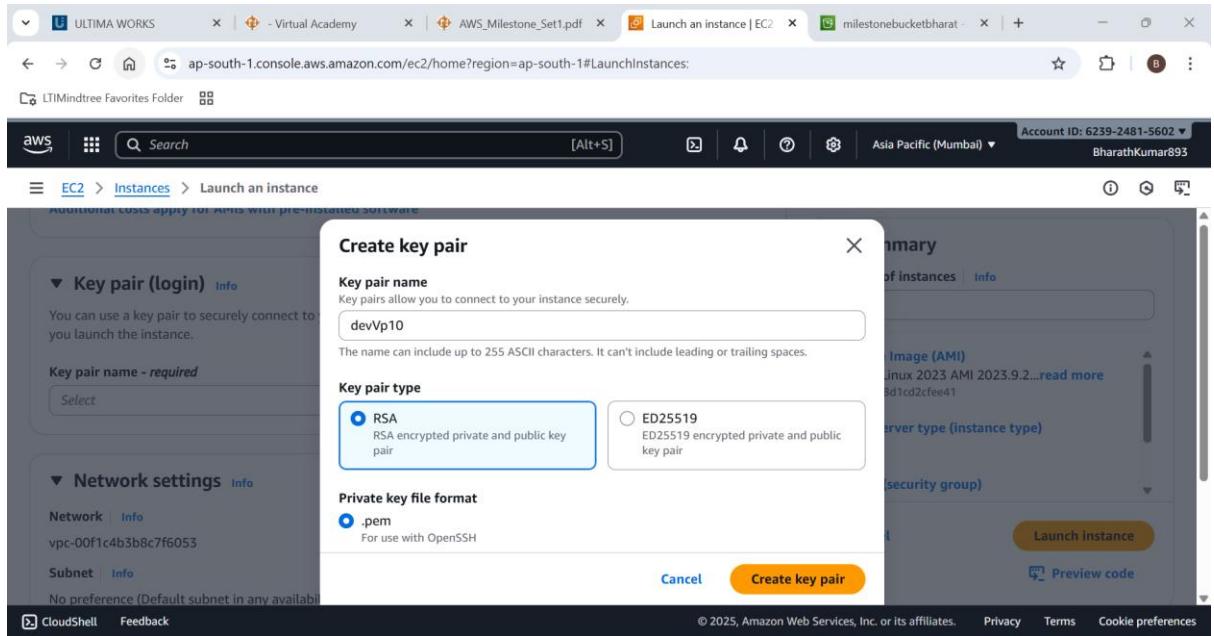
The screenshot shows the AWS Launch an instance page. In the 'Amazon Machine Image (AMI)' section, 'Amazon Linux 2023 kernel-6.1 AMI' is selected. The 'Description' section provides details about the AMI, stating it is a modern, general purpose Linux-based OS with 5 years of long term support. The 'Summary' section on the right shows the selected AMI as 'Amazon Linux 2023 AMI 2023.9.2...', the 'Virtual server type (instance type)' as 't3.micro', and the 'Launch instance' button.

Step 4: Choose instance type as t3.micro



The screenshot shows the AWS Launch an instance page. In the 'Instance type' section, 't3.micro' is selected. The 'Additional costs apply for AMIs with pre-installed software' note is visible. The 'Summary' section on the right shows the selected AMI as 'Amazon Linux 2023 AMI 2023.9.2...', the 'Virtual server type (instance type)' as 't3.micro', and the 'Launch instance' button.

Create a new key-value pair



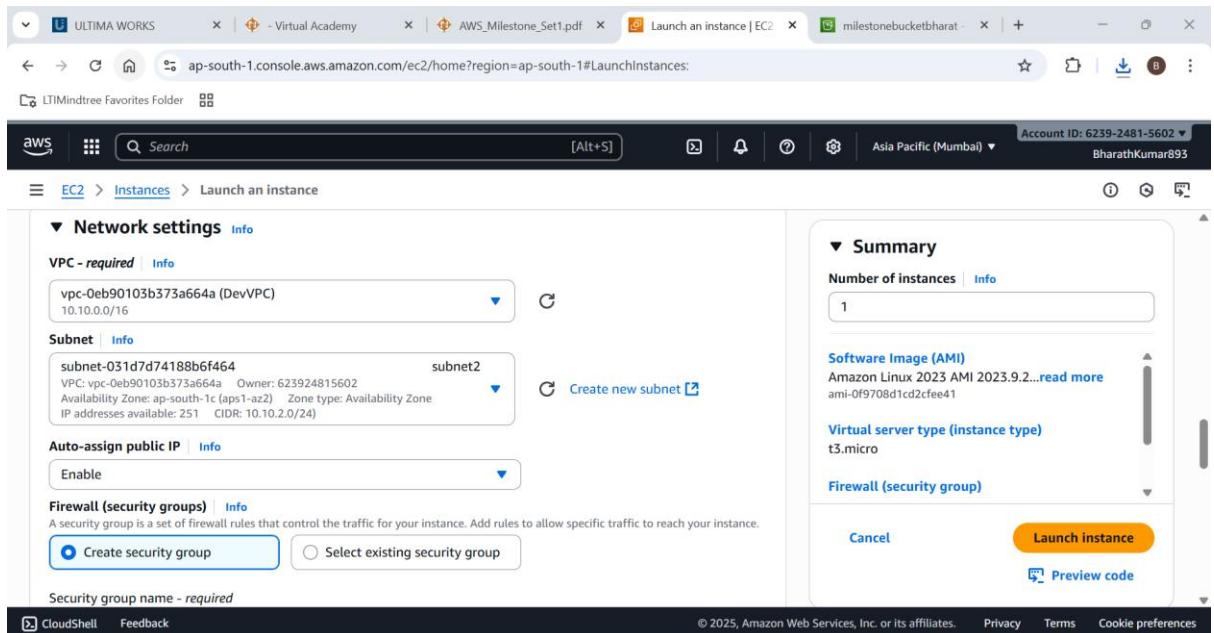
The screenshot shows the AWS EC2 'Launch an instance' wizard. The 'Create key pair' step is open, prompting for a key pair name ('devVp10') and type ('RSA'). The 'Private key file format' is set to '.pem'. The background shows the instance configuration: AMI (Ubuntu 20.04 LTS (HVM)), instance type (t3.micro), and security group (Subnet2). The 'Launch instance' button is visible at the bottom right.

Edit Network settings

VPC=DevVPC

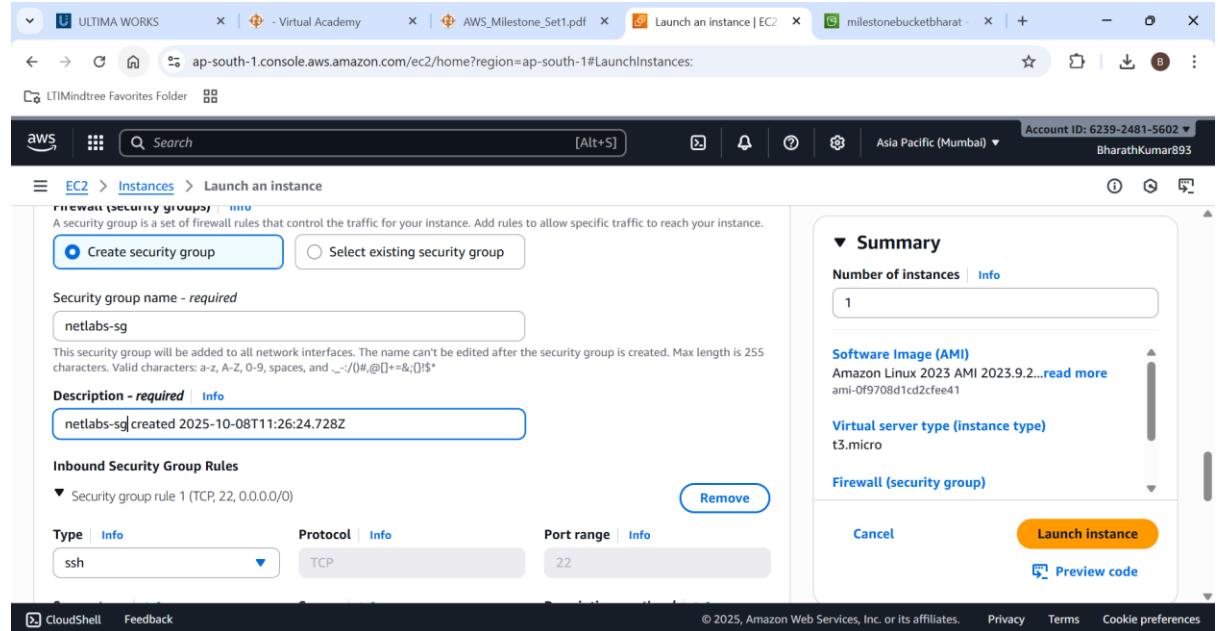
Subnet=Subnet2

Auto assign Public IP=Enable



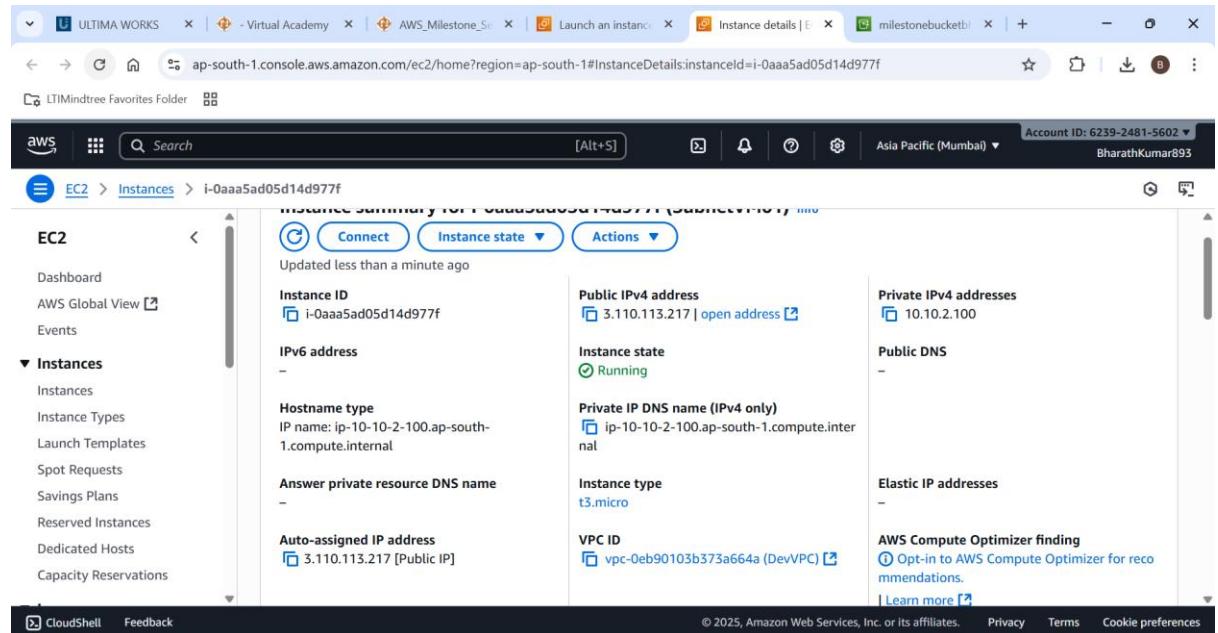
The screenshot shows the AWS EC2 'Launch an instance' wizard. The 'Network settings' step is open, showing the selected VPC (vpc-0eb90103b373a664a) and subnet (subnet2). The 'Auto-assign public IP' is set to 'Enable'. The background shows the instance configuration: AMI (Ubuntu 20.04 LTS (HVM)), instance type (t3.micro), and security group (Subnet2). The 'Launch instance' button is visible at the bottom right.

Assign New Security group for it



The screenshot shows the AWS EC2 'Launch an instance' wizard. The 'Create security group' tab is selected. A new security group named 'netlabs-sg' is being created. The 'Inbound Security Group Rules' section contains one rule: 'Security group rule 1 (TCP, 22, 0.0.0.0/0)' with 'ssh' as the type and 'TCP' as the protocol. The 'Port range' is set to 22. The 'Summary' panel on the right shows 1 instance, the AMI 'Amazon Linux 2023 AMI 2023.9.2...', the instance type 't3.micro', and a 'Launch instance' button.

New EC2 instance is created successfully



The screenshot shows the 'Instance details' page for the instance 'i-0aaa5ad05d14d977f'. The instance is in a 'Running' state with a public IP of 3.110.113.217 and a private IP of 10.10.2.100. It is associated with the VPC ID 'vpc-0eb90103b373a664a'. The page also shows the instance type 't3.micro' and the private DNS name 'ip-10-10-2-100.ap-south-1.compute.internal'.

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)

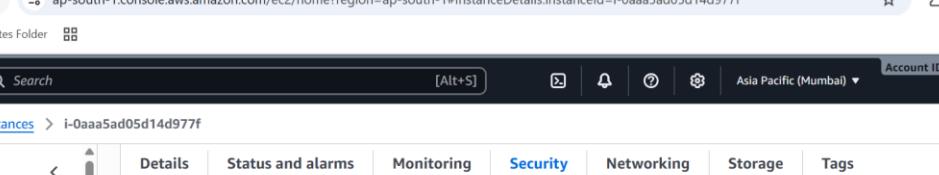
- Associate this Security Group with EC2 in Subnet2 you are creating

Lets choose the security group we created

The screenshot shows the AWS EC2 'Change security groups' interface. At the top, the instance ID is i-Oaaa5ad05d14d977f. The 'Associated security groups' section shows an empty list. The 'Security groups associated with the network interface (eni-Occa78384756db30c)' section lists the following security groups:

Security group ID	Security group name	Description	Owner ID
sg-01234567890abcdef	MySecurityGroup	My security group description	123456789012

Lets edit the Inbound rules first

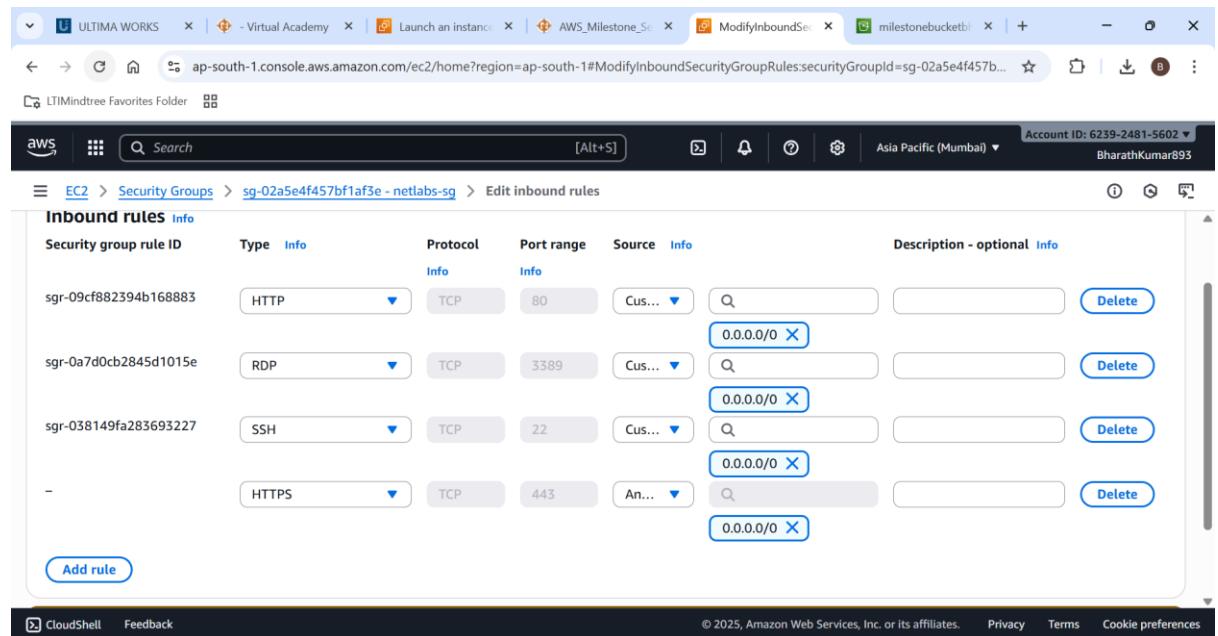


The screenshot shows the AWS EC2 Instances page for an instance with ID `i-0aaa5ad05d14d977f`. The Security tab is selected, displaying the following details:

- Owner ID:** 623924815602
- Launch time:** Wed Oct 08 2025 17:02:33 GMT+0530 (India Standard Time)
- Security groups:** sg-02a5e4f457bf1af3e (netlabs-sg)

The Inbound rules table is empty.

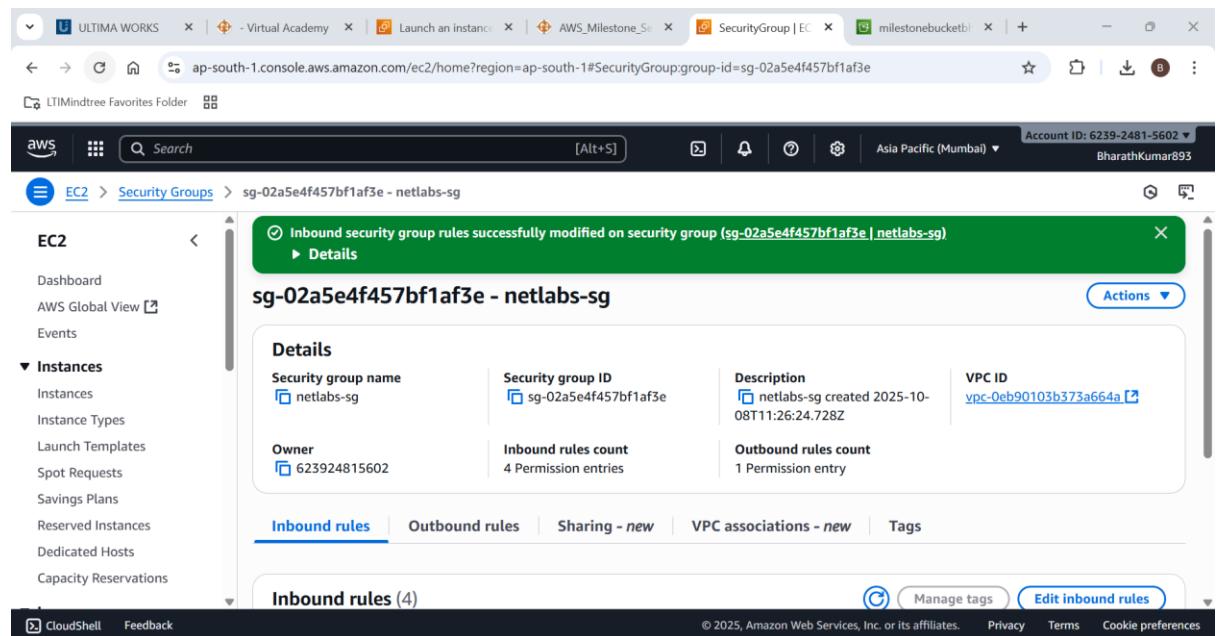
Adding RDP(3389), HTTPS(443), HTTP(80)



The screenshot shows the AWS EC2 Security Groups Inbound rules configuration page. The security group is named 'sg-02a5e4f457bf1af3e - netlabs-sg'. The table lists the following inbound rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-09cf882394b168883	HTTP	TCP	80	Custom (Cus...)	0.0.0.0/0
sgr-0a7d0cb2845d1015e	RDP	TCP	3389	Custom (Cus...)	0.0.0.0/0
sgr-038149fa283693227	SSH	TCP	22	Custom (Cus...)	0.0.0.0/0
-	HTTPS	TCP	443	Anywhere (An...)	0.0.0.0/0

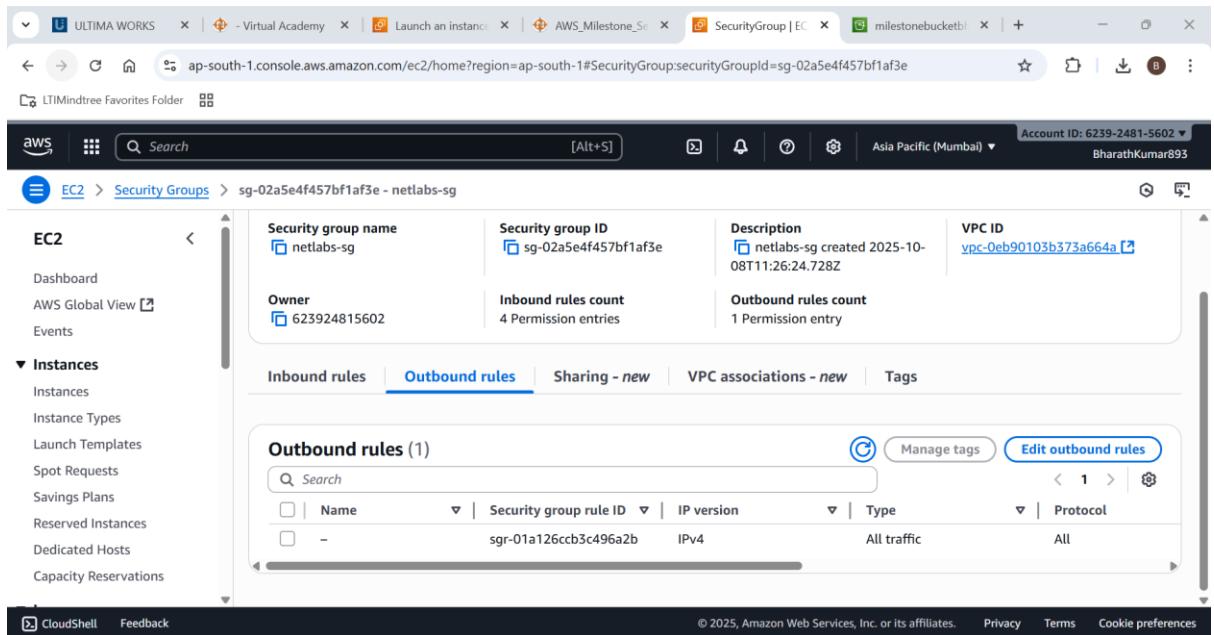
Inbound security is associated to the EC2 successfully



The screenshot shows the AWS EC2 Security Groups details page for the security group 'sg-02a5e4f457bf1af3e - netlabs-sg'. A green success message box is displayed: 'Inbound security group rules successfully modified on security group (sg-02a5e4f457bf1af3e | netlabs-sg)'. The 'Inbound rules' tab is selected, showing 4 entries. The 'Details' section shows the following information:

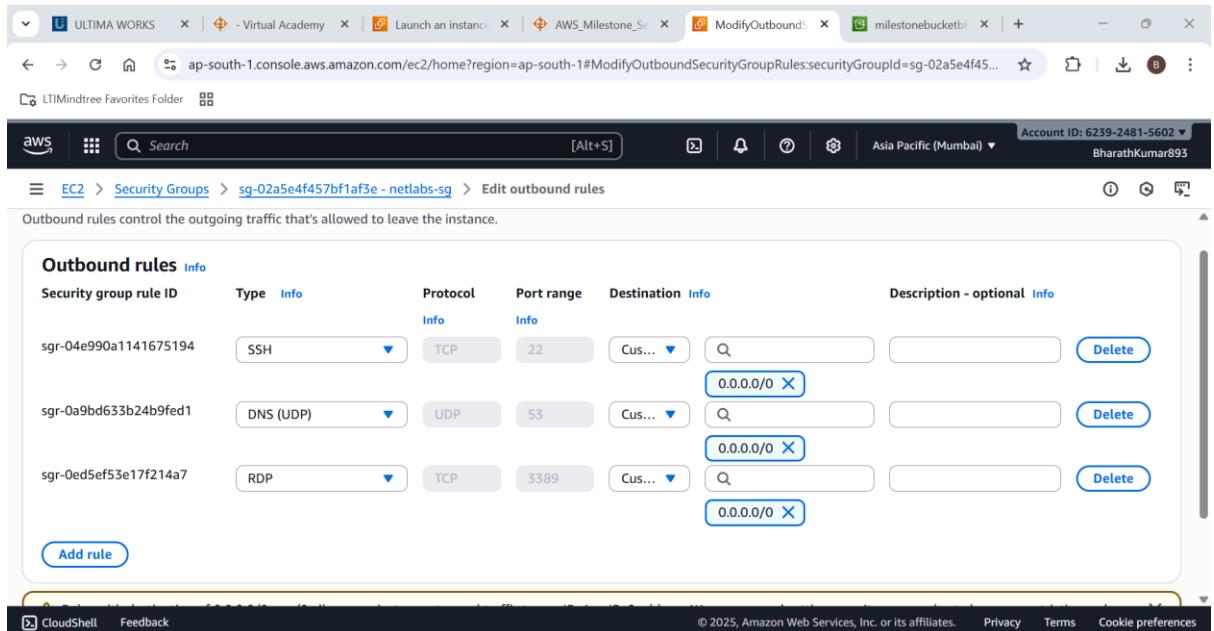
Security group name	sg-02a5e4f457bf1af3e
Owner	623924815602
Inbound rules count	4 Permission entries
Outbound rules count	1 Permission entry

Now lets edit the outbound rules



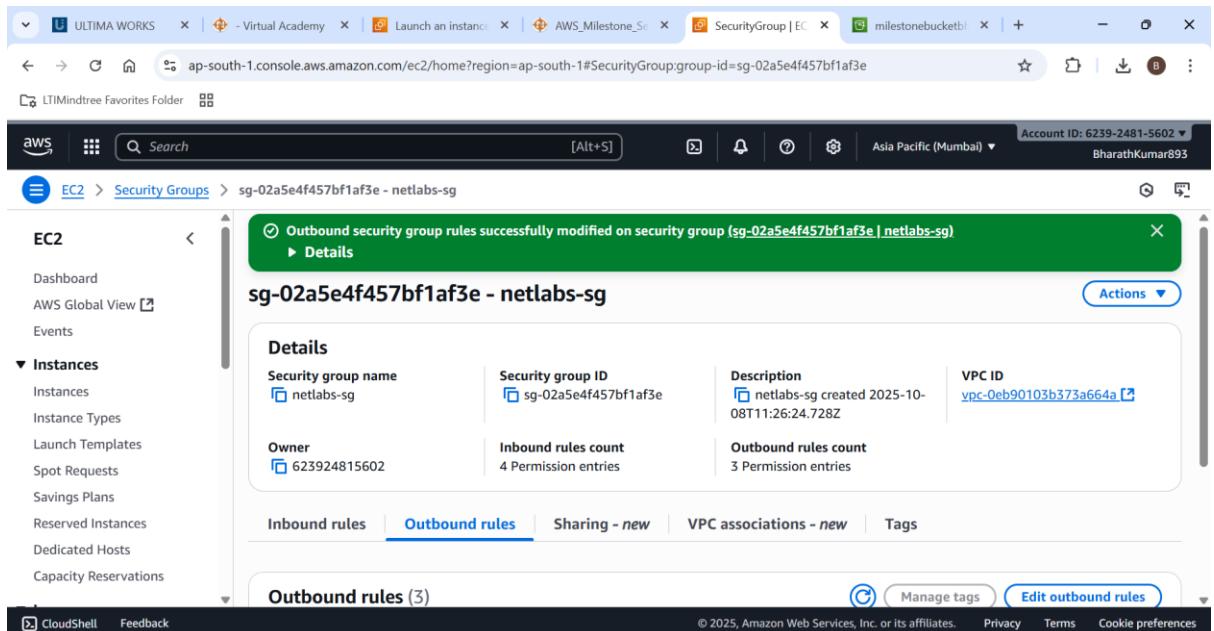
The screenshot shows the AWS EC2 Security Groups page. The security group selected is 'sg-02a5e4f457bf1af3e - netlabs-sg'. The 'Outbound rules' tab is active, displaying one rule: 'sgr-01a126ccb3c496a2b' (IPv4, All traffic, All). The 'Inbound rules' tab is also visible. The left sidebar shows navigation options for EC2, Instances, and other AWS services.

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.



The screenshot shows the 'Edit outbound rules' screen for the same security group. It lists three existing rules: SSH (TCP port 22), DNS (UDP port 53), and RDP (TCP port 3389). Each rule has a 'Delete' button next to its destination IP range (0.0.0.0/0). Below the table is an 'Add rule' button.

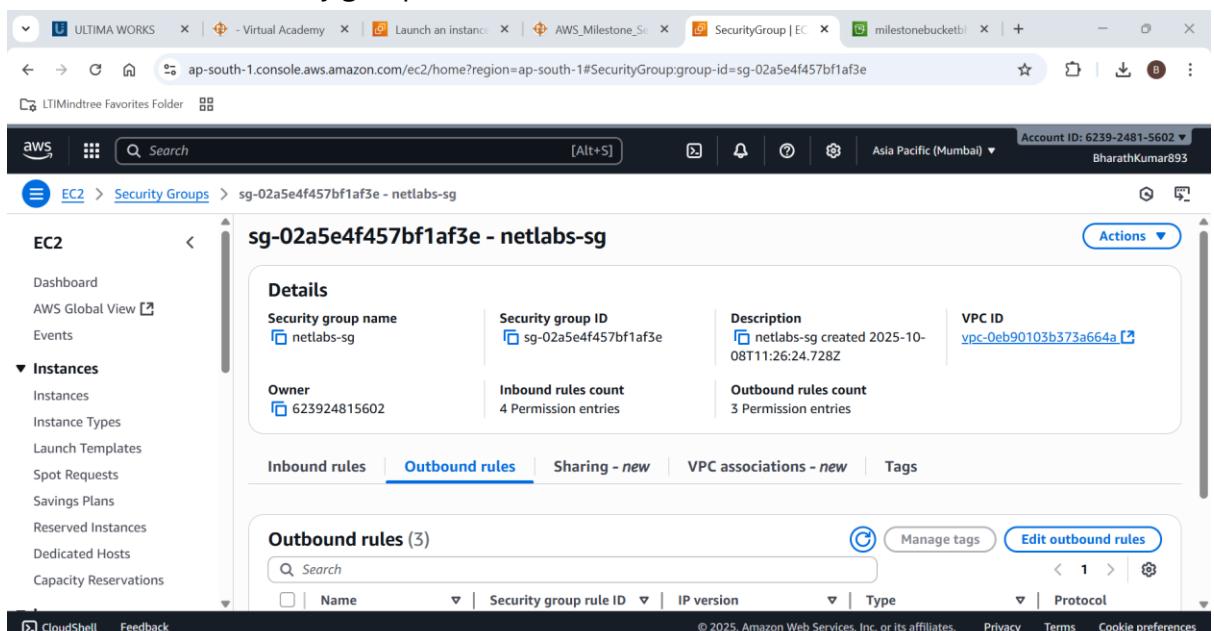
Add it to the EC2 Instance



The screenshot shows the AWS EC2 Security Groups page. A green success message at the top states: "Outbound security group rules successfully modified on security group (sg-02a5e4f457bf1af3e | netlabs-sg)". Below this, the security group details for "sg-02a5e4f457bf1af3e - netlabs-sg" are shown. The "Outbound rules" tab is selected, showing 3 entries. The "Details" section includes fields for Security group name (netlabs-sg), Security group ID (sg-02a5e4f457bf1af3e), Description (netlabs-sg created 2025-10-08T11:26:24.728Z), Owner (623924815602), Inbound rules count (4 Permission entries), and Outbound rules count (3 Permission entries). The page also includes tabs for Inbound rules, Sharing - new, VPC associations - new, and Tags. At the bottom, there are buttons for Manage tags and Edit outbound rules.

Outbound Security has been added to the instance successfully

Now lets see in Security group for verification



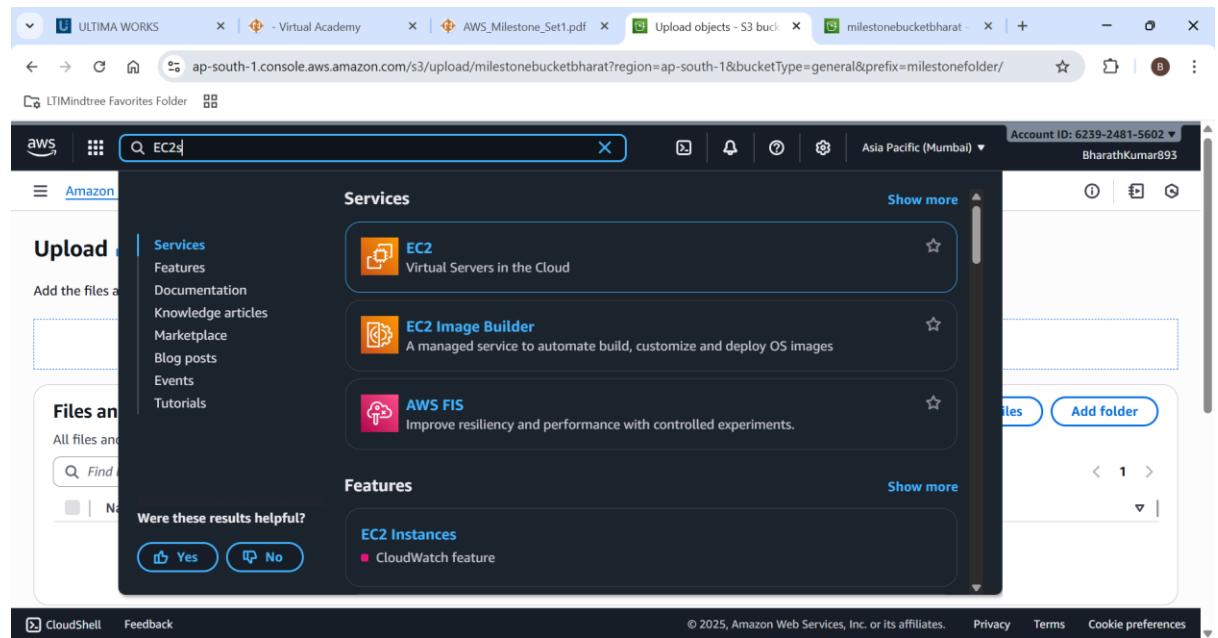
The screenshot shows the AWS EC2 Security Groups page for the same security group. The "Outbound rules" tab is selected, showing 3 entries. The "Details" section includes fields for Security group name (netlabs-sg), Security group ID (sg-02a5e4f457bf1af3e), Description (netlabs-sg created 2025-10-08T11:26:24.728Z), Owner (623924815602), Inbound rules count (4 Permission entries), and Outbound rules count (3 Permission entries). The page also includes tabs for Inbound rules, Sharing - new, VPC associations - new, and Tags. At the bottom, there are buttons for Manage tags and Edit outbound rules. The "Outbound rules" table has columns for Name, Security group rule ID, IP version, Type, and Protocol.

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

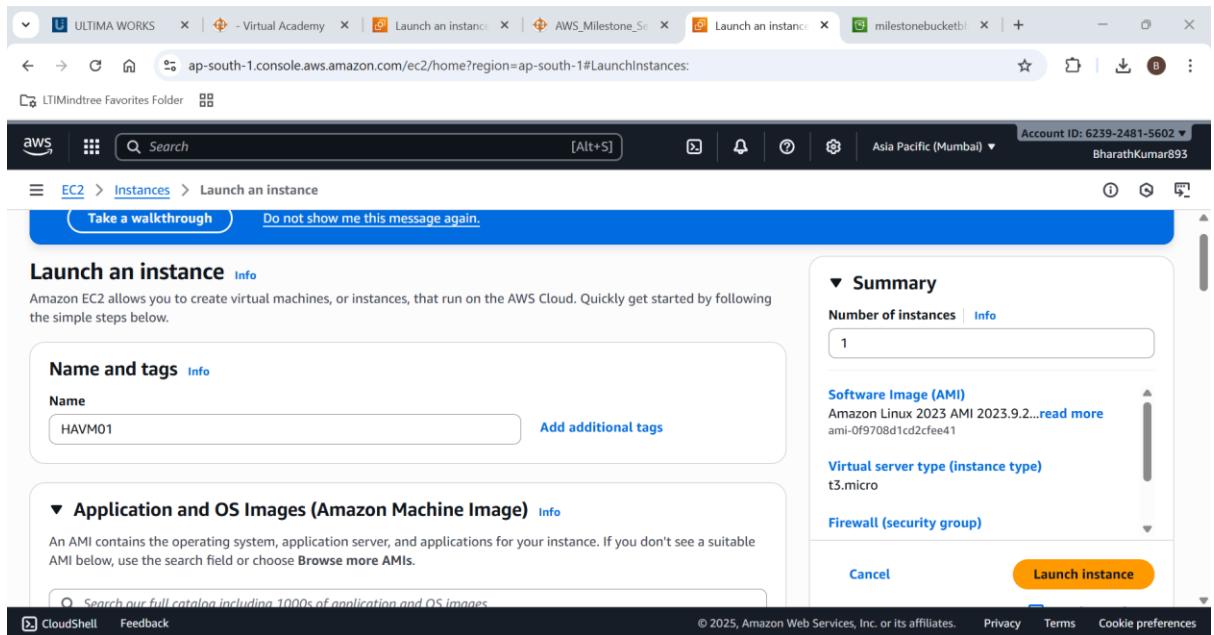
4. Create an Availability Zone Setup

- **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

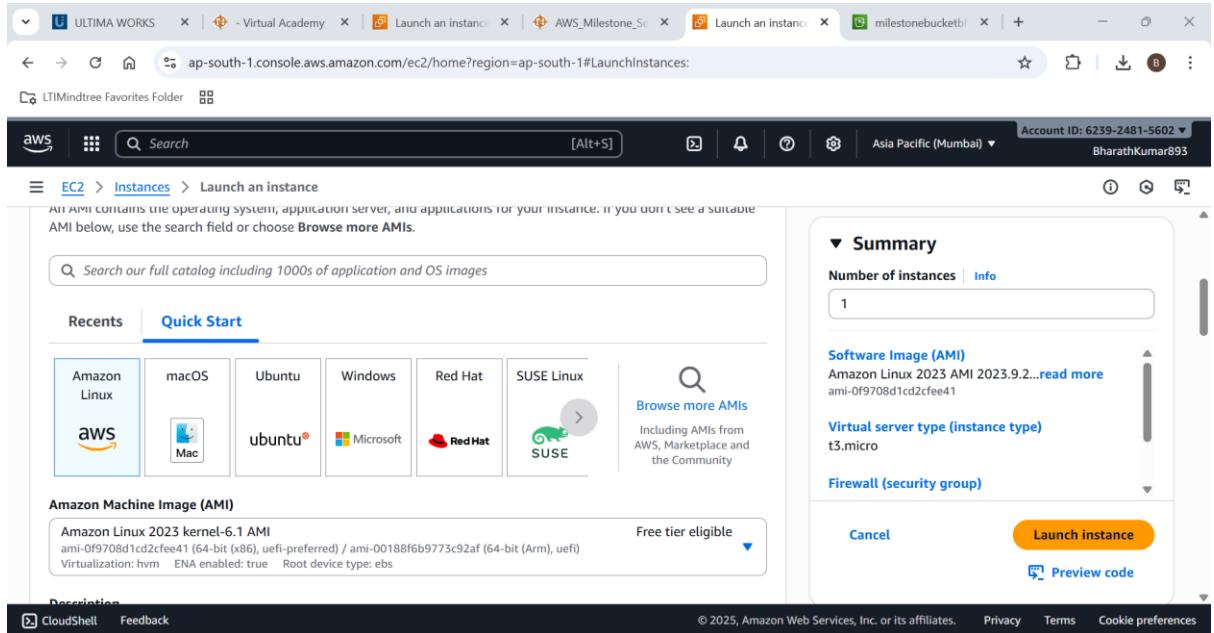


Lets create a EC2 instance with the name HAVM01



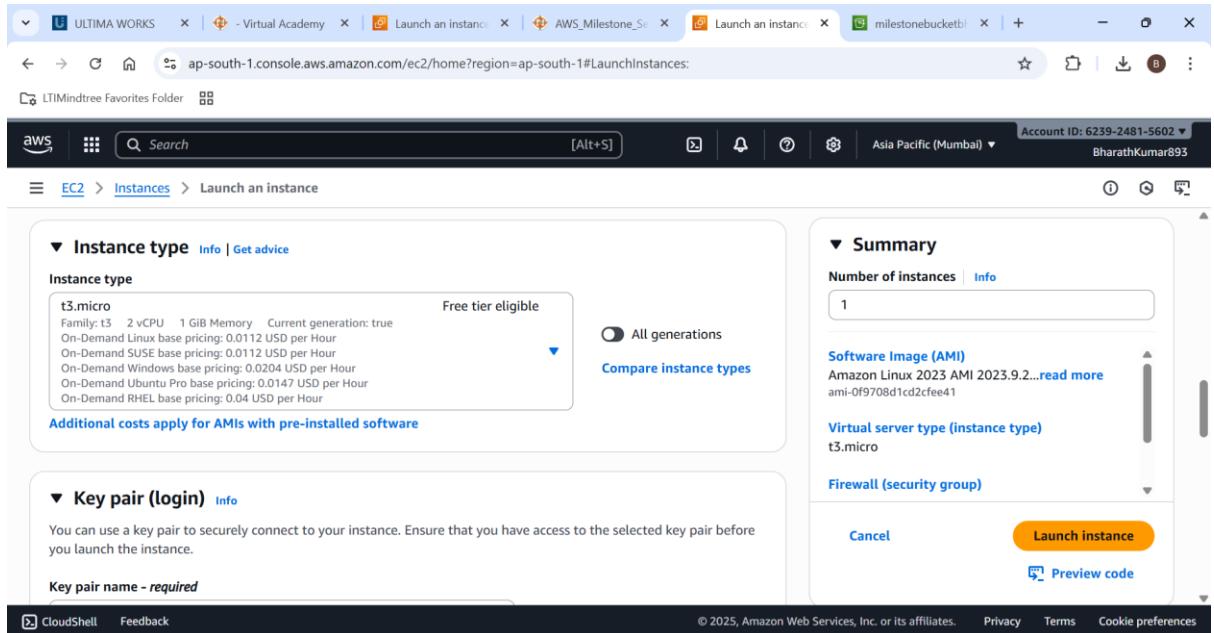
The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Name and tags' step, the instance name is set to 'HAVM01'. The 'Software Image (AMI)' section is expanded, showing 'Amazon Linux 2023 AMI 2023.9.2...' with a 'Launch instance' button. Other visible options include 'Virtual server type (instance type)' set to 't3.micro' and 'Firewall (security group)'.

Lets Choose Amazon linux-2023 kernel-6.1 AMI



The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Amazon Machine Image (AMI)' step, the 'Amazon Linux 2023 kernel-6.1 AMI' is selected. The 'Launch instance' button is visible. Other visible options include 'Virtual server type (instance type)' set to 't3.micro' and 'Firewall (security group)'.

Lets choose t3.micro as instance type



ULTIMA WORKS - Virtual Academy - Launch an instance - AWS_Milestone_Set - Launch an instance - milestonebucketb1 - +

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances: LTMindtree Favorites Folder

aws Search [Alt+S] Account ID: 6239-2481-5602

Asia Pacific (Mumbai) BharathKumar893

EC2 Instances Launch an instance

Instance type [Info](#) | [Get advice](#)

Instance type

t3.micro Family: t3 2 vCPU 1 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.0112 USD per Hour

On-Demand SUSE base pricing: 0.0112 USD per Hour

On-Demand Windows base pricing: 0.0204 USD per Hour

On-Demand Ubuntu Pro base pricing: 0.0147 USD per Hour

On-Demand RHEL base pricing: 0.04 USD per Hour

Free tier eligible

All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Summary

Number of instances [Info](#)

1

Software Image (AMI) Amazon Linux 2023 AMI 2023.9.2... [read more](#)

ami-0f9708d1cd2cfe41

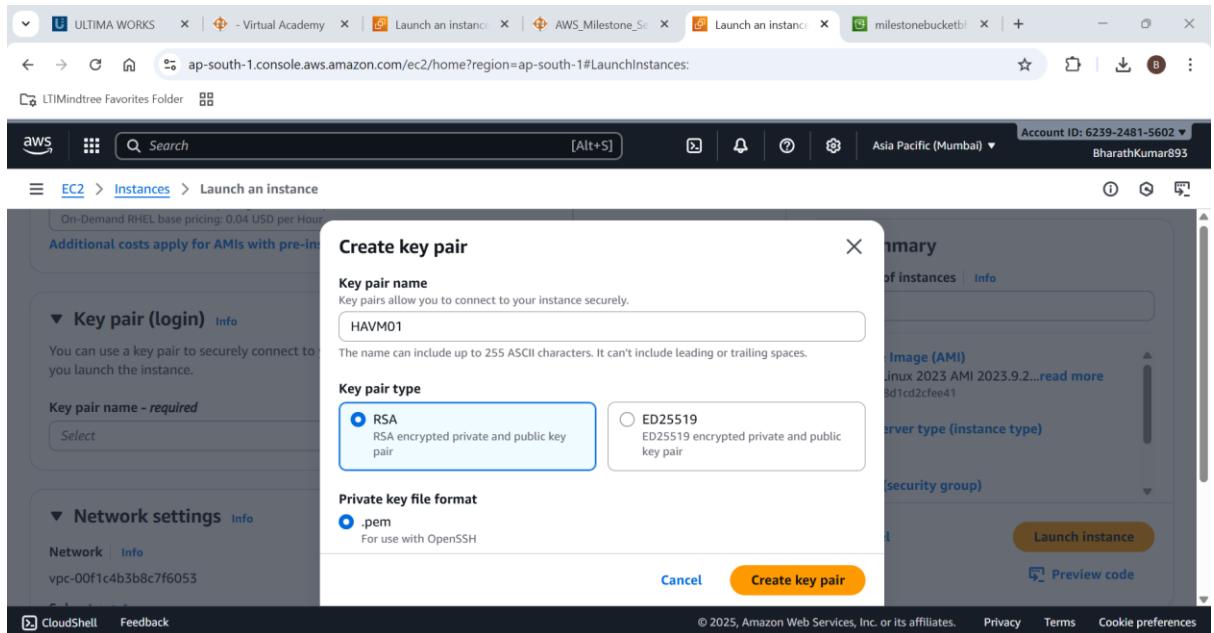
Virtual server type (instance type) t3.micro

Firewall (security group)

[Cancel](#) [Launch instance](#) [Preview code](#)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Create a new key value pair.



ULTIMA WORKS - Virtual Academy - Launch an instance - AWS_Milestone_Set - Launch an instance - milestonebucketb1 - +

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances: LTMindtree Favorites Folder

aws Search [Alt+S] Account ID: 6239-2481-5602

Asia Pacific (Mumbai) BharathKumar893

EC2 Instances Launch an instance

Create key pair

Key pair name Key pairs allow you to connect to your instance securely.

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

RSA RSA encrypted private and public key pair

ED25519 ED25519 encrypted private and public key pair

Private key file format

.pem For use with OpenSSH

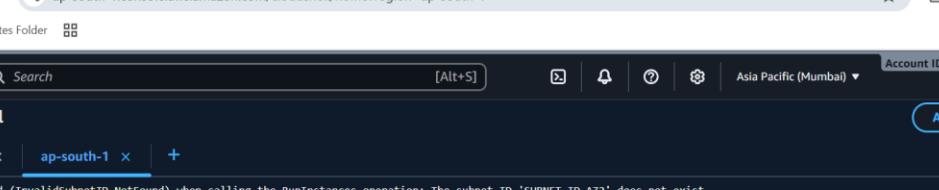
[Cancel](#) [Create key pair](#) [Preview code](#)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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

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

Use Cloudshell to create multiple Availability zones



ap-south-1.console.aws.amazon.com/cloudshell/home?region=ap-south-1#

```
aws CloudShell
Search [Alt+S] Actions ▾
CloudShell | 6239-2481-5602 ▾
BharathKumar893
CloudShell
ap-south-1 × ap-south-1 × +
An error occurred (InvalidSubnetID.NotFound) when calling the RunInstances operation: The subnet ID 'SUBNET_ID_AZ2' does not exist
~ $ # Check all your instances
~ $ aws ec2 describe-instances \
>   --filters "Name>tag:Name,Values=HVMW01" \
>   --query 'Reservations[].[Instances].[{Name:Tags[?Key==`Name`].Value[0],InstanceId:InstanceId,State:State.Name,AZ:Placement.AvailabilityZone,SubnetId:SubnetId,PublicIP:PublicIpAddress,PrivateIP:PrivateIpAddress}]" \
>   --region ap-south-1 \
>   --output table
| DescribeInstances
| AZ | InstanceId | Name | PrivateIP | PublicIP | State | SubnetId |
| ap-south-1c | i-02174a3336d10498 | HVMW01 | 10.18.2.200 | 3.7.62.24 | running | subnet-031d7d74188b6f464 |
| ap-south-1c | i-0b7d523346568696 | HVMW01-AZ1 | 10.18.2.186 | 52.66.223.22 | running | subnet-031d7d74188b6f464 |
~ $ # Check instance distribution across AZs
~ $ aws ec2 describe-instances \
>   --filters "Name>tag:Name,Values=HVMW01" "Name=instance-state-name,Values=running" \
>   --query 'Reservations[].[Instances].[Placement.AvailabilityZone]" \
>   --region ap-south-1 \
>   --output text | sort | uniq -c
 1 ap-south-1c  ap-south-1c
~ $
```

Instances is deployed at multiple zones successfully

Thus now lets verify instances connection



The screenshot shows the AWS CloudShell interface in a browser window. The URL is `ap-south-1.console.aws.amazon.com/cloudshell/home?region=ap-south-1#`. The AWS logo is in the top left, and the top navigation bar includes 'Search' and 'Actions' dropdowns. The main area shows a terminal session with the following command history and output:

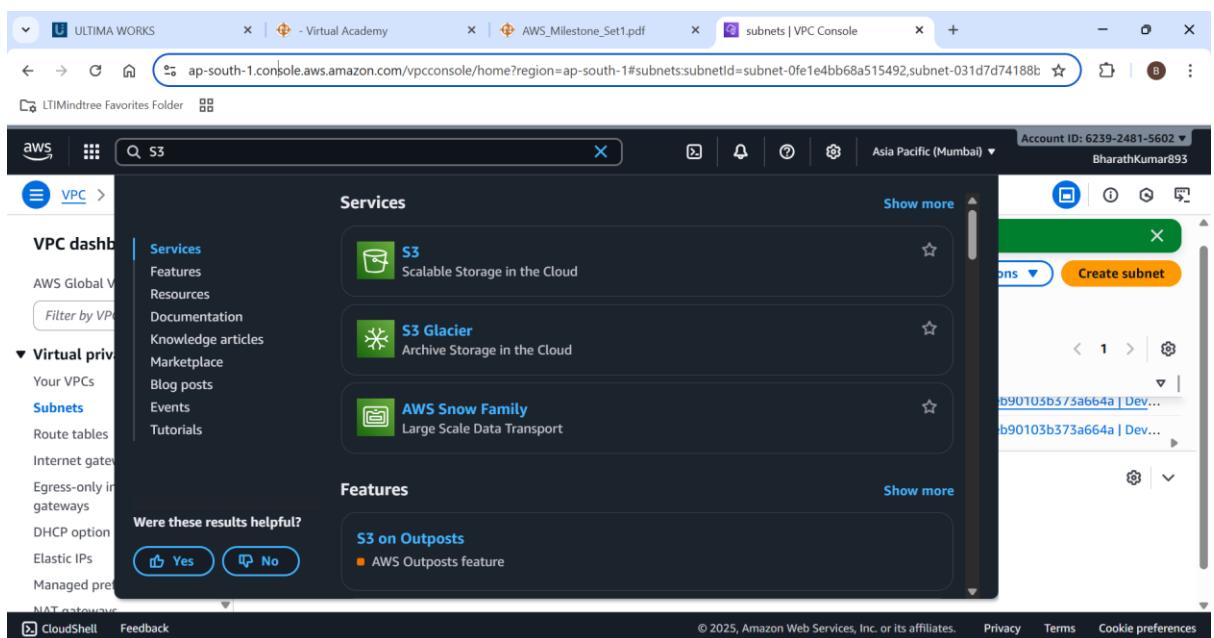
```
~ $ VPC_ID="vpc-0eb90103b373a664a"
~ $ SUBNET_ID="subnet-031d7d4188b6f464"
~ $ AMI_ID="ami-0f979801cd2cfe41"
~ $ SG_ID="sg-07f796a615360399a"
~ $ REGION="ap-south-1"
~ $ # Launch first instance
~ $ echo "Launching instance 1..."
Launching instance 1...
~ $ aws ec2 run-instances \
>   --image-id $AMI_ID \
>   --instance-type t3.micro \
>   --key-name HAVM-KeyPair \
>   --security-group-ids $SG_ID \
>   --subnet-id $SUBNET_ID \
>   --associate-public-ip-address \
>   --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=HAVM01-Primary}]' \
>   --region $REGION
{
  "ReservationId": "r-02c09c30b3b35357a",
  "OwnerId": "63924815602",
  "Groups": [],
  "Instances": [
    {
      "Architecture": "x86_64",
      "CreationTime": "2024-01-15T10:30:00.000Z",
      "ImageId": "ami-0f979801cd2cfe41",
      "InstanceId": "i-0f979801cd2cfe41",
      "InstanceType": "t3.micro",
      "KeyName": "HAVM-KeyPair",
      "NetworkInterfaces": [
        {
          "Association": {
            "AllocationId": "eni-0f979801cd2cfe41",
            "AssociationId": "eni-0f979801cd2cfe41",
            "Primary": true
          },
          "Description": "Primary network interface for instance i-0f979801cd2cfe41",
          "MacAddress": "54-12-97-01-00-00",
          "NetworkInterfaceId": "eni-0f979801cd2cfe41",
          "PrivateDns": "ip-172-31-1-1.ap-south-1.compute.internal",
          "PrivateIpAddress": "172.31.1.1",
          "Status": "in-use"
        }
      ],
      "Placement": {
        "AvailabilityZone": "ap-south-1a",
        "GroupName": null,
        "Tenancy": "default"
      },
      "State": {
        "Name": "pending",
        "Reason": null
      }
    }
  ],
  "OwnerId": "63924815602"
}
```

The requested has been done and verified successfully

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

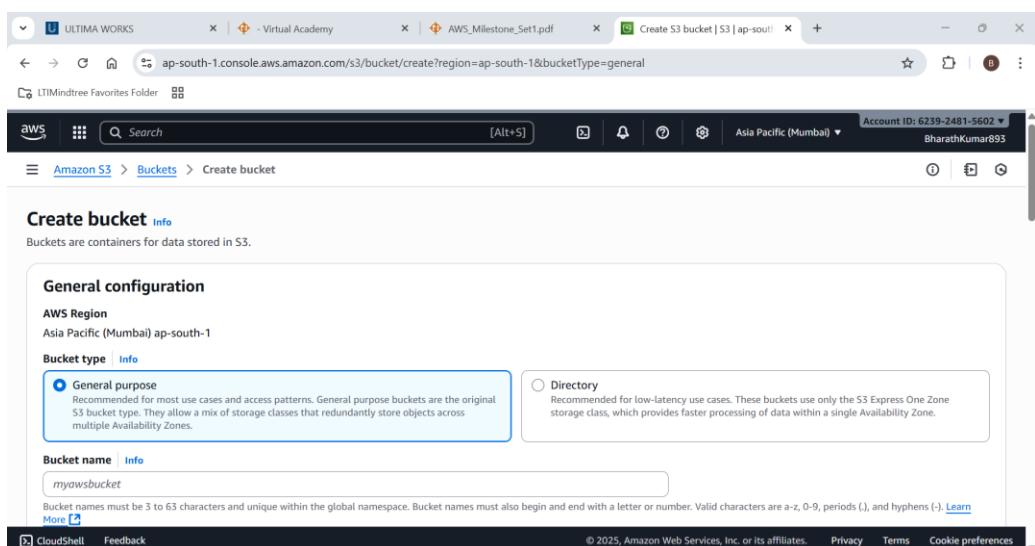
- Create a private S3 bucket (Do not make it Public).
- Enable ACL
- Create Folder name it container
- Upload a file into a container (folder) inside the bucket.
- Use Object Url and check whether you can see it or not, if No. How can you see with object ACL enablement.
- Ensure that the bucket and objects are private by default.

Step 1: Lets open the S3 bucket service



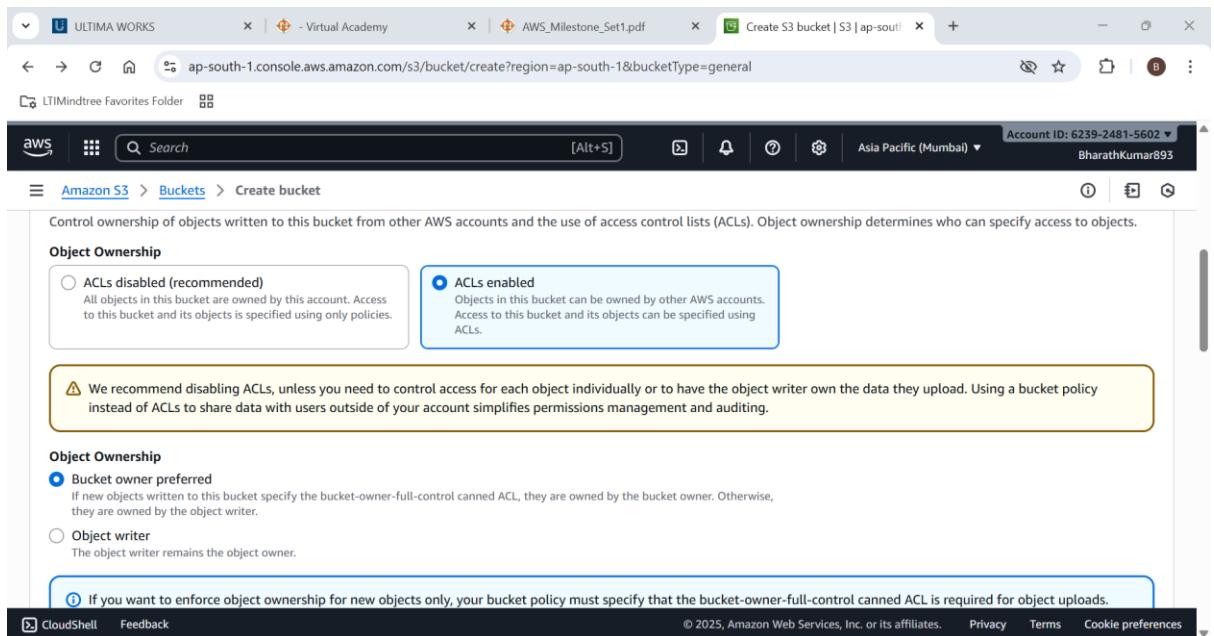
The screenshot shows the AWS VPC console. The left sidebar is collapsed, showing the VPC dashboard, AWS Global View, and a search bar for 'S3'. The main content area is titled 'Services' and lists 'S3 Scalable Storage in the Cloud', 'S3 Glacier Archive Storage in the Cloud', and 'AWS Snow Family Large Scale Data Transport'. Below this is a 'Features' section with a link to 'S3 on Outposts' and the 'AWS Outposts feature'. A sidebar on the right shows a list of subnets with buttons for 'Edit' and 'Create subnet'.

Create a new bucket



The screenshot shows the 'Create bucket' wizard in the AWS S3 console. The 'General configuration' step is displayed. The 'AWS Region' is set to 'Asia Pacific (Mumbai) ap-south-1'. The 'Bucket type' is set to 'General purpose'. The 'Bucket name' is 'myawsbucket'. A note at the bottom states: 'Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn More](#)'.

Enabling the ACL



Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

Object Ownership

ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

⚠️ We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

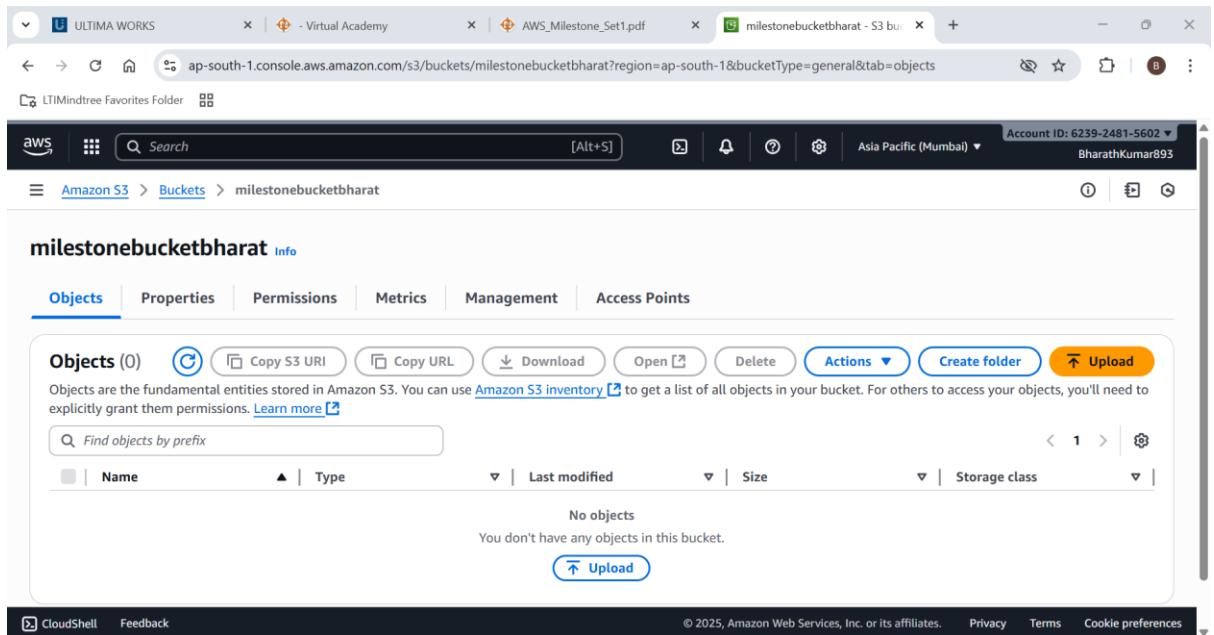
Object Ownership

Bucket owner preferred
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

Object writer
The object writer remains the object owner.

ⓘ If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads.

The new bucket is created successfully



ULTIMA WORKS - Virtual Academy AWS_Milestone_Set1.pdf Create S3 bucket | S3 | ap-south-1 Create S3 bucket | S3 | ap-south-1

LTIMindtree Favorites Folder

aws Search [Alt+S] Account ID: 6239-2481-5602 Asia Pacific (Mumbai) BharathKumar893

Amazon S3 > Buckets > Create bucket

milestonebucketbharat [Info](#)

Objects (0) [Actions](#) [Create folder](#) [Upload](#)

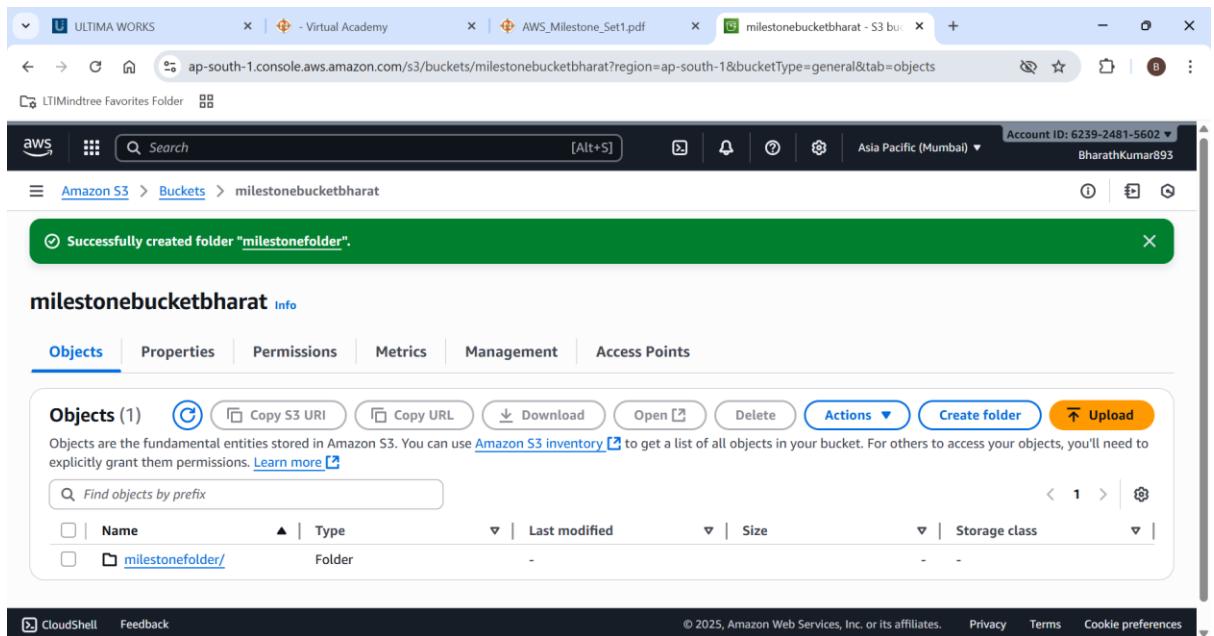
Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

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 © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Lets create a folder in it

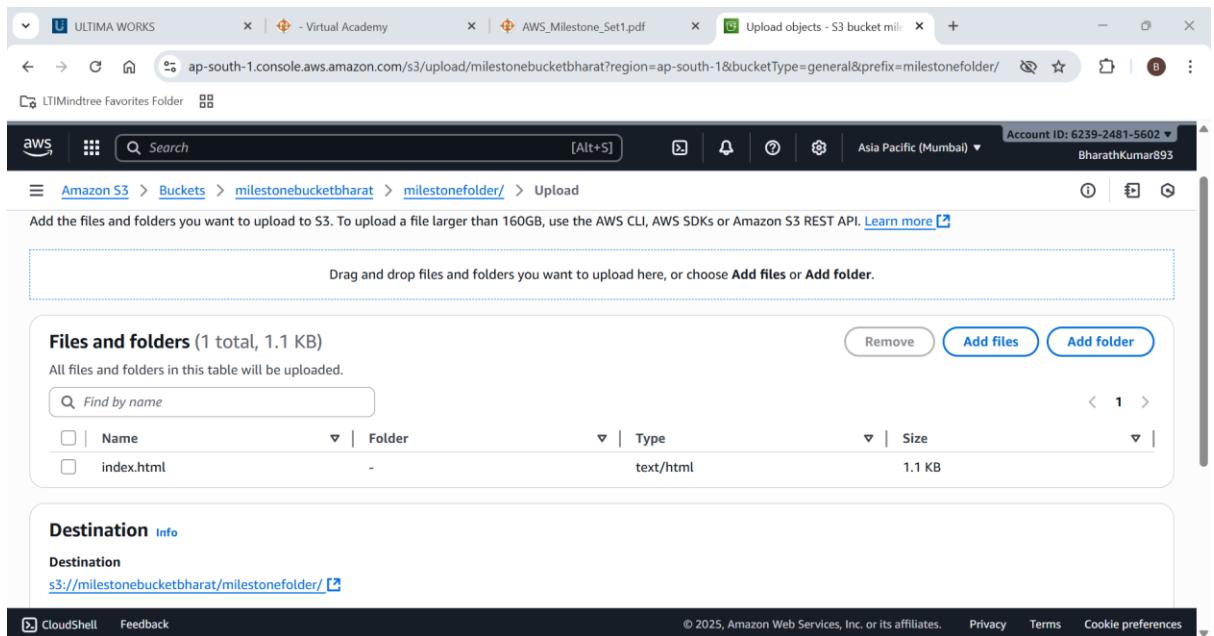


The screenshot shows the AWS S3 console with the following details:

- Tab bar: Ultima Works, Virtual Academy, AWS_Milestone_Set1.pdf, milestonebucketbharat - S3 bu...
- Header: AWS logo, Search bar, Account ID: 6239-2481-5602, Region: Asia Pacific (Mumbai), User: BharathKumar893
- Breadcrumbs: Amazon S3 > Buckets > milestonebucketbharat
- Message: Successfully created folder "milestonefolder".
- Section: Objects
- Table: Objects (1)
 - Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, Upload
 - Find objects by prefix: milestonefolder/
 - Columns: Name, Type, Last modified, Size, Storage class
 - Data: milestonefolder/ (Folder)
- Footer: CloudShell, Feedback, © 2025, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, Cookie preferences

New folder created successfully

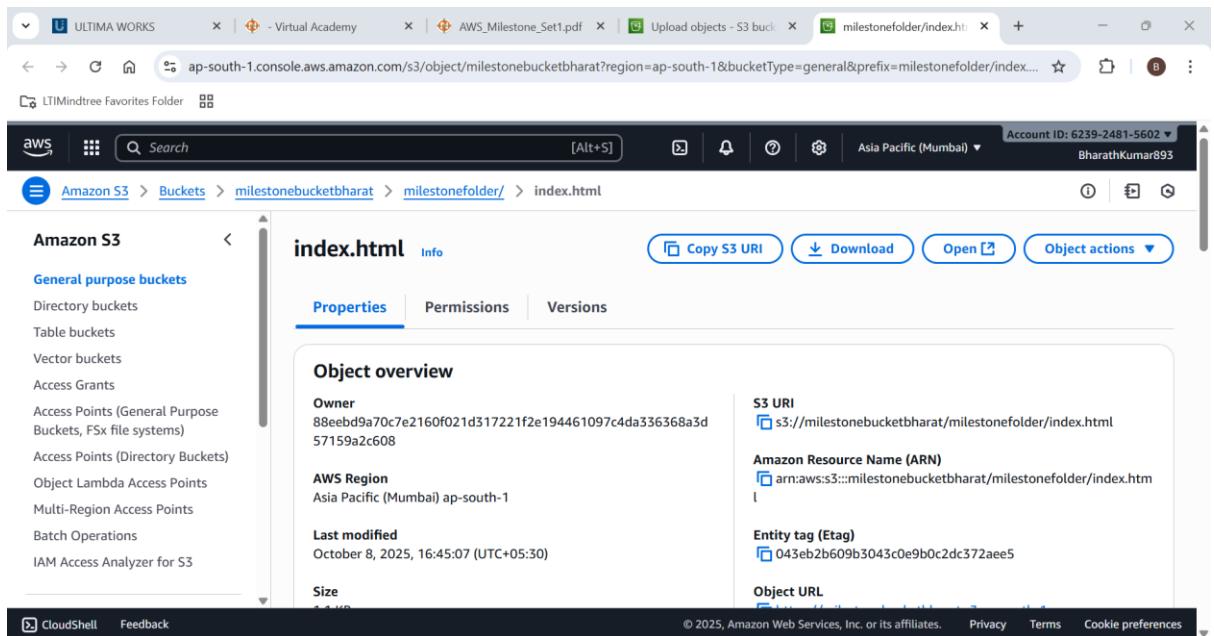
Lets upload a file (index.html)



The screenshot shows the AWS S3 console with the following details:

- Tab bar: Ultima Works, Virtual Academy, AWS_Milestone_Set1.pdf, Upload objects - S3 bucket mil...
- Header: AWS logo, Search bar, Account ID: 6239-2481-5602, Region: Asia Pacific (Mumbai), User: BharathKumar893
- Breadcrumbs: Amazon S3 > Buckets > milestonebucketbharat > milestonefolder/ > Upload
- Text: Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDKs or Amazon S3 REST API. [Learn more](#)
- Form: Drag and drop files and folders you want to upload here, or choose Add files or Add folder.
- Section: Files and folders (1 total, 1.1 KB)
 - Remove, Add files, Add folder
 - Find by name: index.html
 - Columns: Name, Type, Size
 - Data: index.html (text/html, 1.1 KB)
- Section: Destination [Info](#)
 - Destination: <s3://milestonebucketbharat/milestonefolder/>
- Footer: CloudShell, Feedback, © 2025, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, Cookie preferences

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



Amazon S3

General purpose buckets

index.html

Properties Permissions Versions

Object overview

Owner: 88eebd9a70c7e2160f021d317221f2e194461097c4da336368a3d57159a2c608

AWS Region: Asia Pacific (Mumbai) ap-south-1

Last modified: October 8, 2025, 16:45:07 (UTC+05:30)

Size: 1.1 KB

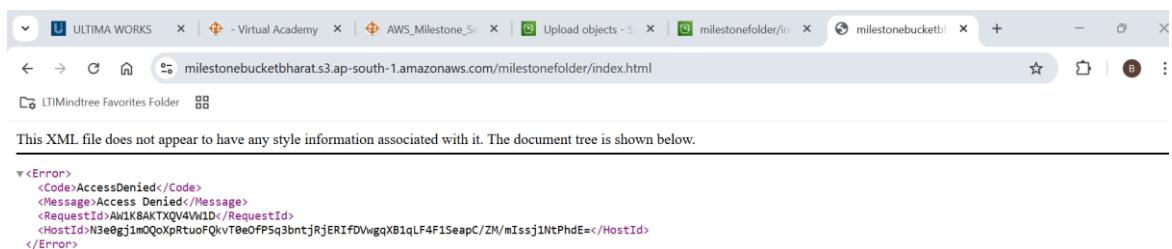
S3 URI: s3://milestonebucketbharat/milestonefolder/index.html

Amazon Resource Name (ARN): arn:aws:s3:::milestonebucketbharat/milestonefolder/index.html

Entity tag (Etag): 043eb2b609b3043c0e9b0c2dc372aee5

Object URL: <https://s3://milestonebucketbharat/milestonefolder/index.html>

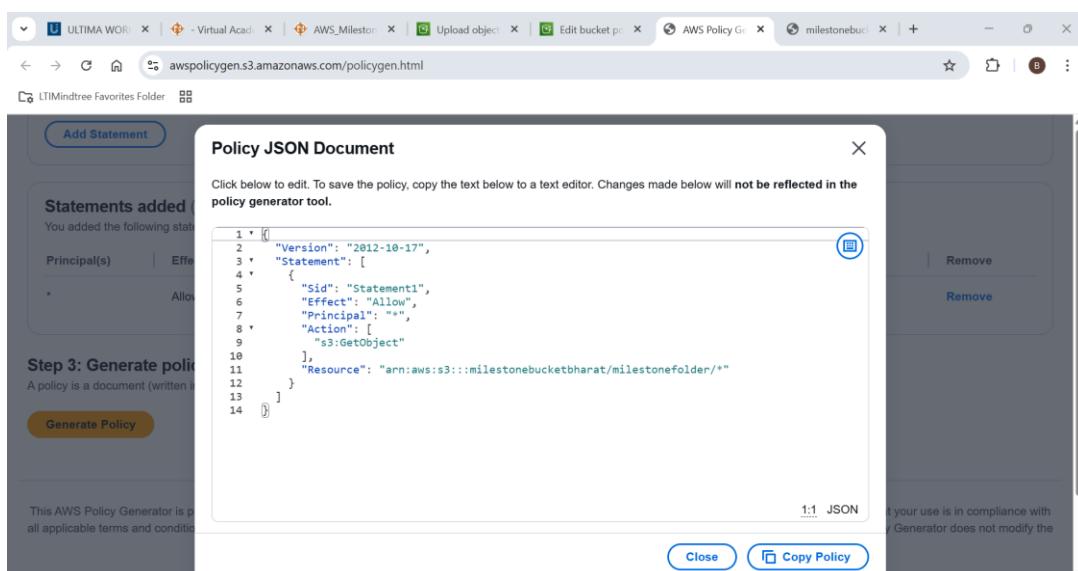
The Object is not accessible to public



```
<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>A91K8AKTXQV4VWID</RequestId>
  <HostId>N3e0gjm0QoXpRtuofQkv10e0FP5q3bntjRjERIfDwggXB1qLF4F1SeapC/ZM/mIssj1NtPhdE=</HostId>
</Error>
```

To make it public we need to edit the bucket policy

Genarate a Bucket policy



```
1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Sid": "Statement1",
6        "Effect": "Allow",
7        "Principal": "*",
8        "Action": [
9          "s3:GetObject"
10        ],
11        "Resource": "arn:aws:s3:::milestonebucketbharat/milestonefolder/*"
12      }
13    ]
14  }
```

Policy JSON Document

Click below to edit. To save the policy, copy the text below to a text editor. Changes made below will not be reflected in the policy generator tool.

Add Statement

Statements added

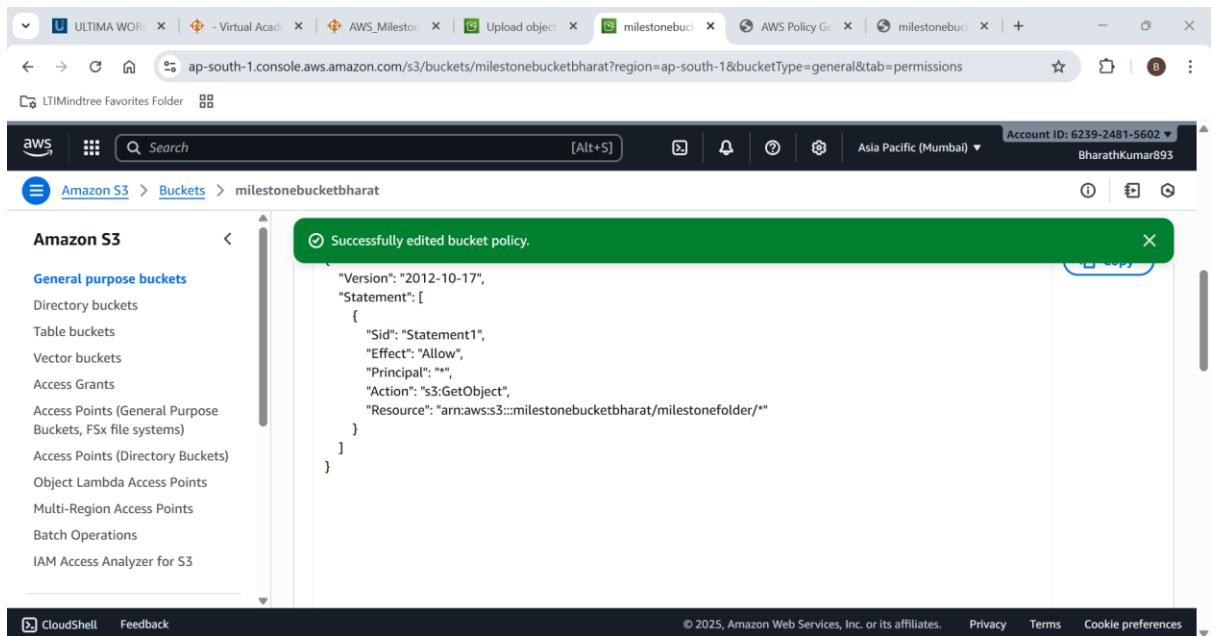
Principal(s) Effect

Step 3: Generate policy

Generate Policy

1:1 JSON

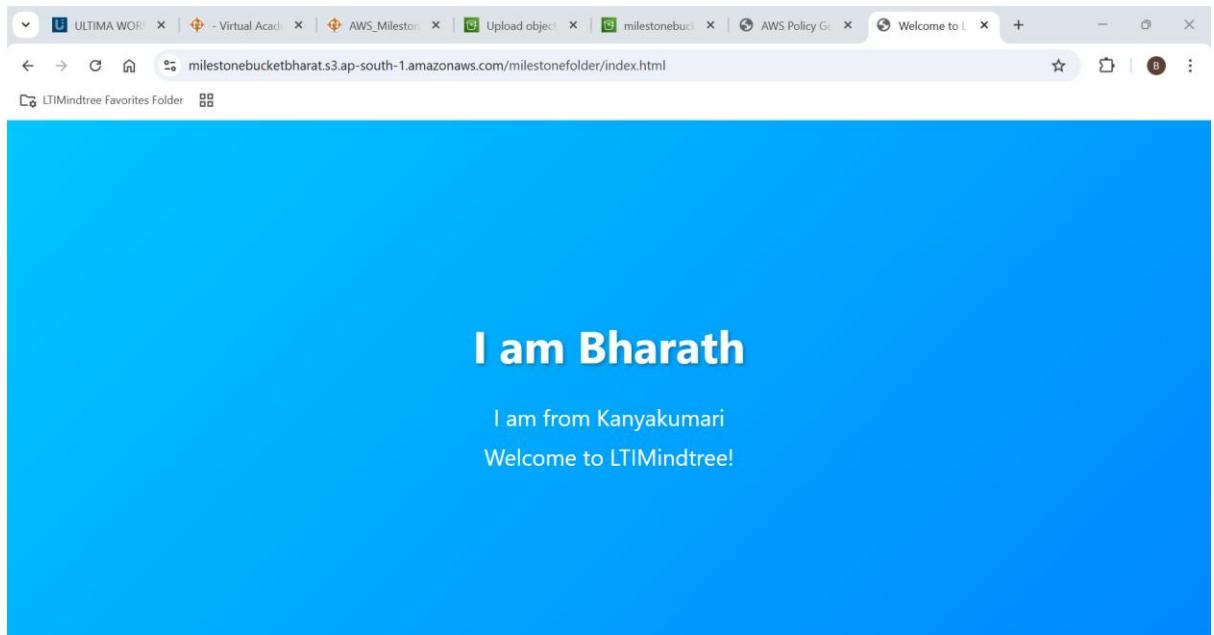
Bucket policy added successfully



The screenshot shows the AWS S3 Bucket policy editor. A green success message box at the top right says "Successfully edited bucket policy." Below it is the JSON policy code:

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::milestonebucketbharat/milestonefolder/*"
        }
    ]
```

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



The screenshot shows a public AWS S3 object at the URL <https://milestonebucketbharat.s3.ap-south-1.amazonaws.com/milestonefolder/index.html>. The page content is:

I am Bharath
I am from Kanyakumari
Welcome to LTIMindtree!

Now the Object is visible from public source

