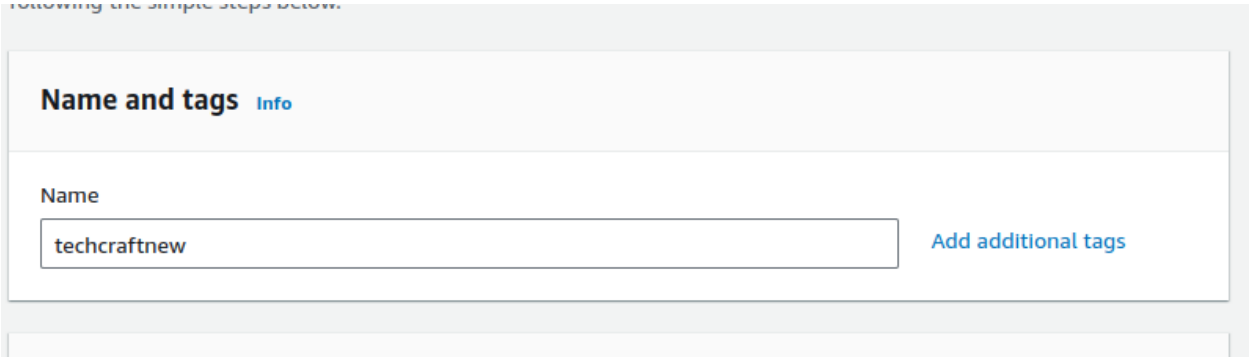
1. **EC2 Basics Lab**
    - **Objective**: To understand the process of setting up and managing an Amazon EC2 instance.
    - **Approach**: Students will start by launching a new EC2 instance, selecting an appropriate instance type and configuring the instance details. They will then create and configure a new Security Group, and allocate an Elastic IP address to the instance. The lab will also include connecting to the instance via SSH.
    - **Goal**: By the end of this lab, students should be able to launch and manage an EC2 instance, understand instance types, security groups, and IP addressing in AWS.

==========

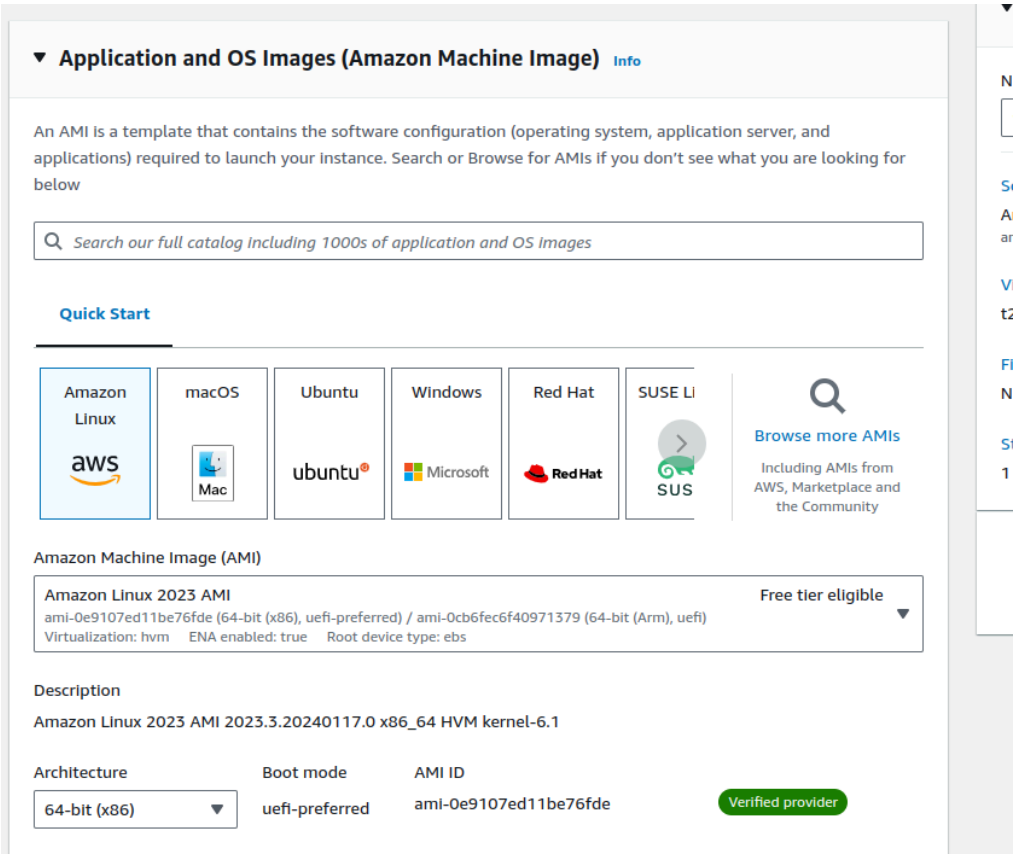
BASIC STEPS

1. Open the AWS Console and click EC2

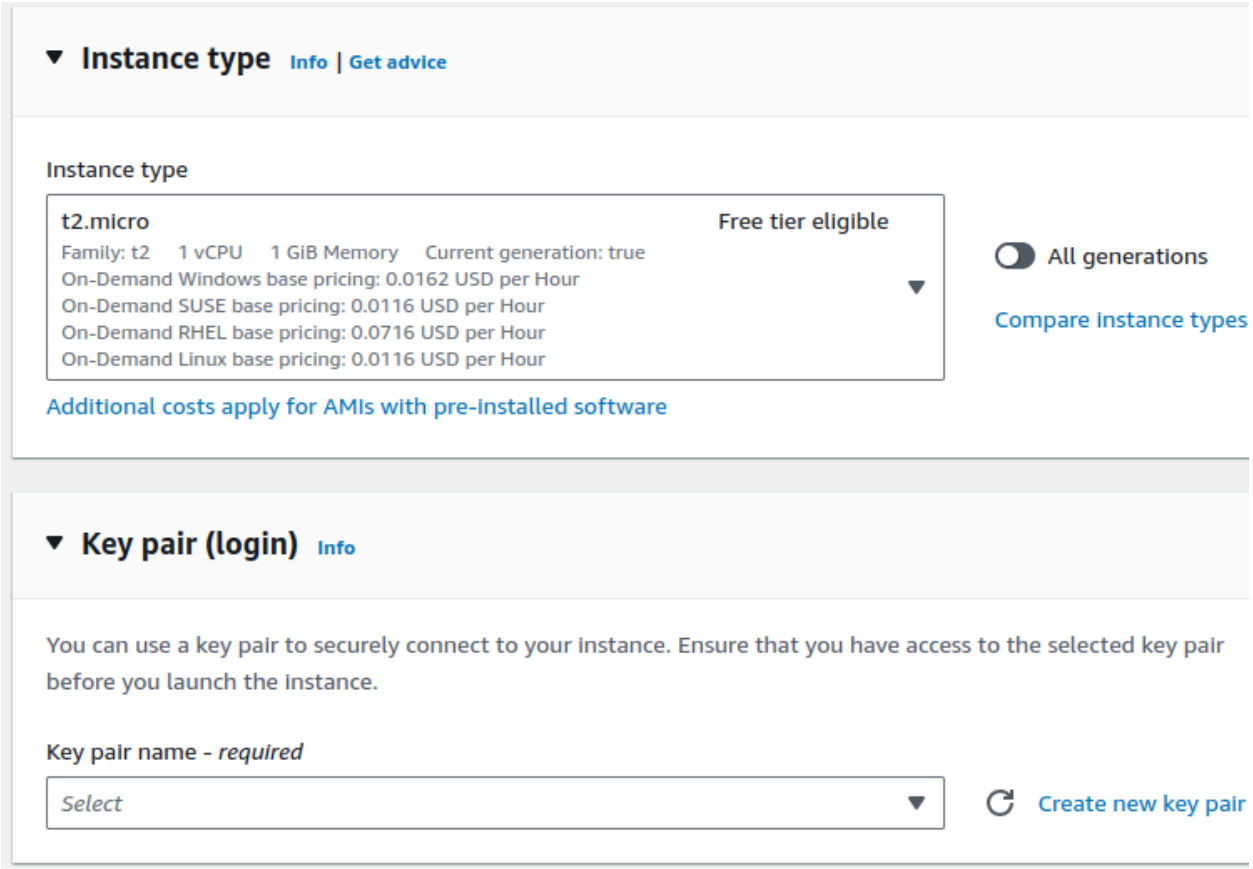
   Give the EC2 instance name



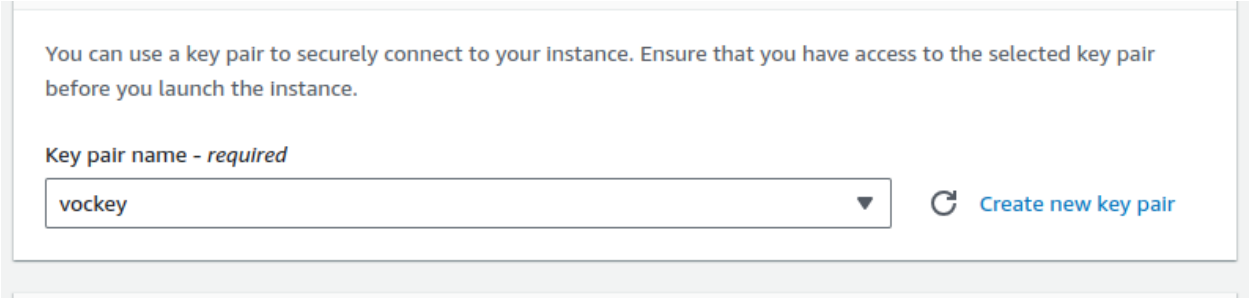
2. Select OS images and AMI



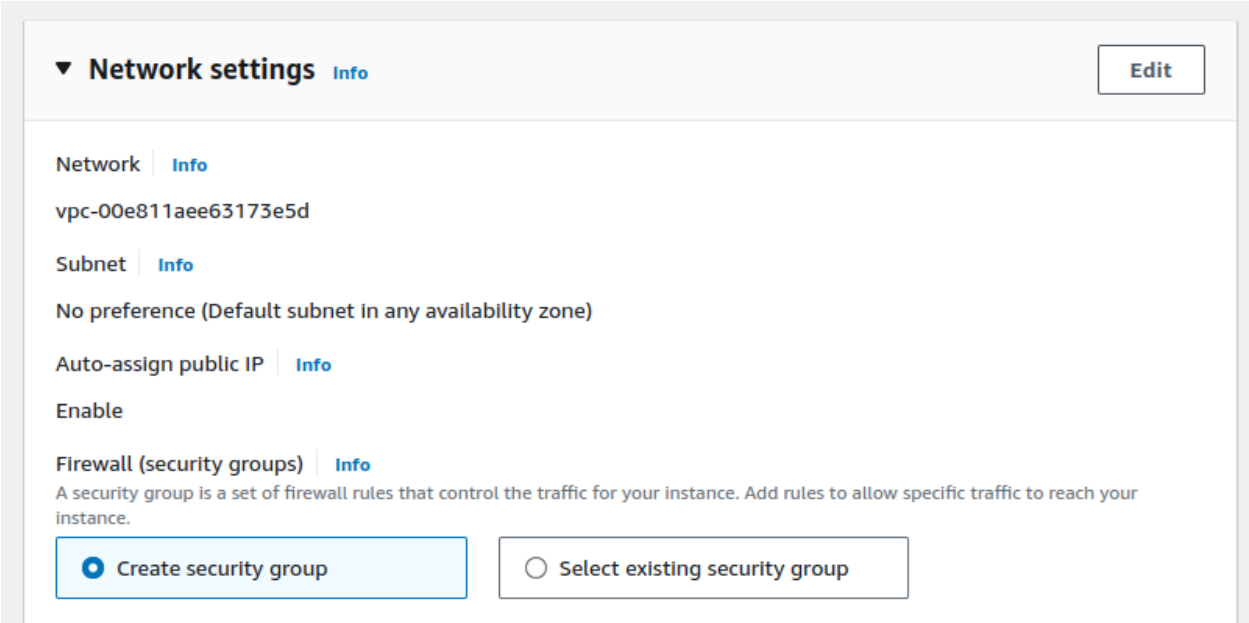
3. Select instance type as t2.micro



Select key pair login

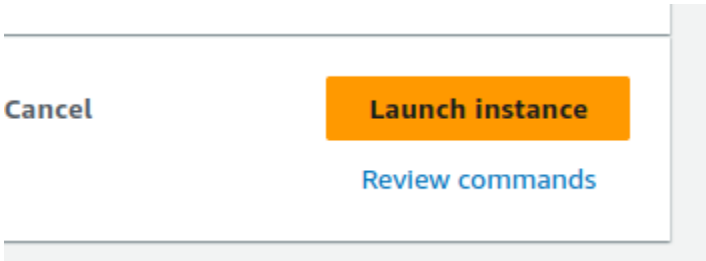


4. Select create security group and edit the rules

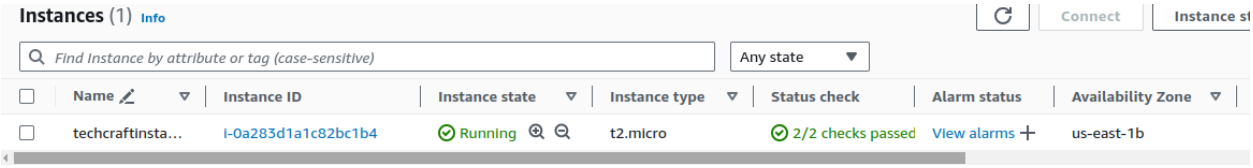


5. Disable auto assign public ip

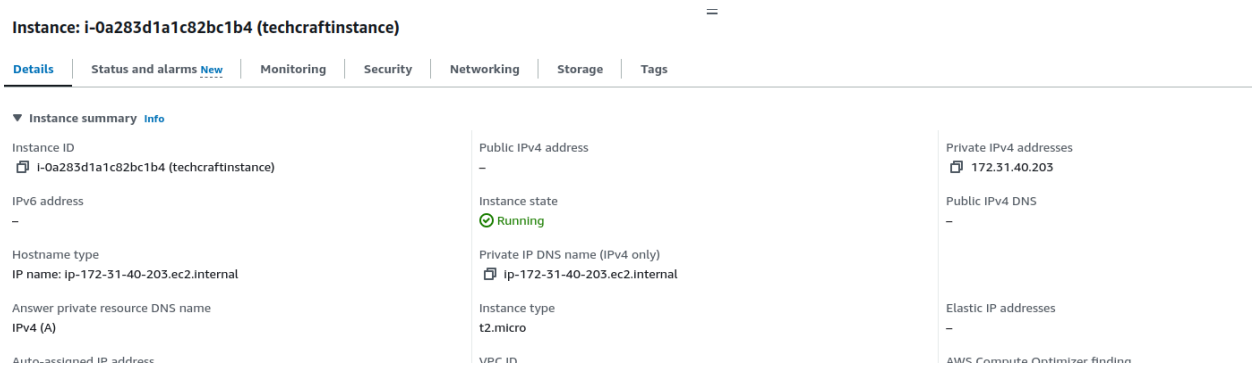
6. Click launch instance to create new EC2 instance



7. The instance is now created

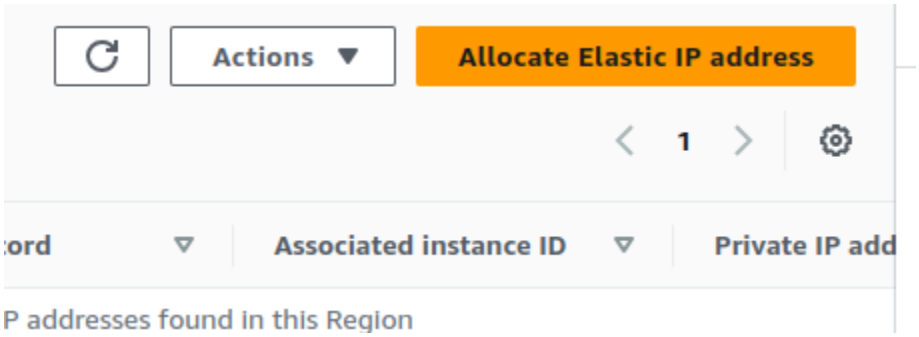


8. Not assign any ip for the EC2 instance

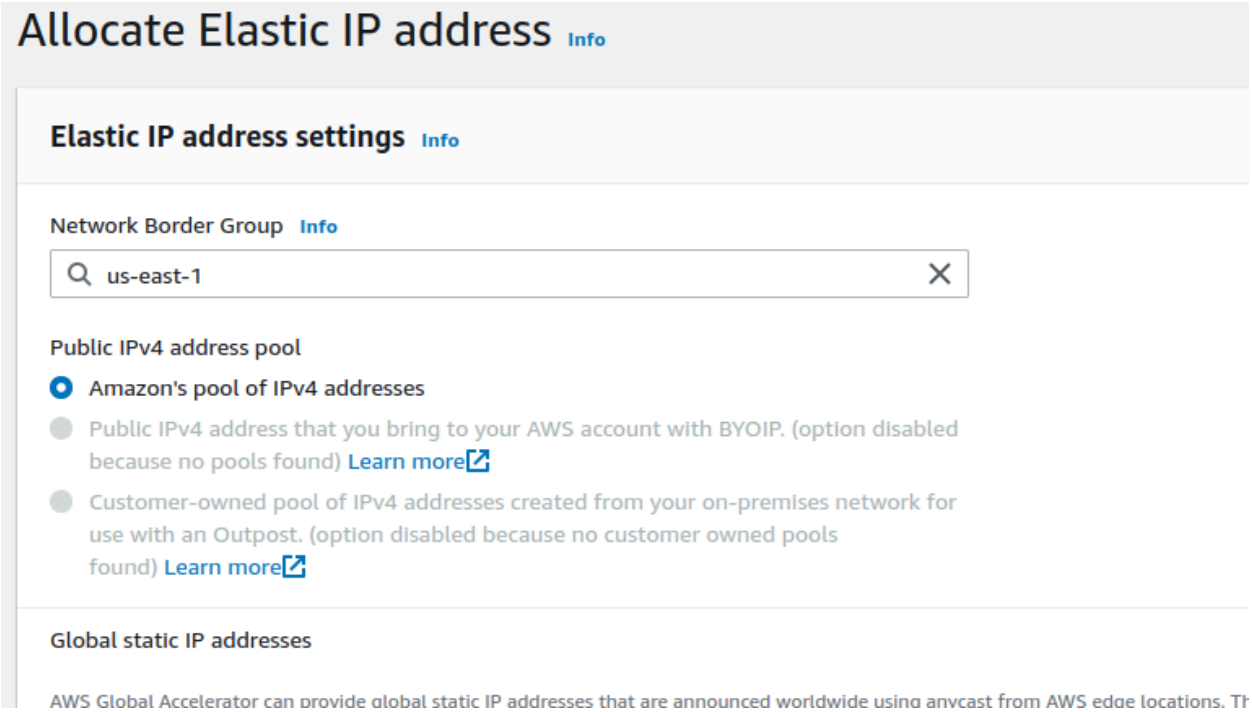


9. Now assing the Elastic ip address
   Select Elastic ips

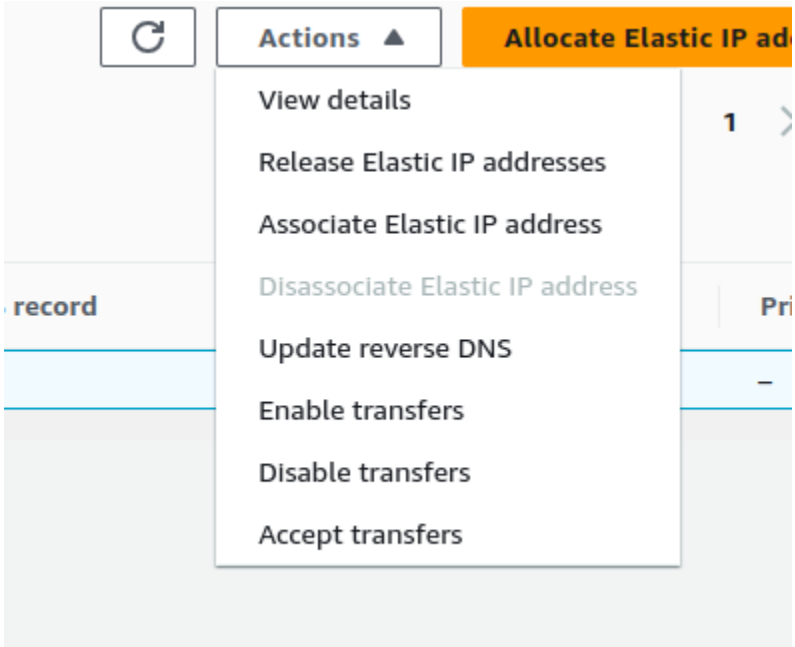
10. Select Allocate Elastic ip address



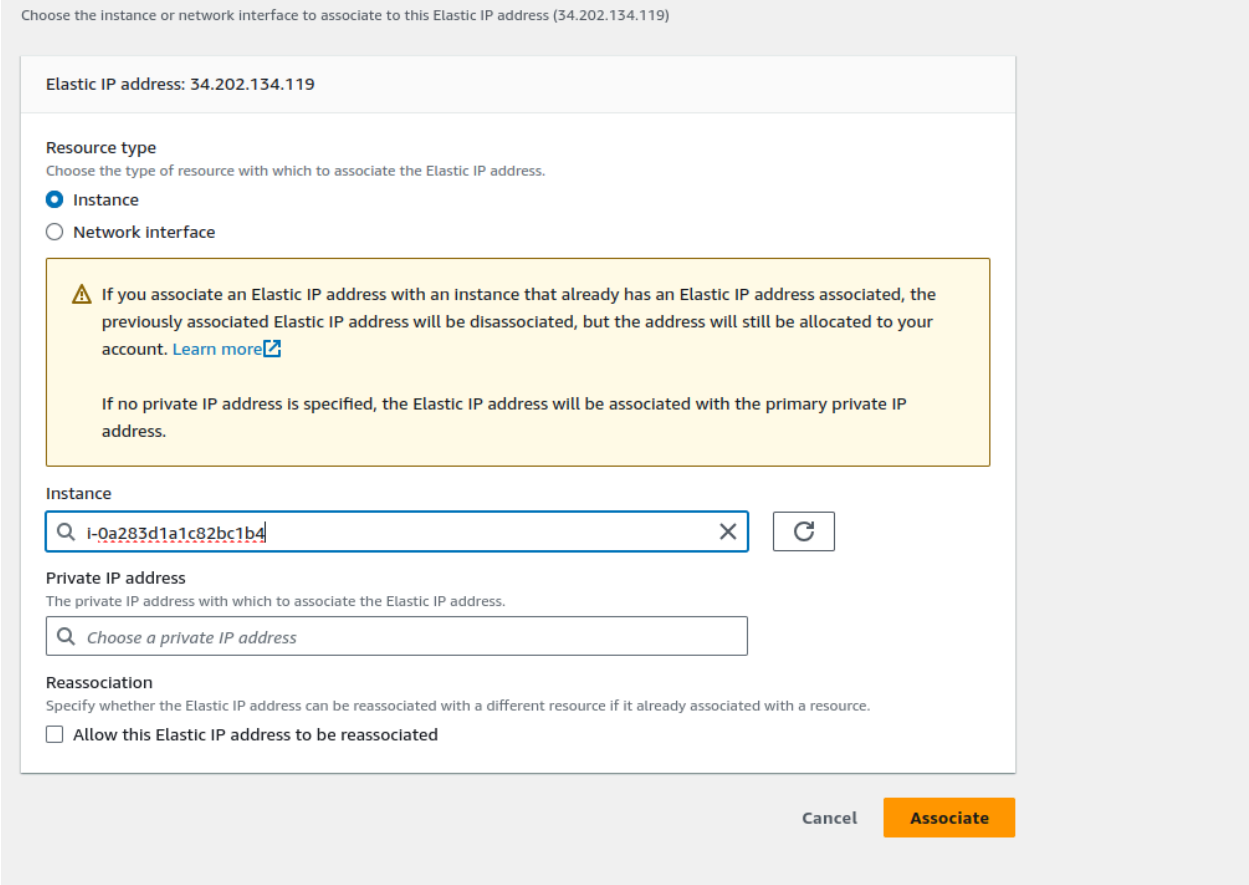
11. Select IPv4 to set IP address



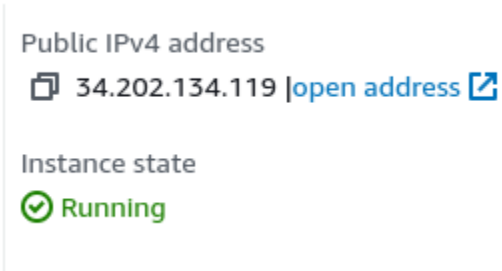
12. Keep the created ipv4 selected and click action and select associate elastic ip address



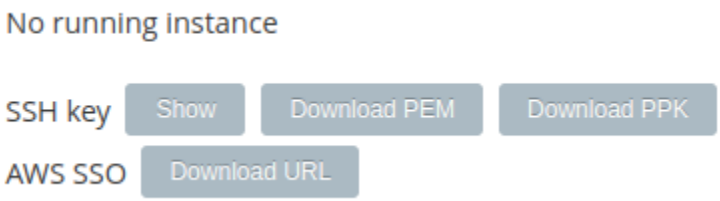
13. Select the EC2 instance and click associate



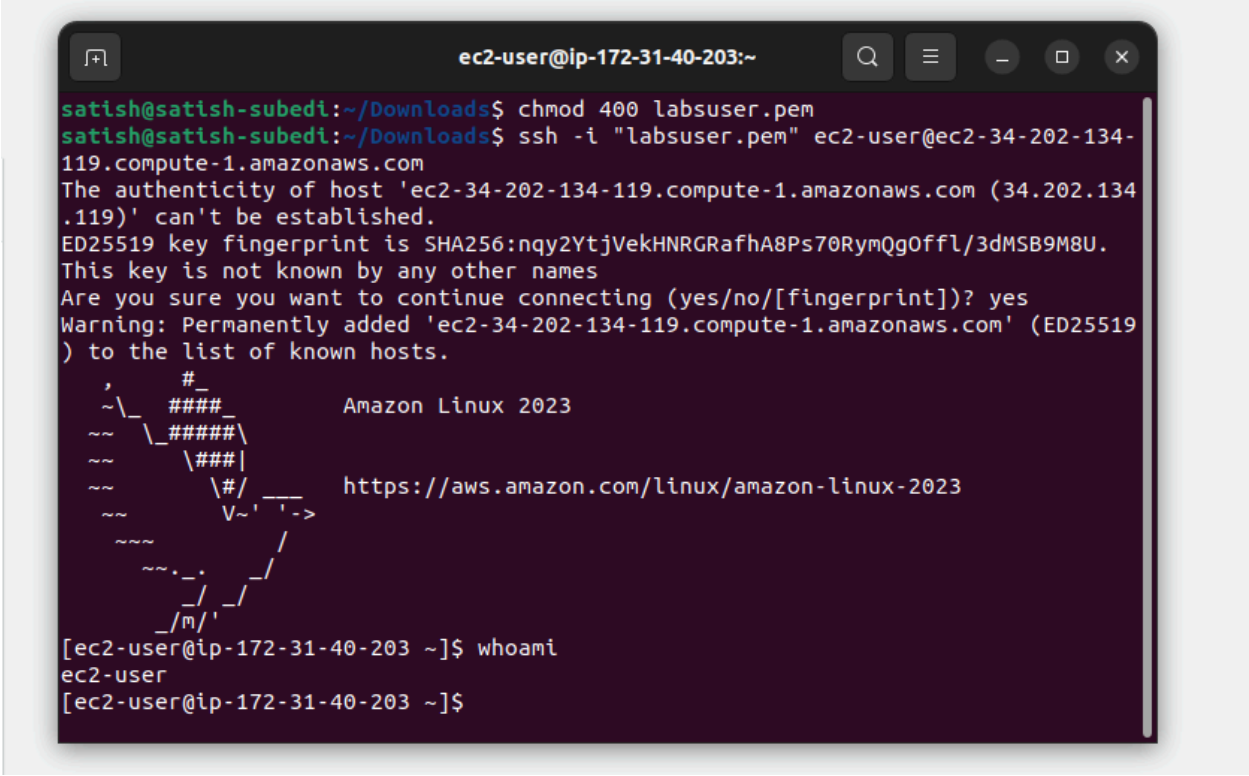
14. Now we can see the EC2 is assign an IP



15. To connect to the ec2 instance via ssh download the pem file in the aws details



16. Now connect to the ec2 instance

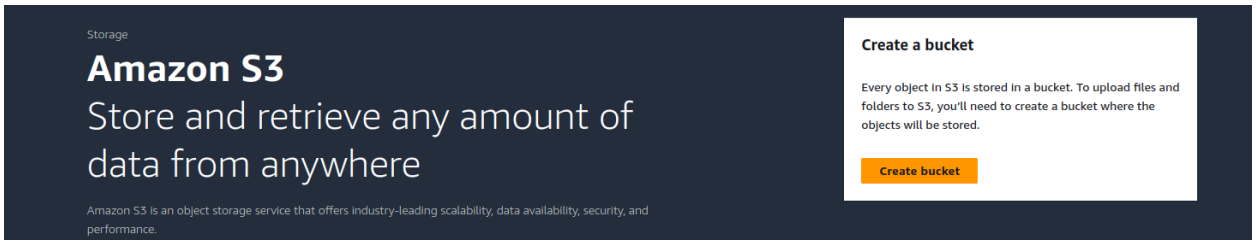


2. **S3 Storage Fundamentals Lab**
    - **Objective**: To gain hands-on experience with Amazon S3 by performing basic storage operations.
    - **Approach**: This lab involves creating an S3 bucket, uploading files to it, and setting up bucket policies for access control. Students will explore the S3 management console, learn about object storage, and understand the concepts of buckets and objects.
    - **Goal**: Students will understand how to use S3 for storing and managing data, learn about S3 security and permissions, and become familiar with S3's user interface.

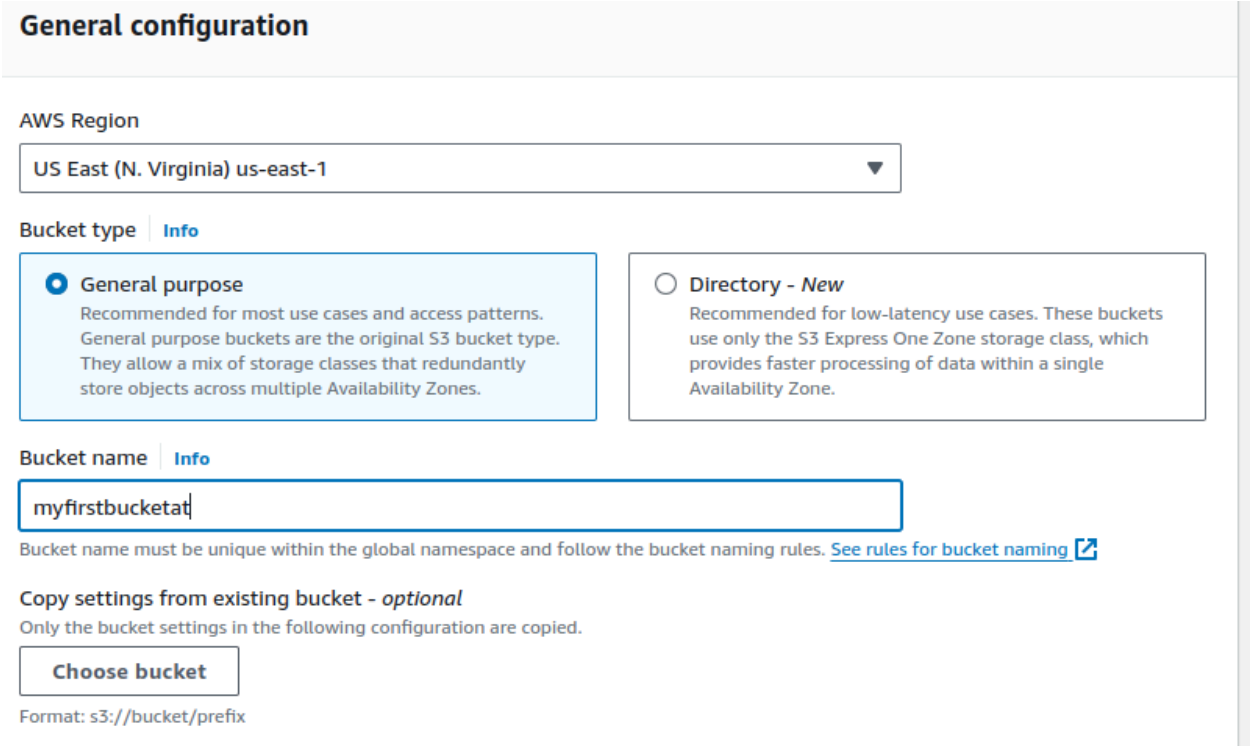
##

BASIC STEPS

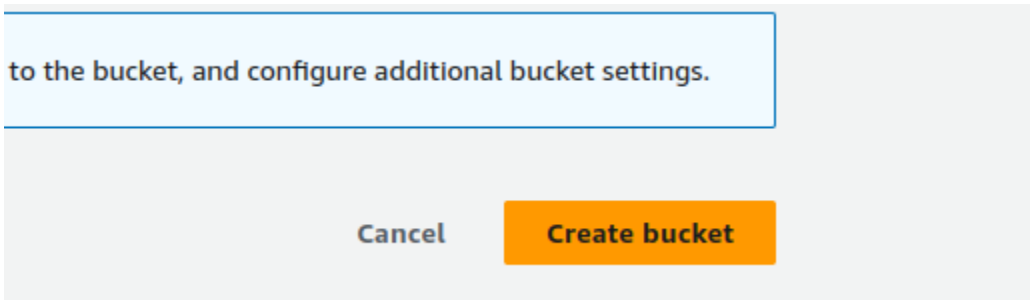
1. Go to the AWS Console Management and select S3
   Click create bucket to create the S3 bucket



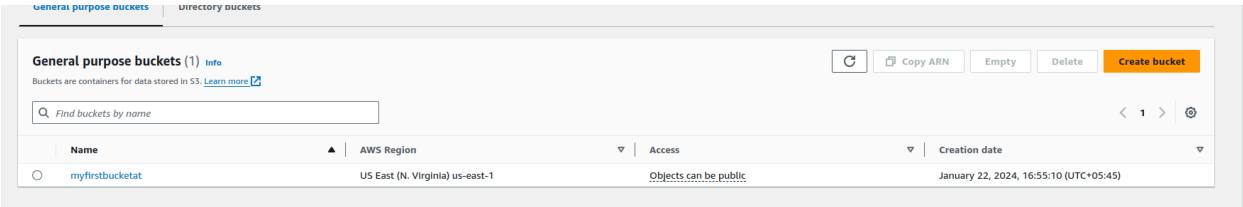
2. Choose the bucket name the bucket name must be unique



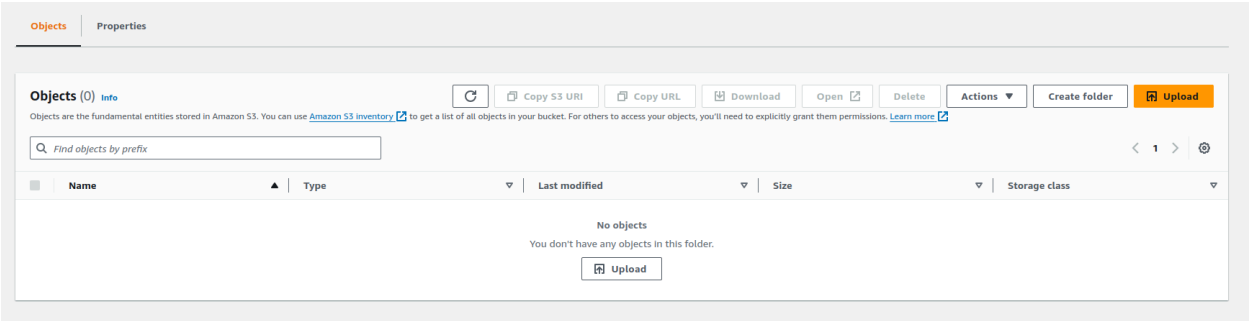
3. Click create bucket



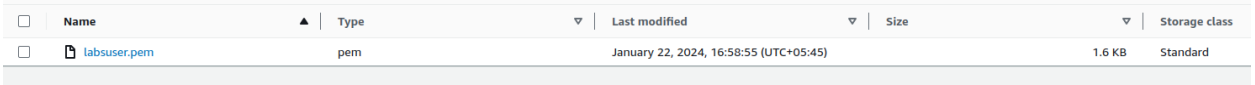
4. Bucket is now created



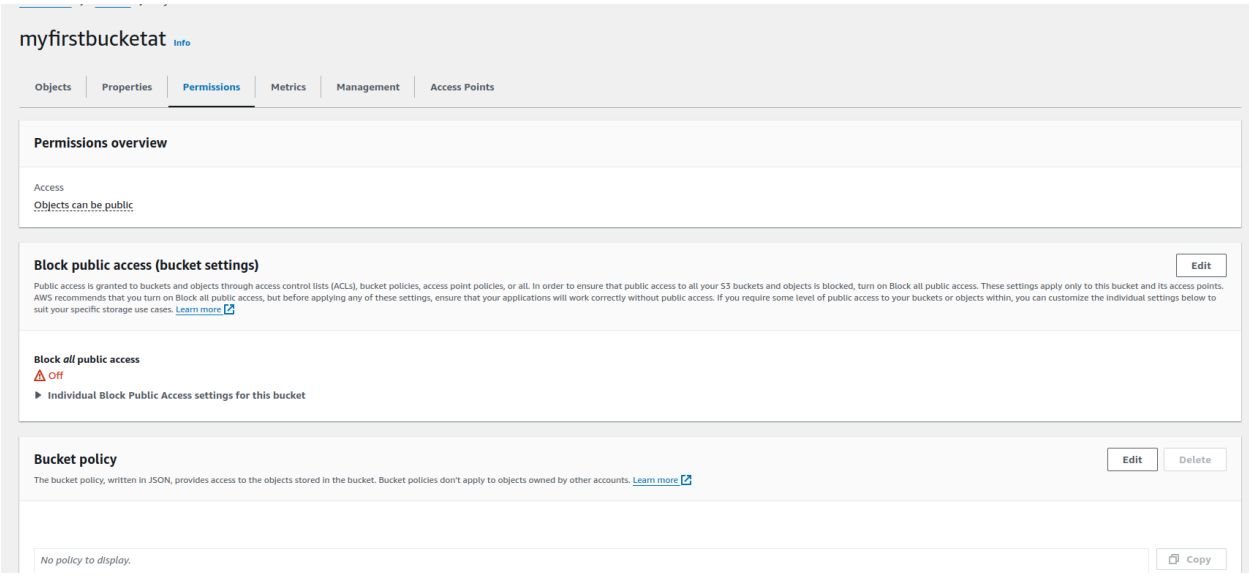
5. Click upload to upload the file



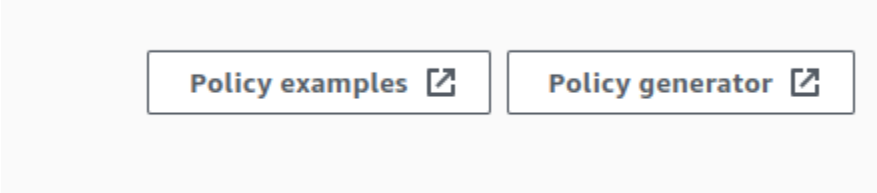
6. File is uploaded



7. To set the policies click the bucket name and select permission tab and you can see the Bucket policy now click edit to set the polices to the bucket



8. Click to policy generator to generate policy



9. Add necessary policy and click to add Statement

AWS Policy Generator

The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about c
concepts in Using AWS Identity and Access Management. Here are sample policies.

Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an

Select Type of Policy S3 Bucket Policy

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a description of elements that you can use in statements.

Effect (•) Allow ( ) Deny

Principal *

Use a comma to separate multiple values.

AWS Service Amazon S3 [ ] All Services ('*')

Use multiple statements to add permissions for more than one service.

Actions 1 Action(s) Selected [ ] All Actions ('*')

Amazon Resource Name (ARN) arn:aws:s3:::myfirstbucketat

ARN should follow the following format: arn:aws:s3:::${BucketName}/${KeyName}.
Use a comma to separate multiple values.

Add Conditions (Optional)

Add Statement

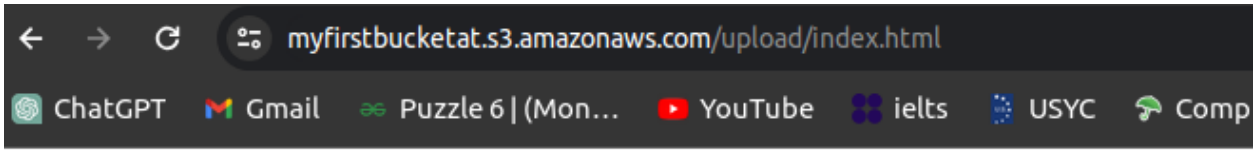
10. Now place the generated JSON in the policy and click save changes

```
1  {
2    "Id": "Policy1705933731096",
3    "Version": "2012-10-17",
4    "Statement": [
5      {
6        "Sid": "Stmt1705933707120",
7        "Action": [
8          "s3:GetObject"
9        ],
10       "Effect": "Allow",
11       "Resource": "arn:aws:s3:::myfirstbucketat",
12       "Principal": "*"
13     }
14   ]
15 }
```

Edit stateme

Select an e

11. Now access the bucked through public URL



Hello Amazon Web Services (AWS)

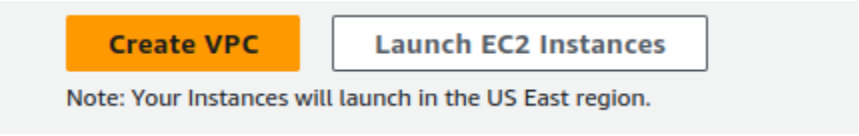
This is a simple HTML page to greet AWS!

3. **VPC Configuration Lab**
   - **Objective**: To understand the fundamentals of AWS networking through the configuration of a Virtual Private Cloud (VPC).
   - **Approach**: Students will create a new VPC, add subnets, set up an Internet Gateway, and configure route tables. The lab might also include setting up a simple EC2 instance within this VPC to demonstrate how resources are deployed in a custom network environment.
   - **Goal**: By the end of this lab, students should be able to create and configure a VPC, understand subnetting, and the role of route tables and internet gateways in AWS.

##

BASIC STEPS

1. **Open AWS Console and search for VPC and click create VPC**



2. **Configure the VPC details in the *VPC settings* panel on the left:**
   Click VPC and more
   Choose VPC and more.
   Under Name tag auto-generation, keep *Auto-generate* selected, however change the value from project to `lab`.
   Keep the IPv4 CIDR block set to 10.0.0.0/16
   For Number of Availability Zones, choose 1.
   For Number of *public* subnets, keep the 1 setting.
   For Number of *private* subnets, keep the 1 setting.

Name tag auto-generation Info

Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

☑ Auto-generate

lab

IPv4 CIDR block Info

Determine the starting IP and the size of your VPC using CIDR notation.

10.0.0.0/16 65,536 IPs

CIDR block size must be between /16 and /28.

IPv6 CIDR block Info

◉ No IPv6 CIDR block

○ Amazon-provided IPv6 CIDR block

Tenancy Info

Default

Number of Availability Zones (AZs) Info

Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

1 | 2 | 3

▶ Customize AZs

Number of public subnets Info

The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 | 1

Number of private subnets Info

The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

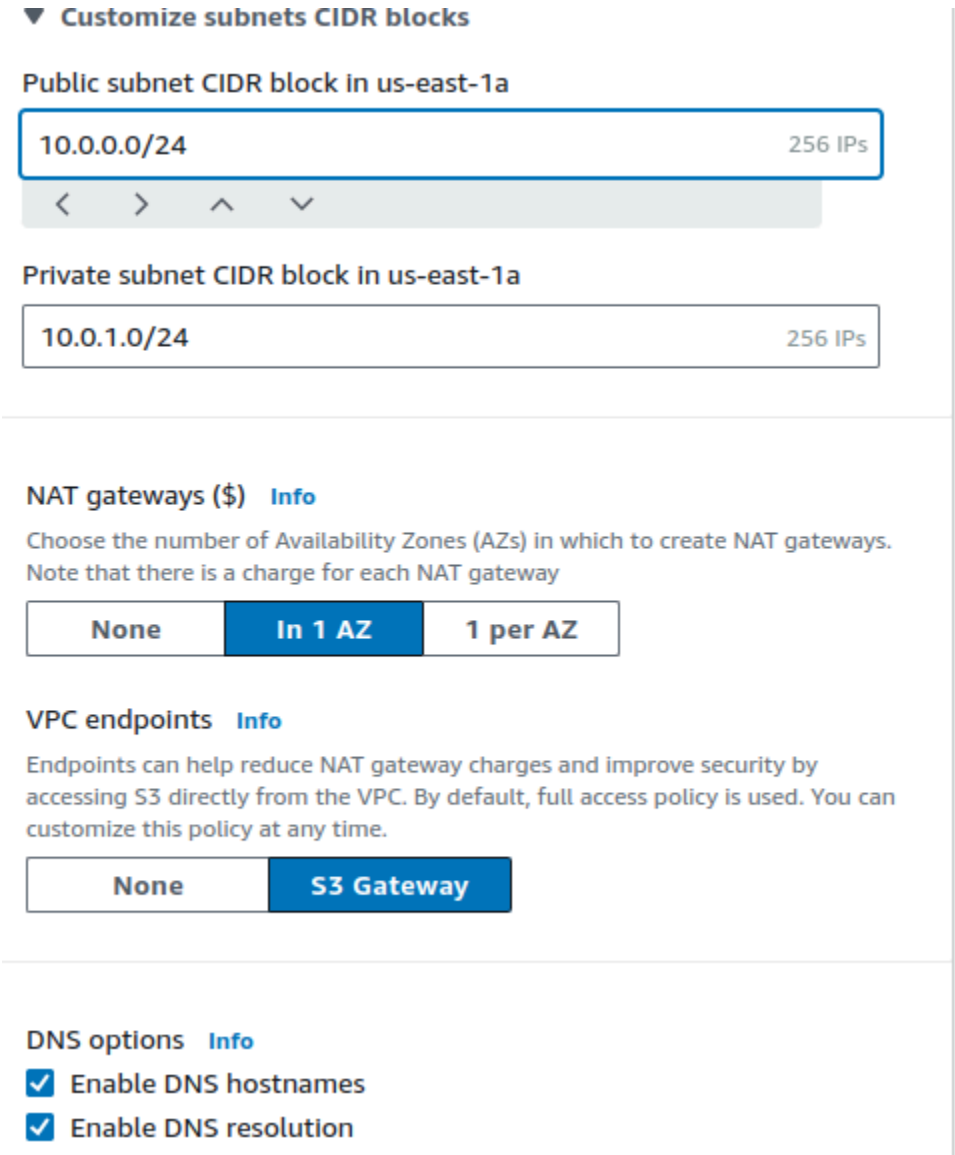
0 | 1 | 2

3. **Expand the Customize subnets CIDR blocks section**

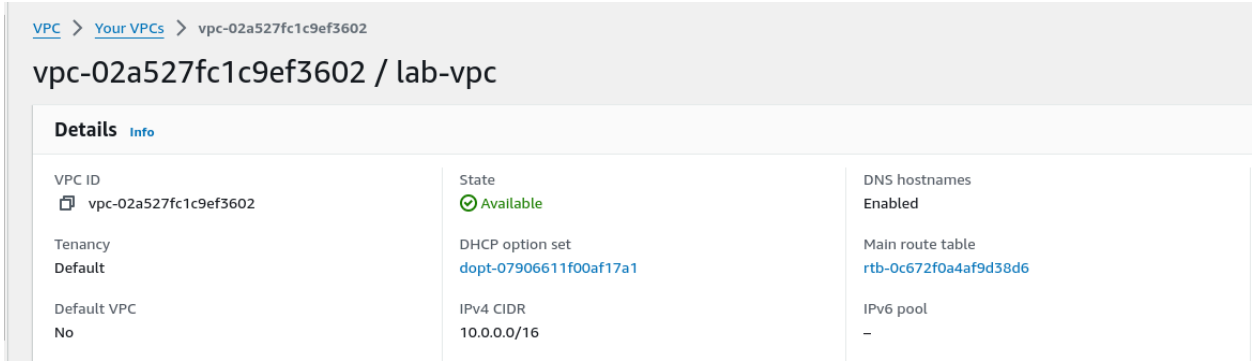
   Change Public subnet CIDR block in us-east-1a to `10.0.0.0/24`

   Change Private subnet CIDR block in us-east-1a to `10.0.1.0/24`

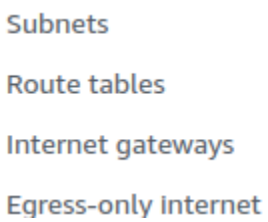
Set NAT gateways to In 1 AZ.
Set VPC endpoints to None.
Keep both DNS hostnames and DNS resolution *enabled*.



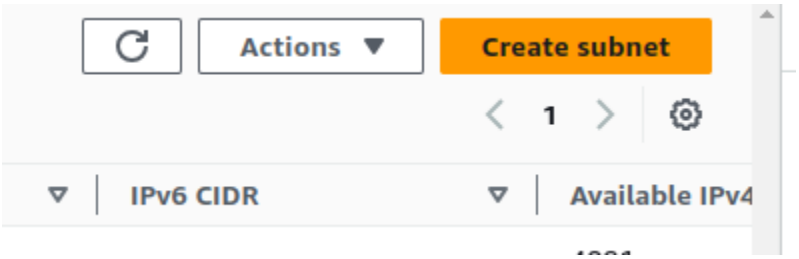
4. **Now click to create VPC the VPC will be created after some time**



5. **To add subnet click to subnet in left panel**

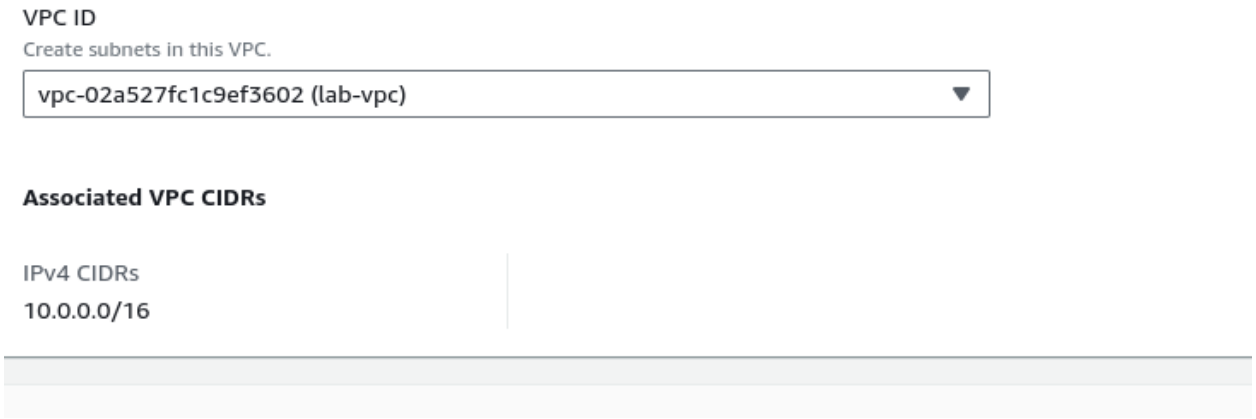


6. **Click to create subnet**



7.

7. Select the created VPC



8. **Now give subnet name, availability zone, and add the IPv4-CIDR subnet**

Subnet 1 of 1

Subnet name

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

lab-subnet_public2

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.

US East (N. Virginia) / us-east-1b

IPv4 VPC CIDR block Info

Choose the IPv4 VPC CIDR block to create a subnet in.

10.0.0.0/16

IPv4 subnet CIDR block

10.0.2.0/24

▼ Tags - *optional*

Again do same to create another subnet for with different CIDR block id you can see newly created private and public subnet

☐	lab-subnet-private1-us-east-1a	subnet-0c603c3
☐	lab-subnet-public1-us-east-1a	subnet-03188ab
☐	lab-subnet_public2	subnet-022f5c2
☐	lab-subnetprivate2	subnet-04b0887

9. **To setup the internet gateway select my vpc and in the left bar click internet gateway click create gateway to create**

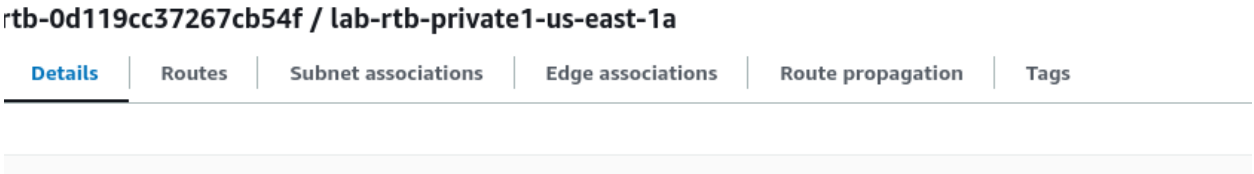
Actions ▼ | Create internet gateway

1

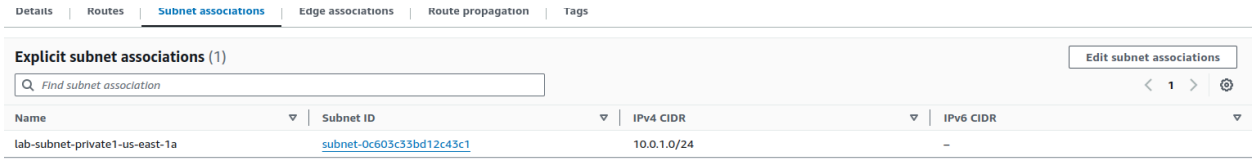
Owner

10. Now to configure the route table click route tables in the left navigation panel and select lab-rtb-private1-us-east-1a

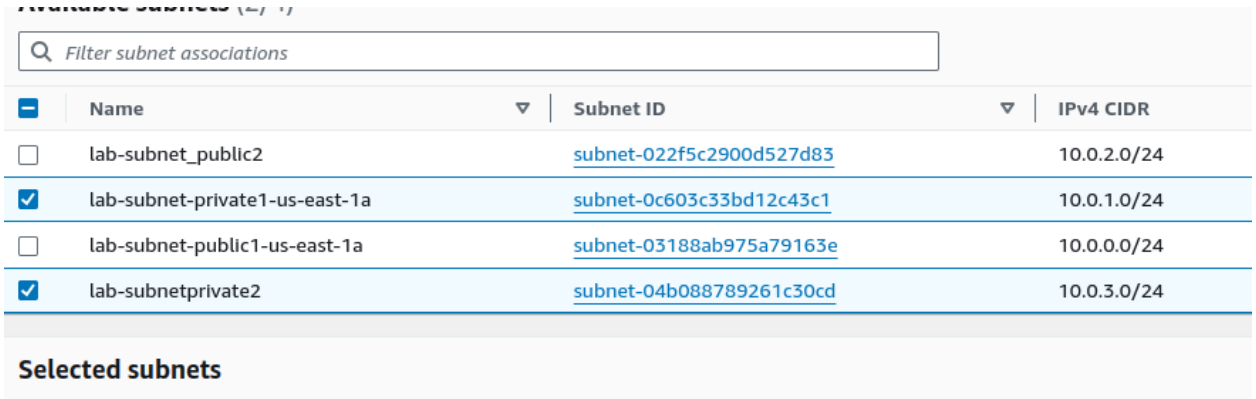
11. In the lower panel choose routes



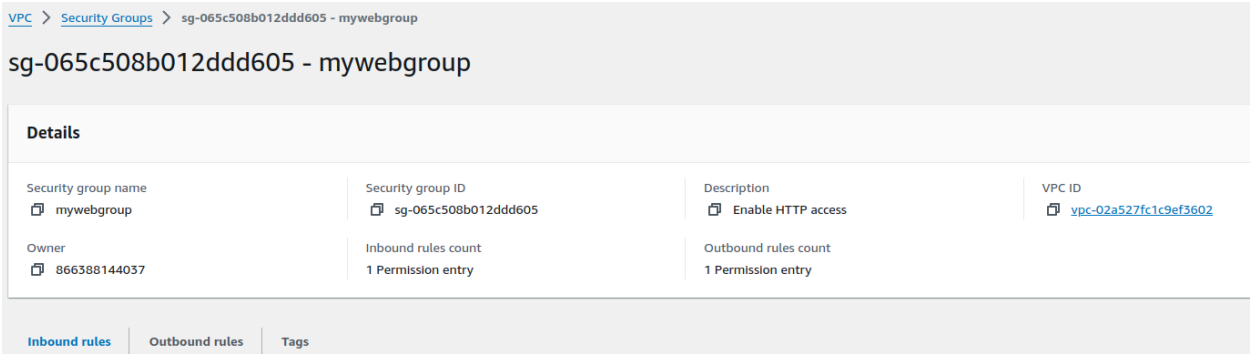
12. Choose subnet association and click edit subnet association and click explicit subnet association



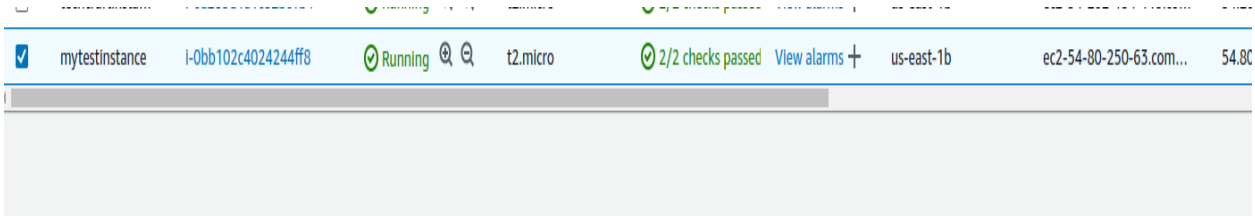
13. Leave lab-rtb-private1-us-east-1a selected and select lab-subnetprivate2 and do same for public subnet



14. Create security group with necessary rule for the vpc



15. **Now create a EC2 instance with the created security group and the vpc And launch by some code in user data box and go back to the cretated instance when status shows 2 check pass select the instance and get the public IP**



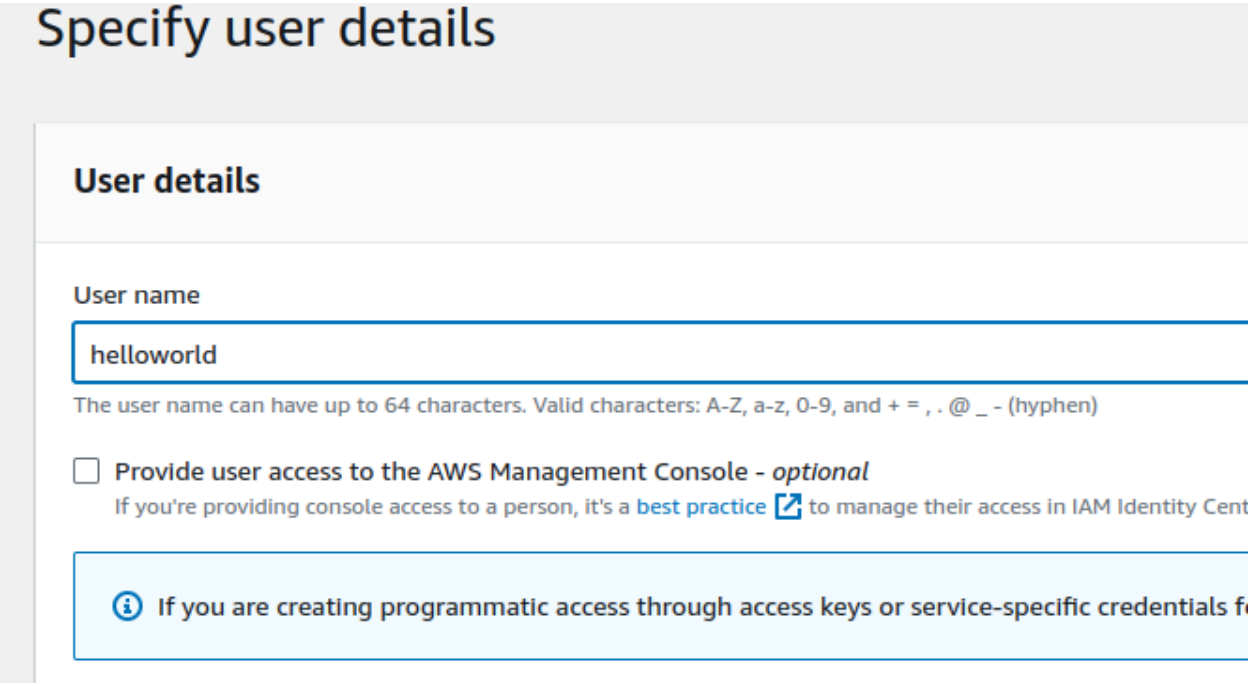
4. **IAM Users and Roles Lab**
   - **Objective**: To understand AWS Identity and Access Management (IAM) by creating and managing users, groups, and roles.
   - **Approach**: Students will create new IAM users, assign them to groups, and apply policies to manage permissions. The lab will also involve creating roles for AWS services and understanding the use of IAM roles for cross-service access.
   - **Goal**: Students will learn about user and permission management in AWS, the importance of roles for security and best practices for IAM.

##

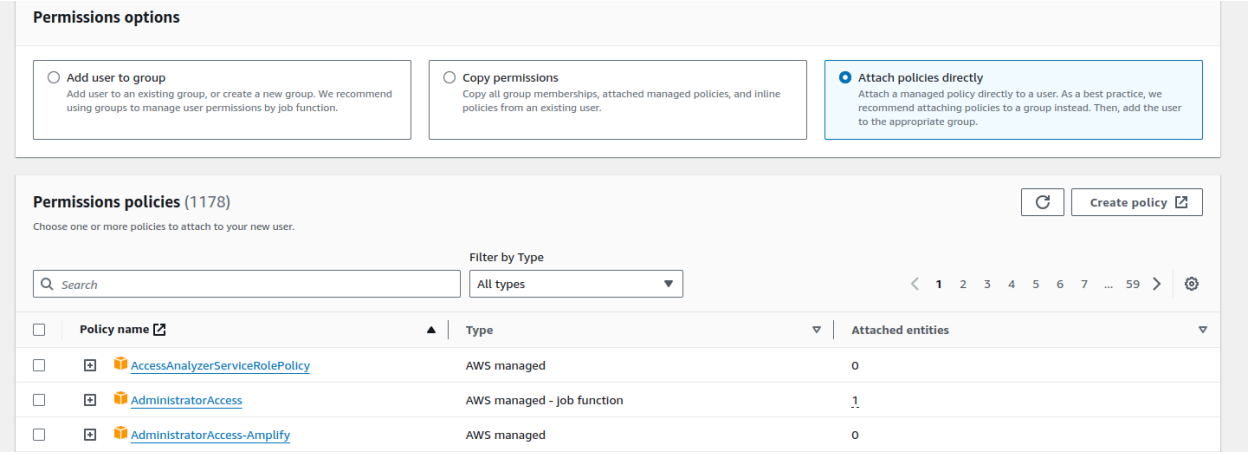
BASIC STEPS

We Can't create any groups and assign the policy, so I used the foundation course lab to assign the user with groups

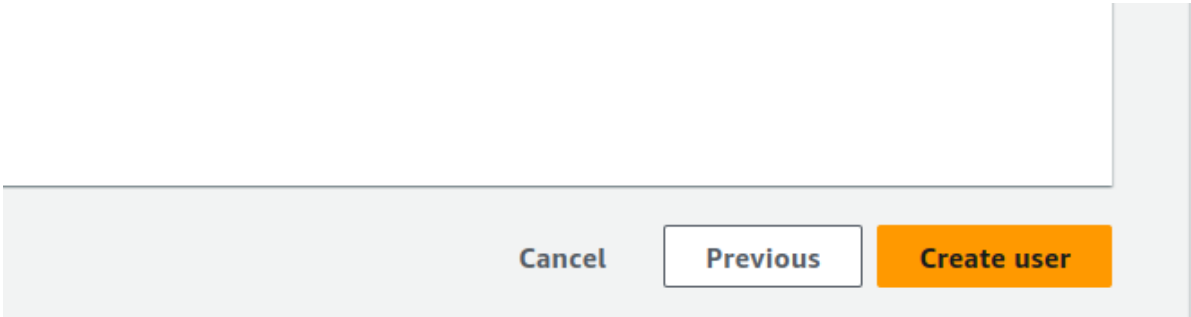
1. **Create a IAM User first, to create this, open the AWS Console Management tab and search for IAM. Click the User on the left side and create IAM user with necessary attribute**



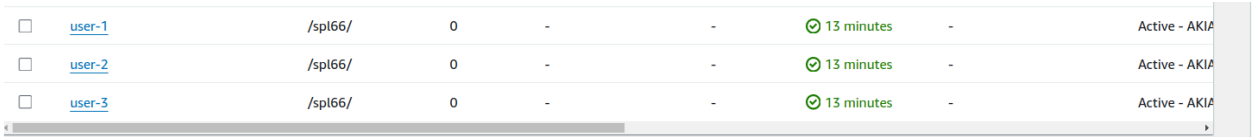
2. **Attach the policy for that user, you have to select attach policies directly radio button to do so**



3. **Click to create user to create user**

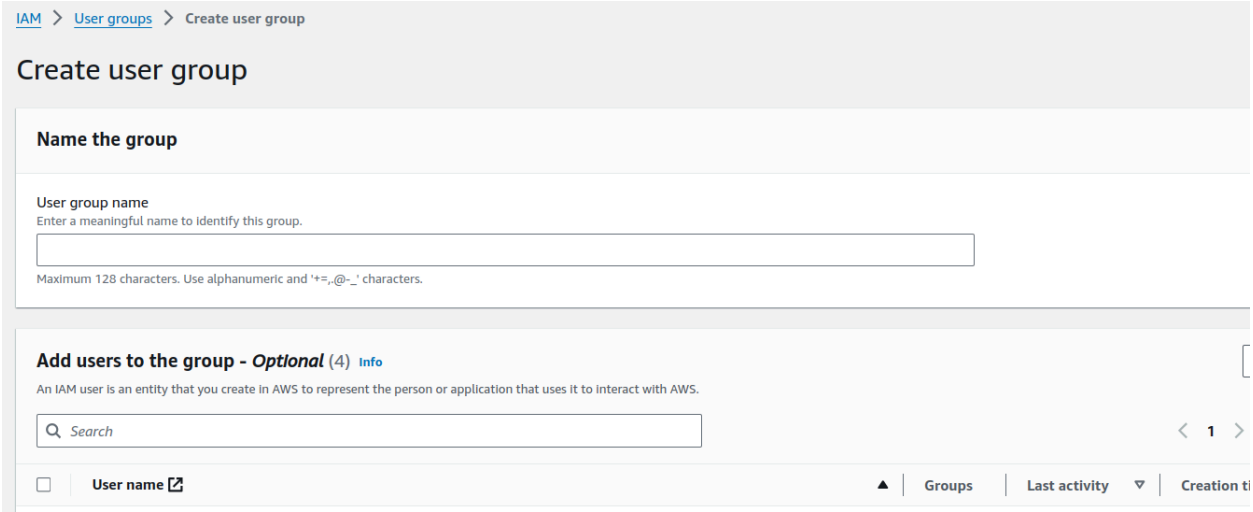


These are the users

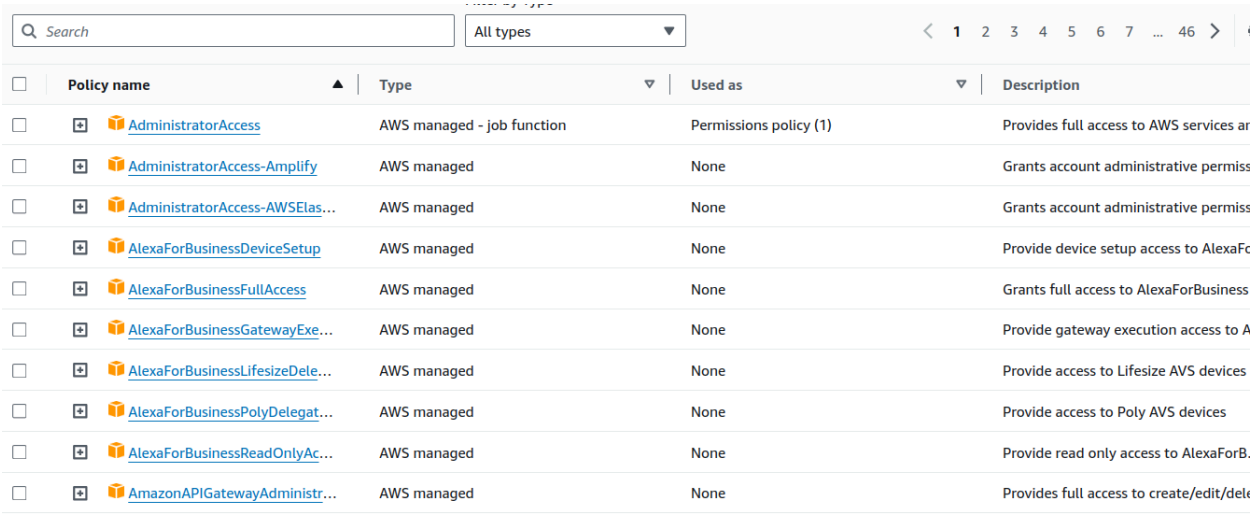


☐	user-1	/spl66/	0	-	-	13 minutes	-	Active - AKIA
☐	user-2	/spl66/	0	-	-	13 minutes	-	Active - AKIA
☐	user-3	/spl66/	0	-	-	13 minutes	-	Active - AKIA

4. **For creating groups, click the group section in the left nav bar and select create groups**

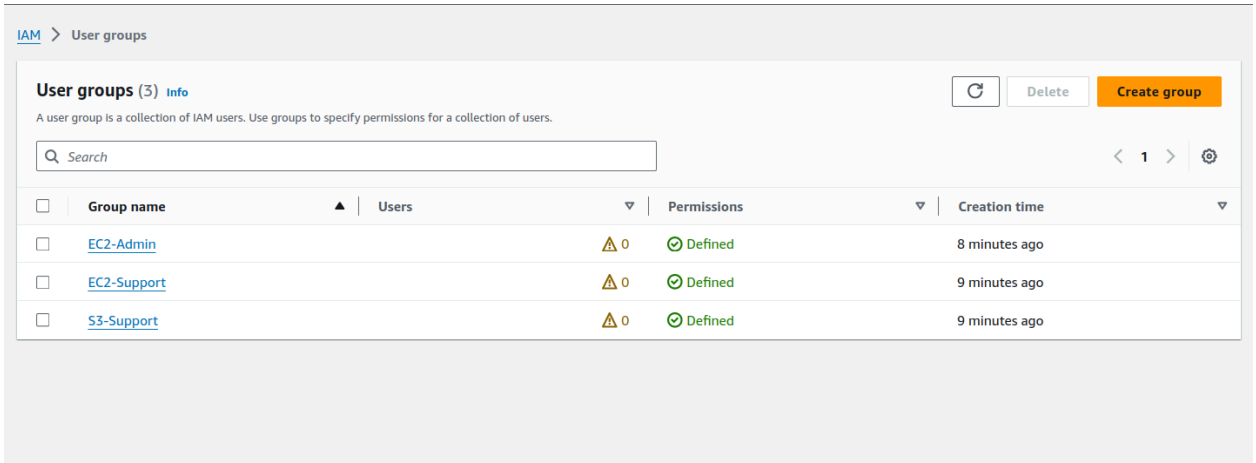


Again for the group we can assign policy directly by selecting the available policy

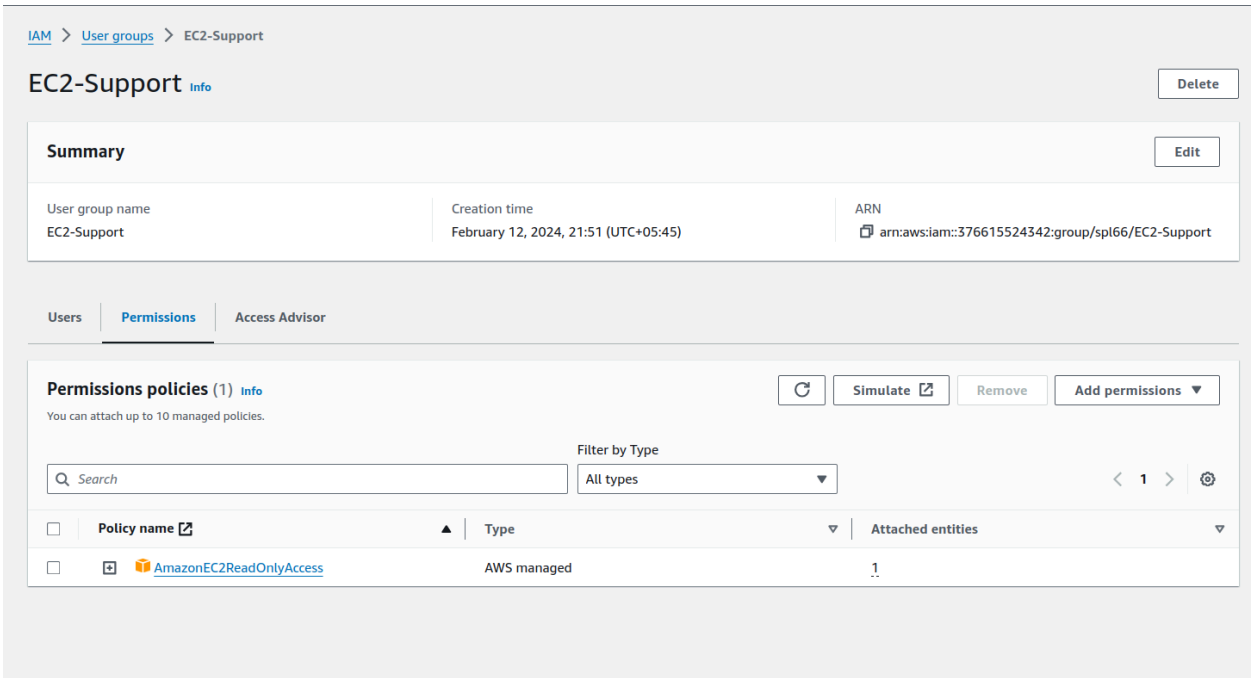


	Policy name	Type	Used as	Description
☐	AdministratorAccess	AWS managed - job function	Permissions policy (1)	Provides full access to AWS services ar
☐	AdministratorAccess-Amplify	AWS managed	None	Grants account administrative permiss
☐	AdministratorAccess-AWSElas...	AWS managed	None	Grants account administrative permiss
☐	AlexaForBusinessDeviceSetup	AWS managed	None	Provide device setup access to AlexaFc
☐	AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBusiness
☐	AlexaForBusinessGatewayExe...	AWS managed	None	Provide gateway execution access to A
☐	AlexaForBusinessLifesizeDele...	AWS managed	None	Provide access to Lifesize AVS devices
☐	AlexaForBusinessPolyDelegat...	AWS managed	None	Provide access to Poly AVS devices
☐	AlexaForBusinessReadOnlyAc...	AWS managed	None	Provide read only access to AlexaForB.
☐	AmazonAPIGatewayAdministr...	AWS managed	None	Provides full access to create/edit/dele

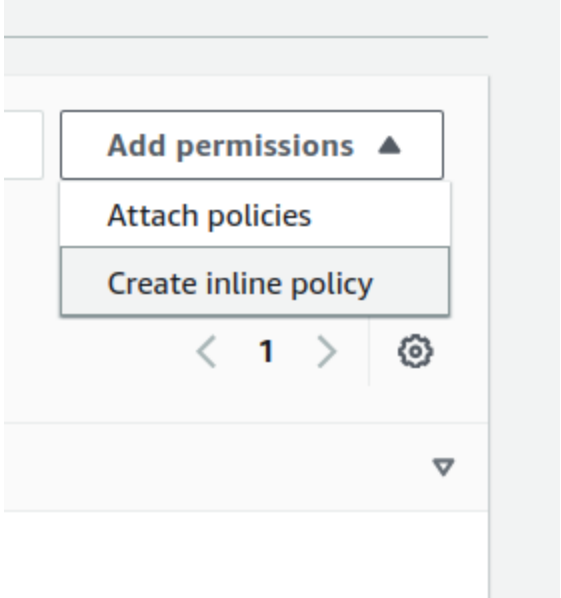
These are the pre created group, now we have to assign the users to it



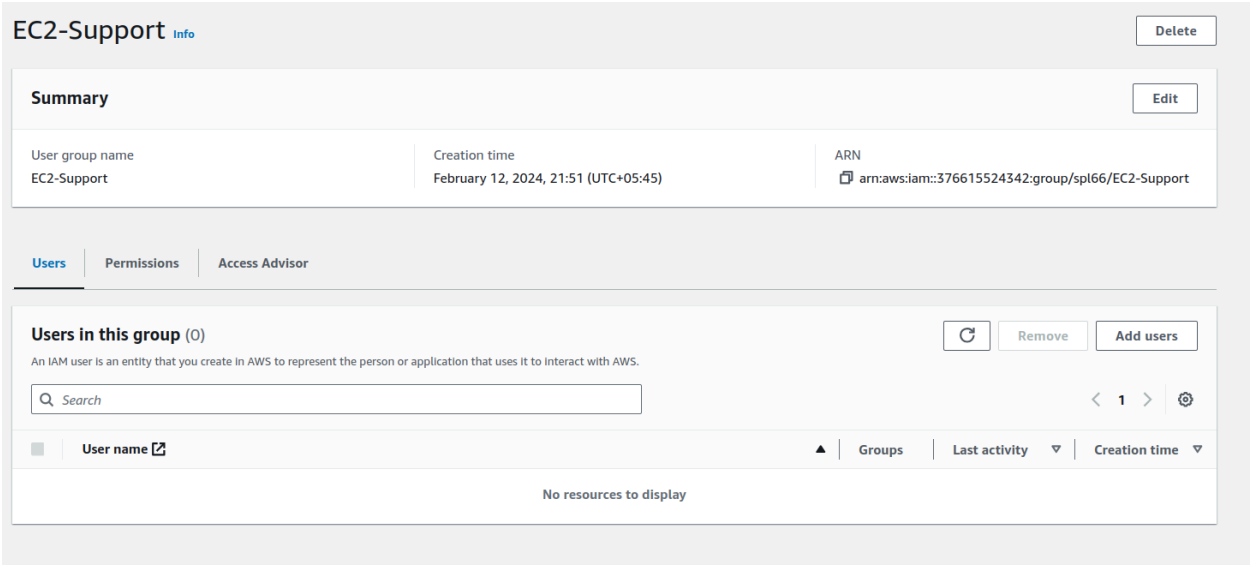
5. **Go to the user group which is under the users in the left panel,click the group name**



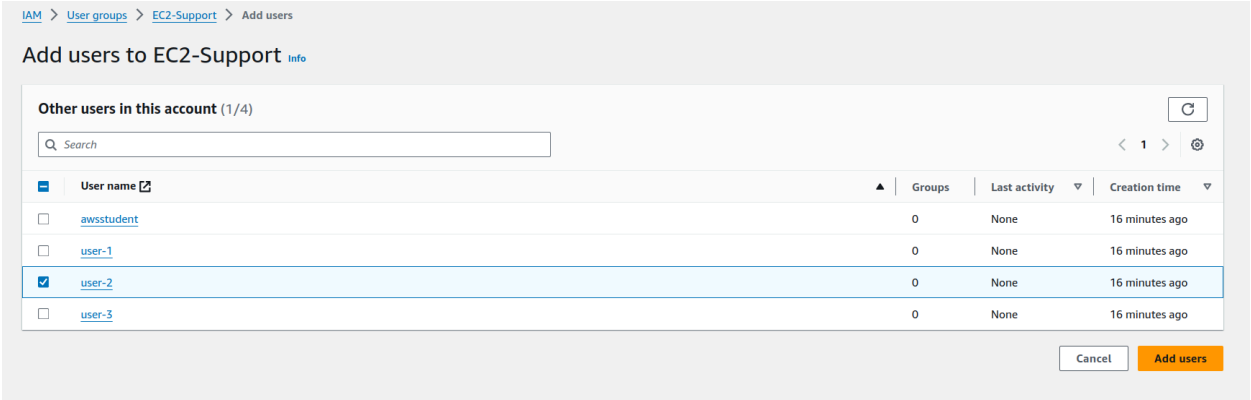
Now click the permission tab as you can see the group has read only. We can again reassign the policy by clicking add permission but here we are not allowed to do that



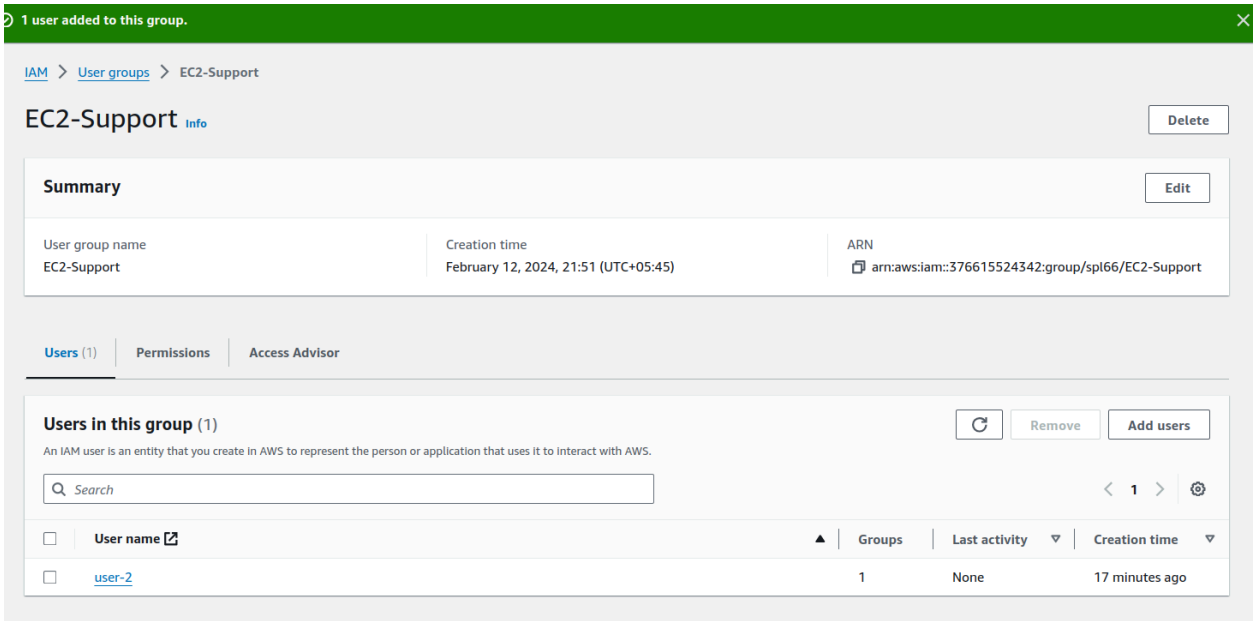
6. **Now click the user tab and click add user**



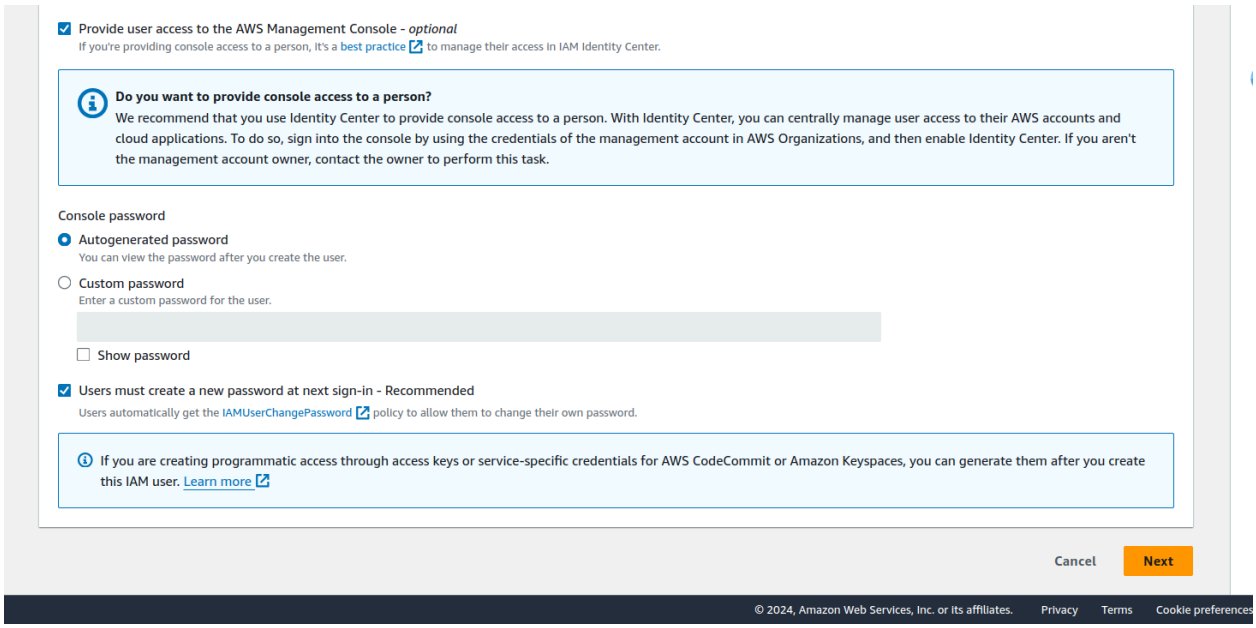
And select the user you want to assign to the particular group, after selecting the user click add user



You can see the user is added to the group



Now we can signout from aws and use the created user as a userid,generated pasword and login with the assign roles within a group using the



We can auto generate password or use custom password when creating the user