**Tasks on IAM**

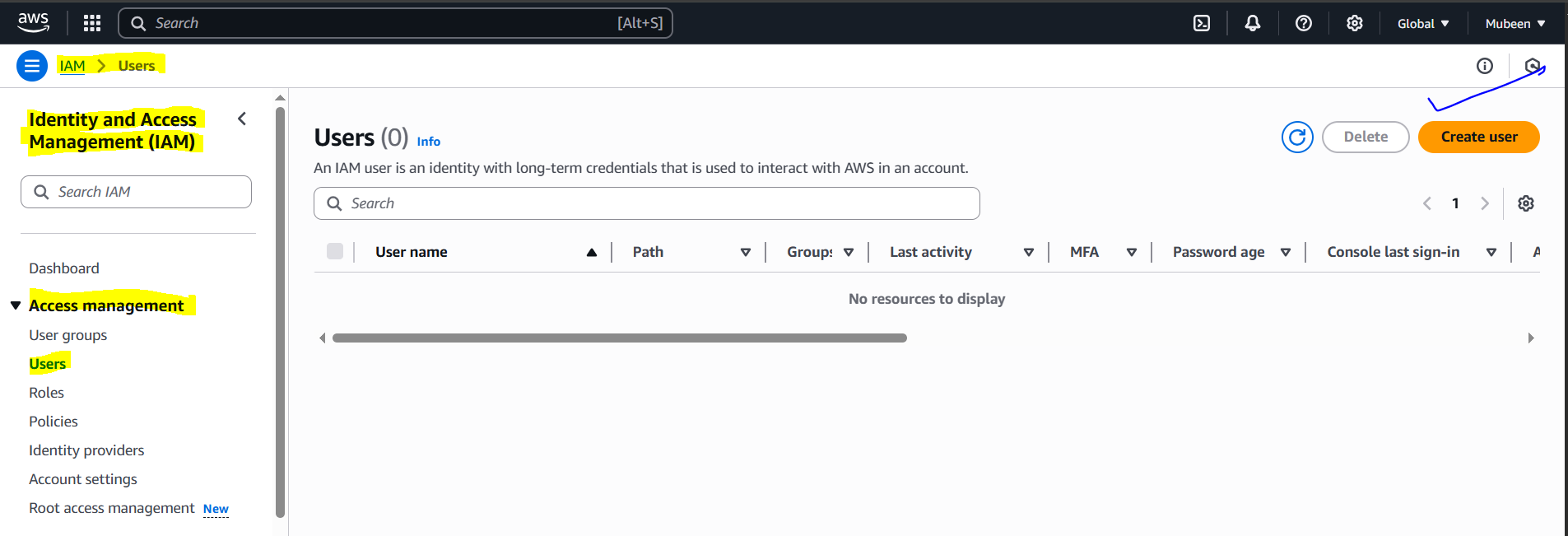
**1) Create one IAM user and assign ec2, s3 full access role.   
  
Step 1: Create an IAM User**

1. Log in to the AWS Management Console.

2. Navigate to the IAM dashboard.

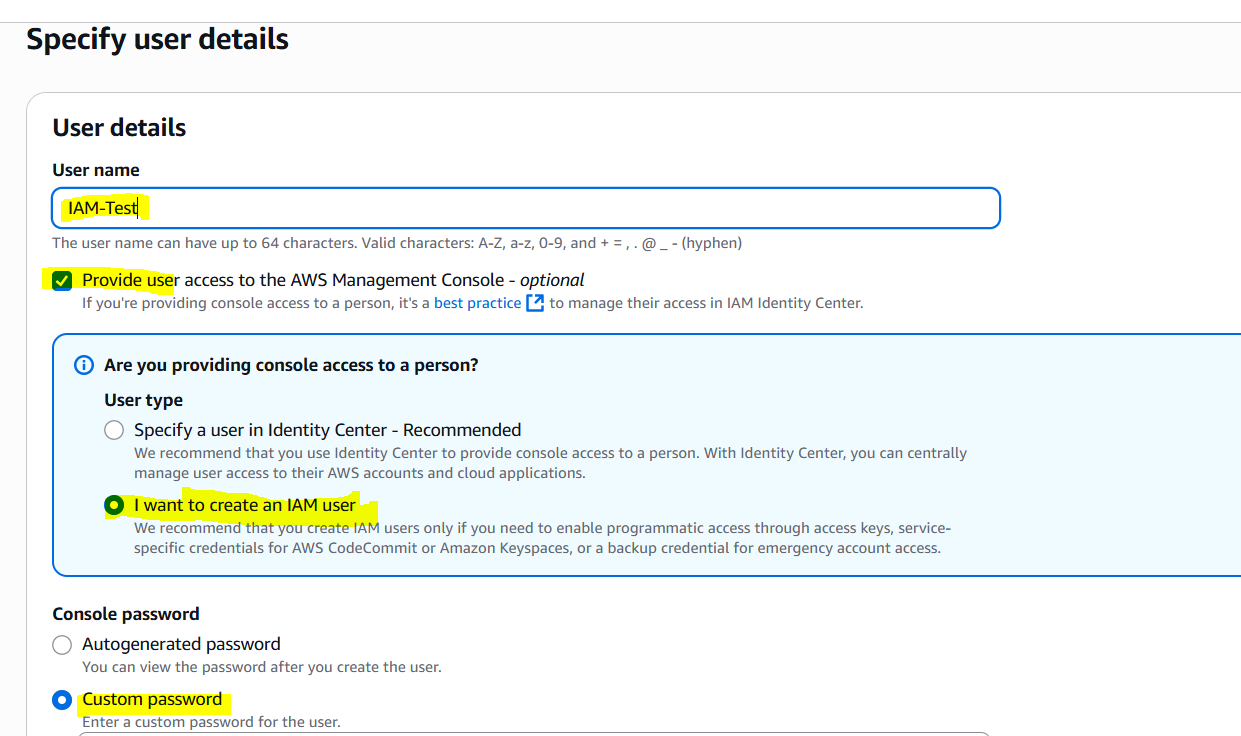
3. Click on Users in the left sidebar.

4. Click on Create user.

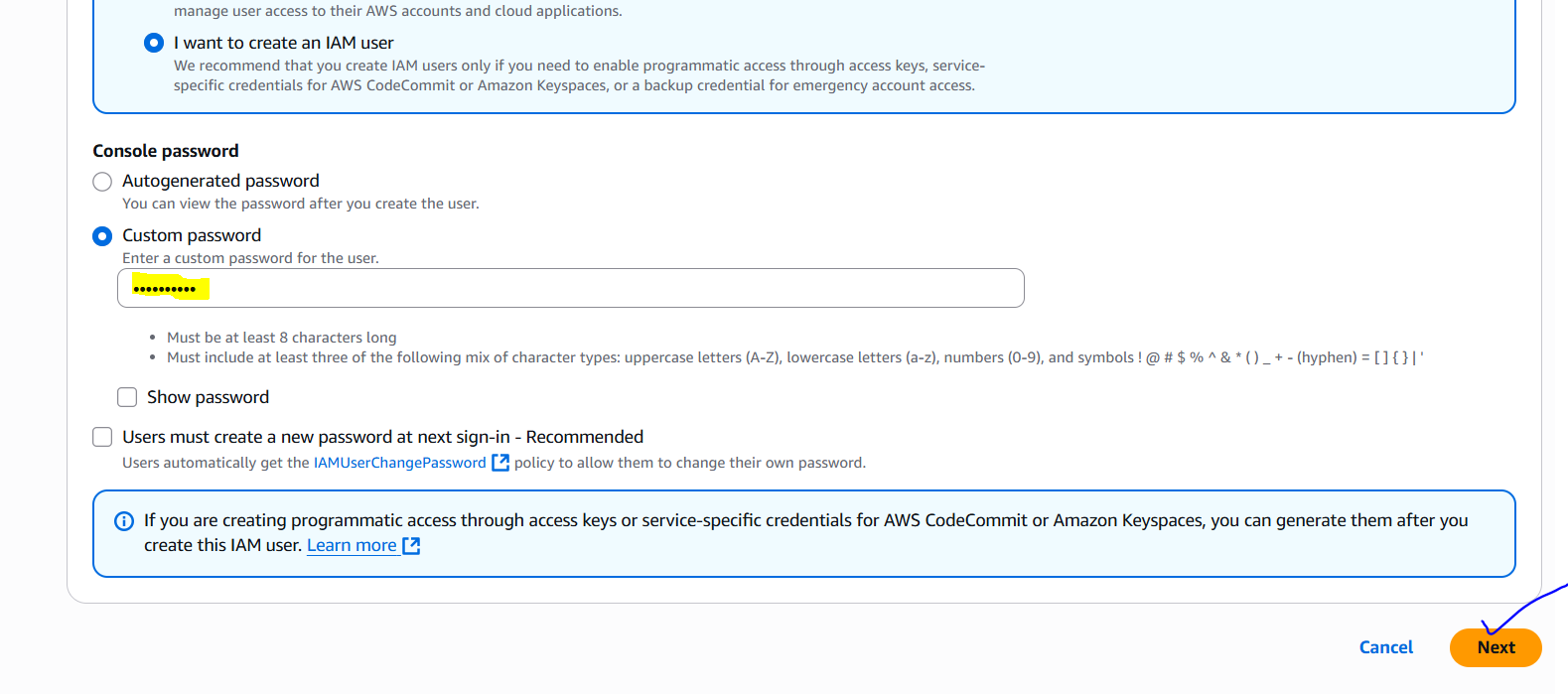


5. Choose Programmatic access(for aws cli) or AWS Management Console access (or both) depending on your needs.

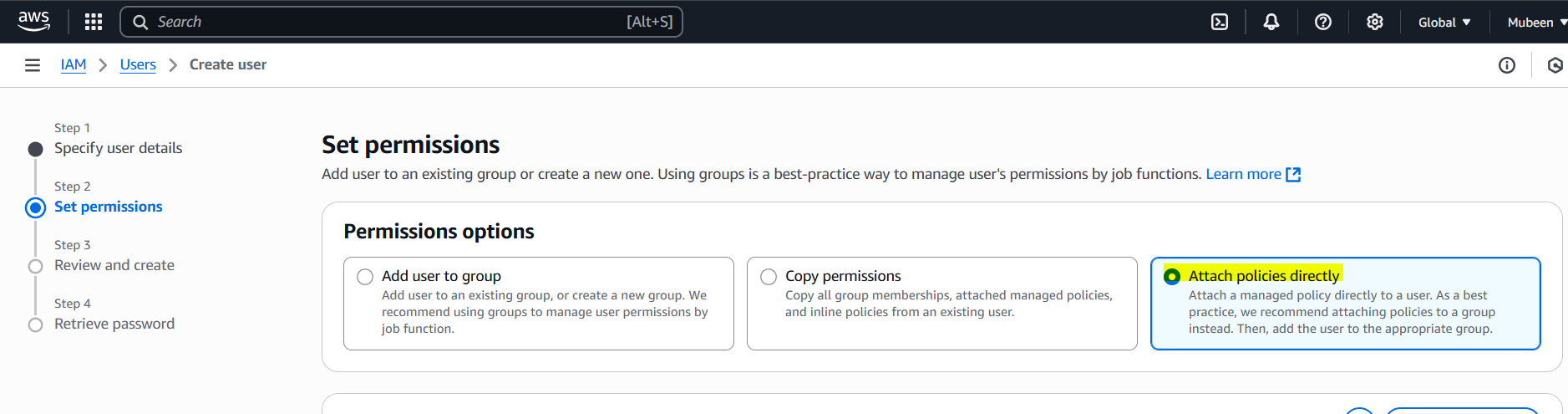
6. Enter a Username with custom password.



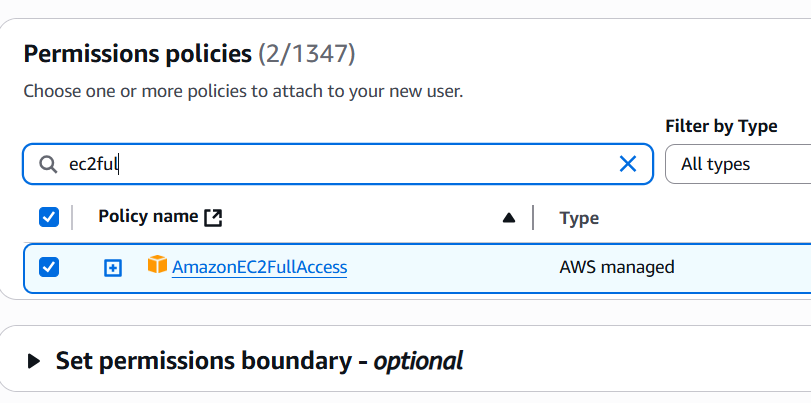
7. Click Next: Permissions.



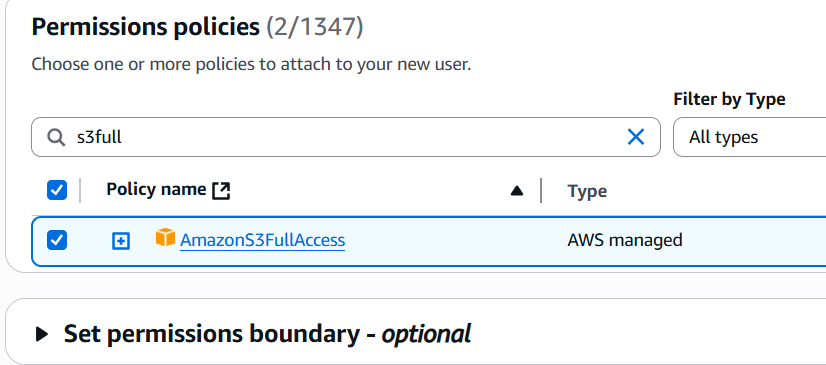
**Step 2: Attach Policies**

1. Click on Attach policy directly.  
  


2. Search for AmazonEC2FullAccess and select it.



3. Search for AmazonS3FullAccess and select it.

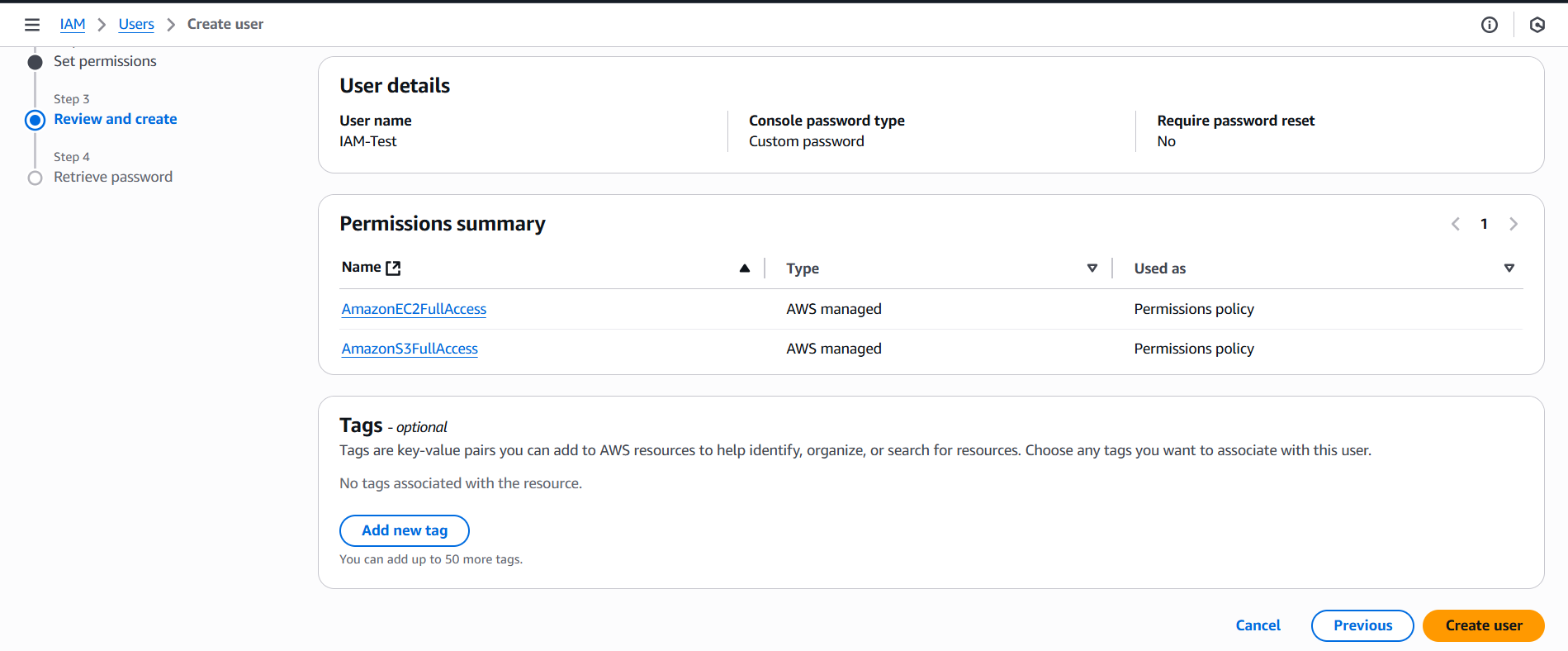


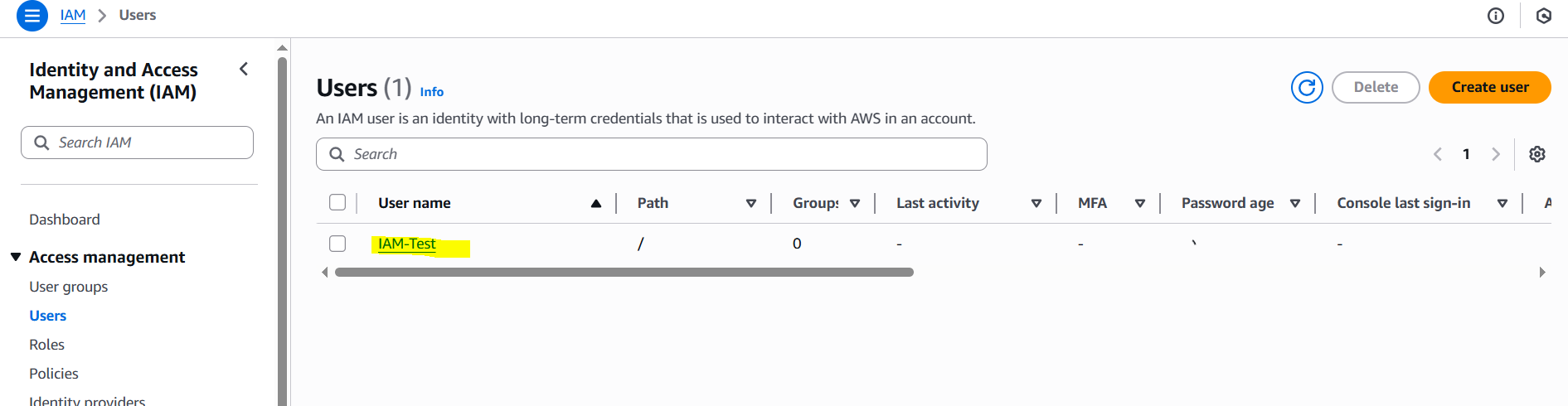
4. Click Next: Review.

Step 3: Review and Create

1. Review the user details and policies.

2. Click Create user.





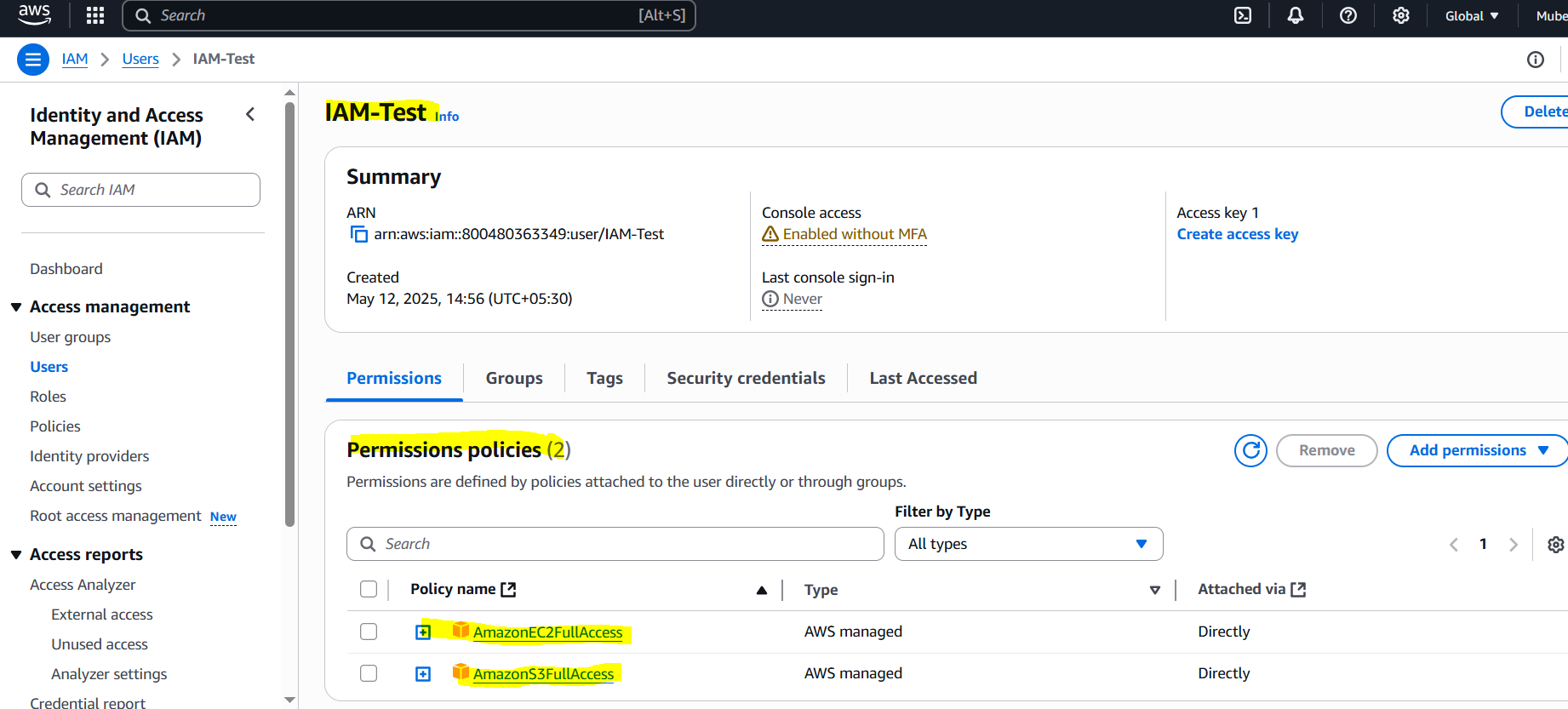
**Step 4: Verify Permissions**

1. Go to the Users page.

2. Find the newly created user.

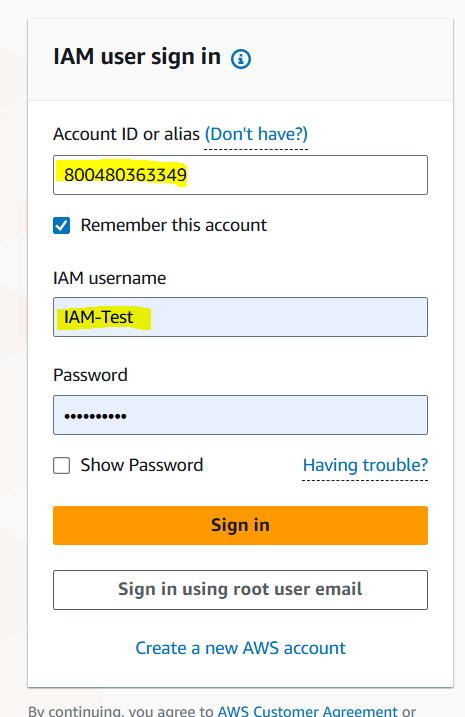
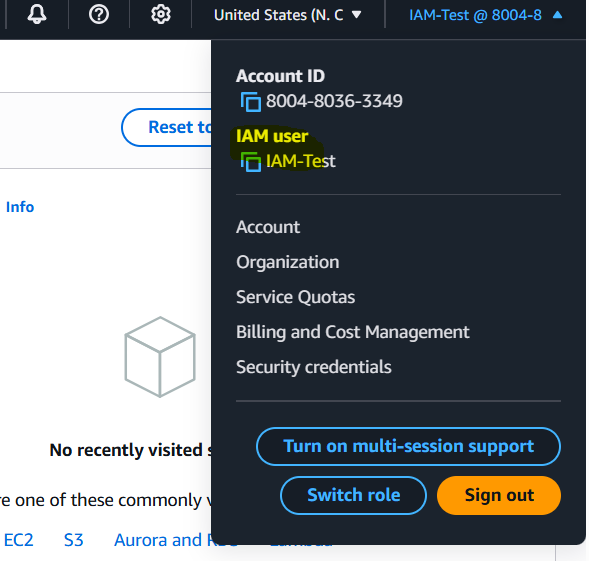
3. Click on the Permissions tab.

4. Verify that AmazonEC2FullAccess and AmazonS3FullAccess policies are attached.

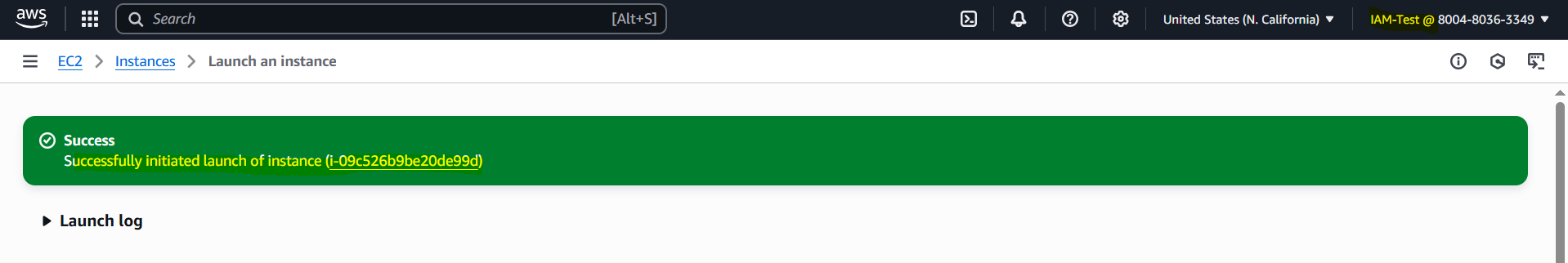


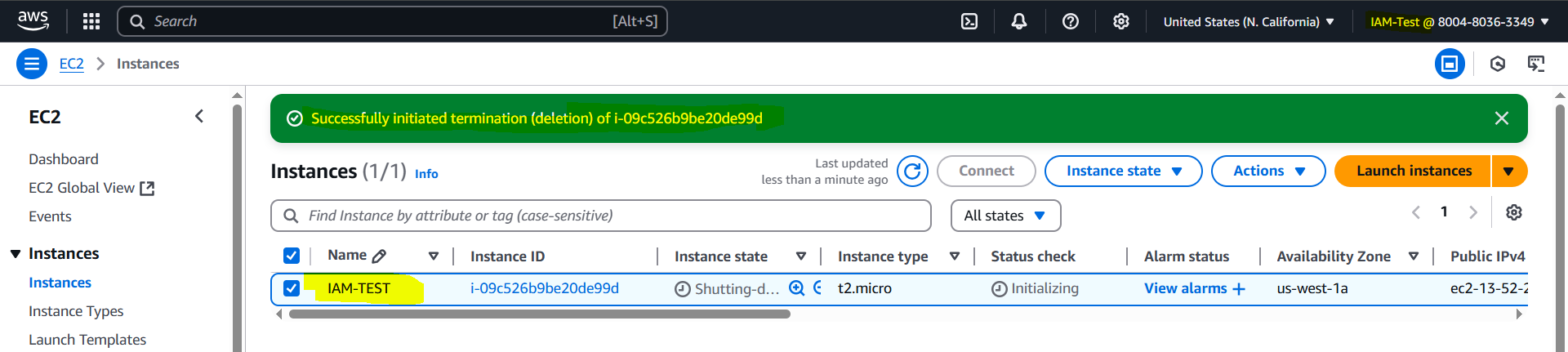
Your IAM user now has full access to EC2 and S3. You can use this user's credentials to access these services.

Use aws Account id, user name and password to login as user.

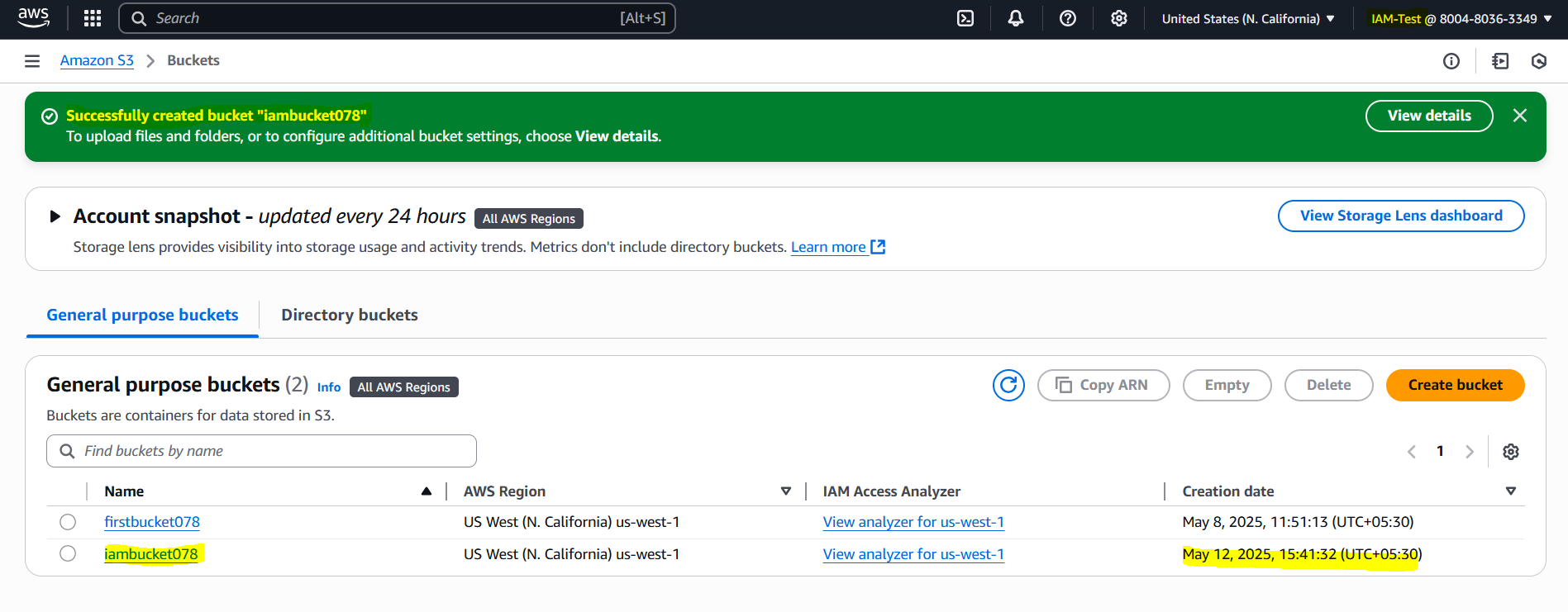
 

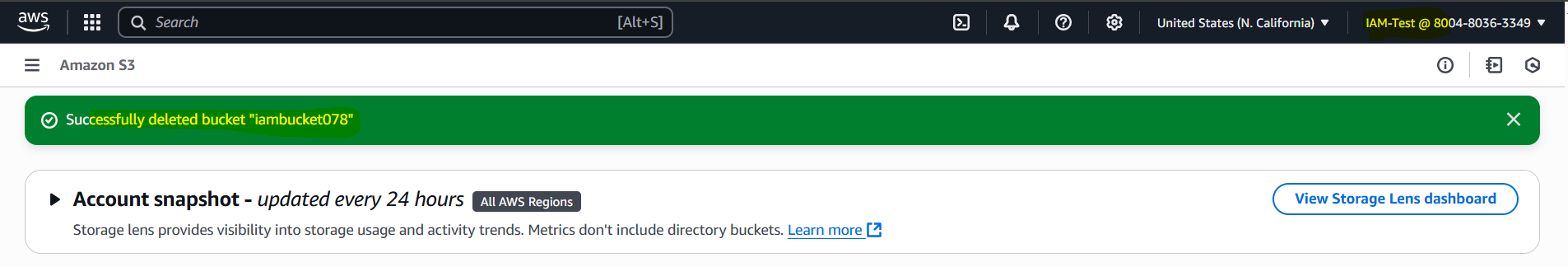
Now we have full access for S3 Ec2 in IAM-Test user.





Now we have full access for S3 Bucket in IAM-Test user.



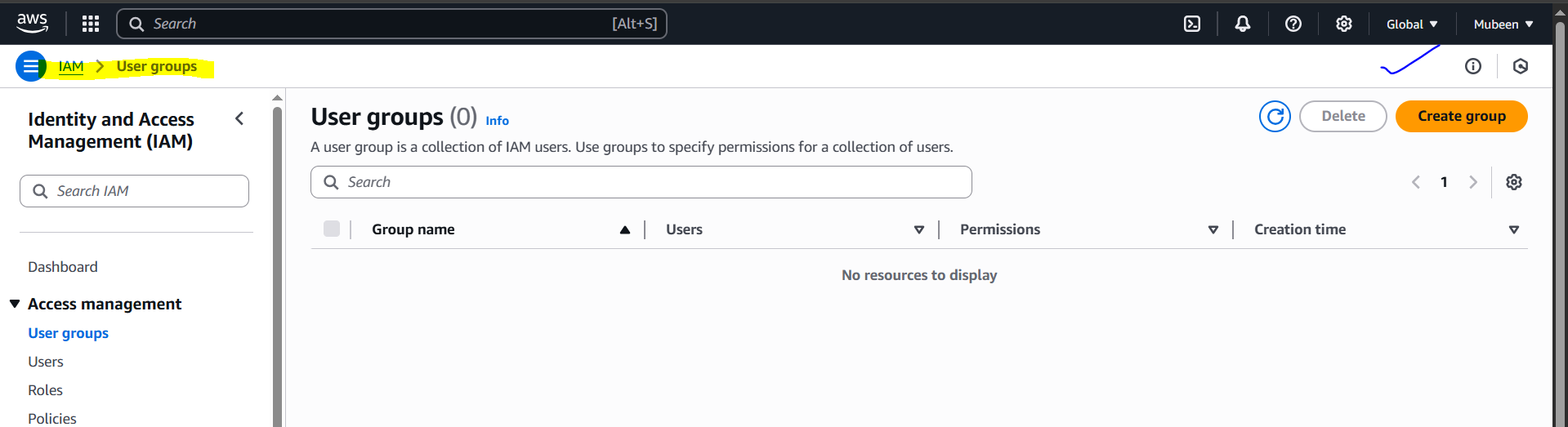


**2) Create one Group in IAM and Assign Read access for ec2.   
  
Step 1: Create an IAM Group**

1. Log in to the AWS Management Console.

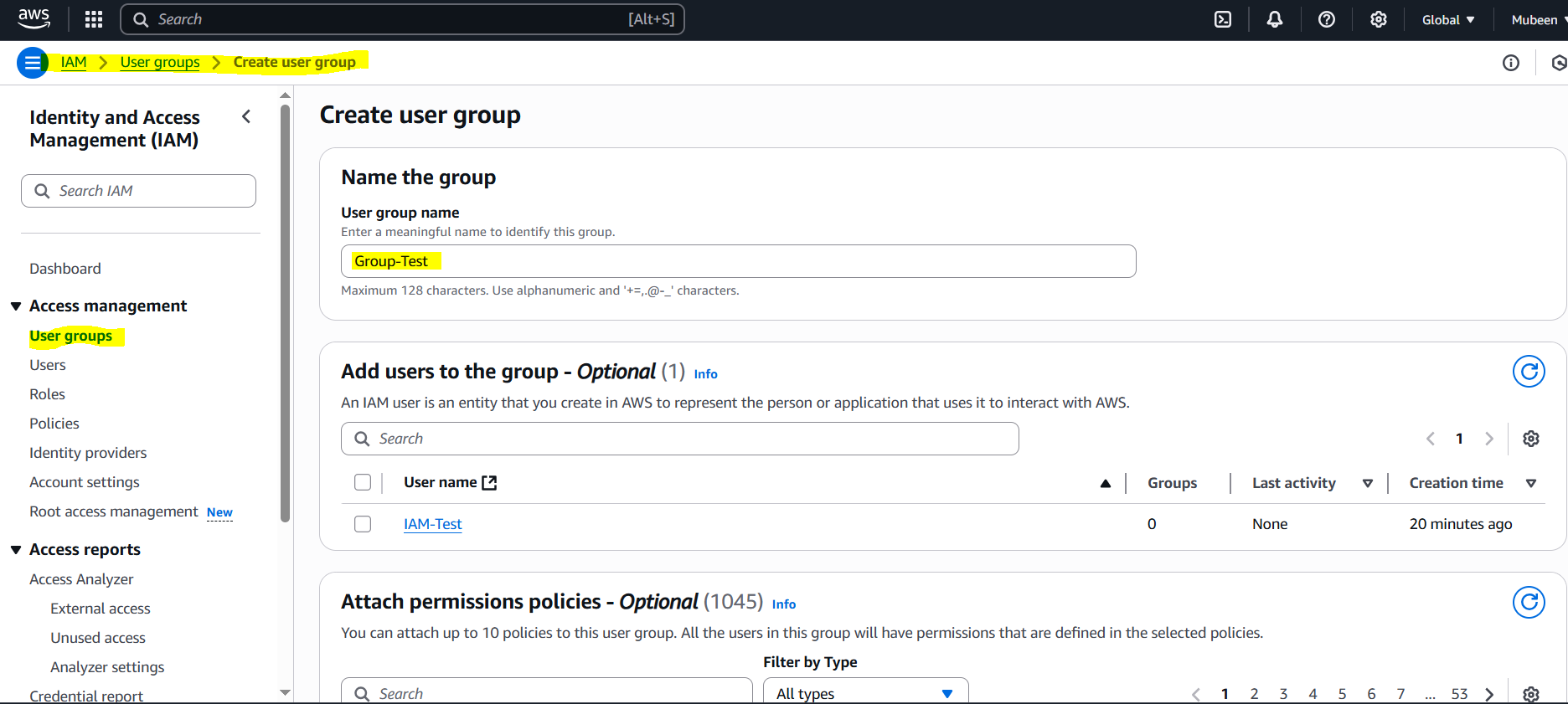
2. Navigate to the IAM dashboard.

3. Click on Groups in the left sidebar.



4. Click on Create group.

5. Enter a Group name.



**Step 2: Attach Policy**

1. Search for AmazonEC2ReadOnlyAccess policy.

2. Select the policy.

3. Click Create group.



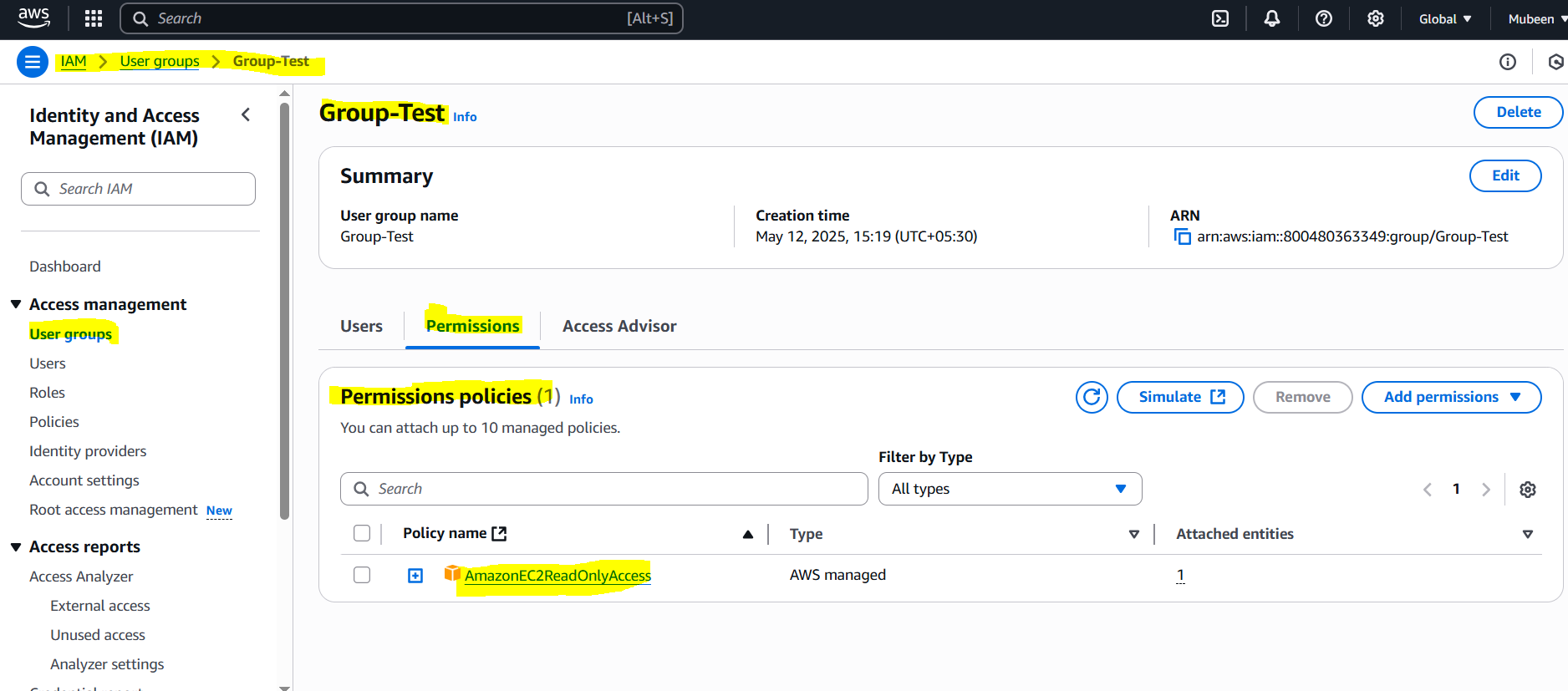
**Step 3: Verify Group Permissions**

1. Go to the Groups page.

2. Find the newly created group.

3. Click on the Permissions tab.

4. Verify that AmazonEC2ReadOnlyAccess policy is attached.



Your IAM group now has read-only access to EC2. You can add users to this group to grant them EC2 read-only permissions.

**3) Create a new user with name Devops and add to the group created in task2.   
  
Step 1: Create a New IAM User**

1. Log in to the AWS Management Console.

2. Navigate to the IAM dashboard.

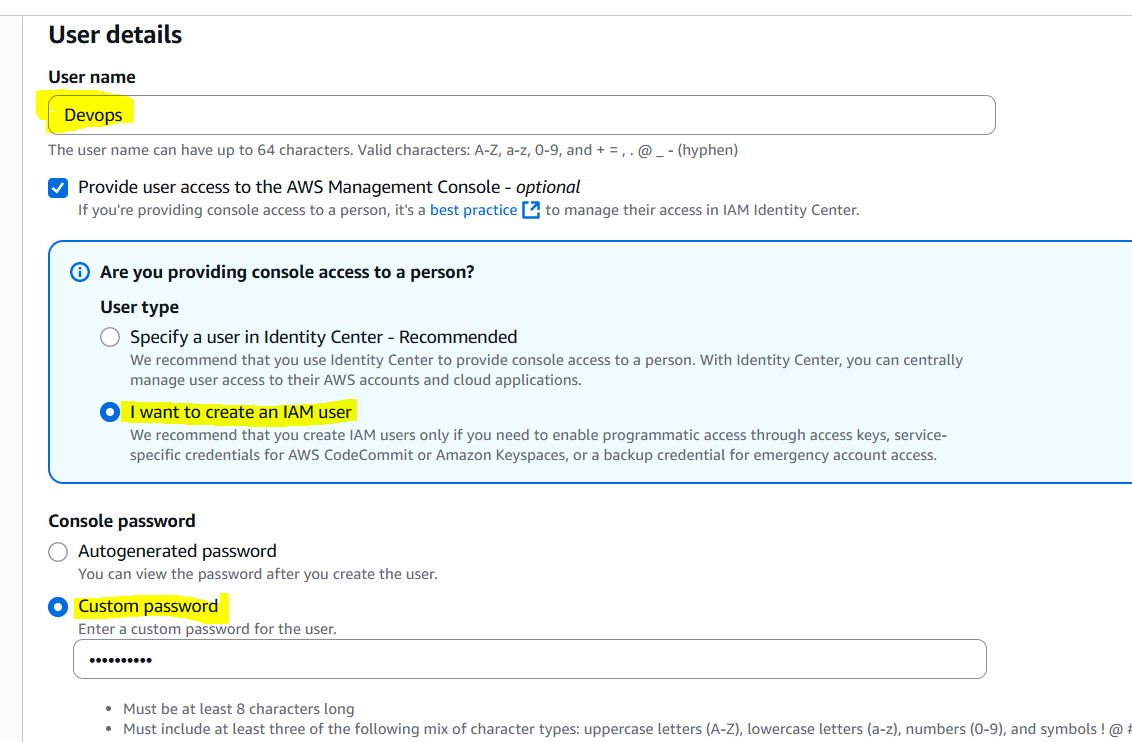
3. Click on Users in the left sidebar.

4. Click on Create user.

5. Choose Programmatic access or AWS Management Console access (or both) depending on your needs.

6. Enter Devops as the username with Custom password.

7. Click Next: Permissions.



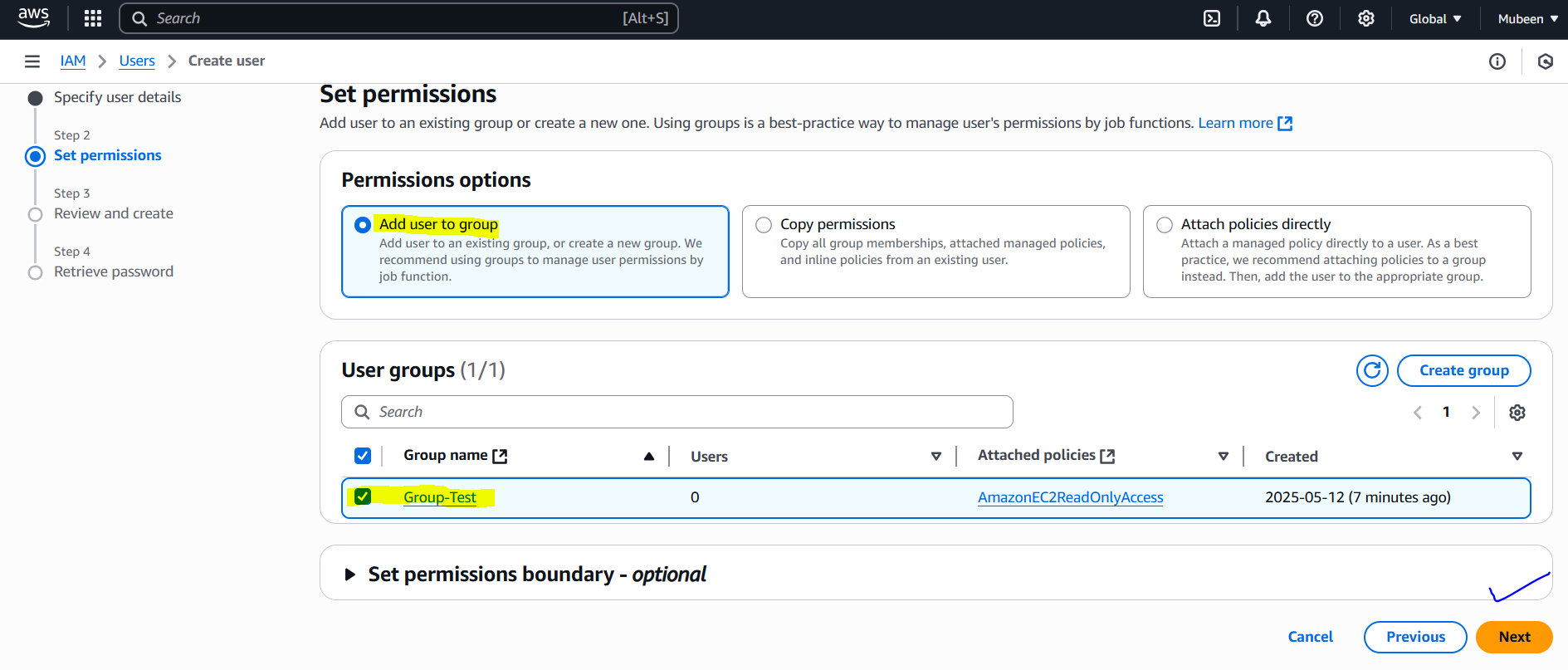
**Step 2: Add User to Group**

1. Click on Add user to group.

2. Select the Group-Test group.

3. Click Next: Tags (optional).

4. Click Next: Review.



**Step 3: Review and Create**

1. Review the user details.

2. Click Create user.



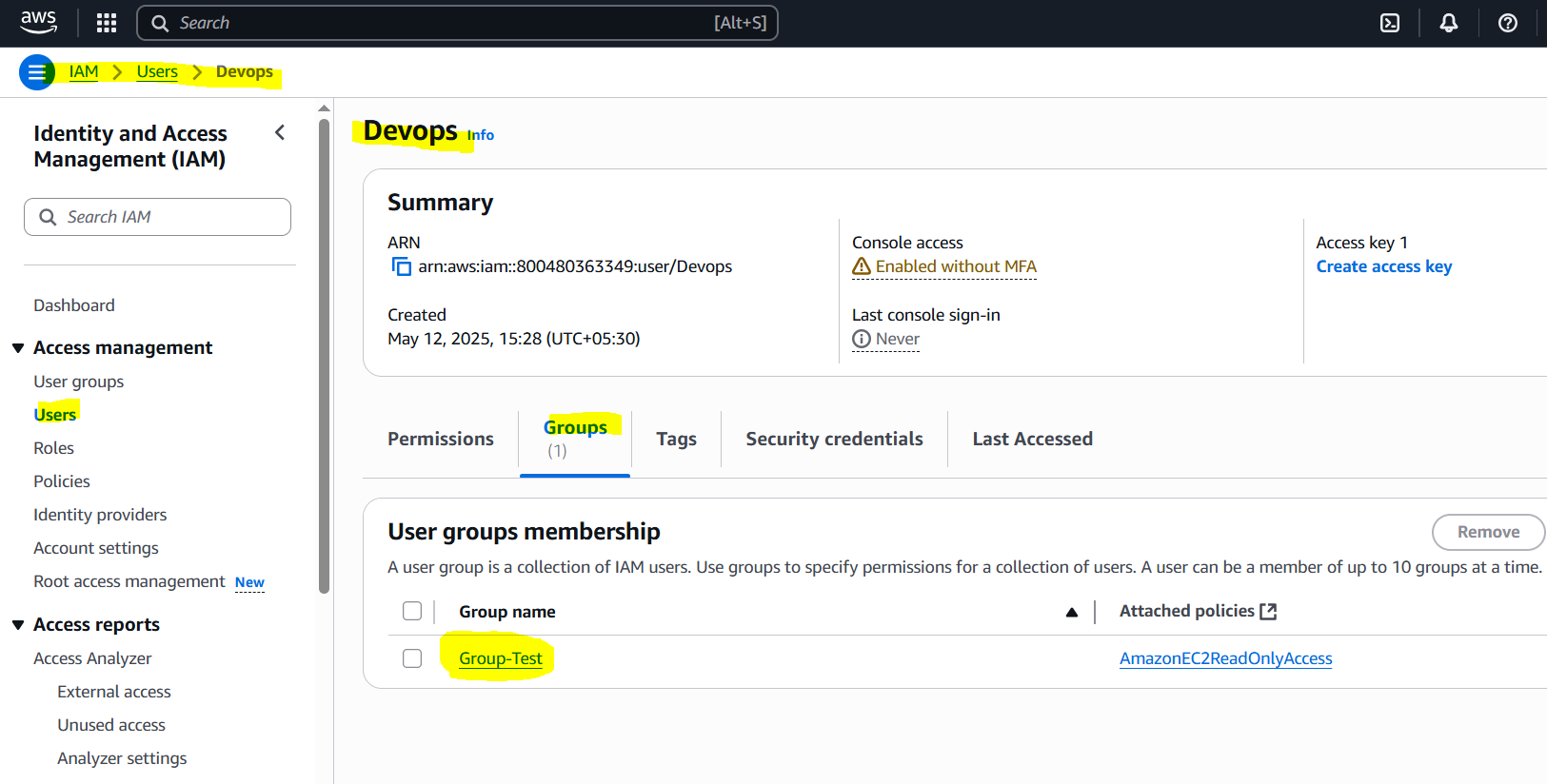
**Step 4: Verify Group Membership**

1. Go to the Users page.

2. Find the Devops user.

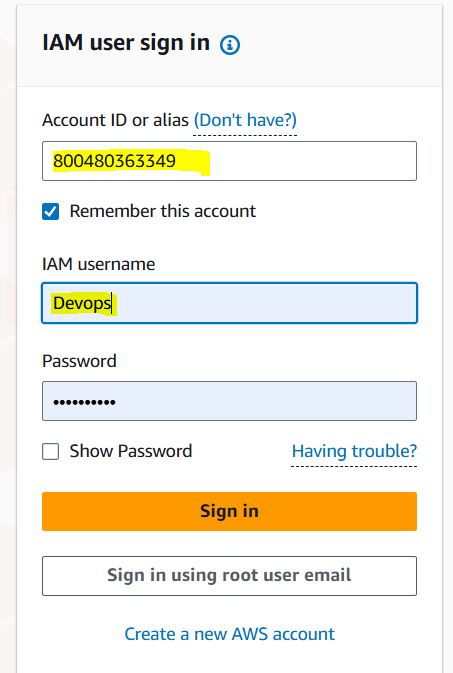
3. Click on the Groups tab.

4. Verify that Group-Test is listed.

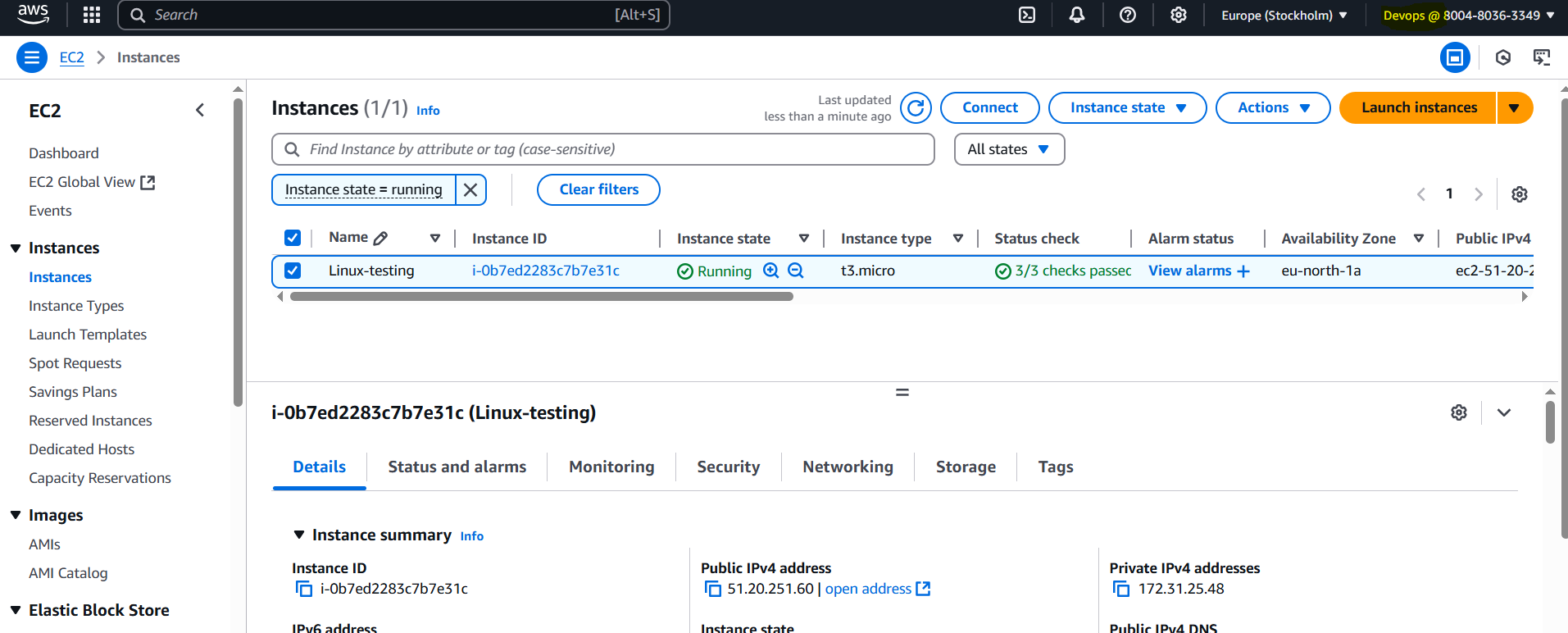


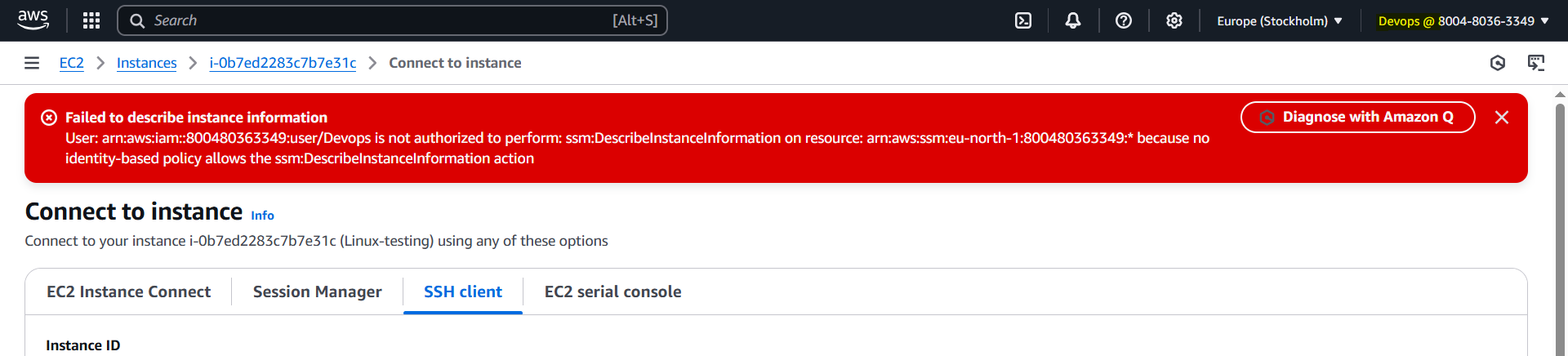
The Devops user now has the same permissions as the Group-Test group, which includes read-only access to EC2.

Login in to Devops user.



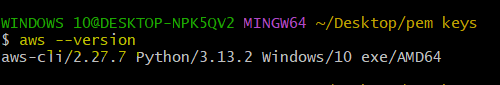
We have Read access but no other access.





**4) Write a bash script to create an IAM user with VPC full access.   
  
Step 1: Create a Bash Script**

Make sure whether aws cli is configured or not by using **aws --version**.



Create a file named create **iam-user.sh** with the following content:

**#!/bin/bash**

**USERNAME="mubeen-vpc"**

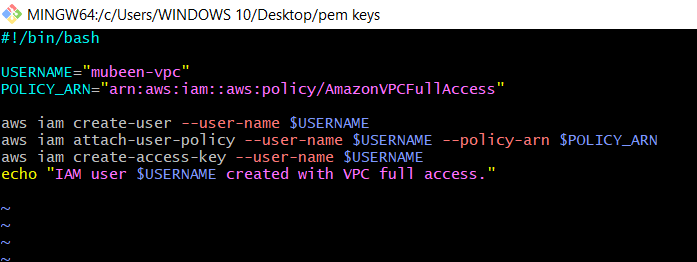
**POLICY\_ARN="arn:aws:iam::aws:policy/AmazonVPCFullAccess"**

**aws iam create-user --user-name $USERNAME**

**aws iam attach-user-policy --user-name $USERNAME --policy-arn $POLICY\_ARN**

**aws iam create-access-key --user-name $USERNAME**

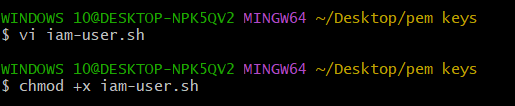
**echo "IAM user $USERNAME created with VPC full access."**



**Step 2: Make the Script Executable**

Run the following command to make the script executable:

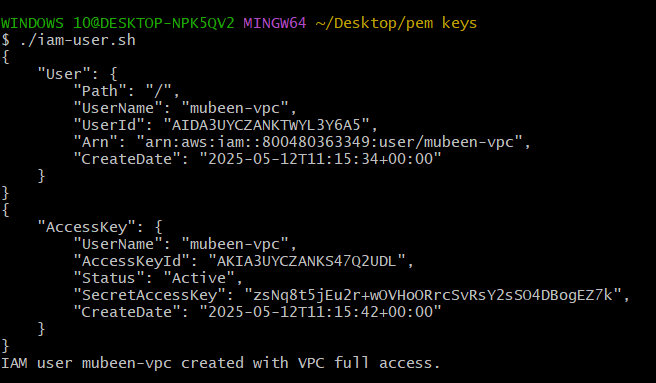
**chmod +x iam-user.sh**



**Step 3: Run the Script**

Run the script using the following command:

**./iam-user.sh**



**Step 4: Verify IAM User Creation**

1. Log in to the AWS Management Console.

2. Navigate to the IAM dashboard.

3. Click on Users in the left sidebar.

4. Verify that the user "mubeen-vpc" is listed.

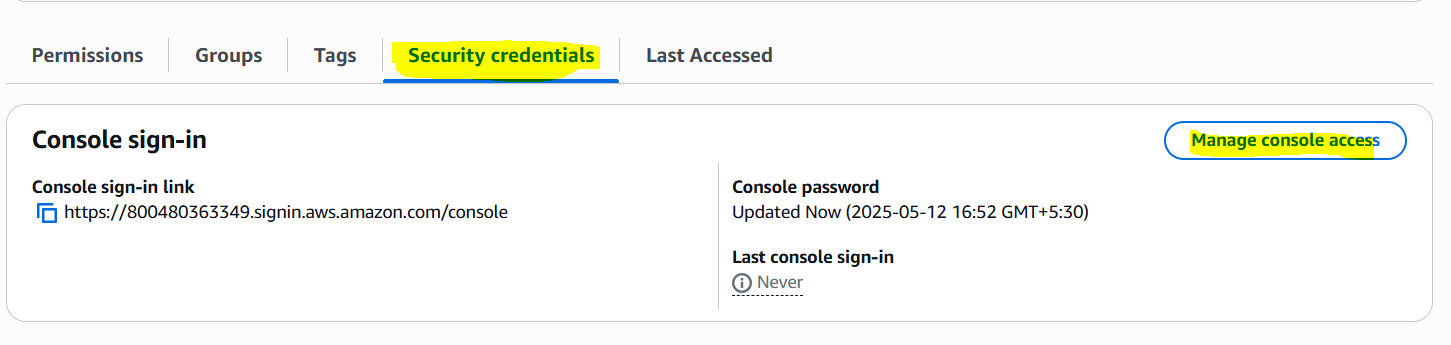


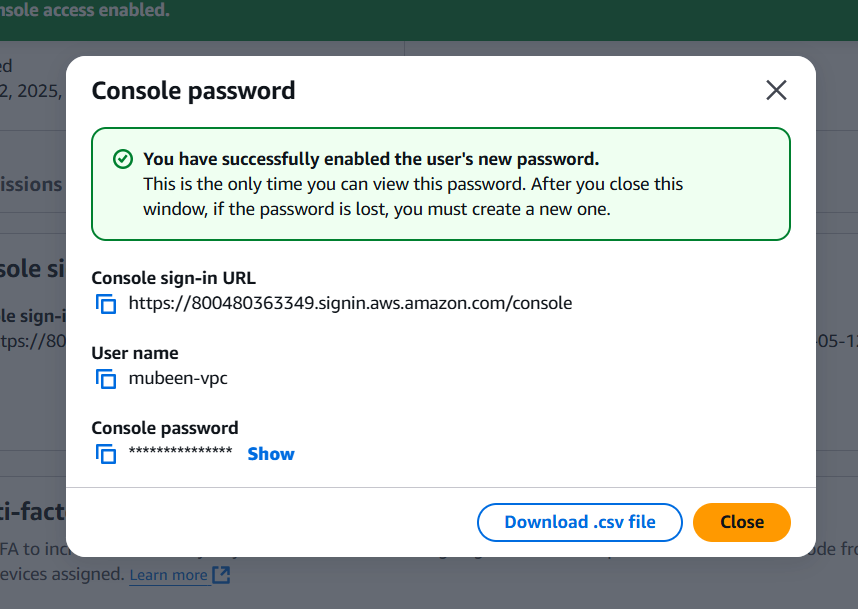
**Step 5: Login as the New IAM User**

1. Click on the user "mubeen-vpc" in the IAM dashboard.

2. Click on the Security credentials tab.

3. Add console password to the Manage console access.



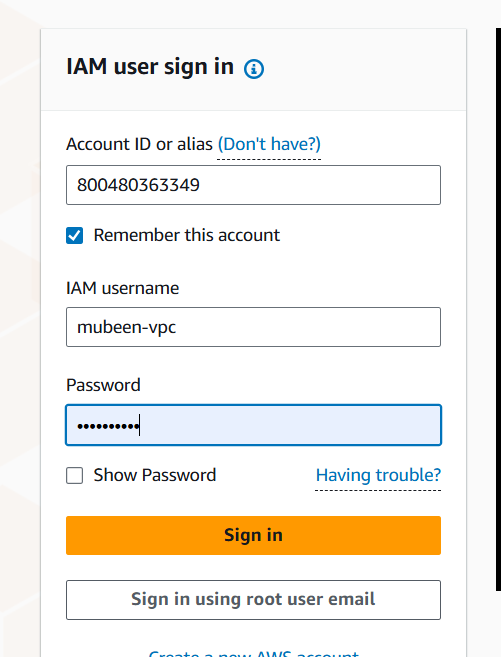


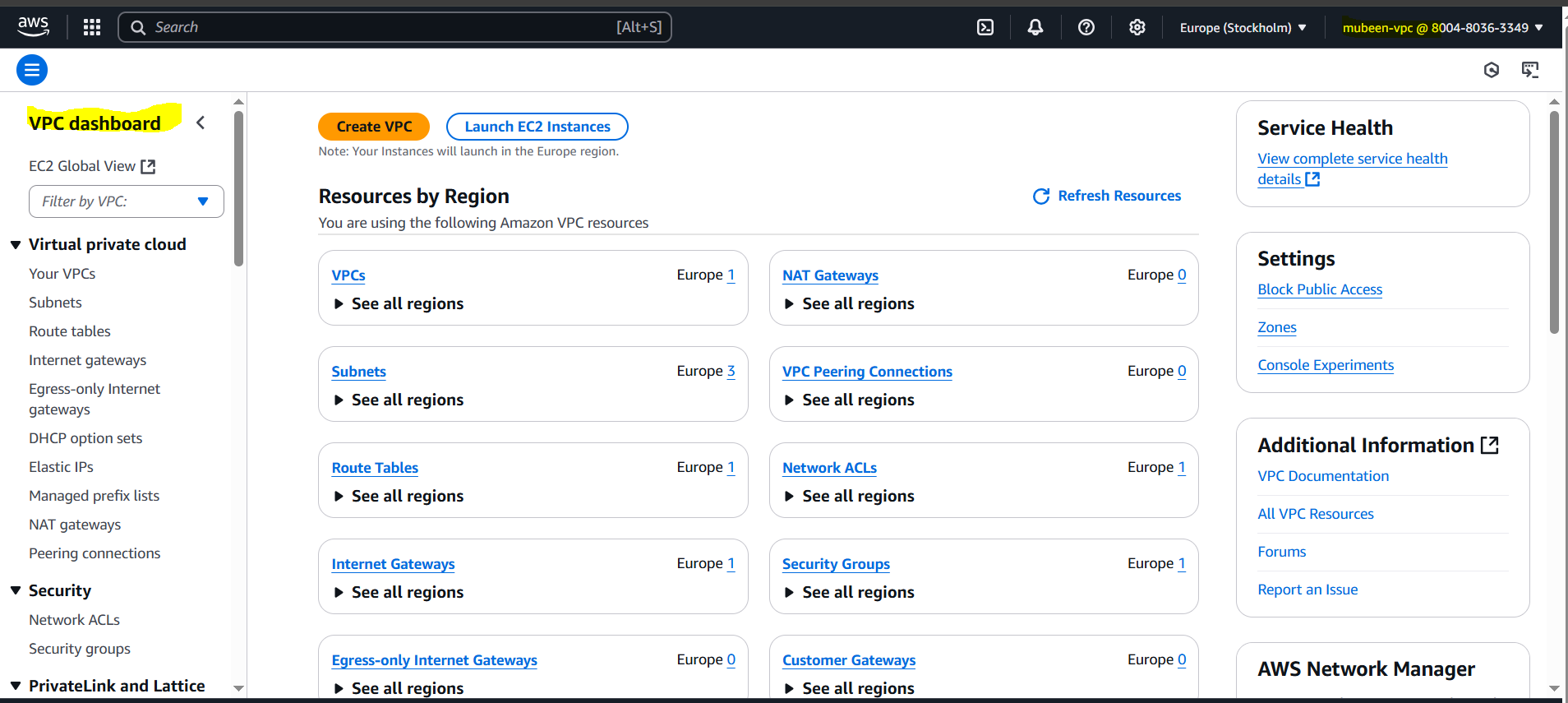
**Step 6: Verify VPC Access**

1. Log in to the AWS Management Console as the new IAM user.

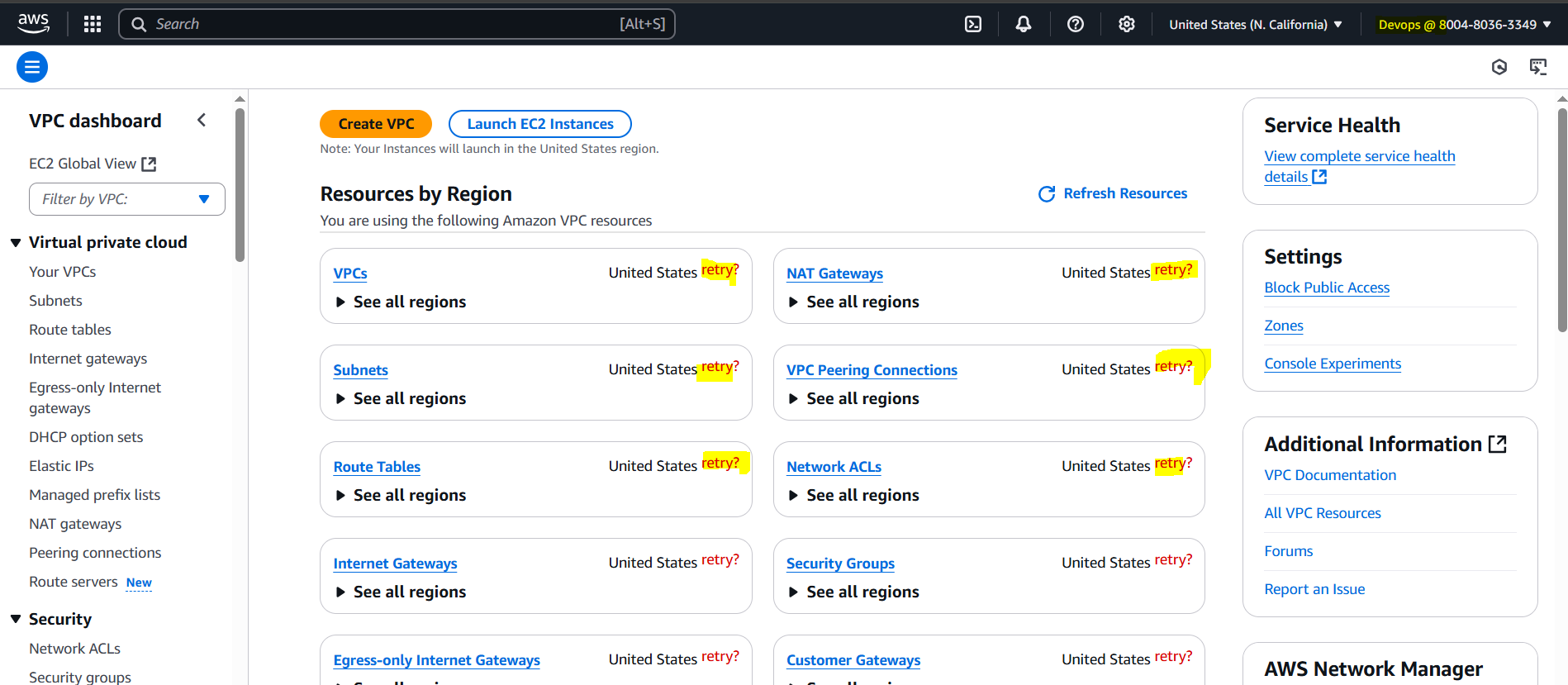
2. Navigate to the VPC dashboard.

3. Verify that you can view and manage VPC resources.





We can see that other users don’t have access for VPC.



**5) Create an IAM policy to access ec2 for a specific user in specific regions only.   
  
Step 1: Create a New Policy**

1. Log in to the AWS Management Console.

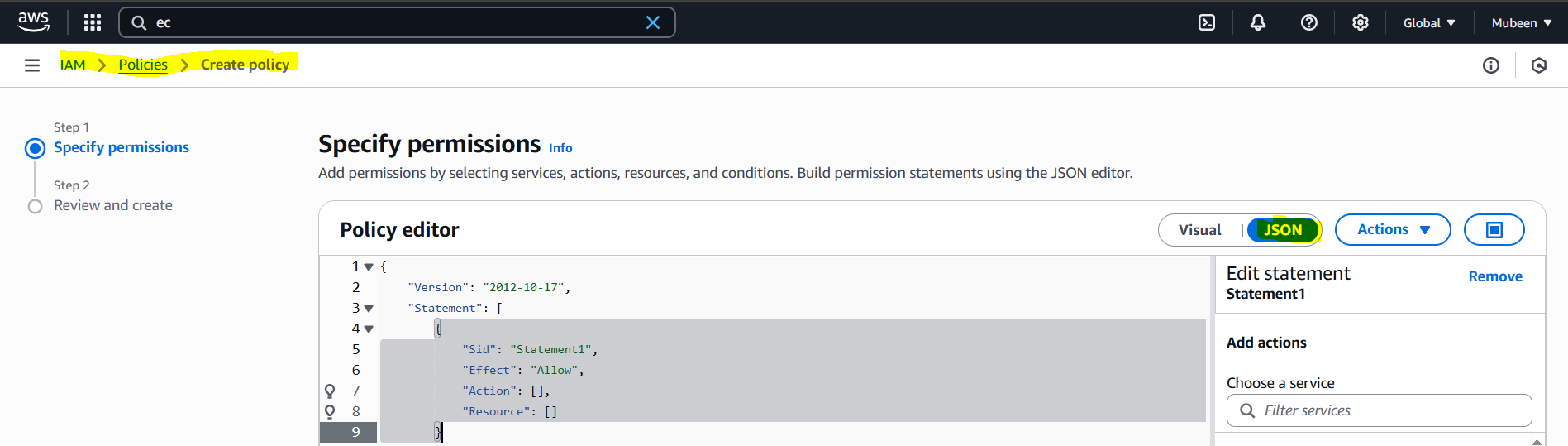
2. Navigate to the IAM dashboard.

3. Click on Policies in the left sidebar.

4. Click on Create policy.

5. Choose Custom policy.

6. Click JSON.



**Step 2: Define Policy Document**

Enter the following policy document:

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": "ec2:\*",

"Resource": "\*",

"Condition": {

"StringEquals": {

"aws:RequestedRegion": [

"us-east-2",

"us-west-2"

]

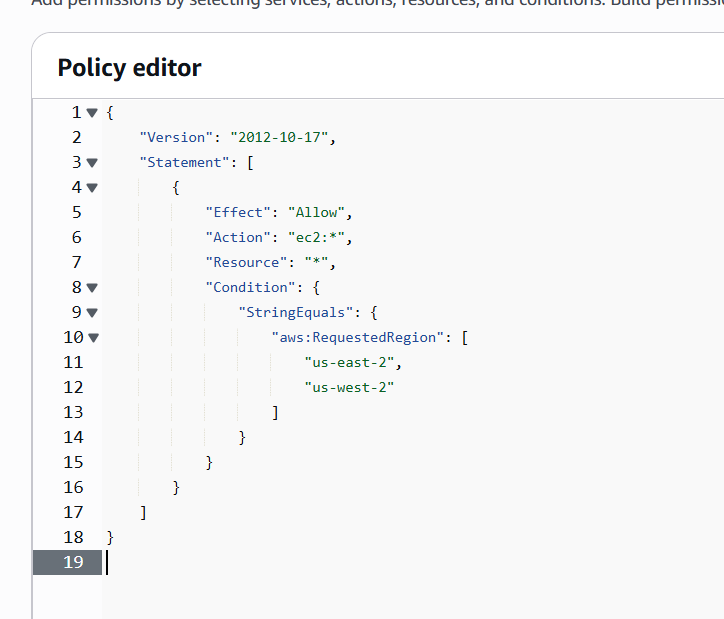
}

}

}

]

}



{

"Effect": "Allow", // Allow the action

"Action": "ec2:\*", // Allow all EC2 actions (start, stop, launch, etc.)

"Resource": "\*", // Applies to all EC2 resources

"Condition": { // Apply this only when...

"StringEquals": {

"aws:RequestedRegion": [ // ...the region is either of these:

"us-east-2", // Ohio

"us-west-2" // Oregon

]

}

}

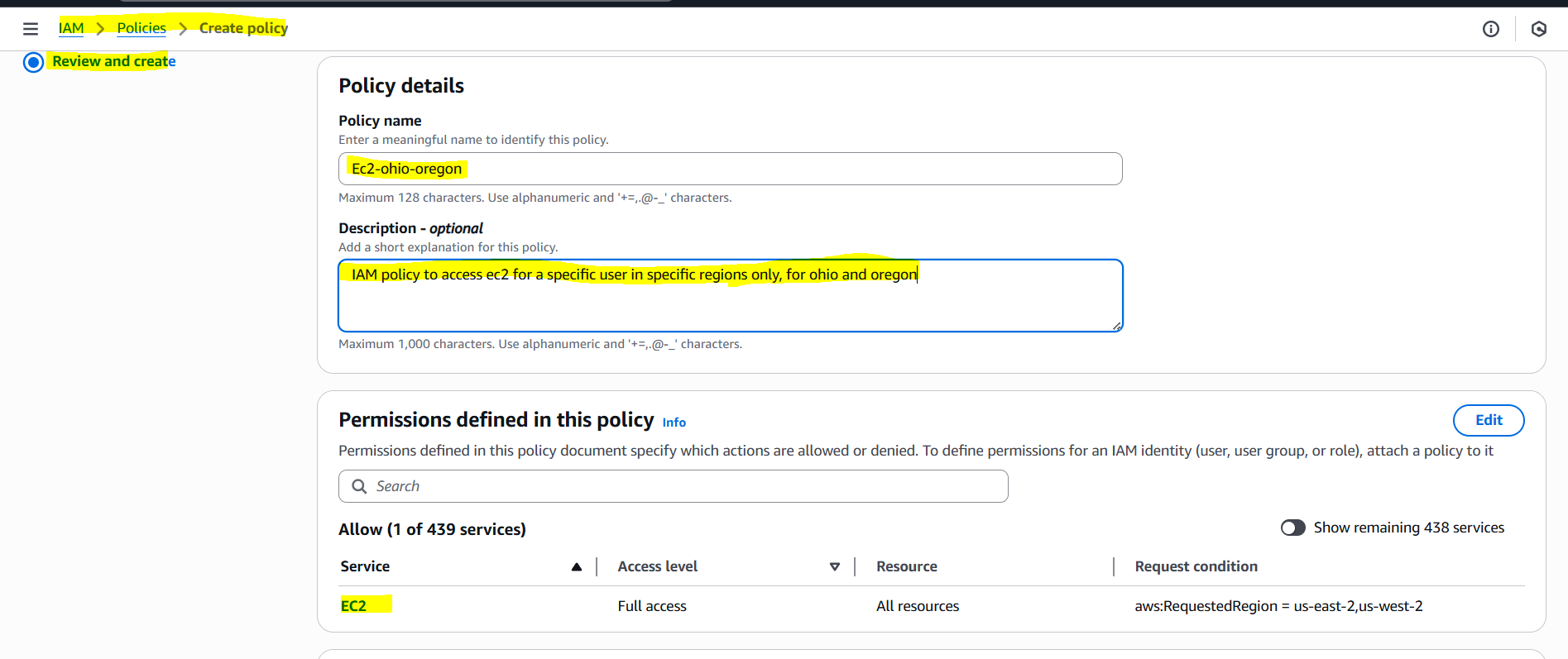
}

**Step 3: Review and Create**

1. Click Review policy.

2. Enter a Policy name.

3. Click Create policy.



**Step 4: Attach Policy to User**

1. Go to the Users page.

2. Find the Devops user.

3. Click on the Permissions tab.

4. Click Attach policy.

5. Search for the newly created policy.

6. Select the policy.

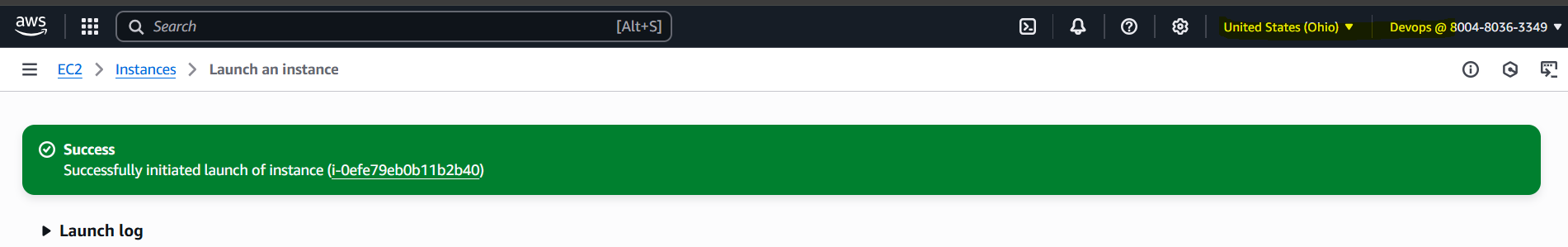
7. Click Attach policy.

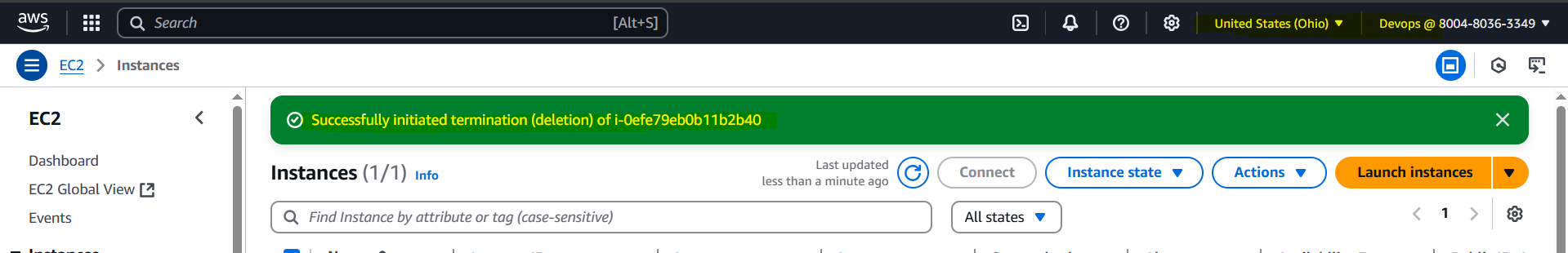


New policy added to user Devops and removed all previous policies.

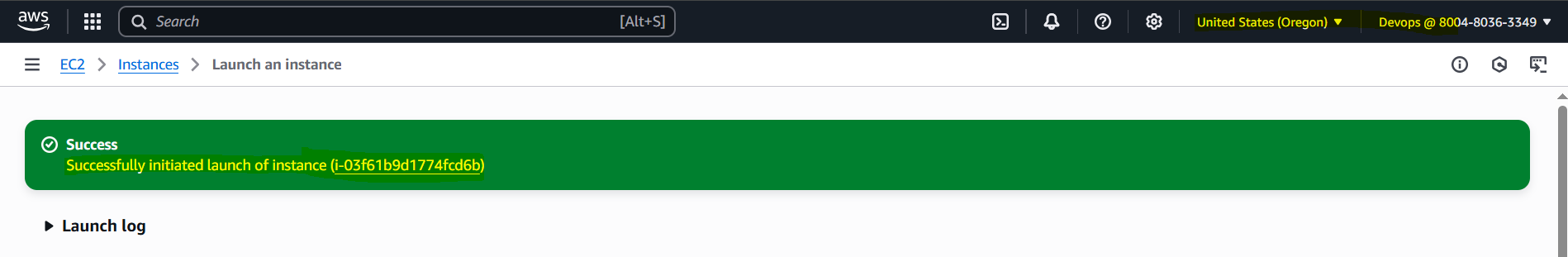
The Devops user now has EC2 access restricted to the Ohio (us-east-2) and Oregon (us-west-2) regions.

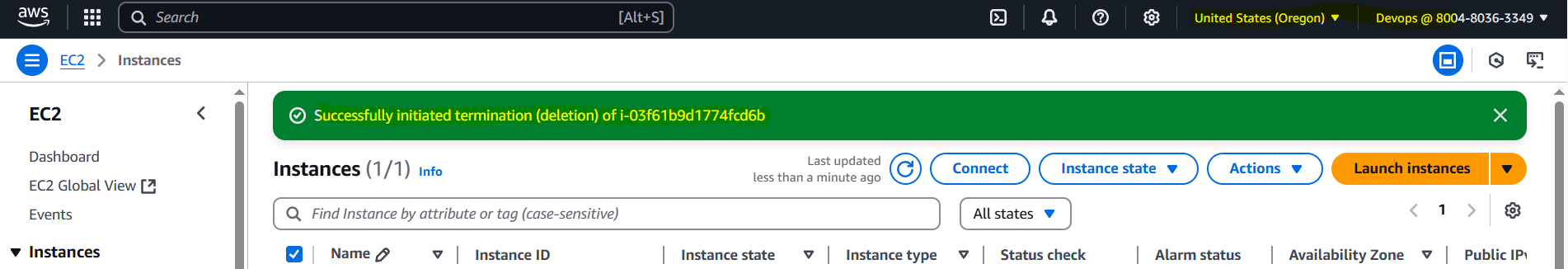
**Ohio**



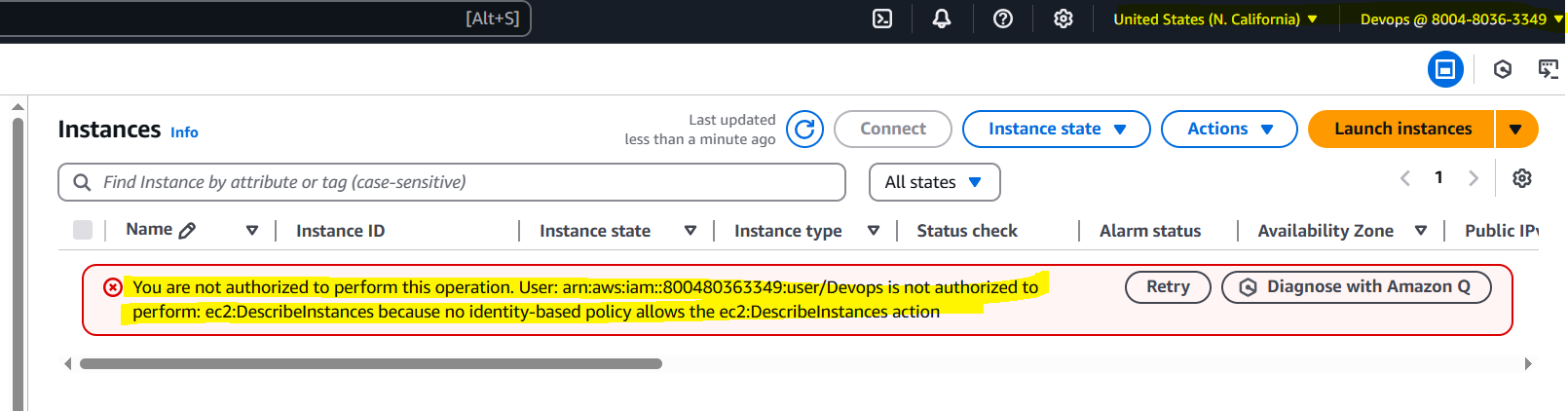


**Oregon**





**Another regions**

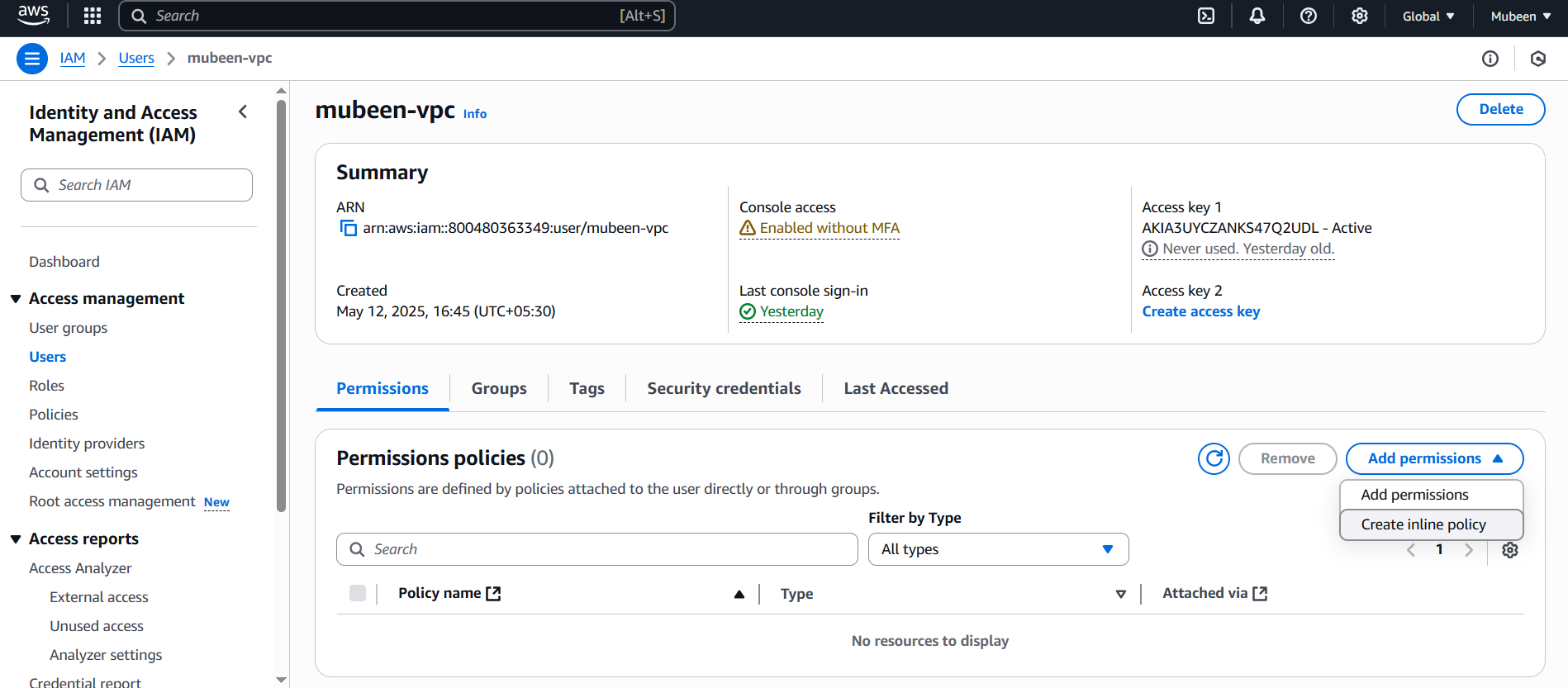


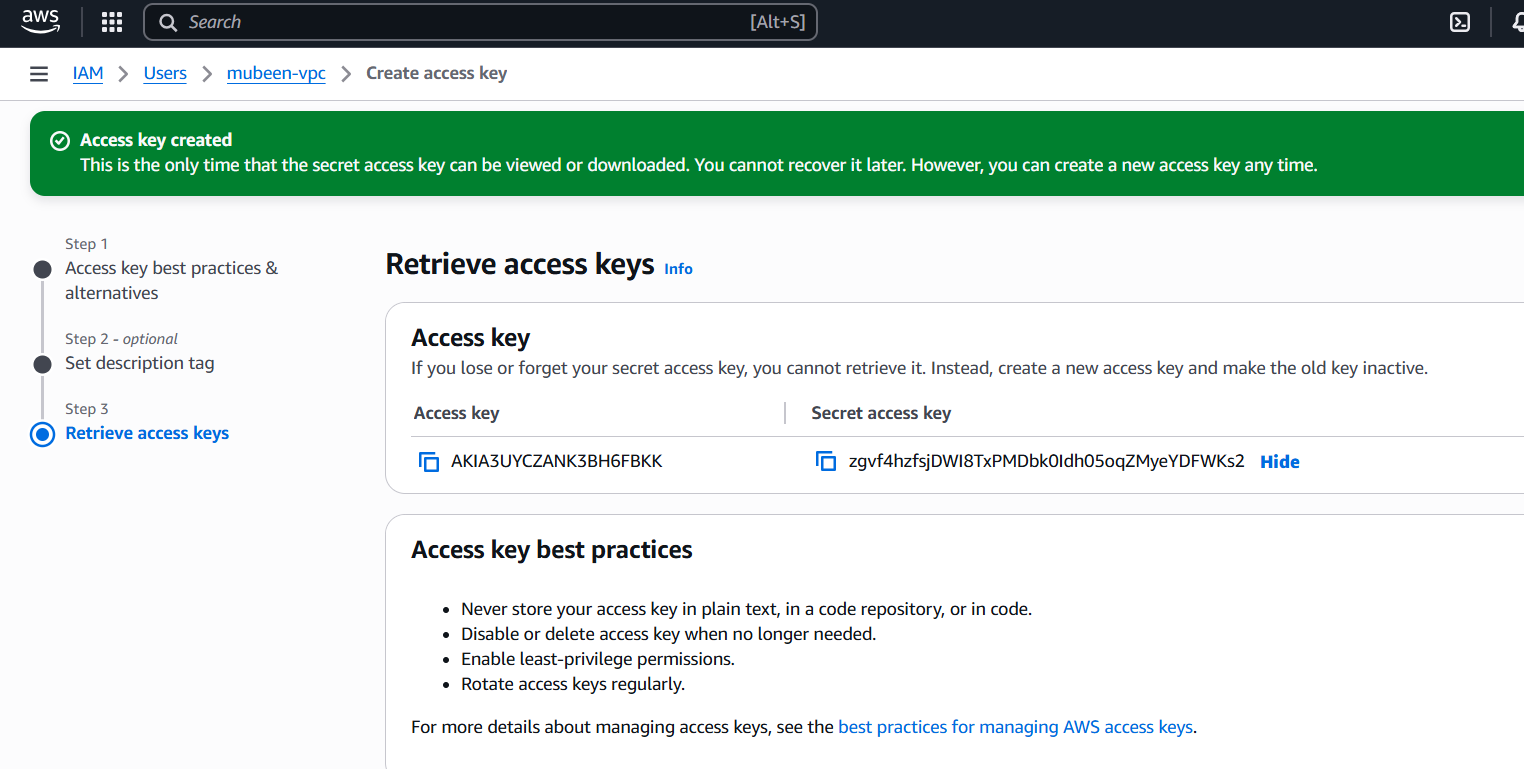
**6) We have two accounts Account A and Account B, Account A user should access s3 bucket in Account B. (Collaborate with team member and execute this. Mostly asked in every interview)**

Account A is Mubeen – User with inline Policy.

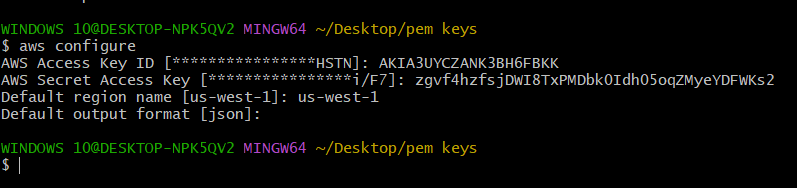
Account B is SGM (Ghouse) – S3 bucket with Bucket Policy.

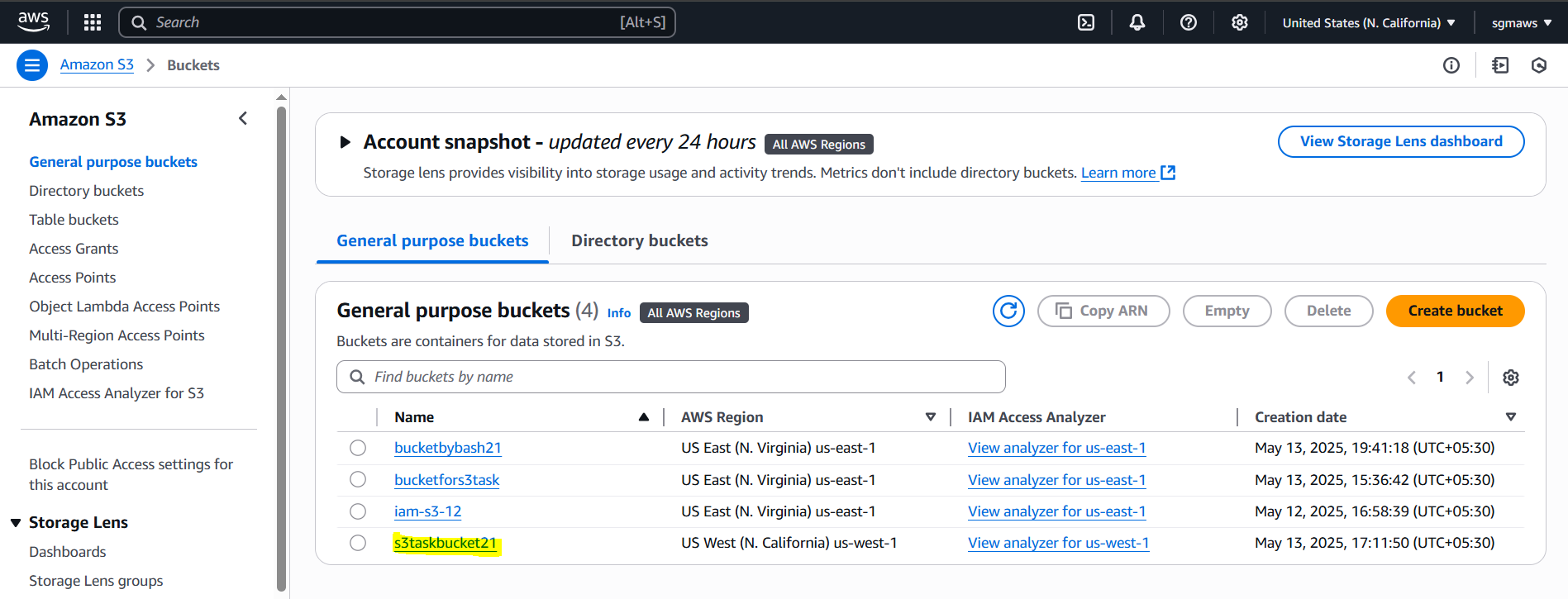
Account A:



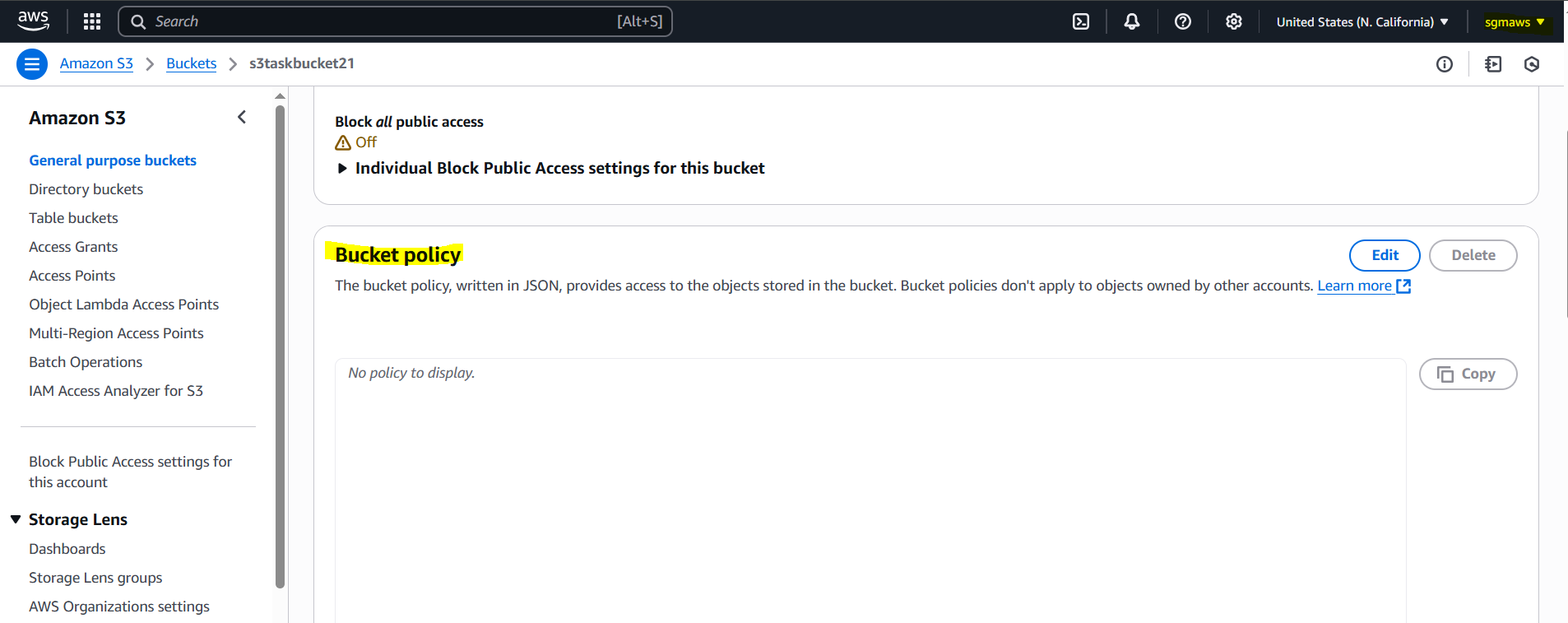
We have a existing user with name mubeen-vpc which we are using to proceed with the task.  


Create access key and configure in cli.

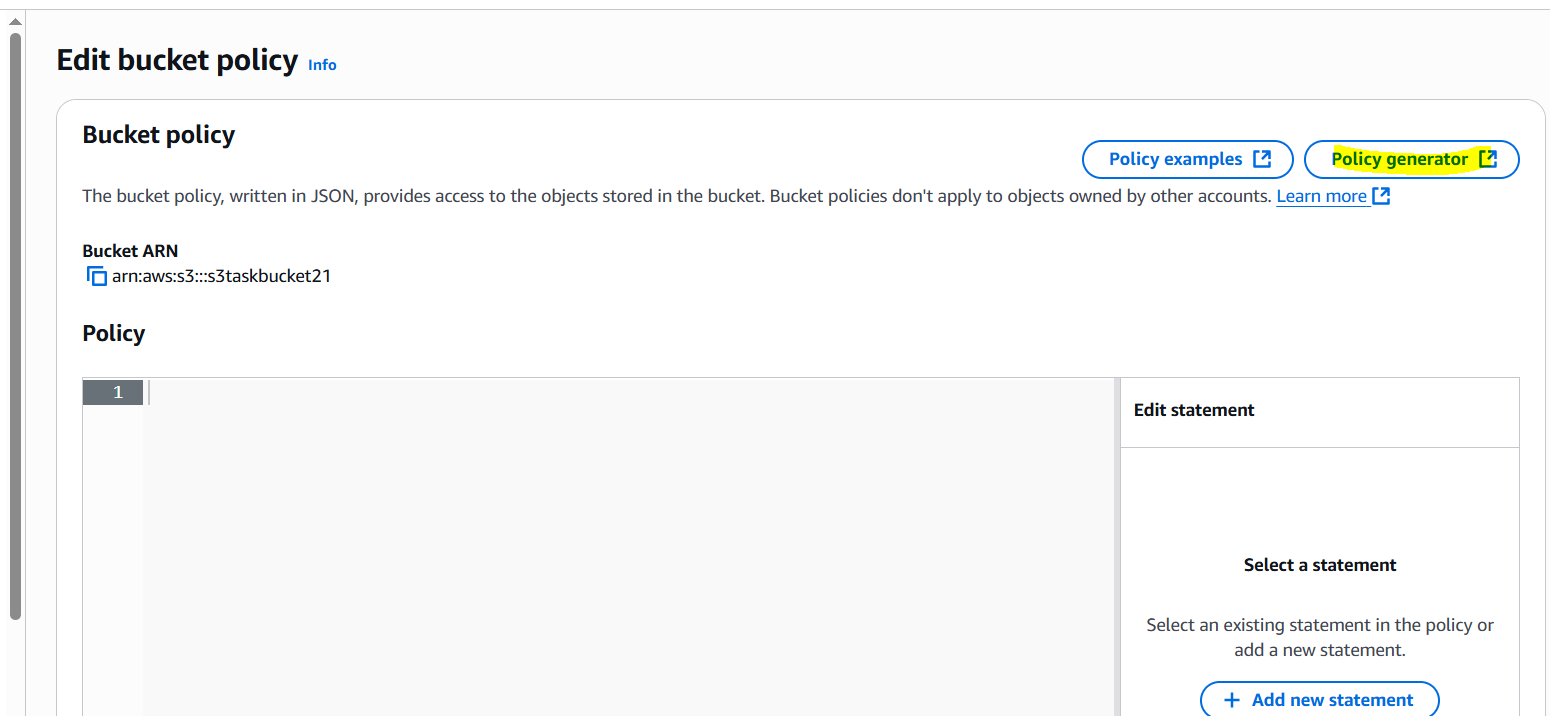
Account B:



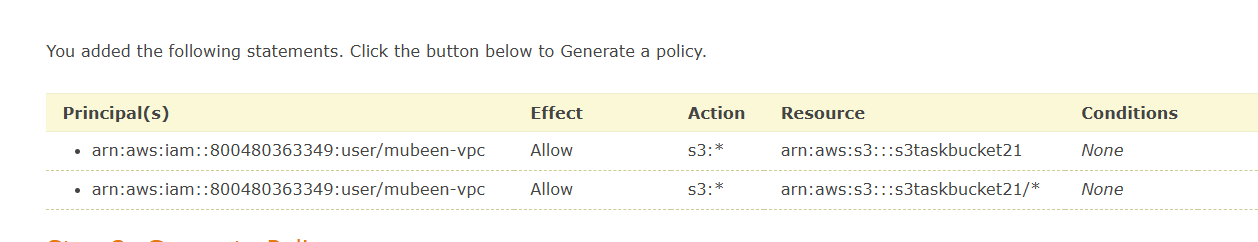
Using existing bucket **s3taskbucket21**



Adding bucket policy.



Click on policy generator to create a policy.



{

"Id": "Policy1747147499177",

"Version": "2012-10-17",

"Statement": [

{

"Sid": "Stmt1747147410495",

"Action": "s3:\*",

"Effect": "Allow",

"Resource": "arn:aws:s3:::s3taskbucket21",

"Principal": {

"AWS": [

"arn:aws:iam::800480363349:user/mubeen-vpc"

]

}

},

{

"Sid": "Stmt1747147491586",

"Action": "s3:\*",

"Effect": "Allow",

"Resource": "arn:aws:s3:::s3taskbucket21/\*",

"Principal": {

"AWS": [

"arn:aws:iam::800480363349:user/mubeen-vpc"

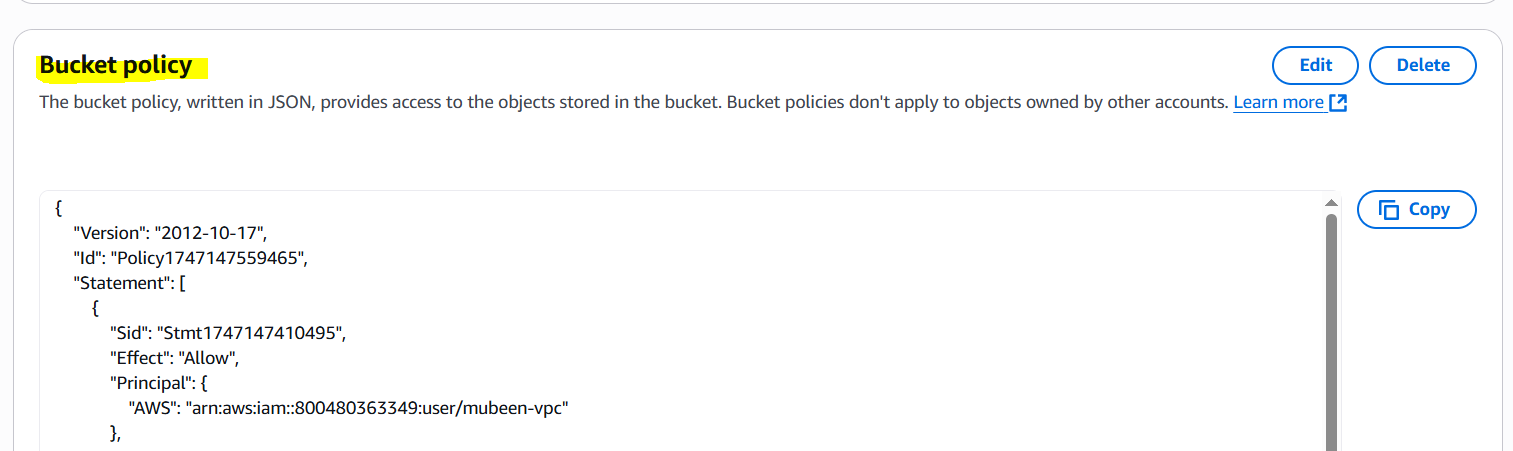
]

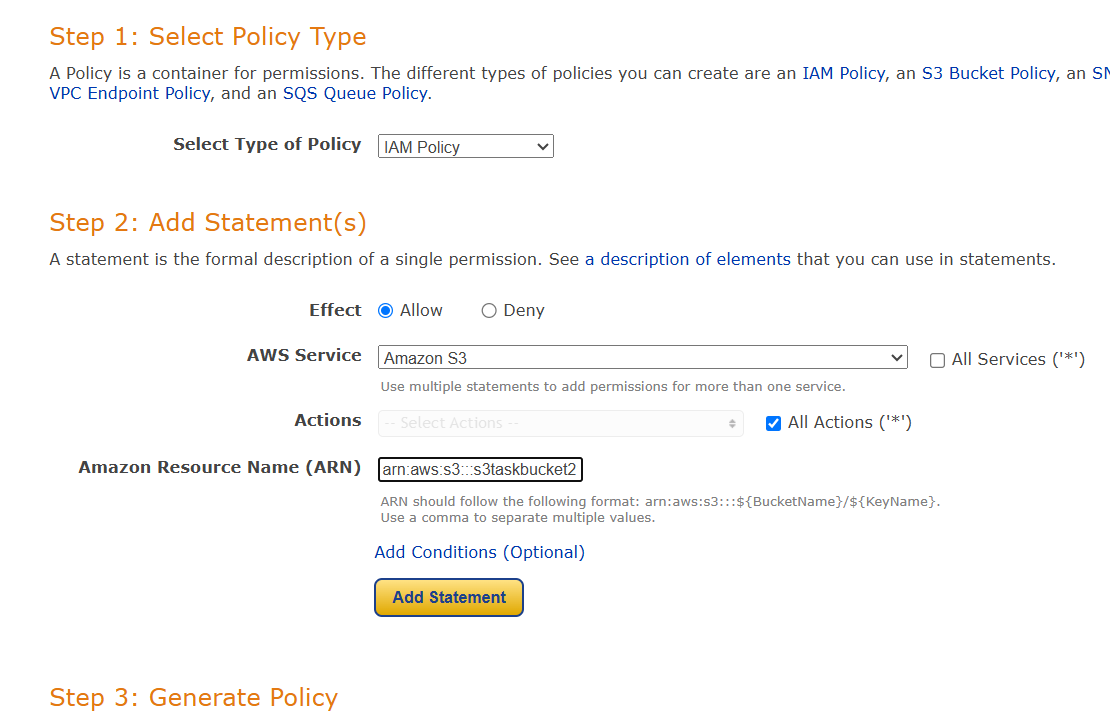
}

}

]

}

Add the generated policy to the bucket.  
  


Now generate inline policy for **user mubeen-vpc in Account A.**  


{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "Stmt1747147773860",

"Action": "s3:\*",

"Effect": "Allow",

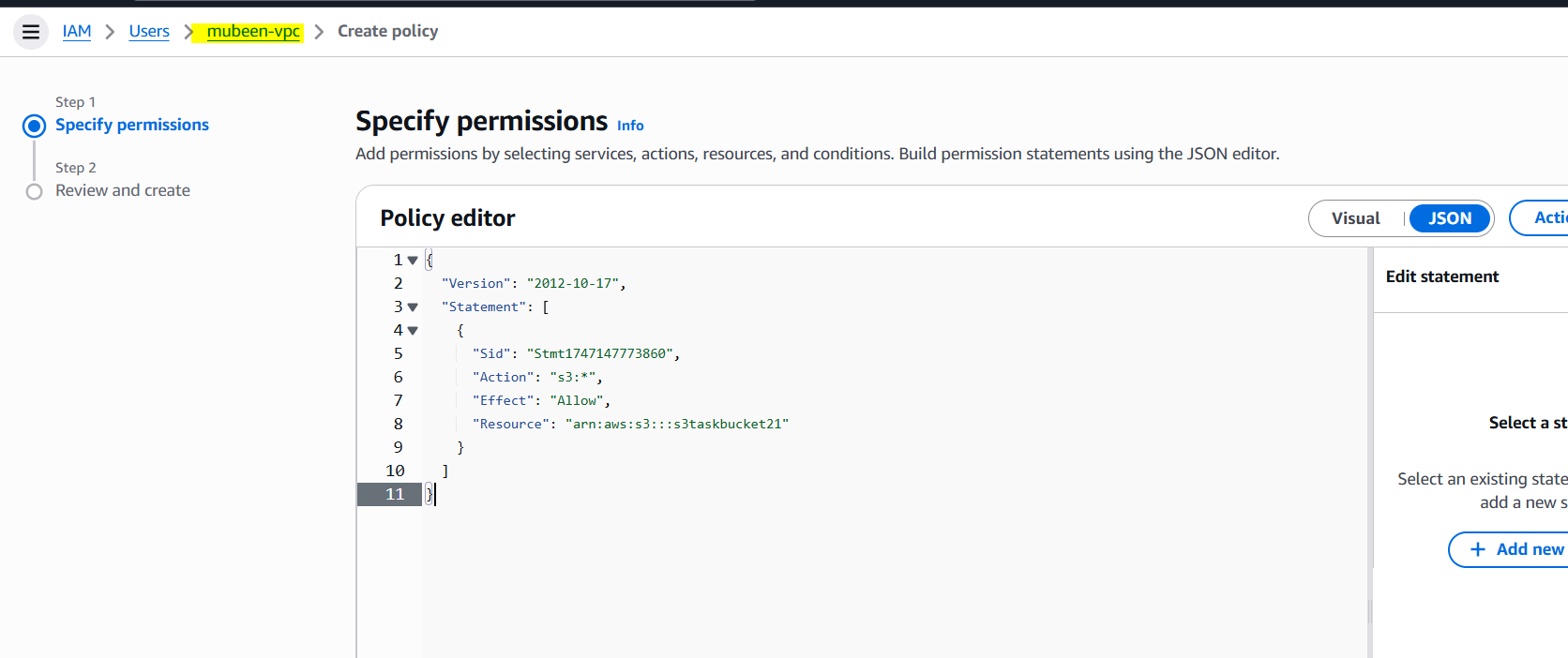
"Resource": "arn:aws:s3:::s3taskbucket21"

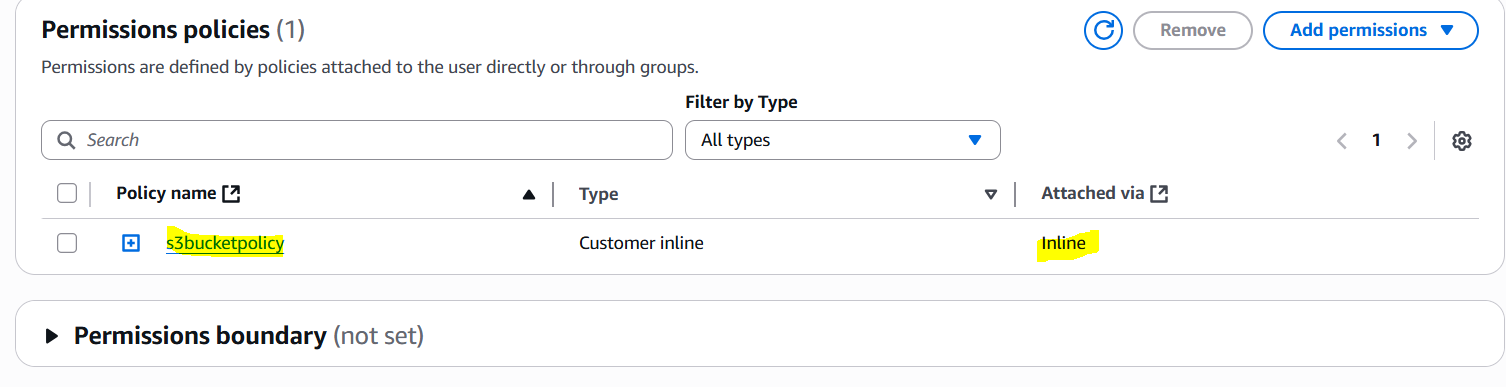
}

]

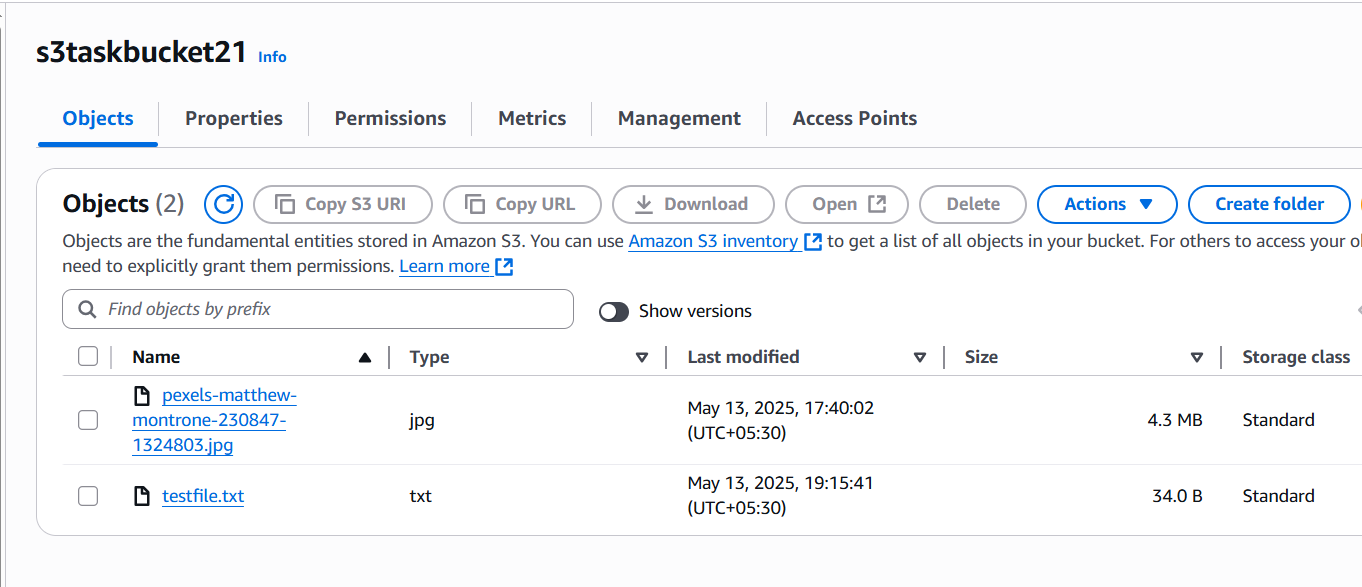
}

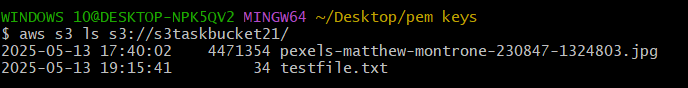
Add inline policy to the user.





Finally we can access Account b bucket from account a cli using command **aws s3 ls s3://s3taskbucket21/**





**We can see that files in the console are same in CLI.**

**The -End**