Placement Empowerment Program

Cloud Computing and DevOps Centre

Implement Role-Based Access Control in the Cloud: Create different IAM roles for accessing cloud resources (e.g., read-only, admin). Test their permissions.

Name: SUBASHINI P

Department: IT

Introduction

In modern cloud environments, **security and access control** are crucial for managing resources effectively. **Role-Based Access Control (RBAC)** in AWS Identity and Access Management (IAM) ensures that users, applications, and services **only have the permissions they need**, reducing security risks.

This PoC demonstrates how to **create, assign, and test IAM roles** with different permissions for AWS resources. We will implement **least privilege access** by assigning:

- **Read-only access to S3** for a user.
- **Full access to EC2** for another user.

Overview

This PoC focuses on **configuring IAM roles with specific permissions** and validating their effectiveness. The key steps include: 1. **Creating IAM Roles**

S3ReadOnlyRole (Grants read-only access to S3). EC2FullAccessRole (Grants full control over EC2).

2. **Assigning IAM Roles to Users**

Attach S3ReadOnlyRole to User A. Attach EC2FullAccessRole to User B.

3. **Testing Permissions**

   Validate that User A can only list S3 buckets but cannot create/delete them.Verify that User B can launch and manage EC2 instances but cannot access S3.

Objectives

1. Implement **IAM roles with least privilege access**.
2. Demonstrate **secure access control** using AWS IAM.
3. Ensure **users can only perform authorized actions**.
4. Improve **security posture** by restricting unnecessary permissions.

Importance

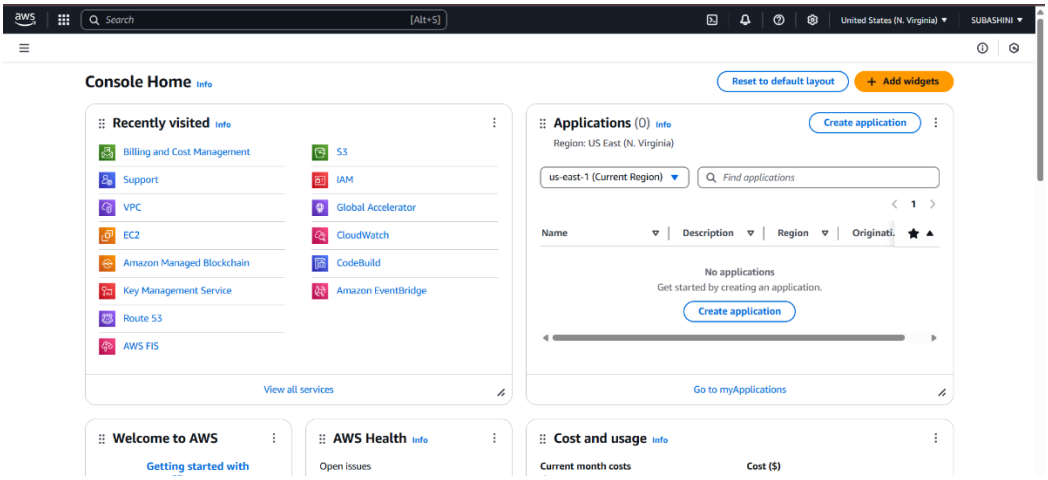
1. **Enhances Cloud Security** – Prevents unauthorized access and enforces least privilege.
2. **Simplifies Permission Management** – IAM roles reduce manual policy management.
3. **Ensures Compliance** – Helps meet security and governance requirements.
4. **Prevents Costly Mistakes** – Avoids accidental resource modifications/deletions.

5. **Encourages Best Practices** – Follows AWS security guidelines for IAM.

Step-by-Step Overview

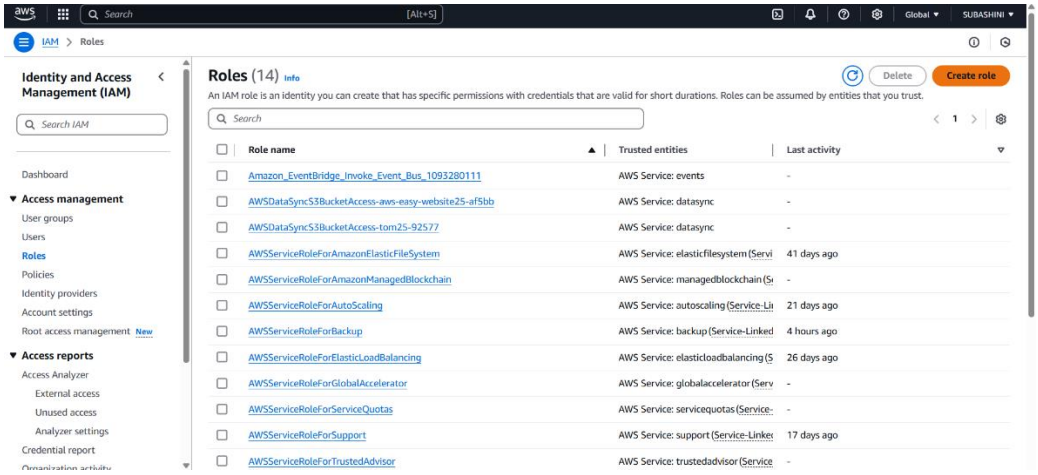
Step 1:

1. Go to AWS Management Console.
2. Enter your username and password to log in.



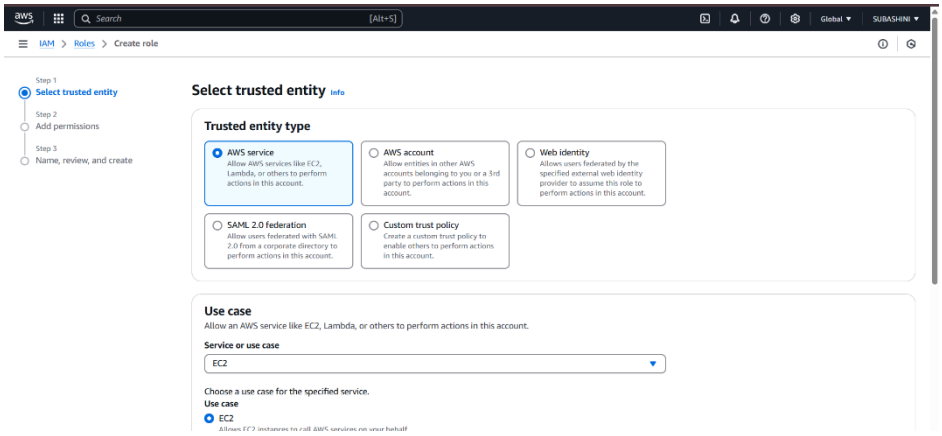
Step 2:

1. **Sign in to AWS Management Console.**
2. Go to **IAM → Roles → Create Role.**



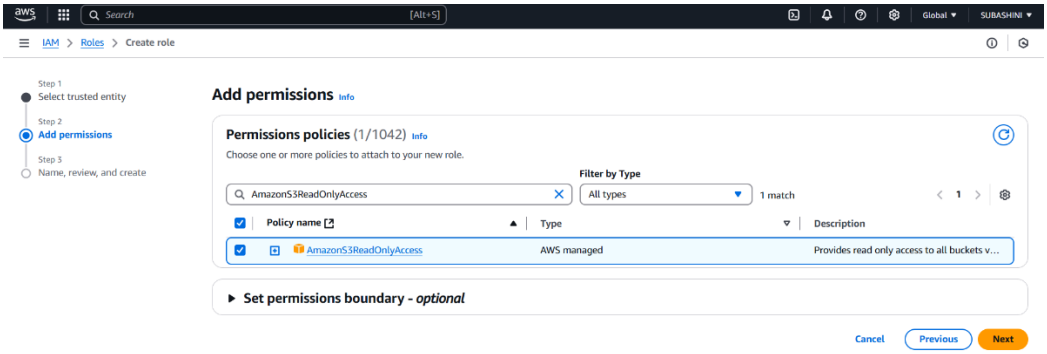
Step 3:

1. **Select trusted entity:** Choose **AWS Service**.
2. **Use case:** Select **EC2** role for an instance.
3. Click **Next**.



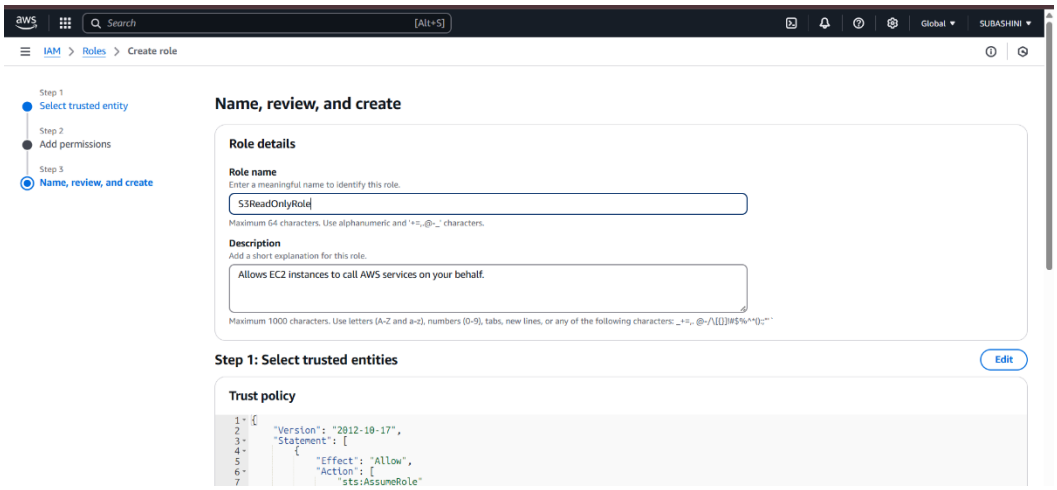
Step 4:

Search for **AmazonS3ReadOnlyAccess** and select it.



Step 5:

1. Click **Next** → Name the role **S3ReadOnlyRole**.
2. Click **Create Role**.



Step 6

1. Go to **IAM** → **Roles** → **Create Role**.
2. **Select trusted entity:** Choose **AWS Service**.

3. **Use case:** Select **EC2**.

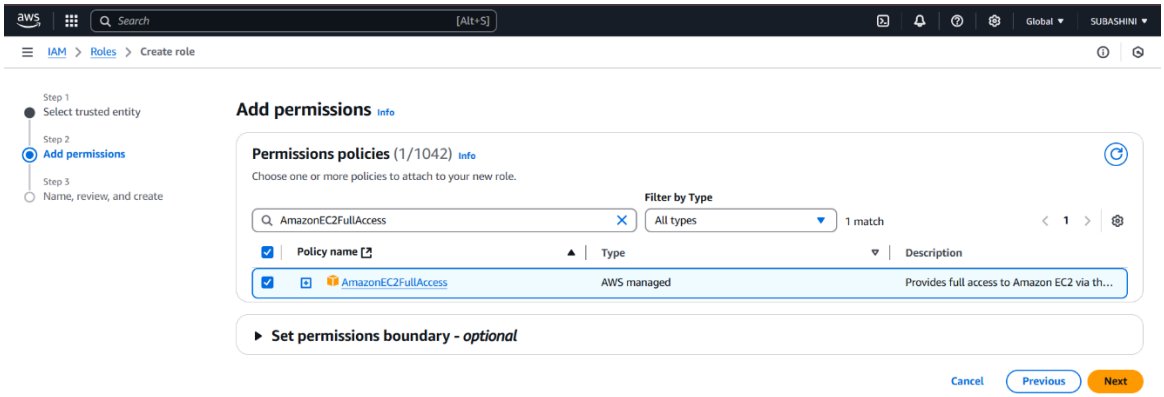
4. Click **Next**.

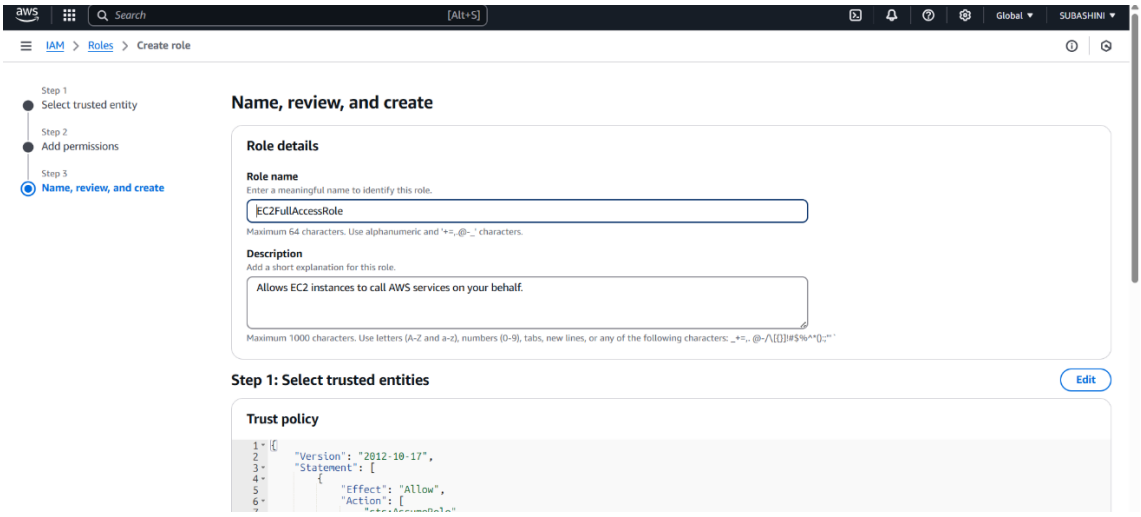
5. **Attach permissions**:

Search for **AmazonEC2FullAccess** and select it.

6. Click **Next** → Name the role **EC2FullAccessRole**.

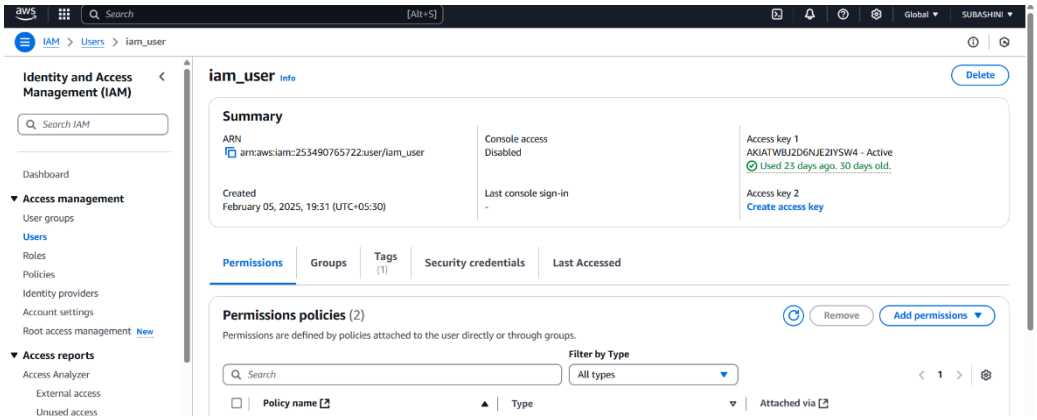
7. Click **Create Role**.





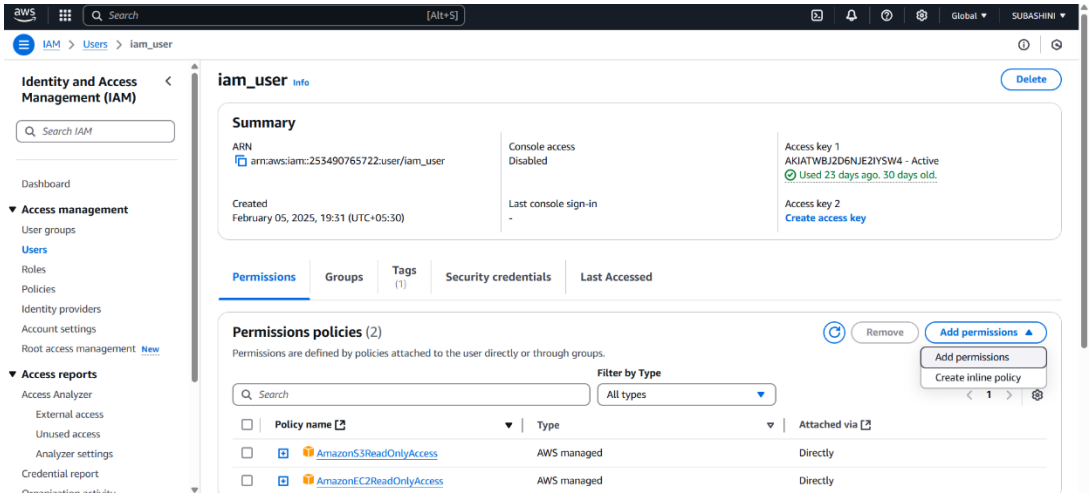
Step 7

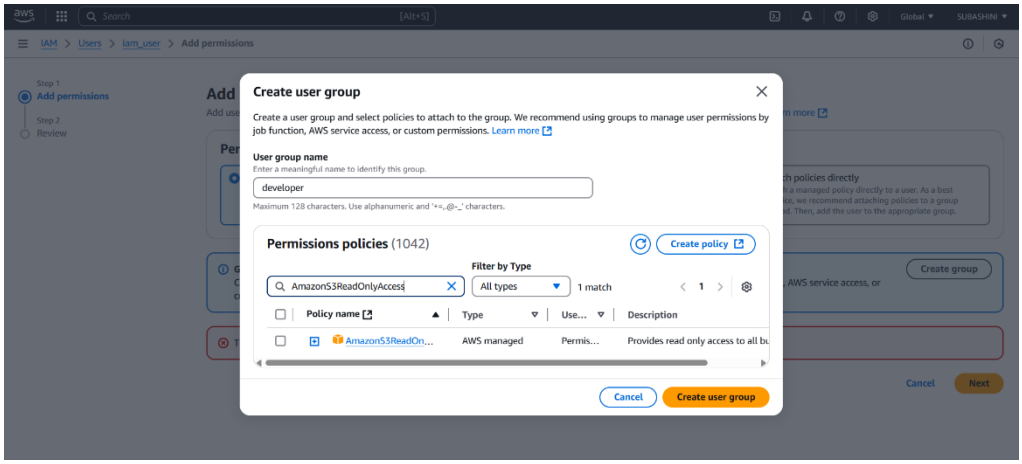
1. Go to **IAM → Users.**
2. Select a user.

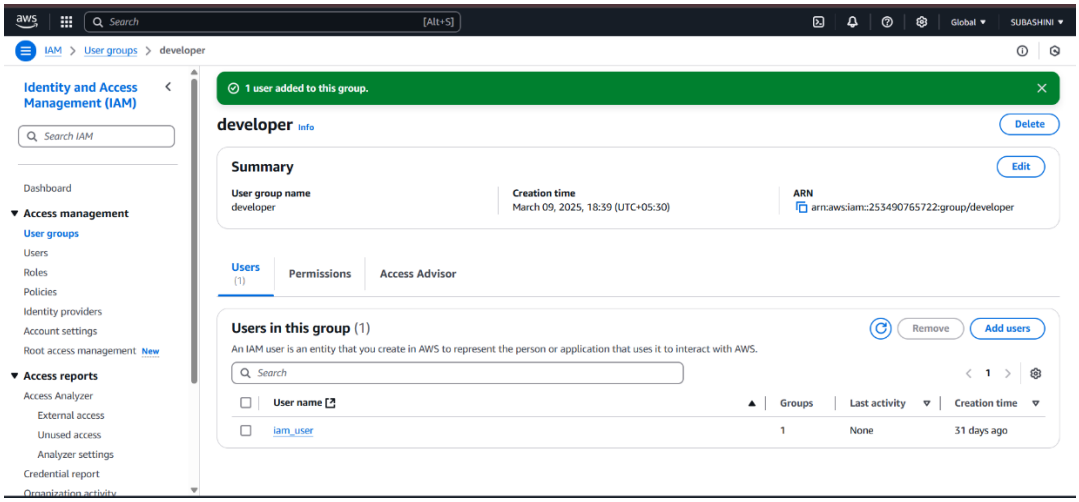


Step 8

1. Assign:
   - **S3ReadOnlyRole** to one user.
   - **EC2FullAccessRole** to another user.
2. Click **Next → Review → Add permissions.**







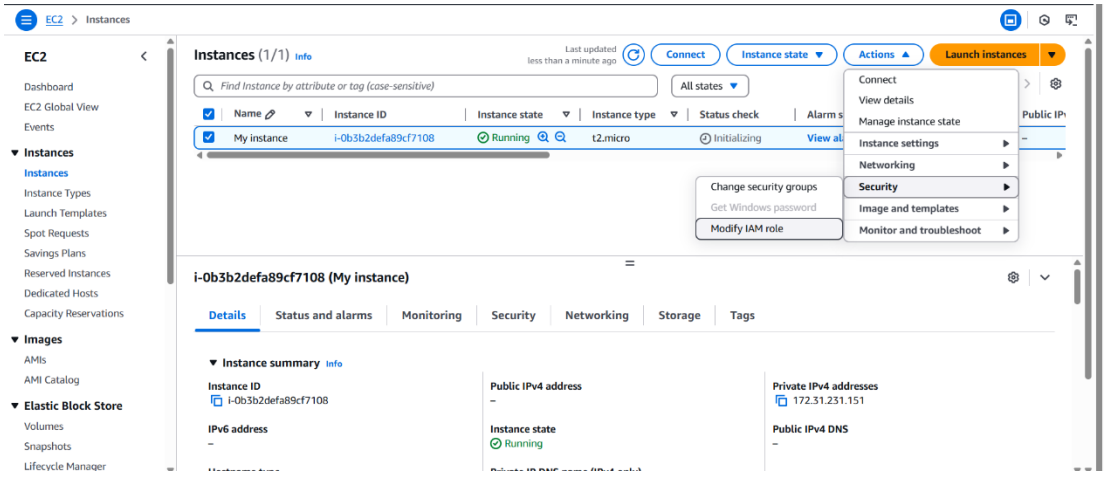
Step 8

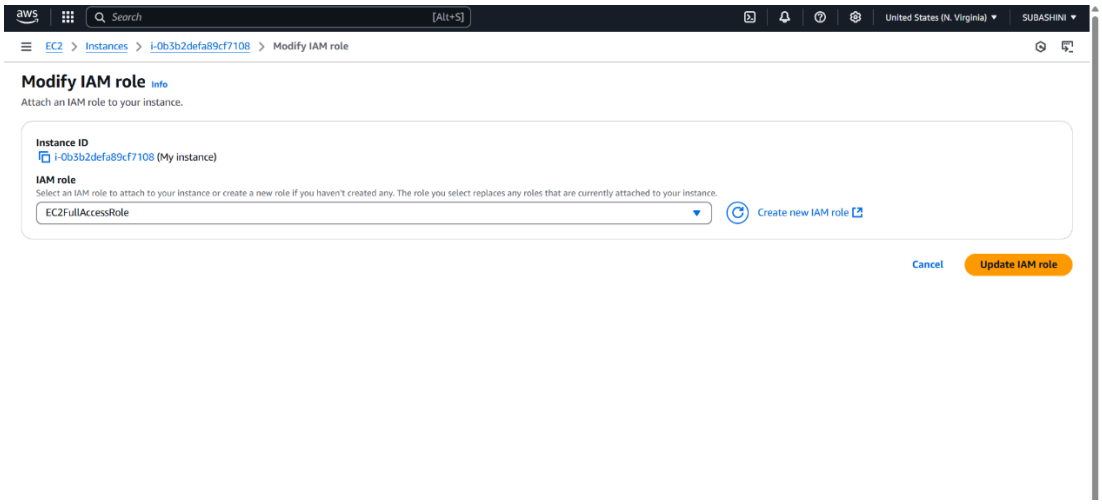
1. Go to **EC2 → Select an Instance.**

2. Click **Actions → Security → Modify IAM Role.**

3. Attach **EC2FullAccessRole** to the instance.

4. Click **Update IAM Role.**





Step 9

Open Command prompt

1. Run:

   aws s3 ls

   ☑ It should list S3 buckets.

2. Try creating a bucket: aws s3 mb s3://test-bucket ✗ It should **deny access**.

```
C:\Users\subam>aws s3 ls
2025-02-07 12:49:11 strbucket-1
2025-02-07 12:49:54 strbucket-2

C:\Users\subam>aws s3 mb s3://strbucket-1
make_bucket failed: s3://strbucket-1 An error occurred (AccessDenied) when calling the CreateBucket operation: User: arn
:aws:iam::253490765722:user/iam_user is not authorized to perform: s3:CreateBucket on resource: "arn:aws:s3:::strbucket-
1" because no identity-based policy allows the s3:CreateBucket action
```

Step 10

1. Sign in as the user with **EC2FullAccessRole**.
2. Try launching an EC2 instance:

aws ec2 run-instances --image-id ami-12345678 --instance-type t2.micro –subnet-id subnet-1234567

3. It should succeed.
4. Try listing S3 buckets:

aws s3 ls

5. It should **deny access**.

```
C:\Users\subam>aws ec2 run-instances --image-id ami-08b5b3a93ed654d19 --instance-type t2.micro --subnet-id subnet-004337
654f3dd6d99

An error occurred (UnauthorizedOperation) when calling the RunInstances operation: You are not authorized to perform thi
s operation. User: arn:aws:iam::253490765722:user/iam_user is not authorized to perform: ec2:RunInstances on resource: a
rn:aws:ec2:us-east-1:253490765722:instance/* because no identity-based policy allows the ec2:RunInstances action. Encode
d authorization failure message: dDsQ3W_AtUdBtPyPwcj1GgqxL8TVmGuNlCjg3dpkcWZm8r58eyvnFJybzdvuylFMrRug9xUne5uvYW27ZVkQsN0
MxFKgAGre6xC-Q6rn6Tf1uLI8PQgTR_gMSL01xpZrinTIKViCZ8G9eaYlIg9FGwhxeAyyKyc35aaNmoTlPnoJ4MyHqi3v7Ca0Ng5X8NU8RXZr-z8FtL6LV6x
4Vy-hhttq495yOMIui12gATmbVxEY_n4iNWxeIntGdWTbVEDXbtDW27vjeu9J7Dmk75DYl516MpBdU4UupEvkWhgIrVbEymus9XkzewevoUqxprEwxOTNNvd
aP11-4GeLHWHiKosYm7-amC4eCXWjn9u9-k_JuuN4-jpzrETjHcnsPptYW4r6OBBcrCT_HOOjjYRwZTnIvD-uFRogyVhX15qr1X7PswGvL2jTS3xiBIfCmg9
Wq4VnHMrGXw05CJZHUPe6pBhGTKaWCEhgV7icQ0A3-rgmyanQfEoJ3LHmyn8szrjKHLwDH3UKjjLouupICTn7dzBbg8M6Hwv7Vjzis_nZTtSH33Q6wdj6kfd
6Vw0stXIotHAyVwAnbxD0BlRp4eeLG3zhFqGHHk1jmC3Zk4GuXtG36AiYyQaFoZoHngfZ3xp3fPhhKZnen7cwS6uQFi38XgXhDmPfce-rhZOs9O_0LtVWiGR
E3OGCEB0aWCDUUGy0ZX38AUbBXzf_kh_3Hlw5Xb3nIjNf-lHdZQ3QUMU4CpswIdVlsSMFlQPXfF7cJWOZx8tg8mE9
```

Outcomes

By completing this **Role-Based Access Control (RBAC) in AWS IAM** PoC, you will:

1. **Understand AWS IAM Roles & Policies** – Gain hands-on experience in creating and managing IAM roles with different levels of access control.
2. **Implement Least Privilege Access** – Learn how to restrict permissions effectively, ensuring users and services only have the minimum access required.
3. **Assign IAM Roles to Users** – Practice attaching predefined IAM policies (AmazonS3ReadOnlyAccess and AmazonEC2FullAccess) to different users securely.
4. **Test & Validate Permissions** – Verify that IAM users can perform only the allowed actions, ensuring security by testing access to S3 and EC2.
5. **Enhance Cloud Security Best Practices** – Improve AWS security posture by reducing the risk of unauthorized access and preventing accidental resource modifications.
6. **Use AWS CLI for IAM Management** – Execute AWS CLI commands to list, create, and verify permissions assigned through IAM roles efficiently.