

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: Vishali V

Department:
CSE

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) only have the ensures that users, applications, and services 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.

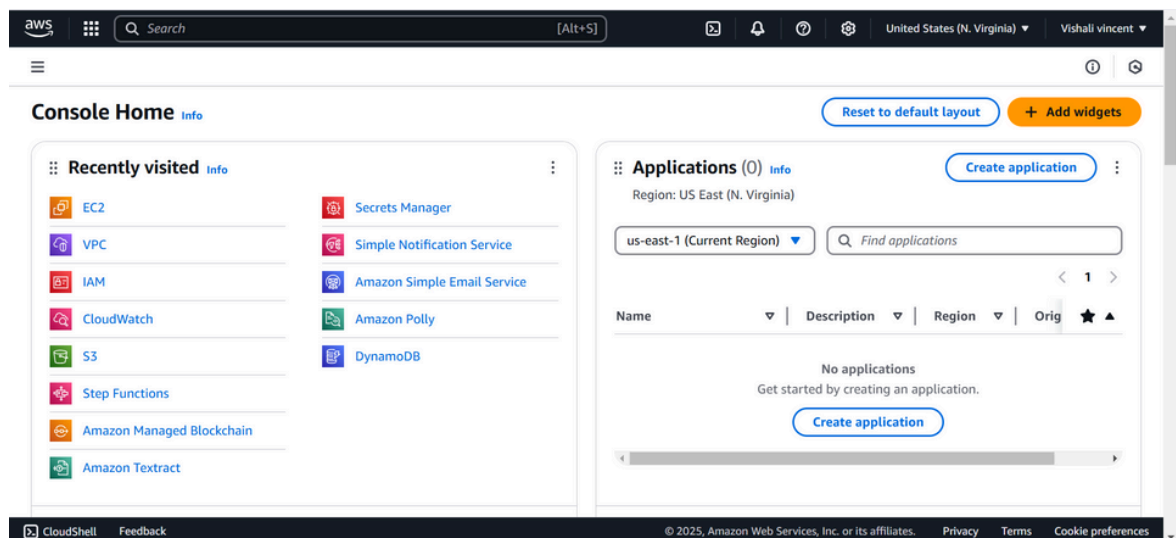
Importance

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

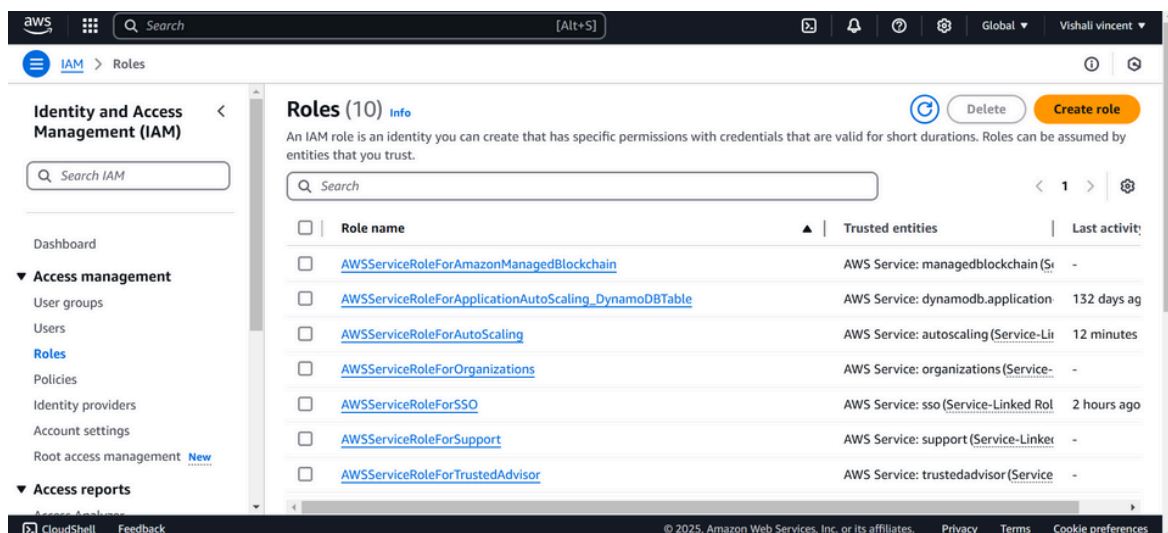
Step 1:

1. Go to [AWS Management Console](#).
2. Enter your username and password to log in.



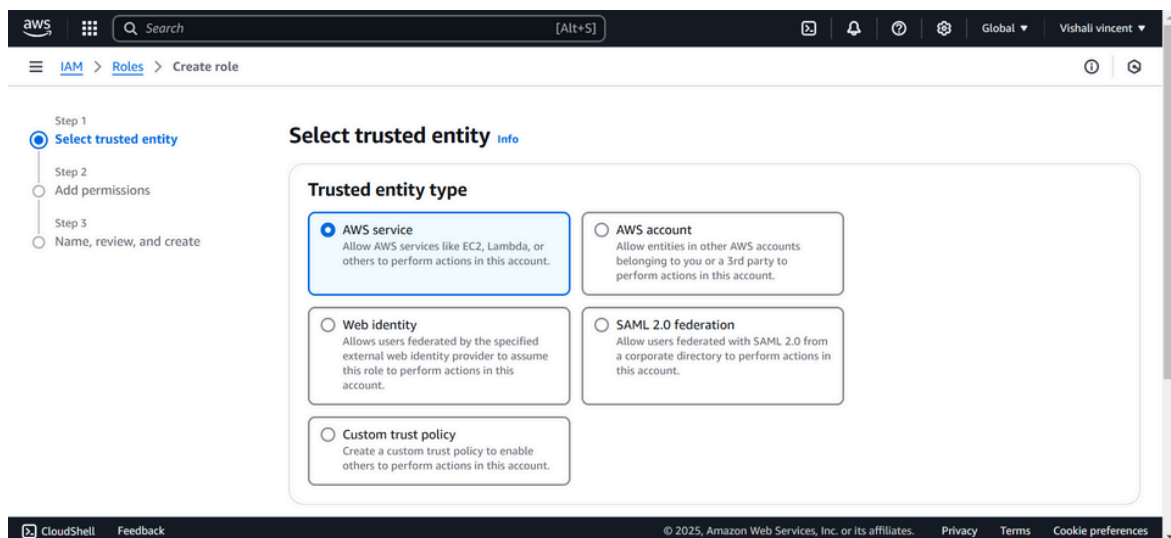
Step 2:

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



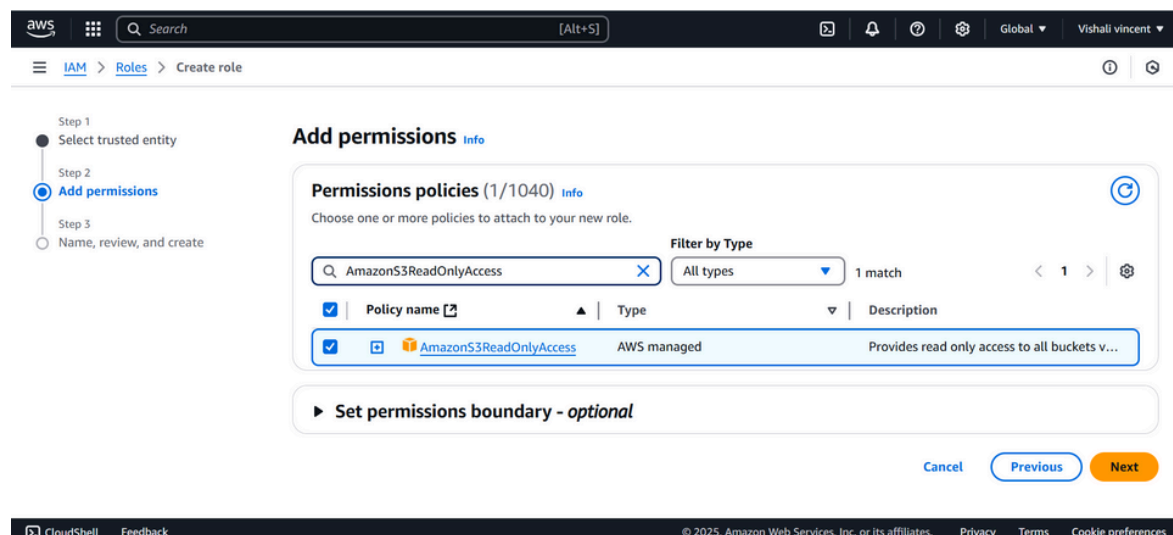
Step 3:

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



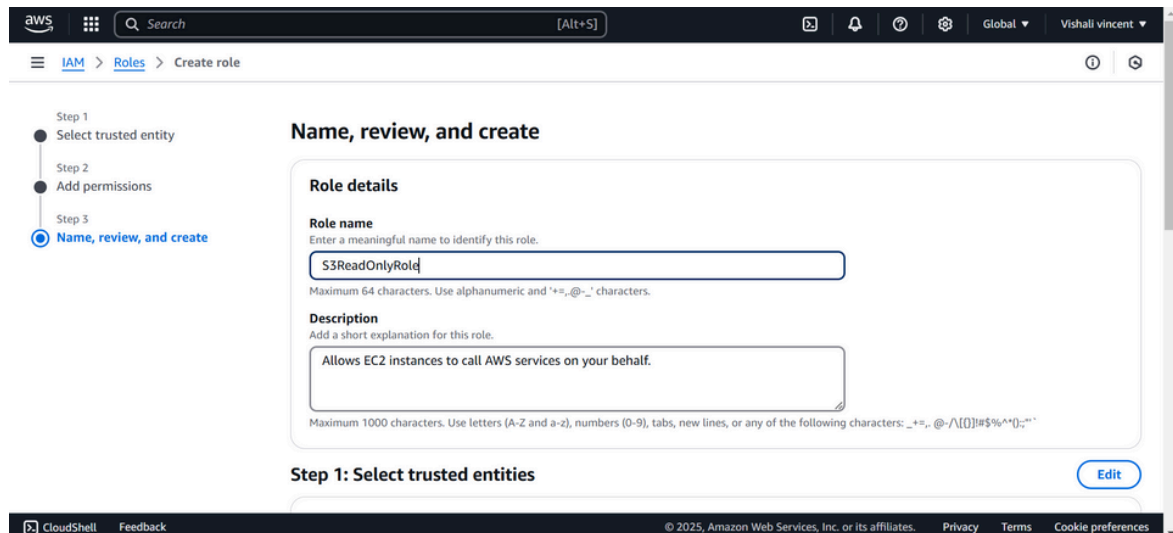
Step 4:

Search for AmazonS3ReadOnlyAccess and select it.



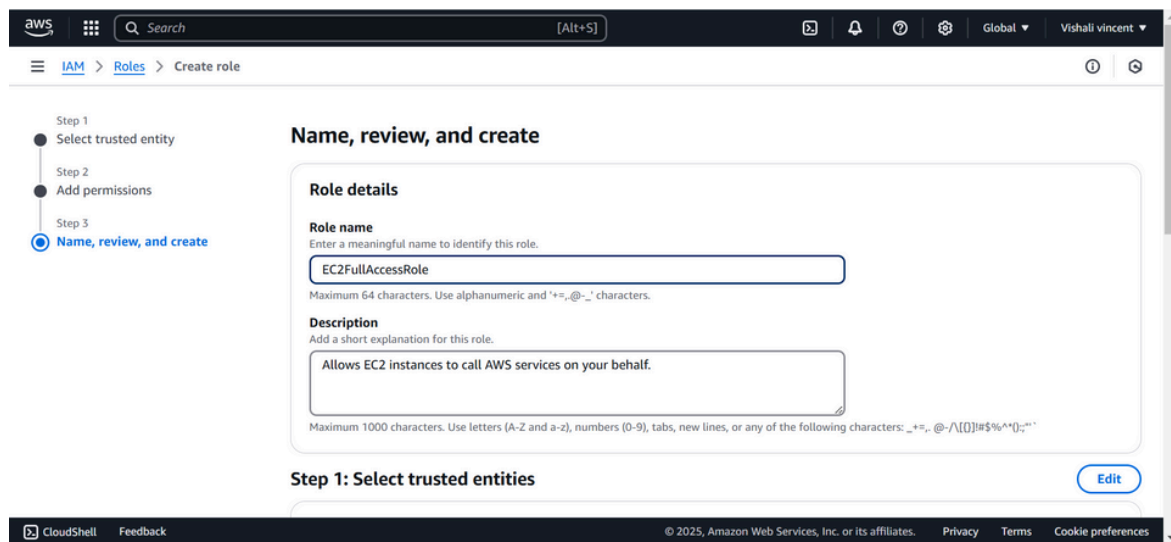
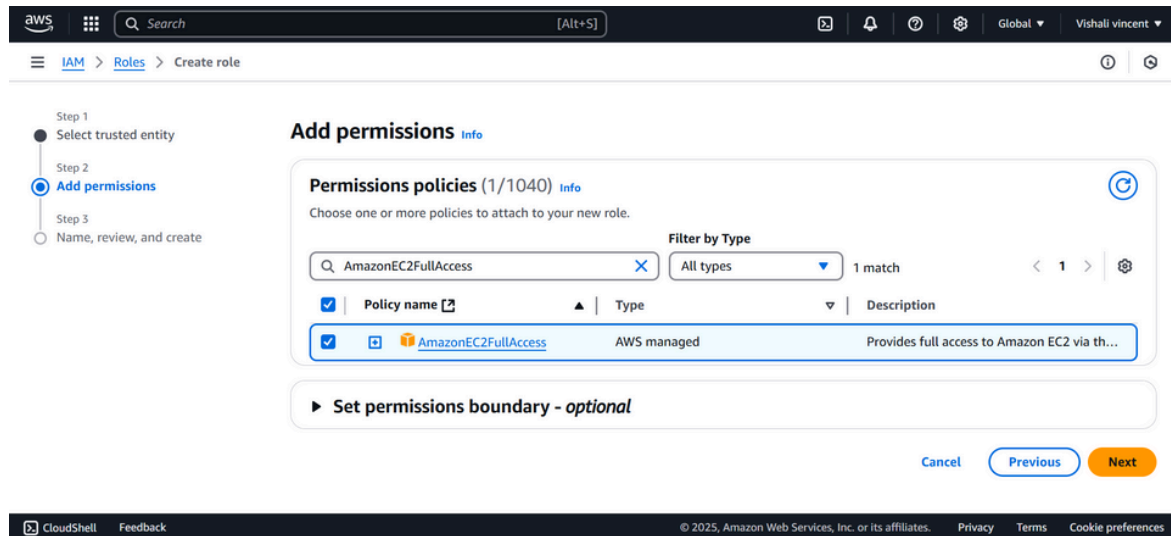
Step 5:

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



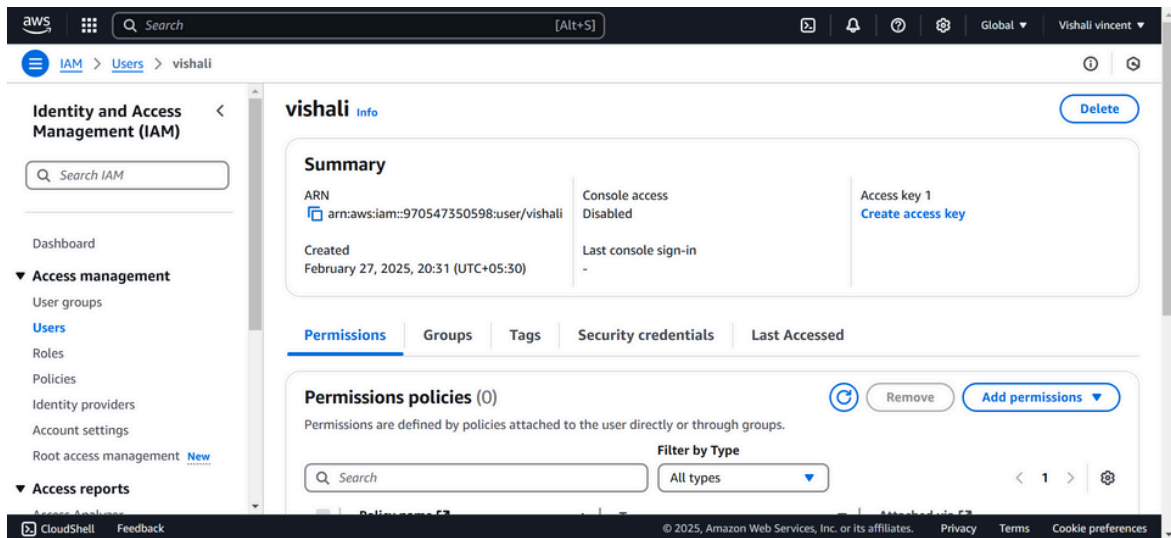
Step 6

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.

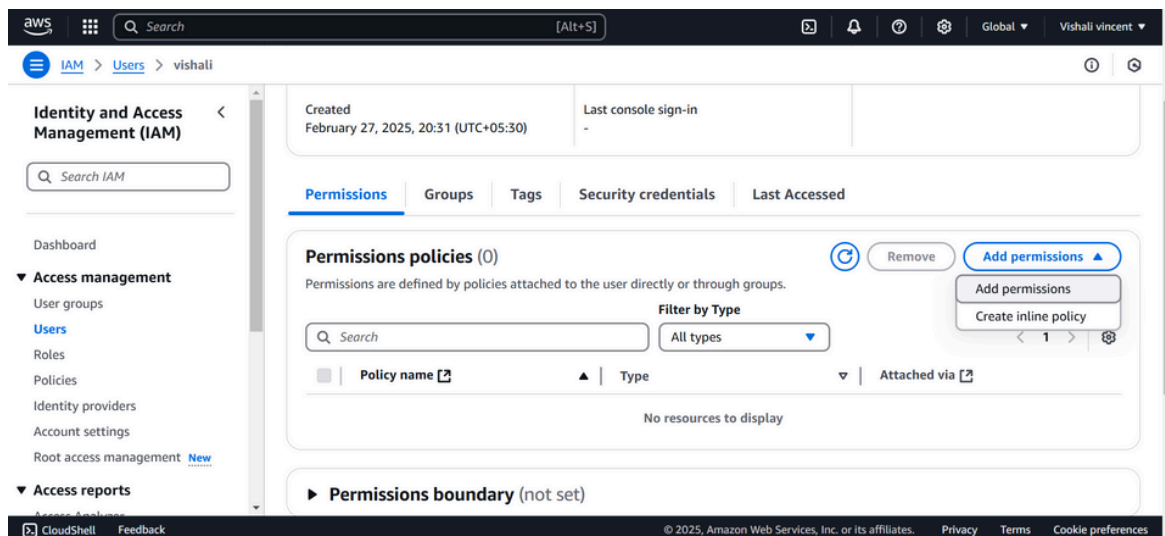


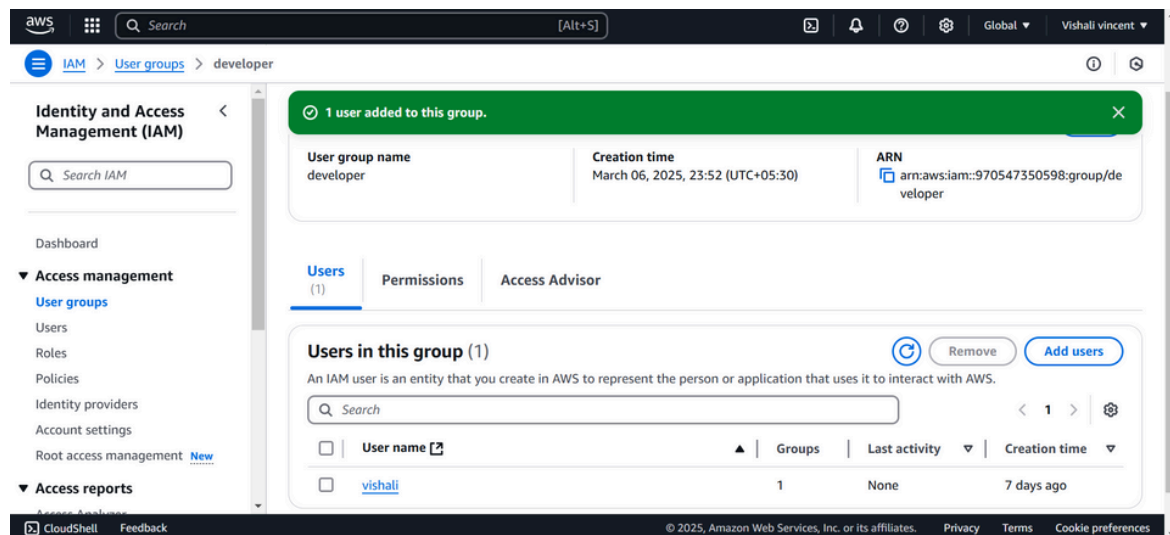
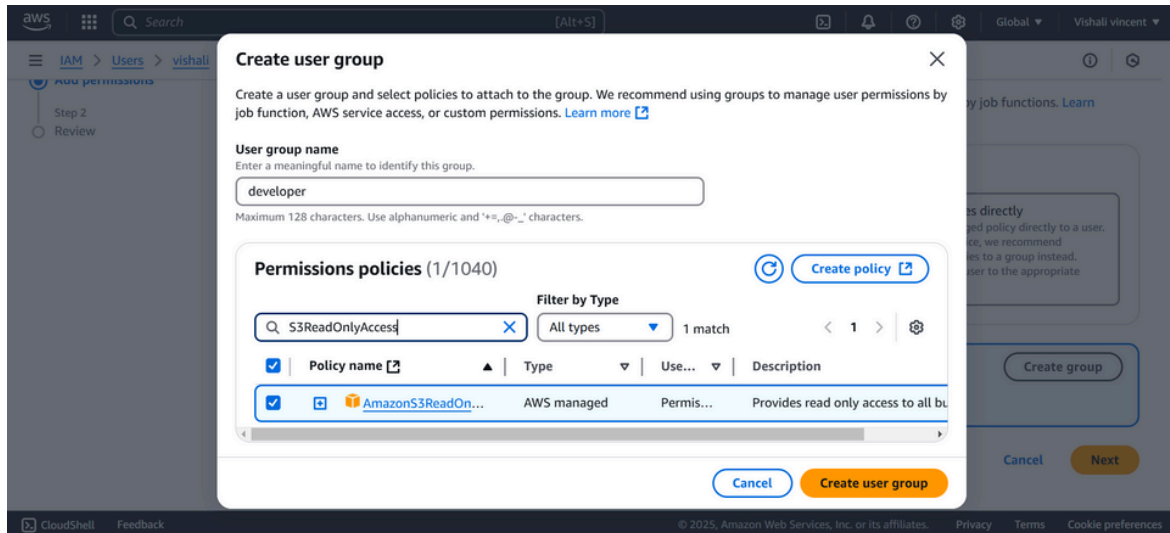
Step 8

1. Assign:

- ☒ S3ReadOnlyRole to one user.
- ☒ EC2FullAccessRole to another user.

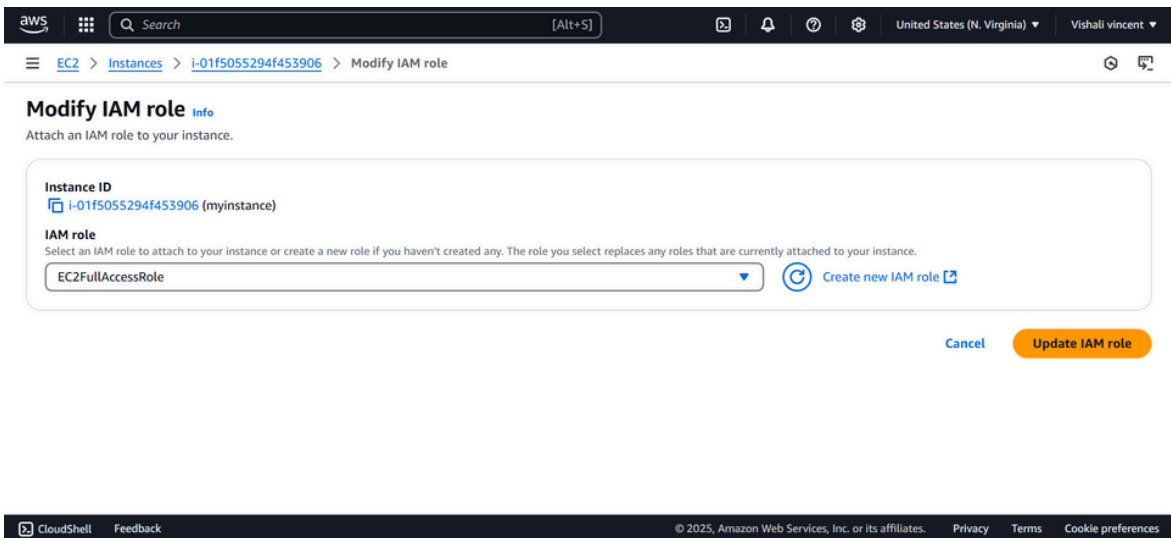
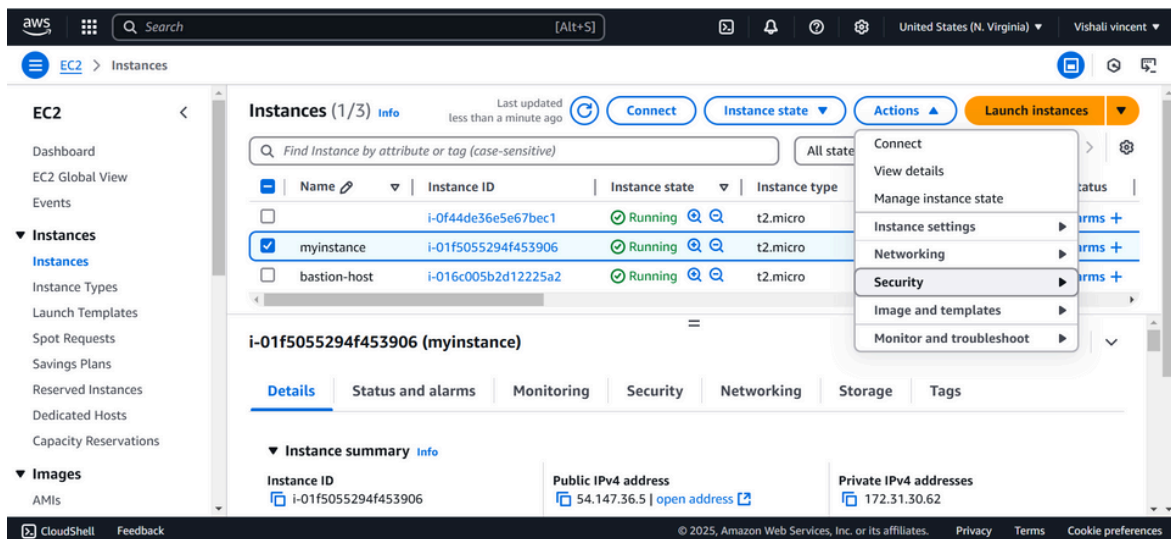
2. Click Next → Review → Add permissions.





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\Hi>aws s3 ls
2025-02-28 18:38:29 my-bucket--poc
2025-02-28 18:39:12 my-unique-bucket-123456789xyz

C:\Users\Hi>aws s3 mb s3://my-unique-bucket-123456789xyz
make_bucket failed: s3://my-unique-bucket-123456789xyz An error occurred (AccessDenied) when calling the CreateBucket operation: User: arn:aws:iam::343218194491:user/SaravanaKrishnan is not authorized to perform: s3:CreateBucket on resource: "arn:aws:s3::my-unique-bucket-123456789xyz" 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
```

3. It should succeed.

4. Try listing S3 buckets:

```
aws s3 ls
```

5. It should deny access.

```
C:\Users\Hi>aws ec2 run-instances --image-id ami-05b10e08d247fb927 --instance-type t2.micro

An error occurred (UnauthorizedOperation) when calling the RunInstances operation: You are not authorized to perform this operation . User: arn:aws:iam::343218194491:user/SaravanaKrishnan is not authorized to perform: ec2:RunInstances on resource: arn:aws:ec2:us-east-1:343218194491:instance/* because no identity-based policy allows the ec2:RunInstances action. Encoded authorization failure message: YP4wgAxS9_oFA-3UCgaB5_gkAaKCF9DHnfs_0rMbH35hskxcDvh1qL_fg460nK7Nn30GVcC_F2u0HSCI2Z4IQFfkErV2ZfkVdDf5Fk9WD_6LyS3oLkVgBcfAizn CtX4gaxxvandTSdoqex3jPeIAGxqEdS1Z2VHVYH6zzZ2wP00wPraeM6AEws_R5MNF9iRjXqqXB-2GTo51jas3vA8LM1eiyDALbYVgN8N4fFbwJrd6x43wm9-kmCeaHwb7t7 i-Okcx_wkNGZYAZXHzUniA4Z9x-n6_HQvqODCFXLYVOoz_yxDfvesnNHvSXGXhovOH0iB_rMAM4YD3Z6NGm8h0mKX3hRVpJ6oU-Byu0hbpXHJhK-WMDEuBM38Y_vcSyj5KG ynSaj4MwyGTpuRMFi1waNKF2ArPT4D2fIHmDjC2QmSf7Nme8xx8vCAOo-LhwW0gYTi-EvgXMTiHyfP4WgnMRoUuSTAst_gecucKgvAbJWqgAhIiUtFmH1w63T2QbPQjwTS7 2pli0H4_JGbBJIBK3CaQln1BaTncY1Ye2XTXyULe0mQVYujPqibkT-M08uqQde24Hn8GPAjHBT9QpiSv45Ji--VmB9FKVln2ip6kx74pt9r5PIr6YD5D_YP81GEeDeJdBsx 8yQoV2JsrtdI0sJ--AYSepy9cT4NAVQGoZiMyPyiaW0u3sygtg3E1H1zXkZEhX_xB-0H4hw7SVyrOp4c2zmOuPEQ0hs0cp5Wm0J5Q5h0m4PLueRIBYw7s974resjYkKzIx YFnt--CIV_
```

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.