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: Saravana Krishnan J   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.

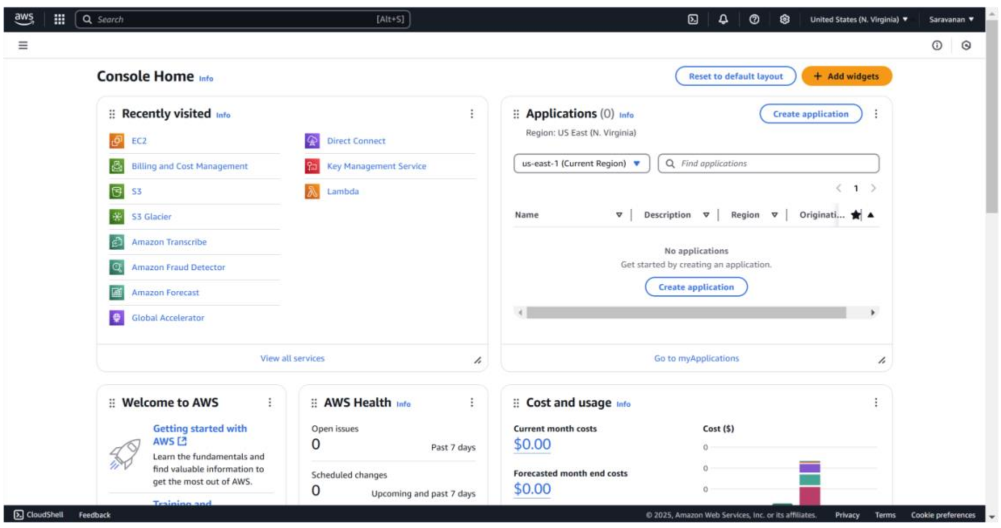
4. **Prevents Costly Mistakes** – Avoids accidental resource modifications/deletions.

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.



Console Home Info

Reset to default layout | + Add widgets

Recently visited Info

- EC2
- Billing and Cost Management
- S3
- S3 Glacier
- Amazon Transcribe
- Amazon Fraud Detector
- Amazon Forecast
- Global Accelerator
- Direct Connect
- Key Management Service
- Lambda

View all services

Applications (0) Info

Create application

Region: US East (N. Virginia)

us-east-1 (Current Region) | Find applications

Name	Description	Region	Originati...

No applications

Get started by creating an application.

Create application

Go to myApplications

Welcome to AWS

Getting started with AWS

Learn the fundamentals and find valuable information to get the most out of AWS.

AWS Health Info

Open issues

0 Past 7 days

Scheduled changes

0 Upcoming and past 7 days

Cost and usage Info

Current month costs

$0.00

Forecasted month end costs

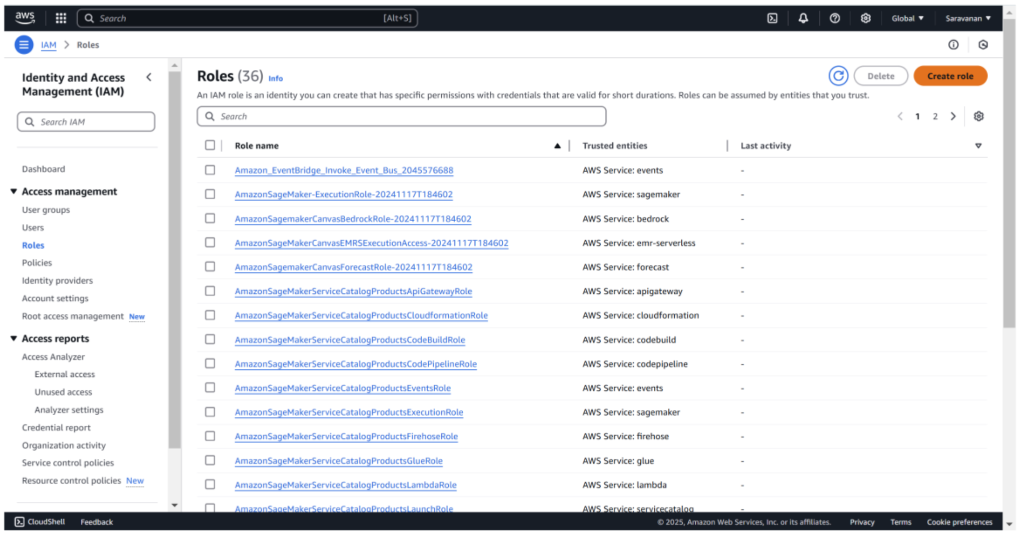
$0.00

Cost ($)

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

Step 2:

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



IAM > Roles

Identity and Access Management (IAM)

- Dashboard
- Access management
  - User groups
  - Users
  - Roles
  - Policies
  - Identity providers
  - Account settings
  - Root access management New
- Access reports
  - Access Analyzer
    - External access
    - Unused access
    - Analyzer settings
  - Credential report
  - Organization activity
  - Service control policies
  - Resource control policies New

Roles (36) Info

Delete | Create role

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
Amazon_EventBridge_Invoke_Event_Bus_2045576688	AWS Service: events	-
AmazonSageMaker-ExecutionRole-20241117T184602	AWS Service: sagemaker	-
AmazonSagemakerCanvasBedrockRole-20241117T184602	AWS Service: bedrock	-
AmazonSageMakerCanvasEMRSExecutionAccess-20241117T184602	AWS Service: emr-serverless	-
AmazonSagemakerCanvasForecastRole-20241117T184602	AWS Service: forecast	-
AmazonSageMakerServiceCatalogProductsApiGatewayRole	AWS Service: apigateway	-
AmazonSageMakerServiceCatalogProductsCloudformationRole	AWS Service: cloudformation	-
AmazonSageMakerServiceCatalogProductsCodeBuildRole	AWS Service: codebuild	-
AmazonSageMakerServiceCatalogProductsCodePipelineRole	AWS Service: codepipeline	-
AmazonSageMakerServiceCatalogProductsEventsRole	AWS Service: events	-
AmazonSageMakerServiceCatalogProductsExecutionRole	AWS Service: sagemaker	-
AmazonSageMakerServiceCatalogProductsFirehoseRole	AWS Service: firehose	-
AmazonSageMakerServiceCatalogProductsGlueRole	AWS Service: glue	-
AmazonSageMakerServiceCatalogProductsLambdaRole	AWS Service: lambda	-
AmazonSageMakerServiceCatalogProductsLaunchRole	AWS Service: servicecatalog	-

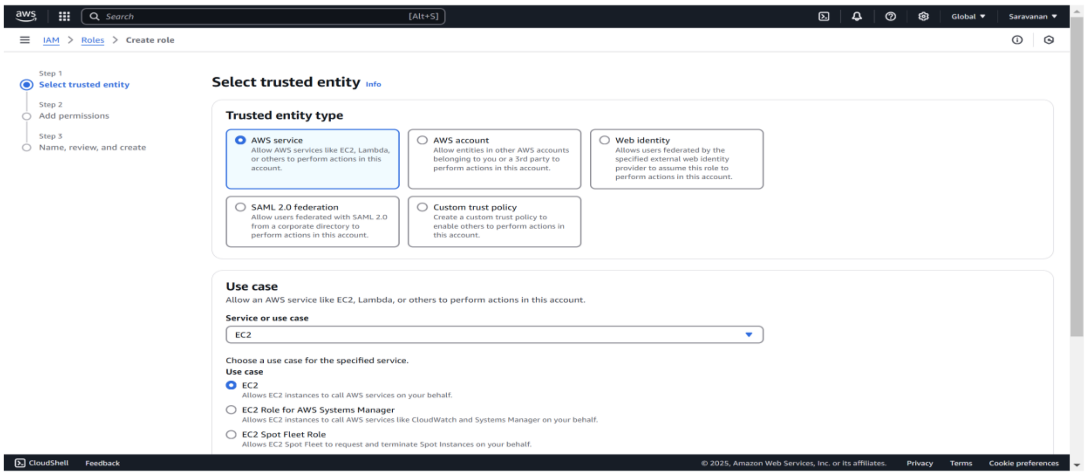
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 3:

1. **Select trusted entity:** Choose **AWS Service**.

2. **Use case:** Select **EC2** role for an instance.

3. Click **Next**.



IAM > Roles > Create role

Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create

Select trusted entity Info

Trusted entity type

- [✓ AWS service] Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account: Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity: Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation: Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy: Create a custom trust policy to enable others to perform actions in this account.

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

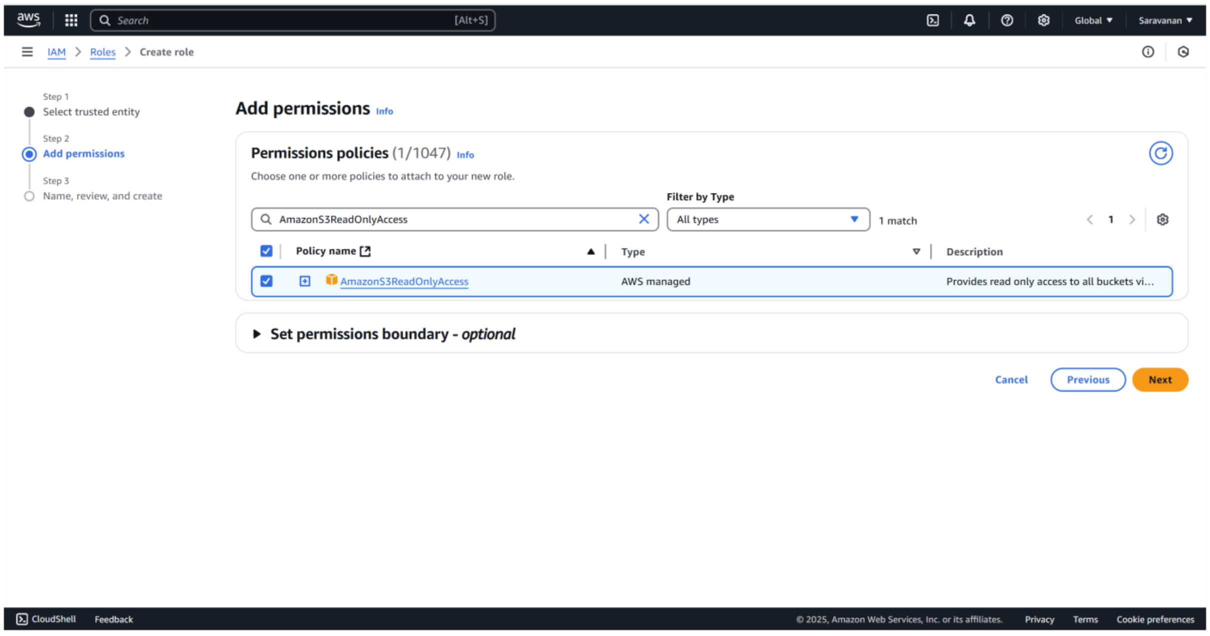
Service or use case: EC2

Choose a use case for the specified service.
Use case
- [✓ EC2] Allows EC2 instances to call AWS services on your behalf.
- EC2 Role for AWS Systems Manager: Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
- EC2 Spot Fleet Role: Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.

© 2025, Amazon Web Services, Inc. or its affiliates.

Step 4:

Search for **AmazonS3ReadOnlyAccess** and select it.



IAM > Roles > Create role

Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create

Add permissions Info

Permissions policies (1/1047) Info
Choose one or more policies to attach to your new role.

Search: AmazonS3ReadOnlyAccess
Filter by Type: All types
1 match

	Policy name	Type	Description
✓	AmazonS3ReadOnlyAccess	AWS managed	Provides read only access to all buckets vi…

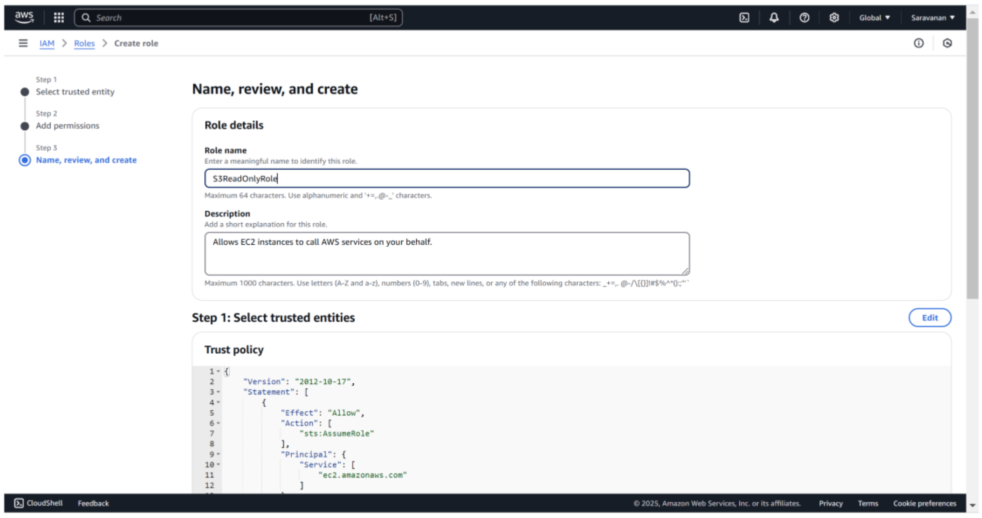
Set permissions boundary - *optional*

Cancel Previous Next

© 2025, Amazon Web Services, Inc. or its affiliates.

Step 5:

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



Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.
S3ReadOnlyRole
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{}]!#$%^*();:"'

Step 1: Select trusted entities

Trust policy

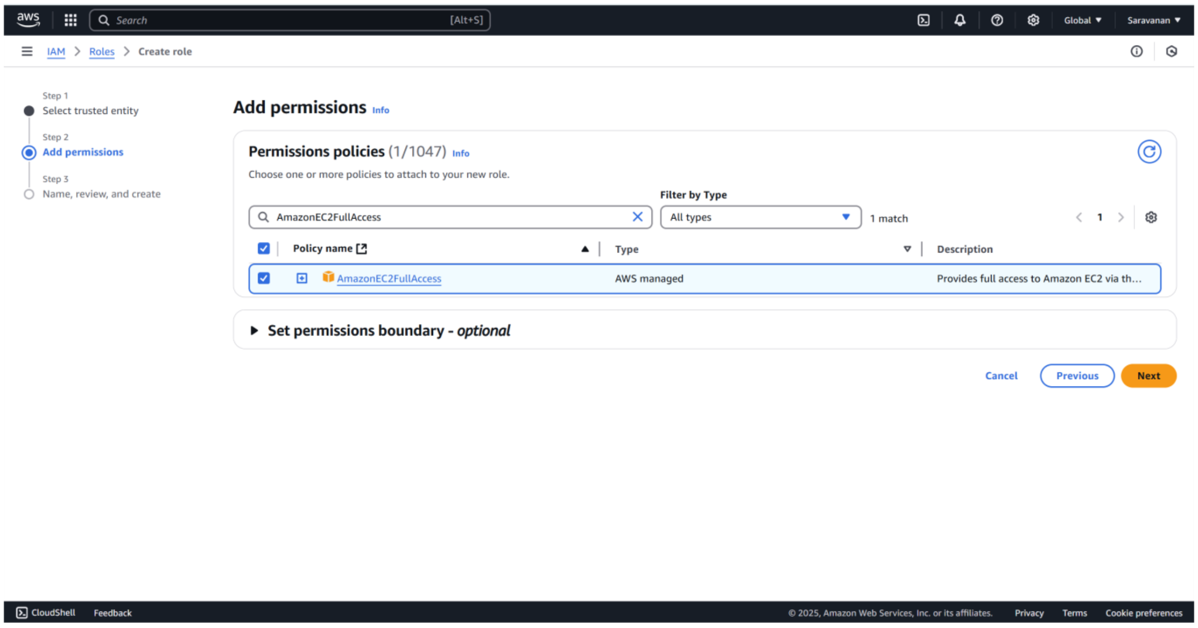
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
                "Service": [
                    "ec2.amazonaws.com"
                ]
```

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**.



IAM > Roles > Create role

Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create

Add permissions Info

Permissions policies (1/1047) Info

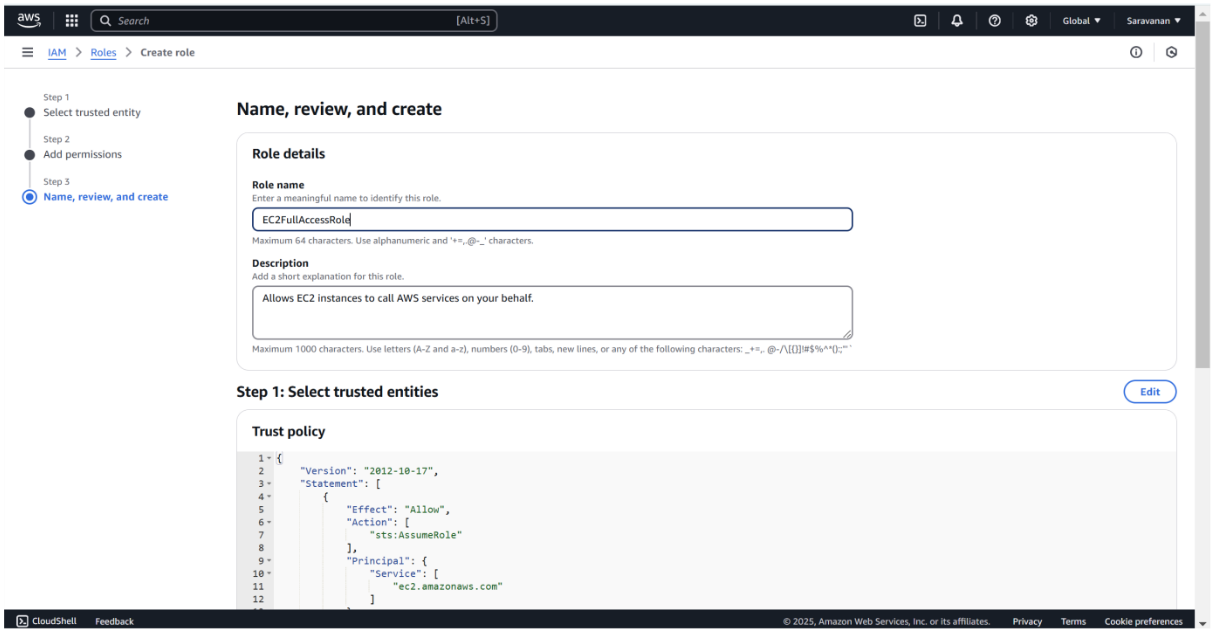
Choose one or more policies to attach to your new role.

Filter by Type

AmazonEC2FullAccess | All types | 1 match

Policy name	Type	Description
✓ AmazonEC2FullAccess	AWS managed	Provides full access to Amazon EC2 via th...

▶ **Set permissions boundary** - *optional*



IAM > Roles > Create role

Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

EC2FullAccessRole

Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,.@-/\[{]}!#$%^*();:"'`

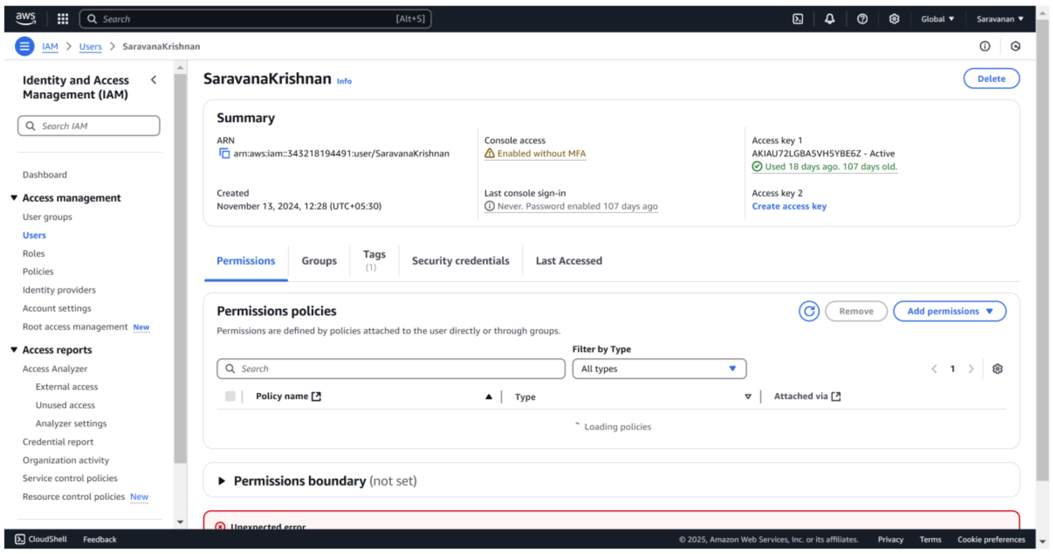
Step 1: Select trusted entities

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
                "Service": [
                    "ec2.amazonaws.com"
                ]
```

Step 7

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


IAM > Users > SaravanaKrishnan

Identity and Access Management (IAM)

Dashboard
Access management: User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management New
Access reports: Access Analyzer, External access, Unused access, Analyzer settings, Credential report, Organization activity, Service control policies, Resource control policies New

SaravanaKrishnan Info

Summary
ARN: arn:aws:iam::343218194491:user/SaravanaKrishnan
Console access: Enabled without MFA
Access key 1: AKIAU72LGBA5VH5YBE6Z - Active. Used 18 days ago. 107 days old.
Created: November 13, 2024, 12:28 (UTC+05:30)
Last console sign-in: Never. Password enabled 107 days ago
Access key 2: Create access key

Permissions | Groups | Tags (1) | Security credentials | Last Accessed

Permissions policies
Permissions are defined by policies attached to the user directly or through groups.
Filter by Type: All types
Policy name | Type | Attached via
Loading policies

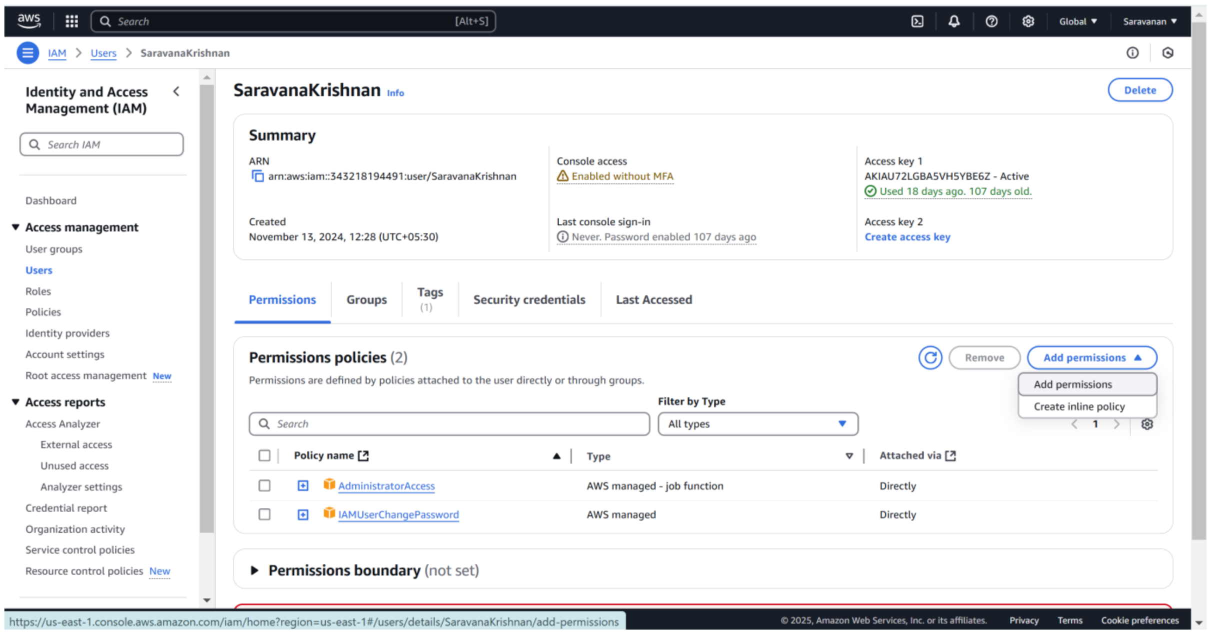
Permissions boundary (not set)

Unexpected error

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

Step 8

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


IAM > Users > SaravanaKrishnan

SaravanaKrishnan Info

Summary
ARN: arn:aws:iam::343218194491:user/SaravanaKrishnan
Console access: Enabled without MFA
Access key 1: AKIAU72LGBA5VH5YBE6Z - Active. Used 18 days ago. 107 days old.
Created: November 13, 2024, 12:28 (UTC+05:30)
Last console sign-in: Never. Password enabled 107 days ago
Access key 2: Create access key

Permissions | Groups | Tags (1) | Security credentials | Last Accessed

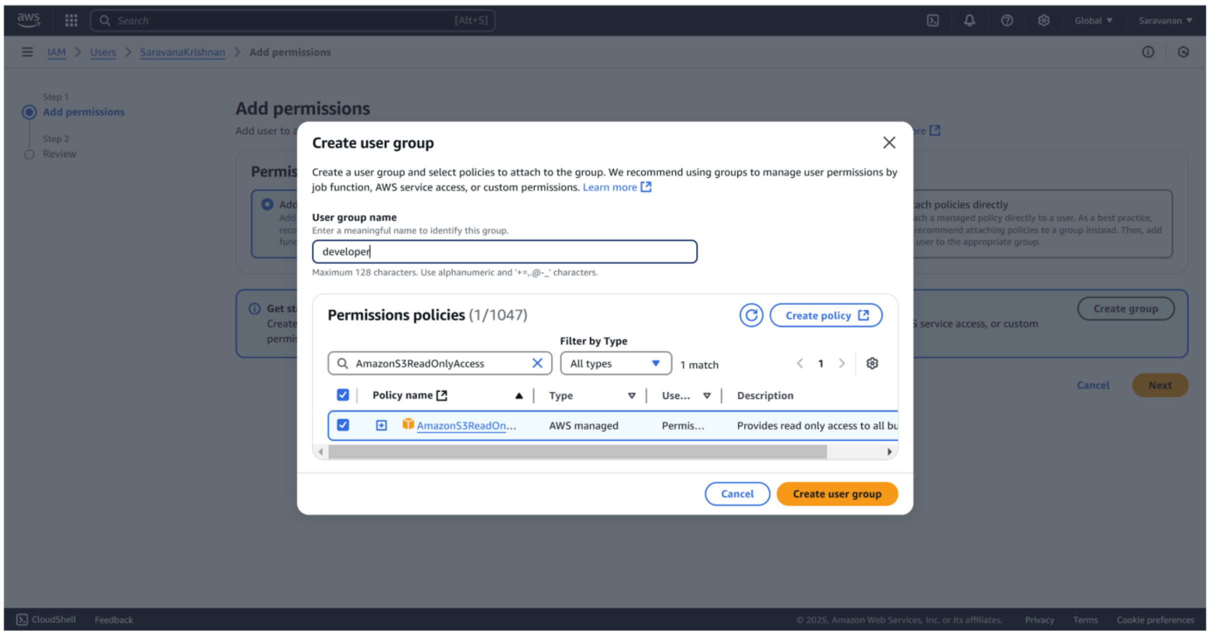
Permissions policies (2)
Permissions are defined by policies attached to the user directly or through groups.
Add permissions ▲: Add permissions / Create inline policy
Filter by Type: All types

Policy name	Type	Attached via
AdministratorAccess	AWS managed - job function	Directly
IAMUserChangePassword	AWS managed	Directly

Permissions boundary (not set)

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/details/SaravanaKrishnan/add-permissions

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


IAM > Users > SaravanaKrishnan > Add permissions

Step 1
Add permissions

Step 2
Review

Create user group

Create a user group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. Learn more

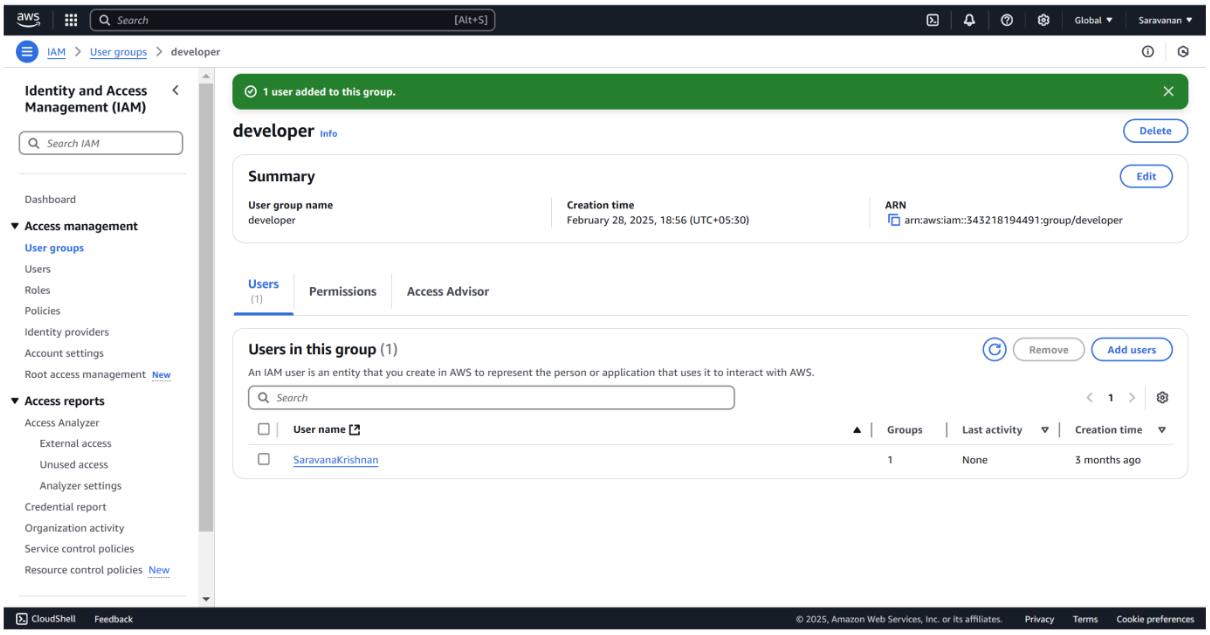
User group name
Enter a meaningful name to identify this group.
developer
Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Permissions policies (1/1047)

Filter by Type
AmazonS3ReadOnlyAccess | All types | 1 match

✓	Policy name	Type	Use...	Description
✓	AmazonS3ReadOn...	AWS managed	Permis...	Provides read only access to all bu

Cancel | Create user group


IAM > User groups > developer

1 user added to this group.

developer Info

Summary

User group name	Creation time	ARN
developer	February 28, 2025, 18:56 (UTC+05:30)	arn:aws:iam::343218194491:group/developer

Users (1) | Permissions | Access Advisor

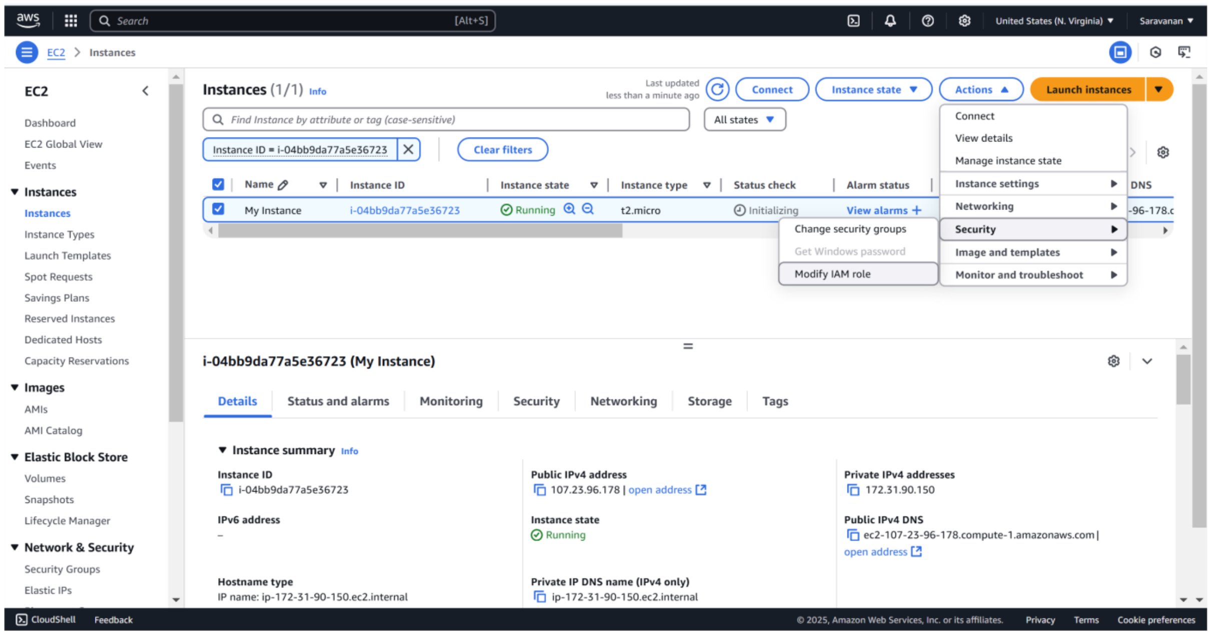
Users in this group (1)

An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

User name	Groups	Last activity	Creation time
SaravanaKrishnan	1	None	3 months ago

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**.



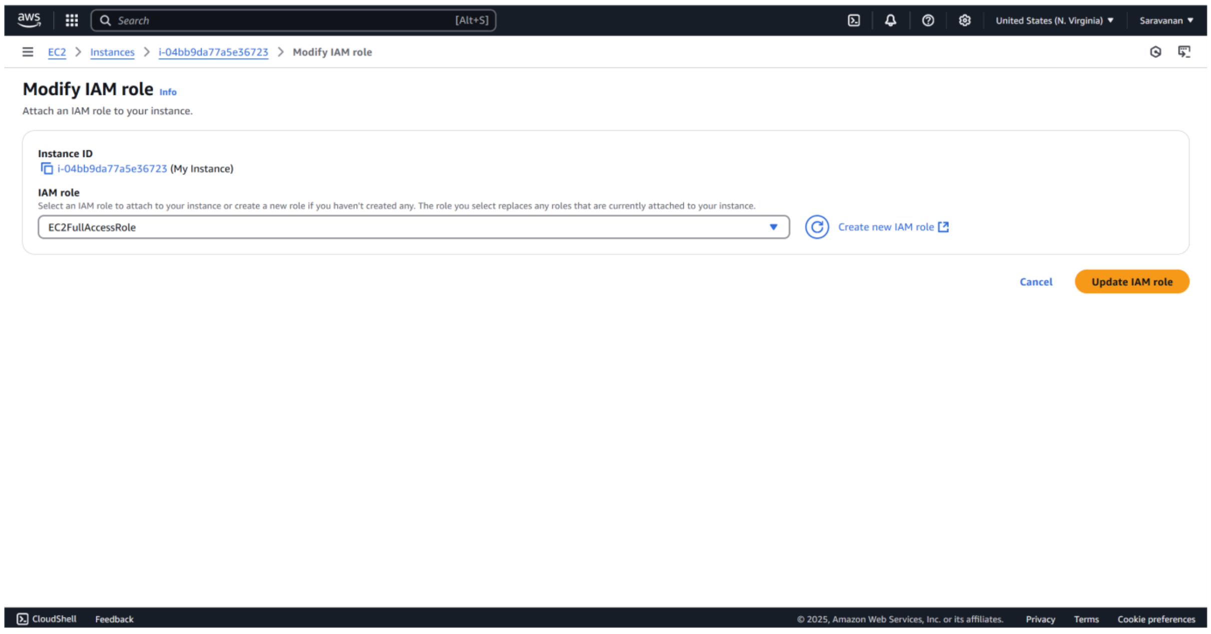
Instance ID = i-04bb9da77a5e36723

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
My Instance	i-04bb9da77a5e36723	Running	t2.micro	Initializing	View alarms

i-04bb9da77a5e36723 (My Instance)

Instance summary

Instance ID: i-04bb9da77a5e36723
Public IPv4 address: 107.23.96.178 | open address
Private IPv4 addresses: 172.31.90.150
IPv6 address: –
Instance state: Running
Public IPv4 DNS: ec2-107-23-96-178.compute-1.amazonaws.com | open address
Hostname type: IP name: ip-172-31-90-150.ec2.internal
Private IP DNS name (IPv4 only): ip-172-31-90-150.ec2.internal



Modify IAM role

Attach an IAM role to your instance.

Instance ID: i-04bb9da77a5e36723 (My Instance)

IAM role: Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

EC2FullAccessRole

Step 9

Open Command prompt

1. Run:

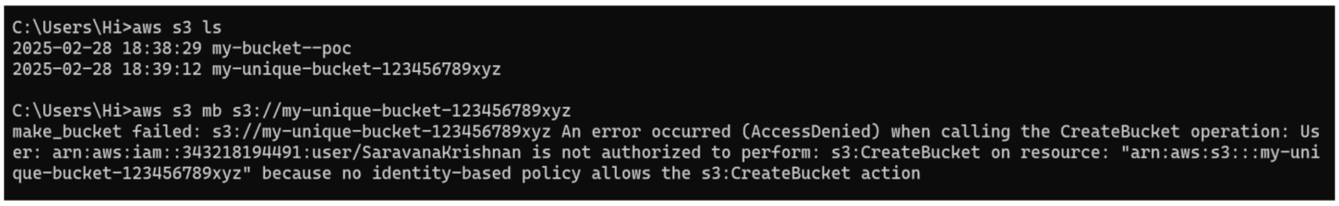
   aws s3 ls

   ✅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: Us
er: arn:aws:iam::343218194491:user/SaravanaKrishnan is not authorized to perform: s3:CreateBucket on resource: "arn:aws:s3:::my-uni
que-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:

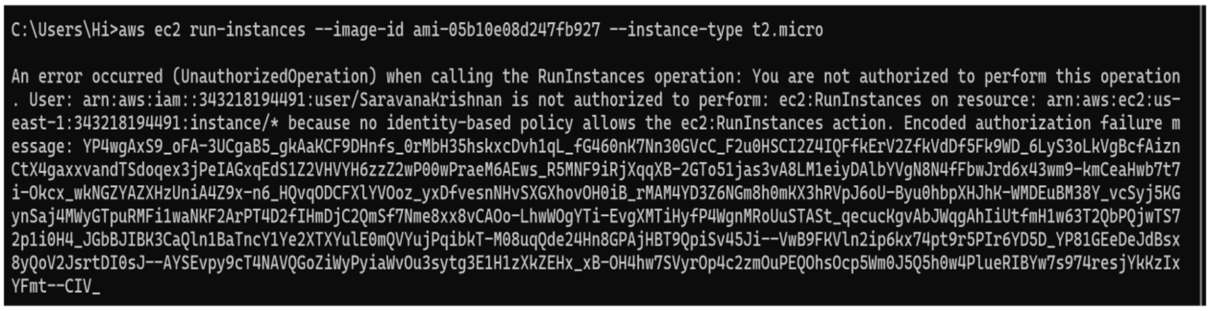
aws ec2 run-instances --image-id ami-12345678 --instance-type t2.micro

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 m
essage: YP4wgAxS9_oFA-3UCgaB5_gkAaKCF9DHnfs_0rMbH35hskxcDvh1qL_fG460nK7Nn30GVcC_F2u0HSCI2Z4IQFfkErV2ZfkVdDf5Fk9WD_6LyS3oLkVgBcfAizn
CtX4gaxxvandTSdoqex3jPeIAGxqEdS1Z2VHVYH6zzZ2wP00wPraeM6AEws_R5MNF9iRjXqqXB-2GTo51jas3vA8LM1eiyDAlbYVgN8N4fFbwJrd6x43wm9-kmCeaHwb7t7
i-Okcx_wkNGZYAZXHzUniA4Z9x-n6_HQvqODCFXlYVOoz_yxDfvesnNHvSXGXhovOH0iB_rMAM4YD3Z6NGm8h0mKX3hRVpJ6oU-Byu0hbpXHJhK-WMDEuBM38Y_vcSyj5KG
ynSaj4MWyGTpuRMFi1waNKF2ArPT4D2fIHmDjC2QmSf7Nme8xx8vCAOo-LhwWOgYTi-EvgXMTiHyfP4WgnMRoUuSTASt_qecucKgvAbJWqgAhIiUtfmH1w63T2QbPQjwTS7
2p1i0H4_JGbBJIBK3CaQln1BaTncY1Ye2XTXYulE0mQVYujPqibkT-M08uqQde24Hn8GPAjHBT9QpiSv45Ji--VwB9FKVln2ip6kx74pt9r5PIr6YD5D_YP81GEeDeJdBsx
8yQoV2JsrtDI0sJ--AYSEvpy9cT4NAVQGoZiWyPyiaWvOu3sytg3E1H1zXkZEHx_xB-OH4hw7SVyrOp4c2zmOuPEQOhsOcp5Wm0J5Q5h0w4PlueRIBYw7s974resjYkKzIx
YFmt--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.