

## **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: Hema S

Department: ECE

# 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.

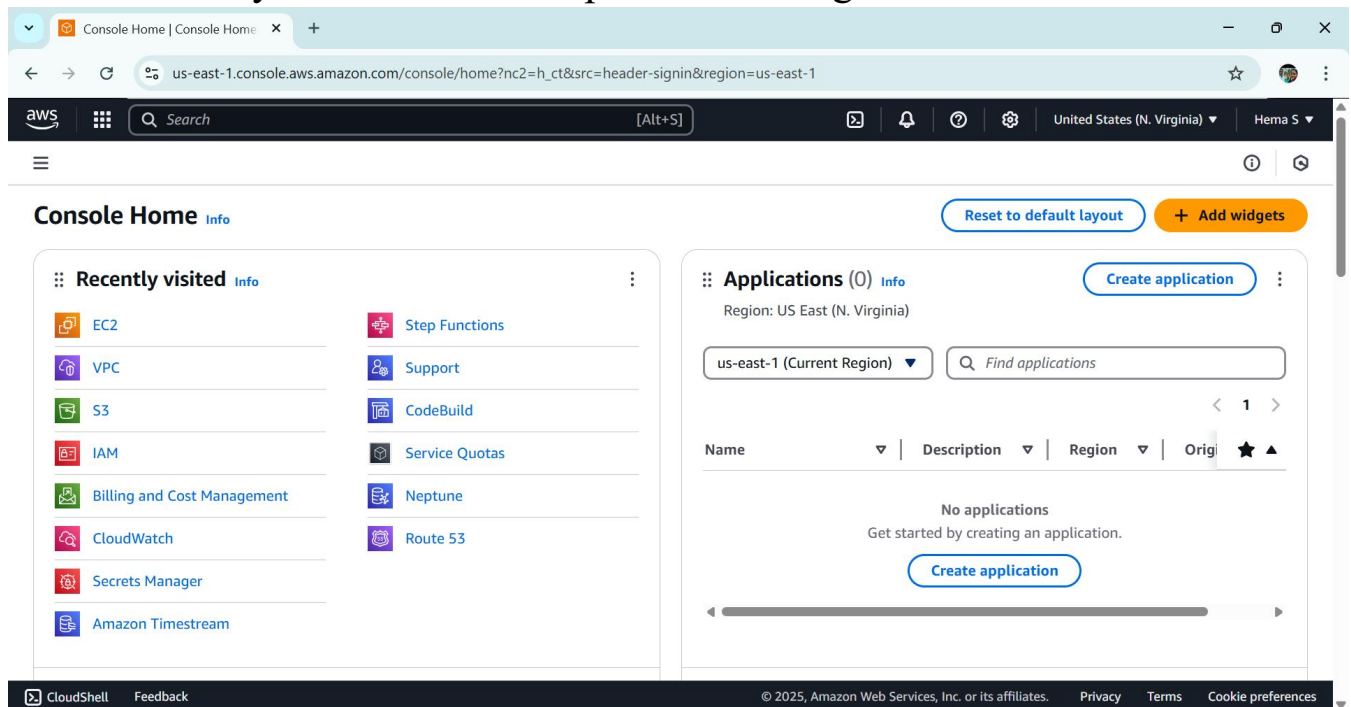
# Importance

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](https://aws.amazon.com/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.**

Roles | IAM | Global

us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles

aws

Search

[Alt+S]

Global

Hema S

IAM > Roles

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

Root access management

Access reports

Roles (8)

Info

Refresh

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.

Search

< 1 >

Settings

<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	<a href="#">AWSServiceRoleForAutoScaling</a>	AWS Service: autoscaling (Service-Linker)	27 days ago
<input type="checkbox"/>	<a href="#">AWSServiceRoleForElasticLoadBalancing</a>	AWS Service: elasticloadbalancing (Service-Linker)	-
<input type="checkbox"/>	<a href="#">AWSServiceRoleForSupport</a>	AWS Service: support (Service-Linker)	114 days ago
<input type="checkbox"/>	<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor (Service-Linker)	-
<input type="checkbox"/>	<a href="#">My-EC2-S3-Access-Role</a>	AWS Service: ec2	-
<input type="checkbox"/>	<a href="#">StepFunctions-MyStateMachine-9xg922nhi-role-h3g1thvak</a>	AWS Service: states	-
<input type="checkbox"/>	<a href="#">StepFunctions-MyStateMachine-dg1mayoq2-role-vyho58tv9</a>	AWS Service: states	-

CloudShell

Feedback

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

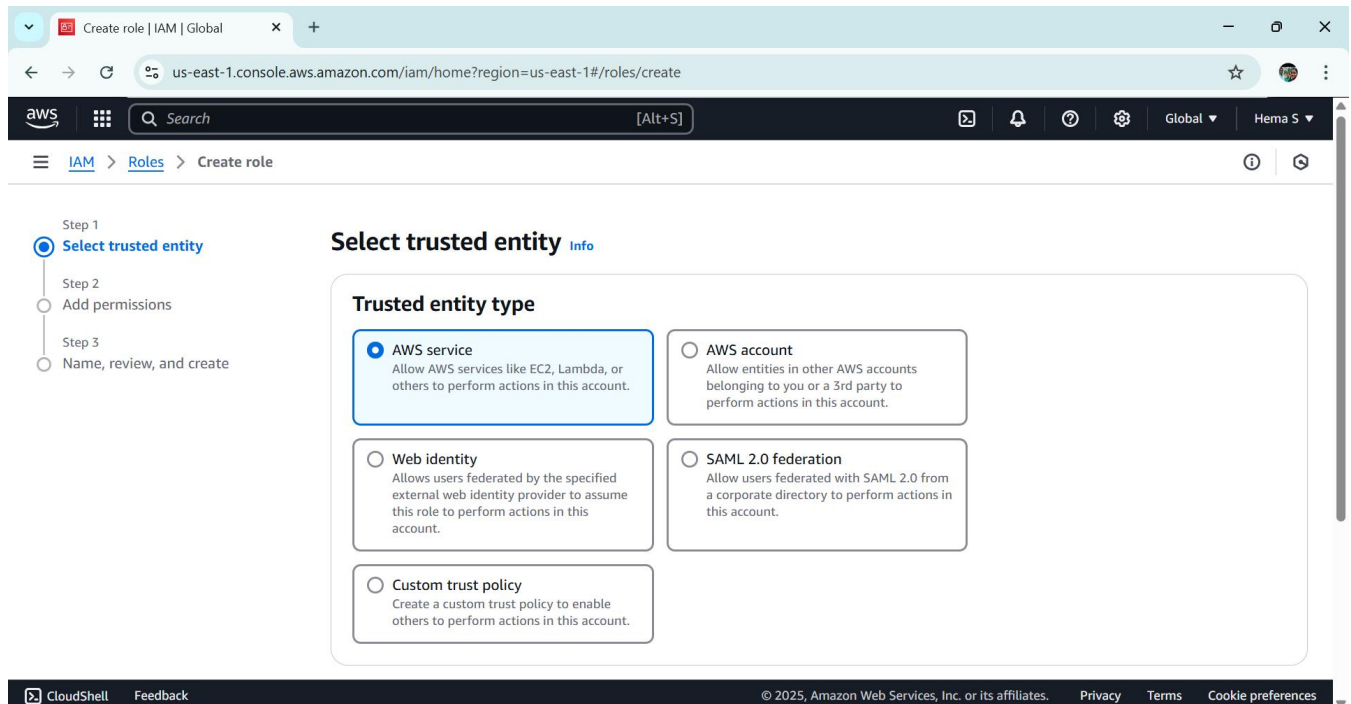
Privacy

Terms

Cookie preferences

## Step 3:

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



## Step 4:

Search for **AmazonS3ReadOnlyAccess** and select it.

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

### Add permissions [Info](#)

#### Permissions policies (1/1041) [Info](#)

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

Filter by Type

<input type="text" value="amazons3readonlyaccess"/>	<input type="button" value="X"/>	<input type="text" value="All types"/>	1 match	<input type="button" value="1"/>	<input type="button" value="X"/>
<input checked="" type="checkbox"/>	Policy name <a href="#">?</a>	▲	Type	▼	Description
<input checked="" type="checkbox"/>	<input type="button" value="+"/> <a href="#">AmazonS3ReadOnlyAccess</a>		AWS managed		Provides read only access to all buckets v...

► Set permissions boundary - *optional*

[Cancel](#) [Previous](#) [Next](#)

## Step 5:

1. Click **Next** → Name the role **S3ReadOnlyRole**.

2. Click **Create Role**.

The screenshot shows the AWS IAM console 'Create role' page. The breadcrumb trail is 'IAM > Roles > Create role'. On the left, a progress bar shows three steps: 'Step 1: Select trusted entity', 'Step 2: Add permissions', and 'Step 3: Name, review, and create' (which is the active step). The main content area is titled 'Name, review, and create'. It contains a 'Role details' section with two fields: 'Role name' and 'Description'. The 'Role name' field has the value 'S3ReadOnlyRole' and a note: 'Enter a meaningful name to identify this role. Maximum 64 characters. Use alphanumeric and '+=, @-\_' characters.' The 'Description' field has the value 'Allows EC2 instances to call AWS services on your behalf.' and a note: 'Add a short explanation for this role. Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: \_+=, @-/[()!#\$%^&\*()-~'''. Below the 'Role details' section is a section titled 'Step 1: Select trusted entities' with an 'Edit' button. The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates, along with links for 'Privacy', 'Terms', and 'Cookie preferences'.

## 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**.



The image displays two screenshots of the AWS IAM console interface for creating a new role.

**Top Screenshot: Add permissions**

- Navigation:** IAM > Roles > Create role
- Steps:** Step 1: Select trusted entity, Step 2: Add permissions (active), Step 3: Name, review, and create
- Section:** Add permissions [Info](#)
- Permissions policies (1/1041) [Info](#)**
  - Choose one or more policies to attach to your new role.
  - Filter by Type:  All types 1 match
  - Table:

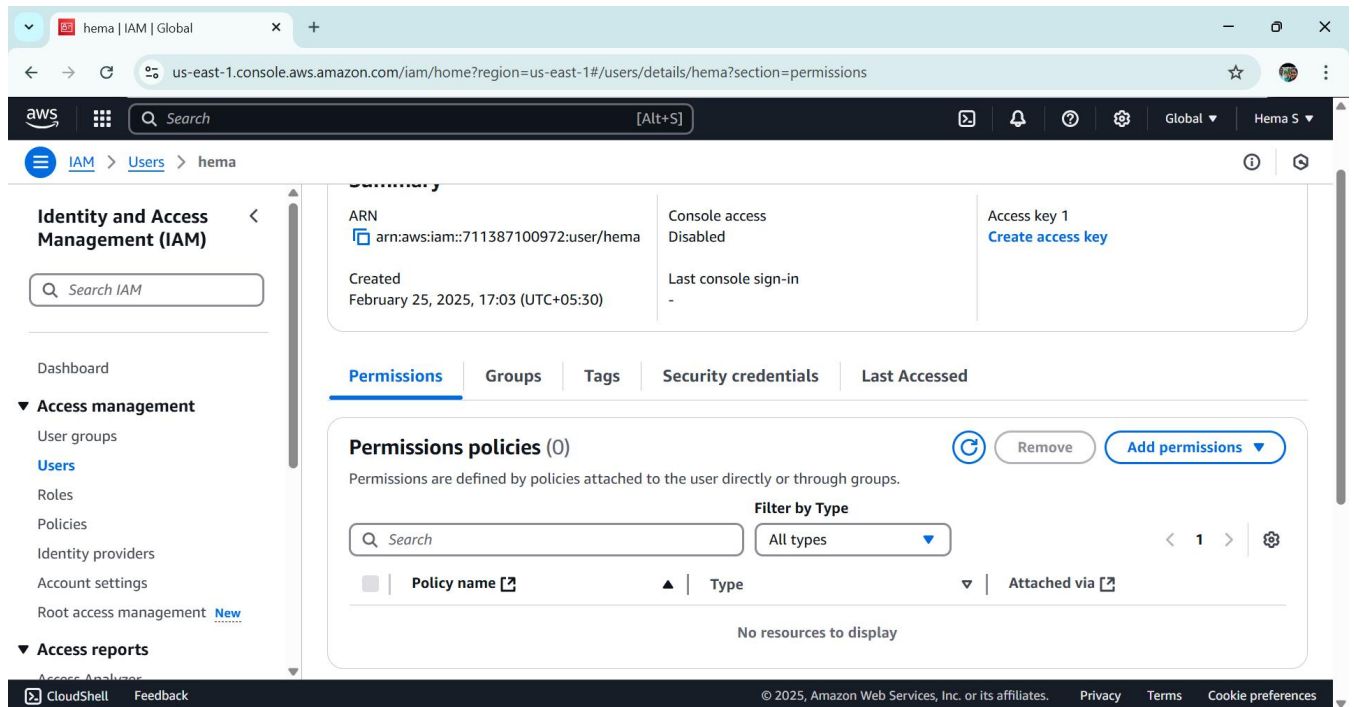
<input checked="" type="checkbox"/>	Policy name <a href="#">?</a>	Type	Description
<input checked="" type="checkbox"/>	AmazonEC2FullAccess	AWS managed	Provides full access to Amazon EC2 via t...
- Buttons:** Cancel, Previous, Next

**Bottom Screenshot: Name, review, and create**

- Navigation:** IAM > Roles > Create role
- Steps:** Step 1: Select trusted entity, Step 2: Add permissions, Step 3: Name, review, and create (active)
- Section:** Name, review, and create
- Role details**
  - Role name**  
Enter a meaningful name to identify this role.  
  
Maximum 64 characters. Use alphanumeric and '+=, @-\_' characters.
  - Description**  
Add a short explanation for this role.  
  
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: \_+=, @-/\[\]:!#\$%^&\*()~'"
- Buttons:** Edit

## 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**.





Add permissions | IAM | Global

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

aws

Search

[Alt+S]



Global

Hema S

IAM > Users > hema > Add permissions

Step 1

Add permissions

Step 2

Review

Review

The following policies will be attached to this user. [Learn more](#)

User details

User name  
hema

Permissions summary (2)

Name	Type	Used as
<a href="#">AdministratorAccess</a>	AWS managed - job function	Permissions policy
<a href="#">IAMUserChangePassword</a>	AWS managed	Permissions policy

Cancel

Previous

Add permissions

CloudShellFeedback

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

The top screenshot shows the 'Add permissions' page for a user named 'hema'. A modal titled 'Create user group' is open. It prompts to 'Create a user group and select policies to attach to the group'. The 'User group name' field contains 'developer'. Under 'Permissions policies (1/1041)', the 'AmazonS3ReadOnlyAccess' policy is selected. The bottom screenshot shows the 'User groups' page with one group listed: 'developer'.

Group name	Users	Permissions	Creation time
<a href="#">developer</a>	0	Defined	Now

## 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**.

The image shows two screenshots of the AWS Management Console. The top screenshot displays the 'Instances' page for the 'us-east-1' region. A table lists instances, with 'My Instance' (ID: i-098cd2410788d1a39) in a 'Running' state. The 'Actions' menu is open, showing options like 'Connect', 'View details', and 'Security'. The bottom screenshot shows the 'Modify IAM role' page for the same instance. It displays the current IAM role as 'EC2FullAccessRole' and provides a 'Create new IAM role' link. The footer of both screenshots includes the AWS logo, a search bar, and navigation links for 'CloudShell' and 'Feedback'.

**Instances (1/1)** Info Last updated 1 minute ago

Name	Instance ID	Instance state	Instance type
My Instance	i-098cd2410788d1a39	Running	t2.micro

**i-098cd2410788d1a39 (My Instance)**

**Details** Status and alarms Monitoring Security Networking Storage Tags

**Instance summary** Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-098cd2410788d1a39	18.232.125.63   open address	172.31.13.24

**Modify IAM role** Info

Attach an IAM role to your instance.

Instance ID: i-098cd2410788d1a39 (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 | Create new IAM role

Cancel 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
```

```
C:\Windows\System32>aws s3 ls
2025-03-11 22:12:35 my-bucket-poc-21
2025-03-11 22:15:39 my-unique-bucket-123456789xy

C:\Windows\System32>aws s3 mb s3://my-unique-bucket-123456789xy
make_bucket: my-unique-bucket-123456789xy
```

## 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. It should **deny access**.

```
C:\Windows\System32>aws ec2 run-instances --image-id ami-08b5b3a93ed654d19 --instance-type t2.micro
An error occurred (MissingInput) when calling the RunInstances operation: No subnets found for the default
```

# 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.