



## **Placement Empowerment Program**

### ***Cloud Computing and DevOps Centre***

**Task:**

**Set Up IAM Roles and Permissions** Create an IAM role on your cloud platform. Assign the role to your VM to restrict/allow specific actions.

**Name: Vishnu K**

**Department: CSE**



# Introduction

## Introduction: Setting Up IAM Roles and Permissions for Cloud VMs

In modern cloud computing environments, security and access control play a crucial role in managing resources effectively. Identity and Access Management (IAM) enables administrators to define granular permissions, ensuring that users and services have only the necessary access required to perform specific tasks.

This Proof of Concept (PoC) focuses on creating and assigning IAM roles to a Virtual Machine (VM) on a cloud platform (such as AWS, Google Cloud, or Azure). By doing so, we can restrict or allow certain actions for the VM, enhancing security while maintaining operational efficiency.

In this guide, we will:

1. Create an IAM Role with specific permissions.
2. Assign the IAM Role to a cloud VM instance.
3. Verify the Role's Effectiveness by testing access to cloud services.

By implementing IAM roles, organizations can enforce the principle of least privilege, ensuring that VMs only have the necessary permissions required for their tasks, reducing the risk of unauthorized access or data exposure.

Let me know if you need further refinements! 🚀

## Overview

This Proof of Concept (PoC) demonstrates how to create and assign IAM roles to a cloud VM to control access and permissions securely. By implementing IAM roles, we ensure the VM has the least privilege required to interact with cloud services. The process includes creating a role, attaching policies, assigning it to the VM, and verifying access. This enhances security by restricting unauthorized actions while enabling necessary operations.



## Key Steps Involved:

- Step 1: Create an IAM Role
- Step 2: Attach Policies to the IAM Role
- Step 3: Assign the IAM Role to the VM
- Step 4: Verify Role Permissions

# Objectives

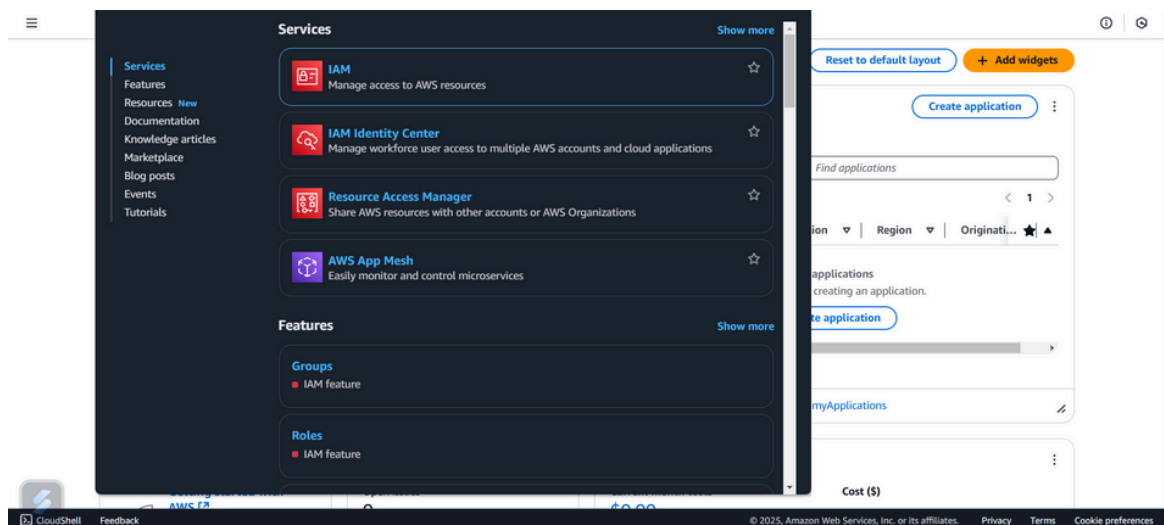
- Implement Secure Access Control – Establish IAM roles to manage and restrict access to cloud resources, ensuring only authorized actions are allowed.
- Enforce Least Privilege Principle – Grant the VM only the necessary permissions required for its tasks, reducing security risks.
- Streamline Role-Based Access Management – Simplify permission management by using IAM roles instead of hardcoded credentials within the VM.
- Enhance Cloud Security and Compliance – Prevent unauthorized access and maintain compliance with security best practices by using IAM policies.
- Validate Role Functionality – Test and verify that the assigned IAM role allows or restricts access as intended by executing relevant commands from the VM.

## Step-by-Step Overview

### Step 1:

#### Navigate to the AWS Management Console

In a Management Console open IAM in the left side navigate to the roles



Step 2: Add permissions

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.
- ☐ **EC2 - Spot Fleet Auto Scaling**  
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.
- ☐ **EC2 - Spot Fleet Tagging**  
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.
- ☐ **EC2 - Spot Instances**  
Allows EC2 Spot instances to launch and manage spot instances on your behalf.

## Step 2:

Navigate to the roles enter the role name and give access to the EC2 and create the role

Step 3: Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

EC2Access

Description

Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Step 1: Select trusted entities

Trust policy

```

1- {
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Effect": "Allow",
6-       "Action": [
7-         "sts:AssumeRole"
8-       ],
9-       "Principal": {
10-        "Service": [
11-          "ec2.amazonaws.com"
12-        ]
13-      }
14-    ]
15-  }

```

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings
- Root access management

Access reports

- Access Analyzer
- External access
- Unused access
- Analyzer settings
- Credential report
- Organization activity

Role EC2Access created.

Roles (5)

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
<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor (Service)	-
<a href="#">EC2Access</a>	AWS Service: ec2	-
<a href="#">shortdemo-role-r1nfdn20</a>	AWS Service: lambda	-

Roles Anywhere

Authenticate your non AWS workloads and securely provide access to AWS services.

Access AWS from your non AWS workloads

Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.

X.509 Standard

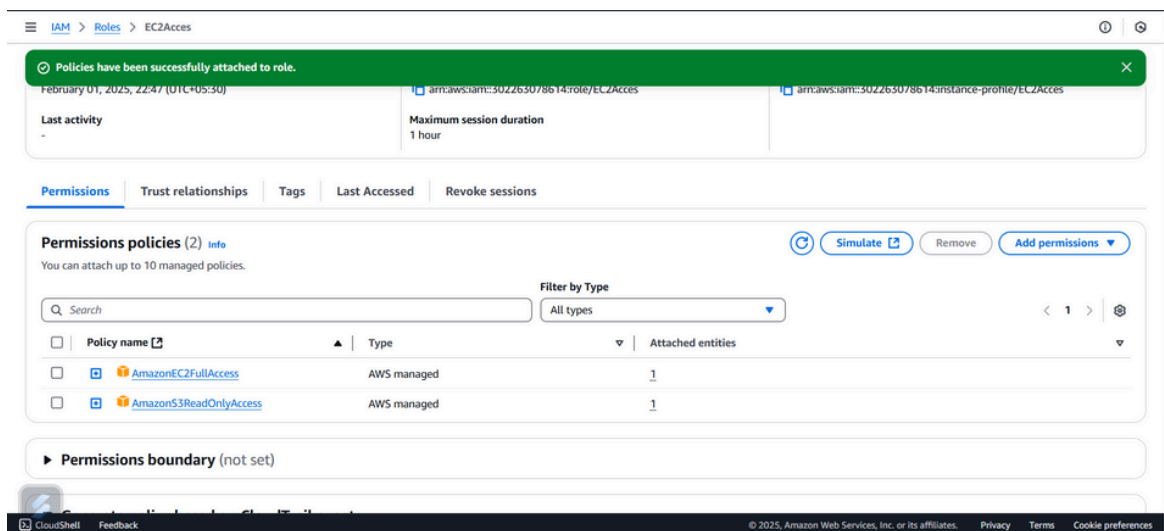
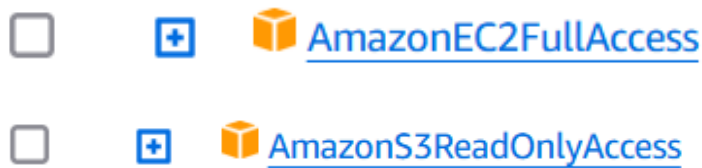
Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities.

Temporary credentials

Use temporary credentials with ease and benefit from the enhanced security they provide.

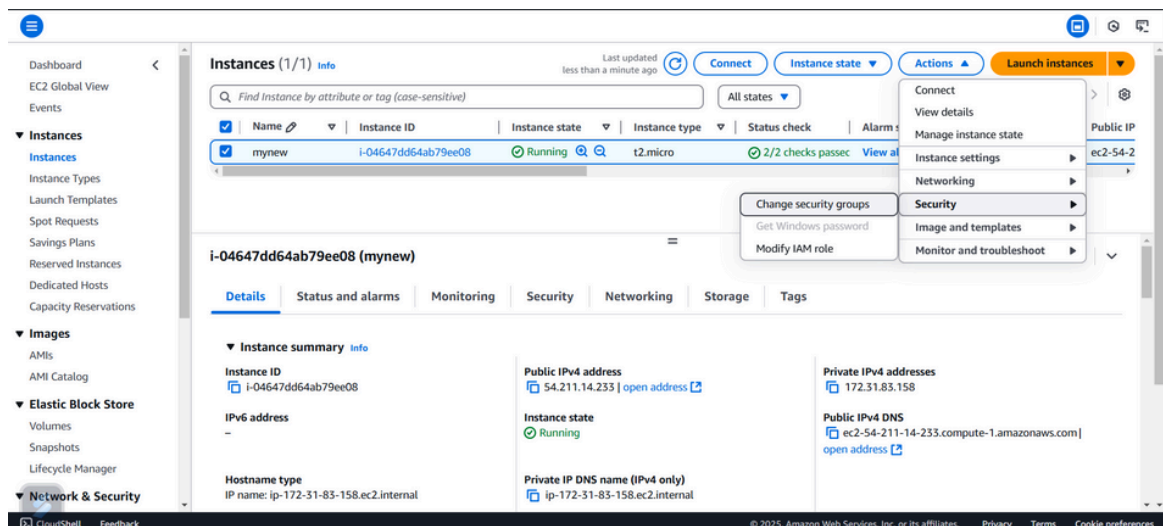
## Step 3:

Open the created role and give access the given policy



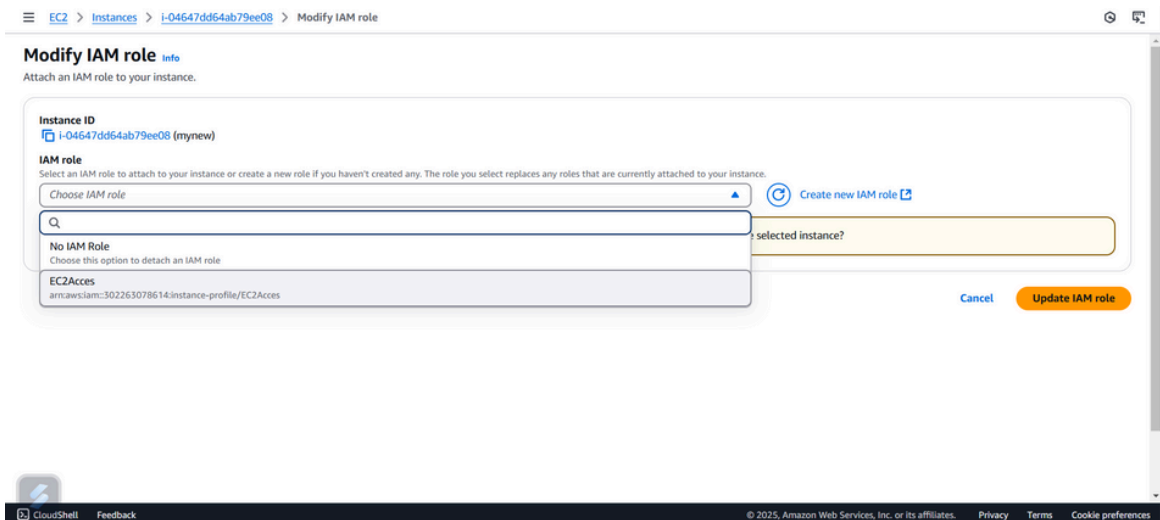
## Step 4:

In this step first simply launch an ec2 instance and click the tick mark on the created instances. Then click on the action open security the click modify IAM role

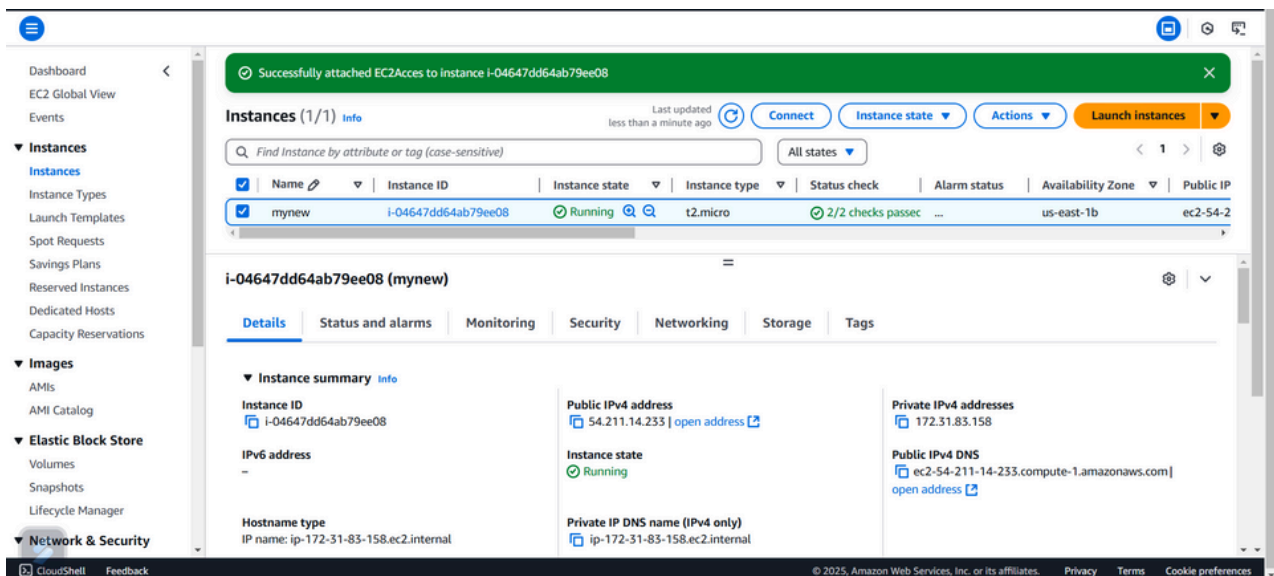


# Step 5:

Then finally choose the IAM role which have been created so far.



Then click on the update IAM role



# Expected Outcomes:

## Expected Outcomes for This Proof of Concept (PoC)

1. **Successful IAM Role Creation** – An IAM role with the necessary permissions is created and available for assignment.
2. **Role Assigned to the VM** – The IAM role is successfully attached to the cloud VM instance without using static credentials.
3. **Controlled Access to Cloud Services** – The VM is able to perform only the permitted actions (e.g., accessing specific cloud storage, databases, or APIs) based on assigned policies.
4. **Prevention of Unauthorized Actions** – Attempts to perform actions outside the defined role permissions result in access denied errors, confirming security enforcement.
5. **Improved Security & Compliance** – The setup adheres to the principle of least privilege, reducing risks of unauthorized access and security breaches.
6. **Verification and Testing Success** – Running test commands from the VM confirms that access permissions function as expected.

By achieving these outcomes, the PoC ensures secure, role-based access control for cloud VMs, enhancing security and operational efficiency. 🚀