



Placement Empowerment Program

Cloud Computing and DevOps Centre

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: Jason Peniel Raj. S

Department: CSE

Introduction

This Proof of Concept (PoC) demonstrates the process of setting up and utilizing IAM roles and permissions in AWS. The goal is to show how to secure AWS resources by managing access through roles rather than hardcoding credentials. Specifically, this PoC focuses on creating an IAM role, assigning it to an EC2 instance, and verifying the instance's access to AWS services such as Amazon S3.

Overview

The process is divided into several key steps:

- 1. Create an IAM Role:** Define a role in AWS IAM and attach policies that grant permissions for specific AWS services.
- 2. Launch an EC2 Instance:** Create a virtual machine (VM) in AWS and configure it for testing the assigned IAM role.
- 3. Assign the IAM Role to the EC2 Instance:** Attach the created IAM role to the EC2 instance to enable access to AWS services without using access keys.
- 4. Verify Access:** Test the EC2 instance to confirm that it has the appropriate permissions by interacting with services like Amazon S3.

Objectives

This PoC aims to achieve the following objectives:

1. **Secure Access:** Implement IAM roles to grant temporary permissions to AWS resources without embedding credentials.
2. **Demonstrate Role-Based Permissions:** Show how roles can restrict or allow actions based on attached policies.
3. **Test Least Privilege Principle:** Ensure that the EC2 instance only has the permissions it needs to perform specific tasks.
4. **Hands-On Learning:** Provide practical experience with IAM roles and their applications in a cloud environment.

Importance

IAM roles and permissions are fundamental to securing cloud environments. They allow for fine-grained access control and improve operational efficiency by:

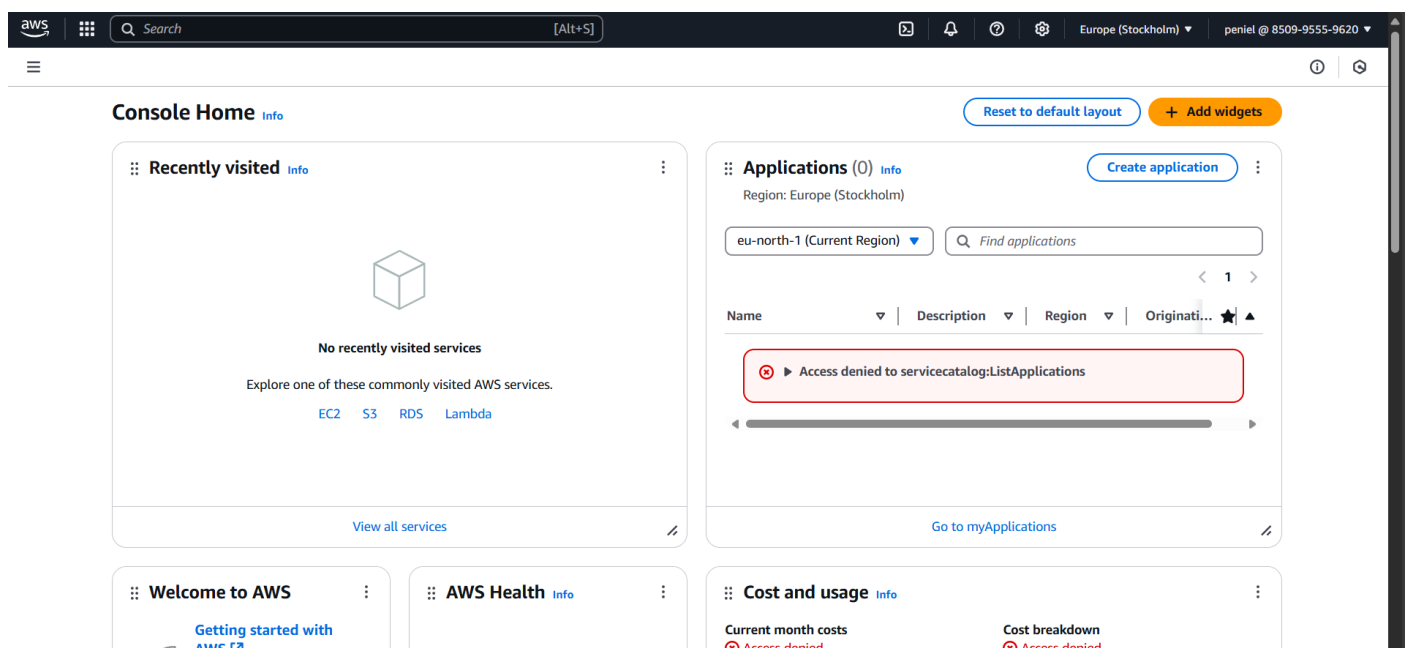
1. **Eliminating Hardcoded Credentials:** Reducing security risks by avoiding the storage of access keys in applications or instances.
2. **Granting Least Privilege Access:** Ensuring users and resources only have the permissions they require, minimizing potential misuse.
3. **Improving Compliance:** Enforcing organizational policies and audit requirements.

4. **Enhancing Automation:** Allowing resources like EC2 instances to securely interact with other AWS services.

Step-by-Step Overview Step

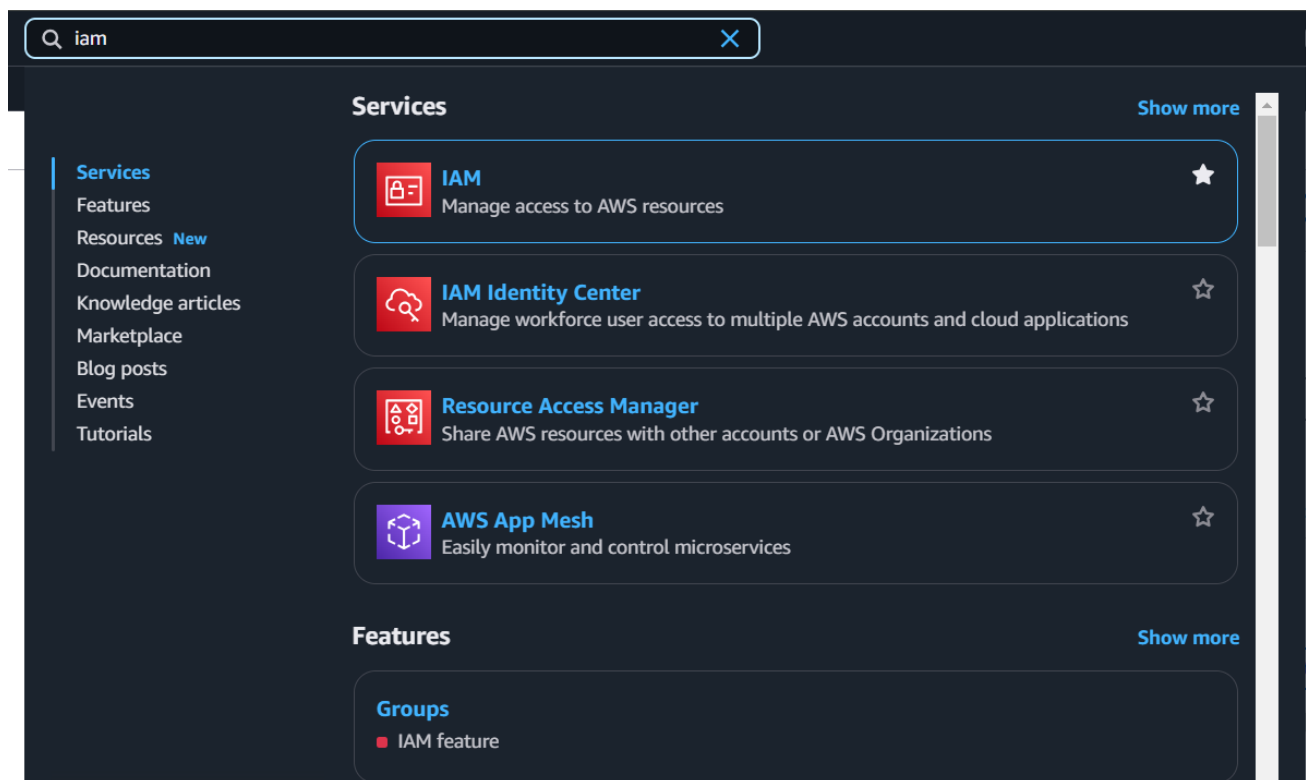
1:

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



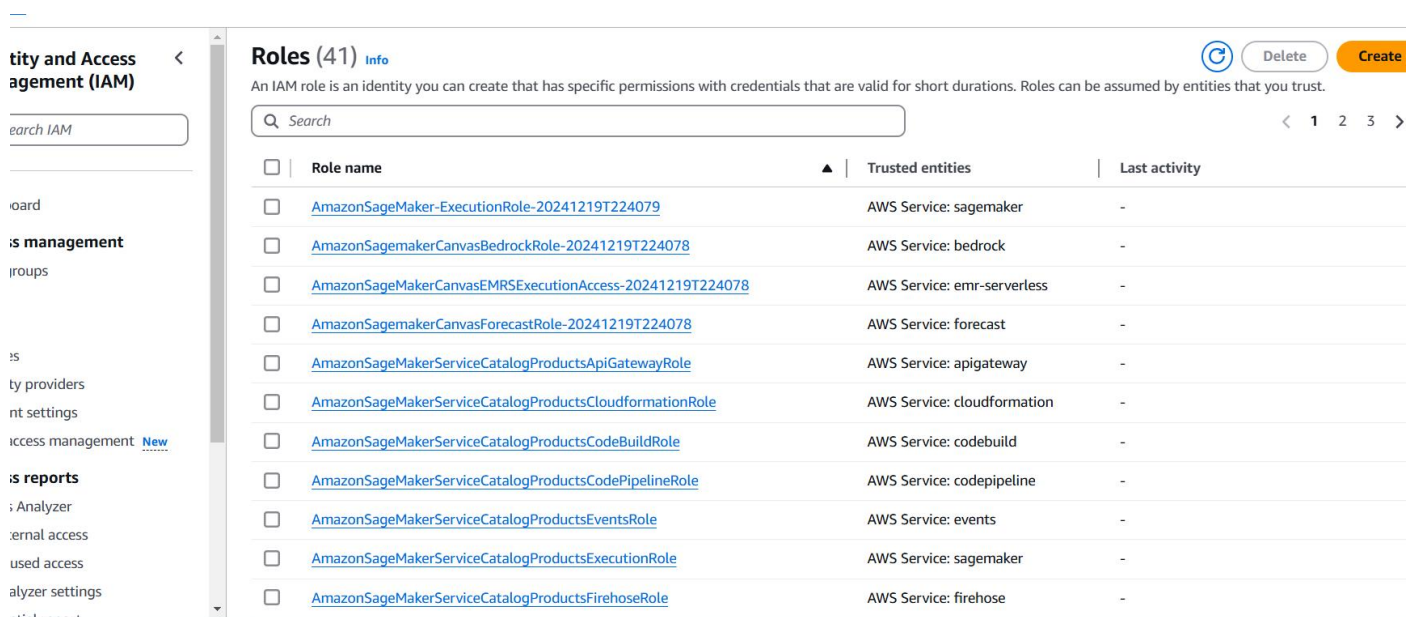
Step 2:

1. In the AWS Management Console, type "IAM" in the search bar at the top.
2. Click on **IAM** from the search results.



Step 3:

1. On the IAM dashboard, click on **"Roles"** in the left-hand menu.
2. On the Roles page, click the **"Create Role"** button.



Step 4:

1. On the **"Create Role"** page, under **Trusted Entity Type**, select **AWS Service** (it should be selected by default).
2. In the **Use Case** dropdown, choose **EC2**.

Click **Next** to continue

The screenshot shows the 'Create Role' page in the AWS IAM console, specifically Step 4. On the left, a progress bar indicates the current step: 'Step 4: Name, review, and create'. The main content area is titled 'Trusted entity type' and contains five radio button options: 'AWS service' (selected), 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. Below this, the 'Use case' section is visible, which includes a dropdown menu labeled 'Service or use case' with 'EC2' selected. The page is designed with a clean, modern aesthetic using light gray borders and blue accents for the selected options.

Step 5:

1. On the **Permissions** page, you'll see a list of policies.
2. Select a policy based on what actions you want the VM to perform. For example:

To give the VM **read-only access to S3**, select **AmazonS3ReadOnlyAccess**.

You can search for policies in the search bar (e.g., type "S3" for S3 policies).


3. Once you've selected a policy, click **Next**.

```
4 {  
5   "Effect": "Allow",  
6   "Action": [  
7     "sts:AssumeRole"  
8   ],  
9   "Principal": {  
10    "Service": [  
11      "ec2.amazonaws.com"  
12    ]  
13  }  
14 }  
15 ]  
16 }
```

Step 2: Add permissions

[Edit](#)

Permissions policy summary

Policy name 	Type	Attached as
AmazonS3ReadOnlyAccess	AWS managed	Permissions policy

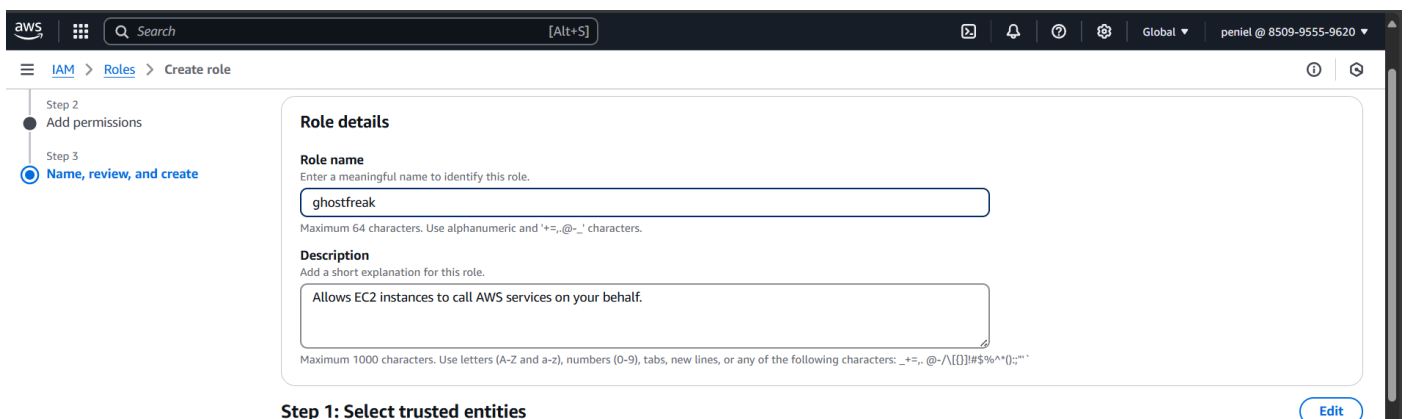
Step 3: Add tags

Step 6:

1. On the **Role Details** page:

- Enter a name for your role (e.g., My-EC2-S3-Access-Role).
- (Optional) Add a description or tags if you'd like.

2. Click **Create Role** to finish.



aws | Search [Alt+S] | Global | peniel @ 8509-9555-9620

IAM > Roles > Create role

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

Role details

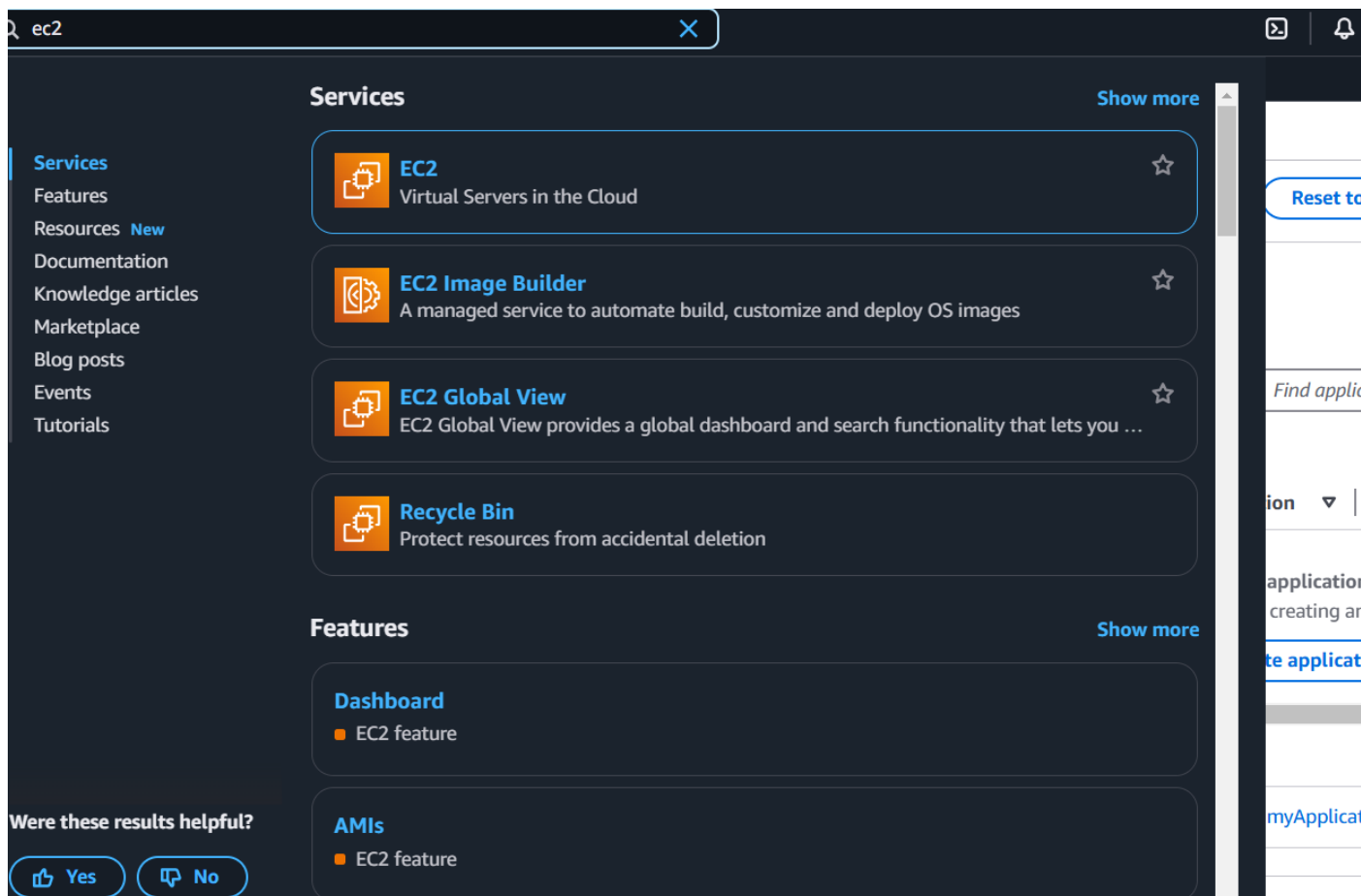
Role name
Enter a meaningful name to identify this role.
ghostfreak
Maximum 64 characters. Use alphanumeric and "+=, @-_" characters.

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 [Edit](#)

Step 7:

1. In the AWS Management Console, search for **EC2** and click to open the **EC2 Dashboard**.
2. Select the instance (VM) you want to assign the IAM role to.



Step 8:

1. In the **Instance details** section, click **Actions** in the top right corner.
2. From the dropdown, choose **Security** > **Modify IAM Role**.

Instances (1/1) Info

Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm state	Public IPv4 address
benny04	i-088d3877d83c11418	Running	t3.micro	Initializing	View alarms	13.60.193.221

i-088d3877d83c11418 (benny04)

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

▼ Instance summary Info

Instance ID i-088d3877d83c11418	Public IPv4 address 13.60.193.221 open address	Private IPv4 addresses 172.31.39.57
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-13-60-193-221.eu-west-1.compute.amazonaws.com

Step 9:

1. In the **Modify IAM role** window, you should see a dropdown for **IAM role**.
2. Select the role you created earlier (e.g., My-EC2-S3-AccessRole).
3. Click **Update IAM role** to apply the changes.

EC2 > Instances > i-0817b5628cce06248 > Modify IAM role

Modify IAM role Info

Attach an IAM role to your instance.

Instance ID
i-0817b5628cce06248 (jenz03)

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.

dasan12

Create new IAM role

Cancel Update IAM role

Step 10:

1. Open your terminal (if you're using Linux or macOS) or Command Prompt (Windows).
2. Use SSH to log in to your EC2 instance. For example:

```
ssh -i "your-key-pair.pem" ec2-user@your-ec2-public-ip
```

Step 11:

```
[ec2-user@ip-172-31-80-54 ~]$ aws ec2 describe-regions --query "Regions[*].RegionName"
```

The error confirms that your IAM role (My-EC2-S3-Access-Role) does not have permissions to perform the **ec2:DescribeRegions** action. The role currently only has S3-related permissions (e.g., AmazonS3ReadOnlyAccess) and doesn't include broader EC2 permissions.

[illegible]

Outcome

By completing this PoC of setting up IAM roles and permissions with an EC2 instance, you will:

1. Create an IAM role and attach policies to control access to specific AWS services.
2. Launch and configure an EC2 instance for testing purposes.
3. Assign the IAM role to the EC2 instance securely without using access keys.
4. Verify permissions by interacting with AWS services (e.g., listing S3 buckets) from the EC2 instance.
5. Demonstrate the principle of least privilege by ensuring only necessary permissions are granted.