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: Krishna bhattad J
Department: IT

Introduction and Overview

IAM (Identity and Access Management) roles in AWS provide secure and controlled access to AWS services without requiring long-term credentials. By assigning IAM roles to EC2 instances, applications can interact with services like S3 based on predefined permissions. This ensures better security, streamlined access management, and adherence to the **Principle of Least Privilege**. Properly configuring IAM roles helps prevent unauthorized actions while enabling seamless resource access within AWS environments.

Objective

The goal of this project is to:

1. **Secure Access Management**: Ensure AWS resources (e.g., EC2 instances) can securely interact with other AWS services (e.g., S3) without using hardcoded credentials.
2. **Controlled Resource Access**: Assign specific permissions to limit what actions an instance or user can perform, preventing unauthorized access or modifications.
3. **Improve Compliance and Security**: Enforce best practices like the **Principle of Least Privilege**, reducing security vulnerabilities and meeting compliance requirements.

Importance

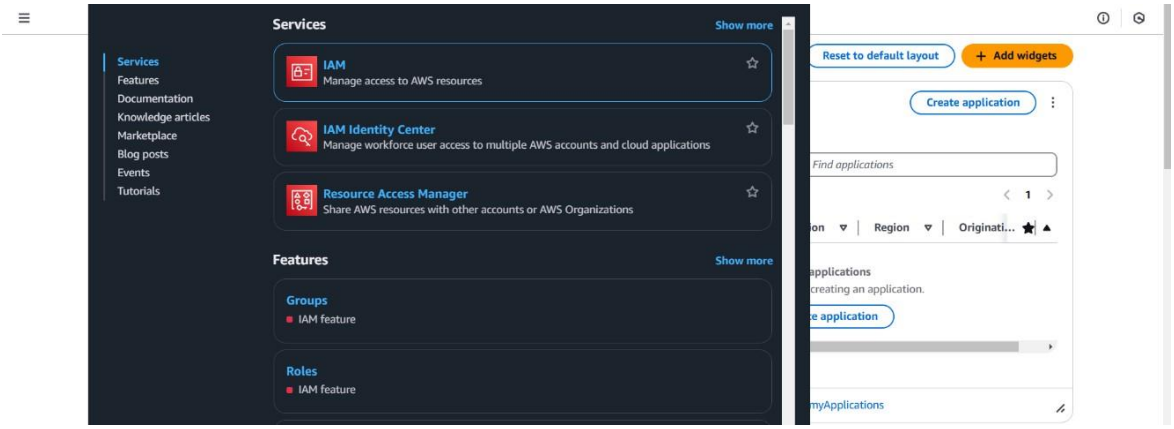
1. **Enhanced Security**: IAM roles provide controlled access to AWS services without exposing long-term credentials, reducing security risks.

2. **Granular Access Control**: Permissions can be precisely defined to ensure that instances and applications only access the resources they need, following the **Principle of Least Privilege**.
3. **Simplified Credential Management**: Since IAM roles use temporary credentials, there's no need to manually manage or rotate access keys, improving security and reducing operational overhead.

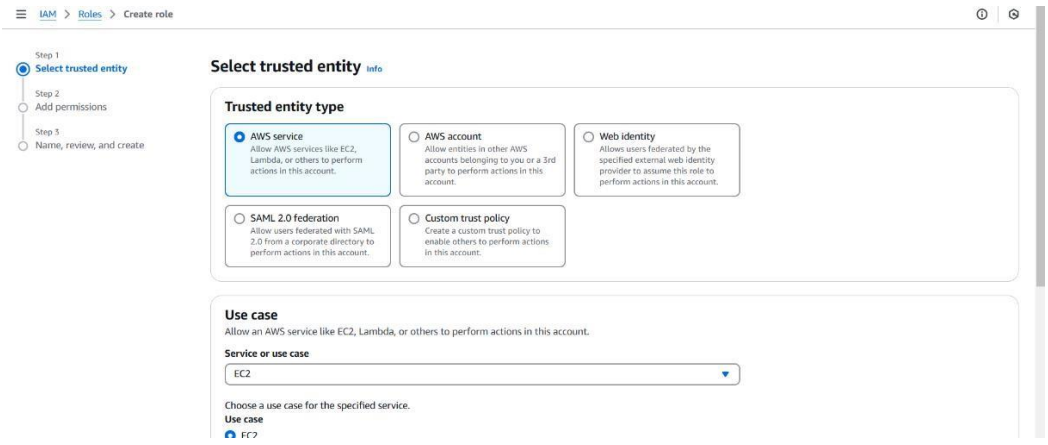
Step-by-Step Overview

Step1: **Create an IAM Role**

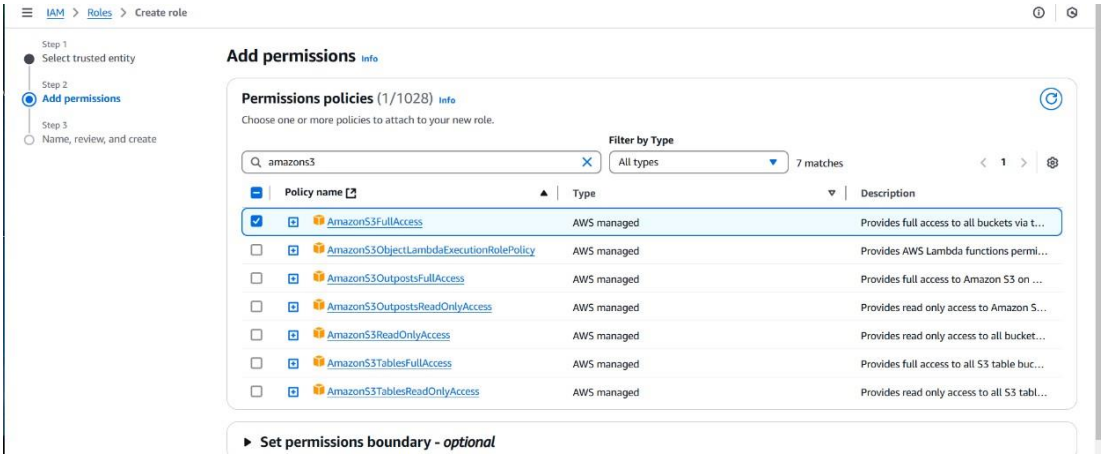
1. **Go to IAM Console** ◦ Sign in to your console, Navigate to **IAM (Identity & Access Management)**.



2. **Create a New Role** ◦ In the IAM console, click **Roles** on the left sidebar.
   ◦ Click **Create role**. ◦ Choose **AWS service** as the trusted entity. ◦ Select **EC2** (since we are assigning the role to a VM). ◦ Click **Next**.



3. **Attach Permissions** ◦ Search for **AmazonS3FullAccess** (if you want full access to S3) or **AmazonS3ReadOnlyAccess** (for read-only). ◦ Select the policy and click **Next**.

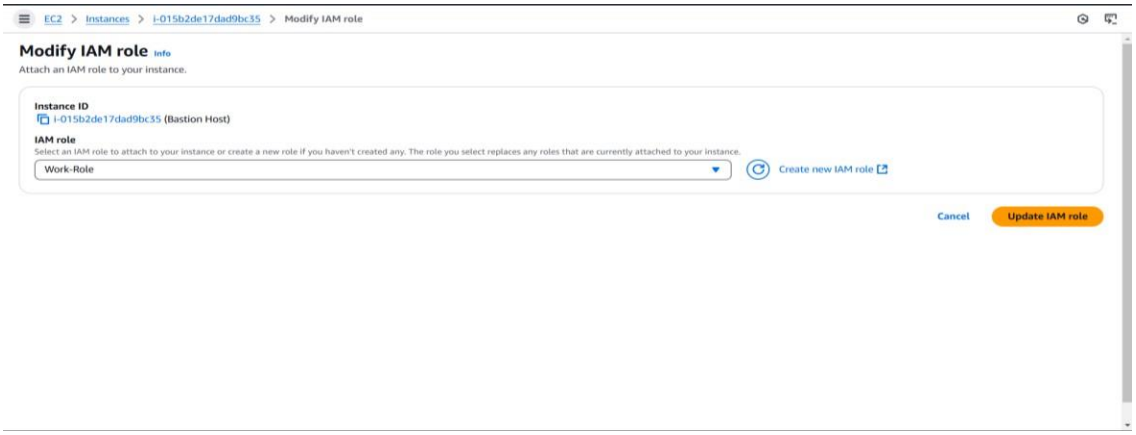


4. **Name & Create the Role** ◦ Enter a role name (e.g., `EC2-S3-Access-Role`).
    - Review and click **Create role**.

Step 2: **Attach the Role to Your EC2 Instance**

1. **Go to EC2 Console** ◦ Open the **EC2 dashboard**.

2. **Modify IAM Role for the Instance** ◦ Select your **EC2 instance**. ◦ Click **Actions → Security → Modify IAM Role**. ◦ Choose the IAM role you just created (`EC2-S3-Access-Role`). ◦ Click **Update IAM Role**.



Step 3:Connect to Your Instance via SSH

1. **Find the Public IP**:
   - Go back to the EC2 Dashboard. ◦ In the **Instances** section, find the instance you just launched.
   - Note the **Public IPv4 address** of your instance. You'll need this for SSH.
2. **Set Permissions for the Key**: Before you can use the `.pem` key file to connect, you need to set the correct permissions:

3. **SSH into the Instance**:
   - Open your terminal (or use an SSH client like PuTTY if you're on Windows). ◦ Run the following command to connect to your EC2 instance via SSH. Replace `<your-key.pem>` with the path to your downloaded key, and `<Public_IP>` with the public IP address you noted earlier.

   For Windows (using PuTTY): ◦ Convert the `.pem` file to `.ppk` format using PuTTYgen. ◦ In PuTTY, enter the IP address and select your private key file under **Connection > SSH > Auth**. ◦ Click **Open** to initiate the connection.
4. **Accept the SSH Key Fingerprint**:
   - The first time you connect, you'll be asked to confirm the host's authenticity. Type `yes` to continue.

   You should now be logged into your EC2 instance!

Step 4: Verify IAM Role Permissions

1. Connect to the EC2 Instance

- Use **SSH** to log into the instance:

```
ssh -i your-key.pem ec2-user@your-instance-publicip
```

2. Check IAM Role Attached

- Run the following command to verify the IAM role:

  ```
  curl
  http://169.254.169.254/latest/metadata/iam/securit
  y-credentials/
  ```

3. Try Accessing S3

- Run the following command to list S3 buckets:

  ```
  aws s3 ls
  ```

  - If your IAM role has **AmazonS3FullAccess**, it should list all S3 buckets.
  - If it has **AmazonS3ReadOnlyAccess**, you can view but not modify buckets.
- Try creating a new bucket (only works if the role has **write permissions**):

  ```
  aws s3 mb s3://your-new-bucket-name
  ```

4. Test Denied Actions

- If you **don't have write permissions**, try deleting an S3 bucket:

  ```
  aws s3 rb s3://your-new-bucket-name
  ```

  You should get a **permission denied** error.

Expected Outcome

By completing this POC, you will:

1. **Successful IAM Role Attachment**: Running `curl http://169.254.169.254/latest/metadata/iam/security-credentials/` should return the IAM role name, confirming it is attached to the EC2 instance.
2. **Permitted Actions Work**: If the role has the correct permissions, running `aws s3 ls` should list all accessible S3 buckets, and `aws s3 mb s3://your-new-bucket-name` should create a new bucket (if write access is granted).

3. **Denied Actions Generate Errors**: If the role lacks necessary permissions, attempting unauthorized actions like `aws s3 rb s3://your-newbucket-name` (bucket deletion) will result in an **Access Denied** error.