**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: Mathimalar.P Department : ADS



Introduction and Overview

IAM (Identity and Access Management) roles in AWS allow secure access to AWS resources without using static credentials. By assigning an IAM role to an EC2 instance, services like S3, DynamoDB, and others can be accessed securely.

Objective

.

Set up an **IAM Role** and attach it to an **EC2 instance**.

Allow the EC2 instance to interact with AWS services (e.g., S3) **without manual credentials**.

Verify access by running AWS CLI commands like aws s3 ls..

Importance

 **Enhanced Security** – Eliminates the need for storing access keys on the instance.

 **Fine-Grained Access Control** – Ensures only necessary permissions are granted.

 **Automated Credential Management** – AWS handles temporary credentials, improving security and ease of use.

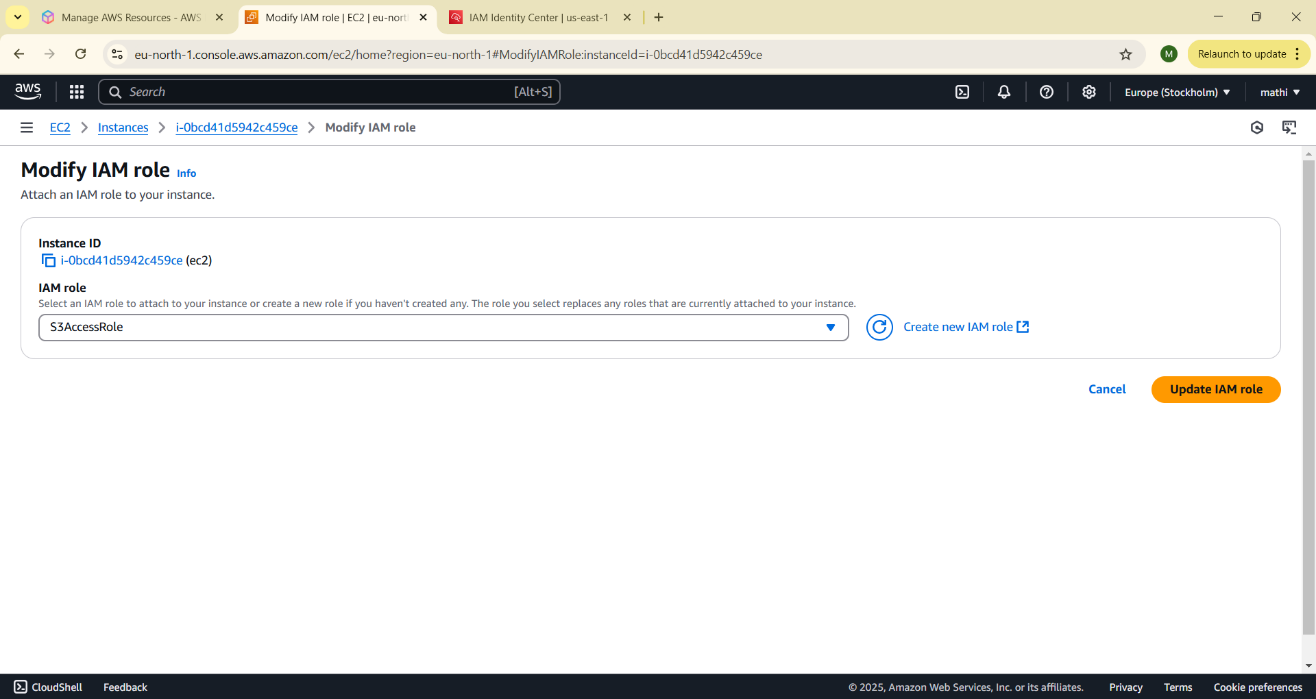
 **Scalability** – IAM roles can be assigned to multiple instances, streamlining permission management.

**Step-by-Step Overview**

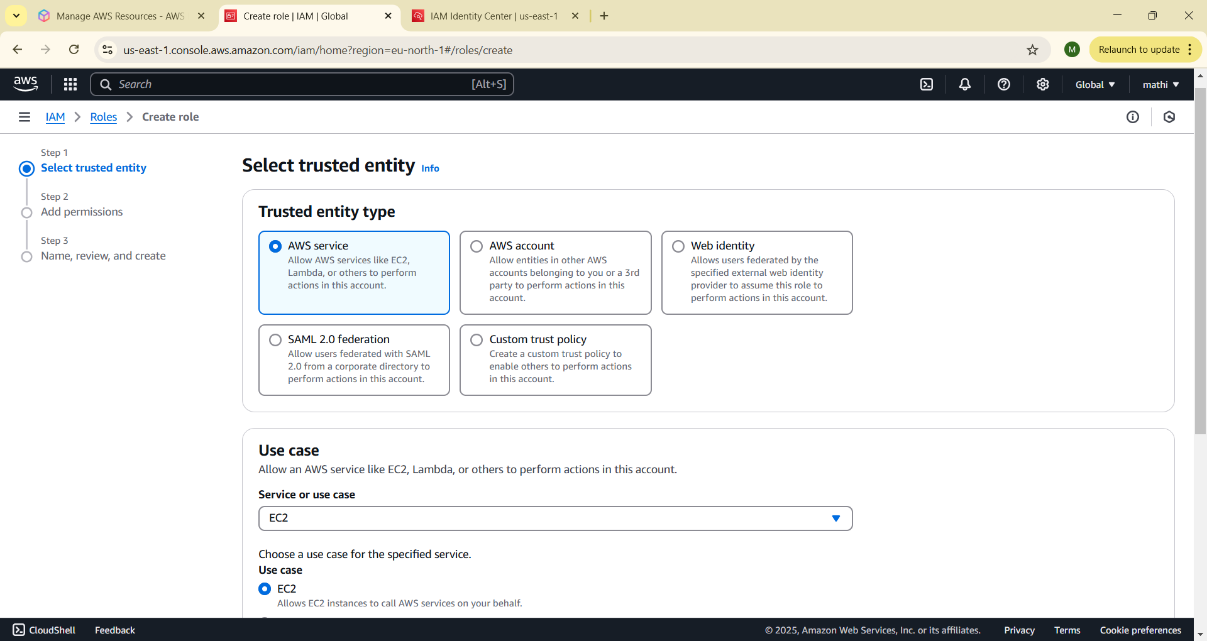
**Step by step overview**

**Step 1: Create an IAM Role**

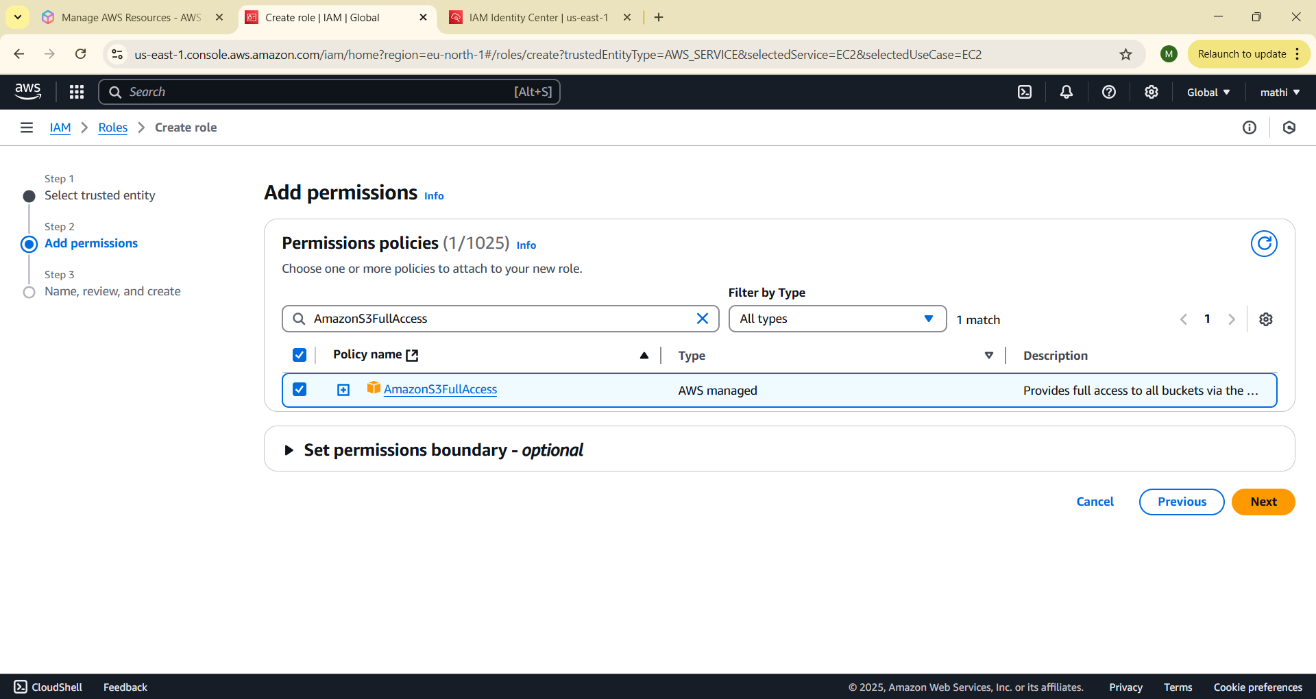
**1. Open the AWS IAM Console**

* Go to the AWS [**IAM Console**](https://console.aws.amazon.com/iam/).
* In the left sidebar, click on **Roles**.
* Click **Create role**.
* 

**2. Select a Trusted Entity**

* Choose **AWS service**.
* Under **Use case**, select **EC2** (for an EC2 instance).
* Click **Next**.
* 

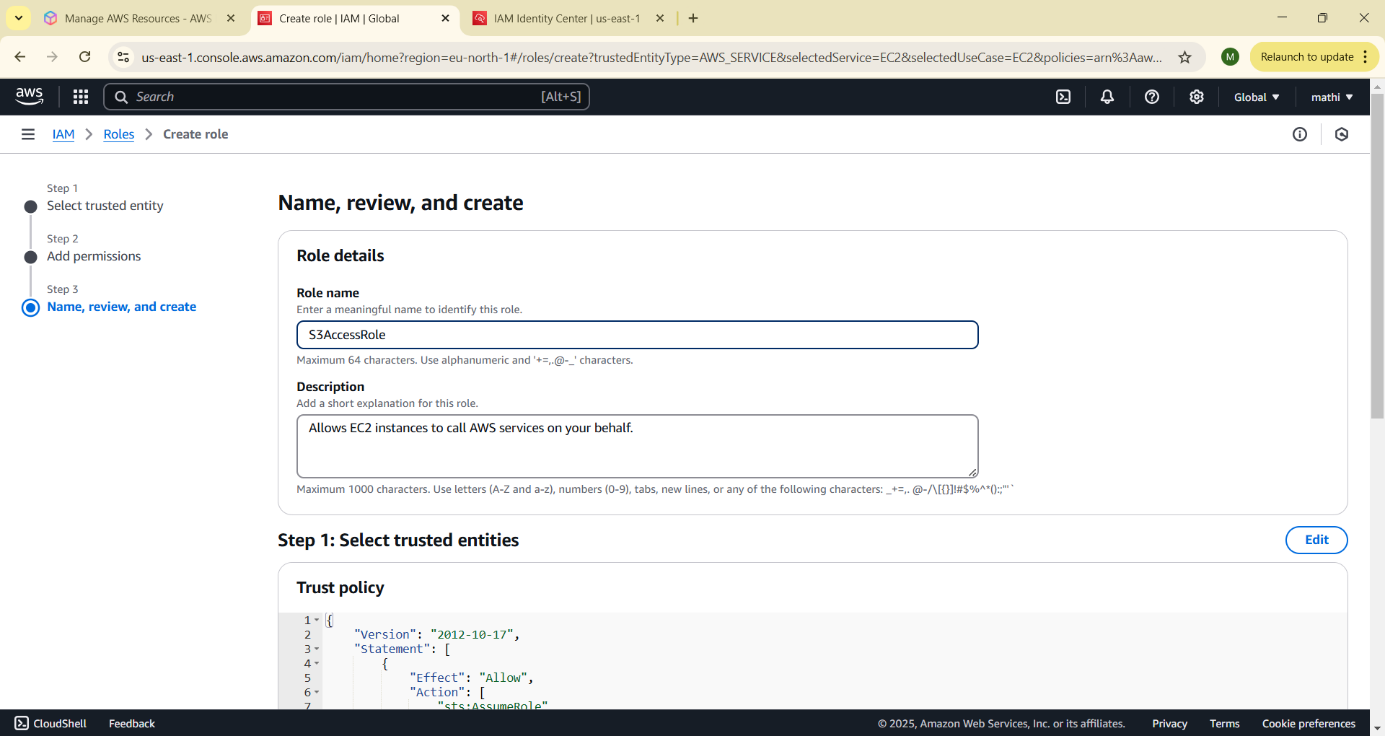
**3. Attach Permissions**

* Choose the required permissions policy:
  + If you need S3 access, search for **AmazonS3FullAccess** and select it.
  + If you need only read access, select **AmazonS3ReadOnlyAccess**.
* Click **Next**.
* 

**4. Add Tags (Optional)**

* You can add tags to help identify your role (e.g., Project: TestRole).
* Click **Next**.

**5. Name and Create the Role**

* Enter a **Role name** (e.g., S3AccessRole).
* Click **Create role**.
* 

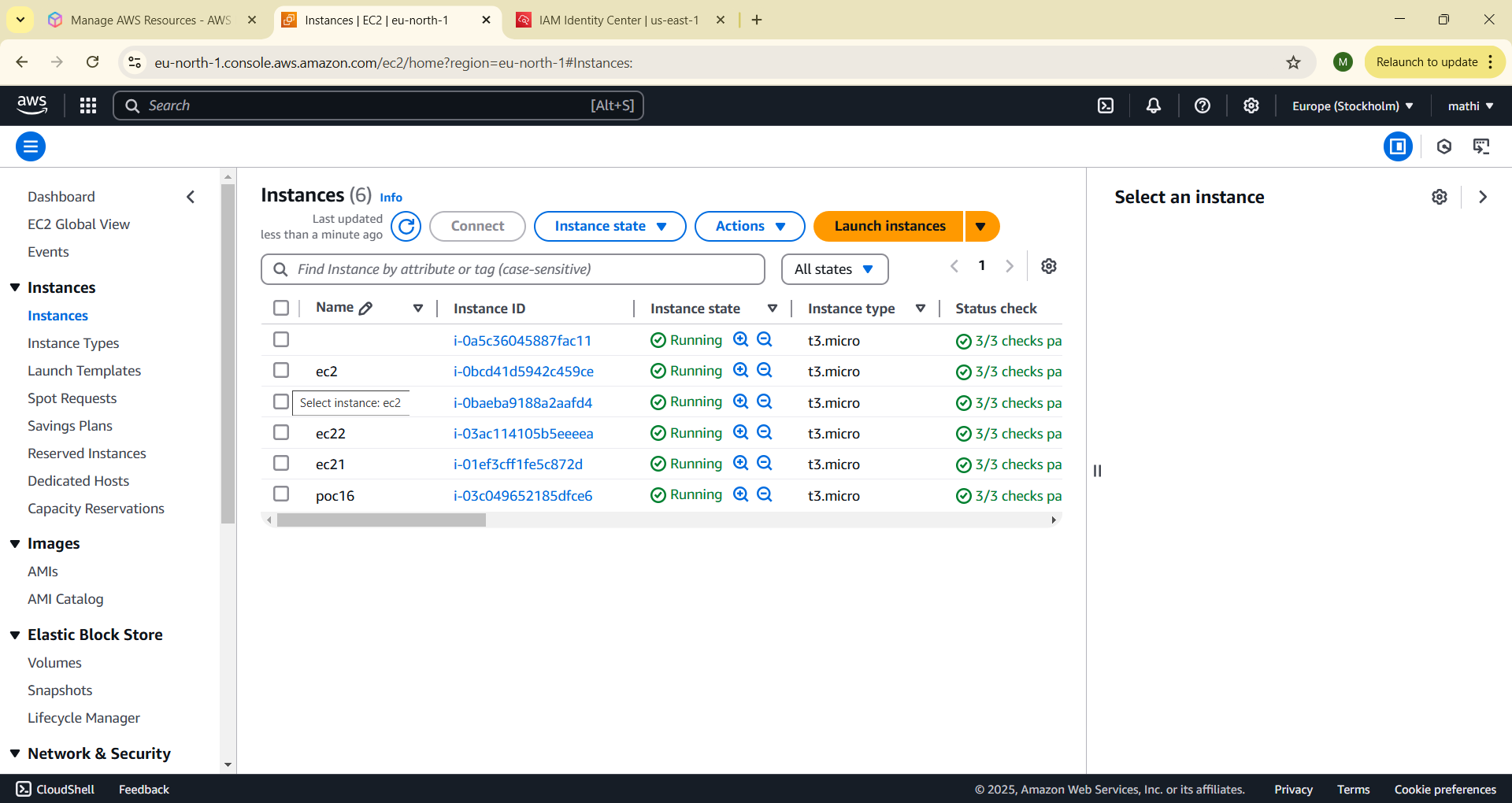
**Step 2: Attach the IAM Role to an EC2 Instance**

**1. Go to the EC2 Console**

* Open the [**EC2 Console**](https://console.aws.amazon.com/ec2/).
* In the left sidebar, click on **Instances**.
* Find and **select your EC2 instance**.

**2. Modify the IAM Role**

* Click on **Actions** → **Security** → **Modify IAM role**.
* In the **IAM Role** dropdown, select the role you created (S3AccessRole).
* Click **Update IAM role**.



**Step 3: Verify the IAM Role is Working**

**1. Connect to Your EC2 Instance**

* Open your EC2 instance via **SSH** (Linux) or **RDP** (Windows).

**2. Run AWS CLI Commands**

* If the AWS CLI is not installed, install it first:

sudo apt update && sudo apt install awscli -y # Ubuntu/Debian

* Check if the instance can access S3 using the assigned IAM role:

aws s3 ls

* If your IAM role has S3 permissions, this should list all your S3 buckets.

**Troubleshooting**

If aws s3 ls gives a **permissions error**, check:

1. That the **IAM role is correctly attached** to the EC2 instance.
2. That the **IAM role has the correct policies** (e.g., AmazonS3FullAccess).
3. That the **AWS CLI is installed** on your EC2 instance.

This setup ensures your EC2 instance has secure access to AWS services **without needing Access Keys**.

**Expected outcomes**

The outcomes of your task (setting up IAM roles and permissions for an EC2 instance) are:

1. **Secure Access** – Your EC2 instance can access AWS services (like S3) without needing manual credentials.
2. **Least Privilege Principle** – The IAM role restricts access to only the required services.
3. **Automatic Credential Management** – AWS handles temporary credentials, reducing security risks.
4. **Successful AWS CLI Execution** – Running commands like aws s3 ls works without configuring access keys.
5. **Easier Maintenance** – No need to manage or rotate long-term credentials manually.