St. JOSEPH'S GROUP OF INSTITUTIONS
OMR, CHENNAI - 119

aws

**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: Esly Abro K Department :IT

St. JOSEPH'S COLLEGE OF ENGINEERING | St. JOSEPH'S INSTITUTE OF TECHNOLOGY
AUTONOMOUS INSTITUTIONS, AFFILIATED TO ANNA UNIVERSITY

# Introduction

In AWS, Identity and Access Management (IAM) allows you to define roles and permissions that control access to your resources. With IAM roles, you can manage who can do what within your AWS account. This document will walk you through creating an IAM role, assigning it to an EC2 instance, and verifying the permissions.

# Objectives

By following this guide, you will:

1. Learn how to create IAM roles and assign permissions.
2. Attach the IAM role to an EC2 instance.
3. Verify that the permissions work as intended by attempting both permitted and denied actions.

---

# Step 1: Create an IAM Role 1.1. Log in to AWS Management Console

- Open your browser and go to the AWS Management Console.
- Log in with your AWS account credentials.

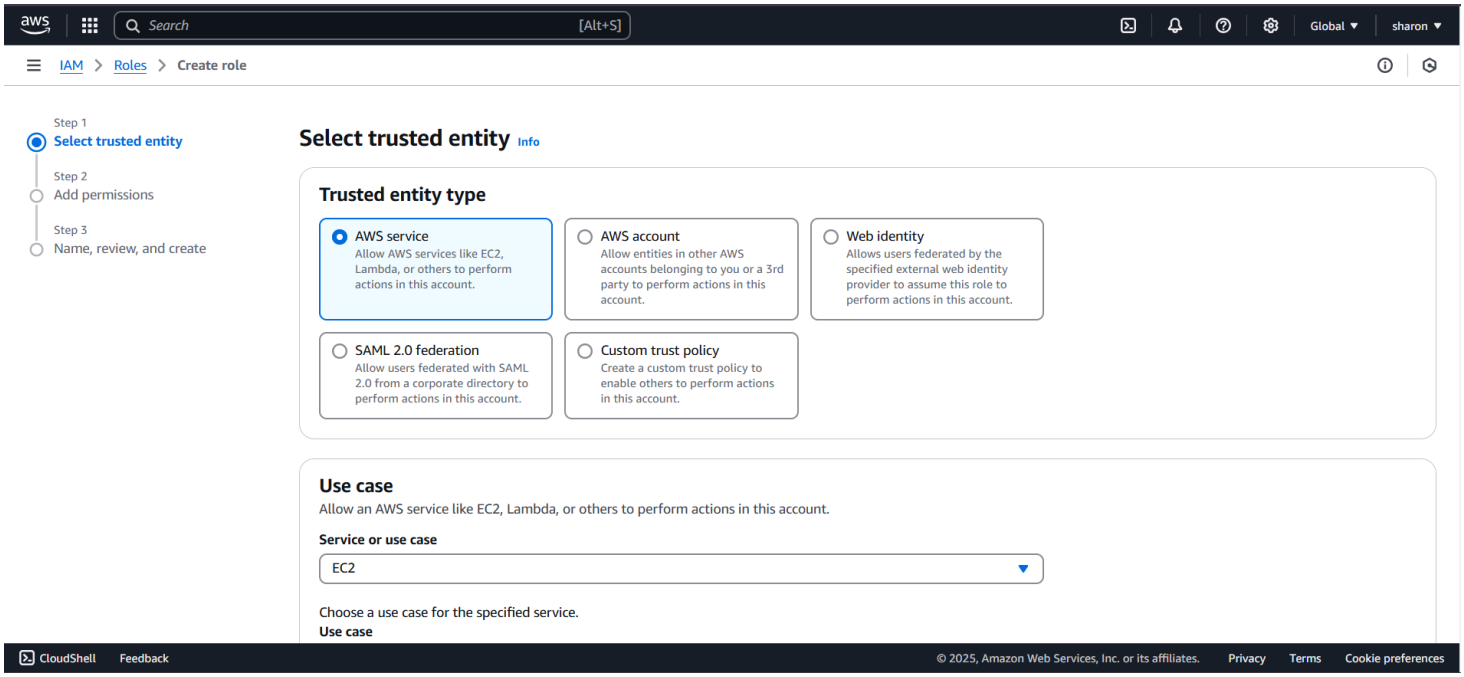
# 1.2. Navigate to IAM

- In the AWS Management Console, search for **IAM** in the search bar and select **IAM**

from the list.

## 1.3. Create a New Role

- In the IAM dashboard, select **Roles** from the left sidebar.
- Click the **Create role** button.




## 1.4. Select Trusted Entity

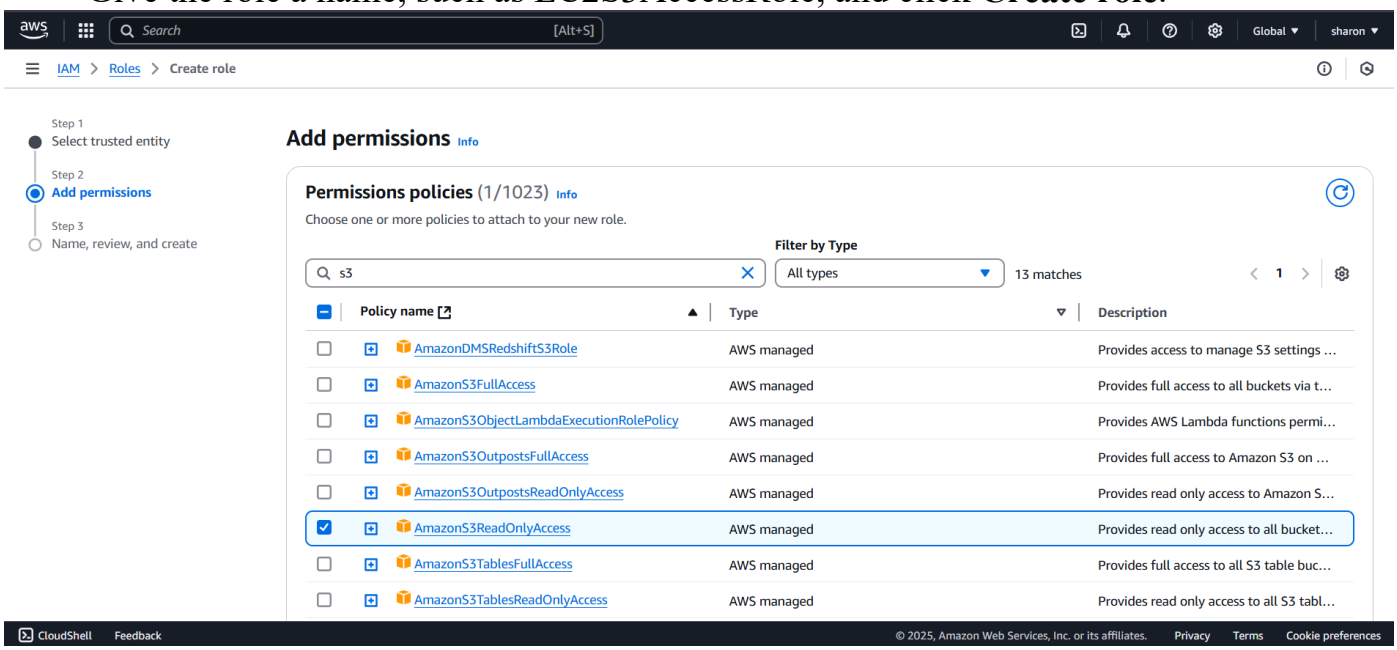
- Choose the **AWS service** option.
- Under **Use case**, select **EC2** (This will allow EC2 instances to assume this role).

## 1.5. Attach Permissions to the Role

- On the permissions page, you'll see a list of policies that you can attach to your role.
- For example, to allow your EC2 instance to access S3, search for AmazonS3FullAccess and check the box next to it.
- You can also search for other permissions you might need (e.g., AmazonEC2ReadOnlyAccess, etc.).

## 1.6. Name the Role

- After selecting the necessary permissions, click **Next: Tags**.
- Optionally, add tags to the role.
- Click **Next: Review**.
- Give the role a name, such as EC2S3AccessRole, and click **Create role**.




---

**Step 2: Attach the IAM Role to an EC2 Instance 2.1. Navigate to EC2**

- From the AWS Management Console, search for **EC2** and select **EC2** from the list.

## 2.2. Select Your EC2 Instance

- In the EC2 dashboard, click **Instances** in the left sidebar.
- Select the EC2 instance to which you want to attach the IAM role.

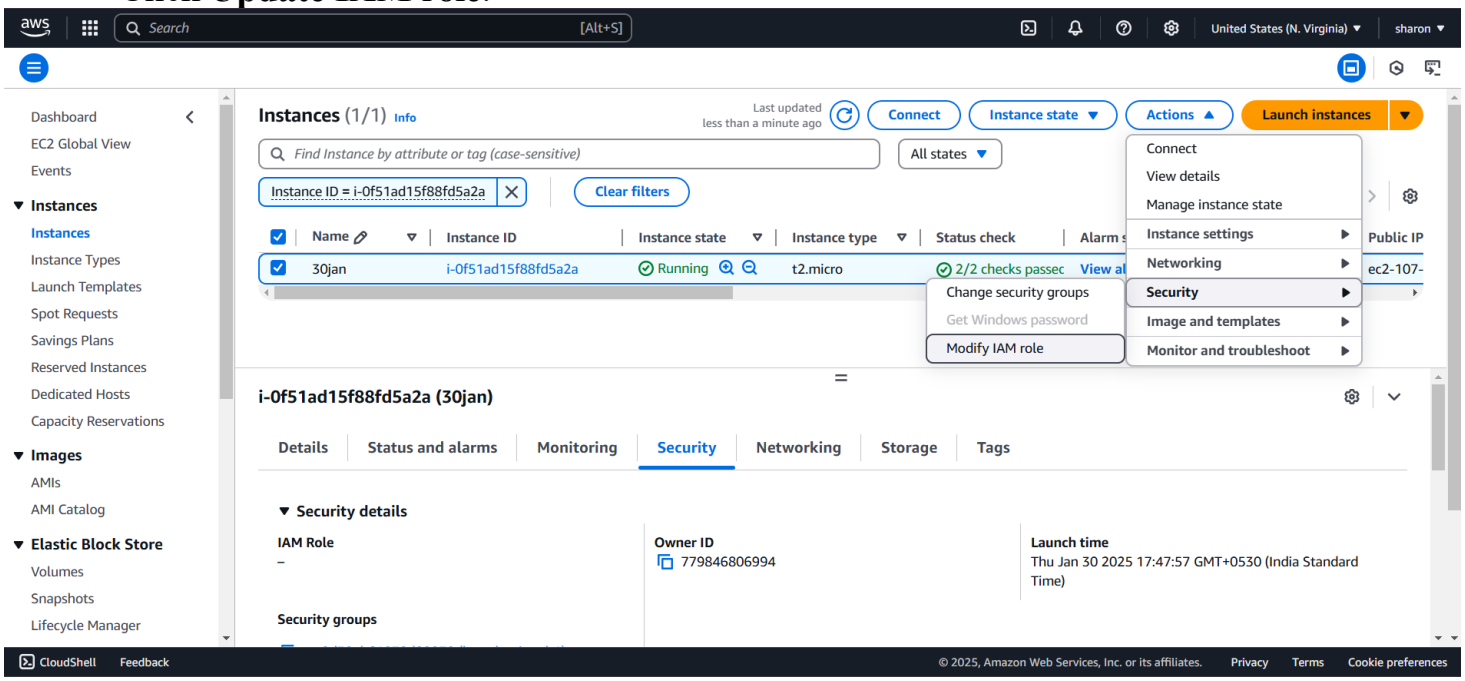
## 2.3. Modify IAM Role

- With your instance selected, click on the **Actions** dropdown at the top right.

- Choose **Security** and then select **Modify IAM role**.

## 2.4. Assign the Role

- In the **Modify IAM role** window, you will see a dropdown labeled **IAM role**.
- Select the IAM role you created earlier (EC2S3AccessRole).
- Click **Update IAM role**.



# Step 3: Test the Permissions on the EC2 Instance 3.1. Connect to Your EC2 Instance

- In the EC2 dashboard, click **Connect** with your instance selected to get the connection details (e.g., SSH or EC2 Instance Connect).
- Use the appropriate method to access your instance.

## 3.2. Test Permitted Actions

- Once connected to your EC2 instance, open a terminal (or command prompt for Windows) and attempt to use AWS CLI commands.

For example, to test S3 access:

aws s3 ls

If the role has the correct permissions, this command should return a list of S3 buckets.

## 3.3. Test Denied Actions

- To test denied actions, try to access a resource or perform an operation that the IAM role doesn’t have permission to do.

For example, if your role doesn't have permission to list EC2 instances, try: aws

ec2 describe-instances

This should return an error indicating permission is denied.

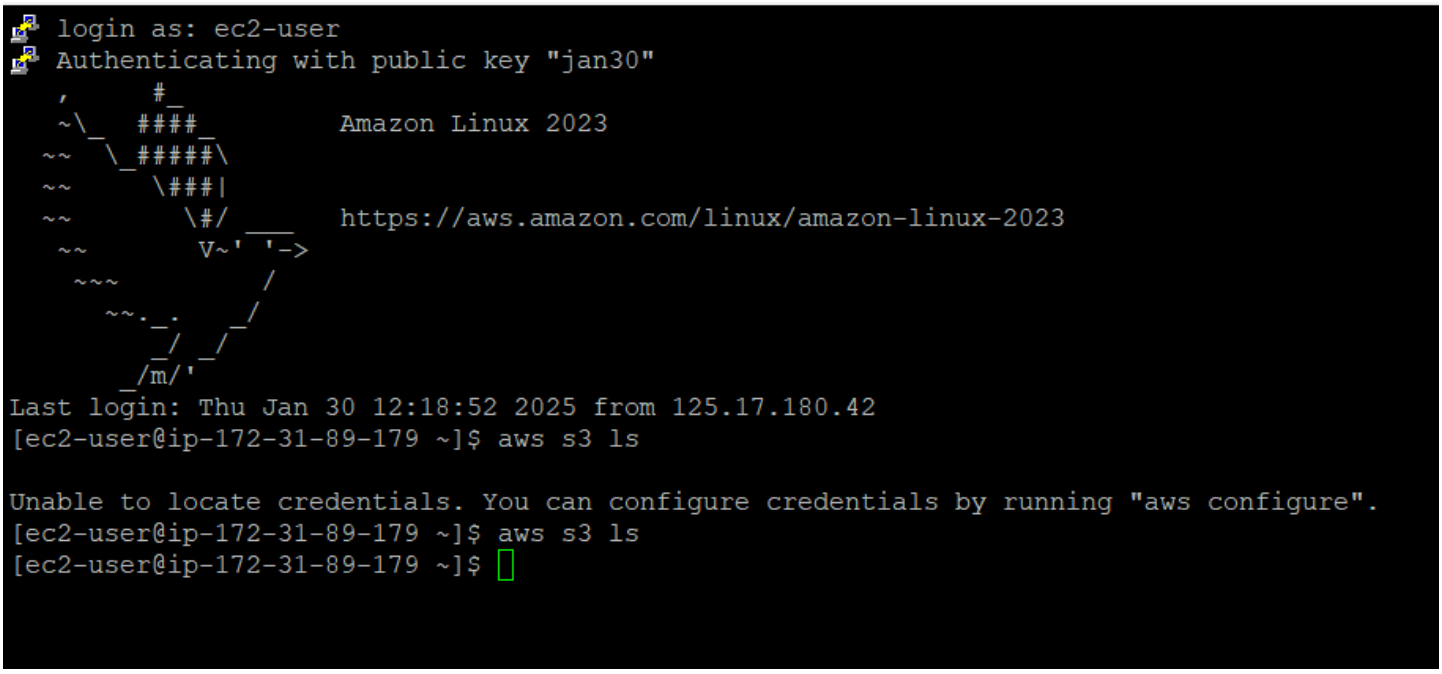

---

# Step 4: Verify the Effect of Permissions 4.1. Confirm Permitted Access

- Ensure that the actions defined by your IAM role's policies (e.g., S3 access) are allowed when you run the respective commands.

## 4.2. Confirm Denied Access

- Ensure that any actions outside the scope of your role’s permissions (e.g., accessing EC2 instances without permissions) result in access being denied.




---

## Conclusion

By following this process, you've successfully created an IAM role, assigned it to an EC2 instance, and tested the permissions to ensure it works as intended. IAM roles provide

finegrained control over access to AWS resources, ensuring your EC2 instance can only perform the actions you've authorized.