

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

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.

The screenshot shows the AWS IAM 'Create role' wizard. The top navigation bar includes the AWS logo, a search bar, and user information. The breadcrumb trail is 'IAM > Roles > Create role'. On the left, a progress indicator shows three steps: 'Step 1: Select trusted entity' (active), 'Step 2: Add permissions', and 'Step 3: Name, review, and create'. The main content area is titled 'Select trusted entity' with an 'Info' link. It contains two sections: 'Trusted entity type' and 'Use case'. The 'Trusted entity type' section has five radio button options: 'AWS service' (selected), 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. The 'Use case' section has a description 'Allow an AWS service like EC2, Lambda, or others to perform actions in this account.' and a dropdown menu labeled 'Service or use case' with 'EC2' selected. Below the dropdown is a note: 'Choose a use case for the specified service. Use case'.

Step 1
● **Select trusted entity**
○ Step 2: Add permissions
○ Step 3: Name, review, and create

Select trusted entity [Info](#)

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

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**.

The screenshot shows the AWS IAM console interface. On the left, a sidebar indicates the current step in the role creation process: Step 1 (Select trusted entity), Step 2 (Add permissions - highlighted), and Step 3 (Name, review, and create). The main area is titled 'Add permissions' and shows a search for 's3' with 13 matches. A table lists several AWS managed policies, with 'AmazonS3ReadOnlyAccess' selected. The table columns are Policy name, Type, and Description.

Policy name	Type	Description
<input type="checkbox"/> AmazonDMSRedshiftS3Role	AWS managed	Provides access to manage S3 settings ...
<input type="checkbox"/> AmazonS3FullAccess	AWS managed	Provides full access to all buckets via t...
<input type="checkbox"/> AmazonS3ObjectLambdaExecutionRolePolicy	AWS managed	Provides AWS Lambda functions permi...
<input type="checkbox"/> AmazonS3OutpostsFullAccess	AWS managed	Provides full access to Amazon S3 on ...
<input type="checkbox"/> AmazonS3OutpostsReadOnlyAccess	AWS managed	Provides read only access to Amazon S...
<input checked="" type="checkbox"/> AmazonS3ReadOnlyAccess	AWS managed	Provides read only access to all bucket...
<input type="checkbox"/> AmazonS3TablesFullAccess	AWS managed	Provides full access to all S3 table buc...
<input type="checkbox"/> AmazonS3TablesReadOnlyAccess	AWS managed	Provides read only access to all S3 tabl...

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**.

The screenshot displays the AWS Management Console interface. On the left, the navigation pane shows the 'Instances' section. The main content area shows a list of instances with one instance, '30jan', selected. The instance details for 'i-0f51ad15f88fd5a2a (30jan)' are shown below the list. The 'Security' tab is active, displaying the 'IAM Role' as '-' and the 'Owner ID' as 779846806994. The 'Actions' menu is open, and 'Modify IAM role' is highlighted.

Name	Instance ID	Instance state	Instance type	Status check	Alarms
30jan	i-0f51ad15f88fd5a2a	Running	t2.micro	2/2 checks passed	

i-0f51ad15f88fd5a2a (30jan)

Security details

IAM Role	Owner ID	Launch time
-	779846806994	Thu Jan 30 2025 17:47:57 GMT+0530 (India Standard Time)

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.

```
login as: ec2-user
Authenticating with public key "jan30"

      #_
~\#### Amazon Linux 2023
~~\#####\
~~\###|
~~\#/ https://aws.amazon.com/linux/amazon-linux-2023
~~V~'-'>
~~~~
~~.-.
~/m/'-
```

Last login: Thu Jan 30 12:18:52 2025 from 125.17.180.42
[ec2-user@ip-172-31-89-179 ~]\$ aws s3 ls

Unable to locate credentials. You can configure credentials by running "aws configure".
[ec2-user@ip-172-31-89-179 ~]\$ █

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.

```
login as: ec2-user
Authenticating with public key "jan30"

#_
~\####_      Amazon Linux 2023
~~~\#####\
~~~\_###|
~~~\#/      https://aws.amazon.com/linux/amazon-linux-2023
~~~V~'-'>
~~~~
~~~~
~/m/'-/
```

Last login: Thu Jan 30 12:18:52 2025 from 125.17.180.42
[ec2-user@ip-172-31-89-179 ~]\$ aws s3 ls

Unable to locate credentials. You can configure credentials by running "aws configure".
[ec2-user@ip-172-31-89-179 ~]\$ aws s3 ls
[ec2-user@ip-172-31-89-179 ~]\$

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.