

## Roles in IAM Service

IAM Roles: Temporary IDs for Secure Access.

Imagine you have a toolbox with different tools for various tasks. IAM roles are like sets of specific tools (permissions) that you can temporarily give to people or applications who need them. **These roles don't have permanent access** like keys, so they're more secure.

Key Points:

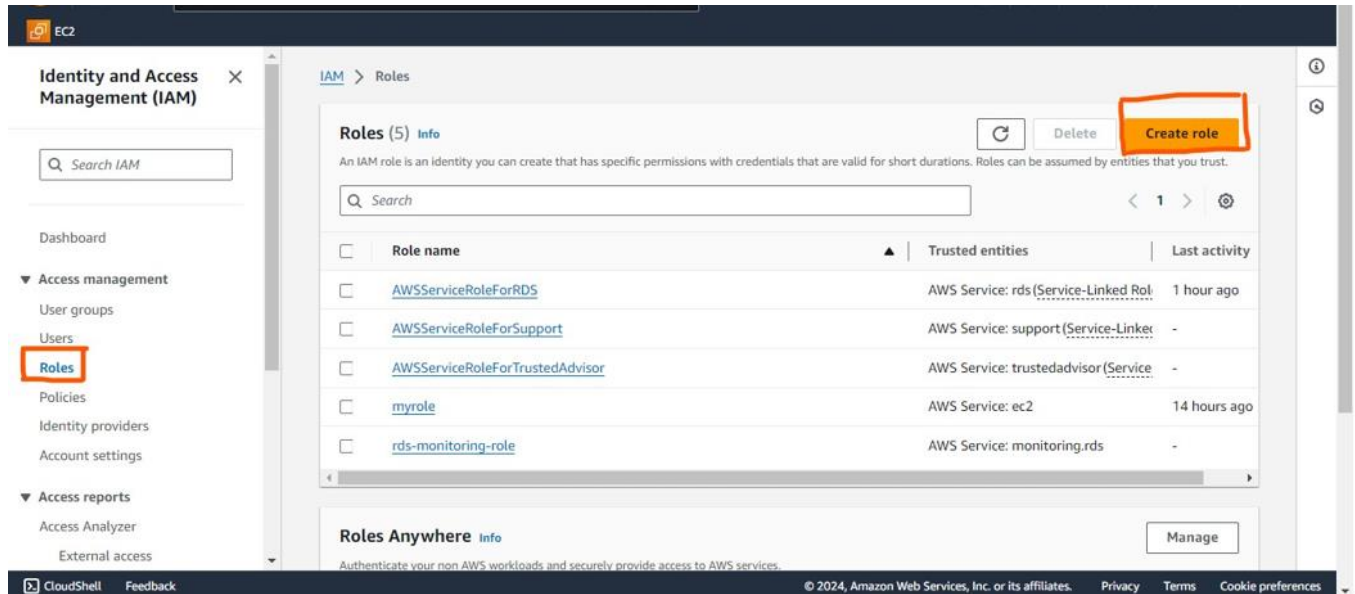
1. Roles are temporary identities, unlike permanent user accounts.
2. They provide secure access with defined permissions.
3. They're ideal for automation and shared access scenarios.

How Roles Work:

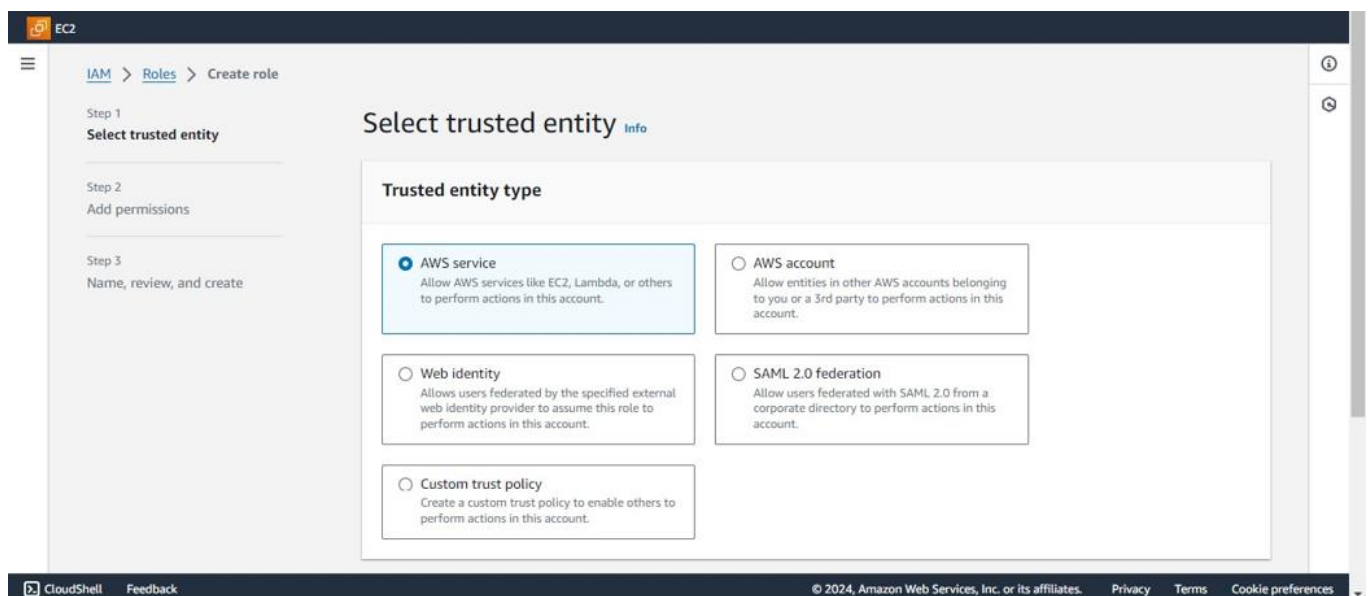
1. Create a Role: You define what actions the role can perform (e.g., starting an EC2 instance, uploading files to S3).
2. Grant Access: You provide temporary credentials (like a passcode) for someone or an application to "assume" the role, granting them the defined permissions.
3. Use the Tools: Whoever assumes the role can use the allowed tools for a limited time, like in a construction project where workers use specific tools for specific tasks.
4. Return the Tools: When they're done, the temporary credentials expire, and the access goes away.

## Creating a role :-

### 1. Click on **Roles** and **Create role** option



### 2. Select **AWS service** option



### 3. Select the service type (in this case EC2 is selected)

The screenshot shows the 'Use case' selection screen in the AWS IAM console. The 'Service or use case' dropdown is set to 'EC2'. Below it, a list of use cases is displayed, with 'EC2' selected. The other use cases include 'EC2 Role for AWS Systems Manager', 'EC2 Spot Fleet Role', 'EC2 - Spot Fleet Auto Scaling', 'EC2 - Spot Fleet Tagging', 'EC2 - Spot Instances', 'EC2 - Spot Fleet', and 'EC2 - Scheduled Instances'. The 'Next' button is highlighted in orange.

**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

- ☒ **EC2**  
Allows EC2 instances to call AWS services on your behalf.
- ☐ **EC2 Role for AWS Systems Manager**  
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
- ☐ **EC2 Spot Fleet Role**  
Allows EC2 Spot Fleet to request and terminate Spot instances on your behalf.
- ☐ **EC2 - Spot Fleet Auto Scaling**  
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.
- ☐ **EC2 - Spot Fleet Tagging**  
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.
- ☐ **EC2 - Spot Instances**  
Allows EC2 Spot instances to launch and manage spot instances on your behalf.
- ☐ **EC2 - Spot Fleet**  
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.
- ☐ **EC2 - Scheduled Instances**  
Allows EC2 Scheduled Instances to manage instances on your behalf.

Cancel Next

### 4. Select policy as per you choice (in this case Admin all access is selected)

The screenshot shows the 'Add permissions' screen in the AWS IAM console. The 'Permissions policies' section is active, showing a list of policies. The 'AdministratorAccess' policy is selected. The 'Set permissions boundary - optional' section is also visible. The 'Next' button is highlighted in orange.

**Add permissions** info

Permissions policies (1/912) info

Choose one or more policies to attach to your new role.

Filter by Type  
All types 4 matches

<input type="checkbox"/>	Policy name	Type	Description
<input checked="" type="checkbox"/>	<a href="#">AdministratorAccess</a>	AWS managed - job function	Provides full access to AWS services and r...
<input type="checkbox"/>	<a href="#">AdministratorAccess-Amplify</a>	AWS managed	Grants account administrative permission...
<input type="checkbox"/>	<a href="#">AdministratorAccess-AWSElasticBeanstalk</a>	AWS managed	Grants account administrative permission...
<input type="checkbox"/>	<a href="#">AWSAuditManagerAdministratorAccess</a>	AWS managed	Provides administrative access to enable o...

► Set permissions boundary - optional

Cancel Previous Next

## 5. Assign any name as per your choice

The screenshot shows the AWS IAM console 'Create role' page. The left sidebar indicates the current step is 'Step 3: Name, review, and create'. The main content area is titled 'Name, review, and create' and contains a 'Role details' section. In this section, the 'Role name' field is populated with 'role\_for\_ec2'. The 'Description' field contains the text 'Allows EC2 instances to call AWS services on your behalf.' Below the role details, there is a section for 'Step 1: Select trusted entities' with an 'Edit' button. A 'Trust policy' section shows a JSON snippet for the trust policy. The bottom of the page features a footer with 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates.

EC2

IAM > Roles > Create role

Step 1  
[Select trusted entity](#)

Step 2  
[Add permissions](#)

Step 3  
**Name, review, and create**

### Name, review, and create

**Role details**

**Role name**  
Enter a meaningful name to identify this role.  
  
Maximum 64 characters. Use alphanumeric and '+', '@', '-' characters.

**Description**  
Add a short explanation for this role.  
  
Maximum 1000 characters. Use alphanumeric and '+', '@', '-' characters.

**Step 1: Select trusted entities** Edit

**Trust policy**

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Principal": {  
7         "Service": "ec2.amazonaws.com"  
8       }  
9     }  
10  ]  
11 }
```

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## 6. Scroll down and click on create role

The screenshot shows the AWS IAM console 'Create role' page, continuing from Step 1. The left sidebar indicates the current step is 'Step 2: Add permissions'. The main content area shows the 'Permissions policy summary' table, which lists the 'AdministratorAccess' policy as an 'AWS managed - job function' type, attached as a 'Permissions policy'. Below this, the 'Step 3: Add tags' section is visible, showing 'Add tags - optional' and a message that 'No tags associated with the resource.' are present. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Create role'. The bottom of the page features a footer with 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates.

EC2

IAM > Roles > Create role

Step 1  
[Select trusted entity](#)

Step 2  
**Add permissions**

Step 3  
[Add tags](#)

### Permissions policy summary

Policy name	Type	Attached as
<a href="#">AdministratorAccess</a>	AWS managed - job function	Permissions policy

### Step 3: Add tags

**Add tags - optional** [Info](#)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

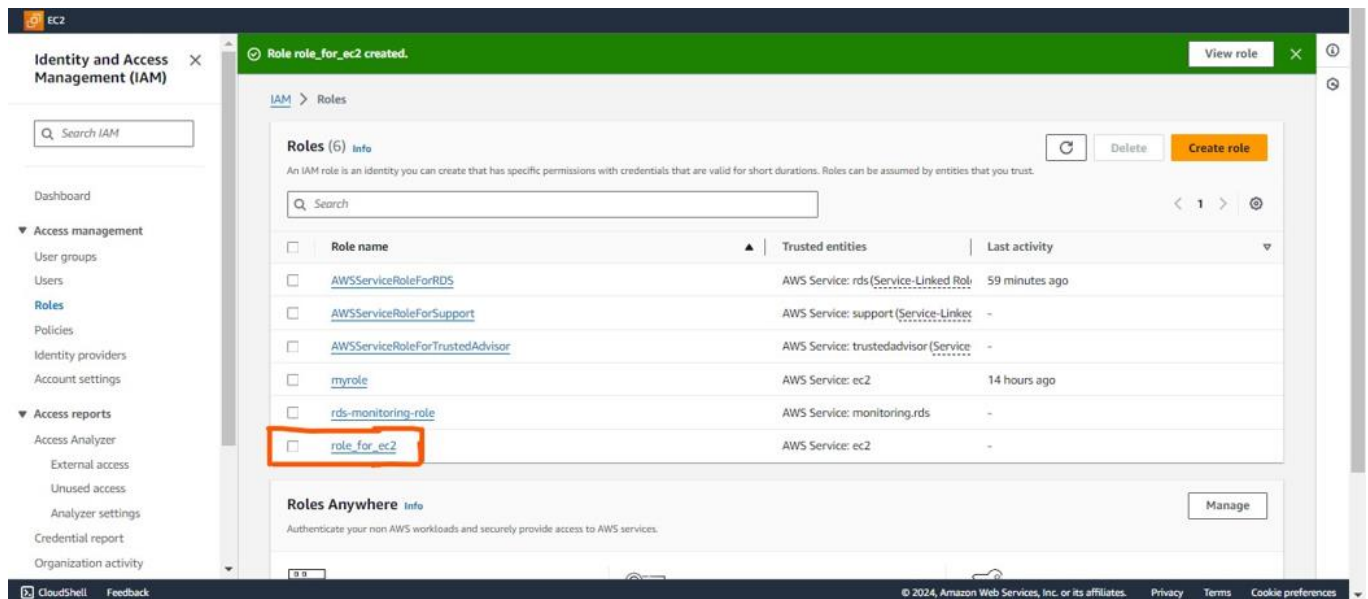
Add new tag

You can add up to 50 more tags.

Cancel Previous Create role

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

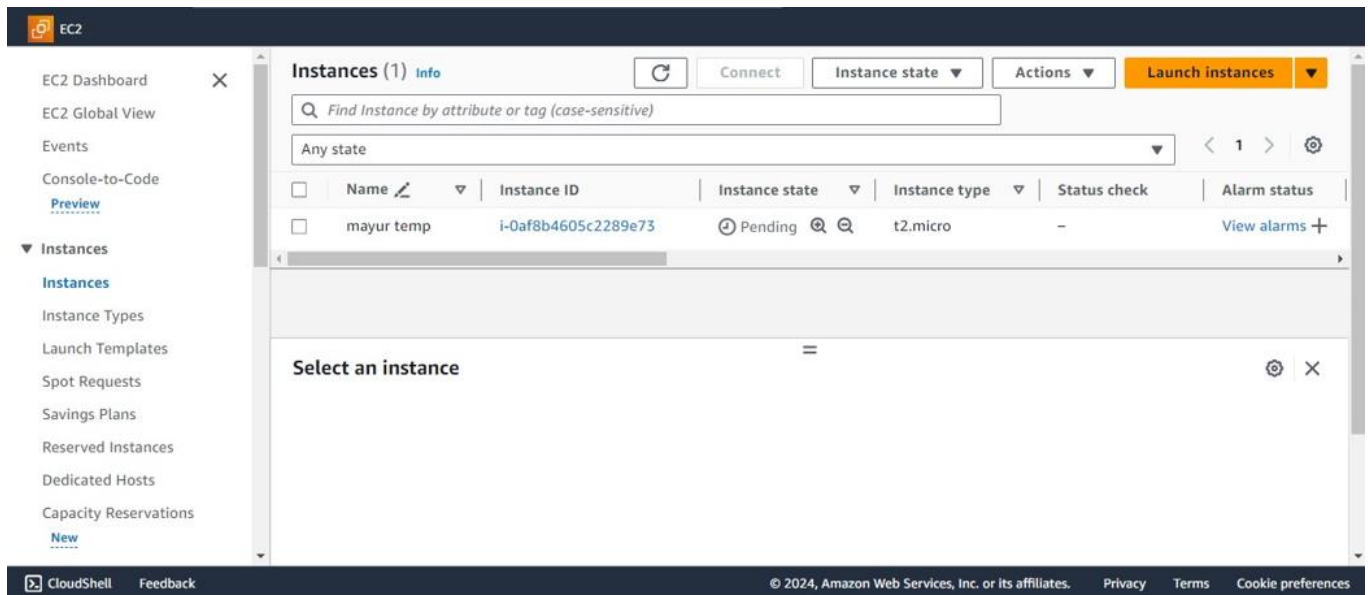
## 7. Role created successfully



## Assigning a role :-

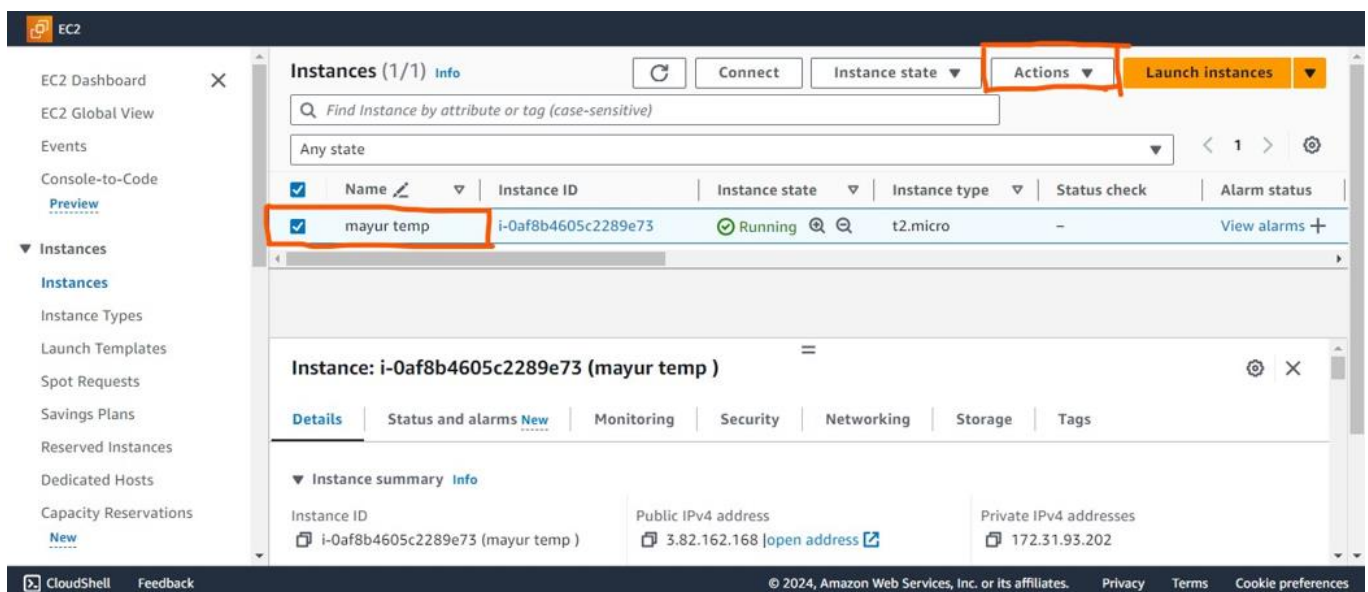
### Step1:- Create EC2 Instance

- Search EC2 service in aws search bar
- Click on create instance
- Assign any name as per your choice
- Select any image as per your choice ( aws linux is selected)
- Assign a name to key value pair and click on **create new key value pair**
- Download the key value pairs
- Click on launch instance
- Instance Created successfully

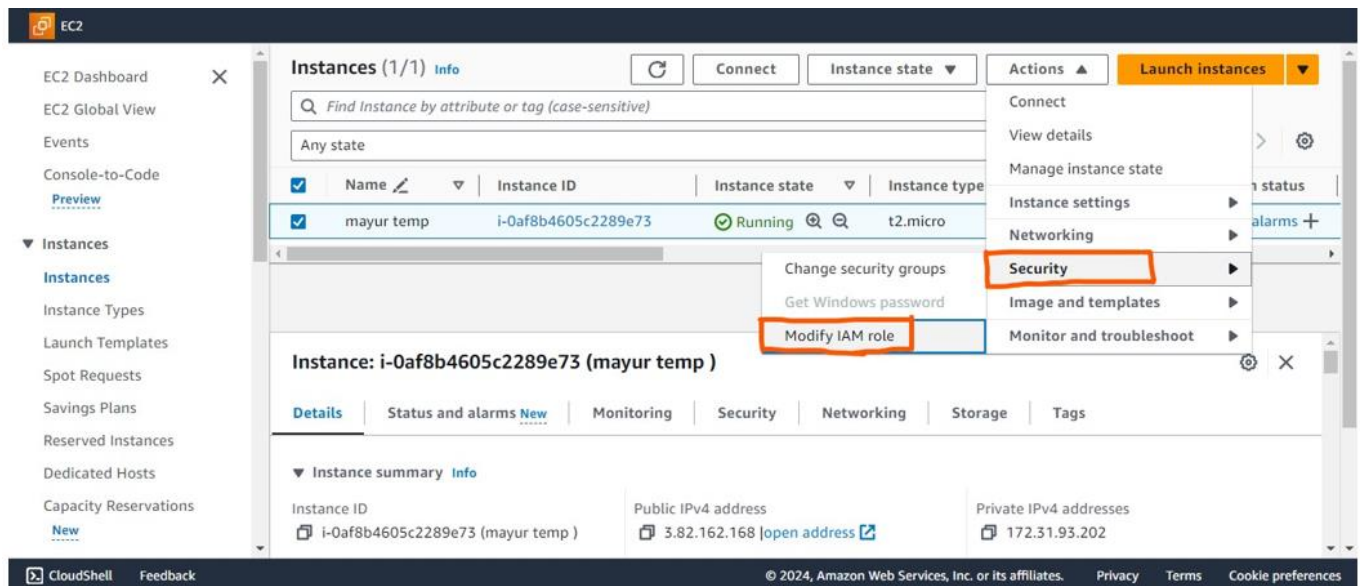


## Step 2: Assign role to EC2 instance

1. Select the created instance and click on **actions**

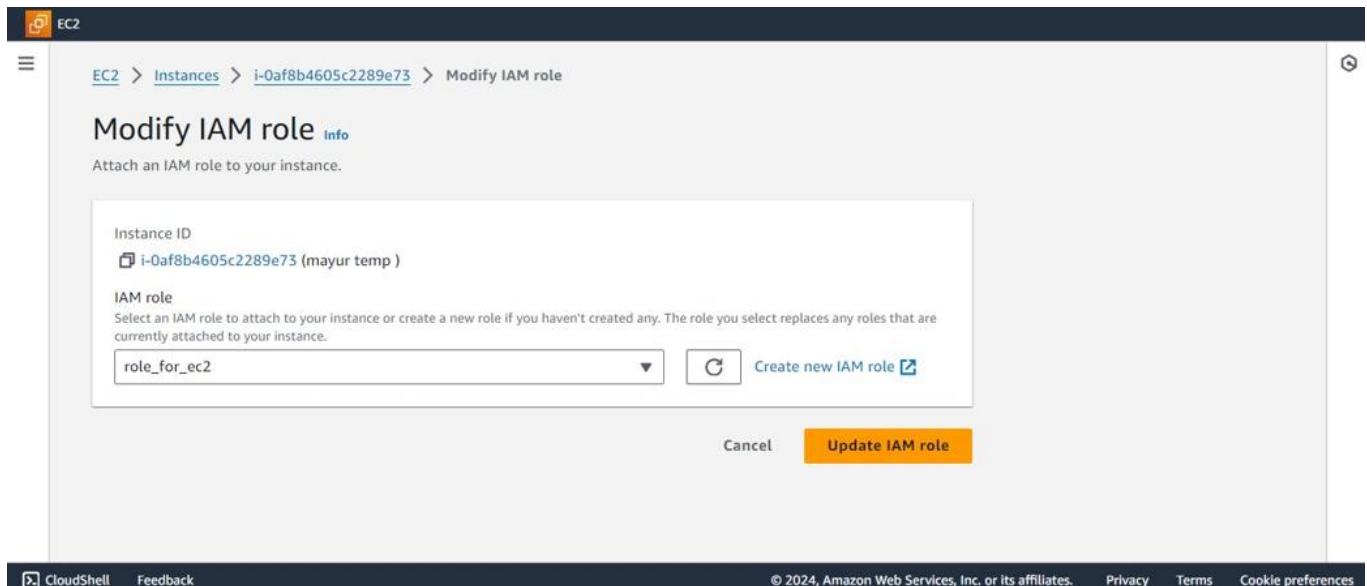


## 2. Click on **security** and **Modify IAM role**



The screenshot shows the AWS Management Console for EC2. In the left sidebar, the 'Instances' section is expanded, and the 'Security' tab is highlighted. The main content area shows a list of instances. The instance 'mayur temp' with ID 'i-0af8b4605c2289e73' is selected. The 'Actions' dropdown menu is open, and the 'Modify IAM role' option is highlighted. Below the instance list, the 'Instance: i-0af8b4605c2289e73 (mayur temp)' details are visible, including the 'Security' tab.

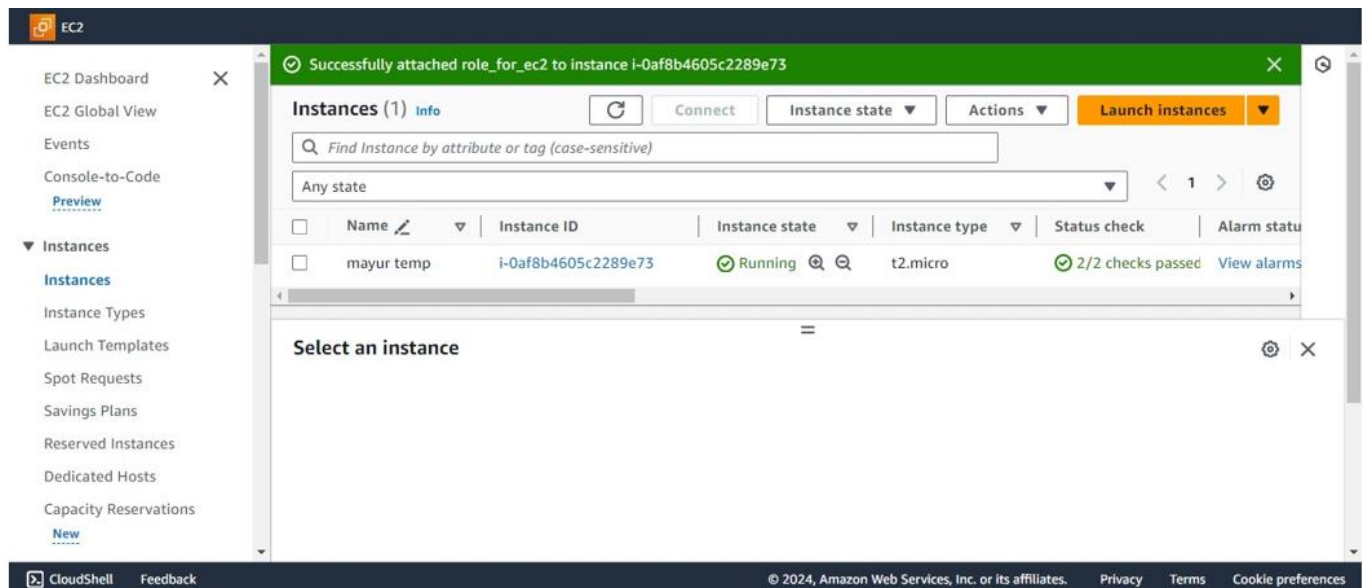
## 3. Select a role which you created in previous step and click on **update IAM role**



The screenshot shows the 'Modify IAM role' page in the AWS Management Console. The instance ID 'i-0af8b4605c2289e73' is selected. The 'IAM role' dropdown menu is open, and the 'role\_for\_ec2' role is selected. The 'Update IAM role' button is highlighted.

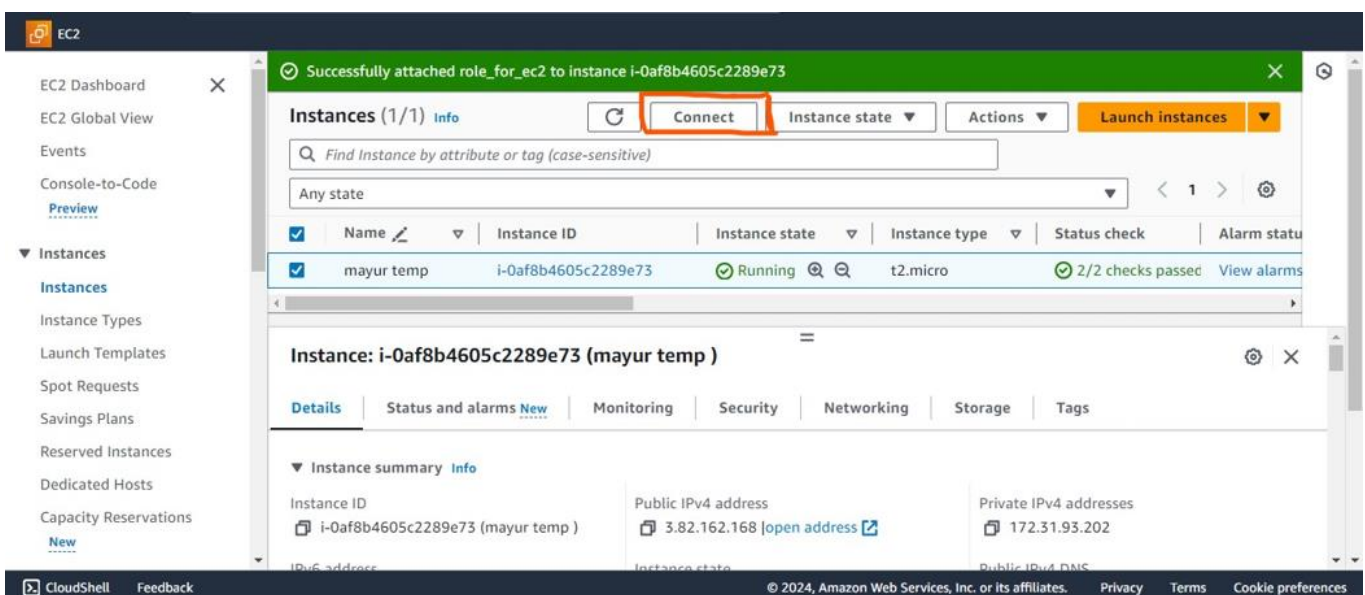


#### 4. Role attached successfully to instance



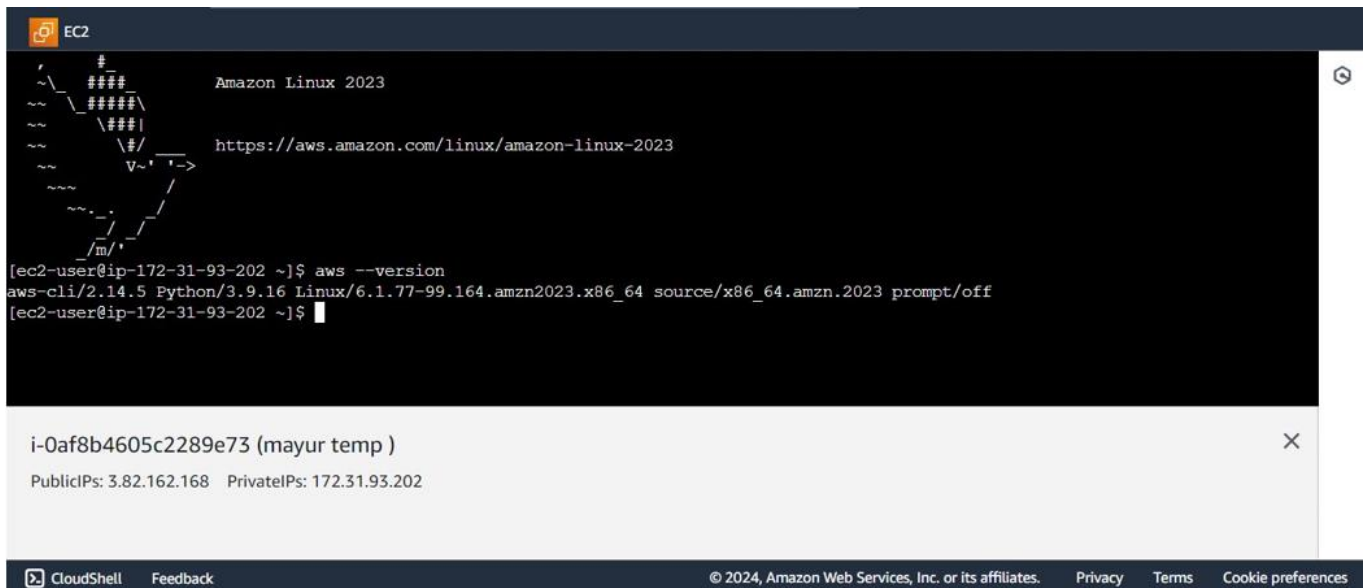
### Step 3 :- performing the actions through ec2

#### 1. Select the instance and click on **connect** option





2. Make sure that the aws-cli is installed in ec2 instance using **aws – version** command



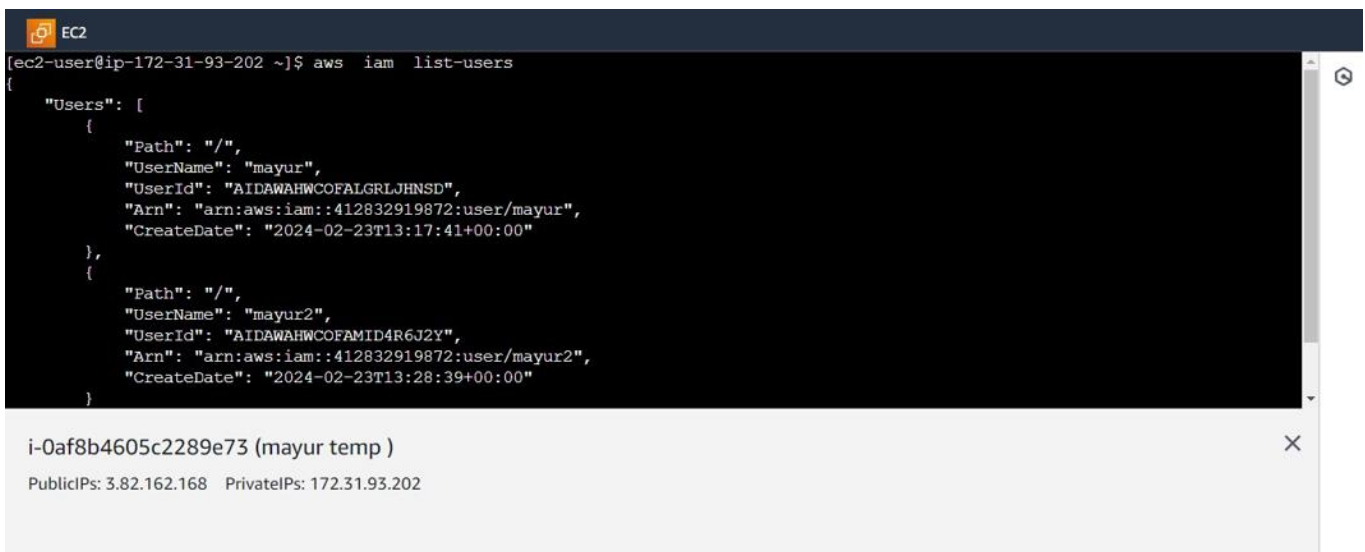
The screenshot shows a terminal window titled "EC2" with the Amazon Linux 2023 logo and URL. The user runs the command `aws --version`, which outputs: `aws-cli/2.14.5 Python/3.9.16 Linux/6.1.77-99.164.amzn2023.x86_64 source/x86_64.amzn.2023 prompt/off`. Below the terminal, a metadata bar shows the instance ID `i-0af8b4605c2289e73` (mayur temp) and IP addresses: PublicIPs: 3.82.162.168, PrivateIPs: 172.31.93.202. The footer includes "CloudShell", "Feedback", and copyright information for Amazon Web Services, Inc. or its affiliates.

```
[ec2-user@ip-172-31-93-202 ~]$ aws --version
aws-cli/2.14.5 Python/3.9.16 Linux/6.1.77-99.164.amzn2023.x86_64 source/x86_64.amzn.2023 prompt/off
[ec2-user@ip-172-31-93-202 ~]$
```

i-0af8b4605c2289e73 (mayur temp )  
PublicIPs: 3.82.162.168 PrivateIPs: 172.31.93.202

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

3. Successfully accessing the aws iam users lists



The screenshot shows a terminal window titled "EC2" with the Amazon Linux 2023 logo and URL. The user runs the command `aws iam list-users`, which outputs a JSON list of two IAM users: "mayur" and "mayur2". Below the terminal, a metadata bar shows the instance ID `i-0af8b4605c2289e73` (mayur temp) and IP addresses: PublicIPs: 3.82.162.168, PrivateIPs: 172.31.93.202. The footer includes "CloudShell", "Feedback", and copyright information for Amazon Web Services, Inc. or its affiliates.

```
[ec2-user@ip-172-31-93-202 ~]$ aws iam list-users
{
  "Users": [
    {
      "Path": "/",
      "UserName": "mayur",
      "UserId": "AIDAWAHWCOFALGRLJHNSD",
      "Arn": "arn:aws:iam::412832919872:user/mayur",
      "CreateDate": "2024-02-23T13:17:41+00:00"
    },
    {
      "Path": "/",
      "UserName": "mayur2",
      "UserId": "AIDAWAHWCOFAMID4R6J2Y",
      "Arn": "arn:aws:iam::412832919872:user/mayur2",
      "CreateDate": "2024-02-23T13:28:39+00:00"
    }
  ]
}
```

i-0af8b4605c2289e73 (mayur temp )  
PublicIPs: 3.82.162.168 PrivateIPs: 172.31.93.202

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences