SET UP IAM Role And Permission Management

ABOUT:

AWS Identity and Access Management (IAM) is a security service that helps manage access to AWS resources by defining **who can access what** within an AWS account. IAM allows you to create **users, groups, and roles**, assign **permissions** using **policies**, and enforce **fine-grained access control** across AWS services.

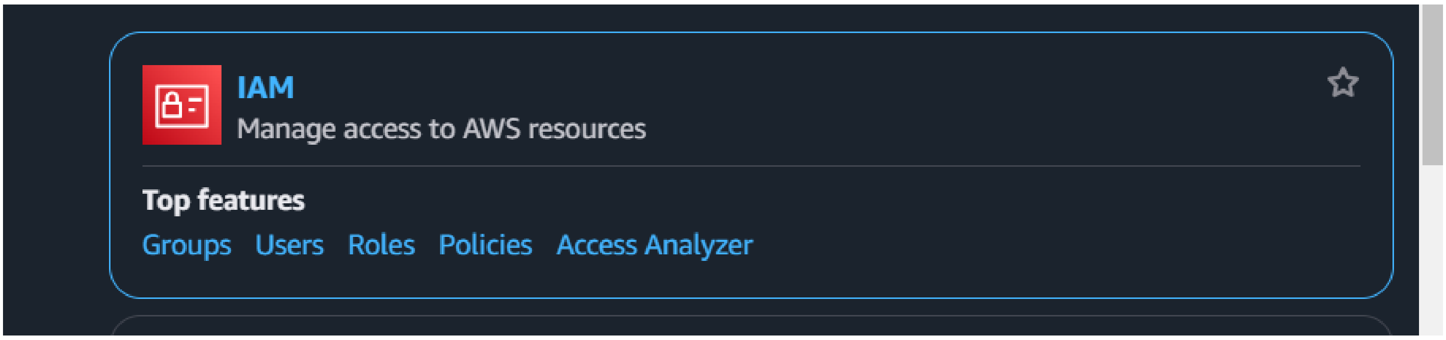
With **IAM Role and Permission Management**, AWS ensures secure authentication and authorization, following the **principle of least privilege** to prevent unauthorized access. IAM is essential for **controlling user actions, integrating with AWS services, and ensuring compliance** with security best practices.

SIGNIFICANCE:

1. **Enhanced Security** – Controls access to AWS resources, reducing the risk of unauthorized access.
2. **Granular Access Control** – Allows fine-tuned permission settings for users, groups, and services.
3. **Simplified User Management** – Enables role-based access, reducing the need for sharing credentials.
4. **Regulatory Compliance** – Helps meet security standards and audit requirements by managing permissions effectively.

STEP 1:

Go to AWS console and search for IAM

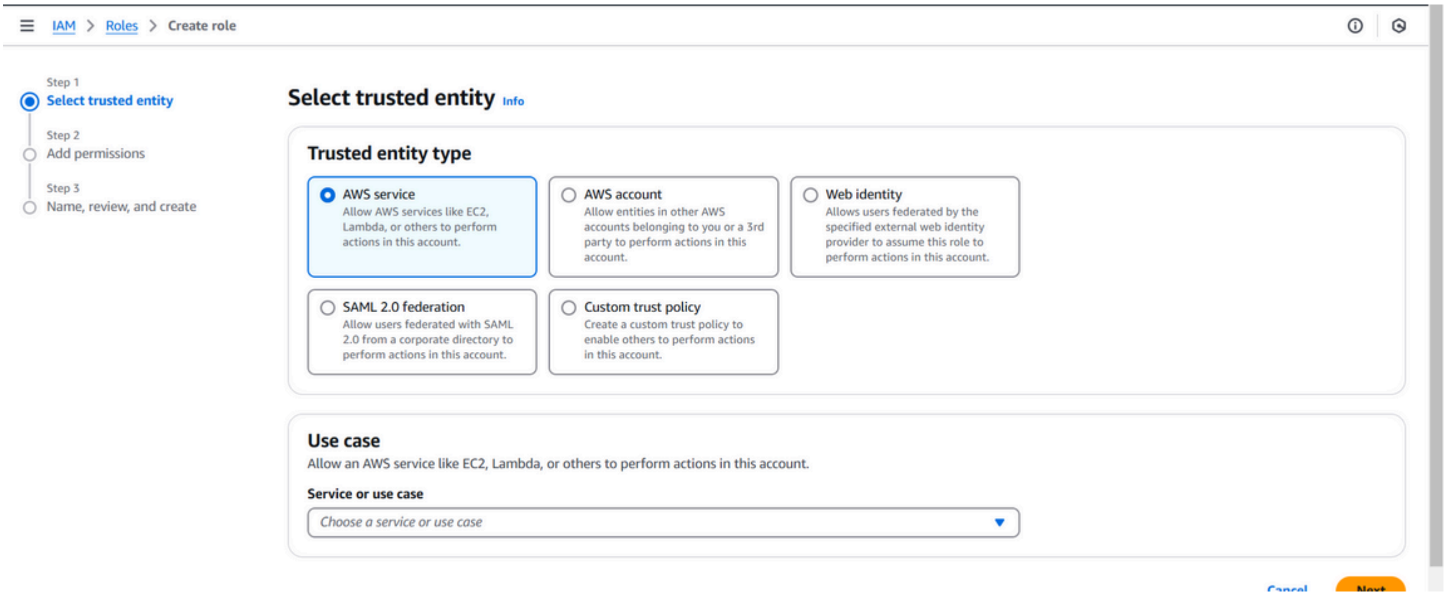


IAM

Manage access to AWS resources

STEP 2:

On the left tab ,click on roles-select create roles.



Select trusted entity

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.

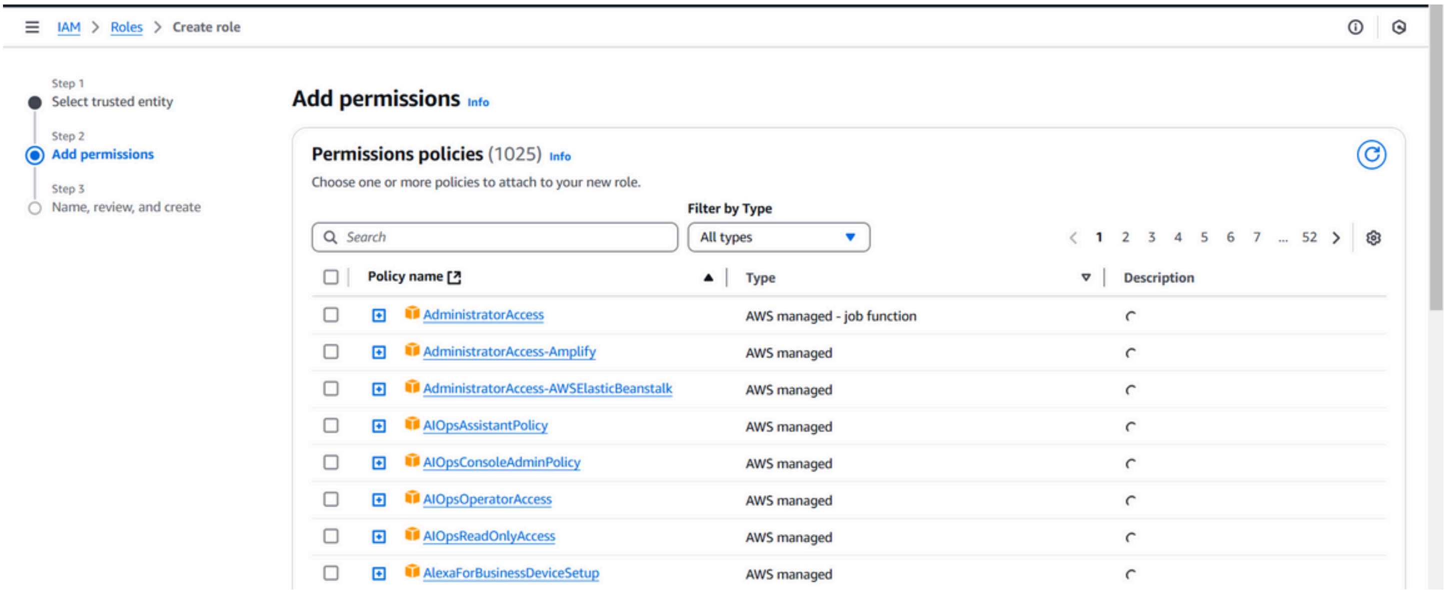
Service or use case

Choose a service or use case

Choose AWS Service under trusted entity AND Choose EC2 under Use Case

STEP 3:

Click on next to go to the permission tab



Step 1
● Select trusted entity

Step 2
◉ Add permissions

Step 3
○ Name, review, and create

Add permissions Info

Permissions policies (1025) Info

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

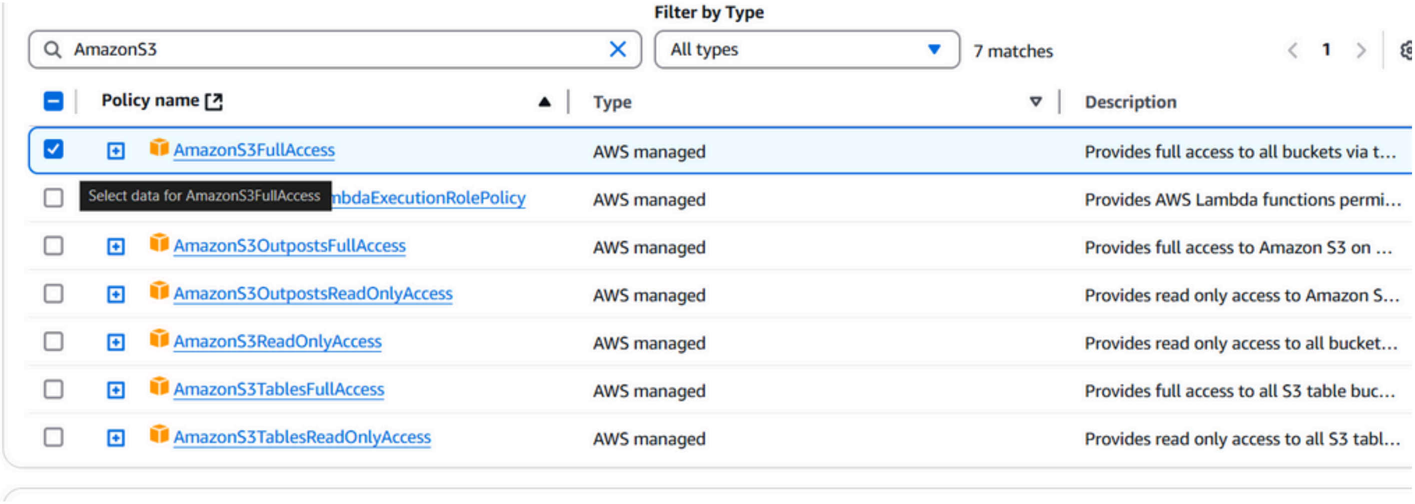
Filter by Type

Search | All types ▼ | < 1 2 3 4 5 6 7 ... 52 >

□	Policy name	Type	Description
□	AdministratorAccess	AWS managed - job function	
□	AdministratorAccess-Amplify	AWS managed	
□	AdministratorAccess-AWSElasticBeanstalk	AWS managed	
□	AIOpsAssistantPolicy	AWS managed	
□	AIOpsConsoleAdminPolicy	AWS managed	
□	AIOpsOperatorAccess	AWS managed	
□	AIOpsReadOnlyAccess	AWS managed	
□	AlexaForBusinessDeviceSetup	AWS managed	

STEP 4:

Select the policy name as per your choice



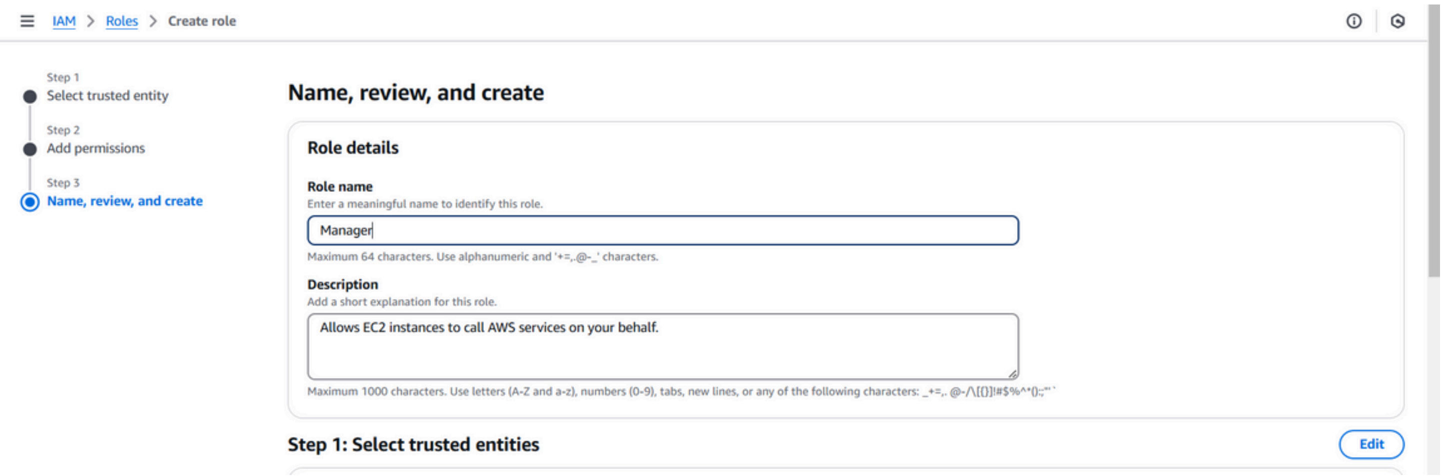
Filter by Type

AmazonS3 ✕ | All types ▼ | 7 matches | < 1 >

☑	Policy name	Type	Description
☑	AmazonS3FullAccess	AWS managed	Provides full access to all buckets via t...
□	Select data for AmazonS3FullAccess mbdaExecutionRolePolicy	AWS managed	Provides AWS Lambda functions permi...
□	AmazonS3OutpostsFullAccess	AWS managed	Provides full access to Amazon S3 on ...
□	AmazonS3OutpostsReadOnlyAccess	AWS managed	Provides read only access to Amazon S...
□	AmazonS3ReadOnlyAccess	AWS managed	Provides read only access to all bucket...
□	AmazonS3TablesFullAccess	AWS managed	Provides full access to all S3 table buc...
□	AmazonS3TablesReadOnlyAccess	AWS managed	Provides read only access to all S3 tabl...

STEP 5:

Give a name of the role you’ve given permission . Click on create role



Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

Manager

Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.

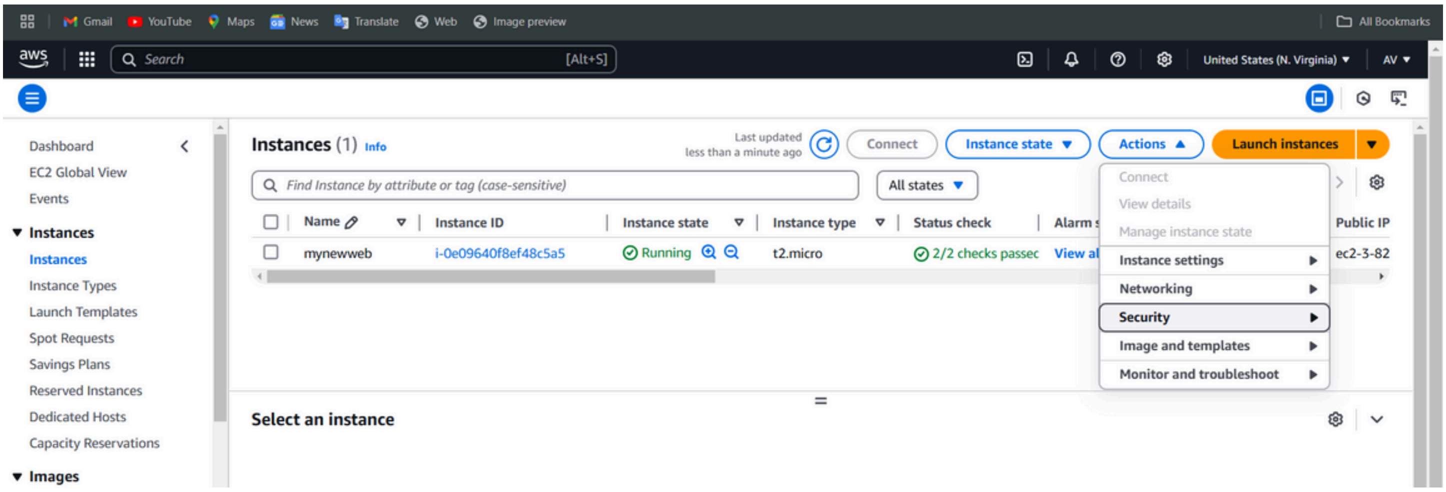
Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{()}]!#$%^*();:"'`

Step 1: Select trusted entities

STEP 6:

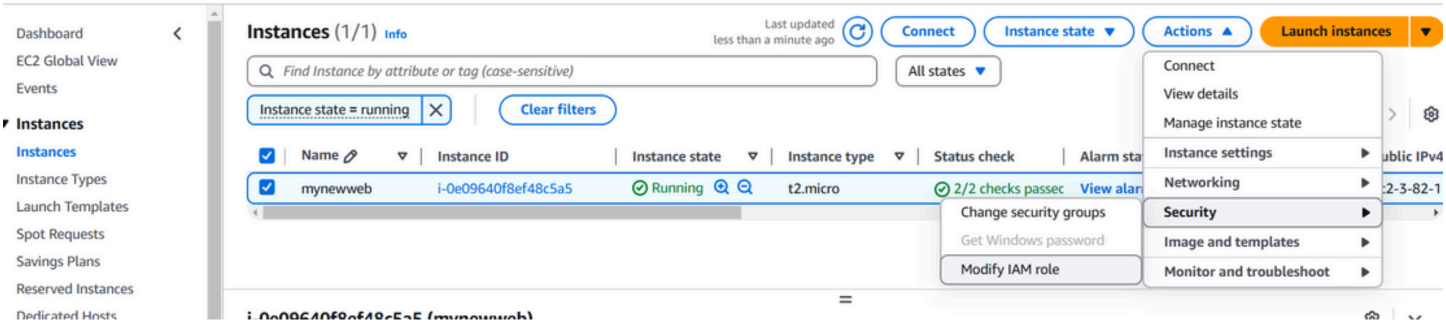
Go to EC2 instance and click on actions , choose security



☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm s	Public IP
☐	mynewweb	i-0e09640f8ef48c5a5	Running	t2.micro	2/2 checks passec	View al	ec2-3-82

STEP 8:

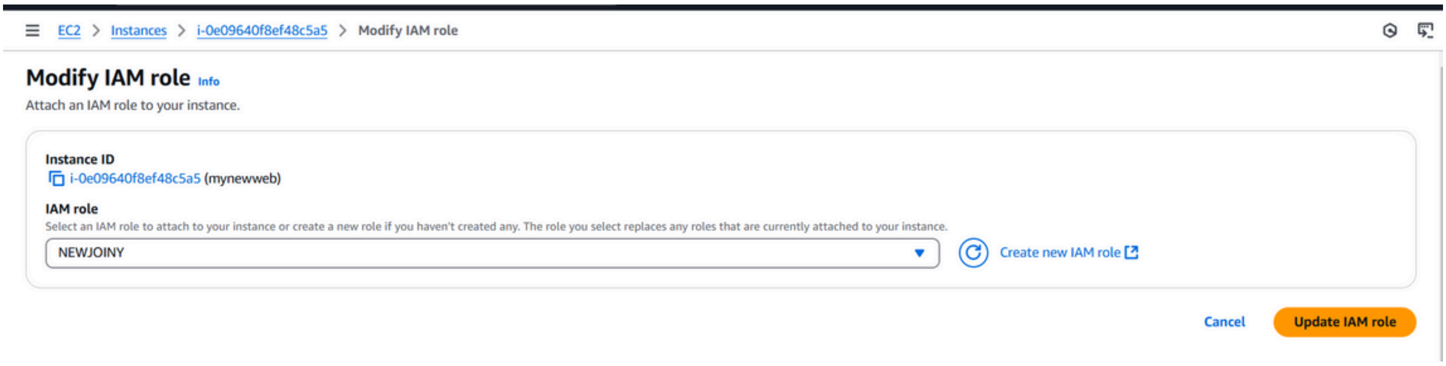
Click on security and choose modify IAM role



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm sta	ublic IPv4
☑	mynewweb	i-0e09640f8ef48c5a5	Running	t2.micro	2/2 checks passec	View alar	2-3-82-1

STEP 9:

Assign the permission



Modify IAM role Info

Attach an IAM role to your instance.

Instance ID
i-0e09640f8ef48c5a5 (mynewweb)

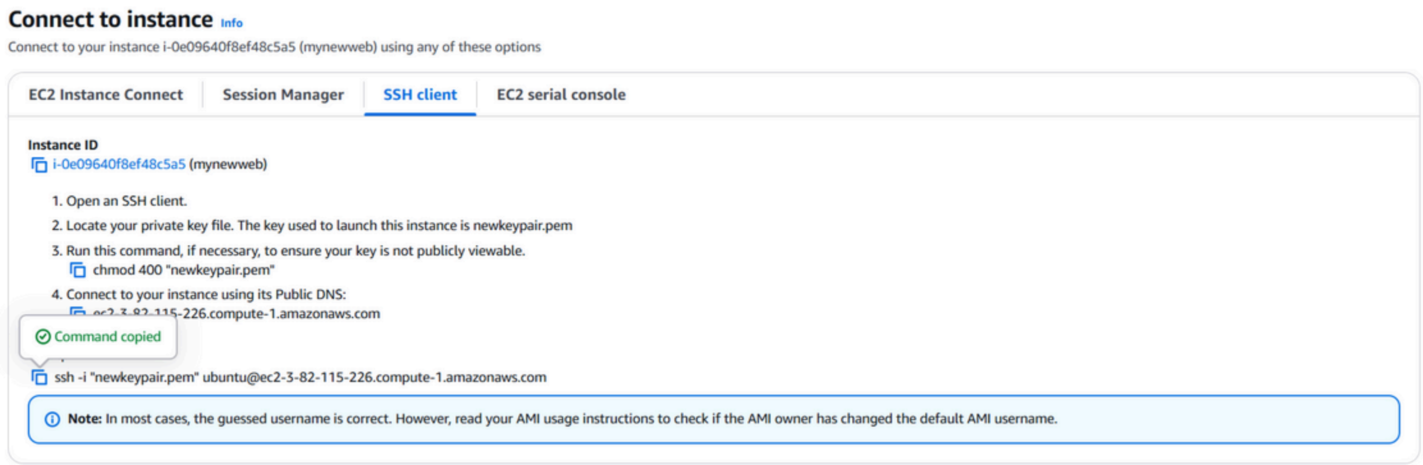
IAM role
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

NEWJOINY

Select update IAM role

STEP 10:

Go to EC2 instance, click on connect and under SSH , copy the link


Connect to instance Info

Connect to your instance i-0e09640f8ef48c5a5 (mynewweb) using any of these options

Instance ID
i-0e09640f8ef48c5a5 (mynewweb)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is newkeypair.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 "newkeypair.pem"
4. Connect to your instance using its Public DNS:
   ec2-3-82-115-226.compute-1.amazonaws.com

Command copied

```
ssh -i "newkeypair.pem" ubuntu@ec2-3-82-115-226.compute-1.amazonaws.com
```

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

STEP 11:

Copy link to command prompt and check if it shows error.

OUTCOME:

Improved Security Posture – Ensures that only authorized users and services can access specific AWS resources.

Efficient Access Management – Simplifies user provisioning and permission control, reducing administrative overhead.

Seamless Service Integration – Allows AWS services to interact securely using IAM roles without hardcoded credentials.

Better Compliance and Auditing – Provides detailed logs and monitoring for security audits and regulatory compliance.