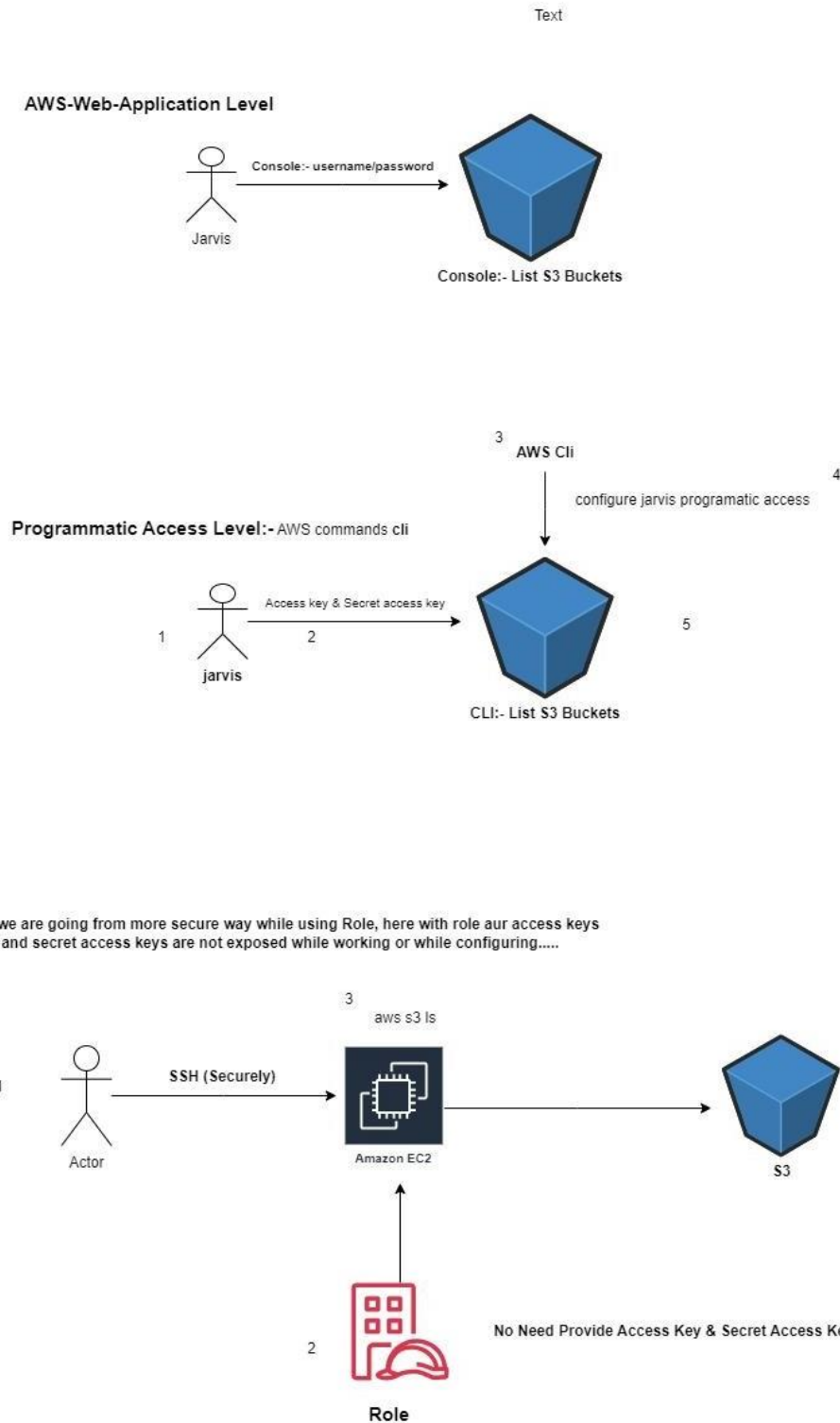


# IAM - Identity and Access Management

I want to check my all S3 Buckets :- Using Console/Programmatic Access/Roles

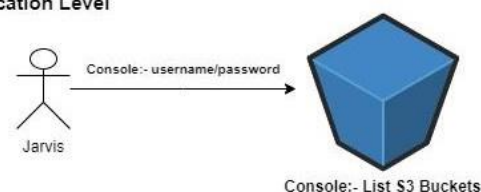


## AWS-Web-Application Level

## Use IAM Service

## 1. AWS – web application level.

Create a user in IAM

We can access any service using **AWS – web application level**. (As per above diagram -1)

## User details

User name

Om

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ \_ - (hyphen)

☒ Provide user access to the AWS Management Console - *optional*If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

## Are you providing console access to a person?

User type

☐ Specify a user in Identity Center - Recommended

We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

☒ I want to create an IAM user

We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

Console password

☐ Autogenerated password

You can view the password after you create the user.

☒ Custom password

Enter a custom password for the user.

\*\*\*\*\*

- Must be at least 8 characters long
- Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & \* ( ) \_ + - (hyphen) = [ ] { } ' "

☐ Show password☒ Users must create a new password at next sign-in - RecommendedUsers automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

IAM &gt; Users &gt; Om &gt; Add permissions

Step 1

Add permissions

Step 2

Review

## Add permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

## Permissions options

☐ Add user to group

Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ Copy permissions

Copy all group memberships, attached managed policies, inline policies, and any existing permissions boundaries from an existing user.

☒ Attach policies directly

Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

## Permissions policies (1/1221)

s3rea

Filter by Type

All types

1 match

<input checked="" type="checkbox"/>	Policy name	Type	Attached entities
<input checked="" type="checkbox"/>	AmazonS3ReadOnlyAccess	AWS managed	0

Cancel

Next

Review and create user.

Download .csv file (Contain User name, Password, Console sign-in URL & paste console URL to incognito to check login.

**User created successfully** You can view and download the user's password and email instructions for signing in to the AWS Management Console. [View user](#)

Step 1: Specify user details  
Step 2: Set permissions  
Step 3: Review and create  
Step 4: **Retrieve password**

### Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

**Console sign-in details** [Email sign-in instructions](#)

**Console sign-in URL**  
https://905418477144.signin.aws.amazon.com/console

User name  
Om

Console password  
om@12345 [Hide](#)

[Cancel](#) [Download .csv file](#) [Return to users list](#)

Here need to change the password.

Amazon Web Services Sign-In

ap-southeast-2.signin.aws.amazon.com/clm?action=changepassword&userType=iam&redirect\_uri=https%3A%2F%2Fconsole.aws.amazon.com%2Fconsole

**aws**

You must change your password to continue

**AWS account** 905418477144

**IAM user name** Om

**Old password**

**New password**

**Retype new password**

[Confirm password change](#)

[Sign in using root user email](#)

English

[Terms of Use](#) [Privacy Policy](#) © 1996-2024 Amazon Web Services, Inc. or its affiliates

Create a demo S3 bucket in aws root user & check using IAM user that can be access.

aws Services Search [Alt+S] Sydney Om @ 9054-1847-7144

### Amazon S3

Buckets  
Access Grants  
Access Points  
Object Lambda Access Points  
Multi-Region Access Points  
Batch Operations  
IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens  
Dashboards  
Storage Lens groups  
AWS Organizations settings

Feature spotlight

**Account snapshot - updated every 24 hours** [View Storage Lens dashboard](#)

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

**General purpose buckets** [Directory buckets](#)

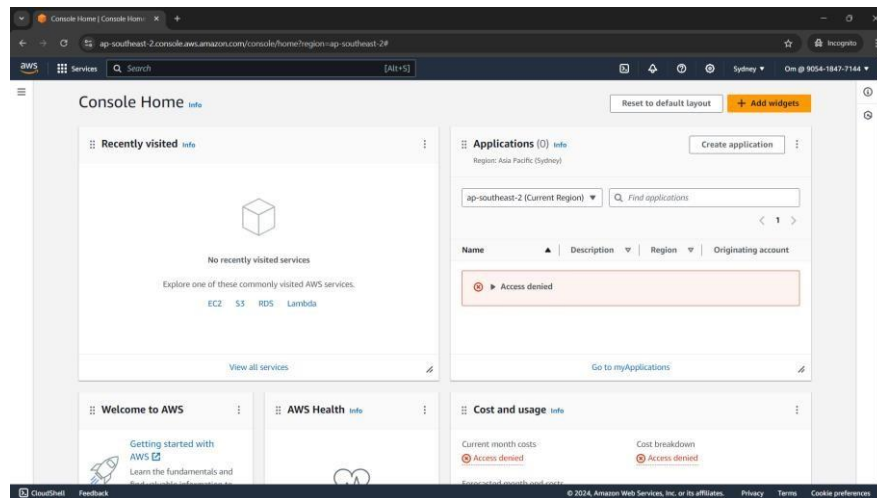
**General purpose buckets (1)** [Info](#) [All AWS Regions](#)

Buckets are containers for data stored in S3.

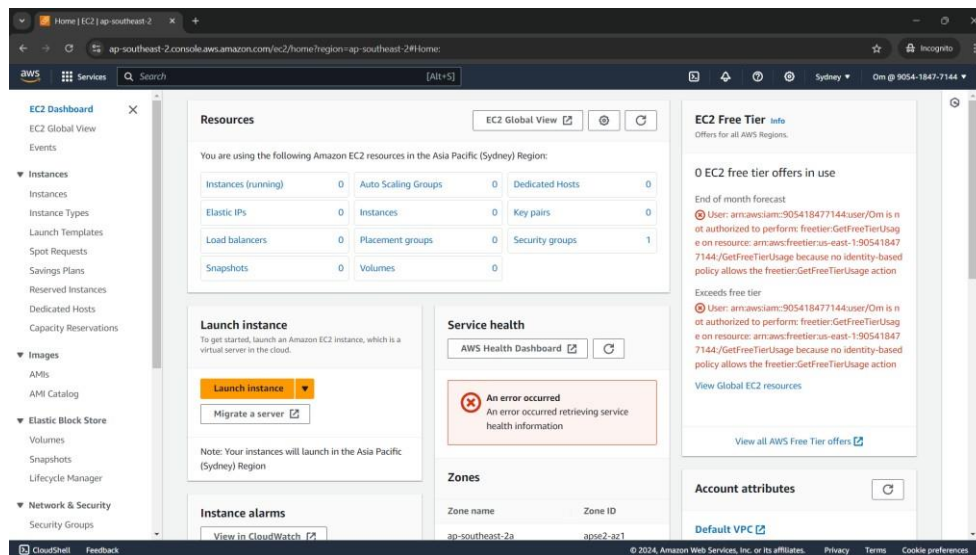
[Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Name	AWS Region	IAM Access Analyzer	Creation date
<a href="#">demo-cli-bucket</a>	Asia Pacific (Mumbai) ap-south-1	<a href="#">View analyzer for ap-south-1</a>	August 8, 2024, 14:33:24 (UTC+05:30)

Console home for newly created IAM user with assigned limited policies. (ie. Ec2 full access



We can access only ec2 services.



## 2. Programmatic Access Level - AWS Command CLI (Command Line Interface) We work on same user User security credentials access key

Programmatic Access Level:- AWS commands cli



us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users/details/Om?section=security\_credentials

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users**
- Roles
- Policies
- Identity providers
- Account settings

Access reports

- Access Analyzer
- External access
- Unused access
- Analyzer settings
- Credential report
- Organization activity
- Service control policies

IAM > Users > Om

Om info

Summary

ARN arn:aws:iam::905418477144:user/Om	Console access Enabled without MFA	Access key 1 Create access key
Created August 08, 2024, 13:41 (UTC+05:30)	Last console sign-in Today	

Permissions Groups Tags **Security credentials** Access Advisor

Console sign-in

Console sign-in link  
https://905418477144.signin.aws.amazon.com/console

Console password  
Not enabled

Last console sign-in  
19 minutes ago (2024-08-08 13:47 GMT+5:30)

Multi-factor authentication (MFA) (0)

Remove Resync Assign MFA device

## Create access key

Access keys (0)

Create access key

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. [Learn more](#)

No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. [Learn more](#)

Create access key

IAM > Users > Om > Create access key

Step 1  
Access key best practices & alternatives

Step 2 - optional  
Set description tag

Step 3  
Retrieve access keys

### Access key best practices & alternatives

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

- ☒ **Command Line Interface (CLI)**  
You plan to use this access key to enable the AWS CLI to access your AWS account.
- ☐ **Local code**  
You plan to use this access key to enable application code in a local development environment to access your AWS account.
- ☐ **Application running on an AWS compute service**  
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- ☐ **Third-party service**  
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- ☐ **Application running outside AWS**  
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
- ☐ **Other**  
Your use case is not listed here.

Alternatives recommended

- Use [AWS CloudShell](#), a browser-based CLI, to run commands. [Learn more](#)
- Use the [AWS CLI V2](#) and enable authentication through a user in IAM Identity Center. [Learn more](#)

Confirmation

☒ I understand the above recommendation and want to proceed to create an access key.

Cancel Next

Download .csv file & secret access key is one time shown.

**Access key created**  
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

[IAM](#) > [Users](#) > [Om](#) > Create access key

Step 1  
[Access key best practices & alternatives](#)

Step 2 - optional  
[Set description tag](#)

Step 3  
**Retrieve access keys**

### Retrieve access keys [Info](#)

**Access key**  
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIA5FTZFRZMBOFNAB5A	90IUc5Zp04i93dLCfyQBapnOtNwxnyOhALz5xj58 <a href="#">Hide</a>

**Access key best practices**

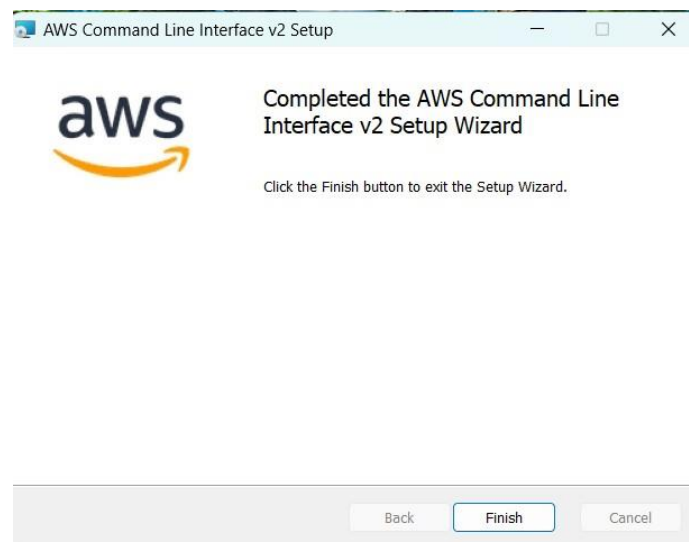
- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

[Download .csv file](#) [Done](#)

Now, install amazon CLI that is for Windows

<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>



Create a sample S3 bucket (that we already created and try to access them using CLI using AWS Access key ID & AWS Secret Access key ID

```
Command Prompt
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rushi>aws --version
aws-cli/2.17.25 Python/3.11.9 Windows/10 exe/AMD64

C:\Users\rushi>aws configure
AWS Access Key ID [None]: AKIA5FTZFRZMBOFNAB5A
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:

C:\Users\rushi>aws s3 ls
2024-08-08 14:33:24 demo-cli-buckett

C:\Users\rushi>
```

## Now for Linux-

### Create an Instance & connect it to MobaXterm

<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

Follow AWS CLI install and update instructions for Linux

- **Downloading from the URL** – To download the installer with your browser, use the following URL:  
[https://awscli.amazonaws.com/awscli-exe-linux-x86\\_64.zip](https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip)

### Commands-

sudo apt update

wget https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip & sudo apt-get install zip -y

ls → awscli-exe-linux-x86\_64.zip wget-log

unzip awscli-exe-linux-x86\_64.zip

ls → aws awscli-exe-linux-x86\_64.zip wget-log

cd aws

ls → README.md THIRD\_PARTY\_LICENSES dist install

sudo ./install

aws --version

aws configure

AWS Access Key ID [None]: AKIA5FTZFRZMBOFNAB5A

AWS Secret Access Key [None]: 90lUc5Zp04i93dLCFyQBpN0tNwxnyOhALz5xjS8

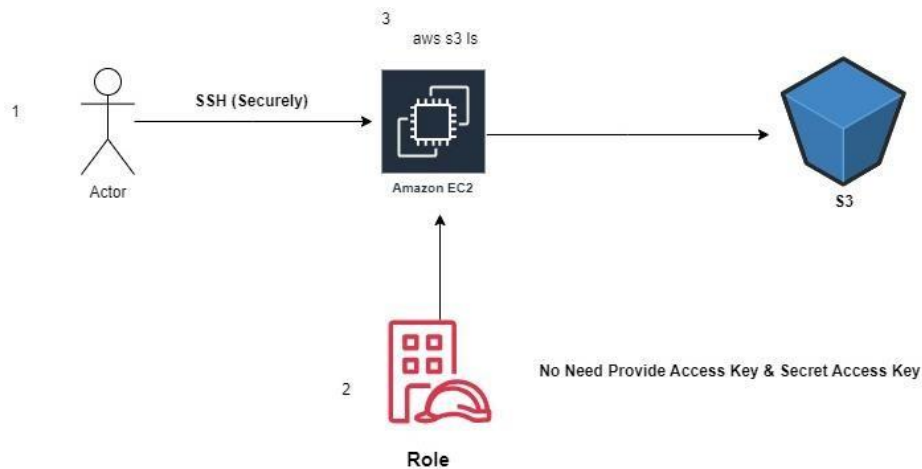
aws s3 ls

```
ubuntu@ip-172-31-38-155:~$ wget https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip & sudo apt-get install zip -y
[1] 1638

Redirecting output to 'wget-log'.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
zip is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
[1]+  Done                  wget https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip
ubuntu@ip-172-31-38-155:~$ ls
awscli-exe-linux-x86_64.zip  wget-log
ubuntu@ip-172-31-38-155:~$ unzip awscli-exe-linux-x86_64.zip
Archive:  awscli-exe-linux-x86_64.zip
   creating: aws/
   creating: aws/dist/
   inflating: aws/dist/docutils/parsers/rst/include/isogr1.txt
ubuntu@ip-172-31-38-155:~$ ls
aws  awscli-exe-linux-x86_64.zip  wget-log
ubuntu@ip-172-31-38-155:~$ cd aws
ubuntu@ip-172-31-38-155:~/aws$ ls
README.md  THIRD_PARTY_LICENSES  dist  install
ubuntu@ip-172-31-38-155:~/aws$ sudo ./install
You can now run: /usr/local/bin/aws --version
ubuntu@ip-172-31-38-155:~/aws$ aws --version
aws-cli/2.17.25 Python/3.11.9 Linux/6.8.0-1009-aws exe/x86_64.ubuntu.24
ubuntu@ip-172-31-38-155:~/aws$ aws configure
AWS Access Key ID [None]: AKIA5FTZFRZMB0FNAB5A
AWS Secret Access Key [None]: 90lUc5Zp04i93dLCFyQBpN0tNwxnyOhALz5xjS8
Default region name [None]:
Default output format [None]:
ubuntu@ip-172-31-38-155:~/aws$ aws s3 ls
2024-08-08 09:03:26 demo-cli-buckett
ubuntu@ip-172-31-38-155:~/aws$
```



Now we are going from more secure way while using the Role, here with the role our access key and secret access keys are not exposed while working or while configuring



## Create new user

### User details

User name

Sanket

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ \_ - (hyphen)

☒ Provide user access to the AWS Management Console - *optional*

If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.



#### Are you providing console access to a person?

User type

☐ Specify a user in Identity Center - Recommended

We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

☒ I want to create an IAM user

We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

Console password

☐ Autogenerated password

You can view the password after you create the user.

☒ Custom password

Enter a custom password for the user.

sanket@123

- Must be at least 8 characters long

- Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & \* ( ) \_ + - (hyphen) = [ ] { } ' "

☒ Show password

☒ Users must create a new password at next sign-in - Recommended

## Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

### Permissions options

☐ Add user to group

Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ Copy permissions

Copy all group memberships, attached managed policies, and inline policies from an existing user.

☒ Attach policies directly

Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

### Permissions policies (1/1223)

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



[Create policy](#)

Q s3read

Filter by Type

All types

1 match

< 1 > ⚙

<input checked="" type="checkbox"/>	Policy name	Type	Attached entities
<input checked="" type="checkbox"/>	AmazonS3ReadOnlyAccess	AWS managed	1

► Set permissions boundary - optional

Cancel

Previous

Next

## Create Role for user (Sanket

[IAM](#) > [Roles](#) > Create role

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

Choose a service or use case

Cancel

Next

[IAM](#) > [Roles](#) > Create role

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

☒ EC2

Allows EC2 instances to call AWS services on your behalf.

☐ EC2 Role for AWS Systems Manager

IAM > Roles > Create role

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

## Add permissions Info

Permissions policies (1/945) Info

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

Filter by Type: All types 1 match

Policy name	Type	Description
<input checked="" type="checkbox"/> AmazonS3ReadOnlyAccess	AWS managed	Provides read only access to all buckets vi...

► Set permissions boundary - optional

Cancel Previous Next

## Add one more permission policy

IAM > Roles > Sanket-Role > Add permissions

Attach policy to Sanket-Role

► Current permissions policies (1)

Other permissions policies (1/944)

Filter by Type: All types 1 match

Policy name	Type	Description
<input checked="" type="checkbox"/> AmazonEC2FullAccess	AWS managed	Provides full access to Amazon EC2 via th...

Cancel Add permissions

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.

Sanket-Role

Maximum 64 characters. Use alphanumeric and "+,=,@,\_" characters.

Description  
Add a short explanation for this role.

Allows S3 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 Edit

Trust policy

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Principal": {
7         "Service": "s3.amazonaws.com"
8       },
9       "Action": "sts:AssumeRole"
10    }
11  ]
12 }

```

## Create new instance with aws root & Action→Security→modify IAM role

Instances (1/2) Info

Find Instance by attribute or tag (case-sensitive)

All states

Name	Instance ID	Instance state	Instance type	Status check
Demo-IAM-Se...	i-087e0f3dede3c1730	Running	t2.micro	2/2 checks passed
Sanket-Role-S...	i-07516c52fedbdfa8	Running	t2.micro	2/2 checks passed

Change security groups  
Get Windows password  
Modify IAM role

This action is available for Windows instances.

Launch instances

Connect  
View details  
Manage instance state  
Instance settings  
Networking  
Security  
Image and templates  
Monitor and troubleshoot


i-07516c52fedbdfa8 (Sanket-Role-Server)

EC2 > Instances > i-07516c52fedbda8 > Modify IAM role

## Modify IAM role [Info](#)


Attach an IAM role to your instance.

Instance ID

 i-07516c52fedbda8 (Sanket-Role-Server)

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.


[Create new IAM role](#)

[Cancel](#)
[Update IAM role](#)

### Connect to MobaXterm

wget https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip & sudo apt-get install zip -y

ubuntu@ip-172-31-45-198:~\$ ls

awscli-exe-linux-x86\_64.zip wget-log

ubuntu@ip-172-31-45-198:~\$ unzip awscli-exe-linux-x86\_64.zip

To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo\_root" for details.

```
ubuntu@ip-172-31-45-198:~$ wget https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip & sudo apt-get install zip -y
[1] 1165
```

```
Redirecting output to 'wget-log'.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  unzip
```

```
[1]+ Done wget https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip
```

```
ubuntu@ip-172-31-45-198:~$ ls
awscli-exe-linux-x86_64.zip wget-log
ubuntu@ip-172-31-45-198:~$ unzip awscli-exe-linux-x86_64.zip
Archive: awscli-exe-linux-x86_64.zip
  creating: aws/
  creating: aws/dist/
  inflating: aws/THIRD_PARTY_LICENSES
  inflating: aws/README.md
  inflating: aws/install
  creating: aws/dist/awscli/
  creating: aws/dist/cryptography/
```

```
ubuntu@ip-172-31-45-198:~$ ls
aws awscli-exe-linux-x86_64.zip wget-log
ubuntu@ip-172-31-45-198:~$ cd aws
ubuntu@ip-172-31-45-198:~/aws$ ls
README.md THIRD_PARTY_LICENSES dist install
ubuntu@ip-172-31-45-198:~/aws$ sudo ./install
You can now run: /usr/local/bin/aws --version
ubuntu@ip-172-31-45-198:~/aws$ aws --version
aws-cli/2.17.25 Python/3.11.9 Linux/6.8.0-1009-aws exe/x86_64.ubuntu.24
ubuntu@ip-172-31-45-198:~/aws$ aws s3 ls
2024-08-08 09:03:26 demo-cli-bucket
ubuntu@ip-172-31-45-198:~/aws$ \
```

# Multi-factor authentication (MFA)

Identity and Access Management (IAM)

Q Search IAM

Dashboard

Access management

- User groups
- Users**
- Roles
- Policies
- Identity providers
- Account settings

Access reports

- Access Analyzer
  - External access
  - Unused access
  - Analyzer settings
- Credential report
- Organization activity
- Service control policies

Related console

- IAM Identity Center
- AWS Organizations

IAM > Users > Om

Om

Delete

Summary

ARN

- arnawsiam:905418477144user/Om

Created

- August 08, 2024, 13:41 (UTC+05:30)

Console access

- Enabled without MFA

Last console sign-in

- Today

Access key 1

- AKIASFTZF8ZMBQFMABSA - Active
- Never used. Created today.

Access key 2

- Create access key

Permissions Groups Tags **Security credentials** Access Advisor

Console sign-in

Manage console access

Console sign-in link

- https://905418477144.signin.aws.amazon.com/console

Console password

- Not enabled

Last console sign-in

- 2 hours ago (2024-08-08 13:47 GMT+5:30)

Multi-factor authentication (MFA) (0)

Remove Resync Assign MFA device

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. [Learn more](#)

Type	Identifier	Certifications	Created on
No MFA devices. Assign an MFA device to improve the security of your AWS environment			

Assign MFA device

IAM > Users > Om > Assign MFA device

Step 1

Select MFA device

Step 2

Set up device

Select MFA device

MFA device name

Device name

This name will be used within the identifying ARN for this device.

Sanket-MFA

Maximum 64 characters. Use alphanumeric and \* , , @ , - , \_ characters.

MFA device

Device options

In addition to username and password, you will use this device to authenticate into your account.

Passkey or security key

Authenticate using your fingerprint, face, or screen lock. Create a passkey on this device or use another device, like a FIDO2 security key.

Authenticator app

Authenticate using a code generated by an app installed on your mobile device or computer.

Hardware TOTP token

Authenticate using a code generated by Hardware TOTP token or other hardware devices.

13

IAM > Users > Om > Assign MFA device

Step 1

Select MFA device

Step 2

**Set up device**

Set up device

Authenticator app


A virtual MFA device is an application running on your device that you can configure by scanning a QR code.

1

Install a compatible application such as Google Authenticator, Duo Mobile, or Authy app on your mobile device or computer.

[See a list of compatible applications](#)

2



Open your authenticator app, choose **Show QR code** on this page, then use the app to scan the code. Alternatively, you can type a secret key. [Show secret key](#)

3

Type two consecutive MFA codes below

Enter a code from your virtual app below

976009

Wait 30 seconds, and enter a second code entry.

337245