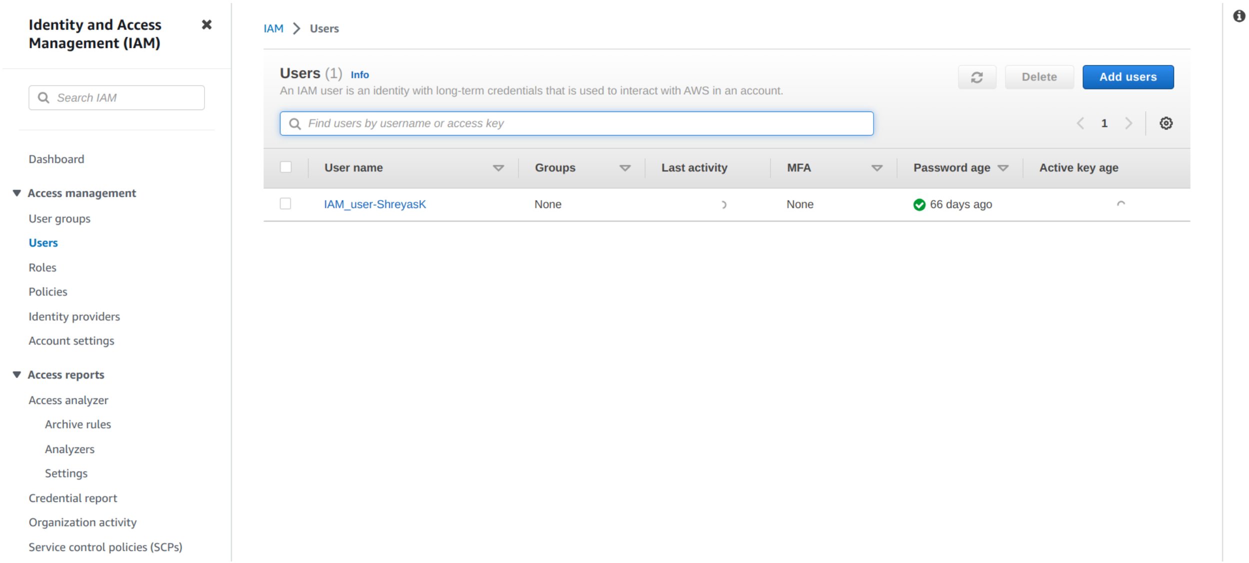
Task

Day 1 30th June 2023

1. Create an IAM user and assign S3 bucket permissions.

Step 1 - Login to the AWS management console and browse to the IAM console.

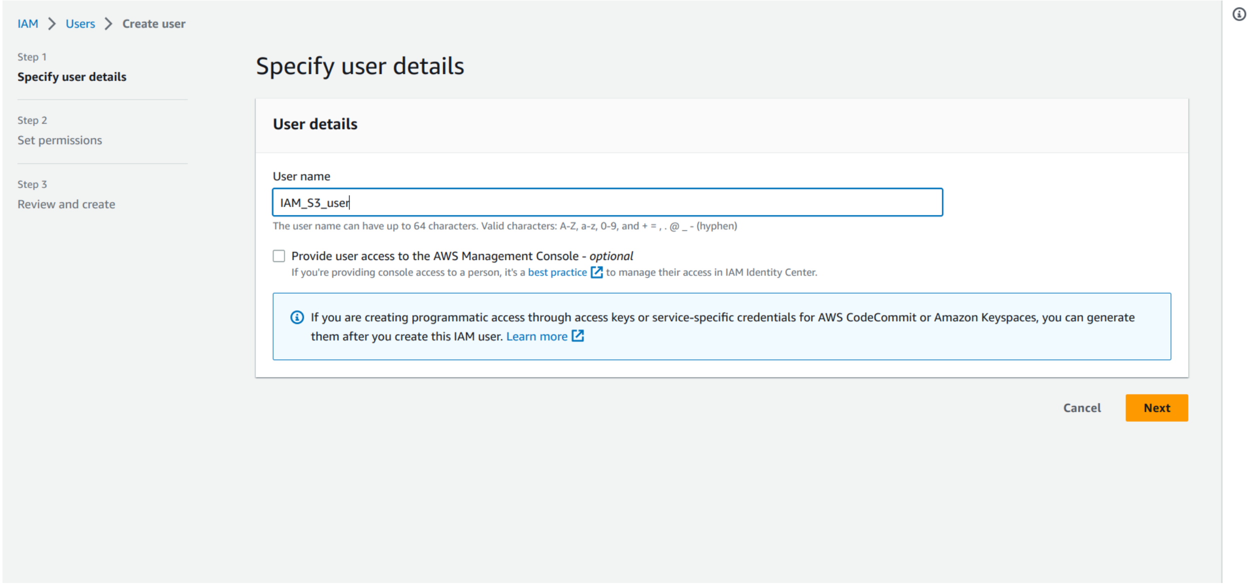


Users (1) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Groups	Last activity	MFA	Password age	Active key age
IAM_user-ShreyasK	None		None	✓ 66 days ago	

Step 2 - Click on add user button to create a new IAM user. Enter the user name for the user.



Specify user details

User details

User name: IAM_S3_user

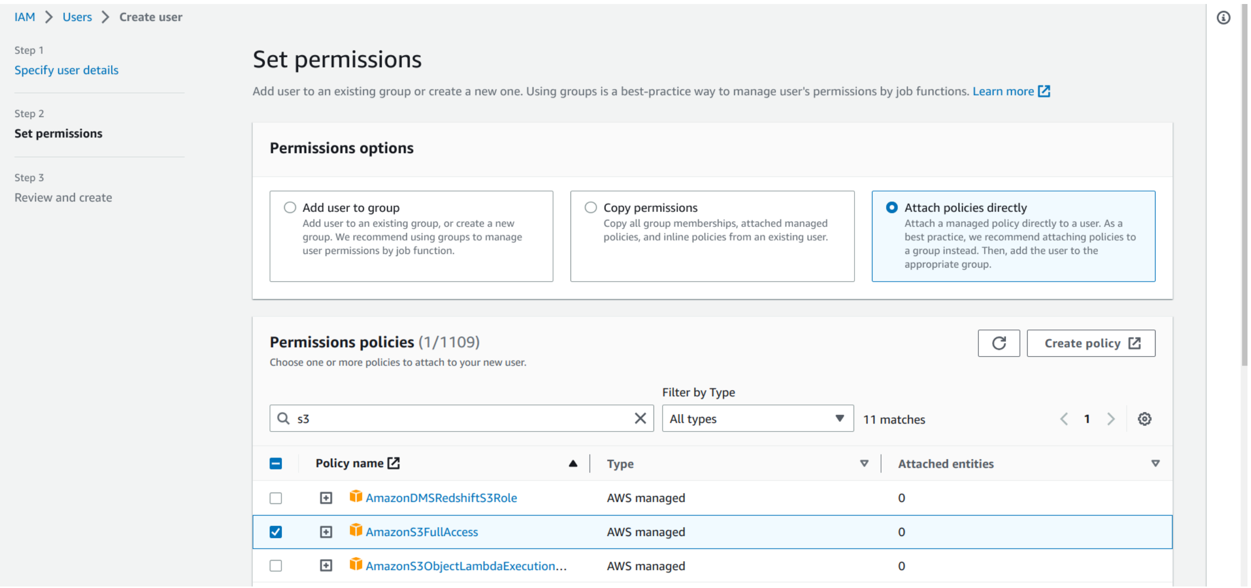
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

□ 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.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more

Step 3 - Click next and select permissions for the user. Here we need to give full S3 access permission to the IAM user.



IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

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 (1/1109)

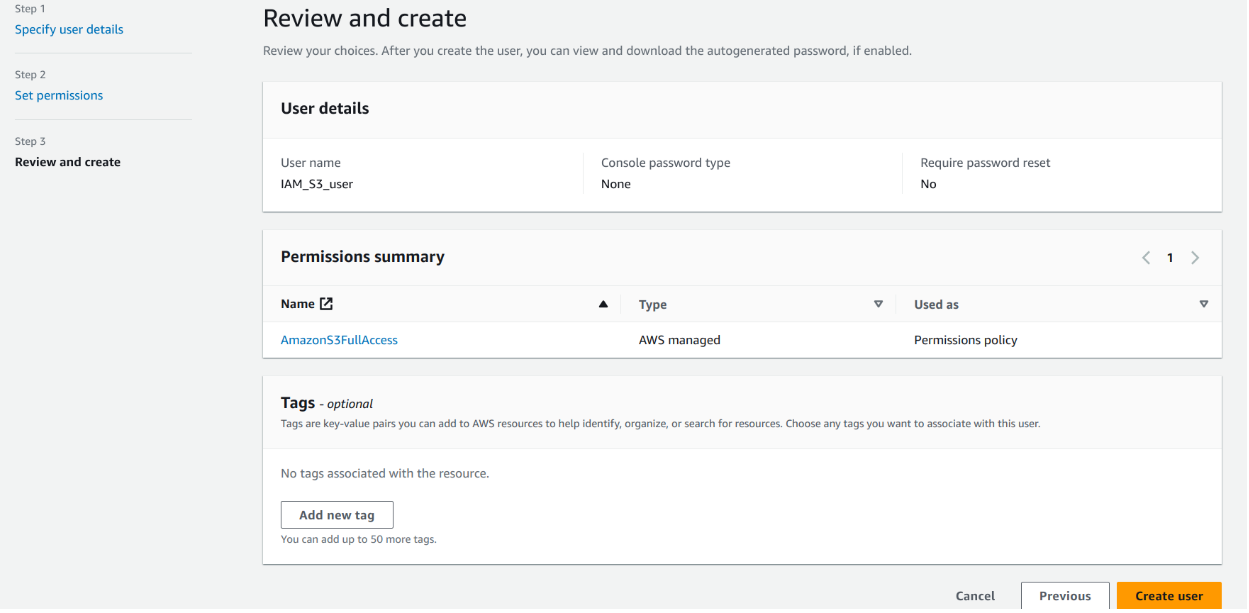
Choose one or more policies to attach to your new user.

Filter by Type

s3 | All types | 11 matches

	Policy name	Type	Attached entities
□	AmazonDMSRedshiftS3Role	AWS managed	0
☑	AmazonS3FullAccess	AWS managed	0
□	AmazonS3ObjectLambdaExecution...	AWS managed	0

Step 4 - Next review the IAM user details and click on create user button.



Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name	Console password type	Require password reset
IAM_S3_user	None	No

Permissions summary

Name	Type	Used as
AmazonS3FullAccess	AWS managed	Permissions policy

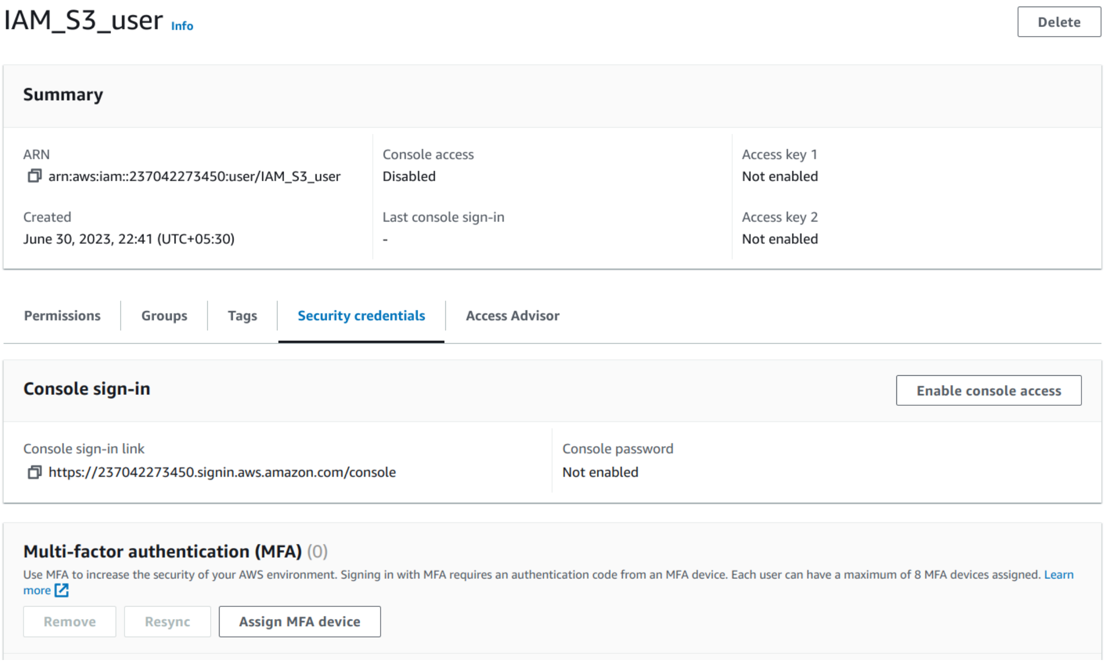
Tags - *optional*

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

You can add up to 50 more tags.

Step 5 - Click on the IAM user just created in the IAM users console. Select security credentials and click on Enable console access for the user.



IAM_S3_user

Summary

ARN arn:aws:iam::237042273450:user/IAM_S3_user	Console access Disabled	Access key 1 Not enabled
Created June 30, 2023, 22:41 (UTC+05:30)	Last console sign-in -	Access key 2 Not enabled

Console sign-in

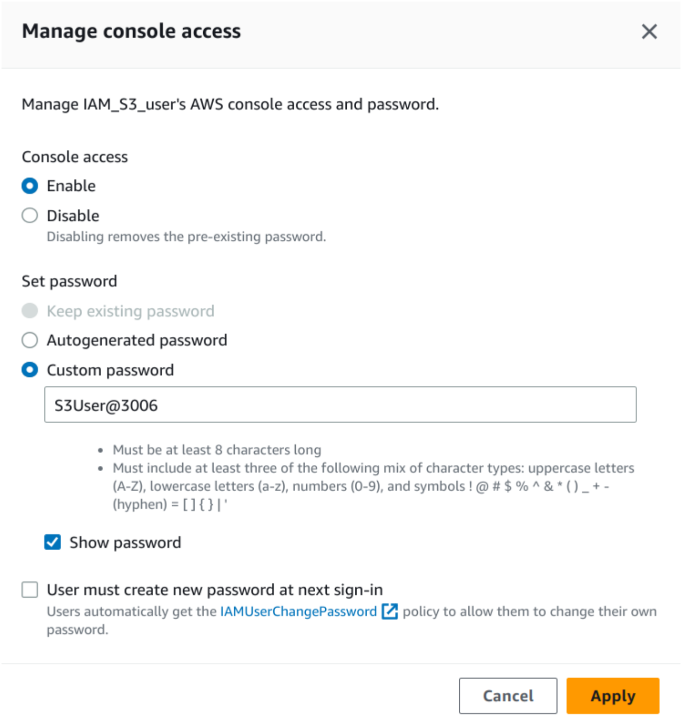
Console sign-in link
https://237042273450.signin.aws.amazon.com/console

Console password
Not enabled

Multi-factor authentication (MFA) (0)

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

Step 6 - Select the checkbox for custom password and create a new password for the user.



Manage console access

Manage IAM_S3_user's AWS console access and password.

Console access

● Enable

○ Disable
Disabling removes the pre-existing password.

Set password

○ Keep existing password

○ Autogenerated password

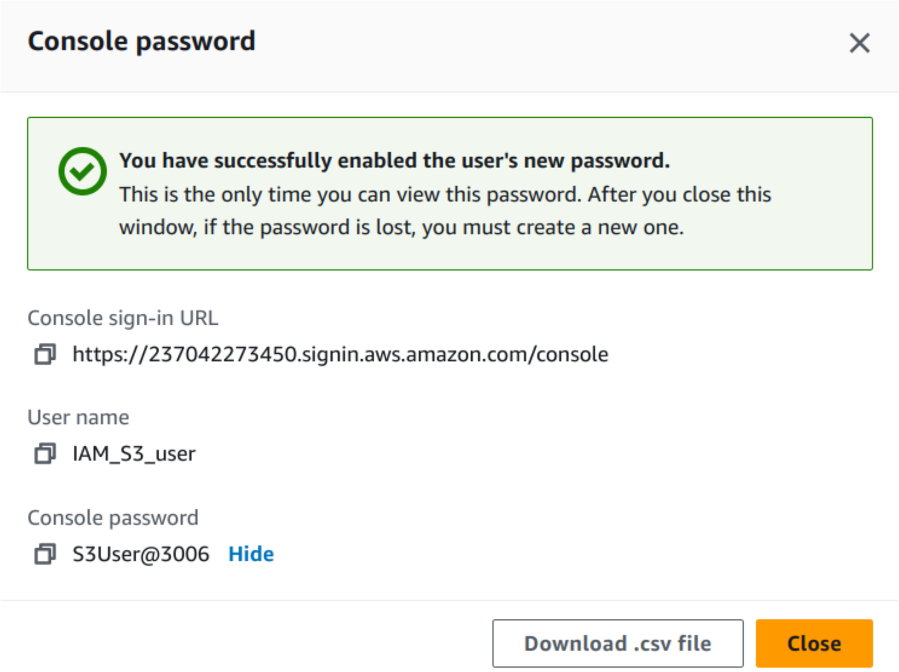
● Custom password

S3User@3006

- 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

□ User must create new password at next sign-in
Users automatically get the IAMUserChangePassword policy to allow them to change their own password.



Console password

You have successfully enabled the user's new password.
This is the only time you can view this password. After you close this window, if the password is lost, you must create a new one.

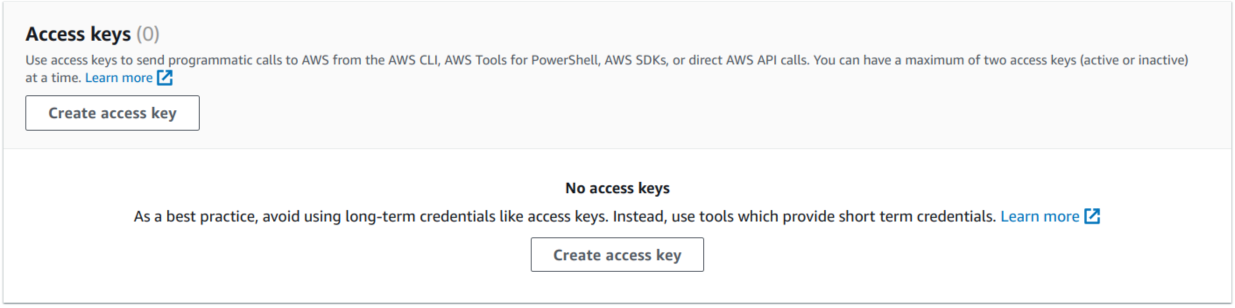
Console sign-in URL
https://237042273450.signin.aws.amazon.com/console

User name
IAM_S3_user

Console password
S3User@3006

2. Create an access key and secret access key for the user.

Step 1 - Scroll down to the access key section and click on create access key.



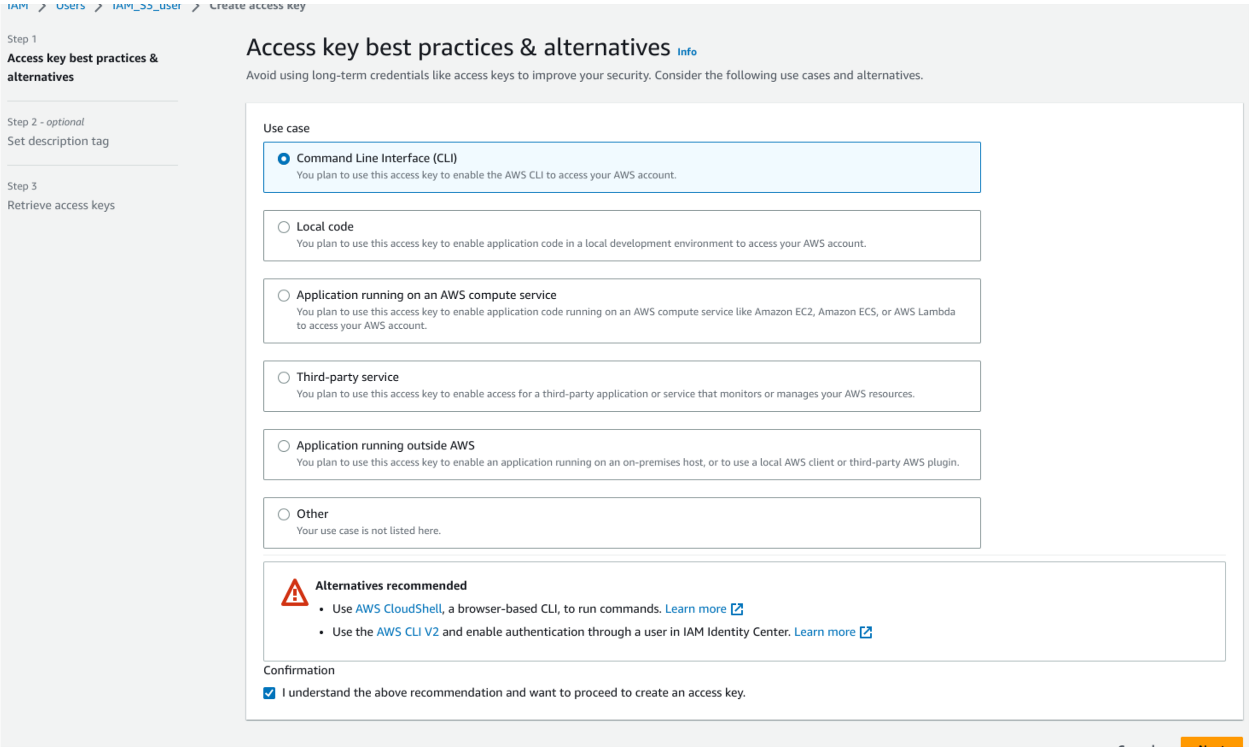
Access keys (0)

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

Step 2 - On the create access key page, select the radio box command line to use the access key to connect AWS CLI to the AWS account. Click on next.



IAM > Users > IAM_S3_user > 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 Info

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 enable an application running on an on-premises host, or to use a local AWS client or third-party AWS plugin.

○ Other
Your use case is not listed here.

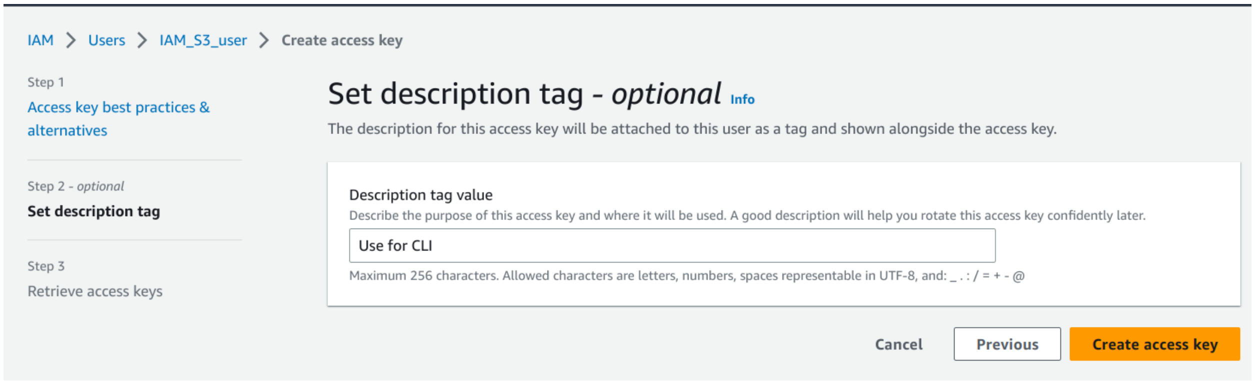
Alternatives recommended

- 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.

Step 3 - Enter a short description for the access key. And click on create access key.



Set description tag - *optional*

The description for this access key will be attached to this user as a tag and shown alongside the access key.

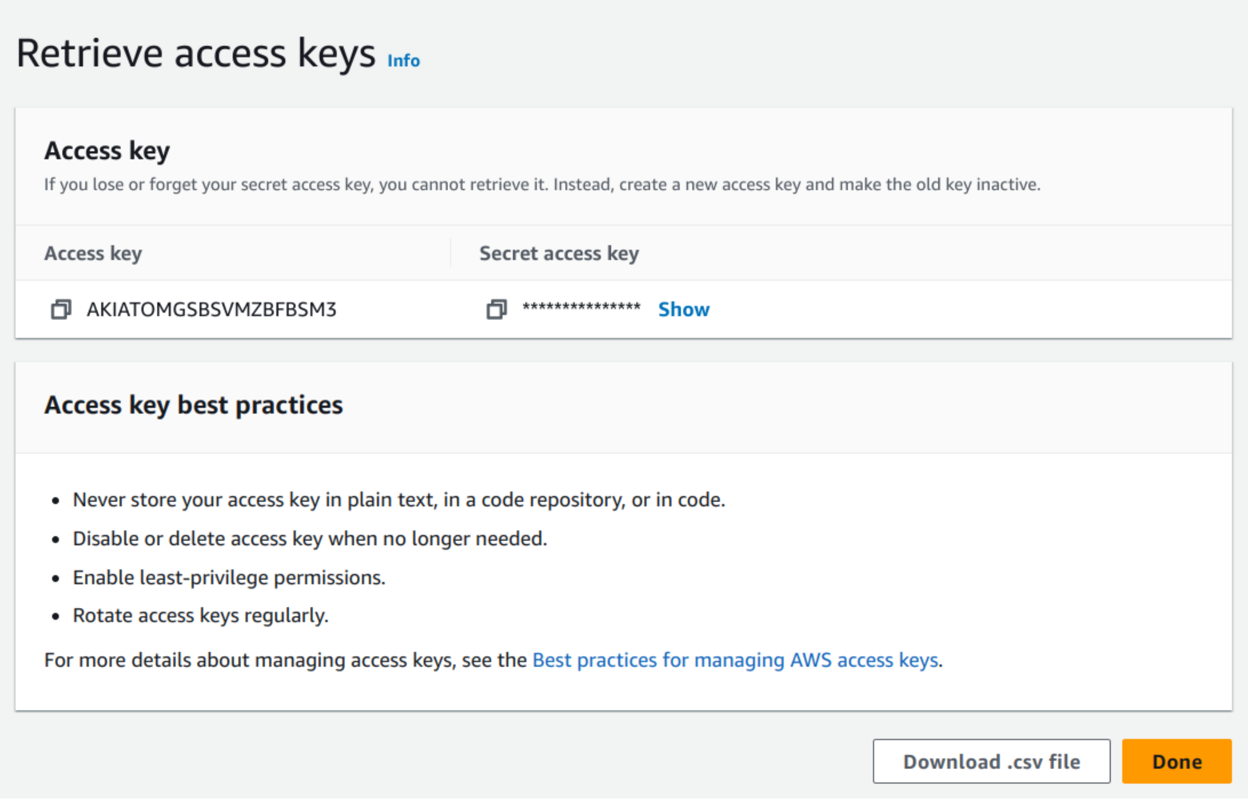
Description tag value

Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

Use for CLI

Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . : / = + - @

Step 4 - Access key and Secret access key is created. These can be used to connect AWS account to AWS CLI.



Retrieve access keys

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
AKIATOMGSBSVMZBFBSM3	***************

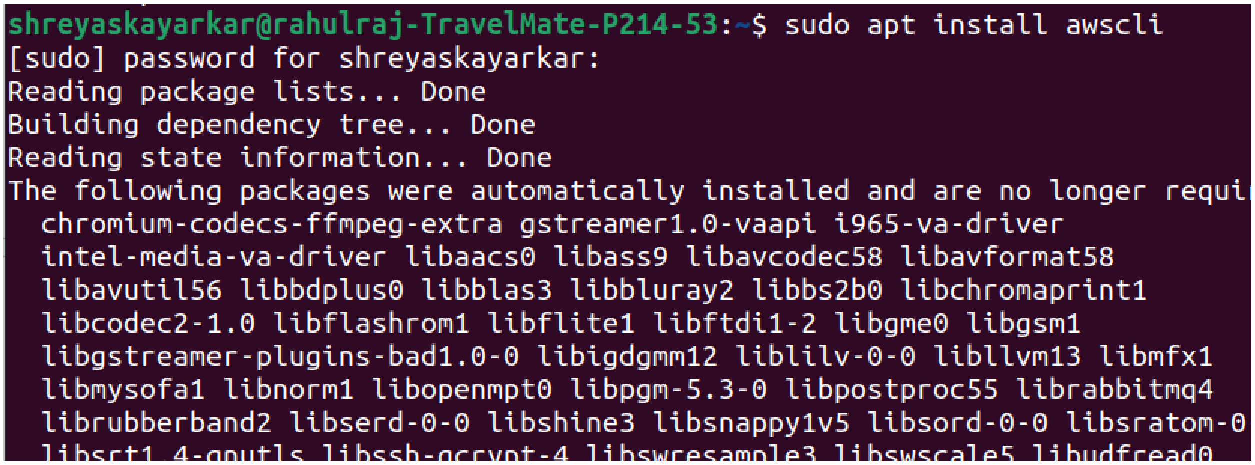
Access key best practices

- 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.

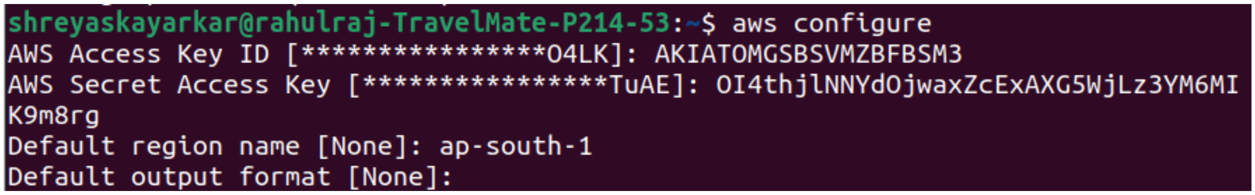
3. Configure AWS CLI in local machine

Step 1 - Install “awscli” with apt install command in terminal.



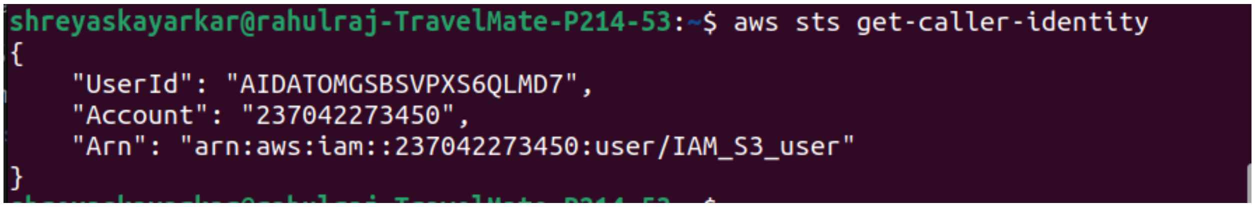
```
shreyaskayarkar@rahulraj-TravelMate-P214-53:~$ sudo apt install awscli
[sudo] password for shreyaskayarkar:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer requi
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver
  intel-media-va-driver libaacs0 libass9 libavcodec58 libavformat58
  libavutil56 libbdplus0 libblas3 libbluray2 libbs2b0 libchromaprint1
  libcodec2-1.0 libflashrom1 libflite1 libftdi1-2 libgme0 libgsm1
  libgstreamer-plugins-bad1.0-0 libigdgmm12 liblilv-0-0 libllvm13 libmfx1
  libmysofa1 libnorm1 libopenmpt0 libpgm-5.3-0 libpostproc55 librabbitmq4
  librubberband2 libserd-0-0 libshine3 libsnappy1v5 libsord-0-0 libsratom-0
  libsrt1.4-gnutls libssh-gcrypt-4 libswresample3 libswscale5 libudfread0
```

Step 2 - Login to the IAM user using its access key and secret access key. Run command “aws configure” and enter the details.



```
shreyaskayarkar@rahulraj-TravelMate-P214-53:~$ aws configure
AWS Access Key ID [****************O4LK]: AKIATOMGSBSVMZBFBSM3
AWS Secret Access Key [****************TuAE]: OI4thjlNNYdOjwaxZcExAXG5WjLz3YM6MI
K9m8rg
Default region name [None]: ap-south-1
Default output format [None]:
```

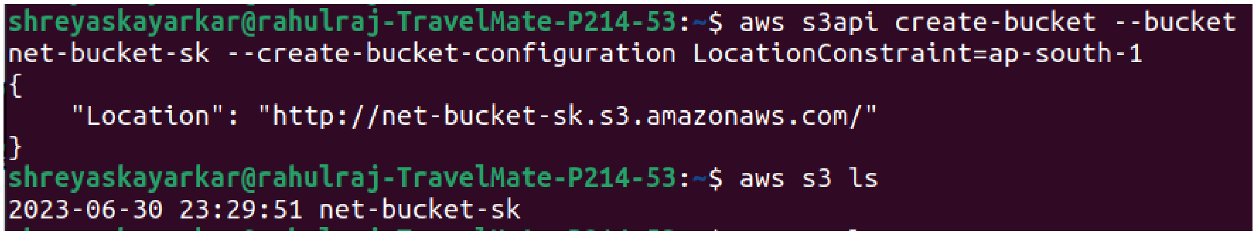
Step 3 - Check if the user is logged in. Run “sts get-caller-identity” command which returns the userId, account Id, and ARN of the caller account.



```
shreyaskayarkar@rahulraj-TravelMate-P214-53:~$ aws sts get-caller-identity
{
    "UserId": "AIDATOMGSBSVPXS6QLMD7",
    "Account": "237042273450",
    "Arn": "arn:aws:iam::237042273450:user/IAM_S3_user"
}
```

4. Create S3 bucket in mumbai region.

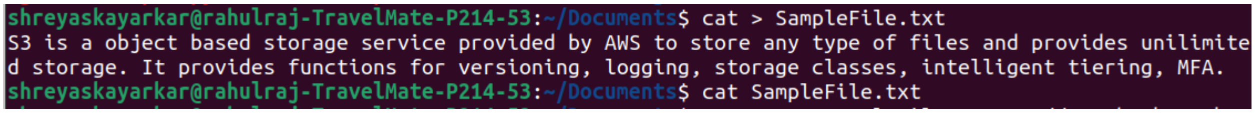
Step 1 - Run the “create-bucket” command to create a bucket. A location constraint has to be specified if the bucket is being created in a region other than us-region-1.



```
shreyaskayarkar@rahulraj-TravelMate-P214-53:~$ aws s3api create-bucket --bucket
net-bucket-sk --create-bucket-configuration LocationConstraint=ap-south-1
{
    "Location": "http://net-bucket-sk.s3.amazonaws.com/"
}
shreyaskayarkar@rahulraj-TravelMate-P214-53:~$ aws s3 ls
2023-06-30 23:29:51 net-bucket-sk
```

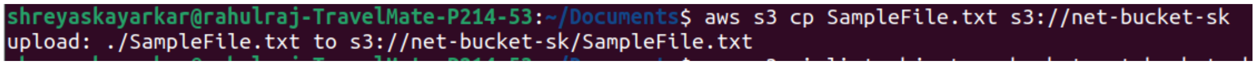
5. Upload a file to the S3 bucket.

Step 1 - Create a sample file to upload it to S3.



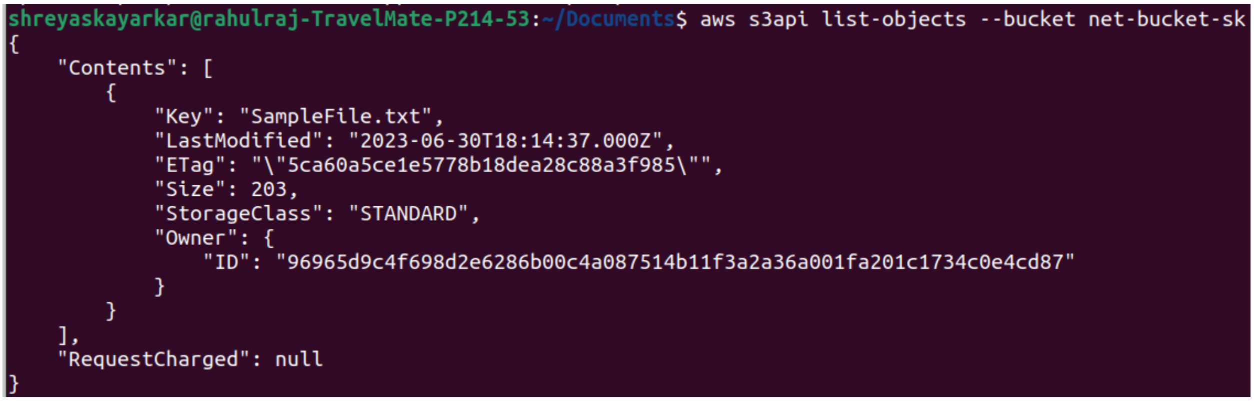
```
shreyaskayarkar@rahulraj-TravelMate-P214-53:~/Documents$ cat > SampleFile.txt
S3 is a object based storage service provided by AWS to store any type of files and provides unilimite
d storage. It provides functions for versioning, logging, storage classes, intelligent tiering, MFA.
shreyaskayarkar@rahulraj-TravelMate-P214-53:~/Documents$ cat SampleFile.txt
```

Step 2 - Run “cp” command to copy local file to S3 bucket.



```
shreyaskayarkar@rahulraj-TravelMate-P214-53:~/Documents$ aws s3 cp SampleFile.txt s3://net-bucket-sk
upload: ./SampleFile.txt to s3://net-bucket-sk/SampleFile.txt
```

Step 3 - List objects in the S3 bucket to confirm if the object was uploaded to the bucket successfully. Run “list-objects” command on the specified bucket.



```
shreyaskayarkar@rahulraj-TravelMate-P214-53:~/Documents$ aws s3api list-objects --bucket net-bucket-sk
{
    "Contents": [
        {
            "Key": "SampleFile.txt",
            "LastModified": "2023-06-30T18:14:37.000Z",
            "ETag": "\"5ca60a5ce1e5778b18dea28c88a3f985\"",
            "Size": 203,
            "StorageClass": "STANDARD",
            "Owner": {
                "ID": "96965d9c4f698d2e6286b00c4a087514b11f3a2a36a001fa201c1734c0e4cd87"
            }
        }
    ],
    "RequestCharged": null
}
```