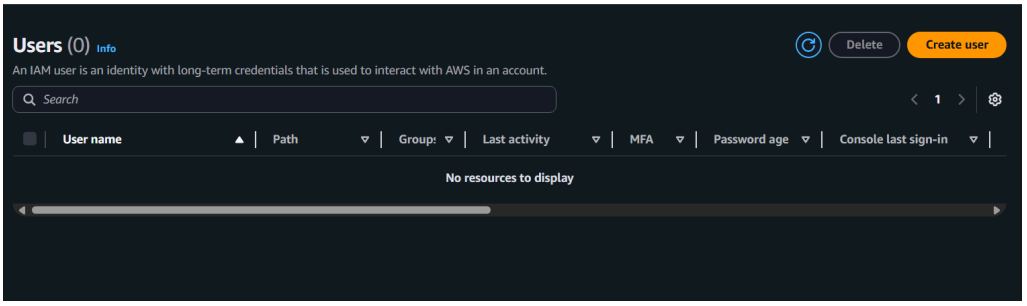

# Create an IAM user from root and connect CLI with root and IAM User

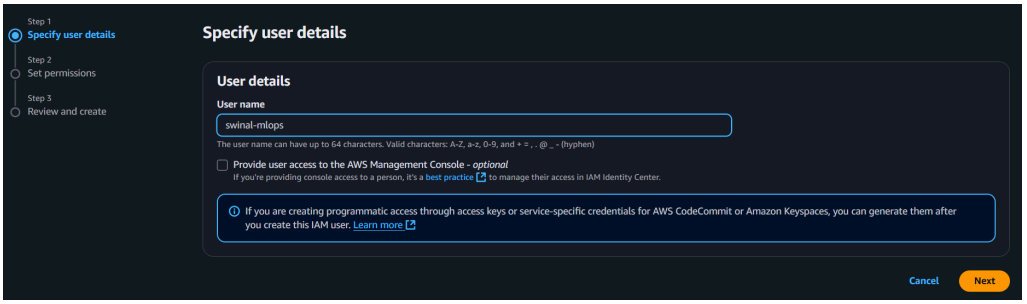
| | |
|---|---|
| Date | @September 8, 2025 |
| Multi-select | IAM Lab-1 |
| Status | Done |

## Create IAM User

- To create an IAM user, go to **IAM**, then under **Access Management**, click on **Users**, and select **Create user**.




- Provide the user details under **User details**. Enter the **User name** (in our case, `swinal-mlops`), then click **Next**.



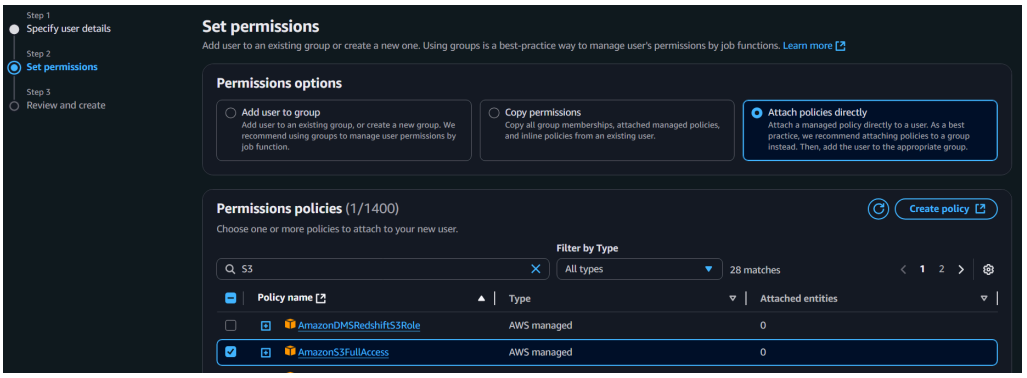

- Next, set permissions for the user. You have three options:
  - **Add user to group**

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

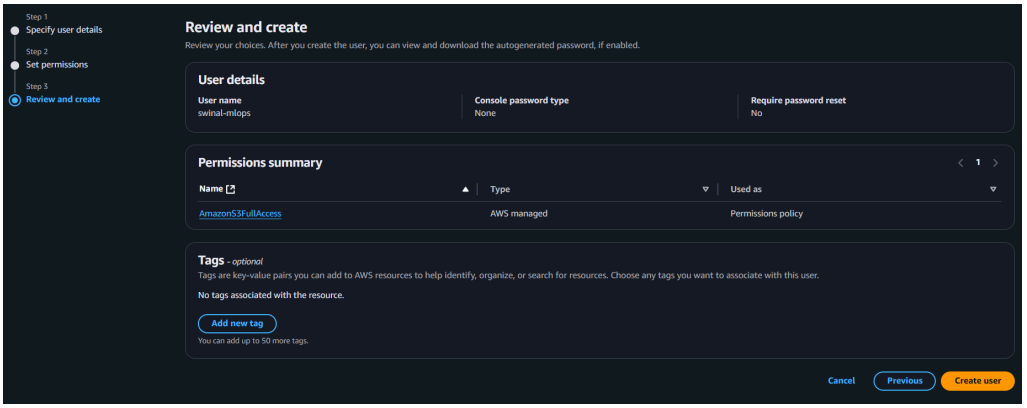
    Copy all group memberships, attached managed policies, and inline policies from an existing user.
  - **Attach policies directly**

    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.

  Here, we are creating a new policy, so we select **Attach policies directly** and attach the policy `AmazonS3FullAccess`. Then click **Next**.

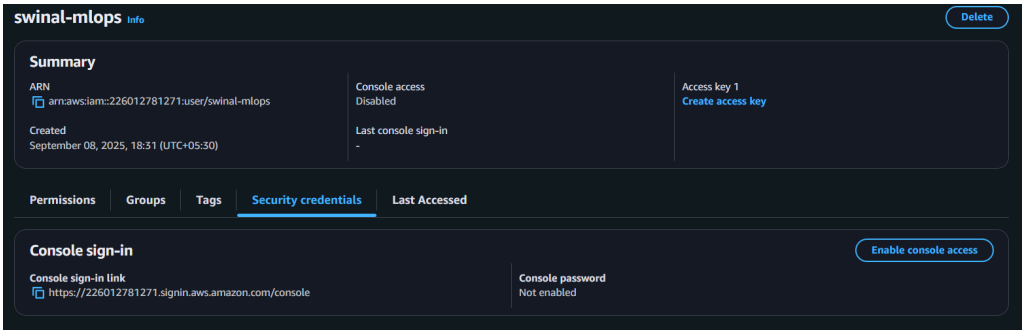



- Review the details, then click **Create user**.

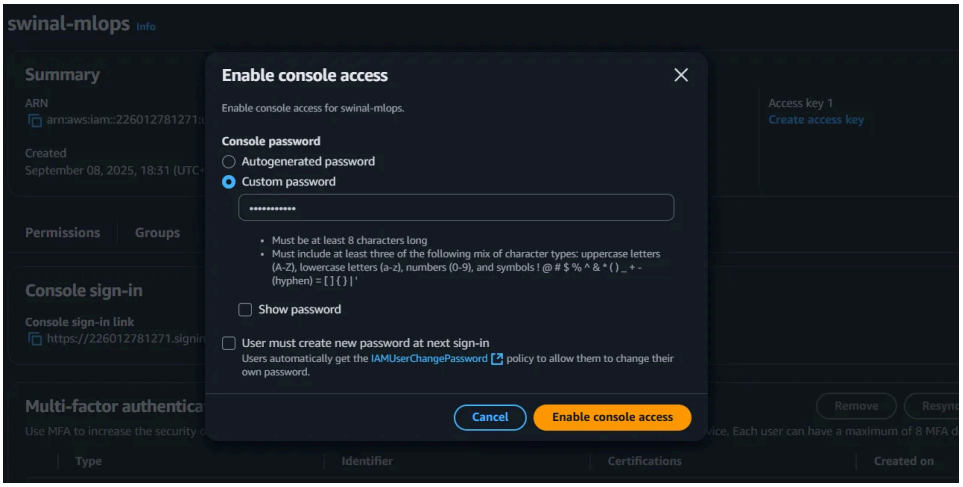



## Users Console Access

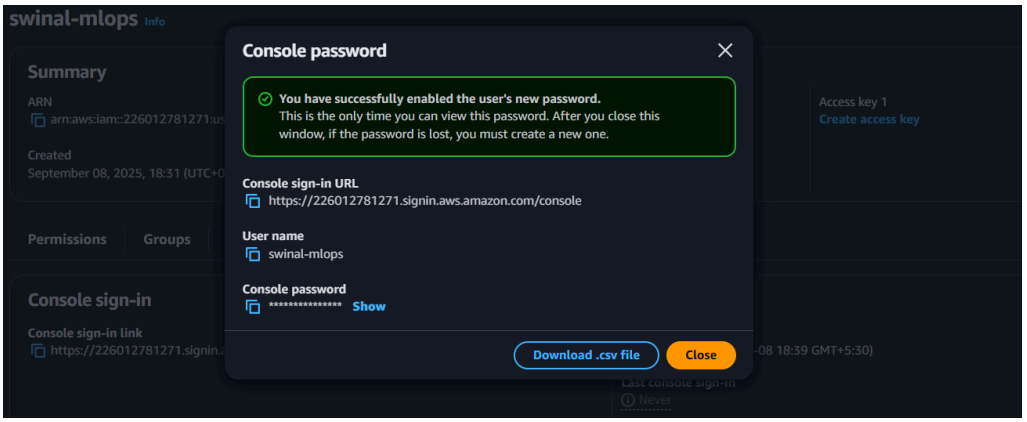
- Open the user `swinal-mlops`, go to the **Security Credentials** tab, and click **Enable console access**.



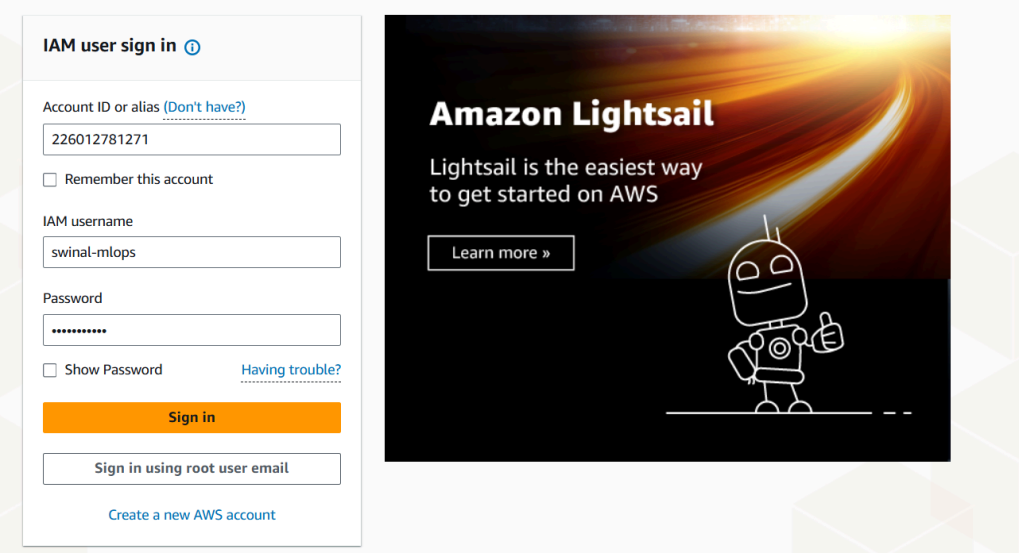

- Set a custom password for the user `swinal-mlops`, then click **Enable console access**.



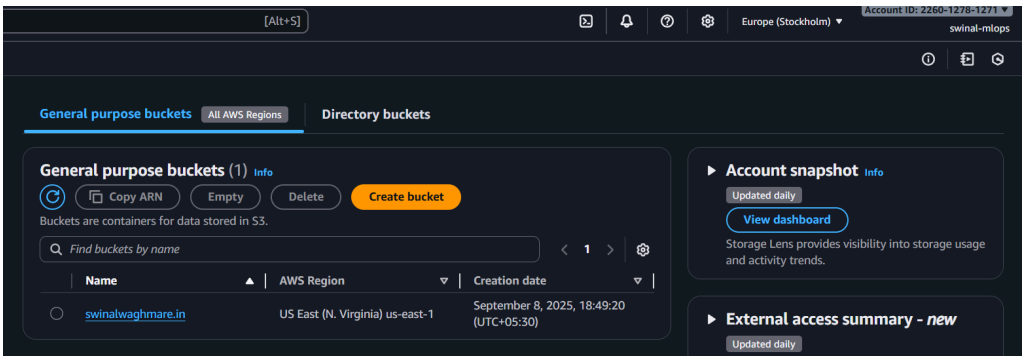

- Download the user credentials as a `.csv` file.



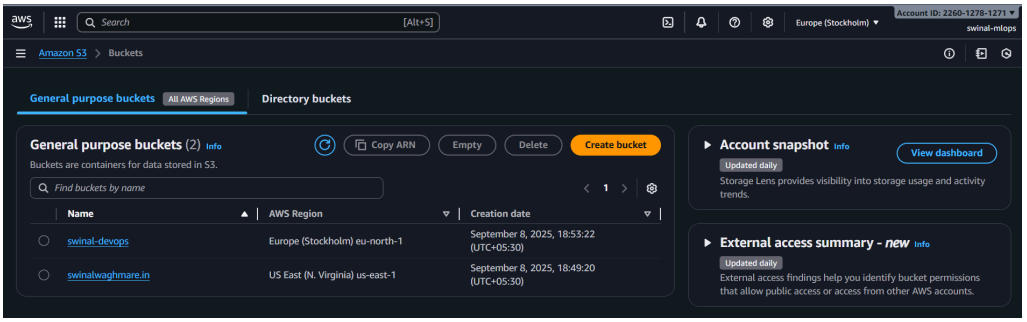

- Now, sign in as the IAM user `swinal-mlops`. Provide the **Account ID**, **IAM username**, and **Password**.
  - **Account ID** → URL account number
  - **IAM username** → `swinal-mlops`
  - **Password** → the custom password

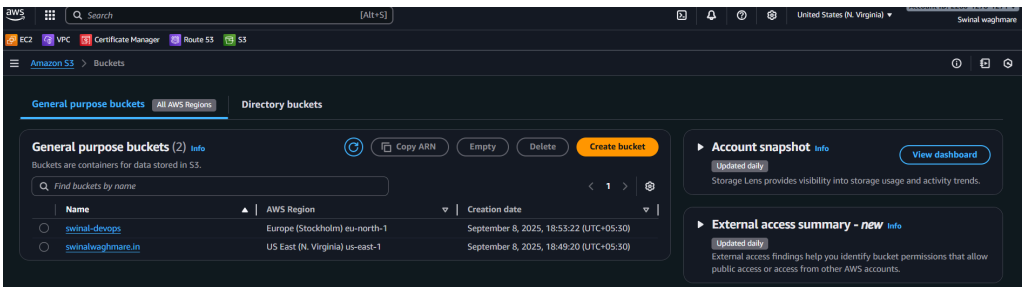



- Since only S3 access is granted, the user can access only **Amazon S3**.




- Any changes made by the user will also reflect in the root account. For example, if the IAM user creates a new bucket ( `swinal-devops` ), it will be visible to the root user.



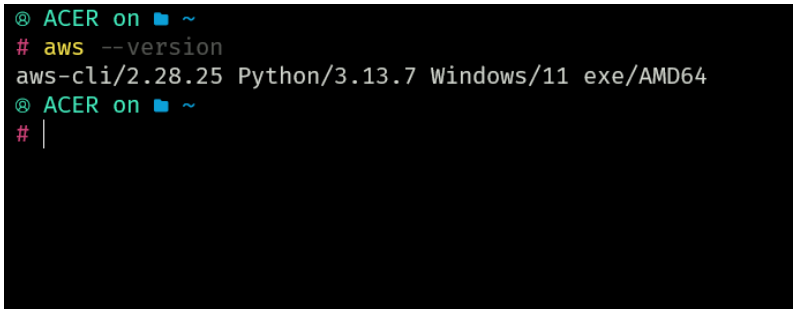

## Create Credentials for Root User and IAM User (CLI Access)

- To access AWS through the **Command Line Interface (CLI)**, first install the AWS CLI:

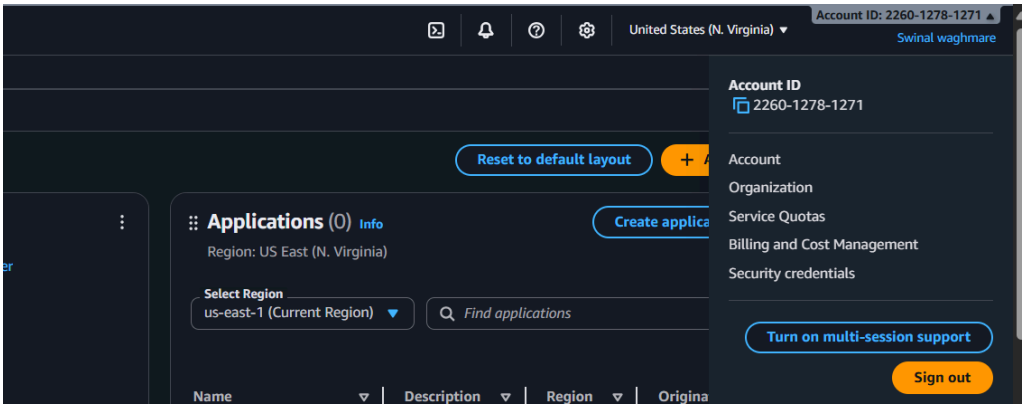
> Installing or updating to the latest version of the AWS CLI - AWS Command Line Interface
> The AWS CLI is an open source tool built using the AWS SDK for Python (Boto) that provides commands for interacting with AWS services. With minimal configuration, you can start using all of the functionality provided by the AWS Management Console from your favorite terminal program.
> https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

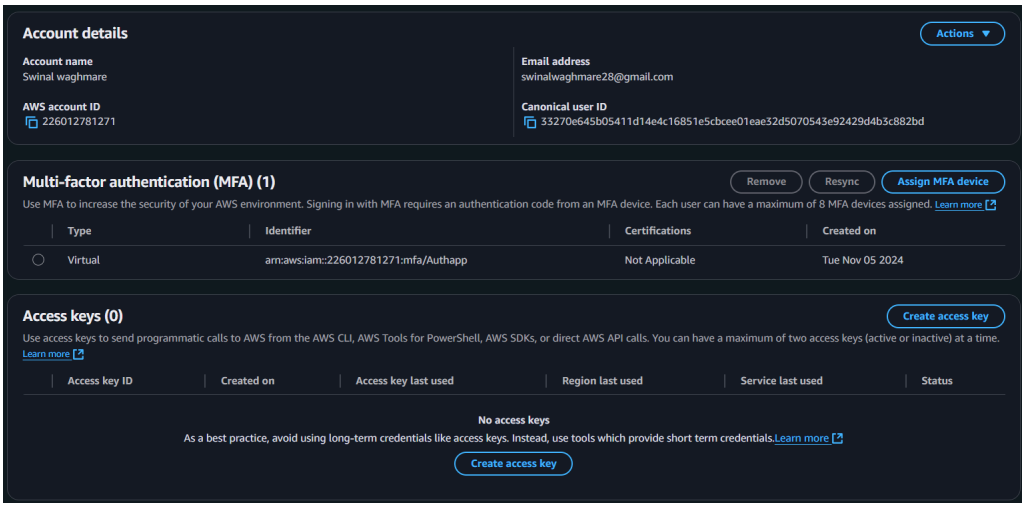
- Verify installation:

```
aws --version
```



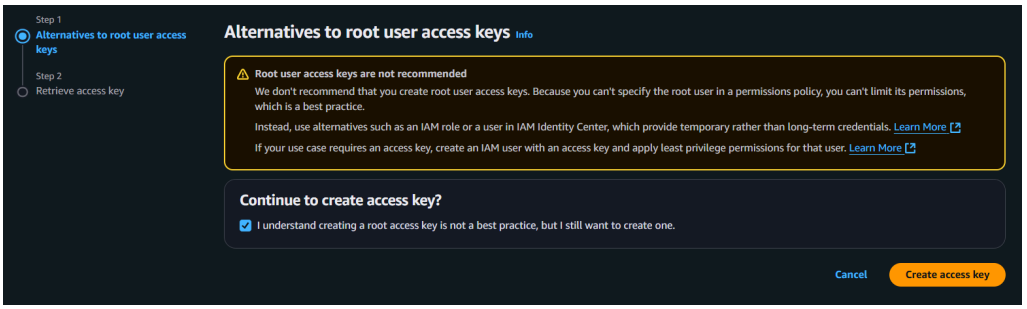

- First, create credentials for the **root user**.
  Go to the **root account**, click on your account ID, then select **Security credentials**.



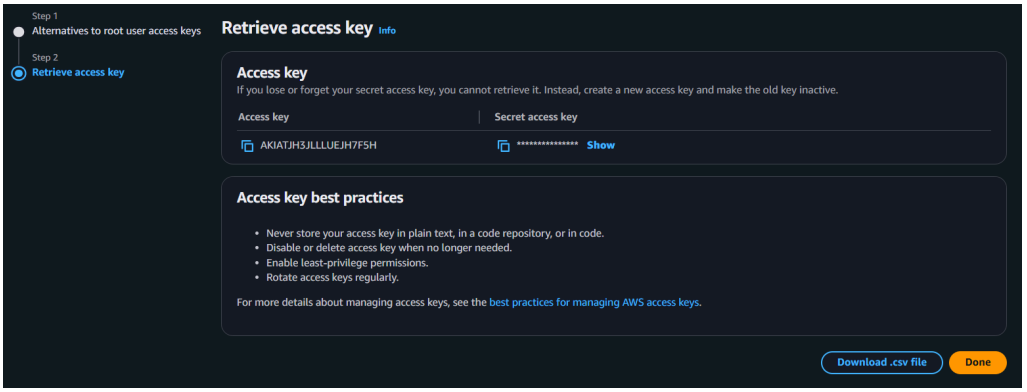



- Create an **Access Key** and **Secret Key** by selecting **Create access key**.

> **Root user access keys are not recommended**
>
> We don't recommend that you create root user access keys. Because you can't specify the root user in a permissions policy, you can't limit its permissions, which is a best practice.
>
> Instead, use alternatives such as an IAM role or a user in IAM Identity Center, which provide temporary rather than long-term credentials. Learn More
>
> If your use case requires an access key, create an IAM user with an access key and apply least privilege permissions for that user. Learn More




- Download the keys, then click **Done**.




> **Note**: You can only create **two access keys per user** (applies to both root and IAM users).

- Configure the AWS CLI with root user credentials:

```
aws configure
```

```
AWS Access Key ID [**********QUQS]: AKIATJH3JLLLUEJH7F5H
AWS Secret Access Key [**********kQBv]: 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
Default region name [ap-southeast-2]: ap-south-1
Default output format [json]:
```

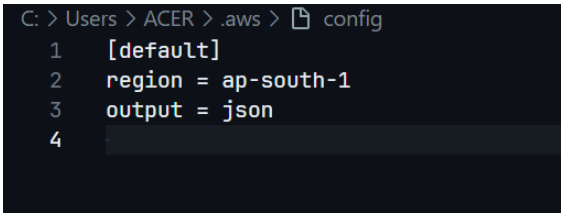
```
® ACER on ■ ~
# aws --version
aws-cli/2.28.25 Python/3.13.7 Windows/11 exe/AMD64
® ACER on ■ ~
# aws configure
AWS Access Key ID [****************QUQS]: AKIATJH3JLLLUEJH7F5H
AWS Secret Access Key [****************kQBv]: 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
Default region name [ap-southeast-2]: ap-south-1
Default output format [json]:
® ACER on ■ ~
```

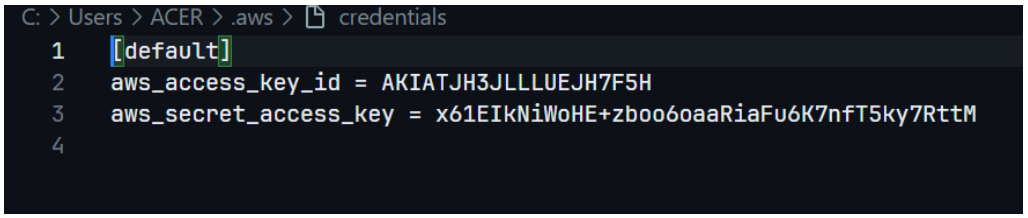
- Verify S3 access (root user has admin privileges):

```
aws s3 ls
```

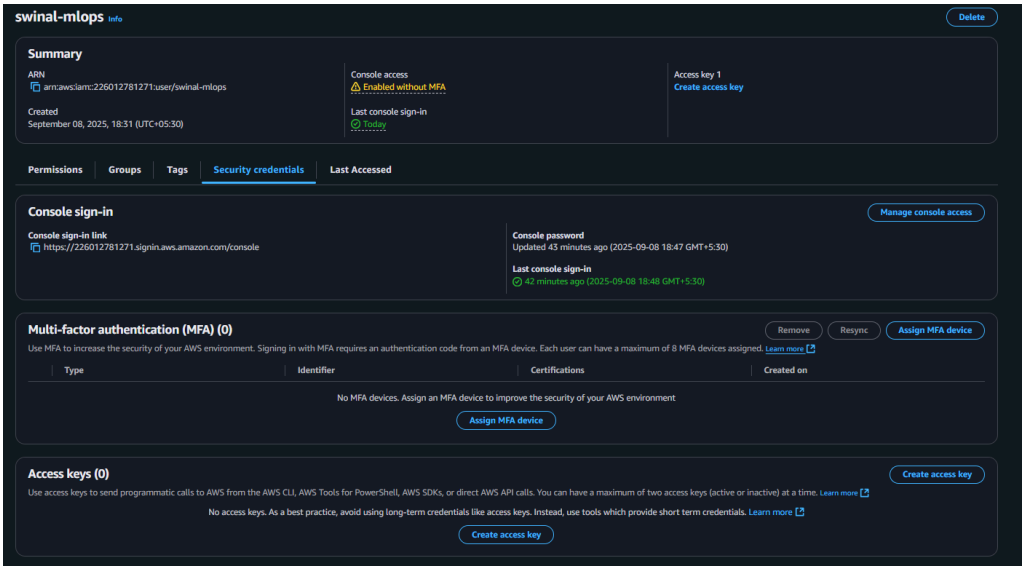
```
® ACER on ■ ~
# aws --version
aws-cli/2.28.25 Python/3.13.7 Windows/11 exe/AMD64
® ACER on ■ ~
# aws configure
AWS Access Key ID [****************QUQS]: AKIATJH3JLLLUEJH7F5H
AWS Secret Access Key [****************kQBv]: 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
Default region name [ap-southeast-2]: ap-south-1
Default output format [json]:
® ACER on ■ ~
# aws s3 ls
2025-09-08 18:53:24 swinal-devops
2025-09-08 18:49:21 swinalwaghmare.in
® ACER on ■ ~
```

- AWS CLI credentials are stored in the `.aws` folder:
  - `config` → region and output format
  - `credentials` → Access Key ID and Secret Access Key

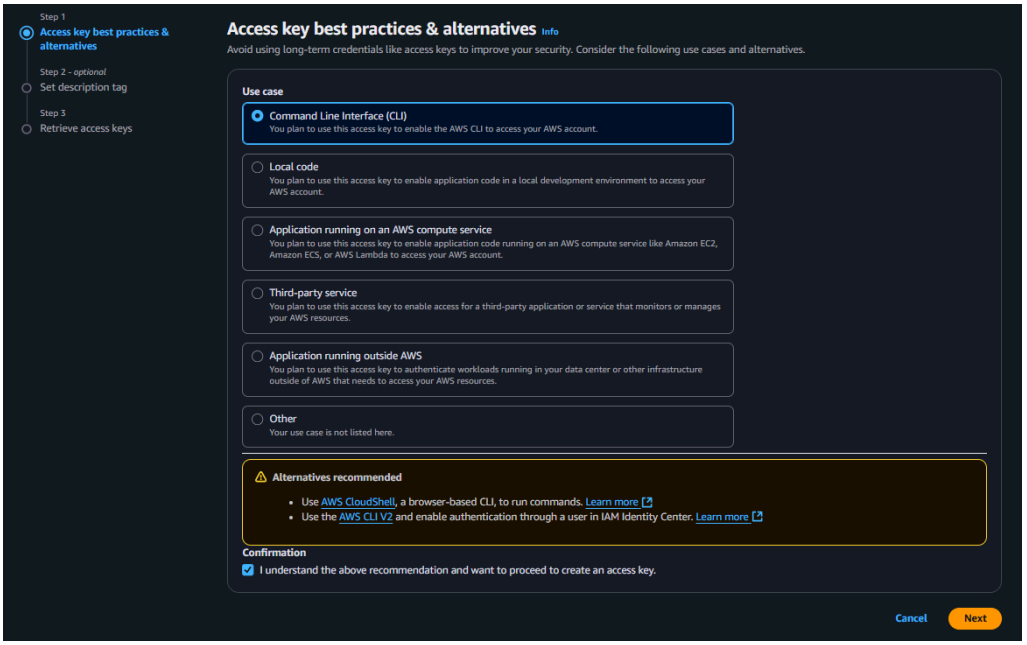


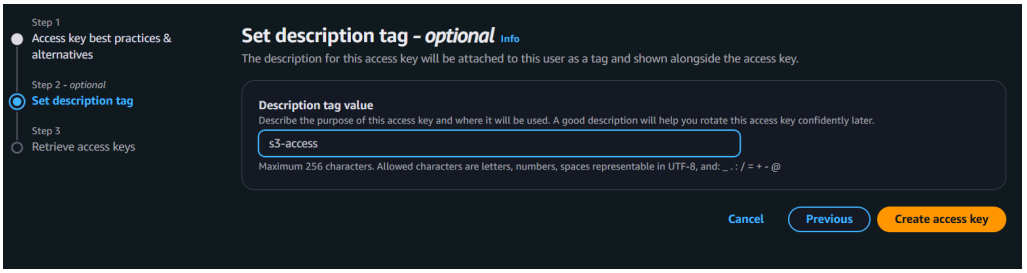

- Now create credentials for the IAM user `swinal-mlops` Go to the IAM user → **Security credentials** → **Create access key**



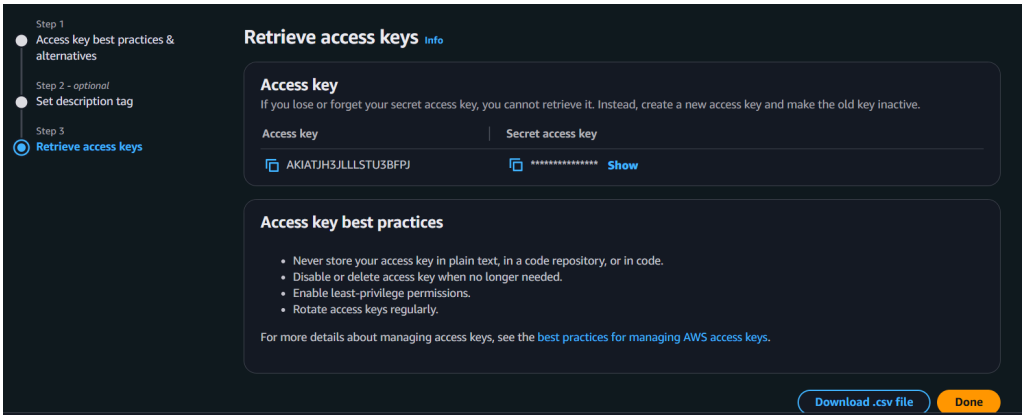

- Choose **Command Line Interface (CLI)** as the use case, then click **Next**.




- Add a description, e.g., `s3-access`, then click **Create access key**.




- Download the keys in `.csv` format and click **Done**.




- Configure IAM user CLI profile (to avoid overwriting root credentials):

```
aws configure --profile swinal-mlops
```

```
® ACER on ■ ~
# aws configure --profile swinal-mlops
AWS Access Key ID [None]: AKIATJH3JLLLSTU3BFPJ
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]:
® ACER on ■ ~
#
```

- Now the `.aws/credentials` file contains both users:

```
C: > Users > ACER > .aws > credentials
1   [default]
2   aws_access_key_id = AKIATJH3JLLLUEJH7F5H
3   aws_secret_access_key = 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
4   [swinal-mlops]
5   aws_access_key_id = AKIATJH3JLLLSTU3BFPJ
6   aws_secret_access_key = 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
7
```

- Verify IAM user permissions (only S3 access):

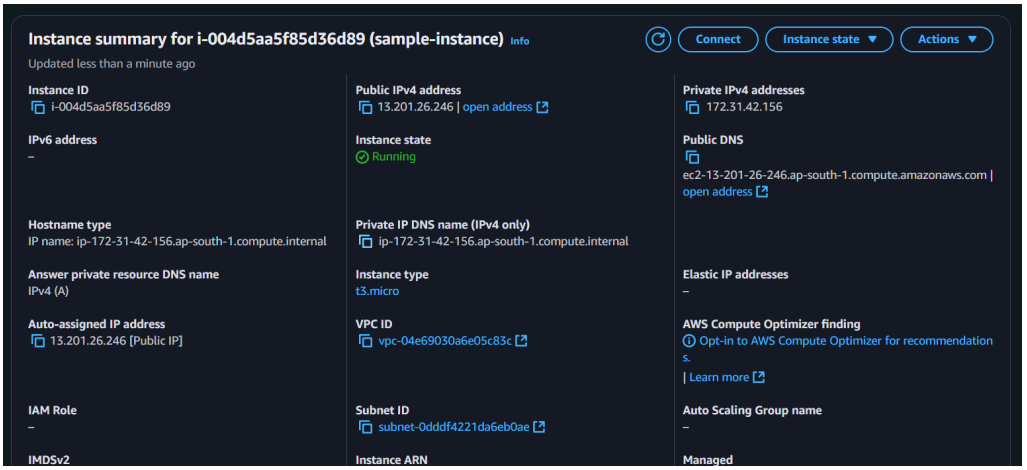
```
aws s3 ls --profile swinal-mlops
```

```
® ACER on ■ ~
# aws s3 ls --profile swinal-mlops
2025-09-08 18:53:24 swinal-devops
2025-09-08 18:49:21 swinalwaghmare.in
® ACER on ■ ~
#
```

- Test EC2 access with **root user**:

```
aws ec2 describe-instances
```

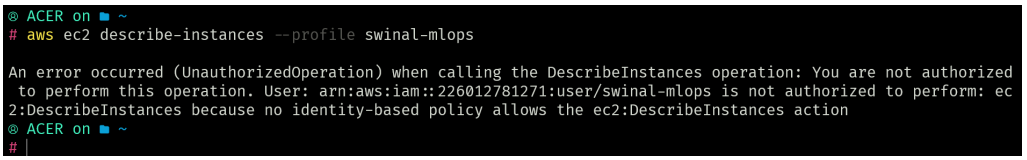
```
# aws ec2 describe-instances
{
   "Reservations": [
      {
         "ReservationId": "r-0980349ac185e82f6",
         "OwnerId": "226012781271",
         "Groups": [],
         "Instances": [
            {
               "Architecture": "x86_64",
               "BlockDeviceMappings": [
                  {
                     "DeviceName": "/dev/xvda",
                     "Ebs": {
                        "AttachTime": "2025-09-08T14:18:00+00:00",
                        "DeleteOnTermination": true,
                        "Status": "attached",
                        "VolumeId": "vol-033432b0b42e1d8f6"
                     }
                  }
               ],
               "ClientToken": "b74e4ac3-bdbd-498d-81ed-1b0f3e054ffa",
               "EbsOptimized": true,
               "EnaSupport": true,
               "Hypervisor": "xen",
               "NetworkInterfaces": [
                  {
                     "Association": {
                        "IpOwnerId": "amazon",
                        "PublicDnsName": "ec2-13-201-26-246.ap-south-1.compute.amazonaws.com",
                        "PublicIp": "13.201.26.246"
                     },
                     "Attachment": {
                        "AttachTime": "2025-09-08T14:18:00+00:00",
```


- Test EC2 access with **IAM user (** `swinal-mlops` **)**:

```
aws ec2 describe-instances --profile swinal-mlops
```

→ Error (since IAM user only has S3 access).




✅ **Lab Completed**