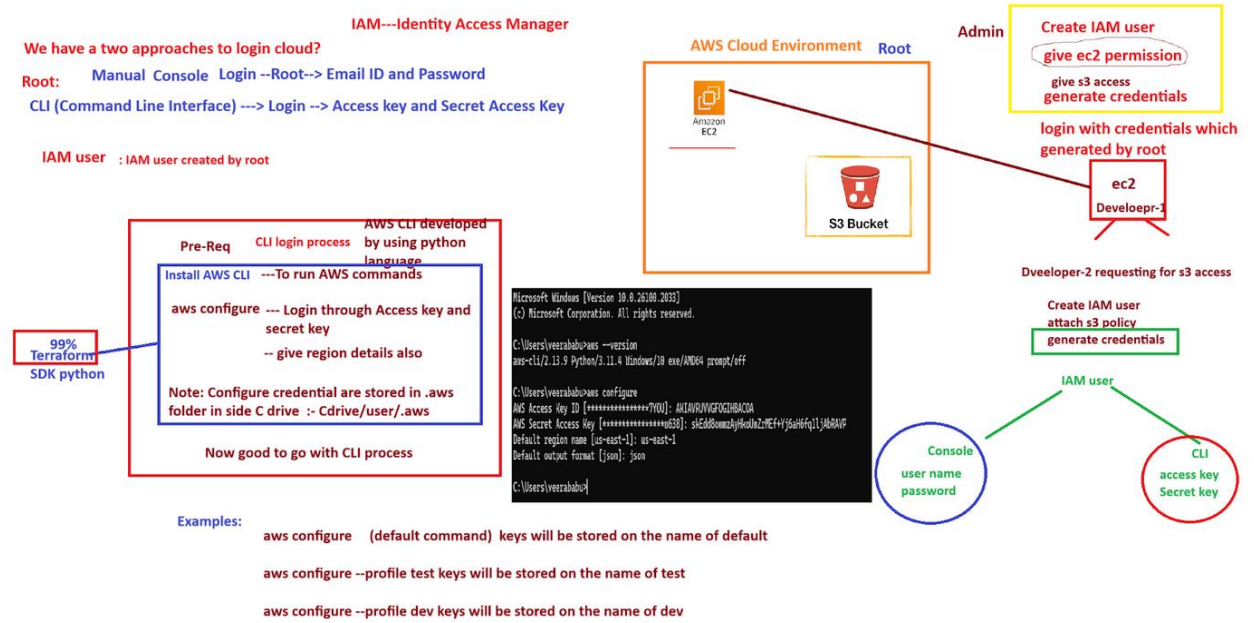
Table of Contents

AWS CLI

Class Room Discussion



Examples:

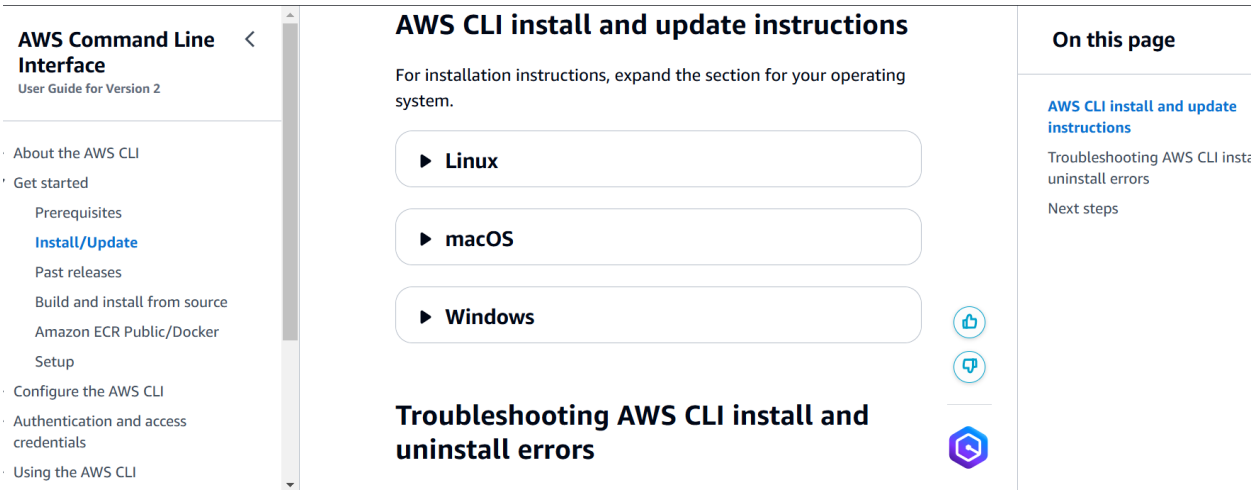
aws configure (default command) keys will be stored on the name of default

aws configure --profile test keys will be stored on the name of test

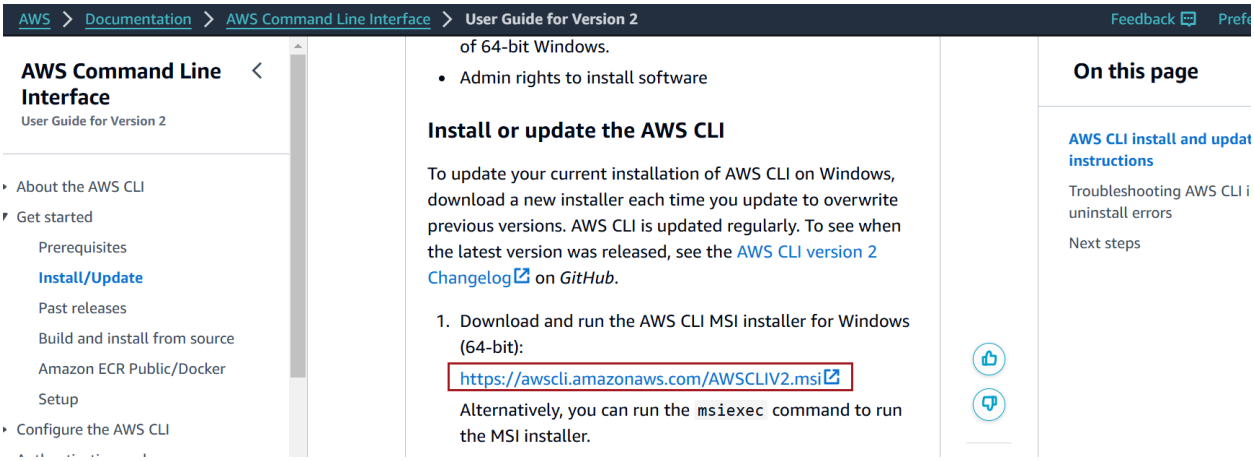
aws configure --profile dev keys will be stored on the name of dev

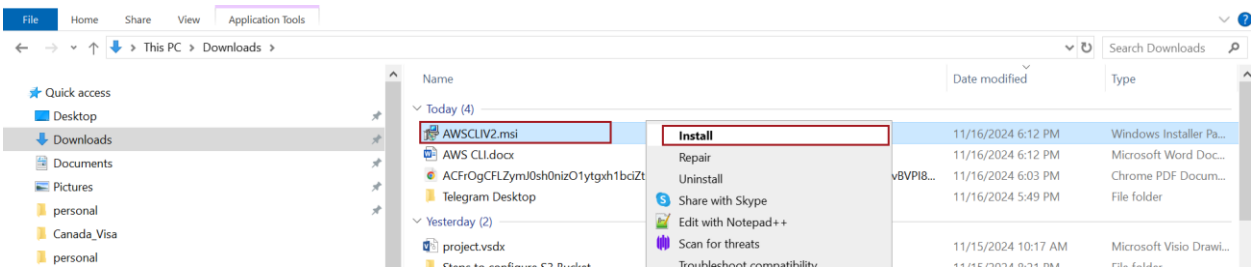
Screenshots for Windows/Mac/Linux Installation

Visit the official AWS CLI download page: AWS CLI Installation



Download and install the appropriate version for your operating system

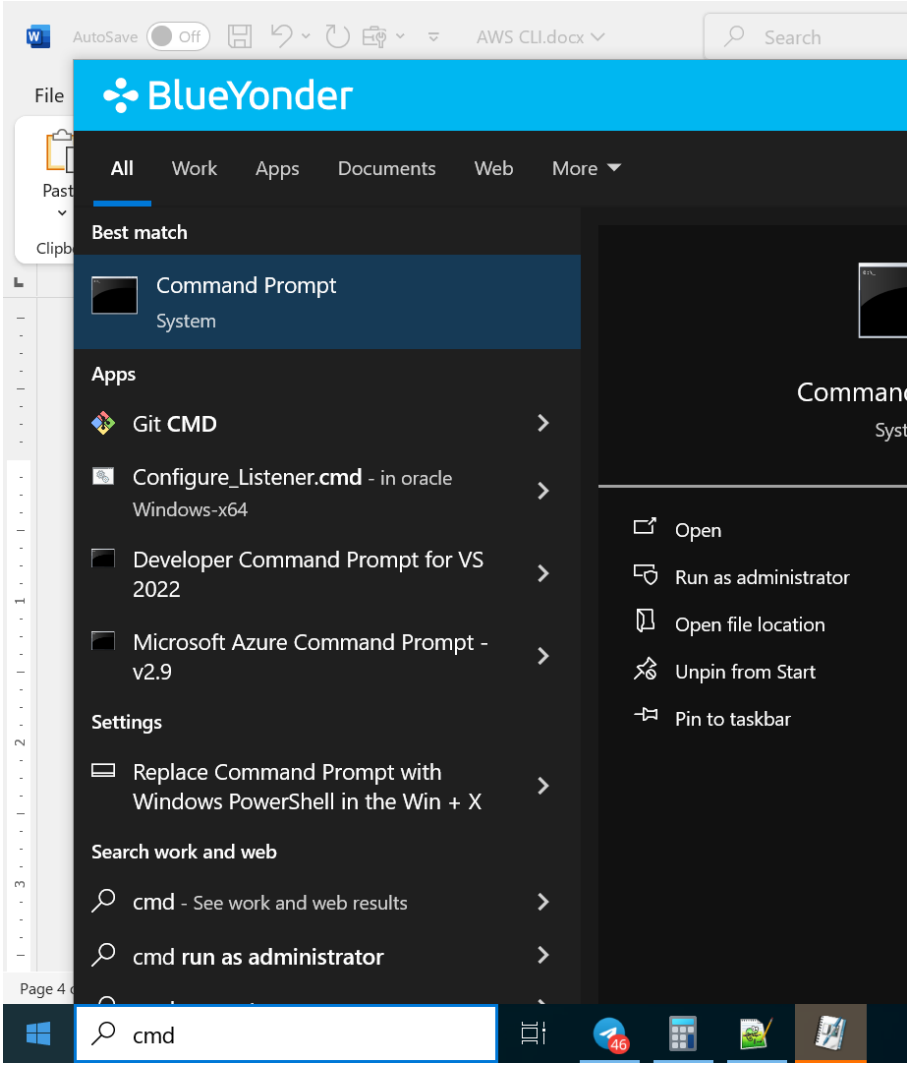




Proceed with the defaults by clicking "Next."

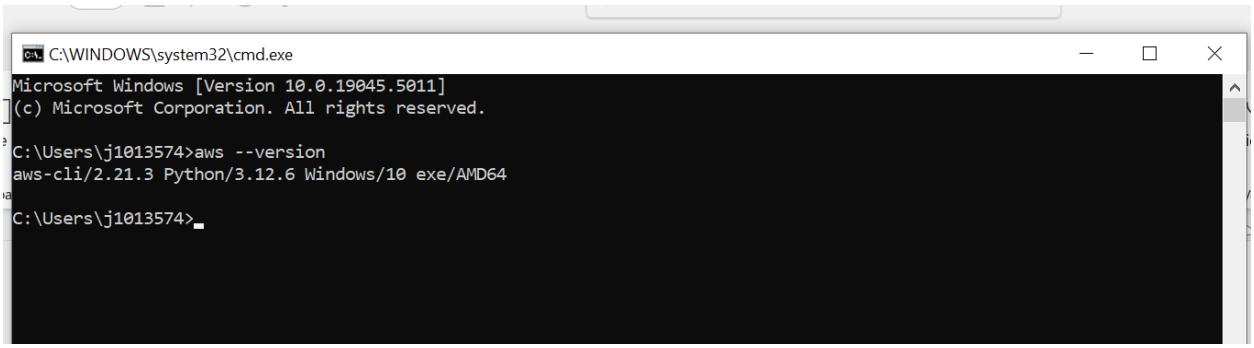
Verify Installation

Open a terminal or command prompt



Run below command

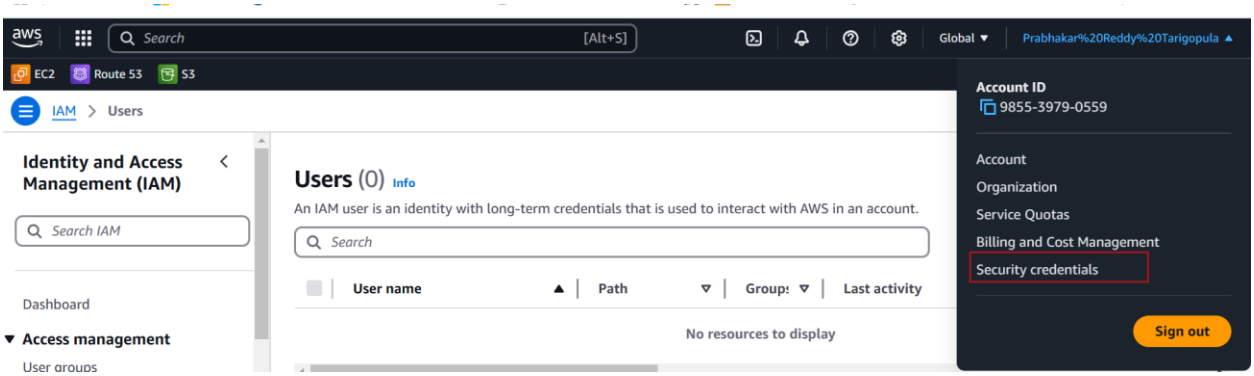
aws --version



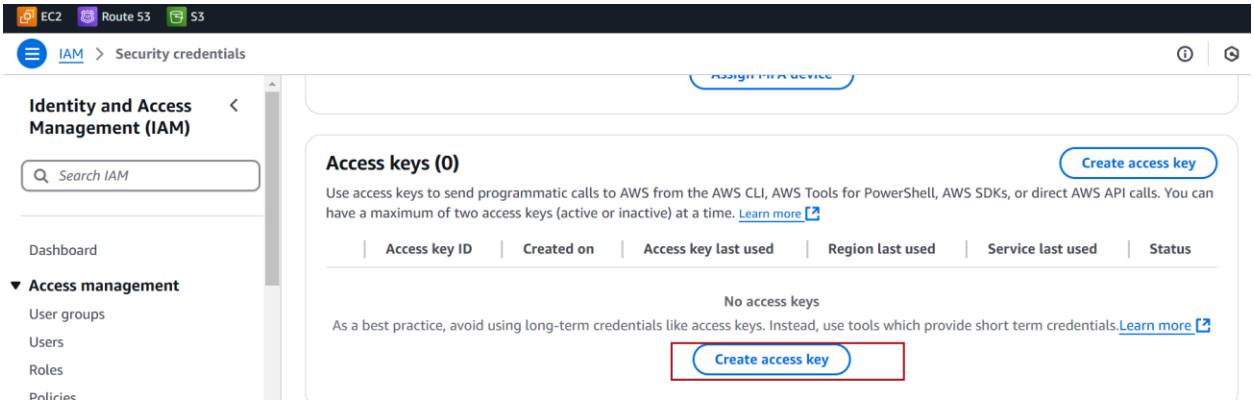
Screenshot for configuring Access Keys for AWS CLI

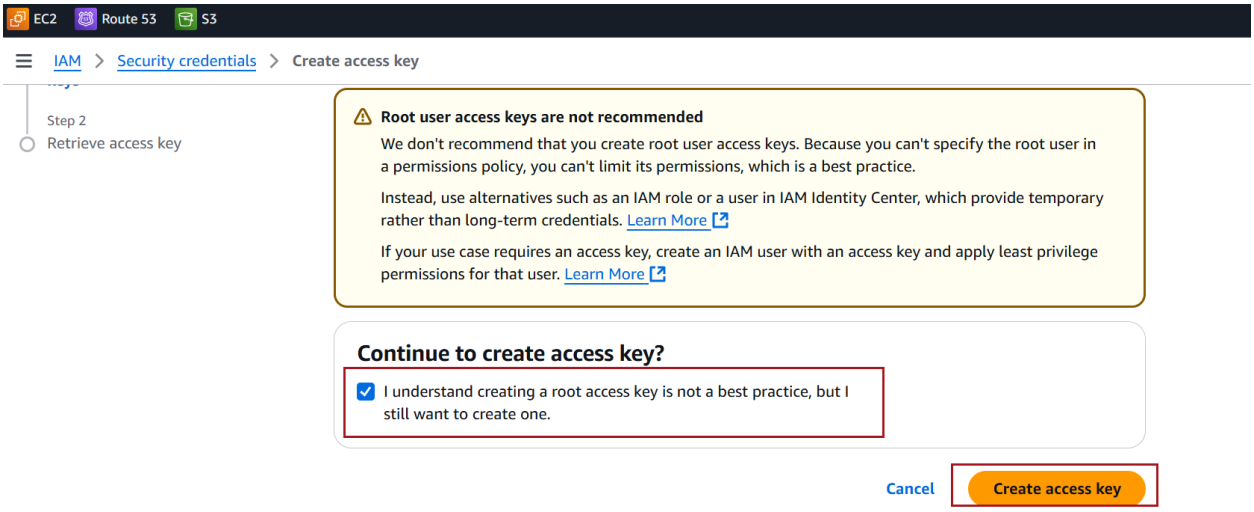
In the top-right corner of the AWS Management Console, click your **account name** or **account ID** (depending on your setup).

Select **Security Credentials** from the dropdown menu

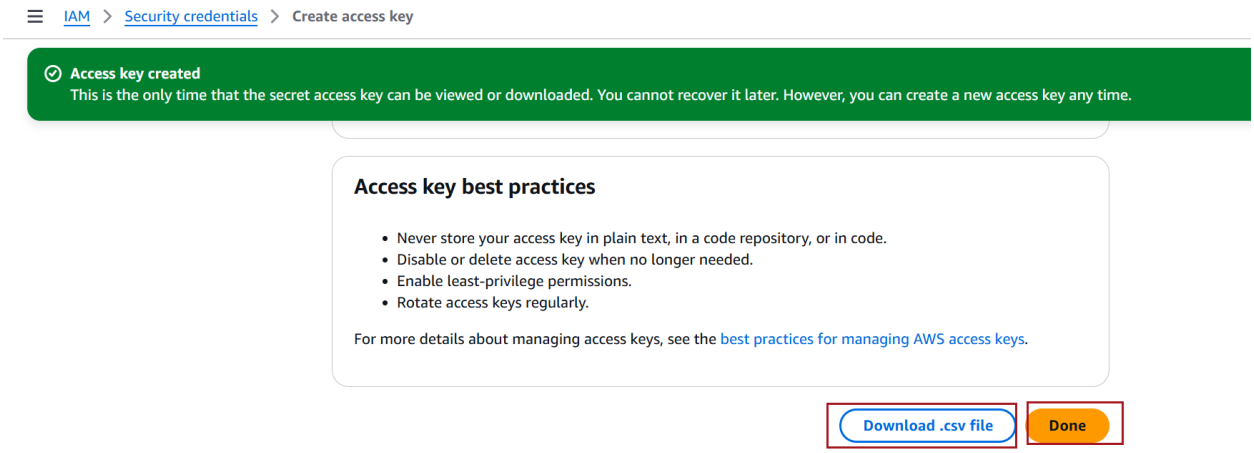


Scroll down to the **Access keys** section and click **Create access key**

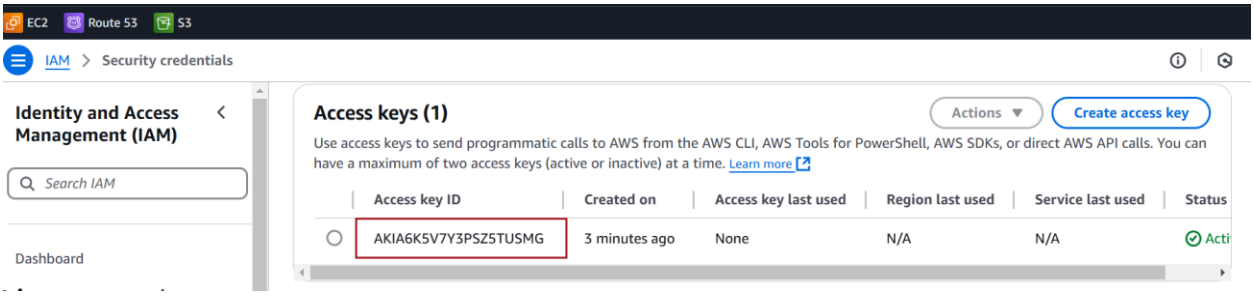




Save the .csv file in a secure location and click on Done



Observe that access keys created



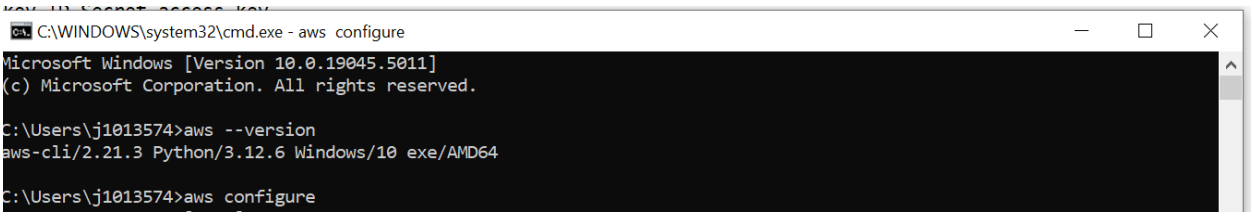
Screenshot for Configuring AWS CLI

Open a terminal or command prompt.

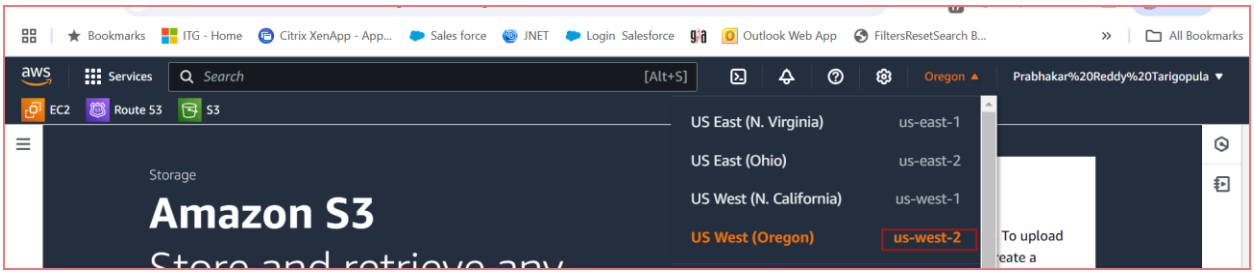
Run aws configure

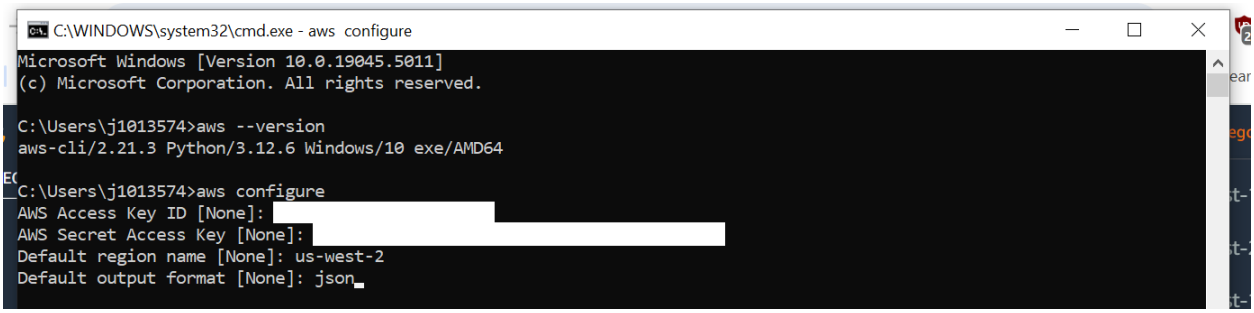
Enter the following details:

- **AWS Access Key ID**: Enter your Access Key ID.
- **AWS Secret Access Key**: Enter your Secret Access Key.
- **Default region name**: Enter the AWS region (e.g., us-east-1 or ap-south-1).
- **Default output format**: Choose json, table, or text (default is json).



Region





Test Configuration

Try listing your **S3 buckets** to test if AWS CLI can interact with your AWS account

C:\WINDOWS\system32\cmd.exe

```
C:\Users\j1013574>aws s3 ls
2024-11-16 18:53:21 mysourcebuckets3test

C:\Users\j1013574>
```

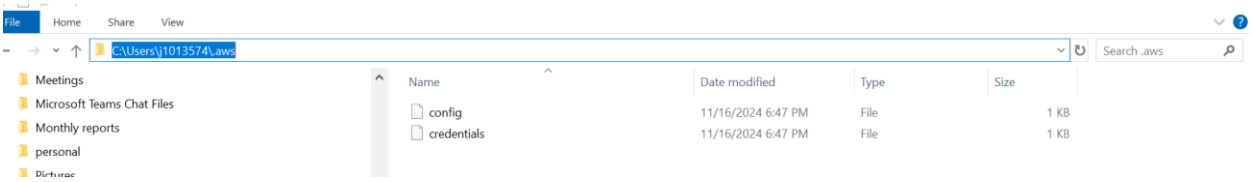
Screenshot for Checking AWS folder on Windows

Open **File Explorer**.

C:\Users\<YourName>\.aws

Inside the .aws folder, you should see two key files:

- **config**: Contains settings like the default region and output format.
- **credentials**: Contains the Access Key ID and Secret Access Key for your configured profiles.



Screenshot for accessing another AWS account

Run the aws configure command with a custom profile name for the other account. You can give any name to the profile e.g., another-account

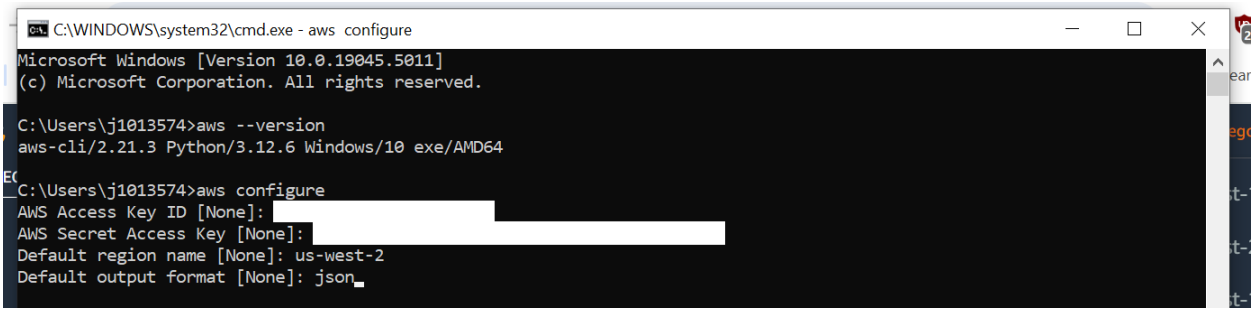
C:\WINDOWS\system32\cmd.exe

```
C:\Users\j1013574>aws configure --profile another-account
```

Enter the credentials for the other account:

- **AWS Access Key ID**: Enter the access key for the other account.
- **AWS Secret Access Key**: Enter the secret key for the other account.

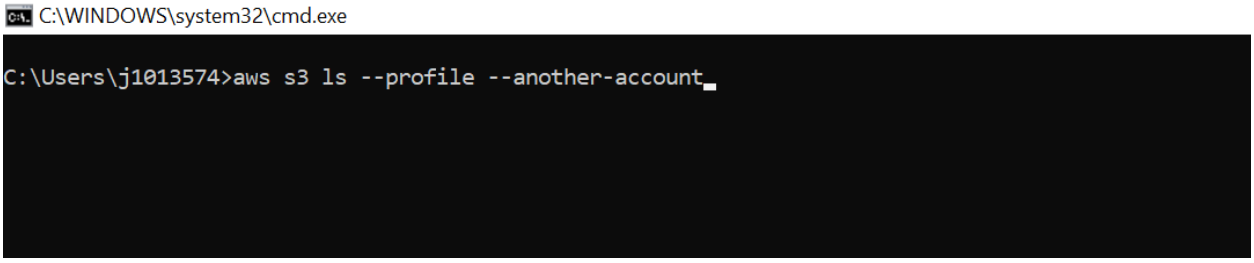
- **Default region name**: You can press **Enter** to accept the default region, or specify a region (e.g., us-west-2).
- **Default output format**: Press **Enter** to accept the default json, or specify your preference (e.g., text or table).



Use the Other Account via the Named Profile

Once the profile for the other account is configured, you can run AWS CLI commands using the --profile option to specify the account you want to use.

aws s3 ls --profile another-account



Screenshot for launching a New EC2 Instance using CLI

Creating an **EC2 instance** using the AWS CLI involves multiple steps

Prerequisite list required for EC2 instance creation through the AWS CLI:

- **AMI ID**: Amazon Machine Image ID (e.g., ami-12345).
- **Instance Type**: Instance size/type (e.g., t2.micro).
- **Key Pair**: For SSH access to the instance.
- **Security Group**: For firewall rules.

Run the following command to create an EC2 instance:

aws ec2 run-instances --image-id ami-22111148 --count 1 --instance-type t1.micro --key-name stage-key --security-groups my-aws-security-group

Note - If you are in a default VPC and want AWS to automatically assign the default security group, omit the --security-group-ids parameter entirely:

aws ec2 run-instances --image-id ami-01f9f821c37661b97 --count 1 --instance-type t2.micro --key-name newkey


```
            "PrivateDnsNameOptions": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            },
            "MaintenanceOptions": {
                "AutoRecovery": "default"
            },
            "CurrentInstanceBootMode": "legacy-bios",
            "InstanceId": "i-0061665e47d72b272",
            "ImageId": "ami-01f9f821c37661b97",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "PrivateDnsName": "ip-172-31-20-174.us-west-2.compute.internal",
            "PublicDnsName": "",
            "StateTransitionReason": "",
            "KeyName": "newkey",
            "AmiLaunchIndex": 0,
            "ProductCodes": [],
            "InstanceType": "t2.micro",
            "LaunchTime": "2024-11-17T16:10:56+00:00",
            "Placement": {
                "GroupName": "",
                "Tenancy": "default",
                "AvailabilityZone": "us-west-2a"
            },
            "Monitoring": {
                "State": "disabled"
            },
            "SubnetId": "subnet-06e089686ad696750",
            "VpcId": "vpc-05903633ec364f6f3",
            "PrivateIpAddress": "172.31.20.174"
        }
    ]
}

C:\Users\j1013574>
```


To see the list of all EC2 instances in your AWS account, use the following AWS CLI command

aws ec2 describe-instances --query "Reservations[*].Instances[*].[InstanceId,State.Name]"

```
C:\Users\j1013574>
C:\Users\j1013574>aws ec2 describe-instances --query "Reservations[*].Instances[*].[InstanceId,State.Name]"
[
    [
        [
            "i-0c26062fa455b67cd",
            "terminated"
        ]
    ],
    [
        [
            "i-0061665e47d72b272",
            "terminated"
        ]
    ],
    [
        [
            "i-0a470fd3a13f69bab",
            "terminated"
        ]
    ],
    [
        [
            "i-074b4638527327c1f",
            "running"
        ]
    ]
]
```