😊 AWS CLI (Command Line Interface)

The AWS Command Line Interface (AWS CLI) is a unified tool to manage your Amazon Web Services (AWS) resources from the command line. With just one tool to download and configure, you can control multiple AWS services and automate them through scripts. Here are some key features of the AWS CLI:

1. **Unified Tool**: The AWS CLI provides a single command-line interface to interact with all AWS services.
2. **Scriptable**: It allows you to create scripts to automate tasks such as launching instances, managing S3 buckets, and more.
3. **Configuration**: You can easily configure the CLI with your AWS credentials, region, and output format using the AWS configure command.
4. **Support for All AWS Services**: The AWS CLI supports all the AWS services, allowing you to interact with services like EC2, S3, Lambda, and more.
5. **Cross-Platform**: It is available on multiple platforms, including Windows, macOS, and Linux.

Common AWS CLI Commands

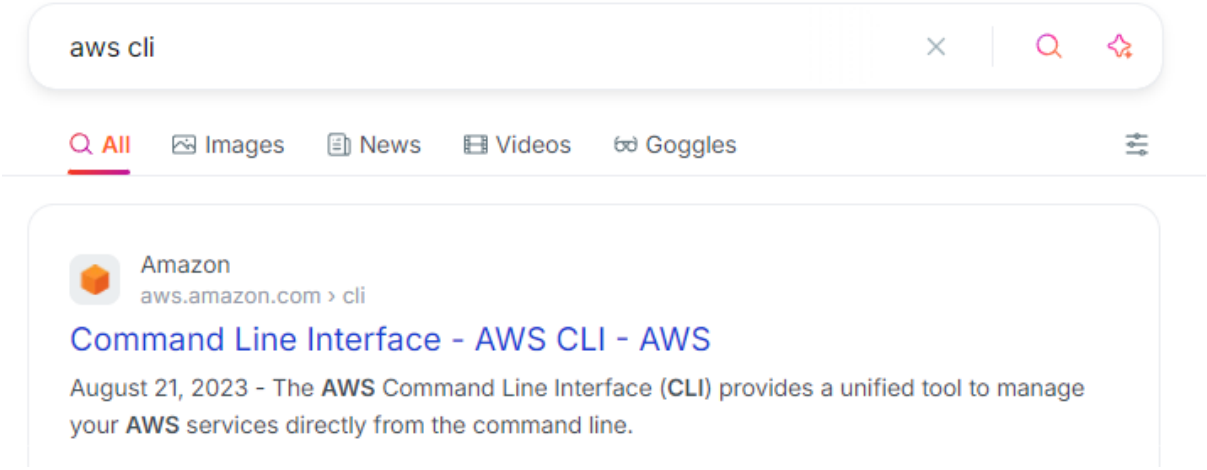
- **aws s3 ls**: List the contents of an S3 bucket.
- **aws ec2 describe-instances**: Retrieve details about your EC2 instances.
- **aws lambda invoke**: Invoke a Lambda function from the command line.
- **aws cloudformation deploy**: Deploy a CloudFormation stack.
- **aws configure**: Set up your AWS CLI by providing your AWS Access Key ID, Secret Access Key, region, and output format.

The goal of the steps provided is to help you install and configure the AWS Command Line Interface (CLI) on your local machine, so you can easily manage and interact with Amazon Web Services (AWS) resources. The process involves downloading the AWS CLI based on your operating system, installing it, and then verifying the installation. Next, you configure the AWS CLI with access credentials from your AWS account, which allows you to perform operations like listing, uploading, downloading, and deleting files in Amazon S3 buckets directly from your command prompt.

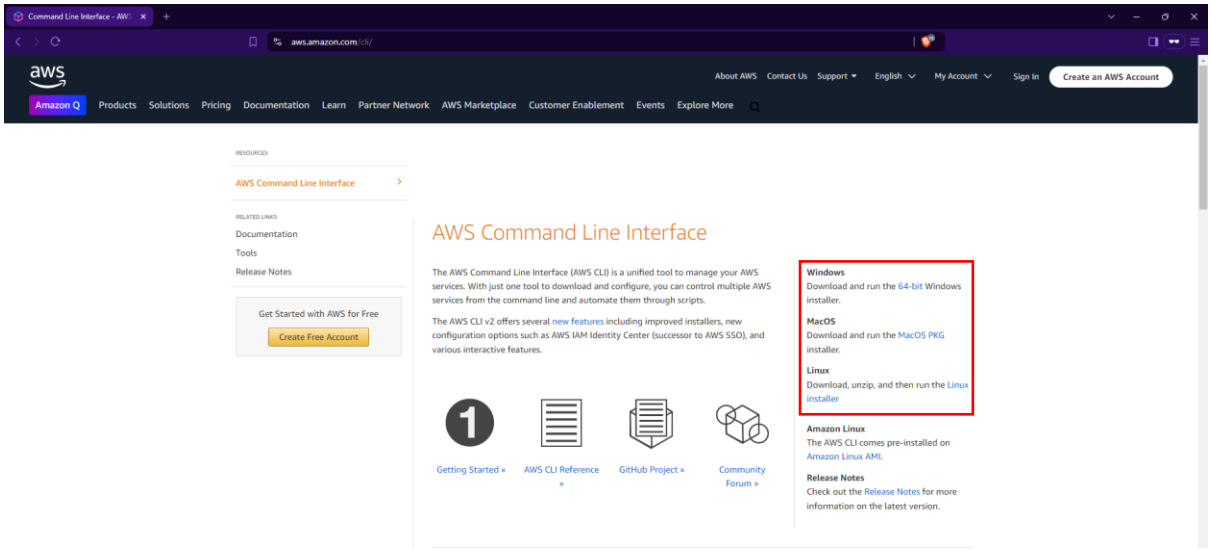
In summary, by following these steps, you'll be able to effectively use AWS CLI to manage your AWS resources, with the specific example of managing files in S3 buckets. The end goal is to empower you to interact with AWS services through the command line, making it easier to automate and manage tasks.

😄 To begin with the Lab:

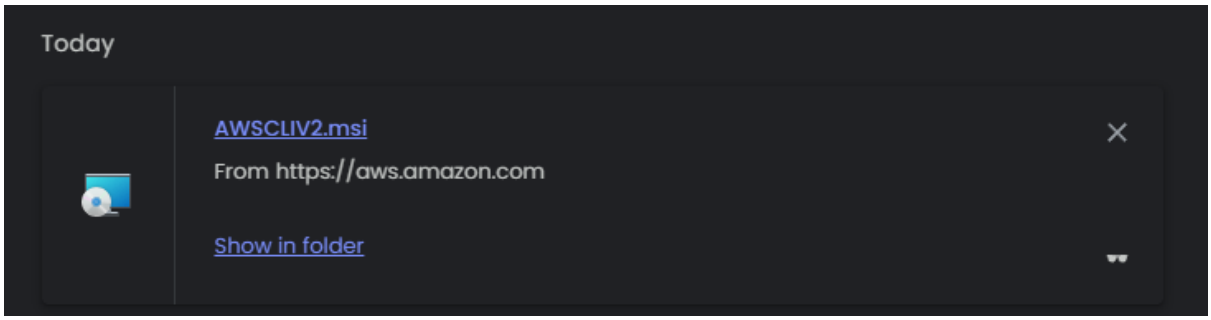
1. The first thing we need to do is, install AWS CLI in our Local Machine. So, for that open any browser and search AWS CLI on it, then click on the first link you see.



2. Now you will be on the AWS official site, as shown below. So, you need to choose your operating system it could be Windows, MacOS, or Linux.
3. Now based on your OS you need to download AWS CLI. Just click on that 64-bit for Windows and your download will start, similarly do for others.



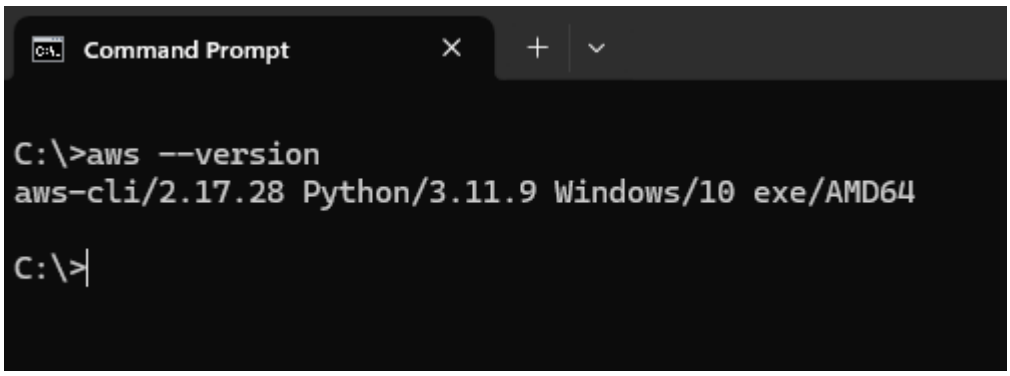
4. Once it is downloaded now you need to install it.



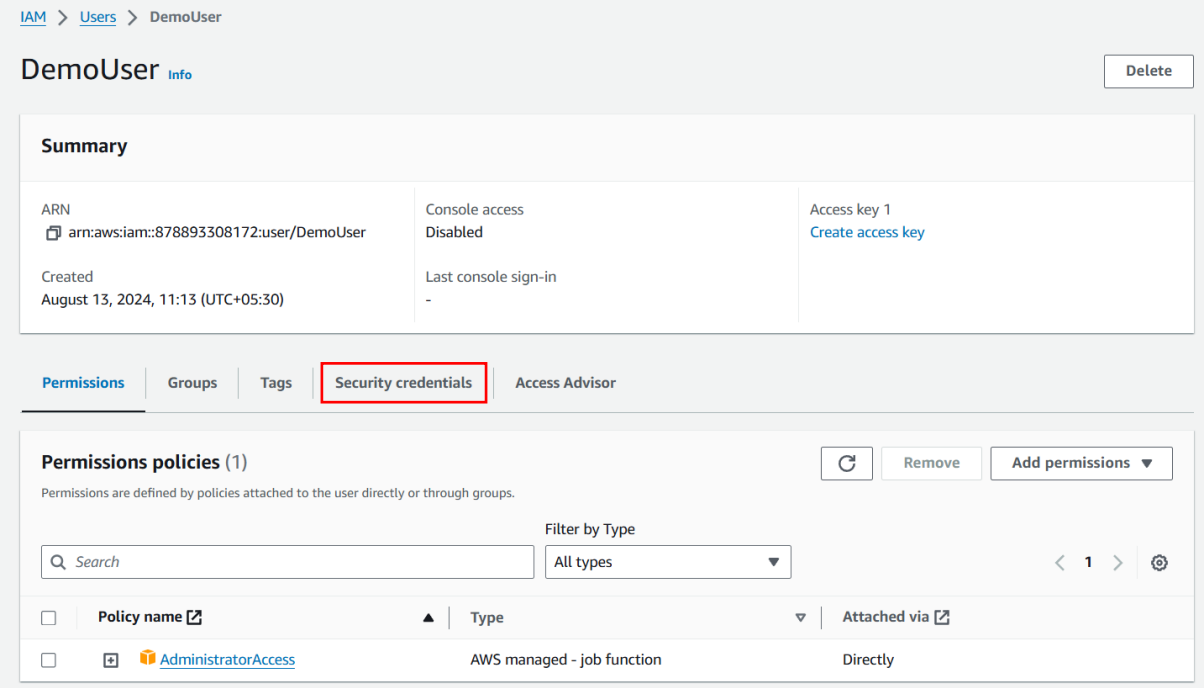
5. The installation process is simple like you do for other applications. Now just install it.

6. Once CLI is installed then you need to open Command Prompt in your local machine you need to run this command shown below. It will show you the version of AWS CLI. It also means that CLI has been installed properly in your Local Machine.

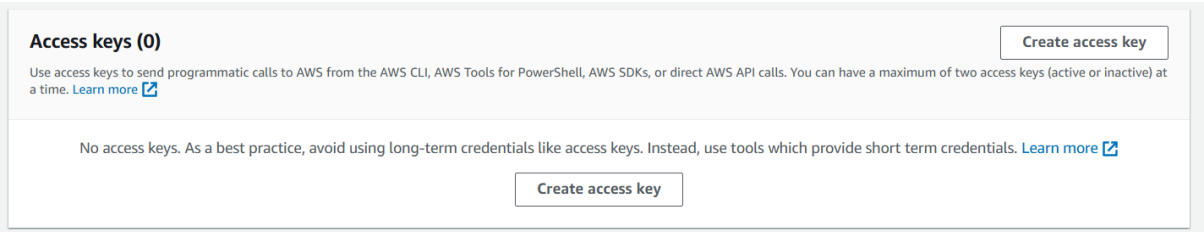
   aws --version



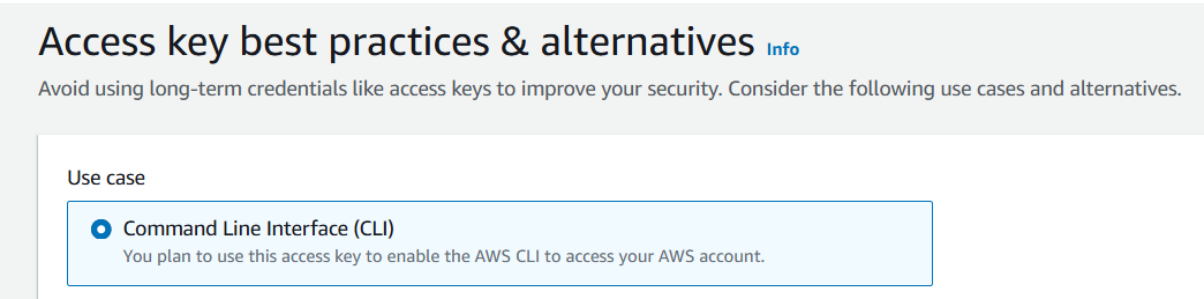
7. Now if you want to run CLI commands on your laptop you need to configure the access and secret access key of your AWS account. So, for that, you need to create a user and then assign that user with access and a secret access key.
8. Below you can see that we have a user with Administrator Access to it. Now you need to go to Security Credentials.



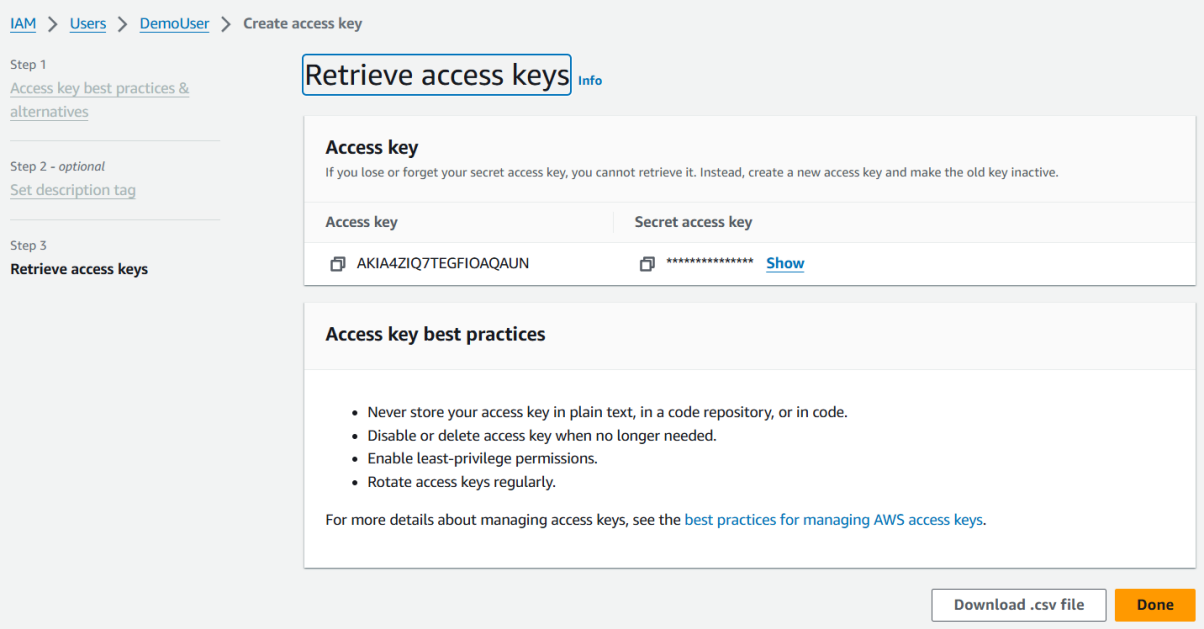
9. Then you need to scroll down to Access Keys and click on Create.



10. Now you need to choose command line interface (CLI) as your use case and move forward.



11. Below you can see that our Access keys have been created, you can also download these because once you have closed this page you cannot retrieve your access and secret access key. You must create new ones to use them again. Or you can just copy them to your notepad for the time being.



12. Now come back to the command prompt and write this command. Then it will ask you the Access key ID and the Secret Access Key. So, you need to copy those and paste them here. Then you need to provide your region and the default output format is JSON as you can see below.
13. Now if you are doing this for the first time then everything for you will be blank.

    aws configure

```
C:\>aws --version
aws-cli/2.17.28 Python/3.11.9 Windows/10 exe/AMD64

C:\>aws configure
AWS Access Key ID [****************FH5C]: AKIA4ZIQ7TEGFIOAQAUN
AWS Secret Access Key [****************5HE7]: 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
Default region name [ap-south-1]: ap-south-1
Default output format [none]: json

C:\>
```

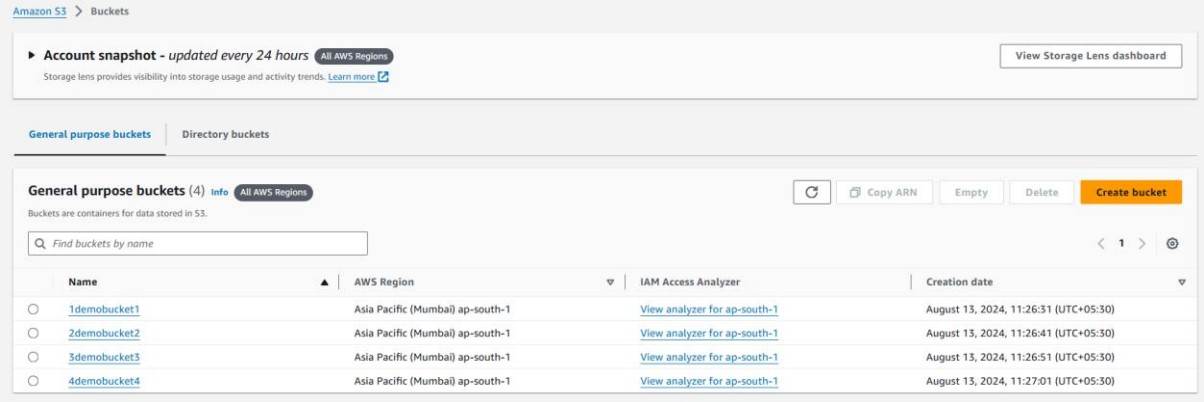
14. After configuring your Access Keys, you need to write this command. This command is to list every bucket in your account. As you can see below currently, I don't have any buckets, so I am going to create 4 new buckets.

    aws s3 ls

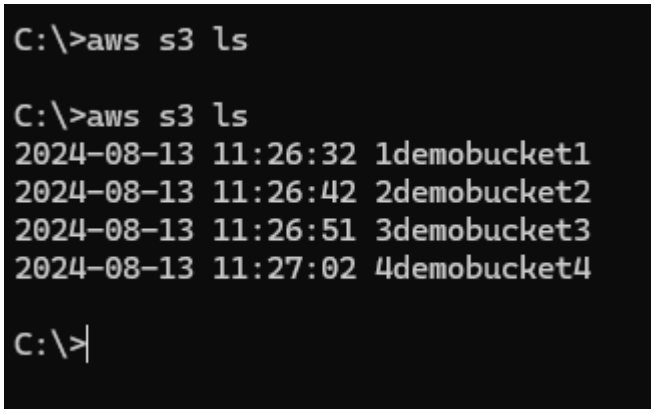
```
C:\>aws s3 ls

C:\>
```

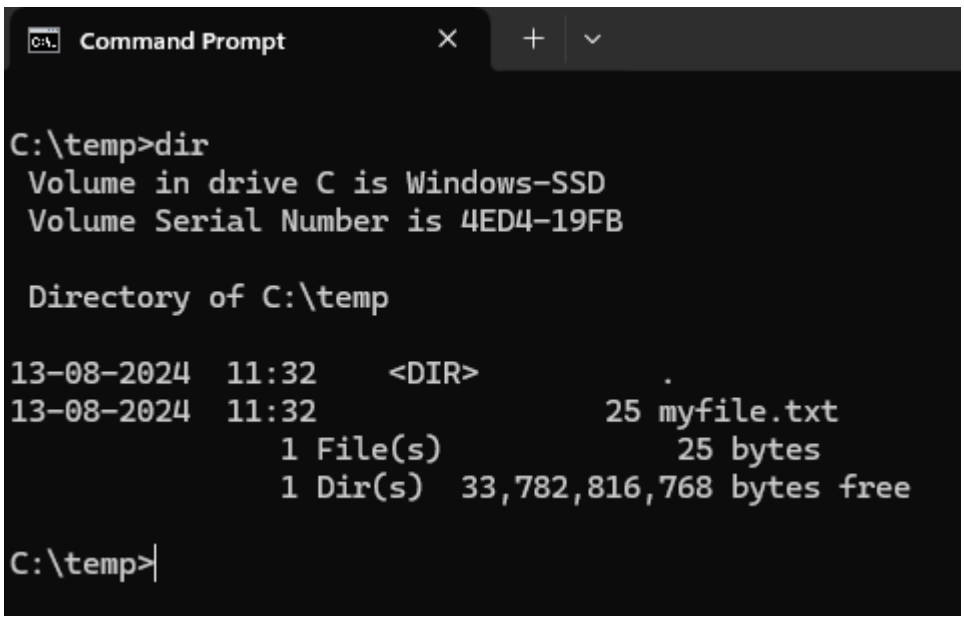
15. Below you can see that I have created 4 buckets in my S3. So, now if I run the command again in the command prompt to list the buckets in my account.



16. You will see that I can see my buckets.



17. Now we are going to upload a file from our laptop to one of our S3 buckets. For that, we are going to run a command. Below you can see that I am in the temp folder on my laptop and here you will see that this folder has only one file and we are going to upload this particular file to one of our S3 buckets.



18. Now we are going to run the command and you will see that our file has been uploaded.

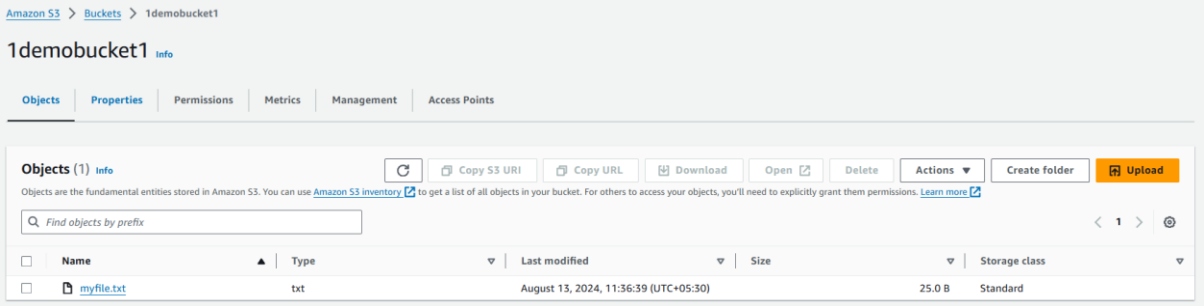
- **aws**: This is the command-line tool for interacting with Amazon Web Services (AWS).
- **s3**: This specifies that you are using the S3 service, which is Amazon's Simple Storage Service.
- **cp**: This stands for "copy" and is a command used to copy files or directories between your local machine and an S3 bucket, or between S3 buckets.
- **myfile.txt**: This is the source file on your local machine that you want to upload.
- **s3://1demobucket1/myfile.txt**: This is the destination path in the S3 bucket.
- **s3://**: Indicates that the destination is an S3 bucket.
- **1demobucket1**: This is the name of the S3 bucket where you want to upload the file.
- **myfile.txt**: This is the name of the file within the bucket.

aws s3 cp myfile.txt s3://1demobucket1/myfile.txt

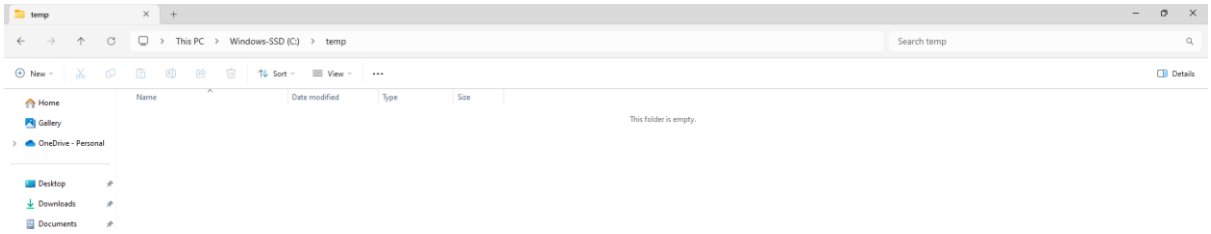
```
C:\temp>aws s3 cp myfile.txt s3://1demobucket1/myfile.txt
upload: .\myfile.txt to s3://1demobucket1/myfile.txt

C:\temp>
```

19. Just to confirm in my S3 bucket you can also see the file uploaded successfully.



20. Now we are going to download the same file from S3 to our laptop but first, we must delete this file from our temp folder.

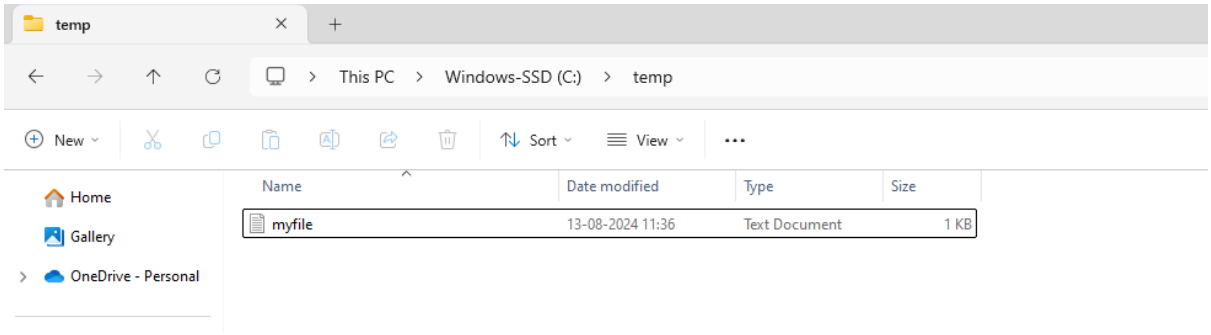


21. Now we are going to run the command and you can see that our file has been downloaded successfully.

aws s3 cp s3://1demobucket1/myfile.txt myfile.txt

```
C:\temp>aws s3 cp s3://1demobucket1/myfile.txt myfile.txt
download: s3://1demobucket1/myfile.txt to .\myfile.txt

C:\temp>
```



22. Now we are going to run a command to delete the file from our S3 bucket. Here you can see that we have deleted the file from our bucket.

    - **aws**: The command-line tool for interacting with AWS.
    - **s3**: Specifies that you are working with the S3 service.
    - **rm**: Stands for "remove," which is the command used to delete a file.
    - **s3://1demobucket1/myfile.txt**: The full path to the file you want to delete, which includes the bucket name and the file path.

aws s3 rm s3://1demobucket1/myfile.txt

```
C:\temp>aws s3 rm s3://1demobucket1/myfile.txt
delete: s3://1demobucket1/myfile.txt
```