**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

**DAY 14 TASK:**

Use Cloud CLI Tools: Install the CLI for your cloud provider (e.g., AWS CLI). Use it to list resources, upload files to storage, and manage VMs.

Name: SHAHANA.M.S Department: ADS



**Introduction**

Cloud Command Line Interface (CLI) tools provide a powerful way to interact with cloud services directly from the terminal. By installing and using the CLI for cloud providers like AWS, Azure, or Google Cloud Platform (GCP), users can efficiently manage cloud resources, automate tasks, and deploy applications without needing to use the web console. The CLI enables operations such as listing resources, uploading files to cloud storage, and managing virtual machines (VMs) with simple commands.

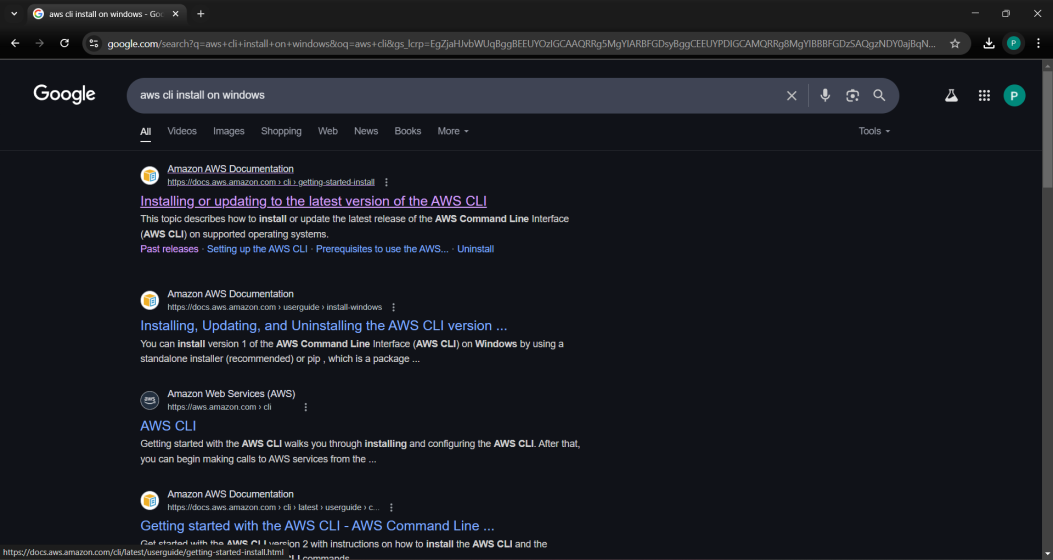
**Objective**

* To install the CLI tool for a cloud provider (AWS CLI, Azure CLI, or Google Cloud SDK).
* To use the CLI to list cloud resources.
* To upload files to cloud storage using CLI commands.
* To manage cloud VMs via CLI (start, stop, and configure instances).

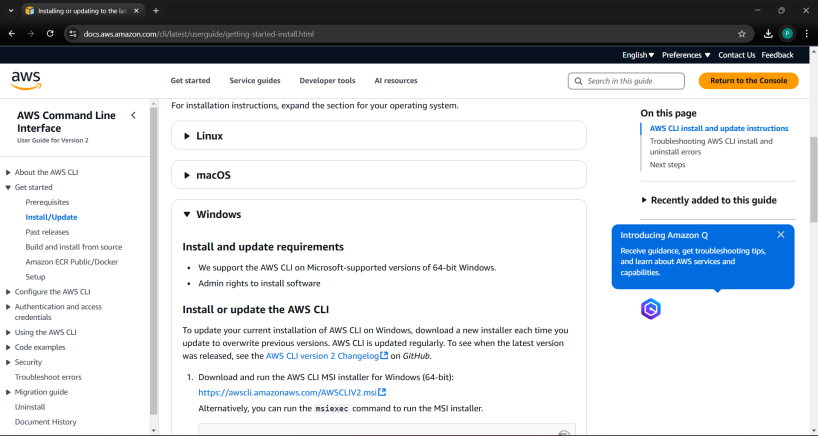
**Importance**

* **Efficiency**: Automates cloud management tasks, reducing manual effort.
* **Flexibility**: Enables remote management of cloud resources from any system.
* **Scripting & Automation**: Supports scripting for repetitive cloud operations.
* **Faster Deployment**: Allows quick provisioning and configuration of resources.

Step 1: Search for aws cli for windows and click the first link to proceed.



Step 2: In that page for windows click the download link it will start downloading.



Step 3: Follow up the installation wizard.



Step 4: Open Command Prompt and Type the following command and press Enter: aws --version If this appears, the AWS CLI is installed successfully!



**Step 4:** Configure the AWS CLI Now that the AWS CLI is installed, we need to configure it with your AWS account credentials so that you can interact with AWS services like S3 and EC2.

1. Open Command Prompt

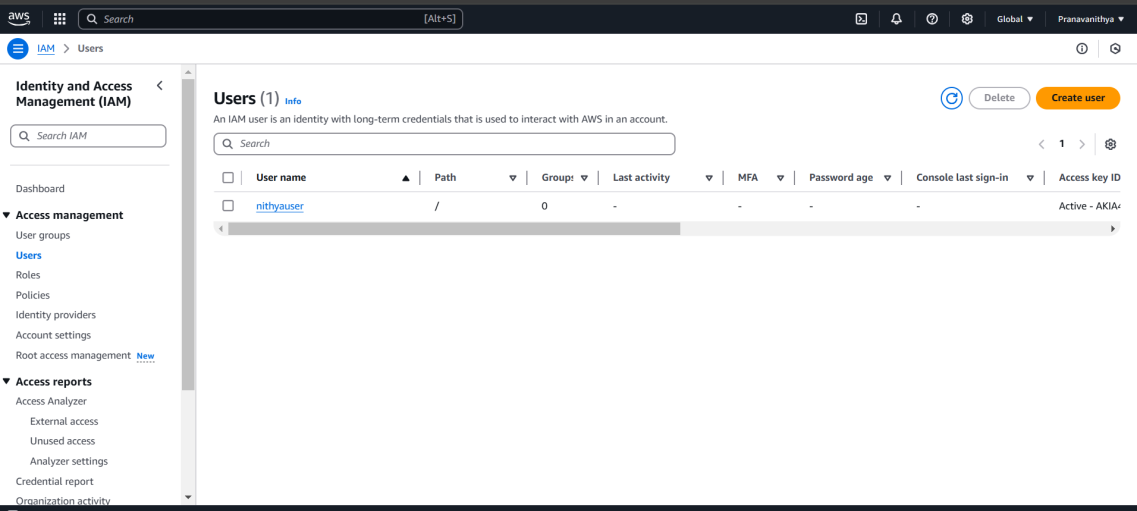
2. Run the Configuration Command: In the command prompt, type the following command and press Enter:

**aws configure**

3. Enter Your AWS Credentials: This will prompt you for four pieces of information: AWS Access Key ID: You need to get this from your AWS account.

5. Default Region Name: You can enter a region code like us-east-1

6. Default Output Format: Choose json



**Step 5:** List AWS Resources Using the AWS CLI S3 (Simple Storage Service) is where you store files in AWS. To list your S3 buckets, follow these steps:

1. List S3 Buckets: Run the following command in Command Prompt **aws s3 ls**



2. Expected Result:

If you have any S3 buckets, they will appear as a list.

If you don't have any, you might see an empty output like this.

A new bucket will be listed here when created.

This command helps you verify the S3 storage you currently have.

**Step 6**: Launch an Ec2 instance in the console and Run the following command:

**aws ec2 describe-instances --query "Reservations[].Instances[].{ID:InstanceId,State:State.Name,Typ e:InstanceType,Name:Tags[?Key=='Name'].Value | [0]}"**

Expected Result:

This will display details about your EC2 instances. You’ll see something like.

**Step 7:** Upload Files to S3 Using AWS CLI If you don’t already have an S3 bucket, create a new one.

Run this command to upload a file (replace file.txt with the path to the file you want to upload, and my-unique-bucket-name with your bucket name). Note : The whatever the file u want to store it must be in separate folder.

**aws s3 cp C:\path\to\your\file.txt s3://my-unique-bucket-name/ - recursive**

**Step 8:** To check if the file is uploaded successfully, you can list the contents of your S3 bucket:

**aws s3 ls s3://my-unique-bucket-name/**

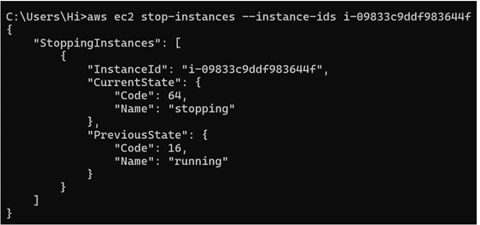
**Step 9:** Run the following command to list your EC2 instances, including their ID, state, type, and name (if tagged with "Name"):

**aws ec2 describe-instances --query "Reservations[].Instances[].{ID:InstanceId,State:State.Name,Typ e:InstanceType,Name:Tags[?Key=='Name'].Value | [0]}"**

**Step 10:** Now just stop the instance and if you need to restart the terminated instance then use this :

Replace with the ID of the instance you want to start

**aws ec2 start-instances --instance-ids <instance-id>**

****

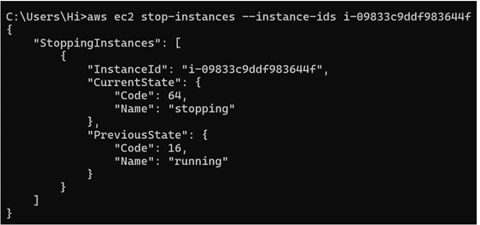
**Step 11:** Verify the status of the instance:

**aws ec2 describe-instances --instance-ids --query "Reservations[].Instances[].State.Name"**

****

**Step 12:** To stop an EC2 instance, use this command. Again, replace with the actual instance ID:

**aws ec2 stop-instances --instance-ids**

****

**Step 13:**Do again verification of the status of the instance:

**aws ec2 describe-instances --instance-ids --query "Reservations[].Instances[].State.Name"**

****

**Step 14:** Clean Up Resources

• Delete a Single File:

**aws s3 rm s3://my-unique-bucket-name/file.pdf**

• Delete All Files in a Bucket (if you want to delete everything):

**aws s3 rm s3://my-unique-bucket-name/--recursive**

**Step 15:** Once the files are deleted, you can delete the bucket itself.

• Delete an S3 Bucket:

**aws s3 rb s3://my-unique-bucket-name/ --force**

Then verify it by :

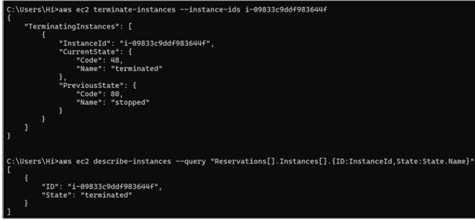
**aws s3**

**Step 16**: If you no longer need the EC2 instances, you can terminate them to avoid ongoing charges.

**aws ec2 terminate-instances --instance-ids**

Then verify it by :

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

****

**Outcome**

* Successfully installing and configuring the cloud provider’s CLI tool.
* Listing cloud resources such as instances, storage buckets, and services.
* Uploading and downloading files using CLI commands.
* Managing cloud VMs (starting, stopping, and configuring instances).