Placement Empowerment Program

Cloud Computing and DevOps Centre

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: CHANDRU S

Department: INFORMATION TECHNOLOGY

St. JOSEPH'S COLLEGE OF ENGINEERING | St. JOSEPH'S INSTITUTE OF TECHNOLOGY

AUTONOMOUS INSTITUTIONS, AFFILIATED TO ANNA UNIVERSITY

Introduction

Cloud Command Line Interface (CLI) tools, such as the **AWS CLI**, enable users to interact with cloud services programmatically, enhancing efficiency and automation. The **AWS CLI** simplifies the management of cloud resources like **Amazon S3 (storage), EC2 (compute), and other AWS services** directly from the command line. This **Proof of Concept (PoC)** demonstrates how to use AWS CLI on a **Windows system** to perform essential cloud operations, including **listing resources, uploading files, and managing virtual machines (VMs).**

Overview

This PoC provides hands-on experience in using the AWS CLI for efficient cloud resource management. By completing this PoC, you will learn to:

1. Install and configure AWS CLI on a Windows system.
2. List **AWS resources** such as S3 buckets and EC2 instances.
3. Upload files to **Amazon S3** using CLI commands.
4. Manage **EC2 instances** (start, stop, and terminate) directly from the command line.

Following a structured, step-by-step approach, this PoC will help users understand how CLI tools streamline cloud operations, reducing reliance on the AWS Management Console.

Objectives

The key learning objectives of this PoC include:

1. **Install and Configure AWS CLI** – Set up AWS CLI on a **Windows system** and configure it using **Access Keys**.
2. **List AWS Resources** – Use CLI commands to retrieve details about **S3 buckets** and **EC2 instances**.
3. **Upload Files to S3** – Transfer files from a **local system to Amazon S3** via CLI.
4. **Manage EC2 Instances** – Start, stop, and terminate **EC2 instances** directly from the command line.
5. **Optimize Resource Usage** – Clean up unused resources to prevent unnecessary costs.

Importance

This PoC is valuable for professionals aiming to enhance their cloud management skills through real-world applications:

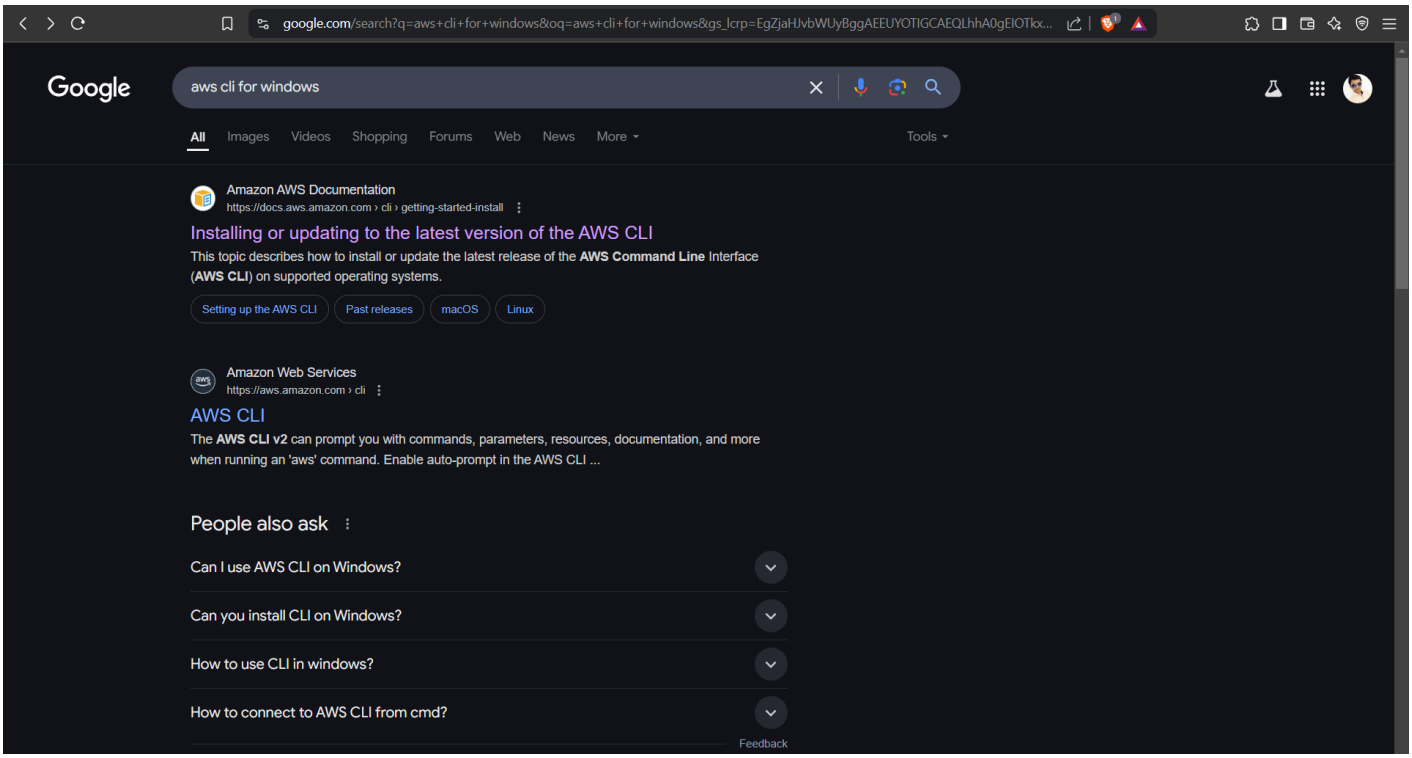
1. **Efficiency & Automation** – AWS CLI allows faster execution of tasks compared to the web console, improving workflow automation.
2. **Skill Development** – Hands-on CLI experience builds expertise in cloud management, crucial for **Cloud Engineers and DevOps Specialists**.

3. **Cost Optimization** – CLI-based resource management ensures better control over cloud usage, helping to minimize unnecessary expenses.

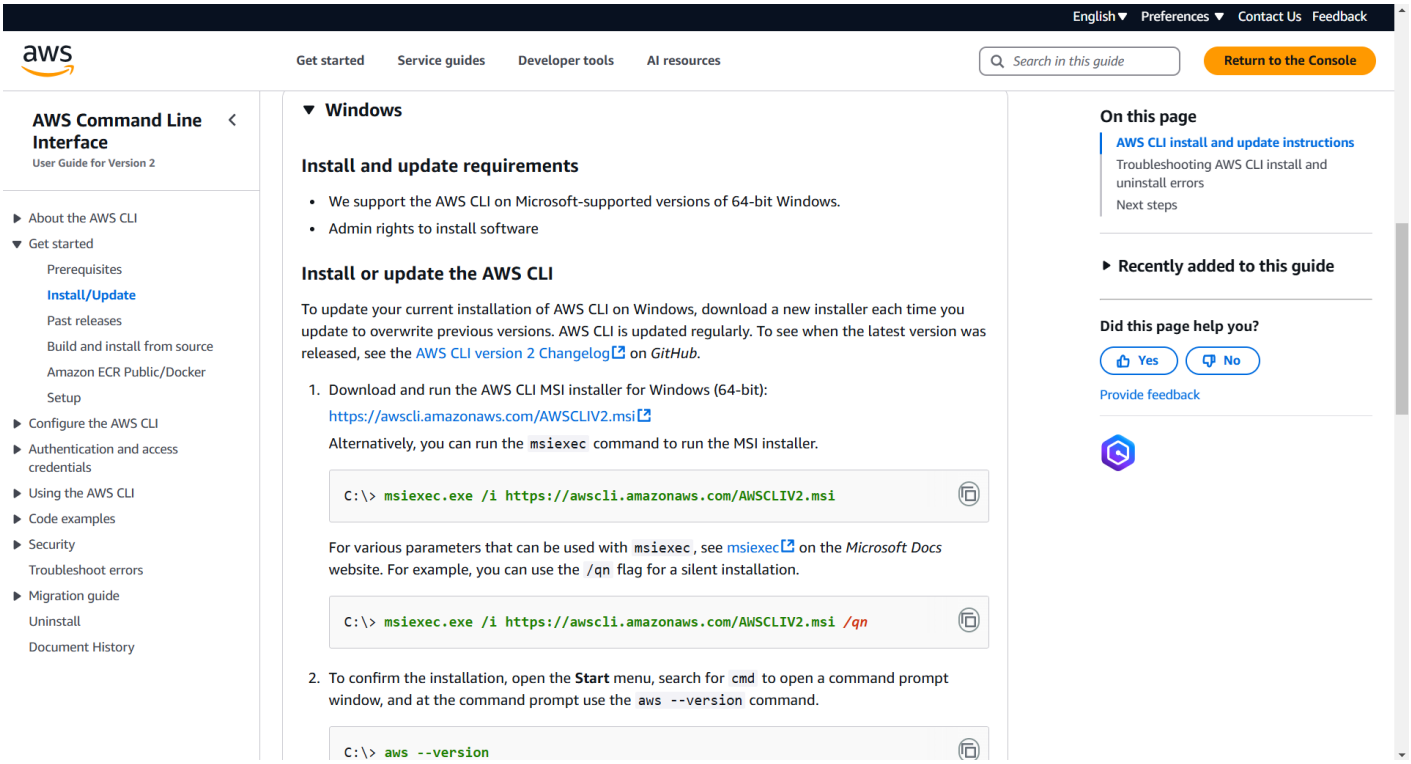
Step-by-Step Overview

Step 1:

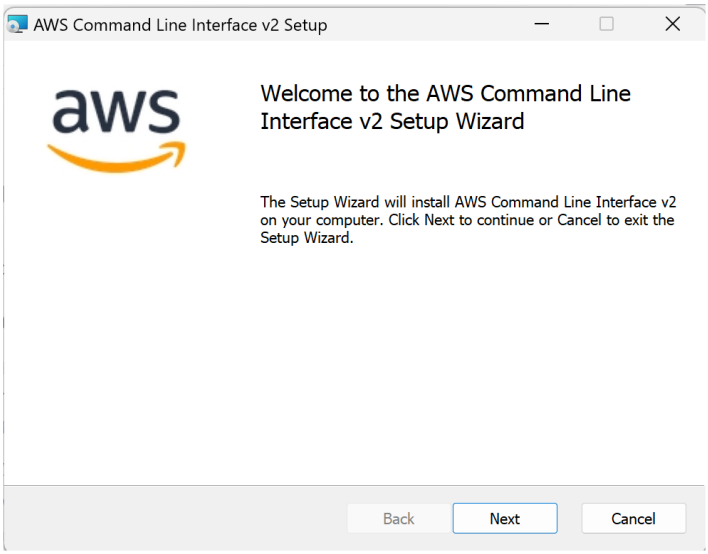
- Search for **"AWS CLI for Windows"** and click the first link.



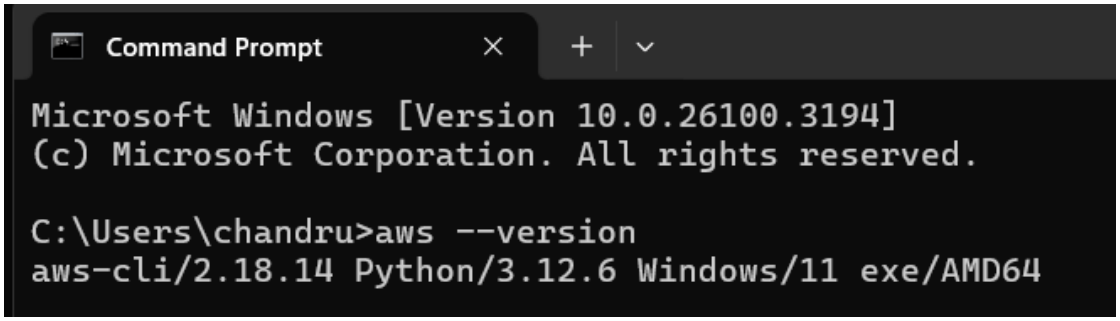
- Download the installation file for Windows.



- Follow the installation wizard to complete the setup.



- Open **Command Prompt** and verify the installation by running: **aws --version**
- If it returns a version number, AWS CLI is installed successfully.



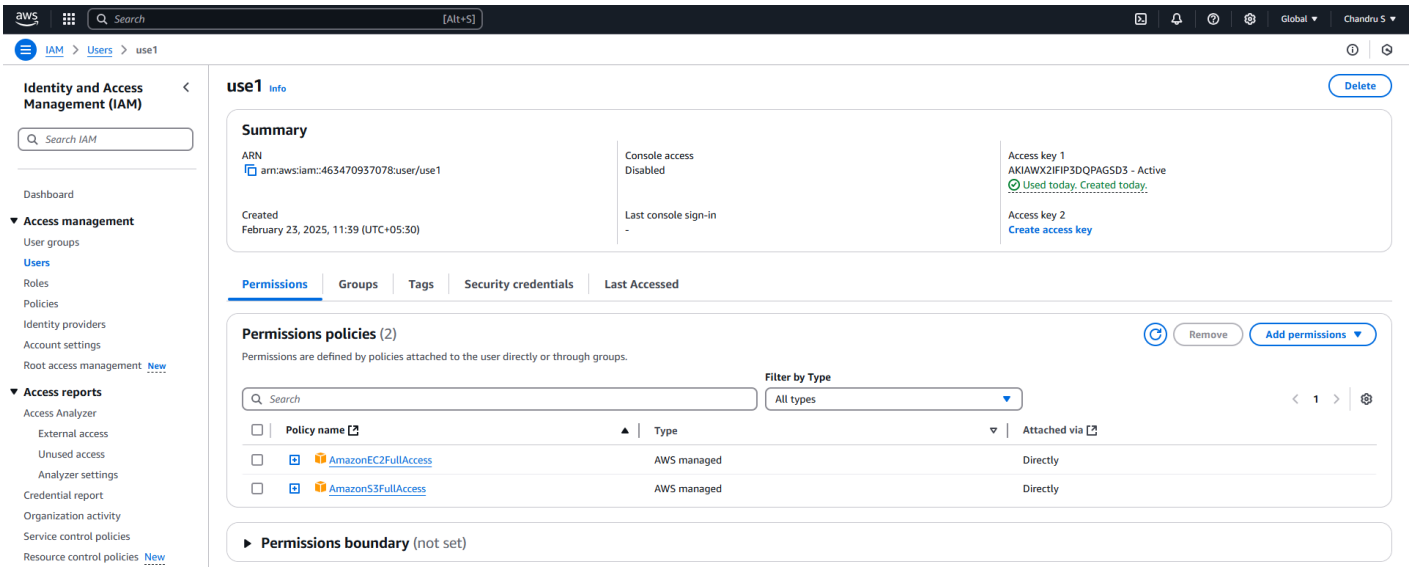
Step 2:

Now, configure AWS CLI with your AWS account credentials:

1. Open **Command Prompt** and run:
   aws configure
2. Enter the required details when prompted:
   - AWS Access Key ID
   - AWS Secret Access Key
   - Default Region Name (e.g., us-east-1)
   - Default Output Format (choose json)

To get your AWS Access Key & Secret Key:

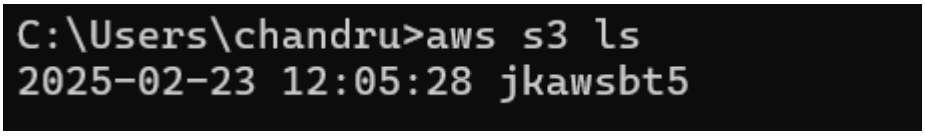
- Go to **AWS IAM Console** → Click **Users** → Select your username → Under **Security Credentials**, create a new access key.



Step 3:

To check existing S3 buckets, run: **aws s3 ls**

- If buckets exist, they will be listed.
- If no buckets exist, the output will be empty.



Step 4:

- Launch an EC2 instance using the AWS Management Console.
- List EC2 instances using:
  aws ec2 describe-instances --query "Reservations[].Instances[].{ID:InstanceId,State:State.Name,Type:InstanceType,Name:Tags[?Key=='Name'].Value | [0]}"
- This will display Instance ID, State, Type, and Name (if tagged).

```
C:\Users\chandru>aws ec2 describe-instances --query "Reservations[].Instances[].{ID:InstanceId,State:State.Name,Type:InstanceType,Name:Tags[?Key=='Name'].Value | [0]}"
[
    {
        "ID": "i-06b0dcd876537e8c4",
        "State": "running",
        "Type": "t2.micro",
        "Name": "jkec2"
    }
]
```

Step 5:

- If you don't have an S3 bucket, create one in the AWS console.
- Upload a file to S3:
  aws s3 cp C:\path\to\your\file.txt s3://my-unique-bucket-name/

```
C:\Users\chandru>aws s3 cp "C:\Users\chandru\Downloads\h2slogo.png" s3://jkawsbt5/
upload: Downloads\h2slogo.png to s3://jkawsbt5/h2slogo.png
```

- Verify the upload:
  aws s3 ls s3://my-unique-bucket-name/

```
C:\Users\chandru>aws s3 ls s3://jkawsbt5/
2025-02-23 12:19:58      15801 h2slogo.png
```

Step 6:

- Start an EC2 instance:
  aws ec2 start-instances --instance-ids <instance-id>

```
C:\Users\chandru>aws ec2 start-instances --instance-ids i-06b0dcd876537e8c4
{
    "StartingInstances": [
        {
            "InstanceId": "i-06b0dcd876537e8c4",
            "CurrentState": {
                "Code": 16,
                "Name": "running"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

- Check instance state:
  aws ec2 describe-instances --instance-ids <instance-id> --query "Reservations[].Instances[].State.Name"

```
C:\Users\chandru>aws ec2 describe-instances --instance-ids i-06b0dcd876537e8c4 --query "Reservations[].Instances[].State
.Name"
[
    "running"
]
```

- Stop an EC2 instance:
  aws ec2 stop-instances --instance-ids <instance-id>

```
C:\Users\chandru>aws ec2 stop-instances --instance-ids i-06b0dcd876537e8c4
{
    "StoppingInstances": [
        {
            "InstanceId": "i-06b0dcd876537e8c4",
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

Step 7:

- Delete a specific file from S3:
  aws s3 rm s3://my-unique-bucket-name/file.pdf

```
C:\Users\chandru>aws s3 rm s3://jkawsbt5/h2slogo.png
delete: s3://jkawsbt5/h2slogo.png
```

- Delete all files in an S3 bucket:
  aws s3 rm s3://my-unique-bucket-name/ --recursive
- Delete an S3 bucket:
  aws s3 rb s3://my-unique-bucket-name/ --force

```
C:\Users\chandru>aws s3 rb s3://jkawsbt5/ --force
remove_bucket: jkawsbt5
```

- Verify bucket deletion:
  aws s3 ls

Step 8:

- Terminate an EC2 instance:
  aws ec2 terminate-instances --instance-ids <instance-id>

```
C:\Users\chandru>aws ec2 terminate-instances --instance-ids i-06b0dcd876537e8c4
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-06b0dcd876537e8c4",
            "CurrentState": {
                "Code": 48,
                "Name": "terminated"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
```

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

```
C:\Users\chandru>aws ec2 describe-instances --query "Reservations[].Instances[].{ID:InstanceId,State:State.Name}"
[
    {
        "ID": "i-06b0dcd876537e8c4",
        "State": "terminated"
    }
]
```

That's the final step in this POC! You've successfully:

- Installed and configured AWS CLI
- Managed S3 and EC2 using CLI commands
- Uploaded and deleted files from S3
- Started, stopped, and terminated EC2 instances
- Cleaned up resources to avoid unnecessary costs

Outcome

By completing this **PoC on AWS CLI tools**, you will gain hands-on experience in efficiently managing cloud resources. Specifically, you will:

1. **Install and Configure AWS CLI** – Successfully set up AWS CLI on a **Windows system**, authenticate it, and enable seamless interaction with AWS services.
2. **List and Retrieve Cloud Resources** – Use CLI commands to list **S3 buckets, EC2 instances, and other AWS resources**, understanding their states and configurations.
3. **Upload Files to S3** – Transfer files from a **local system to an S3 bucket** using AWS CLI, demonstrating efficient cloud storage management.
4. **Manage EC2 Instances** – Programmatically **start, stop, and terminate EC2 instances**, showcasing essential virtual machine management skills.
5. **Optimize Cloud Resources** – Clean up **unused S3 buckets and EC2 instances** to prevent unnecessary costs and maintain a well-organized AWS environment.