**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: Kirubalini S I Department: IT



**Introduction**

Cloud CLI (Command Line Interface) tools are essential for modern cloud computing, providing a powerful and efficient way to interact with cloud services directly from your terminal. Instead of relying solely on graphical user interfaces, CLIs offer a text-based interface that allows for automation, scripting, and precise control over your cloud resources. By installing the CLI specific to your cloud provider, such as the AWS CLI, Azure CLI, or Google Cloud CLI, you gain the ability to perform a wide range of tasks, from simple resource listing to complex infrastructure management. This introduction will focus on the fundamental uses of these tools, specifically listing resources, uploading files to storage, and managing virtual machines (VMs), highlighting their importance in streamlining cloud operations.

**Objective**

The goal of this project is to:

**Automation and Efficiency:** The primary objective is to automate repetitive cloud management tasks, reducing manual effort and increasing operational efficiency.

**Precise Control:** To provide granular control over cloud resources, enabling users to perform specific actions that might be cumbersome or impossible through a graphical interface.

**Scripting and Integration:** To facilitate scripting and integration with other tools and systems, enabling the creation of complex workflows and infrastructure-as-code (IaC) solutions.

**Resource Management:** to provide the ability to list resources, upload files to storage, and manage VMs.

**Importance**

**Scalability and Speed:** CLIs enable rapid scaling and deployment of cloud resources, crucial for handling dynamic workloads.

**DevOps and CI/CD:** They are fundamental to DevOps practices and continuous integration/continuous delivery (CI/CD) pipelines, enabling automated deployments and infrastructure management.

**Troubleshooting and Diagnostics:** CLIs provide powerful tools for troubleshooting and diagnosing issues, allowing users to quickly access logs and resource information.

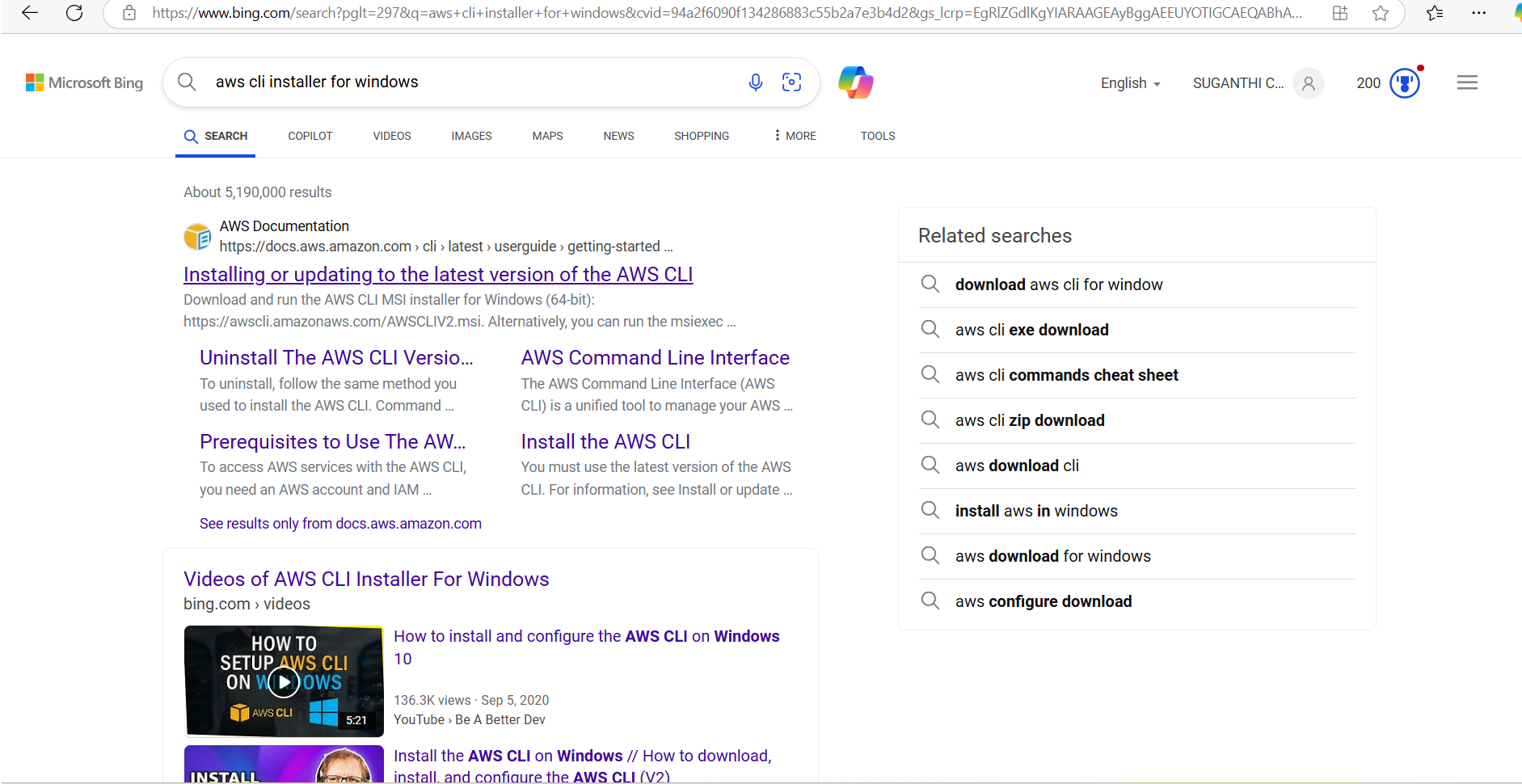
**Security:** CLIs, when used correctly, can enhance security by enabling fine-grained access control and automated security checks.

**Cost Optimization:** by scripting resource management, you can easily turn off unused resources, and monitor costs.

**Step-by-Step Overview**

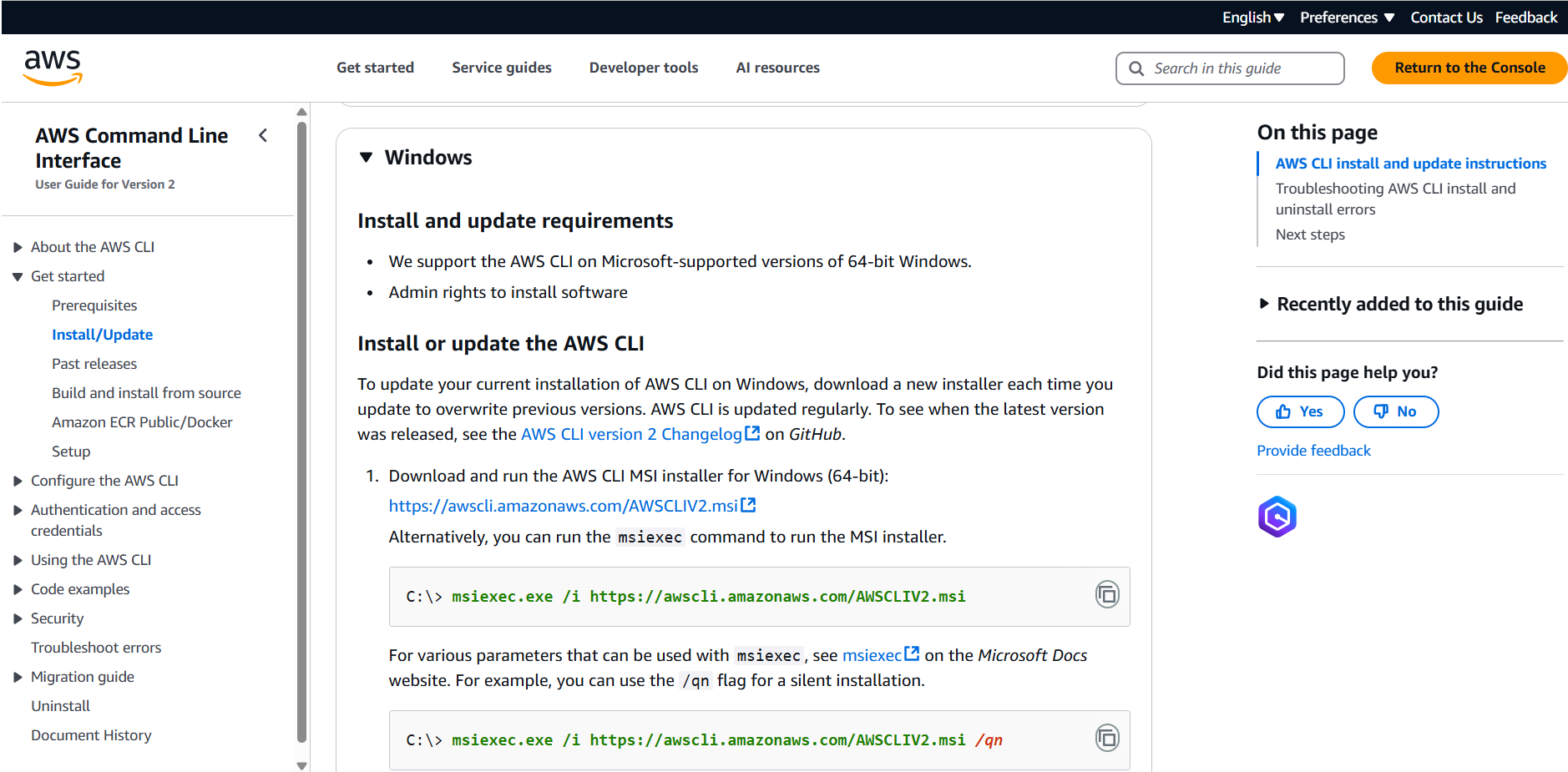
Step1:

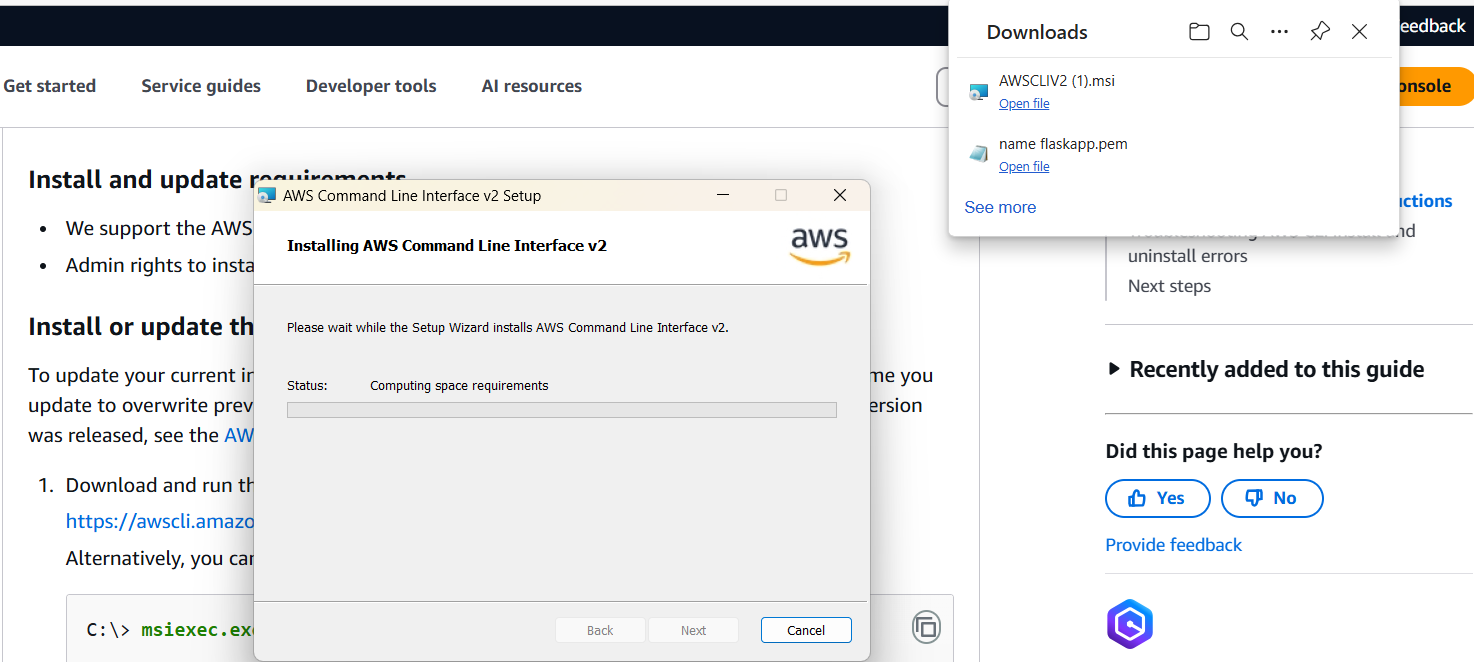
Search for "AWS CLI Installer for Windows" on Google and click the first link to access the official website.



Step 2:

Click on the "Install/Update" option located on the left-hand side of the Apache Lounge website. Select the link regarding your OS, Install by using the link provided else use the ***msiexec*** command





Step 3:

Once installed, verify the installation by opening Command Prompt (cmd) or PowerShell and running **aws --version**

It should return something like

aws-cli/2.x.x Python/3.x.x Windows/x86\_64

Step 4:

Before using AWS CLI, you need to configure it with your AWS credentials.

Open Command Prompt and type **aws configure**

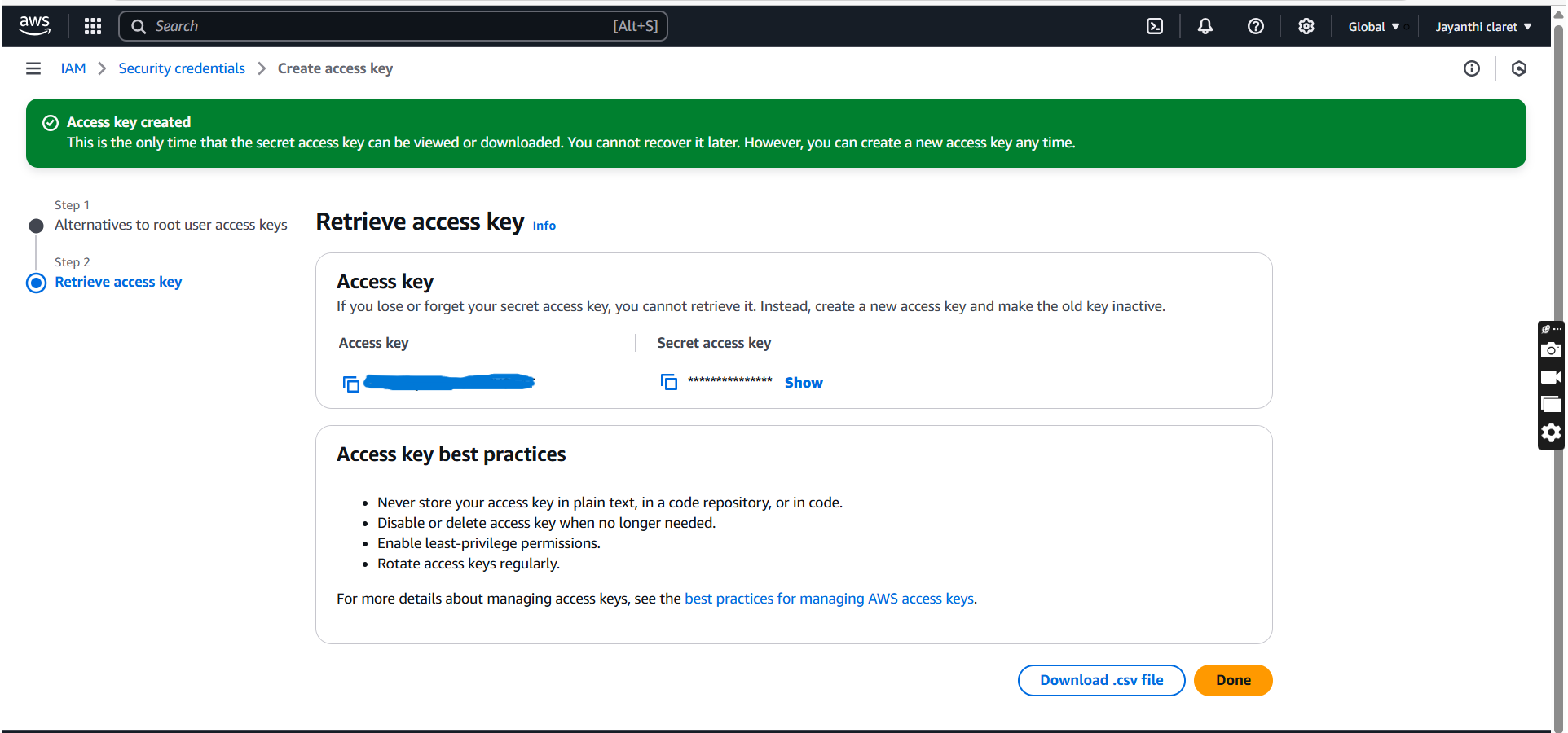
It will ask for:

AWS Access Key ID → Get it from AWS IAM > Security Credentials

AWS Secret Access Key → Get it from AWS IAM > Security Credentials

Default region name → Example: us-east-1 (Find yours in AWS Console)

Default output format → Keep it as json or press Enter for default



Step 5:

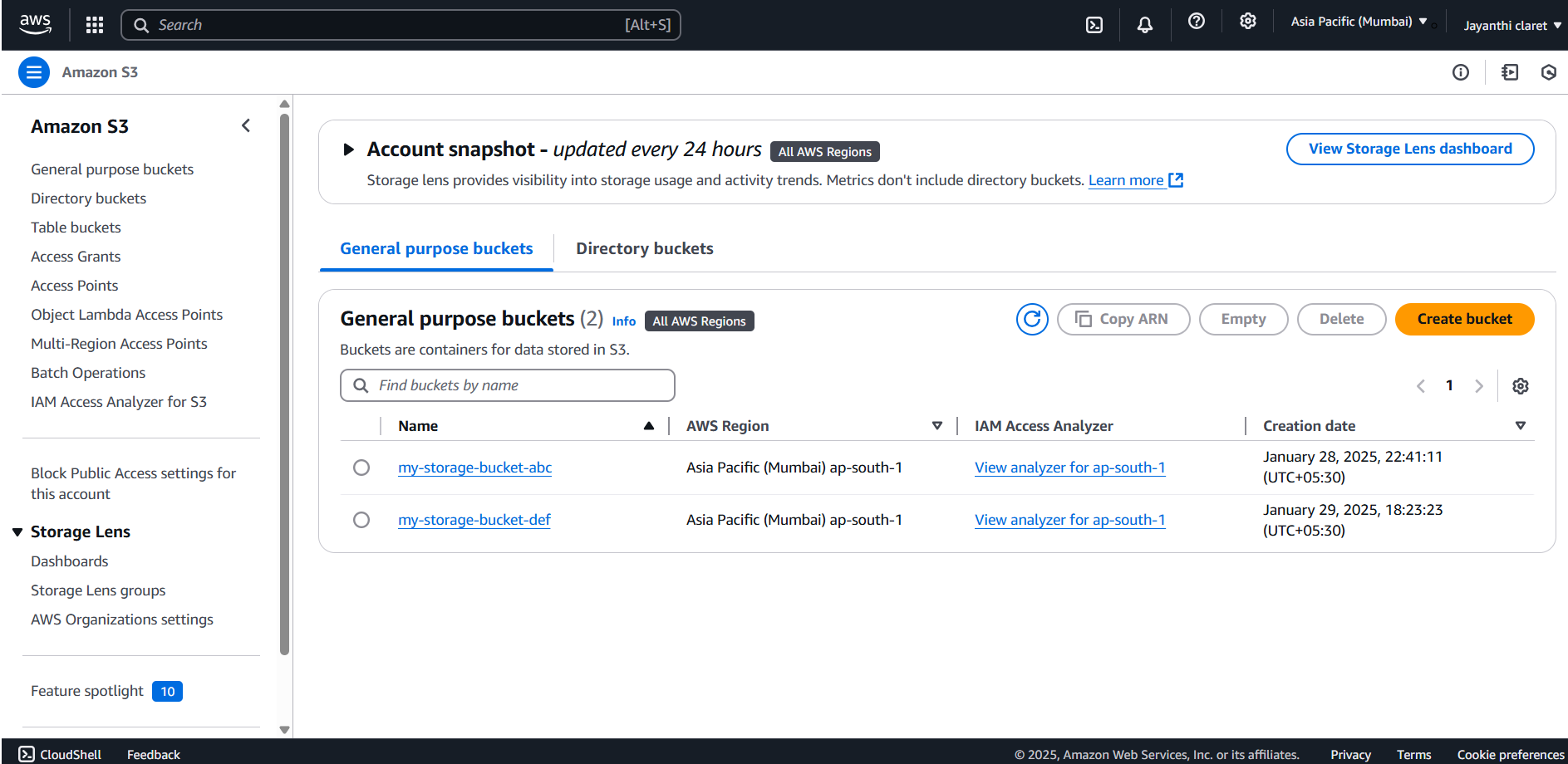
To see all storage buckets, Type **aws s3 ls** in cmd

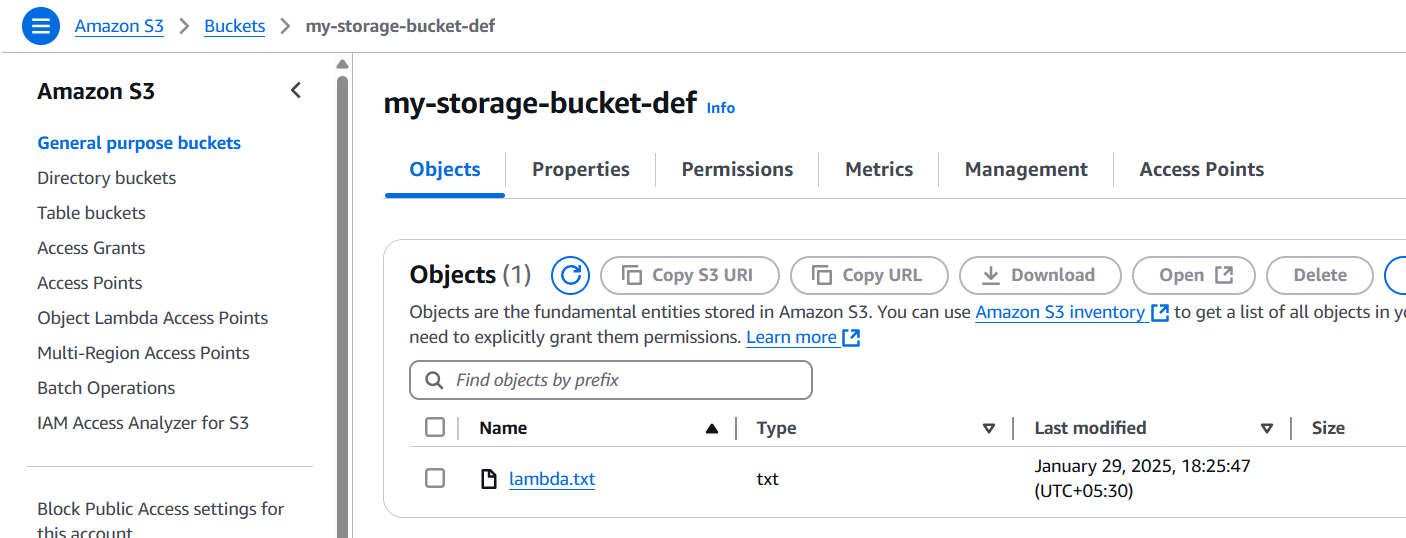
To check running EC2 instances **aws ec2 describe-instances** in cmd

Step 6:

Create an S3 Bucket by typing **aws s3 mb s3://your-unique-bucket-name** in cmd

Upload a file to S3 Bucket by typing **aws s3 cp yourfile.txt s3://your-unique-bucket-name/** in cmd

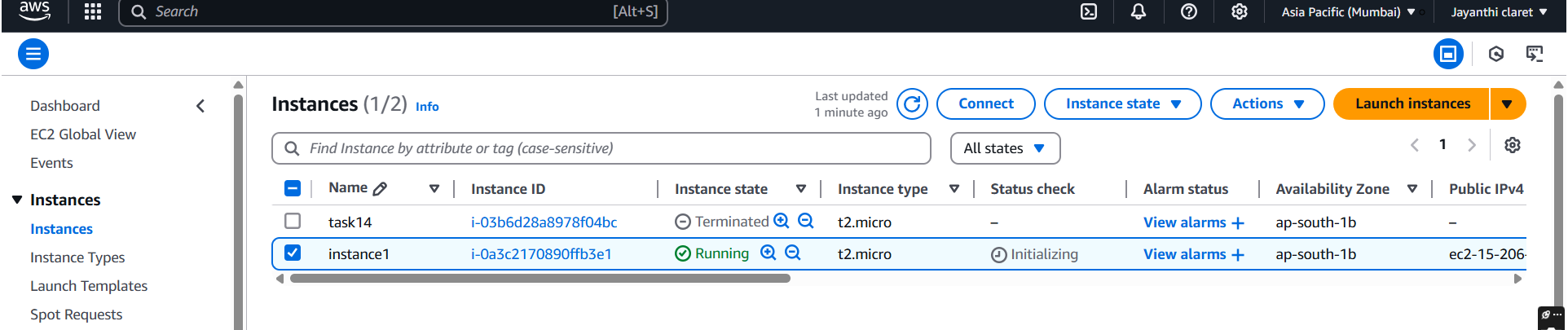




Step 7:

To Start an EC2 Instance, Type **aws ec2 start-instances --instance-ids <INSTANCE\_ID>** in cmd

Replace <INSTANCE\_ID> with your actual instance ID



**OUCOMES**

**Increased Productivity:** Automation and scripting lead to significant time savings and increased productivity for developers and operations teams.

**Reduced Errors:** Automation minimizes human error, leading to more consistent and reliable cloud deployments.

**Improved Infrastructure Management:** CLIs enable better visibility and control over cloud infrastructure, resulting in more efficient resource utilization.

**Faster Deployment Cycles:** Automated deployments and infrastructure provisioning accelerate development and deployment cycles.

**Enhanced Security Posture:** Automated security checks and fine-grained access control strengthen the security posture of cloud environments.

**Consistent resource management:** by using scripts, the resources are managed in a consistent manor, reducing configuration drift.

**Reproducible infrastructure:** by using scripts, infrastructure can be easily reproduced.