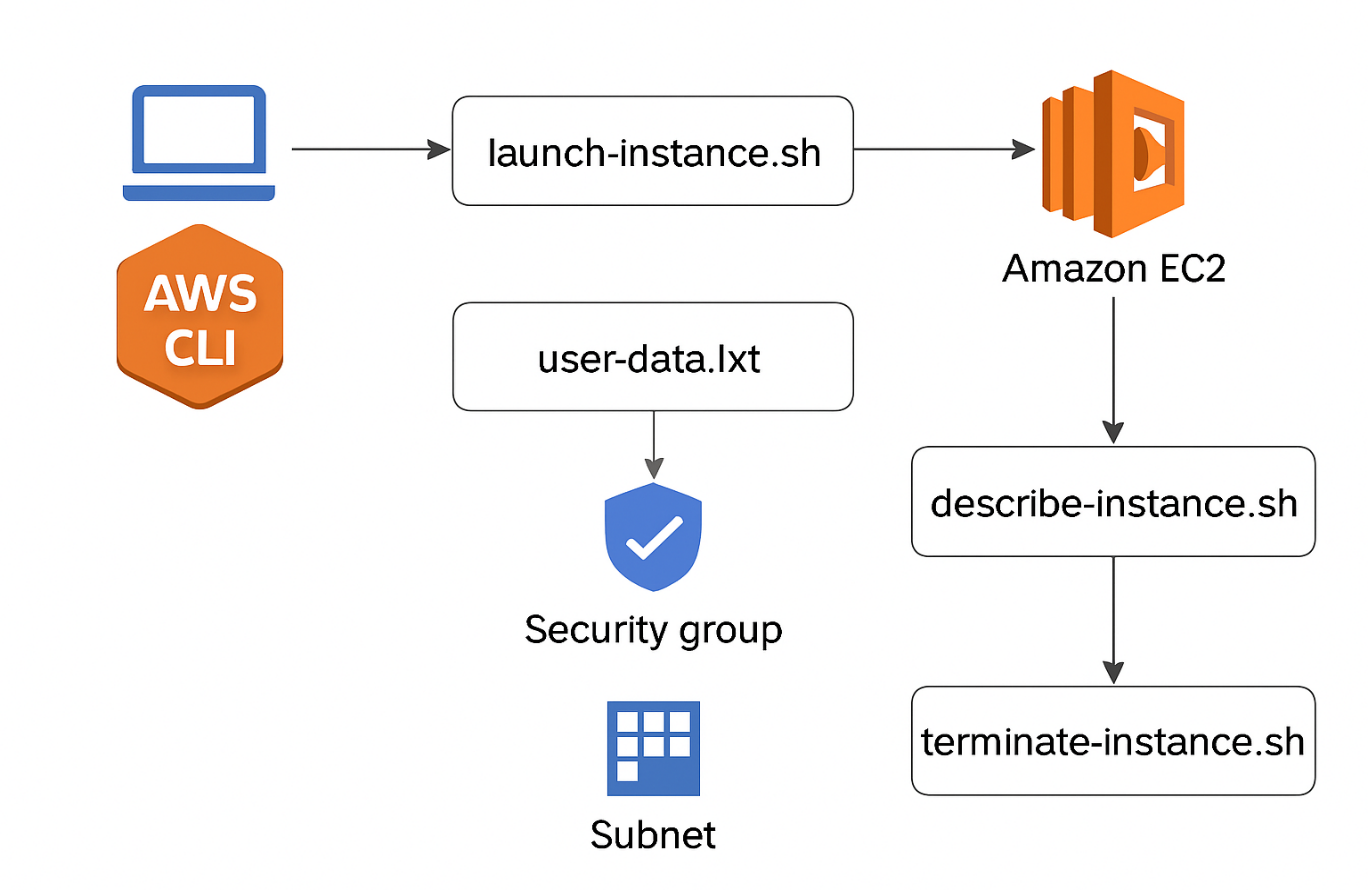
# **AWS PROJECT 4 CLI EC2 LAUNCH PROJECT – MUMBAI REGION**



This architecture diagram illustrates the flow of operations in the AWS CLI EC2 launch project:

• The process begins with the user executing AWS CLI commands from a local machine.

• The **launch-instance.sh** script provisions an Amazon EC2 instance in the Mumbai region, using a specified AMI, key pair, security group, and subnet.

• A user-data script (user-data.txt) is injected during launch to automatically install and start Apache, serving a custom HTML page.

• The instance is tagged as CLI-Instance for easy identification.

• The **describe-instance.sh** script retrieves the instance’s public IP and status.

• Finally, the terminate-instance.sh script safely decommissions the instance to avoid unnecessary charges.

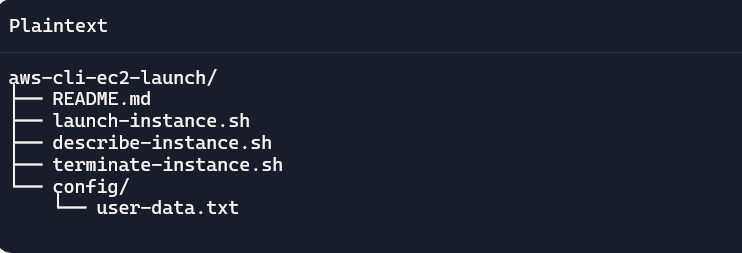
# Overview:

This project demonstrates how to launch, manage, and terminate an EC2 instance using AWS CLI in the ap-south-1 (Mumbai) region. It’s designed to showcase hands-on DevOps skills, automation scripting, and infrastructure-as-code principles—all from the command line.

# Prerequisites:

* AWS CLI installed and configured (**aws\_configure)**
* IAM user with EC2 permissions
* Existing:
* Key Pair (e.g., AWS\_CLI Key)
* Security Group ID
* Subnet ID
* Mumbai AMI ID

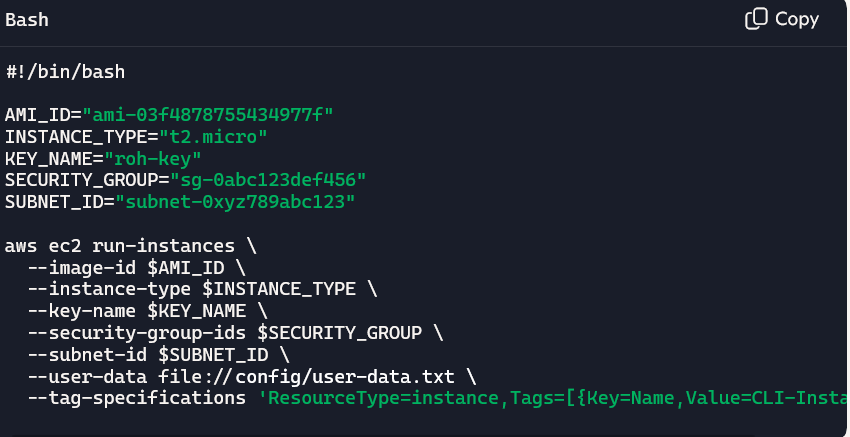
# Project Structure:



# Scripts :

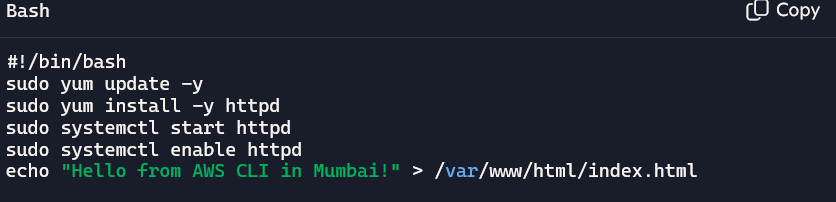
1. launch-instance.sh

Launches an EC2 instance with a basic Apache setup via user data.



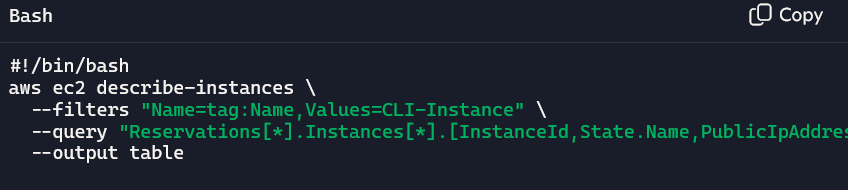
2.config/user-data.txt

Sets up Apache and a welcome page.



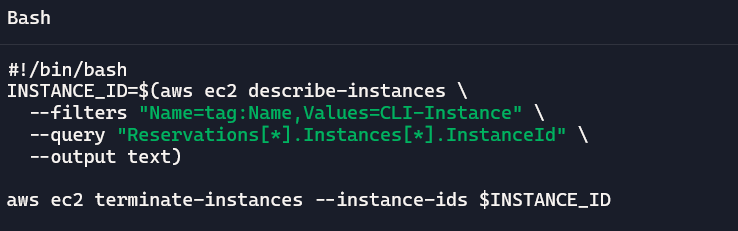
3. describe.instance.sh

Display instance details



4.terminate-instance.sh

Terminate the instances



✅ How to Use

1. Clone or download the repo.
2. Update variables in the scripts with your actual AWS values.
3. Run **launch-instance.sh** to create the EC2 instance.
4. Use **describe-instance.sh** to view instance details.
5. Visit the public IP in your browser to see the Apache welcome page.
6. Run terminate-instance.sh to clean up.

📸 Optional Enhancements

* Add logging to each script (tee or >> log.txt)
* Extend with S3 bucket creation and file upload
* Integrate with CloudWatch for monitoring
* Document outputs with screenshots in your GitHub repo