Title: Automating Apache Airflow Infrastructure on AWS Using Terraform & Jenkins

The Airflow environment will include:

A base EC2 instance running Docker

Airflow services (webserver, scheduler, worker) running as Docker containers

PostgreSQL database (RDS) for Airflow metadata

S3 bucket to store DAGs, logs, and plugins

Jenkins server to automate Terraform execution and deployment workflows

Key Tools & Services:

AWS: EC2, RDS (PostgreSQL), S3, IAM

Terraform: For provisioning infrastructure

Jenkins: For automation and CI/CD pipelines

Docker: For containerized Airflow services

GitHub: For storing code and Jenkinsfiles

Project Modules:

Terraform Infrastructure Setup

Launch EC2 for Airflow and Jenkins

Provision RDS PostgreSQL

Create S3 bucket for DAG storage

Manage IAM roles and security groups

Dockerized Airflow Deployment

Use docker-compose to deploy Airflow (webserver, scheduler, worker, etc.)

Connect Airflow to the RDS PostgreSQL

Sync DAGs from S3 into Airflow via Docker volume

Jenkins CI/CD Setup

Install Jenkins on a separate EC2

Create a Jenkins pipeline to:

Run Terraform plan, apply, and destroy

Trigger Airflow container deployment after infra setup

Optionally upload DAGs to S3

🔄 Pipeline Flow:

Jenkins triggers Terraform code to:

Provision EC2 (Airflow)

Provision RDS and S3

Once infrastructure is ready:

Jenkins triggers a shell script to deploy Airflow via Docker

Jenkins also syncs DAG files from GitHub or S3

Optional stages:

Ansible (or shell) to install extra software

Auto clean-up using terraform destroy via Jenkins

📝 Deliverables:

Terraform code for AWS infra (EC2, RDS, S3)

Jenkinsfile with parameterized pipeline stages:

Plan / Apply / Destroy infra

Deploy or Skip Airflow container

Docker-compose file to launch Airflow

Documentation (README.md) explaining full setup flow

🧑‍🏫 Intern Learning Outcomes:

Hands-on with AWS services like EC2, RDS, S3

Write Terraform modules and Jenkins pipelines

Understand Airflow architecture and deployment

Build CI/CD pipelines using Jenkins

Gain experience in containerized deployment using Docker