CI/CD Pipeline using Jenkins, Docker & AWS EC2

Project Overview

This project demonstrates the implementation of a CI/CD pipeline using Jenkins, Docker, and Docker Compose, deployed on an AWS EC2 instance. The pipeline automatically pulls source code from GitHub, builds a Docker image, and deploys the application in a containerized environment.

System Architecture

Developer pushes code to GitHub →Jenkins pulls the code →Jenkins builds Docker image (CI) →Jenkins deploys the application using Docker Compose (CD) →Application runs on AWS EC2 and is accessed via public IP.

Steps

This project is divided into two parts :
Part A - For CI
Part B - For CD

Part A Section

Step 1: Create a folder for CI files, then pushes the files to github. [Github repo: CI-CD]

Step 2: Create EC2 instance.

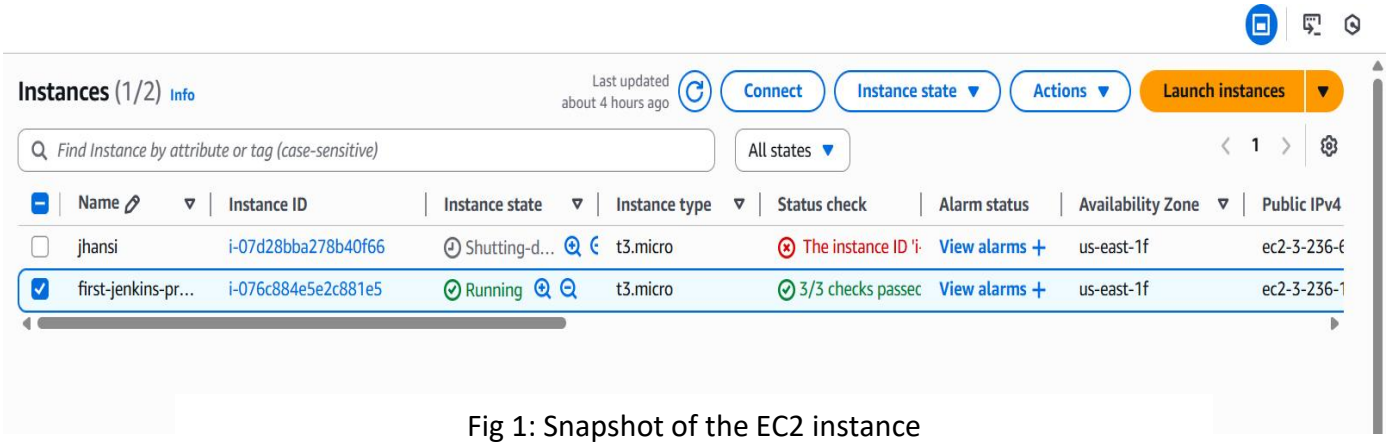

Fig 1: Snapshot of the EC2 instance

Add inbound rules for jenkins. [port:8080]

Step 3: Run the ssh - i command in the terminal.
ssh - I <key name> aws@<instance ip address>

Step 4: Download the Jenkins and Docker inside this instance.

Step 5: Open the jenkins through url: http://<ip address>:8080

Step 6: Open and install inline pluggins.

Step 7: Then restart the jenkins.

Step 8: In settings inside pluggins, search for “Docker Pipeline” select it and install it. Then once again restart the jenkins.

Step 9: Jenkins pull the files from the github, then runs it.

Step 10:

Fig 2 : Final console output

Part B Section

Step 1: Create a folder for CD files, then pushes the files to github. [Github repo: Jenkins-continue]

Step 2: Create EC2 instance.

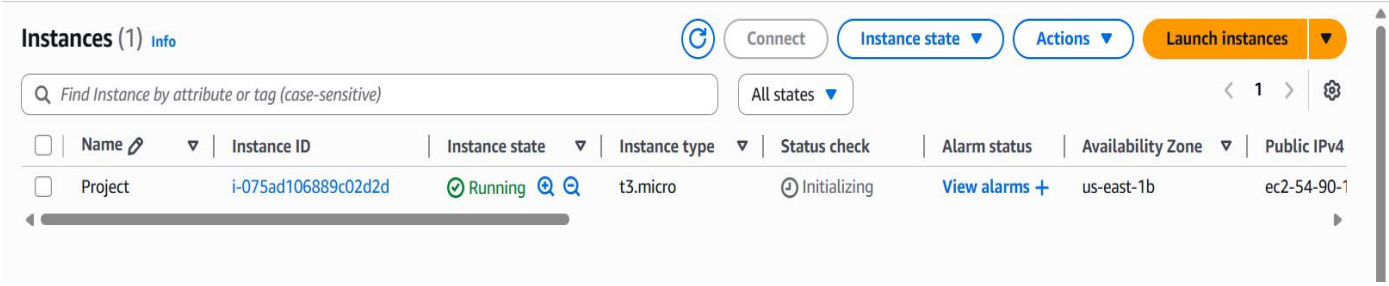

Fig 3: Snapshot of the EC2 instance

Add 2 inbound rules for jenkins. [port:8080 - jenkins and port:8081 to avoid the conflict]

Step 3: Run the ssh - i command in the terminal.
ssh - I <key name> aws@<instance ip address>

Step 4: Download the Jenkins, Docker and Docker compose inside this instance.

Step 5: Open the jenkins through url: http://<ip address>:8080

Step 6: Open and install inline pluggins.

Step 7: Then restart the jenkins.

Step 8: In settings inside pluggins, search for “Docker Pipeline” select it and install it. Then once again restart the jenkins.

Step 9: Jenkins pull the files from the github, then runs it.

Step 10:

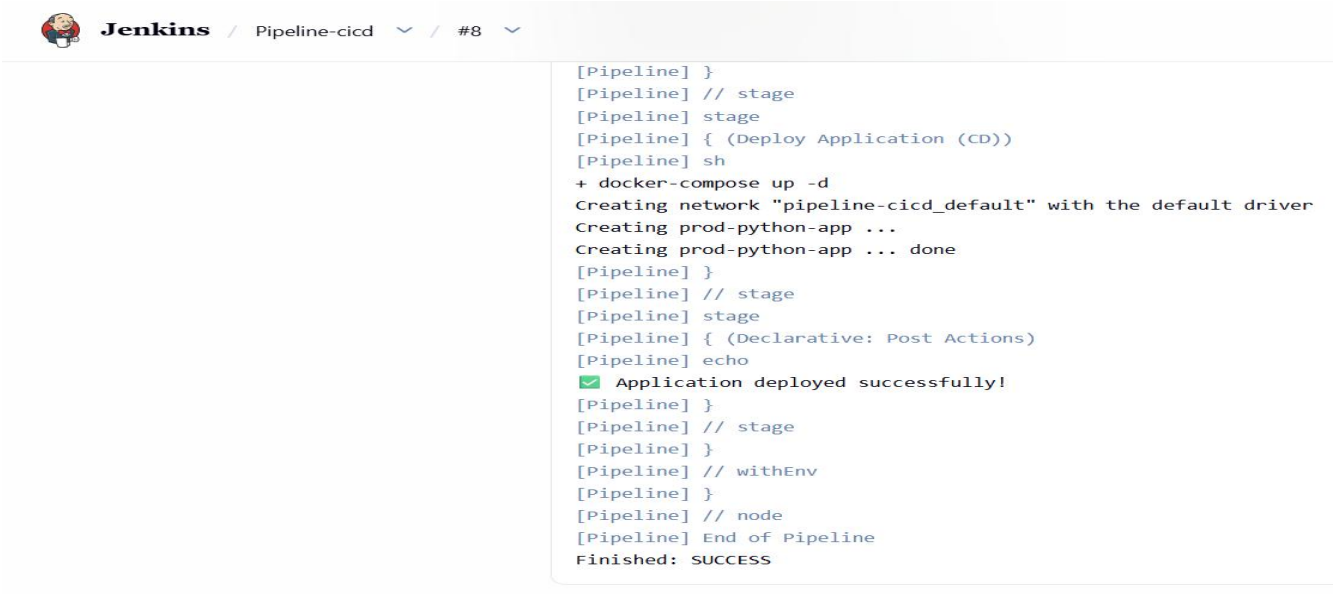

Fig 4 : Final console output

Step 11: To access the application. Open the url : http://<EC2-PUBLIC-IP>:8081

Step 12:

Fig 5 : Application Successfully Deployed using Jenkins