□ Standard Operating Procedure (SOP)

🏷️ Project Title: Full Stack DevOps – Cloud Deployment, Monitoring & CI/CD Automation

🥇 1 – Cloud Deployment

🎯 Objective

Deploy the full stack (frontend + backend + MySQL) on an AWS EC2 Ubuntu instance using Docker and Docker Compose with persistent MySQL storage.

□ Steps Followed

- **Clone the Repository**

  git clone https://github.com/DhruvShah0612/fusionpact-devops-challenge.git

  cd fusionpact-devops-challenge

- **Build and Run Containers**

  sudo docker-compose up -d –build

- **Verify Running Containers**

  sudo docker ps

Frontend Dockerfile

```
FROM nginx:alpine

# Remove default nginx html
RUN rm -rf /usr/share/nginx/html/*

# Copy everything in current folder (frontend/) to nginx html folder
COPY . /usr/share/nginx/html/

# Rename the file to index.html safely
RUN [ -f /usr/share/nginx/html/Devops_Intern.html ] && mv /usr/share/nginx/html/Devops_Intern.html /usr/share/nginx/html/index.html || true

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]
```

Backend Dockerfile

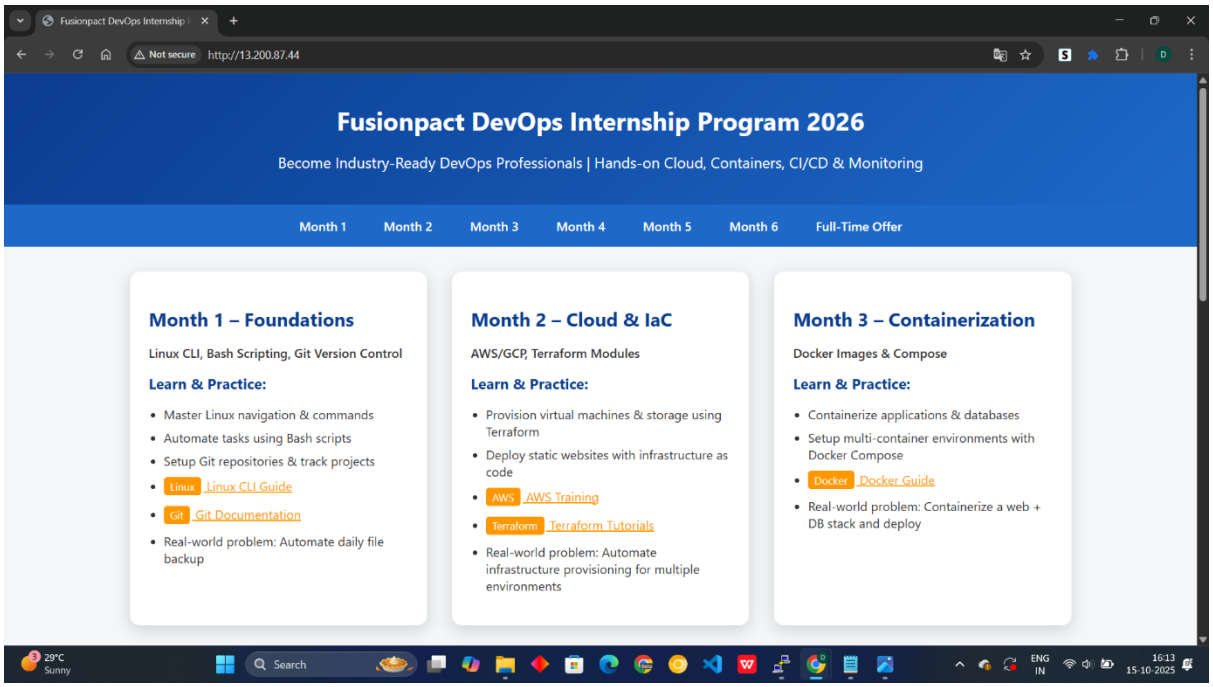
```
FROM python:3.11-slim

WORKDIR /app

RUN apt-get update && apt-get install -y default-libmysqlclient-dev build-essential

# Copy requirements (no 'backend/' here)
COPY requirements.txt .

RUN pip install --no-cache-dir -r requirements.txt

# Copy app folder (no 'backend/' here)
COPY app ./app

EXPOSE 8000

CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]
```

Docker-compose.yml file

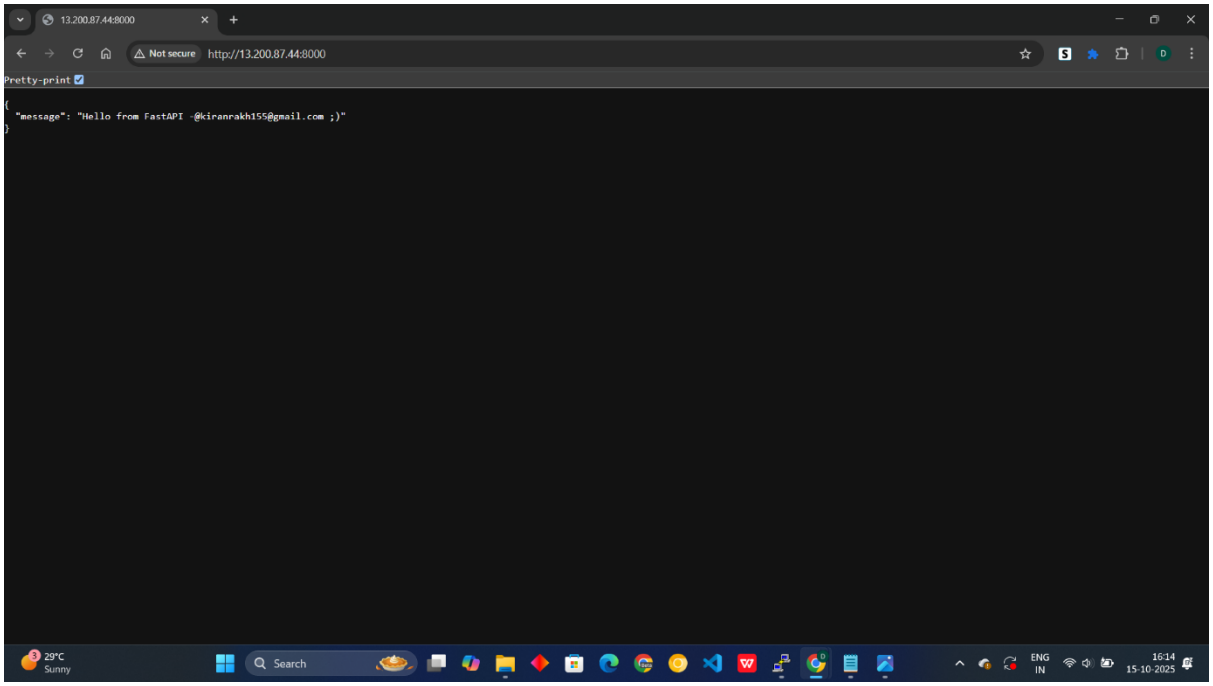
```yaml
version: '3.9'

services:
  # -----------------
  # MySQL Database
  # -----------------
  db:
    image: mysql:8.0
    container_name: mysql-db
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: fusionpact
      MYSQL_USER: fusionuser
      MYSQL_PASSWORD: fusionpass
    ports:
      - "3306:3306"
    volumes:
      - mysql_data:/var/lib/mysql
    networks:
      - fusionnet

  # -----------------
  # Backend - FastAPI
  # -----------------
  backend:
    build: ./backend
    container_name: fusion-backend
    restart: always
    environment:
      DATABASE_HOST: db
      DATABASE_USER: fusionuser
      DATABASE_PASSWORD: fusionpass
      DATABASE_NAME: fusionpact
      DATABASE_PORT: 3306
    depends_on:
      - db
    ports:
      - "8000:8000"
    networks:
      - fusionnet

  # -----------------
  # Frontend - React/Vite
  # -----------------
  frontend:
    build: ./frontend
    container_name: fusion-frontend
    restart: always
    ports:
      - "80:80"
    depends_on:
      - backend
    networks:
      - fusionnet

# -----------------
# Named Volumes & Network
# -----------------
volumes:
  mysql_data:

networks:
  fusionnet:
    driver: bridge
```

➢ **Access Services:**

Frontend → http://<EC2-IP>/



Backend → http://<EC2-IP>:8000/



💾 MySQL Data Persistence Test

CREATE TABLE test_table (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(50));

INSERT INTO test_table (name) VALUES ('Level1 Test');

SELECT * FROM test_table;

```
root@ip-172-31-33-166:/home/ubuntu/fusionpact-devops-challenge# docker exec -it mysql-db mysq
l -u fusionuser -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.43 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE fusionpact;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+----------------------+
| Tables_in_fusionpact |
+----------------------+
| test_table           |
+----------------------+
1 row in set (0.01 sec)

mysql>
```

🥈 2 – Monitoring & Observability

🎯 Objective

Implement observability using Prometheus, Node Exporter, and Grafana for both application and infrastructure metrics.

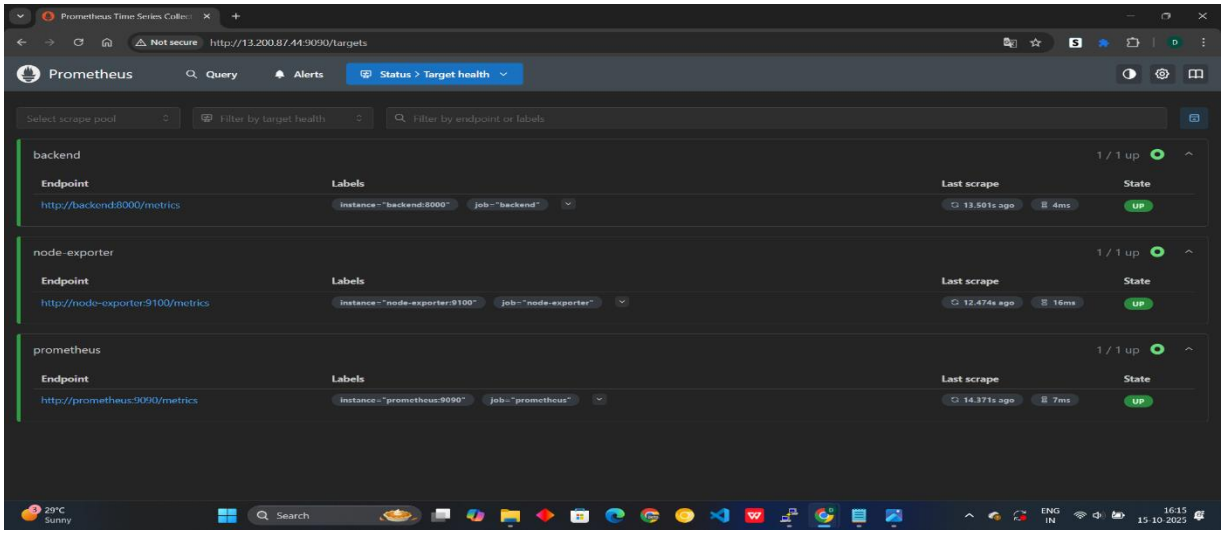
📄 prometheus.yml

```
global:
  scrape_interval: 15s

scrape_configs:
  # Prometheus itself
  - job_name: 'prometheus'
    static_configs:
      - targets: ['prometheus:9090']

  # FastAPI backend
  - job_name: 'backend'
    static_configs:
      - targets: ['backend:8000']

  - job_name: 'node-exporter'
    static_configs:
      - targets: ['node-exporter:9100']
```

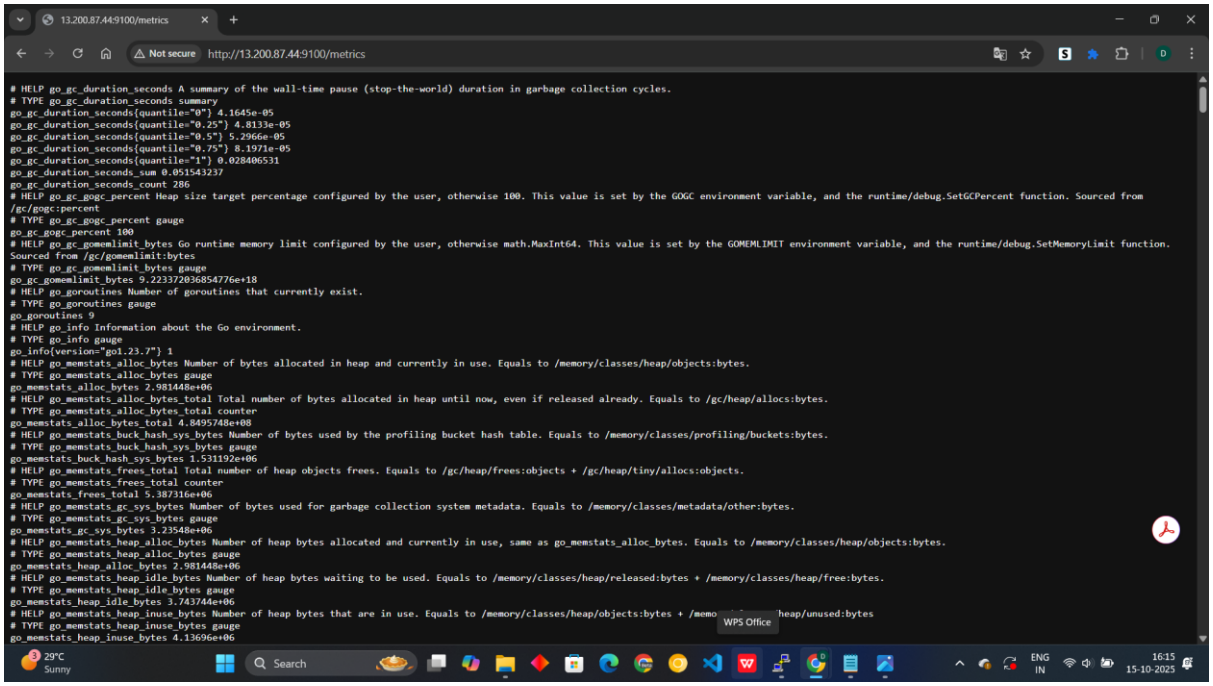
⚙️ Setup Commands

docker-compose -f docker-compose.yml up -d

Prometheus → http://<EC2-IP>:9090



Prometheus targets page (showing backend + node_exporter)

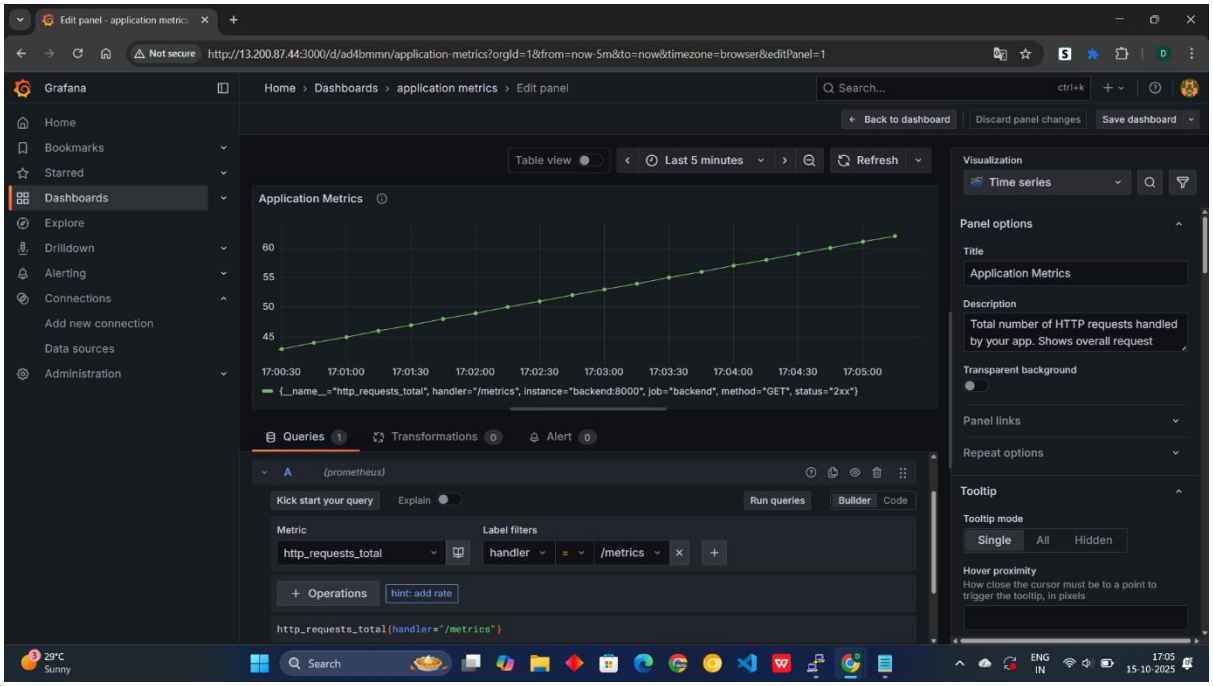


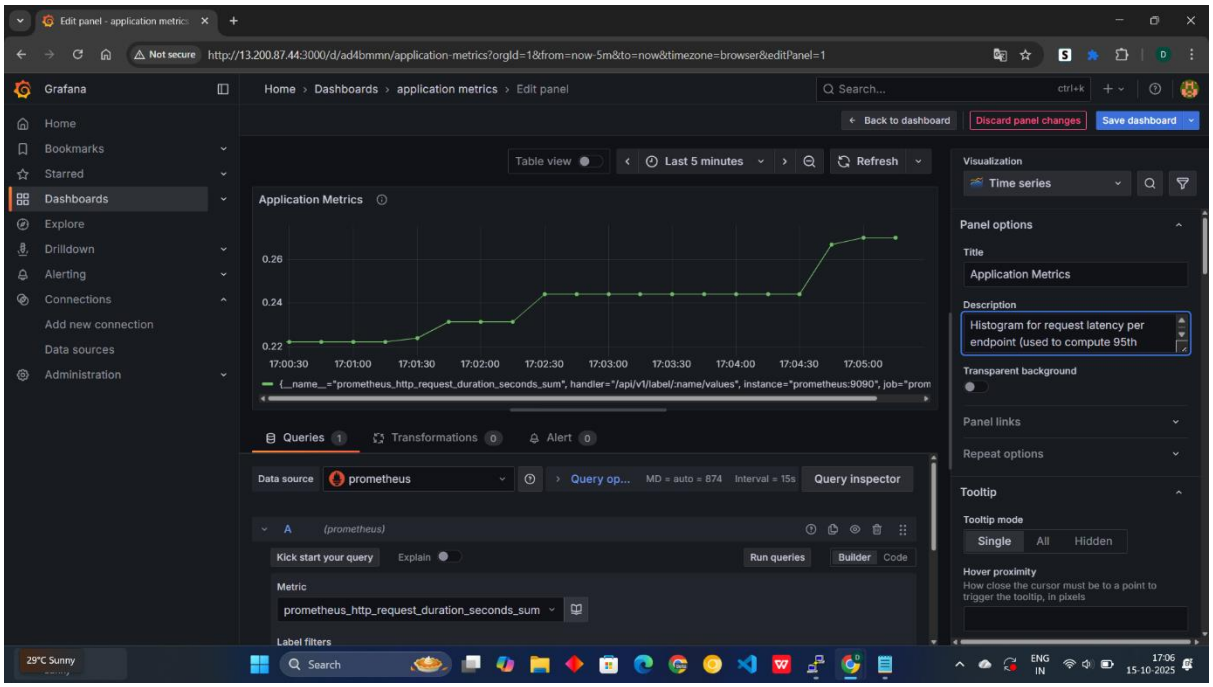
- **Grafana →** http://<EC2-IP>:3000
    - Username: admin
    - Password: admin

➢ **Add Prometheus as data source:**
http://prometheus:9090

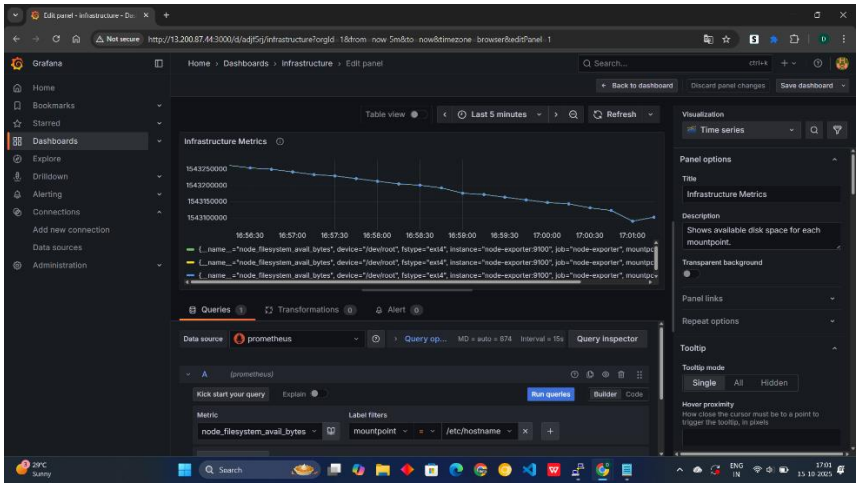
Dashboards Created

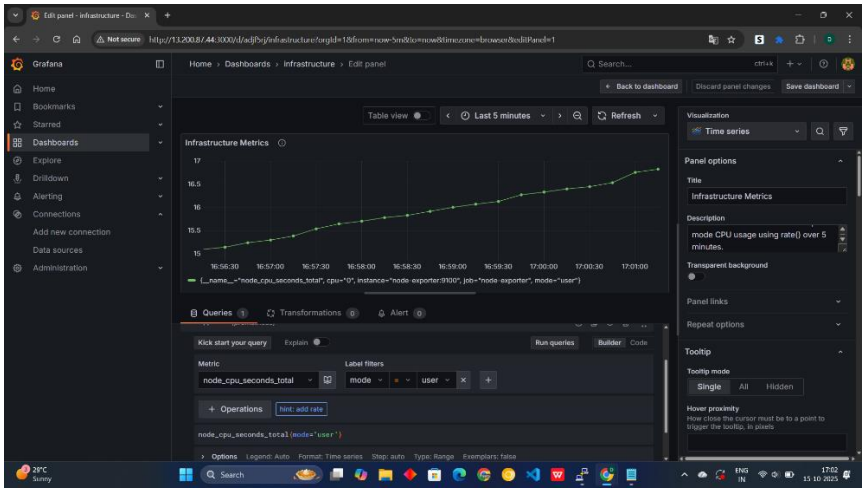
Application Metrics

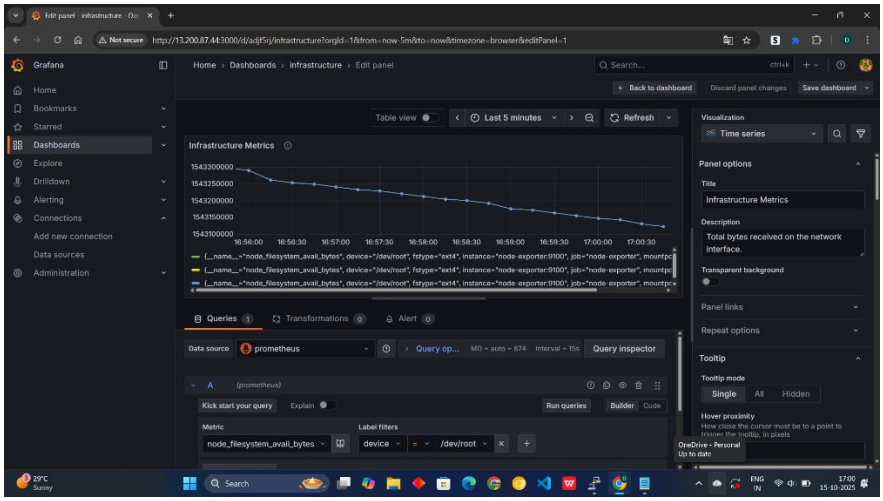




Infrastructure Metrics







🥉 3 – CI/CD Automation

🎯 Objective

Automate the build, push, and deploy stages using Jenkins with Docker integration on AWS EC2.

✅ Final Jenkinsfile

```
pipeline {
    agent any
    environment {
        DOCKER_HUB_REPO = 'dhruvshah0612/fusionpact'
    }

    stages {
        stage('Checkout Code') {
            steps {
                git branch: 'main', credentialsId: 'github-https', url: 'https://github.com/DhruvShah0612/fusionpact-devops-challenge.git'
            }
        }

        stage('Docker Test') {
            steps {
                sh 'docker --version'
            }
        }
        stage('Build Docker Image') {
            steps {
                sh 'docker build -t $DOCKER_HUB_REPO:${BUILD_NUMBER} .'
            }
        }
        stage('Push to Docker Hub') {
            steps {
                withCredentials([usernamePassword(credentialsId: 'dockerhub-creds', usernameVariable: 'USER', passwordVariable: 'PASS')]) {
                    sh '''
                    echo "$PASS" | docker login -u "$USER" --password-stdin
                    docker push $DOCKER_HUB_REPO:${BUILD_NUMBER}
                    '''
                }
            }
        }

        stage('Deploy') {
            steps {
                sh '''
                docker stop fusionpact-app || true
                docker rm fusionpact-app || true
                docker run -d --name fusionpact-app -p 8080:80 $DOCKER_HUB_REPO:${BUILD_NUMBER}
                '''
            }
        }
    }

    post {
        always {
            echo 'Cleaning up dangling images...'
            sh 'docker image prune -f || true'
        }
        success {
            echo '✅ Pipeline executed successfully!'
        }
        failure {
            echo '❌ Pipeline failed! Please check logs.'
        }
    }
}
```

⚙️ Jenkins Setup Command

```
docker run -d \
  --name jenkins-docker \
  -p 8080:8080 -p 50000:50000 \
  -v /home/ubuntu/jenkins_home:/var/jenkins_home \
  -v /var/run/docker.sock:/var/run/docker.sock \
  -e DOCKER_GROUP_ID=$(getent group docker | cut -d: -f3) \
  jenkins/jenkins:lts
```

🔑 Jenkins Credentials

ID	Type	Description
github-https	GitHub Token	For code checkout
dockerhub-creds	Docker Hub Login	For image push
ec2-ssh-key	SSH Key (optional)	For remote deployment

🌐 Access

- Jenkins Web UI: http://:8080

This setup ensures a fully automated CI/CD pipeline from GitHub → Jenkins → Docker → EC2 deployment.