# AI Artistic Style Service Assignment

## Dockerized AI Artistic Style Service

### Docker Compose File:

version: '3.8'

services:

  ai-artistic-style-service:

    image: urmsandeep/ai-artistic-style-service:latest

    container\_name: ai-artistic-style

    ports:

      - "5001:5001"

    restart: always

  prometheus:

    image: prom/prometheus:latest

    container\_name: prometheus

    volumes:

      - ./prometheus/prometheus.yml:/etc/prometheus/prometheus.yml

    ports:

      - "9090:9090"

  grafana:

    image: grafana/grafana:latest

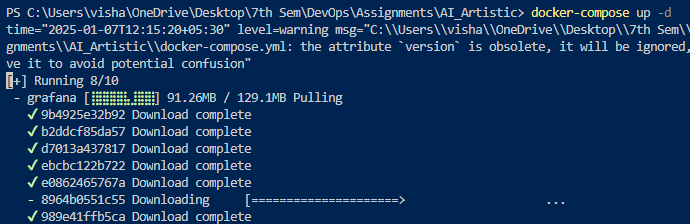
    container\_name: grafana

    ports:

      - "3000:3000"

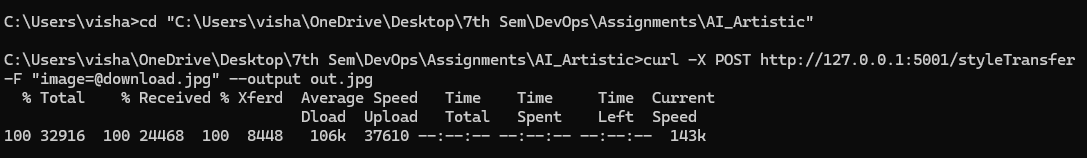
### Running Docker Compose:

Command:  
docker-compose up -d



## 2. API Testing

### Using Curl to Test the API Endpoint:

Command:  
curl -X POST http://127.0.0.1:5001/styleTransfer -F "image=@input.jpg" --output out.jpg  
  


Input and Output Images:



## 

## 3. API Monitoring

### Prometheus Configuration:

global:

  scrape\_interval: 15s

scrape\_configs:

  - job\_name: "ai-artistic-style-service"

    static\_configs:

      - targets: ["ai-artistic-style:5001"]

  - job\_name: "prometheus"

    static\_configs:

      - targets: ["localhost:9090"]

### Grafana Dashboard:

Screenshot Placeholder:  
Insert a screenshot of Grafana showing API performance metrics here.

## 5. CI/CD Pipeline

### Jenkinsfile Content:

pipeline {

    agent any

    stages {

        stage('Pull Docker Image') {

            steps {

                script {

                    powershell '''

                    docker pull urmsandeep/ai-artistic-style-service:latest

                    '''

                }

            }

        }

        stage('Run Tests') {

            steps {

                script {

                    powershell '''

                    # Start the container in detached mode

                    docker run -d --name ai-artistic-style-service urmsandeep/ai-artistic-style-service:latest

                    # Install pytest inside the running container

                    docker exec ai-artistic-style-service pip install pytest

                    # Run the tests

                    docker exec ai-artistic-style-service pytest tests/

                    # Stop and remove the container after testing

                    docker stop ai-artistic-style-service

                    docker rm ai-artistic-style-service

                    '''

                }

            }

        }

        stage('Deploy Service') {

            steps {

                script {

                    powershell '''

                    docker-compose down

                    docker-compose up -d

                    '''

                }

            }

        }

        stage('Verify Deployment') {

    steps {

        script {

            bat '''

            ping 127.0.0.1 -n 6 > nul

            if exist test.jpg (

                curl -X POST http://127.0.0.1:5001/styleTransfer -F "image=@test.jpg" --output styled\_output.jpg

            ) else (

                echo File "test.jpg" not found & exit /b 1

            )

            '''

        }

    }

}

    }

    post {

        always {

            script {

                powershell '''

                docker system prune -f

                '''

            }

        }

        success {

            echo 'Pipeline executed successfully. The service is running and functional!'

        }

        failure {

            echo 'Pipeline failed. Check logs for errors.'

        }

    }

}

### Pipeline Output:

## 

