#### **Problem Statement:**

You are tasked with setting up a CI/CD pipeline using Jenkins to streamline the deployment process of a simple Java application. The pipeline should accomplish the following tasks:

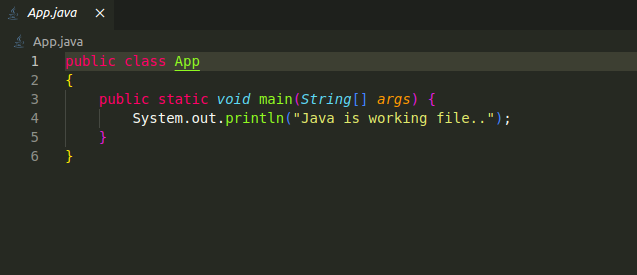
1. **Fetch the Dockerfile**: The pipeline should clone a GitHub repository containing the source code of the Java application and a Dockerfile.
2. **Create a Docker Image**: The pipeline should build a Docker image from the fetched Dockerfile.
3. **Push the Docker Image**: The pipeline should push the created Docker image to a specified DockerHub repository.
4. **Deploy the Container**: The pipeline should deploy a container using the pushed Docker image.

#### **Deliverables:**

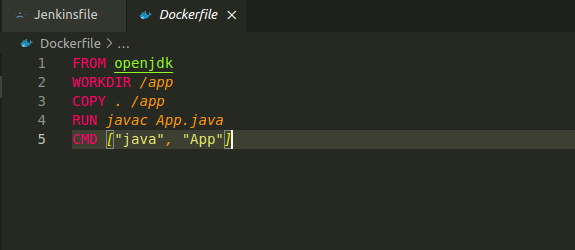
1. **GitHub Repository**: A GitHub repository containing:
   * The source code of a simple Java application.
   * A Dockerfile for building the Docker image.
2. **Jenkins Pipeline Script**: A Jenkinsfile (pipeline script) that:
   * Clones the GitHub repository.
   * Builds the Docker image.
   * Pushes the Docker image to DockerHub.
   * Deploys a container using the pushed image.
3. **DockerHub Repository**: A DockerHub repository where the Docker images will be stored.
4. **Jenkins Setup**:
   * Jenkins installed and configured on a local Ubuntu machine.
   * Required plugins installed (e.g., Git, Docker, Pipeline).
5. **Documentation**: Detailed documentation explaining:
   * How to set up the local Jenkins environment.
   * Configuration steps for the pipeline.
   * Instructions for verifying the deployment.

Here is screenshot:

Here is the Java app



Step2: create a docker file for this app



Step3:

Created the Jenkinsfile for that:  
pipeline {

agent any

environment {

DOCKER\_CREDENTIALS\_ID = 'docker'

DOCKERHUB\_REPOSITORY = 'mtaori/javaimage:v14'

// here we have to parameterised the image so that if we run the pipeline it creates new image enery time

}

stages {

stage('Checkout') {

steps {

git 'https://github.com/mtaori/day14\_task.git'

}

}

stage('Build Docker Image') {

steps {

script {

*def* customImage = docker.build("${env.DOCKERHUB\_REPOSITORY}")

docker.withRegistry('https://index.docker.io/v1/', "${env.DOCKER\_CREDENTIALS\_ID}") {

customImage.push()

}

}

}

}

stage('Run Docker Image') {

steps {

script {

*def* container = docker.image("${env.DOCKERHUB\_REPOSITORY}").run('-d')

sleep(time: 10, unit: 'SECONDS')

sh "docker logs ${container.id}"

}

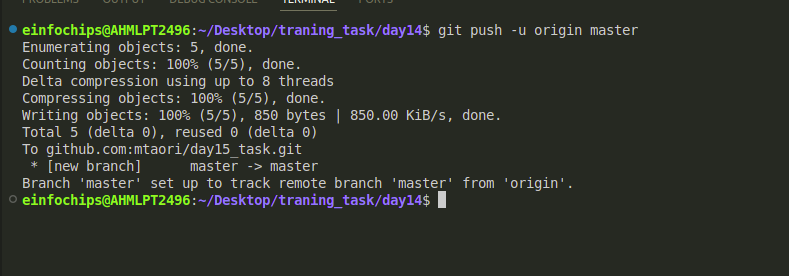
}

}

}

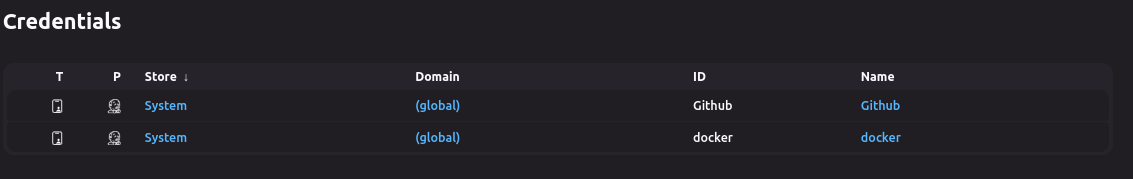
}

Step 5: Push the all the file in github repository



Step6:

Add the github credential



Step 7:

Run the pipeline:

