**DevOps Assignments (Local Setup, No Cloud Required)**

**1. CI/CD Pipeline with Jenkins + Docker**

**Objective:** Set up a local Jenkins pipeline to build, test, and run a Dockerized app.

**• Tasks:**

**o** Set up Jenkins locally (use Docker)

**o** Use GitHub repo as the source

**o** Build and test a sample app (Node.js, Python, etc.)

**o** Build Docker image and run container on successful build

**Setup instructions :**

* Runs Jenkins inside a Docker container
* Pulls your Node.js app code from GitHub
* Builds and runs your app using Docker
* Sends email or Slack notifications based on build success or failure

### **Run Jenkins Using Docker (Locally)**

Open your terminal and run this command to start Jenkins in Docker:  
  
docker run -d --name jenkins \

-p 8080:8080 -p 50000:50000 \

-v jenkins\_home:/var/jenkins\_home \

jenkins/jenkins:lts

After Jenkins starts, go to<http://localhost:8080> in your browser.  
To get the admin password for the first-time setup, run:  
  
docker exec jenkins cat /var/jenkins\_home/secrets/initialAdminPassword

Paste this password into the Jenkins setup screen and finish the setup (install suggested plugins).

**Install Jenkins Plugins**

In Jenkins, go to **Manage Jenkins → Plugins**, and install these:

* Docker
* Git
* Pipeline

### **Connect Jenkins to GitHub**

In Jenkins, click **“New Item” → Choose “Pipeline” → Name your job → Click OK**.  
In the job config:  
 Set **Pipeline script from SCM** Choose **Git** Add your GitHub repository URL

### **Create a Sample Node.js App**

**app.js file:**

const express = require('express');

const app = express();

const port = process.env.PORT || 3000;

app.get('/', (req, res) => {

res.send('Hello from AWS Elastic Beanstalk!');

});

app.listen(port, () => {

console.log(`App listening on port ${port}`);

});

**Dockerfile** to containerize the app:

FROM node:18

WORKDIR /app

COPY package\*.json ./

RUN npm install

COPY . .

EXPOSE 3000

CMD ["node", "app.js"]

**Jenkins Pipeline Script (Jenkinsfile)**

Jenkinsfile in GitHub repo root:

groovy

pipeline {

agent any

tools {

nodejs 'Node18'

}

environment {

DOCKER\_IMAGE = 'my-node-app'

}

stages {

stage('Clone Repo') {

steps {

git branch: 'main', url: 'https://github.com/ramya-create/nodeJs-application.git'

}

}

stage('Install Dependencies') {

steps {

bat 'npm install'

}

}

stage('Build Docker Image') {

steps {

bat 'docker build -t my-node-app .'

}

}

stage('Run Docker Container') {

steps {

bat 'docker rm -f node-app || true'

bat 'docker run -d --name node-app -p 3000:3000 my-node-app'

}

}

}

post {

always {

echo 'Cleaning workspace and releasing agent...'

cleanWs()

}

success {

echo '✅ Build and deployment successful!'

}

failure {

echo '❌ Build failed!'

}

}

}