

Jenkins + GitHub + Docker CI/CD Pipeline

Jenkins + GitHub + Docker CI/CD Pipeline

Tools Used

GitHub: Source code hosting

Docker Hub: Image registry

Jenkins: CI/CD automation

AWS EC2 (Amazon Linux): Jenkins host

Step-by-Step Setup

1. Install Required Packages on EC2

```
sudo yum update -y
```

```
sudo yum install git -y
```

2. Install Docker

```
sudo yum install docker -y
```

```
sudo systemctl start docker
```

```
sudo systemctl enable docker
```

3. Install Jenkins

```
wget -O /etc/yum.repos.d/jenkins.repo <https://pkg.jenkins.io/redhat-stable/jenkins.repo>
```

```
sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key>
```

```
sudo yum install java-21-amazon-corretto
```

```
sudo yum install jenkins -y
```

```
sudo systemctl start jenkins
```

```
sudo systemctl enable jenkins
```

4. Setup Jenkins

Jenkins + GitHub + Docker CI/CD Pipeline

Visit: `http://<your-ec2-public-ip>:8080`

Unlock Jenkins:

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

Install suggested plugins.

Create an admin user.

Local Project Setup

1. Initialize Your Project Directory

```
mkdir node-hello-world
```

```
cd node-hello-world
```

2. Add Project Files

Include at least:

`app.js`

`Dockerfile`

`docker-compose.yml`

`package.json`

`Jenkinsfile`

3. Create Dockerfile

FROM node:18-alpine

```
WORKDIR /usr/src/app
```

```
COPY package*.json ./
```

```
RUN npm install
```

```
RUN npm ci --only=production)
```

```
COPY . .
```

Jenkins + GitHub + Docker CI/CD Pipeline

EXPOSE 3000

CMD ["npm", "start"]

4. Create docker-compose.yml

version: '3'

services:

app:

image: gopi02/node-hello-world:latest

ports:

- "3000:3000"

restart: always

5. Create Jenkinsfile

pipeline {

agent any

environment {

DOCKER_IMAGE = 'gopi02/node-hello-world'

DOCKER_TAG = "\${env.BUILD_NUMBER}"

}

stages {

stage('Checkout') {

steps {

checkout([

\$class: 'GitSCM',

branches: [[name: '*/main']],

extensions: [],

userRemoteConfigs: [[

Jenkins + GitHub + Docker CI/CD Pipeline

```
    url: '<https://github.com/Gopik02/Final-Project.git>',
    credentialsId: 'git-creds'
  ]]
})
}
```

stage('Install Dependencies') {

```
  steps {
    sh 'npm install'
  }
}
```

stage('Build Docker Image') {

```
  steps {
    script {
      sh "docker build -t ${DOCKER_IMAGE}:${DOCKER_TAG} ."
    }
  }
}
```

stage('Push to Docker Hub') {

```
  steps {
    script {
      withCredentials([usernamePassword(
        credentialsId: 'docker-creds',
        usernameVariable: 'DOCKER_USER',
        passwordVariable: 'DOCKER_PASS'
      )]) {
```

Jenkins + GitHub + Docker CI/CD Pipeline

```
sh """
    echo ${DOCKER_PASS} | docker login -u ${DOCKER_USER} --password-stdin
    docker push ${DOCKER_IMAGE}:${DOCKER_TAG}
"""

}

}

}

}

}

}

}
```

Jenkins Freestyle Job (Alternative to Jenkinsfile)

If you do not use a Jenkinsfile, you can configure a freestyle project as follows:

Source Code Management: Git

Git URL: `https://github.com/<your-username>/node-hello-world.git`(git repo.)

Build Step (Execute Shell):

```
docker login -u <your-dockerhub-username> -p <your-dockerhub-password>
```

```
docker build -t <your-dockerhub-username>/quothub-app:v1 .
```

```
docker push <your-dockerhub-username>/quothub-app:v1
```

Set Up GitHub Webhook

Expose Jenkins to the Web (optional: configure a reverse proxy or use a service such as ngrok).

Find Jenkins Webhook URL:`http://<jenkins-url>/github-webhook/`

In Your GitHub Repository:

Go to Settings > Webhooks.

Payload URL: your Jenkins webhook URL

Content type: application/json

Event: Just push events

Final Test

Push your code to GitHub:

Jenkins + GitHub + Docker CI/CD Pipeline

```
git add .
```

```
git commit -m "Initial commit"
```

```
git push origin main
```

Jenkins should automatically trigger the pipeline on push.

Credentials Setup in Jenkins

Go to Jenkins Dashboard Manage Jenkins Credentials and add:

DockerHub username/password (ID: docker-creds)

GitHub token (if using a private repository)