

Jenkins

Jenkins is an open-source automation server used for Continuous Integration (CI) and Continuous Deployment (CD) in software development. It helps automate the build, test, and deployment process, making DevOps workflows efficient and reliable.

Maven Overview

Maven is a build automation tool primarily used for Java projects. It simplifies project management by handling dependencies, building the project, and automating testing and deployment.

GitHub

GitHub is a web-based platform for version control and collaboration using Git. It allows developers to store, manage, track changes, and collaborate on projects.

Create GitHub account and create one repository

Name: puropale project

AWS EC2

Create Jenkins instance and run connect to git bash

```
ssh -i "Jenkins.pem" ec2-user@ec2-13-61-14-170.eu-north-1.compute.amazonaws.com
```

Then install Jenkins

Add Jenkins Repository: °

Import the GPG key:

```
→ wget -q -O - https://pkg.jenkins.io/debian/ jenkins.io.key |  
sudo apt-key add
```

Append the Jenkins repository to your sources list:

```
→ echo deb http://pkg.jenkins.io/debian-stable binary/ |  
sudo tee /etc/apt/sources.list.d/ jenkins.list
```

```
→ sudo yum install jenkins -y
```

Jenkins status

```
→ sudo systemctl status Jenkins
```

Jenkins port number

```
→ http://13.48.70.178:8080
```

Start Jenkins

```
→ sudo systemctl start jenkins
```

Install java

```
→ sudo yum install java-17-openjdk -y
```

```
→ java -version
```

Added Jenkins repository

```
→ sudo yum install -y wget
```

```
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.key>

install maven

→ sudo yum install -y apache-maven

Verify version

→ mvn -version

You can Create one project like programme use vs code

Hello world java

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

Push the GitHub account then write pom.xml file

Open connect es2 Jenkins and login username password

Open mange Jenkins

Open credentials go to

System → open global credentials → scope → add

Username password

Id – GitHub Credentials

Description

Added user name& password of GitHub

Go to tools

Add maven ->install automatically

Finish setting

Then start new item

HELLO WORLD click freestyle project

Click ok

Description

This build a java project and to understand GitHub &maven integration with junkies.

Next step

Git url

<https://github.com/Bajisai/Puopale-project-.git>

Credentials

Saibajivellulli/***** (added user)

Change branch

Environment add

→Add timestamps to the Console Output

BUILD STEPS

Maven version

→maven

Goals

→clean package

Save

Then build Now

Then success to deployed

Maven project in Jenkins

Then install go to plugins and check maven integration then install.

Start new item maven project

Description

→this is test maven integration with Jenkins

Git url

→<https://github.com/Bajisai/Puopale-project-.git>

Credential

→saibajivellulli/***** (added user)

Change branch

Build

Pom.xml

Clean install

Save and build now run success

.....

- **Git/GitHub** – Version control system to manage the project source code.
- **Jenkins** – CI/CD automation server to build, test, and deploy the project.
- **Maven** – Build automation tool to compile, package, and manage dependencies.
- **Pipeline (Jenkinsfile)** – Scripted automation for CI/CD.
- **Web Server (Tomcat, Docker, or Kubernetes)** – For deployment.

Jenkins file stored in git git repo

```
pipeline {  
    agent any  
    tools {  
        maven 'Maven-3.8.6' // Ensure Maven is installed in  
Jenkins  
    }  
    stages {  
        stage('Clone Repository') {
```

```
    steps {
        git branch: 'main', url: 'https://github.com/your-
repo.git'
    }
}
stage('Build') {
    steps {
        sh 'mvn clean package'
    }
}
stage('Test') {
    steps {
        sh 'mvn test'
    }
}
stage('Deploy') {
    steps {
        echo 'Deploying the application...'
        // Add deployment scripts (Tomcat, Docker,
Kubernetes, etc.)
    }
}
```

}

}