

```
#####
DevOps Project Setup with CI CD Pipeline
#####
```

- 1) Maven
- 2) Git Hub
- 3) Tomcat
- 4) SonarQube
- 5) Nexus Repo
- 6) Jenkins

```
#####
JENKINS VM Setup
#####
```

```
1) Create Ubuntu VM using AWS EC2

2) Install Java & Jenkins using below commands

$ sudo apt-get update

$ sudo apt-get install default-jdk

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

$ sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ >
/etc/apt/sources.list.d/jenkins.list'

$ sudo apt-get update

$ sudo apt-get install jenkins

$ sudo systemctl status jenkins

# Copy jenkins admin pwd
$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword

# Open jenkins server in browser using VM public ip

URL : http://public-ip:8080/

-> Finish Jenkins setup
```

```
#####
Pipeline creation
#####
```

=> Create CI CD Pipeline with below stages

```
=====
1) Create github stage
=====
```

```
=====
2) Create Maven Build Stage (Add maven in global tools)
=====
```

```
stage ('Maven Clean Build'){
    def mavenHome = tool name: "Maven-3.8.6", type: "maven"
    def mavenCMD = "${mavenHome}/bin/mvn"
    sh "${mavenCMD} clean package"
```

```
}
```

```
=====
```

3) Create SonarQube stage

```
=====
```

-> Connect to sonar vm and run sonar server

```
$ cd /opt/sonarqube-7.8/bin/linux-x86-64
```

```
$ sh sonar.sh start
```

-> Manage Jenkins -> Plugins -> Available -> Sonar Qube Scanner Plugin -> Install it

-> Generate Sonar Token and Add it as Secret text in Global Credentials

-> Manage Jenkins -> Configure System -> Sonar Qube Servers -> Add Sonar Qube Server

```
- Name : Sonar-Server-7.8
```

```
- Server URL : http://52.66.247.11:9000/ (Give your sonar server
```

```
url here)
```

```
- Add Sonar Server Token
```

```
(Token we should add as secret text)
```

```
(Save it)
```

```
stage('SonarQube analysis') {
    withSonarQubeEnv('Sonar-Server-7.8') {
        def mavenHome = tool name: "Maven-3.8.6", type: "maven"
        def mavenCMD = "${mavenHome}/bin/mvn"
        sh "${mavenCMD} sonar:sonar"
    }
}
```

```
=====
```

4) Create Nexus Stage

```
=====
```

-> Run nexus VM and create nexus repository

-> Create Nexus Repository

-> Install Nexus Repository Plugin

-> Generate Nexus Pipeline Syntax

```
stage('upload war to nexus'){
    nexusArtifactUploader artifacts: [[artifactId: '01-maven-web-app', classifier: '', file:
'target/01-maven-web-app.war', type: 'war']], credentialsId: 'NEXUS_CREDENTIALS', groupId:
'in.ashokit', nexusUrl: '65.0.182.179:8081', nexusVersion: 'nexus3', protocol: 'http', repository:
'ashokit-snapshot', version: '1.0-SNAPSHOT'
}
```

```
=====
```

5) Create Deploy Stage

```
=====
```

-> Start Tomcat Server

-> Generate SSH Agent and configure stage

```
stage ('Deploy'){
    sshagent(['Tomcat-Server-Agent']) {
        sh 'scp -o StrictHostKeyChecking=no target/01-maven-web-app.war ec2-
user@13.235.75.254:/home/ec2-user/apache-tomcat-9.0.64/webapps'
    }
}
```

```
}

#####
Final Pipeline
#####

node {

    stage ('Git Clone'){
        git credentialsId: 'c87aff7e-f5f1-4756-978f-3379694978e6', url:
'https://github.com/ashokitschool/maven-web-app.git'
    }

    stage ('Maven Clean Build'){
        def mavenHome = tool name: "Maven-3.8.6", type: "maven"
        def mavenCMD = "${mavenHome}/bin/mvn"
        sh "${mavenCMD} clean package"
    }

    stage('SonarQube analysis') {
        withSonarQubeEnv('Sonar-Server-7.8') {
            def mavenHome = tool name: "Maven-3.8.6", type: "maven"
            def mavenCMD = "${mavenHome}/bin/mvn"
            sh "${mavenCMD} sonar:sonar"
        }
    }

    stage('upload war to nexus'){
        nexusArtifactUploader artifacts: [[artifactId: '01-maven-web-app', classifier: '', file:
'target/01-maven-web-app.war', type: 'war']], credentialsId: 'NEXUS_CREDENTIALS', groupId:
'in.ashokit', nexusUrl: '65.0.182.179:8081', nexusVersion: 'nexus3', protocol: 'http', repository:
'ashokit-snapshot', version: '1.0-SNAPSHOT'
    }

    stage ('Deploy'){
        sshagent(['Tomcat-Server-Agent']) {
            sh 'scp -o StrictHostKeyChecking=no target/01-maven-web-app.war ec2-
user@13.235.75.254:/home/ec2-user/apache-tomcat-9.0.64/webapps'
        }
    }

}
```