**How to create Jenkinsfile | How to configure Declarative Pipeline | create Jenkins Pipeline As code**

Please find steps below for configuring pipeline as a code - Jenkinsfile.  
  
***Pre-requistes:***  
  
1. Project setup in Bitbucket/GitHub/GitLab  
2. Jenkins and Tomcat (web container) set up.  
3. Maven installed in Jenkins  
4. Sonarqube setup and integrated with Jenkins  
5. Nexus configured and integrated with Jenkins  
6. Slack channel configured an integrated with Jenkins  
  
**Create Jenkinsfile (pipeline code) to your MyWebApp**  
  
**Step 1**  
  
Go to GitHub and choose the Repo where you setup MyWebApp in Lab exercise # 2  
  
**Step 2**  
Click on create new file.  
Graphical user interface, text, application, email

Description automatically generated  
  
**Step 3 - Enter Jenkinsfile as a file name**  
**Graphical user interface, text, application

Description automatically generated**  
 **Step 4**  
  
Copy and paste the below code and make sure what ever is highlighted in red color needs to be changed per your settings.  
  
That's it. Pipeline as a code - Jenkinsfile is setup in GitHub.  
  
pipeline {  
  agent any  
   
  tools {  
  maven 'Maven3'  
  }  
  stages {  
    stage ('Build') {  
      steps {  
      sh 'mvn clean install -f MyWebApp/pom.xml'  
      }  
    }  
    stage ('Code Quality') {  
      steps {  
        withSonarQubeEnv('SonarQube') {  
        sh 'mvn -f MyWebApp/pom.xml sonar:sonar'  
        }  
      }  
    }  
    stage ('JaCoCo') {  
      steps {  
      jacoco()  
      }  
    }  
    stage ('Nexus Upload') {  
      steps {  
      nexusArtifactUploader(  
      nexusVersion: 'nexus3',  
      protocol: 'http',  
      nexusUrl: 'nexus\_url:8081',  
      groupId: 'myGroupId',  
      version: '1.0-SNAPSHOT',  
      repository: 'maven-snapshots',  
      credentialsId: 'fc0f1694-3036-41fe-b3e3-4c5d96fcfd26',  
      artifacts: [  
      [artifactId: 'MyWebApp',  
      classifier: '',  
      file: 'MyWebApp/target/MyWebApp.war',  
      type: 'war']  
      ])  
      }  
    }  
    stage ('DEV Deploy') {  
      steps {  
      echo "deploying to DEV Env "  
      deploy adapters: [tomcat9(credentialsId: '268c42f6-f2f5-488f-b2aa-f2374d229b2e', path: '', url: 'http://your\_public\_dns:8080')], contextPath: null, war: '\*\*/\*.war'  
      }  
    }  
    stage ('Slack Notification') {  
      steps {  
        echo "deployed to DEV Env successfully"  
        slackSend(channel:'your slack channel\_name', message: "Job is successful, here is the info - Job '${env.JOB\_NAME} [${env.BUILD\_NUMBER}]' (${env.BUILD\_URL})")  
      }  
    }  
    stage ('DEV Approve') {  
      steps {  
      echo "Taking approval from DEV Manager for QA Deployment"  
        timeout(time: 7, unit: 'DAYS') {  
        input message: 'Do you want to deploy?', submitter: 'admin'  
        }  
      }  
    }  
     stage ('QA Deploy') {  
      steps {  
        echo "deploying to QA Env "  
        deploy adapters: [tomcat9(credentialsId: '268c42f6-f2f5-488f-b2aa-f2374d229b2e', path: '', url: 'http://your\_dns\_name:8080')], contextPath: null, war: '\*\*/\*.war'  
        }  
    }  
    stage ('QA Approve') {  
      steps {  
        echo "Taking approval from QA manager"  
        timeout(time: 7, unit: 'DAYS') {  
        input message: 'Do you want to proceed to PROD?', submitter: 'admin,manager\_userid'  
        }  
      }  
    }  
    stage ('Slack Notification for QA Deploy') {  
      steps {  
        echo "deployed to QA Env successfully"  
        slackSend(channel:'your slack channel\_name', message: "Job is successful, here is the info - Job '${env.JOB\_NAME} [${env.BUILD\_NUMBER}]' (${env.BUILD\_URL})")  
      }  
    }    
  }  
}  
  
**Step 5**  
That's it. Pipeline as a code - Jenkinsfile is setup in GitHub.  
  
Click on commit to save into GitHub.  
  
**Create Pipeline and Run pipeline from Jenkinsfile**  
  
1. Login to Jenkins  
2. Click on New item, give some name and choose Pipeline and say OK

[Graphical user interface, text, application, email

Description automatically generated](https://1.bp.blogspot.com/-d_lAouxW7vc/W4r8bqsoXNI/AAAAAAAAAV0/VdTKezeGtwc0C0OUcM3Kwe7a3VhzydxhgCLcBGAs/s1600/Screen%2BShot%2B2018-09-01%2Bat%2B3.53.30%2BPM.png)

3. Under build triggers, choose Poll SCM,  
Enter H/02 \* \* \* \*

[Graphical user interface, text, application, email

Description automatically generated](https://2.bp.blogspot.com/-1pdTF3NS8rM/W4r8-pxIasI/AAAAAAAAAV8/Mne2rbh3ErAwnQQWfOZb6HwWpLQnLg_pgCLcBGAs/s1600/Screen%2BShot%2B2018-09-01%2Bat%2B3.55.51%2BPM.png)

4. Under Pipeline section. click on choose pipeline script from SCM

[A picture containing application

Description automatically generated](https://4.bp.blogspot.com/-kQngHNYePYo/W4r9WJEvDTI/AAAAAAAAAWE/7V5pryiKbFAAZQDNHERPrQti-Pz0t5HyACLcBGAs/s1600/Screen%2BShot%2B2018-09-01%2Bat%2B3.57.12%2BPM.png)

5. Under SCM, choose Git

[](https://2.bp.blogspot.com/-tkOeVbgQgLg/W4r9no8tFqI/AAAAAAAAAWM/_hOuk-cOV3cn2VMQtrbI0H0iplI6DjqZACLcBGAs/s1600/Screen%2BShot%2B2018-09-01%2Bat%2B3.58.34%2BPM.png)

6. Enter HTTPS URL of repo and choose credentials - enter user name/password of GitHub.  
Script path as Jenkinsfile

[Graphical user interface, application

Description automatically generated](https://1.bp.blogspot.com/-kbzZiw_WLMY/W4r-ZSfqSZI/AAAAAAAAAWY/Os2VzfmgAaou7AEeP2FUhoqQkdWnn_5bwCLcBGAs/s1600/Screen%2BShot%2B2018-09-01%2Bat%2B4.00.44%2BPM.png)

7. Click on Apply and Save  
8. Click on Build now.  
  
You should see pipeline running and application is deployed to Tomcat.

[Graphical user interface, application, table

Description automatically generated](https://1.bp.blogspot.com/-8fS_QXclYjw/XmqvGn24haI/AAAAAAAABgI/yKWh-eyy-pU7TfR0Qcru-9E0AnYtrV5ywCLcBGAsYHQ/s1600/pipeline%2Bview.png)