**JENKINS**

* New Item – Name it – Freestyle Project
* **Source Code** – GIT – Add Repository URL
* Add Credentials if REPO is Private
* BUILD – Will Clone the REPO – Check Workspace
* **MAVEN** uses pom.xml file for everything.

**GLOBAL TOOL CONFIGURATION**

* Install GIT on server and get the location by “whereis GIT”
* Name – Default
* Path – specify the path from server
* Add **MAVEN** :

Name – Any name

Install Automatically

* Add **JDK**:

Name: Any name

Install by giving Oracle ID/Password

**Code Compile**

* New Items – Name the project and select Freestyle Project.
* Source Code Management – REPO URL
* BUILD – Invoke top level Maven targets

Version – Name from Global Tool Config

Goal – Compile

SAVE and Build Now

* Check the TARGET directory - .class file

**Code Review**

* New Items -Name the project and select Freestyle Project.
* Source Code Management – REPO URL
* BUILD – Invoke top level Maven targets

Version – Name from Global Tool Config

Goal – [-P metrics pmd:pmd]

SAVE and Build Now

* Check Target directory – pmd.xml
* To make this File readable – install PMD plugin from PLUGIN Manager.
* Go to – POST BUILD ACTIONS – Publish PMD analysis
* PMD result path : Get the pmd.xml file location (target/pmd.xml)
* Build now – Check PMD warnings

**QA\_UNIT\_TEST**

* New Items -Name the project and select Freestyle Project.
* Source Code Management – REPO URL
* BUILD – Invoke top level Maven targets

Version – Name from Global Tool Config

Goal - Test

SAVE and Build Now

* Check Target reports
* To make this File readable – install JUNIT plugin from Manage PLUGIN Manager.
* Go to – POST BUILD ACTIONS – Publish Junit result
* Test Report xml : target/surefire-reports/\*xml
* Build now – Check LATEST TEST RESULT

**QA\_Metrics\_check**

* New Items -Name the project and select Freestyle Project.
* Source Code Management – REPO URL
* BUILD – Invoke top level Maven targets

Version – Name from Global Tool Config

Goal – cobertura:cobertura -Dcobertura.report.format=xml

SAVE and Build Now

* Check Target reports – site/cobertura – coverage.xml
* To make this File readable – install cobertura plugin from Manage PLUGIN Manager.
* Go to – POST BUILD ACTIONS – Publish cobertura coverage report
* Test Report xml : \*\*/target/site/cobertura/coverage.xml
* Build now – Check Coverage report

**Package**

* New Items -Name the project and select Freestyle Project.
* Source Code Management – REPO URL
* BUILD – Invoke top level Maven targets

Version – Name from Global Tool Config

Goal – package

SAVE and Build Now

* Check Target reports – addressbook.war (war file)
* **Location of war file on Jenkins Server : /var/lib/jenkins/workspace/Package/target**
* Build section – Execute shell

cd /root

./jenkins.sh

**{Where cd/root is the location of the Jenkins file where below mentioned codes are stored}**

**Create a <file.sh> on the Jenkins server and add this:**

#!/bin/bash

ssh -i /home/ec2-user/masterkey.pem ec2-user@18.188.47.59 "mkdir -p /home/ec2-user/deployment"

scp -i /home/ec2-user/masterkey.pem /var/lib/jenkins/workspace/Package/target/addressbook.war ec2-user@18.188.47.59:/home/ec2-user/deployment

ssh -i /home/ec2-user/masterkey.pem ec2-user@18.188.47.59 "sudo /home/ec2-user/triggerfile.sh"

TRIGGERFILE 🡪 It’s the file on the Tomcat server where below mentioned codes are stored.

**Create a <file.sh> on the TOMCAT server and add this:**

#! bin/bash

cd /opt/apache-tomcat-9.0.36/bin

./shutdown.sh

tar -cvzf /home/ec2-user/logs.tar.gz /opt/apache-tomcat-9.0.36/logs

tar -cvzf /home/ec2-user/webapps.tar.gz /opt/apache-tomcat-9.0.36/webapps

rm -rf /opt/apache-tomcat-9.0.36/logs/\*

rm -rf /opt/apache-tomcat-9.0.36/webapps/\*

cp /home/ec2-user/deployment/addressbook.war /opt/apache-tomcat-9.0.36/webapps/

./startup.sh

Build periodically - \*/2\*\*\*\* - Automatically builds in every 2 minutes

Poll SCM - \*/2\*\*\*\* - Automatically checks for any new commit - Doesn't BUILD, just checks

Email settings - Configure System - Extended Email Notification - SMTP Server

Username/password

Go to Post build Actions under any project

Add the recipient list

Advanced – Triggers – Send to **RECIPIENTS**

**Pipeline (POM – Project Object Module)**

Jenkins read POM file with the help of Maven plugin

**Compile** – Compiles the Code pulled from GitHub. Compiles all the Java code and get it into TARGET directory. (CLASS FILE) – Extension changes from .JAVA to .CLASS

**Code Review** – Checks if Code meets the Organization’s standard – Extra Characters, Variables but not required. Uses PMD plugin to read PMD codes and gets the .XML file.

**Unit** **Test** – Checks test cases using JUnit Plugin – Test cases are written by the developers.

Checks if code is logical. Check according to the Test scripts and the source code.

**QA Metrics Check** – Coverage Analysis – How much percent of code is checked. Calculates the Percentage of code accessed by testing. Shows if complete test cases are checked or not and shows the percentage of code checked.

**QA Package check** – Gets the executable file here (war file).

**Built in environment variables**

Jenkins provides a set of environment variables. You can also define your own. Here is a list of built in environment variables:

* **BUILD\_NUMBER** - The current build number. For example "153"
* **BUILD\_ID** - The current build id. For example "2018-08-22\_23-59-59"
* **BUILD\_DISPLAY\_NAME** - The name of the current build. For example "#153".
* **JOB\_NAME** - Name of the project of this build. For example "foo"
* **BUILD\_TAG** - String of "jenkins-${JOB\_NAME}-${BUILD\_NUMBER}".
* **EXECUTOR\_NUMBER** - The unique number that identifies the current executor.
* **NODE\_NAME** - Name of the "slave" or "master". For example "linux".
* **NODE\_LABELS** - Whitespace-separated list of labels that the node is assigned.
* **WORKSPACE** - Absolute path of the build as a workspace.
* **JENKINS\_HOME** - Absolute path on the master node for Jenkins to store data.
* **JENKINS\_URL** - URL of Jenkins. For example *http://server:port/jenkins/*
* **BUILD\_URL** - Full URL of this build. For example *http://server:port/jenkins/job/foo/15/*
* **JOB\_URL** - Full URL of this job. For example *http://server:port/jenkins/job/foo/*