How to write a Jenkinsfile to build a Maven project and deploy to Apache Tomcat webserver

Prerequisites

* maven, Deploy to container plugins should be installed in Jenkins.
* Configure specific version of maven in Jenkins Global Tool Configuration

Jenkinsfile

pipeline {

agent any

tools {

maven 'maven-3.6.3'

}

stages {

stage('Clone') {

steps {

git 'https://github.com/xxxxxxxxxx/maven-web-app.git'

}

}

stage ('Build') {

steps {

sh 'mvn clean package'

}

}

stage ('Deploy') {

steps {

script {

deploy adapters: [tomcat9(credentialsId: 'tomcat\_credential', path: '', url: 'http://dayal-test.letspractice.tk:8081')], contextPath: '/pipeline', onFailure: false, war: 'webapp/target/\*.war'

}

}

}

}

}

In the tools block we have used maven definition to refer the maven installation **maven-3.6.3** configured in Jenkins Global tool configuration.

In the stages block we have created two stages Build and Deploy.

In the Build stage we are executing mvn clean package command to compile and package the java application.

It will compile the java code and generate the package in **targets** folder.

In the Deploy stage we are using the Deploy to container plugin to deploy the hello-world.war file to tomcat webserver.

Parameters passed to Deploy to container plugin definition.

* credentialsId: ‘tomcat\_credential’ –> Store the tomcat username and password in Jenkins credentials and pass the tomcat credential id here. I have stored the tomcat credentals in Jenkins and created the id as tomcat\_credential

Before storing the credentials in jenkins, create a user in Tomcat with manager-script role.

To create users in Tomcat, open the file **/var/lib/tomcat9/conf/tomcat-users.xml**

sudo vi /var/lib/tomcat9/conf/tomcat-users.xml

Go to end of the file and paste the following lines inside tomcat-users block and save it.

<role rolename="manager-script"/>

<user username="deployer" password="deployer" roles="manager-script"/>

Here we have defined one role manager-script and created one user deployer and assigned the manager-script role to the deployer user.

Then restart the tomcat9

sudo systemctl restart tomcat9

url: ‘http://152.70.71.239:8080/’ –> Your tomcat url

contextPath: ‘/pipeline’ –> Context path to deploy in Tomcat

onFailure: false –> Flag used to control the deployment, I dont want to deploy If my pipeline JOb fails, thatswhy I am setting onFailure flag to false

war: ‘target/\*.war’ –> Your war file name