**Java Web Application CI/CD Pipeline with deployment on Tomcat**

1. Tomcat Installation and Configuration

--------------------------------------------------

1. Install tomcat on a VM. In this example Tomcat 7.0 distribution is used. Java 8 or above must be installed.

Download Link: <https://archive.apache.org/dist/tomcat/tomcat-7/v7.0.32/bin/>

Download the [apache-tomcat-7.0.32.zip](https://archive.apache.org/dist/tomcat/tomcat-7/v7.0.32/bin/apache-tomcat-7.0.32.zip) file.

1. Unzip the tomcat file in C:\ drive\
2. Set JAVA\_HOME to the JDK directory. If not set already

C:\ProgramFIles\Java\JDK\_1.8\_X

1. Create an admin user for tomcat manager app.

IN CATALINA\_HOME/conf/tomcat-users.xml , add below code in <tomcat-users> tag

<tomcat-users>

<role rolename="admin-gui,manager-gui,manager-script,manager-jmx,manager-status,admin-gui"/>

<user username="admin" password="password" roles="admin-gui,manager-gui,manager-script"/>

</tomcat-users>

1. Tomcat works on port#8080. To change port , change it in below tag in CATALINA\_HOME/conf/server.xml file

<Connector port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

1. Start tomcat from command prompt. Enter the tomcat home and bin.

C:\apache-tomcat-7.0.42\bin>startup

[tomcat will start]

To shut down , C:\apache-tomcat-7.0.42\bin>shutdown

1. Test the tomcat and tomcat manager in browser.

URL: <http://localhost:8080> [tomcat home page will display][change localhost to public ip of host machine]

URL: <http://localhost:8080/manager/html> [manager app will open, enter username and password]

[tomcat home page will display][change localhost to public ip of host machine]

1. Creating A Java Web Application in Eclipse and testing.
2. Change Perspective to JavaEE
3. File->New->Dynamic Project
4. Create a JSP file index.jsp
5. Write in JSP file
6. Run on Server
7. Convert the project in maven project
8. Add dependency for Servlet API and Update . refer below pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>TestApp</groupId>

<artifactId>TestApp</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>war</packaging>

<dependencies>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

<scope>runtime</scope>

</dependency>

</dependencies>

<build>

<sourceDirectory>src</sourceDirectory>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.0</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>3.2.1</version>

<configuration>

<warSourceDirectory>WebContent</warSourceDirectory>

</configuration>

</plugin>

</plugins>

</build>

</project>

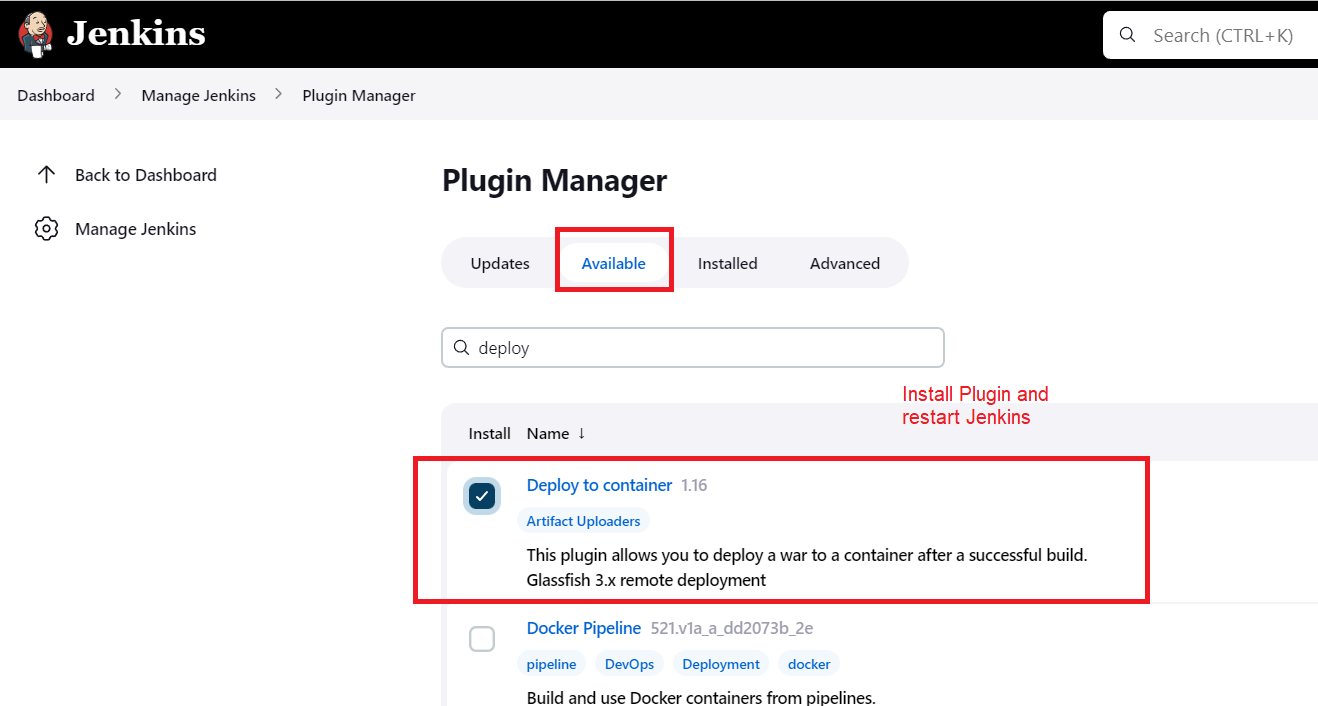
1. Run as mvn install
2. Push the code to GitHub.

<https://github.com/gauravsinghai/TestApp.git> [Project being used]

----------------------------------------------------------------------------------------------------------------------------------------

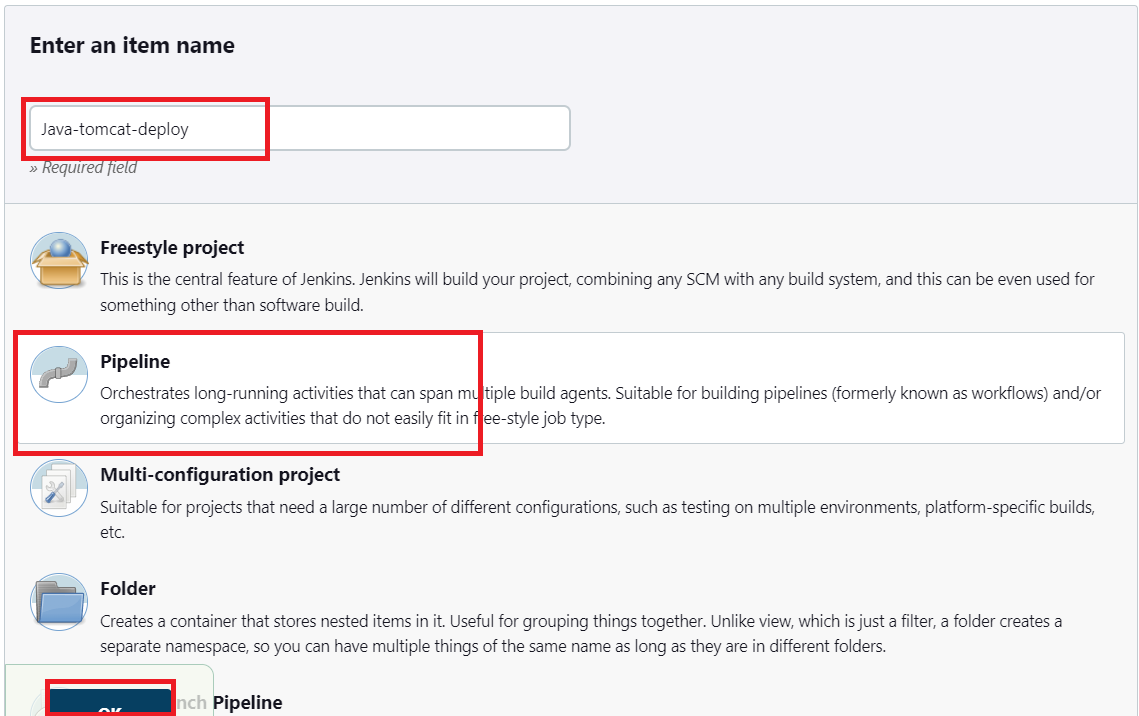
1. **Creating Jenkins Pipeline**
2. We will require Deploy to Container plugin. Install it.

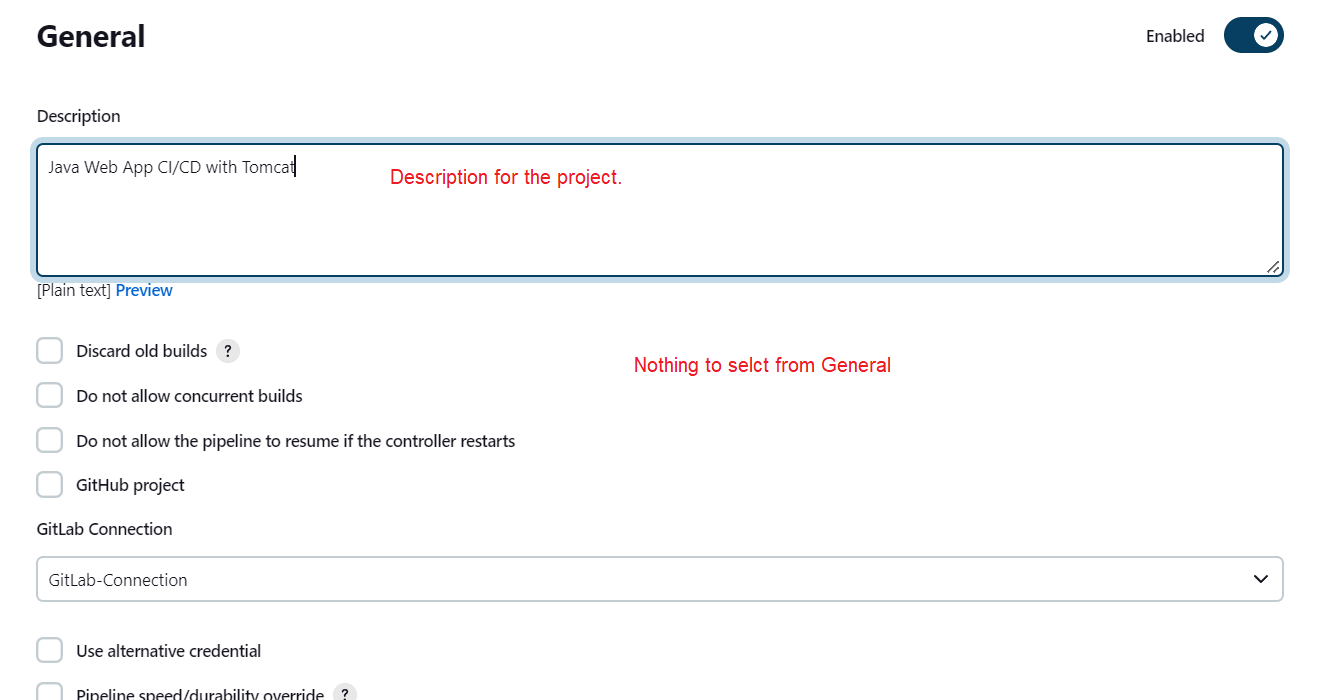
Manage Jenkins->Manage Plugins->in available Deploy to container->Install and restart Jenkins.

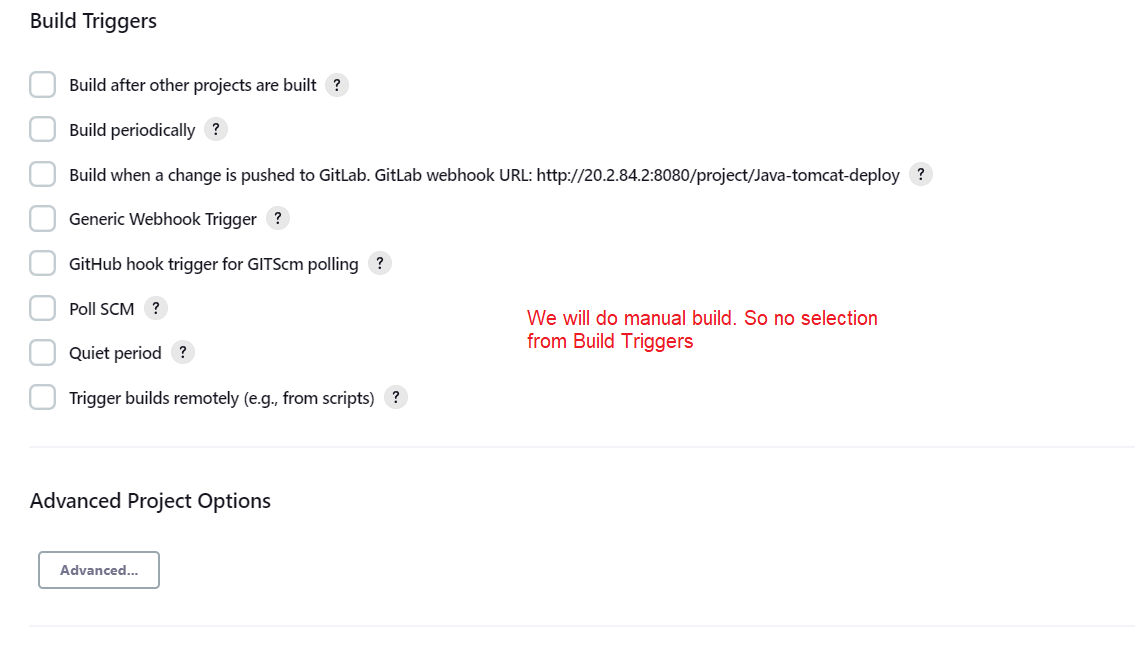


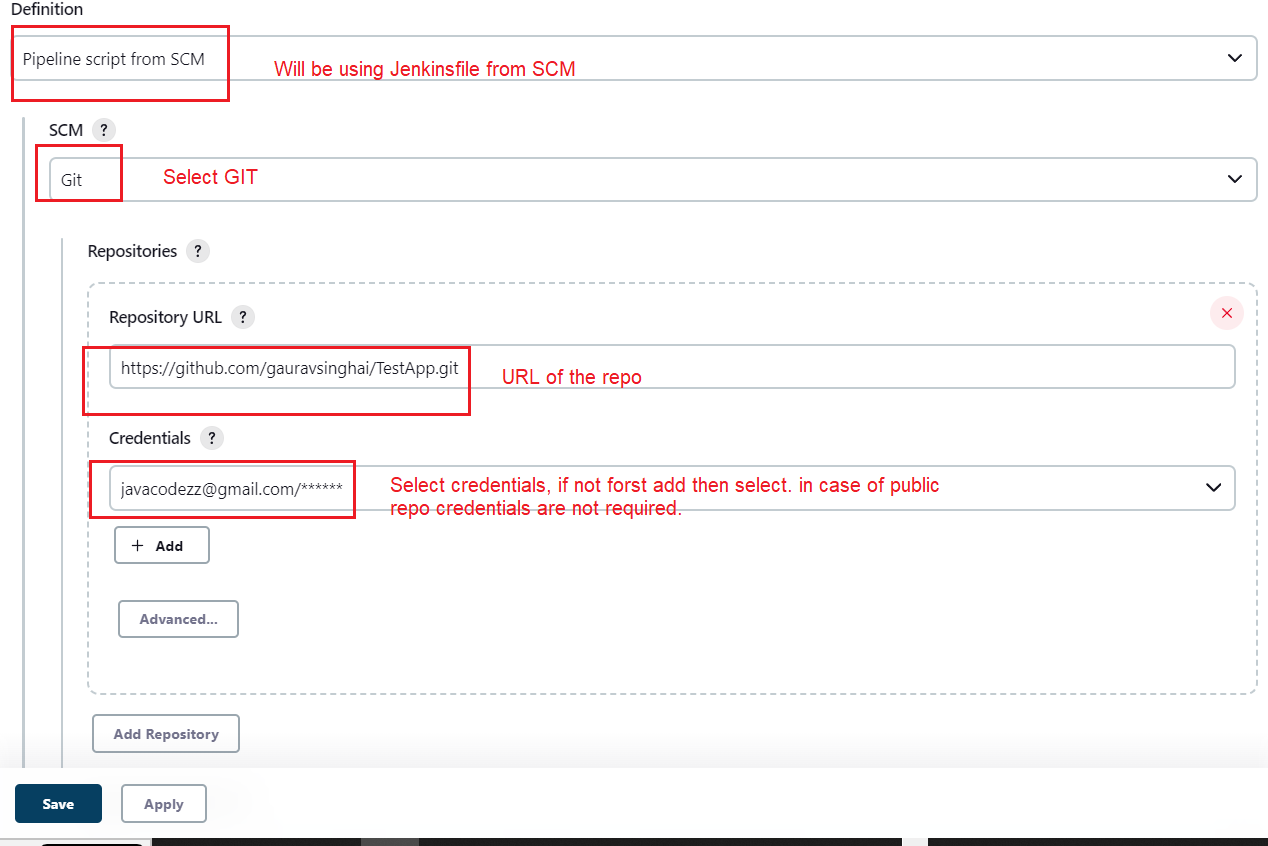
1. Create a Jenkins Pipeline

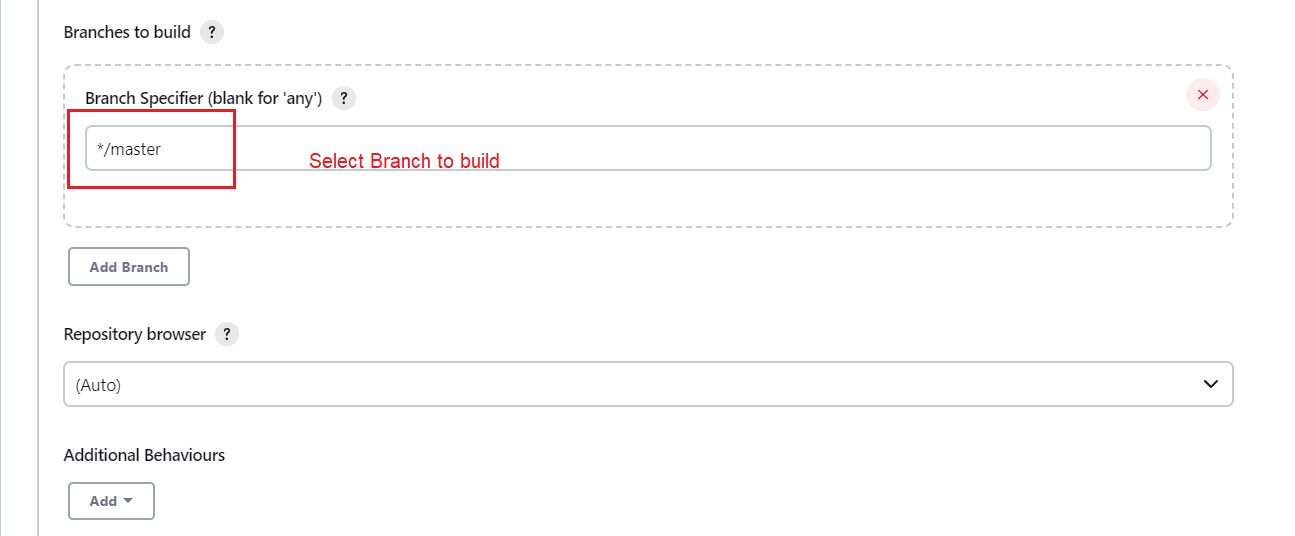
New Item->Pipeline











1. Write pipeline syntax.

Refer below syntax used in example.

pipeline{

agent any

tools{

maven 'mvn1' (Name of maven installation as configured in global tools)

}

stages{

stage ('Build'){

steps{

sh 'mvn clean package' (building project)

}

post{ (post build actions)

success{ (if build success)

echo "Archiving the Artifacts"

archiveArtifacts artifacts: '\*\*/target/\*.war' (war file location)

}

}

}

stage ('Deploy to tomcat server') {

steps{

deploy adapters: [tomcat7(credentialsId: 'tomcatCred', path: '', url: 'http://13.94.40.245:8089/')], contextPath: '/webapps', war: '\*\*/\*.war' (this should be generated from pipeline snippet generator, refer steps below.)

echo "Deployment.."

}

}

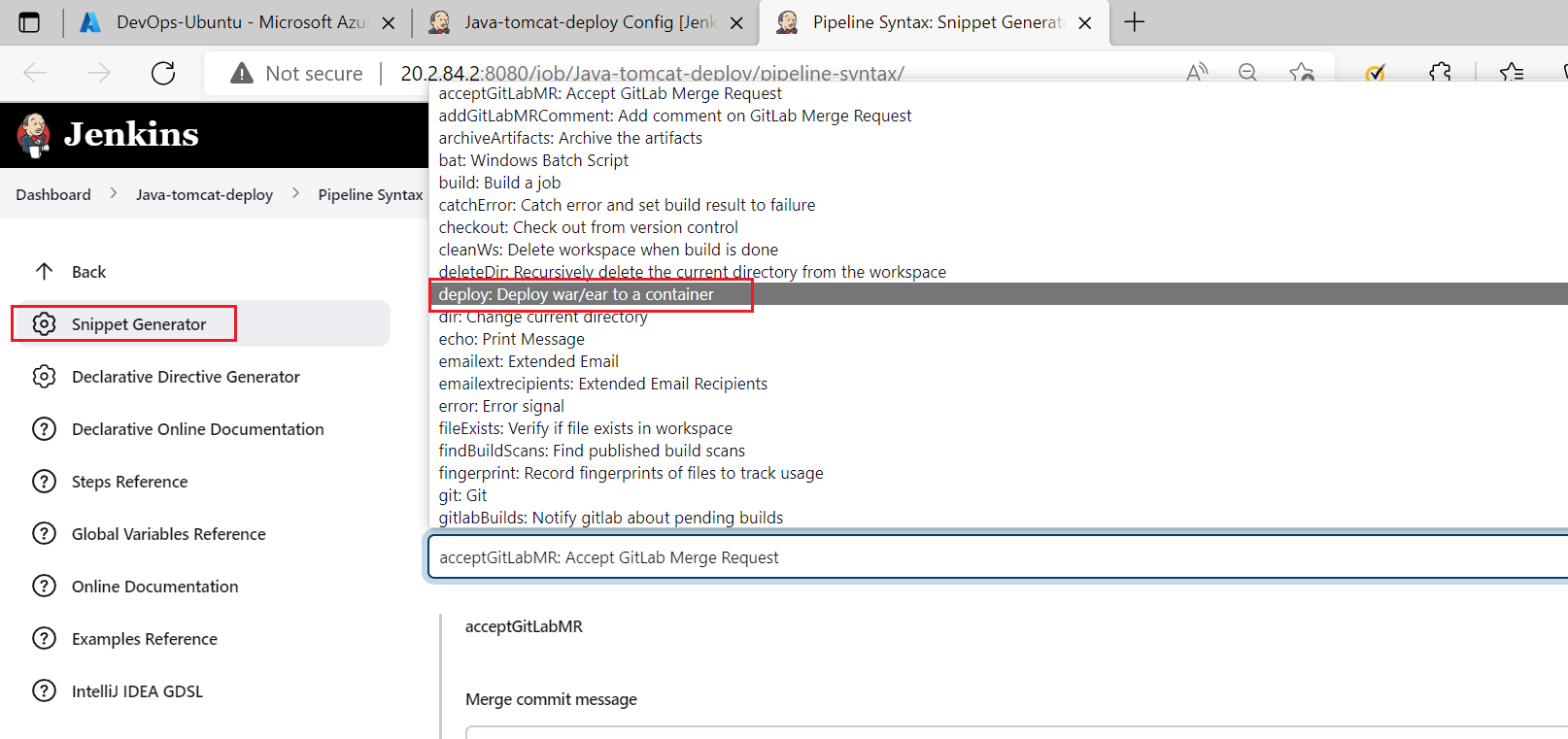
}

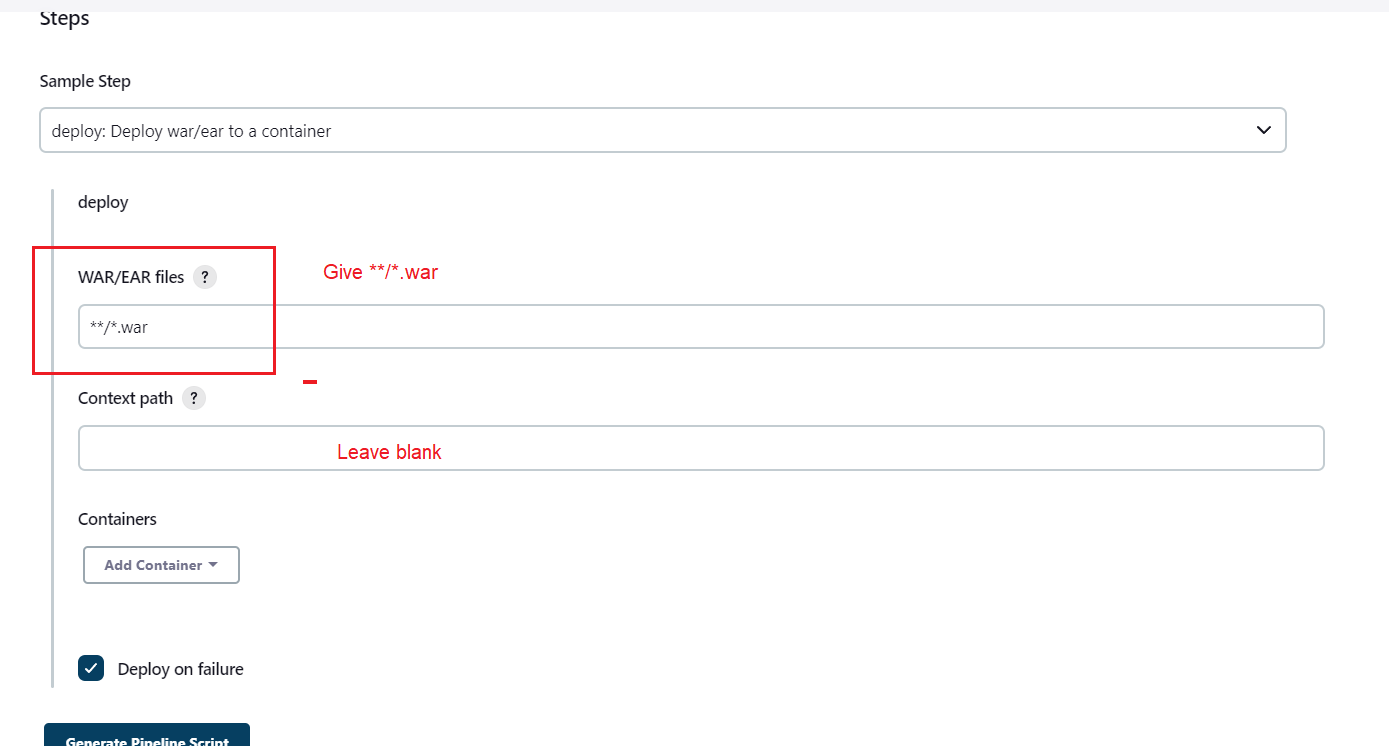
}

1. Steps to generate script to deploy to tomcat.
2. In Configuration, click on pipeline syntax.

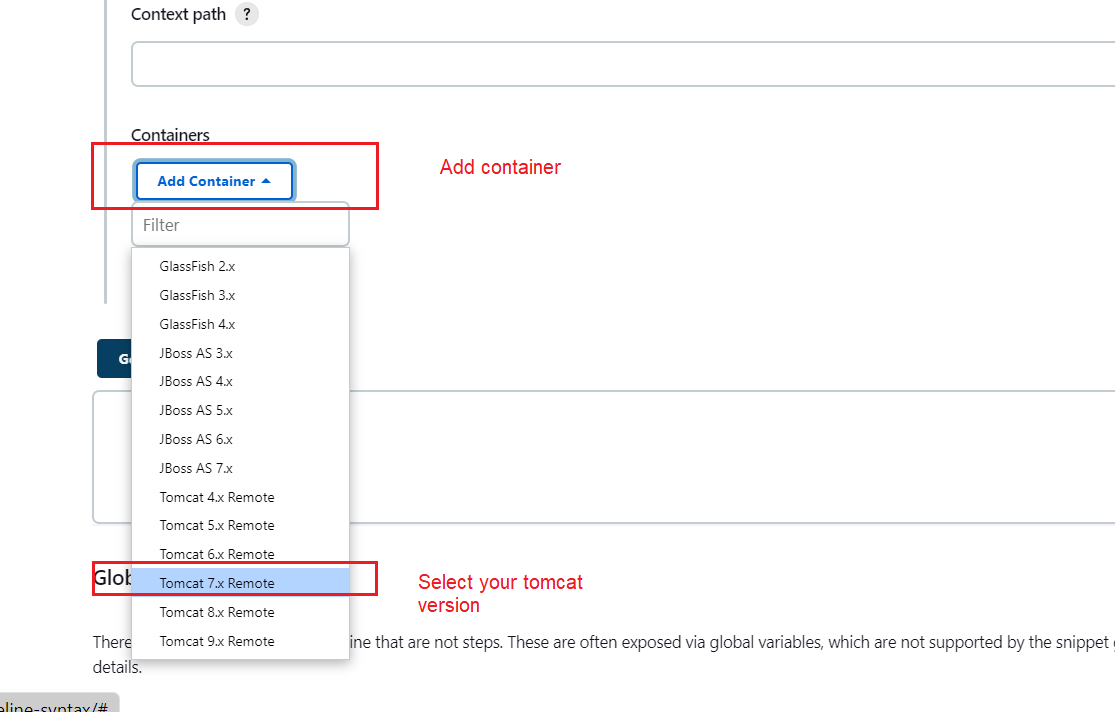


1. In Snippet Generator, select deploy:Deploy war/ear to a container. This option will come only when you have installed deploy to container plugin.

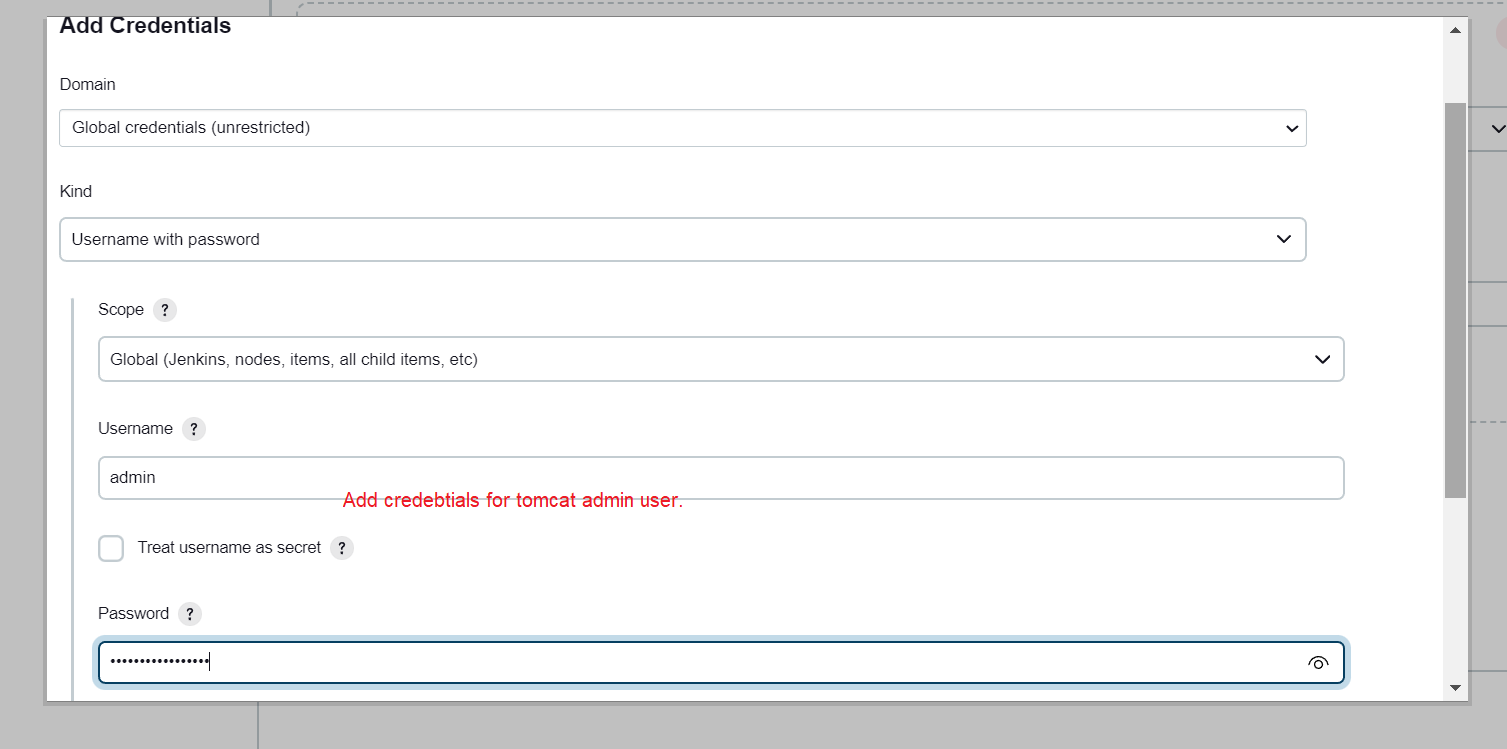


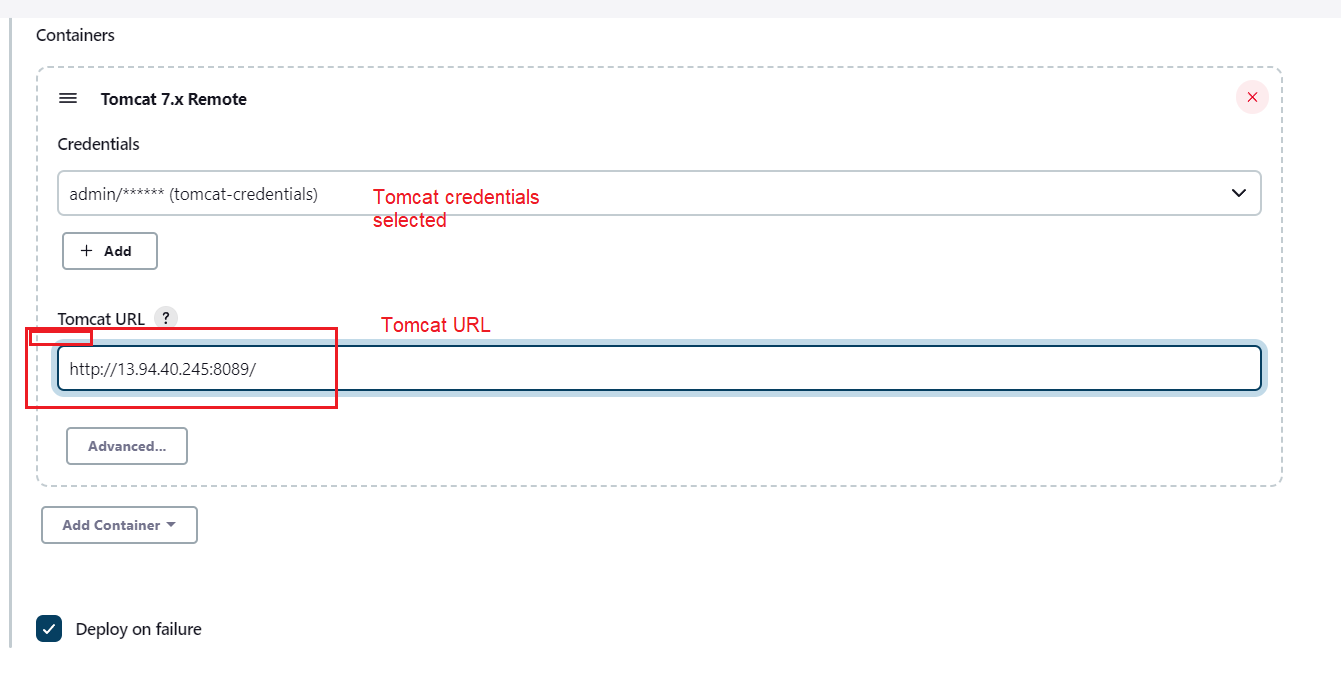


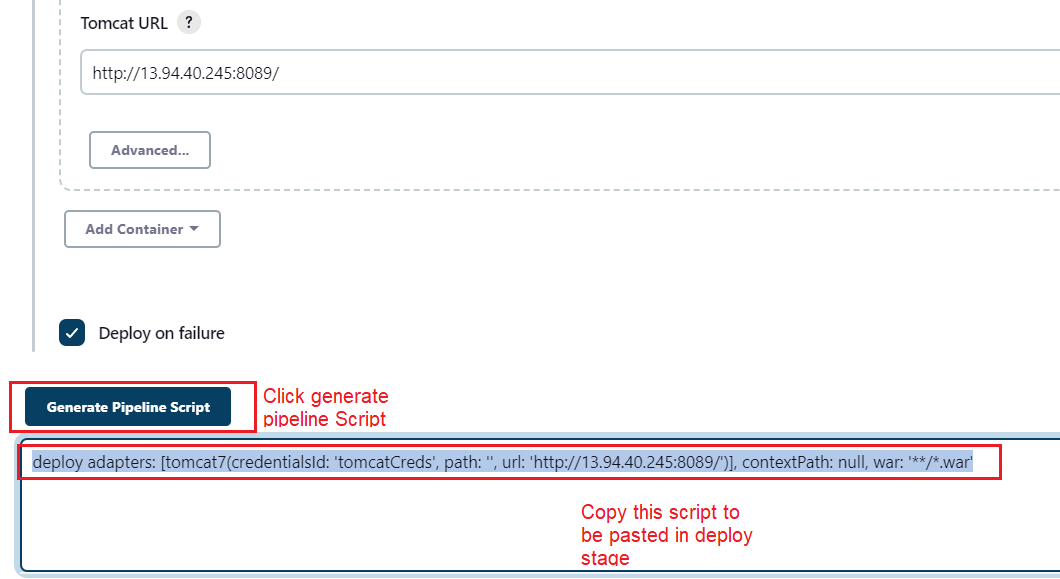
1. Select the Add Container and select the appropriate Tomcat version from dropdown.



1. Add credential for tomcat user.







1. Now create Jenkinsfile in project and also .gitignore file.
2. Build the project

