**Step-by-step Jenkins Tomcat deploy of a WAR file**

### Jenkins Tomcat deploy prerequisites

The following pieces of software are required to follow this example of a Jenkins deployment of a WAR file to Tomcat:

* a Java web application that Jenkins can deploy to Tomcat;
* a Git source code management tool installation;
* a Java 8 or newer installation of the JDK;
* a build tool – Apache Maven
* a Jenkins CI tool installation.

### A WAR file for Jenkins to deploy

If you need a web application for Jenkins to deploy to Tomcat, clone this rock-paper-scissors Java web app from GitHub:

<https://github.com/cameronmcnz/rock-paper-scissors.git>

git clone <https://github.com/cameronmcnz/rock-paper-scissors.git>  
cd rock\*  
git checkout patch-1

(here patch-1 is the branch from where we need to get web-app)

You may use this git url also:

<https://github.com/sebsto/webapp.git>

### Add to Tomcat a user with deployment rights

Change tomcat’s conf/server.xml with port 8080 to 9090

edit of the tomcat-users.xml file

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

Now start tomcat server:

Go to its folder and open it in gitbash prompt:

Tomcat’s bin> ./startup.bat

Tomcat starts in port: 9090 , url is <http://localhost:9090>

### Configure Jenkins

Now go to Jenkin’s folder where Jenkins.war file is present and open gitbash prompt

Java –jar Jenkins.war –hhtpPort=9092

Now Jenkins start : <http://localhost:9092>

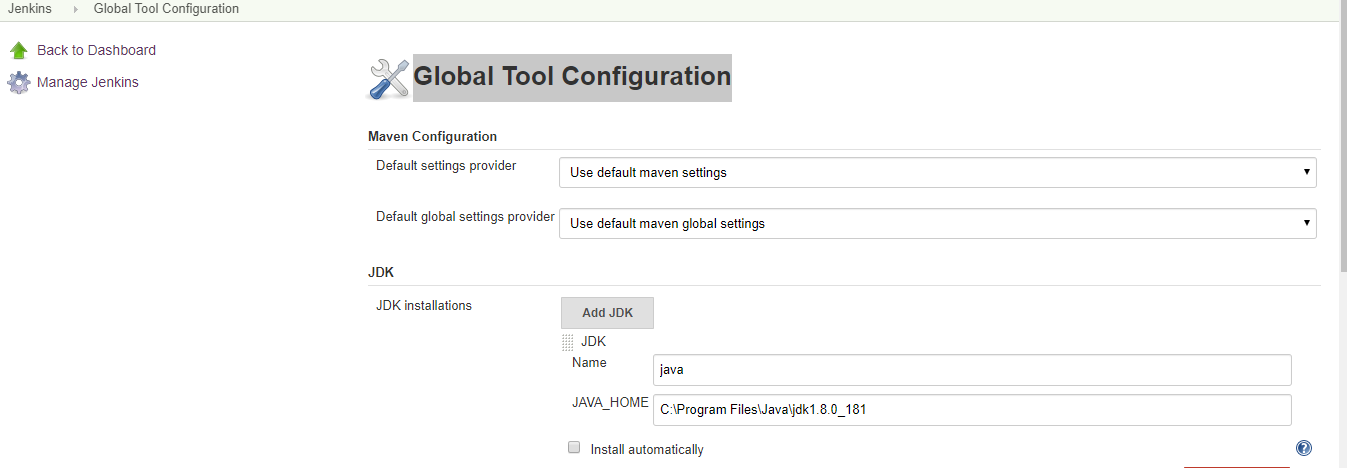
Asks for credientials for first time : get its password :

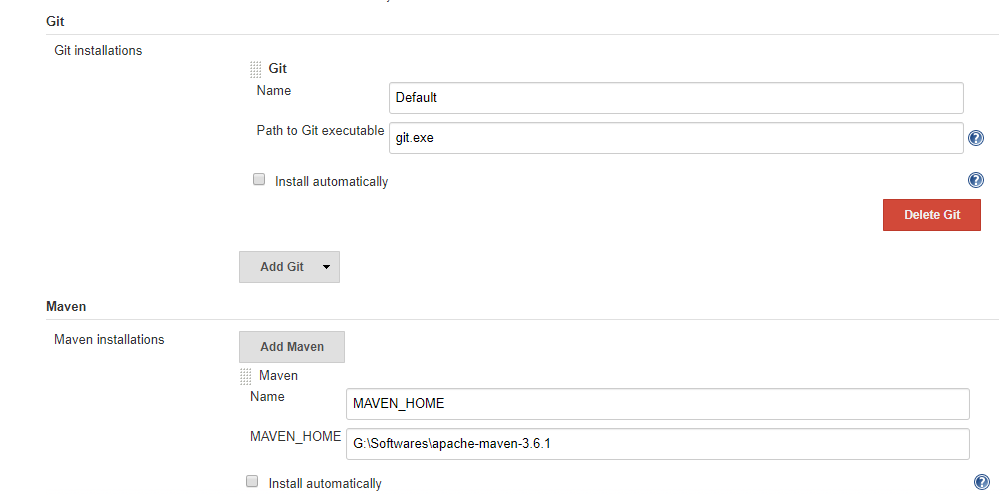
C:\Users\Dibakar\.jenkins\secrets folder’s under initialAdminPassword file, copy password from there and username is admin

# Global Tool Configuration

Set java , maven, git path also.

Need to install these tools also in local pc.

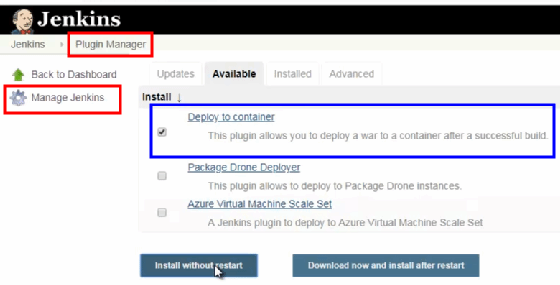




### Add Jenkins' plugin

Out of the box, there are no built-in features that perform a Jenkins WAR file deployment to Tomcat. That means a Jenkins Tomcat deploy plugin must be installed in the CI tool to make a deployment happen.

The most popular Jenkins Tomcat deployment plugin is named *Deploy to container*, which can be installed through the Plugin Manager tab under the "Manage Jenkins" section of the tool.



### Add maven plugin as well as git plugin also.

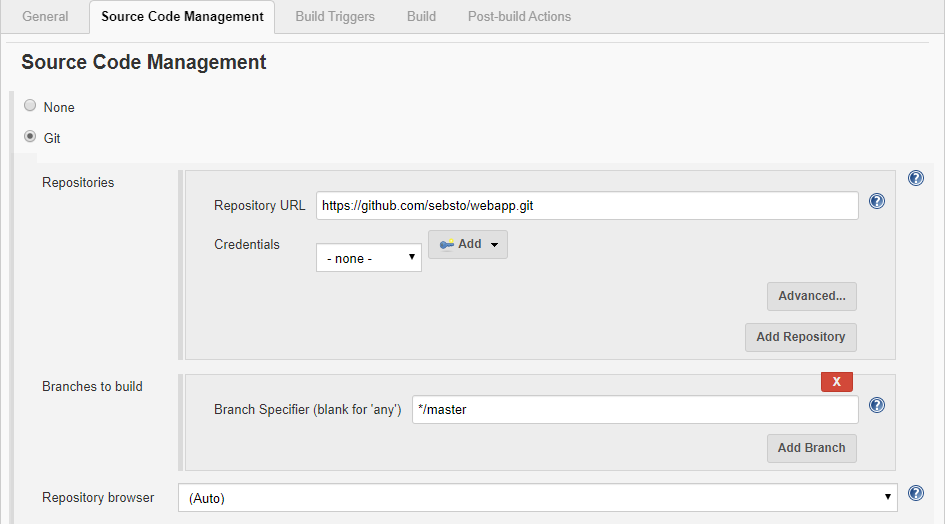
### Create a Jenkins freestyle project



### Configure standard build job properties

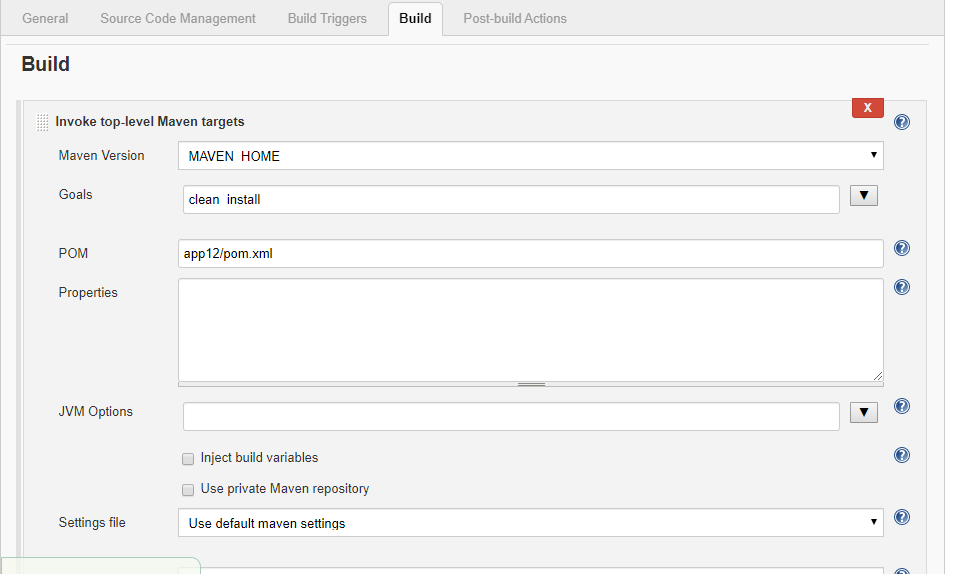
Git Repository URL: <https://github.com/sebsto/webapp.git>

Git branch specifier: \*/master



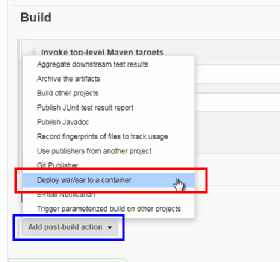
Maven Goals: clean install

Set Maven path: MAVEN\_HOME



### Jenkins Tomcat deploy plugin post-build action

After a build, the final step of a Jenkins pipeline deploy to Tomcat is to use the Deploy to container plugin in a post-build action.

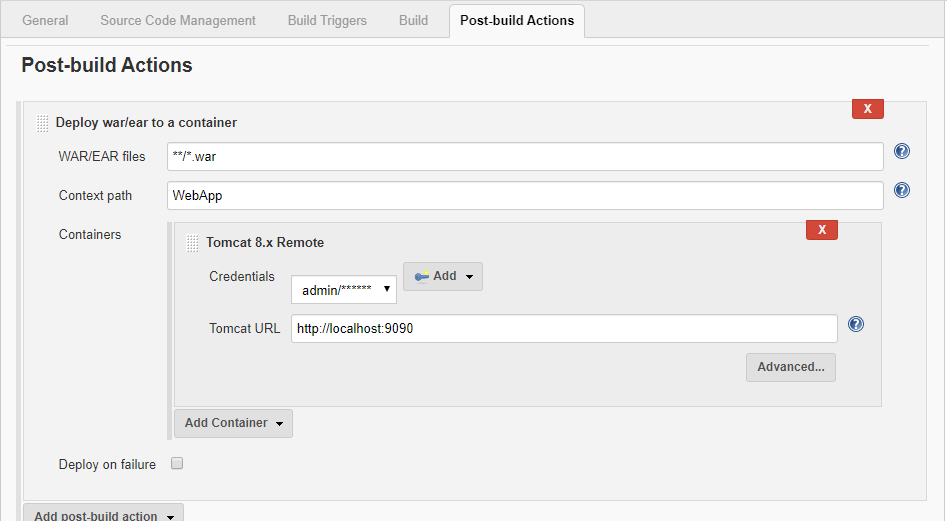


WAR/EAR files: \*\*/\*.war

Context path: WebApp

Containers: Tomcat 8.x credentials : admin/admin

Tomcat URL: http://localhost:9090



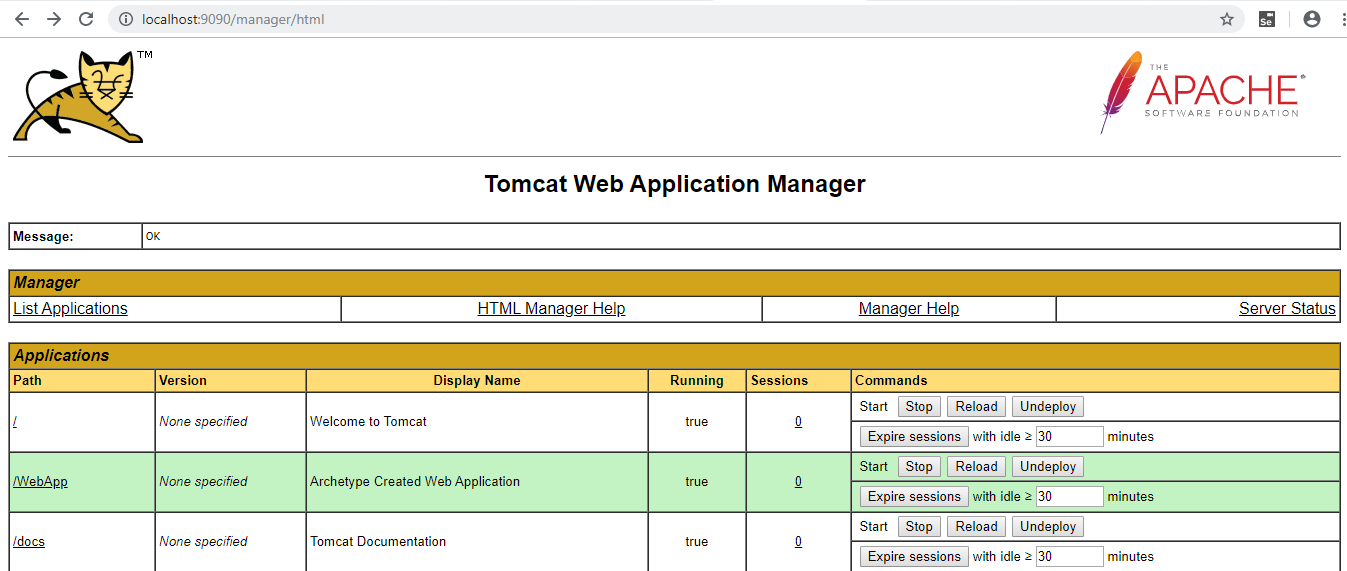
### Run the Jenkins build job

Now that you have specified all of the configurations, the Jenkins build job can be saved and run.

At the time of building , you can check console from Jenkins also .



When the build job finishes, the Jenkins Tomcat deploy of a WAR file will have also completed, and a file named .war will be visible in the webapps directory of Tomcat. Check Tomcat’s manager console as:



Reference :

<https://www.theserverside.com/video/Step-by-step-Jenkins-Tomcat-deploy-of-a-WAR-file>

## Poll SCM:

(\* \* \* \* \* ) represents (**minute**, **hour**, **day\_of\_month**, **week**, **day\_of\_week**)

range (0–59, 0–23, 1–31, 1–12, 0–7)

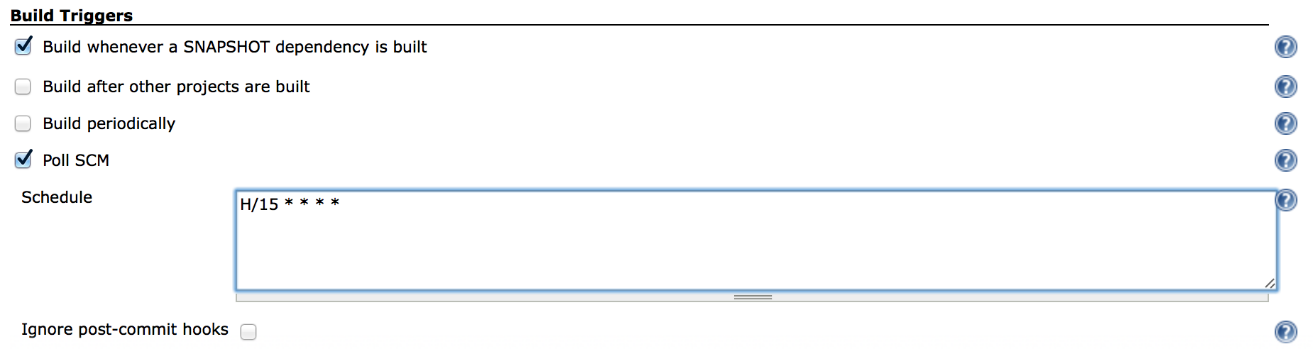
sunday -0, monday-1, tuesday-2 ,…,saturday-6

January-1, Feb-2, …,Dec-12

**Example :**

* H/15 \* \* \* \* — every 15 min
* 0 \* \* \* \* — every hour

## **How to configure**

1. Go to configure page of a project.
2. Go to “Build Triggers” section.
3. Check “Poll SCM”
4. Enter the schedule that you want in cron-style.
   * The help that placed side of input area might be useful
   * Example for poll at every 15minutes is here:
5. Save

## **See the log**

* When any change is detected at polling, we can see that message in log of Jenkins.
  + In this setup, the log file is located at /var/log/jenkins/jenkins.log