**Deploying your war file from Jenkins to Tomcat:** The steps to extend a Jenkins job to automatically deploy a built war file to a [Tomcat](http://tomcat.apache.org/) instance. (**set up a continuous deployment pipeline**).

Step 1: I assume that Tomcat up and running, you also have Jenkins up and running,

Have administrator privileges to set up a job in Jenkins that produces a war file.

Step 2. Install Jenkins plugin

Jenkins > Manage Jenkins> Manage Plugins> Available>[Deploy to container](https://wiki.jenkins-ci.org/display/JENKINS/Deploy+Plugin) plugin > install it.

Step 3. Edit tomcat-users.xml

In order for Tomcat to accept remote deployments,

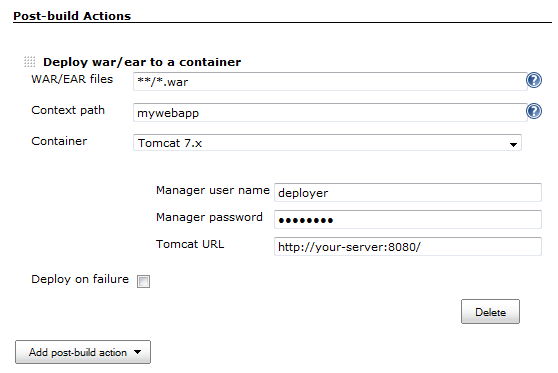
Edit tomcat-users,xml> add a user with the role manager-scipt.

(To do so, edit the file ../conf/tomcat-users.xml and add the following line):

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

Step 4. Edit the Jenkins job

Job > Configure >**Post-build Actions>** Add post-build action>**S**elect the option “**Deploy war/ear to a container**” from the Fill in the new fields, e.g.:

[](https://www.jdev.it/blog/wp-content/uploads/2013/12/jenkins-tomcat.png)

For WAR/EAR files you can use wild cards, e.g. \*\*/\*.war.

The ***context path*** is the ***context path part of the URL under which your application will be published***

***in Tomcat***.

Select the appropriate T**omcat version** from the Container dropdown box (note that you **can also** **deploy to Glassfish or JBoss** using **this Jenkins plugin**).

For the manager’s user name and password just copy over what you’ve entered in the tomcat-users.xml file.

The Tomcat URL is the base URL through which your Tomcat instance can be reached.  
Finally, don’t forget to save the configuration.

Step 5. Run the Job project and verify the end results

Schedule a build for your job in Jenkins. If you check out the log file you should see one or more lines near the end indicating that the war file has been deployed, e.g.:

Deploying /home/maven/hudson/hudson-work/jobs/MyWebApp/workspace/trunk/target/MyWebApp-1.0-SNAPSHOT.war to container Tomcat 7.x Remote

Redeploying [/home/maven/hudson/hudson-work/jobs/MyWebApp/workspace/trunk/target/MyWebApp-1.0-SNAPSHOT.war]

Undeploying [/home/maven/hudson/hudson-work/jobs/MyWebApp/workspace/trunk/target/MyWebApp-1.0-SNAPSHOT.war]

Deploying [/home/maven/hudson/hudson-work/jobs/MyWebApp/workspace/trunk/target/MyWebApp-1.0-SNAPSHOT.war]

Finished: SUCCESS

If you check the logfiles in Tomcat (catalina.out) you should also see that your application has been succesfully deployed.

Lastly, if you point your browser to the URL and context path you’ve specified in the job configuration in Jenkins (e.g., http://your-server:8080/mywebapp), you should be able to open your freshly deployed application.