

Setting Up Jenkins Job for Deployment to Tomcat Server

Jenkins – Maven Web App

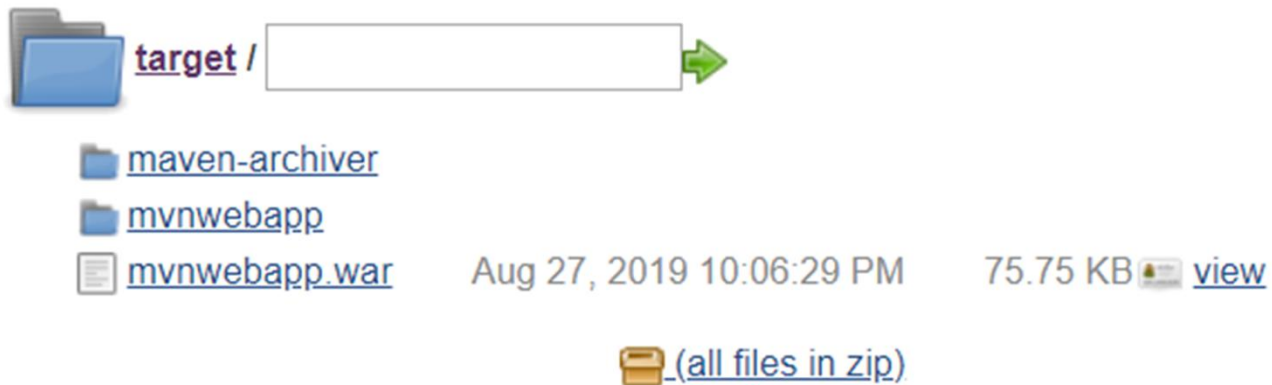
- Put up a Free Style Job to pull this Maven Web Application
Repo : <https://github.com/rchidana/mvnwebapp>
- First check by building it > mvn package



Jenkins – Maven Web App

- Check if a war file is properly generated

Workspace of mvn_webapp on master



Deploy to Container Plugin

- Install “Deploy to Container” plugin

This plugin allows you to store credentials in Jenkins.

Deploy to container Plugin



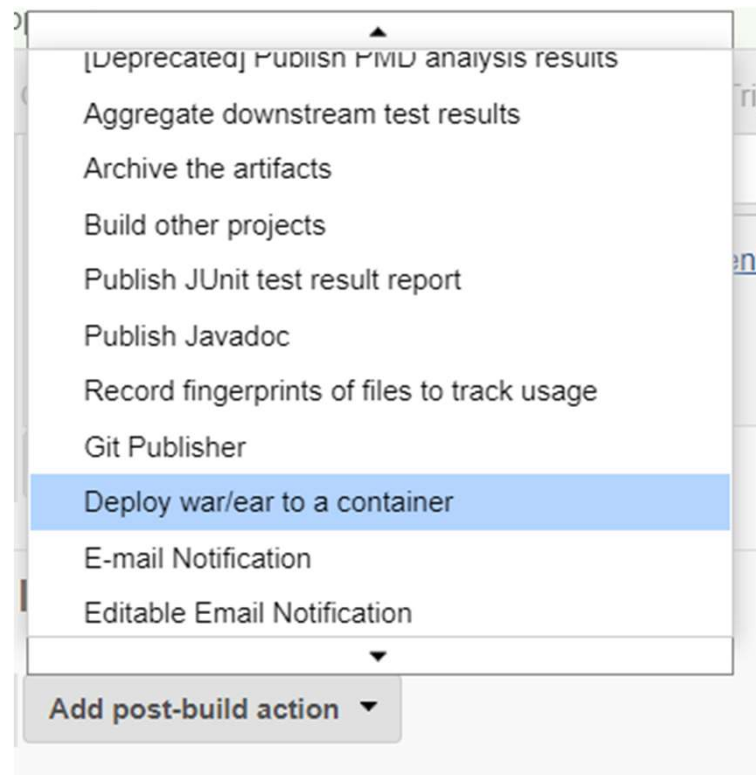
This plugin allows you to deploy a war to a container after a successful build.

Glassfish 3.x remote deployment

1.15

Jenkins – Post Build Step

- As a Post Build Step, put in a task to Deploy to Tomcat Container



Jenkins – Tomcat 7.X Remote

Deploy war/ear to a container

WAR/EAR files

Context path

Containers

Tomcat 7.x Remote

Credentials

Add

Tomcat URL

Advanced...

Add Container

Jenkins web-hook Integration

- Web-Hook is a call back mechanism from any Git Server that can trigger a Jenkins Job
- Any Jenkins Job can be configured to be triggered based on web-hook trigger from specific Git Server

Jenkins web-hook Integration

- Configure a Build Trigger for our job

Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts)
- ☐ Build after other projects are built
- ☐ Build periodically
- ☒ GitHub hook trigger for GITScm polling



If Jenkins will receive PUSH GitHub hook from repo defined in Git SCM section it will trigger Git SCM polling logic.
So polling logic in fact belongs to Git SCM.

(from [GitHub plugin](#))

GitHub web-hook set up

- Set up a web-hook trigger on desired repository to invoke call back to Jenkins Job based on desired user events
- URL for call back is :

`http://JENKINS-URL:PORT/github-webhook/`