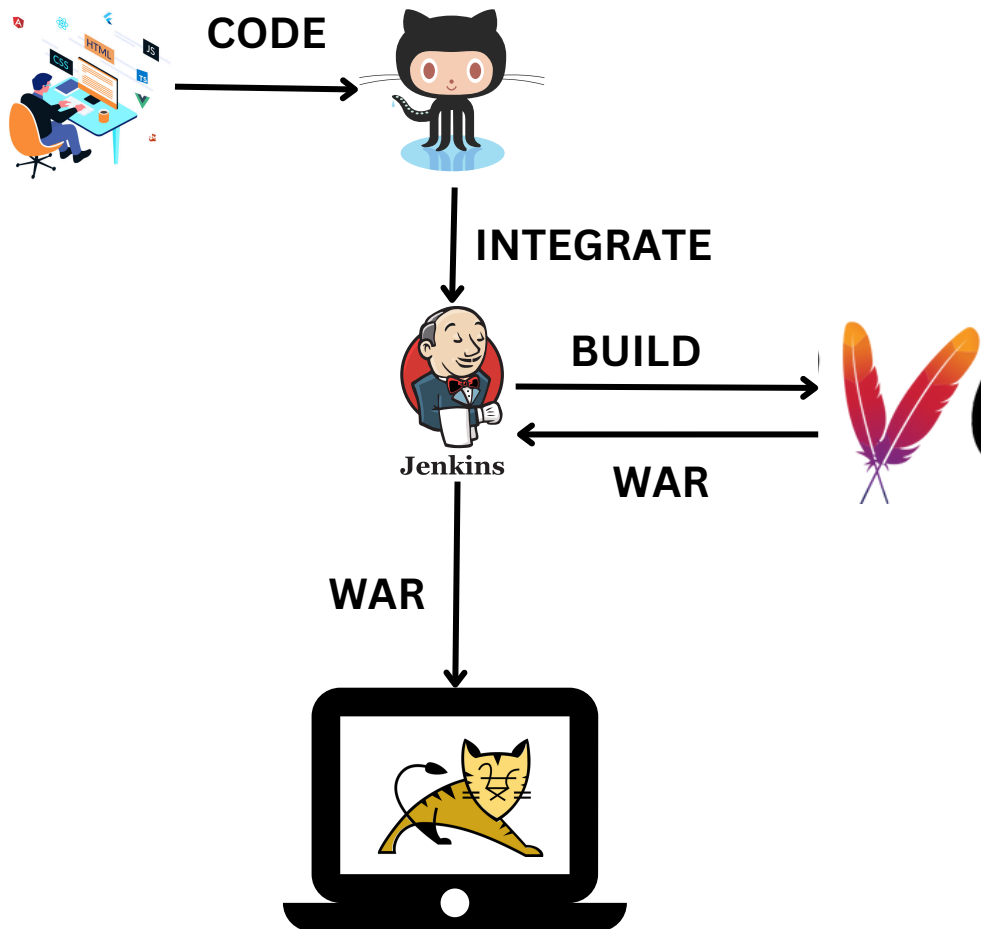


# DEPLOY A WEB-APP USING JENKINS IN TOMCAT SERVER

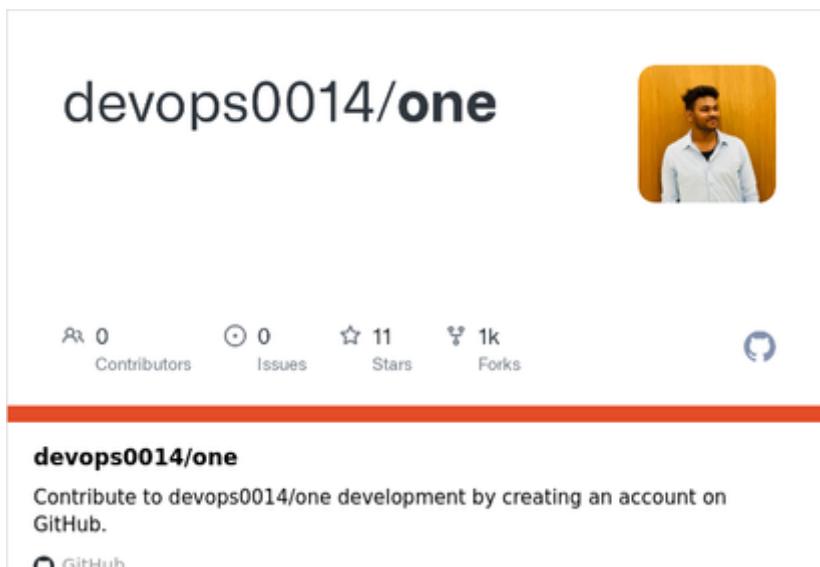


STEP-1: LAUNCH AN EC2 INSTANCE WITH 8080 PORT AND KEYPAIR

STEP-2: SETUP JENKINS

STEP-3: INSTALL GIT IN SERVER

STEP-4: CREATE A FREE STYLE JOB AND INTEGRATED GIT TO JENKINS



STEP-5: INSTALL JAVA1.8.0 & MAVEN

STEP-6: CONFIGURE THE JOB AND GO TO BUILD STEP AND SELECT INVOKE TOP-LEVEL MAVEN TARGETS (clean package) THEN BUILD NOW

## STEP-7: TOMCAT SETUP

- download tomcat file from dlcdn: [wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.96/bin/apache-tomcat-9.0.96.tar.gz](https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.96/bin/apache-tomcat-9.0.96.tar.gz)
- untar the file: `tar -zxvf apache-tomcat-9.0.96.tar.gz`
- go to the folder: `cd apache-tomcat-9.0.96/webapps/manager/META-INF`
- open the context.xml in vim editor and make some change (delete 2 lines (21 and 22 lines))
- go to three steps back: `cd ../../../../`
- and go to conf folder and open tomcat-users.xml file in vim editor

```
-->
<!--
<role rolename="tomcat"/>
<role rolename="role1"/>
<user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
<user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
<user username="role1" password="<must-be-changed>" roles="role1"/>
-->
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<user username="tomcat" password="123456" roles="manager-gui, manager-script"/>
</tomcat-users>
```

- go to one step back: `cd ..`
- go to bin folder and execute startup.sh file  
`./startup.sh`

**note:** tomcat will not run in 8080 port now. because jenkins is already running in 8080 port

- CHANGE TOMCAT PORT NUMBER:
- to stop the tomcat: `./shutdown.sh`
- go to conf folder and open server.xml file in vim editor
- change the port number in 69 line from 8080 to 8081
- and save & exit from the file
- go to one step back `cd ..`
- go to bin folder again and execute startup.sh file

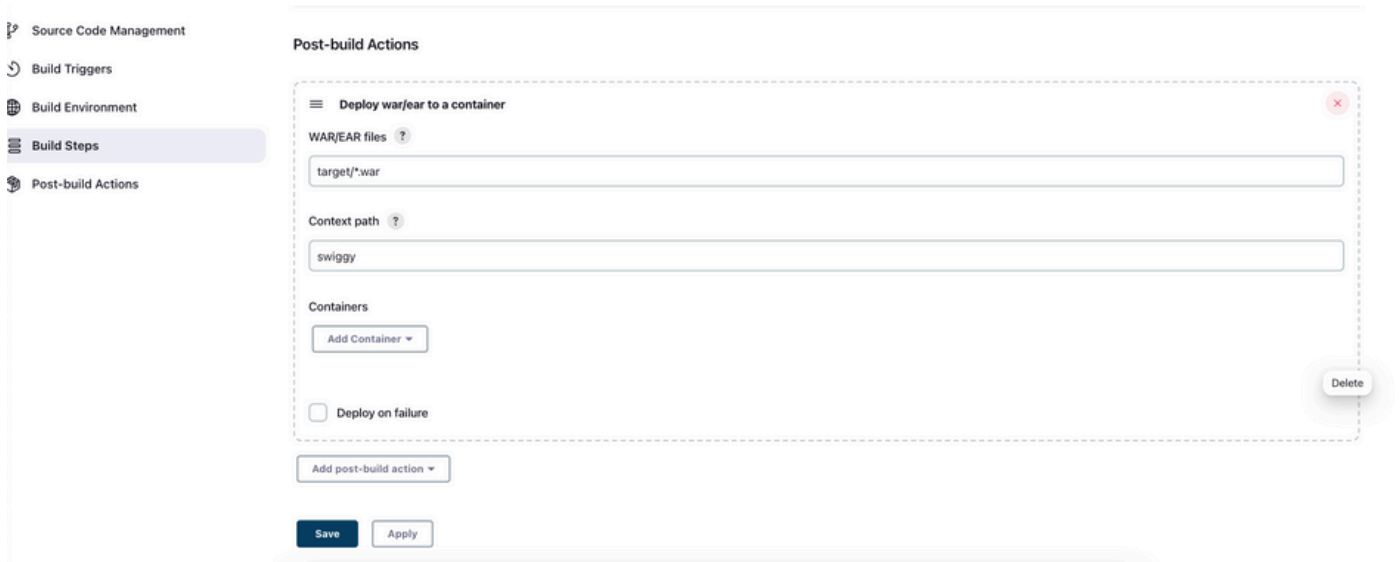
PUBLICIP:8081 (CONNECT IN THE BROWSER)

## STEP-8:

- go to jenkins dashboard
- install plugin (manage jenkins --> manage plugin --> available plugin --> **deploy to container**)
- after installing the plugin go to our job and select post build actions --> add post build actions

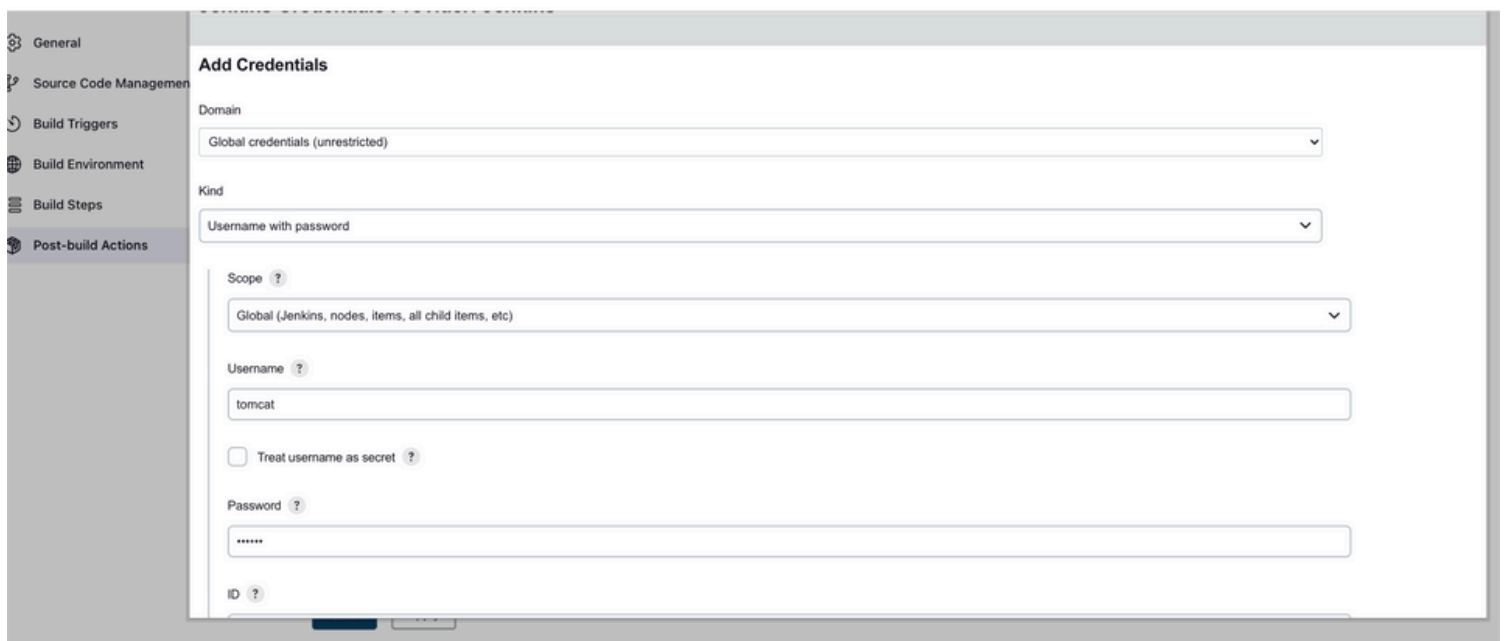
## STEP-9:

- select deploy war/ear to container



The screenshot shows the Jenkins 'Post-build Actions' configuration page. On the left sidebar, 'Post-build Actions' is selected. The main area is titled 'Post-build Actions' and contains a dashed box for the 'Deploy war/ear to a container' action. Inside this box, the 'WAR/EAR files' field is set to 'target/\*.war', the 'Context path' is 'swiggy', and there is an 'Add Container' button. Below the box, there is a checkbox for 'Deploy on failure' and an 'Add post-build action' button. At the bottom, there are 'Save' and 'Apply' buttons.

- click on add container(9th version). and add credentials (username & password of tomcat)



The screenshot shows the 'Add Credentials' dialog in Jenkins. The 'Domain' is set to 'Global credentials (unrestricted)'. The 'Kind' is 'Username with password'. The 'Scope' is 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Username' field is filled with 'tomcat'. There is a checkbox for 'Treat username as secret' which is unchecked. The 'Password' field is filled with '\*\*\*\*\*'. The 'ID' field is empty. At the bottom, there are 'Save' and 'Apply' buttons.

The screenshot shows the Jenkins configuration interface. On the left is a sidebar with navigation links: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions (which is highlighted). The main panel is titled 'Containers' and contains a configuration for 'Tomcat 9.x Remote'. This configuration includes a 'Credentials' dropdown menu with 'tomcat/\*\*\*\*\* (tomcat credentials)' selected, an '+ Add' button, a 'Tomcat URL' field with the value 'http://54.241.136.194:8081', an 'Advanced...' button, and an 'Add Container' button at the bottom.

## STEP-10:

- save and build the job and go to tomcat  
you will see swiggy folder. click on the folder you can access the client application