JENKINS PROJECT:

DEPLOY A WAR FILE IN TOMCAT SERVER

CLIENT REQUIREMENT: TO DEVLOPE A STATIC WEBSITE

PLAN:

1. CODE: HTML, CSS TO DEVELOPE THE APP
2. WE WILL USE AWS TO DEPLOY
3. TO AUTOMATE THE PROJECT WE NEED TO USE JENKINS
4. 2 DEVELOPERS
5. 24 HOURS
6. COST

CODE:

1. HTML & CSS IS USED TO DEVELOP A BASIC WEBSITE

BUILD:

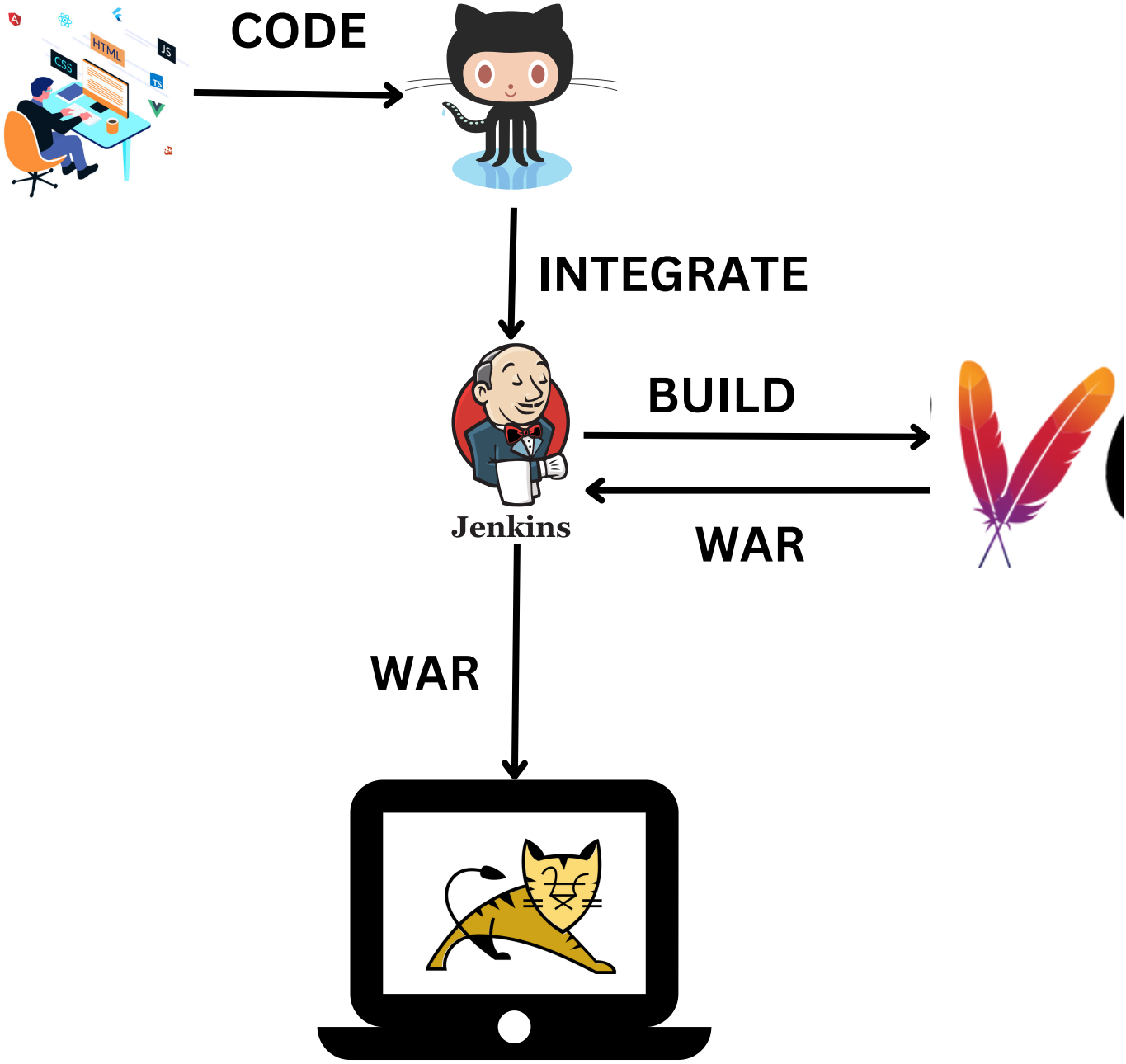
1. USE MAVEN TO BUILD THE SOURCE.
2. ONCE WE BUILD WE WILL GET A JAR/WAR
3. THESE FILES WE WILL USE TO TEST AND DEPLOY

TEST:

1. TESTING ENGINEERS WILL DO THE TESTING
2. SO WE WILL SKIP THIS PART.

DEPLOY:

1. TO DEPLOY THE WAR FILE WE NEED HAVE A CLIENT SERVER.
2. SETUP APACHE TOMCAT IN SERVER AND DEPLOY THE CODE IN SERVER.



REQUIREMENTS:

- LAUNCH 2 SERVER (jenkins, prod)
- GET THE CODE FROM THE DEVELOPERS
- SETUP JENKINS IN JENKINS SERVER
- SETUP TOMCAT IN PROD SERVER

PROCEDURE:

STEP-1: LAUNCH 2 INSTANCES WITH 8080 PORT

STEP-2: SETUP JENKINS IN SERVER

- sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
- sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
- amazon-linux-extras install java-openjdk11 -y
- yum install jenkins -y && systemctl restart jenkins

STEP-3: FORK THE GITHUB (https://github.com/devops0014/one.git)

STEP-4: INSTALL GIT IN OUR SERVER
yum install git -y

STEP-5: CREATE JOB AND INTEGRATE GIT TO JENKINS AND BUILD IT.
ONCE WE BUILD THE JOB, FILES PRESENT IN MASTER BRANCH WILL COMES INTO CI SERVER

STEP-6: NEXT STEP IS BUILD THE SOURCE CODE WHICH ARE PRESENT IN CI SERVER. TO BUILD THE WE NEED TO USE MAVEN

INSTALL JAVA-1.8.0 & MAVEN IN OUR SERVER

yum install java-1.8.0-openjdk -y
yum install maven -y

STEP-7: **CONFIGURE** THE SAME JOB AND CLICK ON **BUILD STEP** AND SELECT **ADD BUILD STEP** SELECT **invoke top level maven targer.**

in the goal : **clean package**

SAVE THE JOB AND BUILD.
SO WE WILL GET A WAR FILE IN TARGET FOLDER.

STEP-8: SETUP THE TOMCAT SERVER IN PROD SERVER.

DEVOPS BY MUSTAFA

- download tomcat file from dlcdn: wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.70/bin/apache-tomcat-9.0.70.tar.gz
- untar the file: tar -zxvf apache-tomcat-9.0.70.tar.gz
- go to the folder: cd apache-tomcat-9.0.70//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-user.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

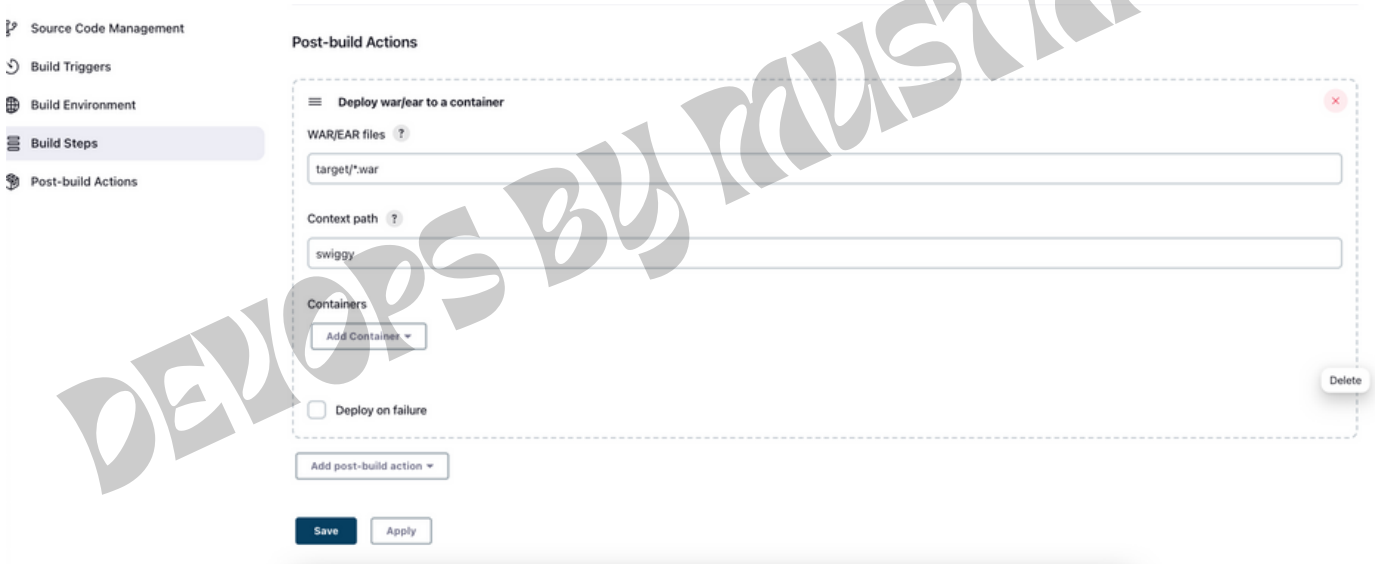
STEP-9:

GO to manager apps and it will ask the user name and password enter it

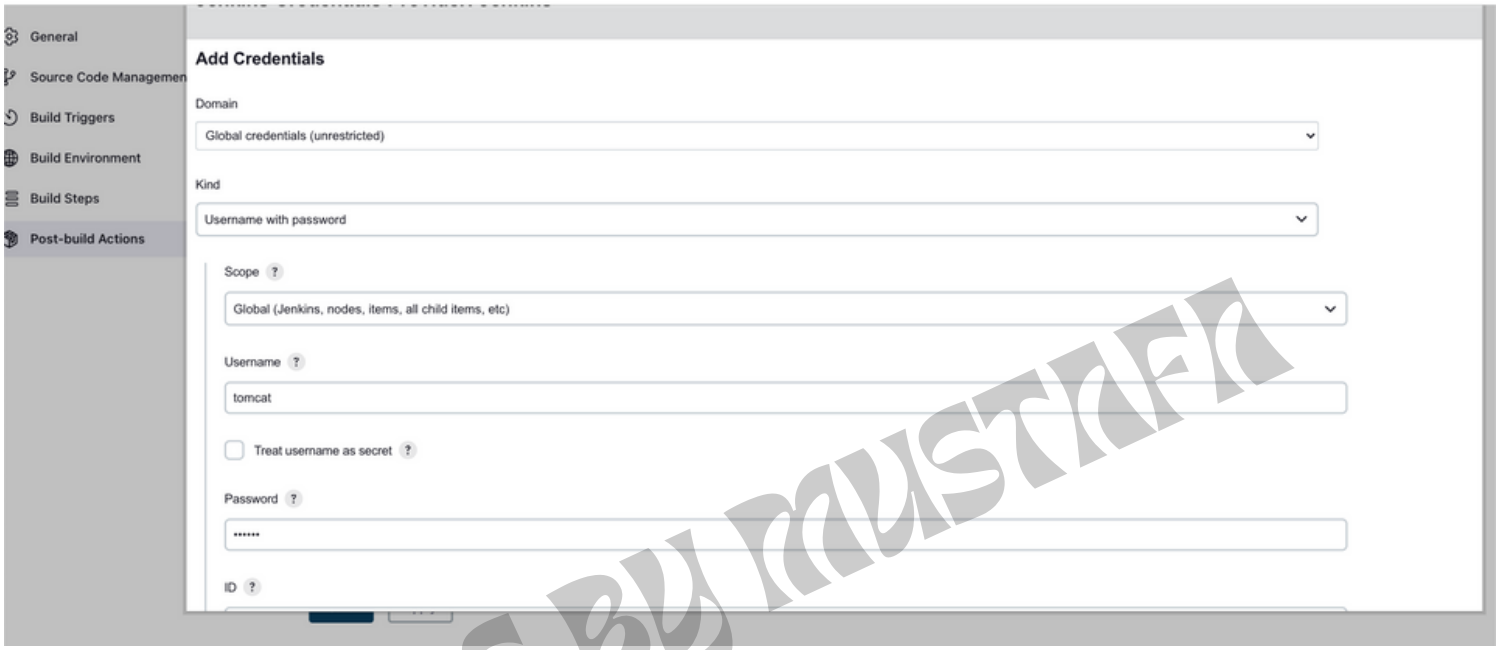
STEP-10: DEPLOY THE WAR FILE IN TOMCAT

- 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

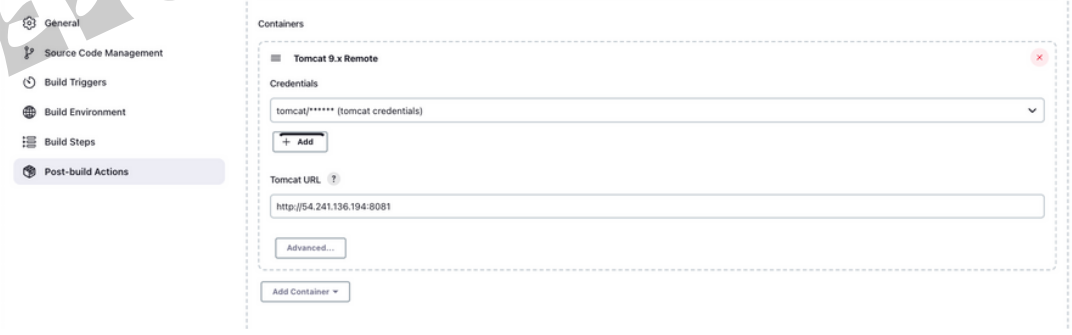
- select deploy war/ear to container



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



- add tomcat url



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

TASKS:

1. IMPLEMENT CI/CD IN ABOVE PROJECT
2. SETUP TOMCAT IN ANOTHER SERVER
3. PRACTICE BRANCH AND MERGE CONFLICTS IN GIT
4. UNDERSTAND THE PROJECT FOLDER STRUCTURE IN MAVEN
5. PRACTICE GIT WITH BITBUCKET
6. CREATE A PIPELINE WITH 4 JOBS

(JOB1 NEEDS TO CREATE FILE USING PARAMETER
JOB2 NEEDS TO CREATE FOLDER USING PARAMETER
JOB3 NEEDS TO GET THE SOURCE CODE FROM GITHUB
JOB4 NEEDS TO CREATE A FILE AND INSERT SOME DATA INTO FILE)