

SOURCE CODE

Build war file :

- Connect dev terminal through git
- check java is installed or not. Install it if not already installed.
- Check if maven is installed or not. If not, install it.
- Check if jenkins is installed or not. If not, install it.

Maven download :

- Go to the root directory : sudo su -
- cd /opt
- Open the browser and type maven download
- Right click on the apache-maven-3.9.9-bin.tar.gz and copy the link.
- enter command : wget <https://dlcdn.apache.org/maven/maven-3/3.9.9/binaries/apache-maven-3.9.9-bin.tar.gz>
- Unzip the maven file :


```
tar -xvzf apache-maven-3.9.9-bin.tar.gz
```
- ll
- Rename apache-maven-3.9.9 to maven :


```
mv apache-maven-3.9.9 maven
```
- ll
- cd maven
- ll
- Note down the maven path : /opt/maven
- Move to bin : cd bin
- ll
- Note down maven bin path : /opt/maven/bin

- pwd
- Now check if maven is installed or not :
mvn --version
(or)

- mvn --version
- Now go back to the root directory : cd ~

- Check maven version : mvn --version
It shows maven not found.
So we need to create environment variable.

- Go to root directory : cd ~

- pwd

- ll

we can see .profile file

- Open the file : sudo vi m.profile

- Go to insert mode (click on I) and give the maven, java home and m2 paths here.

- M2_HOME = /opt/maven

M2 = /opt/maven/bin

JAVA_HOME = /usr/lib/jvm/java-21-openjdk-amd64

PATH = \$PATH:\$HOME/bin:\$JAVA_HOME:\$M2_HOME:\$M2
:wg.

- To get java_home path : find / -name java-21*

- echo \$PATH

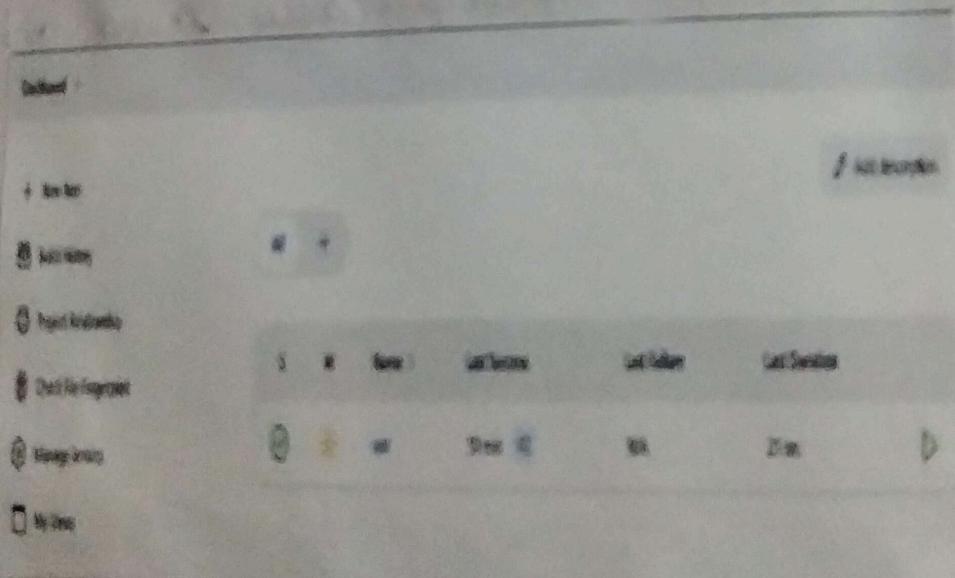
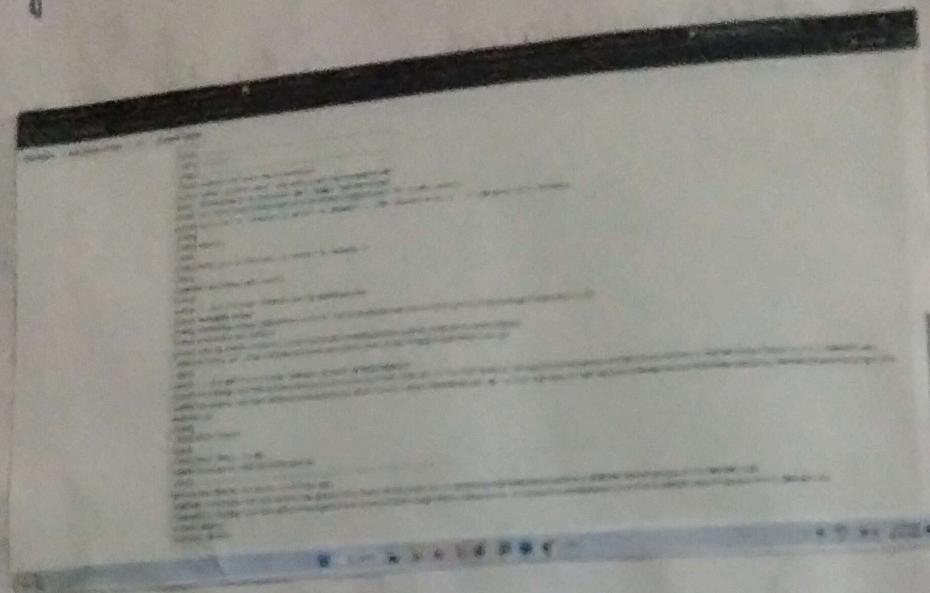
You can see the java and maven path above.
So we need to restart the .profile file with the below command

- source .profile

- Now we can see the java and maven path.
- Check maven version.
mvn -version (we can see maven now)
- Go to jenkins dashboard, we need to install one plugin (maven integration).
- Manage Jenkins > Plugins > Available Plugins > maven integration plugin.
- Without this plugin, we can't see the maven project.
- Once installed, click on restart jenkins.
- Add java and maven paths in jenkins.
- Go to jenkins dashboard > Manage Jenkins > Tools
Add JDK and add MAVEN
- Uncheck install automatically and add maven.
- Click on apply and save.
- Now we can create job for war file.
- Dashboard > New Item > war (give any name)
Select maven project, click on OK.
- Go to GitHub, select / copy repo url with pom.xml file and paste in jenkins repo url.
- Select build goal and options, write the command clean install and click on apply and save.
- Click on Build Now
- If job execution is taking a lot of time, then log out jenkins
stop the dev instance

OUTPUT

restart the dev instance
start the jenkins service
Run the job
Now your job will be executed



VIVA QUESTIONS

1. What is a WAR file, and where is it used?

Ans. A WAR file is a compressed archive used to deploy and package Java web applications on servers.

2. What steps would you take if the Jenkins server is not up?

Ans. Check Jenkins service status, review logs, verify open ports, ensure efficient resources and restart the service.

3. What is the difference between "Install without restart" and "Install and restart" in plugin installation?

Ans. "Install without restart" installs the plugin without restarting Jenkins. "Install and restart" installs the plugin and immediately restarts Jenkins to apply changes.

Experiment - 6:

Aim: Deploy the artifact on the Test Server.

Procedure:

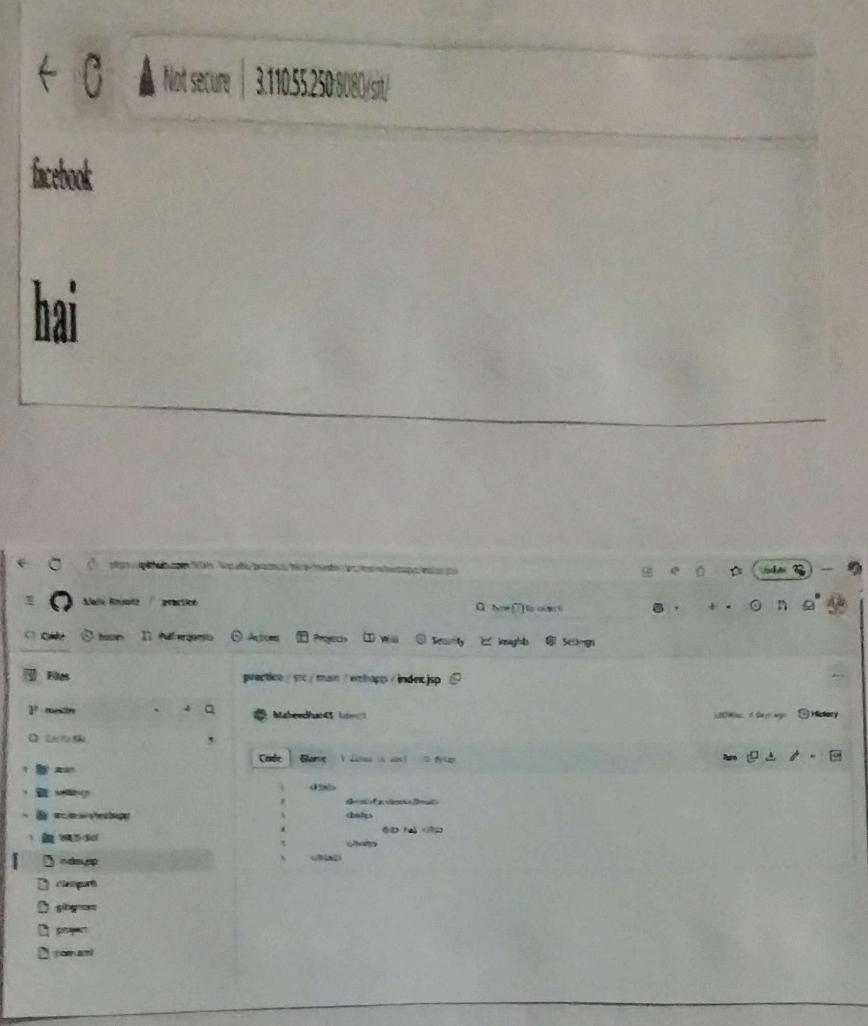
- Setup Jenkins on AWS
- Build WAR file
- Connect to sit machine with git bash
- Update the apt repository.
 > sudo apt-get update
- Install tomcat 9 : sudo apt-get install -y tomcat9
- Now install tomcat9 admin as well
 sudo apt-get install tomcat9-admin
- To access the tomcat, take the public IP of sit server and add :8080, copy paste this command on browser
- Now go to the path : cd /etc/tomcat9
- ll
- Now we need to add user in the tomcat-users.xml
 sudo vim tomcat-users.xml
 Go to insert mode (I)
 Add the following lines:
 <user username="ranya" password="ranya123"
 roles="manager-script,manager-status,manager-gui"/>
 esc :wq
- Now restart the tomcat service.
 sudo service tomcat9 restart

- Now you can add one plugin in the jenkins
(deploy to container)
- Manage jenkins > plugins > available plugins >
select deploy to container and install.
- Go to jenkins dashboard, select the war job >
select configure.
- Select the post build action and search deploy
war/ear to a container.
- Add container tomcat 9.
- Add the credentials
- Select the credentials and give the tomcat url
(sit server path along with 8080 port number)
- Click on Apply and Save.
- Run the build now.
- How can we check the artifact is deployed or not?
- It can be done as follows :

Take the sit public IP address and port no.
give the content path name.

ex : <http://3.110.55.250:8080/sit>

- The output will be displayed.



whenever we make some changes in the source file, jenkins job will automatically take the code from github to dev server and build the war file and deploy into it

Git commands involved in the automation process are:

- git clone <https://github.com/RaniyaReddyM/DevOps2.git>
this copies the remote repository to your local machine

• cd practice

• ll

• cd src

• ll

• cd main

• ll

• cd webapp

• ll

• vim index.jsp

• git status

shows modified/untracked files in working directory

• git add .

Stages all changed files for commit

• git status

• git commit -m "updated"

Creates a snapshot of changes with message "updated"

• git push origin master

uploads local commits to the remote repository's master branch

- Go to Jenkins > profile right click on project > configure > triggers > give " * * * * * " in schedule > Apply > save.
- Now we don't have to build again.
- Jenkins will automatically trigger the job.

OUTPUT

Builds

Filter

Today

- #5 3:21PM
- #4 3:18PM
- #3 2:59PM
- #2 2:36PM
- #1 2:19PM



Not secure | 3.110.55.250:8080/sit/

facebook

hello DHONI

VIVA QUESTIONS

1. How does Jenkins support Continuous Integration and Continuous Deployment (CI/CD)?

Ans. By automating code integration, testing, deployment through pipelines triggered by code changes.

2. Which Java version is recommended for Jenkins installation?

Ans. Java 11.

3. What are the key plugins required for basic automation in Jenkins, and how do you install them?

Ans. Key plugins : Git, Pipeline, Docker and Ansible. Installation Manage Jenkins > Manage Plugins > Available > search and install plugin.

4. What are the different build triggers available in Jenkins?

Ans. Poll SCM, Build periodically, Github Hook trigger, Trigger via URL.

5. How do you configure email notifications for build results in Jenkins?

Ans. Install Email Extension Plugin, go to manage Jenkins' Configure System, set SMTP details, then add post build actions to send emails.