Experiment # 3
Aim: To deploy Java Web application using Jenkins.

To deploy Java Web App we need to configure some of the additional software such as Maven and Apache Tomcat. We will do automated development from online repository for that purpose we will use Git.

Step 1: Download maven from official website https://maven.apache.org/ . Go to download section and download the binary zip archive such as apache-maven-3.8.4-bin.zip

Step 2: extract the folder at particular drive and set the environment variable. To set the environment variable go to extracted folder-> Apache maven -> bin
Copy the path and go on system environment variable setting. Click on environmental variable, select path and click on edit. Add new path and paste the maven path.

Step 3: To check maven is configured or not on system, go on command prompt (open command prompt as administrator) and type mvn -v , If you find the maven Version then maven system is installed and configured successfully.

Step 4: Now, Download Apache Tomcat from official website. Visit https://tomcat.apache.org/ . On the left side panel you can see various tomcat version, will refer Tomcat 9.0 click on it.
Scroll down the page till binary distribution section. 32-bit/64-bit Windows Service Installer click on this installer. The .exe file will download.

Step 5: Once it is download, install the Apache Tomcat. While installation it will ask you for username & password and also the port number. Remember the port number.
Username: test
Password: 12345
Click on next and finish the procedure.

Step 6: For Apache Tomcat configuration it will ask for JAVA_HOME variable. To set JAVA_HOME variable, go on drive at which you installed Java mostly in c drive. Copy the path such as local disk c-> program files -> Java -> jdk 11.0.8
Copy the path and open system environment variable setting. Add new variable where variable name will be JAVA_HOME and paste the path at path section.
To check JAVA_HOME path is configured or not properly, open command prompt as administrator and type java -version it will show the Java version and Java home version.

Step 7: Copy jenkins.war file at Apache Tomcat webapps folder.

Step 8: Now, go on Apache Tomcat folder -> bin and copy the path, open the command prompt as administrator and change the directory as per copied path and type the command startup. The system will open Apache Tomcat server.

Step 9: command prompt will show the port number for Apache Tomcat. To check Apache Tomcat installed successfully or not then type http://localhost:portnumber/Jenkins
Jenkins will ask you administrator password, paste the password and you can see Tomcat server successfully installed message.
If you can't find the port then go on Apache tomcat folder and open conf folder and edit server.xml and change the port as per your wish.

Step 10: Now close the Apache tomcat and open Jenkins dashboard. To open Jenkins dashboard first we will run war file through command prompt and then open browser.
Type jenkins port such as http://localhost:portnumber and it will open the login page. Type the username and password.

Step 11: click on manage Jenkins and visit configure tool.
Step 11. 1: Add JDK Installer, name will be JDK name such as JDK 11.0.8 and path will be your

system JDK path.
Step 11.2 : Add maven installer, name will be maven name such as maven-3.0.8 and path will be your system maven path.

Step 12: visit manage plugin section and click on available tab and install Deploy to container plugin.

Step 13: At new tab open this git link https://github.com/seedstack/store-webapp-sample and copy the .git repository URL.

Step 14: Now back to Jenkins dashboard and click on create new item.

Step 15: Enter the project name and select freestyle project(which type of your project), then click on ok.

Step 16: In the description, the section adds about your project. Such as "sample web application deployment"

Step 17: Tick mark on GitHub project checkbox. add the repository of your project URL.

Step 18: In build triggers section tick mark on poll SCM (here you can schedule your job (like Cron jobs) here * * * * * means for every one minute clean and build and deploy the project into a tomcat server.

Step 19: In Build Environment section tick mark on Delete workspace before build starts.

Step 20: In build section select invoke top-level maven targets(here I am creating the maven project right that why here I am selecting maven targets). In the maven version select maven Version which we configured at manage Jenkins. And type install at goal section

Step 21: In Post-build Actions add username and password which we used for Apache Tomcat sever and select the deploy war/ear to a container, in WAR/EAR files section add the **/*.war(it means fetch the war file from the workspace).

Step 22: In theContext path add /

Step 23: Add the Tomcat URL (http://localhost:portnumber).

Step 24: Click on save and apply buttons and then click on the build now button. your build process is running after the successful build (it will show the blue button) if the build fails it will show the red button and you can also check the console output.

Step 25: After successful deployment, you can check your tomcat and test the application whether it's successfully deployed or not.