Experiment No. 5: To create and build CI/CD Pipeline in Jenkins to test and deploy an application over tomcat server.

Configuration of Jenkins:

Step1: Now, we need to specify the Java location to Jenkins. Go back to your server command prompt and use the code below to fetch the directory of Java. Multiple directories will be listed using the below code. In our case, the directory is: '/usr/lib/jvm/java-11-openjdk-amd64/bin/java'.

find / -type f -name java

Step 2) Copy the above location and go back to your Jenkins Dashboard. Look for Global Tool

Configuration under the Manage Jenkins menu.

Step 3) Unselect the Install automatically button from the JDK window and fill the fields. Paste the java location path and trim it as shown in the image below. In the below Git window, select the install automatically checkbox.

For Git, check the Install automatically box.

Step 4) Similarly, check the box for Maven and fill the name field. Save all the settings, and now the configuration of Jenkins is completed.

Name: Mavenlatest

Install automatically: checked

Install from Apache Version: 3.9.10

Creating CI/CD Pipeline:

Step 1) Create New Item, select Freestyle Project and provide a name to your item.

Step 2) Switch to the Source Code Management window and paste your Github repository link. Specify your branch name of the repository below and Save it.

[Note: The above-linked Github repository 'https://github.com/sujataoak799/helloworld2025.git' contains a 'pom.xml' file used for Java compilation and generates a web app. It will be deployed to the server]

Step 3) Now click on Build Now button from the menu. With this step, all the repository files will be fetched by Jenkins. Click on Configure to go back to the same settings page.

Step 4) Click on Build Tab and select build step as ‘Invoke top-level Maven targets‘.

Step 5) Select your Maven name from the drop-down menu. Fill the goals with the multiple jobs you need to perform and separate them with one space. These goals are available in your repository, and you need to invoke them using Maven. The goals are ‘clean compile package’. Save it and again click on the ‘Build Now’ button from the menu as we did in the previous steps. Now the maven commands will be executed that will generate a war file.

Click on Build Now

Step 6) If you want to check the war file created in the previous steps, visit the workspace on your Jenkins dashboard or just run the directory commands in your server. Your directories and project name can vary, so you can use the ‘ls‘ command to see the list inside that directory and also keep in mind the directory name is case sensitive.

cd /var/lib/jenkins/workspace/

ls

cd <Pipeline Name>

ls

cd webapp

ls

cd target

ls

Step 7) Now go back to Configure and visit the ‘Post Build Actions’ tab. Click the drop-down and select ‘Archive the Artifacts’ from the options. In the field, write down ‘**/*.war’ as shown in the image below. It will fetch all the directories and get the war file wherever it is present. Click again on Build Now button, and you will now see the Artifacts in the Jenkins dashboard.

Click on build now.