**🧩 Experiment No. 5 — Jenkins CI/CD Pipeline with Tomcat Deployment**

**🧩 Part 1: Jenkins Setup**

**Step 1: Install Jenkins**

If not installed:

sudo apt update

sudo apt install openjdk-11-jdk -y

curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \

/usr/share/keyrings/jenkins-keyring.asc > /dev/null

echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \

https://pkg.jenkins.io/debian-stable binary/ | sudo tee \

/etc/apt/sources.list.d/jenkins.list > /dev/null

sudo apt update

sudo apt install jenkins -y

Start Jenkins:

sudo systemctl enable jenkins

sudo systemctl start jenkins

Access it in browser:  
👉 http://localhost:8080

Copy the password:

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

**🧩 Part 2: Configure Jenkins Tools**

**Step 2: Configure Java Path**

1. In the terminal, find Java path:
2. find / -type f -name java

Example: /usr/lib/jvm/java-11-openjdk-amd64/bin/java

1. Go to **Jenkins Dashboard → Manage Jenkins → Global Tool Configuration**
2. Under **JDK**, uncheck “Install Automatically”
3. Enter Name: JDK11
4. Paste the Java path (remove /bin/java, keep only /usr/lib/jvm/java-11-openjdk-amd64)
5. Under **Git**, check “Install Automatically”
6. Under **Maven**, enter Name: Maven and check “Install Automatically”
7. Click **Save**

**🧩 Part 3: Create the Jenkins Pipeline (CI)**

**Step 3: Create a New Job**

1. Go to **Dashboard → New Item**
2. Select **Freestyle Project**
3. Name it → CI-CD-Pipeline
4. Click **OK**

**Step 4: Connect GitHub Repository**

1. In the job configuration, go to **Source Code Management**
2. Select **Git**
3. Paste your repository URL (for example):
4. https://github.com/sujataoak799/hello-world2025.git
5. Enter branch name:
6. main

This repo must contain pom.xml (used by Maven to build .war files).

**Step 5: Add Build Step (Maven Build)**

1. Go to the **Build** tab
2. Click **Add build step → Invoke top-level Maven targets**
3. Select your Maven installation name from the dropdown
4. In **Goals**, type:
5. clean package

(This will clean and build your project into a .war file.)

1. Click **Save**
2. Now click **Build Now** → Jenkins will build the project and generate a .war file.

**Step 6: Verify the WAR File**

1. Go to Jenkins terminal or workspace:
2. cd /var/lib/jenkins/workspace/
3. ls
4. You’ll see your project folder → open it:
5. cd <project\_name>
6. ls

You’ll find a .war file created under target/.

**🧩 Part 4: Deploy to Tomcat (CD)**

**Step 7: Install Apache Tomcat**

cd /tmp

curl -O https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.109/bin/apache-tomcat-9.0.109.tar.gz

sudo mkdir /opt/tomcat

sudo tar xzvf apache-tomcat-9.0.109.tar.gz -C /opt/tomcat --strip-components=1

Set permissions:

sudo groupadd tomcat

sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat

sudo chgrp -R tomcat /opt/tomcat

sudo chmod -R g+r conf

sudo chmod g+x conf

sudo chown -R root:root /opt/tomcat

**Step 8: Change Tomcat Port**

1. Edit server.xml to avoid conflict with Jenkins (uses port 8080):
2. sudo vi /opt/tomcat/conf/server.xml
3. Change:
4. <Connector port="8080" ... />

to

<Connector port="8090" ... />

**Step 9: Set Tomcat Users**

Edit file:

sudo vi /opt/tomcat/conf/tomcat-users.xml

Add this before </tomcat-users>:

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<user username="admin" password="admin" roles="manager-gui,manager-script"/>

**Step 10: Remove IP Restriction**

sudo vi /opt/tomcat/webapps/manager/META-INF/context.xml

Delete everything inside <Context> tags and save.

**Step 11: Restart Tomcat**

cd /opt/tomcat/bin

./shutdown.sh

./startup.sh

Now Tomcat runs on:  
👉 http://localhost:8090

**🧩 Part 5: Jenkins Deployment Setup**

**Step 12: Install Jenkins Plugin**

1. Go to **Manage Jenkins → Manage Plugins → Available**
2. Search for:  
   ✅ *Deploy to Container Plugin*
3. Click **Install without restart**

**Step 13: Configure Post-Build Deployment**

1. Go to your job → **Configure**
2. Scroll to **Post-Build Actions → Add post-build action → Deploy war/ear to a container**
3. In WAR file field, type:
4. \*\*/\*.war
5. Click **Add Container → Tomcat 9.x Remote**
6. Enter:
   * **URL:** http://localhost:8090
   * **Credentials:** Add → Username: admin, Password: admin (as per Tomcat users file)
7. Save configuration.

**Step 14: Run the Pipeline**

Click **Build Now** again.  
If successful, Jenkins will automatically deploy your .war file to Tomcat.

**🧩 Step 15: Verify Deployment**

Open your browser:  
👉 http://localhost:8090/webapp/  
(or the name of your project folder)

You’ll see your web application running.