**Lesson 07 Demo 01**

**Configuring Deploy Plugin for Performing Automated CD**

**Objective:** To configure a CI/CD pipeline in Jenkins for deploying a Java application to Tomcat Apache

**Tools required:** GitHub, Jenkins, and Tomcat Apache

**Prerequisites:** You need to have a Jenkins up and running.

Steps to be followed:

1. Install Tomcat Apache 9 on Ubuntu VM
2. Log in to the Jenkins CI tool and install the Deploy to container plugin
3. Configure the deployment stage in the Jenkins pipeline

**Step 1: Install Tomcat Apache 9 on Ubuntu VM**

1. Open the terminal in your lab and use the following command to switch to the root user:

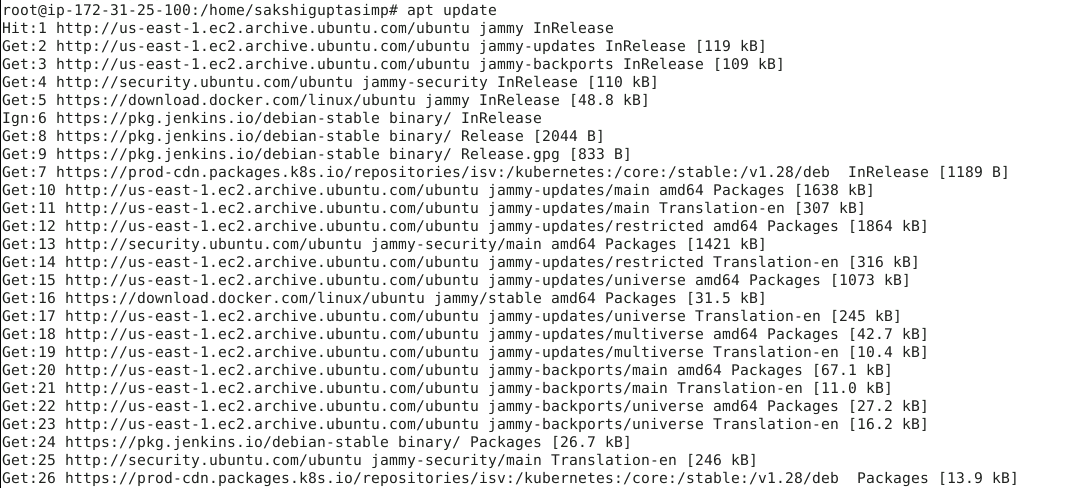
**sudo su**

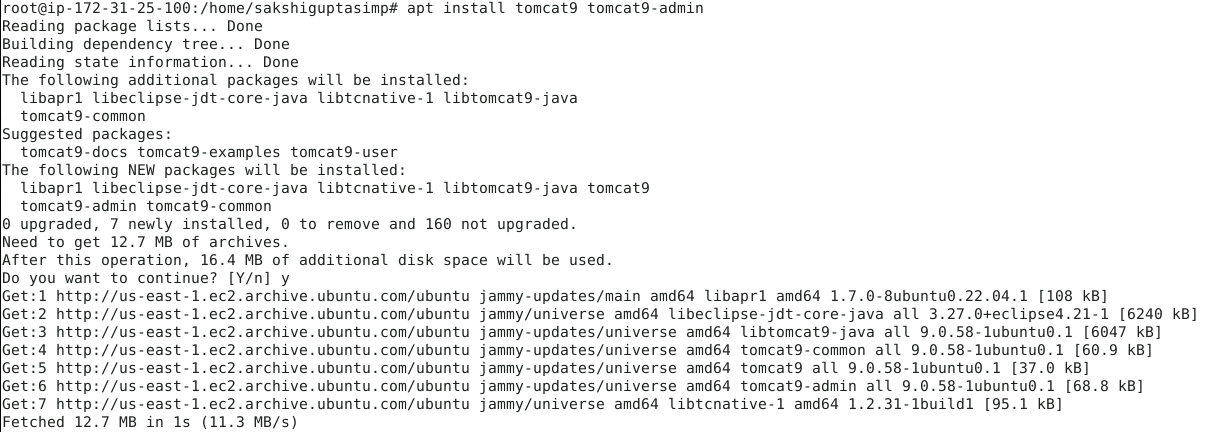


1. Install Tomcat Apache and other required packages using the following command:

**apt update**

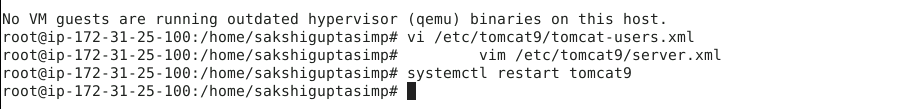
**apt install tomcat9 tomcat9-admin**





1. Once the installation is complete, open the **tomcat-users.xml** file using the following command:

**vi /etc/tomcat9/tomcat-users.xml**



1. Add the following content in **tomcat-users.xml** file:

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

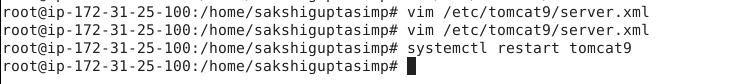
A screenshot of a computer program

Description automatically generated

**Note**: To save the file and exit, press **Esc**, then type **:wq**, and press **Enter**

1. Open the **server.xml** file using the following command and scroll down to change the connector port number of Tomcat to **9090**:

**vim /etc/tomcat9/server.xml**



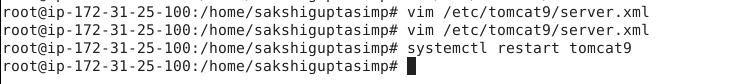
**A screenshot of a computer code

Description automatically generated**

**Note**: To save the file and exit, press **Esc**, then type **:wq**, and press **Enter**

1. Restart Tomcat using the following command:

**systemctl restart tomcat9**



1. Navigate to **localhost:9090** in your web browser and access Tomcat

A screenshot of a computer

Description automatically generated

1. Click and access **manager webapp** to make sure the Tomcat setup is complete

A screenshot of a computer

Description automatically generated

1. Enter the credentials and click on **Sign in**

A screenshot of a login box

Description automatically generated

**Note:** The credentials for accessing Tomcat manager web app are

Username: **tomcat** and Password: **password**.

A screenshot of a computer

Description automatically generated

**Step 2: Log in to Jenkins CI tool and install Deploy to container plugin**

1. Navigate to **localhost:8080** in your web browser, enter your credentials, and click on **Sign in**

A screenshot of a login form

Description automatically generated

|  |
| --- |
| **Note:** The credentials for accessing Jenkins in the lab are Username: **admin** and Password: **admin**. |

1. Click on **Manage Jenkins** on the Jenkins dashboard

**A screenshot of a computer

Description automatically generated**

1. Scroll down and click on **Plugins** under **System Configuration**

**A screenshot of a computer

Description automatically generated**

1. Navigate to **Available plugins** and search for **Deploy to container** plugin

A screenshot of a computer

Description automatically generated

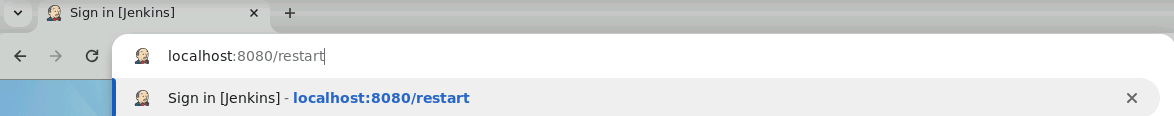
1. Select **Deploy to container** plugin, click on **Install**

A screenshot of a computer

Description automatically generated

1. After installation, navigate to the following URL to restart Jenkins:

[**http://localhost:8080/restart**](http://localhost:8080/restart)



A screenshot of a computer

Description automatically generated

**Step 3: Configure the deployment stage in the Jenkins pipeline**

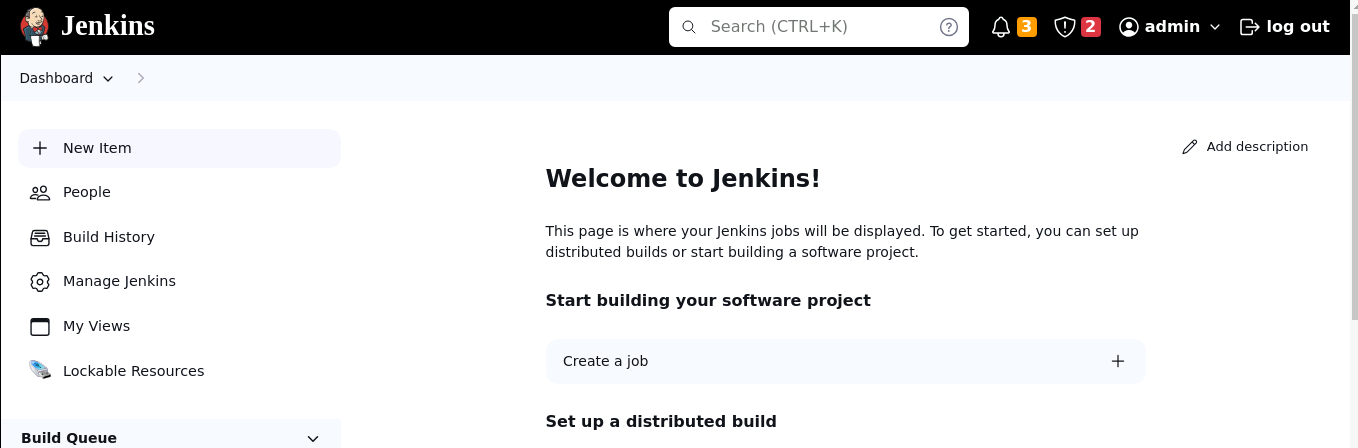
1. Enter your credentials and **Sign in** to Jenkins CI tool

|  |
| --- |
| **Note:** The credentials for accessing Jenkins in the lab are Username: **admin** and Password: **admin**. |

A screenshot of a login form

Description automatically generated

1. Click on **New Item** to create new Jenkins job

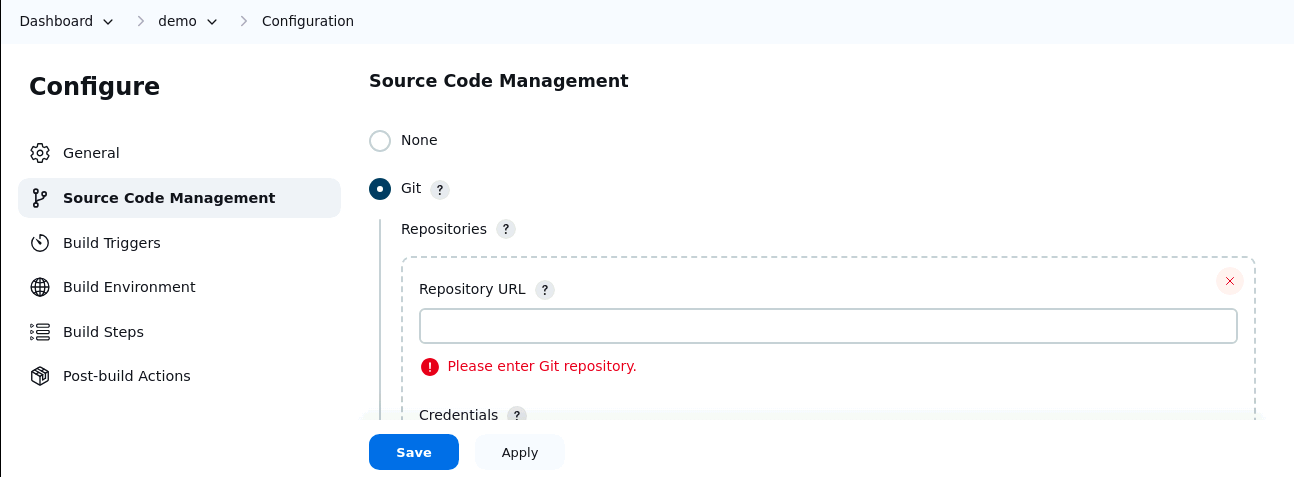


1. Select **Freestyle project** while creating Jenkins job and provide a custom job name, then click on **OK** to continue

A screenshot of a computer

Description automatically generated

1. Go to **Source Code Management** and select **Git**



1. Enter the following **Repository URL**:

**https://github.com/github-simplilearn-net/MavenBuild**

A screenshot of a computer

Description automatically generated

1. Go to **Build Steps** under **Configure**:

A screenshot of a chat

Description automatically generated

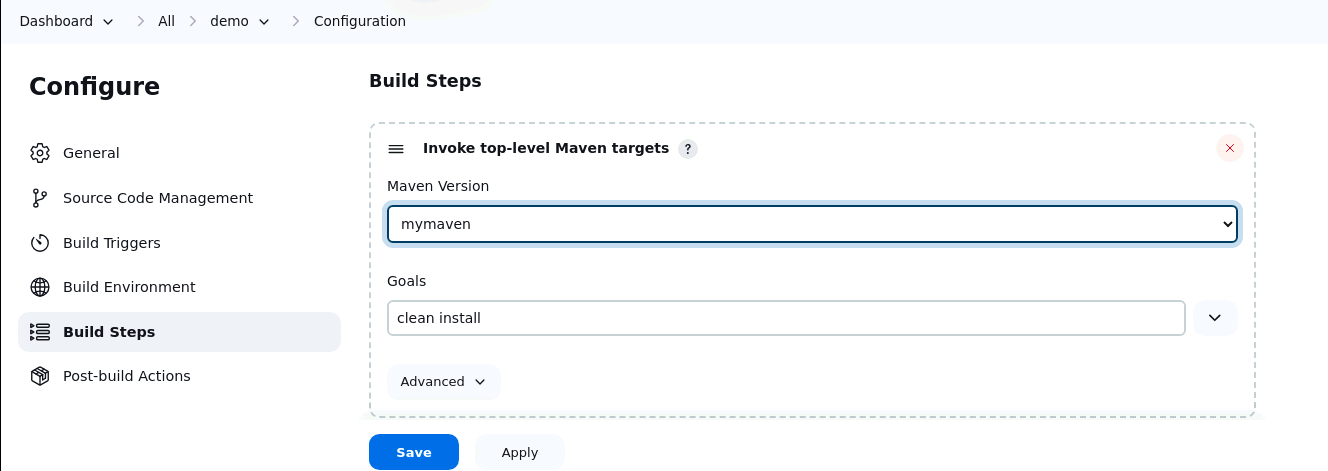
1. Click on **Add build step** and select **Invoke top-level Maven targets**

A screenshot of a computer

Description automatically generated

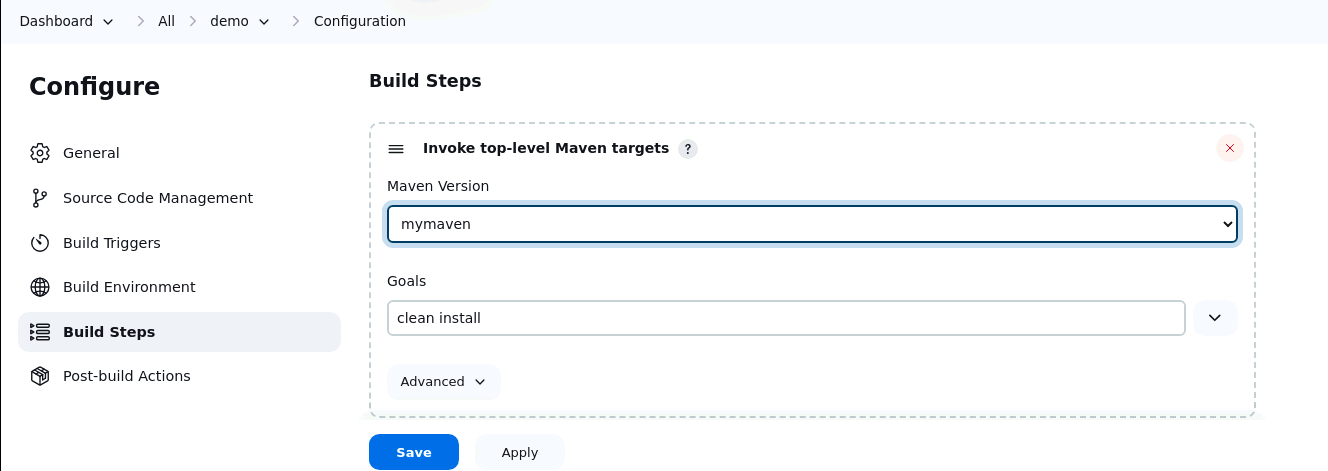
1. Enter the following **Maven Version**:

**mymaven**



1. Enter the following **Goals**:

**clean install**



1. Go to **Post-build Actions**

A screenshot of a chat

Description automatically generated

1. Click on **Add post-build action** and select **Deploy war/ear to a container**

A screenshot of a chat

Description automatically generated

1. Enter the following path under **WAR/EAR files**:

**\*\*/\*.war**

A screenshot of a computer

Description automatically generated

1. Click on **Add Container** and select **Tomcat 9.x Remote**

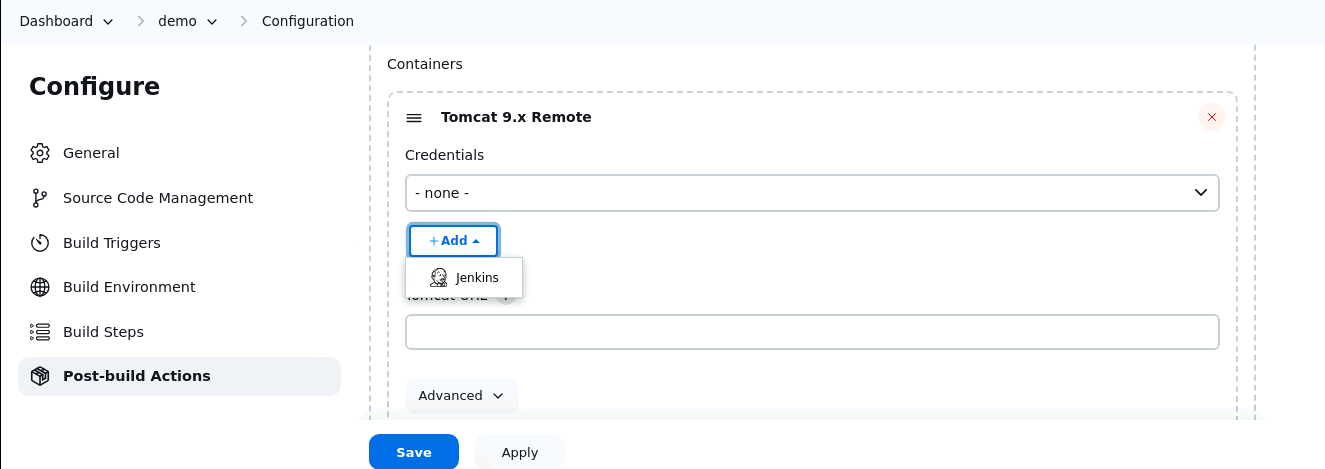
A screenshot of a computer

Description automatically generated

1. Click on **Add** under **Credentials** and select **Jenkins** to add the Tomcat credentials

A screenshot of a computer

Description automatically generated



1. Scroll down and enter the following details:

**Username: tomcat**

**Password: password**

**ID: tomcat-id**

**Description: tomcat-id**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

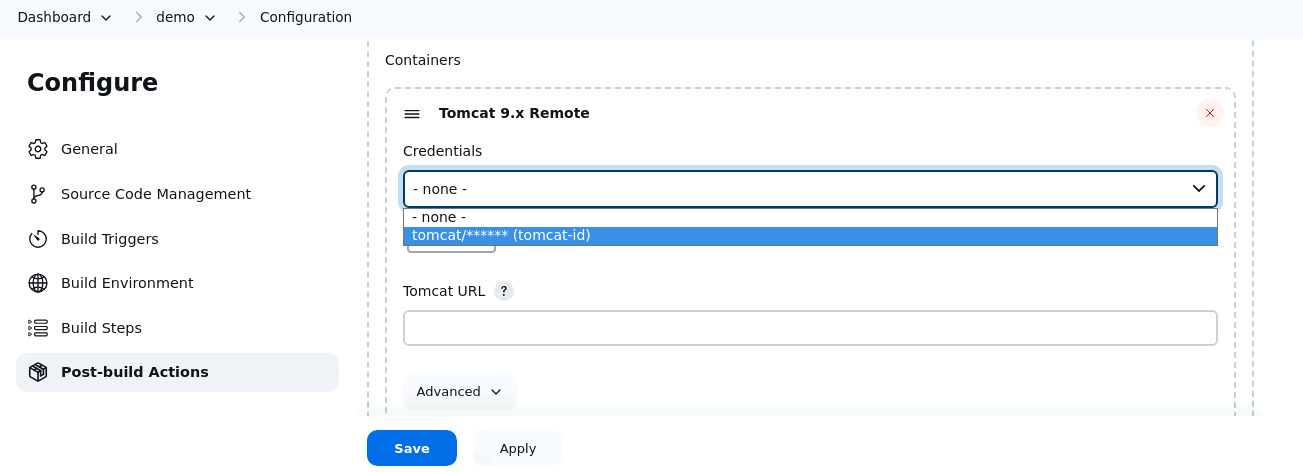
Description automatically generated

1. Click on **Add** to save the credentials

A screenshot of a computer

Description automatically generated

1. In the **Credentials** dropdown menu, select the added credentials



A screenshot of a computer

Description automatically generated

1. Enter the following **Tomcat URL** and **Save** the job:

**http://localhost:9090/**

A screenshot of a chat

Description automatically generated

1. Click on **Build Now** and then select **Console Output** to check the output

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

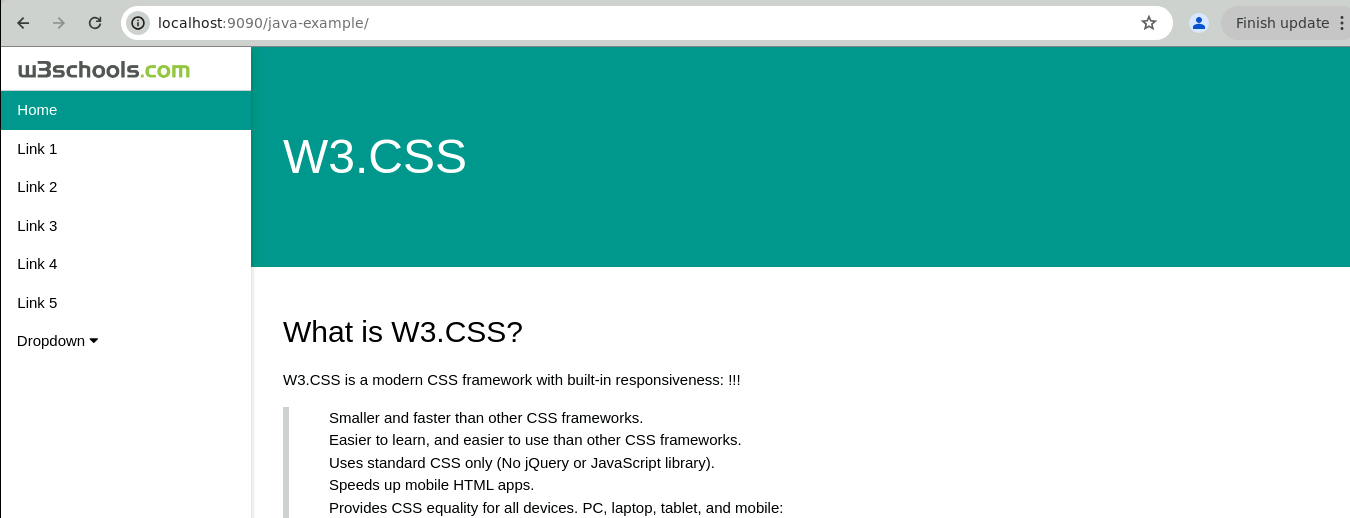
Description automatically generated

A screenshot of a computer

Description automatically generated

1. Access the deployed application in your web browser with the following URL:

**http://localhost:9090/java-example/**



By following these steps, you have successfully configured a CI/CD pipeline in Jenkins for deploying a Java application to Apache Tomcat.