**Lesson-End Project**

**Implementing CI/CD Pipeline for Deploying Application to Tomcat Apache**

**Project agenda**: To implement a continuous integration and continuous deployment (CI/CD) pipeline in Jenkins for deploying a Java application to Tomcat Apache

**Description:** You work as a DevOps Engineer in an IT firm. Your company is undertaking a project to modernize its application deployment processes, aiming to streamline the deployment of new software updates. As part of this initiative, you are tasked with setting up a continuous integration and continuous deployment (CI/CD) pipeline to automate the deployment of applications to the Tomcat Apache server hosted on an Ubuntu VM.

**Tools required:** Jenkins and Tomcat Apache

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

**Expected deliverables**: A step-by-step guide for setting up and implementing a CI/CD pipeline for deploying applications to Tomcat Apache on an Ubuntu VM, including integration with Jenkins for automated builds and deployments.

Steps to be followed:

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

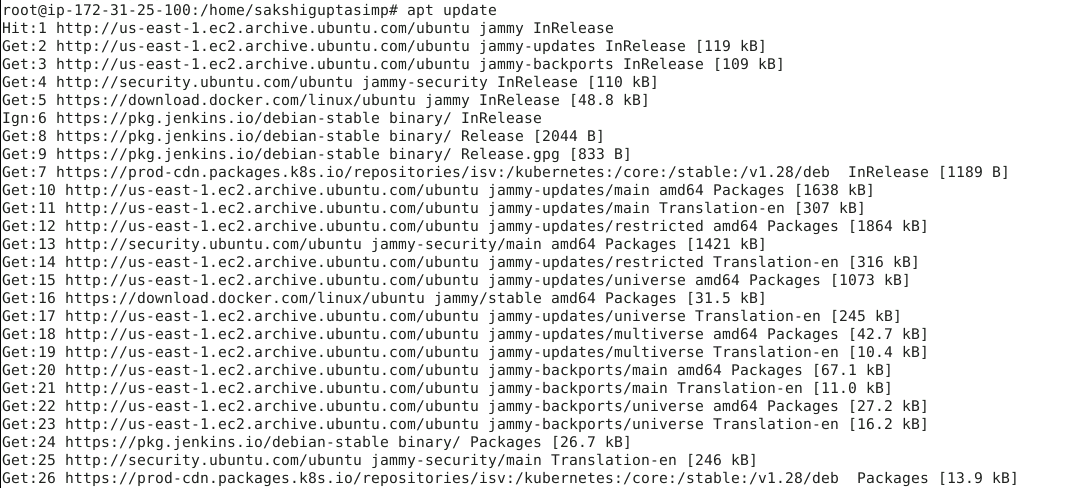
**sudo su**

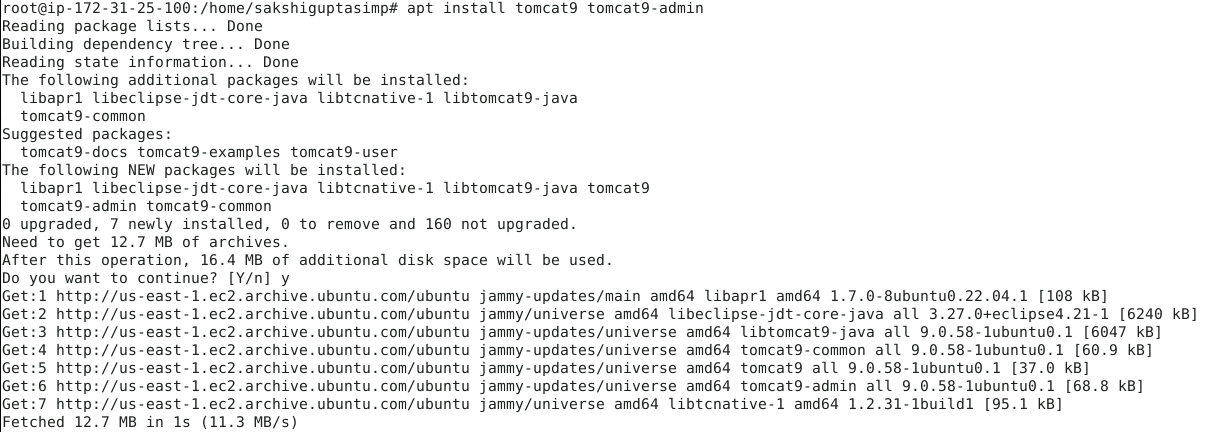


1. Install Tomcat Apache and other required packages using the following command:

**apt update**

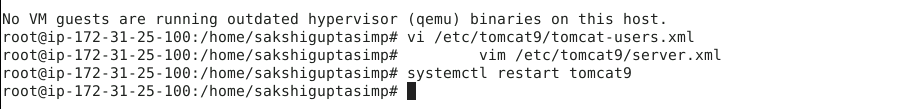
**apt install tomcat9 tomcat9-admin**





1. Once the installation is done, 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"/>**

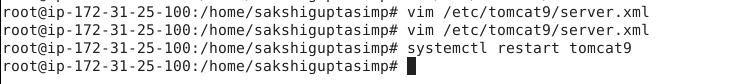
A screenshot of a computer program

Description automatically generated

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



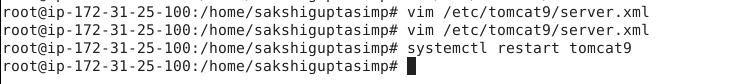
**A screenshot of a computer code

Description automatically generated**

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

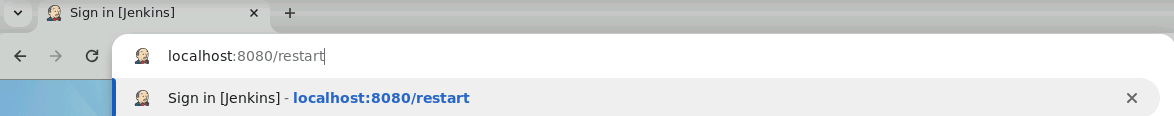
1. Select the **Deploy to container** plugin and click on **Install**

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 Jenkins pipeline**

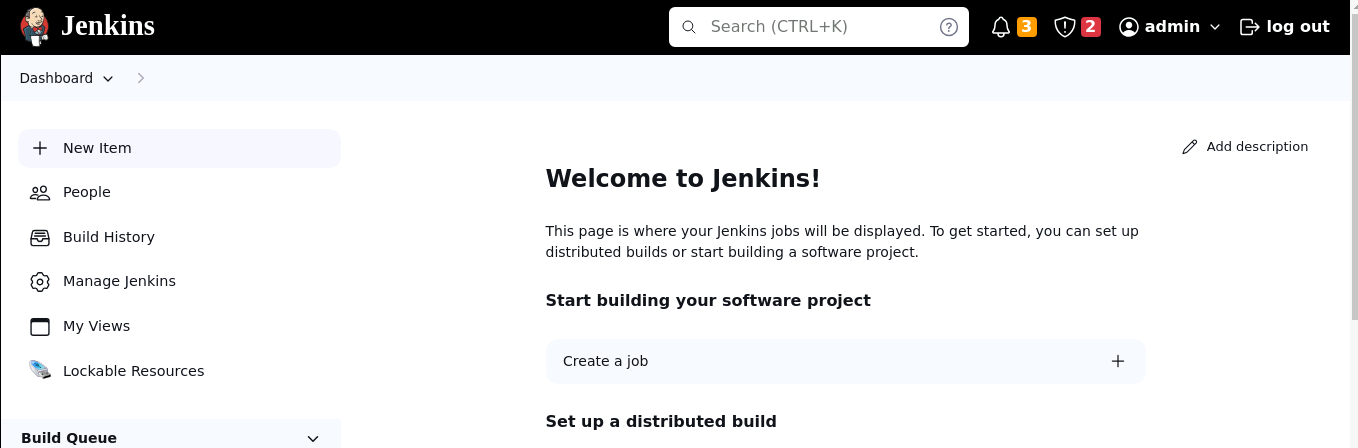
1. Enter your credentials and **Sign in** to the Jenkins CI tool

|  |
| --- |
| **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 a new Jenkins job



1. Select **Pipeline** when creating a Jenkins job, provide a custom job name, and then click **OK** to continue

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Go to the **Pipeline** section

A screenshot of a computer

Description automatically generated

1. Enter the following code under **Script** and **Save** the job:

**pipeline{**

**tools{**

**maven 'mymaven'**

**}**

**agent any**

**stages{**

**stage ('Clone a repo')**

**{**

**steps {**

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

**}**

**}**

**stage ('Package the code')**

**{**

**steps{**

**sh 'mvn package'**

**}**

**}**

**stage ('Deploy the code')**

**{**

**steps{**

**deploy adapters: [tomcat9(credentialsId: 'tomcat-id', path: '', url: 'http://localhost:9090/')], contextPath: null, war: '\*\*/\*.war'**

**}**

**}**

**}**

**}**

****

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

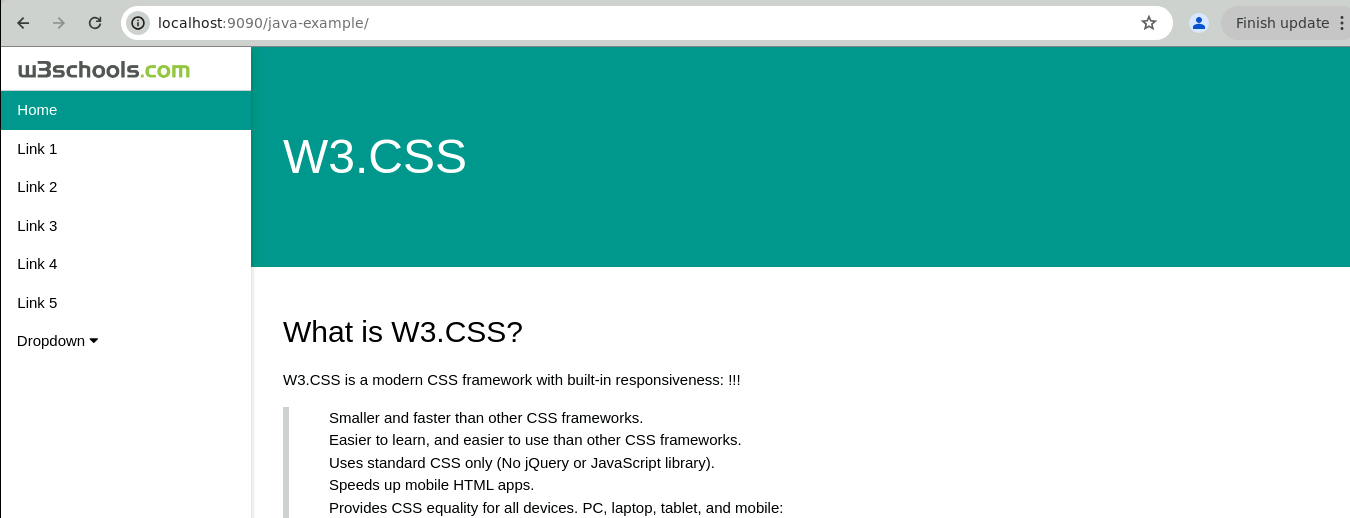
Description automatically generated

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 implemented a continuous integration and continuous deployment (CI/CD) pipeline in Jenkins for deploying a Java application to Tomcat Apache.