Logo

Description automatically generated with low confidence Logo

Description automatically generated

**Devops Lab**

Pre-requisite for the lab

JDK or JRE will need to be installed on the Windows Server before you can configure Tomcat 9 on the server. OpenJDK and Amazon Corretto are two examples of open-source Java Development Kit providers.

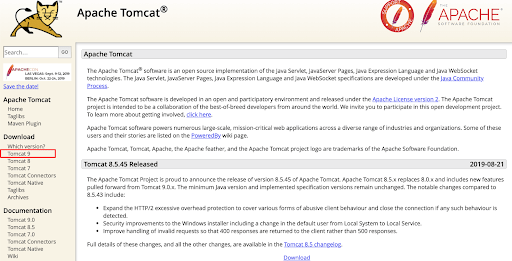
Logo

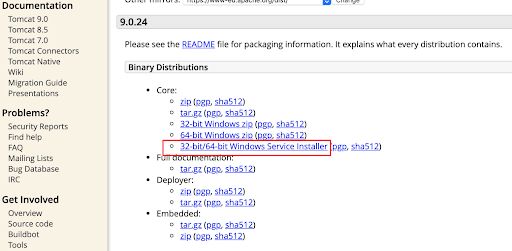
Description automatically generated

**Lab1**

**Use Case: Download and Install Apache Tomcat**

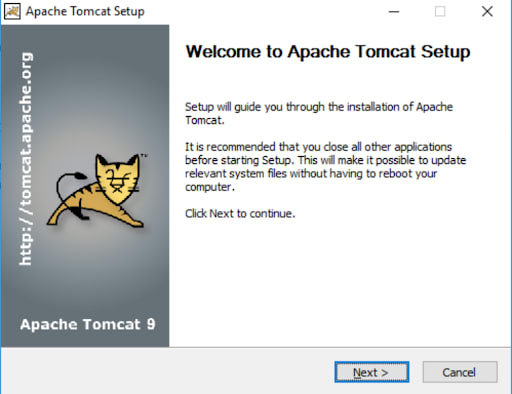
Open your browser and head over to https://tomcat.apache.org.  
Scroll down a little to locate and click on the [Tomcat 9 link](https://tomcat.apache.org/download-90.cgi) located within the left menu bar.





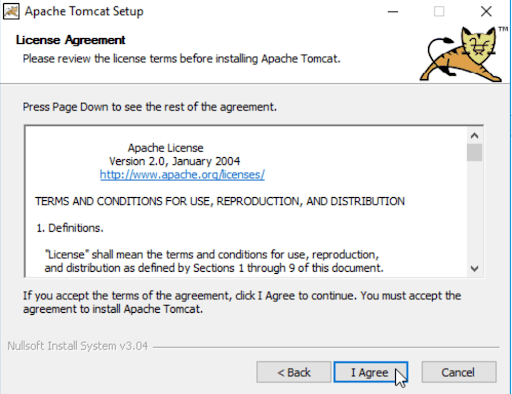
Step 1.

The first page of the install explains what the installer will do and what to expect. Go ahead and click Next on the first page.



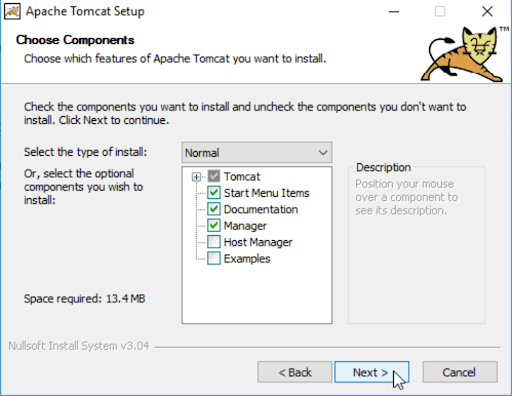
Step 2.

Before you can start the install, you must agree to the Apache License Agreement for the Tomcat 9 service



Step 3.

Click on “Select the type of install” dropdown list and choose the “Full” install option and then click Next.



Step 4.

The next screen is the configuration screen. This screen will allow you to set up any default ports that you want the service to connect through, and will also allow you to set an Administrator username and password.

Please note that you can configure these settings later if you choose to.

Choose your selections and click Next.

Graphical user interface, application

Description automatically generated

Step 5.

Next, you will choose the location where you want to install the Tomcat 9 service. Click Next

Graphical user interface, text, application, email

Description automatically generated

Step 6.

To correctly install Tomcat 9 on your server, the Wizard will want you to choose the location of any other Java related software on your server.

Graphical user interface, text, application, email

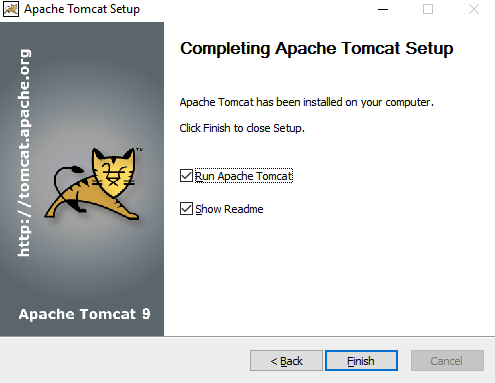
Description automatically generated

Step 7.

Once the install is complete, click Next.

Lastly, uncheck the Readme checkbox.

If you prefer to start Apache Tomcat 9 after the install, leave the checkbox marked. If not, then you will want to uncheck that checkbox.



To make sure the service is running, go to the Windows startup menu and type services.cmd. From the list of available services, find the Apache Tomcat 9 service, right-click on the service name, select Start, and make sure the service starts successfully. You should see a “running” status next to the service name.

To test the Tomcat 9 install and verify it is running on your server, open and point your browser to localhost:8080 (or whatever custom port you put into the configuration).

Graphical user interface, text, application

Description automatically generated

**Lab2- Integrate Tomcat with Jenkins**

**Use Case: Environment Configuration – Tomcat**

**Got to C:\apache-tomcat-9.0.43\conf directory and look for tomcat-users.xml file.**

Graphical user interface, table

Description automatically generated

Open the file in your favorite editor like notepad or vscode and add the following lines right before the last line

<**user** username="tomcatmanager" password="password" roles="manager-gui"/>

<**user** username="deployer" password="password" roles="manager-script"/>

Text

Description automatically generated

Here we are creating two usernames named tomcat manager and deployer. here the deployer account would be used to deploy the WAR file over http.

manager-gui based tomcat manager user would be used to manage the manager web application at http://<HostName>:8080/manager

Note\*: If you are using Tomcat8+ version, Steps to enable manager application might be different. Refer the product version specific documentation if you get stuck.

Now we are ready with the Tomcat Servlet Container Aka Application server, and it is ready to be connected from Jenkins.

**Jenkins Configuration:**

Install Deploy to container plugin

Manage Jenkins -> Manage Plugins -> Available -> Deploy to Container Plugin

Graphical user interface, text, application

Description automatically generated

Note:\* For the Next step we have selected a Maven Job as our Choice. you can also create a Free Style Project and use Gradle or Ant as your build tool

Create and Configure a Maven Job with Source Code Management (Gitlab)

Configure the Post-build Action and Specify the Tomcat Server Details

* Drag to the bottom and Go to the Post-build Actions section
* Click on Add post-build action button
* On the available options click on the Deploy war/ear to container

Graphical user interface, text, application, email

Description automatically generated

Fill the required parameters for the plugin. Use the following Screen Shot as the reference

Choose the Context Path in which the application should be installed. It would rename the WAR file before deploying to the server and thereby the application context root would be changed.

Tomcat URL http://[Tomcat Server Host]:[Primary http port]/

Graphical user interface, text, application, email

Description automatically generated

Build Jenkins Job

Console Output after the Successful build.

At the last line you can see that the WAR file has been generated and deployed on the remote server.

in our case, <http://192.168.0.107:8081/>

<http://192.168.0.107:8081/>

Table

Description automatically generated with low confidence

Testing the application

As the deployment is completed and the Jenkins Job ran Successfully without issues. ( Or So I presume)

Let us test our application.

In my case, the URL should be as follows

<http://192.168.0.107:8081/TomcatMavenApp>

Graphical user interface, application

Description automatically generated