

TOMCAT

Deployment types of applications:

1. Manual Deployment - Webserver (Tomcat) - Least preferred
2. Automated Deployment - Jenkins - Most preferred

Manual deployment of a Web application - Tomcat

=> In order to deploy any application, we need a server (EC2 Instance). If we want to deploy a java based web application, for this we need a web sever.

=> The web server which is available to deploy the java based web application is Apache Tomcat.

=> Maven (war file) ----> Deployment (Tomcat) = Deployment process.

=> Developers will write the code ---> The code will be tested by testing team ---> Once we get the approval from Testing team, Developers will keep the code into the GitHub Repository -----> As a DevOps Engineer, we have to take the code from GitHub Repo ---> Then we need to deploy the code into the Servers (EC2 Instances)

=> For java web applications, we use below servers;

1. Apache Tomcat, 2. JBoss, 3. Weblogic, 4. Websphere

Manual deployment

=> Java web applications will be deployed in the form of war files.

=> Java standalone applications will be deployed in the form of jar files.

=> Here we don't use any physical servers. So here we will use virtual servers (EC2 Instances) provided by the cloud service providers (AWS, GCP, Azure).

=> AWS ----> EC2 ----> Launch an instance (linux) ----> Install Tomcat, Install Java ----> Deploy the war of the app. ----> Access the application (We will see the 'Hello World' sample page).

=> Apache is the name of the organization. Tomcat is the name of the server.

=> Tomcat is an open-source s/w and it is available for FREE.

=> Tomcat server will run on port number 8080.

How to download the tomcat server into the local system

<https://tomcat.apache.org/download-90.cgi>

Click on Tomcat 9 in the downloads section

Folder structure of tomcat server

=> BIN (Binaries) - It contains the files to start and stop the tomcat server.

(.sh files - are used for linux OS) ----> startup.sh (This is used to start the server), shutdown.sh (This is used to stop the server)

(.bat files - are used for window OS) ----> startup.bat (This is used to start the server), shutdown.bat (This is used to stop the server)

=> CONF (Configuration) - It contains configuration files

(server.xml ----> To configure the tomcat port number)

(tomcat-users.xml ----> To configure the user details)

=> LIB (Library) - Library folder contains libraries i.e jar files

=> LOGS - It contains the server logs i.e execution details

=> TEMP (Temporary) - It contains the temporary data

=> WEBAPPS - In this folder we will keep the application war files. It is also called as "Deployment" folder.

Process

Developers (Write the code) ----> Push the code repo. ----> Clone the repo. ----> Package the file
----> war file ----> Keep the war file in the webapps folder.

=> We can keep multiple war files in the webapps folder. But it is not suggested.

Working with Tomcat in the linux ec2 instance

~ Launch the Linux based VM

~ Connect to the Instance

~ Install Java Software:

To install Java 11 on Amazon Linux:

```
sudo amazon-linux-extras install java-openjdk11
```

To install Java 08 on Amazon Linux:

```
sudo yum install java-1.8.0-openjdk
```

~ Check the Java Version:

```
java -version
```

Installation of Tomcat Web Server in Linux based VM

~ Since the VM is Linux based OS, we will install the "tar" file of Apache Tomcat Web Server

<https://tomcat.apache.org/download-90.cgi>

~ Keep the cursor on tar.gz file and right click and select 'Copy link address'

~ To install the Webserver: `wget <Paste the tar.gz link>` (Tar file of Apache Tomcat will get installed)

~ Extract the tar file

`tar -xvf <Name visible in the red colour in the above link>`

`ls -l` (You can see the extracted file in blue color)

~ `cd <Enter the name visible in blue colour in above line>`

`ls -l` (To see the folder structure)

Starting the Tomcat Server

~ To start the server, we need to access the `startup.sh` file. This file is available in BIN Folder

~ `cd bin/`

~ To start the server,

`./startup.sh` (You can see "Tomcat Started")

Accessing Tomcat Server

~ Copy the Public IP of EC2 Instance and paste in new tab ---- You will see the error page. This is because, the port no. 8080 is not enabled.

~ Enable Port No. 8080 in SGs of EC2 Instance.

Type: Custom TCP, Port: 8080, Source: Anywhere

~ After enabling the port no 8080, go to new tab,

<Paste the public IP of instance>:8080/ -----You can see tomcat server page

Deploy the Web Application in Tomcat Server

The war file has to be deployed in the WEBAPPS Folder

Goto 02-Maven-WebApp-CivilEdu ----> Open Command Prompt in that path ----> Lets package the project ----> mvn clean package ----> You can see Build Success ----> You can see the war file in target folder

Since we are accessing the Tomcat in our PC, we can only access the HOME PAGE of TOMCAT. We cannot perform the Admin activities from our PC.

To perform the admin activities from our PC, we need to edit the "context.xml" and configure "tomcat-user.xml" files.

Editing the context.xml to give the Admin permissions

File path of context.xml:

```
cd tomcat/webapps/manager/META-INF/context.xml
```

```
vi context.xml ----> Edit the allow tag ----> ".*" />
```

Configuring the Users in Tomcat

we need to edit the tomcat-users.xml to configure the users.

The "tomcat-users.xml" file is available in "conf" folder

```
<role rolename="manager-gui" />
```

```
<user username="tomcat" password="tomcat" roles="manager-gui" />
```

```
<role rolename="admin-gui" />
```

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

Click on 'Server Status' in the Tomcat Browser ----> Enter the username (admin) and password (admin)

Deploying the "war" file of our application

Scroll down to see "War file to deploy" -----> Click on Choose File ----> Select the war file from our PC ----> Click on "Deploy"

You can see our application under "List Applications"

Click on the Application Name (02-Maven-WebApp) -----> You will be able to see the HELLO WORLD!

Note: HELLO WORLD! is seen because it is available in index.jsp file which is there in "target folder"

This is all about the Manual Deployment.

By
Kastro Kiran V

LinkedIn: <https://www.linkedin.com/in/kastro-kiran/>

YouTube: https://www.youtube.com/playlist?list=PLs-PsDpuAuTdOcZa-DDgG8KRbtMI_XRrC

