

# DEPLOY APACHE WEB SERVER USING FREE STYLE JOBS

**Deployment :** Installing a application inside a server

(or)

Installing web application on web application server

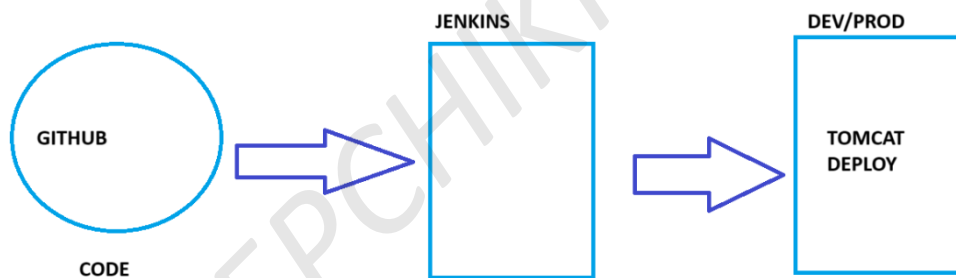
**Web Server :** A web server is a computer that runs websites. It stores web server software and a websites component files (Eg: HTML docs,images,css,js files)

(or)

A web server is a computer system capable of delivering web content to end users over the internet via a web browser

**Eg for web servers:** Apache tomcat, Nginx, IBM, HTTPD, websphere , etc...,

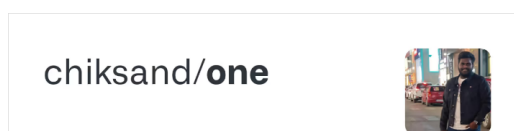
- So, based on our requirements we will use the web servers
- Present, we are using apache to deploy our code
- For java apps, we almost use apache web server
- For react and node JS we use HTTPD (or) Nginx



Here, we have to follow 4 steps i.e., code, build, test, deploy.

## Step-1 : (code)

- First create the jenkins setup, Create a job
- For github code, fork my github project. Below is my project url and username is “chiksand”



▲ Copy the url , clone and build and see the output in server

- Copy the url, save and build and see the output in server
- Go to workspace → myproject → src → main → webapp → index.jsp
- In this index.jsp file , we are having source code. This code will done by developers



## Step-2 : (Build)

- Here, we have to do maven setup
  - First install maven dependency i.e., java → yum install java-1.8.0-openjdk -y
    - So, here java version are not override. In linux, this is the benefit we had. We can maintain multiple versions
    - So, present you want to check, which java version is running means we have command
    - update-alternatives --config java
    - Through this command we are getting java versions “\* +” symbol was there in java version that means present that java is running
    - If you want to change java version, you can enter the number
    - If you want current version just click ENTER
  - Second, install maven → yum install maven -y
- Check the version → mvn -v

```
[root@ip-172-31-1-34 ~]# mvn -v
Apache Maven 3.0.5 (Red Hat 3.0.5-17)
Maven home: /usr/share/maven
Java version: 1.8.0_382, vendor: Red Hat, Inc.
Java home: /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.382.b05-1.amzn2.0.1.x86_64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "5.10.186-179.751.amzn2.x86_64", arch: "amd64", family: "unix"
[root@ip-172-31-1-34 ~]#
```

- make sure you have to get like above image means from server side you are clear

Go to jenkins dashboard → job → Configure → build steps → invoke top level maven target → inside goals → clean package → save → build

Now, Go to server → cd /var/lib/jenkins/workspace/Deployment → cd target → you got war file

Overall, Through maven we generated a war file

## Step -3 : (Test)

- Here, Testing we can't do. And this testing will done by testing engineers
- But previously we use maven, So testing already done through mvn test

## Step -4 : (Deploy)

Here, This deploy will be done inside web server. For that, we have to launch a new instance. We have to set up the Tomcat

1. Launch one normal server (t2.micro, normal-security group, 8gb volume, no key-pair)
2. Install tomcat dependency i.e., java → yum install java-1.8.0-openjdk -y
3. Go to browser → dlcdn.apache.org → tomcat/ → tomcat9/ → v9.0.78/ → bin → [apache-tomcat-9.0.78.tar.gz](https://dlcdn.apache.org/tomcat/9.0.78/bin/apache-tomcat-9.0.78.tar.gz)
4. Go to terminal → wget [apache-tomcat-9.0.78.tar.gz](https://dlcdn.apache.org/tomcat/9.0.78/bin/apache-tomcat-9.0.78.tar.gz)
5. gunzip the file → tar -zxuf fileName → ll → apache-tomcat-9.0.78/
6. Start the tomcat, Go to → cd apache-tomcat-9.0.78/ → cd bin/ → ./ startup.sh

7. Default port number for tomcat is 8080
8. Open tomcat in browser i.e. publicIP:8080. and select Manager App you will get below image

403 Access Denied

You are not authorized to view this page.

By default the Manager is only accessible from a browser running on the same machine as Tomcat. If you wish to modify this restriction, you'll need to edit the Manager's `context.xml` file.

If you have already configured the Manager application to allow access and you have used your browser's back button, used a saved book-mark or similar then you may have triggered the cross-site request forgery (CSRF) protection that has been enabled for the HTML interface of the Manager application. You will need to reset this protection by returning to the main Manager page. Once you return to this page, you will be able to continue using the Manager application's HTML interface normally. If you continue to see this access denied message, check that you have the necessary permissions to access this application.

If you have not changed any configuration files, please examine the file `conf/tomcat-users.xml` in your installation. That file must contain the credentials to let you use this webapp.

For example, to add the `manager-gui` role to a user named `tomcat` with a password of `secret`, add the following to the config file listed above.

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

Note that for Tomcat 7 onwards, the roles required to use the manager application were changed from the single `manager` role to the following four roles. You will need to assign the role(s) required for the functionality you wish to access.

- `manager-gui` - allows access to the HTML GUI and the status pages
- `manager-script` - allows access to the text interface and the status pages
- `manager-jmx` - allows access to the JMX proxy and the status pages
- `manager-status` - allows access to the status pages only

The HTML interface is protected against CSRF but the text and JMX interfaces are not. To maintain the CSRF protection:

- Users with the `manager-gui` role should not be granted either the `manager-script` or `manager-jmx` roles.
- If the text or jmx interfaces are accessed through a browser (e.g. for testing since these interfaces are intended for tools not humans) then the browser must be closed afterwards to terminate the session.

For more information - please see the [Manager App How-To](#).

Now above image is displaying tomcat is not access for us. So, read the description

1. Edit the manager's folder in context.xml file
- a. cd apache-tomcat-9.0.78/ → webapps → manager → META-INF → ui context.xml

i. Here, delete 21, 22 lines i.e., <value ..... > save and exit
2. Again try to access tomcat browser, it's asking credentials
- a. cd apache-tomcat-9.0.78/ → conf → ui tomcat-users.xml

i. Here, we need to add the credentials, we have this description in apache

ii. we have to add 2 roles i.e., manager-gui, manager-script

```
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<user username="tomcat" password="123456" roles="manager-gui,manager-script"/>
</tomcat-users>
```

Perform same above steps and you can give any username and password

b. So, here we changed the configuration file, we need to restart

- i. Go to cd apache-tomcat-9.0.78/ → cd bin → perform commands
- a. ./shutdown.sh

b. now, start the tomcat ./startup.sh



Tomcat Web Application Manager

Message: OK

Manager

List ApplicationsHTML Manager HelpManager HelpServer Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

- Now, you will get the tomcat web page. This is Tomcat Web Application Manager
- Default we're having 5 paths. So, whenever we did deploy it will comes after the manager like left side

From Terminal end we did successfully tomcat setup

- Now, Go to Jenkins → configure the job → post-build actions → we need deploy war container plugin
- we need to install the plugin. So, go to manage jenkins → plugins → available plugins → Deploy war/ear container → select and click install without restart
- Go to job → configure → post-build → we have deploy container → select
- we have options to fill. So, look into the below image and fill the options

Dashboard > Deployment > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Post-build Actions

Deploy war/ear to a container

WAR/EAR files ?

target/\*.war

Context path ?

SandeepChikkala

Containers

Tomcat 9.x Remote

Credentials

tomcat/\*\*\*\*\* (tomcat-credentials)

Add +

Tomcat URL ?

http://54.81.237.109:8080/

Save Apply

- context path means, we have to give the application name
- we have to create tomcat-credentials → select add → username and password → give tomcat username and password we did in tomcat setup. Then after give description like tomcat-credentials
- After done everything, save and build

Now, Refresh your tomcat page, we got “SandeepChikkala” in paths



Tomcat Web Application Manager

Message:	OK				
Manager					
<a href="#">List Applications</a>	<a href="#">HTML Manager Help</a>		<a href="#">Manager Help</a>	<a href="#">Server Status</a>	
Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<div>Start Stop Reload Undeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
/SandeepChikkala	None specified	Archetype Created Web Application	true	0	<div>Start Stop Reload Undeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
/docs	None specified	Tomcat Documentation	true	0	<div>Start Stop Reload Undeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
/examples	None specified	Servlet and JSP Examples	true	0	<div>Start Stop Reload Undeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
/host-manager	None specified	Tomcat Host Manager Application	true	0	<div>Start Stop Reload Undeploy</div>

So, now click on /SandeepChikkala. now we can see the output/Application output

**HEY OUR APP WAS DEPLOYED**

**hey this is second version**

webhook activated

So, we successfully Deployed the Apache Web application in Jenkins.