# Assignment No. 1

**Title :**

Installation and configuration of web servers and application servers

**Problem Statement :**

Study assignment

1. Installation and configuration of Apache Tomcat server on Linux.
2. Installation and configuration of JBoss server on Linux.
3. Installation and configuration of GlassFish server on Linux.
4. Installation and configuration of WebSphere server on Linux.

**Objective :**

To understand commands to install mentioned application servers.

To understand difference between web server and application server

**Theory :**

**Web Server and Application server**

Most of the times these terms Web Server and Application server are used interchangeably. Following are some of the key differences in features of Web Server and Application Server:

Web Server is designed to serve HTTP Content. App Server can also serve HTTP Content but is not limited to just HTTP. It can be provided other protocol support such as RMI/RPC

Web Server is mostly designed to serve static content, though most Web Servers have plugins to support scripting languages like Perl, PHP, ASP, JSP etc. through which these servers can generate dynamic HTTP content.

Most of the application servers have Web Server as integral part of them, that means App Server can do whatever Web Server is capable of. Additionally App Server have components and features to support Application level services such as Connection Pooling, Object Pooling, Transaction Support, Messaging , services etc.

As web servers are well suited for static content and app servers for dynamic content, most of the production environments have web server acting as reverse proxy to app server. That means while servicing a page request, static contents (such as images/Static HTML) are served by web server that interprets the request. Using some kind of filtering technique (mostly extension of requested resource) web server identifies dynamic content request and transparently forwards to app server .

Components of Tomcat

1. Catalina : It is the Servlet Container of Tomcat.
2. Coyote : Coyote acts as a connector and supports HTTP 1.1
3. Jasper : It is the Tomcat‘s JSP Engine.
4. Cluster : A component for load balancing to manage large applications.
5. High availability : A Tomcat component to schedule system upgrades and changes without affecting live environment.
6. Web Application : Manage Sessions, Support deployment across different environments.

This article will walk you throughout the process of installing Apache Tomcat 8 (i.e. 8.5.14) on Linux systems which includes RHEL, CentOS, Fedora, Debian, Ubuntu, etc.

Step 1: Installing Java 8

1. Before installing Tomcat make sure you have the latest version of Java Development Kit (JDK) installed and configured on the system. It is preferred to use oracle Java.

To install latest Oracle Java JDK (jdk-8u131) on Linux, you may like to refer our recent posts on Oracle jdk/jre/jar.

Step 2: Download and Install Apache Tomcat 8

2. Once latest Java installed and configured correctly on the system, we will move forward to download and install latest stable version of Tomcat 8

Steps for Tomcat installation

1. Download tomcat tar file.
2. create tomcat installtion directory anywhere using command mkdir command.
   e.g. mkdir /opt/tomcat_installation
3. cp apache-tomcat-{version}.tar.gz /opt/tomcat_installation
4. cd /opt/tomcat_installation
5. tar -xvf apache-tomcat-{version}.tar.gz
6. It will extract apache-tomcat-{version}.tar.gz
7. It will also include bin directory where there are binaries for tomcat.
8. cd bin.
9. ./startup.sh
   It will show Tomcat started.
10. Now copy html file that you want to host on tomcat server into tomcat_installation/webapps/
11. Open browser type http://localhost:8080/hello. if hello.html is copied in step number 10 in /opt/tomcat_installation/ webapps/ROOT/
12. Create lib directory using mkdir command inside tomcat_installation/webapps/
13. Now in order to host jsp pages which will connect to database we have to copy jstl-1.2.jar and mysql-connector.jar to tomcat_installation/webaps/ftp://192.168.4.87/pub.
14. shutdown tomcat server using ./shutdown.sh command from bin directory.
15. cp jsp files tomcat_installation/webapps/

**Getting started with JBoss AS 7 in Fedora**

From a terminal, install JBoss AS 7 using dnf or yum:
sudo dnf -y install jboss-as
sudo yum -y install jboss-as

Start the JBoss AS 7 system service:
sudo systemctl start jboss-as.service

Connect to the JBoss AS 7 management console (for the system instance):
sudo -u jboss-as sh -c "jboss-cli -c"

When you connect to the management console, the server will send a secret key challenge to the client. The client can only pass the challenge if it has physical direct access to the file system and the same permissions as the user running the server. Otherwise, you'd need to create and use a proper management user.

Stop the JBoss AS 7 system service:

sudo systemctl stop jboss-as.service

Create a user instance of JBoss AS 7:

jboss-as-cp -l $HOME/jboss-as-user-instance

Start the JBoss AS 7 user instance:

$HOME/jboss-as-user-instance/bin/standalone.sh

standalone.sh is a script generated by jboss-as-cp that effectively runs this command

JBOSS_BASE_DIR=$HOME/jboss-as-user-instance /usr/share/jboss-as/bin/standalone.sh -c standalone-web.xml

Connect to the JBoss AS 7 management console (for the user instance):

jboss-cli -c

**Conclusion :** By performing this experiment we have successfully learned installation and configuration of Apache Tomcat server , Jboss server , GlassFish server .