

ASSIGNMENT 1

TITLE INSTALLATION AND CONFIGURATION OF
WEB SERVERS AND APPLICATION SERVERS

DATE

PROBLEM

Study Assignment:

- STATEMENT
- Installation and configuration of Apache Tomcat Server on Linux.
 - Installation and configuration of JBoss Server on Linux.
 - Installation and configuration of GlassFish server on Linux.
 - Installation and configuration of WebSphere server on Linux.

- LEARNING
- To understand commands to install mentioned application servers.
 - To understand difference between web server & application server.

- OBJECTIVES
- LEARNING
- Students will be able to install Apache Tomcat, JBoss, GlassFish and Websphere software on Linux.
 - Students will know the difference between webserver and application server.

SW & HW REQUIREMENTS

Apache-tomcat 8.5.24, jdk 8, Jboss, Glassfish Servers, mysql, PC with the configurations latest version of G4 or OS, Open Source Fedora - GHz, 8 GB RAM, 500 GB HDD, 15" Color Monitor, Keyboard, Mouse.

THEORY WEB SERVER AND APPLICATION SERVER

Although 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. Additional App Server have components - & features to support Application level services such as Connection Pooling.
- 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 are served by web server that interprets the request. Using some kind of filtering technique web server identifies dynamic content request & transparently forward to app server.

Components of Tomcat :

1. Catalina : It is the Servlet Container of Tomcat.
2. Coyote : Coyote acts as a connector & supports HTTP.
3. Jasper : It is the Tomcat's JSP engine.
4. Cluster : A component for load balancing to manage large applications.
5. ^{Web Application} High Availability : Manage sessions, support deployment across different deployment.
6. High Availability : A Tomcat component to schedule system upgrades & changes without affecting live environment.

STEP 1 : Installing Java 8

Before installing Tomcat make sure you have latest version of JDK installed and configured on the system.

STEP 2 : Download and Install Apache Tomcat 8

- Download tomcat tar file.
- Create tomcat installation directory anywhere using command `mkdir` command.
- `cp apache-tomcat-{version}.tar.gz /opt/tomcat-installation`
- `cd /opt/tomcat-installation`
- `tar -xvf apache-tomcat-{version}.tar.gz`
- It will extract the contents and also include bin directory.
- `cd bin`
- `./ startup.sh`
- Copy html file that you want to host on tomcat server into `tomcat-installation/webapps/`
- Open browser `http://localhost:8080/hello` if `hello.html` is copied in `/Root/`
- Create lib directory using `mkdir` command inside `tomcat-installation/webapps/`

- Now in order to host jsp pages which will connect to database we have to copy jstl-1.2.jar & mysql-connector.jar to tomcat installation/webapps/ ftp://192.168.4.87/pub
- Shutdown tomcat server using ./shutdown.sh command from bin directory.
- cp jsp files tomcat installation/webapps/

Getting Started with JBoss AS 7 in Fedora

From a terminal, install JBoss AS 7 using drif or yum:

```
> sudo drif -y install jboss-as
> sudo yum -y install jboss-as
```

Start the JBoss AS7 system service :

```
> sudo systemctl start jboss-as.service
```

Connect to the JBoss AS 7 management console

> 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 & the same permissions as the user running the server.

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 AS7 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.xml
```

Connect to the JBoss AS 7 management console

```
> jboss-cli -c
```

GlassFish

- Download GlassFish 4 JEE 7 App Server for Linux

```
> sudo su
```

```
> cd </path/2>/latest-glassfish-unixsh
```

```
> chmod +x /latest-glassfish-unix.sh
```

Access at <http://localhost:4848>.

Conclusion Thus we have successfully installed and configured Tomcat, GlassFish and JBoss servers.