Application Server

Trainee Name : Chhavi Sharma
Newers ID : 4023
Mentor Name : Nishith Kulshrestha
College : UPES

1.What is the difference between an Application Server and a Web Server?

Ans.

	Application Server	Web Server
What is it?	It exposes business logic to the client applications through various protocols including HTTP.	A server that handles HTTP protocol.
Job	Application server is used to serve web based applications and enterprise based applications	Web server is used to serve web based applications.
Functions	To deliver various applications to another device, it allows everyone in the network to run software off of the same machine.	Keeping HTML, PHP, ASP, etc files available for the web browsers to view when a user accesses the site on the web.
Resource utilization	High	Low

2.What is Catalina?

Ans. Catalina is Tomcat's servlet container.It provides Tomcat's actual implementation of the servlet specification. So when you start up theTomcat server, you're actually starting Catalina. It is the Java Engine (JRE / JVM) that's built into Tomcat and provides an environment in which Servlets can be run.

3.Describe tomcat directory structure.

Ans.

   /bin : Contains Start/Stop scripts for tomcat for both windows and linux operating system.

Also consists of JAR files with classes required to control the tomcat server.

/conf : Contains main configuration files (web.xml and server.xml).

/lib : Contains the Tomcat Java Archive (jar) files, shared across all Tomcat components.

/logs : Contains tomcat log files.
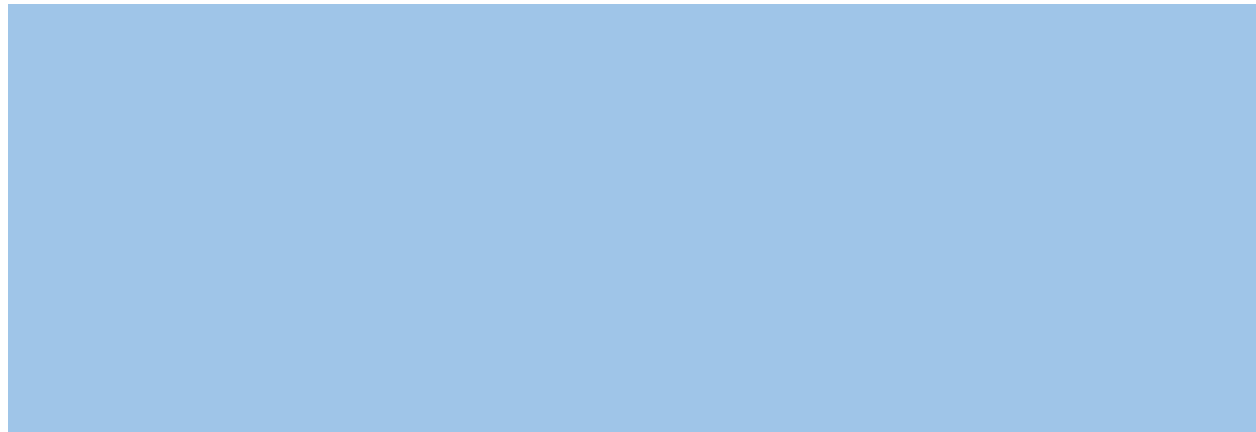
/temp: Temporary file storage.

/webapps : Deployment staging directory, where you place your WAR file.

/work : This is Tomcat`s working directory where Tomcat places all servlets that are generated from JSPs.

4.Connect any sample.war to MySQL running on localhost.

Ans.

In mysql create a user , a database and a table in that database.

In context.xml of tomcat9 paste the following.

```
    <!-- Uncomment this to disable session persistence across Tomcat restarts -->
    <!--
    <Manager pathname="" />
    -->

  <Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataSource"
                  maxActive="100" maxIdle="30" maxWait="10000"
                             username="chhavi" password="pass" driverClassName="com.mysql.jdbc.Driver"
                                         url="jdbc:mysql://localhost:3306/mydb"/>
</Context>
"context.xml" 36L, 1697C                                                              35,63-91      Bot
```

In sampleApp create a WEB-INF folder. IN that folder make a file web.xml and paste the following in it .

```
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
    version="2.4">
  <description>MySQL Test App</description>
  <resource-ref>
      <description>DB Connection</description>
      <res-ref-name>jdbc/TestDB</res-ref-name>
      <res-type>javax.sql.DataSource</res-type>
      <res-auth>Container</res-auth>
  </resource-ref>
</web-app>
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

In sampleApp create a test.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<sql:query var="rs" dataSource="jdbc/TestDB">
select id, foo, bar from testdata
</sql:query>

<html>
  <head>
    <title>DB Test</title>
  </head>
  <body>

  <h2>Results</h2>

<c:forEach var="row" items="${rs.rows}">
    Foo ${row.foo}<br/>
    Bar ${row.bar}<br/>
</c:forEach>

  </body>
</html>
~
~
~
~
~
~
~
~
~
~
:wq
```

Download standard.jar and jstl.jar, unzip it and move it into the bin folder in WEB-INF.

```
chhavi@chhavi:/var/lib/tomcat9/webapps/sampleApp/WEB-INF/lib$ sudo wget http://www.java2s.com/Code/JarDownload/jstl/jstl.
--2020-02-26 17:20:50--  http://www.java2s.com/Code/JarDownload/jstl/jstl.jar.zip
Resolving www.java2s.com (www.java2s.com)... 52.216.88.122
Connecting to www.java2s.com (www.java2s.com)|52.216.88.122|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 20047 (20K) [application/zip]
Saving to: 'jstl.jar.zip'

jstl.jar.zip              100%[===================================================================>]  19.58K  75.

2020-02-26 17:20:51 (75.5 KB/s) - 'jstl.jar.zip' saved [20047/20047]

chhavi@chhavi:/var/lib/tomcat9/webapps/sampleApp/WEB-INF/lib$ ls
jstl.jar.zip  standard.jar.zip
chhavi@chhavi:/var/lib/tomcat9/webapps/sampleApp/WEB-INF/lib$ unzip jstl.jar.zip
Archive:  jstl.jar.zip
error:  cannot create jstl.jar
        Permission denied
chhavi@chhavi:/var/lib/tomcat9/webapps/sampleApp/WEB-INF/lib$ sudo !!
sudo unzip jstl.jar.zip
Archive:  jstl.jar.zip
  inflating: jstl.jar
chhavi@chhavi:/var/lib/tomcat9/webapps/sampleApp/WEB-INF/lib$ sudo unzip standard.jar.zip
Archive:  standard.jar.zip
  inflating: standard.jar
chhavi@chhavi:/var/lib/tomcat9/webapps/sampleApp/WEB-INF/lib$ ls
jstl.jar  jstl.jar.zip  standard.jar  standard.jar.zip
```

Download jdbc driver for mysql connector

Extract the downloaded file. After downloading paste the file into WEB-INF/lib.

Now hit localhost:8080/sampleApp/test.jsp
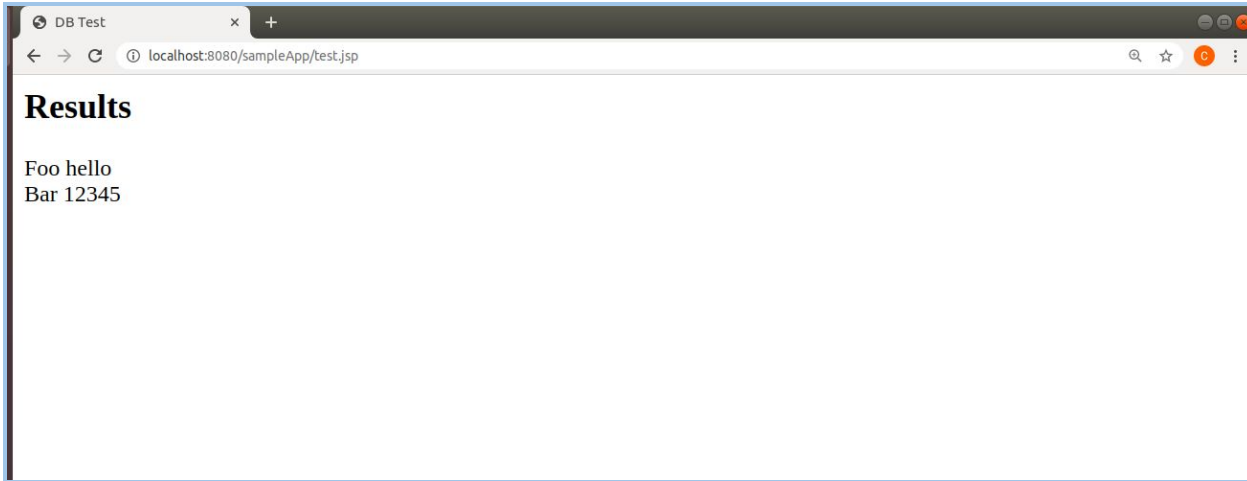
5.Run multiple services on different ports with different connectors (AJP/HTTP) on the same tomcat installation.

Make a new service block in the server.xml.

```
<Service name="Catalina2">

<Connector port="8081" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
<Engine name="Catalina2" defaultHost="localhost">
<Realm className="org.apache.catalina.realm.LockOutRealm">
       <!-- This Realm uses the UserDatabase configured in the global JNDI
            resources under the key "UserDatabase".  Any edits
            that are performed against this UserDatabase are immediately
            available for use by the Realm.  -->
<Realm className="org.apache.catalina.realm.UserDatabaseRealm"
           resourceName="UserDatabase"/>
</Realm>

<Host name="localhost"  appBase="webappsNew"
        unpackWARs="true" autoDeploy="true">
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
           prefix="localhost_access_log" suffix=".txt"
           pattern="%h %l %u %t &quot;%r&quot; %s %b" />

</Host>
</Engine>
</Service>
/Server>
```
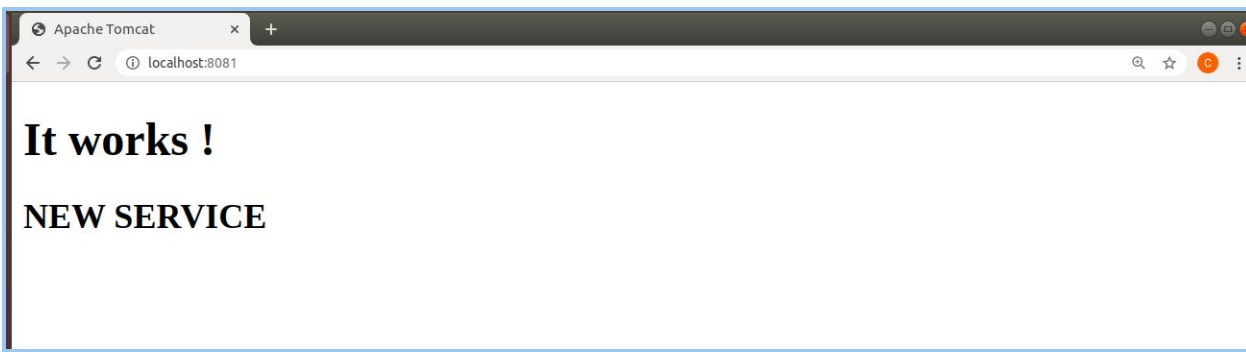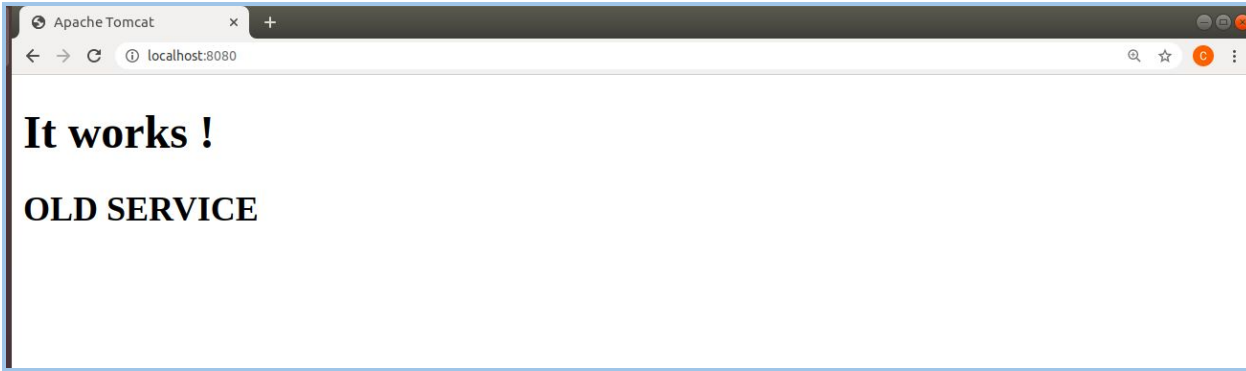
Make a new appbase and specify it in the new server block.Also specify a different port in the connector section.

```
chhavi@chhavi:/var/lib/tomcat9$ ls
conf  lib  logs  policy  webapps  webappsNew  work
```
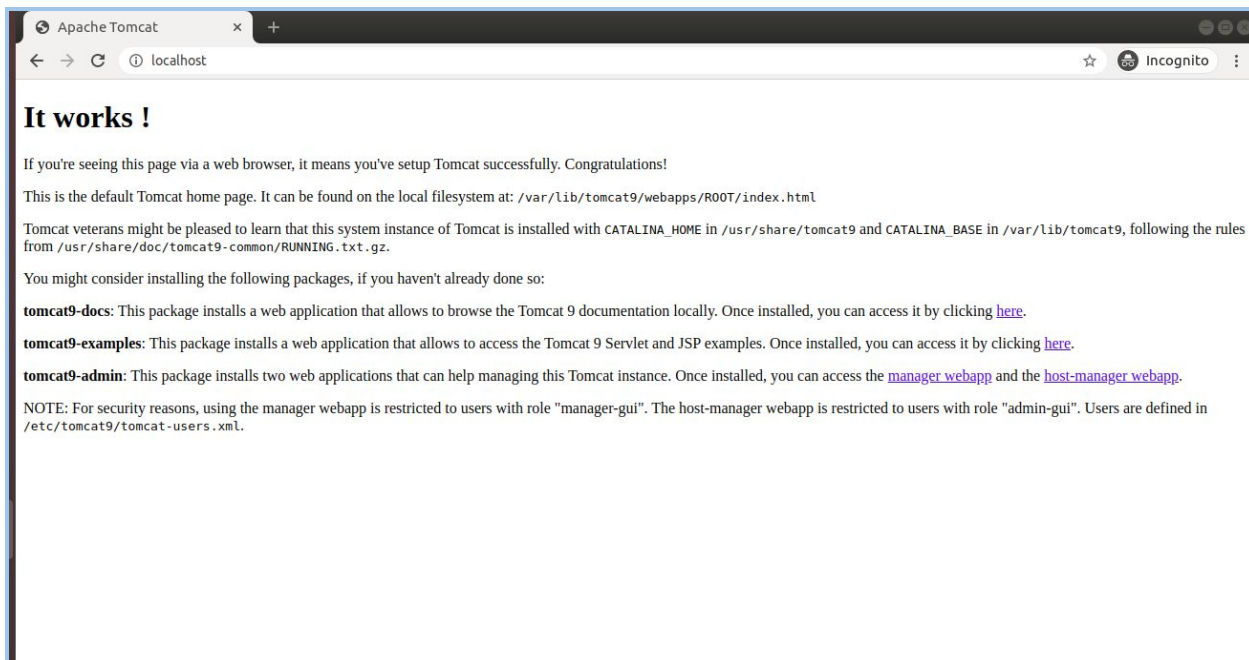
Hit localhost on different ports.(8080,8081)

Apache Tomcat

localhost:8080

It works !

OLD SERVICE

Apache Tomcat

localhost:8081

It works !

NEW SERVICE

6. Use nginx as reverse proxy for tomcat application.

Make the following changes in bootcamp.com(virtual host).
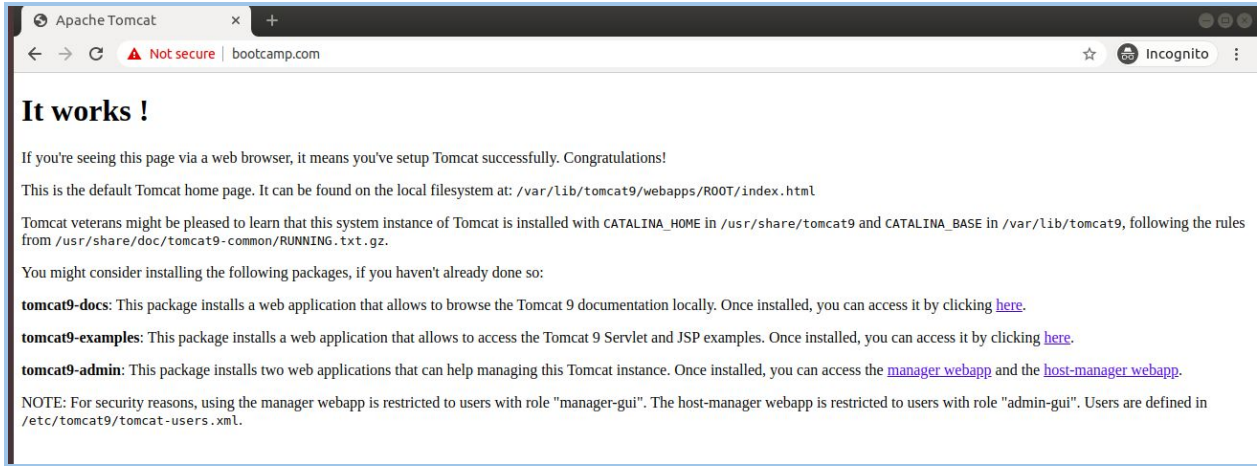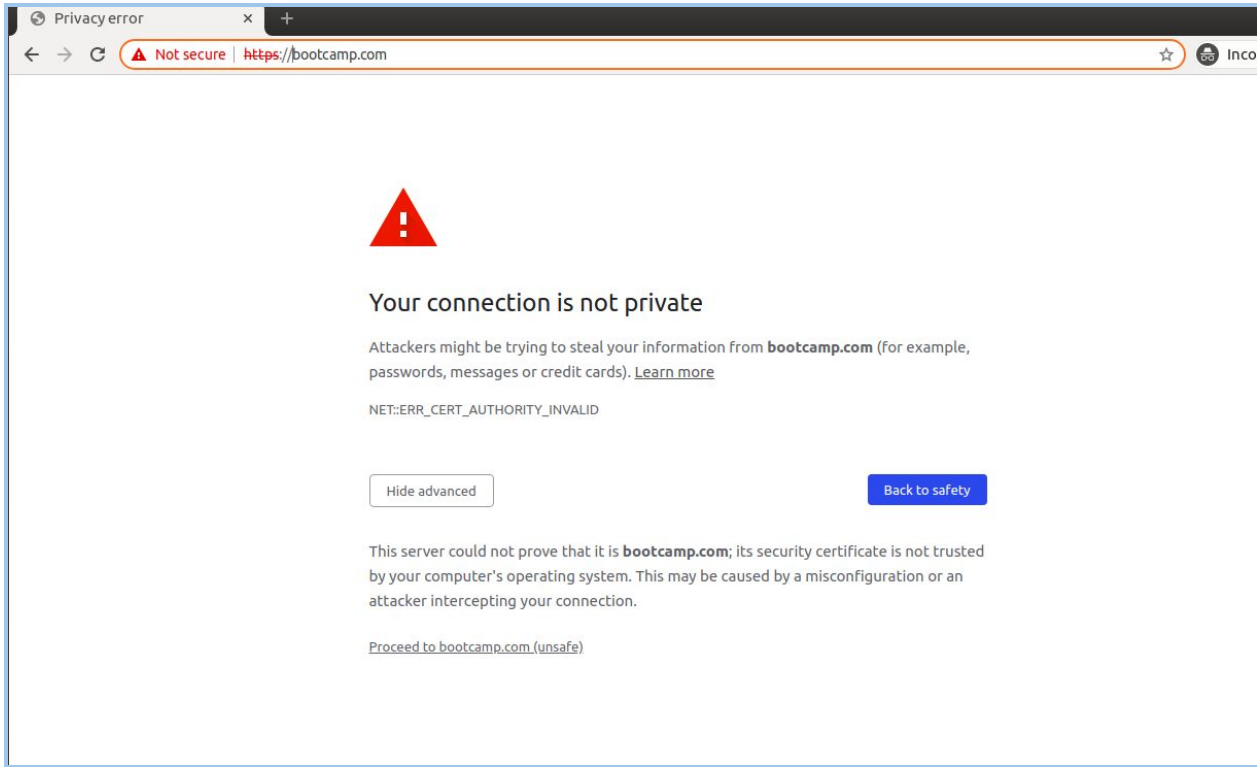

Write localhost in url

Nginx gets proxy passed to tomcat

6.1. Setup self signed certificate on that nginx for bootcamp.com.

Ans.

6.2.What is the difference between proxy_pass & proxy_pass reverse?

Ans.

Sno.	Forward proxy	Reverse proxy
1	forward **proxy is** used by the client such as a web browser	**reverse proxy is** used by the server such as a web server.
2.	Used for caching	Used in load balancing