APPLICATION SERVER

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

Ans 1.

Difference between web server and application server:

S.NO	WEB SERVER	APPLICATION SERVER
1.	Web server encompasses web container only.	While application server encompasses Web container as well as EJB container.
2.	Web server is useful or fitted for static content.	Whereas application server is fitted for dynamic content.
3.	Web server consumes or utilizes less resources.	While application server utilize more resources.
4.	Web servers arrange the run environment for web applications.	While application servers arrange the run environment for enterprises applications.
5.	In web servers, multithreading is not supported.	While in application server, multithreading is supported.
6.	Web server's capacity is lower than application server.	While application server's capacity is higher than web server.
7.	In web server, HTML and HTTP protocols are used.	While in this, GUI as well as HTTP and RPC/RMI protocols are used.

Ques 2. What is Catalina?

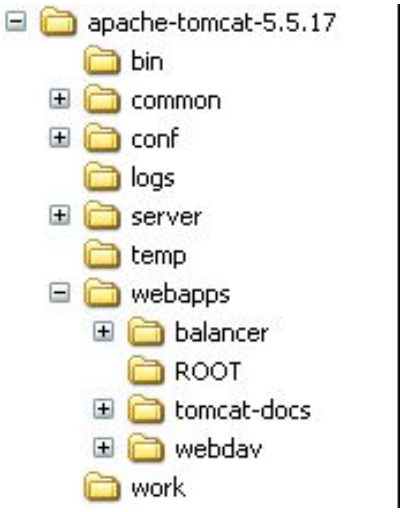
Ans 2.

Catalina is Tomcat's servlet container. Catalina implements Sun Microsystems' specifications for servlet and JavaServer Pages (JSP). In Tomcat, a Realm element represents a "database" of usernames, passwords, and roles (similar to Unix groups) assigned to those users. Different implementations of Realm allow Catalina to be integrated into environments where such authentication information is already being created and maintained, and then use that information to implement Container Managed Security as described in the Servlet Specification.

Ques 3. Describe tomcat directory structure.

Ans 3.

Once Tomcat has been installed, you will see a directory structure something like:



(This one was installed with my NetBeans installation.)

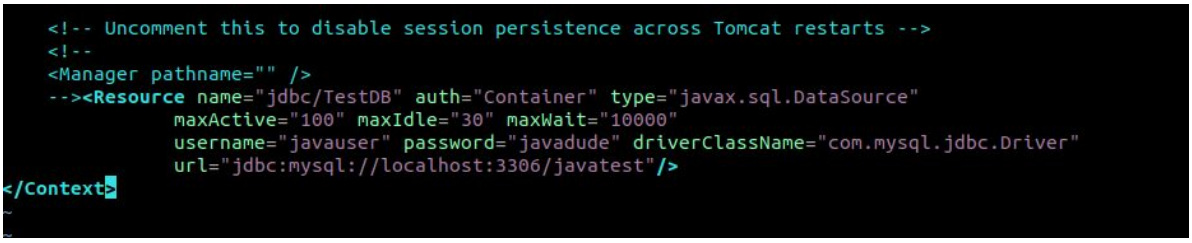
The typical directory hierarchy of a Tomcat installation consists of the following:

- **bin** - startup, shutdown and other scripts and executables
- **common** - common classes that Catalina and web applications can use
- **conf** - XML files and related DTDs to configure Tomcat
- **logs** - Catalina and application logs
- **server** - classes used only by Catalina
- **shared** - classes shared by all web applications
- **webapps** - directory containing the web applications
- **work** - temporary storage for files and directories

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

Ans 4.

Make changes in the context.xml file and create a resource for your mysql credentials.



Create the resource reference in web.xml file present in the WEB-INF directory

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<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">

    <display-name>Hello, World Application</display-name>
    <description>
        This is a simple web application with a source code organization
        based on the recommendations of the Application Developer's Guide.
    </description>

    <servlet>
        <servlet-name>HelloServlet</servlet-name>
        <servlet-class>mypackage.Hello</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>HelloServlet</servlet-name>
        <url-pattern>/hello</url-pattern>
    </servlet-mapping>
  <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>
~
```

Created a file test.jsp to display results

```
<%@ 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>
~
```

Entries in table

```
mysql> select * from testdata;
+----+-------+-------+
| id | foo   | bar   |
+----+-------+-------+
|  1 | hello | 12345 |
+----+-------+-------+
1 row in set (0.00 sec)

mysql>
```

JDBC Connector to connect from table

```
kaushlendra@kaushlendra:tomcat9 $ cd lib
kaushlendra@kaushlendra:lib $ ls
mysql-connector-java-8.0.19.jar
kaushlendra@kaushlendra:lib $
```

Ques 5. Run multiple services on different ports with different connectors (AJP/HTTP) on same tomcat installation.

Ans 5.

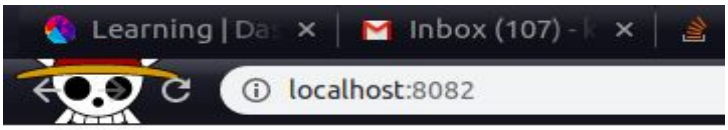
Created two directories in tomcat9 for instance

```
kaushlendra@kaushlendra:tomcat9 $ ls
conf  f1  f2  lib  logs  policy  webapps  work
kaushlendra@kaushlendra:tomcat9 $ cd conf
```

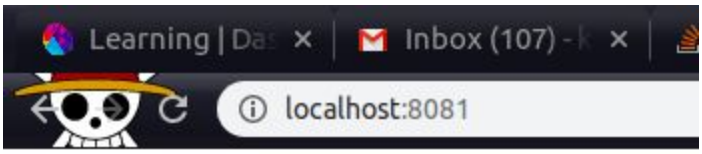
Then done some changes in server.xml

```
<Service name="app1">
   <Connector port="8081" protocol="org.apache.coyote.http11.Http11NioProtocol"
           connectionTimeout="20000"
           redirectPort="8443" />
   <Engine name="Catalina" defaultHost="localhost">
      <Host name="localhost"  appBase="f1"
        unpackWARs="true" autoDeploy="true">
      </Host>
   </Engine>
</Service>
<Service name="app2">
   <Connector port="8082" protocol="org.apache.coyote.http11.Http11NioProtocol"
           connectionTimeout="20000"
           redirectPort="8443" />
   <Engine name="Catalina" defaultHost="localhost">
      <Host name="localhost"  appBase="f2"
        unpackWARs="true" autoDeploy="true">
      </Host>
```

After this searched on browser



this is instance 2

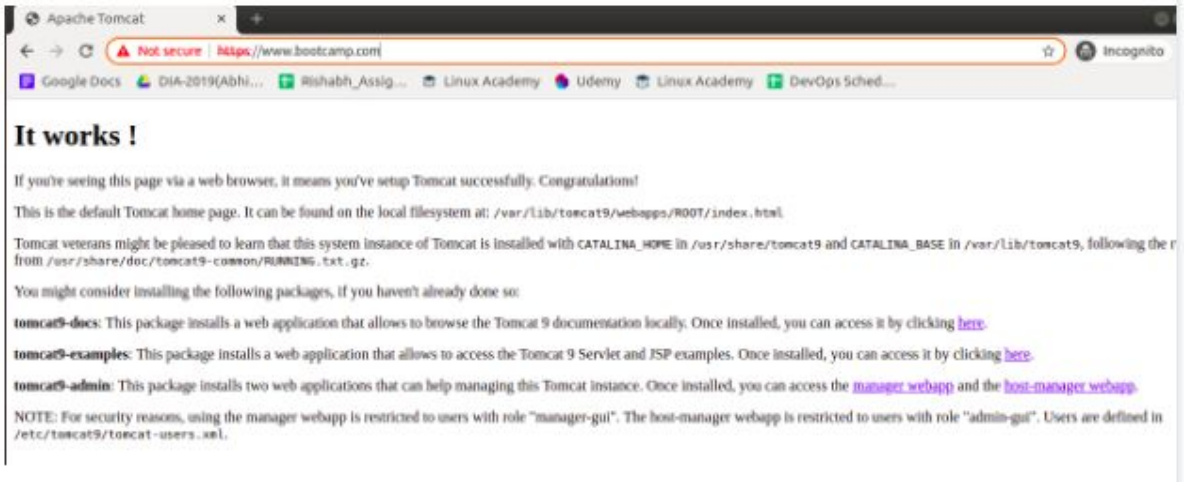


this is first instance

Ques 6.1 Use nginx as reverse proxy for tomcat application. Setup self signed certificate on that nginx for bootcamp.com.

Ans 6.1

```
server{
        listen 80;
        server_name www.bootcamp.com;
return 302 https://www.bootcamp.com;
}
server{
listen 443 ssl;
server_name www.bootcamp.com;
        ssl_certificate /etc/nginx/ssl/nginx.pem;
        ssl_certificate_key /etc/nginx/ssl/nginx.key;
location /{
                proxy_pass http://127.0.0.1:8080;
        }
}
~
```



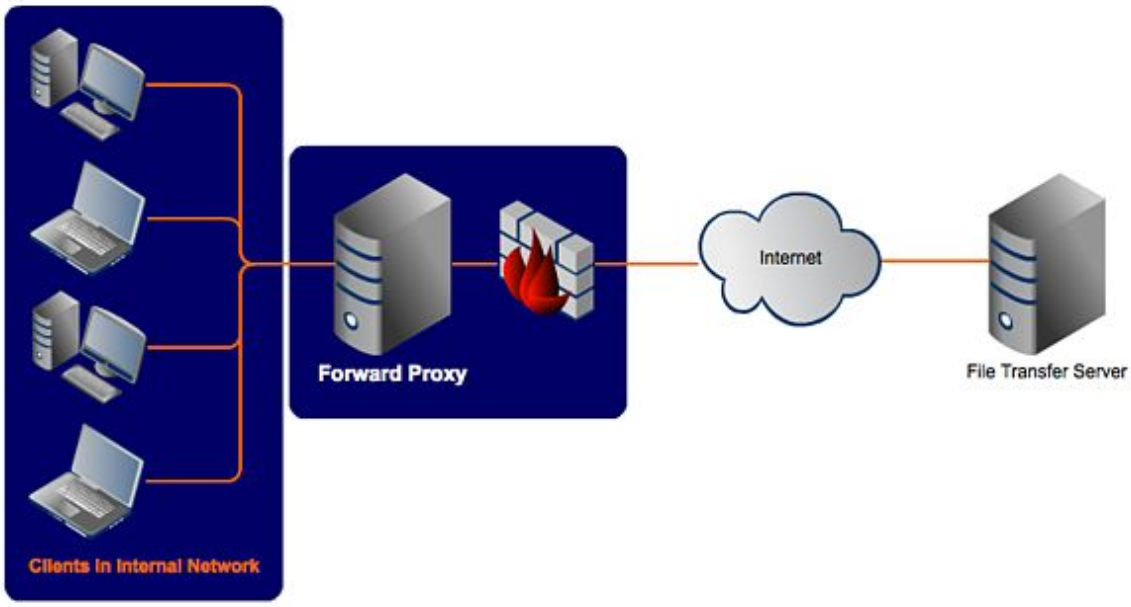
Ques 6.2 What is the difference between proxy_pass & proxy_pass reverse?

Ans 6.2

Proxy_pass

When people talk about a proxy server (often simply known as a "proxy"), more often than not they are referring to a forward proxy. Let me explain what this particular server does.

A forward proxy provides proxy services to a client or a group of clients. Oftentimes, these clients belong to a common internal network like the one shown below.

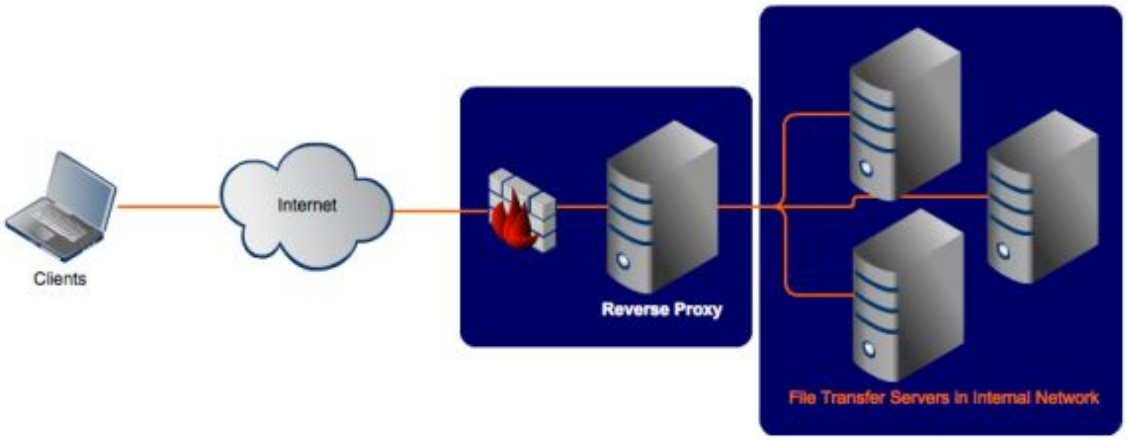


When one of these clients makes a connection attempt to that file transfer server on the Internet, its requests have to pass through the forward proxy first.

The Reverse Proxy

What is a reverse proxy? As its name implies, a reverse proxy does the exact opposite of what a forward proxy does. While a forward proxy proxies in behalf of clients (or requesting hosts), a reverse proxy proxies in behalf of servers. A reverse proxy accepts requests from external

clients on behalf of servers stationed behind it just like what the figure below illustrates.



To the client in our example, it is the reverse proxy that is providing file transfer services. The client is oblivious to the file transfer servers behind the proxy, which are actually providing those services. In effect, whereas a forward proxy hides the identities of clients, a reverse proxy hides the identities of servers.