Application Server:

Q1. **What is the difference between an Application Server and a Web Server?**

Web Server	Application Server
Web server encompasses web container only.	While application server encompasses Web container as well as EJB container.
Web server is useful or fitted for static content.	Whereas application server is fitted for dynamic content.
Web server consumes or utilizes less resources.	While application server utilize more resources.
Web servers arrange the run environment for web applications.	While application servers arrange the run environment for enterprises applications.
Web server's capacity is lower than application server.	While application server's capacity is higher than web server.

Q2. What is Catalina?

- 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.
- Catalina's default behavior can be directly configured by editing the six configuration files located in Tomcat's "$CATALINA_BASE/conf" directory.

Q3. Describe tomcat directory structure.

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

Q4. **Connect any sample.war to MySQL running on localhost.**

Entering data in database.

```
mysql> GRANT ALL PRIVILEGES ON *.* TO javauser@localhost
    -> IDENTIFIED BY '#viking0' WITH GRANT OPTION;
ERROR 1819 (HY000): Your password does not satisfy the current polic
y requirements
mysql> GRANT ALL PRIVILEGES ON *.* TO javauser@localhost
    -> IDENTIFIED BY '#Viking0' WITH GRANT OPTION;
Query OK, 0 rows affected, 1 warning (0.03 sec)

mysql> create database javatest;
Query OK, 1 row affected (0.00 sec)

mysql> use javatest;
Database changed
mysql> create table testdata (
    -> id int not null auto_increment primary key,
    -> foo varchar(25),
    -> bar int);
Query OK, 0 rows affected (0.32 sec)

mysql>
mysql>
mysql>
mysql>
mysql> insert into testdata values(null, 'hello', 12345);
Query OK, 1 row affected (0.06 sec)

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

mysql>
```

Making entry in context.xml

```
<Context>

    <!-- Default set of monitored resources. If one of these changes
, the    -->
    <!-- web application will be reloaded.
         -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
    <WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
    <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>

    <!-- Uncomment this to disable session persistence across Tomcat
 restarts -->
    <!--
    <Manager pathname="" />
    -->
 <Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataS
ource"
               maxActive="100" maxIdle="30" maxWait="10000"
               username="javauser" password="#Viking0" driverClassNa
me="com.mysql.jdbc.Driver"
               url="jdbc:mysql://localhost:3306/javatest"/>

</Context>
-- INSERT --                                    33,54         Bot
```

Putting the content in web.xml

```
vedant@Vedant-Sharma:/var/lib/tomcat9/webapps/sample1/WEB-INF$ cat web.xml
<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>
```

Downloading the connector for myql

```
vedant@Vedant-Sharma:~$ wget https://dev.mysql.com/get/Downloads/Connector
-J/mysql-connector-java_8.0.19-1ubuntu18.04_all.deb
--2020-02-26 13:33:21--  https://dev.mysql.com/get/Downloads/Connector-J/m
ysql-connector-java_8.0.19-1ubuntu18.04_all.deb
Resolving dev.mysql.com (dev.mysql.com)... 137.254.60.11
Connecting to dev.mysql.com (dev.mysql.com)|137.254.60.11|:443... connecte
d.
HTTP request sent, awaiting response... 302 Found
Location: https://cdn.mysql.com//Downloads/Connector-J/mysql-connector-jav
a_8.0.19-1ubuntu18.04_all.deb [following]
--2020-02-26 13:33:23--  https://cdn.mysql.com//Downloads/Connector-J/mysq
l-connector-java_8.0.19-1ubuntu18.04_all.deb
Resolving cdn.mysql.com (cdn.mysql.com)... 104.89.112.4
Connecting to cdn.mysql.com (cdn.mysql.com)|104.89.112.4|:443... connected
.
HTTP request sent, awaiting response... 200 OK
Length: 2327348 (2.2M) [application/x-debian-package]
Saving to: ‘mysql-connector-java_8.0.19-1ubuntu18.04_all.deb’

mysql-connector-ja 100%[===============>]   2.22M  5.12MB/s    in 0.4s

2020-02-26 13:33:24 (5.12 MB/s) - ‘mysql-connector-java_8.0.19-1ubuntu18.0
4_all.deb’ saved [2327348/2327348]
```

Downloading JSTL jar and standard.jar

```
vedant@Vedant-Sharma:~$ ls
assignments  Downloads                                      Public
bootcamp     Music                                          snap
Desktop      mysql-connector-java_8.0.19-1ubuntu18.04_all.deb  Videos
Documents    Pictures
vedant@Vedant-Sharma:~$ wget http://www.java2s.com/Code/JarDownload/jstl/j
stl.jar.zip
--2020-02-26 13:34:41--  http://www.java2s.com/Code/JarDownload/jstl/jstl.
jar.zip
Resolving www.java2s.com (www.java2s.com)... 52.216.108.242
Connecting to www.java2s.com (www.java2s.com)|52.216.108.242|:80... connec
ted.
HTTP request sent, awaiting response... 200 OK
Length: 20047 (20K) [application/zip]
Saving to: ‘jstl.jar.zip’

jstl.jar.zip        100%[===============>]  19.58K  68.0KB/s    in 0.3s

2020-02-26 13:34:42 (68.0 KB/s) - ‘jstl.jar.zip’ saved [20047/20047]

vedant@Vedant-Sharma:~$ ls
assignments  Downloads                                      Pictures
bootcamp     jstl.jar.zip                                   Public
Desktop      Music                                          snap
Documents    mysql-connector-java_8.0.19-1ubuntu18.04_all.deb  Videos
vedant@Vedant-Sharma:~$ wget http://www.java2s.com/Code/JarDownload/standa
rd/standard.jar.zip
--2020-02-26 13:35:11--  http://www.java2s.com/Code/JarDownload/standard/s
tandard.jar.zip
Resolving www.java2s.com (www.java2s.com)... 52.216.200.98
Connecting to www.java2s.com (www.java2s.com)|52.216.200.98|:80... connect
ed.
```

Unzipping:

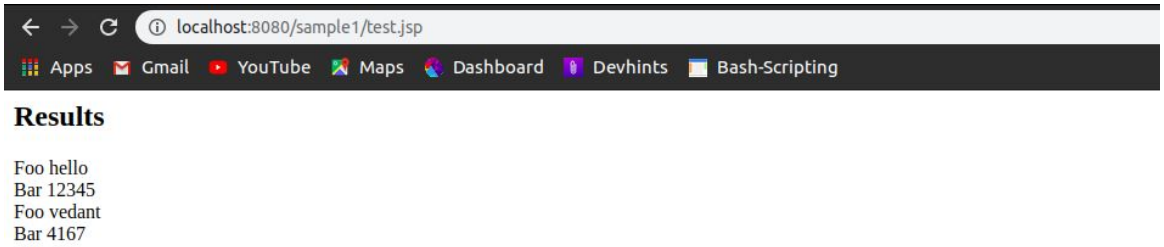
```
vedant@Vedant-Sharma:~$ ls
assignments    mysql-connector-java_8.0.19-1ubuntu18.04_all.deb
bootcamp       Pictures
Desktop        Public
Documents      snap
Downloads      standard.jar
jstl.jar       standard.jar.zip
jstl.jar.zip   Videos
Music
vedant@Vedant-Sharma:~$ sudo cp standard.jar jstl.jar /var/lib/tomcat9/web
apps/sample1/WEB-INF/lib
```

```
vedant@Vedant-Sharma:~$ dpkg -x mysql-connector-java_8.0.19-1ubuntu18.04_a
ll.deb
dpkg-deb: error: --extract needs a target directory.
Perhaps you should be using dpkg --install ?

Type dpkg-deb --help for help about manipulating *.deb files;
Type dpkg --help for help about installing and deinstalling packages.
vedant@Vedant-Sharma:~$ sudo !!
sudo dpkg -x mysql-connector-java_8.0.19-1ubuntu18.04_all.deb
dpkg-deb: error: --extract needs a target directory.
Perhaps you should be using dpkg --install ?

Type dpkg-deb --help for help about manipulating *.deb files;
Type dpkg --help for help about installing and deinstalling packages.
vedant@Vedant-Sharma:~$ sudo dpkg -x mysql-connector-java_8.0.19-1ubuntu18
.04_all.deb .
vedant@Vedant-Sharma:~$ ls
assignments    mysql-connector-java_8.0.19-1ubuntu18.04_all.deb
bootcamp       Pictures
Desktop        Public
Documents      snap
Downloads      standard.jar
jstl.jar       standard.jar.zip
jstl.jar.zip   usr
Music          Videos
vedant@Vedant-Sharma:~$ cd usr
vedant@Vedant-Sharma:~/usr$ ls
share
```

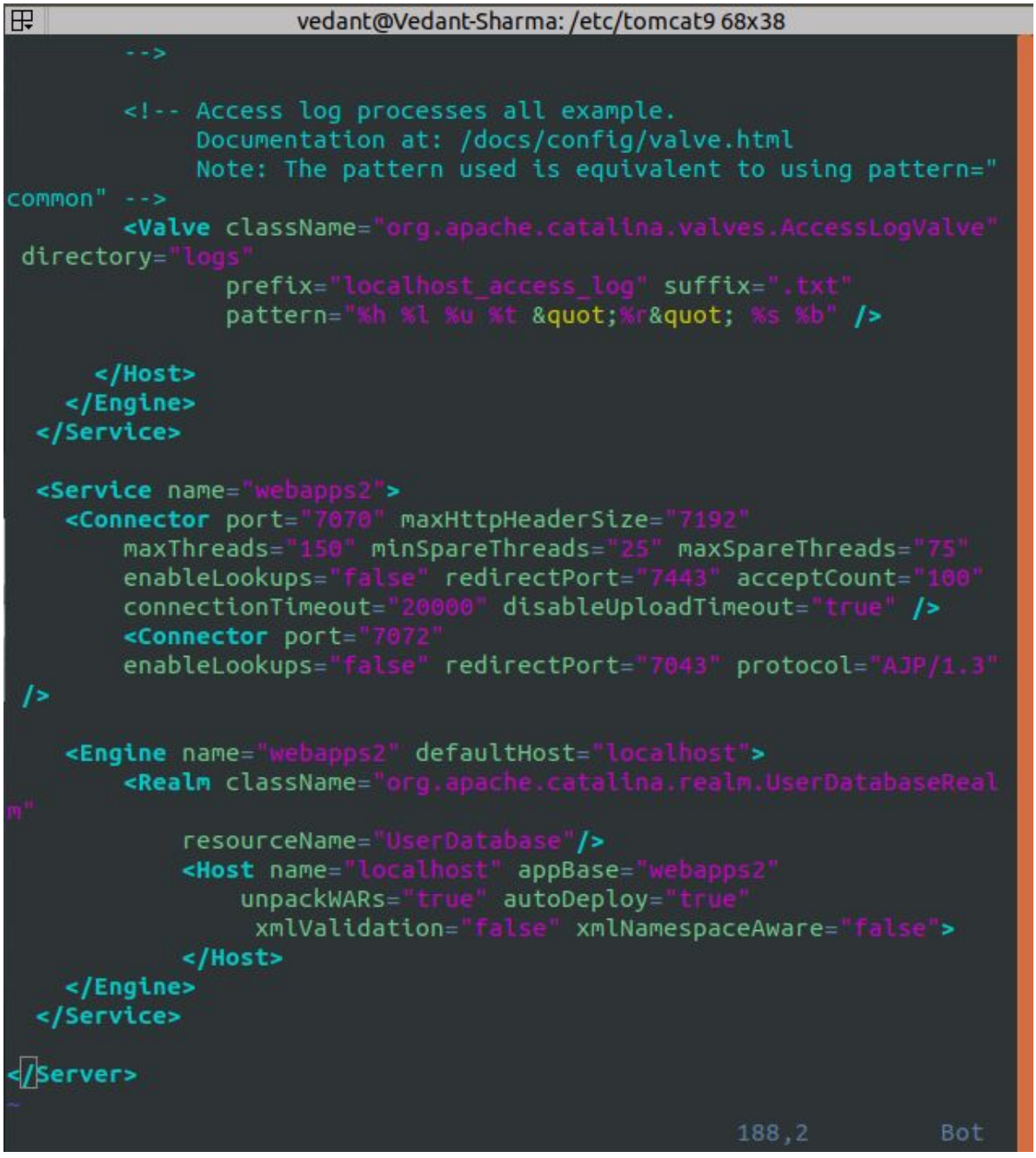
Test.jsp fetches the data from sql database

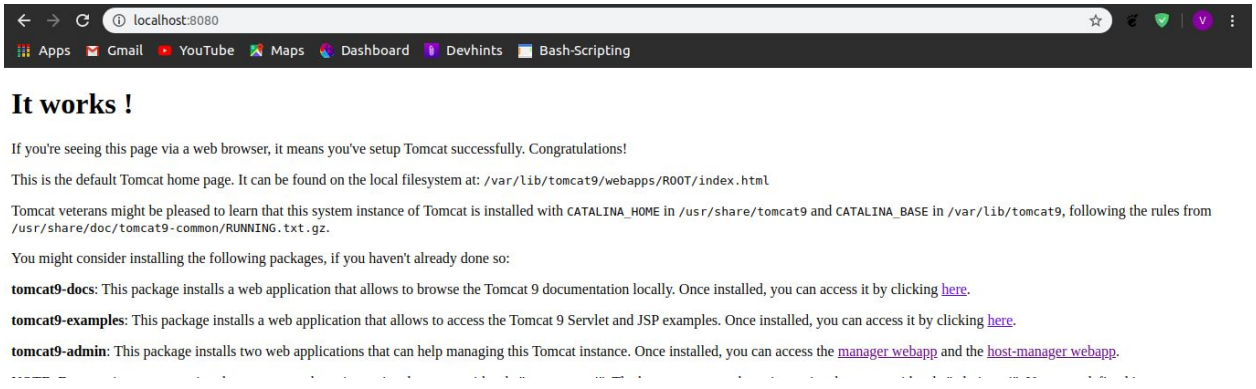


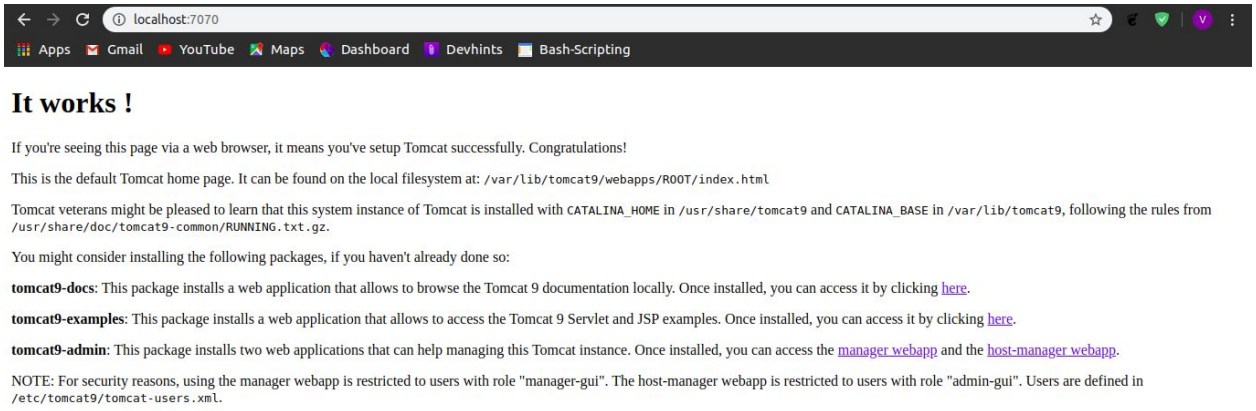
Q5. Run multiple services on different ports with different connectors (AJP/HTTP) on same tomcat installation.

Making a webapps2 in /var/lib/nginx and making it as a new appbase for new service block

```
vedant@Vedant-Sharma:/var/lib/tomcat9$ cd webapps
vedant@Vedant-Sharma:/var/lib/tomcat9/webapps$ cd ..
vedant@Vedant-Sharma:/var/lib/tomcat9$ cd webapps2
vedant@Vedant-Sharma:/var/lib/tomcat9/webapps2$ ls
ROOT  sample  sample.war
vedant@Vedant-Sharma:/var/lib/tomcat9/webapps2$ cd ..
vedant@Vedant-Sharma:/var/lib/tomcat9$ cd webapps
vedant@Vedant-Sharma:/var/lib/tomcat9/webapps$ ls
ROOT  sample  sample1  sample.war
vedant@Vedant-Sharma:/var/lib/tomcat9/webapps$
```


vedant@Vedant-Sharma: /etc/tomcat9 68x38
-->
<!-- Access log processes all example.
Documentation at: /docs/config/valve.html
Note: The pattern used is equivalent to using pattern="common" -->
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
prefix="localhost_access_log" suffix=".txt"
pattern="%h %l %u %t "%r" %s %b" />
</Host>
</Engine>
</Service>
<Service name="webapps2">
<Connector port="7070" maxHttpHeaderSize="7192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" redirectPort="7443" acceptCount="100"
connectionTimeout="20000" disableUploadTimeout="true" />
<Connector port="7072"
enableLookups="false" redirectPort="7043" protocol="AJP/1.3" />
<Engine name="webapps2" defaultHost="localhost">
<Realm className="org.apache.catalina.realm.UserDatabaseRealm"
resourceName="UserDatabase"/>
<Host name="localhost" appBase="webapps2"
unpackWARs="true" autoDeploy="true"
xmlValidation="false" xmlNamespaceAware="false">
</Host>
</Engine>
</Service>
</Server>
~
188,2 Bot





Q6. Use nginx as reverse proxy for tomcat application.

- **Setup self signed certificate on that nginx for bootcamp.com.**
- **What is the difference between proxy_pass & proxy_pass reverse?**

```
vedant@Vedant-Sharma: /etc/nginx/sites-available 74x41
server {
        listen 443 ssl;
        server_name bootcamp.com;
        root /var/www/html;
        index ssl.html;
        ssl on;
        ssl_certificate /etc/nginx/ssl/nginx.crt;
        ssl_certificate_key /etc/nginx/ssl/nginx.key;
        location / {
                proxy_pass http://127.0.0.1:8080/;
        }
}
~
~
~
~
```

```
vedant@Vedant-Sharma:/etc/nginx/ssl$ curl bootcamp.com
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <title>Apache Tomcat</title>
</head>

<body>
<h1>It works !</h1>

<p>If you're seeing this page via a web browser, it means you've setup Tom
cat successfully. Congratulations!</p>

<p>This is the default Tomcat home page. It can be found on the local file
system at: <code>/var/lib/tomcat9/webapps/ROOT/index.html</code></p>

<p>Tomcat veterans might be pleased to learn that this system instance of
Tomcat is installed with <code>CATALINA_HOME</code> in <code>/usr/share/to
mcat9</code> and <code>CATALINA_BASE</code> in <code>/var/lib/tomcat9</cod
e>, following the rules from <code>/usr/share/doc/tomcat9-common/RUNNING.t
xt.gz</code>.</p>

<p>You might consider installing the following packages, if you haven't al
ready done so:</p>

<p><b>tomcat9-docs</b>: This package installs a web application that allow
s to browse the Tomcat 9 documentation locally. Once installed, you can ac
cess it by clicking <a href="docs/">here</a>.</p>

<p><b>tomcat9-examples</b>: This package installs a web application that a
llows to access the Tomcat 9 Servlet and JSP examples. Once installed, you
 can access it by clicking <a href="examples/">here</a>.</p>
```

Difference ;

A forward proxy provides proxy services to a client or a group of clients.At times, these clients belong to a common internal network.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. Depending on the forward proxy's settings, a request can be allowed or denied. If allowed, then the request is forwarded to the firewall and then to the file transfer

server. From the point of view of the file transfer server, it is the proxy server that issued the request, not the client. So when the server responds, it addresses its response to the proxy.

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 it is the reverse proxy that is providing file transfer services. The client is unknown 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.