Application Server

ASSIGNMENT

Name – Maithely Sharma
College – University of Petroleum and Energy Studies
EmployeeID – 4057

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

- **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

- **Describe tomcat directory structure.**

```
maithely@maithely:/opt/tomcat$ tree
.
├── bin [error opening dir]
├── BUILDING.txt
├── conf [error opening dir]
├── CONTRIBUTING.md
├── lib [error opening dir]
├── LICENSE
├── logs [error opening dir]
├── NOTICE
├── README.md
├── RELEASE-NOTES
├── RUNNING.txt
├── temp [error opening dir]
├── webapps [error opening dir]
└── work [error opening dir]

7 directories, 7 files
maithely@maithely:/opt/tomcat$
```

```
.
├── conf.d
├── fastcgi.conf
├── fastcgi_params
├── koi-utf
├── koi-win
├── mime.types
├── modules-available
├── modules-enabled
│   ├── 50-mod-http-geoip.conf -> /usr/share/nginx/modules-available/mod-http-geoip.conf
│   ├── 50-mod-http-image-filter.conf -> /usr/share/nginx/modules-available/mod-http-image-filter.conf
│   ├── 50-mod-http-xslt-filter.conf -> /usr/share/nginx/modules-available/mod-http-xslt-filter.conf
│   ├── 50-mod-mail.conf -> /usr/share/nginx/modules-available/mod-mail.conf
│   └── 50-mod-stream.conf -> /usr/share/nginx/modules-available/mod-stream.conf
├── nginx.conf
├── proxy_params
├── scgi_params
├── sites-available
│   └── default
├── sites-enabled
│   └── default -> /etc/nginx/sites-available/default
├── snippets
│   ├── fastcgi-php.conf
│   └── snakeoil.conf
├── uwsgi_params
└── win-utf
```

The directory structure is as follows:-

jsp

This directory holds all the jsp files for the application, both system jsp files copied from the "src/jsp" directory (),

<u>images</u>

This hold image files, primarily "gif", for particular questions. This directory is structured by package name.

<u>html</u>

This hold pure html files for particular questions. This directory is structured by package name.

<u>WEB-INF</u>

Tomcat hides the contents of this directory from users, and is the location where Java class files are stored as well as the Tomcat "web.xml" file which defines a number of parameters for the application in particular security information and the mapping of user requests, i.e. URIs, to servlets. The contents of this directory are as follows:-

<u>web.xml</u>

This is a key file for running a Tomcat application and defines various features of the application.

<u>classes</u>

This holds the Java classes for the application, both system jsp files compiled from the "src/java" directory ()

<u>help</u>

This holds the help for the application, both system help files compied from the "src/help" directory ()

<u>logs</u>

This holds the log files produced from the execution of a coursework. There are 3 basic log files, usually called *applicationlog.text*, *activitylog.txt*, and *mysqlLog.txt*.

<u>lib</u>

This contains jar files needed by Tomcat to run the @systemname; application. These are copied from "src/lib" on a coursework build.

<u>images-xml</u>

This hold Xml files that define an image to be rendered by the "synDrawings" package: these images can be modified at run-time in response to user input. This directory is structured by package name.

initParameters

This holds files holding name-value pairs for use by jsp and java packages so that the ExerTran application can be varied without editing the program sources

xml

This holds Xml files with definitions of the database table columns. Changes to these files changes the structure of the database table on the next database write

files

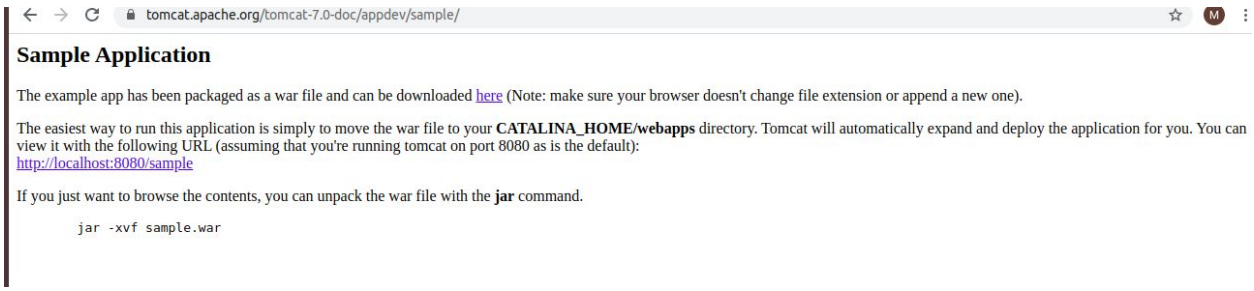
This is available for holding save files from the application, particularly csv files created from database tables.

src/java/dynDrawings

Objects that handle the dynDrawing system for modifying diagrams on the fly in respnse to user actions.

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

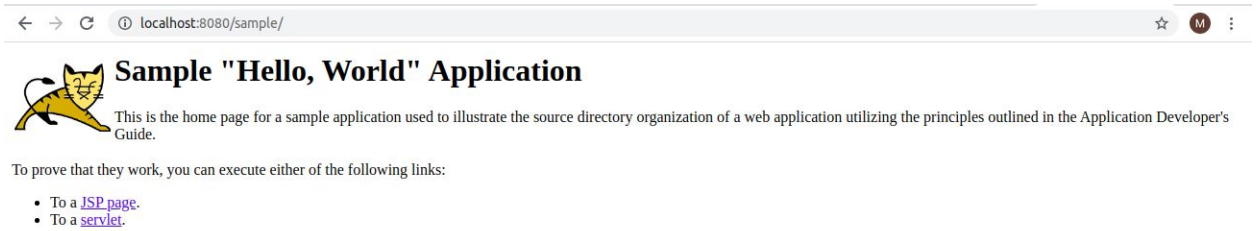
Download sample.war file



Copy this file in /opt/tomcat/webapps

```
maithely@maithely:/opt/tomcat$ sudo su
root@maithely:/opt/tomcat# cd webapps/
root@maithely:/opt/tomcat/webapps# ls
docs  examples  host-manager  manager  ROOT  sample  sample.war
root@maithely:/opt/tomcat/webapps#
```

Now open your browser and open sample file



TO RUN USING SQL:

Create a user and grant privileges to it, then create a database and a table along with it

```
mysql> CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'Ttn@1234';
Query OK, 0 rows affected (0.01 sec)

mysql> grant all privileges on *.* to 'newuser'@'localhost';
Query OK, 0 rows affected (0.00 sec)

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

mysql> use test;
Database changed
mysql> create table testdata ( id int not null auto_increment primary key, foo varchar(25), bar int );
Query OK, 0 rows affected (0.03 sec)

mysql>
```

```
mysql> insert into testdata values( null ,'hello',12345);
Query OK, 1 row affected (0.02 sec)

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

mysql>
```

Then add these lines in context.xml

```
maithely@maithely:/opt/tomcat$ sudo su
[sudo] password for maithely:
root@maithely:/opt/tomcat# cd conf/
root@maithely:/opt/tomcat/conf# ls
Catalina          catalina.properties  jaspic-providers.xml
catalina.policy   context.xml          jaspic-providers.xsd
```

```
    <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.DataSource"
               maxActive="100" maxIdle="30" maxWait="10000"
               username="newuser" password="Ttn@1234" driverClassName="com.mysql.jdbc.Driver"
               url="jdbc:mysql://localhost:3306/test"/>
</Context>
"context.xml" 35L, 1684C
```

Now create web.xml

```
root@maithely:/opt/tomcat/webapps# mkdir samplenew
root@maithely:/opt/tomcat/webapps# cd samplenew/
root@maithely:/opt/tomcat/webapps/samplenew# mkdir WEB-INF
root@maithely:/opt/tomcat/webapps/samplenew# cd WEB-INF/
root@maithely:/opt/tomcat/webapps/samplenew/WEB-INF# sudo vim web.xml
```

Add these lines to 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>
```

Now create a file test.jsp in samplenew directory

```
root@maithely:/opt/tomcat/webapps/samplenew/WEB-INF# cd ..
root@maithely:/opt/tomcat/webapps/samplenew# sudo vim test.jsp
```

Add these lines to 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>
```

Now create a directory lib and download jdbc connector and also download standard.jar file

```
root@maithely:/opt/tomcat/webapps/samplenew/WEB-INF# mkdir lib
root@maithely:/opt/tomcat/webapps/samplenew/WEB-INF# cd lib/
root@maithely:/opt/tomcat/webapps/samplenew/WEB-INF/lib# wget http://www.java2s.com/Code/JarDownload/standard/standard.jar.zip
--2020-02-26 17:39:23--  http://www.java2s.com/Code/JarDownload/standard/standard.jar.zip
Resolving www.java2s.com (www.java2s.com)... 52.216.107.83
Connecting to www.java2s.com (www.java2s.com)|52.216.107.83|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 366241 (358K) [application/zip]
Saving to: ‘standard.jar.zip’

standard.jar.zip                    100%[===================================================================>] 357.66K   233KB/s    in 1.5s
```

Copy the link address from below and wget it

Download standard.jar

- 💾 standard/standard.jar.zip(366 k)

The download jar file contains the following class files or Java source files.

Now download one more file

```
root@maithely:/opt/tomcat/webapps/samplenew/WEB-INF/lib# wget http://www.java2s.com/Code/JarDownload/jstl/jstl.jar.zip
--2020-02-26 17:39:55--  http://www.java2s.com/Code/JarDownload/jstl/jstl.jar.zip
Resolving www.java2s.com (www.java2s.com)... 52.216.147.107
Connecting to www.java2s.com (www.java2s.com)|52.216.147.107|: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  7

2020-02-26 17:39:56 (73.3 KB/s) - ‘jstl.jar.zip’ saved [20047/20047]
```

Copy link address from here of jstl.jar

Jar File Download / j / jstl /

Download jstl.jar

jstl/jstl.jar.zip(20 k)

The download jar file contains the following class files or Java source files.

Unzip the files

```
root@maithely:/opt/tomcat/webapps/samplenew/WEB-INF/lib# ls
jstl.jar.zip  standard.jar.zip
root@maithely:/opt/tomcat/webapps/samplenew/WEB-INF/lib# unzip jstl.jar.zip
Archive:  jstl.jar.zip
  inflating: jstl.jar
root@maithely:/opt/tomcat/webapps/samplenew/WEB-INF/lib# unzip standard.jar.zip
Archive:  standard.jar.zip
  inflating: standard.jar
root@maithely:/opt/tomcat/webapps/samplenew/WEB-INF/lib# ls
jstl.jar  jstl.jar.zip  standard.jar  standard.jar.zip
```

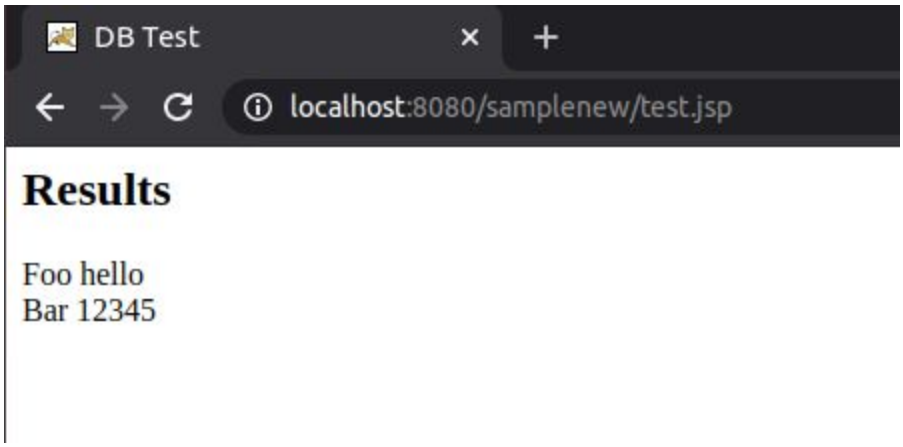
Now move jdbc connector

```
root@maithely:/opt/tomcat/webapps/samplenew/WEB-INF/lib# sudo mv /home/maithely/Downloads/u
ubuntu@3.92.209.206   ubuntu@52.91.106.94   ubuntu@52.91.213.125  usr/
root@maithely:/opt/tomcat/webapps/samplenew/WEB-INF/lib# sudo mv /home/maithely/Downloads/usr/share/java/mysql-connector-java-8.0.19.jar
root@maithely:/opt/tomcat/webapps/samplenew/WEB-INF/lib# ls
jstl.jar  jstl.jar.zip  mysql-connector-java-8.0.19.jar  standard.jar  standard.jar.zip
```

Start tomcat service

```
maithely@maithely:~$ sudo service tomcat restart
[sudo] password for maithely:
maithely@maithely:~$
```

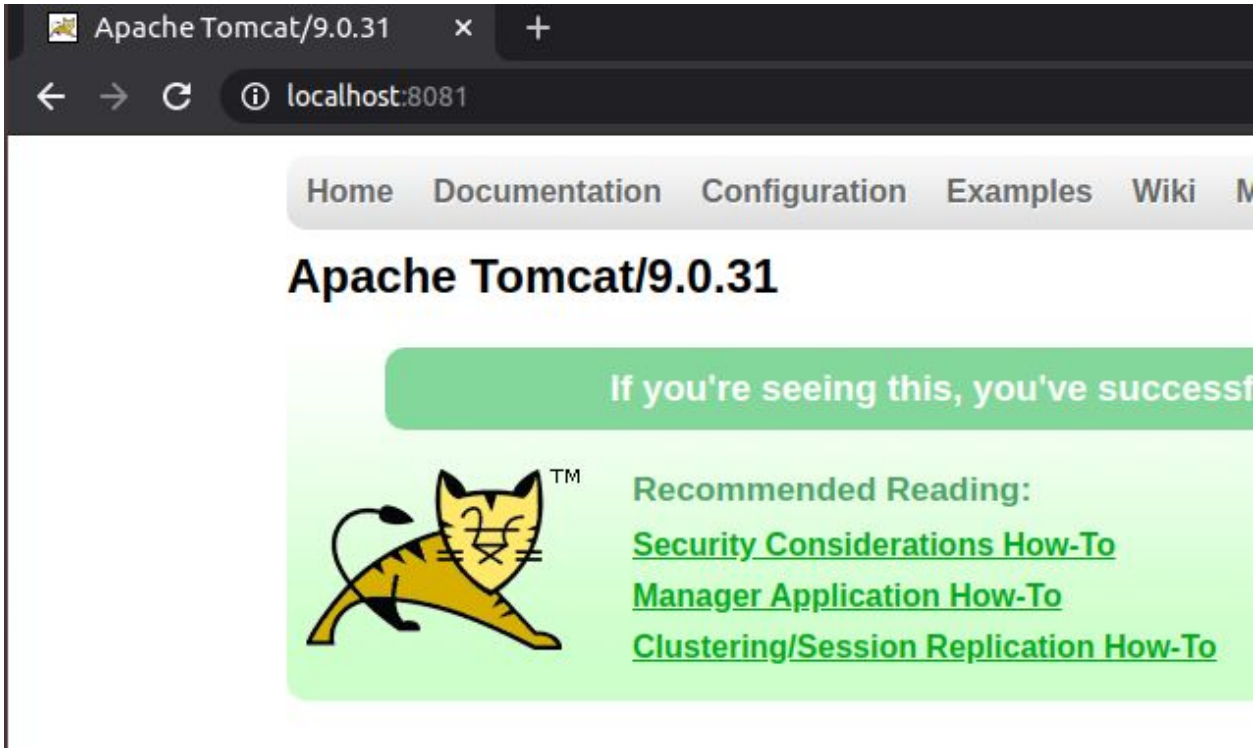
Open the browser and open test.jsp at localhost

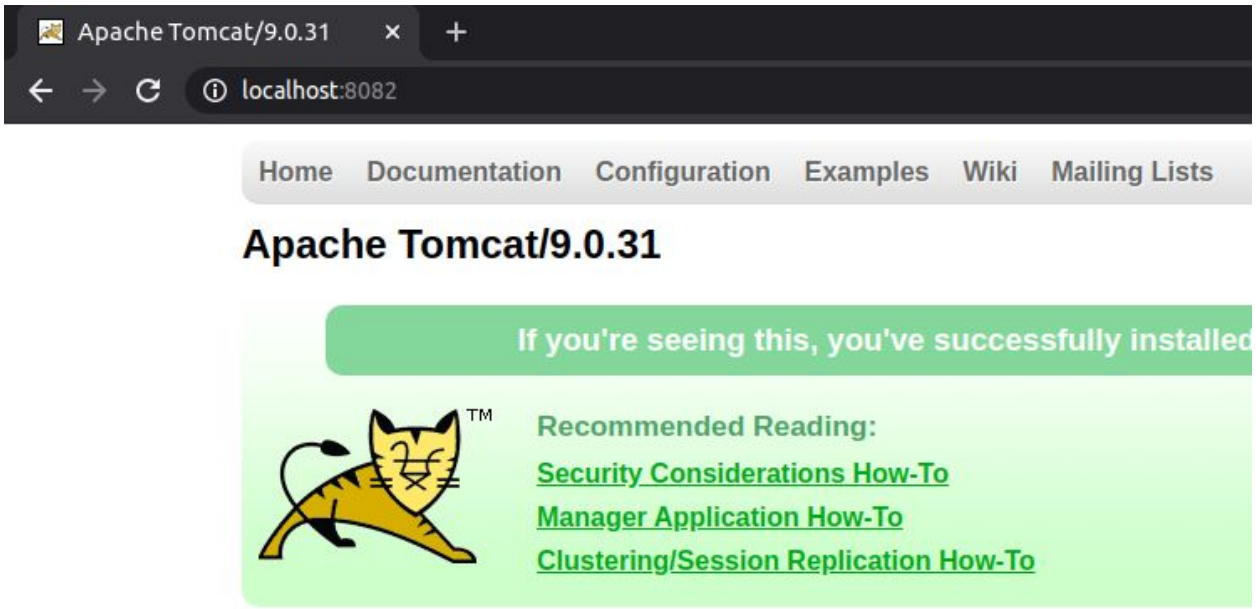


- **Run multiple services on different ports with different connectors (AJP/HTTP) on same tomcat installation.**

Add the service 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="webapps"
        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="webapps"
        unpackWARs="true" autoDeploy="true">
      </Host>
   </Engine>
</Service>
</Server>
"server.xml" 189L, 8285C
```


Apache Tomcat/9.0.31
localhost:8081
Home
Documentation
Configuration
Examples
Wiki
Apache Tomcat/9.0.31
If you're seeing this, you've successf
Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To


Apache Tomcat/9.0.31
localhost:8082
Home
Documentation
Configuration
Examples
Wiki
Mailing Lists
Apache Tomcat/9.0.31
If you're seeing this, you've successfully installed
Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To

For HTTP:

```
    -->
    <Connector port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
    <!-- A "Connector" using the shared thread pool-->
    <!--
    <Connector executor="tomcatThreadPool"
```

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

```
maithely@maithely:/etc/nginx$ cd ssl/
maithely@maithely:/etc/nginx/ssl$ ls
private.key  public.pem
maithely@maithely:/etc/nginx/ssl$
```

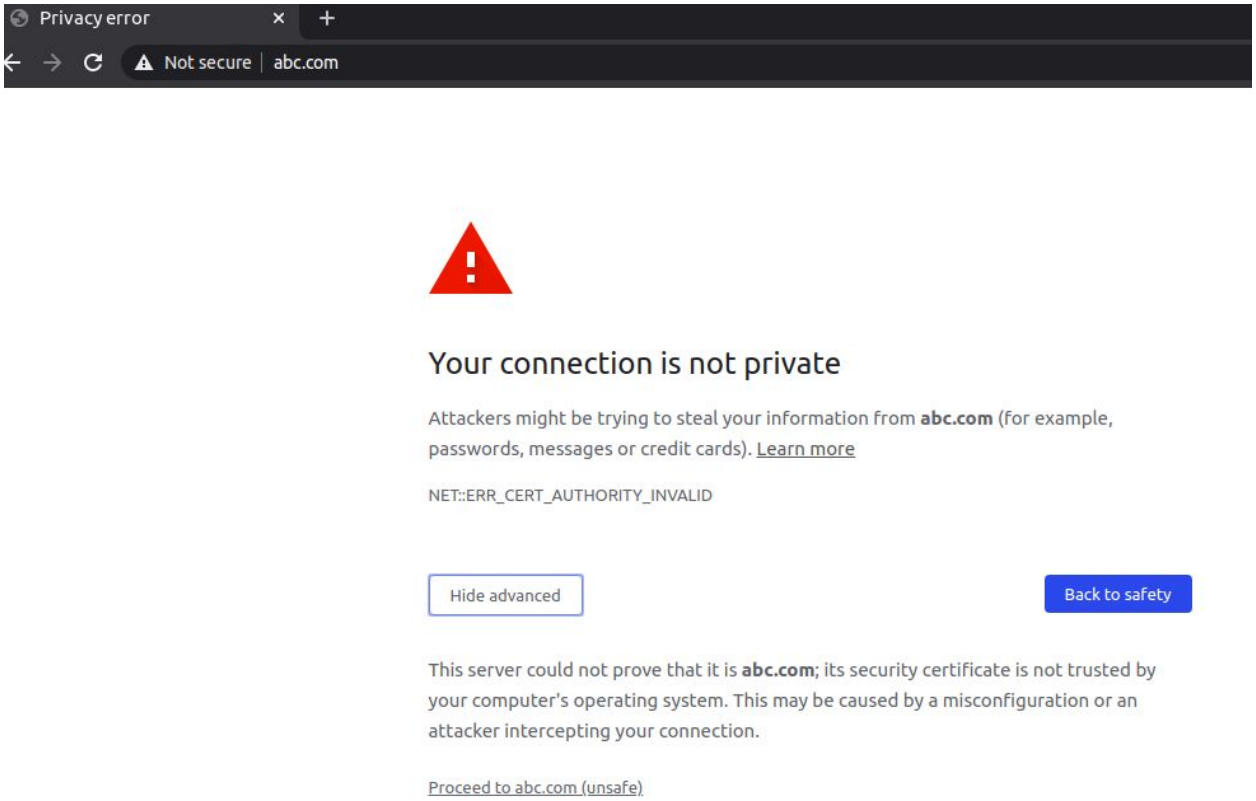
```
server {
listen 80;
server_name _ ;
return 302 https://www.abc.com;
}
#upstream backend  {
#  server 127.0.0.1:81;
#  server 127.0.0.1:82;
# server 127.0.0.1:83;
#  server 127.0.0.1:84;
#  server 127.0.0.1:85;
#}

  server {

        listen 443 ssl;
        server_name www.abc.com;
        #error_page 404 error.html;
        ssl_certificate /etc/nginx/ssl/public.pem;
        ssl_certificate_key /etc/nginx/ssl/private.key;
#       allow 10.1.211.99;
#       deny all;
        error_page 403 404 /failpage.html;

#location /failpage.html{
        #return 405 "<h1>ERROR 405</h1>";
        #proxy_pass http://weevil.info/;
        #return 302 http://weevil.info/;
        location /
{               proxy_pass http://localhost:8080/;


    }
}
~
~
~
"abc" 34L, 675C
```


Privacy error
Not secure
abc.com
Your connection is not private
Attackers might be trying to steal your information from abc.com (for example, passwords, messages or credit cards). Learn more
NET::ERR_CERT_AUTHORITY_INVALID
Hide advanced
Back to safety
This server could not prove that it is abc.com; its security certificate is not trusted by your computer's operating system. This may be caused by a misconfiguration or an attacker intercepting your connection.
Proceed to abc.com (unsafe)

Apache Tomcat/9.0.31

Not secure | abc.com

Home Documentation Configuration Examples Wiki Mailing Lists

Apache Tomcat/9.0.31

If you're seeing this, you've successfully installed Tomcat. Congratu

Recommended Reading:

Security Considerations How-To

Manager Application How-To

Clustering/Session Replication How-To

Developer Quick Start

Tomcat Setup | Realms & AAA | Examples | Serv

First Web Application | JDBC DataSources | Tom

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 9.0 access to the manager

Documentation

Tomcat 9.0 Documentation

Tomcat 9.0 Configuration

Tomcat Wiki

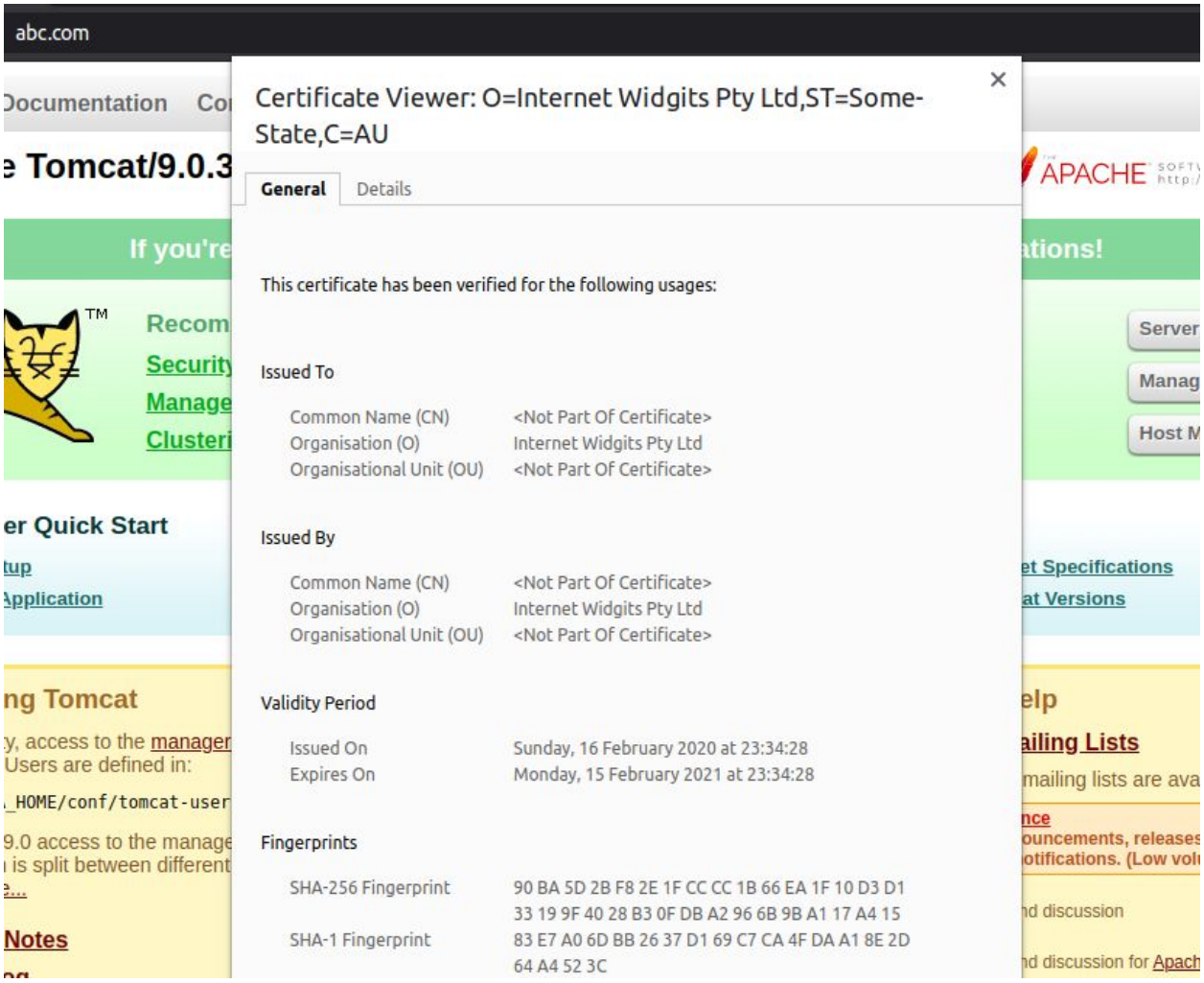
Getting H

FAQ and M

The following

tomcat-anno

Important an



- **What is the difference between proxy_pass & proxy_pass reverse?**

Proxy	Reverse Proxy
Sits between our clients and the internet	Sits between internet traffic and our servers
Intermediate layer often used within organizations to monitor web traffic.	Intermediate layer often used to load balance traffic & serve content from a cache.