**Application Server**

**ASSIGNMENT**

****

Name – Maithely Sharma

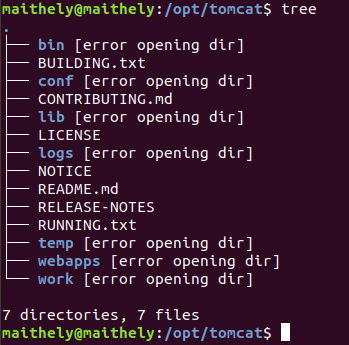
College – University of Petroleum and Energy Studies

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.**

****

****

**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 (),**

**images**

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

**html**

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

**WEB-INF**

**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:-**

**web.xml**

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

**classes**

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

**help**

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

**logs**

**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*.**

**lib**

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

**images-xml**

**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**

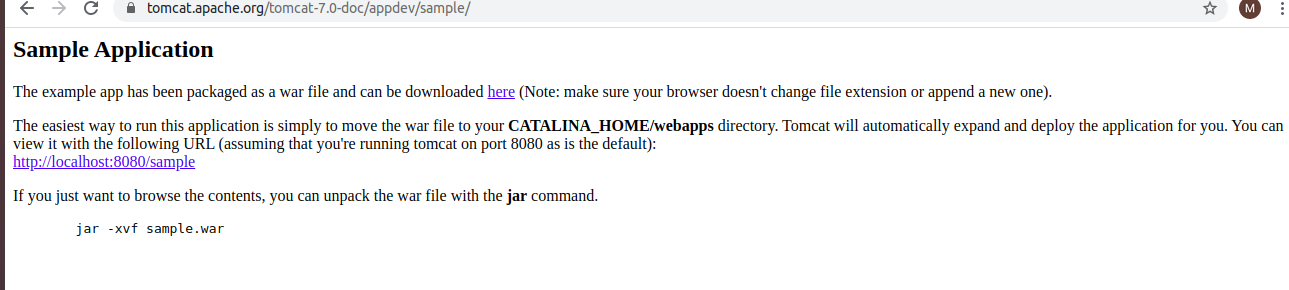
**This is available for holding save files from the application, particularly csv files created from database tables.**

**src/java/dynDrawings**

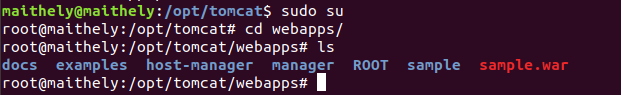
**Objects that handle the dynDrawing system for modifying diagrams on the fly in respnse to user actions.**

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

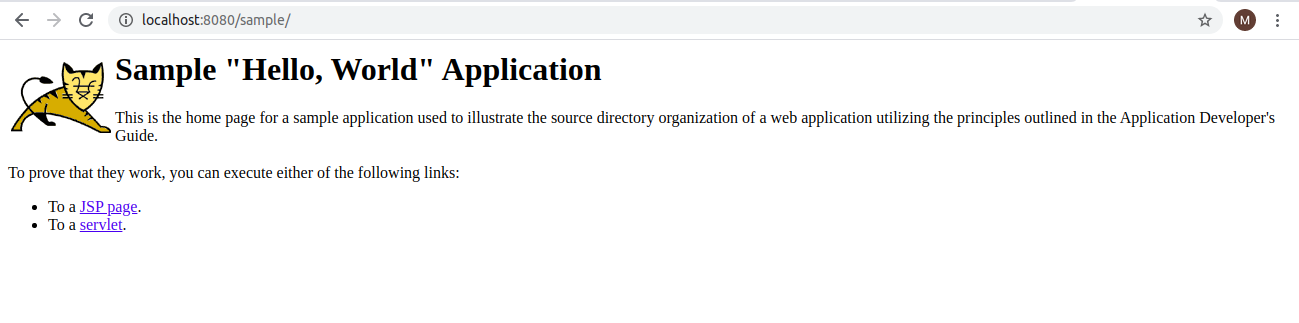
**Download sample.war file**

****

**Copy this file in /opt/tomcat/webapps**

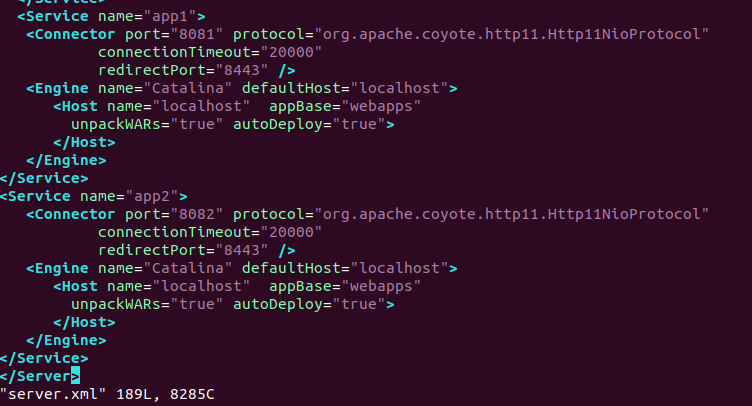
****

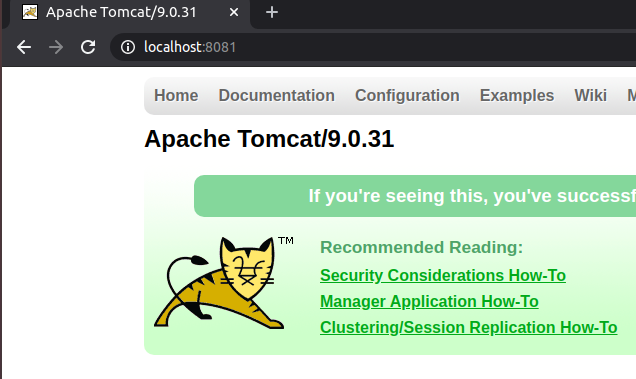
**Now open your browser and open sample file**

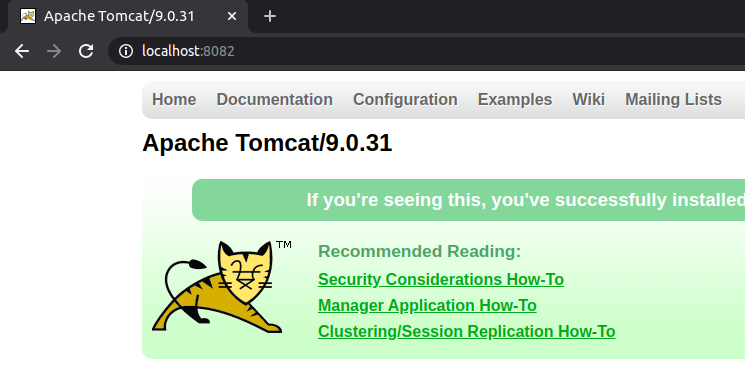
****

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

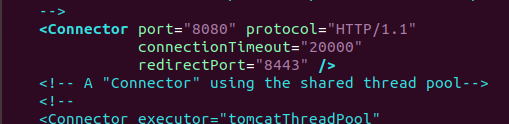
**Add the service in server.xml**

****

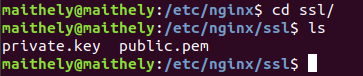
****

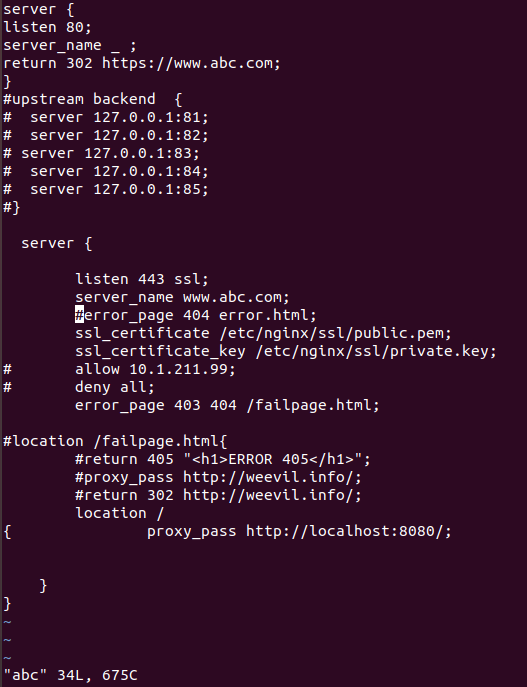
****

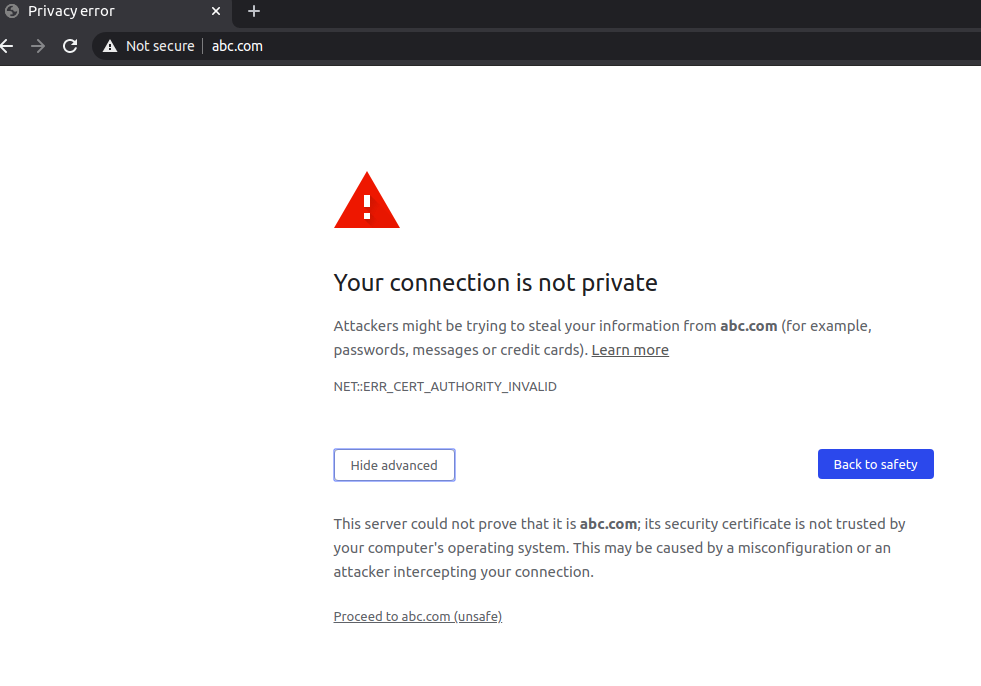
**For HTTP:**

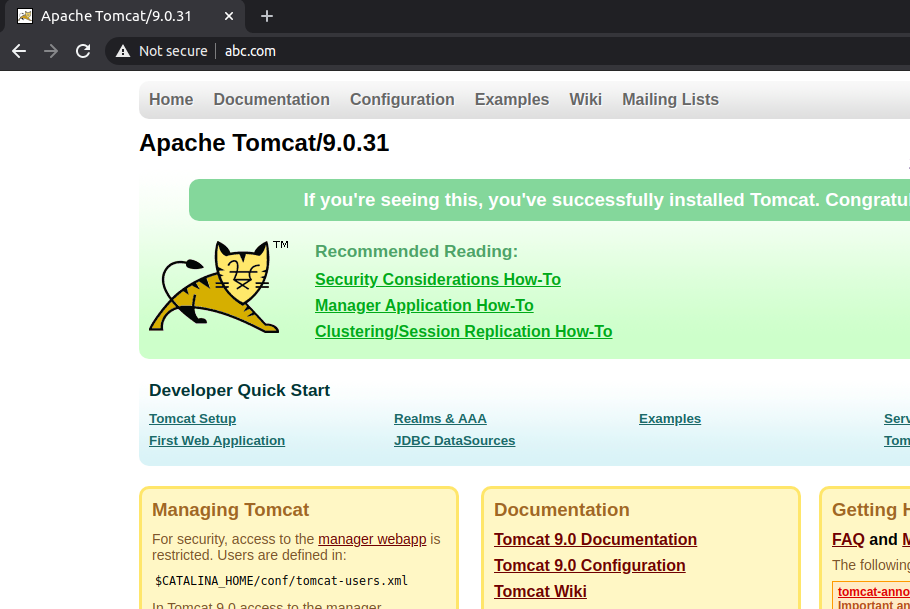
****

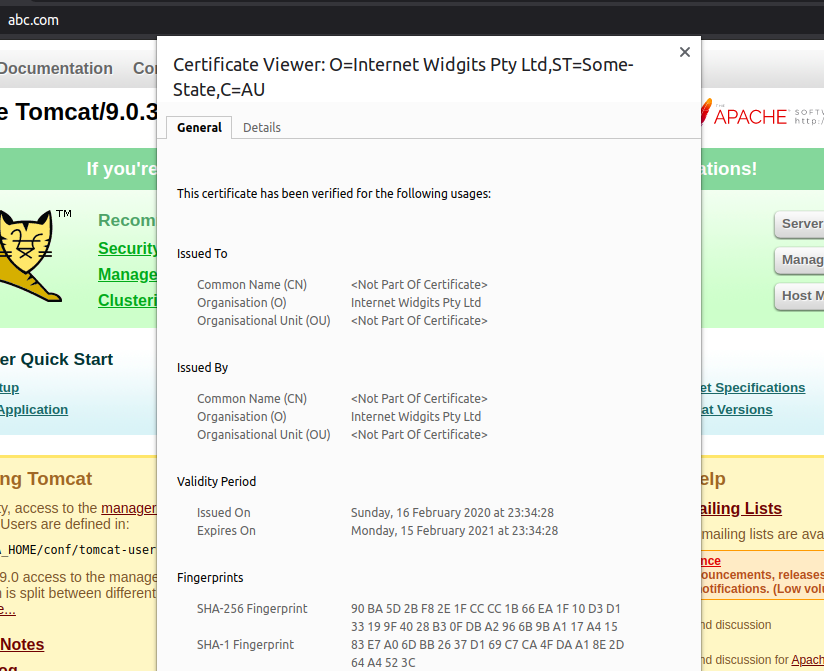
* **Use nginx as reverse proxy for tomcat application.** 
  + **Setup self signed certificate on that nginx for bootcamp.com.**

****

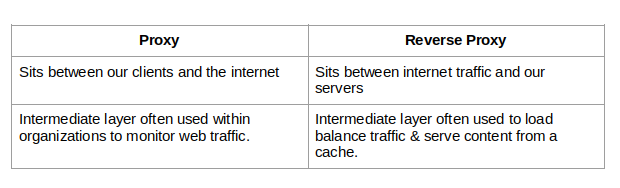
****

****

****

****

* + **What is the difference between proxy\_pass & proxy\_pass reverse?**

****