**Assessment 8 – Application Server**

**Trainee Name : Gargi Sharma**

**Mentor Name : Mr. Ravi Kumar**

**College Name : UPES**

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

>Web server is useful or fitted for static content, whereas application server is fitted for dynamic content.

>In web servers, multithreading is not supported, while in application server, multithreading is supported.

>Application Server supports distributed transaction and EJB. While Web Server only supports Servlets and JSP.

>Application Server can contain web server in them. most of App server e.g. JBoss or was has Servlet and JSP container.

>Though its not limited to Application Server but they used to provide services like Connection pooling, Transaction management, messaging, clustering, load balancing and persistence. Now Apache tomcat also provides connection pooling.

>In terms of logical difference between web server and application server. web server is supposed to provide http protocol level service while application server provides support to web service and expose business level service e.g. EJB.

>Application server are more heavy than web server in terms of resource utilization.

1. What is Catalina?

**Catalina** in 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.

1. Describe tomcat directory structure.

The webapp directory is held in the coursework directory within the "courses" directory and is named after the coursework. If the coursework's name is D2, the coursework directory is "courses/D2Home" in the ExerTran installation directory, and the webapp directory is named "courses/D2Home/D2". 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 (), but also question related jsp files from packages imported into the coursework or generated by the coursework lecturer (the chapter called Topic Packages). The directory structure of this directory is the same as that of the "src/jsp" directory, except that the "jsp/user/" directory also hold jsp files from the packages: with directories structured by package name.

**images**

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

**html**

This holds 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. This file is created automatically during a coursework build from a template file, "web.tmpl", held in "src/templates", with substitution of parameters defined in the coursework definition file (the chapter called Coursework Definition File. The contents of this file are described in .

**classes**

This holds the Java classes for the application, both system jsp files compiled from the "src/java" directory (), but also from the question related java files from packages imported into the coursework or generated by the coursework lecturer (the chapter called Topic Packages). The directory structure of this directory is the same as that of the "src/java" directory, except that the "java/user/" directory also hold java files from the packages: with directories structured by package name.

**help**

This holds the help for the application, both system help files compied from the "src/help" directory (), but also from the question related help files from packages imported into the coursework or generated by the coursework lecturer (the chapter called Topic Packages). The structure of the help files is by question package, with the system help files in the directory "system".

**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. The first of these logs debug messages from the coursework; the second logs user activity, particularly question attempts and marks awarded; the third error messages from the mysql database, which is the default database. When the coursework is re-started, these log files are compressed when they are larger then a set limit, and versions created. It may be necessary to change the compression to occur while the system is running, if regular closure of the application does not occur. A further log file is also created when the connection to the marks database fails and marks cannot be stored in the database. The marks are then recorded in a log file, from which they are read stored to the database when the coursework system is restarted: on successful storage of these marks, the log file is removed.

**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.

**initParameter**

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**

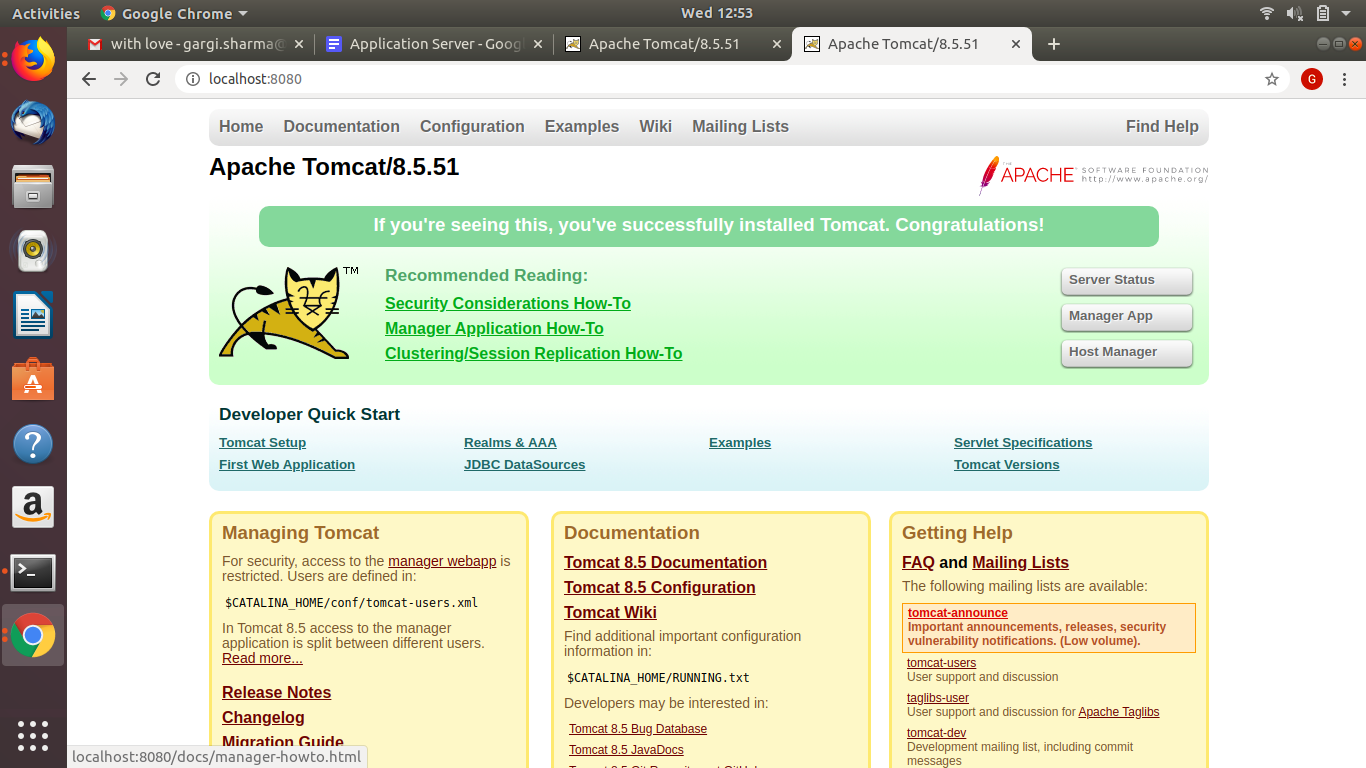
This holds Xml files with definitions of the database table columns. Changes to these files changes the structure of the database table on the next database write: note that this has only been tested for adding extra columns to a table not for removing columns. Note that the only database table that needs creating to start running a coursework is a "users" table, defined by "userd\_def.xml" in the directory, with a manager user and password, all other tables can be created from the manager interface by saving to the database, even if there are no rows to save.

**files**

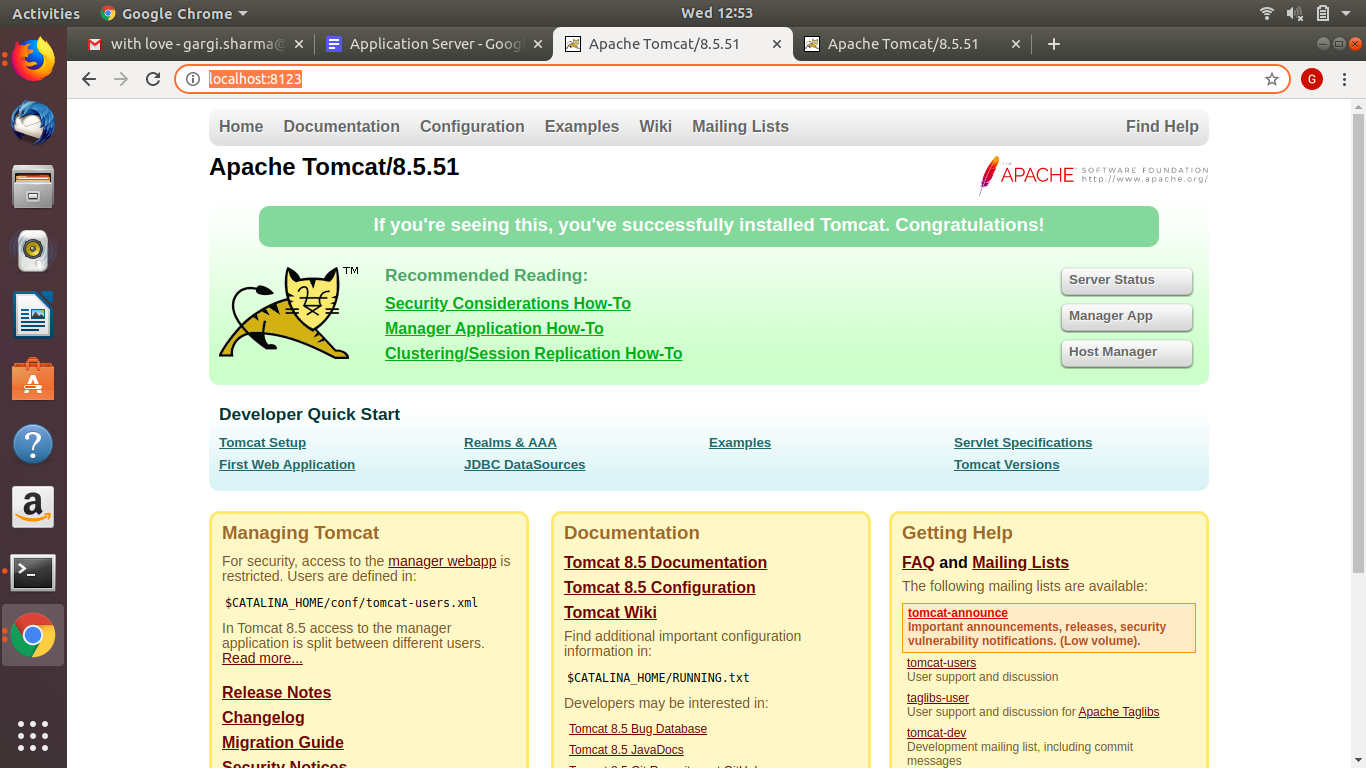
This is available for holding save files from the application, particularly csv files created from database tables.

1. Connect any sample.war to MySQL running on localhost.
2. Run multiple services on different ports with different connectors (AJP/HTTP) on same tomcat installation.

At localhost:8080



At localhost:8123

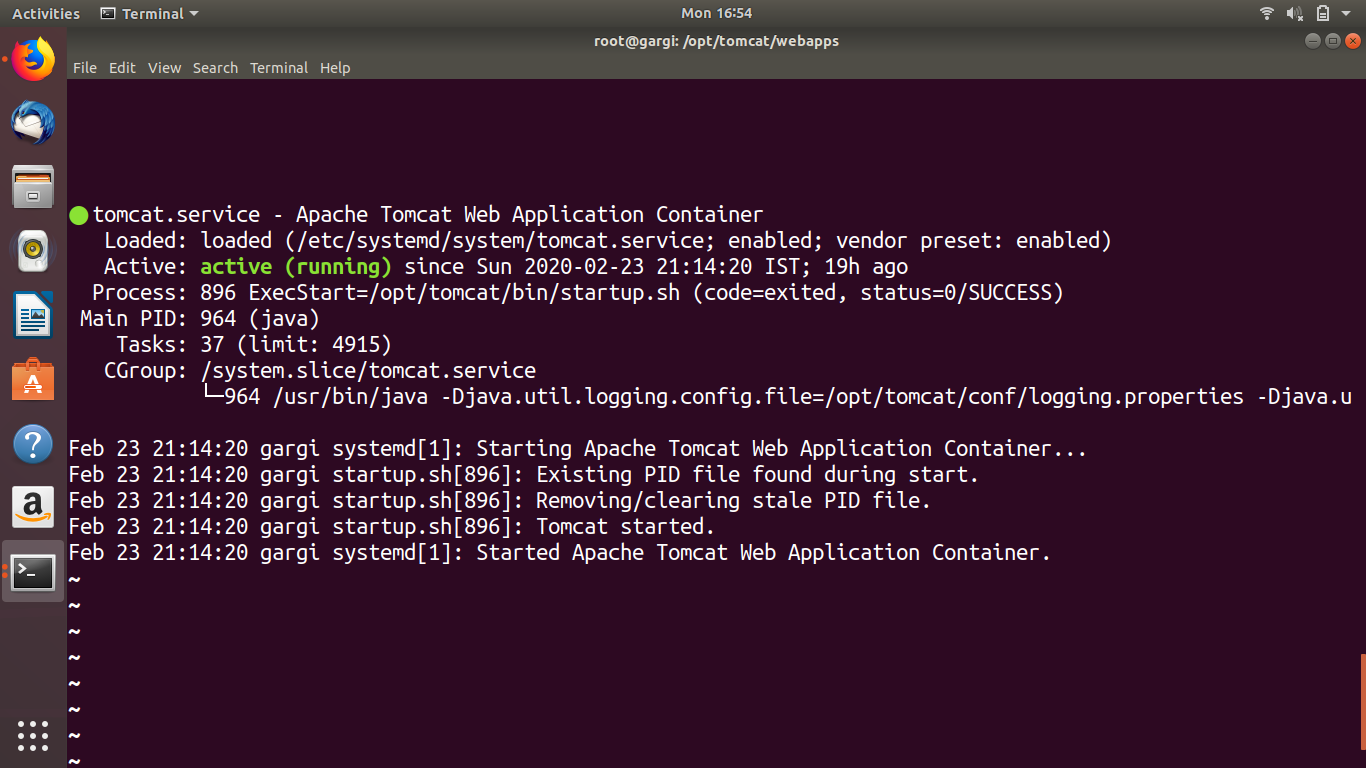


1. How to change default port in tomcat?

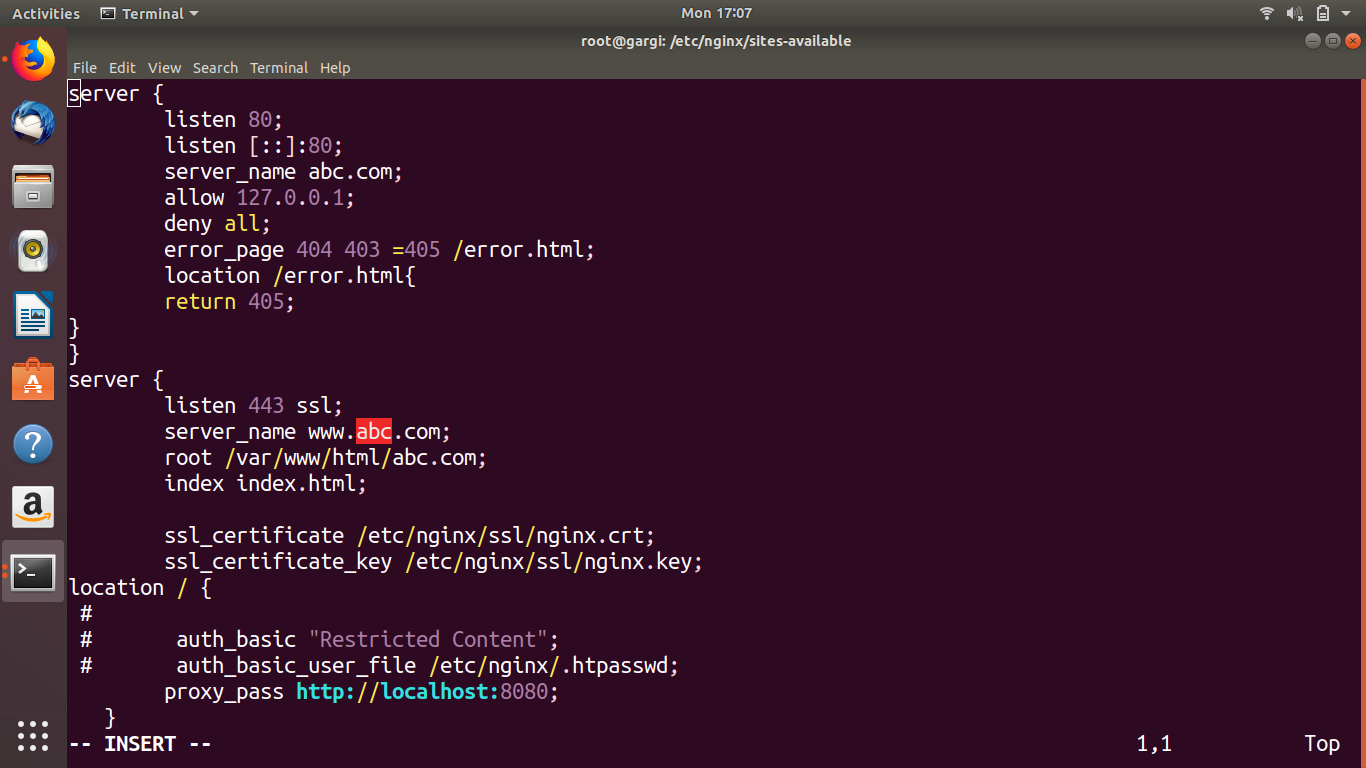
By making changes in server.xml in conf folder.

1. Install tomcat and Use nginx as reverse proxy for tomcat application.

Installed:



Make the appropriate changes in nginx.conf to apply proxy pass:



Now if open abc.com(hosted on nginx), tomcat will serve us instead of nginx:

