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

A Web server exclusively handles HTTP requests, whereas an application server serves business logic to application programs through any number of protocols.

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

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

3. 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.

4. 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.

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

* **Write down name of Application Servers and Web Servers.**

**Web Servers:**

Nginx

Nginx is a web server which can also be used as a reverse proxy, load balancer, mail proxy and HTTP cache.

## Apache Tomcat

## Apache Tomcat is an open source Java servlet container that functions as a web server. A Java servlet is a Java program that extends the capabilities of a server. Although servlets can respond to any types of requests, they most commonly implement applications hosted on Web servers.

## Apache HTTP Server

## The Apache HTTP Server—often referred to as httpd, or simply Apache.

**Java and JavaEE Application Servers:**

## Apache Tomcat

Apache Tomcat is an open source software implementation of the Java Servlet and JavaServer Pages technologies and was developed in an open and participatory environment and released under the Apache License version 2 and developed by the Apache Software Foundation (ASF).

## Glassfish

GlassFish is an open-source application server project originally started by Sun Microsystems for the Java EE platform and now part of the Oracle Corporation. It is available under a dual-license: the Common Development and Distribution License (CDDL) and the GNU General Public License (GPL) with the classpath exception.

## Wildfly

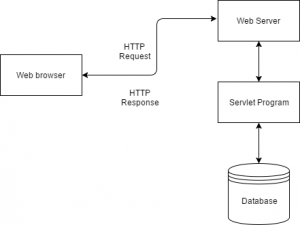
WildFly, was better known as JBoss AS is an application server created by JBoss, but now under continuous development by Red Hat. WildFly is written in Java, and implements the Java Platform, Enterprise Edition (Java EE) specification separately/standalone from the JBoss Enterprise Application Platform.

## Jetty

Jetty is a Java based HTTP (Web) server and Java Servlet container and is developed as a free and open source project as part of the Eclipse Foundation (originally it was developed as an independent open source project).

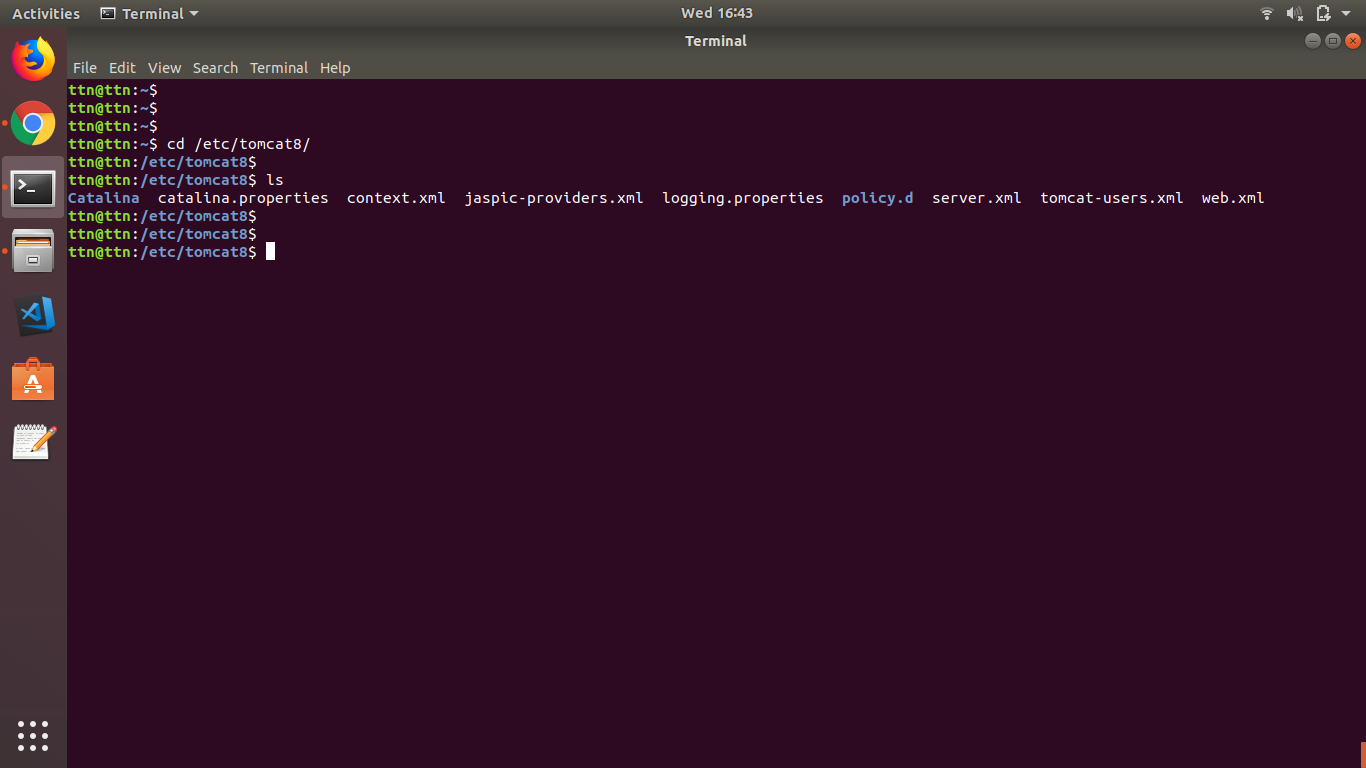
* **What are java servlet?**

A Java servlet is a [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) [software component](https://en.wikipedia.org/wiki/Software_component) that extends the capabilities of a [server](https://en.wikipedia.org/wiki/Server_(computing)). Although servlets can respond to any types of requests, they most commonly implement [web containers](https://en.wikipedia.org/wiki/Web_container) for hosting [web applications](https://en.wikipedia.org/wiki/Web_application) on [web servers](https://en.wikipedia.org/wiki/Web_server) and thus qualify as a server-side servlet [web API](https://en.wikipedia.org/wiki/Web_API). Such web servlets are the [Java](https://en.wikipedia.org/wiki/Java_(software_platform)) counterpart to other [dynamic web content](https://en.wikipedia.org/wiki/Dynamic_web_page) technologies such as [PHP](https://en.wikipedia.org/wiki/PHP) and [ASP.NET](https://en.wikipedia.org/wiki/ASP.NET).



**Properties of Servlets :**

* Servlets work on the server-side.
* Servlets capable of handling complex request obtained from web server.
* **Describe tomcat directory structure.**

****

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. 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*](http://www0.cs.ucl.ac.uk/staff/P.Rounce/myhtml/elecTran_technical_manual/c1601.html). 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 ()ies structured by package name.

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: see [the chapter called *initParameters: Initialisation parameters*](http://www0.cs.ucl.ac.uk/staff/P.Rounce/myhtml/elecTran_technical_manual/c1848.html) and .

xml

This holds Xml files with definitions of the database table columns.

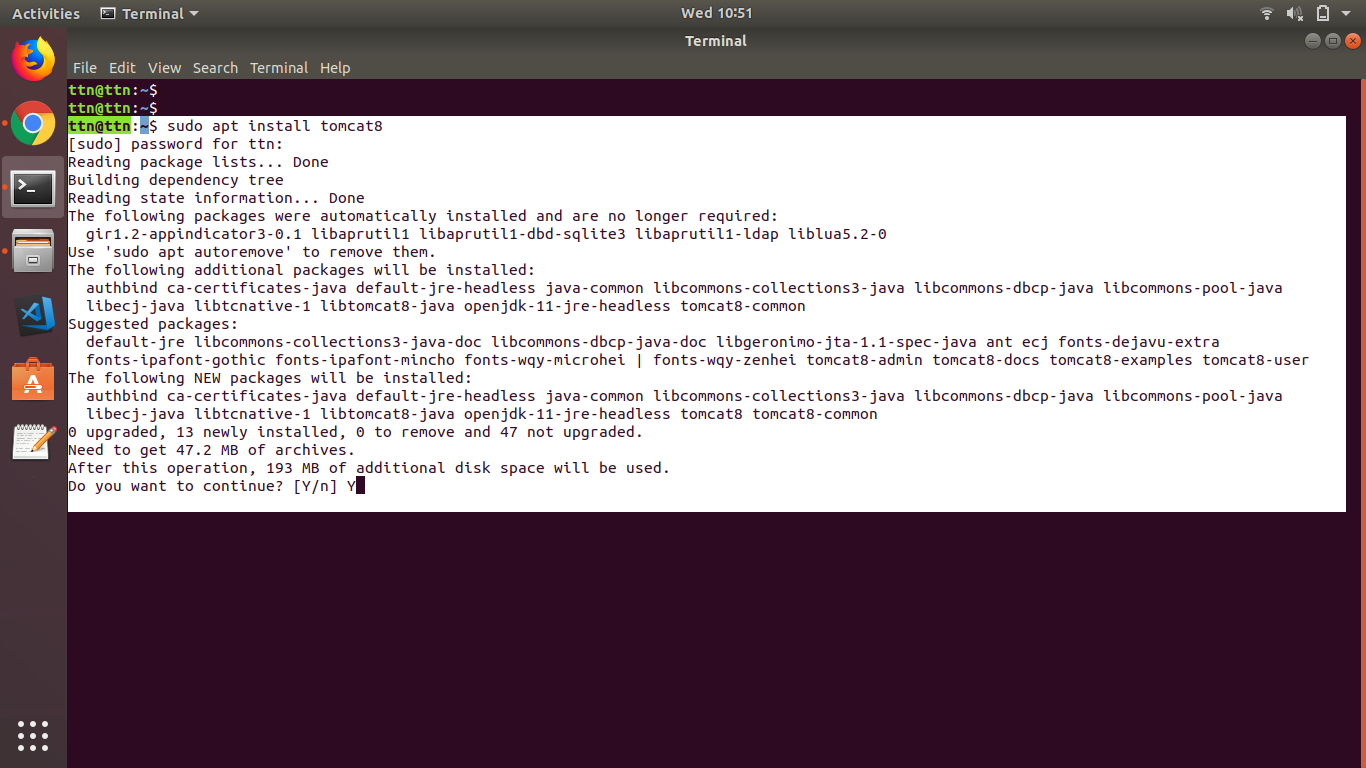
files

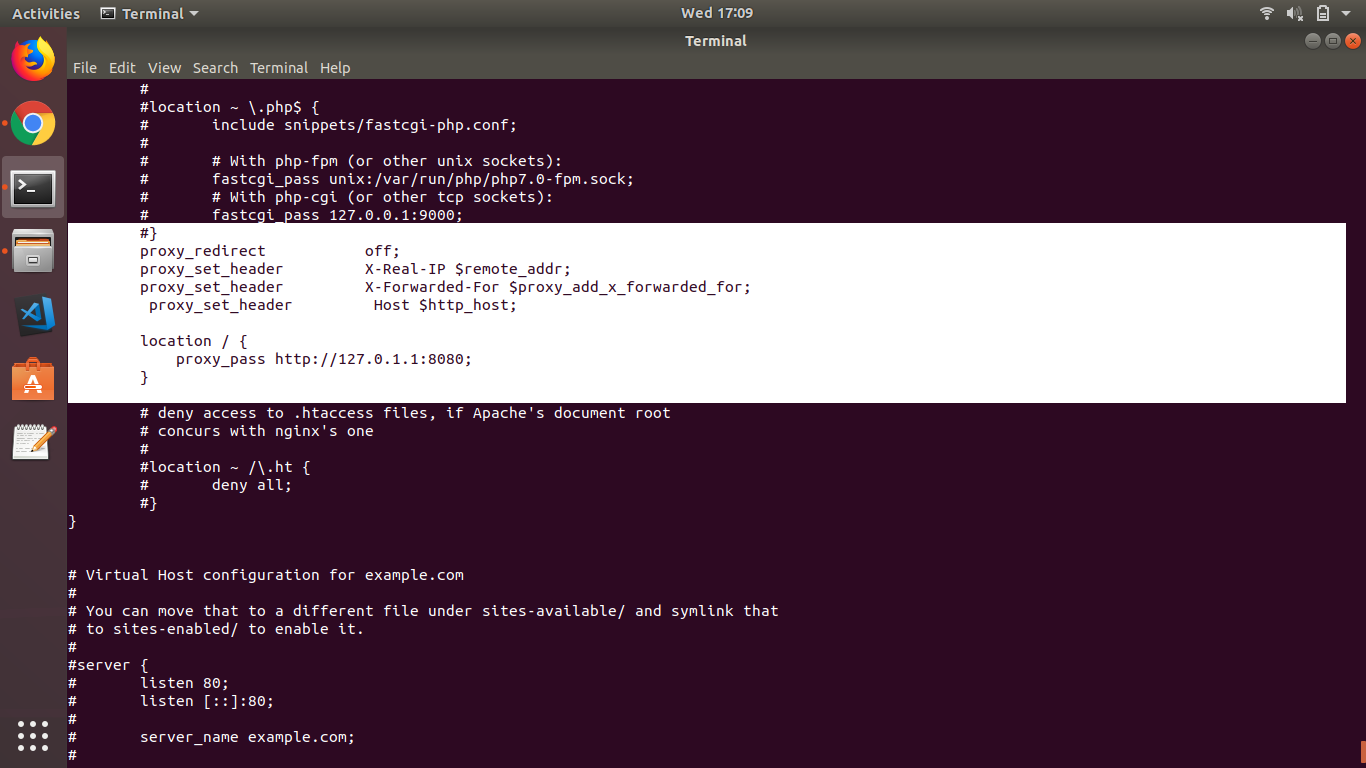
This is available for holding save files from the application, particularly csv files created from database tables.

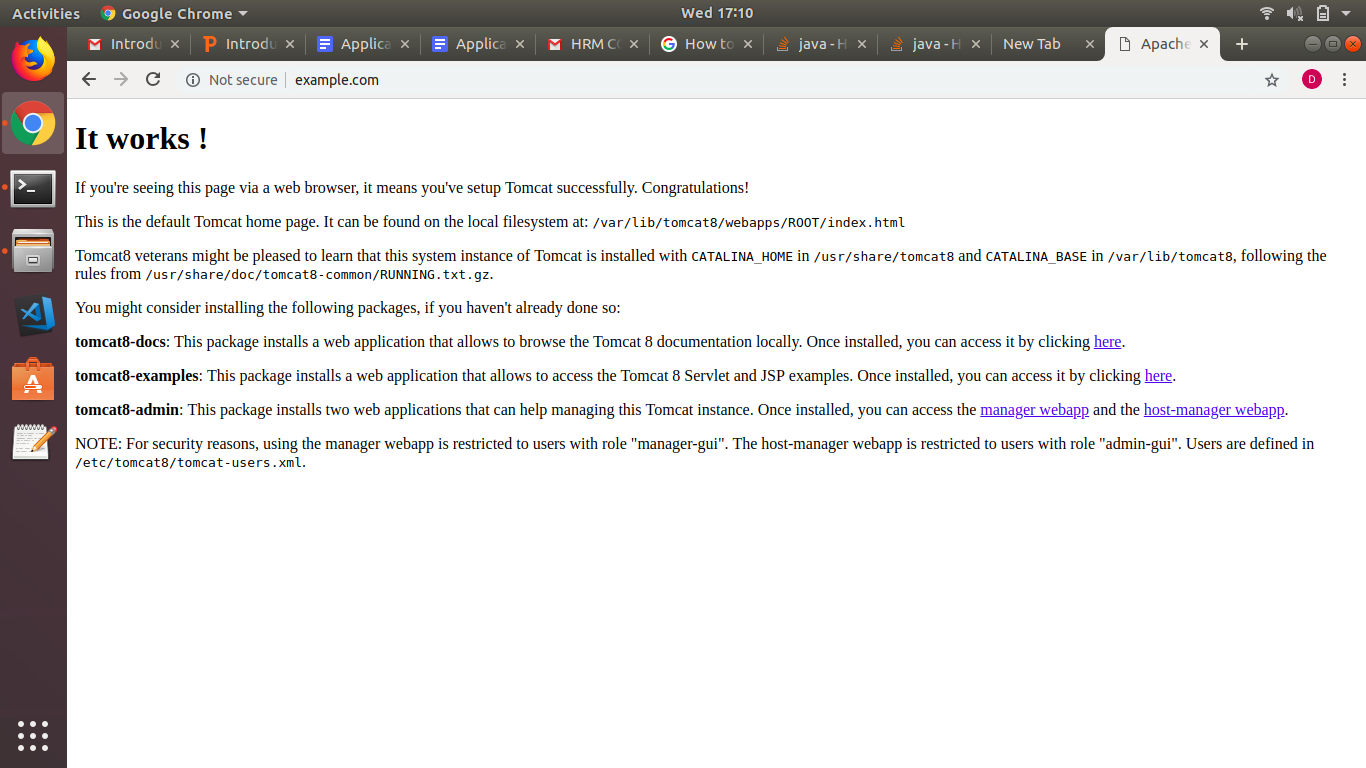
* **How to change default port in tomcat?**
  + Go to tomcat>conf folder
  + Edit server.xml
  + Search "Connector port"
  + Replace "8080" by your port number
  + Restart tomcat server.

<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />

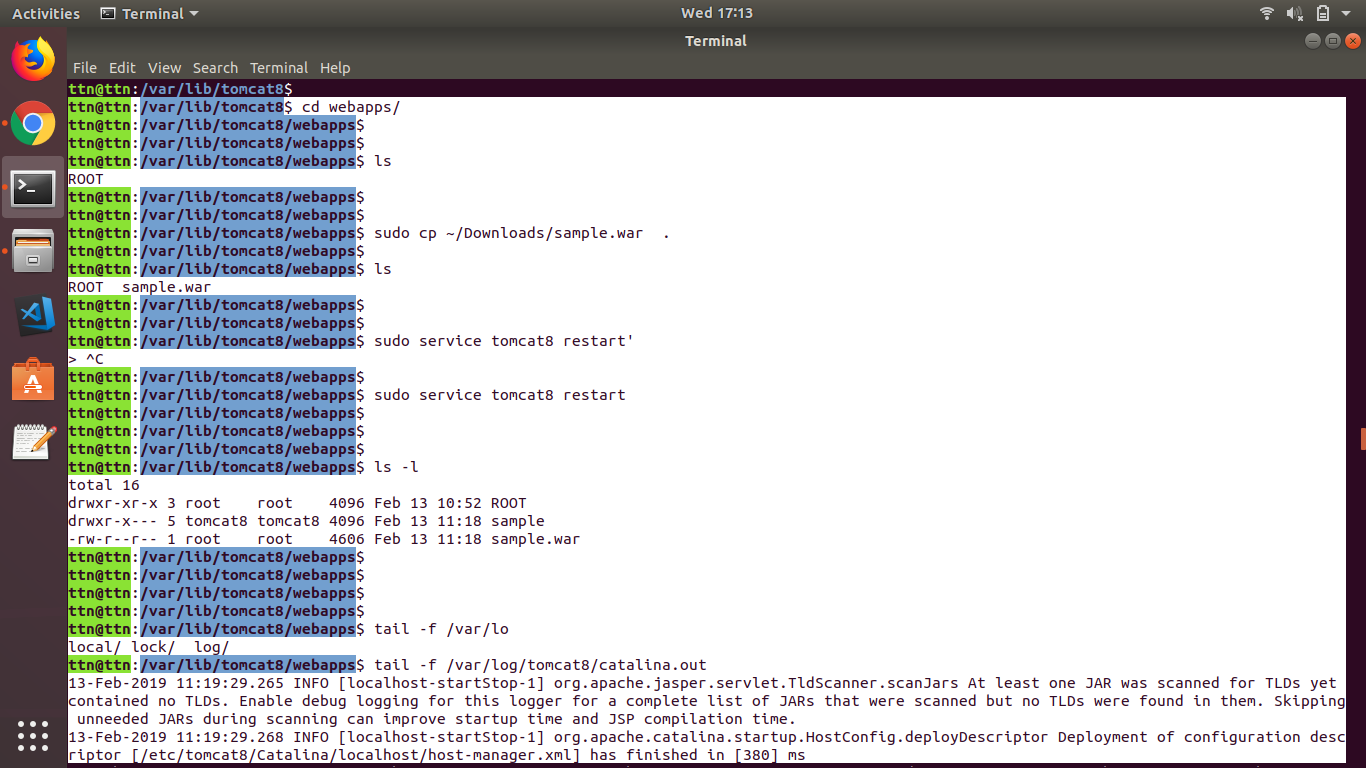
* **Install tomcat and nginx and set proxy\_pass in nginx to display tomcat.**

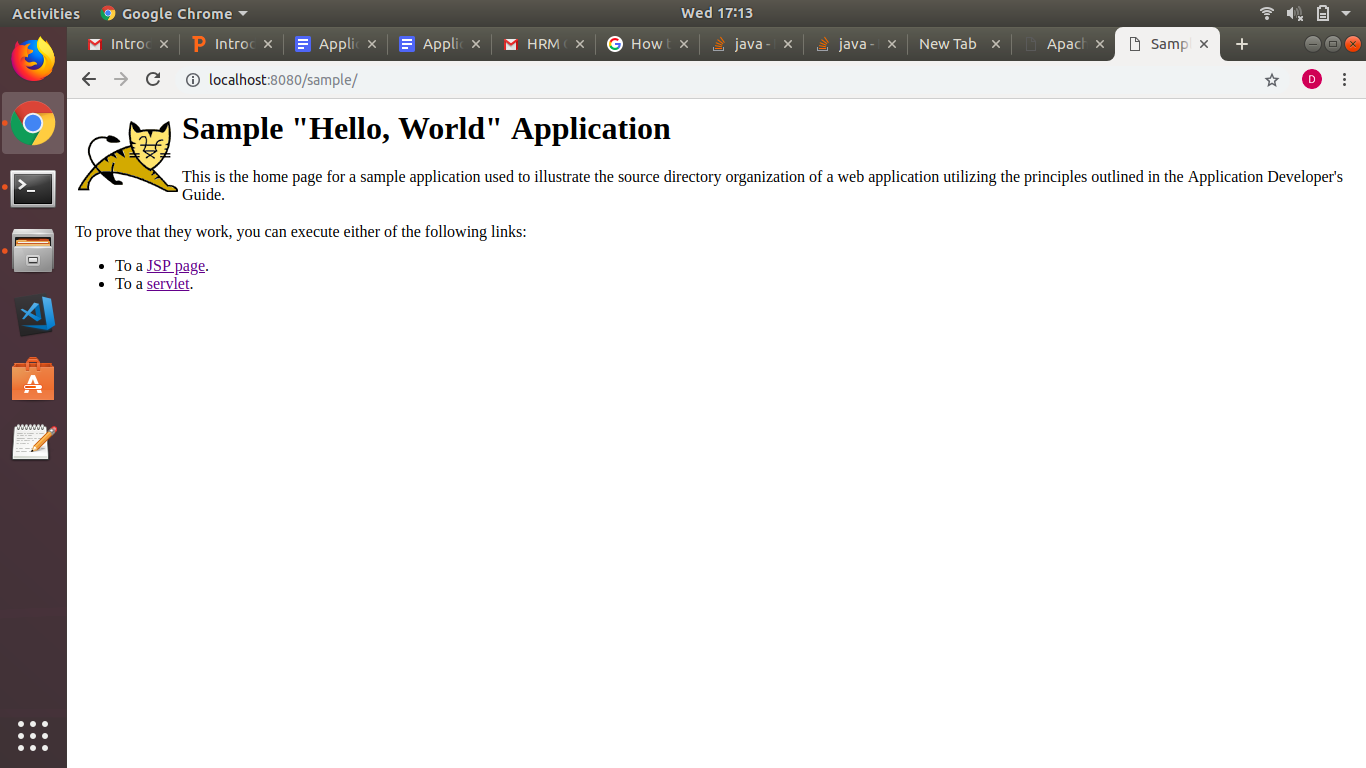
****

****

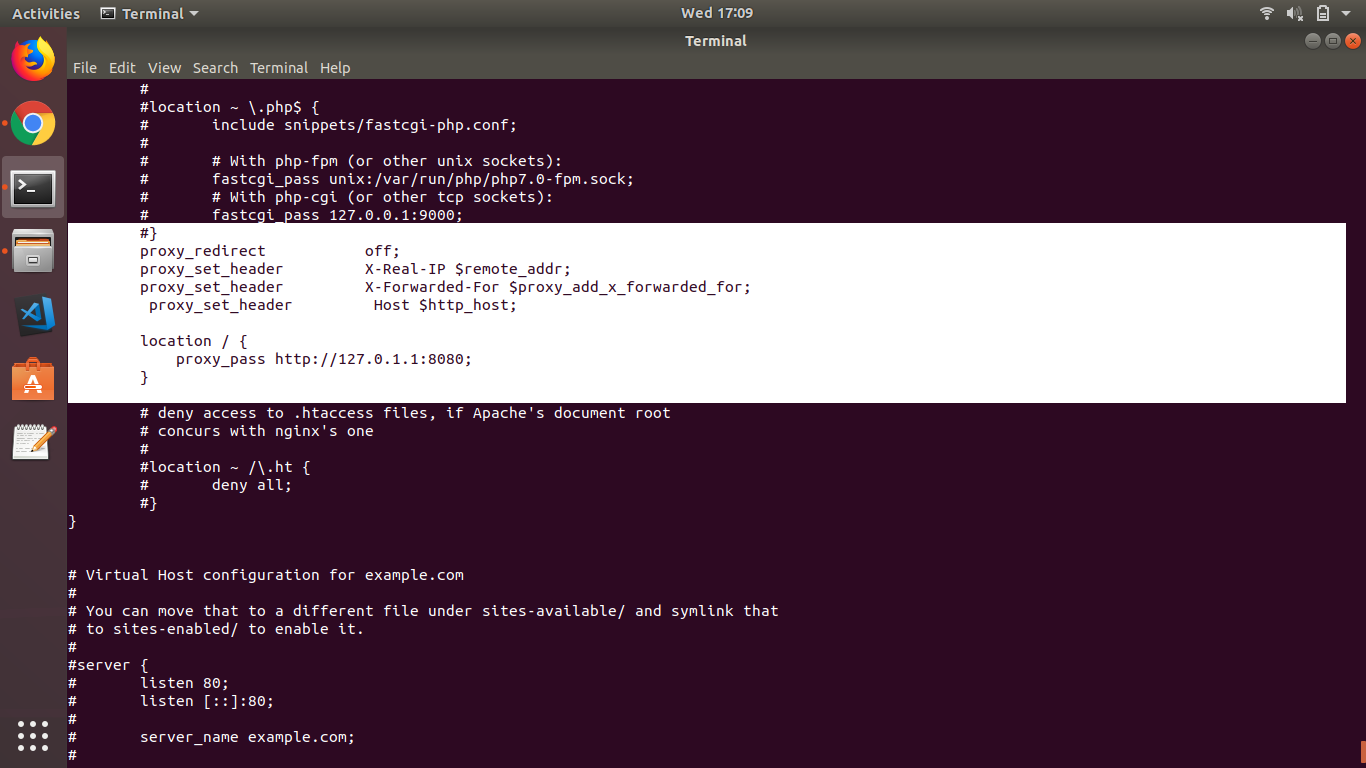
****

* **Configure Tomcat as a Service**

****

****

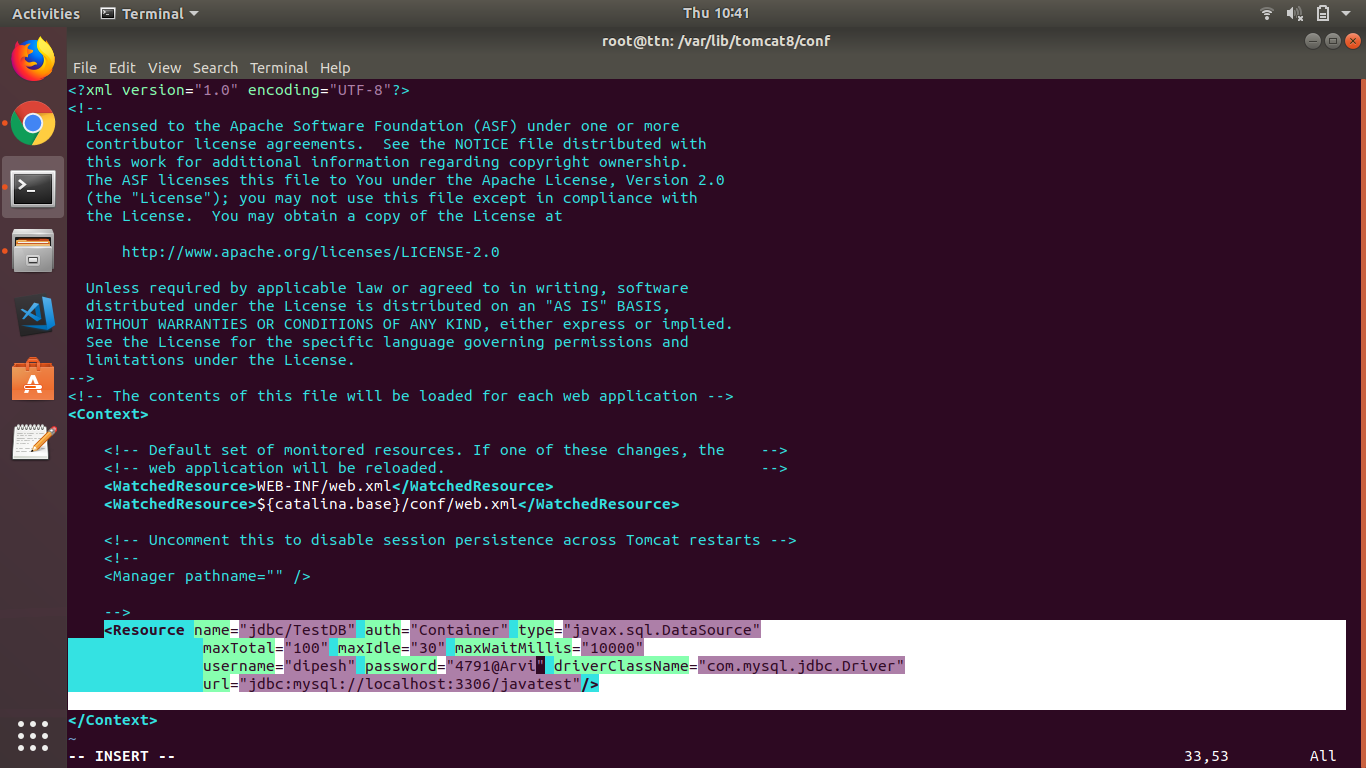
* **Use any web server in front of tomcat using AJP and HTTP connector.**

****

* **Configure tomcat to use any mysql server.**

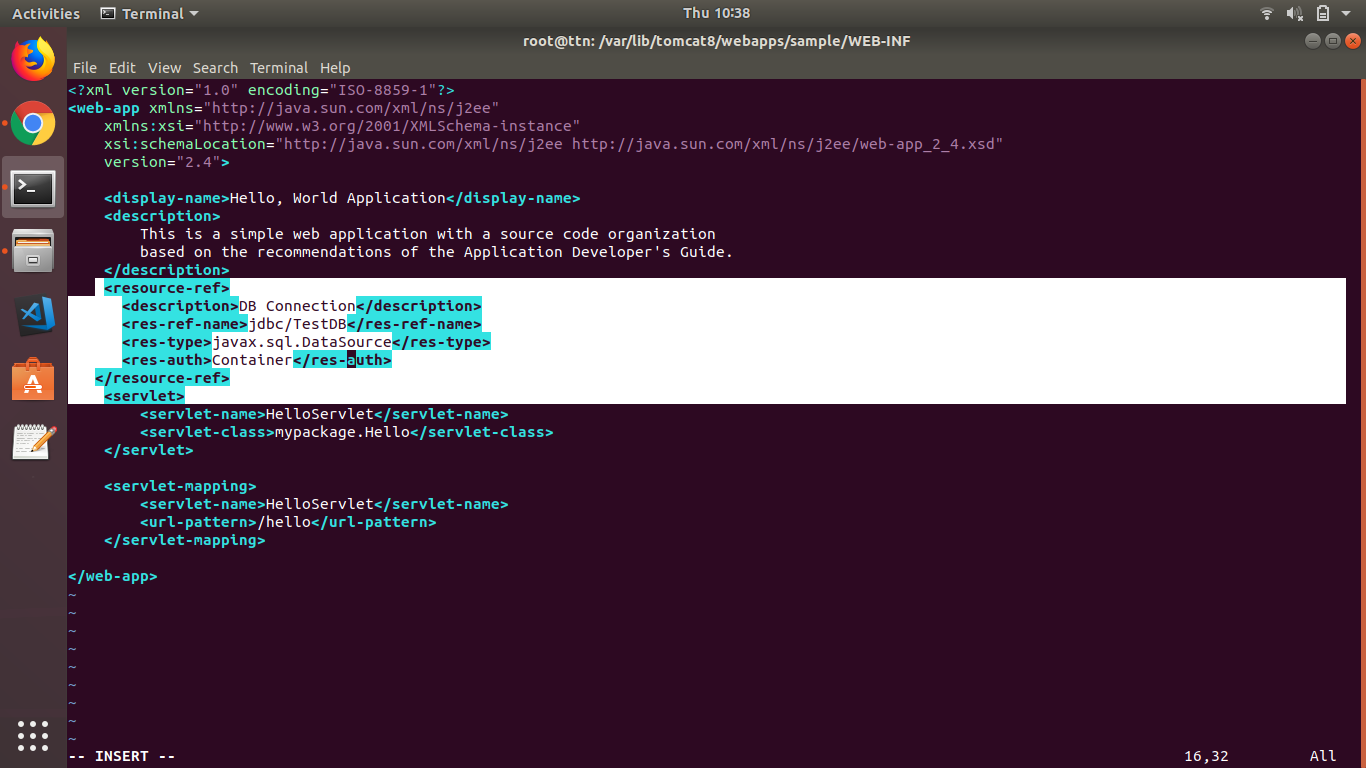
Configure the JNDI DataSource in Tomcat by adding a declaration for your resource to your Context.

root@ttn:/var/lib/tomcat8/conf# vim context.xml

****

**]**

root@ttn:/var/lib/tomcat8/webapps/sample/WEB-INF# vim web.xml

****