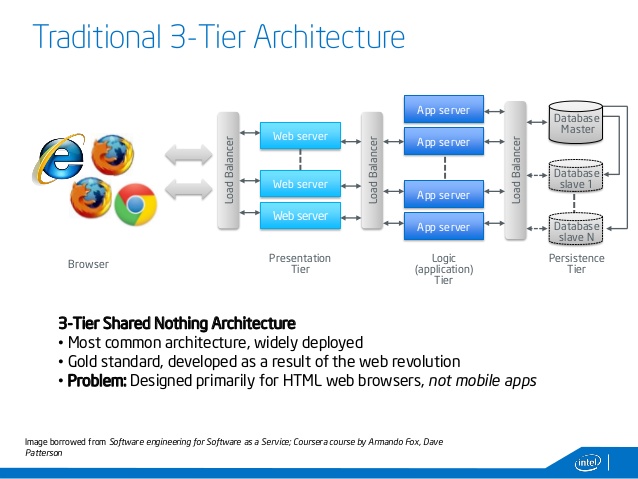
# 3-Tier Architecture

“3-tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms.”

A “tier” in this case can also be referred to as a “layer”. The three tiers, or layers, involved include:

1. A **Presentation Layer** that sends content to browsers in the form of HTML/JS/CSS. This might leverage frameworks like React, Angular, Ember, Aurora, etc.
2. An **Application Layer** that uses an application server and processes the business logic for the application. This might be written in C#, Java, C++, Python, Ruby, etc.
3. A **Data Layer** which is a database management system that provides access to application data. This could be MSSQL, MySQL, Oracle, or PostgreSQL, Mongo, et

# Examples

* Presentation Layer
  + **Apache HTTP Server**
  + NGINX
  + Apache Tomcat
  + Node.js
  + Lighttpd
  + Google Web Server
  + Microsoft IIS (Internet Information Services)
* Application Layer
  + **Tomcat**
  + JBoss/WildFly
  + Weblogic
  + Jetty
  + GlassFish
  + IBM WebSphere
* Data Layer
  + Oracle
  + MySQL (**MariaDB**)
  + Microsoft SQL Server
  + PostgreSQL
  + MongoDB

# Deploy Web Server - Apache HTTP Server

Install Apache HTTP Server on RH 7

yum install httpd -y

Start Web Service

systemctl start httpd

Check the Status Service

systemctl status httpd

To run Service at Startup

sytemctl enable httpd

To check the process of service

ps -ef | grep httpd

To check the port of service

netstat -lntp

Check the web server running on browser by typing IP Address

<http://IP-ADDRS/>

To add website content

/var/www/html/

Create the index.html as below

<h1>Hello Chetan</h1>

<pr>How are you?</pr>

httpd configuration location

/etc/httpd/

Default Configuration

/etc/httpd/conf

Project Configuration Directory will contain .conf files of projects

/etc/httpd/conf.d

Module configuration files directory

/etc/httpd/conf.modules.d

Main Configuration File

/etc/httpd/conf/httpd.conf

# Why Web Server?

Web Server is Secured more than Application Server

Web Server can handle load in form of Queues.

Web Server is designed to serve HTTP Content.

# Deploy App Server

Install Tomcat Server

yum install tomcat -y

To start the Tomcat Server

systemctl start tomcat

To check the status of Tomcat Server

systemctl status tomcat

To enable Tomcat to run at startup

systemctl enable tomcat

To check the process of Tomcat

ps -ef | grep httpd

To check the port of Tomcat

netstat -lntp

Tomcat Ports

8080: End User Port

8005: Shutdown Tomcat

8009: Backend Port

Location to Deploy Application

/var/lib/tomcat/webapps/

Applications will be as .jar/.war files

Log files in RH

/var/logs

Tomcat log files

/var/logs/tomcat/

# Installing Tomcat via Binaries

Download the Binary Distributions of Tomcat & Extract

cd /lib/

wget -O- <http://apache.mirror.serversaustralia.com.au/tomcat/tomcat-9/v9.0.7/bin/apache-tomcat-9.0.7.tar.gz> | tar -xzv

mv apache-tomcat-9.0.7 tomcat-9

Download the .war file to webapps directory

cd /lib/tomcat-9/webapps/

wget <https://github.com/cit-ager/APP-STACK/raw/master/student.war>

Start the Tomcat by running startup script

sh /lib/tomcat-9/bin/startup.sh

Check the status of tomcat

ps -ef | grep tomcat

Check the port status by

netstat -lntp

Open the Application by

<http://IP-ADDRS:8080/student/>

Check the logs by

tail -f /lib/tomcat-9/logs/catalina.out

# Database Creation

Install MariaDB Server

yum install mariadb-server -y

Enable & Start the MariaDB Server

systemctl enable mariadb

systemctl start mariadb

ps -ef | grep mariadb

MySQL Basics

show databases;

create database sample;

use sample;

show tables;

CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20), species VARCHAR(20), sex CHAR(1), birth DATE, death DATE);

desc pet;

Download the SQL Script for Student App

wget <https://raw.githubusercontent.com/cit-ager/APP-STACK/master/db-schema.sql>

Configure the table from Script

mysql < db-schema.sql

Grant Privileges

grant all privileges on studentapp.\* to 'student'@'demo-appserver.c.chetub-1406.internal' identified by 'student@1';

# Tomcat Database Configuration

Download the MariaDB Client

yum install mariadb -y

Check the DB Connection

mysql -h demo-dbserver -u student -p student@1

Configure the Tomcat context.xml

vim /lib/tomcat-9/conf/context.xml

<Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataSource"

maxActive="50" maxIdle="30" maxWait="10000"

username="student" password="student@1"

driverClassName="com.mysql.jdbc.Driver"

url="jdbc:mysql://demo-dbserver:3306/studentapp" />

Download JDBC connector to library directory of Tomcat

cd /lib/tomcat-9/lib/

wget <https://github.com/cit-ager/APP-STACK/raw/master/mysql-connector-java-5.1.40.jar>

Start the tomcat

sh /lib/tomcat-9/bin/startup.sh

# Web Server Configuration

Tomcat Provides Connector (modJK) to connect Application Server

Download the Source & Build the Connector

Download Source

wget -O- http://apache.mirror.digitalpacific.com.au/tomcat/tomcat-connectors/jk/tomcat-connectors-1.2.43-src.tar.gz | tar -xz

Check apxs installed?

which apxs

Check which program provides apxs

yum provides apxs

Download apxs

yum list all | grep httpd-devel

yum install httpd-devel.x86\_64

Install the GCC – C Compiler

yum install gcc -y

Run the Makefile with configuration

cd /root/tomcat-connectors-1.2.43-src/native

./configure --with-apxs=/bin/apxs

make

Check the mod\_jk.so connector after build

/root/tomcat-connectors-1.2.43-src/native/apache-2.0/

Install the mod\_jk.so in Modules in httpd

make install

Check the availabale modules in httpd

/usr/sbin/httpd -M

Modules Location

/etc/httpd/modules

Module configuration files location

/etc/httpd/conf.modules.d

Module configuration for mod\_jk.so

vim /etc/httpd/conf.modules.d/mod-jk.conf

LoadModule jk\_module modules/mod\_jk.so

Check for the mod\_jk module load

/usr/sbin/httpd -M | grep jk

Connector Port Details

cd /lib/tomcat-9/conf/

server.xml

Mod\_JK Configuration

cd /etc/httpd/conf.d

**# vim workers.properties**

worker.list=worker1

worker.worker1.type=ajp13

worker.worker1.host=10.142.0.3

worker.worker1.port=8009

**# vim mod-jk.conf**

LoadModule jk\_module modules/mod\_jk.so

JkWorkersFile conf.d/workers.properties

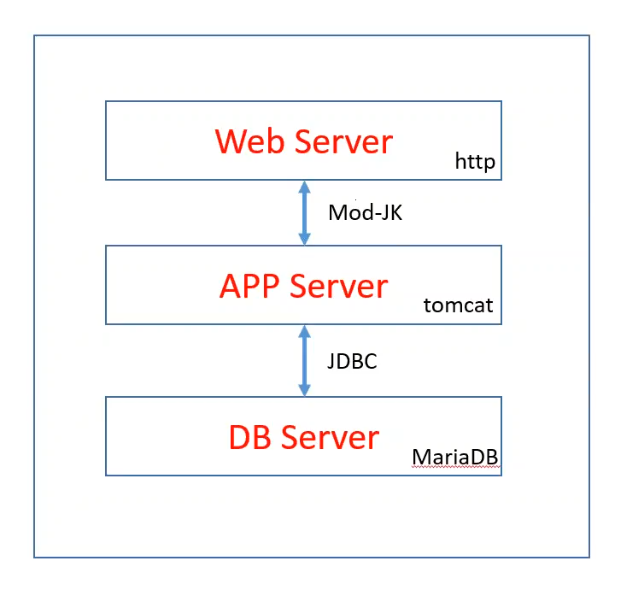
JkLogFile logs/mod\_jk.log

JkMount /student\* worker1

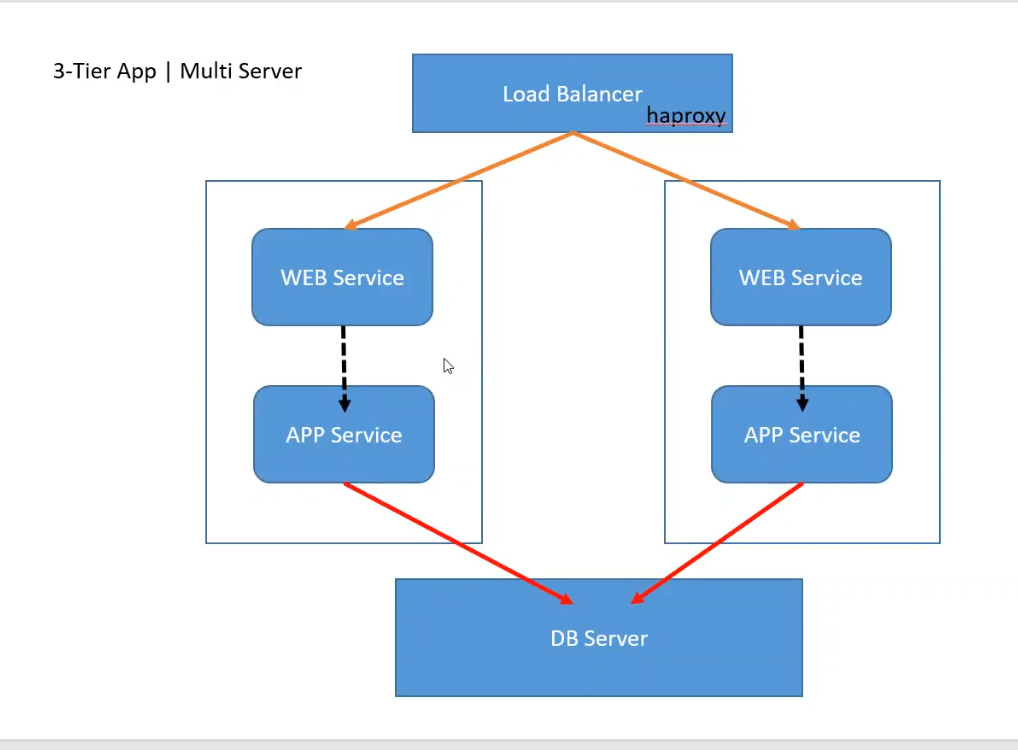
Check the httpd error

systemctl status httpd.service -l

Our 3 tier



Load Balancing



Install haproxy

yum install haproxy -y

systemctl enable haproxy

Configure haproxy.cfg

vim /etc/haproxy/haproxy.cfg

frontend main \*:80

default\_backend web

backend web

balance roundrobin

server web01 10.142.0.5:80 check

server web02 10.142.0.6:80 check