File name	YTM Enterprise Deployment Appliance
Author	Malloc Inc
Confidentiality	Internal
Last save date	Friday, January-10-2020 at 10:12:00 AM

Table of Contents

1 Introduction 2
2 Enterprise Configurations 2
2.1 High Performance Solution 2
2.2 High Performance and High Availability Solution 3
3 Apache HTTP Server 4
3.1 Step by Step Configuration Guide 4
3.2 Apache HTTPD and MOD_JK Installation 4
3.3 MOD_JK Configuration 5
3.4 Tomcat Setup 6
4 Useful links 6
5 PostgreSQL Hot Standby Configuration 6

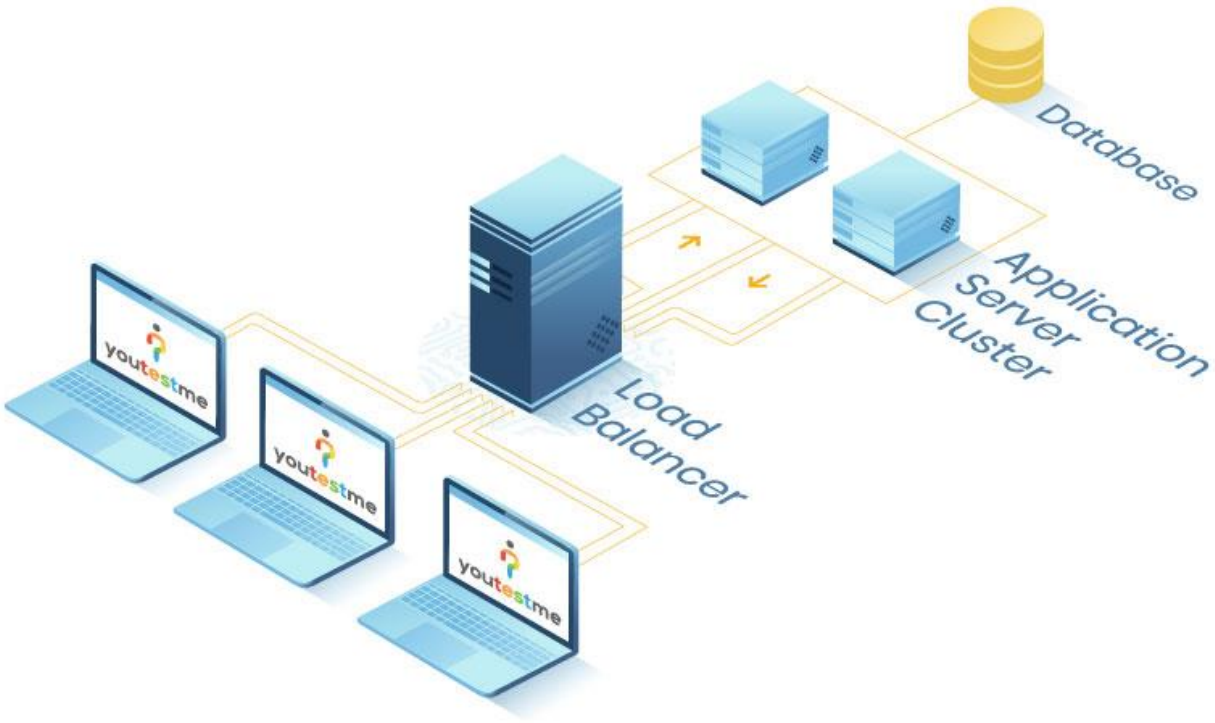
1 Introduction

This document describes the configuration of the YTM Enterprise Deployment Appliance.

2 Enterprise Configurations

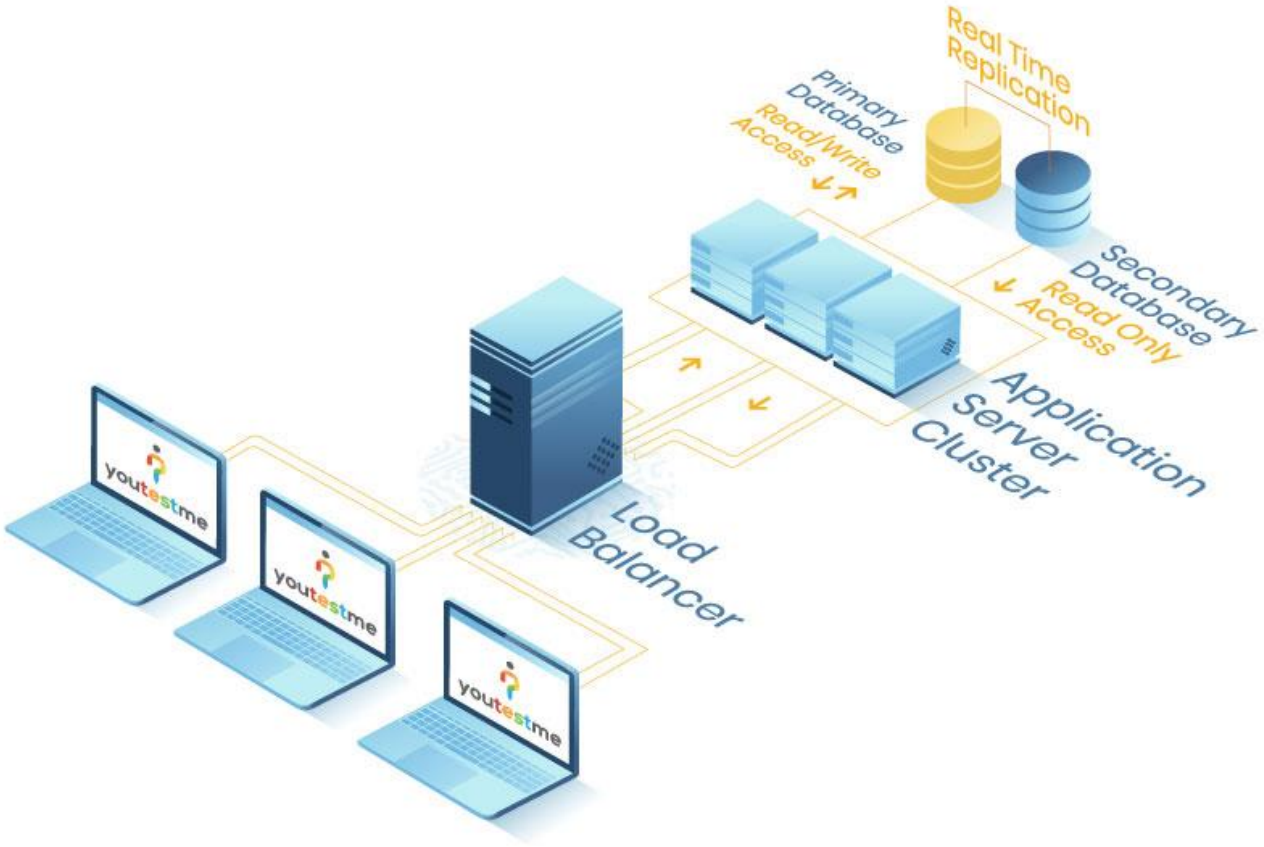
2.1 High Performance Solution

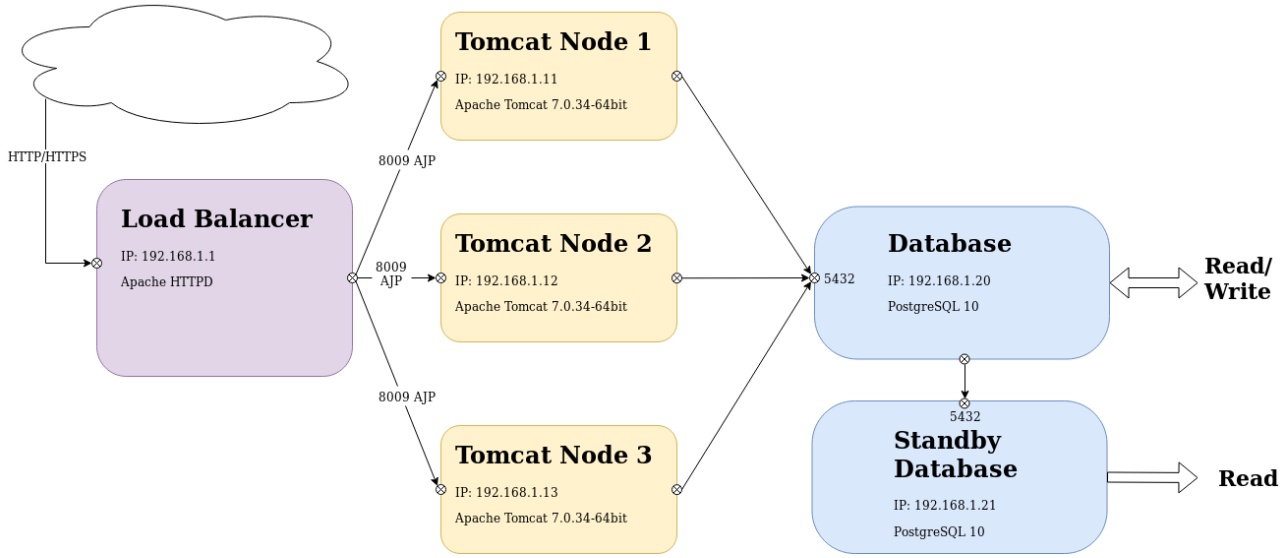
#	Layer	Min Number of VMs Required	Description
1.	Apache HTTP Server https://httpd.apache.org/docs/2.4/	1	v2.4
2.	A number of Tomcat Servers (default 2)	2+	Apache tomcat 7.0.34-64bit
3.	PostgreSQL database server	1	v10



2.2 High Performance and High Availability Solution

#	Layer	Min Number of VMs Required	Description
1.	Apache HTTP Server	1	v2.4
2.	A number of Tomcat Servers (default 2)	2+	Apache tomcat 7.0.34-64bit
3.	A number of PostgreSQL servers (default 2)	2	Primary and Hot Standby database - v10





3 Apache HTTP Server

3.1 Step by Step Configuration Guide

https://www.mulesoft.com/tcat/tomcat-clustering

3.2 Apache HTTPD and MOD_JK Installation

Install HTTPD package with other useful packages by running the following command:

```
# yum install httpd-devel apr apr-devel apr-util apr-util-devel gcc gcc-
c++ make autoconf libtool
```

Download and install Tomcat connector “mod_jk” from the official site:
http://tomcat.apache.org/download-connectors.cgi.

```
# mkdir -p /opt/mod_jk/
# cd /opt/mod_jk
# wget http://www.eu.apache.org/dist/tomcat/tomcat-connectors/jk/tomcat-
connectors-1.2.41-src.tar.gz
# tar -xvzf tomcat-connectors-1.2.41-src.tar.gz
# cd tomcat-connectors-1.2.41-src/native

# ./configure --with-apxs=/usr/sbin/apxs --enable-api-compatibility
# make
# libtool --finish /usr/lib64/httpd/modules
# make install
```

3.3 MOD_JK Configuration

Open the file "/etc/httpd/conf/httpd.conf" and add the following:

```
LoadModule jk_module "/etc/httpd/modules/mod_jk.so"

JkWorkersFile /etc/httpd/conf/workers.properties
# Where to put jk shared memory
JkShmFile   /var/run/httpd/mod_jk.shm
# Where to put jk logs
JkLogFile   /var/log/httpd/mod_jk.log
# Set the jk log level [debug/error/info]
JkLogLevel   info
# Select the timestamp log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "

<Location /status>
       JkMount jkstatus
       Require ip [Allowed_IP]
</Location>

<IfModule mod_jk.c>
       JkMount /* LoadBalancer
</IfModule>
```

Create new directory to store shared memory of the module:

```
# mkdir -p /var/run/httpd/mod_jk
# chown apache:apache /var/run/httpd/mod_jk
```

Next step will be to create "worker.properties" file with the following content:

```
# vim /etc/httpd/conf/workers.properties
```

```
worker.list=jkstatus, LoadBalancer
worker.jkstatus.type=status
worker.LoadBalancer.type=lb

worker.jvm1.type=ajp13
worker.jvm1.host=[TOMCAT1_IP]
worker.jvm1.port=[TOMCAT1_AJP_PORT]
worker.jvm1.lbfactor=1

worker.jvm2.type=ajp13
worker.jvm2.host=[TOMCAT2_IP]
worker.jvm2.port=[TOMCAT2_AJP_PORT]
worker.jvm2.lbfactor=1

worker.LoadBalancer.balance_workers=jvm1,jvm2
worker.LoadBalancer.sticky_session=true
```

3.4 Tomcat Setup

The "jvmRoute" attribute of the Engine element allows the load balancer to match requests to the JVM currently responsible for updating the relevant session. It does this by appending the name of the JVM to the JSESSIONID of the request, and matching this against the worker name provided in "workers.properties".

In order to configure *jvmRoute*, make sure that the value of "jvmRoute" for all your Engines is paired with an identically named Worker name entry in mod_jk's worker.properties configuration file.

We should set "jvmRoute" to support load-balancing via AJP by modifying Engine element:

```
<Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
```

At the end of the configuration process it is necessary to restart Tomcat server and Apache HTTPD.

4 Useful links

1. Configuring tomcat cluster with load balancer - https://www.mulesoft.com/tcat/tomcat-clustering
2. Official tomcat workers document - http://tomcat.apache.org/connectors-doc/common_howto/loadbalancers.html
3. Official tomcat cluster document - https://tomcat.apache.org/tomcat-8.5-doc/cluster-howto.html
4. Example with session replication in Java - https://examples.javacodegeeks.com/enterprise-java/tomcat/tomcat-clustering-session-replication-tutorial/
5. Tomcat Clustering configuration tutorial: http://www.datadisk.co.uk/html_docs/java_app/tomcat6/tomcat6_clustering.htm
6. https://geekflare.com/tomcat-load-balancer-using-mod-proxy-and-session-sticky/
7. https://access.redhat.com/documentation/mod_jk

5 PostgreSQL Hot Standby Configuration

All relevant information can be found in the following document:

https://svn.youtestme.com/admin/trunk/System Administration/PostgreSQL/YTM PostgreSQL Replication and Hot Standby.docx