In many cases, we have chosen Apache-tomcat as a java container. But many of us get confused about how to install it properly. Here I have tried to present it in a simple way. Hope you will like it.

# Install Web server

Install nginx

[root@websrv ~]# yum install nginx -y

[root@websrv ~]# service httpd restart

[root@websrv ~]# chkconfig httpd on

# Install java

Download the latest version of java from “http://www.oracle.com/technetwork/java/javase/downloads/index.html”.

[root@websrv ~]# mkdir -p /usr/jdk

[root@websrv ~]# tar -zxvf jdk-7u75-linux-x64.tar.gz -C /usr/jdk/

[root@websrv ~]# ls /usr/jdk/jdk1.7.0\_75/

Now, we will set it as our new java location for system uses. First we will check the current java version.

[root@websrv ~]# java -version

Then we will set up new java for our system.

[root@websrv ~]# export JAVA\_HOME=/usr/jdk/jdk1.7.0\_75/

[root@websrv ~]# export PATH=$JAVA\_HOME/bin:$PATH

But it will be removed after a server restart. So, Let's make it permanent.

[root@localhost ~]# vim /etc/profile

done

export JAVA\_HOME=/usr/jdk/jdk1.7.0\_75/

export PATH=$JAVA\_HOME/bin:$PATH

unset i

unset pathmunge

And check new/updated java version

[root@websrv ~]# java -version

**Advance tuning**

Install jce:

[root@db27 ~]# wget http://download.oracle.com/otn-pub/java/jce/7/UnlimitedJCEPolicyJDK7.zip?AuthParam=1477216623\_536262e065576926055a0faf4656fa13

[root@db27 ~]# unzip UnlimitedJCEPolicyJDK7.zip

[root@db27 ~]# cd UnlimitedJCEPolicy

[root@db27 ~]# chmod 777 \*

[root@db27 ~]# cp -prf local\_policy.jar /usr/jdk/jdk1.7.0\_75/jre/lib/security/

[root@db27 ~]# cp -prf US\_export\_policy.jar /usr/jdk/jdk1.7.0\_75/jre/lib/security/

http://download.oracle.com/otn-pub/java/jce/8/jce\_policy-8.zip?AuthParam=1477851060\_73faa5d6640bda70827b00053a289bfa

# Install apache-tomcat

Download the latest version of apache-tomcat from “https://tomcat.apache.org/”.

[root@websrv ~]# tar -zxvf apache-tomcat-8.0.36.tar.gz -C /usr/share/

Define Java location:

[root@websrv ~]# vim /usr/share/apache-tomcat-8.0.36/bin/catalina.sh

JAVA\_HOME=/usr/jdk/jdk1.7.0\_75/

Set Java Heap size: (memory allocation for java)

[root@App1 ~]# vim /usr/share/apache-tomcat-8.0.36/bin/setenv.sh

CATALINA\_OPTS="-Djava.awt.headless=true -Xms1024m -Xmx20480m -XX:PermSize=24576m -XX:MaxPermSize=24576m"

Create a user for tomcat manager to access from the web.

[root@websrv ~]# vim /usr/share/apache-tomcat-8.0.36/conf/tomcat-users.xml

# Append the below lines under the <tomcat-users> tag

<role rolename="manager-gui"/>

<user username="tomcat" password="our\_Password\_Here" roles="manager-gui"/>

**Allow remote tomcat manager access**

For Tomcat v8.5.4 and above, the file <tomcat>/webapps/manager/META-INF/context.xml has been adjusted:

[root@ps2 ~]# vim /usr/share/apache-tomcat-8.5.13/webapps/manager/META-INF/context.xml

<Context antiResourceLocking="false" privileged="true" >

<Valve className="org.apache.catalina.valves.RemoteAddrValve"

allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />

</Context>

Change this file to comment the Valve

<Context antiResourceLocking="false" privileged="true" >

<!--

<Valve className="org.apache.catalina.valves.RemoteAddrValve"

allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />

-->

</Context>

# 

# 

# Redirect request with nginx

root@testserver1:~# vim /etc/nginx/sites-enabled/mahidul.conf

server {

listen 80;

server\_name mahidul.com;

location / {

proxy\_pass http://127.0.0.1:8080/SampleWebApp/;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

}

}

root@testserver1:~# service nginx restart

# Advance tuning

**Tune tomcat default session timeout**

[root@websrv ~]# vim /usr/share/apache-tomcat-8.0.36/conf/web.xml

<session-config>

<session-timeout>15</session-timeout>

</session-config>

**Tuned tomcat's a little bit**

[root@websrv ~]# vim /usr/share/apache-tomcat-8.0.36/conf/server.xml

<Connector port="9090" protocol="HTTP/1.1"

connectionTimeout="10000"

acceptCount="10"

enableLookups="false"

compression="off"

maxConnections="8192"

maxThreads="2000"

minSpareThreads="25"

maxSpareThreads="75"

tcpNoDelay="true"

redirectPort="9443" />

<Connector port="9009" protocol="AJP/1.3"

maxThreads="2000" redirectPort="8443" tomcatAuthentication="false"

keepAliveTimeout="300000" connectionTimeout="300000" />

**Increase maximum upload file size**

[root@ps1 shaown]# vim /usr/share/apache-tomcat-8.0.36/webapps/manager/WEB-INF/web.xml

<multipart-config>

<!-- 50MB max -->

<max-file-size>552428800</max-file-size>

<max-request-size>52428800</max-request-size>

<file-size-threshold>0</file-size-threshold>

</multipart-config>

**Set database connection numbers**

[root@ps2 webapps]# vim /usr/share/apache-tomcat-8.0.36/webapps/profino\_fsibl/WEB-INF/classes/profino.properties

hibernate.c3p0.maxPoolSize = 100

[root@ps2 webapps]# vim /usr/share/apache-tomcat-8.0.36/webapps/surecash\_bcbl/WEB-INF/rest-api-dispatcher-servlet.xml

p:maxActive="200"

Now let's add a script to get “service tomcat {start|stop|restart}” options. Please change the JAVA\_HOME, TOMCAT\_HOME, START\_TOMCAT and STOP\_TOMCAT script location as per your configuration.

**Start the tomcat server**

[root@websrv ~]# vim /etc/init.d/tomcat

#!/bin/bash

# chkconfig: 234 20 80

# description: Tomcat Server basic start/shutdown script

# processname: tomcat

JAVA\_HOME=/usr/jdk/jdk1.7.0\_75/

export JAVA\_HOME

TOMCAT\_HOME=/usr/share/apache-tomcat-8.0.36/bin

START\_TOMCAT=/usr/share/apache-tomcat-8.0.36/bin/startup.sh

STOP\_TOMCAT=/usr/share/apache-tomcat-8.0.36/bin/shutdown.sh

tomcat\_pid() {

echo `ps aux | grep org.apache.catalina.startup.Bootstrap | grep -v grep | awk ‘{ print $2 }’`

}

start() {

echo -n “Starting tomcat: “

cd $TOMCAT\_HOME

${START\_TOMCAT}

echo “done.”

}

stop() {

echo -n “Shutting down tomcat: “

cd $TOMCAT\_HOME

${STOP\_TOMCAT}

echo “done.”

}

case “$1″ in

start)

start

;;

stop)

stop

;;

restart)

stop

sleep 10

start

;;

status)

pid=$(tomcat\_pid)

# pid=`ps -ef | grep java | grep tomcat | cut -c10-14`

if [ -n “$pid” ]

then

echo “Tomcat is running with pid: $pid”

else

echo “Tomcat is not running”

fi

;;

\*)

echo “Usage: $0 {start|stop|restart|status}”

esac

exit 0

So, let's start the tomcat service,

[root@websrv ~]# chmod 755 /etc/init.d/tomcat

[root@websrv ~]# service tomcat restart

[root@websrv ~]# chkconfig tomcat on

We can also restart apache-tomcat services manually like below-

[root@websrv ~]# killall java

[root@websrv ~]# sh /usr/share/apache-tomcat-8.0.36/bin/shutdown.sh

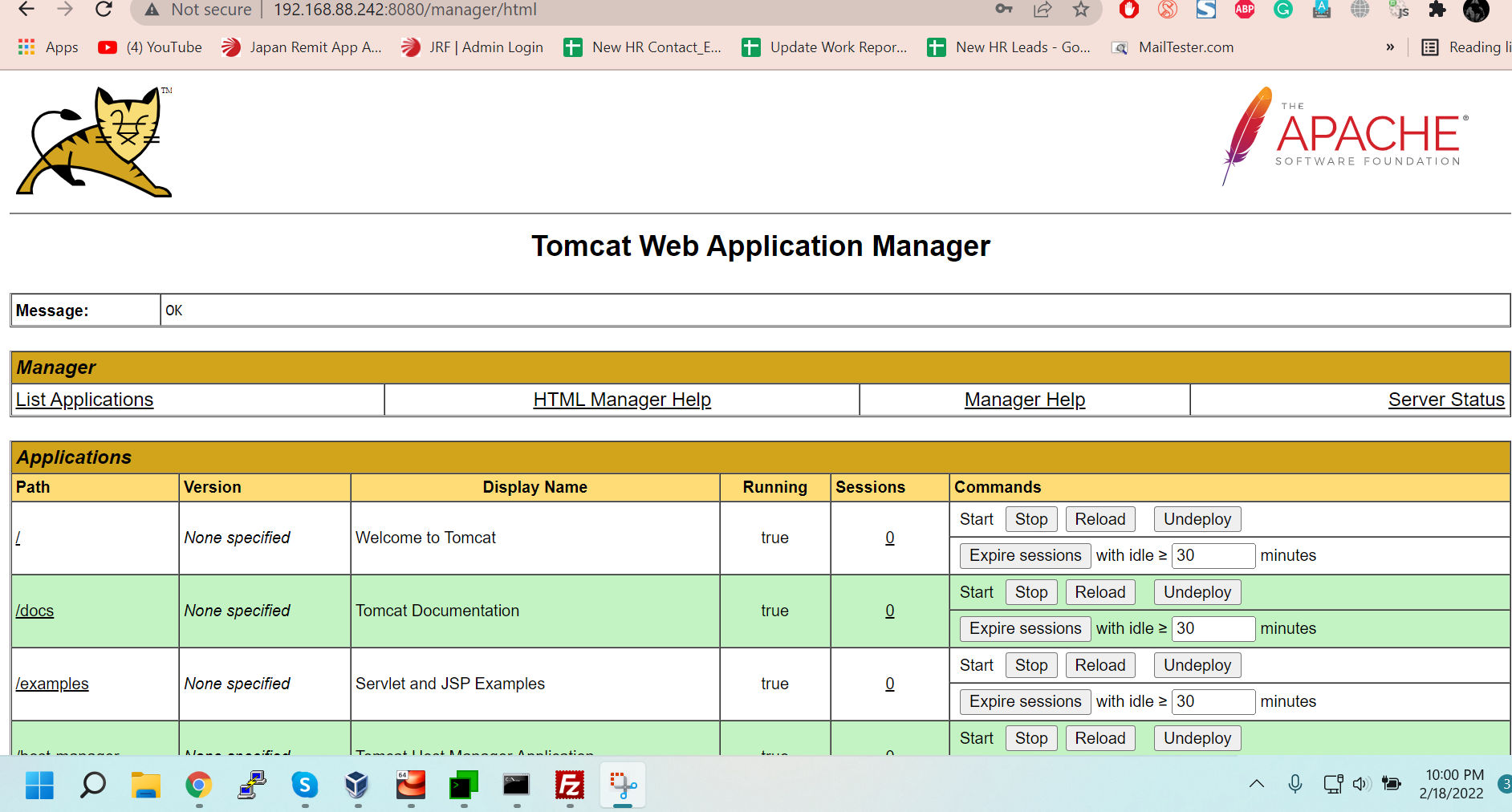
[root@websrv ~]# sh /usr/share/apache-tomcat-8.0.36/bin/startup.sh

Monitoring the status from log file:

[root@websrv ~]# tail -f /usr/share/apache-tomcat-8.0.36/logs/catalina.out

**Check the tomcat manager web page**

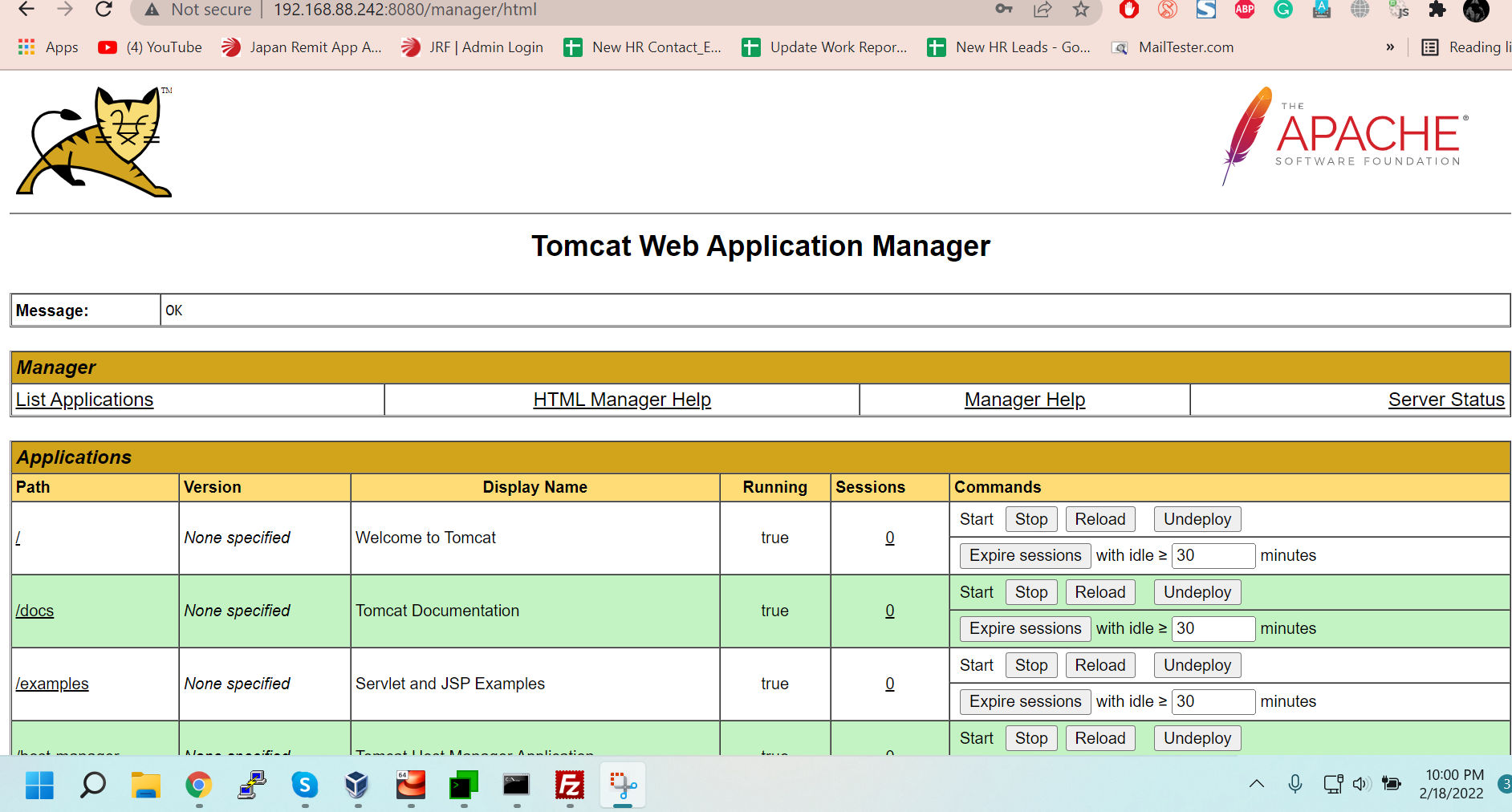
<http://server_ip_address:8080/manager/html>



# Java software deployment

Upload the war from web interface or copy the war to the webapps folder

[root@websrv ~] # cp [.war] /usr/share/apache-tomcat-8.5.75/webapps



# Log Rotation

Tomcat log file size gets more than 2GB, it may crash. so, let's rotate the log.

[root@websrv ~]# vim /etc/logrotate.d/tomcat

/usr/share/apache-tomcat-8.0.36/logs/catalina.out {

copytruncate

#maxsize 200M

daily

rotate 5

compress

dateext

missingok

}

It will rotate tomcat log when catalina.out is

1. copytruncate: copytruncate itself creates a copy of the original log file before truncating that original file to zero size. Continue to write the log information in the newly created file after rotating the old log file.

2. daily, weekly, monthly, yearly option: Rotate the log file weekly/daily/monthly

3. rotate 60: keep 60 days/weeks/months worth of backlogs

4. dateext: Rotate the old log file with date in the log filename

5. missingok: Don't return error if the log file is missing

6. compress: Compress the files after rotating them. the rotated files will be compressed with gzip utility.

8. size: If you want to rotate a log file (for example, /tmp/output.log) for every 1KB, create the logrotate.conf as shown below.

Let's check what this log rotation actually trying to do-

[root@websrv ~]# logrotate -vdf /etc/logrotate.d/tomcat

If everything looks good, then execute it manually once to check live performance.

[root@websrv ~] # logrotate -vf /etc/logrotate.d/tomcat

# Configure NGINX for serving tomcat requests