TOMCAT INSTALLATION AND CONFIGURATION ON AWS EC2

Introduction

Tomcat or Apache Tomcat is light weight, open source web container used to deploy and running the java based web applications, developed by Apache Software Foundation (ASF)

What is Webserver?

A Web server is a program that uses HTTP(Hypertext Transfer Protocol) to serves web content (HTML and static content) to users

Tomcat Directory Structure

bin : The bin folder contains .exe files and .bat files and most important file for us which is used to start and stop the tomcat web server.

/opt/apache-tomcat-8.5.72/bin

.sh will run in linux server

.bat will be run in windows

conf

The conf contains configuration, \*.xml files. Some of the regularly used files are:- server.xml, tomcat-users.xml, context.xml. The server.xml is used to modify the server configuration like a port number. The tomcat-users.xml file is used to add/modify/update admin details like username, password, and their role. Similarly, the context.xml file is used to modify the monitoring resources.

lib

The lib folder contains all the library files in the form of JAR files which are required for tomcat server execution. Examples of some files: - serverlet-api.jar, jsp-api.jar, jaspar.jar, catalina.jar, and e.t.c. The catalina.jar represents the servlet container and jaspar.jar represents the JSP container. The serverlet-api.jar and jsp-api.jar represent the original servlet and JSP technology API respectively.

webapps

The webapps folder will contain all the deployed web applications. After developing the web applications having web components, we deploy them in the tomcat server by copying them to this folder either in the form of directories or in the form of the WAR file. WAR means Web Application Archive, a kind of JAR file.

logs: - All the generated log files will be stored here in the form of \*.log and \*.txt file. Every day new log files will be generated.

temp: - It will contain all the temporary files.

work: - It contains a Catalina folder, and that contains the servlet container internal folder.

contains the translated servlet source files and classes of JSP/JSF. Organized in hierarchy of engine name (Catalina), host name (localhost), webapp name followed by the java classes package structure

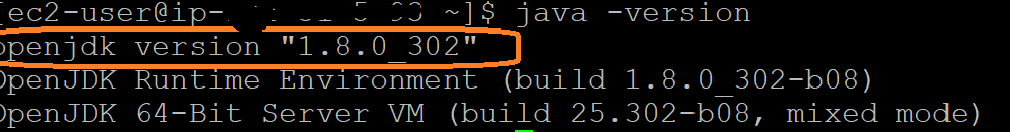
Tomcat Installation in Linux server

1. Launch a red-hot Linux server
2. Install Java

* #yum install java-1.8\* -y

check point

* java -version



1. Install Apache Tomcat

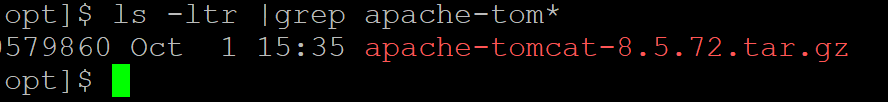
Get the package from apache website:

<https://tomcat.apache.org/download-80.cgi>

Go to /opt folder

#cd /opt

# sudo wget [**https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.72/bin/apache-tomcat-8.5.72.tar.gz**](https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.72/bin/apache-tomcat-8.5.72.tar.gz)



1. # sudo tar -zvxf apache-tomcat-8.5.72.tar.gz



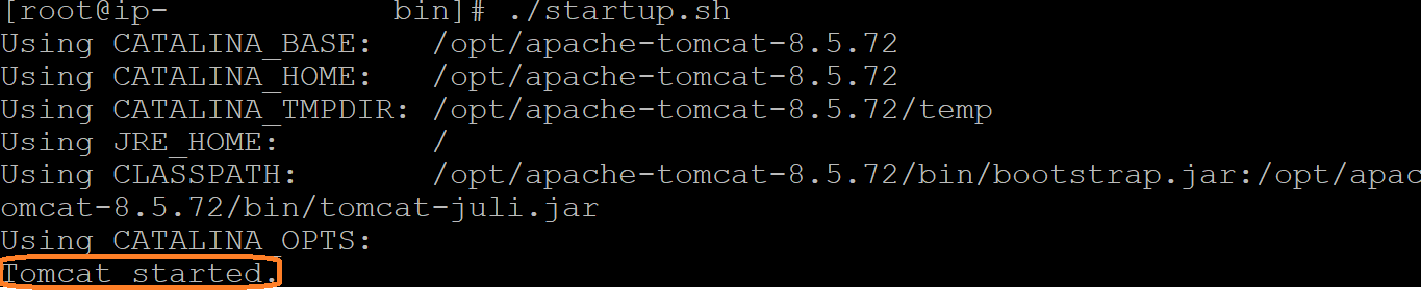
1. #Giving all permission to tomcat folder including subfolders

# sudo chmod 777 -R apache-tomcat-8.5.72

1. # cd apache-tomcat-8.5.72/bin

# start the tomcat

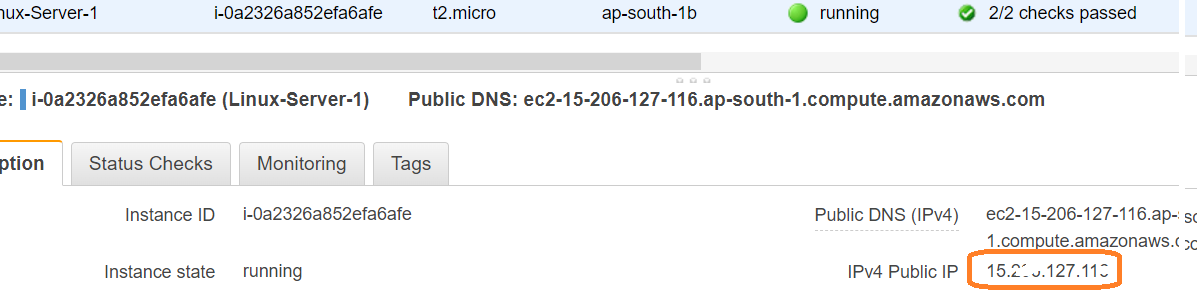
/opt/apache-tomcat-8.5.72/bin/startup.sh



1. #stop the tomcat

/opt/apache-tomcat-8.5.72/bin/shutdown.sh

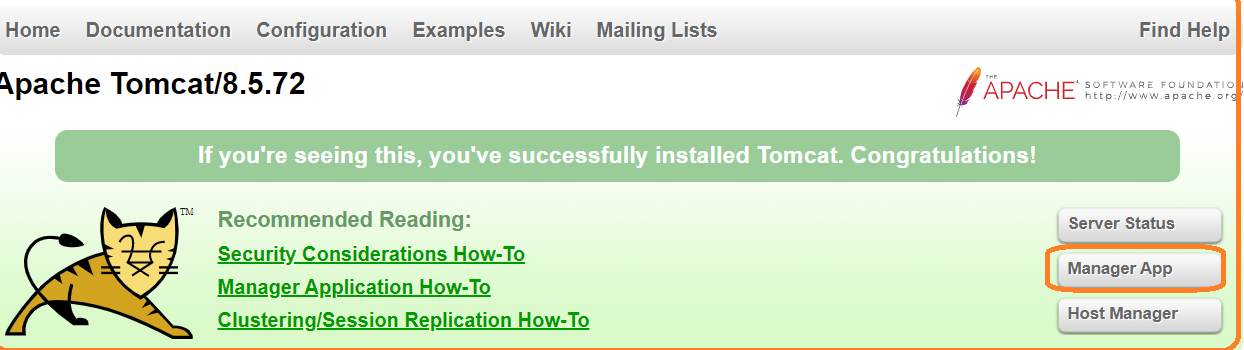
AWS Instance:



1. **#Checkpoints**

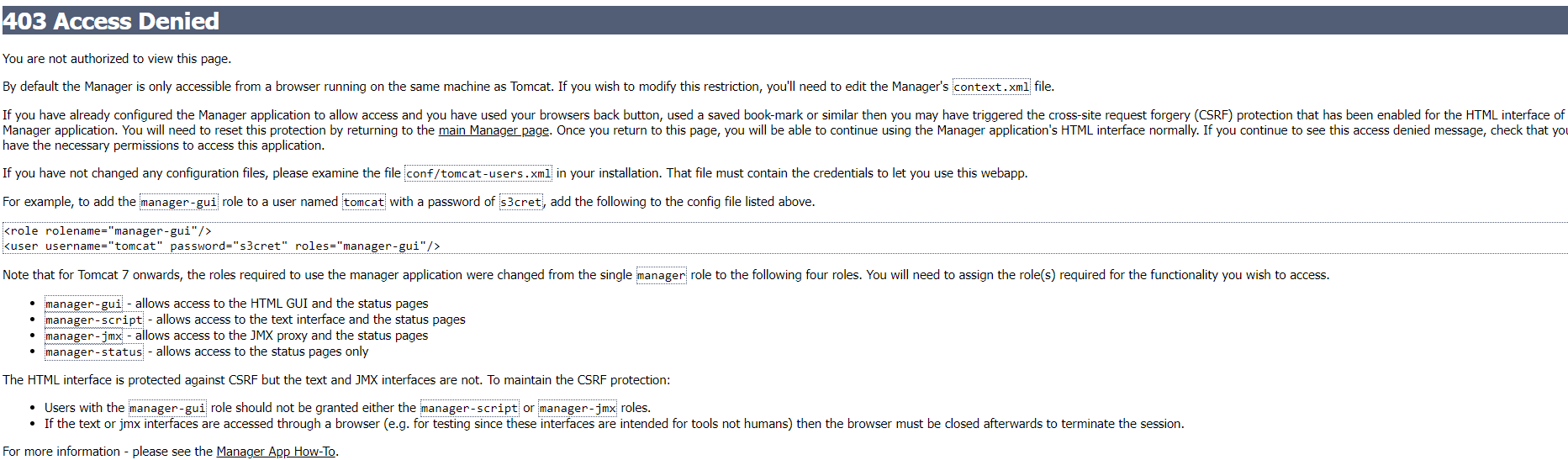
Access tomcat application from browser on port 8080

http://<Public\_IP of AWS Instance>:8080



1. If we click **Manager-App,** we will get below message

By default, we can’t access outside network, but we can access same ec2 instance and in order to access outside as well we need to configure few changes in **context.xml** file



10.To resolve above issue, need to follow below steps

Search for context.xml file

**#find / -name context.xml**

Above command gives 3 context.xml files.

Comment ***()ValueClassNamefield*** on files which are under webapp directory. After that restart tomcat services to effect these changes

**i./opt/apache-tomcat-8.5.69/webapps/host-manager/META-INF/context.xml**

***<!--<Valve className="org.apache.catalina.valves.RemoteAddrValve"***

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

**ii./opt/apache-tomcat-8.5.69/webapps/manager/META-INF/context.xml**

***<!--<Valve className="org.apache.catalina.valves.RemoteAddrValve"***

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

iii. update users’ information in the tomcat-users.xml file, go to Tomcat home directory and add below users to **conf/tomcat-user.xml** file

**Path:**

***/opt/apache-tomcat-8.5.72/conf/tomcat-users.xml***

***<user username="admin" password="admin" roles="manager-gui,admin-gui"/>***

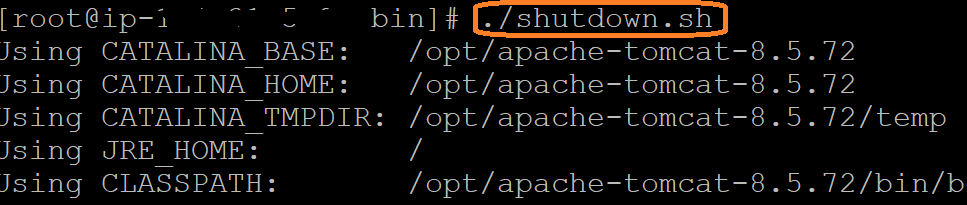
**Note:**

if you give **manager-gui**, we can access only **manager-app** and **server-status**

if you give **admin-gui,** we can access **host-manager**

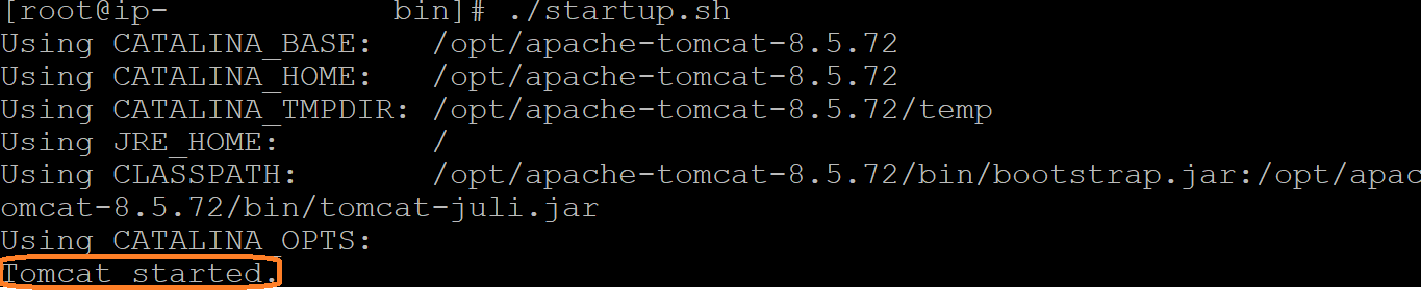
#**shutdown the tomcat**

**/opt/apache-tomcat-8.5.72/bin/shutdown.sh**

****

**#start the tomcat**

**/opt/apache-tomcat-8.5.72/bin/startup.sh**



**If we want(optional**), Change the port number in conf/server.xml file under tomcat home

# cd /opt/apache-tomcat-8.5.54/conf

# update port number in the “connecter port” field in server.xml

# stop and start the tomcat after configuration update

**<Connector port="8080" protocol="HTTP/1.1"**

**connectionTimeout="20000"**

**redirectPort="8443" />**

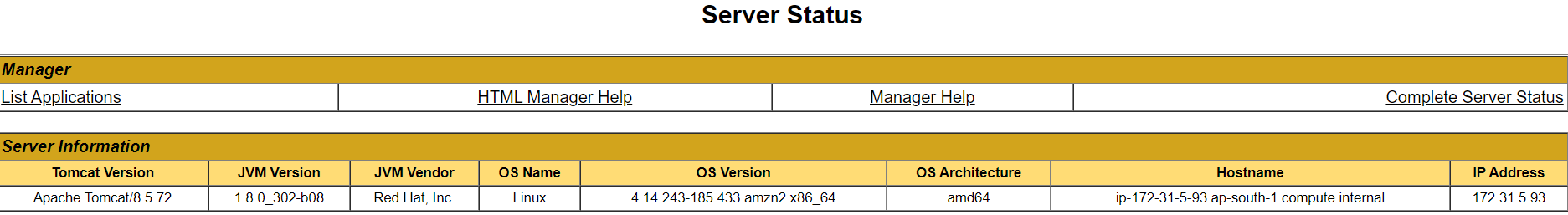
**Note:** **By default, port number is 8080**

**Checkpoints**

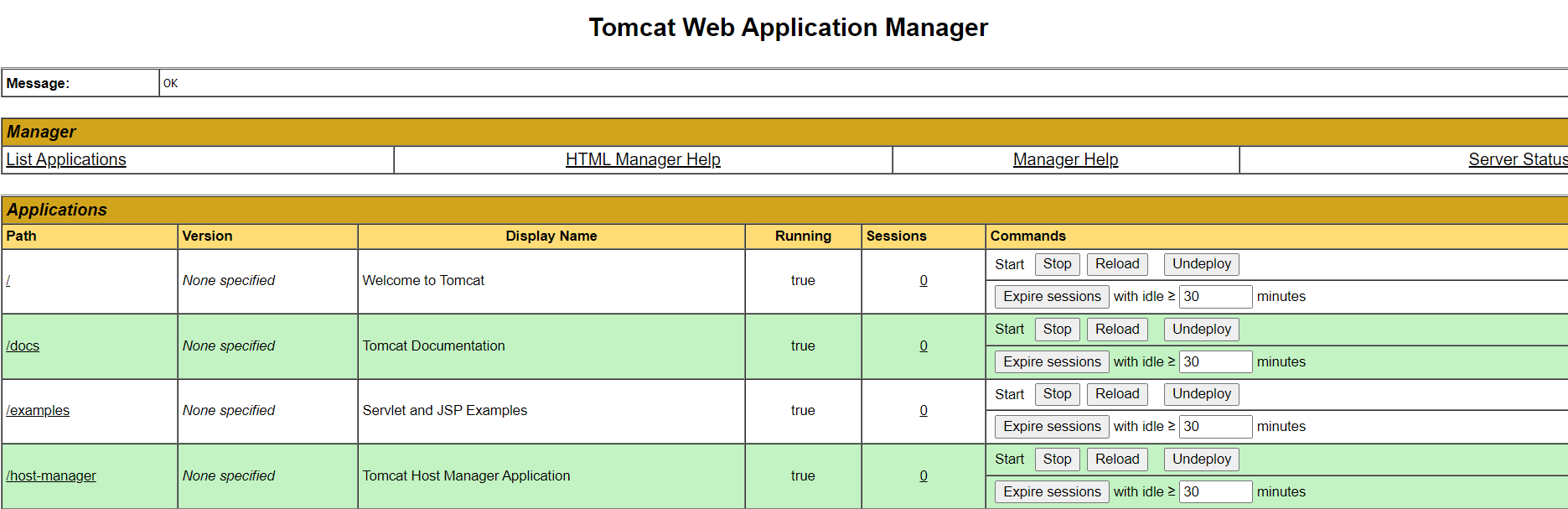
Access tomcat application from browser on port 8080

http://<Public\_IP of AWS Instance>:8080

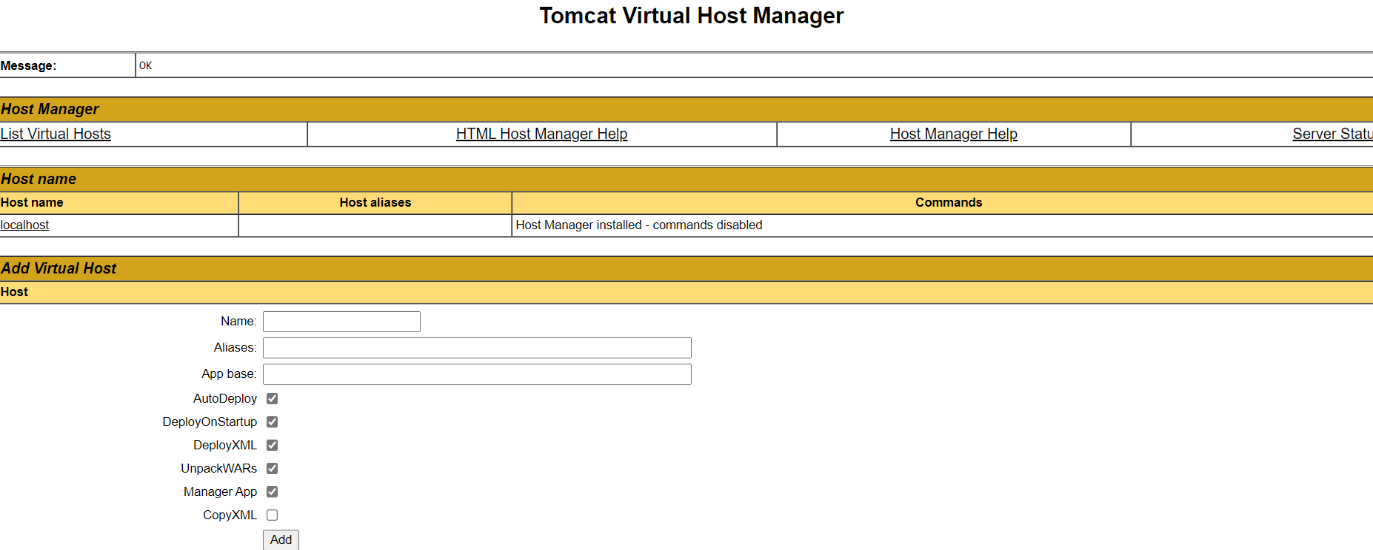
From Tomcat home page, click on ‘**Server Status’**

******

***Click on Manager-App button***

******

***Click on Host-Manager***

******

**Server Status & Manager App**

To access the server status application and Manager app, tomcat7 on words below roles as introduced

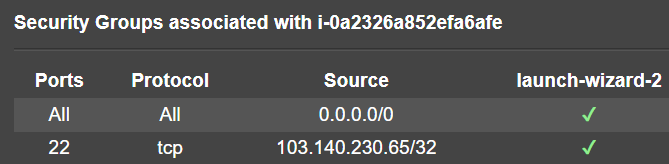
* **manager-gui :** allows access to the HTML GUI and Server and Manger App pages

<user username="admin" password="admin" roles="manager-gui"/>

* **admin-gui :** allows access to the Host Manager page

<user username="admin" password="admin" roles="admin-gui"/>

We need to enable security Group in AWS instance

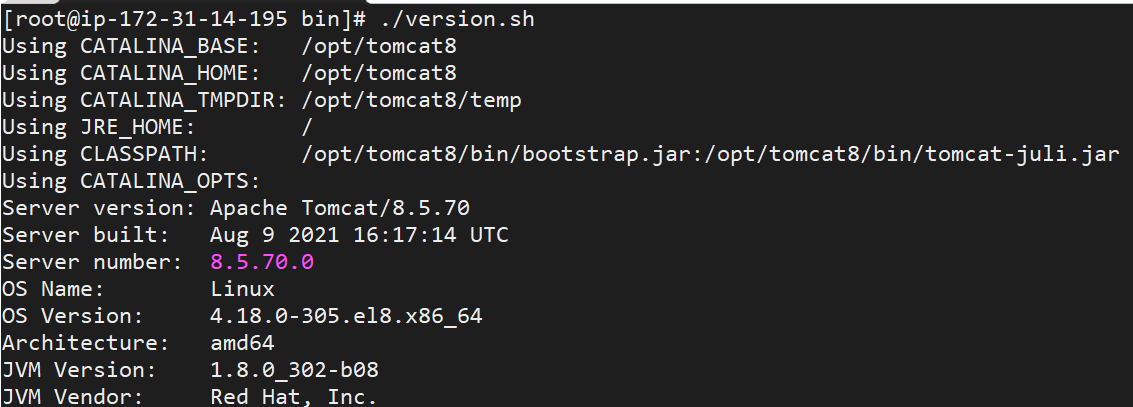
**

Restart service and try to login to tomcat application from the browser.

To check tomcat version in EC2 instance

$ cd /opt/tomcat/bin

**$./version.sh**

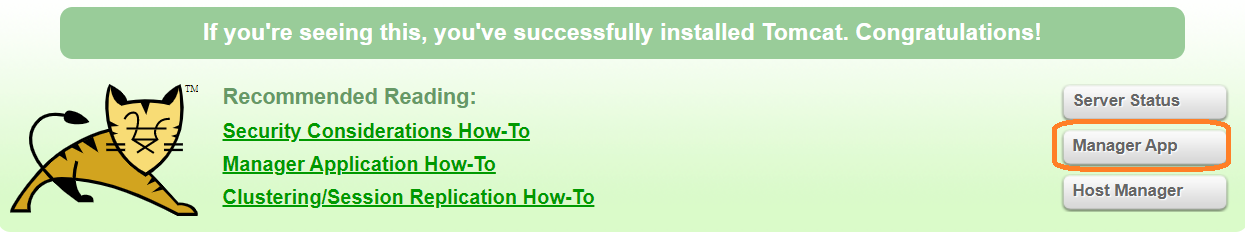


Clear cache if not working as expected

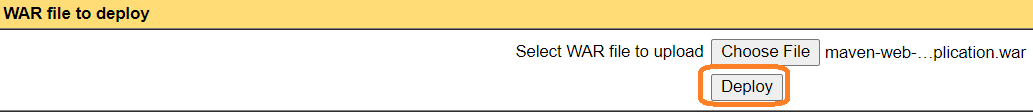
**Ctr + Shift + Delete**

**.War file Deployment:**

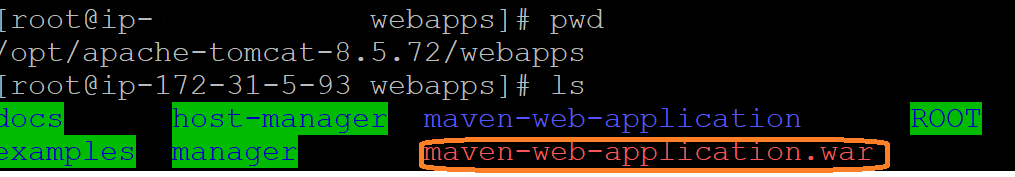
Click on **Manager App** button

****

Browse and upload .war file and Click on **Deploy** button



File will be upload into webapps directory



We can access application with the help of context URL, default context is .war file name

http://<public-ip-of-aws>:<port-number>/context-name

<http://15.206.151.64:8090/maven-web-application/>

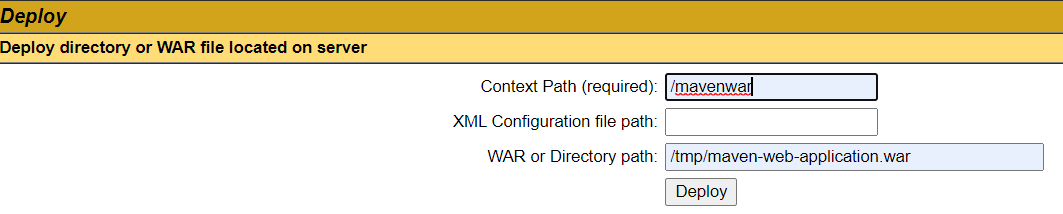
**Note:**

1. Not allowed to use same context path with multiple applications
2. Different context path with same application is possible

**Custom Context path:**

If we want deploy the application in shared directory

We have to copy .war file to tmp location which is accessible for every one



Context path: give custom context path

WAR or Directory path: Give complete path

Once, deployment success we will see below entry



Note:

1. If we change the .war file name context path will be change
2. Dev team will give context path and .war file details

**Other way to deploy:**

We can deploy application directly copying in web-apps folder

If .war is available in tmp or other accessible location we need to copy .war into webapp folder using cp command

**cp /tmp/maven-web-application.war .**

**Stop the specific application:**

If we click Stop button application will be stopped

****

1. Once we click stop button application will be down and we will access that application
2. We will get 404 exceptions

**Remove application from tomcat server:**

If we click on **Undeploy** application will be removed from tomcat server and webapp location

**Start and stop the tomcat from any folder**

ln -s /opt/apache-tomcat-8.5.72/bin/startup.sh /usr/bin/startTomcat

ln -s /opt/apache-tomcat-8.5.72/bin/shutdown.sh /usr/bin/stopTomcat

**How to increase the Heap size in Tomcat?**

Go to <Apache TomcatHomeDir>/bin

cd /opt/apache-tomcat>bin

Create a file with name setenv.sh and add below lines

vi setenv.sh

export CATALINA\_OPTS="$CATALINA\_OPTS -Xms512m"

export CATALINA\_OPTS="$CATALINA\_OPTS -Xms8192m"

export CATALINA\_OPTS="$CATALINA\_OPTS -XX:MaxPermSize=256m"

Note: The default -Xmx value is 64MB

Give the execute permissions

chmod u+x setenv.sh

Restart Tomcat server once u check the console out log file, you will see below lines as output

04-Jul-2019 12.125:526 org.apache.catalina.startup.VersionLoggerListener.logCommand Line argument:-Xmx512