Lab 1. Installation of Tomcat 8

Apache Tomcat is an open source Java-based web and servlet container, which is used to host Java-based applications. It was first developed for Jakarta Tomcat. Due to an increase in demand, it was later hosted as a separate project called Apache Tomcat, which is supported by The Apache Software Foundation. It was initially developed by James Duncan Davidson, a software architect at Sun Microsystems. He later helped make this project open source and played a key role in donating this project from Sun Microsystems to The Apache Software Foundation. Tomcat implements the **Java Servlet** and the **JavaServer Pages** (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run.

In this lab, we will discuss the following topics:

- Introduction to Tomcat 8
- Features of Tomcat 8
- Installation on Linux operating systems
- Common areas of troubleshooting during installation

History of Tomcat

Tomcat was first introduced to the open source group in 1999 and its first version was released with 3.0.x version. Since then, it has been greatly supported by the open source community and widely accepted in the IT industry. In the current scenario, Tomcat is running in production environments, as well as being used for mission-critical projects in various industries. The following mentioned details give us a quick history of the versions.

Web application memory leak detection and prevention

Tomcat had a chronological problem of memory leaks in 4.x/5.x versions. While reloading the applications in the entire life cycle of Tomcat, OutOfMemoryError exceptions were generated. Tomcat has put an exceptional effort in tracking down the bugs and issues related to memory, in order to avoid memory leaks.

Servlet 3.0

Tomcat 8 offers great support for Servlet 3.0. Servlet 3.0 helps developers to code very easily and also provides significant support for asynchronous programming techniques. The types of support provided are:

- **Asynchronous Support:**
- **Dynamic Configuration:**
- **Annotation-based Configuration:**

Improved logging

Tomcat 8 includes two new features for logging, in order to provide a good understanding to the users for log analysis:

- **Asynchronous file handler:** The asynchronous handler allows Tomcat to write logs to the disk by a dedicated thread, so that logging operations do not cause any delay in processing threads.
- **Single line log formatter:** The single line formatter writes logs in a single line, which is a better feature for administrators.

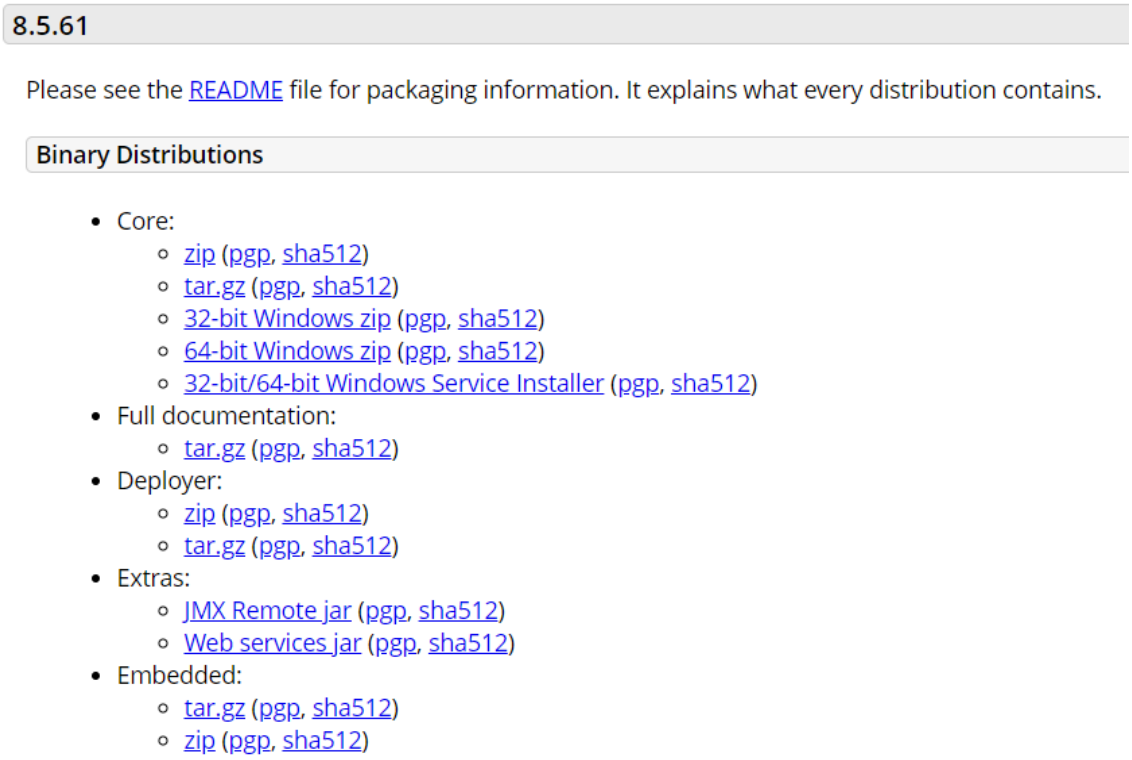
Installation of Tomcat 8

In the previous section, we have discussed the new enhancements in Apache Tomcat 8. Now, it's time to move on to the Tomcat installation.

How to download the Tomcat software

Perform the following steps to download the software:

- Before we start the installation of Apache Tomcat 8 software, the first thing that comes to mind is where can you download the software from and also how much does the license cost? By default, Apache comes with Apache License, Version 2.0 ,which is compatible to GPL (General Public License). In simple terms, it is free of cost! For more information on licenses, you can visit http://www.apache.org/licenses/. Now, the second problem is how to download the software.
- It is always recommended to download the software from its official site, http://tomcat.apache.org/download-80.cgi. By default, on http://tomcat.apache.org/, we get the latest stable version of Tomcat package and we have to download the package based on the operating system, where we want to install it.



Tomcat comes with different packages for installation such as binary, source, and RPM. Based on the requirement, the package should be taken from the official site. Let's have a brief discussion on which package should be implemented in real time and why.

Binary package

It comes with a pre-set library and customized configuration which are implemented and tested as per industry standards. A few advantages of using the binary package are:

- It is a standard package that suits most of the real-time environments
- In a non-DOS environment (such as Linux, UNIX, and so on), we can configure multiple Tomcat instances on a single OS
- It is path independent; we can configure Tomcat in any part of the OS based on our resources available (hardware)

RPM/exe

RPM is defined as a system installer, which is developed and compiled on each OS independently. It has a pre-defined library, which will work only on the respective OS. A few advantages of using RPM are:

- It does not require installation of any dependent libraries for the package

- RPM is built with the shared libraries for the respective OS
- It does not need to configure separate startup services

The only disadvantage is, we cannot configure multiple instances in a single operating system and it has predefined paths.

Source

You can customize the installation based on your requirements using the source package. Suppose you want to customize during installation of the software, it can be done in this package.

- Customization of Tomcat can be done very effectively (only required services are installed)
- In a non-DOS environment (such as Linux, UNIX, and so on), we can configure multiple Tomcat instances on a single OS
- It is path independent; we can configure Tomcat in any part of the OS based on our resources available (hardware)
- In a production environment, it's always recommended to use the source or binary instead of the RPM

Prerequisites for the Tomcat 8 installation

Following are the prerequisites mentioned for Apache Tomcat 8:

- Java Installed
- Configuration of the OS environment variables

JAVA_HOME and the PATH environment variable in Linux

In Linux, we can use the following command to verify the environment variables:

```
echo $VARIABLE_NAME
```

For `JAVA_HOME:`

```
[root@localhost ~]# echo $JAVA_HOME
/usr/lib/jvm/java-8-openjdk-amd64
```

For `PATH:`

```
[root@localhost ~]# echo $PATH
/usr/lib/jvm/java-8-openjdk-amd64/bin:/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/
sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
```

After verifying the environment variable on both the OSes, we are sure that `JAVA_HOME` and `PATH` are properly set in the environment. We have completed the prerequisites of installation of Apache Tomcat 8. Now, we can proceed with the installation of Apache Tomcat 8.

Installation of Apache Tomcat 8

Installation of Tomcat 8 is quite simple in a Linux environment. It can be done in just three steps:

Note: Setup apache-tomcat-8.5.61.zip has been downloaded and unzipped in /opt directory already.

1. Download the latest stable version from Tomcat's official site http://tomcat.apache.org/download-80.cgi. Once the download is complete, save it in the /opt location. Unzip the Tomcat 8 source, that is, apache-tomcat-8.5.61.zip using the following command:

   ```
   [root@localhost opt]# unzip apache-tomcat-8.5.61.zip
   ```

2. After you unzip the apache-tomcat-8.5.61.zip, it will create the directory named apache-tomcat-8.5.61 in the opt directory. Go to the bin directories of apache-tomcat-8.5.61 using the following command:

```
[root@localhost opt]# cd apache-tomcat-8.5.61/bin/
```

3. Run the following command. If you fail to run the following command, then Tomcat services will not come up. By default, the package comes with read/write permissions, but no execution permissions are given to the package. We have to manually change the permissions:

```
[root@localhost bin]# chmod 0755 *.sh
[root@localhost bin]# pwd
/opt/apache-tomcat-8.5.61/bin
```

Note

```
The` chmod 0755 file` is equivalent
to` u=rwx (4+2+1),go=rx (4+1 & 4+1)`. The` 0`
specifies no special modes.
```

After this step, the installation of Tomcat is complete in Linux.

Startup and shutdown of Tomcat services

Let us start the services on Linux to verify the installation.

Before that, let's quickly verify the configuration. Tomcat 8 comes with different scripts, through which we will verify the complete installation. There is a very good script placed in the Tomcat `bin` directory named as `version.sh`, through which we can verify the complete Tomcat version and system information. Let's run the script using the following command:

```
[root@localhost bin]# ./version.sh

Using CATALINA_BASE:   /opt/apache-tomcat-8.5.61
Using CATALINA_HOME:   /opt/apache-tomcat-8.5.61
Using CATALINA_TMPDIR: /opt/apache-tomcat-8.5.61/temp
Using JRE_HOME:        /usr/lib/jvm/java-8-openjdk-amd64
Using CLASSPATH:       /opt/apache-tomcat-8.5.61/bin/bootstrap.jar:/opt/apache-tomcat-
8.5.61/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Server version: Apache Tomcat/8.5.61
Server built:   Dec 3 2020 14:03:28 UTC
Server number:  8.5.61.0
OS Name:        Linux
OS Version:     4.18.0-193.28.1.el8_2.x86_64
Architecture:   amd64
JVM Version:    1.8.0_275-8u275-b01-0ubuntu1~20.04-b01
JVM Vendor:     Private Build
```

There is one more script in the Tomcat `bin` directory that is very useful. `configtest.sh` is used to check any configuration changes in scripts. This script performs a quick configuration check on the system and finds the errors. Let's run the script using the following command:

```
[root@localhost bin]# ./configtest.sh
Using CATALINA_BASE: /opt/apache-tomcat-8.5.61
Using CATALINA_HOME: /opt/apache-tomcat-8.5.61
Using CATALINA_TMPDIR: /opt/apache-tomcat-8.5.61/temp
Using JRE_HOME: /usr/lib/jvm/java-8-openjdk-amd64
Using CLASSPATH: /opt/apache-tomcat-8.5.61/bin/bootstrap.jar:/opt/apache-tomcat-
8.5.61/bin/tomcat-juli.jar
May 22, 2011 4:06:16 PM org.apache.coyote.AbstractProtocolHandler init
INFO: Initializing ProtocolHandler ["http-bio-8080"]
May 22, 2011 4:06:16 PM org.apache.coyote.AbstractProtocolHandler init
INFO: Initializing ProtocolHandler ["ajp-bio-8009"]
May 22, 2011 4:06:16 PM org.apache.catalina.startup.Catalina load
INFO: Initialization processed in 1401 ms
```

Note

`configtest.sh` is available in a Linux environment only.

After doing the configuration check, start the Tomcat services. The Tomcat services can be started using the `startup.sh` in the `bin` directory.

Startup script

To start the Tomcat services, you have to perform the following mentioned steps:

1. The first step is to change the directory from the current location to the Tomcat directory.

   ```
   [root@localhost bin]# cd /opt/apache-tomcat-8.5.61/bin/
   ```

2. In the bin directory, we will find the entire executable for Tomcat. To start the services, we have to use the following command. Once you execute the startup command, it will display the parameters which are essential for booting Tomcat. Some of them are CATALINA_BASE, CATALINA_HOME, JRE_HOME, and so on.

   ```
   [root@localhost bin]# ./startup.sh
   Using CATALINA_BASE: /opt/apache-tomcat-8.5.61
   Using CATALINA_HOME: /opt/apache-tomcat-8.5.61
   Using CATALINA_TMPDIR: /opt/apache-tomcat-8.5.61/temp
   Using JRE_HOME: /usr/lib/jvm/java-8-openjdk-amd64
   Using CLASSPATH: /opt/apache-tomcat-8.5.61/bin/bootstrap.jar:/opt/apache-
   tomcat-8.5.61/bin/ tomcat-juli.jar
   ```

Shutdown script

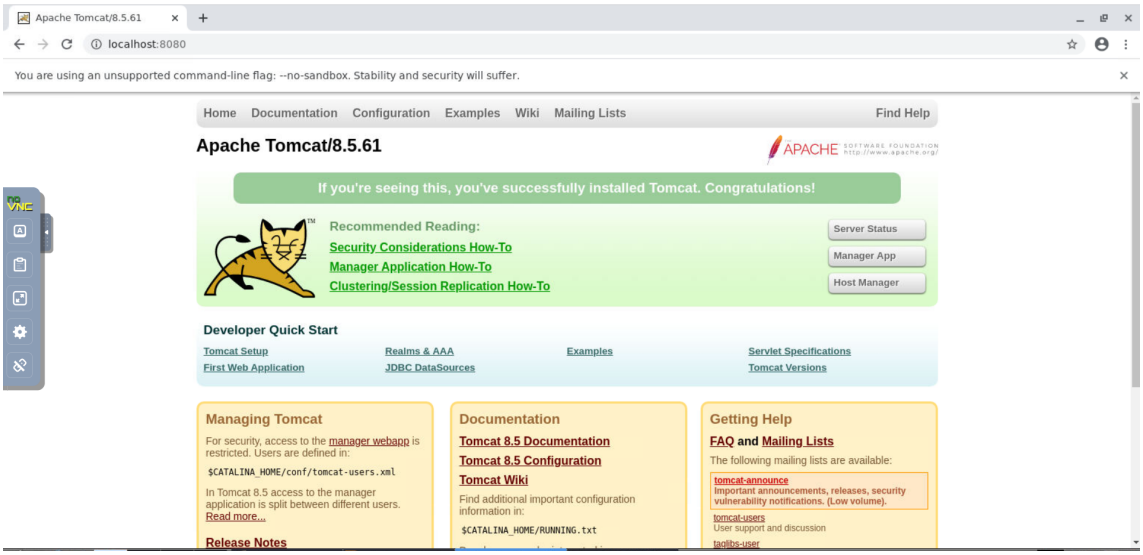
A Tomcat shutdown script is also available in the `bin` directory named as `./shutdown.sh`. Let's execute the script to know the output. The details are as follows:

```
[root@localhost bin]# cd /opt/apache-tomcat-8.5.61/bin/
[root@localhost bin]# ./shutdown.sh
Using CATALINA_BASE: /opt/apache-tomcat-8.5.61
Using CATALINA_HOME: /opt/apache-tomcat-8.5.61
Using CATALINA_TMPDIR: /opt/apache-tomcat-8.5.61/temp
Using JRE_HOME: /usr/lib/jvm/java-8-openjdk-amd64
```

```
Using CLASSPATH: /opt/apache-tomcat-8.5.61/bin/bootstrap.jar:/opt/apache-tomcat-
8.5.61/bin/tomcat-juli.jar
```

Verification of Tomcat status

Once we have executed the startup scripts, the next step is the verification of the Tomcat services, to check whether services are coming up fine or not. By default, Tomcat runs on HTTP port 8080 and can be accessed on the web browser using the URL, `http://localhost:8080`. We then find the Tomcat welcome page, which shows that Tomcat is installed correctly and running fine in the environment, as shown in the following screenshot:



Once the welcome page for Tomcat 8 is displayed, we can verify the server status by clicking on **Server Status.**

It will prompt for the user ID/password. Let's create a user admin that the user ID will be used here for access.

Tomcat users

Tomcat users are defined in the file – /opt/apache-tomcat-8.5.61/conf/tomcat-users.xml, by default, there is NO user, it means no one can access the Tomcat manager page.

To enable users to access the Tomcat manager page, add a user as the role manager-gui.

Original:

```
<tomcat-users>
<!--
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user username="both" password="tomcat" roles="tomcat,role1"/>
  <user username="role1" password="tomcat" roles="role1"/>
-->
</tomcat-users>
```

Updated:

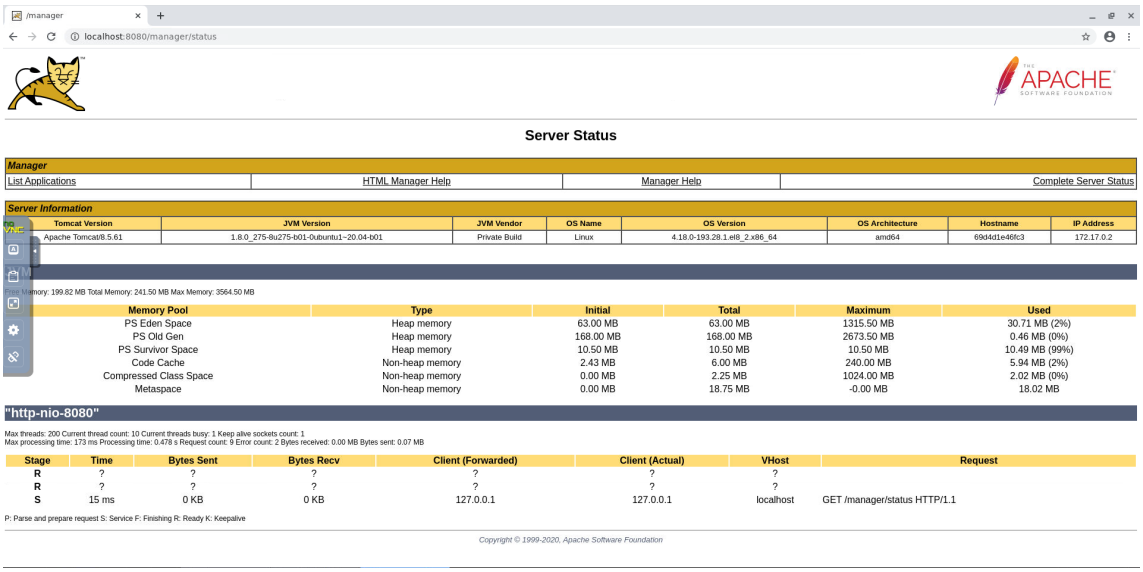
```
<tomcat-users>
<!--
```

```
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <user username="tomcat" password="tomcat" roles="tomcat"/>
  <user username="both" password="tomcat" roles="tomcat,role1"/>
  <user username="role1" password="tomcat" roles="role1"/>
-->

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

</tomcat-users>
```

Username: **admin**

Password: **admin**

Shutdown and start Tomcat again, now you should able to access the **Server Status.** page with user = “admin” and password = “admin”



Common problems and troubleshooting in installation

There are multiple issues which may arise during the installation of Tomcat 8. Let's discuss these issues:

Error: Tomcat is not able to find JAVA_HOME

Scenario 1: While starting the Tomcat startup script, the following error occurs:

```
[root@localhost bin]# ./startup.sh
Neither the JAVA_HOME nor the JRE_HOME environment variable is defined
At least one of these environment variables is needed to run this program
```

Fix: Check the `~/.bashrc` and find out whether the following mentioned entry is present in the file:

```
source /etc/environment
# JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

```
PATH=$JAVA_HOME/bin:$PATH:$HOME/bin
export PATH
```

Error: Error in the logs showing port already in use

Scenario 2: Tomcat services is not displayed after running `startup.sh`.

Issue: This service is already running on the server.

Fix: Check for any Java process running in the system using the following command in Linux:

```
ps -ef |grep tomcat
```

This command will show all Tomcat processes. If any process is running on an OS, kill it and run the startup scripts again.

Summary

In this lab, we have covered the Apache Tomcat history and new features introduced in Tomcat 8. We have done a step-by-step installation of Tomcat on Linux operating system.

In the next lab, we will discuss the various methods used for deployment in Tomcat 8 and solution of issues that may occur during the deployment process.