

Apache Tomcat 8





Table of Contents



1. Installation of Tomcat 8 :3
2. Configuration and Deployment: 62
3. Performance Tuning: 119
4. Integration of Tomcat with the Apache Web Server: 157
5. Securing Tomcat 8: 230
6. Logging in Tomcat 8: 270
7. Troubleshooting in Tomcat: 305
8. Monitoring and Management of Tomcat 8: 353
9. Clustering in Tomcat 8: 381
10. Tomcat Upgrade: 413
11. Advanced Configuration for Apache Tomcat 8: 439



1. Installation of Tomcat 8



Installation of Tomcat 8

In this lesson, we will discuss the following topics:

- Introduction to Tomcat 8
- Features of Tomcat 8
- Installation of Tomcat
- 1. Prerequisites for Tomcat 8 installation
- 2. Installation on Linux and Windows 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.



| Version | Release Date | Description |
|------------------------|------------------|---|
| 0.x. (initial release) | 1999 | Merger of donated Sun Java Web Server code and ASF and implements Servlet 2.2 and JSP 1.1 specifications. |
| 3.2 | March 9, 2004 | Latest 3.x release. |
| 1.40 | June 25, 2009 | Latest 4.x release. |
| 5.32 | February 1, 2011 | Latest 5.x release. |
| 0.32 | February 3, 2011 | Latest 6.x release. |
| 0.0 beta | June 29, 2010 | First Apache Tomcat release to support Servlet 3.0, JSP 2.2, and EL 2.2 specifications. |
| 0.11 | March 11, 2011 | Fourth stable version. |
| 0.12 | April 6, 2011 | Current stable version. |

Tomcat support matrix

- Apache Tomcat can be classified based on different components, such as the JDK version, enhancement, stability, and so on.
- Let's take a real time example, where you want to take a decision on which Apache Tomcat version to deploy for an application.
- For example, if an application is using Servlet 2.4 and JSP 2.0, then we should always go for the 5.x version.

| Features | X | X | X | X | X |
|--------------------------|------------------------------|-------------------------------|----------------------|----------------------|--------------------------|
| Version specifications | Servlet 3.0, JSP 2.2, EL 2.2 | Servlet 2.5, JSP 2.1 | Servlet 2.4, JSP 2.0 | Servlet 2.3, JSP 1.2 | Servlet 2.2, JSP 1.1 |
| Stable: | Yes | Yes | Yes | Yes | Yes |
| Enhancements | Yes | Yes | Unlikely | Highly unlikely | Highly unlikely |
| Bug Fixes | Yes | Yes | Yes | Highly unlikely | Highly unlikely |
| Security Fixes | Yes | Yes | Yes | Highly unlikely | Highly unlikely |
| Releases | Yes | Yes | Yes | Highly unlikely | Highly unlikely |
| Release Manager | Mark Thomas (markt) | Jean-Frederic Clere (jfclere) | Filip Hanik (fhanik) | Mark Thomas (markt) | Bill Barker (billbarker) |
| Process | CTR | RTC | RTC | CTR | CTR |
| Listed on download pages | Yes | Yes | Yes | No | No |
| JDK version | 6 | 5 | 4 | 3 | 1 |

Features and enhancements of Apache Tomcat 8

- In the previous section, we discussed the various support matrices for Tomcat versions, we are now aware of the support specifications (JDK support, EJB, and Servlet) for Tomcat.
- Let's try to understand, and quickly review the new features/enhancements for Tomcat 8.
- Apache Tomcat 8.x was released with some key improvements over Tomcat 6.x and real time implementation of Servlet 3.0, JSP 2.2, and EL 2.2 specifications.
- Apart from these, it also solves some major issues from previous releases.

How to download the Tomcat software

Binary Distributions

- Core:
 - [zip \(pgp, md5\)](#)
 - [tar.gz \(pgp, md5\)](#)
 - [32-bit Windows zip \(pgp, md5\)](#)
 - [64-bit Windows zip \(pgp, md5\)](#)
 - [64-bit Itanium Windows zip \(pgp, md5\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, md5\)](#)
- Full documentation:
 - [tar.gz \(pgp, md5\)](#)
- Deployer:
 - [zip \(pgp, md5\)](#)
 - [tar.gz \(pgp, md5\)](#)
- Extras:
 - [JMX Remote jar \(pgp, md5\)](#)
 - [Web services jar \(pgp, md5\)](#)
 - [JULI adapters jar \(pgp, md5\)](#)
 - [JULI log4j jar \(pgp, md5\)](#)
- Embedded:
 - [tar.gz \(pgp, md5\)](#)
 - [zip \(pgp, md5\)](#)

Source Code Distributions

- [tar.gz \(pgp, md5\)](#)

Installation of Tomcat 8

Apache Tomcat



- Download the MD5 checksum from the website apache.org.
- The MD5 check sum is integrated with every package we download.
- The previous screenshot shows the packages which are available in MD5, and by clicking on md5 on the website, we can compare the checksum generated in our system with the value given on the site.

Installation of Tomcat 8

- Run the following command to generate the checksum for the downloaded software. See the following screenshot:

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

Prerequisites for the Tomcat 8 installation

Before we begin with the Apache Tomcat 8 installation, we have to configure the prerequisites and they are very important for the Tomcat 8 installation to start. Following are the prerequisites mentioned for Apache Tomcat 8:

- Java SE 1.6 or later
- Configuration of the OS environment variables

Installation of Tomcat 8

JDK/JRE comes in 32 bit and 64-bit editions, so we can use it based on the application requirement. Some of the performance characteristics of the 64 bit versus 32-bit Virtual Machine (VM) are:

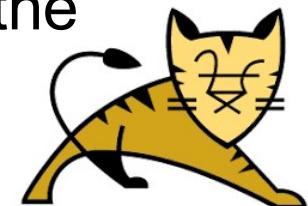
- The benefits of using 64 bit are being able to address larger amounts of memory which comes with a small performance loss in 64-bit VMs, versus running the same application on a 32-bit VM
- You can allocate more than 4 GB to JVM for memory intensive applications

Installation of Java in Linux

In this topic, we will discuss the steps performed during installation of Java on Linux:

- Download the JDK from the Oracle site on the Linux system.
- Change the permission of the package using the following command:

`chmod 0755 jdk-6u24-linux-i586.bin`



- Run the following command to install the JDK:

[root@localhost opt]# ./jdk-6u24-linux-i586.bin

- You will see an output like the following screenshot:

```
[root@localhost opt]# ./jdk-6u24-linux-i586.bin
Unpacking...
Checksumming...
Extracting...
UnZipSFX 5.50 of 17 February 2002, by Info-ZIP (Zip-Bugs@lists.wku.edu).
  creating: jdk1.6.0_24/
  creating: jdk1.6.0_24/jre/
  creating: jdk1.6.0_24/jre/bin/
  inflating: jdk1.6.0_24/jre/bin/java
  inflating: jdk1.6.0_24/jre/bin/keytool
  inflating: jdk1.6.0_24/jre/bin/policytool
  inflating: jdk1.6.0_24/jre/bin/rmiregistry
  inflating: jdk1.6.0_24/jre/bin/rmid
  inflating: jdk1.6.0_24/jre/bin/tnameserv
  inflating: jdk1.6.0_24/jre/bin/orbd
  inflating: jdk1.6.0_24/jre/bin/servrtool
  inflating: jdk1.6.0_24/jre/bin/unpack200
  inflating: jdk1.6.0_24/jre/bin/pack200
  inflating: jdk1.6.0_24/jre/bin/jcontrol
    linking: jdk1.6.0_24/jre/bin/ControlPanel  -> ./jcontrol
  inflating: jdk1.6.0_24/jre/bin/java_vm
  inflating: jdk1.6.0_24/jre/bin/javaws
  creating: jdk1.6.0_24/jre/lib/
  creating: jdk1.6.0_24/jre/lib/applet/
  creating: jdk1.6.0_24/jre/lib/ext/
  inflating: jdk1.6.0_24/jre/lib/ext/sunjce_provider.jar
  inflating: jdk1.6.0_24/jre/lib/ext/sunpkcs11.jar
  inflating: jdk1.6.0_24/jre/lib/ext/dnsns.jar
  inflating: jdk1.6.0_24/jre/lib/ext/localesdata.pack
  inflating: jdk1.6.0_24/jre/lib/ext/meta-index
  creating: jdk1.6.0_24/jre/lib/i386/
  creating: jdk1.6.0_24/jre/lib/i386/native_threads/
  inflating: jdk1.6.0_24/jre/lib/i386/native_threads/libhpi.so
  creating: jdk1.6.0_24/jre/lib/i386/server/
  inflating: jdk1.6.0_24/jre/lib/i386/server/libi18n.so
```

Java(TM) SE Development Kit 6 successfully installed.

Product Registration is FREE and includes many benefits:

- * Notification of new versions, patches, and updates
- * Special offers on Oracle products, services and training
- * Access to early releases and documentation

Product and system data will be collected. If your configuration supports a browser, the JDK Product Registration form will be presented. If you do not register, none of this information will be saved. You may also register your JDK later by opening the register.html file (located in the JDK installation directory) in a browser.

For more information on what data Registration collects and how it is managed and used, see:

<http://java.sun.com/javase/registration/JDKRegistrationPrivacy.html>

Press Enter to continue.....



Installation of Tomcat 8

- After pressing Enter it will exit from the command prompt, as shown in the following screenshot:

```
Press Enter to continue.....
```

```
Done.
```

```
[root@localhost opt]#
```



Installation of Tomcat 8

```
root@localhost jdk1.6.0_24]# ls -ltrh
total 19M
-rw-r--r-- 1 root root 19M Feb  2 17:30 exec.zip
r--r--r-- 1 root root  76 Feb  2 17:30 THIRDPARTYLICENSEREADME.txt
r--r--r-- 1 root root 21K Feb  2 17:30 README_zh_CN.html
r--r--r-- 1 root root 25K Feb  2 17:30 README_ja.html
r--r--r-- 1 root root 28K Feb  2 17:30 README.html
r--r--r-- 1 root root 3.3K Feb  2 17:30 COPYRIGHT
drwxr-xr-x 2 root root 4.0K Feb  2 19:36 bin
drwxr-xr-x 10 root root 4.0K Feb  2 19:36 conf
drwxr-xr-x 9 root root 4.0K Feb  2 19:36 examples
drwxr-xr-x 4 root root 4.0K Feb  2 19:36 lib
drwxr-xr-x 3 root root 4.0K Feb  2 19:36 logs
drwxr-xr-x 7 root root 4.0K Feb  2 19:36 temp
drwxr-xr-x 3 root root 4.0K May 16 20:34 webapps
drwxr-xr-x 7 root root 4.0K May 16 20:34 work
r--r--r-- 1 root root 4.8K May 16 20:35 register_zh_CN.html
r--r--r-- 1 root root 6.6K May 16 20:35 register_ja.html
r--r--r-- 1 root root 5.2K May 16 20:35 register.html
```

Installation of Java in Windows

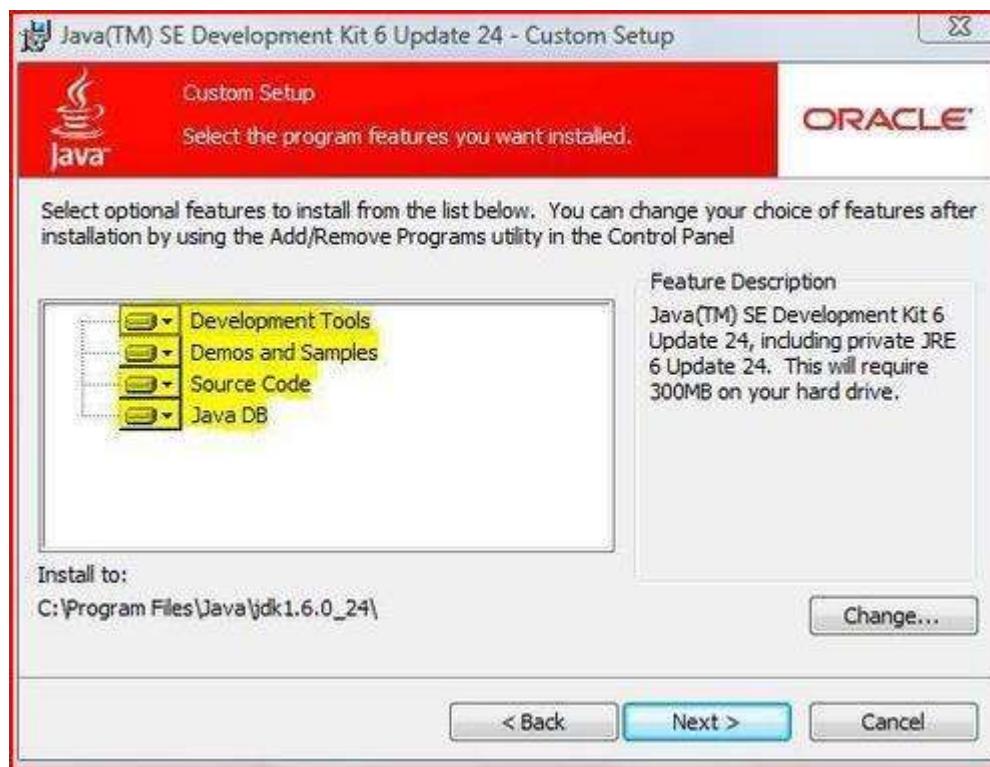
In this topic, we will discuss the steps performed during installation of Java in the Windows operating system:

- Download the JDK from the Oracle site on the Windows system from the following link:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>.
- Once the download is complete, jdk-6u24-windows-i586.exe is created in the download location.

Installation of Tomcat 8



- Check the field Installation type: Default



Installation of Tomcat 8



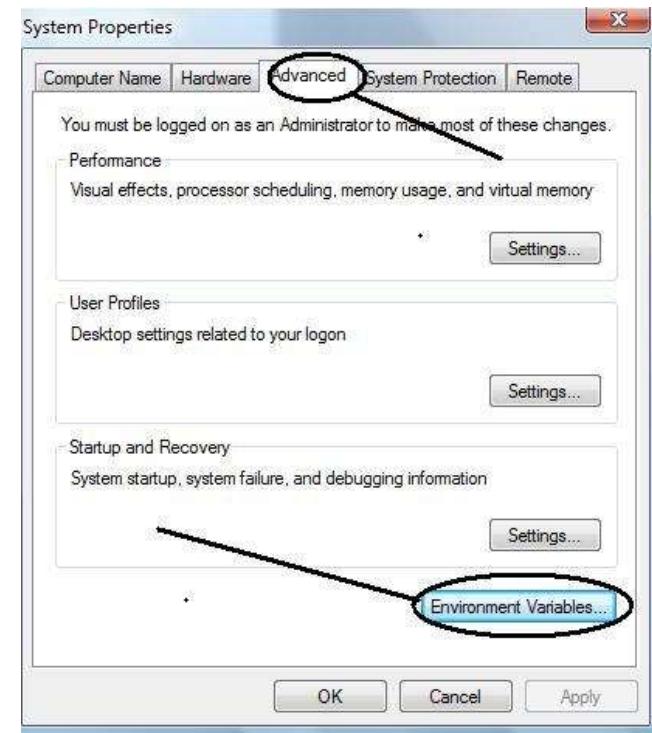
What is JAVA_HOME?

- In a production environment, it is always recommended to use the permanent environment variable.

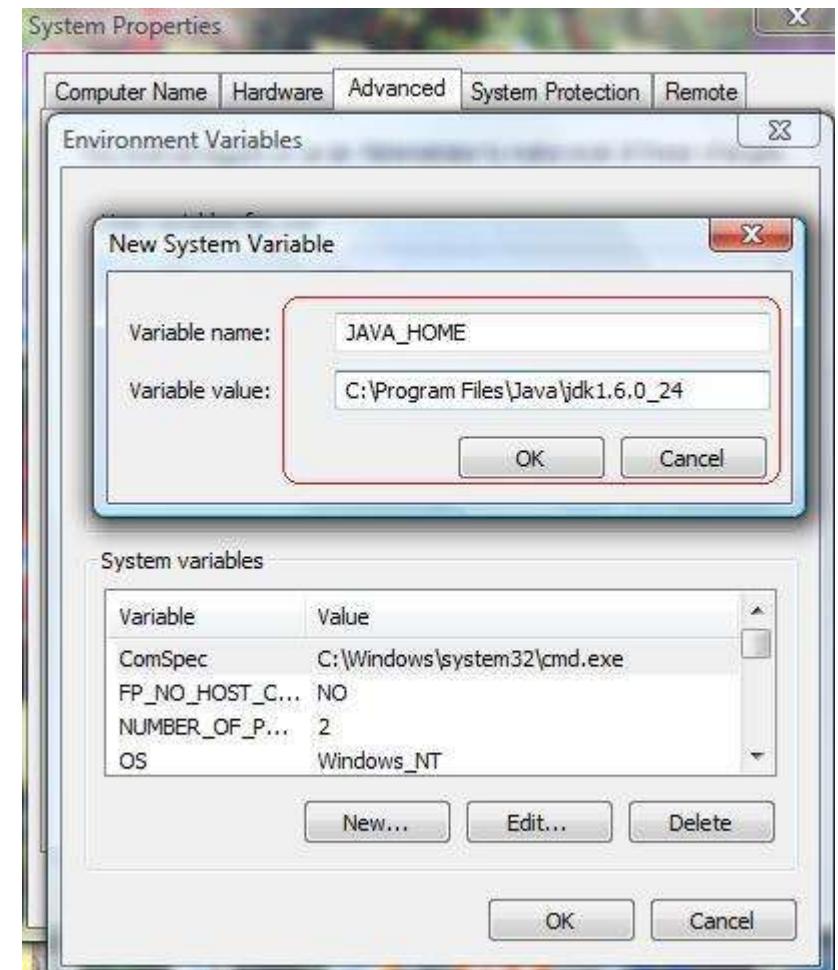
```
# For Cygwin, ensure paths are in UNIX format before anything is touched
if $cygwin; then
    [ -n "$JAVA_HOME" ] && JAVA_HOME=`cygpath --unix "$JAVA_HOME"`
    [ -n "$JRE_HOME" ] && JRE_HOME=`cygpath --unix "$JRE_HOME"`
    [ -n "$CATALINA_HOME" ] && CATALINA_HOME=`cygpath --unix "$CATALINA_HOME"`
    [ -n "$CATALINA_BASE" ] && CATALINA_BASE=`cygpath --unix "$CATALINA_BASE"`
    [ -n "$CLASSPATH" ] && CLASSPATH=`cygpath --path --unix "$CLASSPATH"`
fi
```

Setting the JAVA_HOME and PATH variable in Windows

- Right-click on the My Computer icon on your desktop and then click Properties, as shown in the following screenshot:

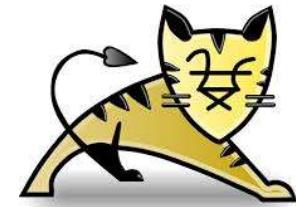


- In the Variable name field, enter JAVA_HOME, as shown in the following screenshot:

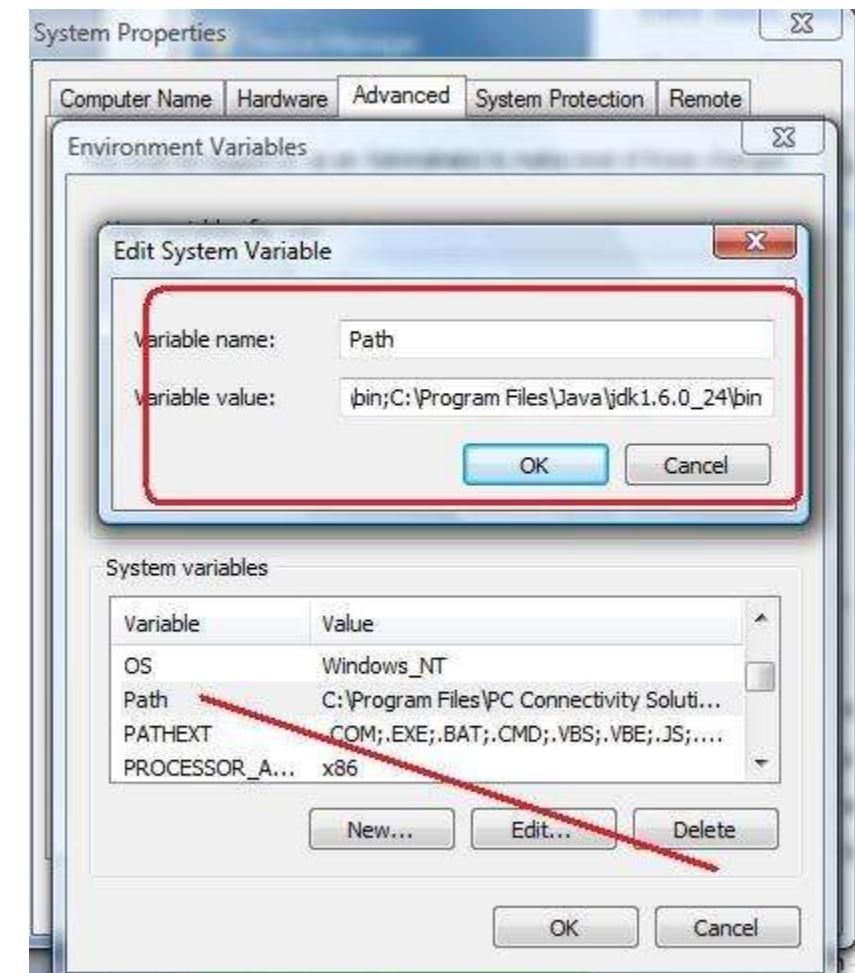


Setting the global path variable in Windows

- Right-click on the My Computer icon on your desktop and then click Properties.
- Click Advanced | Environment Variables.
- Under System Variables, click on Path.



- Edit the path and add the Java path in the end.



Setting the JAVA_HOME and the PATH environment variable in Linux

The environment variable and path are set differently in Linux as compared to Windows. Perform the following steps to set the environment variable in Linux:

- Open the .bash_profile using the vi editor for the root user.



Installation of Tomcat 8



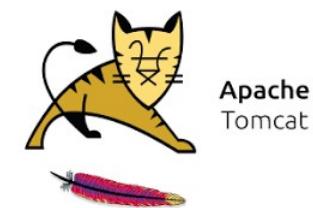
```
# .bash_profile
# Get the aliases and functions
if [ -f ~/.bashrc ];then . ~/.bashrc
fi
# User specific environment and start-up programs
JAVA_HOME=/opt/jdk1.6.0_24
PATH=$JAVA_HOME/bin:$PATH:$HOME/bin
export PATH JAVA_HOME
unset USERNAME
```

Installation of Tomcat 8

- Save the .bash_profile using the :wq command.
- Once you have saved the .bash_profile, then you have to logout and re-log in to the environment to activate the changes using the following command:

su - username

su - root (as our user is root)



- Also, you can run the env command to verify the environment variables are configured properly, as shown in the following screenshot

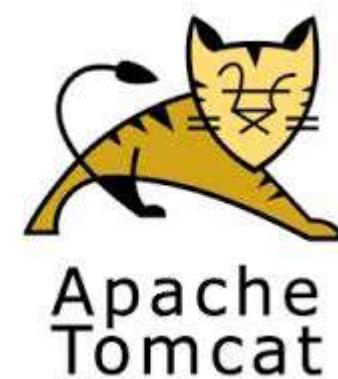
Installation of Tomcat 8

- In the Windows environment, variables can be verified using the following command:

echo %VARIABLE_NAME%

- For JAVA_HOME:

C:\Users\user>echo %JAVA_HOME%
C:\Program Files\Java\jdk1.6.0_24



Installation of Tomcat 8

- For PATH:

```
C:\Users\user>echo %PATH%
C:\Program Files\PC Connectivity
Solution\;C:\Windows\system32;C:\Windows;C:\Window
s\System32\Wbem;C:\ Program
Files\Broadcom\Broadcom 802.11\Driver;
C:\Program Files\Java\jdk1.6.0_24\bin
```

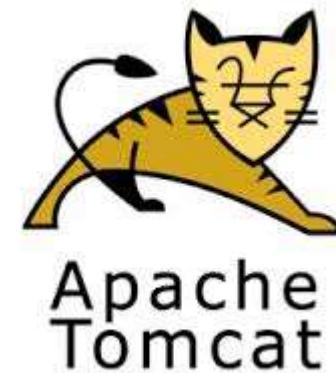
Installation of Tomcat 8

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

```
echo $VARIABLE_NAME
```

- For JAVA_HOME:

```
[root@localhost ~]# echo $JAVA_HOME  
/opt/jdk1.6.0_24
```



Installation of Tomcat 8

- For PATH:

```
[root@localhost ~]# echo $PATH  
/opt/jdk1.6.0_24/bin:/usr/kerberos/sbin:/usr/kerberos/bin  
:/usr/local/  
sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
```

Installation on a Windows environment

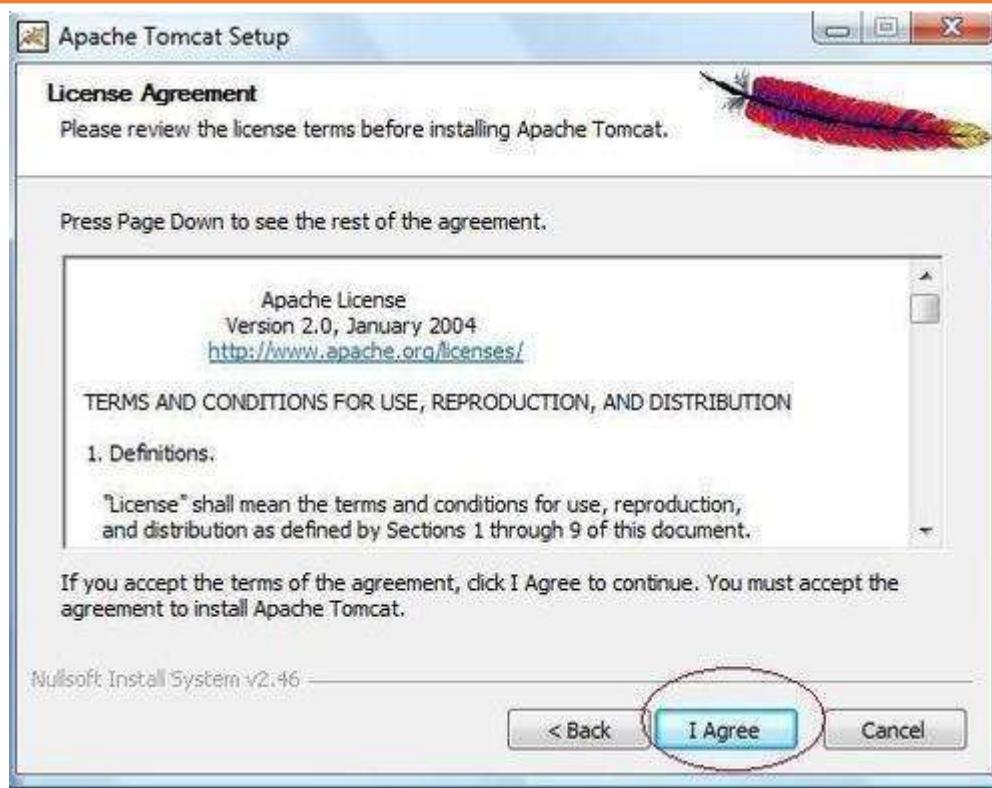
In this topic, we will discuss the steps involved during the software installation of Tomcat 8. Following are the steps:

- Download the latest stable version from the Tomcat official site, <http://tomcat.apache.org/download-70.cgi>. We are downloading the 32-bit/64-bit Windows Service Installer (pgp, md5). Once the download is complete, save it in the software folder.
- Double-click on apache-tomcat-7.0.14.exe. It will launch the setup wizard.

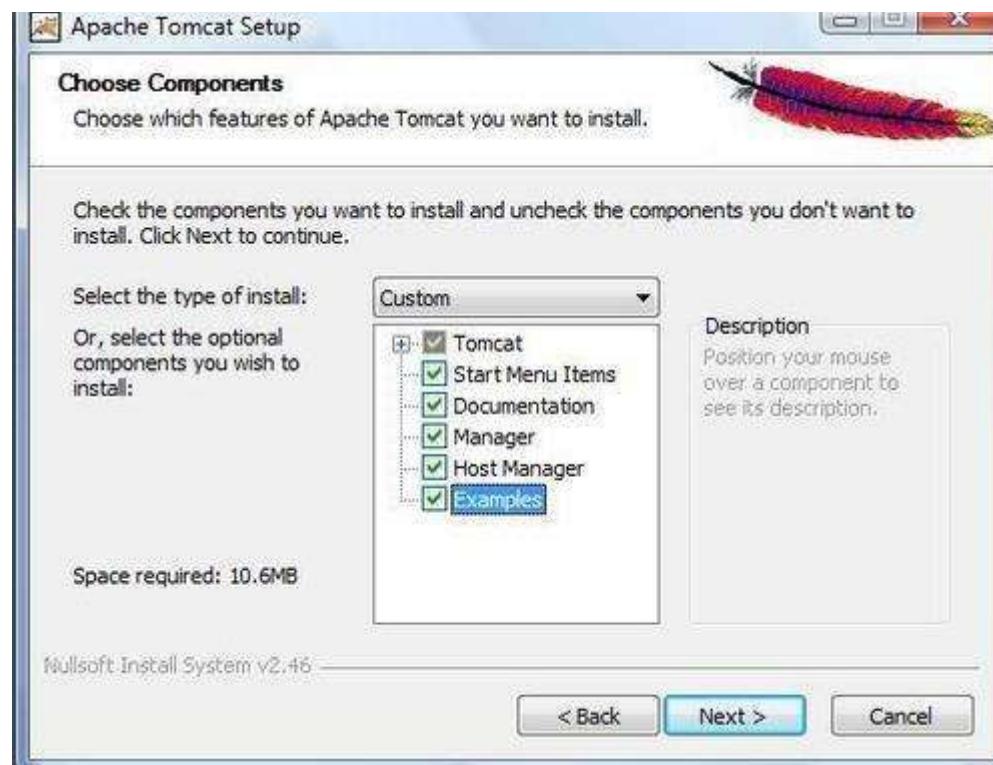
- Click on Next button to continue, as shown in the following screenshot



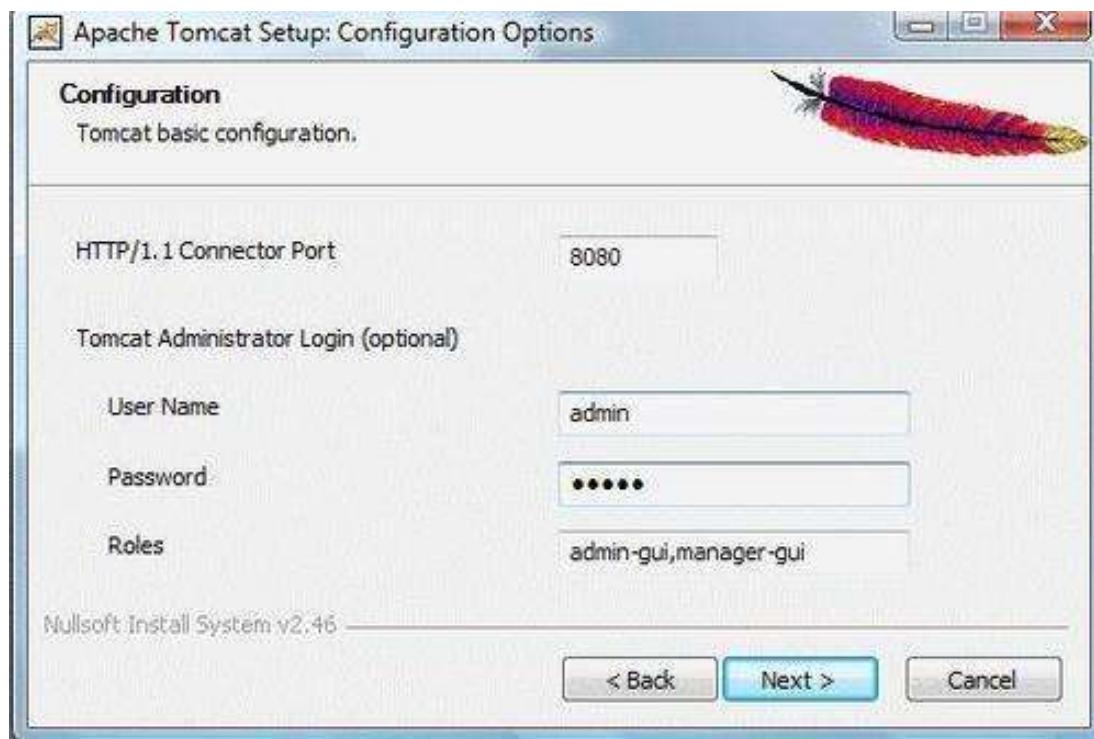
Installation of Tomcat 8



- The following window shows us the different components we need to install:

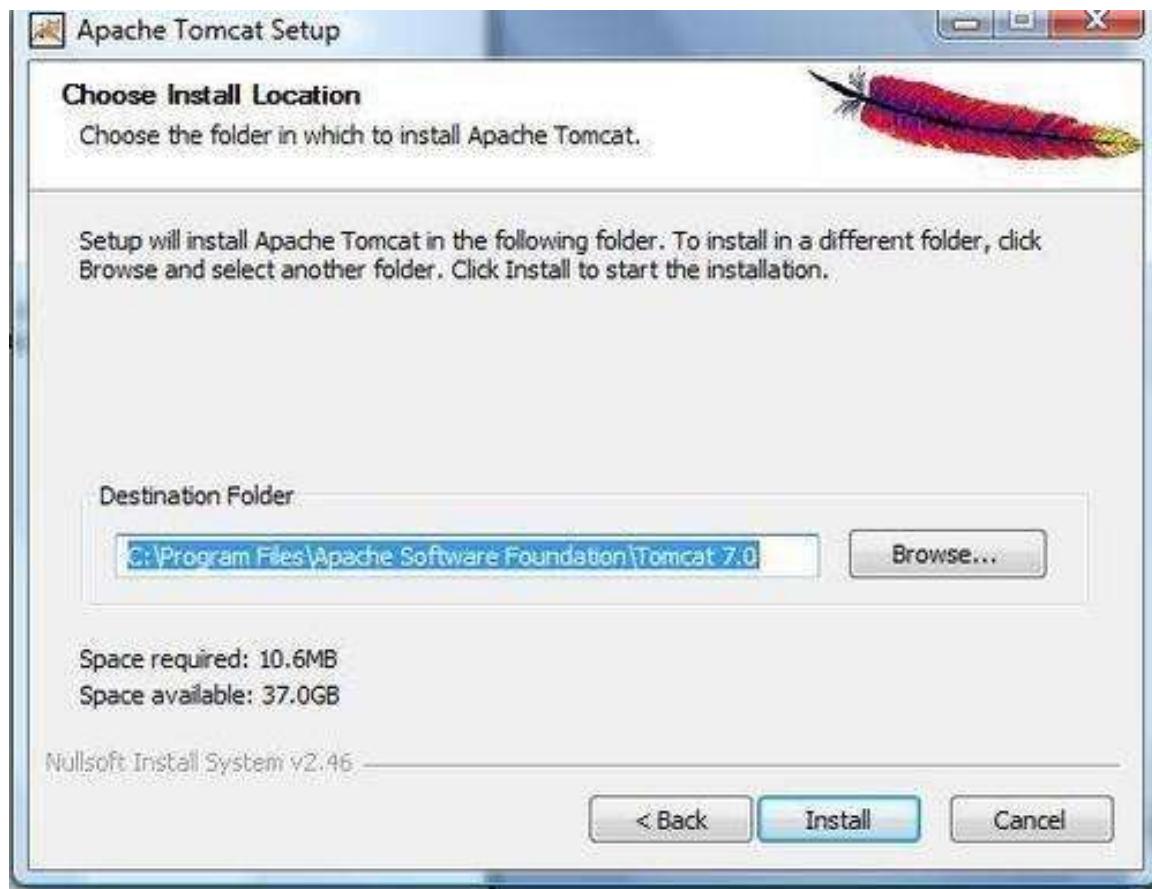


Installation of Tomcat 8

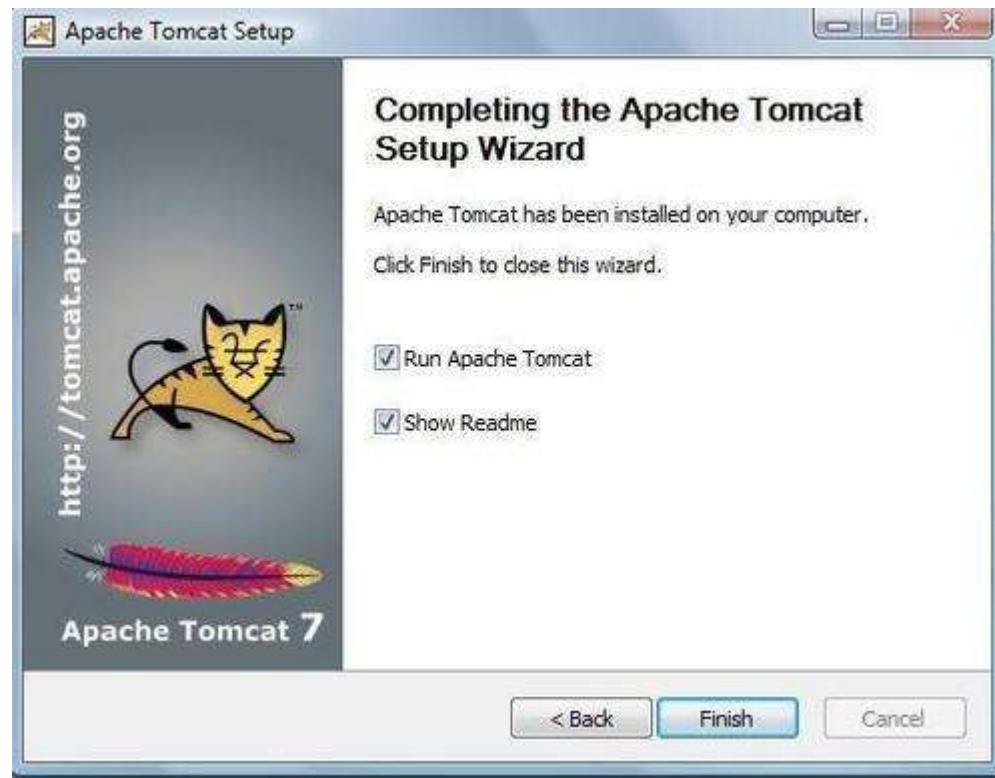


Installation of Tomcat 8





Installation of Tomcat 8

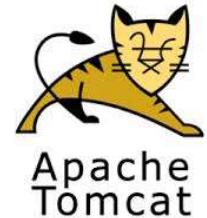


Installation on a Linux environment

- Download the latest stable version from Tomcat's official site <http://tomcat.apache.org/download-70.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
```

Installation of Tomcat 8



- 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:

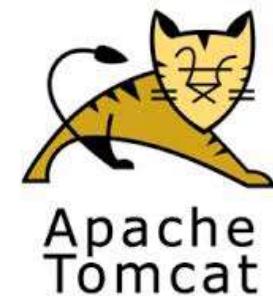
```
[root@localhost opt]# cd apache-tomcat-8.5.61/bin/
```

Installation of Tomcat 8

- 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 must manually change the permissions:

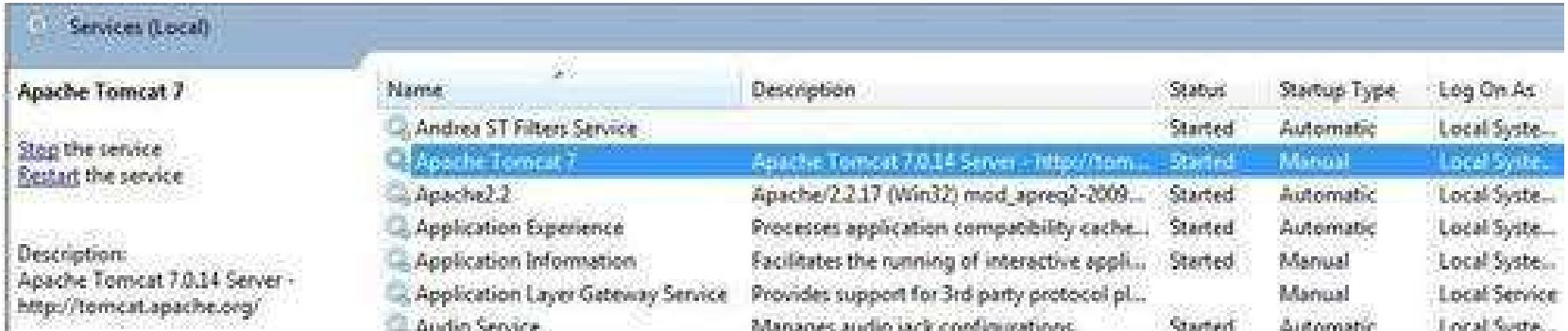
```
[root@localhost bin]# chmod 0755 *.sh
```

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



Services in Windows

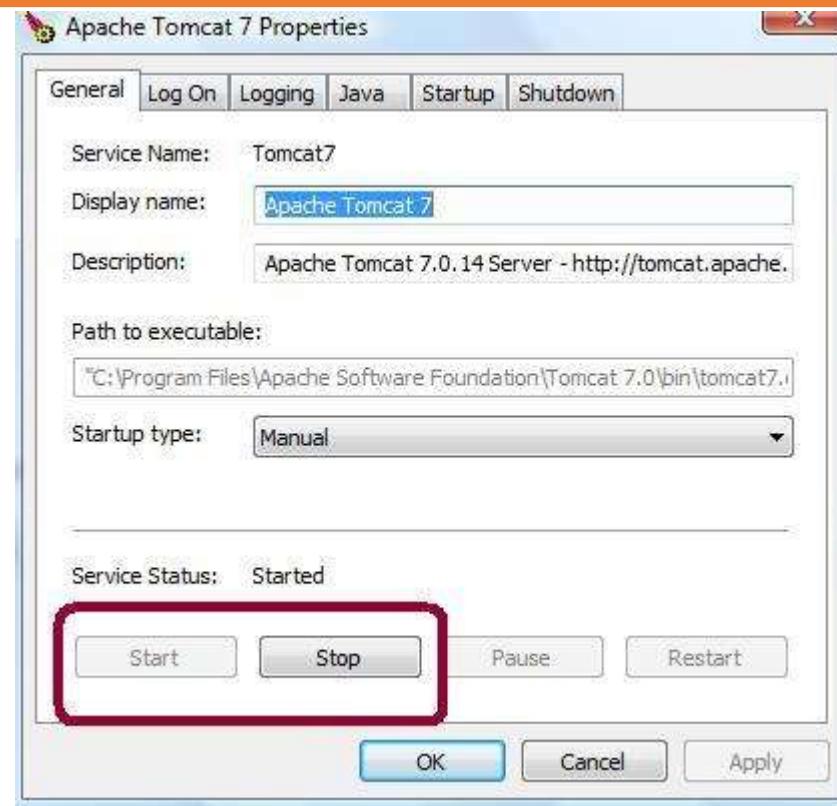
- Through the Microsoft Management Console (MMC): Go to Start | Run | services.msc.
- When the MMC opens, as shown in the following screenshot, you can start/stop services based on the requirement:



The screenshot shows the Windows Services (Local) console. On the left, there is a context menu for the 'Apache Tomcat 7' service with options 'Stop the service' and 'Restart the service'. Below the menu, a 'Description:' label is followed by the text 'Apache Tomcat 7.0.14 Server - http://tomcat.apache.org/'. The main table lists various services with their details:

| Name | Description | Status | Startup Type | Log On As |
|-----------------------------------|---|---------|--------------|---------------|
| Andrea ST Filter Service | | Started | Automatic | Local System |
| Apache Tomcat 7 | Apache Tomcat 7.0.14 Server - http://tomcat.apache.org/ | Started | Manual | Local System |
| Apache2.2 | Apache/2.2.17 (Win32) mod_apreq2-2009... | Started | Automatic | Local System |
| Application Experience | Processes application compatibility cache... | Started | Automatic | Local System |
| Application Information | Facilitates the running of interactive appli... | Started | Manual | Local System |
| Application Layer Gateway Service | Provides support for 3rd party protocol pl... | Started | Manual | Local Service |
| Audio Service | Manages audio jack configurations. | Started | Automatic | Local System |

Apache monitor console



Services in Linux

- The Linux startup process is completely different. Here, we have to run the startup/shutdown scripts manually to bring the services online.
- 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.

- 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: /opt/jdk1.6.0_24
Using CLASSPATH: /opt/apache-tomcat-
8.5.61/bin/bootstrap.jar:/opt/apache-tomcat-8.5.61/bin/tomcat-juli.jar
Server version: Apache Tomcat/8.5.61
Server built: Apr 1 2011 06:13:02
Server number: 8.5.61.0
OS Name: Linux
OS Version: 2.6.18-8.el5
Architecture: i386
JVM Version: 1.6.0_24-b07
JVM Vendor: Sun Microsystems Inc.
```



```
[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: /opt/jdk1.6.0_24
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
```

Startup script

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

- 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/
```

Installation of Tomcat 8

```
[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: /opt/jdk1.6.0_24
Using CLASSPATH: /opt/apache-tomcat-
8.5.61/bin/bootstrap.jar:/opt/apache-tomcat-8.5.61/bin/
tomcat-juli.jar
```

Shutdown script

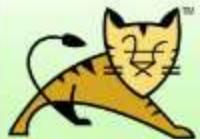
```
[root@localhost bin]# cd /opt/apache-tomcat-8.5.61/bin/  
[root@localhost bin]# ./shutdown.sh  
Using CATALINA_BASE: /opt/apache-tomcat-8.5.61  
Using CATALINA_HOME: /opt/apache-tomcat-8.5.61  
Using CATALINA_TMPDIR: /opt/apache-tomcat-8.5.61/temp  
Using JRE_HOME: /opt/jdk1.6.0_24  
Using CLASSPATH: /opt/apache-tomcat-  
8.5.61/bin/bootstrap.jar:/opt/apache-tomcat-  
8.5.61/bin/tomcat-juli.jar
```

Verification of Tomcat status

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/8.5.61

If you're seeing this, you've successfully installed Tomcat. Congratulations!



Recommended Reading:

- [Security Considerations How-To](#)
- [Manager Application How-To](#)
- [Clustering/Session Replication How-To](#)

Server Status Manager App Host Manager

Developer Quick Start

Tomcat Setup First Web Application Realms & AAA JDBC DataSources Examples Servlet Specifications Tomcat Versions

Managing Tomcat
For security, access to the `manager.webapp` is restricted. Users are defined in:
`$CATALINA_HOME/conf/tomcat-users.xml`
In Tomcat 8.5 access to the manager application is split between different users.
[Read more...](#)

Release Notes
[Changelog](#)

Documentation
[Tomcat 8.5 Documentation](#)
[Tomcat 8.5 Configuration](#)
[Tomcat Wiki](#)
Find additional important configuration information in:
`$CATALINA_HOME/RUNNING.txt`
Developers may be interested in:

Getting Help
FAQ and Mailing Lists
The following mailing lists are available:
[tomcat-announce](#)
Important announcements, releases, security vulnerability notifications. (Low volume).
[tomcat-users](#)
User support and discussion
[taglibs-user](#)
User support and discussion for Apache Taglibs

http://localhost:8080/manager/status

Apache Page Status News E-mail Weather

The Apache Software Foundation http://www.apache.org/

Server Status

Manager List Applications HTML Manager Help Manager Help Complete Setup

Server Information

| Tomcat Version | JVM Version | JVM Vendor | OS Name | OS Version | OS Architecture |
|----------------------|--------------|-----------------------|---------------|------------|-----------------|
| Apache Tomcat/7.0.14 | 1.6.0_24-b07 | Sun Microsystems Inc. | Windows Vista | 6.0 | x86 |

JVM

free memory: 3.64 MB Total memory: 15.96 MB Max memory: 247.56 MB

ajp-bio-8009

No threads: 200 Current thread count: 0 Current thread busy: 0
Free processing time: 0 ms Processing time: 0.0 s Request count: 0 Error count: 0 Bytes received: 0.00 MB Bytes sent: 0.00 MB

| Stage | Time | B Sent | B Recv | Client | VHost | Request |
|-------|------|--------|--------|--------|-------|---------|
|-------|------|--------|--------|--------|-------|---------|

← Back to Apache Manager → Remove this instance ← Back to Manager

Error: Permission denied for the Java binary

- **Scenario 1:** The Java installation is not working, while executing the Java binary.

```
[root@localhost opt]# ./jdk-6u24-linux-i586.bin  
-bash: ./jdk-6u24-linux-i586.bin: Permission denied
```

- **Issue:** The Java binary doesn't have execute permissions with a specific user.
- **Fix:** Change the permission to 0755 for ./jdk-6u24-linux-i586.bin using the following command:
`chmod 0755 jdk-6u24-linux-i586.bin`

Error: Tomcat is not able to find JAVA_HOME

- **Scenario 2:** While starting the Tomcat startup script, the following error occurs:

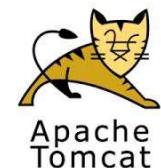
```
[root@localhost bin]# ./startup.sh
```

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 .bash_profile and find out whether the following mentioned entry is present in the file:

```
JAVA_HOME=/opt/jdk1.6.0_24  
PATH=$JAVA_HOME/bin:$PATH:$HOME/bin  
export PATH JAVA_HOME
```



Error: Error in the logs showing port already in use

- **Scenario 3:** 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

Summary



- In this lesson, 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 Windows and Linux operating systems.
- Also, we have discussed the common problems that may arise during the installation and their possible solutions.

2. Configuration and Deployment



Configuration and Deployment

In this lesson, we will discuss the following topics:

- Configuration of Tomcat
- Configuration of the virtual directory
- Deployment of an application on Tomcat 8



Configuration files and their usage

- Apache Tomcat 8 comes with a default setup, which can be directly used for a QA environment.
- We can customize Tomcat based on the environment specification; components such as Services, Servers, Engine, Connectors, Realm, and Valve can be configured.
- The Tomcat configuration files are available in the conf folder.
- Let's discuss the configuration properties and their usage.

Configuration files and their usage

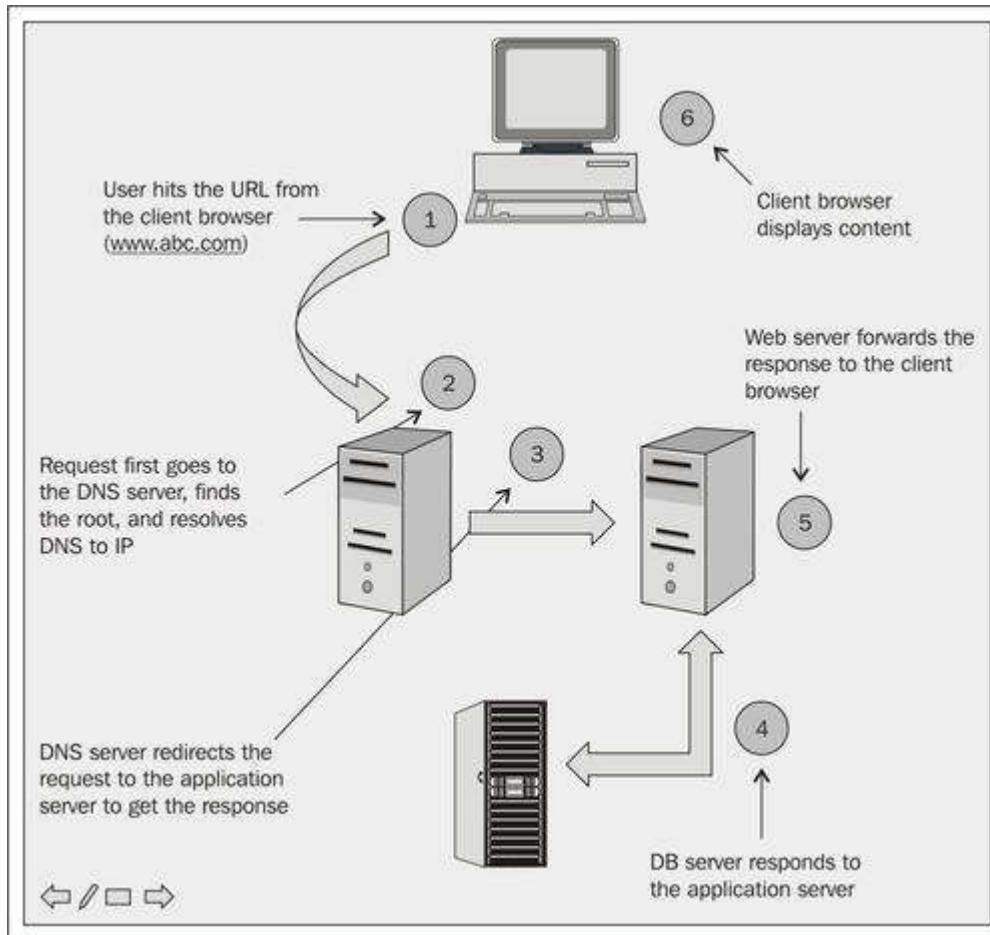
- The following screenshot shows the directory structure of the configuration directory for Tomcat 8:

```
root@localhost conf]# ls -l
total 136
drwxr-xr-x 3 root root 4096 May 16 21:03 .
rwxr--r-- 1 root root 11888 Apr  1 18:15 catalina.policy
rwxr--r-- 1 root root 5089 Apr  1 18:15 catalina.properties
rwxr--r-- 1 root root 1428 Apr  1 18:15 context.xml
rwxr--r-- 1 root root 3213 Apr  1 18:15 logging.properties
rwxr--r-- 1 root root 6645 Apr  1 18:15 server.xml
rwxr--r-- 1 root root 1566 Apr  1 18:15 tomcat-users.xml
rwxr--r-- 1 root root 53273 Apr  1 18:15 web.xml
```



Configuration of Tomcat 8

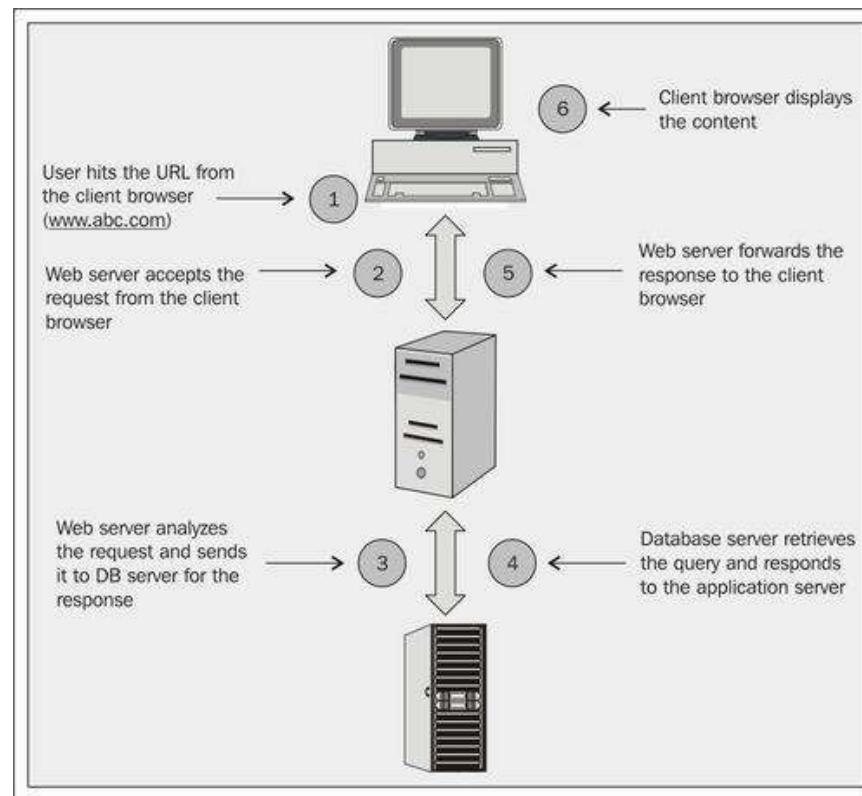
1. Whenever you hit the URL (for example, www.abc.com), the browser will contact the DNS server.
2. The DNS server will contact the ISP for the required information.
3. Once the web server accepts the request from the client browser, it will redirect it to the database server.
4. In turn, the database server will retrieve the query and respond it back to the web server.
5. The web server then forwards the same response to the client browser, and finally, the client browser will display the content to the user.



Data Source configuration

1. Whenever you hit the URL (for example, www.abc.com), the request goes to the web server.
2. Once the web server accepts the request from the client browser, it will analyze the request based on the query. If it requires the database (DB) response, then it redirects the request to the database server.
3. Based on the query, the database server will retrieve the content and respond to the web server. The web server then forwards the response from the database server to the client browser.

This process flow is also explained in the following figure:



Data Source

The following are the parameters required for any database server to connect Tomcat 8 with the database and are also the prerequisites for data source configuration:

- IP address
- Port number
- JNDI name
- Database user ID/password



Data Source

- Database Connection Pool (DBCP) configuration is located in the TOMCAT_HOME or CATALINA_HOME/lib/tomcat-dbcp.jar.
- This specific JAR is responsible for connection pooling.
- The following screenshot shows the location of tomcat-dbcp.jar.
- The following are the built-in properties of the Tomcat 8 server for accomplishing a connection with the database:

Database Connection pool

Common DBCP properties

```
[root@localhost lib]# cd /opt/apache-tomcat-7.0.12/lib/
[root@localhost lib]# ls -l tomcat-dbcp.jar
-rw-r--r-- 1 root root 234639 Apr  1 18:15 tomcat-dbcp.jar
[root@localhost lib]#
```



DataSource for Oracle

- By default, the definition of datasource values are defined in the global section of server.xml.
- The following screenshot shows the datasource details in server.xml:

```
<!-- Global JNDI resources Documentation at /docs/jndi-resources-howto.html-->
<GlobalNamingResources>
    <!-- Editable user database that can also be used by UserDatabaseRealm to
        authenticate users-->
    <Resource name="jdbc/tomcat8" auth="Container"
        type="javax.sql.DataSource" driverClassName="oracle.jdbc.OracleDriver"
        url="jdbc:oracle:thin:@127.0.0.1:1521:test"
        description="test database for Tomcat 8"
        username="admin" password="admin" maxActive="20" maxIdle="10"
        maxWait="-1"/>
</GlobalNamingResources>
```



DataSource for Oracle

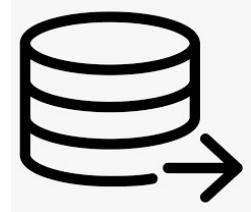
```
<Resource name="jdbc/tomcat7" auth="Container"  
    type="javax.sql.DataSource" driverClassName="oracle.jdbc.OracleDriver"  
    url="jdbc:oracle:thin:@127.0.0.1:1521:mysid"  
    description="User database that can be updated and saved"  
    username="admin" password="admin" maxActive="20" maxIdle="10"  
    maxWait="-1"/>
```

- Oracle JDBC driver classes should be placed in the **CATALINA_HOME/lib/** folder of the Tomcat instance. For Oracle, either class 12.jar or ojdbc14.jar is used.

DataSource for Oracle

- The server web.xml should be used only for the server properties changes, such as the session parameter and so on, which references to the web/application server specific.

```
<resource-ref>
<description>Oracle Datasource for tomcat </description>
<res-ref-name>jdbc/tomcat8 </res-ref-name>
<res-type>javax.sql.DataSource</res-type>
<res-auth>Container</res-auth>
</resource-ref>
```



DataSource for MySQL

- The following lines of code provide the definition of datasource in server.xml
- By default, these values are defined in the global section.

```
<Resource name="jdbc/tomcat8" auth="Container" type="javax.sql.DataSource"  
maxActive="100" maxIdle="30" maxWait="10000" username="tomcatuser"  
password="tomcat" driverClassName="com.mysql.jdbc.Driver"  
url="jdbc:mysql://localhost:3306/tomcat8"/>
```

DataSource for MySQL

```
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee  
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"  
version="2.4">  
  <description>Tomcat 8 test DB</description>  
  <resource-ref>  
    <description>DB Connection</description>  
    <res-ref-name>jdbc/tomcat8</res-ref-name>  
    <res-type>javax.sql.DataSource</res-type>  
    <res-auth>Container</res-auth>  
  </resource-ref>  
</web-app>
```



DataSource for MySQL

- One of the most important points which the Tomcat administrator should keep in mind is that, in MySQL, the DB should be configured with all privileges for the DB server user.
- Log in to the MySQL prompt and run the following command to grant the

```
mysql> GRANT ALL PRIVILEGES ON *.* TO tomcatuser@localhost  
IDENTIFIED BY 'tomcat8' WITH GRANT OPTION;
```

```
mysql> create database tomcat8;
```

```
mysql> use tomcat8;
```

```
mysql> create table testdata ( id int not null auto_increment primary  
key,foo varchar(25), bar int);
```

DataSource for PostgreSQL

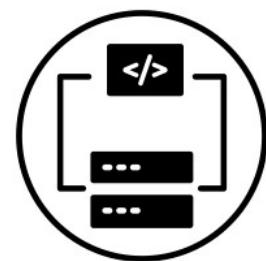
- The following code provides the definition of datasource in server.xml.
- By default, these values are defined in the global section.

```
<Resource name="jdbc/tomcat8" auth="Container" type="javax.sql.DataSource"  
driverClassName="org.postgresql.Driver"  
url="jdbc:postgresql://127.0.0.1:5432/tomcat8" username="tomcat8"  
password="tomcat" maxActive="20" maxIdle="10" maxWait="-1"/>
```

DataSource for PostgreSQL

- For the web.xml configuration of the application, use the following lines of code.
- This should be placed in the WEB-INF/web.xml for the application-specific content.

```
<resource-ref>
<description>postgreSQL Tomcat datasource </description>
<res-ref-name>jdbc/tomcat8 </res-ref-name>
<res-type>javax.sql.DataSource</res-type>
<res-auth>Container</res-auth>
</resource-ref>
```



Comparison of the datasource for common databases

- **Oracle:** The following mentioned code describes the datasource parameter for the Oracle database:

```
<Resource name="jdbc/tomcat8" auth="Container" type="javax.sql.DataSource"  
driverClassName="oracle.jdbc.OracleDriver"  
url="jdbc:oracle:thin:@127.0.0.1:1521:test" description="test database for Tomcat 8"  
username="admin" password="admin" maxActive="20" maxIdle="10" maxWait="-  
1"/>
```

- **MySQL:** The following mentioned code describes the datasource parameter for the MySQL database:

```
<Resource name="jdbc/tomcat8" auth="Container" type="javax.sql.DataSource"  
driverClassName="org.postgresql.Driver"  
url="jdbc:postgresql://127.0.0.1:5432/tomcat8" username="tomcat8"  
password="tomcat" maxActive="20" maxIdle="10" maxWait="-1"/>
```

Comparison of the datasource for common databases

- **PostgreSQL:** The following mentioned code describes the datasource parameter for the PostgreSQL database:

```
<Resource name="jdbc/tomcat8" auth="Container"  
type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"  
url="jdbc:postgresql://127.0.0.1:5432/tomcat8" username="tomcat8"  
password="tomcat" maxActive="20" maxIdle="10" maxWait="-1"/>
```

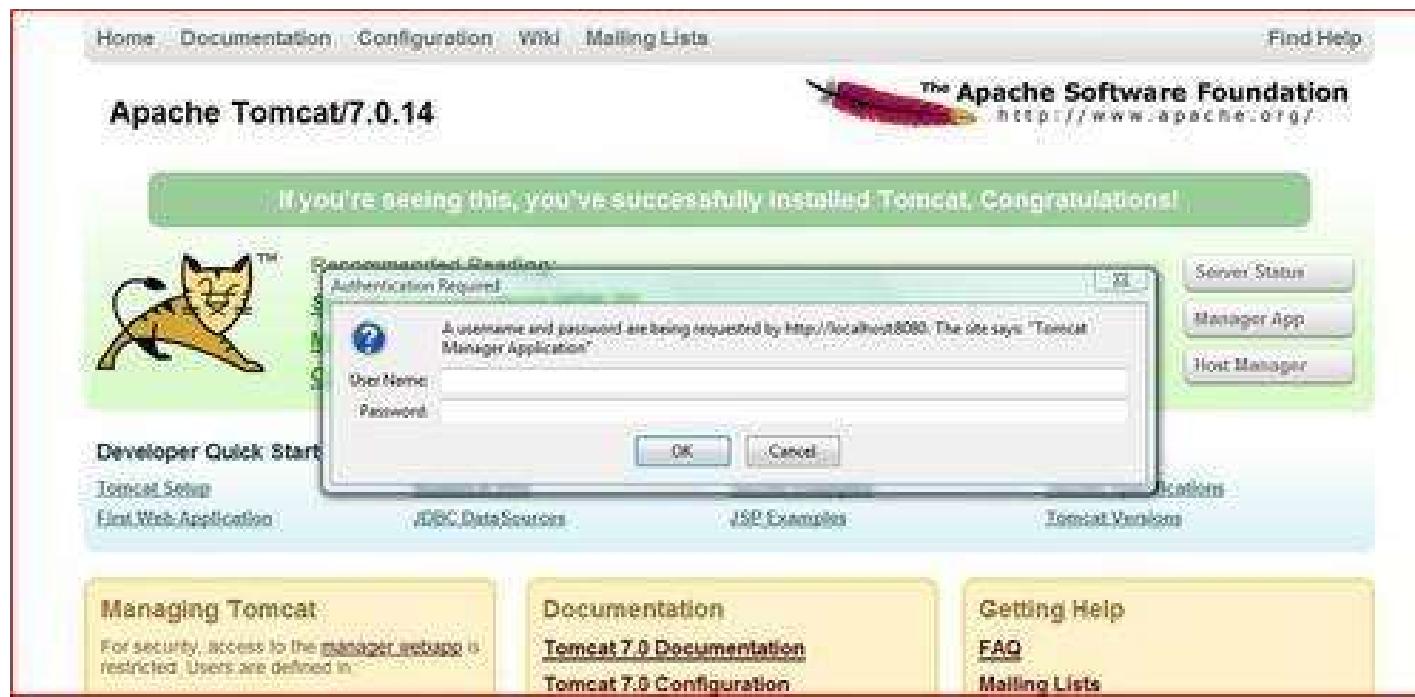
| Database | oracle | Mysql | Postgresql |
|-------------|--------------------------|-----------------------|------------------------------|
| classes | oracle.jdbc.OracleDriver | com.mysql.jdbc.Driver | org.postgresql.Driver |
| Port | 1521 | 3306 | 5432 |
| JDBC driver | ojdbc14.jar | MySQL-3.23.47 | postgresql-9.0-801.jdbc3.jar |

Tomcat Manager configuration

- Deployment of a new application remotely
- Idle session clearing
- Unemployment of an application without restarting the container
- Analysis of memory leaks
- JVM status
- Server status



Enabling the Tomcat Manager



Enabling the Tomcat Manager

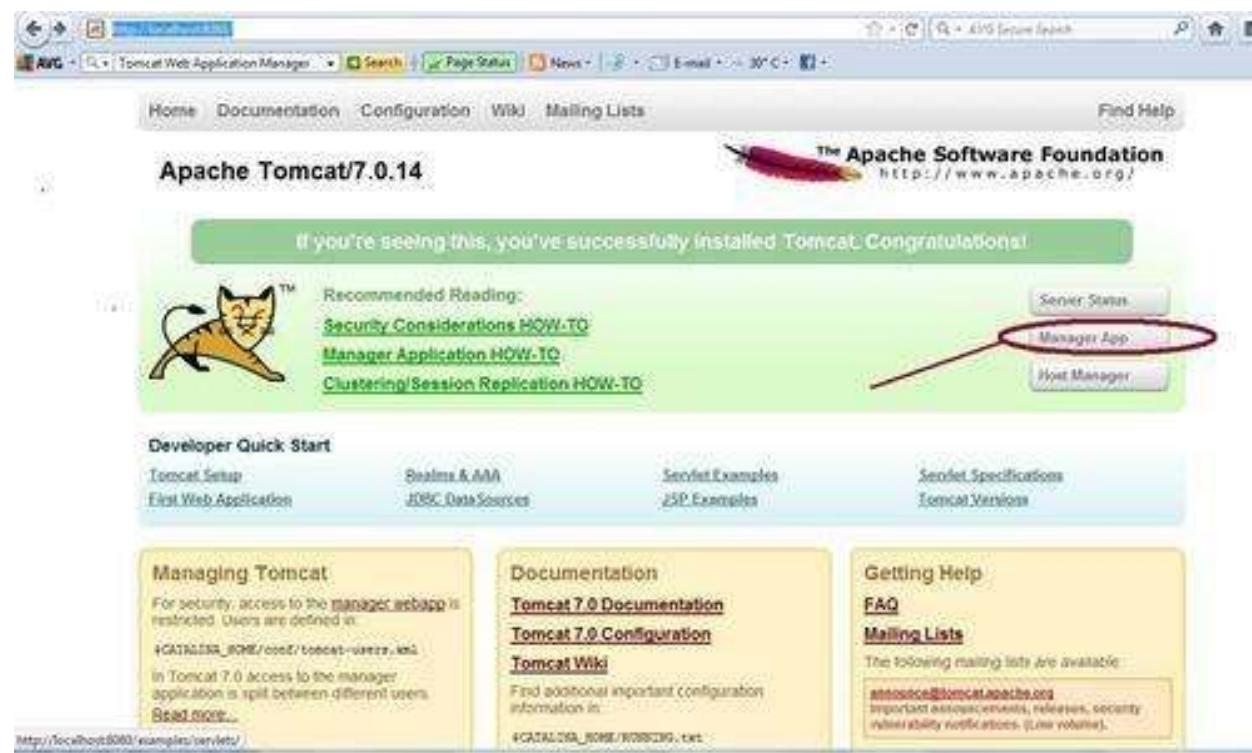
```
<tomcat-users>
<!--
  NOTE: By default, no user is included in the "manager-gui" role required
  to operate the "/manager/html" web application. If you wish to use this ap-
  you must define such a user - the username and password are arbitrary.
-->
<!--
  NOTE: The sample user and role entries below are wrapped in a comment
  and thus are ignored when reading this file. Do not forget to remove
  <!... ...> that surrounds them.
-->
<!--
<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>
```

Enabling the Tomcat Manager

- After enabling the Tomcat Manager, the user will get a message in the command prompt, as shown in the following screenshot:

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

Enabling the Tomcat Manager



Enabling the Tomcat Manager

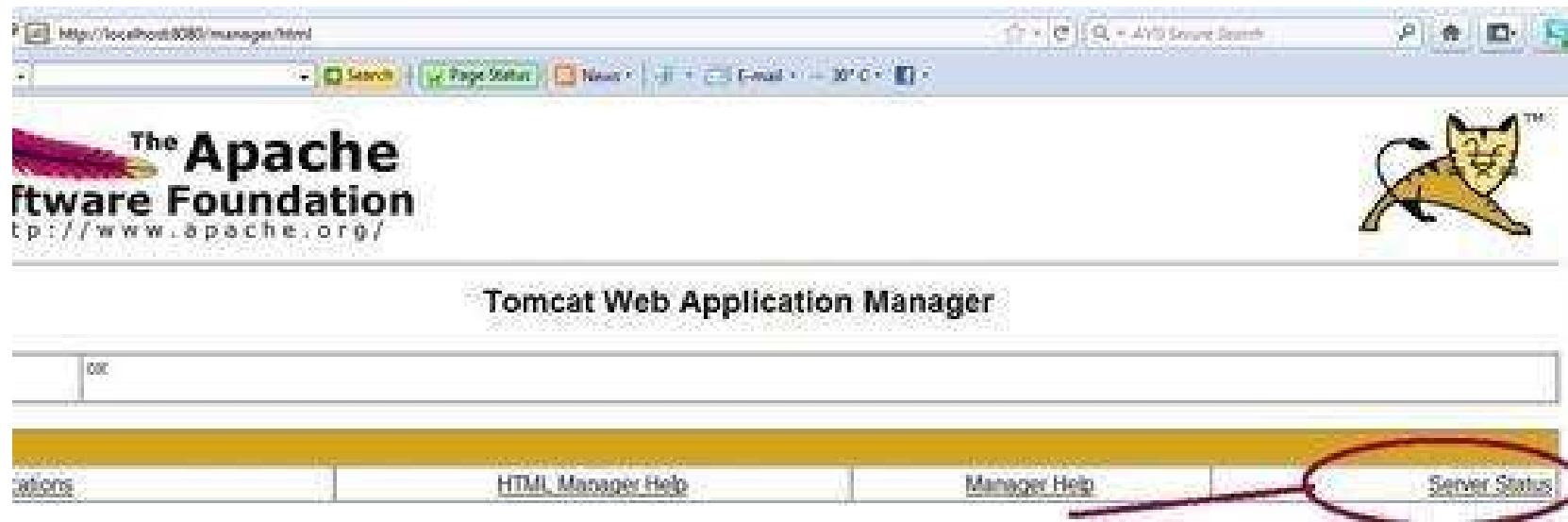
The screenshot shows a web browser window displaying the Tomcat Web Application Manager. The URL in the address bar is `http://localhost:8080/manager/html`. The page header includes the Apache Software Foundation logo and a cartoon cat icon.

The main content area is titled "Tomcat Web Application Manager". It features a "Message" input field with the value "OK". Below this is a navigation bar with tabs: "Manager", "List Applications", "HTML Manager Help", "Manager Help", and "Server Status".

The central part of the page is a table titled "Applications" with the following data:

| Path | Version | Display Name | Running | Sessions | Commands |
|---------------|----------------|---------------------------------|---------|----------|--|
| / | None specified | Welcome to Tomcat | true | 1 | Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes |
| /docs | None specified | Tomcat Documentation | true | 0 | Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes |
| /examples | None specified | Servlet and JSP Examples | true | 0 | Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes |
| /host-manager | None specified | Tomcat Host Manager Application | true | 0 | Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes |
| /manager | None specified | Tomcat Manager Application | true | 0 | Start Stop Reload Undeploy |

Enabling the Tomcat Manager



- The following screenshot shows the Server Status:

Manager

[List Applications](#) [HTML Manager Help](#) [Manager Help](#) [Complete Server Status](#)

Server Information

| Tomcat Version | JVM Version | JVM Vendor | OS Name | OS Version | OS Architecture |
|----------------------|--------------|-----------------------|---------------|------------|-----------------|
| Apache Tomcat/7.0.14 | 1.6.0_24-b07 | Sun Microsystems Inc. | Windows Vista | 6.0 | x86 |

JVM

Free memory: 8.99 MB Total memory: 15.56 MB Max memory: 247.58 MB

"ajp-bio-8009"

Max threads: 200 Current thread count: 0 Current thread busy: 0
Max processing time: 0 ms Processing time: 0.0 ms Request count: 0 Error count: 0 Bytes received: 0.00 MB Bytes sent: 0.00 MB

| Stage | Time | B Sent | B Recv | Client | VHost | Request |
|-------|------|--------|--------|---------------|-----------|--|
| R | ? | ? | ? | ? | ? | ? |
| R | ? | ? | ? | ? | ? | ? |
| S | 4 ms | 0 KB | 0 KB | 0.0.0.0.0.0.1 | localhost | GET /manager/status?org.apache.catalina.filters.CSRF_NONCE=CAF3D8ABED5AABB298A5628E9BC586E0 HTTP/1.1 |
| R | ? | ? | ? | ? | ? | ? |

Context path

- The context path is a key element of a web application. It's also used for a virtual host.
- Virtual hosting can be defined as a method through which you can host multiple domain names on the same web server or a single IP.
- The context path is also used to define the URL mapping for the .war files.

Enabling the context path

The context path in Tomcat can be enabled in two ways:

- GUI using the Tomcat Web Application Manager
- Command-line configuration in server.xml

GUI using the Tomcat Web Application Manager

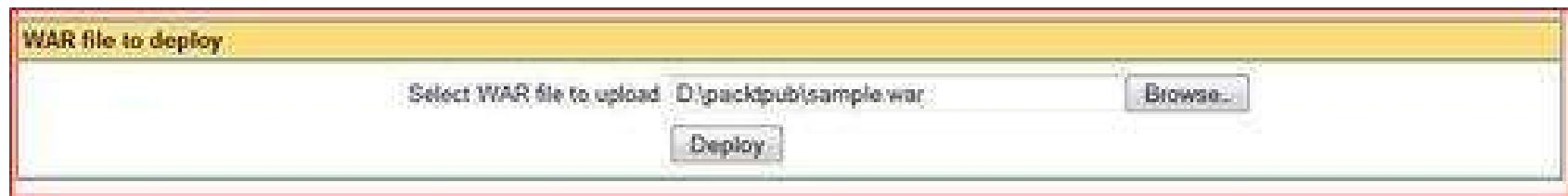


GUI using the Tomcat Web Application Manager

The screenshot shows a web browser window displaying the Tomcat Web Application Manager. The URL in the address bar is `http://localhost:8080/manager/html`. The page header includes the Apache Software Foundation logo and a cartoon cat icon. The main content area is titled "Tomcat Web Application Manager". A message box at the top left contains the text "Message: OK". Below it is a navigation bar with tabs: "List Applications" (selected), "HTML Manager Help", "Manager Help", and "Server Status". The main section is titled "Applications" and lists the following entries:

| Path | Version | Display Name | Running | Sessions | Commands |
|--------------|----------------|---------------------------------|---------|----------|---|
| | None specified | Welcome to Tomcat | true | 1 | <button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <input type="button" value="Expire sessions"/> with idle > 30 minutes |
| docs | None specified | Tomcat Documentation | true | 0 | <button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <input type="button" value="Expire sessions"/> with idle > 30 minutes |
| examples | None specified | Servlet and JSP Examples | true | 0 | <button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <input type="button" value="Expire sessions"/> with idle > 30 minutes |
| host-manager | None specified | Tomcat Host Manager Application | true | 0 | <button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> <input type="button" value="Expire sessions"/> with idle > 30 minutes |
| manager | None specified | Tomcat Manager Application | true | 2 | <button>Start</button> <button>Stop</button> <button>Reload</button> <button>Undeploy</button> |

GUI using the Tomcat Web Application Manager

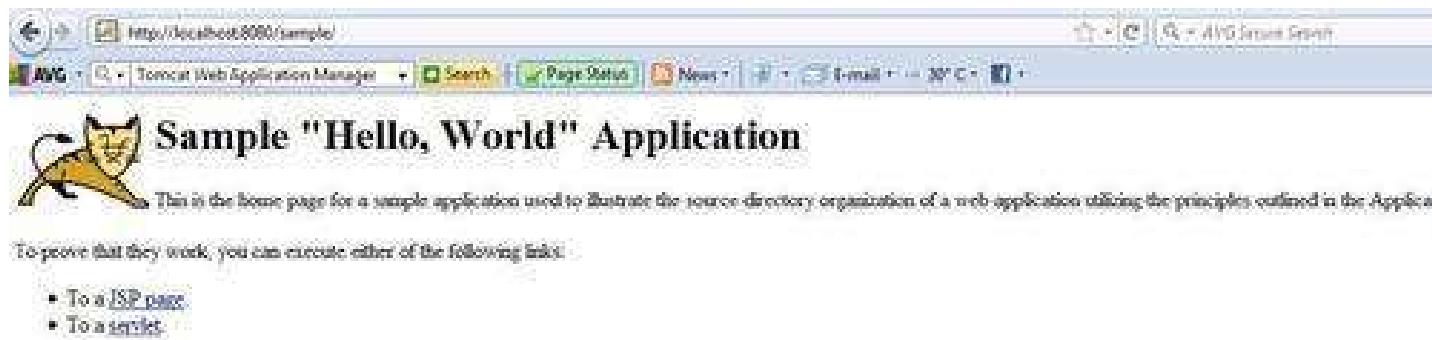


- The following screenshot shows the application deployment status and administrative controls such as Stop, Reload, and Undeploy:



GUI using the Tomcat Web Application Manager

- Once the application is deployed successfully, you can browse the application using the URL `http://localhost:8080/sample`, as shown in the following screenshot



Command-line configuration in server.xml

- Another way of adding the context path in Tomcat 8 is by editing server.xml.
- But, you need to have a good understanding of XML.
- Let's quickly discuss the changes that need to be done on the Tomcat server.

```
<Context path="/sample" docBase="/opt/" reloadable="true"  
swallowOutput="true">  
    <WatchedResource>WEB-INF/web.xml</WatchedResource>  
    <Logger className="org.apache.catalina.logger.FileLogger"  
prefix="www-sample-com-log." suffix=".txt" timestamp="true"/>  
</Context>
```

Command-line configuration in server.xml

```
<Context path="/sample" docBase="/opt/" reloadable="true"
    swallowOutput="true">
<WatchedResource>WEB-INF/web.xml</WatchedResource>
    <Logger className="org.apache.catalina.logger.FileLogger"
        prefix="www-example-com-log." suffix=".txt"
        timestamp="true"/>
</Context>
```

Deployment in Tomcat 8

- Deployment is basically defined as the installation of the WAR files in the web application.
- In other words, we can define the unpacking of the WAR file in the Tomcat webapps directory.

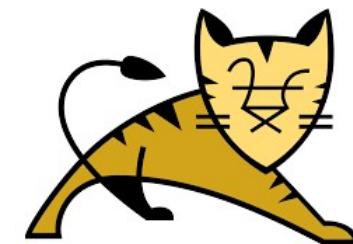
Apache Tomcat



Structure of the WebArchive

- In the previous snippet, we are mapping the name to the servlet class (when Tomcat 8 starts, it will create an object of the class and map it to the name we have provided in the servlet-name field).

```
classB =new class.classB ()  
<servlet-mapping>  
<servlet-name> classB </servlet-name>
```



Archive Files

- EJBs are packaged as .jar files
- Web applications are packaged as .war files
- Resource adapters are packaged as .rar files
- Enterprise applications are packaged as .ear files, and can contain any combination of EJBs, web applications, and resource adapters
- Web services can be packaged either as .ear files or as .war files

Exploded archive directories

- You want to perform partial updates to a deployed application without redeploying the entire application.
- You want to use the Tomcat Manager to dynamically edit and persist selected deployment descriptor values for the deployment.
- You are deploying a web application that contains static files that you will periodically update.
- In this case, it is easier to deploy the application as an exploded directory, because you can update and refresh the static files without re-creating the archive.

Deployment operations

The deployment tools provide support for performing these common deployment operations:

- Deploy
- Redeploy
- Stop
- Start
- Undeploy



Types of deployment

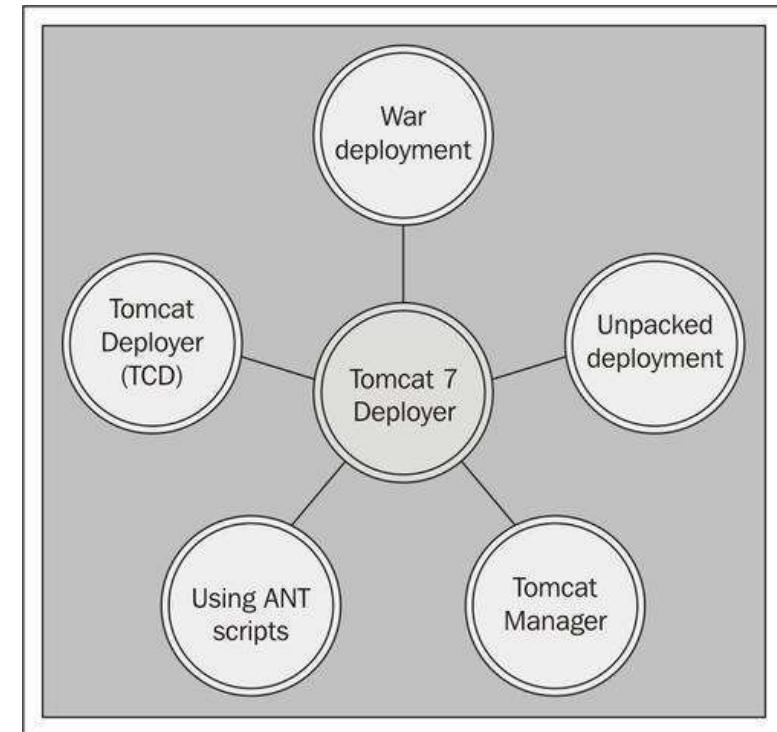
The deployment staging mode determines how deployment files are made available to the target servers that must deploy an application or standalone module. The Tomcat 8 server provides three different options for staging files listed as follows:

- Stage mode
- Nostage mode
- External_stage mode

| Deployment Staging Mode | Behavior | When to Use |
|-------------------------|--|--|
| Stage | The Tomcat administrator first copies the deployment unit source files to the staging directories of the target servers and then the target servers deploy them using their local copy of the deployment files. | Deploying small or moderate-sized applications to multiple Tomcat 8 server instances. Deploying small or moderate-sized applications to a cluster. |
| Nostage | The Tomcat administrator does not copy the deployment unit files. Instead, all servers deploy using the same physical copy of the deployment files, which must be directly accessible by the Tomcat administrator and target servers. Nostage deployments of exploded archive directories is not recommended | Deploying to a single server instance. Deploying to a cluster on a multi-homed machine. Deploying very large applications to multiple targets or to a cluster where deployment files are placed on the server. |
| External_stage | The Tomcat administrator does not copy the deployment files. Instead, the administrator must ensure that deployment files are distributed to the correct staging directory location before deployment (for example, by manually copying files prior to deployment). With external_stage deployments, the Tomcat administrator requires a copy of the deployment files for validation purposes. Copies of the deployment files that reside in the target servers' staging directories are not validated before deployment. | Deployments where you want to manually control the distribution of deployment files to the target servers. Deploying to the server instance where third-party applications or scripts manage the copying of deployment files to the correct staging directories. Deployments that do not require a dynamic update of selected deployment descriptors via the Tomcat Manager (not supported in external stage mode). Deployments that do not require partial redeployment of the application components. |

Ways of application deployment in Tomcat 8

- Deployment of applications can be done in many ways in Tomcat 8.
- There are five different ways which are widely known and accepted in the various industries displayed in the following figure:



Http://localhost:8080/manager/html

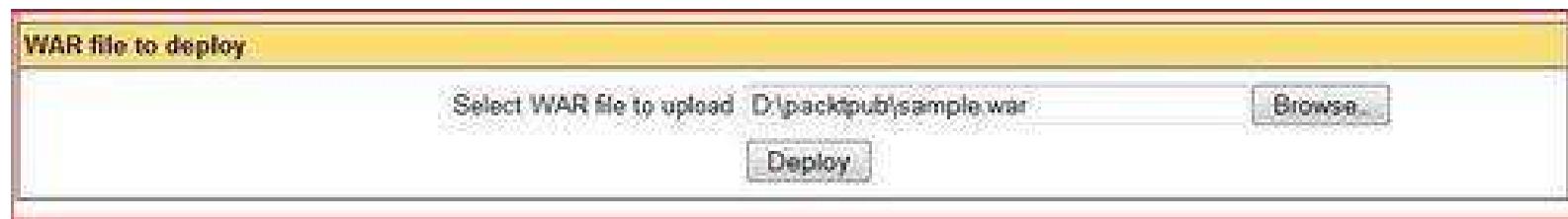
The Apache Software Foundation http://www.apache.org/

Tomcat Web Application Manager

| Manager | | HTML Manager Help | | Manager Help | | Server Status | |
|-------------------|------|-------------------|---------------------------------|--------------|----------|--|--|
| List Applications | | | | | | | |
| Applications | | | | | | | |
| | Path | Version | Display Name | Running | Sessions | Commands | |
| | | None specified | Welcome to Tomcat | true | 1 | <input type="button" value="Start"/> <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> | <input type="button" value="Expire sessions with idle > 30 minutes"/> |
| docs | | None specified | Tomcat Documentation | true | 0 | <input type="button" value="Start"/> <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> | <input type="button" value="Expire sessions with idle > 30 minutes"/> |
| examples | | None specified | Servlet and JSP Examples | true | 0 | <input type="button" value="Start"/> <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> | <input type="button" value="Expire sessions with idle > 30 minutes"/> |
| host-manager | | None specified | Tomcat Host Manager Application | true | 9 | <input type="button" value="Start"/> <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> | <input type="button" value="Expire sessions with idle > 30 minutes"/> |
| manager | | None specified | Tomcat Manager Application | true | 2 | <input type="button" value="Start"/> <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> | |

Ways of application deployment in Tomcat 8

- You can create the context path using the Deploy tab.
- Click on Browse and select the required WAR file.
Then click on Deploy.
- It will take 10 to 15 seconds to deploy the application
and you will see a page like the following screenshot:



Ways of application deployment in Tomcat 8

- The following screenshot shows the application deployment status and administrative control such as Stop, Reload, and Undeploy:



Ways of application deployment in Tomcat 8

- Once the application is deployed successfully, as shown in the following screenshot, you can browse the application using the URL, <http://localhost:8080/sample>:



Common issues in deployment, configuration, and their troubleshooting

Scenario 1:

- Issue: Users complain that after the deployment, they can still view the old code.

Troubleshooting steps:

- Check if the latest file is present on the doc base.
- Check the catalina.out in the log's directory of Tomcat 8 and whether the WAR filename is deployed or not.

Common issues in deployment, configuration, and their troubleshooting

- If both are checked and the issue still persists, then stop the Tomcat service and clear the content of the temp directory under the work/Catalina/localhost using the following command:

```
cd /opt/apache-tomcat-8.5.61/temp/ rm -rf ../*  
cd /opt/apache-tomcat-8.5.61/work/Catalina/localhost/  
rm -rf ../*
```

Common issues in deployment, configuration, and their troubleshooting

Scenario 2:

- Issue: Users complaining that they can view the current deployed code on one node and the other node still displays the previous version of the code.

Troubleshooting steps:

- Check if the latest file is present on the doc base.
- Check the catalina.out in the log's directory of Tomcat 8 and whether the WAR filename is deployed or not.

Common issues in deployment, configuration, and their troubleshooting

- If both are checked and the issue still persists, then stop the Tomcat service on node2.
- Replicate the code from node1 and clear the content of the temp directory under the work/Catalina/localhost using the following command:

```
cd /opt/apache-tomcat-8.5.61/temp/ rm -rf ../*  
cd /opt/apache-tomcat-8.5.61/work/Catalina/localhost/  
rm -rf ../*
```

Common issues in deployment, configuration, and their troubleshooting

Scenario 3:

- Issue: The Tomcat instance is not coming up after the changes made to server.xml.

Troubleshooting steps:

- Go to the Tomcat bin directory.
- Then, run the configtest.sh.

```
[root@localhost ~]# cd /opt/apache-tomcat-8.5.61/bin/  
[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: /opt/jdk1.6.0_24  
Using CLASSPATH: /opt/apache-tomcat-8.5.61/bin/bootstrap.jar:/opt/apache-tomcat-8.5.61/bin/tomcat-juli.jar  
Error:-  
org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:435)  
Caused by: java.net.BindException: Address already in use  
at java.net.PlainSocketImpl.socketBind(Native Method)  
at java.net.PlainSocketImpl.bind(PlainSocketImpl.java:383)  
at java.net.ServerSocket.bind(ServerSocket.java:328)  
at java.net.ServerSocket.<init>(ServerSocket.java:194)  
at java.net.ServerSocket.<init>(ServerSocket.java:150)
```

Summary



- In this lesson, we have discussed the configuration of Tomcat including data source configuration for the different databases (Oracle, MySQL, and PostgreSQL)
- The context path creation using a sample application, various ways to perform deployment including deployment using the Tomcat Manager for the sample application.

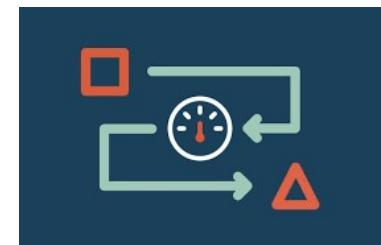
3. Performance Tuning



Performance Tuning

In this lesson, we will cover the major topics of performance tuning including the following:

- Memory related issues
- JVM parameter optimization
- OS level optimization to improve the performance

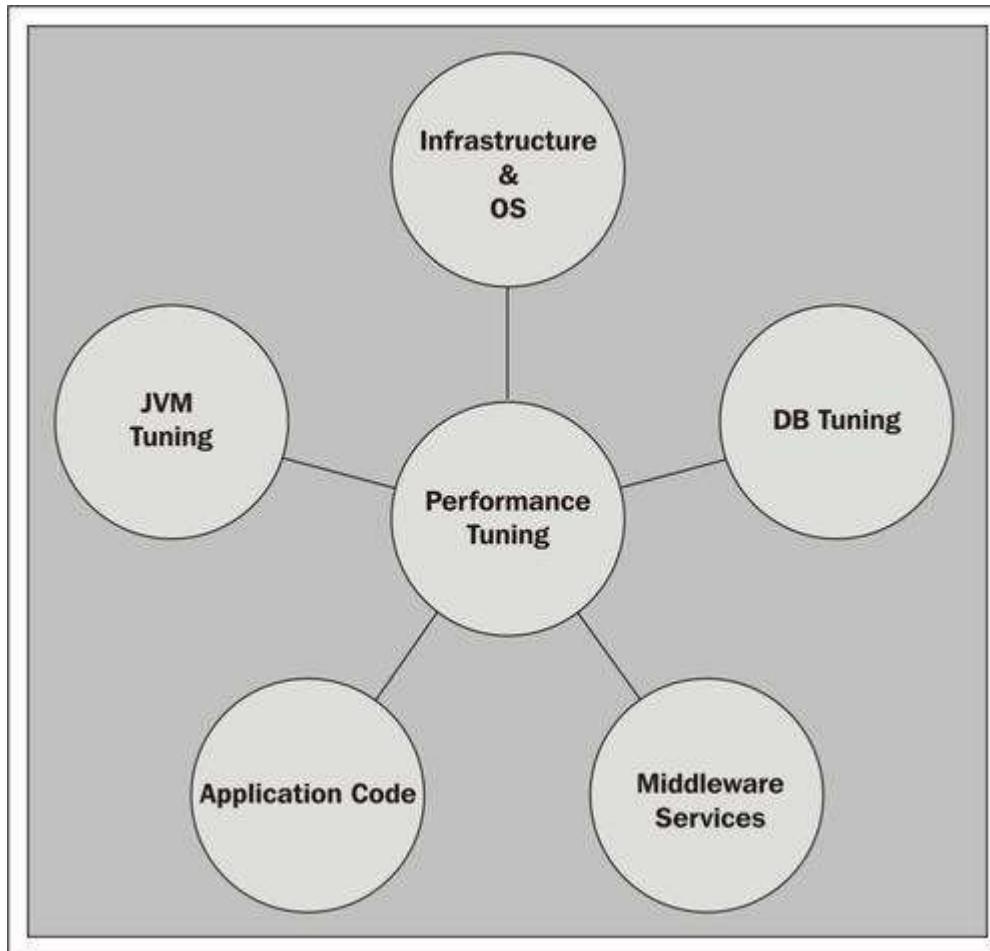


Performance tuning for Tomcat 8

- Performance tuning plays a vital role to run a web application without downtime.
- Also, it helps in improving the performance of Tomcat while running the applications.
- Tuning of the Tomcat server may vary from application to application.
- Since every application has its own requirements, it is a very tricky task to tune Tomcat 8 for every application.

Why we need performance tuning?

- People always ask, why do we need to do performance tuning for Tomcat 8 when, by default, Tomcat 8 packages are customized for the production run.
- The answer to this question is very subjective; every web application has its own requirement.
- Some of the applications require high memory while others require less memory but a high GC pause.



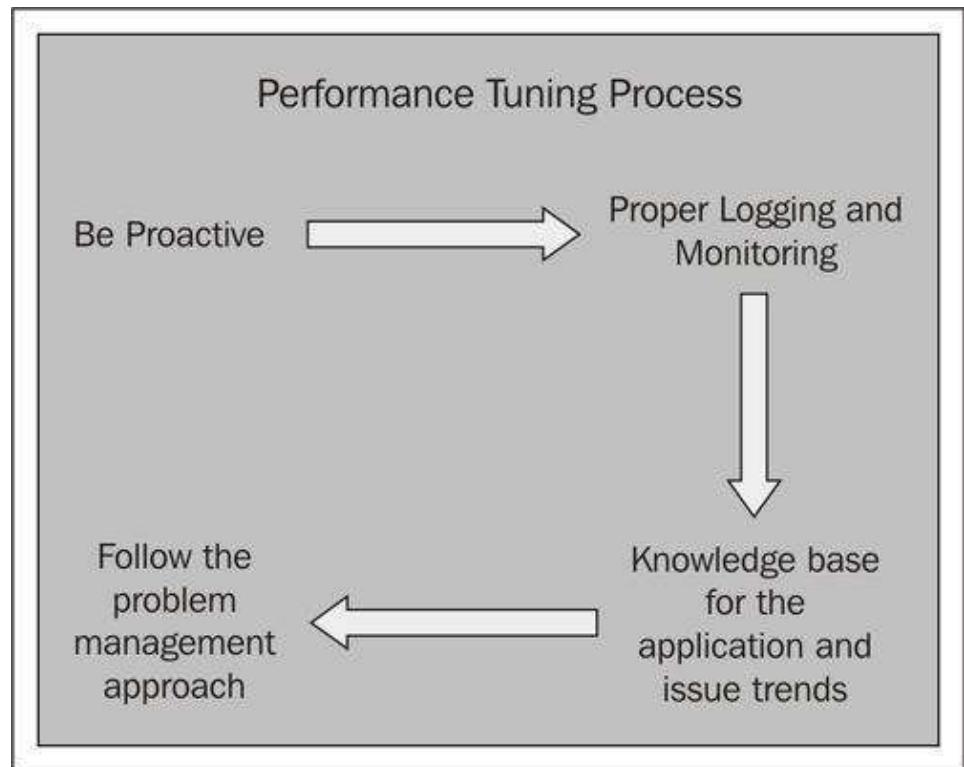
How to start performance tuning

- Performance tuning starts from the day the application deployment stage begins.
- One may ask, as the application is only in the development phase, why do we need to do performance tuning now?
- At the time of the application development, we are in the state to define the architecture of the application, how the application is going to perform in reality, and how many resources are required for an application.



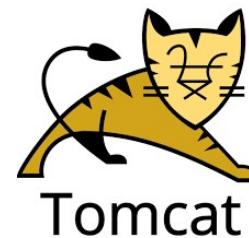
How to start performance tuning

- The following figure shows the process flow for performance tuning:



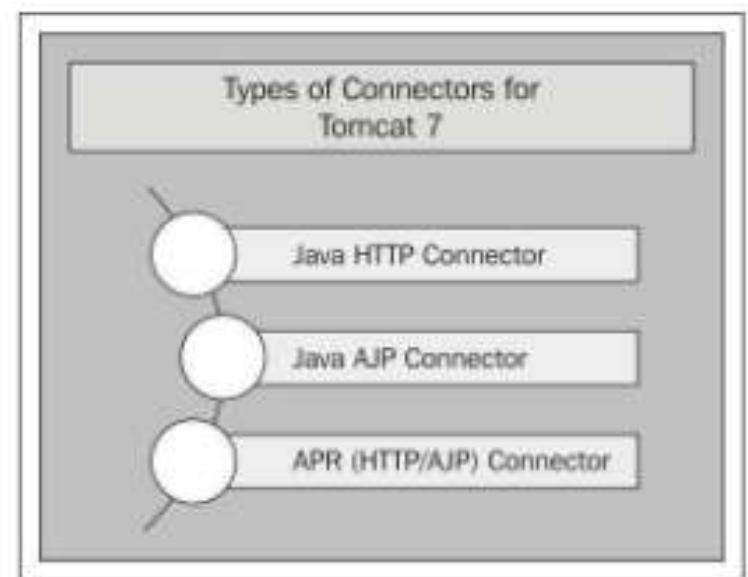
Tomcat components tuning

- In Tomcat 8, you can do many configurations to improve the server performance, threads tuning, port customization, and JVM tuning.
- Let's quickly discuss the major components of Tomcat 8 which are important for performance improvement.



Types of connectors for Tomcat 8

- Connectors can be defined as a point of intersection where requests are accepted, and responses are returned.
- There are three types of connectors used in Tomcat 8, as shown in the following figure.
- These connectors are used according to different application requirements.
- Let's discuss the usability of each connector:



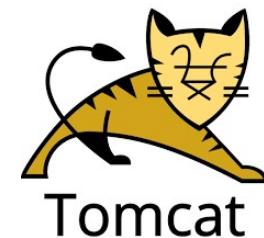
Thread optimization for Tomcat 8

- Thread tuning plays a major role in Tomcat's performance. In most cases, we have seen that a particular application works very well on other industries, but when we implement the same application, its performance is degraded (application performance issues).
- The reason being that there may be the chance that we have done an improper tuning of the thread, which may lead to server-degraded performance.

Shared thread pool (shared executor)

- Edit the server.xml and add the definition of the shared thread pool inside the services section.
- The following highlighted code shows the thread pool:

```
<Executor name="tomcatThreadPool"  
namePrefix="catalina-exec-"  
maxThreads="150"  
minSpareThreads="4"/>
```



Shared thread pool (shared executor)

- Once you have defined the shared thread pool, call the reference of the thread setting in the Connector definition for the services section of server.xml, as shown in the following code:

```
<Connector executor="tomcatThreadPool"  
port="8080"  
protocol="HTTP/1.1"  
connectionTimeout="20000"  
redirectPort="8443" />
```

Dedicated thread pool

- Edit the server.xml and define the configuration of the dedicated thread pool inside the Connector section.
- The following highlighted code shows the thread pool:

```
<Connector port="8443" protocol="HTTP/1.1"
SSLEnabled="true"
maxThreads="150"
scheme="https"
secure="true"
clientAuth="false" sslProtocol="TLS" />
```



Shared thread pool versus dedicated thread pool

- Let us do a quick comparison between shared thread pool and dedicated thread pool to find out where they can be used.
- The following table shows the comparison between both the thread pooling methods for Tomcat 8:

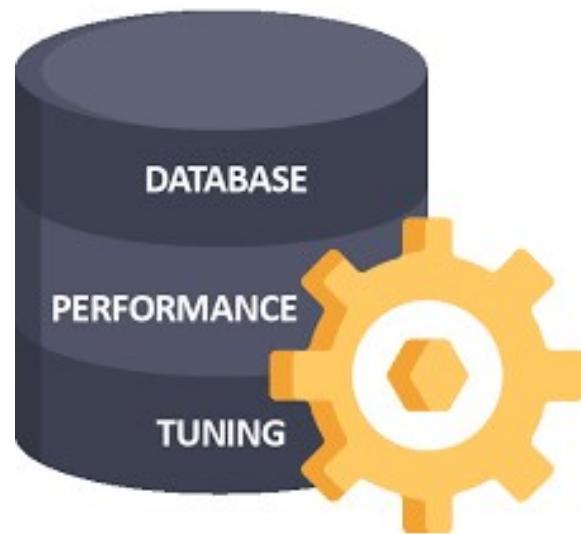
| Features | Shared thread pool | Dedicated thread pool |
|-----------------|--------------------|-----------------------|
| Number of users | Less | High |
| Environment | Development | Production |
| Performance | Low | Good |

maxThreads

- maxThreads can be defined as the highest number of requests a server can accept. By default, Tomcat 8 comes with maxThreads=150.
- In a production environment, we have to tune the maxThreads based on the server performance.
- Let's perform a real time maxThreads tuning.
- Let us assume we have maxThreads=300 for an application.

JVM tuning

- Before we start with JVM tuning, we should note that there are various vendors available in the market for JVM.
- Based on the application requirement, we should select the JDK from the vendor.



Why do we need to tune the JDK for Tomcat?

- Run the following command on the terminal in Linux:
ps -ef |grep java
- This will return all the Java processes running in the system, with all information such as the PID, where Tomcat is running, and so on:

```
root@localhost: bin]# ps -ef |grep java
root      4884  1 0 14:09 pts/1    00:00:00 /opt/apache-tomcat-7.0.14/bin/java -Djava.util.logging.config.file=/opt/apache-tomcat-7.0.14/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.endorsed.dirs=/opt/apache-tomcat-7.0.14/endorsed -classpath /opt/apache-tomcat-7.0.14/bin/bootstrap.jar:/opt/apache-tomcat-7.0.14/bin/tomcat-juli.jar -Dcatalina.base=/opt/apache-tomcat-7.0.14 -Dcatalina.home=/opt/apache-tomcat-7.0.14 -Djava.ext.dirs=/opt/apache-tomcat-7.0.14/lib -Djava.security.manager=catalina.startup.Bootstrap start
root@localhost: bin]#
```

Why do we need to tune the JDK for Tomcat?

```
root 4306 1 0 14:09 pts/1 00:00:04 /opt/jdk1.6.0_24/bin/java -  
Djava.util.logging.config.file=/opt/apache-tomcat-8.5.61/conf/logging.properties -  
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -  
Djava.endorsed.dirs=/opt/apache-tomcat-8.5.61/endorsed-classpath  
/opt/apache-tomcat-8.5.61/bin/bootstrap.jar:/opt/apache-tomcat-  
8.5.61/bin/tomcat-juli.jar -Dcatalina.base=/opt/apache-tomcat-8.5.61 -  
Dcatalina.home=/opt/apache-tomcat-8.5.61 -Djava.io.tmpdir=/opt/apache-tomcat-  
8.5.61/temp org.apache.catalina.startup.Bootstrap start
```

- In the previous output, 4306 is the PID for the Tomcat process.
- Now, we know the process ID for Tomcat, let's find out the memory allocated to the Tomcat instance using the command jmap.

Why do we need to tune the JDK for Tomcat?

- For checking the process ID for Tomcat on Windows, you have to run the following command:
tasklist |find "tomcat"
- The following screenshot shows the output of the previous command, where 2112 is the process ID for the Tomcat process:

```
C:\Users\user>tasklist |find "tomcat"
tomcat7.exe          2112 Services              0      38,656 K
```

JMAP (Memory Map)

| Options | Description |
|-----------------|--|
| -dump | Dumps the Java heap in hprof binary format |
| -finalizer info | Prints information on objects awaiting finalization |
| -heap | Prints a heap summary |
| -histo | Prints a histogram of the heap |
| -permstat | Prints class loader-wise statistics of permanent generation of the Java heap |

Syntax for jmap

- In our current scenario, the PID is 4306:
`[root@localhost bin]# ./jmap -heap 4306`
- The output of the previous command is as follows:
Refer to the file 3_1.txt



Syntax for jmap

The previous command returns the following details:

- The heap configuration for the application (the highlighted code describes the heap configuration)
- The heap utilization for each JVM component
- The algorithm used for the garbage collection

How to increase the heap size in Tomcat 8

- To increase the heap size for Tomcat 8, we need to add the JAVA_OPTS parameter in catalina.sh, which can be found in TOMCAT_HOME/bin.
- Let us take an example of increasing the max heap size to 512 MB instead 256 MB.
- Also setting the Perm Gen = 256 MB.

JAVA_OPTS="-Xms128m -Xmx512m -XX:MaxPermSize=256m"

```
[root@localhost bin]# jmap -heap 21091
```

- The output of the previous command is as follows:

Attaching to process ID 21091, please wait...

Debugger attached successfully.

Client compiler detected.

JVM version is 19.1-b02

using thread-local object allocation.

Mark Sweep Compact GC

Heap Configuration:

MinHeapFreeRatio = 40

MaxHeapFreeRatio = 70

MaxHeapSize = 536870912 (512.0MB)

NewSize = 1048576 (1.0MB)

MaxNewSize = 4294901760 (4095.9375MB)

OldSize = 4194304 (4.0MB)

NewRatio = 2

SurvivorRatio = 8

PermSize = 12582912 (12.0MB)

MaxPermSize = 268435456 (256.0MB)



Garbage collection

- Garbage means waste, but let's find out how this waste term fits in JVM.
- What is garbage in JVM?
- Garbage is nothing but an object that resides in the JVM memory and is not currently used by any program.

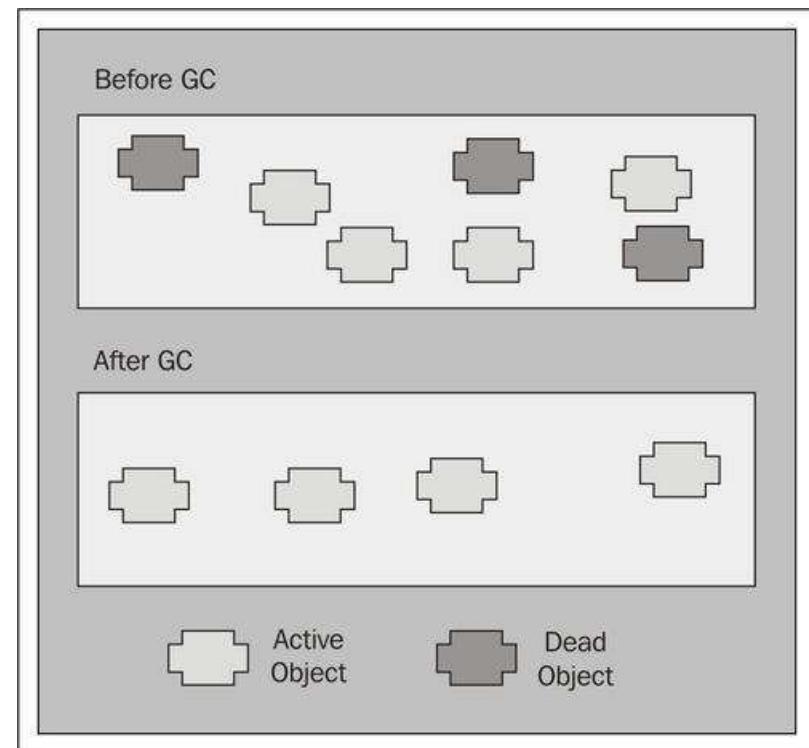
Garbage collection

Following are the facts for garbage collection:

- Garbage collection doesn't work properly with large memory applications
- Garbage collection doesn't consider the fact that some of the objects have a very short life, but some of them remain active for years until the recycle of Tomcat 8

How garbage collection works

- When the GC algorithm is called, it collects all the inactive objects present in the memory and, hence, cleans the memory.
- It can be explained as the opposite of manual memory management:



How garbage collection works

| Features | Serial collector |
|-------------|---|
| Process | Single thread is used for GC |
| GC pause | High |
| Threading | Single threaded |
| Application | Small application (data less than 100 MB) |
| Advantage | There is single thread communication |

How garbage collection works

| Features | Parallel collector |
|-------------|--|
| Process | Parallel thread does minor GC |
| GC pause | Less than Serial |
| Threading | Multithreaded |
| Application | Mid-large |
| Advantage | Used in applications when peak performance is needed |

How garbage collection works

The following table describes the features of the concurrent collector:

| Features | Concurrent collector |
|-------------|--|
| Process | GC is done concurrently |
| GC pause | Short pause |
| Threading | Multithreaded |
| Application | Mid-large |
| Advantage | Used in applications when a response is needed |

Standard options

- Standard options are acknowledged by the Java HotSpot VM mentioned in the Java application launcher page for each OS.
- The following screenshot shows the standard option for the Java application launcher reference page:

```
root@localhost ~]# java -showversion
java version "1.6.0_24"
Java(TM) SE Runtime Environment (build 1.6.0_24-b07)
Java HotSpot(TM) Client VM (build 19.1-b02, mixed mode, sharing)

Usage: java [-options] class [args...]
           (to execute a class)
      or  java [-options] -jar jarfile [args...]
           (to execute a jar file)

where options include:
  -d32          use a 32-bit data model if available
  -d64          use a 64-bit data model if available
  -client        to select the "client" VM
  -server        to select the "server" VM
  -hotspot       is a synonym for the "client" VM [deprecated]
                 The default VM is client.

  -cp <class search path of directories and zip/jar files>
  -classpath <class search path of directories and zip/jar files>
             A : separated list of directories, JAR archives,
             and ZIP archives to search for class files.
  -D<name>=<value>
             set a system property
  -verbose[:class|gc|jni]
             enable verbose output
  -version      print product version and exit
  -version:<value>
             require the specified version to run
  -showversion  print product version and continue
  -jre-restrict-search | -jre-no-restrict-search
                 include/exclude user private JREs in the version search
  -? -help       print this help message
  -X            print help on non-standard options
  -ea[<packagename>...|:<classname>]
  -enableassertions[<packagename>...|:<classname>]
             enable assertions
  -da[<packagename>...|:<classname>]
  -disableassertions[<packagename>...|:<classname>]
             disable assertions
  -esa | -enablesystemassertions
             enable system assertions
  -dsa | -disablesystemassertions
             disable system assertions
```

Non-standard options

- **Behavioral options:** It changes the basic behavior of the VM.
- **Performance tuning options:** It triggers the performance optimization for the VM. also, these options are very useful for server tuning.
- **Debugging options:** It displays the output and printing information of the VM. In addition to that, it enables tracing in logs (these options are very useful in troubleshooting critical issues).

| Options | Parameter | Description |
|---------------------|--|---|
| Behavioral Options | -XX:+ScavengeBeforeFullGC | Do young generation GC prior to a full GC |
| Behavioral Options | --XX:-UseParallelGC | Use parallel garbage collection for scavenges |
| Performance Options | -XX:MaxNewSize=size | Maximum size of new generation (in bytes) |
| Performance Options | -XX:MaxPermSize=64m | Size of the Permanent Generation (after exceeding Xmx value) |
| Performance Options | -Xms | Minimum heap memory for the startup of Tomcat |
| Performance Options | Xmx | Maximum memory allocated to the instance |
| Performance Options | -Xss | Stack size for the heap |
| Debugging Options | -XX:-CITime | Prints time spent in the JIT Compiler |
| Debugging Options | -XX:ErrorFile=./hs_err_pid<pid>.log | If an error occurs, save the error data to this file |
| Debugging Options | -XX:HeapDumpPath=./java_pid<pid>.hprof | Path to the directory or filename for the heap dump |
| Debugging Options | -XX:-HeapDumpOnOutOfMemoryError | Dump the heap to the file when java.lang.OutOfMemoryError is thrown |
| Options | Parameter | Description |
| Debugging Options | -XX:OnError=<cmd args>;<cmd args>" | Run user-defined commands on fatal error |
| Debugging Options | -XX:OnOutOfMemoryError=<cmd args>; | Run user-defined commands when an OutOfMemoryError is first thrown |
| Debugging Options | -XX:-PrintClassHistogram | Print a histogram of class instances on Ctrl-Break |

Parameters displayed in the logs for GC

- GC prints the output of the garbage collection to the stdout stream.
- At every garbage collection, the following five fields are printed:

[%T %B->%A(%C), %D]

SurvivorRatio

- It is defined as a ratio of eden to the survivor space size.
- The default value is 8, meaning that eden is 8 times bigger than from and to, each.
- The syntax for the SurvivorRatio is -
`XX:SurvivorRatio=<size>`.
- The following are some examples:

$Xmn / (\text{SurvivorRatio} + 2) = \text{size of from and to, each}$

$(Xmn / (\text{SurvivorRatio} + 2)) * \text{SurvivorRatio} = \text{eden size}$

OS tuning

- Every OS has its own prerequisites to run Tomcat 8 and the system has to be tuned based on the application's requirement, but there are some similarities between each OS.
- Let's discuss the common module used for optimization of Tomcat 8 for every OS.
- The OS plays a vital role for increasing the performance.
- Depending on the hardware, the application's performance will increase or decrease.

OS tuning

- Huge page size: Many applications send a huge page size causing the application to run slow.
- In this case, you can increase the page size based on the application needs.
- You can check the page size by running the following command:

Refer to the file 3_2.txt

Summary

- In this lesson, we have covered the different ways of performance improvement and techniques in Apache Tomcat 8.
- We went through step-by-step configuration for connectors, JVM performance tuning, and OS parameter optimization.



4. Integration of Tomcat with the Apache Web Server



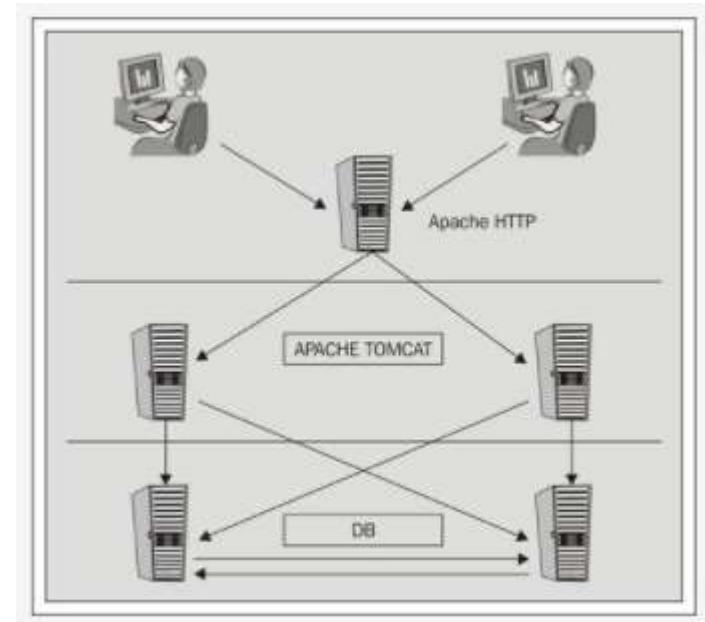
Integration of Tomcat with the Apache Web Server

In this lesson, we will discuss the following topics:

- The Apache HTTP installation
- The various modules of Apache
- Integration of Apache with Tomcat 8
- How IT industry environments are set up

User request flow (web/application level)

- The user hits the URL in the browser and the request goes to the HTTP server instead of Tomcat.
- The HTTP server accepts the request and redirects it to Tomcat for business logic processing.
- Tomcat internally contacts the database server to fetch the data, and sends the response back to the user through the same channel of request:



Why the Apache HTTP server

- The Apache HTTP server is one of the most successful and common web servers used in IT industries.
- The reason being that it is supported by open-source communities.
- In IT industries, the Apache HTTP server is heavily used as a frontend web server for the following reasons.

Efficiently serves static content

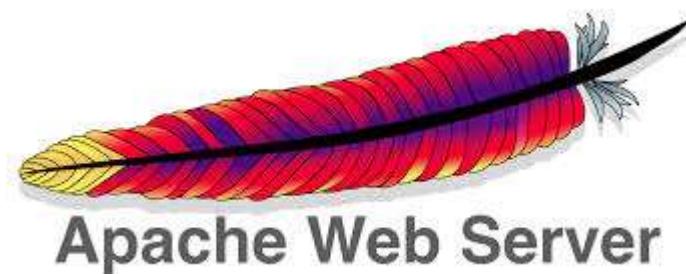
Clustering

Security

Multiple website hosting

Modules

Decorator



Installation of the Apache HTTP

- The Apache installation can be done using various methods, based on the requirement of the infrastructure.
- For example, if you want to run multiple Apache instances on a single machine, then the Source installation will be used.
- There are mainly three types of installations done in various web environments:

Source

Binary

RPM/exe



Apache HTTP installation on Windows

Apache HTTP Server (httpd) 2.2.19 is the best available version. 2011-05-22

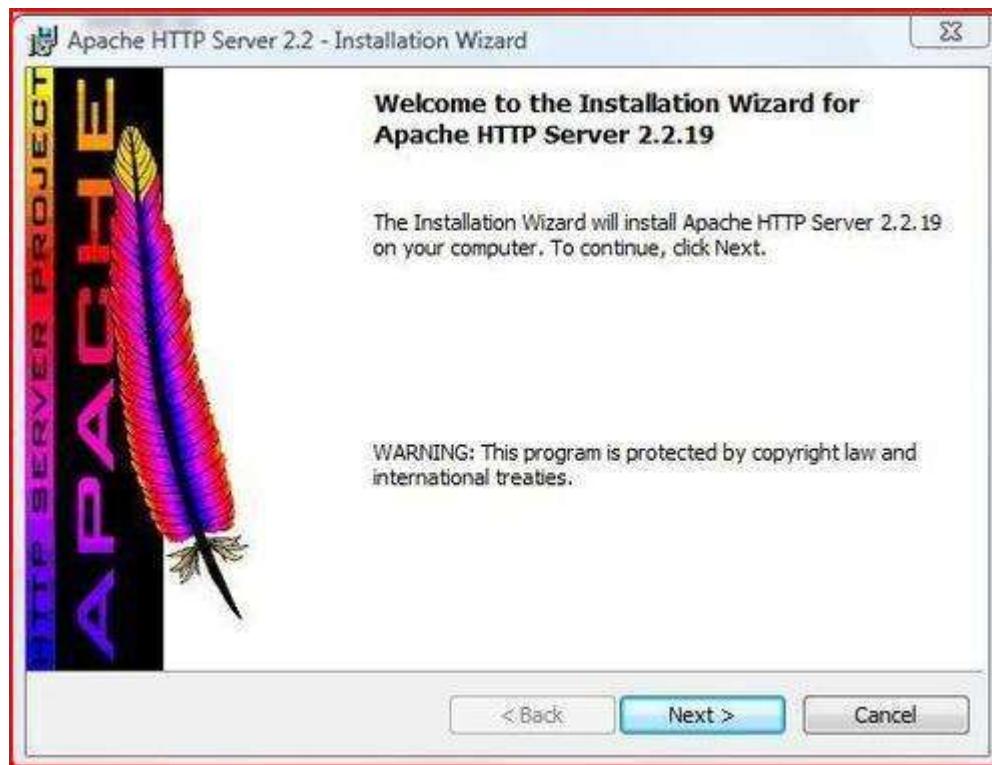
The Apache HTTP Server Project is pleased to announce the release of Apache HTTP Server (httpd) version 2.2.19. This release represents fifteen years of innovation by the project, and is recommended over all previous releases!

For details see the [Official Announcement](#) and the [CHANGES-2.2](#) or condensed [CHANGES-2.2.19](#) lists.

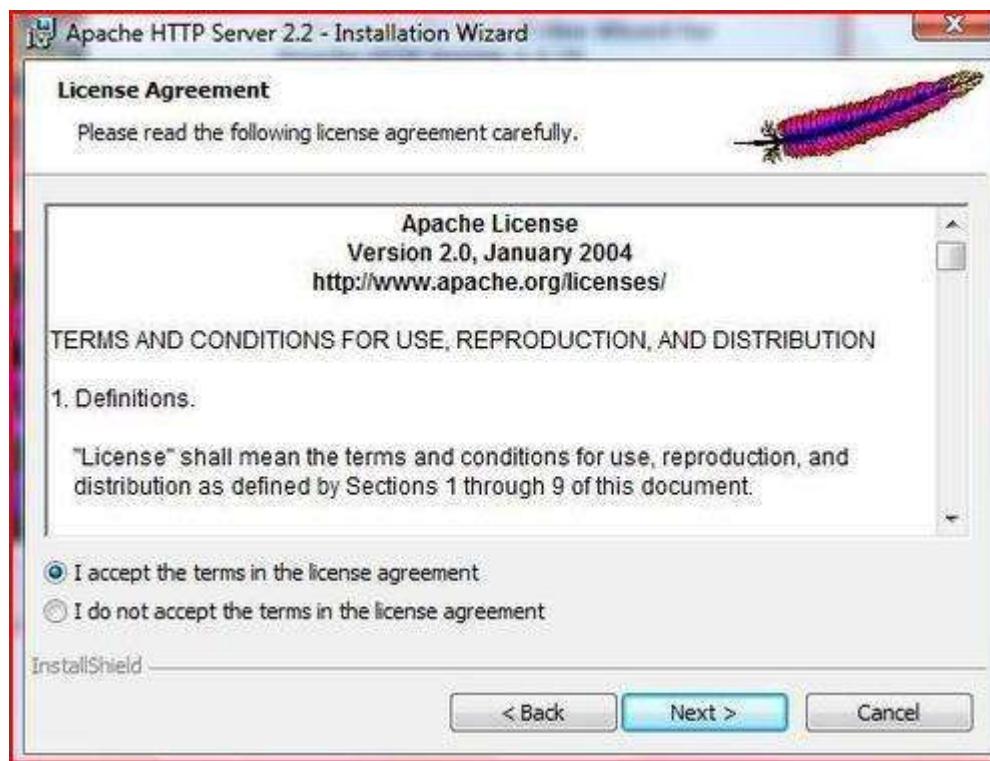
Add-in modules for Apache 2.0 are not compatible with Apache 2.2. If you are running third party add-in modules, you must obtain modules compiled or updated for Apache 2.2 from that third party, before you attempt to upgrade from these previous versions. Modules compiled for Apache 2.2 should continue to work for all 2.2.x releases.

- Unix Source: [httpd-2.2.19.tar.gz](#) [PGP] [MD5] [SHA1]
- Unix Source: [httpd-2.2.19.tar.bz2](#) [PGP] [MD5] [SHA1]
- Win32 Source: [httpd-2.2.19-win32-src.zip](#) [PGP] [MD5] [SHA1]
- Win32 Binary without crypto (no mod_ssl) (MSI Installer): [httpd-2.2.19-win32-x86-no_ssl.msi](#) [PGP] [MD5] [SHA1]
- Win32 Binary including OpenSSL 0.9.8 (MSI Installer): [httpd-2.2.19-win32-x86-openssl-0.9.8.msi](#) [PGP] [MD5] [SHA1]
- Other See:

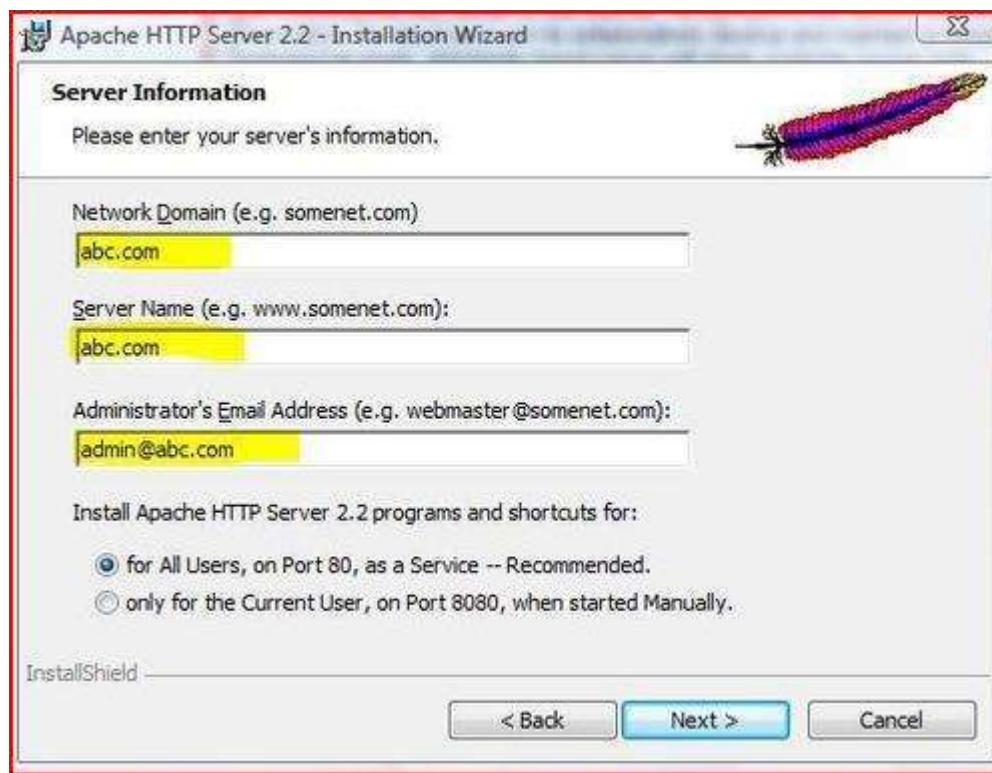
Apache HTTP installation on Windows



Apache HTTP installation on Windows

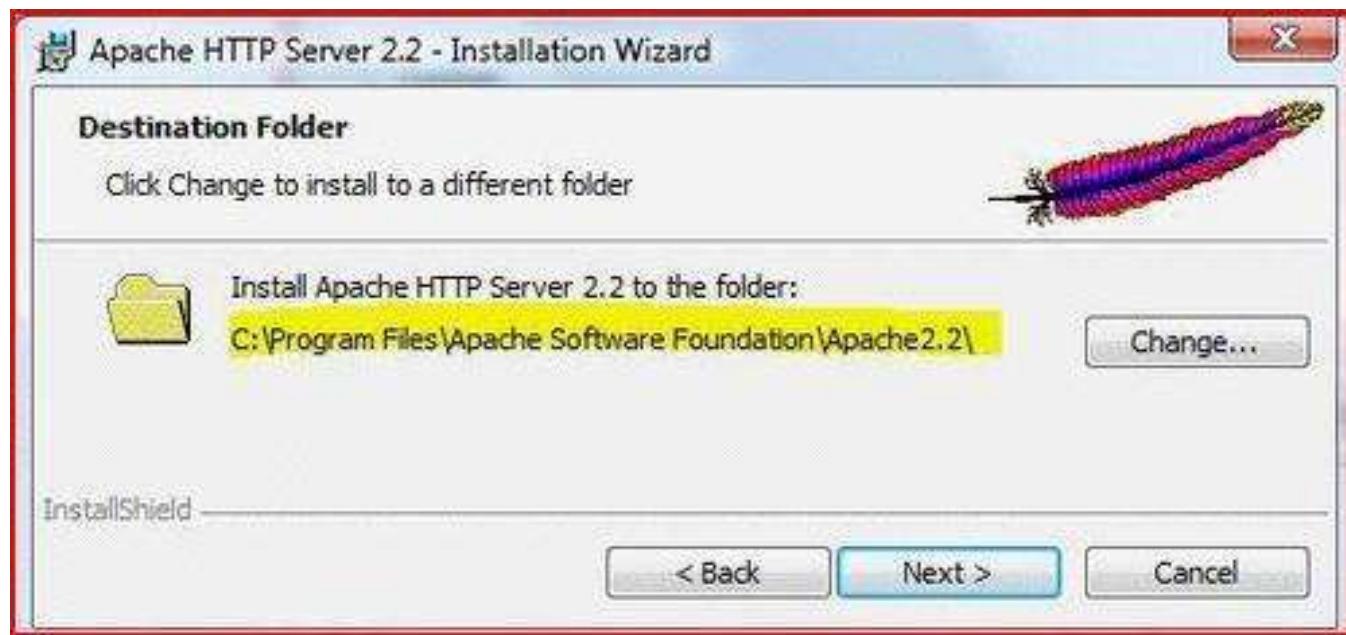


Apache HTTP installation on Windows





Apache HTTP installation on Windows



Apache HTTP installation on Windows



Apache HTTP installation on Windows



How to monitor the Apache service

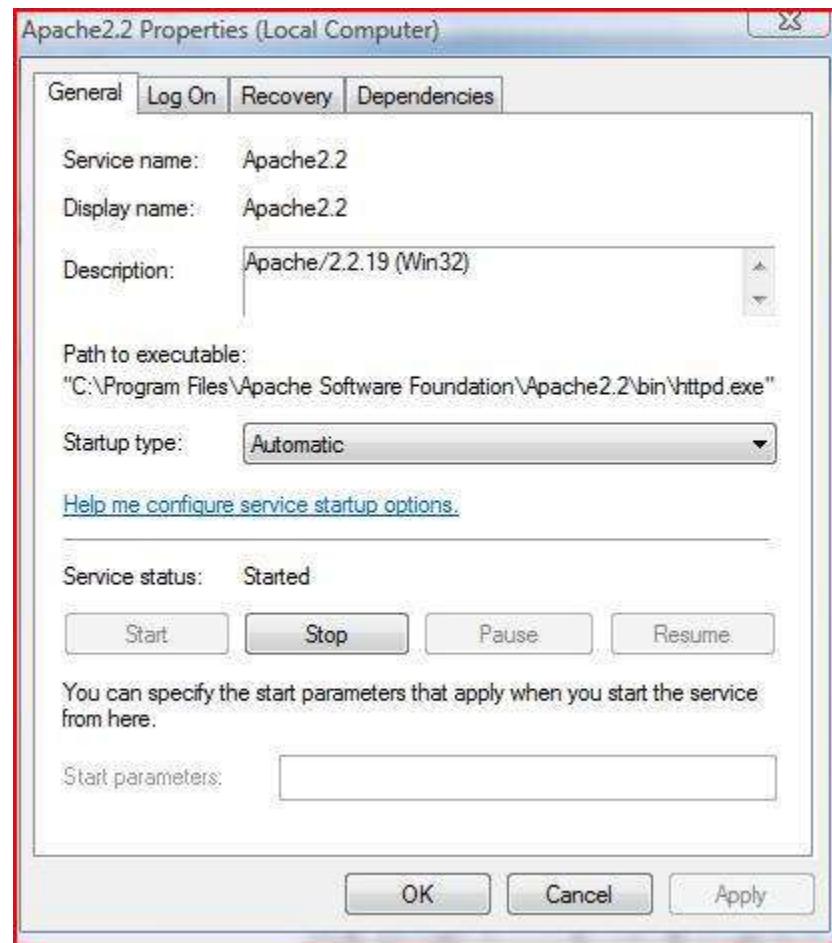
Once Apache is installed, you can start and stop the services in two ways:

- Apache monitoring service
- services.msc



How to monitor the Apache service





Installation of Apache HTTP on Linux

Apache HTTP Server (httpd) 2.2.19 is the best available version. 2011-05-22

The Apache HTTP Server Project is pleased to announce the release of Apache HTTP Server (httpd) version 2.2.19. This release represents fifteen years of innovation by the project, and is recommended over all previous releases!

For details see the [Official Announcement](#) and the [CHANGES 2.2](#) or condensed [CHANGES 2.2.19](#) lists.

Add-in modules for Apache 2.0 are not compatible with Apache 2.2. If you are running third party add-in modules, you must obtain modules compiled or updated for Apache 2.2 from that third party, before you attempt to upgrade from these previous versions. Modules compiled for Apache 2.2 should continue to work for all 2.2.x releases.

- Unix Source: [httpd-2.2.19.tar.gz](#) [PGP] [MD5] [SHA1]
- Unix Source: [httpd-2.2.19.tar.bz2](#) [PGP] [MD5] [SHA1]
- Win32 Source: [httpd-2.2.19-win32-src.zip](#) [PGP] [MD5] [SHA1]
- Win32 Binary without crypto (no mod_ssl) (MSI Installer): [httpd-2.2.19-win32-x86-no_ssl.msi](#) [PGP] [MD5] [SHA1]
- Win32 Binary including OpenSSL 0.9.8r (MSI Installer): [httpd-2.2.19-win32-x86-openssl-0.9.8r.msi](#) [PGP] [MD5] [SHA1]
- NetWare Binary: [apache-2.2.19-netware.nas](#) [PGP] [MD5] [SHA1]
- Other files

Installation of Apache HTTP on Linux

- Once the download is complete, the source file is stored in the home directory of the user (in our case it's /root).
- The source file comes in the form of tar.gz. Run the following command to unzip the source.
- First create a folder httpd and then extract the content in the httpd directory.

```
tar -zxvf httpd-2.2.X.tar.gz
```

 root@localhost:/opt

```
httpd-2.2.19/src/lib/apr-util/dbm/NWGNUdbmgdbm  
httpd-2.2.19/src/lib/apr-util/dbm/NWGNUMakefile  
httpd-2.2.19/src/lib/apr-util/dbm/apr_dbm_sdbm.c  
httpd-2.2.19/src/lib/apr-util/dbm/apr_dbm_db.mak  
httpd-2.2.19/src/lib/apr-util/dbm/NWGNUdbmdb  
httpd-2.2.19/src/lib/apr-util/dbm/apr_dbm_db.dep  
httpd-2.2.19/src/lib/apr-util/dbm/sdbm/  
httpd-2.2.19/src/lib/apr-util/dbm/sdbm/sdbm_lock.c  
httpd-2.2.19/src/lib/apr-util/dbm/sdbm/sdbm_private.h  
httpd-2.2.19/src/lib/apr-util/dbm/sdbm/sdbm.c  
httpd-2.2.19/src/lib/apr-util/dbm/sdbm/sdbm_pair.c  
httpd-2.2.19/src/lib/apr-util/dbm/sdbm/sdbm_hash.c  
httpd-2.2.19/src/lib/apr-util/dbm/sdbm/sdbm_tune.h  
httpd-2.2.19/src/lib/apr-util/dbm/sdbm/sdbm_pair.h  
httpd-2.2.19/src/lib/apr-util/dbm/apr_dbm.c  
httpd-2.2.19/src/lib/apr-util/aprutil.mak  
httpd-2.2.19/src/lib/apr-util/misc/  
httpd-2.2.19/src/lib/apr-util/misc/apu_dso.c  
httpd-2.2.19/src/lib/apr-util/misc/apr_thread_pool.c  
httpd-2.2.19/src/lib/apr-util/misc/apr_date.c  
httpd-2.2.19/src/lib/apr-util/misc/apr_queue.c  
httpd-2.2.19/src/lib/apr-util/misc/apu_version.c  
httpd-2.2.19/src/lib/apr-util/misc/apr_reslist.c  
httpd-2.2.19/src/lib/apr-util/misc/apr_rmm.c  
httpd-2.2.19/src/lib/apr-util/aprutil.dep  
httpd-2.2.19/src/lib/apr-util/libaprutil.rc  
httpd-2.2.19/src/lib/apr-util/apu-config.in
```

Installation of Apache HTTP on Linux

- You can check the directory using the following command.
- The result displayed is similar to the following screenshot:

ls -lthr

```
[root@localhost opt]# ls -lthr
total 124M
-rw-r--r-- 1 root root 81M May 16 20:28 jdk-6u24-linux-i586.bin
-rw-r--r-- 1 root root 7.3M May 16 20:29 apache-tomcat-7.0.12.tar
drwxr-xr-x 10 root root 4.0K May 16 20:35 apache-tomcat-7.0.12
drwxr-xr-x 11 root root 4.0K May 20 10:01 httpd-2.2.19
drwxr-xr-x  9 root root 4.0K Jun 23 02:23 httpd-2.2.19
-rw-r--r-- 1 root root 36M Jul 25 10:58 httpd-2.2.19.tar
[root@localhost opt]#
```

Installation of Apache HTTP on Linux

- Then access the extracted directory using the following command.
- The result is similar to the following screenshot:

```
cd httpd-2.2.19  
ls -ltrh
```

```
[root@localhost httpd-2.2.19]# ls -ltrh  
total 1.4M  
-rw-r--r-- 1 root root 403 Nov 21 2004 emacs-style  
-rw-r--r-- 1 root root 11K Nov 21 2004 config.layout  
-rw-r--r-- 1 root root 15K Nov 21 2004 ABOUT_APACHE  
-rw-r--r-- 1 root root 10K Mar 13 2005 ROADMAP  
-rw-r--r-- 1 root root 8.0K Oct 17 2005 VERSIONING  
-rw-r--r-- 1 root root 5.1K Nov 29 2005 LAYOUT  
-rw-r--r-- 1 root root 2.9K Dec 7 2006 InstallBin.dsp  
-rw-r--r-- 1 root root 5.9K Jan 9 2007 README  
-rw-r--r-- 1 root root 17K Jan 12 2007 libhttpd.dsp  
-rw-r--r-- 1 root root 2.6K Aug 23 2007 Buildall.dsp  
-rw-r--r-- 1 root root 29K Jan 18 2008 LICENSE  
-rw-r--r-- 1 root root 4.1K Jun 11 2008 httpd.dsp  
-rw-r--r-- 1 root root 4.7K Sep 18 2008 INSTALL  
-rw-r--r-- 1 root root 19K Nov 24 2008 acinclude.m4  
-rw-r--r-- 1 root root 8.6K Nov 25 2008 Makefile.in  
-rw-r--r-- 1 root root 928 Jan 5 2009 NOTICE  
-rw-r--r-- 1 root root 2.7K Jul 29 2009 BuildBin.dsp  
-rw-r--r-- 1 root root 5.3K Oct 13 2009 README_platforms  
-rw-r--r-- 1 root root 34K Oct 5 2010 Makefile.win  
-rw-r--r-- 1 root root 56K Oct 5 2010 Apache.dsw  
-rw-r--r-- 1 root root 2.5K Dec 20 2010 README-win32.txt  
-rwxr-xr-x 1 root root 5.7K Feb 9 04:13 buildconf  
-rw-r--r-- 1 root root 13K Apr 1 06:47 NWGNUmakefile  
-rw-r--r-- 1 root root 24K Apr 16 12:09 configure.in  
-rw-r--r-- 1 root root 28K May 6 10:28 libhttpd.mk  
-rw-r--r-- 1 root root 8.8K May 6 10:28 httpd.mk  
-rw-r--r-- 1 root root 30K May 6 21:37 libhttpd.dep  
-rw-r--r-- 1 root root 1.3K May 6 21:37 httpd.dep  
-rw-r--r-- 1 root root 114K May 20 09:54 CHANGES  
drwxr-xr-x 9 root root 4.0K May 20 09:59 .deps  
drwxr-xr-x 3 root root 4.0K May 20 10:00 .headers  
drwxr-xr-x 20 root root 4.0K May 20 10:00 .origins  
drwxr-xr-x 2 root root 4.0K May 20 10:00 .origins  
drwxr-xr-x 4 root root 4.0K May 20 10:00 .headers  
drwxr-xr-x 5 root root 4.0K May 20 10:00 .origins  
drwxr-xr-x 9 root root 4.0K May 20 10:00 .origins  
drwxr-xr-x 5 root root 4.0K May 20 10:01 .origins  
drwxr-xr-x 2 root root 4.0K May 20 10:01 .origins  
-rwxr-xr-x 1 root root 646K May 20 10:01 configure  
-rwxr-xr-- 1 root root 12K May 20 10:01 httpd.spec
```

Installation of Apache HTTP on Linux

- After the verification of the directory, it's time to install the Apache HTTP server on Linux. By default, the execution permission is not set to true on the source folder.
- For that, we have to run the chown command to make it true.

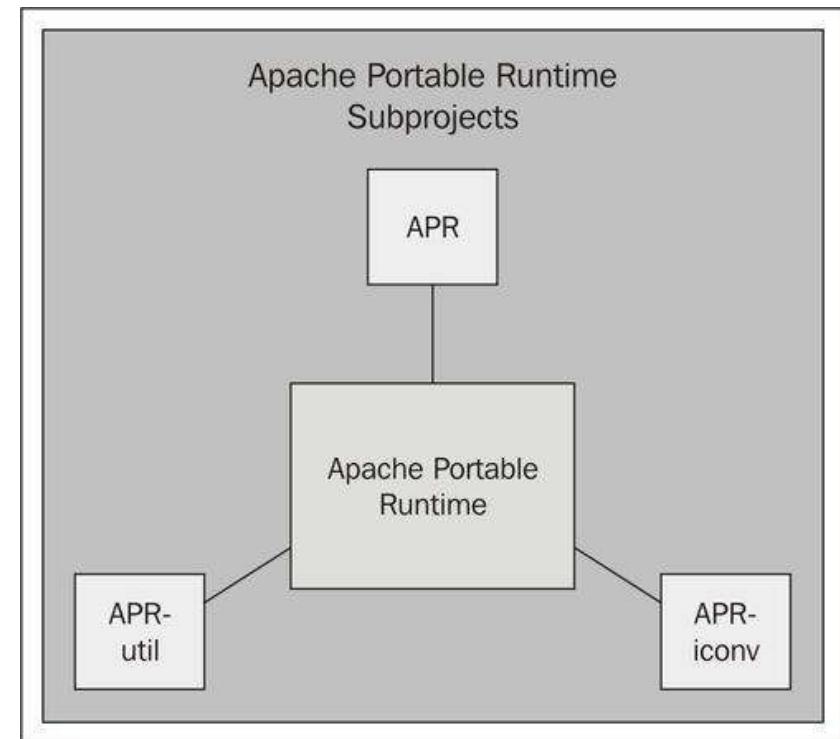
```
[root@localhost httpd-2.2.19]# chown 0755 configure
```

Apache Portable Runtime

- Apache Portable Runtime is an open-source project, which is supported by the Apache Foundation software.
- The main goal of this project is to provide the developer with an API, through which they can code and predict the identical behavior, regardless of different platforms.
- It eliminates the requirement of additional code dependency for different operating systems.

Apache Portable Runtime

- The Apache Portable Runtime project is again divided into three subprojects, to enhance and simplify the capability of this project.
- The following figure shows the different subprojects for APR:



Installation of APR/APR-util

APR/APR-util comes with the source of the Apache package, and they can be found in the following directories as shown in the following screenshot:

- APR: Installdir/src/lib/apr
- APR-util: Installdir/src/lib/apr-util

Installation of APR/APR-util

- As we have extracted the source in /opt/httpd-2.2.19, the source directory is also found in the same directory.

```
[root@localhost httpd-2.2.19]# cd srclib/
[root@localhost srclib]# ls -lrh
total 32K
drwxr-xr-x  4 root root 4.0K May 20 10:01 .
-rw-r--r--  1 root root 121 Feb 11  2005 Makefile.in
drwxr-xr-x 19 root root 4.0K May 20 10:01 .
drwxr-xr-x 25 root root 4.0K Jul 25 11:55 .
[root@localhost srclib]#
```

Installation of APR/APR-util

- Enter the source directory of apr and apr-util, then configure the code using the following commands:

```
[root@localhost srclib]# cd /opt/httpd-2.2.19/srclib/apr
[root@localhost apr]# ./configure --prefix=/opt/httpd/apr-
httpd/
[root@localhost apr-util]# ./configure --prefix=/usr/local/apr-
util-httpd/ --with-apr=/usr/local/apr-httpd/
Make
Make install
```

Installation of APR/APR-util

- You can configure Apache using the following command.
- The following screenshot shows the output of the command when executed:

```
[root@localhost httpd-  
2.2.19]#/configure --with-  
included-apr --  
prefix=/opt/apache-2.2.19
```

```
[root@localhost httpd-2.2.19]# ./configure --with-included-apr  
checking for chosen layout... Apache  
checking for working mkdir -p... yes  
checking build system type... i686-pc-linux-gnu  
checking host system type... i686-pc-linux-gnu  
checking target system type... i686-pc-linux-gnu  
  
Configuring Apache Portable Runtime library ...  
  
configuring package in srclib/apr now  
checking build system type... i686-pc-linux-gnu  
checking host system type... i686-pc-linux-gnu  
checking target system type... i686-pc-linux-gnu  
Configuring APR library  
platform: i686-pc-linux-gnu  
checking for working mkdir -p... yes  
APR Version: 1.4.5  
checking for chosen layout... apr  
checking for gcc... gcc  
checking for C compiler default output file name... a.out  
checking whether the C compiler works... yes  
checking whether we are cross compiling... no  
checking for suffix of executables...  
checking for suffix of object files... o  
checking whether we are using the GNU C compiler... yes  
checking whether gcc accepts -g... yes  
checking for gcc option to accept ISO C89... none needed  
checking for a sed that does not truncate output... /bin/sed  
applying APR hints file rules for i686-pc-linux-gnu  
setting CPPFLAGS to "-DLINUX=2"
```

Installation of APR/APR-util

Installation of APR/APR-util

- The previous screenshot describes the progress of the configure command.
- Once the command is executed, it will get the return code 0 otherwise you will see an error on the screen.
- Then, run the make command on the server to compile the code.
- The following figure shows the output of the make command:

```
[root@localhost httpd-2.2.X]#make
```

```

Making all in srclib
make[1]: Entering directory '/opt/httpd-2.2.19/srclib'
Making all in apr
make[2]: Entering directory '/opt/httpd-2.2.19/srclib/apr'
make[3]: Entering directory '/opt/httpd-2.2.19/srclib/apr'
/bin/sh /opt/httpd-2.2.19/srclib/apr/libtool --silent --mode=compile gcc -g -O2 -pthread -DAPACHE_CONFIG_H -DLINUX=2 -D REENTRANT -O 0
 SOURCE -I./include -I/opt/httpd-2.2.19/srclib/apr/include/arch/unix -I./include/arch/unix -I/opt/httpd-2.2.19/srclib/apr/include/a
2.19/srclib/apr/include -o passwd/apr_getpass.lo -o passwd/apr_getpass.o 4t touch passwd/apr_getpass.lo
/bin/sh /opt/httpd-2.2.19/srclib/apr/libtool --silent --mode=compile gcc -g -O2 -pthread -DAPACHE_CONFIG_H -DLINUX=2 -D REENTRANT -O 0
 SOURCE -I./include -I/opt/httpd-2.2.19/srclib/apr/include/arch/unix -I./include/arch/unix -I/opt/httpd-2.2.19/srclib/apr/include/a
2.19/srclib/apr/include -o strings/apr_cpystrn.lo -c strings/apr_cpystrn.c 4t touch strings/apr_cpystrn.lo
/bin/sh /opt/httpd-2.2.19/srclib/apr/libtool --silent --mode=compile gcc -g -O2 -pthread -DAPACHE_CONFIG_H -DLINUX=2 -D REENTRANT -O 0
 SOURCE -I./include -I/opt/httpd-2.2.19/srclib/apr/include/arch/unix -I./include/arch/unix -I/opt/httpd-2.2.19/srclib/apr/include/a
2.19/srclib/apr/include -o strings/apr_fmatch.lo -c strings/apr_fmatch.c 4t touch strings/apr_fmatch.lo
/bin/sh /opt/httpd-2.2.19/srclib/apr/libtool --silent --mode=compile gcc -g -O2 -pthread -DAPACHE_CONFIG_H -DLINUX=2 -D REENTRANT -O 0
 SOURCE -I./include -I/opt/httpd-2.2.19/srclib/apr/include/arch/unix -I./include/arch/unix -I/opt/httpd-2.2.19/srclib/apr/include/a
2.19/srclib/apr/include -o strings/APR_EGNATTF.lo -c strings/apr_egratft.c 4t touch strings/apr_egratft.lo
/bin/sh /opt/httpd-2.2.19/srclib/apr/libtool --silent --mode=compile gcc -g -O2 -pthread -DAPACHE_CONFIG_H -DLINUX=2 -D REENTRANT -O 0
 SOURCE -I./include -I/opt/httpd-2.2.19/srclib/apr/include/arch/unix -I./include/arch/unix -I/opt/httpd-2.2.19/srclib/apr/include/a
2.19/srclib/apr/include -o strings/apr_strings.lo -c strings/apr_strings.c 4t touch strings/apr_strings.lo
/bin/sh /opt/httpd-2.2.19/srclib/apr/libtool --silent --mode=compile gcc -g -O2 -pthread -DAPACHE_CONFIG_H -DLINUX=2 -D REENTRANT -O 0
 SOURCE -I./include -I/opt/httpd-2.2.19/srclib/apr/include/arch/unix -I./include/arch/unix -I/opt/httpd-2.2.19/srclib/apr/include/a
2.19/srclib/apr/include -o strings/apr_stmncmp.lo -c strings/apr_stmncmp.c 4t touch strings/apr_stmncmp.lo

make[1]: Leaving directory '/opt/httpd-2.2.19'
make[1]: Entering directory '/opt/httpd-2.2.19'
make[2]: Entering directory '/opt/httpd-2.2.19/modules'
make[3]: Entering directory '/opt/httpd-2.2.19/modules/mod_alias'
make[3]: Leaving directory '/opt/httpd-2.2.19/modules/mod_alias'
make[2]: Leaving directory '/opt/httpd-2.2.19/modules'
make[1]: Leaving directory '/opt/httpd-2.2.19'

make[1]: Entering directory '/opt/httpd-2.2.19/modules/mappers'
make[2]: Entering directory '/opt/httpd-2.2.19/modules'
make[3]: Entering directory '/opt/httpd-2.2.19'
make[3]: Leaving directory '/opt/httpd-2.2.19'
make[2]: Leaving directory '/opt/httpd-2.2.19/modules'
make[1]: Leaving directory '/opt/httpd-2.2.19'

make[1]: Leaving directory '/opt/httpd-2.2.19'

```

Installation of APR/APR-util

- The previous and the following screenshots show the completion without any error.
 - To proceed with the installation of make, we have to run the following command:

[root@localhost httpd-2.2.X]#make install

Installation of APR/APR-util

```
Installing configuration files
mkdir /opt/apache-2.2.19/conf
mkdir /opt/apache-2.2.19/conf/extra
mkdir /opt/apache-2.2.19/conf/original
mkdir /opt/apache-2.2.19/conf/original/extra
Installing HTML documents
mkdir /opt/apache-2.2.19/htdocs
Installing error documents
mkdir /opt/apache-2.2.19/error
Installing icons
mkdir /opt/apache-2.2.19/icons
mkdir /opt/apache-2.2.19/logs
Installing CGIs
mkdir /opt/apache-2.2.19/cgi-bin
Installing header files
Installing build system files
Installing man pages and online manual
mkdir /opt/apache-2.2.19/man
mkdir /opt/apache-2.2.19/man/man1
mkdir /opt/apache-2.2.19/man/man8
mkdir /opt/apache-2.2.19/manual
make[1]: Leaving directory '/opt/httpd-2.2.19'
```

Installation of APR/APR-util

```
[root@... apache-2.2.19]# ls -ltrh
total 60K
drwxr-xr-x  2 root root 4.0K May 20 12:59 .
drwxr-xr-x 14 root root 12K May 20 13:01 ..
drwxr-xr-x  3 root root 4.0K Jul 25 16:05 lib
drwxr-xr-x  2 root root 4.0K Jul 25 16:05 modules
drwxr-xr-x  2 root root 4.0K Jul 25 16:05 bin
drwxr-xr-x  4 root root 4.0K Jul 25 16:05 conf
drwxr-xr-x  3 root root 4.0K Jul 25 16:05 logs
drwxr-xr-x  2 root root 4.0K Jul 25 16:05 temp
drwxr-xr-x  3 root root 4.0K Jul 25 16:05 srclib
drwxr-xr-x  2 root root 4.0K Jul 25 16:05 srclib-2.0
drwxr-xr-x  2 root root 4.0K Jul 25 16:05 srclib-2.1
drwxr-xr-x  4 root root 4.0K Jul 25 16:06 sbin
drwxr-xr-x  2 root root 4.0K Jul 25 16:06 test
```

Installation of APR/APR-util

- Before we end the installation, it is necessary to start the services of HTTP to verify the instance is properly installed.
- The best way to check the configuration is by running the configtest script.
- This script comes by default with Apache httpd, only in a non-DOS environment.
- The script can be found in APACHE_HOME/bin.

```
[root@localhost bin]# ./apachectl configtest  
Syntax OK
```

Installation of APR/APR-util

- Then restart Apache using the following command:

```
[root@root@localhost bin]# ./apachectl start
```

Installation of APR/APR-util

- Once you start Apache, it's very important to verify the instance status.
- You can verify the system using the ps command:

ps -ef |grep httpd

| | | | | | | | |
|--------|------|-------|---|-------|-------|----------|---------------------------------------|
| root | 6394 | 1 | 0 | 16:11 | ? | 00:00:00 | /opt/apache-2.2.19/bin/httpd -k start |
| daemon | 6395 | 6394 | 0 | 16:11 | ? | 00:00:00 | /opt/apache-2.2.19/bin/httpd -k start |
| daemon | 6396 | 6394 | 0 | 16:11 | ? | 00:00:00 | /opt/apache-2.2.19/bin/httpd -k start |
| daemon | 6397 | 6394 | 0 | 16:11 | ? | 00:00:00 | /opt/apache-2.2.19/bin/httpd -k start |
| daemon | 6398 | 6394 | 0 | 16:11 | ? | 00:00:00 | /opt/apache-2.2.19/bin/httpd -k start |
| daemon | 6399 | 6394 | 0 | 16:11 | ? | 00:00:00 | /opt/apache-2.2.19/bin/httpd -k start |
| root | 6343 | 27394 | 0 | 16:11 | pts/1 | 00:00:00 | grep httpd |

Apache Jserv protocol

- This protocol was mainly developed to transfer data over the network in binary format instead of plain text. It uses TCP and a packet-based protocol, hence, increasing the performance of the web servers.
- Another informational point is that decryption of requests is done on the web server end so that the application server doesn't have a high load.

Installation and configuration of mod_jk

- mod_jk is an AJP connector which is used to integrate web servers such as Apache or IIS to Tomcat 8.
- In case we don't install mod_jk, then we cannot use frontend web servers for Tomcat.
- This module is very helpful in order to hide Tomcat behind the frontend web server and also eliminates the port number while browsing the URL.
- It involves multiple steps starting from installation and configuration.

Installation of mod_jk

- Once the source is downloaded, we have to extract it in the server directory using the following command:

```
[root@localhost opt]# tar -zxvf tomcat-connectors-1.2.x-src.tar
```

where x is the minor version number.

Installation of mod_jk

```
total 52K
-rw-r--r--  1 root bin  14K May  4  2008 LICENSE
-rw-r--r--  1 root bin 269 Jan  3  2011 NOTICE
-rw-r--r--  1 root bin 1.5K Jun 30 15:16 BUILD.txt
drwxr-xr-x  4 root bin 4.0K Jul  2 01:47 conf
drwxr-xr-x  6 root bin 4.0K Jul  2 01:47 logs
drwxr-xr-x  2 root bin 4.0K Jul  2 01:47 temp
drwxr-xr-x  9 root bin 4.0K Jul  2 01:47 work
drwxr-xr-x 10 root bin 4.0K Jul  2 01:47 worker
drwxr-xr-x  2 root bin 4.0K Jul  2 01:47 workers
drwxr-xr-x 11 root bin 4.0K Jul  2 01:47 workmanager
```

Installation of mod_jk

- Go to the native directory of the mod_jk source using the following command and then run the configure command:

```
[root@localhost opt]# cd /opt/tomcat-connectors-1.2.32-  
src/native ./configure --with-apxs=/opt/apache-  
2.2.19/bin/apxs
```

Installation of mod_jk

```
[root@native]# ./configure --with-apxs=/opt/apache-2.2.19/bin/apxs
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking target system type... i686-pc-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking for test... /usr/bin/test
checking for rm... /bin/rm
checking for grep... /bin/grep
checking for echo... /bin/echo
checking for sed... /bin/sed
checking for cp... /bin/cp
checking for mkdir... /bin/mkdir
need to check for Perl first, apxs depends on it...
checking for perl... /usr/bin/perl
APRINCLUDEDIR is -I/opt/apache-2.2.19/include -I/opt/apache-2.2.19/include
building connector for "apache-2.0"
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ANSI C... none needed
checking for style of include used by make... GNU
checking dependency style of gcc... none
checking for a sed that does not truncate output... /bin/sed
```

100%

- Once the configuration is done, you need to run the make command, which compiles the source code, as shown in the following screenshot:

[root@localhost apache-2.0]# make

```
root@...:~ native]# make
skipping all in common
make[1]: Entering directory '/opt/tomcat-connectors-1.2.32-src/native/common'
/opt/apache-2.2.19/build/libtool --silent --mode=compile gcc -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -g -O2 -pthread
cude -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -DLINUX=2 -D REENTRANT -D GNU_SOURCE -D LARGEFILE64_SOURCE -I /opt/jk
1/include/ -c jk_ajp12_worker.c -o jk_ajp12_worker.lo
/opt/apache-2.2.19/build/libtool --silent --mode=compile gcc -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -g -O2 -pthread
cude -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -DLINUX=2 -D REENTRANT -D GNU_SOURCE -D LARGEFILE64_SOURCE -I /opt/jk
1/include/ -c jk_connect.c -o jk_connect.lo
/opt/apache-2.2.19/build/libtool --silent --mode=compile gcc -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -g -O2 -pthread
cude -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -DLINUX=2 -D REENTRANT -D GNU_SOURCE -D LARGEFILE64_SOURCE -I /opt/jk
1/include/ -c jk_msg_buf.c -o jk_msg_buf.lo
/opt/apache-2.2.19/build/libtool --silent --mode=compile gcc -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -g -O2 -pthread
cude -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -DLINUX=2 -D REENTRANT -D GNU_SOURCE -D LARGEFILE64_SOURCE -I /opt/jk
1/include/ -c jk_util.c -o jk_util.lo
100%
/opt/apache-2.2.19/build/libtool --silent --mode=compile gcc -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -g -O2 -pthread
cude -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -DLINUX=2 -D REENTRANT -D GNU_SOURCE -D LARGEFILE64_SOURCE -I /opt/jk
1/include/ -c jk_ajp13.c -o jk_ajp13.lo
/opt/apache-2.2.19/build/libtool --silent --mode=compile gcc -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -g -O2 -pthread
cude -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -DLINUX=2 -D REENTRANT -D GNU_SOURCE -D LARGEFILE64_SOURCE -I /opt/jk
1/include/ -c jk_pool.c -o jk_pool.lo
/opt/apache-2.2.19/build/libtool --silent --mode=compile gcc -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -g -O2 -pthread
cude -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -DLINUX=2 -D REENTRANT -D GNU_SOURCE -D LARGEFILE64_SOURCE -I /opt/jk
1/include/ -c jk_worker.c -o jk_worker.lo
/opt/apache-2.2.19/build/libtool --silent --mode=compile gcc -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -g -O2 -pthread
cude -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -DLINUX=2 -D REENTRANT -D GNU_SOURCE -D LARGEFILE64_SOURCE -I /opt/jk
1/include/ -c jk_ajp13_worker.c -o jk_ajp13_worker.lo
/opt/apache-2.2.19/build/libtool --silent --mode=compile gcc -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -g -O2 -pthread
cude -I/opt/apache-2.2.19/include -D HAVE_CONFIG_H -DLINUX=2 -D REENTRANT -D GNU_SOURCE -D LARGEFILE64_SOURCE -I /opt/jk
1/include/ -c jk_lb_worker.c -o jk_lb_worker.lo
```

- After the code is compiled using the make command then installation of the code is done using the command make install:

root@localhost apache-2.0]# make install

```
make[1]: Leaving directory '/opt/tomcat-connectors-1.2.32-src/native'
target="all"; \
    list='common apache-2.0'; \
    for i in $list; do \
        echo "Making $target in $i"; \
        if test "$i" := "."; then \
            (cd $i && make $target) || exit 1; \
        fi; \
    done;
Making all in common
make[1]: Entering directory '/opt/tomcat-connectors-1.2.32-src/native/common'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory '/opt/tomcat-connectors-1.2.32-src/native/common'
Making all in apache-2.0
make[1]: Entering directory '/opt/tomcat-connectors-1.2.32-src/native/apache-2.0'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory '/opt/tomcat-connectors-1.2.32-src/native/apache-2.0'
```

100%

Installation of mod_jk

```
total 2.3M
-rw-r--r-- 1 root bin    11K Jun 21  2007 bldjk.qclsrc
-rw-r--r-- 1 root bin    11K Jun 21  2007 bldjk54.qclsrc
-rw-r--r-- 1 root bin   1.4K Sep 13  2010 config.m4
-rw-r--r-- 1 root bin   12K Sep 14  2010 mod_jk.dsp
-rw-r--r-- 1 root bin   3.0K Oct 21  2010 Makefile.in
-rw-r--r-- 1 root bin   1.5K Oct 21  2010 Makefile.apxs.in
-rw-r--r-- 1 root bin   6.5K Mar 18 02:05 NWGNUmakefile
-rw-r--r-- 1 root bin  129K May 23 12:03 mod_jk.c
-rw-r--r-- 1 root bin   7.0K Jun 30 12:13 Makefile.vc
-rw-r--r-- 1 root root  1.6K Jul 25 16:30 Makefile.apxs
-rw-r--r-- 1 root root  3.2K Jul 25 16:30 Makefile
-rw-r--r-- 1 root root 124K Jul 25 16:33 mod_jk.o
-rw-r--r-- 1 root root  309 Jul 25 16:33 mod_jk.lo
-rwxr-xr-x 1 root root 858K Jul 25 16:33 mod_jk.so
-rw-r--r-- 1 root root  788 Jul 25 16:33 mod_jk.la
-rw-r--r-- 1 root root 1.1M Jul 25 16:33 mod_jk.a
```

Configuration of mod_jk in Apache

- Copy the mod_jk.so from the apache 2.0 directory of the connector source to the modules directory of the Apache httpd server by using the following command:

```
[root@localhost apache-2.0]# cp mod_jk.so /opt/apache-  
2.2.19/modules/  
chmod 755 mod_jk.so
```

- The previous command sets the execution permission.
`chown root:root mod_jk.so`
- The previous command sets the ownership to root.

Configuration of mod_jk in Apache

- To edit the configuration of the httpd server, you have to create the new file called as mod_jk.conf in the conf directory of \$APACHE_HOME/conf as follows:

```
[root@localhost apache-2.0]# cd /opt/apache-2.2.19/conf  
vi mod-jk.conf  
LoadModule jk_module modules/mod_jk.so  
JkWorkersFile conf/workers.properties  
JkLogFile logs/mod_jk.log  
JkLogLevel info  
JkMount /sample/* node1
```

Configuration of mod_jk in Apache

- Log level: This parameter captures the different events performed by mod_jk in the logs (JkLogLevel info).

```
LoadModule jk_module modules/mod_jk.so
JkWorkersFile conf/workers.properties
JkLogFile logs/mod_jk.log
JkLogLevel info
JkMount /sameple/* node1
JkMount /* node1
```

Configuration of mod_jk in Apache

- Create a new file named as workers.properties in the conf using the following command:

```
[root@localhost conf]# vi workers.properties
worker.list=node1
worker.node1.port=8009
worker.node1.host=10.130.240.51
worker.node1.type=ajp13
worker.node1.lbfactor=1
```

Configuration of mod_jk in Apache

```
worker.list=node1
worker.node1.port=8009
worker.node1.host=10.130.240.51
worker.node1.type=ajp13
worker.node1.lbfactor=1
worker.node1.cachesize=10
```

- The last step is to include the mod_jk.conf in the main configuration file of the httpd, that is, httpd.conf.

[root@localhost conf]# vi httpd.conf

Configuration of mod_jk in Apache

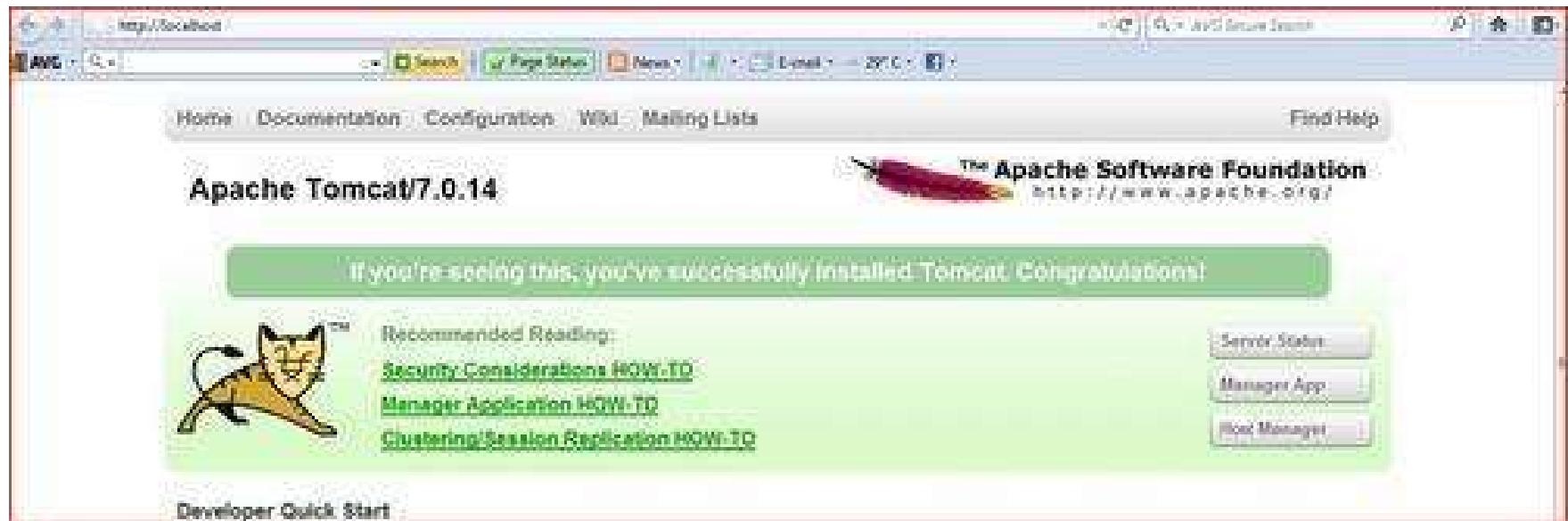
- Now we are done with configuration of mod_jk in the Apache HTTP configuration file (httpd.conf).
- But mod_jk will not work until we recycle the Apache httpd services. So why wait? Let's recycle by running the following command:

```
[root@localhost bin]# ./apachectl stop  
[root@root@localhost bin]# ./apachectl start
```

Configuration of mod_jk in Apache



Configuration of mod_jk in Apache



mod_proxy configuration

- Place the following lines of code after the other LoadModule directives:

```
LoadModule proxy_module modules/mod_proxy.so  
LoadModule proxy_http_module  
modules/mod_proxy_http.so
```

- Place the following lines of code with your other VirtualHost, or at the bottom of the file:

```
NameVirtualHost *
<VirtualHost *>
ServerName abc.com
ProxyRequests Off
<Proxy *>
Order deny,allow
Allow from all
</Proxy>
ProxyPass / http://localhost:8080/
ProxyPassReverse / http://localhost:8080/
<Location />
Order allow,deny
Allow from all
</Location>
</VirtualHost>
```

mod_proxy configuration

- Save the configuration file. Based on the Rule of Thumb, every configuration change is reflected only after a recycle.

```
[root@localhost bin]# ./apachectl stop  
[root@root@localhost bin]# ./apachectl start
```

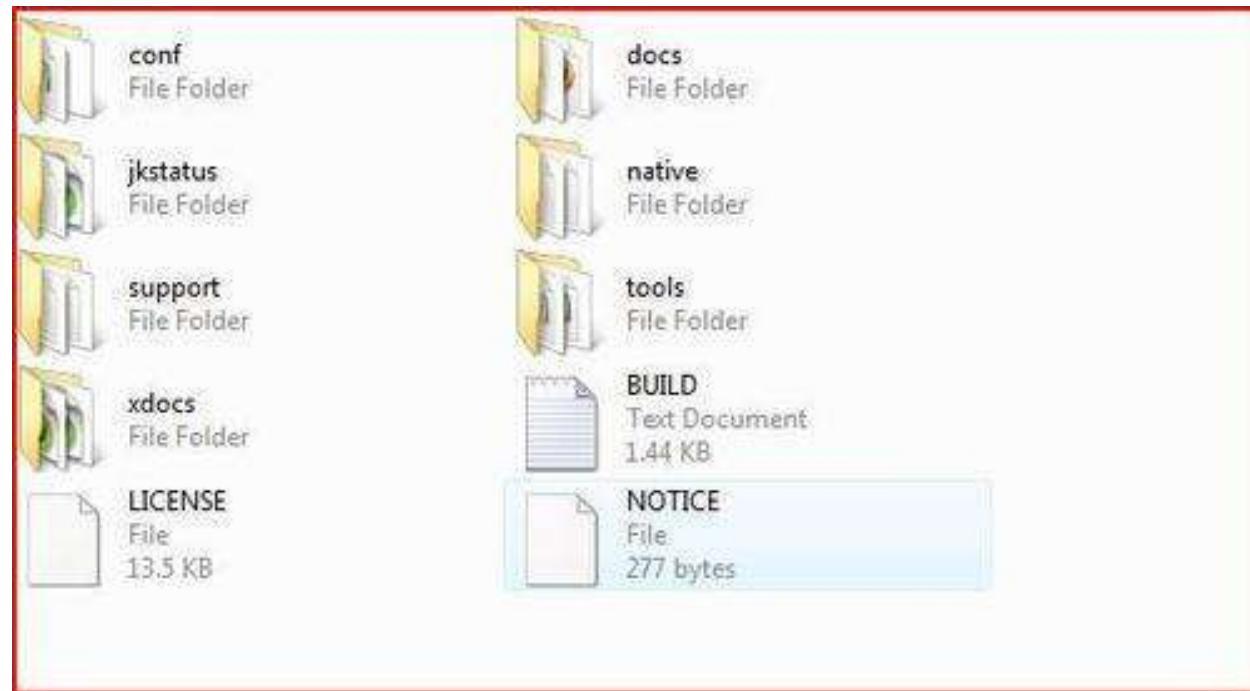
Comparison between mod_jk and mod_proxy

| Feature | mod_jk | mod_proxy |
|----------------------|------------------|--|
| Load balancing | High level | Basic |
| Management interface | Yes | No |
| Compilation | Separate process | Not required. By default comes with Apache |
| Configuration | Huge | Basic |
| Protocol | AJP | HTTP/HTTPS/AJP |
| Node failure | Advance | NA |

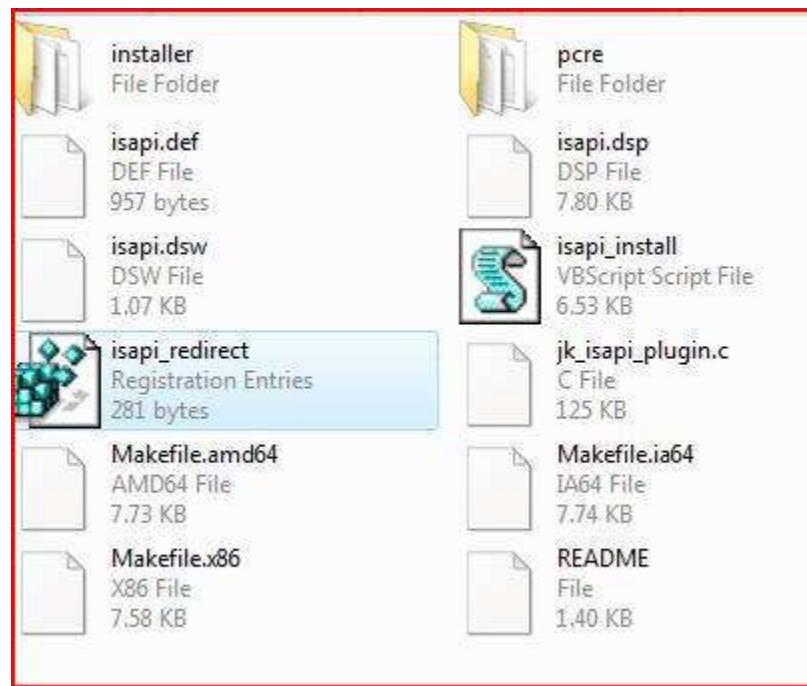
IIS integration with Tomcat 8

- IIS versions vary with different versions of Windows OS, such as Windows 2003 comes with IIS 6 and Windows 2008 comes with IIS 7.
- Here, we discuss the integration of IIS 7 with Tomcat 8, but before we proceed, there are some sets of configuration which we need to configure to make sure the integration works well.

Steps for installation



Steps for installation



Steps for installation

- Let us quickly see the contents in the registry by editing the file:

REGEDIT4

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software Foundation\Jakarta Isapi Redirector\1.0]
"log_file"="C:\\tomcat\\logs\\isapi.log"
"log_level"="debug"
"worker_file"="C:\\tomcat\\conf\\workers.properties"
"worker_mount_file"="C:\\tomcat\\conf\\uriworkermap.properties"
```

Steps for installation

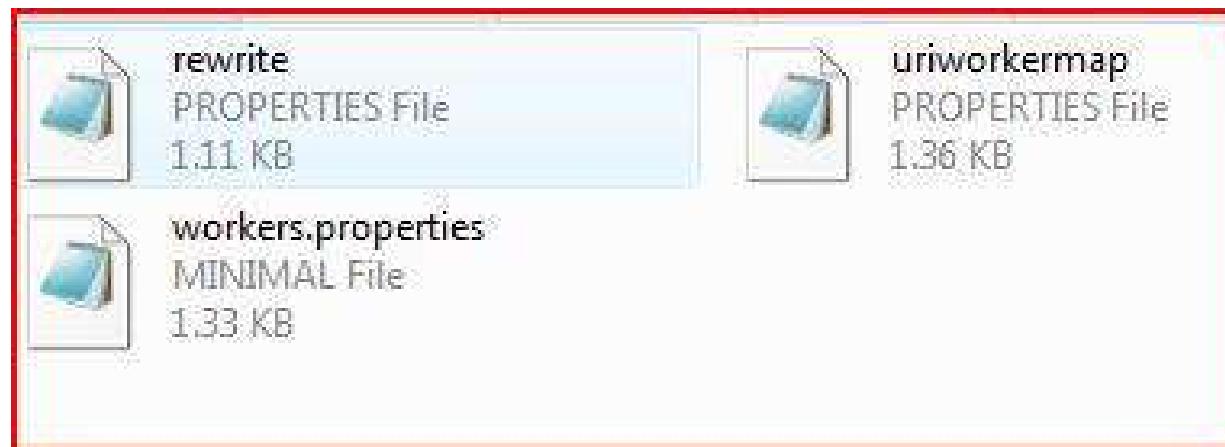
- Once the registry is saved, you have to restart the IIS server using the following commands from the command prompt:

`iisreset stop`

`iisreset start`

Steps for installation

- The following screenshot shows the different configuration files, which are used for mod_jk configuration in IIS:



- The workers.properties file consists of the following code:

workers.properties

```
# The workers that jk should create and work with
#worker.list=wlb,jkstatus
# Defining a worker named ajp13w and of type ajp13
# Note that the name and the type do not have to match.
worker.ajp13w.type=ajp13
worker.ajp13w.host=localhost
worker.ajp13w.port=8009
# Defining a load balancer
worker.wlb.type=lb
worker.wlb.balance_workers=ajp13w
# Define status worker
worker.jkstatus.type=status
```

- The uriworkermap.properties file consists of the following code:

```
uriworkermap.properties
# uriworkermap.properties - IIS
# This file provides sample mappings for example wlb
# worker defined in workermap.properties.minimal
# The general syntax for this file is:
# [URL]=[Worker name]
/admin/*=wlb
/manager/*=wlb
/jsp-examples/*=wlb
/servlets-examples/*=wlb
/examples/*=wlb
# Optionally filter out all .jpeg files inside that context
# For no mapping the url has to start with exclamation (!)
!/servlets-examples/*.jpeg=wlb
# Mount jkstatus to /jkmanager
# For production servers you will need to
# secure the access to the /jkmanager url
/jkmanager=jkstatus
```

Common issues and troubleshooting for integration

- Scenario 1: The httpd server is not able to compile, and this results in the exit from the compilation mode.

Error:

```
configure: error: in `/opt/httpd-2.2.19/src/lib/apr':  
configure: error: no acceptable C compiler found in $PATH  
See `config.log' for more details.  
configure failed for src/lib/apr
```

- Reason: C compilers are missing, such as the GCC and GCC+.

- Solution: Download the GCC compiler from the Internet and compile it, as per the instructions given there:

```
[root@localhost httpd-2.2.19]# ./configure
checking for chosen layout... Apache
checking for working mkdir -p... yes
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking target system type... i686-pc-linux-gnu

Configuring Apache Portable Runtime library ...

checking for APR... reconfig
configuring package in srclib/apr now
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking target system type... i686-pc-linux-gnu
Configuring APR library
Platform: i686-pc-linux-gnu
checking for working mkdir -p... yes
APR Version: 1.4.5
checking for chosen layout... apr
checking for gcc... no
checking for cc... no
checking for cl.exe... no
configure: error: in '/opt/httpd-2.2.19/srclib/apr':
configure: error: no acceptable C compiler found in $PATH
See 'config.log' for more details.
configure failed for srclib/apr
```

Common issues and troubleshooting for integration

- Scenario 2: Apache is not able to compile the make command, displaying an error and exiting the process.
- Error: make is not able to compile the code.
- Reason: make is not able to execute its functions.
- Solution: Run the following command:

make clean

#Then

make

make install

Common issues and troubleshooting for integration

```
gcc -c -DHAVE_CONFIG_H -DLINUX=2 -D_REENTRANT -D_GNU_SOURCE -D_LARGEFILE64_SOURCE -I/opt/httpd-2.2.19/src/lib-util/include/private -I/opt/httpd-2.2.19/src/lib/apr/include -I/opt/httpd-2.2.19/src/lib/apr-util/xml/expat -I/opt/httpd-2.2.19/src/lib/apr-util/xml/expat -o ./aprutil.exp ./aprutil.c
gcc -E -DHAVE_CONFIG_H -DLINUX=2 -D_REENTRANT -D_GNU_SOURCE -D_LARGEFILE64_SOURCE -I/opt/httpd-2.2.19/src/lib-util/include/private -I/opt/httpd-2.2.19/src/lib/apr/include -I/opt/httpd-2.2.19/src/lib/apr-util/xml/expat -o ./aprutil.o ./aprutil.c
sed 's,\`location=\`.*\`linstalled,' < apu-1-config > apu-config.out
make[3]: Leaving directory `/opt/httpd-2.2.19/src/lib/apr-util'
make[2]: Leaving directory `/opt/httpd-2.2.19/src/lib/apr-util'
Making all in pcre
make[2]: Entering directory `/opt/httpd-2.2.19/src/lib/pcre'
Makefile:7: /build/l1tlib.mk: No such file or directory
make[2]: *** No rule to make target `/build/l1tlib.mk'. Stop.
make[2]: Leaving directory `/opt/httpd-2.2.19/src/lib/pcre'
make[1]: *** [all-recursiv] Error 1
make[1]: Leaving directory `/opt/httpd-2.2.19/src/lib'
```

Common issues and troubleshooting for integration

- Scenario 3: The Apache HTTP server is unable to connect to Tomcat 8.
- Error: Unable to connect through AJP.
- Reason: Port might get blocked or the AJP configuration is not correct.
- Solution: Check the logs for more errors using the following command:

Refer to the file 4_1.txt

Common issues and troubleshooting for integration

- Then, run the configtest command on the server to verify the configuration using the following command:

```
[root@root@localhost bin]# ./apachectl configtest  
Syntax OK
```

Summary

- In this lesson, we have discussed the integration of Apache/IIS with Tomcat 8 and their various component integrations.
- Also, you can use ready made solutions for Apache and Tomcat using the link
<http://www.apachefriends.org/en/xampp-windows.html>. After reading this lesson, the reader can expect to have a good command on the integration and the different issues they may encounter during the installation of the integration.

5. Securing Tomcat 8



Securing Tomcat 8

In this lesson, we will discuss the following topics:

- Tomcat security permissions
- Catalina properties
- SSL implementation on Tomcat 8

Tomcat Manager

- The security being a major concern for IT companies, a separate department for IT security administration is created in every company.
- Their major responsibility is to make sure that there are no vulnerabilities in terms of the networks, web, and OS infrastructure.
- We should download Tomcat from the Tomcat website or any secure, known host.

Tomcat security permissions

- Apache Tomcat comes with good security-enabled options, but every environment has its own requirement for security, based on the usage of the application.
- For example, banking sites require a high level of security, on the other hand, user-based applications require little security.

catalina.properties

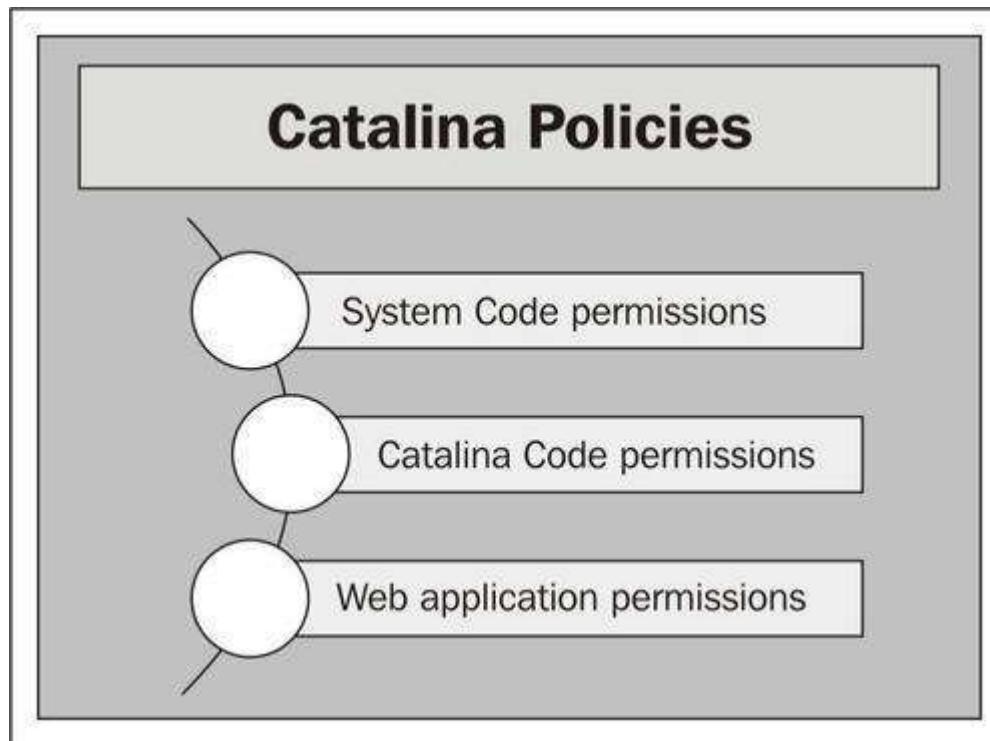
- This file contains information related to the access of the package, package definition, common loader, shared loader, and a list of JAR files, which are not necessary to be scanned at the startup of Tomcat.
- It helps in improving the performance, as adding too many JAR files to the skip list improves memory consumption.
- If you want to add any common JAR, you have to define it under catalina.properties.

Tomcat security permissions

- Following are the key syntaxes used in catalina.properties:

Refer to the file 5_1.txt

catalina.policy



System Code permissions

- This policy gives you access the Java library, which need to be verified at runtime by the Tomcat instance.
- Following code shows that full access permissions are granted for the Java library:

```
grant codeBase "file:${java.home}/lib/-" {permission  
java.security.AllPermission;
```

System Code permissions

```
// ----- SYSTEM CODE PERMISSIONS -----  
  
// These permissions apply to javac  
grant codeBase "file:${java.home}/lib/-" {  
    permission java.security.AllPermission;  
}  
  
// These permissions apply to all shared system extensions  
grant codeBase "file:${java.home}/jre/lib/ext/-" {  
    permission java.security.AllPermission;  
}  
  
// These permissions apply to javac when ${java.home} points at $JAVA_HOME/jre  
grant codeBase "file:${java.home}/../lib/-" {  
    permission java.security.AllPermission;  
}  
  
// These permissions apply to all shared system extensions when  
// ${java.home} points at $JAVA_HOME/jre  
grant codeBase "file:${java.home}/lib/ext/-" {  
    permission java.security.AllPermission;  
}
```

Catalina Code permissions (Tomcat core permission)

- This section contains the Tomcat internal file permissions to access the code.
- It helps in controlling the internal functionality of Tomcat.
- The following policy shows that Catalina/lib has given all the permissions:

```
grant codeBase "file:${catalina.home}/lib/-" {permission  
java.security.AllPermission;
```

```
// ----- CATALINA CODE PERMISSIONS -----  
  
// These permissions apply to the daemon code  
grant CodeBase "file:${catalina.home}/bin/commons-daemon.jar" {  
    permission java.security.AllPermission;  
}  
  
// These permissions apply to the logging API  
// Note: If tomcat-juli.jar is in ${catalina.base} and not in ${catalina.home},  
// update this section accordingly.  
// grant codeBase "file:${catalina.base}/bin/tomcat-juli.jar" {...}  
grant codeBase "file:${catalina.home}/bin/tomcat-juli.jar" {  
    permission java.io.FilePermission  
        "C:/java.home${file.separator}lib${file.separator}logging.properties", "read";  
  
    permission java.io.FilePermission  
        "${catalina.base}${file.separator}conf${file.separator}logging.properties", "read";  
    permission java.io.FilePermission  
        "${catalina.base}${file.separator}logs", "read, write";  
    permission java.io.FilePermission  
        "${catalina.base}${file.separator}logs${file.separator}*", "read, write";  
  
    permission java.lang.RuntimePermission "shutdownHooks";  
    permission java.lang.RuntimePermission "getClassLoader";  
    permission java.lang.RuntimePermission "setContextClassLoader";  
  
    permission java.util.logging.LoggingPermission "control";  
  
    permission java.util.PropertyPermission "java.util.logging.config.class", "read";  
    permission java.util.PropertyPermission "java.util.logging.config.file", "read";  
    permission java.util.PropertyPermission "catalina.base", "read";
```

Web application permissions

- This section contains the policy with reference to the application's resource utilization, such as JVM, JNDI, and so on.
- If you enable the following code, then the Tomcat classes can be accessed from the root directory of the code:

```
// grant codeBase "file:${catalina.base}/webapps/examples/  
WEB-INF/classes/-" {/ / };
```

```
// ----- WEB APPLICATION PERMISSIONS -----  
  
// These permissions are granted by default to all web applications  
// In addition, a web application will be given a read FilePermission  
// and JndiPermission for all files and directories in its document root.  
grant {  
    // Required for JNDI lookup of named JDBC DataSource's and  
    // javamail named MimePart DataSource used to send mail  
    permission java.util.PropertyPermission "java.home", "read";  
    permission java.util.PropertyPermission "java.naming.*", "read";  
    permission java.util.PropertyPermission "javax.sql.*", "read";  
  
    // OS Specific properties to allow read access  
    permission java.util.PropertyPermission "os.name", "read";  
    permission java.util.PropertyPermission "os.version", "read";  
    permission java.util.PropertyPermission "os.arch", "read";  
    permission java.util.PropertyPermission "file.separator", "read";  
    permission java.util.PropertyPermission "path.separator", "read";  
    permission java.util.PropertyPermission "line.separator", "read";  
  
    // JVM properties to allow read access  
    permission java.util.PropertyPermission "java.version", "read";  
    permission java.util.PropertyPermission "java.vendor", "read";  
    permission java.util.PropertyPermission "java.vendor.url", "read";  
    permission java.util.PropertyPermission "java.class.version", "read";  
    permission java.util.PropertyPermission "java.specification.version", "read";  
    permission java.util.PropertyPermission "java.specification.vendor", "read";  
    permission java.util.PropertyPermission "java.specification.name", "read";  
  
    permission java.util.PropertyPermission "java.vm.specification.version", "read";  
    permission java.util.PropertyPermission "java.vm.specification.vendor", "read";  
    permission java.util.PropertyPermission "java.vm.specification.name", "read";  
    permission java.util.PropertyPermission "java.vm.version", "read";  
    permission java.util.PropertyPermission "java.vm.vendor", "read";  
    permission java.util.PropertyPermission "java.vm.name", "read";
```

tomcat-users.xml

- This file contains the roles and security password for Tomcat.
- The following screenshot shows the different roles, users, and passwords for Tomcat 8:

```
<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>
```

server.xml

- This is the main configuration file for Tomcat and it mainly contains the Connector port configuration.
- The following screenshot shows the connector configuration, where Tomcat 8 runs on 8080 and has 20000 as the timeout setting:



```
-->
<Connector port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
           port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
-->
```

Enabling Tomcat Manager

- By default, the Tomcat Manager is disabled in Tomcat 8.
- It is a very powerful tool, but if it goes to the wrong hands, then it can create a problem for the system administrator or the application administrator.
- So it's very important that you enable Tomcat Manager with proper security.

How to enable the Tomcat Manager

```
<!-->
<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"/>
```

- Uncomment the user and save the file, followed by reloading Apache Tomcat 8, as shown in the following screenshot:

```
<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"/>
```

Securing Tomcat 8 for production

- There are different methods of securing Tomcat 8 and these come into picture based on the application's requirement and the security policy used by an IT organization.
- In Tomcat 8, there are different configurations, which need to be changed or enabled in order to secure Tomcat for the external environment.
- Let's discuss each configuration and their usage for a real-time environment.

Connector Port

- By default, Tomcat 8 runs on port 8080 using the HTTP protocol.
- As everyone knows the default port, it is easier for hackers to hit the port and trap the server.
- So it's always recommended to change the connector port and also the AJP port, which runs on 8009, to secure Tomcat.

```
<Connector executor="tomcatThreadPool" port="8080"
protocol="HTTP/1.1" connectionTimeout="20000"
redirectPort="8443" />
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

| Operating system | Location of the <code>services</code> file |
|------------------|--|
| Linux | /etc/services |
| Windows | C:\Windows\System32\drivers\etc |

```
# service-name port/protocol [aliases ...] [# comment]

tcpmux      1/tcp          # TCP port service multiplexer
tcpmux      1/udp          # TCP port service multiplexer
rje        5/tcp          # Remote Job Entry
rje        5/udp          # Remote Job Entry
echo        7/tcp
echo        7/udp
discard    9/tcp          sink null
discard    9/udp          sink null
sysstat    11/tcp         users
sysstat    11/udp         users
daytime    13/tcp
daytime    13/udp
qotd       17/tcp          quote
qotd       17/udp          quote
mspx       18/tcp          # message send protocol
mspx       18/udp          # message send protocol
chargen   19/tcp          ttystat source
chargen   19/udp          ttystat source
ftp-data   20/tcp
ftp-data   20/udp
# 21 is registered to ftp, but also used by fsp
ftp        21/tcp
ftp        21/udp          fsp fspd
ssh        22/tcp          # SSH Remote Login Protocol
ssh        22/udp          # SSH Remote Login Protocol
```

Slimming of Tomcat application

- Reduction in the JVM memory utilization
- Chances of any vulnerability will be less, as unwanted applications (libraries/JAR) are not available
- Easier maintenance of applications

```
[root@localhost webapps]# ls -ltrh
total 40K
drwxr-xr-x  3 root root 4.0K May 22 15:08 .
drwxr-xr-x 13 root root 4.0K May 22 15:08 ..
drwxr-xr-x  5 root root 4.0K May 22 15:08 commons-beanutils-1.8.3.jar
drwxr-xr-x  5 root root 4.0K May 22 15:08 commons-collections-3.2.2.jar
drwxr-xr-x  5 root root 4.0K May 22 15:08 commons-discovery-0.5.jar
```

Disable hot deployment

- Hot deployment or autodeployment is a process through which code will be deployed to the application automatically, without recycling the services.
- To disable the hot deployment, you have to edit server.xml for the following parameter:

```
<Host name="localhost" appBase="webapps"  
unpackWARs="true" autoDeploy="true">
```

- Change the autoDeploy to false.

```
<Host name="localhost" appBase="webapps"  
unpackWARs="true" autoDeploy="false">
```

Non-Tomcat settings

- In the previous section, Tomcat settings, we have discussed about the Tomcat-level configuration to implement security policies for Tomcat 8.
- In a real-time environment, with new and the latest technologies, these settings are not enough to deal with security threats.
- To make the system more secure, we have to secure our infrastructure.
- Let's discuss a few best practices for securing the web infrastructure.

Non-Tomcat settings

- In Windows, we can verify the firewall settings by using the netsh command in the following manner:

netsh firewall show state

- The previous command shows the current status of the firewall rules.

netsh firewall show config

```
C:\Users\user>netsh firewall show state
```

Firewall status:

| | |
|-----------------------------------|--------------------|
| Profile | = Standard |
| Operational mode | = Disable |
| Exception mode | = Enable |
| Multicast/broadcast response mode | = Enable |
| Notification mode | = Enable |
| Group policy version | = Windows Firewall |
| Remote admin mode | = Disable |

Ports currently open on all network interfaces:

| Port | Protocol | Version | Program |
|------|----------|---------|---------|
|------|----------|---------|---------|

No ports are currently open on all network interfaces.

```
C:\Users\user>netsh firewall show config
```

Domain profile configuration:

| | |
|-----------------------------------|-----------|
| Operational mode | = Disable |
| Exception mode | = Enable |
| Multicast/broadcast response mode | = Enable |
| Notification mode | = Enable |

Non-Tomcat settings

- In order to add or remove any firewall policies, we can run the following command:

netsh firewall set opmode enable

- The previous command allows us to edit the configuration rule.
- The following command adds the TCP port 8085 over the subnet. Hence, this port is accessible outside the system.

netsh firewall add portopening TCP 8085 HTTP enable subnet

- In Linux, we can verify the firewall settings using the iptables command in the following manner:

[root@localhost etc]# iptables -L

Non-Tomcat settings

- The following screenshot shows the firewall rule for a Linux environment. Currently, no firewall rule is defined:

```
[root@localhost etc]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

- In case we have to edit the firewall rule, then the following command needs to be executed:

**iptables -A INPUT -s 0/0 -i eth0 -d 192.168.1.2 -p TCP -j
ACCEPT**

Password

- We have to define the password encryption algorithm in the Realm section of server.xml, as in the following line of code:

```
<Realm className=
"org.apache.catalina.realm.MemoryRealm" digest="MD5" />
```

```
<Realm className="org.apache.catalina.realm.UserDatabaseRealm"
      resourceName="UserDatabase"/>
</Realm>
<Realm className="org.apache.catalina.realm.MemoryRealm"
      digest="MD5" />

<Host name="localhost" appBase="webapps"
      unpackWARs="true" autoDeploy="true">
```

Password

- Now go to tomcat_home/bin and run the following command, it will generate the encrypted algorithm, as shown in the following screenshot:

```
[root@localhost bin]# ./digest.sh -a MD5 secret
```

```
[root@localhost bin]# ./digest.sh -a MD5 secret  
secret:5ebe2294ecd0e0f08eab7690d2a6ee69
```

Password

- Copy the MD5 string and replace the password text from `tomcat_user.xml` with the following line of code:

```
<user name="admin"  
password="5ebe2294ecd0e0f08eab7690d2a6ee69 "  
roles="manager-gui" />
```

SSL configuration on Tomcat 8

- Secure Socket Layer (SSL) is another way of securing data communication.
- It is a cryptographic protocol, in which data travels through a secure channel.
- The server sends a secure key to the client browser, the client browser decrypts it and a handshake takes place between the server and the client or we can say it's a two-way handshake over the secure layer.
- When is SSL required for Tomcat?

Types of SSL certificates

Before we go ahead and install SSL, let's discuss the two types of SSL certificates, which are explained as follows:

- Self-signed certificate
- Signed certificate

Process of installing SSL

| CSR Attributes | Domain information as to which CSR needs to be generated |
|----------------|--|
| Common Name | Define the domain name |
| Organization | Organization name |
| Department | Department name of the organization |
| City | City where this organization is located |
| State | State where this organization is located |
| Country | Country where this organization is located |
| Key size | 2048 (encryption bit) |

Process of installing SSL

| CSR Attributes | Domain information as to which CSR needs to be generated |
|----------------|--|
| Common Name | tomcat8 |
| Organization | tomcat8 |
| Department | Tomcat |
| City | Hyd |
| State | AP |
| Country | IN |
| Key size | 2048 |

Process of installing SSL

- The following command will capture parameters for the CSR:

```
[root@localhost conf] # keytool -genkey -alias tomcat8 -keyalg RSA -keysize 2048 -keystore tomcat.jks
```

```
[root@localhost conf]# keytool -genkey -alias tomcat8 -keyalg RSA -keysize 2048 -keystore tomcat.jks
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: tomcat7@packtpub.com
What is the name of your organizational unit?
[Unknown]: tomcat7@packtpub.com
What is the name of your organization?
[Unknown]: tomcat
What is the name of your City or Locality?
[Unknown]: hyd
What is the name of your State or Province?
[Unknown]: AP
What is the two-letter country code for this unit?
[Unknown]: IN
Is CN=tomcat7@packtpub.com, O=tomcat7@packtpub.com, O=tomcat7, L=hyd, ST=AP, C=IN correct?
(yes): yes

Enter key password for <tomcat8>
      (RETURN if same as keystore password):
Re-enter new password:
```

Process of installing SSL

- Generate the certificate in CSR format using the following command.
- It will ask for the password and send it to the respective vendor for signed certificate creation.

```
[root@localhost conf]# keytool -certreq -alias tomcat8 -file csr.txt -keystore tomcat.jks
```

Enter keystore password:

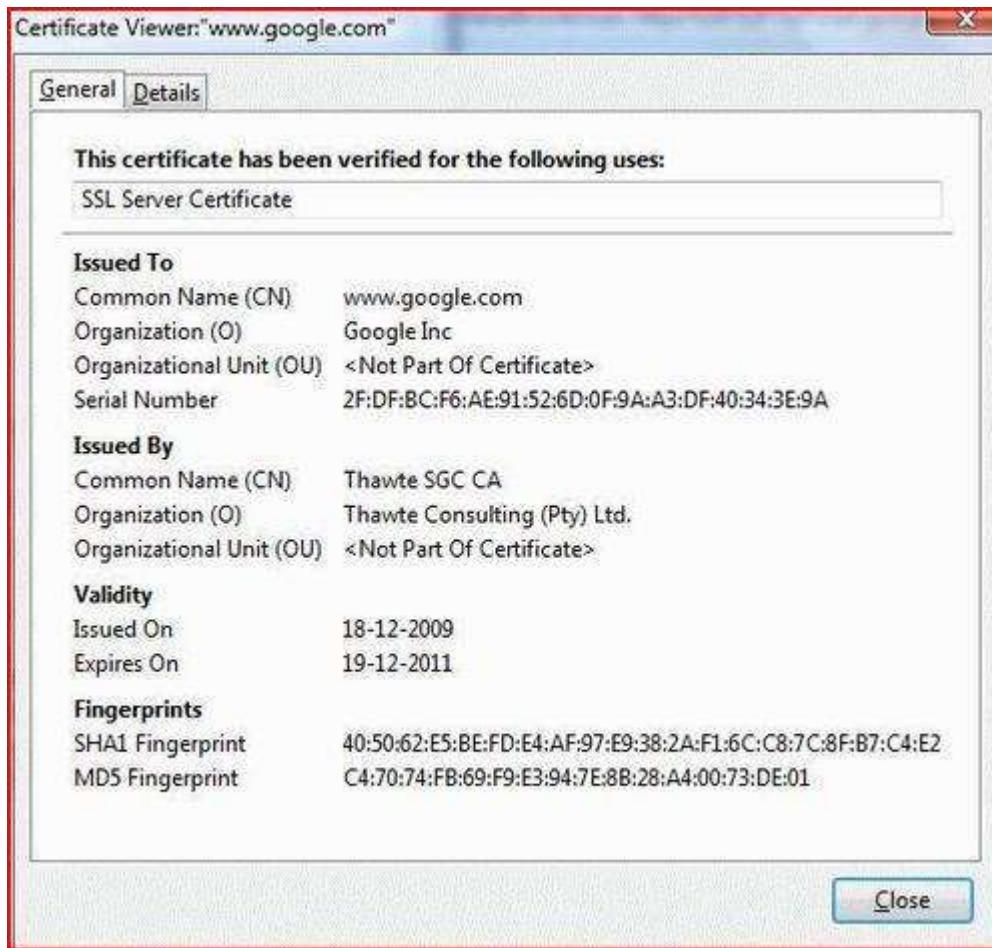
- Import the certificate to the following Tomcat key store.
Copy the tomcat8.jks in the TOMCAT_HOME/conf.

```
[root@localhost conf] # keytool -import -trustcacerts -alias tomcat8 -file tomcat8ernesto.com.pb7 tomcat8.jks
```

Process of installing SSL

- Open server.xml and change the settings, as in the following code snippet:

```
<Connector port="443" maxHttpHeaderSize="8192"  
maxThreads="150" minSpareThreads="25"  
maxSpareThreads="75" enableLookups="false"  
disableUploadTimeout="true" acceptCount="100"  
scheme="https" secure="true" SSLEnabled="true"  
clientAuth="false" sslProtocol="TLS" keyAlias="server"  
keystoreFile="tomcat8.jks" keypass="changeit" />
```



- If you click on Details, it shows that your certificate is successfully installed, as shown in the following screenshot:



Summary

- In this lesson, we have discussed the various policies of Tomcat 8 and their functionalities, such as the Catalina policy and System level policy.
- We have also discussed the different measures of enabling security and their benefits, such as SSL, best practices used in real-time industries to secure Tomcat 8 in the production environment by changing the configuration, and SSL implementation.



6. Logging in Tomcat 8



Logging in Tomcat 8

In this lesson, we will discuss:

- Logging services in Tomcat 8
- JULI
- Log4j
- Log level
- Valve component
- Analysis of logs

JULI

```
[root@localhost bin]# ls -ltrh
total 740K
-rwxr-xr-x 1 root root 1.6K Apr  1 2011 version.sh
-rw-r--r-- 1 root root 2.1K Apr  1 2011 version.bat
-rwxr-xr-x 1 root root 4.6K Apr  1 2011 tool-wrapper.sh
-rw-r--r-- 1 root root 3.6K Apr  1 2011 tool-wrapper.bat
-rw-r--r-- 1 root root 236K Apr  1 2011 tomcat-native.jar.gz
-rw-r--r-- 1 root root 34K Apr  1 2011 tomcat-juli.jar
-rw-r--r-- 1 root root 2.1K Apr  1 2011 startup.bat
-rwxr-xr-x 1 root root 1.6K Apr  1 2011 shutdown.sh
-rw-r--r-- 1 root root 2.1K Apr  1 2011 shutdown.bat
-rwxr-xr-x 1 root root 3.9K Apr  1 2011 setclasspath.sh
-rw-r--r-- 1 root root 3.3K Apr  1 2011 setclasspath.bat
-rwxr-xr-x 1 root root 1.6K Apr  1 2011 digest.sh
-rw-r--r-- 1 root root 2.1K Apr  1 2011 digest.bat
-rw-r--r-- 1 root root 1.4K Apr  1 2011 cpappend.bat
-rwxr-xr-x 1 root root 1.9K Apr  1 2011 configtest.sh
-rw-r--r-- 1 root root 195K Apr  1 2011 commons-daemon-native.jar.gz
-rw-r--r-- 1 root root 23K Apr  1 2011 commons-daemon.jar
-rw-r--r-- 1 root root 2.5K Apr  1 2011 catalina-tasks.xml
-rw-r--r-- 1 root root 12K Apr  1 2011 catalina.bat
-rw-r--r-- 1 root root 27K Apr  1 2011 bootstrap.jar
-rwxr-xr-x 1 root root 2.0K Jul 10 2011 startupbackup.sh
-rwxr-xr-x 1 root root 2.3K Jul 10 2011 startup.sh
-rwxr-xr-x 1 root root 19K Sep 25 10:33 catalina.sh
```

- **Loggers:** It can be defined as the logical name for the log file. This logical name is written in the application code. We can configure an independent logger for each application.
- **Appenders:** The process of generating logs is handled by appenders. There are many types of appenders, such as FileAppender, ConsoleAppender, SocketAppender, and so on, which are available in log4j. The following are some examples of appenders for log4j:

`log4j.appender.CATALINA=org.apache.log4j.DailyRollingFileAppender`

`log4j.appender.CATALINA.File=${catalina.base}/logs/catalina.out`

`log4j.appender.CATALINA.Append=true`

`log4j.appender.CATALINA.Encoding=UTF-8`

Loggers, appenders, and layouts

- The previous four lines of appender define the DailyRollingFileAppender in log4j, where the filename is catalina.out. These logs will have UTF-8 encoding enabled.

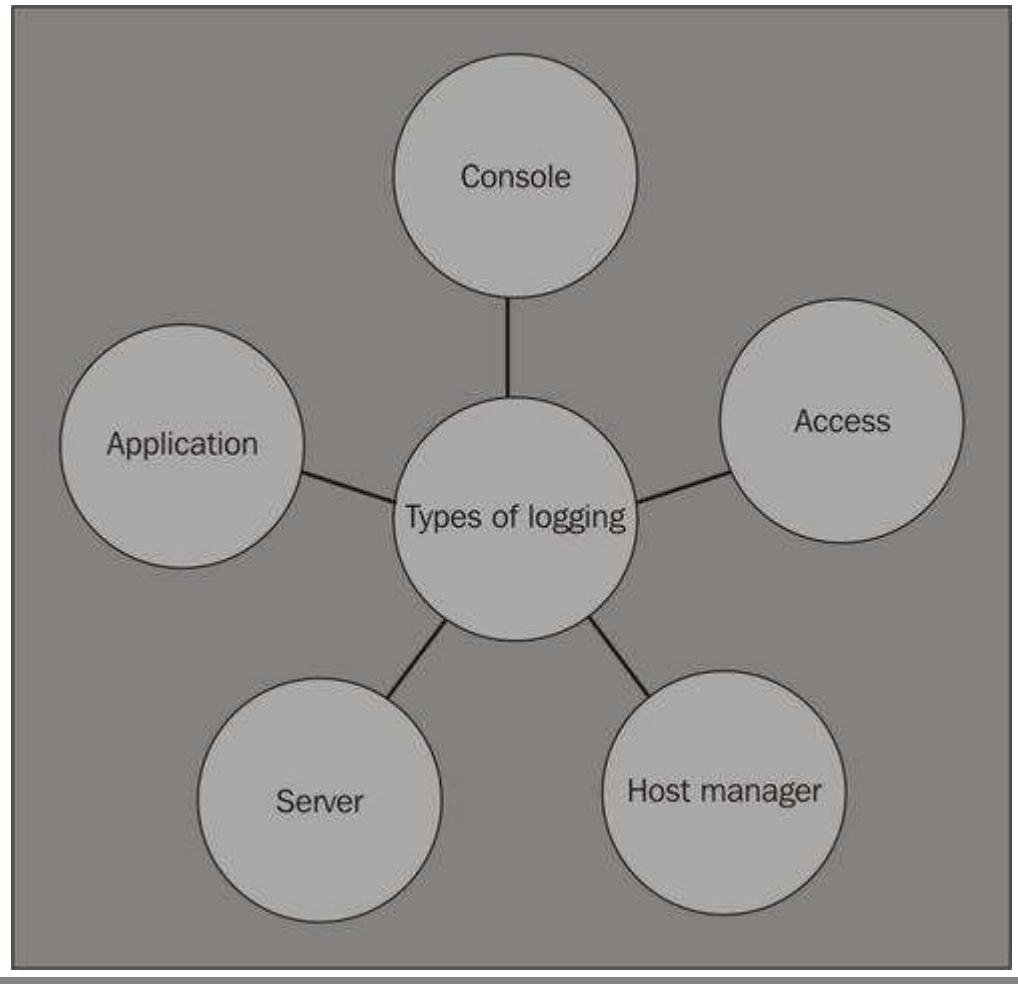
```
# Roll-over the log once per day
log4j.appender.CATALINA.DatePattern='.'dd-MM-yyyy'.log'
log4j.appender.CATALINA.layout =
org.apache.log4j.PatternLayout
log4j.appender.CATALINA.layout.ConversionPattern = %d
[%t] %-5p %c- %m%n
```

Loggers, appenders, and layouts

- Layout: It is defined as the format of logs displayed in the log file. The appender uses the layout to format the log files (also called patterns).
- The highlighted code shows the pattern for the access logs:

```
<Valve  
className="org.apache.catalina.valves.AccessLogValve  
" directory="logs" prefix="localhost_access_log."  
suffix=".txt" pattern="%h %l %u \"%r\" %s  
%b" resolveHosts="false"/>
```

Types of logging in Tomcat 8



Application log

- These logs are used to capture the application event while running the application transaction.
- These logs are very useful in order to identify the application level issues.
- For example, suppose your application performance is slow on a particular transition, then the details of that transition can only be traced in the application log.

Console log

```
CLASSPATH="$CLASSPATH""$CATALINA_HOME"/bin/bootstrap.jar

if [ -z "$CATALINA_BASE" ] ; then
    CATALINA_BASE="$CATALINA_HOME"
fi

if [ -z "$CATALINA_OUT" ] ; then
    CATALINA_OUT="$CATALINA_BASE"/logs/catalina.out
fi

if [ -z "$CATALINA_TMPDIR" ] ; then
    # Define the java.io.tmpdir to use for Catalina
    CATALINA_TMPDIR="$CATALINA_BASE"/temp
fi
```

Console log

```
[root@localhost logs]# ls -lth
total 172K
-rw-r--r-- 1 root root    0 May 16 21:03 manager.2011-05-16.log
-rw-r--r-- 1 root root    0 May 16 21:03 host-manager.2011-05-16.log
-rw-r--r-- 1 root root  714 May 16 21:19 localhost_access_log.2011-05-16.txt
-rw-r--r-- 1 root root  920 May 16 21:20 localhost.2011-05-16.log
-rw-r--r-- 1 root root 5.3K May 16 21:20 catalina.2011-05-16.log
-rw-r--r-- 1 root root    0 May 22 16:15 manager.2011-05-22.log
-rw-r--r-- 1 root root    0 May 22 16:15 host-manager.2011-05-22.log
-rw-r--r-- 1 root root    0 May 22 16:15 localhost_access_log.2011-05-22.txt
-rw-r--r-- 1 root root  460 May 22 16:19 localhost.2011-05-22.log
-rw-r--r-- 1 root root 2.9K May 22 16:19 catalina.2011-05-22.log
-rw-r--r-- 1 root root    0 Jun 23 02:25 manager.2011-06-23.log
-rw-r--r-- 1 root root    0 Jun 23 02:25 host-manager.2011-06-23.log
-rw-r--r-- 1 root root    0 Jun 23 02:26 localhost_access_log.2011-06-23.txt
-rw-r--r-- 1 root root  232 Jun 23 02:26 localhost.2011-06-23.log
-rw-r--r-- 1 root root 2.0K Jun 23 02:26 catalina.2011-06-23.log
-rw-r--r-- 1 root root    0 Jul 10 11:01 manager.2011-07-10.log
-rw-r--r-- 1 root root    0 Jul 10 11:01 host-manager.2011-07-10.log
-rw-r--r-- 1 root root    0 Jul 10 11:01 localhost_access_log.2011-07-10.txt
-rw-r--r-- 1 root root 1.6K Jul 10 15:22 localhost.2011-07-10.log
-rw-r--r-- 1 root root  30K Jul 10 15:22 catalina.out
-rw-r--r-- 1 root root  20K Jul 10 15:22 catalina.2011-07-10.log
[root@localhost logs]# pwd
/opt/apache-tomcat-7.0.12/logs
```

Console log

- The previous screenshot shows the Tomcat log file location, after the start of the Tomcat services.

```
May 22, 2011 4:15:47 PM org.apache.catalina.core.StandardEngine startInternal
INFO: Starting Servlet Engine: Apache Tomcat/7.0.12
May 22, 2011 4:15:47 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory examples
May 22, 2011 4:15:48 PM org.apache.catalina.util.SessionIdGenerator createSecureRandom
INFO: Creation of SecureRandom instance for session ID generation using [SHA1PRNG] took [186] milliseconds.
May 22, 2011 4:15:48 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory host-manager
May 22, 2011 4:15:49 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory docs
May 22, 2011 4:15:49 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory ROOT
May 22, 2011 4:15:49 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory manager
May 22, 2011 4:15:49 PM org.apache.coyote.AbstractProtocolHandler start
INFO: Starting ProtocolHandler ["http-bio-8080"]
May 22, 2011 4:15:49 PM org.apache.coyote.AbstractProtocolHandler start
INFO: Starting ProtocolHandler ["ajp-bio-8009"]
May 22, 2011 4:15:49 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 1903 ms
```

Access log

Access logs are customized logs, which give information about the following:

- Who has accessed the application
- What components of the application are accessed
- Source IP and so on

Access log

- The following screenshot shows the definition of the access logs.
- You can customize them according to the environment and your auditing requirements.

```
<!-- SingleSignOn valve, share authentication between web applications
     Documentation at: /docs/config/valve.html -->
<!--
<Valve className="org.apache.catalina.authenticator.SingleSignOn" />
-->

<!-- Access log processes all example.
     Documentation at: /docs/config/valve.html
     Note: The pattern used is equivalent to using pattern="common" -->
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
      prefix="localhost_access_log." suffix=".txt"
      pattern="%t %l %u %q %b" resolveHosts="false"/>
```

Access log

- Let's discuss the pattern format of the access logs and understand how we can customize the logging format:

```
<Valve  
className="org.apache.catalina.valves.AccessLogValve  
" directory="logs" prefix="localhost_access_log."  
suffix=".txt" pattern="%h %l %u \"%r\" %s  
%b" resolveHosts="false"/>
```

Access log

```
[root@localhost logs]# cat localhost_access_log.2012-01-24.txt
127.0.0.1 - - [24/Jan/2012:09:53:21 -0800] "GET / HTTP/1.1" 200 12079
127.0.0.1 - - [24/Jan/2012:09:53:22 -0800] "GET /tomcat.css HTTP/1.1" 304 -
127.0.0.1 - - [24/Jan/2012:09:53:22 -0800] "GET /favicon.ico HTTP/1.1" 304 -
127.0.0.1 - - [24/Jan/2012:09:53:23 -0800] "GET /asf-logo.png HTTP/1.1" 304 -
127.0.0.1 - - [24/Jan/2012:09:53:23 -0800] "GET /tomcat.png HTTP/1.1" 304 -
127.0.0.1 - - [24/Jan/2012:09:53:23 -0800] "GET /bg-nav.png HTTP/1.1" 304 -
127.0.0.1 - - [24/Jan/2012:09:53:23 -0800] "GET /bg-upper.png HTTP/1.1" 304 -
```

Host manager

```
localhost.org.apache.juli.FileHandler.level = FINE
localhost.org.apache.juli.FileHandler.directory = ${catalina.base}/logs
localhost.org.apache.juli.FileHandler.prefix = localhost.

manager.org.apache.juli.FileHandler.level = FINE
manager.org.apache.juli.FileHandler.directory = ${catalina.base}/logs
manager.org.apache.juli.FileHandler.prefix = manager.

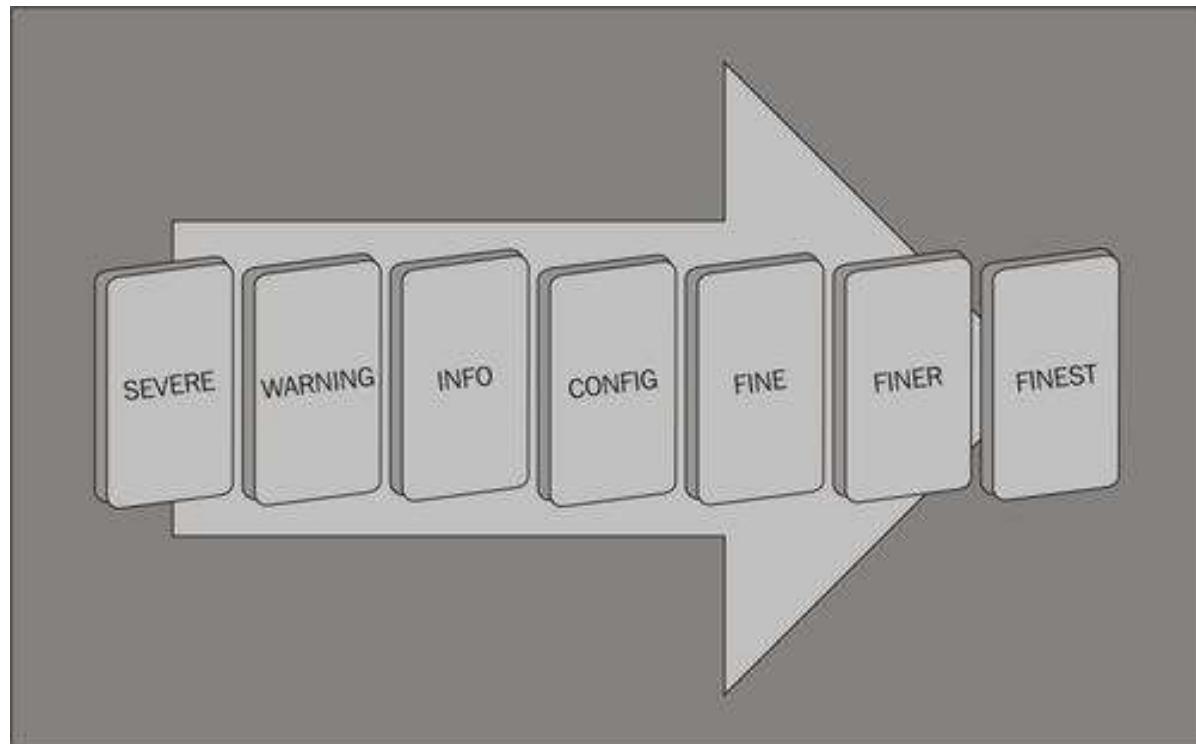
host-manager.org.apache.juli.FileHandler.level = FINE
host-manager.org.apache.juli.FileHandler.directory = ${catalina.base}/logs
host-manager.org.apache.juli.FileHandler.prefix = host-manager.

java.util.logging.ConsoleHandler.level = FINE
java.util.logging.ConsoleHandler.formatter = java.util.logging.SimpleFormatter
```

- The log file for manager looks similar to the following:

```
28 Jun, 2011 3:36:23 AM org.apache.catalina.core.ApplicationContext log
INFO: HTMLManager: list: Listing contexts for virtual host 'localhost'
28 Jun, 2011 3:37:13 AM org.apache.catalina.core.ApplicationContext log
INFO: HTMLManager: list: Listing contexts for virtual host 'localhost'
28 Jun, 2011 3:37:42 AM org.apache.catalina.core.ApplicationContext log
INFO: HTMLManager: undeploy: Undeploying web application at '/sample'
28 Jun, 2011 3:37:43 AM org.apache.catalina.core.ApplicationContext log
INFO: HTMLManager: list: Listing contexts for virtual host 'localhost'
28 Jun, 2011 3:42:59 AM org.apache.catalina.core.ApplicationContext log
INFO: HTMLManager: list: Listing contexts for virtual host 'localhost'
28 Jun, 2011 3:43:01 AM org.apache.catalina.core.ApplicationContext log
INFO: HTMLManager: list: Listing contexts for virtual host 'localhost'
28 Jun, 2011 3:53:44 AM org.apache.catalina.core.ApplicationContext log
INFO: HTMLManager: list: Listing contexts for virtual host 'localhost'
```

Types of log levels in Tomcat 8



Types of log levels in Tomcat 8

| Log level | Description |
|-----------------|--|
| SEVERE(highest) | Captures exception and Error |
| WARNING | Warning messages |
| INFO | Informational message, related to the server activity |
| CONFIG | Configuration message |
| FINE | Detailed activity of the server transaction (similar to debug) |
| FINER | More detailed logs than FINE |
| FINEST(least) | Entire flow of events (similar to trace) |

Types of log levels in Tomcat 8

- For example, let's take an appender from logging.properties and find out the log level used; the first log appender for localhost is using FINE as the log level, as shown in the following code snippet:

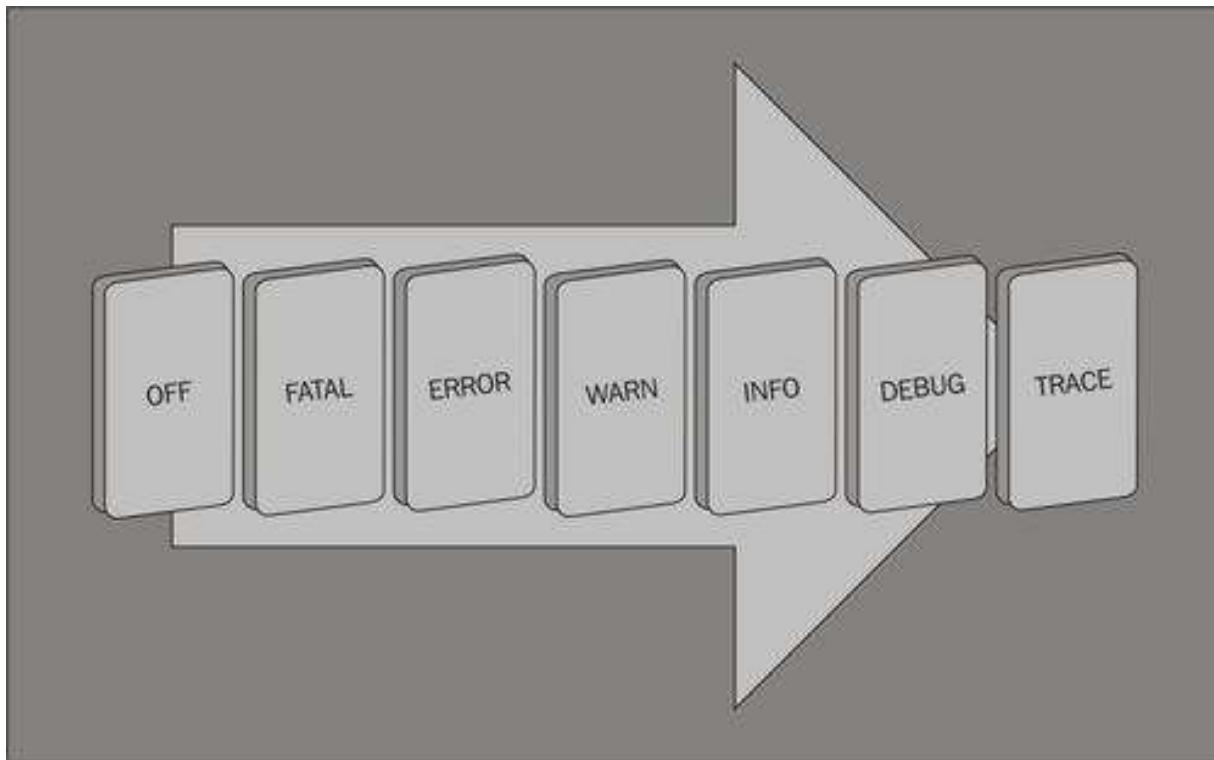
```
localhost.org.apache.juli.FileHandler.level = FINE  
localhost.org.apache.juli.FileHandler.directory =  
${catalina.base}/logs  
localhost.org.apache.juli.FileHandler.prefix = localhost.
```

Types of log levels in Tomcat 8

- The following code shows the default file handler configuration for logging in Tomcat 8 using JULI.
- The properties and log levels are mentioned:

Refer to the file 6_1.txt

Log4j



Log level for log4j

| Log level | Description |
|-----------|--|
| OFF | This level is set when you want logging to be set as false (Stopped logging). |
| FATAL | This log level will print the severe errors that cause premature termination. |
| ERROR | This log level is used to capture runtime errors or unexpected conditions. Expect these to be immediately visible on a status console. |
| WARN | This level is used in the previous version. It gives you almost errors, other runtime situations that are undesirable or unexpected, but not necessarily wrong . Expect these to be immediately visible on a status console. |
| INFO | This log level will define the interesting runtime events (startup/shutdown). It is best practice to put the logs at the INFO level. |
| DEBUG | Detailed information on the flow through the system is defined in this level. |
| TRACE | This log level captures all the events in the system and application. |

How to use log4j

```
log4j.rootLogger=INFO, CATALINA

# Define all the appenders
log4j.appendender.CATALINA=org.apache.log4j.DailyRollingFileAppender
log4j.appendender.CATALINA.File=${catalina.base}/logs/catalina.
log4j.appendender.CATALINA.Append=true
log4j.appendender.CATALINA.Encoding=UTF-8
# Roll-over the log once per day
log4j.appendender.CATALINA.DatePattern='.'dd-MMM-yyyy'.log'
log4j.appendender.CATALINA.layout = org.apache.log4j.PatternLayout
log4j.appendender.CATALINA.layout.ConversionPattern = %d [%t] %-5p %c- %m%n

log4j.appendender.LOCALHOST =org.apache.log4j.DailyRollingFileAppender
log4j.appendender.LOCALHOST.File=${catalina.base}/logs/localhost.
log4j.appendender.LOCALHOST.Append=true
log4j.appendender.LOCALHOST.Encoding=UTF-8
log4j.appendender.LOCALHOST.DatePattern='.'yyyy-MM-dd'.log'
log4j.appendender.LOCALHOST.layout = org.apache.log4j.PatternLayout
log4j.appendender.LOCALHOST.layout.ConversionPattern = %d [%t] %-5p %c- %m%n
```

Log level mapping

| Log level in JULI | Log level in log4j |
|-------------------|--------------------|
| SEVERE | FATAL, ERROR |
| WARNING | WARN |
| INFO | INFO |
| CONFIG | NA |
| FINE | DEBUG |
| FINER | DEBUG |
| FINEST | TRACE |

Values for Tomcat 8

- By default, access logs for Tomcat are defined as follows:

```
<Valve  
className="org.apache.catalina.valves.AccessLogValve  
" directory="logs" prefix="localhost_access_log."  
suffix=".txt" pattern="%h %l %u \"%r\" %s  
%b" resolveHosts="false"/>
```

Values for Tomcat 8

- We want to change the log pattern to show the time taken to process the request.
- We have to add the %T in the patterns. The changed code is shown as follows:

```
<Valve  
className="org.apache.catalina.valves.AccessLogValve  
" directory="logs"  
prefix="localhost_access_log." suffix=".txt"  
pattern="%h %l %u %t %T "%r" %s %b"  
resolveHosts="false"/>
```

| Values | Description |
|--------|---|
| %a | Remote IP address |
| %A | Local IP address |
| %b | Bytes sent, excluding HTTP headers, or " if zero |
| %B | Bytes sent, excluding HTTP headers |
| %h | Remote hostname (or IP address if enableLookups for the connector is false) |
| %H | Request protocol |
| %l | Remote logical username from identd |
| %m | Request method (GET, POST, and so on) |
| %p | Local port on which this request was received |
| %q | Query string (prepended with a '?' if it exists) |
| %r | First line of the request (method and request URI) |
| %s | HTTP status code of the response |
| %S | User session ID |
| %t | Date and time, in Common Log format |
| %u | Remote user that was authenticated (if any) |
| %U | Requested URL path |
| %v | Local server name |
| %D | Time taken to process the request, in milliseconds |
| %T | Time taken to process the request, in seconds |
| %I | Current request thread name (can compare later with stack traces) |

Log analysis

- Check the logs of the last 1 hour from the issue
- Always go to the first exception in the logs when the error has started
- Always keep in mind that issues are not caused due to malfunction of Tomcat, also check the other infrastructure resources

Log analysis

In non-DOS operating systems (Linux, Unix, Ubuntu, and so on), there are two utilities which are very useful in log analysis, grep and awk.

Let's discuss grep and awk utilities briefly:

- grep: This utility prints the lines which match the string searched.

grep Error catalina.out

Log analysis

- awk: This command is used for pattern scanning. Suppose we want to print 10 columns in the entire data file, then this command is very useful.
- The following screenshot shows the output of the command when run for the /opt directory:

```
find "location of directory" -type f -size +10000k -exec ls -lh {} \; | awk '{  
print $9 ":" " $5 }'
```

```
find "/opt" -type f -size +10000k -exec ls -lh {} \; | awk '{ print $9 ":" " $5 }'
```

```
[root@localhost conf]# find /opt -type f -size +10000k -exec ls -lh {} \; | awk '{ print $9 ":" " $5 }'  
/opt/httpd-2.2.19.tar: 36M  
/opt/jdk1.6.0_24/src.zip: 19M  
/opt/jdk1.6.0_24/lib/tools.jar: 13M  
/opt/jdk1.6.0_24/lib/ct.sym: 15M  
/opt/jdk1.6.0_24/jre/lib/rt.jar: 50M  
/opt/jdk1.6.0_24/jre/lib/i386/client/classes.jar: 15M  
/opt/jdk-6u24-linux-i586.bin: 81M
```

Helpful commands for log analysis

- Finding large files and directories in Linux:

```
find "location of directory" type f -size +10000k -exec ls -lh {} \; | awk '{ print $9 ":" "$5" }'
```

- Finding directories with a size over 100MB:

```
find / -type d -size +100000k
```

- Sort directories as per size using du:

```
du --max-depth=1 -m | sort -n -r
```

Helpful commands for log analysis

- Finding directory sizes:

`du -sh folder_name`

`du -ch folder_name`

`du -csh folder_name`

- The following command is used for truncating huge log files on the live system (log rotation can be done without recycle of services):

`cat /dev/null > file_name`

Helpful commands for log analysis

- Finding ERROR exception

`grep ERROR log_file`

- Last 200 lines in log file:

`tail -200 log_file`

- Current logs to be updated

`tail -f log_file`

Summary

- In this lesson, we have discussed the different methods of enabling logs in Tomcat 8 using log4j and JULI.
- Also, we have discussed the best practices used for log analysis, tips, and tricks.





7. Troubleshooting in Tomcat



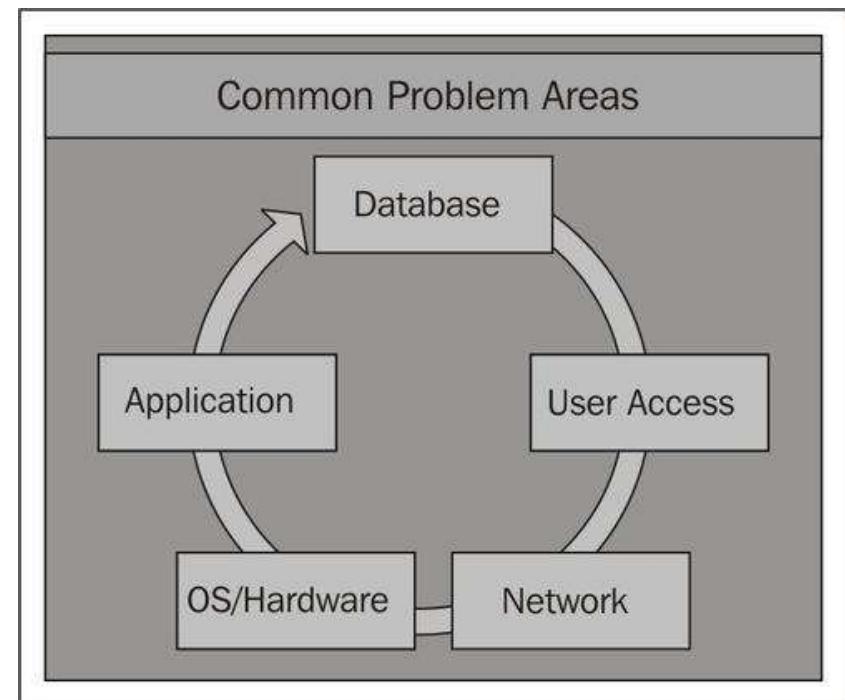
Troubleshooting in Tomcat

In this lesson, we will discuss:

- Common issues
- Third-party tools for thread dump analysis
- Tomcat specific issues related to the OS, JVM, and database
- How to troubleshoot a problem
- Best practices for the production environment

Common problem areas for web administrators

- Web administrators always find issues with applications, not due to server failure of the Tomcat server, but because applications start malfunctioning due to other components as well.
- The following figure shows different components for a typical middleware environment:

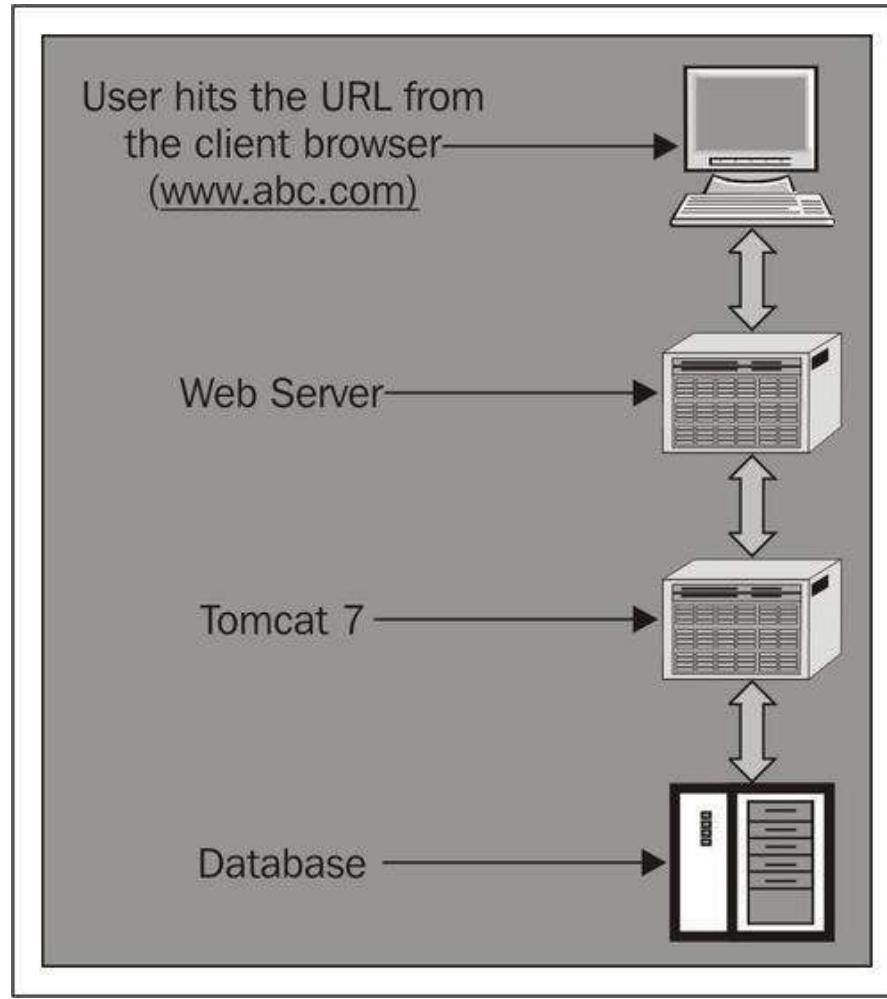


How to troubleshoot a problem

- We cannot troubleshoot any issue by just referring to the user comments or problem statement.
- In order to troubleshoot the issue, we have to narrow down the problem to its root level and fix the issue.
- Many web administrators always ask this question; how do we know that the system has a particular problem?

Slowness issue in applications

- Let's take a real-time situation where users complain about the performance of the application.
- The application comprises of an enterprise setup, which is a combination of the Apache HTTP server as a frontend, Tomcat 8 is used as a servlet container, and the Oracle database running as a backend database server.



User end troubleshooting

Perform the following steps to troubleshoot:

- Try to access the application from the user's browser and check how much time it takes to load the application page.
- Check the ping response of the server from the user side, for example, abc.com, using the command ping.
- If you get an appropriate response, it means the connectivity for the application server and user machine is working fine.

ping abc.com

User end troubleshooting

```
C:\Users\user>ping abc.com

Pinging abc.com [199.181.132.250] with 32 bytes of data:
Reply from 199.181.132.250: bytes=32 time=349ms TTL=232
Reply from 199.181.132.250: bytes=32 time=289ms TTL=230
Reply from 199.181.132.250: bytes=32 time=296ms TTL=232
Reply from 199.181.132.250: bytes=32 time=294ms TTL=230

Ping statistics for 199.181.132.250:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 289ms, Maximum = 349ms, Average = 307ms
```

Web server troubleshooting

- Check whether the web server process is running or not.
- If it is running, check how many processes are running by using the following command.
- This command will show the number of processes and their status.

```
ps -aef |grep httpd
```

Web server troubleshooting

- Then, check the CPU utilization and memory status of the system to see if any Apache processes are consuming a high CPU usage by using the following command:

top|head

```
[root@localhost ~]# top|head
top - 09:01:39 up 1:42, 3 users,  load average: 1.99, 2.09, 1.74
Tasks: 117 total,  3 running, 113 sleeping,  0 stopped,  1 zombie
Cpu(s): 1.1%us, 19.9%sy, 2.5%ni, 73.5%id, 2.5%wa, 0.1%hi, 0.5%si, 0.0%st
Mem: 1571836k total, 604168k used, 967668k free, 65108k buffers
Swap: 2040212k total,      0k used, 2040212k free, 388400k cached

          PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+   COMMAND
        6765 root      39  19  4664 1400 1016 S  3.5  0.1  1:03.79 makewhatis
        27389 root      15   0  2156 1004  740 R  1.8  0.1  0:00.04 top
          1 root      15   0  2032  676  576 S  0.0  0.0  0:01.90 init
[root@localhost ~]#
```

Web server troubleshooting

- The next step is to check the Apache logs and search for errors in the error and access logs.
- The following screenshot shows the system has started successfully:

```
[Tue Jul 16 02:48:01 2011] [notice] Apache/2.2.19 (Win32) configured -- resuming normal operations
[Tue Jul 16 02:48:01 2011] [notice] Server built: May 20 2011 17:39:36
[Tue Jul 16 02:48:01 2011] [notice] Parent: Created child process 2860
[httpd.exe]: Could not reliably determine the server's fully qualified domain name, using 10.0.0.3 for ServerName
[httpd.exe]: Could not reliably determine the server's fully qualified domain name, using 10.0.0.3 for ServerName
[Tue Jul 16 02:48:01 2011] [notice] Child 2860: Child process is running
[Tue Jul 16 02:48:01 2011] [notice] Child 2860: Acquired the start mutex
[Tue Jul 16 02:48:01 2011] [notice] Child 2860: Starting 44 worker threads.
[Tue Jul 16 02:48:01 2011] [notice] Child 2860: Starting thread to listen on port 80.
```

- httpd.exe: Could not reliably determine the server's fully qualified domain name, using 10.0.0.3 for ServerName.

Web server troubleshooting

- Also, there are two commands which are useful for searching the error in the logs. They are as follows:

`tail -f log file |grep ERROR`

- The previous command is used when you want to search the error in the logs.

`grep " 500 " access_log`

The previous command is used to search error codes in the logs.

Web server troubleshooting

- Use the df command to check the mount space, where df = disk free and switch -h = human readable.
- The syntax to use the df command is as follows and the output is shown in the following screenshot:

df -h

```
[root@localhost opt]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2        3.8G  2.4G  1.3G  66% /
/dev/sda1        46M   9.2M   35M  22% /boot
tmpfs           768M    0    768M  0% /dev/shm
/dev/sda3       14G   778M   13G   6% /home
[root@localhost opt]#
```

Tomcat 8 troubleshooting

- Check the Java processes for Tomcat and the load average for the instance machine:

ps -ef |grep java

```
[root@localhost bin]# ps -ef |grep java
root      19488  -1  0 20186 pts/2    00:00:04 /opt/tomcat-7.0.52/conf/bin/startup.sh
root      19488  -1  0 20186 pts/2    00:00:04 /opt/tomcat-7.0.52/bin/java -Djava.awt.headless=true -Dlogging.config=conf/logging.properties -Dtomcat.home=/opt/tomcat-7.0.52 -Dtomcat.base=/opt/tomcat-7.0.52 -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.endorsed.dirs=/opt/tomcat-7.0.52/lib/endorsed -Djava.ext.dirs=/opt/tomcat-7.0.52/lib/ext -Djava.class.path=/opt/tomcat-7.0.52/resources -Djava.io.tmpdir=/opt/tomcat-7.0.52/temp -Djava.library.path=/opt/tomcat-7.0.52/bin -Djava.security.policy=/opt/tomcat-7.0.52/conf/policy.jar
root      19501 19488  -1  0 20186 pts/2    00:00:00 grep java
[root@localhost bin]#
```

Tomcat 8 troubleshooting

- The following screenshot shows the output of the head command on the Tomcat server:

top|head

```
[root@localhost bin]# top|head
top - 10:54:56 up 3:35, 3 users, load average: 0.00, 0.02, 0.01
Tasks: 111 total, 1 running, 109 sleeping, 0 stopped, 1 zombie
Cpu(s): 0.6%us, 12.2%sy, 1.3%ni, 84.1%id, 1.3%wa, 0.1%hi, 0.4%si, 0.0%st
Mem: 1571836k total, 683860k used, 887976k free, 86100k buffers
Swap: 2040212k total, 0k used, 2040212k free, 426956k cached

 PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
  1 root      15   0  2032   676  576 S  0.0  0.0  0:01.95 init
  2 root      RT   0      0      0  0 S  0.0  0.0  0:00.00 migration/0
  3 root      39  19      0      0  0 S  0.0  0.0  0:00.00 ksoftirqd/0
```

- Then check the Tomcat logs which can be found in **TOMCAT_HOME/logs**, and search for the exception in the log files, mainly in **catalina.out**, **localhost.yyyy-mm-dd.log** using the following command:

grep INFO catalina.out

```
Sep 12, 2011 10:45:47 AM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory examples
Sep 12, 2011 10:45:49 AM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory host-manager
Sep 12, 2011 10:45:49 AM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory docs
Sep 12, 2011 10:45:49 AM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory ROOT
Sep 12, 2011 10:45:49 AM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory manager
Sep 12, 2011 10:45:49 AM org.apache.coyote.AbstractProtocolHandler start
INFO: Starting ProtocolHandler ("http-bio-8080")
Sep 12, 2011 10:45:49 AM org.apache.coyote.AbstractProtocolHandler start
INFO: Starting ProtocolHandler ("ajp-bio-8009")
Sep 12, 2011 10:45:49 AM org.apache.catalina.startup.Catalina start
INFO: Server startup in 3133 ms
```

Tomcat 8 troubleshooting

- The previous screenshot shows the Tomcat startup in the logs.
- If there are any errors in the logs, they can be checked using the following command:

```
grep ERROR catalina.out
```

JVM analysis in the Tomcat instance

- There are some chances where the JVM is over utilized in the application.
- To view the memory allocation for the JVM instance, you can use the command-line utility, jmap.
- This command comes with JDK 1.6. It's a Java utility, which determines the entire memory allocation of the Tomcat instance.

```
[root@localhost logs]# jmap -heap "TOMCAT INSTANCE  
PID "
```

Tomcat 8 troubleshooting

- Let us discuss how the previous command performs.
- The jmap command internally collects the JVM memory details, -heap is the switch that tells jmap to collect and display the heap memory footprint, TOMCAT INSTANCE PID is the process ID of the Tomcat instance for which process jmap has to fetch the memory details.

```
[root@localhost logs]# jmap -heap 10638
```

Tomcat 8 troubleshooting

- The jmap command is present in JAVA_HOME/bin and if you set the JAVA_HOME/bin in the path, then you can execute the command from anywhere.

```
Mark Sweep Compact GC

Heap Configuration:
  MinHeapFreeRatio = 40
  MaxHeapFreeRatio = 70
  MaxHeapSize      = 134217728 (128.0MB)
  NewSize          = 1048576 (1.0MB)
  MaxNewSize        = 4294901760 (4095.9375MB)
  OldSize          = 4194304 (4.0MB)
  NewRatio         = 2
  SurvivorRatio    = 8
  PermSize         = 16777216 (16.0MB)
  MaxPermSize      = 268435456 (256.0MB)

Heap Usage:
New Generation (Eden + 1 Survivor Space):
  capacity = 40239104 (38.375MB)
  used     = 22928384 (21.8662109375MB)
  free     = 17310720 (16.5087890625MB)
  56.98035423452769% used
Eden Space:
  capacity = 35782656 (34.125MB)
  used     = 22928384 (21.8662109375MB)
  free     = 12854272 (12.2587890625MB)
  64.07680860805861% used
From Space:
  capacity = 4456448 (4.25MB)
  used     = 0 (0.0MB)
  free     = 4456448 (4.25MB)
  0.0% used
To Space:
  capacity = 4456448 (4.25MB)
  used     = 0 (0.0MB)
  free     = 4456448 (4.25MB)
  0.0% used
tenured generation:
  capacity = 89522176 (85.375MB)
  used     = 1666064 (1.5888824462890625MB)
  free     = 87856112 (83.78611755371094MB)
  1.8610628946284773% used
Perm Generation:
  capacity = 16777216 (16.0MB)
  used     = 11317504 (10.793212890625MB)
  free     = 5459712 (5.206787109375MB)
  67.45758056640625% used
```

How to obtain a thread dump in Tomcat 8

Thread dump using Kill command

- This command generates and redirects the thread dump in catalina.out log.
- But, the limitation to this command is it works in a non-DOS environment such as Linux, Unix, and so on.

Kill -3 java process id

- For example:

Kill -3 10638

```

Full thread dump Java HotSpot(TM) Client VM (19.1-b02 mixed mode, sharing):

"ajp-bio-8009"-AsyncTimeout" daemon prio=10 tid=0x0919b800 nid=0x29cf waiting on condition [0xb4916000]
  java.lang.Thread.State: TIMED_WAITING (sleeping)
    at java.lang.Thread.sleep(Native Method)
    at org.apache.tomcat.util.net.JIoEndpoint$AsyncTimeout.run(JIoEndpoint.java:149)
    at java.lang.Thread.run(Thread.java:662)

"ajp-bio-8009"-Acceptor-0" daemon prio=10 tid=0x0919a400 nid=0xa29ce runnable [0xb4947000]
  java.lang.Thread.State: RUNNABLE
    at java.net.PlainSocketImpl.socketAccept(Native Method)
    at java.net.PlainSocketImpl.accept(PlainSocketImpl.java:109)
    - locked <0xe16cd80> (a java.net.SockeSocketImpl)
    at java.net.ServerSocket.implAccept(ServerSocket.java:462)
    at java.net.ServerSocket.accept(ServerSocket.java:430)
    at org.apache.tomcat.util.net.DefaultServerSocketFactory.acceptSocket(DefaultServerSocketFactory.java:159)
    at org.apache.tomcat.util.net.JIoEndpoint$Acceptor.run(JIoEndpoint.java:211)
    at java.lang.Thread.run(Thread.java:662)

"http-bio-8080"-AsyncTimeout" daemon prio=10 tid=0x09196000 nid=0x29cd waiting on condition [0xb493a000]
  java.lang.Thread.State: TIMED_WAITING (sleeping)
    at java.lang.Thread.sleep(Native Method)
    at org.apache.tomcat.util.net.JIoEndpoint$AsyncTimeout.run(JIoEndpoint.java:149)
    at java.lang.Thread.run(Thread.java:662)

"http-bio-8080"-Acceptor-0" daemon prio=10 tid=0x09197c00 nid=0x29cc runnable [0xb494b000]
  java.lang.Thread.State: RUNNABLE
    at java.net.PlainSocketImpl.socketAccept(Native Method)
    at java.net.PlainSocketImpl.accept(PlainSocketImpl.java:109)
    - locked <0xe16cdcc> (a java.net.SockeSocketImpl)
    at java.net.ServerSocket.implAccept(ServerSocket.java:462)
    at java.net.ServerSocket.accept(ServerSocket.java:430)
    at org.apache.tomcat.util.net.DefaultServerSocketFactory.acceptSocket(DefaultServerSocketFactory.java:159)
    at org.apache.tomcat.util.net.JIoEndpoint$Acceptor.run(JIoEndpoint.java:211)
    at java.lang.Thread.run(Thread.java:662)

"ContainerBackgroundProcessor[StandardEngine[Catalina]]" daemon prio=10 tid=0x09196400 nid=0x29cb waiting on condition [0xb4999000]
  java.lang.Thread.State: TIMED_WAITING (sleeping)
    at java.lang.Thread.sleep(Native Method)
    at org.apache.catalina.core.ContainerBase$ContainerBackgroundProcessor.run(ContainerBase.java:3369)
    at java.lang.Thread.run(Thread.java:662)

```

How to obtain a thread dump in Tomcat 8

```
Heap
def new generation   total 39424K, used 11208K [0x63550000, 0x66010000, 0x6eff0000]
  eden space 35072K,  31% used [0x63550000, 0x640423f0, 0x65790000]
  from space 4352K,  0% used [0x65bd0000, 0x65bd0000, 0x66010000]
  to   space 4352K,  0% used [0x65790000, 0x65790000, 0x65bd0000]
tenured generation
  total 87424K, used 3212K [0x6dfe0000, 0x73550000, 0x83550000]
    the space 87424K,  3% used [0x6dff0000, 0x6e3133f0, 0x6e313400, 0x73550000]
compacting perm gen
  total 12288K, used 6484K [0x83550000, 0x84150000, 0x93550000]
    the space 12288K,  52% used [0x83550000, 0x83ba5298, 0x83ba5400, 0x84150000]
  to space 10240K,  61% used [0x93550000, 0x93b78a38, 0x93b78c00, 0x93f50000]
  zw space 12288K,  60% used [0x93f50000, 0x94688ec0, 0x94689000, 0x94b50000]
```

Thread dump using jstack

- There are some switches which are commonly used with the jstack command, as mentioned in the following table:

| Options | Description |
|---------|---|
| -f | Generates a Java stack forcefully. Majorly used when the process is in the hang state |
| -l | Long listing (displays the additional information on locks) |
| -m | Mixed mode Java stack generation |

Thread dump using jstack

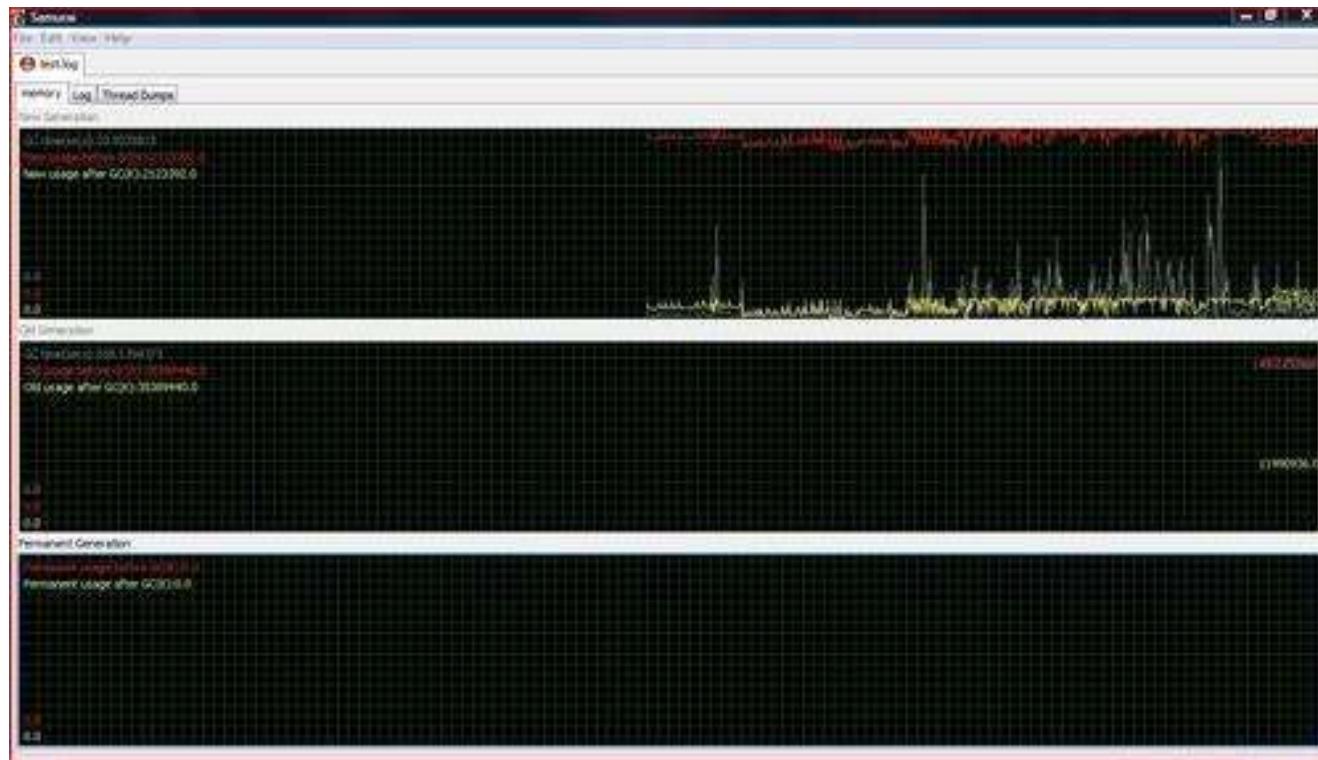
- The following command syntax generates a Java stack for a Java process and redirects the output in a text file:

```
jstack -f Pid > threaddump.txt
```

- For example:

```
jstack -f 10638 > threaddump.txt
```

Thread dump analysis using Samurai



Thread dump analysis using Samurai

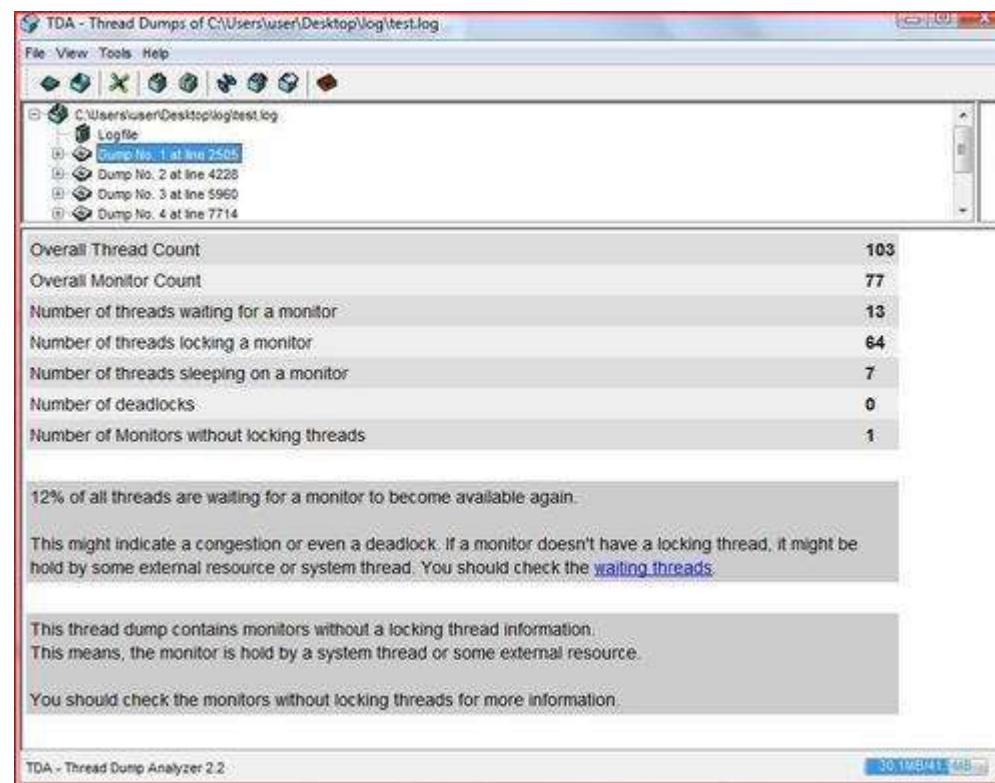


Thread dump analysis using the Thread Dump Analyzer

There is another very powerful tool, and very commonly used by the web administrator, known as Thread Dump Analyzer. This tool is capable of generating a summary for the thread dump. Following are the advantages of using this analyzer:

- Long running threads can be compared among multiple thread dumps
- It visualizes each thread separately
- It generates a summary for each thread dump

Thread dump analysis using the Thread Dump Analyzer



TDA - Thread Dumps of C:\Users\user\Desktop\log\test.log

File View Tools Help

Dump No. 1 at line 2505
Dump No. 2 at line 4233
Dump No. 3 at line 5900
Dump No. 4 at line 7714
Dump No. 5 at line 9530
Dump No. 6 at line 11236

Overall Thread Count: 103
Overall Monitor Count: 70
Number of threads waiting for a monitor: 19
Number of threads locking a monitor: 58
Number of threads sleeping on a monitor: 7
Number of deadlocks: 0
Number of Monitors without locking threads: 1

18% of all threads are waiting for a monitor.

This might indicate a congestion or even a deadlock. If a monitor doesn't have a locking thread, it might be held by some external resource or system thread. You should check the [waiting threads](#).

This thread dump contains monitors without a locking thread information.
This means, the monitor is held by a system thread or some external resource.

You should check the monitors without locking threads for more information.

TDA - Thread Dump Analyzer 2.2 30.9MB/41.1PB

File View Tools Help

C:\Users\user\Desktop\log\test.log

- Logfile
- Dump No. 1 at line 2505
- Dump No. 2 at line 4228
- Dump No. 3 at line 5960
- Dump No. 4 at line 7714
- Dump No. 5 at line 9530
- Dump No. 6 at line 11236
- Long running thread detection
- Long running thread detection

| Name | Type | Prio | Thread-ID | Native-ID | State | Address Range |
|-------------------------------|--------|------|-------------|-----------|-----------------|-----------------------|
| AdapterThread | Daemon | 10 | 7220032 | 3524 | waiting on c... | [0x0000000000000000] |
| CompilerThread0 | Daemon | 10 | 7220752 | 5080 | waiting on c... | [0x0000000000000000] |
| Signal Dispatcher | Daemon | 10 | 7465104 | 6916 | waiting on c... | [0x0000000000000000] |
| [STANDBY] ExecuteThrea... | Daemon | 6 | 1034420000 | 7700 | in Object.w... | [0x0000000045eef... |
| MutlThreadedHttpConnecti... | Daemon | 6 | 1059241520 | 3368 | in Object.w... | [0x0000000045e4f... |
| DoSManager | Daemon | 6 | 933928512 | 1268 | waiting on c... | [0x000000003d12f... |
| weblogic.time.TimeEventG... | Daemon | 10 | 7473360 | 7460 | in Object.w... | [0x00000000000faef... |
| ExecuteThread: '4' for que... | Daemon | 6 | 933925072 | 6340 | runnable | [0x00000000003cef... |
| Finalizer | Daemon | 6 | 7460288 | 2396 | in Object.w... | [0x0000000000776f... |
| ExecuteThread: '1' for que... | Daemon | 6 | 933923008 | 7296 | runnable | [0x00000000001f4f... |
| Surrogate Locker Thread (... | Daemon | 6 | 7460978 | 7648 | waiting on c... | [0x0000000000000000] |
| introscope GC Monitor | Daemon | 6 | 7466480 | 7656 | waiting on c... | [0x00000000007caf... |
| Java2D Disposer | Daemon | 10 | 1052704464 | 6248 | in Object.w... | [0x0000000045acf... |
| UnknownHub Hub Transmit 2 | Daemon | 6 | 1062876144 | 7164 | in Object.w... | [0x00000000001f53f... |
| UnknownHub Hub Receive 2 | Daemon | 6 | 41882501984 | 2420 | runnable | [0x000000003d74f... |
| TestScheduler_Worker-0 | Task | 6 | 225456256 | 6700 | in Object.w... | [0x00000000000efaf... |
| weblogic.GCMonitor | Daemon | 6 | 1057526768 | 6240 | in Object.w... | [0x0000000000129f... |
| LDAPConnThread-1 [ldap://...] | Daemon | 6 | 933929200 | 6556 | runnable | [0x0000000003d16f... |
| [ACTIVE] ExecuteThread: '... | Daemon | 6 | 1041697104 | 7760 | waiting for... | [0x000000004607... |
| [STANDBY] ExecuteThrea... | Daemon | 6 | 1041695728 | 7484 | in Object.w... | [0x0000000045fff... |
| TestScheduler_QuartzSch... | Task | 6 | 1057522640 | 460 | waiting on c... | [0x0000000000106f... |
| [STANDBY] ExecuteThrea... | Daemon | 6 | 225455568 | 1044 | in Object.w... | [0x000000000007f51... |

Long running thread detection

Dump No. 1, Dump No. 2, Dump No. 3, Dump No. 4, Dump No. 5, Dump No. 6

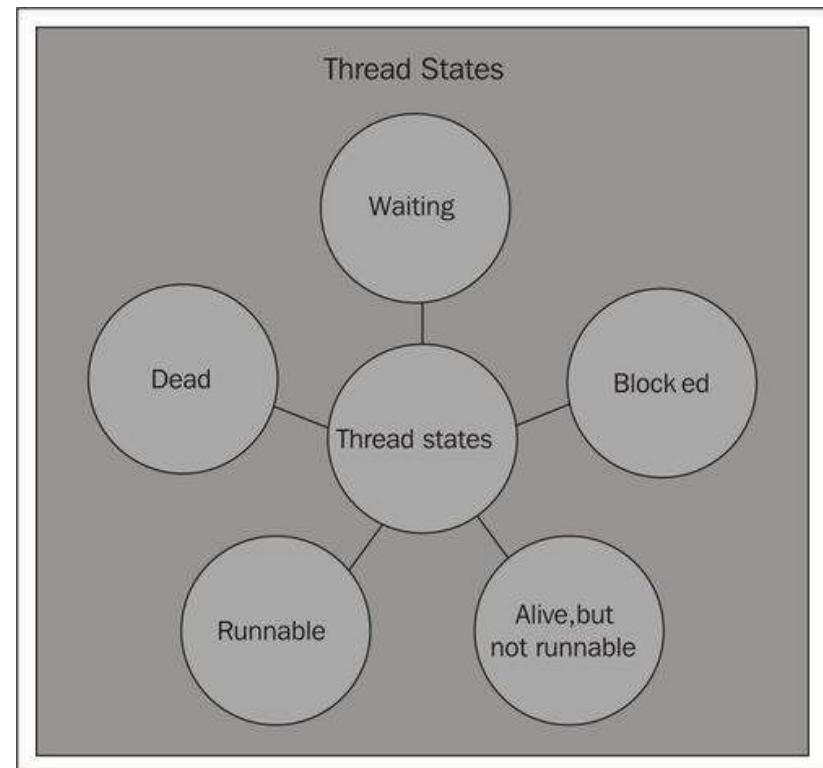
| | |
|-------------------------------|----|
| Overall Thread Count | 74 |
| Minimum Occurrence of threads | 6 |

TDA - Thread Dump Analyzer 2.2

33.9MB/41.9GB

Thread dump analysis using the Thread Dump Analyzer

- Thread states can be classified into five types.
- The following figure shows the different thread states:



Errors and their solutions

Errors can be classified into three major types based on different components of the enterprise application:

- Application
- JVM (memory)
- Database

Let's discuss some of the exceptions and their solutions that help in making the environment stable.

Errors and their solutions

Out of Memory exception

- In an enterprise environment, out of memory issues are encountered on a regular basis due to a high memory requirement of applications and the administrator has to tune the JVM.
- Failure of this causes an Out of Memory exception for the Tomcat instance.

Exception:

SEVERE: Servlet.service() for servlet jsp threw exception
java.lang.OutOfMemoryError: Java heap space

Errors and their solutions

- You have to add the following Java parameters in the startup script of Tomcat, which can be found in **TOMCAT_HOME/bin**, to increase the JVM allocation based on the memory requirement and recycle the Tomcat instance.

JAVA_OPTS="-Xms512m Xmx1048m

OutOfMemoryError: PermGen space

- Tomcat administrators often face the problem with the application's permanent object generation, as every application has different requirements of object generation.
- Hence, application slowness also results in the generation of OutOfMemoryError: PermGen space exception in catalina.out.
- Exception:

MemoryError: PermGen space

java.lang.OutOfMemoryError: PermGen space

Errors and their solutions

Solution:

- The following parameter should be added to the startup script of Tomcat 8.
- The parameter will increase the permanent generation space at the time of startup of Tomcat 8.

-XX:MaxPermSize=(MemoryValue)m

- For example:

-XX:MaxPermSize=128m

Stack over flow exception

- We have come across this issue in many applications. This exception is mainly caused due to recursive class loading (improper coding).
- This issue also causes performance degradation for the application: we observe that the application was working fine an hour ago but then it becomes unresponsive.

Errors and their solutions

Exception:

at java.lang.Thread.run(Thread.java:534)

----- Root Cause -----

java.lang.StackOverflowError

```
java.lang.StackOverflowError: Invoker.service() exception
  at org.apache.catalina.services.InvokerService.servReq(InvokerService.java:524)
  ...
  at org.apache.catalina.core.StandardPipeline$StandardPipelineValveContext.invokeNext(StandardPipeline.java:647)
  at LocalSetValve.invoke(LocalSetValve.java:103)
  at org.apache.catalina.core.StandardPipeline$StandardPipelineValveContext.invokeNext(StandardPipeline.java:641)
  at org.apache.catalina.core.StandardPipeline.invoke(StandardPipeline.java:580)
  at org.apache.catalina.core.ContainerBase.invoke(ContainerBase.java:993)
  at org.apache.coyote.ajp.Ajp13Processor.process(Ajp13Processor.java:457)
  at org.apache.coyote.ajp.Ajp13Processor.run(Ajp13Processor.java:376)
  at java.lang.Thread.run(Thread.java:534)
----- Root Cause -----
java.lang.StackOverflowError.
```

Errors and their solutions

Solution:

- You have to increase the value of the -xss parameter in the startup file of Tomcat.

-Xss=(memory value in k)

- For example:

-Xss=128k

- By default, the stack overflow exception comes with the value of 64 k followed by the recycle.

Errors and their solutions

The following screenshot shows the error in the logs:

- Exception:

at

java.lang.Thread.run(Thread.java:619)
Caused by: ClientAbortException:
java.net.SocketException: Broken pipe

```
        at
org.apache.catalina.core.StandardContextValve.invoke(StandardContextValv
e.java:175)

        at
org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java
:128)

        at
org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java
:102)

        at
org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.
java:109)

        at
org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:2
86)

        at
org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:84
4)

        at
org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.process(
Http11Protocol.java:583)

        at
org.apache.tomcat.util.net.JIoEndpoint$Worker.run(JIoEndpoint.java:447)

        at java.lang.Thread.run(Thread.java:619)

Caused by: ClientAbortException: java.net.SocketException: Broken pipe
```

Timeout waiting for an idle object

- Many times, when we click on the application for any transaction, the application displays a blank page after a while.
- It seems that the application server does not respond but the truth may differ.
- In many cases, the actual culprit is the database.
- What happens is the application server sends the request to the database and waits for the response, but the connection is abnormally terminated at the server causing a connection timeout exception.

Errors and their solutions

- Exception:

at

org.apache.commons.dbcp.PoolingDataSource.getConnection (PoolingDataSource.java:104)

Caused by: java.util.NoSuchElementException: Timeout waiting for idle object

```
[ERROR] 2010-03-05 23:57:38,636 [Servlet - service()] For servlet action threw exception  
java.lang.reflect.InvocationTargetException: Could not open JDBC Connection for transaction.  
nested exception is org.apache.commons.dbcp.SQLNestedException: Cannot get a connection, pool error: Timeout waiting for idle object  
Caused by: org.apache.commons.dbcp.SQLNestedException: Cannot get a connection, pool error:Timeout waiting for idle object  
at org.apache.commons.dbcp.PoolingDataSource.getConnection(PoolingDataSource.java:104)  
Caused by: java.util.NoSuchElementException: Timeout waiting for idle object
```

Database connectivity exception

- This kind of issue is often reported in an enterprise environment, where installation of a new application is in process or the migration of an application is in process.
- It's an issue with an incorrect configuration with JNDI in Tomcat 8.
- Exception:
`java.lang.RuntimeException: Error initializing application. Error
Unable to load any specified brand or the default brand:
net.project.persistence.PersistenceException: Unable to load
brand from database.`

Errors and their solutions

- The previous error often indicates that the database could not be accessed.

Please check your database configuration or contact your system administrator: `java.sql.SQLException: Error looking up data source for name: jdbc/abc`

Web server benchmarking

- Now we know how to troubleshoot problems and find potential solutions in the systems.
- There is one more point left to discuss, Web server benchmarking. Without discussing this topic, troubleshooting in Tomcat 8 cannot be marked as complete.
- It's a process through which we gauge the performance of a web server, also known as Load testing.

Web server benchmarking

ApacheBench

- ApacheBench is a command-line tool for web server benchmarking.
- It comes under the Apache HTTP server and is very useful when we want to generate only HTTP threads. It's a single thread process.

JMeter

- JMeter is one of the widely used open source tools used for load testing.
- This tool is developed under the Apache Jakarta project.
- It is capable of generating traffic for JDBC, web services, HTTP, HTTPS, and JMS services.
- It's a desktop software, which does not support all features of browsers

Summary

- In this lesson, we have discussed different issues faced by the application and web administrators in a real-time environment
- How to avoid these issues in the production environment using different techniques with errors and their solutions, thread dump analysis and tools used for analysis, memory issues, steps for troubleshooting real-time problems, and web server benchmarking.

8. Monitoring and Management of Tomcat 8



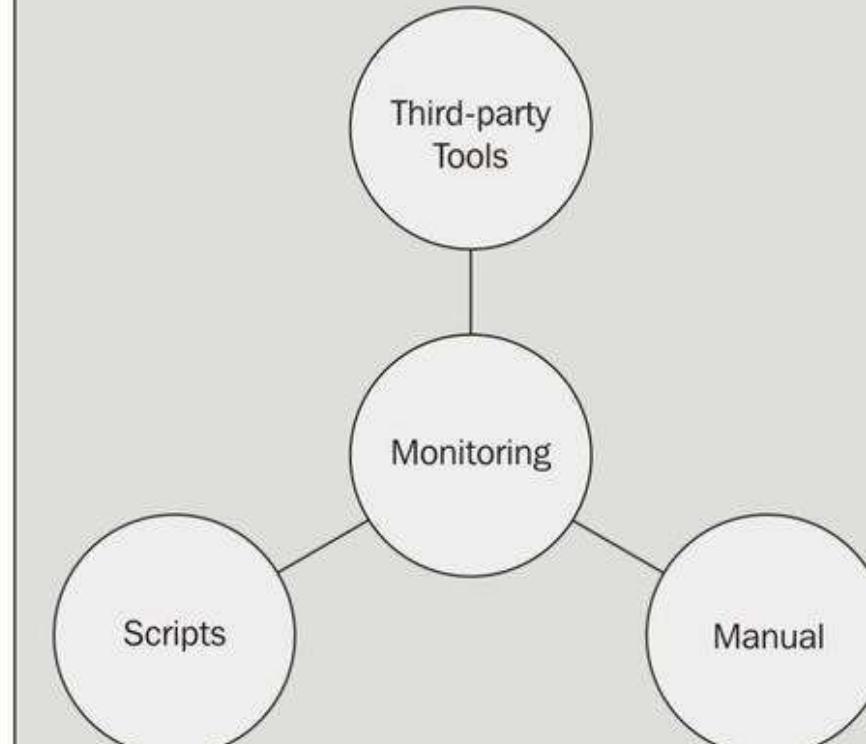
Monitoring and Management of Tomcat 8

In this lesson, we will discuss:

- How to monitor Tomcat 8
- Management of applications using the Tomcat Manager
- A third-party utility used for monitoring Tomcat 8

Different ways of monitoring

Different Ways of Monitoring



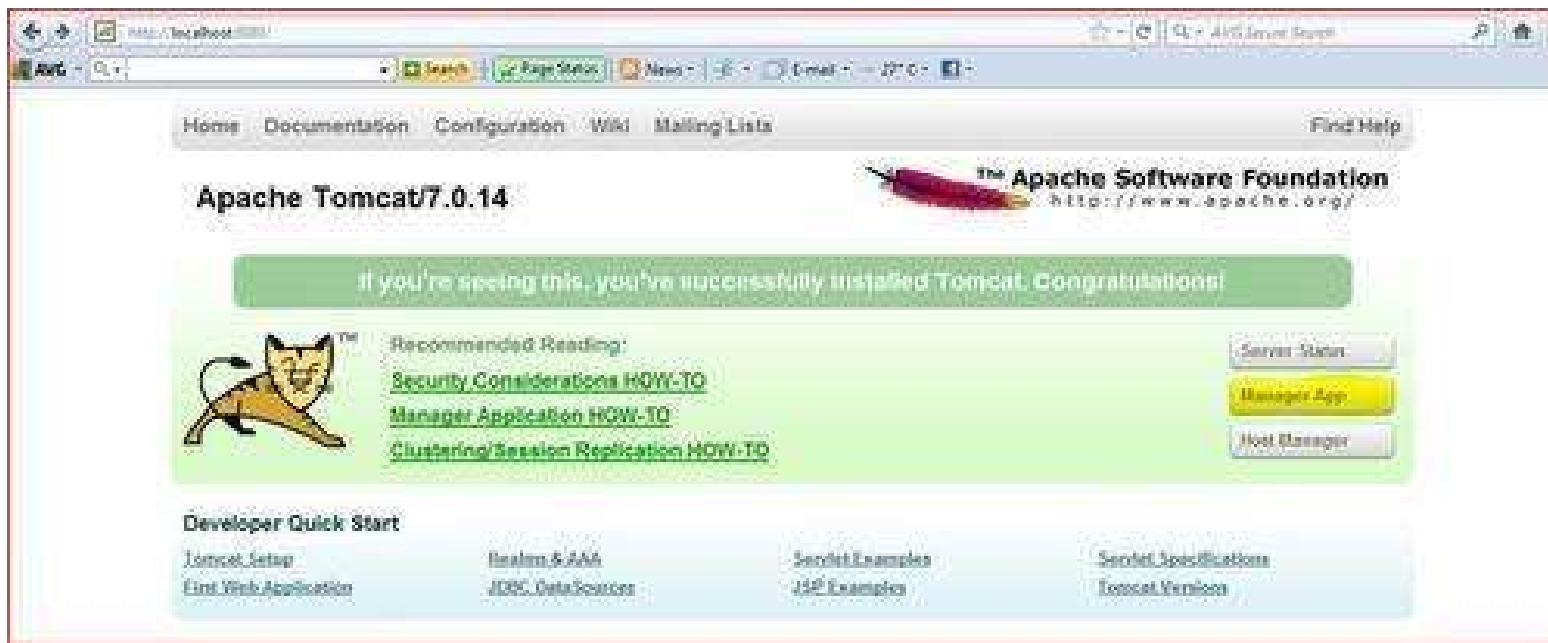
Monitoring setup for a web application and database server

| Monitored component | Benefit | Web server | Application server | Database server |
|---------------------------|--|------------|---|-----------------|
| CPU | Proactive measure to identify system issues | Yes | Yes | Yes |
| Physical Memory (RAM) | Proactive measure to identify system issues | Yes | Yes | Yes |
| JVM | Proactive measure to identify system issues | No | Yes | Yes |
| HTTP connection | Helps us identify the performance of the web server | Yes | Yes (only if HTTP services are running) | No |
| AJP connection | Helps us find out the connectivity of the web/application server | Yes | Yes | No |
| Database connection count | Helps us identify the performance of the database server | No | No | Yes |
| Connection Idle | Helps us identify the issues of the database server | No | No | Yes |
| Disk space | Proactive measure to identify system issues (Disk out of space) | Yes | Yes | Yes |
| Error code in logs | Helps us identify potential issues on systems | No | Yes | Yes |

Tomcat Manager in Tomcat 8

- Allow remote deploy, rollback, start, and stop features for the administrator.
- Provide detailed monitoring status for the application and server.
- Administrators need not stay in the office 24x7, In case of any issues, he/she can log in to the Tomcat Manager to resolve the issue.
- In short, we can say remote administration of Tomcat becomes very easy for administrators.

Tomcat Manager in Tomcat 8



Tomcat Manager in Tomcat 8

- The following screenshot shows the application's status and the different deployment-related tasks performed during application support:

| Applications | | | | | | |
|--------------|----------------|--------------------------|---------|----------|--|--|
| Name | Version | Display Name | Running | Sessions | Commands | |
| / | None specified | Welcome to Tomcat | true | 3 | Start Stop Reload Undeploy | Expire sessions with idle > 30 minutes |
| docs | None specified | Tomcat Documentation | true | 0 | Start Stop Reload Undeploy | Expire sessions with idle > 30 minutes |
| examples | None specified | Servlet and JSP Examples | true | 0 | Start Stop Reload Undeploy | Expire sessions with idle > 30 minutes |

Tomcat Manager in Tomcat 8

The screenshot shows the Tomcat Manager interface with the following sections:

- Deploy:** A form for deploying a WAR file. It includes fields for "Context Path (required)" (with a placeholder "com"), "XML Configuration file (optional)", "WAR or Directory URL...", and a "Deploy" button.
- WAR file to deploy:** A section for uploading a WAR file. It has a "Select WAR file to upload..." input field, a "Browse..." button, and a "Deploy" button.
- Diagnostics:** A section for checking memory leaks. It includes a "Find leaks" button and a note: "This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems."
- Server Information:** A table showing system details:

| Tomcat Version | JVM Version | JVM Vendor | OS Name | OS Version | OS Architecture |
|----------------------|--------------|-----------------------|---------------|------------|-----------------|
| Apache Tomcat/7.0.14 | 1.6.0_24-b07 | Sun Microsystems Inc. | Windows Vista | 6.0 | x86 |

Monitoring in Tomcat 8

- Monitoring in Tomcat 8 can be done using the Tomcat Manager. By default, the Tomcat Manager provides the status of the server with a detailed description of requests and their status.
- This information is very useful to administrators at the time of troubleshooting.
- Apart from this, the administrator need not log in to the machine for collecting this information.

Summary of the Server Status of Tomcat 8

| JVM | | | | | | |
|---|---------|--------|--------|---------------|-----------|------------------------------|
| Free memory: 8.38 MB Total memory: 13.55 MB Max memory: 247.50 MB | | | | | | |
| "ajp-bio-8009" | | | | | | |
| Max threads: 200 Current thread count: 0 Current thread busy: 0 Max processing time: 0 ms Processing time: 0.013 s Request count: 0 Error count: 0 Bytes received: 0.00 KB Bytes sent: 0.00 KB | | | | | | |
| Stage | Time | B Sent | B Recv | Client | VHost | Request |
| R | ? | ? | ? | ? | ? | ? |
| R | ? | ? | ? | ? | ? | ? |
| R | ? | ? | ? | ? | ? | ? |
| S | 2040 ms | 0 KB | 0 KB | 0.0.0.0.0.0.1 | localhost | GET /manager/status HTTP/1.1 |
| ? : Parse and prepare request; S: Service F: Finishing R: Ready K: Keepalive | | | | | | |

Application List

- This gives us the list of all applications hosted in Tomcat 8 and their URL mapping for the application's access.
- The following screenshot shows the application list deployment in the Tomcat 8 instance:



Summary of the Server Status of Tomcat 8

- The following screenshot shows the internal components with their statuses such as the status of the application response, servlet response, and the JSP responses:

The screenshot displays the Tomcat 8 Server Status page with the following sections and data:

- localhost/sample**:
 - Start time: Sun Sep 23 22:29:10 BST 2013 | Startup time: 23 ms | TLC scan time: 76 ms
 - Active sessions: 0 | Session count: 0 | Max active sessions: 0 | Rejected session count: 0 | Expired sessions: 0 | Longest session alive time: 0 s | Average session idle time: 0 s | Processing time: 0 ms
 - 204s issued: 0 | 500s received: 0
- JSP Servlet [/hello]**:
 - Processing time: 0.0 s | Max time: 0 ms | Request count: 0 | Error count: 0 | Load time: 0 ms | Classloading time: 0 ms
- Java Web Script [/]**:
 - Processing time: 0.0 s | Max time: 0 ms | Request count: 0 | Error count: 0 | Load time: 124 ms | Classloading time: 0 ms

Connections on the AJP

| JVM | | | | | | |
|---|--|--------|--------|---------------|-----------|----------------------------------|
| Free memory: 7.72 MB Total memory: 15.26 MB Max memory: 247.50 MB | | | | | | |
| "ajp-bio-8009" | | | | | | |
| Max thread: 200 Current thread count: 0 Current thread busy: 0 | Max processing time: 0 ms Processing time: 0.0 ms Request count: 0 Error count: 0 Bytes received: 0.00 KB Bytes sent: 0.00 KB | | | | | |
| Stage | Time | B Sent | B Recv | Client | VHost | Request |
| S | 22 ms | 0 KB | 0 KB | 0.0.0.0.0.0.1 | localhost | GET /manager/status/all HTTP/1.1 |
| R | ? | ? | ? | ? | ? | ? |
| R | ? | ? | ? | ? | ? | ? |
| "http-bio-8080" | | | | | | |
| Max thread: 200 Current thread count: 10 Current thread busy: 1 | Max processing time: 23009 ms Processing time: 6.004 ms Request count: 12 Error count: 0 Bytes received: 0.00 KB Bytes sent: 0.31 MB | | | | | |
| Stage | Time | B Sent | B Recv | Client | VHost | Request |
| S | 22 ms | 0 KB | 0 KB | 0.0.0.0.0.0.1 | localhost | GET /manager/status/all HTTP/1.1 |
| R | ? | ? | ? | ? | ? | ? |
| R | ? | ? | ? | ? | ? | ? |

JConsole configuration on Tomcat 8

- Detect low memory
- Enable or disable the GC and class loading verbose tracing
- Detect deadlocks
- Control the log level of any loggers in an application
- Access the OS resources—Sun's platform extension
- Manage an application's Managed Beans (MBeans)

Remote JMX enabling

- By default, the following values are added to enable the details:

```
CATALINA_OPTS=-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.port=%my.jmx.port%
\ -Dcom.sun.management.jmxremote.ssl=false \
-Dcom.sun.management.jmxremote.authenticate=false
```

Remote JMX enabling

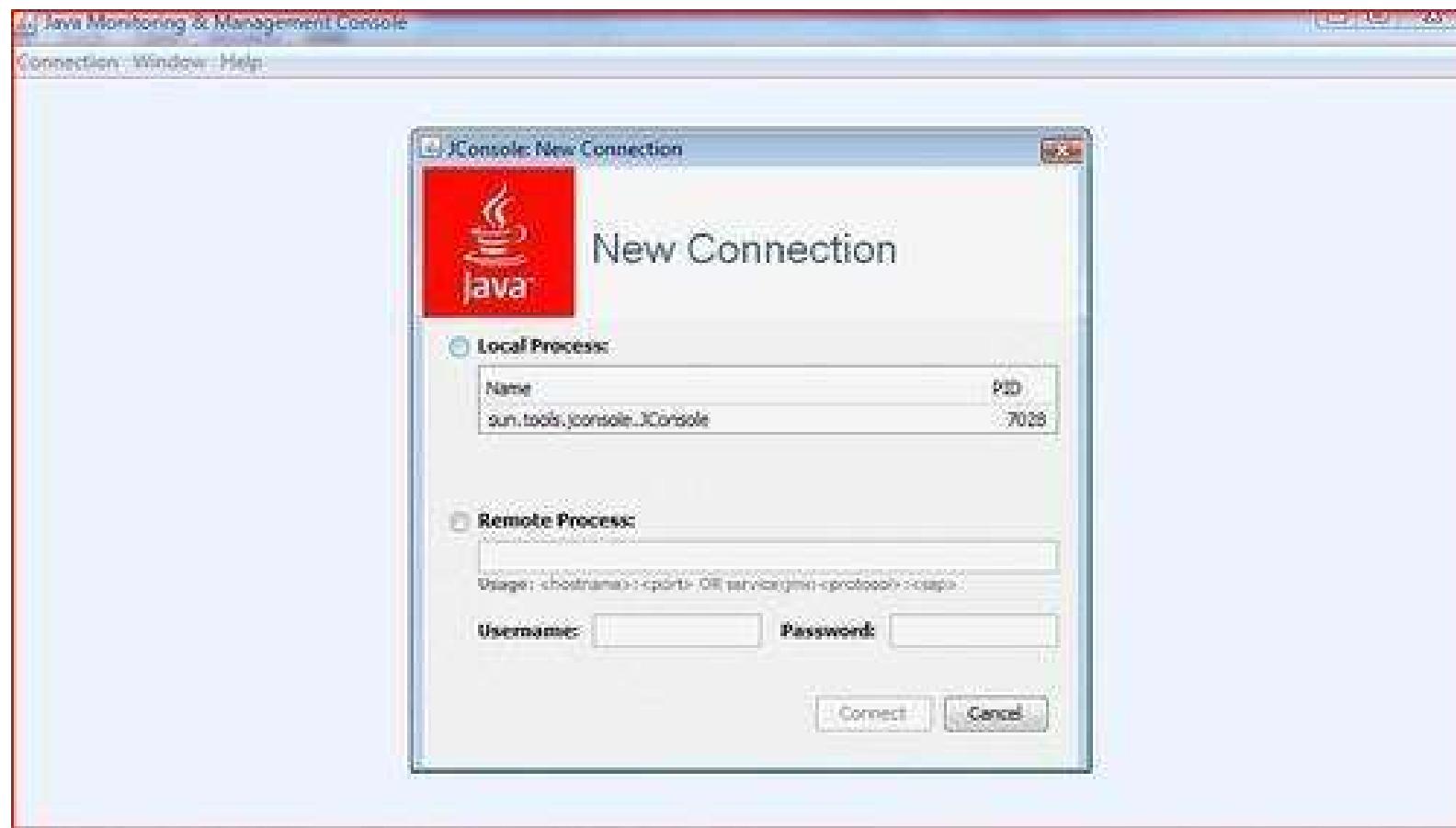
- Let's do the real-time configuration on Tomcat 8 and understand the meaning of each parameter:

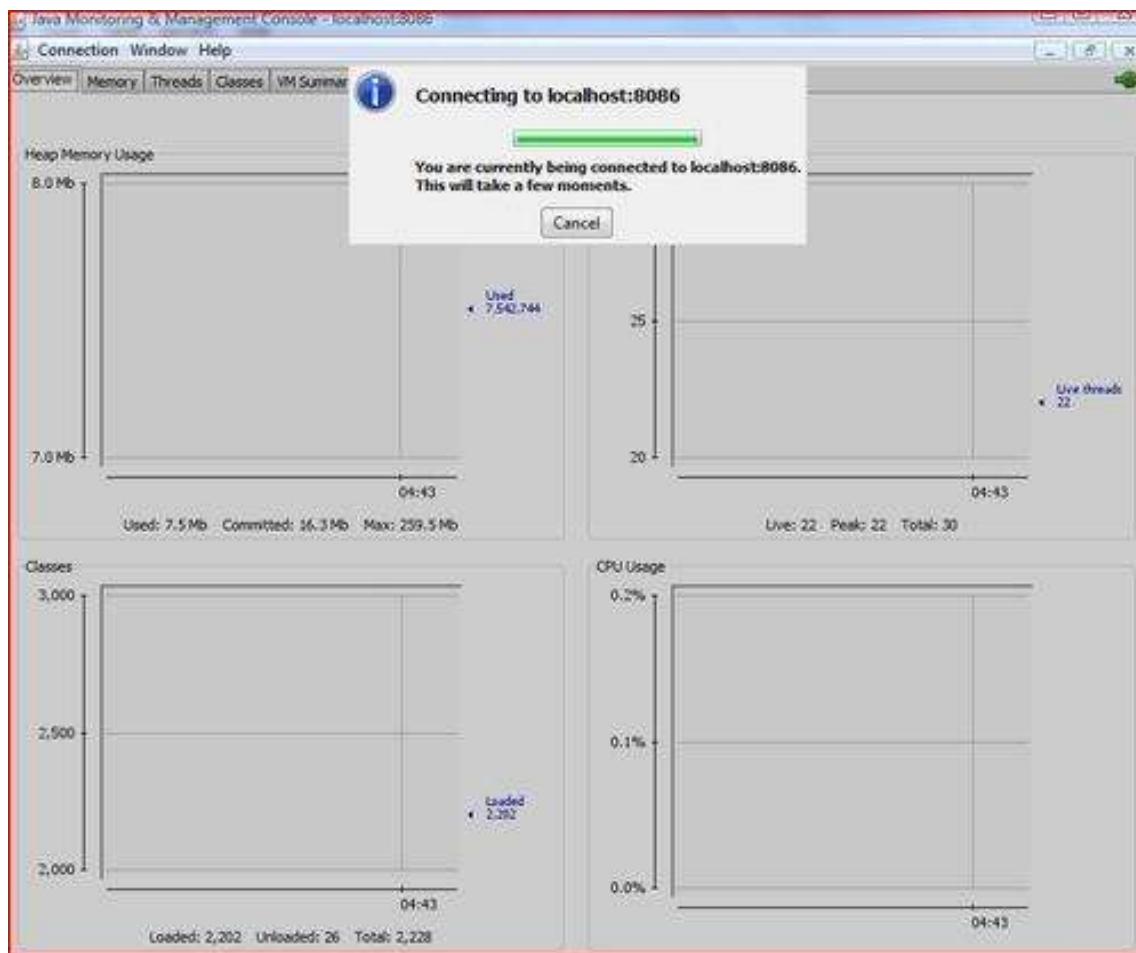
```
CATALINA_OPTS="-Djava.awt.headless=true -  
Xmx128M -server -Dcom.sun.management.jmxremote -  
Dcom.sun.management.jmxremote.port=8086 -  
Dcom.sun.management.jmxremote.authenticate=false -  
Dcom.sun.management.jmxremote.ssl=false"
```

How to connect to the JConsole

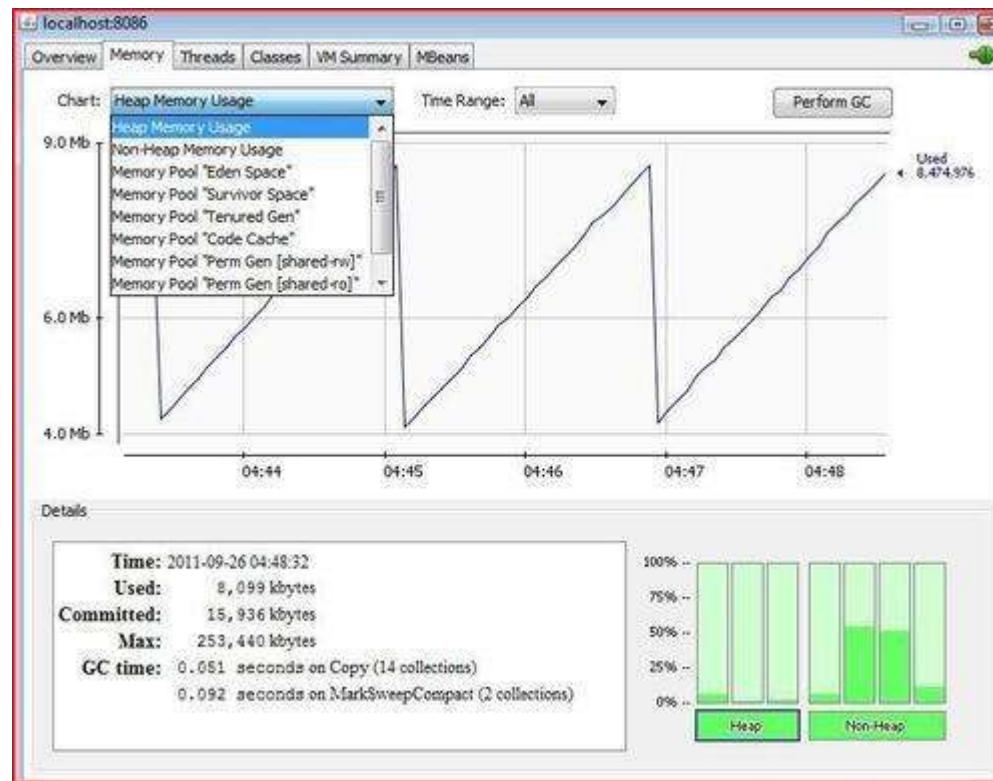
- Once Tomcat 8 configurations are done, it's time to connect to Tomcat 8 through the JConsole remotely using the command jconsole, as follows. It will open the GUI interface.
- We have to provide the IP address and port for the server where we want to connect; in our case, it's localhost and 8086.
- The following screenshot shows the default console for the JConsole:

[root@localhost bin] # jconsole

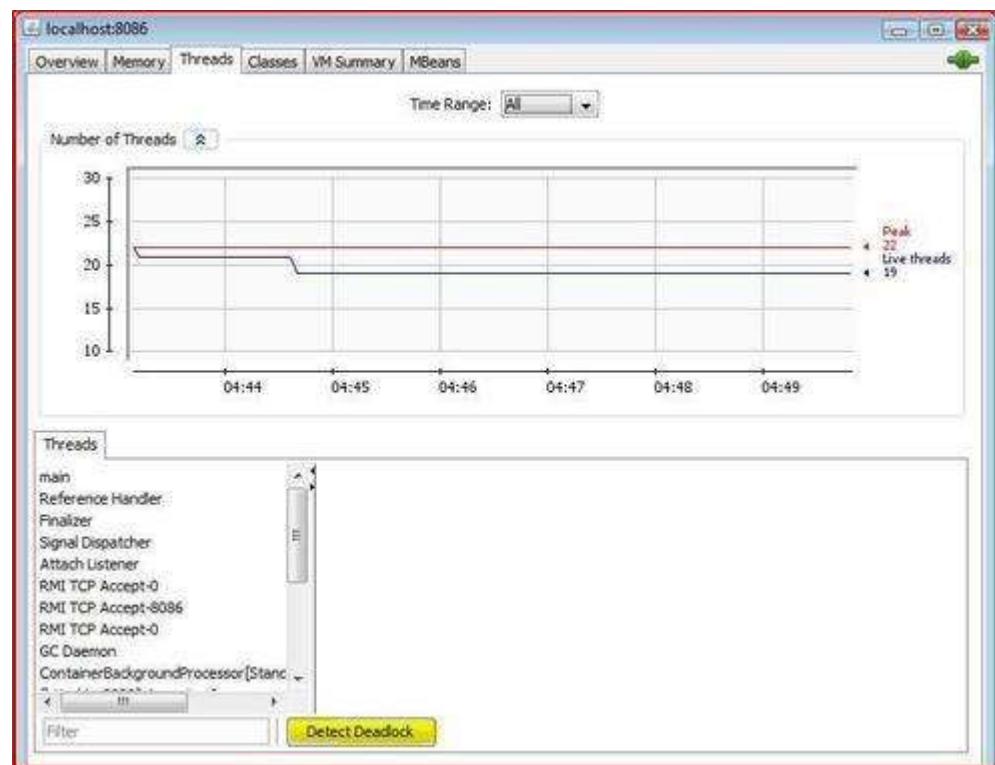




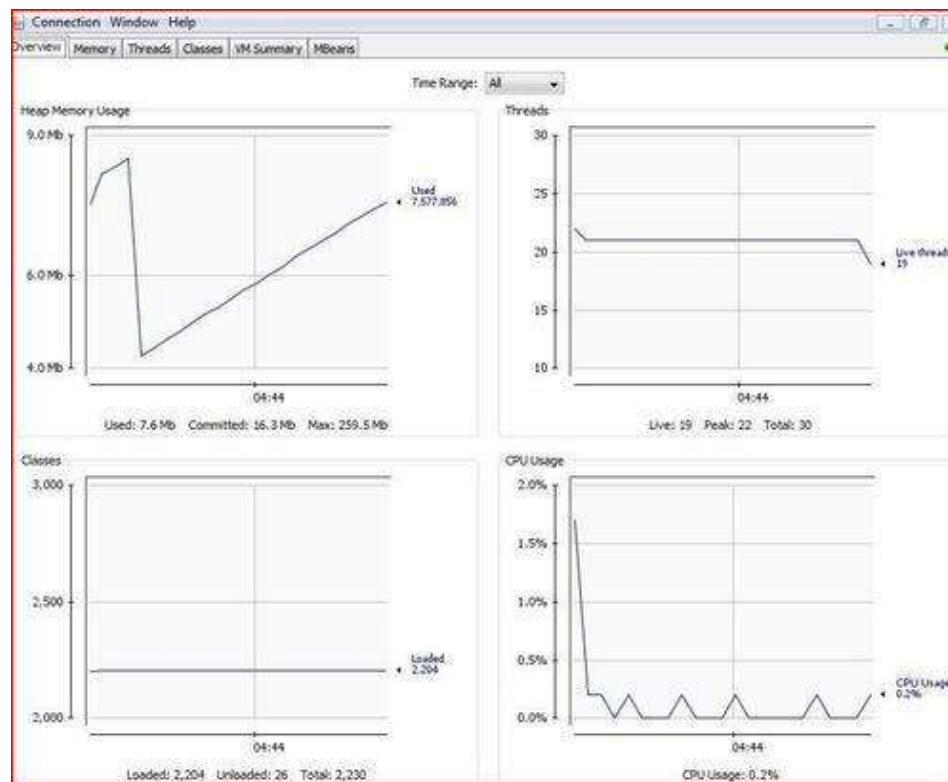
Different tabs for the JConsole and their features

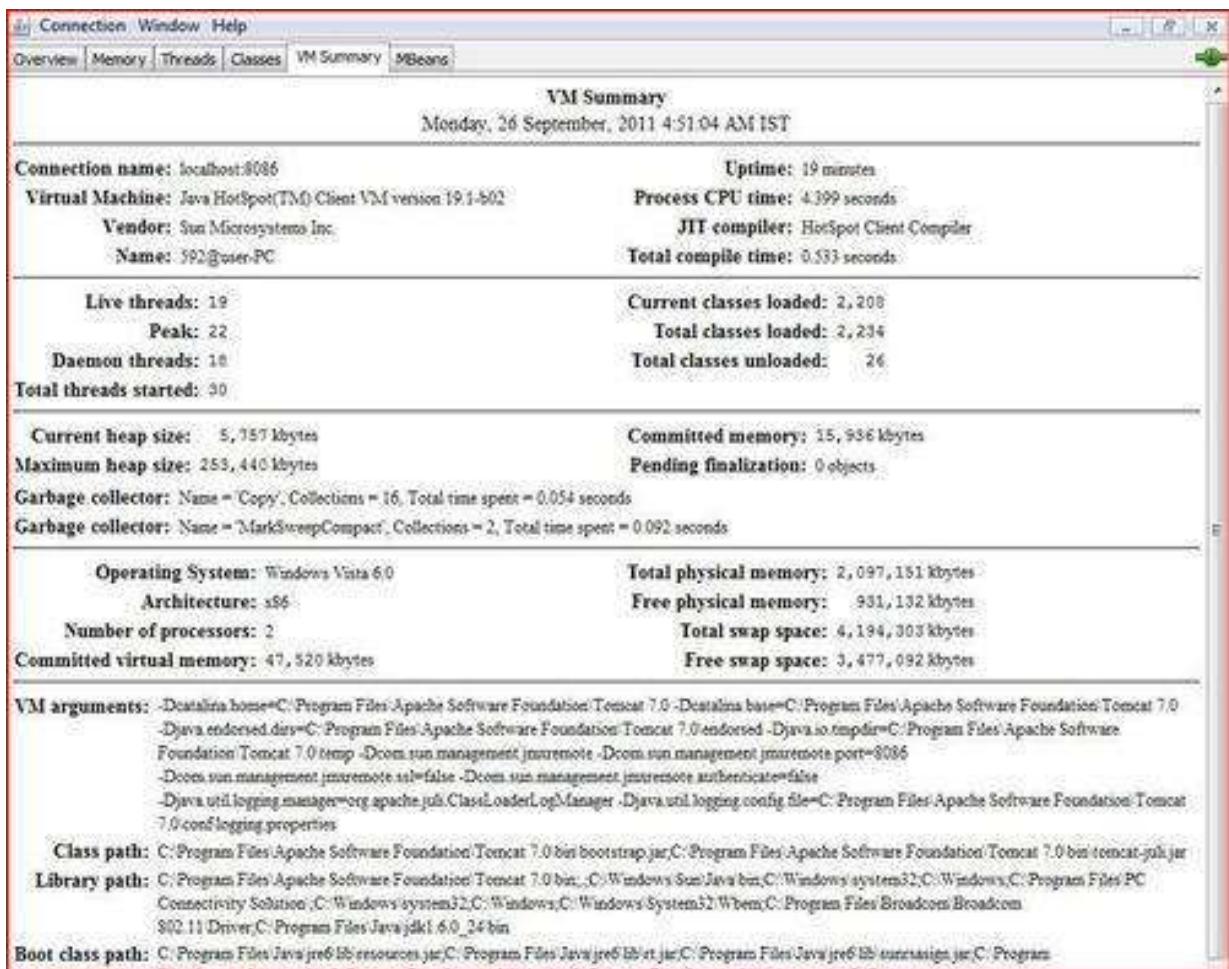


Different tabs for the JConsole and their features

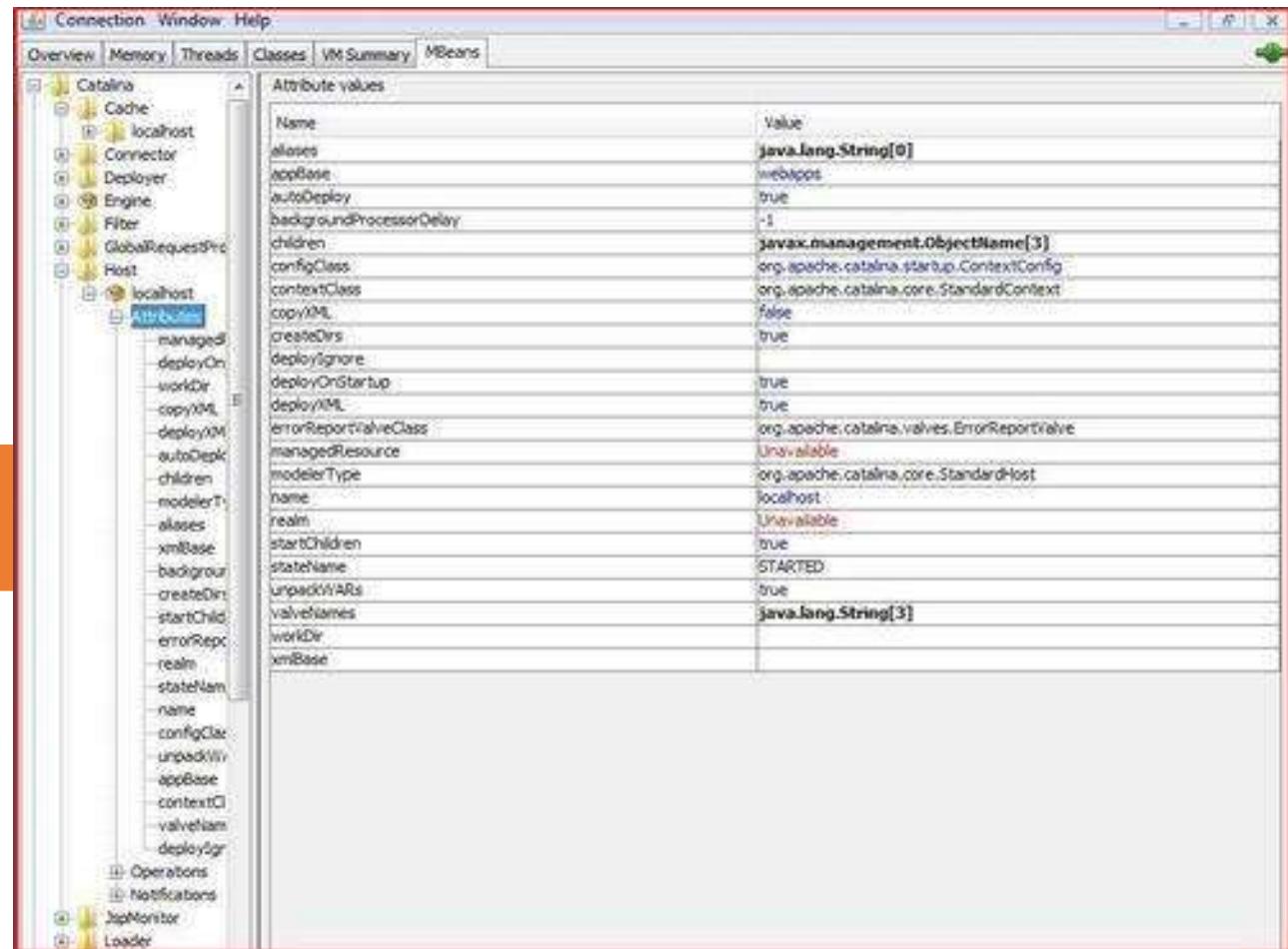


VM Summary and Overview

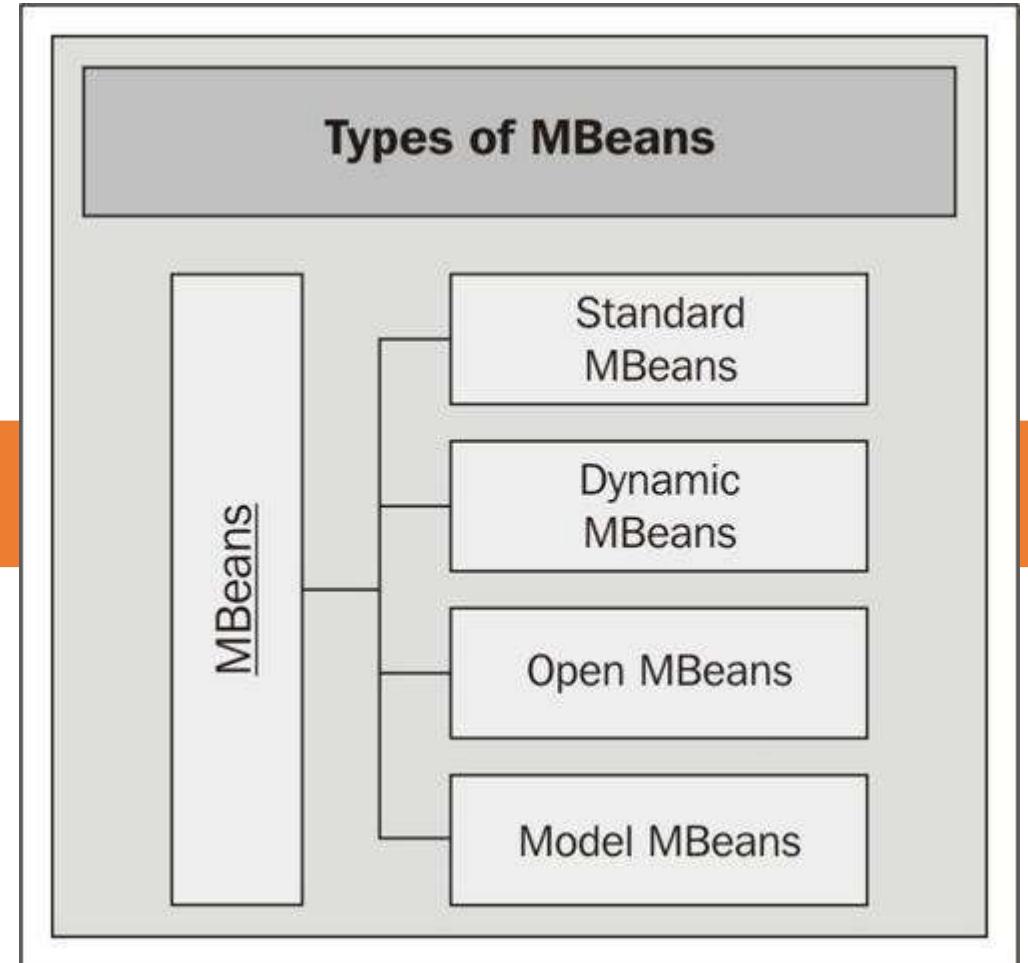




MBeans



Types of MBeans



Java Monitoring & Management Console - localhost:8005

File Connection Window Help

Overview Memory Threads Classes VM Summary MBeans

Catalina

- Cache
- Connector
 - 8009
 - Attributes
 - Operations
 - Notifications
 - 8080
 - Attributes
 - Operations
 - Notifications
 - 8443
 - Attributes
 - Operations
 - Notifications
- Deployer
- Engine
- Filter
- GlobalRequestProcessor
- Host
- JspMonitor
- Loader
- MBeanFactory
- Manager
- Mapper
- NamingResources
- ProtocolHandler
- Realm
- Resource
- Server
 - ServerClassloader
 - Service
 - Servlet
 - StringCache
 - ThreadPool
 - Valve
 - WebModule
 - WebappClassLoader
- Implementation
- MBeanServerDelegate

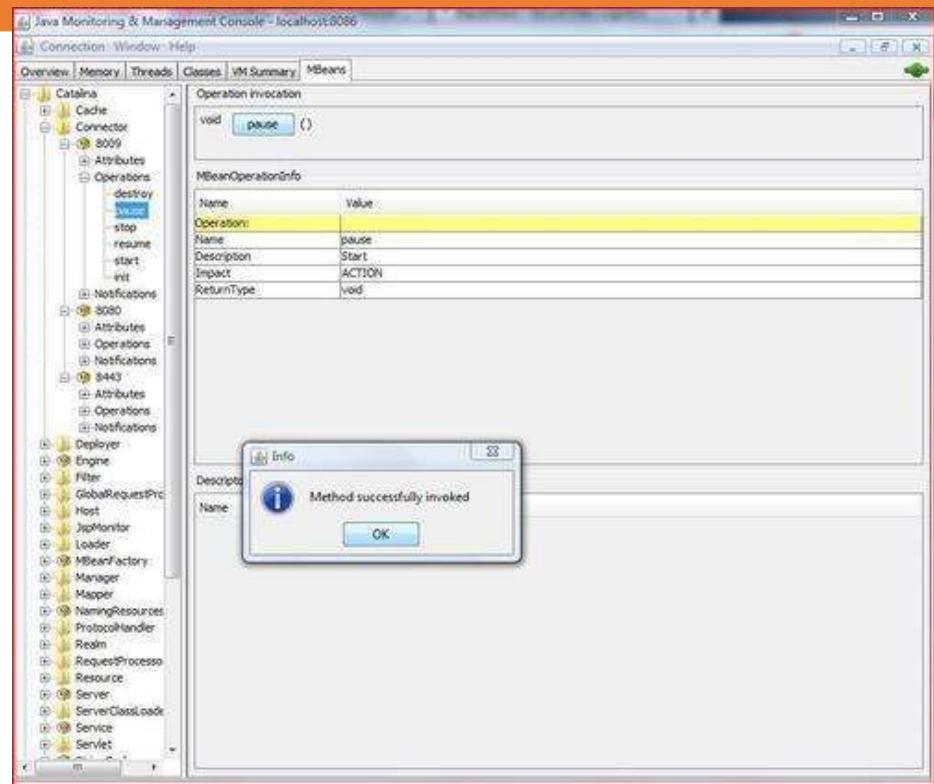
Attribute values

| Name | Value |
|--------------------------|-----------------------------------|
| URIEncoding | |
| acceptCount | 100 |
| address | |
| allowTrace | false |
| className | |
| connectionLinger | -1 |
| connectionTimeout | -1 |
| emptySessionPath | |
| enableLookups | false |
| executorName | Internal |
| keepAliveTimeout | -1 |
| maxKeepAliveRequests | |
| maxPostSize | 2097152 |
| maxSavePostSize | 4096 |
| maxThreads | 200 |
| minSpareThreads | 10 |
| modelerType | |
| packetSize | 8192 |
| port | 8009 |
| processorCache | 200 |
| protocol | AJP/1.3 |
| protocolHandlerClassName | org.apache.coyote.ajp.AjpProtocol |
| proxyName | |
| proxyPort | 0 |
| redirectPort | 8443 |
| scheme | http |
| secret | |
| secure | false |
| sslProtocols | |
| stateName | STARTED |
| tcpNoDelay | true |
| threadPriority | 5 |
| useEncodedForURLs | false |
| useIPHosts | false |
| xpoweredBy | false |

Refresh

Types of MBeans

- The following screenshot shows the pause operation successfully invoked for the HTTP Connector for Tomcat 8:



Summary



- In this lesson, we have discussed the various processes of monitoring in Tomcat 8 and their components using the Tomcat Manager and JConsole, such as different ways of monitoring, how monitoring is done in Tomcat 8, JConsole, and how it is used.
- In the next lesson, we will discuss the high availability setup for Tomcat 8 using clustering, load balancing, high availability concepts, architecture design, scalability, and so on.

9. Clustering in Tomcat 8

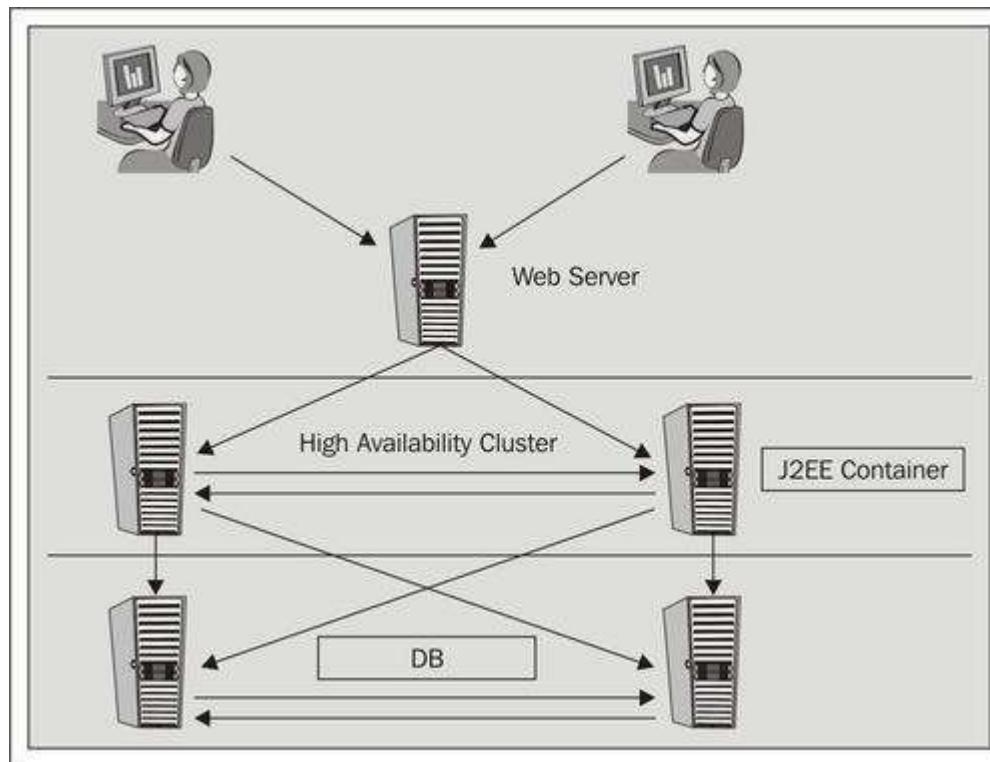


Clustering in Tomcat 8

In this lesson, we discuss:

- High availability architecture and its advantages
- Different types of high availability architectures including load balancing and clustering
- Approaches used by IT industries while building a high availability architecture in an enterprise setup
- How to do Apache Tomcat clustering
- Various clustering architectures
- How to solve common problems in clustering

What is a cluster?



Benefits of clustering

There are many advantages of clustering in a middleware environment. It also depends on which cluster techniques we are using. We will discuss the various advantages of clustering:

- Scalability
- High availability
- High performance
- Cloud computing

Disadvantages of clustering

Until now, we have discussed how useful clustering is for a web environment. Let's discuss the disadvantages of clustering:

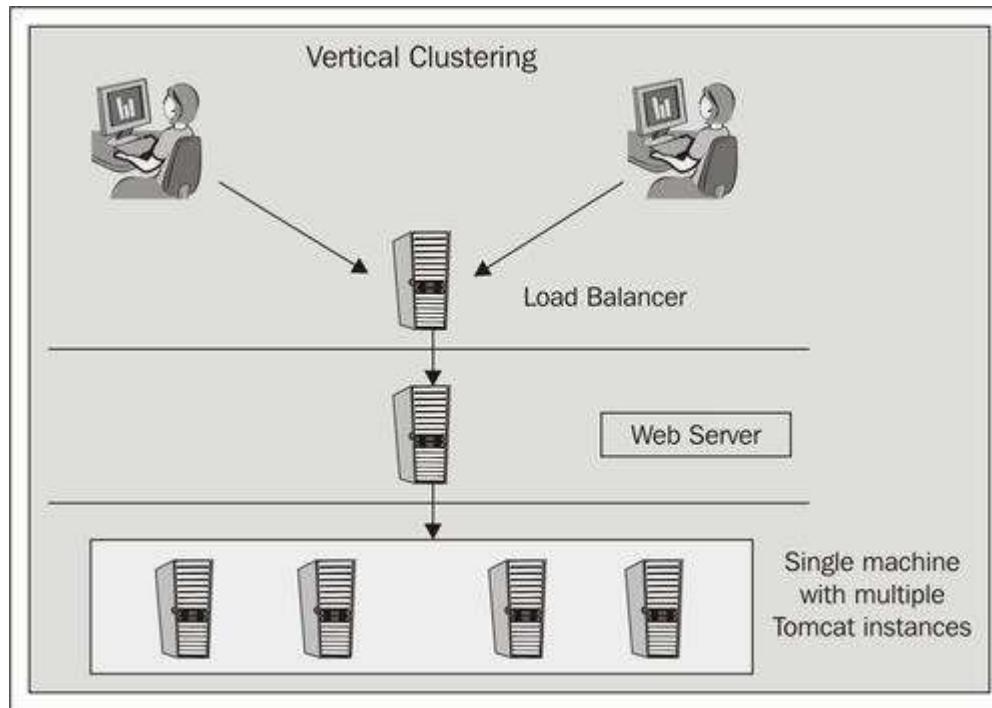
- Cost
- Monitoring

Clustering architecture

In this topic, we will discuss the various architectures of clustering used by IT industries. These architectures may vary on each implementation, depending on the application and business requirements. There are basically two types of clustering architectures implemented in a real-time IT infrastructure:

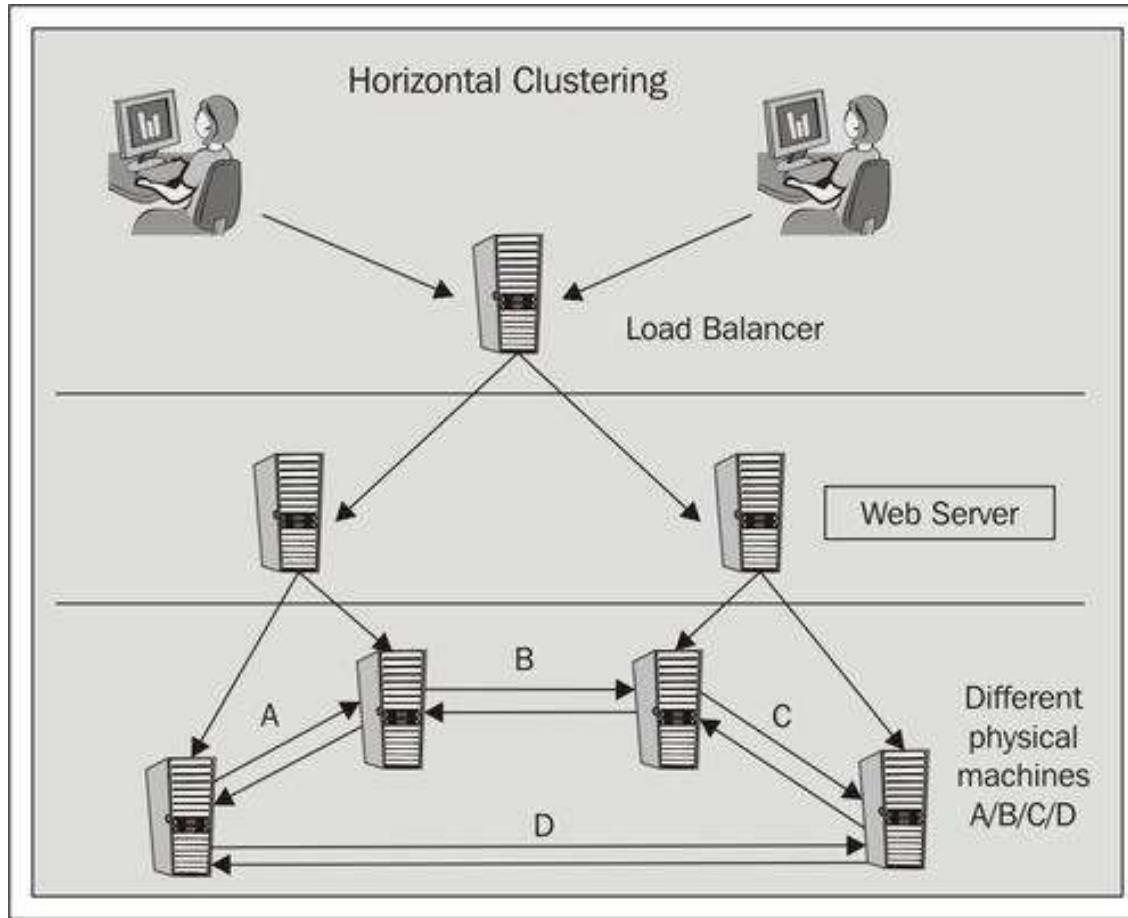
- Vertical clustering
- Horizontal clustering

Vertical clustering



Horizontal clustering

- In this type of clustering method, instances are configured separately on each physical machine and connected through high speed Ethernet.
- It's a very popular implementation technique in the production environment.
- Resources of one machine are not shared with the other machine & Also, failover can be done in the case of hardware failure.



Vertical clustering in Apache Tomcat 8

For vertical clustering, we have to configure at least two instances of Apache Tomcat and the complete process consists of three stages. Let's discuss and implement the steps for vertical cluster in Tomcat 8:

- Installation of the Tomcat instance.
- Configuration of the cluster.
- Apache HTTP web server configuration for the vertical cluster.

Installation of the Tomcat instance

- Download and unzip the software in the required directory.
- Install the JDK and set the JAVA_HOME.
- Copy the Apache Tomcat source code in two different directories, for example, /opt/tomcatX (where X= instance number) and verify that the files are properly copied on both instances, using the following command:

```
[root@localhost opt]# ls -l apache-tomcat*
```

```
apache-tomcat1:  
total 160  
drwxr-xr-x 2 root root 4096 Oct  8 14:15 bin  
drwxr-xr-x 3 root root 4096 Oct  8 13:59 conf  
drwxr-xr-x 2 root root 4096 May 22 15:08 lib  
-rw-r--r-- 1 root root 57851 Apr  1 2011 LICENSE  
drwxr-xr-x 2 root root 4096 Oct 11 15:30 logs  
-rw-r--r-- 1 root root 1230 Apr  1 2011 NOTICE  
-rw-r--r-- 1 root root 9031 Apr  1 2011 RELEASE-NOTES  
-rw-r--r-- 1 root root 6860 Apr  1 2011 RUNNING.txt  
drwxr-xr-x 3 root root 4096 Oct  8 13:59 temp  
drwxr-xr-x 7 root root 4096 Apr  1 2011 work  
drwxr-xr-x 3 root root 4096 May 16 21:03 webapps  
  
apache-tomcat2:  
total 116  
drwxr-xr-x 2 root root 4096 Oct  8 13:54 bin  
drwxr-xr-x 3 root root 4096 Oct  8 13:56 conf  
drwxr-xr-x 2 root root 4096 May 22 15:08 lib  
-rw-r--r-- 1 root root 57851 Apr  1 2011 LICENSE  
drwxr-xr-x 2 root root 4096 Oct 11 15:30 logs  
-rw-r--r-- 1 root root 1230 Apr  1 2011 NOTICE  
-rw-r--r-- 1 root root 9031 Apr  1 2011 RELEASE-NOTES  
-rw-r--r-- 1 root root 6860 Apr  1 2011 RUNNING.txt  
drwxr-xr-x 3 root root 4096 Oct  8 13:57 temp  
drwxr-xr-x 7 root root 4096 Apr  1 2011 work  
drwxr-xr-x 3 root root 4096 May 16 21:03 webapps
```

Configuration of instance 1

- Shutdown port: The following screenshot shows the configuration for the shutdown port for the Tomcat instance.
- While running multiple instances, if by any chance, you have skipped configuring the shutdown port, then the Tomcat instance will be unable to start it.

```
-->
<Server port="8006" shutdown="SHUTDOWN">
  <!-- Security listener. Documentation at /docs/config/listeners.html
  <Listener className="org.apache.catalina.security.SecurityListener" />
  -->
  <!--APR library loader. Documentation at /docs/apr.html -->
```

Configuration of instance 1

```
<!-- A "Connector" represents an endpoint by which requests are received  
and responses are returned. Documentation at :  
Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)  
Java AJP Connector: /docs/config/ajp.html  
APR (HTTP/AJP) Connector: /docs/apr.html  
Define a non-SSL HTTP/1.1 Connector on port 8080  
-->  
<Connector port="8080" protocol="HTTP/1.1"  
connectionTimeout="20000"  
redirectPort="8443" />  
<!-- A "Connector" using the shared thread pool-->  
</>
```

Configuration of instance 1

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

- **Cluster attributes:** Enable the cluster attributes for clustering in server.xml and the following screenshot shows the cluster class used for clustering:

```
<!--For clustering, please take a look at documentation at:
    /docs/cluster-howto.html (simple how to)
    /docs/config/cluster.html (reference documentation) -->

<Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
```

Configuration of instance 1

- Configuration test: Run the configtest.sh script from **TOMCAT_HOME/bin** to check the configuration.
- The following screenshot shows the output for the following config.sh command:

[root@localhost bin]# ./configtest.sh

```
[root@localhost bin]# ./configtest.sh
Using CATALINA_HOME: /opt/apache-tomcat
Using CATALINA_BASE: /opt/apache-tomcat
Using CATALINA_TMPDIR: /opt/apache-tomcat/temp
Using CATALINA_PID: /opt/apache-tomcat.pid
Using CLASSPATH: /opt/apache-tomcat/bin/bootstrap.jar:/opt/apache-tomcat/bin/tomcat-juli.jar
Oct 11, 2011 10:48:01 PM org.apache.catalina.core.AprLifecycleListener init
INFO: The APR based Apache Tomcat Native library which allows optimal performance in production environments was not found on the java.library.path: /opt/jdk1.6.0_24/jre/lib/i386/server:/opt/jdk1.6.0_24/jre/lib/i386:/opt/jdk1.6.0_24/jre/lib:/opt/jdk1.6.0_24/jre:/usr/lib/jvm/java-6-sun/jre/lib/i386:/usr/lib/jvm/java-6-sun/jre:/usr/lib
Oct 11, 2011 10:48:01 PM org.apache.catalina.core.StandardProtocolHandler init
INFO: Initializing ProtocolHandler ["http-bio-8080"]
Oct 11, 2011 10:48:01 PM org.apache.catalina.startup.Catalina load
INFO: Initialization processed in 8148 ms
```

Configuration of instance 1

- Tomcat instance startup: Start the instance 1 configuration using the script startup.sh.
- The following screenshot shows the output for the following startup.sh script:

[root@localhost bin]# ./startup.sh

```
[root@localhost bin]# ./startup.sh
Using CATALINA_BASE:   /opt/apache-tomcat1
Using CATALINA_HOME:  /opt/apache-tomcat1
Using CATALINA_TMPDIR: /opt/apache-tomcat1/temp
Using JRE_HOME:        /opt/jdk1.6.0_24
Using CLASSPATH:       /opt/apache-tomcat1/bin/bootstrap.jar:/opt/apache-tomcat1/bin/tomcat-juli.jar
```

Configuration of instance 1

- Check the Tomcat instance process using the following mentioned command.
- The following screenshot shows the output for the ps command:

[root@localhost bin]# ps -ef |grep java

```
[root@localhost bin]# ps -ef |grep java
root      11766      1 11 17:00 pts/3    00:00:17 /opt/jdk1.6.0_24/bin/java -Djava
m -Xmx512m -XX:MaxPermSize=256m -Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.
.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.awt.headless=true
-e.port=7091 -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.manageme
sspath /opt/apache-tomcat1/bin/bootstrap.jar:/opt/apache-tomcat1/bin/tomcat-juli
-Djava.io.tmpdir=/opt/apache-tomcat1/temp org.apache.catalina.startup.Bootstrap
root      11902 10149  0 17:02 pts/3    00:00:00 grep java
[root@localhost bin]#
```

Configuration of instance 2

- Change the shutdown port for instance 2 in server.xml (increment it by 1).
- The following screenshot shows the configuration:

```
-->
<Server port="8007" shutdown="SHUTDOWN">
    <!-- Security listener. Documentation at /docs/config/listeners.html
        <Listener className="org.apache.catalina.security.SecurityListener" />
```

Configuration of instance 2

- Change the Connector and redirect the port for instance 2 in server.xml (increment it by 1).
- The following screenshot shows the configuration:

```
-->
<Connector port="8081" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8444" />
<!-- A "Connector" using the shared thread pool-->
```

- Change the AJP and redirect the port for instance 2 in server.xml (increment it by 1), The following screenshot shows the configuration:

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8010" protocol="AJP/1.3" redirectPort="8444" />
```

- Enable the cluster attributes for clustering in server.xml, The following screenshot shows the configuration:

```
<!--For clustering, please take a look at documentation at:
    /docs/cluster-howto.html (simple how to)
    /docs/config/cluster.html (reference documentation) -->

<Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
```

- Run the configtest.sh script from TOMCAT_HOME/bin to check the configuration.
- The following screenshot shows the output for the following startup.sh script:

[root@localhost bin]# ./configtest.sh

```
[root@localhost bin]# ./configtest.sh
Using CATALINA_BASE:   /opt/apache-tomcat2
Using CATALINA_HOME:   /opt/apache-tomcat2
Using CATALINA_TMPDIR: /opt/apache-tomcat2/temp
Using JRE_HOME:        /opt/jdk1.6.0_24
Using CLASSPATH:       /opt/apache-tomcat2/bin/bootstrap.jar:/opt/apache-tomcat2
Oct 11, 2011 5:21:07 PM org.apache.catalina.core.AprLifecycleListener init
INFO: The APR based Apache Tomcat Native library which allows optimal performance
1.6.0_24/jre/lib/i386/server:/opt/jdk1.6.0_24/jre/lib/i386:/opt/jdk1.6.0_24/jre/
Oct 11, 2011 5:21:13 PM org.apache.coyote.AbstractProtocolHandler init
INFO: Initializing ProtocolHandler ["http-bio-8081"]
Oct 11, 2011 5:21:14 PM org.apache.coyote.AbstractProtocolHandler init
INFO: Initializing ProtocolHandler ["ajp-bio-8010"]
Oct 11, 2011 5:21:14 PM org.apache.catalina.startup.Catalina load
INFO: Initialization processed in 14385 ms
```

Configuration of instance 2

- Start the configuration of instance 2 using the script startup.sh.
- The following screenshot shows the output for the following startup.sh script:

[root@localhost bin]# ./startup.sh

```
[root@localhost bin]# ./startup.sh
Using CATALINA_BASE:   /opt/apache-tomcat2
Using CATALINA_HOME:   /opt/apache-tomcat2
Using CATALINA_TMPDIR: /opt/apache-tomcat2/temp
Using JRE_HOME:        /opt/jdk1.6.0_24
Using CLASSPATH:       /opt/apache-tomcat2/bin/bootstrap.jar:/opt/apache-tomcat2
/bin/tomcat-juli.jar
[root@localhost bin]#
```

Configuration of instance 2

- Check the Tomcat instance process.
- The following screenshot shows the output for the ps command:

[root@localhost bin]# ps -ef |grep java

```
[root@localhost bin]# ps -ef |grep java
root      11746   1  0 11:49 25.09 /opt/tomcat-8.0_24/bin/java -Djava.awt.headless=true -Djava.logging.config=/opt/apache-tomcat-8.0_24/logging.properties -Dfile.encoding=UTF-8 -Xms512m -Xmx512m -XX:MaxPermSize=512m -Dorg.apache.resolver.mvnrepository.url=http://www.us.apache.org/maven2 -Dmaven.repo.local=/opt/apache-tomcat-8.0_24/maven -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=9999 -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false -Djava.awt.headless=true -Djava.awt.dnsCacheDir=/opt/apache-tomcat-8.0_24/ -Djava.awt.printerjob=SunPrinterJob -Djava.awt.startup=true -Djava.awt.x11FontPath=/opt/apache-tomcat-8.0_24/lib/jni -Djava.awt.headless=true -Djava.awt.printerjob=SunPrinterJob -Djava.awt.x11FontPath=/opt/apache-tomcat-8.0_24/lib/jni -Djava.awt.startup=true
root      12149   1  0 11:49 25.09 /opt/tomcat-8.0_24/bin/java -Djava.awt.headless=true -Djava.logging.config=/opt/apache-tomcat-8.0_24/logging.properties -Dfile.encoding=UTF-8 -Xms512m -Xmx512m -XX:MaxPermSize=512m -Dorg.apache.resolver.mvnrepository.url=http://www.us.apache.org/maven2 -Dmaven.repo.local=/opt/apache-tomcat-8.0_24/maven -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=9999 -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false -Djava.awt.headless=true -Djava.awt.printerjob=SunPrinterJob -Djava.awt.x11FontPath=/opt/apache-tomcat-8.0_24/lib/jni -Djava.awt.startup=true -Djava.awt.x11FontPath=/opt/apache-tomcat-8.0_24/lib/jni -Djava.awt.startup=true
```

Configuration of instance 2

Now, check catalina.out for both the nodes.

- The logs for node 1 are similar to the following:

[Refer to the file 9_1.txt](#)

- The logs for node 2 are similar to the following:

[Refer to the file 9_2.txt](#)

Apache web server configuration for vertical clustering

- We have to create a new file called mod_jk.conf in the conf directory of APACHE_HOME/conf using the following commands:

```
[root@localhost apache-2.0]# cd /opt/apache-2.2.19/conf  
vi mod-jk.conf
```

- The contents of mod_jk include the following lines of code:

```
LoadModulejk_module modules/mod_jk.so
```

```
JkWorkersFile conf/workers.properties
```

```
JkLogFile logs/mod_jk.log
```

```
JkLogLevel info
```

```
JkMount /sample/* loadbalancer
```

```
JkMount /* loadbalancer
```

Apache web server configuration for vertical clustering

- Create a new file named as workers.properties in conf using the following command:

```
[root@localhost conf]# vi workers.properties
```

- worker.list lists all the nodes in Tomcat through which Apache communicates using the AJP protocol.
- In our example, it has two nodes, as shown in the following line of code:

```
worker.list=tomcatnode1, tomcatnode2, loadbalancer
```

Apache web server configuration for vertical clustering

- Define the worker.list for the entire nodes in the cluster:

worker.tomcatnode1.port=8009

worker.tomcatnode1.host=localhost

worker.tomcatnode1.type=ajp13

worker.tomcatnode1.lbfactor=1

Apache web server configuration for vertical clustering

- The previous lines of code define the tomcatnode1 properties.
- The highlighted code shows the AJP port and hostname of tomcatnode1, which is essential for vertical clustering:

worker.tomcatnode2.port=8010

worker.tomcatnode2.host=localhost

worker.tomcatnode2.type=ajp13

worker.tomcatnode2.lbfactor=1

Apache web server configuration for vertical clustering

- The previous lines of code define the tomcatnode2 properties.
- The highlighted code shows the AJP port and hostname of tomcatnode2.
- This is essential for vertical clustering.

worker.loadbalancer.type=lb

worker.loadbalancer.balanced_workers=tomcatnode1,
tomcatnode2

worker.loadbalancer.sticky_session=1

Apache web server configuration for vertical clustering

- The last step is to include the mod_jk.conf in the main configuration file of httpd, that is httpd.conf and reload the Apache services:

```
[root@localhostconf]# vi httpd.conf
```

Monitoring of Tomcat clustering

Once the cluster is up and working, the next stage is to set up the monitoring of the clustering. This can be done in the following ways:

- Various monitoring tools
- Scripts
- Manual

Monitoring of Tomcat clustering

- Check the Tomcat process using the following command:

```
root@localhost bin]# ps -ef |grep java
```

- Check the logs to verify the connectivity of the cluster.
- Verify the URL for both cluster members.

Summary



- In this lesson, we have discussed the clustering of Tomcat 8 and its implementation techniques.
- We have discussed clustering architecture, horizontal and vertical clusters and their benefits, the implementation of horizontal and vertical clustering on Tomcat 8, and the verification of clusters.

10. Tomcat Upgrade

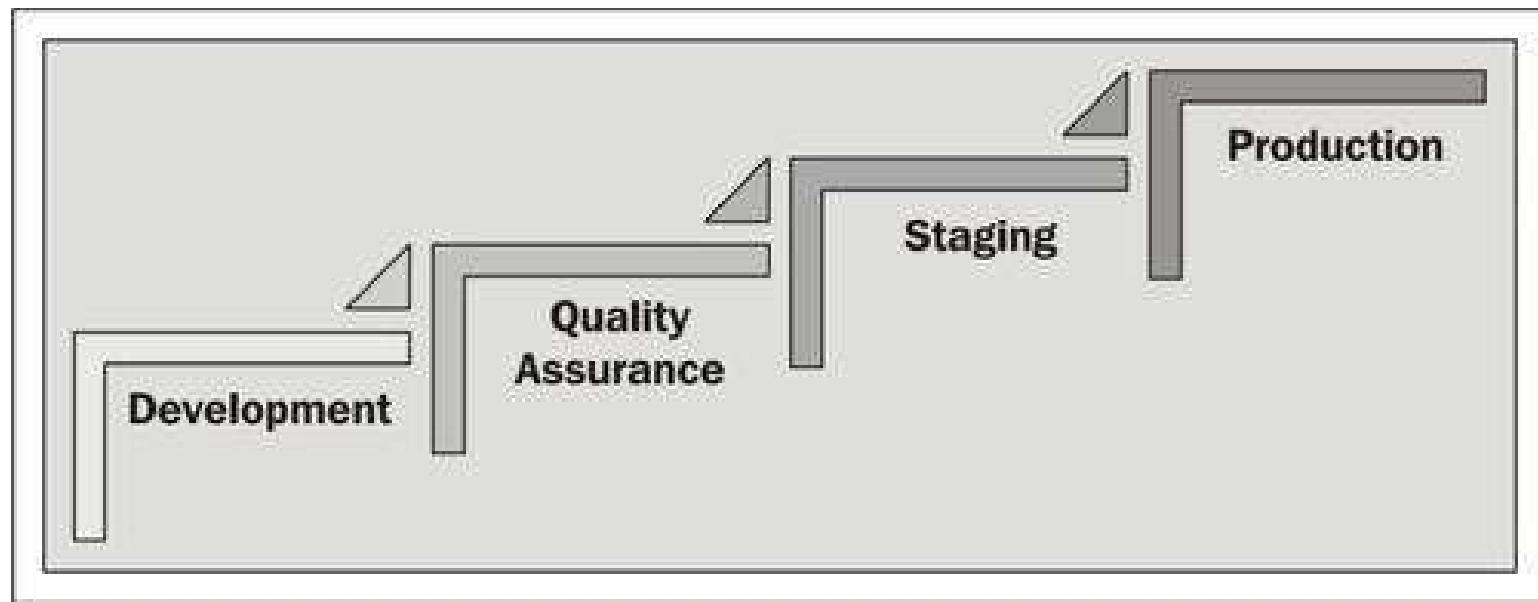


Tomcat Upgrade

In this lesson, we will discuss:

- The life cycle of the upgrade process
- Best practices followed by the IT industry
- How to upgrade Tomcat 6 to Tomcat 8

Different types of environment



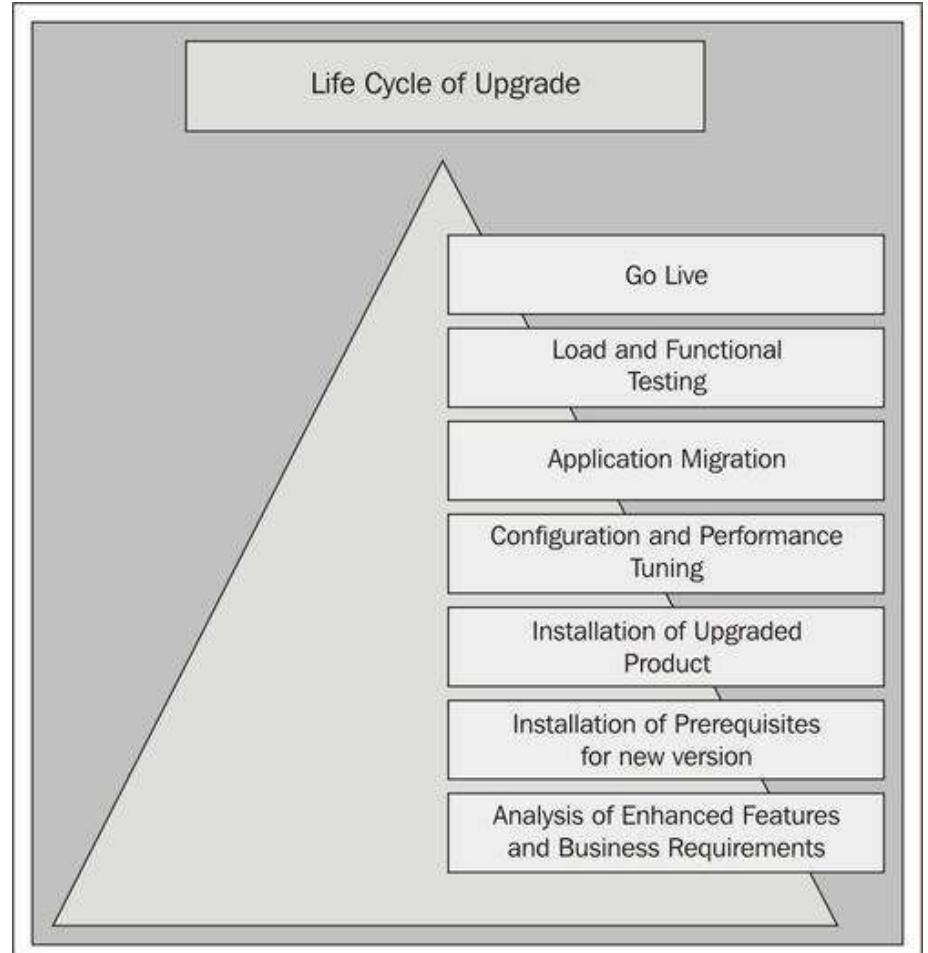
Development environment

- It can be defined as the combination of software and hardware, which is required for a team to build the code and deploy it.
- In simple words, it is a complete package required to build the code and deploy it.

Quality Assurance environment

- This environment is mainly used for the integration of the development module and followed by the functional testing undertaken by the Quality Assurance team.
- If the QA team finds any issues with the functionality of the application, they notify the developers to resolve the issue.

Life cycle of the upgrade



Tomcat upgrade from 6 to 7

Servlet 3.0

- Asynchronous support
- Dynamic configuration
- Extended Servlet API
- Simpler, faster, more developer-friendly
- Simplified embedding
- Improved logging

Tomcat upgrade from 6 to 7

System improvement

- No more memory leaks
- Security improvement

Prerequisites for Tomcat 8

- By default, Tomcat 6 runs on JDK 1.5 and Tomcat 8 requires JDK 1.6, so the major prerequisite for the Tomcat 8 upgrade is the installation of JDK 1.6.
- In Chapter 1, Installation of Tomcat 8, we have discussed the detailed steps of the Java installation.
- Hence, we will move on to the next installation step.

Installation of Tomcat 8 for the upgrade

- Download the latest stable version from the Tomcat official site, <http://tomcat.apache.org/download-70.cgi>.
- Once the download is complete, save it in the /opt location.
- Unzip the Tomcat 8 source, that is apache-tomcat-7.0.12.zip, using the following command:
`[root@localhost opt]# unzip apache-tomcat-7.0.12.zip`

Installation of Tomcat 8 for the upgrade

- After you unzip apache-tomcat-7.0.12.zip, it will create a folder named apache-tomcat-7.0.12 in the opt directory.
- Go to the bin directory of apache-tomcat-7.0.12 using the following command:

```
[root@localhost opt]# cd apache-tomcat-7.0.12/bin/
```

Installation of Tomcat 8 for the upgrade

- Run the following command. If you miss executing the following command, then Tomcat will not start at the time of starting the services.
- The reason is that, the package comes with read/write permissions but no execute permissions are given to the package. We have to manually update the permissions.

```
[root@localhost bin]# chmod 0755 *.sh
```

```
[root@localhost bin]# pwd
```

```
/opt/apache-tomcat-7.0.12/bin
```

Configuration of Tomcat 8

- You must be pondering as to whether we are installing or upgrading Tomcat.
- After reading this section, you will understand the actual upgrade.
- We will discuss the various configurations needed to be done with reference to Tomcat 8.
- It should also perform the same functionality as Tomcat 6, with the integration of new features.

JVM configuration

- How many applications are currently running on Tomcat 6
- The number of concurrent users
- The current configuration
- Upgrade to 64 bit from 32 bit

JVM configuration

```
using thread-local object allocation.  
Mark Sweep Compact GC  
  
Heap Configuration:  
MinHeapFreeRatio = 40  
MaxHeapFreeRatio = 70  
MaxHeapSize      = 268435456 (256.0MB)  
NewSize          = 1048576 (1.0MB)  
MaxNewSize       = 4294901760 (4093.9375MB)  
OldSize          = 5196304 (4.0MB)  
NewRatio         = 2  
SurvivorRatio   = 8  
PermSize         = 12582912 (12.0MB)  
MaxPermSize      = 67108564 (64.0MB)
```

JVM configuration

```
Heap Configuration:  
  MinHeapFreeRatio = 40  
  MaxHeapFreeRatio = 70  
  MaxHeapSize      = 134217728 (128.0MB)  
  NewSize          = 1048576 (1.0MB)  
  MaxNewSize       = 4294901760 (4095.9375MB)  
  OldSize          = 4194304 (4.0MB)  
  NewRatio         = 2  
  SurvivorRatio    = 8  
  PermSize         = 16777216 (16.0MB)  
  MaxPermSize      = 67108864 (64.0MB)
```

JVM configuration

- Let's take an example of the Tomcat 6 configuration and implement it with reference to Tomcat 8, as shown in the following lines of code:

```
JAVA_OPTS="-Xms128m -Xmx512m -  
XX:MaxPermSize=256m -  
Dsun.rmi.dgc.client.gcInterval=3600000 -  
Dsun.rmi.dgc.server.gcInterval=3600000"
```

JVM configuration

- Now, if you want to do the same configuration with Tomcat 8, then you have to increase the configuration based on the availability of the resource and application requirement as shown in the following lines of code:

```
JAVA_OPTS="-Xms1024m Xmx1024m -  
XX:MaxPermSize=256m  
-Dsun.rmi.dgc.client.gcInterval=3600000  
-Dsun.rmi.dgc.server.gcInterval=3600000"
```

Database connection settings

- Tomcat 6:

```
<Resource name="jdbc/myoracle" auth="Container"  
type="javax.sql.DataSource"  
driverClassName="oracle.jdbc.OracleDriver"  
"url="jdbc:oracle:thin:@192.168.0.1:1521:mysid"username="scott" password="tiger" maxActive="20" maxIdle="10"  
maxWait="-1"/>
```

Database connection settings

- Tomcat 8:

```
<Resource name="jdbc/myoracle" auth="Container"  
type="javax.sql.DataSource"  
driverClassName="oracle.jdbc. OracleDriver"  
url="jdbc:oracle:thin:@192.168.2.1:1521:mysid"  
username="scott" password="tiger" maxActive="50"  
maxIdle="10" maxWait="-1"/>
```

Application migration

Following are the steps for application migration:

- Application recompilation on the JDK supported for Tomcat 8.
- Upgrade the third-party JAR to the latest version.
- Deployment of the new application.
- Testing of the newly deployed application.

Alias configuration

- In the current environment, you might have configured many virtual hosts.
- But you cannot use the same URL for the application, as they point to the old environment.
- To resolve this issue, you have to configure the dummy URL in the new Tomcat 8 environment for testing purposes.
- This will help us perform the pre-go live task for the environment.

ITIL process implementation

Availability management

- **Reliability:** It's a process through which IT components are measured, based on the Statement of Work (SOW).
- **Maintainability:** It's a process through which we manage the entire system without any unplanned downtime.
- **Security:** This service is associated to data. It always refers to the confidentiality, integrity, and availability of that data. The term availability means the entire system service available to the environment.

Capacity management

| Process | Environment |
|------------------------|--|
| Application sizing | Stage environment |
| Capacity planning | Development environment or before the project begins |
| Performance management | Stage environment |

Service Transition

| Process | Environment (used) |
|--|------------------------------------|
| Transition planning and support | Post go live support in production |
| Change management | Before go live in production |
| Service asset and configuration management | Before go live in production |
| Release and deployment management | Before go live in production |
| Service validation and testing | Before go live in production |
| Change evaluation | Before go live in production |
| Knowledge management | Development/QA/stage/production |

Summary



- In this lesson, we have discussed the various strategies used in the upgrade of Tomcat 6 to Tomcat 8
- The various steps followed during the upgrade process such as the life cycle of the upgrade, the upgrade configuration of Tomcat 8, and the DataSource configuration.

11. Advanced Configuration for Apache Tomcat 8

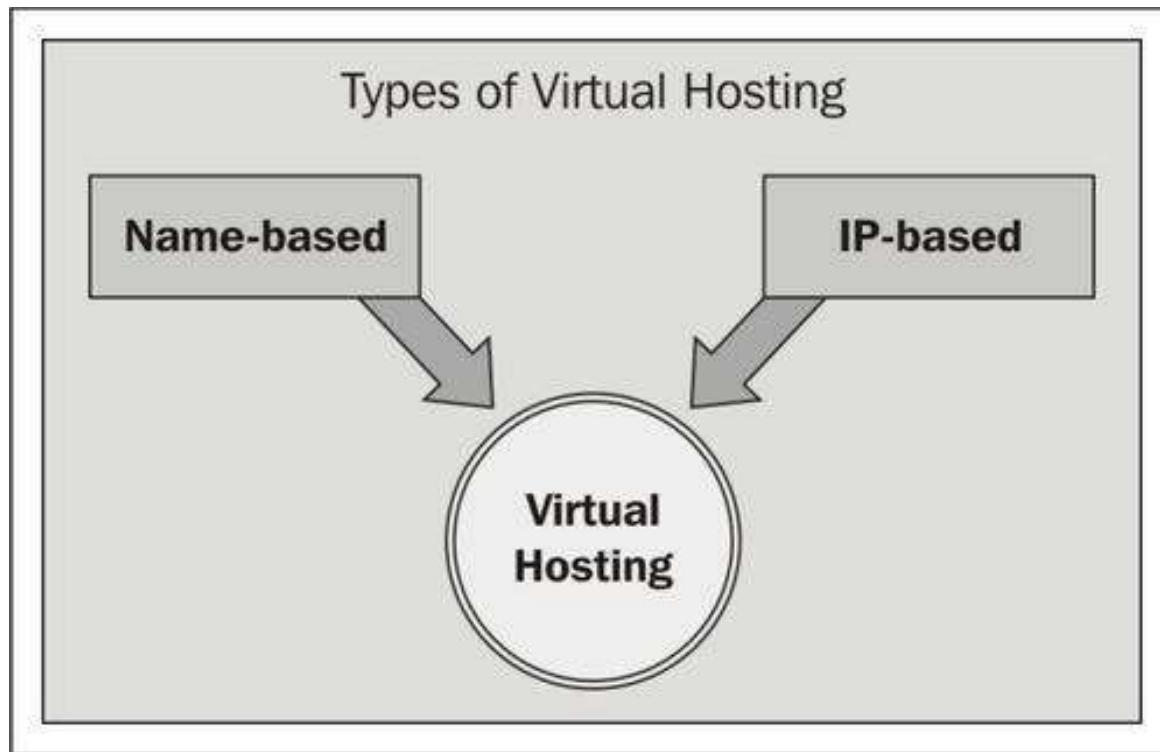


Advanced Configuration for Apache Tomcat 8

In this lesson, we will discuss the following topics:

- Virtual hosting
- Running multiple applications on a single Tomcat server
- Multiple Tomcat environments such as Development, QA, Stage, and Production
- Tuning cache
- Optimization of Tomcat

Virtual hosting



Name-based virtual hosting

Following are the advantages of name-based virtual hosting:

- Putting more than one website on a server using a single IP address
- Easy to configure
- Shared SSL certificates

Name-based virtual hosting

| Alias | Resource record | Domain |
|--|-----------------|--------------------|
| webserver1.yxz.com | A | 168.0.1 |
| www.xyz.com | C | webserver1.yxz.com |
| www.xzy.com | C | webserver1.yxz.com |
| www.abc.com | C | webserver1.yxz.com |

IP-based virtual hosting

Following are the advantages of IP-based virtual hosting:

- Hosting more than one website on a single server using different network interface addresses (different IP address)
- Dedicated network interface
- Dedicated SSL certificates

IP-based virtual hosting

| Alias | Record | Domain |
|--------------------|--------|---------|
| webserver1.yxz.com | A | 168.0.1 |
| www.xyz.com | A | 168.0.2 |
| www.xzy.com | A | 168.0.3 |
| www.abc.com | A | 168.0.4 |

Virtual hosting in Tomcat 8

- Configure the domain names in the DNS server and reload the DNS services so that it can be replicated on the server.
- Following are the DNS records with the address and CNAME:

| Alias | Record | Domain |
|--------------------|--------|--------------------|
| webserver1.yxz.com | A | 168.0.1 |
| www.xyz.com | C | webserver1.yxz.com |
| www.xzy.com | C | webserver1.yxz.com |
| www.abc.com | C | webserver1.yxz.com |

Virtual hosting in Tomcat 8

- For implementing virtual hosting, you have to edit server.xml, which is present in TOMCAT_HOME/conf.
- The following entries need to be added for the virtual host, as shown in the following screenshot:

```
<Host name="www.xyz.com" appBase="../Webapps">
<Context path="" docBase="."/>
</Host>
```

Virtual hosting in Tomcat 8

```
</Host>
<!-- Setting for virtual hosting -->
<Host name="www.xyz.com" appBase="../Webapps">
    <Context path="" docBase=".."/>
</Host>

<Host name="www.xzy.com" appBase="../Webapps">
    <Context path="" docBase=".."/>
</Host>

<Host name="www.abc.com" appBase="../Webapps">
    <Context path="" docBase=".."/>
</Host>

</Engine>
```

Virtual hosting in Tomcat 8

```
127.0.0.1      localhost.localdomain localhost
::1            localhost6.localdomain6 localhost6
# Below are the entries for tomcat virtual host
192.168.0.1    webserver1.yxz.com
192.168.0.1    www.xyz.com
192.168.0.1    www.xzy.com
192.168.0.1    www.abc.com
```

- Save the configuration, followed by the recycle, and check the logs if any errors persist.
- Check the URLs
`www.xyz.com, www.xzy.com, www.abc.com` in the browser.

Hostname aliases

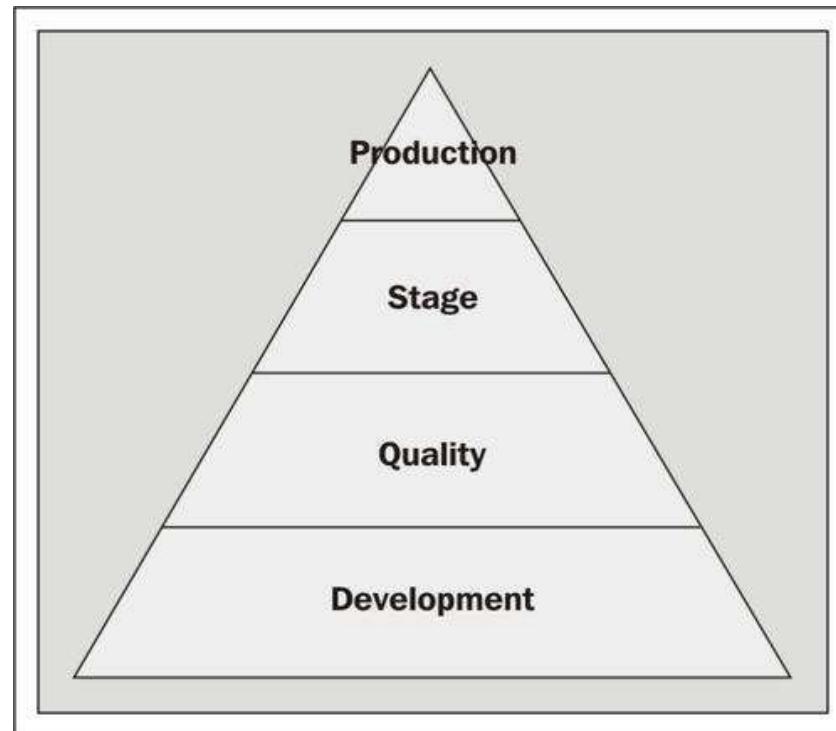
- There is one more important feature that comes with Tomcat 8 called Host name aliases.
- It's a very good feature that gives freedom to the administrator for multiple sites on the same network
- For example, if you have a website which needs to be accessed through a subdomain by different users, then host aliases are created.

Hostname aliases

- The following mentioned code describes how to set the alias for a particular site:

```
<Host name="www.xyz.com" appBase="../Webapps">
<Context path="" docBase="."/>
<Alias>tomcatalias.com</Alias>
</Host>
```

Multiple Tomcat environments— Development/QA/Stage/Production



| Task | Development | QA | Stage | Production |
|---------------------|-------------|-----|-------|------------|
| Auto deployment | Yes | Yes | No | No |
| Single machine | Yes | No | No | No |
| Clustering | No | Yes | Yes | Yes |
| Developer access | Yes | Yes | No | No |
| High-end machine | No | No | Yes | Yes |
| Change control | No | No | No | Yes |
| Performance testing | No | No | Yes | No |
| Functional testing | No | Yes | No | No |

Tuning cache

- When we are running multiple applications on Tomcat 8, it is always recommended to utilize the resource correctly.
- In order to do so, we need to optimize the tuning parameter.
- Every time the server receives a request, it consumes the amount of CPU and memory in the system. In order to resolve this problem, we generate cache on the server from the first request.

Tuning cache

- The following code shows the configuration for adding Expires and Cache-Control: max-age= headers to images, CSS, and JavaScript.
- This code is added in web.xml, which is present in TOMCAT_HOME/CONF.

Refer to the file 11_1.txt

Optimization of Tomcat 8

Running Tomcat 8 as a non privileged user

- It is not recommended way to run Tomcat as the root, because of security reasons and IT compliance policies.
- To resolve this issue, you have to run Tomcat as a non-privileged user.
- To implement this, you have to perform the following changes in the user permission.

Optimization of Tomcat 8

- Let us assume tomcatuser1 will run the Tomcat server as a non-privileged user.

```
# groupadd tomcatuser1  
# useradd -g tomcatuser1 -d /opt/apache-tomcat1  
# chown -R tomcatuser1:tomcatuser1 /opt/apache-tomcat1
```

Optimization of Tomcat 8

- Once you change the permission at the OS level, it's time to set tomcat to run as a service. Now, copy the startup scripts in /etc/init.d using the following commands:

```
cp /opt/apache-tomcat1/bin/startup.sh /etc/init.d/tomcat  
cd /etc/rc5.d  
sudo ln -s ../init.d/tomcat S71tomcat
```

Optimization of Tomcat 8

- You have now created tomcat as a service.
- But before running the service, you have to change the permission to make it executable, then run the service as tomcat using the following command:

```
chown 0755 /etc/init.d/tomcat
```

- Now we are ready to run tomcat as a service.

Summary



- In this lesson, we have discussed the advanced configuration of Tomcat 8 and optimization parameters, key points covered in the environment such as virtual hosting, features of Development/QA/Stage/Production, Tomcat as a service, and running Tomcat as a non privileged user.
- With this lesson, we have completed the journey of Tomcat 8