

Apache Tomcat 8.5

Install and Configure on CentOS 7

***** **Table of Content** *****

1	Information	3
2	Install.....	4
3	Validate	6
3.1	Startup Server	6
3.2	Shutdown Server	6
3.3	Apache Tomcat URL	7
3.4	Apache Tomcat Example URL	7
3.5	Apache Tomcat Manage URL.....	9
3.5.1	Set Manager User.....	9
3.5.2	Update Manager Webapp Viewable Location	10
3.5.3	Manager URL	10
4	Change Port.....	12
4.1	Change HTTP Port	12
4.2	Change Shutdown Port.....	12
4.3	Change AJP Port	13
4.4	Validate	13
5	Deploy Webapp	15
5.1	Browser	15
5.2	FileSystem.....	16
6	LDAP	17
6.1	Setup (server.xml)	17
6.2	Manager Webapp	18
6.2.1	Create LDAP User/Group.....	18
6.2.2	URLTest.....	20
6.3	Example Webapp	21
6.3.1	Create LDAP User/Group.....	21
6.3.2	URL Test.....	22
7	Cluster	24
7.1	Setup Server.xml	24
7.2	Additional Cluster Member.....	28
7.3	Apache HTTP	30

1 Information

This is a document will help with the installation and configuration of Apache Tomcat 8.5. It will cover the following features...

- Install
- Startup/Shutdown
- Apache Tomcat URLs
- Apache manager security
- Change Ports
- LDAP configuration
- Cluster

Useful links:

- Apache Tomcat 8.5 Documentation
<http://tomcat.apache.org/tomcat-8.5-doc/index.html>

This document is not written or supported by Apache

Name	Date	Version	Description
Loc Dang	May 10, 2017	V1	Install and Configure Apache Tomcat 8.5 on CentOS 7

2 Install

1. Login to the Apache Tomcat file system either as root or a user with sudo access.
2. Run the following command to install Java

```
sudo yum install java-devel
```

3. Download Apache Tomcat from the following link. Download either the zip or tar.gz file

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

Example:

```
apache-tomcat-8.5.14.tar.gz
apache-tomcat-8.5.14.zip
```

4. Extract the downloaded file to where you would like Apache Tomcat to be located. Apache Tomcat comes in a self-contained directory. The directory can be renamed after extraction. Make sure you have permission to extract to the location.

Example: UNZIP

```
unzip apache-tomcat-8.5.14.zip -d /opt/apache
```

Example: TAR

```
cd /opt/apache ; tar -xvf <TOMCAT_MEDIA>/apache-tomcat-8.5.14.tar.gz
```

5. OPTIONAL: Rename the Tomcat extracted directory

```
mv <TC_HOME> <TC_HOME_NEW>
```

Example:

```
mv /opt/apache/apache-tomcat-8.5.14 /opt/apache/tomcat
```

NOTE: This will be <TC_HOME>

6. Run the following command to make all “.sh” file executable in the bin directory

```
chmod +x <TC_HOME>/bin/*.sh
```

Example:

```
chmod +x /opt/apache/tomcat/bin/*.sh
```

7. Open the server.xml file with an editor

```
<TC_HOME>/conf/server.xml
```

Example: `/opt/apache/tomcat/conf/server.xml`

8. In the **Engine** stanza, update the **defaultHost** attribute with the correct hostname

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

Example:

```
<Engine name="Catalina" defaultHost="tc85a.loc.com">
```

9. In the **Host** stanza, update the **name** attribute with the correct hostname

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

Example:

```
<Host name="tc85a.loc.com" appBase="webapps" unpackWARs="true"  
autoDeploy="true">
```

10. Save server.xml

11. Start the Apache Tomcat Server

```
<TC_HOME>/bin/startup.sh
```

Example: `/opt/apache/tomcat/bin/startup.sh`

12. Open a browser and set the URL to the Apache Tomcat. By default, Apache Tomcat uses port 8080. If this port is not available then in another section there will be instructions on how to change the port.

```
http://<HOSTNAME><PORT>
```

Example:

```
http://tc85a.loc.com:8080
```


NOTE: If you are not testing on your local Apache Tomcat Server there could be a firewall blocking the port. Run the following command to open the Apache Tomcat port for CentOS

```
sudo firewall-cmd --zone=public --add-port=<NEW_PORT>/tcp --permanent  
sudo firewall-cmd --reload
```


Example:

```
sudo firewall-cmd --zone=public --add-port=8080/tcp --permanent  
sudo firewall-cmd --reload
```

[Home](#) [Documentation](#) [Configuration](#) [Examples](#) [Wiki](#) [Mailing Lists](#) [Find Help](#)

Apache Tomcat/8.5.14 

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 Data Sources](#)

[Examples](#)

[Servlet Specifications](#)
[Tomcat Versions](#)

[Managing Tomcat](#) [Documentation](#) [Getting Help](#)

3 Validate

3.1 Startup Server

There are 2 ways to start the Apache Tomcat. Run one of the following commands to start the Apache Tomcat server

- Startup Option 1:

```
<TC_HOME>/bin/catalina.sh start
```

Example: `/opt/apache/tomcat/bin/catalina.sh start`

- Startup Option 2:

```
<TC_HOME>/bin/startup.sh
```

Example: `/opt/apache/tomcat/bin/startup.sh`

1. Run the following command to check the used ports for Apache Tomcat

```
netstat -tulpn | grep java
```

Example:

```
[tcadmin@tc85a conf]$ netstat -tulpn | grep java
tcp6      0      0  127.0.0.1:8005      :::*           LISTEN       43729/java
tcp6      0      0  :::8009             :::*           LISTEN       43729/java
tcp6      0      0  :::8080             :::*           LISTEN       43729/java
```

3.2 Shutdown Server

There are 2 ways to start the Apache Tomcat. Run one of the following commands to start the Apache Tomcat server

- Shutdown Option 1:

```
<TC_HOME>/bin/catalina.sh stop
```

Example: `/opt/apache/tomcat/bin/catalina.sh stop`

- Shutdown Option 2:

```
<TC_HOME>/bin/shutdown.sh
```

Example: `/opt/apache/tomcat/bin/shutdown.sh`

1. Run the following command to check the used ports for Apache Tomcat. No results should appear

```
netstat -tulpn | grep java
```

3.3 Apache Tomcat URL

1. Start the Tomcat Server
2. Open a browser and set the URL to the Apache Tomcat. By default, Apache Tomcat uses port 8080. If this port is not available then in another section there will be instructions on how to change the port.

<http://<HOSTNAME><PORT>>

Example:

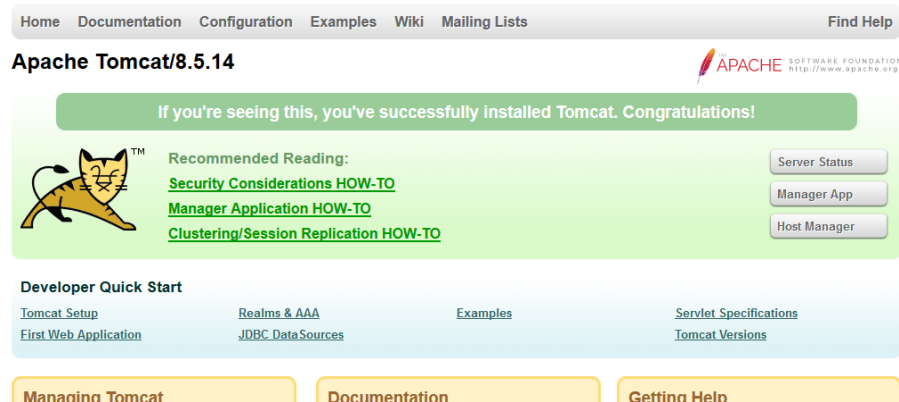
<http://tc85a.loc.com:8080>

NOTE: If you are not testing on your local Apache Tomcat Server there could be a firewall blocking the port. Run the following command to open the Apache Tomcat port for CentOS

```
sudo firewall-cmd --zone=public --add-port=<NEW_PORT>/tcp --permanent
sudo firewall-cmd --reload
```

Example:

```
sudo firewall-cmd --zone=public --add-port=8080/tcp --permanent
sudo firewall-cmd --reload
```



3.4 Apache Tomcat Example URL

1. Start the Tomcat Server
2. Open a browser and set the URL to the Apache Tomcat examples page. By default, Apache Tomcat uses port 8080. If this port is not available then in another section there will be instructions on how to change the port.

<http://<HOSTNAME><PORT>/examples>

Example:

<http://tc85a.loc.com:8080/examples>

NOTE: If you are not testing on your local Apache Tomcat Server there could be a firewall blocking the port. Run the following command to open the Apache Tomcat port for CentOS

```
sudo firewall-cmd --zone=public --add-port=<NEW_PORT>/tcp --permanent
sudo firewall-cmd --reload
```

Example:

```
sudo firewall-cmd --zone=public --add-port=8080/tcp --permanent
sudo firewall-cmd --reload
```

Apache Tomcat Examples

- [Servlets examples](#)
- [JSP Examples](#)
- [WebSocket Examples](#)

3. Click each link and test the examples

○ Servlet examples




Servlet Examples with Code

This is a collection of examples which demonstrate some of the more frequently used parts of the Servlet API. Familiarity with the Java(tm) Programming Language is assumed.

These examples will only work when viewed via an http URL. They will not work if you are viewing these pages via a "file://..." URL. Please refer to the *README* file provide with this Tomcat release regarding how to configure and start the provided web server.

Wherever you see a form, enter some data and see how the servlet reacts. When playing with the Cookie and Session Examples, jump back to the Headers Example to see exactly what your browser is sending the server.

To navigate your way through the examples, the following icons will help:

-  Execute the example
-  Look at the source code for the example
-  Return to this screen

Tip: To see the cookie interactions with your browser, try turning on the "notify when setting a cookie" option in your browser preferences. This will let you see when a session is created and give some feedback when looking at the cookie demo.

Hello World	 Execute	 Source
Request Info	 Execute	 Source
Request Headers	 Execute	 Source




○ JSP Examples

JSP Samples

This is a collection of samples demonstrating the usage of different parts of the Java Server Pages (JSP) specification. Both JSP 2.0 and JSP 1.2 examples are presented below.

These examples will only work when these pages are being served by a servlet engine; of course, we recommend [Tomcat](#). They will not work if you are viewing these pages via a "file://..." URL.

To navigate your way through the examples, the following icons will help:

-  Execute the example
-  Look at the source code for the example
-  Return to this screen

Tip: For session scoped beans to work, the cookies must be enabled. This can be done using browser options.

JSP 2.0 Examples

Expression Language	 Execute	 Source
Basic Arithmetic	 Execute	 Source
Basic Comparisons	 Execute	 Source
Implicit Objects	 Execute	 Source

- WebSocket Examples

Apache Tomcat WebSocket Examples

- [Echo example](#)
- [Chat example](#)
- [Multiplayer snake example](#)
- [Multiplayer drawboard example](#)

3.5 Apache Tomcat Manage URL

3.5.1 Set Manager User

1. Login to the Apache Tomcat file system
2. Open the **tomcat-users.xml** file with an editor

```
<TC_HOME>/conf/tomcat-users.xml
```

Example: `/opt/apache/tomcat/conf/tomcat-users.xml`

3. Add the following line to the bottom of the file in between `<tomcat-users...>...</tomcat-users>`. This will give a user the ability to login to the manage webapp.

```
<user username="<TC_ADMIN>" password="<TC_ADMIN_PWD>" roles="admin-  
gui,manager-gui"/>
```

Example:

```
<user username="tcadmin" password="passw0rd" roles="admin-gui,manager-  
gui"/>
```

Results:

```
...  
<!--  
  <role rolename="tomcat"/>  
  <role rolename="role1"/>  
  <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>  
  <user username="both" password="<must-be-changed>"  
roles="tomcat,role1"/>  
  <user username="role1" password="<must-be-changed>" roles="role1"/>  
-->  
<user username="tcadmin" password="passw0rd" roles="admin-gui,manager-  
gui"/>  
</tomcat-users>
```

4. Save **tomcat-users.xml**

3.5.2 Update Manager Webapp Viewable Location

By default, the manager webapp can only be viewed using 127.0.0.1 and localhost. This update is optional but will give the ability to view the manager webapp using other systems.

1. Login to the Apache Tomcat file system
2. Open the **context.xml** file with an editor

```
<TC_HOME>/webapps/manager/META-INF/context.xml
```

Example: `/opt/apache/tomcat/webapps/manager/META-INF/context.xml`

3. Update the allow variable with the updated information

Before: Only 127.0.0.1 and localhost

```
<Context antiResourceLocking="false" privileged="true" >
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\.\d+\.\d+\.\d+|:::1|0:0:0:0:0:0:0:1" />
</Context>
```

After: Any IP address can view the manager webapp

```
<Context antiResourceLocking="false" privileged="true" >
<!--
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\.\d+\.\d+\.\d+|:::1|0:0:0:0:0:0:0:1" />
-->
</Context>
```

After: Any IP address that starts with 192.168.1.*, localhost and 127.0.0.1

```
<Context antiResourceLocking="false" privileged="true" >
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\.\d+\.\d+\.\d+|:::1|0:0:0:0:0:0:0:1|192.168.1.*" />
</Context>
```

After: IP address 192.168.1.133, localhost and 127.0.0.1 only

```
<Context antiResourceLocking="false" privileged="true" >
  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\.\d+\.\d+\.\d+|:::1|0:0:0:0:0:0:0:1|192.168.1.133"
  />
</Context>
```

NOTE: Each IP address is separated by a ‘|’

3.5.3 Manager URL

1. Start the Tomcat Server

```
<TC_HOME>/bin/catalina.sh start
```

Example: `/opt/apache/tomcat/bin/catalina.sh start`

2. Open a browser and set the URL to the Apache Tomcat manager page. By default, Apache Tomcat uses port 8080. If this port is not available then in another section there will be instructions on how to change the port.

```
http://<HOSTNAME><PORT>/manager
```

Example:

```
http://tc85a.loc.com:8080/manager
```

NOTE: If you are not testing on your local Apache Tomcat server or the context.xml was not updated then there could be a firewall blocking the port. Run the following command to open the Apache Tomcat port for CentOS

```
sudo firewall-cmd --zone=public --add-port=<NEW_PORT>/tcp --permanent
sudo firewall-cmd --reload
```

Example:

```
sudo firewall-cmd --zone=public --add-port=8080/tcp --permanent
sudo firewall-cmd --reload
```

3. Login with the user set in the tomcat-users.xml file



Tomcat Web Application Manager

Message:	OK
----------	----

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> <input type="button" value="Expire sessions"/> with idle ≥ <input type="text" value="30"/> minutes
					Start <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/>

4. Verify the page appears

4 Change Port

If you have 2 or more Apache Tomcat server on the same system, all ports listed in this section will need to be update for the additional Apache Tomcat server.

4.1 Change HTTP Port

1. Login to the Apache Tomcat file system
2. Open the server.xml file with an editor

```
<TC_HOME>/conf/server.xml
```

Example: `/opt/apache/tomcat/conf/server.xml`

3. Update the port referencing **HTTP/1.1** to a new port. In the example, 9090 was used.

Before:

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

After:

```
<Connector port="9090" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
```

4. Save server.xml

4.2 Change Shutdown Port

1. Shutdown the Apache Tomcat server before changing the SHUTDOWN port
2. Login to the Apache Tomcat file system
3. Open the server.xml file with an editor

```
<TC_HOME>/conf/server.xml
```

Example: `/opt/apache/tomcat/conf/server.xml`

4. Update the port referencing **SHUTDOWN** to a new port. In the example, 9005 was used.

Before:

```
<Server port="8005" shutdown="SHUTDOWN">
  <Listener
    className="org.apache.catalina.startup.VersionLoggerListener" />
```

After:

```
<Server port="9005" shutdown="SHUTDOWN">
  <Listener
className="org.apache.catalina.startup.VersionLoggerListener" />
```

5. Save server.xml

4.3 Change AJP Port

1. Login to the Apache Tomcat file system
2. Open the server.xml file with an editor

```
<TC_HOME>/conf/server.xml
```

Example: `/opt/apache/tomcat/conf/server.xml`

3. Update the port referencing **AJP/1.3** to a new port. In the example, 9009 was used.

Before:

```
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

After:

```
<Connector port="9009" protocol="AJP/1.3" redirectPort="8443" />
```

4. Save server.xml

4.4 Validate

1. Start the Apache Tomcat server
2. Login to the Apache Tomcat server file system
3. Run the following command to check the used ports for Apache Tomcat

```
netstat -tulpn | grep java
```

Example:

```
[tcadmin@tc85a conf]$ netstat -tulpn | grep java
tcp6      0      0  127.0.0.1:9005      :::*           LISTEN        43729/java
tcp6      0      0  :::9009             :::*           LISTEN        43729/java
tcp6      0      0  :::9090             :::*           LISTEN        43729/java
```

4. Verify the new ports are being used
5. Open a browser and set the URL to the Apache Tomcat using the new HTTP/1.1 port. In the example 9090 was used.

```
http://<HOSTNAME><PORT>
```

Example:

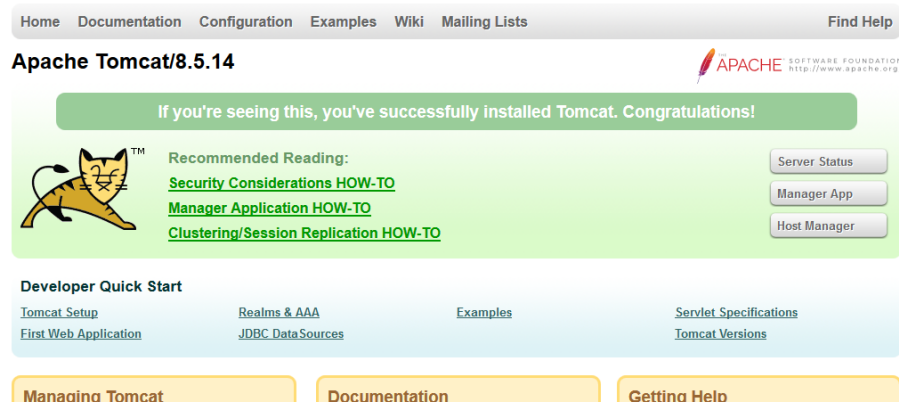
```
http://tc85a.loc.com:9090
```

NOTE: If you are not testing on your local Apache Tomcat Server there could be a firewall blocking the port. Run the following command to open the Apache Tomcat port for CentOS

```
sudo firewall-cmd --zone=public --add-port=<NEW_PORT>/tcp --permanent  
sudo firewall-cmd --reload
```

Example:

```
sudo firewall-cmd --zone=public --add-port=9090/tcp --permanent  
sudo firewall-cmd --reload
```



6. Verify the Apache Tomcat page appears

5 Deploy Webapp

5.1 Browser



Tomcat Web Application Manager

Message:	OK
----------	----

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/> <input type="button" value="Expire sessions"/> with idle ≥ <input type="text" value="30"/> minutes
					Start <input type="button" value="Stop"/> <input type="button" value="Reload"/> <input type="button" value="Undeploy"/>

1. Open a browser on a system that has access to the Manager webapp of Apache Tomcat. By default, only the local Apache Tomcat has access.

<http://<HOSTNAME>:<PORT>/manager>

Example: <http://tc85a.loc.com:8080/manager>

Deploy	
Deploy directory or WAR file located on server	
Context Path (required):	<input type="text"/>
XML Configuration file URL:	<input type="text"/>
WAR or Directory URL:	<input type="text"/>
<input type="button" value="Deploy"/>	
WAR file to deploy	
Select WAR file to upload	<input type="button" value="Browse..."/> No file selected.
<input type="button" value="Deploy"/>	

2. Scroll down the page to the Deploy section
3. If the war file is located on your local desktop, click Browse and browse to the war file.
4. Click Deploy



Tomcat Web Application Manager

Message:	OK
----------	----

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy

5. After the deployment is complete, verify the message for an OK message

5.2 FileSystem

1. Login to the Apache Tomcat file system
2. Copy the custom webapp to the Apache Tomcat webapps directory
`<TC_HOME>/webapps`

Example: `/opt/apache/tomcat/webapps`



Tomcat Web Application Manager

Message:	OK
----------	----

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/LoginModule	None specified	LoginModule	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy

3. After some time the webapp will be installed. Check the Tomcat Manager URL to verify that it has been installed. In the example, a webapp called LoginModule was deployed.

6 LDAP

This section will help create a new realm to configure to a Microsoft Active Directory user repository.

6.1 Setup (server.xml)

1. Login to the Apache Tomcat file system
2. Open the server.xml file with an editor

```
<TC_HOME>/conf/server.xml
```

Example: `/opt/apache/tomcat/conf/server.xml`

3. Add the following lines to the bottom of the <Engine> stanza

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
  connectionName="<BIND_USER>"
  connectionPassword="<BIND_PASSWORD>"
  connectionURL="ldap://<LDAP_HOSTNAME>:<LDAP_PORT>"
  userBase="<USER_BASE>"
  userSearch="( (<LOGIN_ATTRIBUTE>={0}) )"
  userRoleName="memberOf"
  roleBase="<GROUP_BASE>"
  roleName="cn"
  roleSearch="(member={0})"
/>
```

Example: Login with **sAMAccountName** attribute

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
  connectionName="CN=ldapbind,CN=Users,DC=loc,DC=com"
  connectionPassword="passwd"
  connectionURL="ldap://my2008ad.loc.com:389"
  userBase="cn=users,dc=loc,dc=com"
  userSearch="(sAMAccountName={0})"
  userRoleName="memberOf"
  roleBase="ou=groups,dc=loc,dc=com"
  roleName="cn"
  roleSearch="(member={0})"
/>

</Engine>
```

Example: Login with **mail** attribute

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
  connectionName="CN=ldapbind,CN=Users,DC=loc,DC=com"
  connectionPassword="passwd"
  connectionURL="ldap://my2008ad.loc.com:389"
  userBase="cn=users,dc=loc,dc=com"
  userSearch="(mail={0})"
  userRoleName="memberOf"
  roleBase="ou=groups,dc=loc,dc=com"
```

```

        roleName="cn"
        roleSearch="(member={0})"
    />

</Engine>

```

4. Save server.xml
5. Start/restart the Apache Tomcat Server
6. Verify there are no LDAP exception in the catalina.log
`<TC_HOME>/logs/catalina.log`

Example: `/opt/apache/tomcat/logs/catalina.log`

6.2 Manager Webapp

6.2.1 Create LDAP User/Group

The users and groups created in the LDAP will be used with the Apache Tomcat manager webapp.

1. Create the following groups for the manager webapp in the external LDAP repository
 - admin-gui
 - manager-gui
 - manager-script
 - manager-jmx
 - manager-status

NOTE: If you don't want to create new groups and want to use existing groups, update the web.xml file of the manager webapp and replace the existing groups with the update groups. Update the groups in green below and then save.

`<TC_HOME>/webapp/manager/WEB-INF/web.xml`

```

<security-constraint>
  <web-resource-collection>
    <web-resource-name>HTML Manager interface (for humans)</web-
resource-name>
    <url-pattern>/html/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>manager-gui</role-name>
  </auth-constraint>
</security-constraint>
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Text Manager interface (for scripts)</web-
resource-name>
    <url-pattern>/text/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>manager-script</role-name>
  </auth-constraint>
</security-constraint>

```

```

    </auth-constraint>
</security-constraint>
<security-constraint>
  <web-resource-collection>
    <web-resource-name>JMX Proxy interface</web-resource-name>
    <url-pattern>/jmxproxy/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>manager-jmx</role-name>
  </auth-constraint>
</security-constraint>
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Status interface</web-resource-name>
    <url-pattern>/status/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>manager-gui</role-name>
    <role-name>manager-script</role-name>
    <role-name>manager-jmx</role-name>
    <role-name>manager-status</role-name>
  </auth-constraint>
</security-constraint>

```

...

```

<security-role>
  <description>
    The role that is required to access the HTML Manager pages
  </description>
  <role-name>manager-gui</role-name>
</security-role>
<security-role>
  <description>
    The role that is required to access the text Manager pages
  </description>
  <role-name>manager-script</role-name>
</security-role>
<security-role>
  <description>
    The role that is required to access the HTML JMX Proxy
  </description>
  <role-name>manager-jmx</role-name>
</security-role>
<security-role>
  <description>
    The role that is required to access to the Manager Status pages
  </description>
  <role-name>manager-status</role-name>
</security-role>

```

manager-gui = _____

manager-script = _____

manager-jmx = _____

manager-status = _____

2. Create a user and add it to the manager-gui group or its replacement

ADMIN_USER = _____

Example: `tcadmin`

6.2.2 URLTest

1. Open a browser and set the URL to the manager webapp
<http://<HOSTNAME>:<PORT>/manager>

Example: <http://tc85a.loc.com:8080/manager>

2. Login with the user that is part of the **manager-gui** group.



Tomcat Web Application Manager

Message:

OK

Manager

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

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	<div>Start Stop Reload Undeploy</div> <div>Expire sessions with idle ≥ 30 minutes</div>
					<div>Start Stop Reload Undeploy</div>

3. Verify that you are logged in
4. If the user does not exist, the login prompt will appear again

403 Access Denied

You are not authorized to view this page.

By default the Manager is only accessible from a browser running on the same machine as Tomcat. If you wish to modify this restriction, you'll need to edit the Manager's `context.xml` file.

If you have already configured the Manager application to allow access and you have used your browsers back button, used a saved book-mark or similar then you may have triggered the cross-site request forgery (CSRF) protection that has been enabled for the HTML interface of the Manager application. You will need to reset this protection by returning to the [main Manager page](#). Once you return to this page, you will be able to continue using the Manager application's HTML interface normally. If you continue to see this access denied message, check that you have the necessary permissions to access this application.

If you have not changed any configuration files, please examine the file `conf/tomcat-users.xml` in your installation. That file must contain the credentials to let you use this webapp.

For example, to add the `manager-gui` role to a user named `tomcat` with a password of `s3cret`, add the following to the config file listed above.

```
<role rolename="manager-gui"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
```

Note that for Tomcat 7 onwards, the roles required to use the manager application were changed from the single `manager` role to the following four roles. You will need to assign the role(s) required for the functionality you wish to access

5. If the user does not have access to the manager page, they will receive as **403 Access Denied**

6.3 Example Webapp

6.3.1 Create LDAP User/Group

The users and groups created in the LDAP will be used by the Apache Tomcat examples webapp.

1. Create the following groups for the examples webapp in the external LDAP repository
 - o tomcat
 - o role1

NOTE: If you don't want to create new groups and want to use existing groups, update the web.xml file of the examples webapp and replace the existing groups with the update groups. Update the groups in green below and then save.

```
<TC_HOME>/webapp/examples/WEB-INF/web.xml
```

```
<auth-constraint>
  <!-- Anyone with one of the listed roles may access this area
-->
  <role-name>tomcat</role-name>
  <role-name>role1</role-name>
</auth-constraint>
```

...

```
<security-role>
  <role-name>role1</role-name>
</security-role>
<security-role>
  <role-name>tomcat</role-name>
</security-role>
```

tomcat = _____
role1 = _____

2. Create a user and add it to the tomcat group

TC_USER = _____

Example: user1

3. Create a user and add it to the role1 group

ROLE_USER = _____

Example: user2

4. Create a user and add it to both the tomcat and role one group

TR_USER = _____

Example: user3

6.3.2 URL Test

1. Open a browser and set the URL to the Apache Tomcat Example's **FORM Authentication** page

<http://<HOSTNAME>:<PORT>/examples/jsp/security/protected/index.jsp>

Example:

<http://tc85a.loc.com:8080/examples/jsp/security/protected/index.jsp>

Username:

Password:

2. Login with the LDAP user that is part of the **tomcat** group or its replacement (tomcat-ad)

Example: user1

You are logged in as remote user **user1** in session CA0D89BEB00FC3862F78BAE586C6B643.jvm1

Your user principal name is **user1**

To check whether your user name has been granted a particular role, enter it here:

If you have configured this application for form-based authentication, you can log off by clicking [here](#). This should cause you to be returned to the login page after the redirect that is performed.

3. Enter the **tomcat** group or its replacement (tomcat-ad)
4. Click **Submit Query**

You are logged in as remote user **user1** in session CA0D89BEB00FC3862F78BAE586C6B643.jvm1

Your user principal name is **user1**

You have been granted role **tomcat-ad**

To check whether your user name has been granted a particular role, enter it here:

tomcat-ad

If you have configured this application for form-based authentication, you can log off by clicking [here](#). This should cause you to be returned to the login page after the redirect that is performed.

5. The user should be member of the group. The following message will appear

You have been granted role <GROUP>

Example: You have been granted role tomcat-ad

6. Enter the **role1** group or its replacement (role1-ad)
7. Click **Submit Query**

You are logged in as remote user **user1** in session **CA0D89BEB00FC3862F78BAE586C6B643.jvm1**

Your user principal name is **user1**

You have *not* been granted role **role1-ad**

To check whether your user name has been granted a particular role, enter it here:

If you have configured this application for form-based authentication, you can log off by clicking [here](#). This should cause you to be returned to the login page after the redirect that is performed.

8. The user is not a member of the group. The following message will appear. If the group does not exist, you will receive the same message.

You have *not* been granted role <GROUP>

Example: You have *not* been granted role **role1-ad**

7 Cluster

<http://tomcat.apache.org/tomcat-8.5-doc/cluster-howto.html>

7.1 Setup Server.xml

1. Login to the Apache Tomcat file system
2. Open the server.xml file with an editor

```
<TC_HOME>/conf/server.xml
```

Example: `/opt/apache/tomcat/conf/server.xml`

3. In the **Engine** stanza, add the attribute `jvmRoute` with a value for the cluster member

Before:

```
<Engine name="Catalina" defaultHost="<HOSTNAME>">
```

After:

```
<Engine name="Catalina" defaultHost="<HOSTNAME>" jvmRoute="<VALUE>">
```

Example:

```
<Engine name="Catalina" defaultHost="tc85a.loc.com" jvmRoute="jvm1">
```

4. Add the following to the bottom of the **Host** stanza

```
<Cluster
className="org.apache.catalina.ha.tcp.SimpleTcpCluster"
channelSendOptions="8">

  <Manager
className="org.apache.catalina.ha.session.DeltaManager"
expireSessionsOnShutdown="false"
notifyListenersOnReplication="true"/>

  <Channel
className="org.apache.catalina.tribes.group.GroupChannel">
    <Membership
className="org.apache.catalina.tribes.membership.McastService"
address="228.0.0.4"
port="45564"
frequency="500"
dropTime="3000"/>

    <Receiver
className="org.apache.catalina.tribes.transport.nio.NioReceiver"
address="auto"
port="4000"
autoBind="100"
selectorTimeout="5000"
maxThreads="6"/>
```



```

        <Sender
className="org.apache.catalina.tribes.transport.ReplicationTransmitter"
>
        <Transport
className="org.apache.catalina.tribes.transport.nio.PooledParallelSender"/>
        </Sender>
        <Interceptor
className="org.apache.catalina.tribes.group.interceptors.TcpFailureDetector"/>
        <Interceptor
className="org.apache.catalina.tribes.group.interceptors.MessageDispatchInterceptor"/>
        </Channel>

        <Valve
className="org.apache.catalina.ha.tcp.ReplicationValve"
        filter=""/>
        <Valve
className="org.apache.catalina.ha.session.JvmRouteBinderValve"/>

        <Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
        tempDir="/tmp/war-temp/"
        deployDir="/tmp/war-deploy/"
        watchDir="/tmp/war-listen/"
        watchEnabled="false"/>

        <ClusterListener
className="org.apache.catalina.ha.session.ClusterSessionListener"/>
        </Cluster>

```

5. In the **Receiver** stanza, update the **address** attribute to the IP or hostname of the system.

Before:

```

<Receiver
className="org.apache.catalina.tribes.transport.nio.NioReceiver"
        address="auto"
        port="4000"
        autoBind="100"
        selectorTimeout="5000"
        maxThreads="6"/>

```

After:

```

<Receiver
className="org.apache.catalina.tribes.transport.nio.NioReceiver"
        address="tc85a.loc.com"
        port="4000"
        autoBind="100"
        selectorTimeout="5000"
        maxThreads="6"/>

```

6. In the **Deployer** stanza, update the **tempDir**. The tempDir stores binary data when downloading a war from the cluster. You may specify an absolute pathname, or a pathname that is relative to the \$CATALINA_BASE directory. Place this directory in the Apache Tomcat home directory to make it portable when creating a new Apache Tomcat cluster member. Create the directory if it does not exist.

Before:

```
<Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
    tempDir="/tmp/war-temp/"
    deployDir="/tmp/war-deploy/"
    watchDir="/tmp/war-listen/"
    watchEnabled="false"/>
```

After:

```
<Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
    tempDir="<TC_HOME>/war-temp/"
    deployDir="/tmp/war-deploy/"
    watchDir="/tmp/war-listen/"
    watchEnabled="false"/>
```

Example:

```
<Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
    tempDir="/opt/apache/tomcat/war-temp/"
    deployDir="/tmp/war-deploy/"
    watchDir="/tmp/war-listen/"
    watchEnabled="false"/>
```

7. In the **Deployer** stanza, update the **deployDir**. This is the pathname of a directory where the web applications are deployed. Set the path name to the webapp directory of the Tomcat home directory

Before:

```
<Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
    tempDir="/opt/apache/tomcat/war-temp/"
    deployDir="/tmp/war-deploy/"
    watchDir="/tmp/war-listen/"
    watchEnabled="false"/>
```

After:

```
<Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
    tempDir="/opt/apache/tomcat/war-temp/"
    deployDir="<TC_HOME>/webapps"
    watchDir="/tmp/war-listen/"
    watchEnabled="false"/>
```

Example:

```

    <Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
    tempDir="/opt/apache/tomcat/war-temp/"
    deployDir="/opt/apache/tomcat/webapps/"
    watchDir="/tmp/war-listen/"
    watchEnabled="false"/>

```

8. In the **Deployer** stanza, update the **watchDir**. This is the pathname of a directory where changes(add/modify/remove) of web applications is being watched. You may specify an absolute pathname, or a pathname that is relative to the \$CATALINA_BASE directory. Place this directory in the Apache Tomcat home directory to make it portable when creating a new Apache Tomcat cluster member. Create the directory if it does not exist.

Note: if **watchEnabled** is false, this attribute will have no effect.

Before:

```

    <Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
    tempDir="/opt/apache/tomcat/war-temp/"
    deployDir="/opt/apache/tomcat/webapps/"
    watchDir="/tmp/war-listen/"
    watchEnabled="false"/>

```

After:

```

    <Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
    tempDir="/opt/apache/tomcat/war-temp/"
    deployDir="/opt/apache/tomcat/webapps/"
    watchDir="<TC_HOME>/war-listen/"
    watchEnabled="false"/>

```

Example::

```

    <Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
    tempDir="/opt/apache/tomcat/war-temp/"
    deployDir="/opt/apache/tomcat/webapp"
    watchDir="/opt/apache/tomcat/war-listen/"
    watchEnabled="false"/>

```

9. In the **Deployer** stanza, update the **watchEnabled** if needed. Set to true if you want to watch for changes of web applications. Only when this attribute set to true, you can trigger a deploy/undeploy of web applications. The flag's value defaults to false. In the example, watchEnabled was set to true.

Before:

```

    <Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
    tempDir="/opt/apache/tomcat/war-temp/"
    deployDir="/opt/apache/tomcat/webapp"
    watchDir="/opt/apache/tomcat/war-listen/"
    watchEnabled="false"/>

```

After:

```
<Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
tempDir="/opt/apache/tomcat/war-temp/"
deployDir="/opt/apache/tomcat/webapp"
watchDir="/opt/apache/tomcat/war-listen/"
watchEnabled="true"/>
```

10. Save server.xml

7.2 Additional Cluster Member

1. Login to the Apache Tomcat file system either as root or a user with sudo access.
2. Run the following command to install Java

```
sudo yum install java-devel
```

3. Copy the Apache Tomcat home directory from the primary server to the additional server

Example: `/opt/apache/tomcat`

4. Open the server.xml file with an editor

```
<TC_HOME>/conf/server.xml
```

Example: `/opt/apache/tomcat/conf/server.xml`

5. In the Engine stanza, update the defaultHost and jvmRoute.

Before:

```
<Engine name="Catalina" defaultHost="tc85a.loc.com" jvmRoute="jvm1">
```

After:

```
<Engine name="Catalina" defaultHost="tc85b.loc.com" jvmRoute="jvm2">
```

6. In the **Host** stanza, update the **name** attribute with the correct hostname

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

Example:

```
<Host name="tc85a.loc.com" appBase="webapps" unpackWARs="true"
autoDeploy="true">
```

7. In the **Receiver** stanza, update the **address** attribute to the IP or hostname of the system

Before:

```
<Receiver
className="org.apache.catalina.tribes.transport.nio.NioReceiver"
```

```
address="auto"
port="4000"
autoBind="100"
selectorTimeout="5000"
maxThreads="6"/>
```

After:

```
<Receiver
className="org.apache.catalina.tribes.transport.nio.NioReceiver"
address="tc85b.loc.com"
port="4000"
autoBind="100"
selectorTimeout="5000"
maxThreads="6"/>
```

8. In the **Deployer** stanza, update the **tempDir**. If the tempDir was not pointing to a directory in the Apache Tomcat home, create the directory on the additional node.

```
<Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
tempDir="<TC_HOME>/war-temp/"
deployDir="/tmp/war-deploy/"
watchDir="/tmp/war-listen/"
watchEnabled="false"/>
```

Example:

```
<Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
tempDir="/opt/apache/tomcat/war-temp/"
deployDir="/tmp/war-deploy/"
watchDir="/tmp/war-listen/"
watchEnabled="false"/>
```

9. In the **Deployer** stanza, update the **deployDir**. If the Apache Tomcat home directory is a different directory then the primary server, update the deployDir to the correct webapp directory.

```
<Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
tempDir="/opt/apache/tomcat/war-temp/"
deployDir="<TC_HOME>/webapps"
watchDir="/tmp/war-listen/"
watchEnabled="false"/>
```

Example:

```
<Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
tempDir="/opt/apache/tomcat/war-temp/"
deployDir="/opt/apache/tomcat/webapps"
watchDir="/tmp/war-listen/"
watchEnabled="false"/>
```

10. In the **Deployer** stanza, update the **watchDir**. If the watchDir was not pointing to a directory in the Apache Tomcat home, create the directory on the additional node.

```
<Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
tempDir="/opt/apache/tomcat/war-temp/"
deployDir="/opt/apache/tomcat/webapps"
watchDir="<TC_HOME>/war-listen/"
watchEnabled="false"/>
```

Example::

```
<Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
tempDir="/opt/apache/tomcat/war-temp/"
deployDir="/opt/apache/tomcat/webapps"
watchDir="/opt/apache/tomcat/war-listen/"
watchEnabled="false"/>
```

11. In the **Deployer** stanza, update the **watchEnabled** to false.

Before:

```
<Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
tempDir="/opt/apache/tomcat/war-temp/"
deployDir="/opt/apache/tomcat/webapps"
watchDir="/opt/apache/tomcat/war-listen/"
watchEnabled="true"/>
```

After:

```
<Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
tempDir="/opt/apache/tomcat/war-temp/"
deployDir="/opt/apache/tomcat/webapps"
watchDir="/opt/apache/tomcat/war-listen/"
watchEnabled="false"/>
```

12. Save server.xml
13. If there is a firewall, open the firewall port for the Apache Tomcat port

7.3 Apache HTTP

1. Login to the webserver file system
2. Run the following command to install the Apache HTTP server

```
sudo yum install httpd
```

3. Open the 00-proxy.conf file with an editor

```
/etc/httpd/conf.modules.d/00-proxy.conf
```

4. Verify the following modules are uncommented

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

5. If any changes were done, save the 00-proxy.conf

6. Open the httpd.conf file with an editor

```
sudo vi /etc/httpd/conf/httpd.conf
```

7. Add the following lines to the bottom of the httpd.conf

```
<IfModule proxy_module>
ProxyRequests Off

<Proxy balancer://<CLUSTERNAME>>
BalancerMember http://<TC_HOSTNAME>:<TC_PORT> route=<JVMROUTE>
BalancerMember http://<TC_HOSTNAME>:<TC_PORT> route=<JVMROUTE>
</Proxy>

ProxyPass /<CONTEXT_ROOT> balancer://<CLUSTERNAME>/<CONTEXT_ROOT>
stickysession=JSESSIONID
ProxyPassReverse /<CONTEXT_ROOT>
balancer://<CLUSTERNAME>/<CONTEXT_ROOT> stickysession=JSESSIONID

</IfModule>
```

Example:

```
<IfModule proxy_module>
ProxyRequests Off

<Proxy balancer://tccluster>
BalancerMember http://tc85a.loc.com:8080 route=jvm1
BalancerMember http://tc85b.loc.com:8080 route=jvm2
</Proxy>

ProxyPass /examples balancer://tccluster/examples
stickysession=JSESSIONID
ProxyPassReverse /examples balancer://tccluster/examples
stickysession=JSESSIONID

</IfModule>
```

- CLUSTERNAME = Cluster name for the Apache Tomcat Server. This is defined here
- TC_HOSTNAME = hostname of the Apache Tomcat Server
- TC_PORT = port of the Apache Tomcat Server
- JVMROUTE = the jvmRoute value in the server.xml of the Apache Tomcat Server
- CONTEXT_ROOT

8. Start the Apache HTTPD service

```
sudo service httpd start
```

9. Open a browser and set the URL to the WebServer Hostname and the tomcat context root that is mapped in the httpd.conf file

`http://<WEB_HOSTNAME>/<TC_CONTEXTROOT>`

Example: <http://myweb.loc.com/examples>

Apache Tomcat Examples

- [Servlets examples](#)
- [JSP Examples](#)
- [WebSocket Examples](#)

10. Click each link and test the examples

○ Servlet examples




Servlet Examples with Code

This is a collection of examples which demonstrate some of the more frequently used parts of the Servlet API. Familiarity with the Java(tm) Programming Language is assumed.

These examples will only work when viewed via an http URL. They will not work if you are viewing these pages via a "file://..." URL. Please refer to the *README* file provide with this Tomcat release regarding how to configure and start the provided web server.

Wherever you see a form, enter some data and see how the servlet reacts. When playing with the Cookie and Session Examples, jump back to the Headers Example to see exactly what your browser is sending the server.

To navigate your way through the examples, the following icons will help:

-  Execute the example
-  Look at the source code for the example
-  Return to this screen

Tip: To see the cookie interactions with your browser, try turning on the "notify when setting a cookie" option in your browser preferences. This will let you see when a session is created and give some feedback when looking at the cookie demo.

Hello World	 Execute	 Source
Request Info	 Execute	 Source
Request Headers	 Execute	 Source




○ JSP Examples

JSP Samples

This is a collection of samples demonstrating the usage of different parts of the Java Server Pages (JSP) specification. Both JSP 2.0 and JSP 1.2 examples are presented below.

These examples will only work when these pages are being served by a servlet engine; of course, we recommend [Tomcat](#). They will not work if you are viewing these pages via a "file://..." URL.

To navigate your way through the examples, the following icons will help:

-  Execute the example
-  Look at the source code for the example
-  Return to this screen

Tip: For session scoped beans to work, the cookies must be enabled. This can be done using browser options.

JSP 2.0 Examples

Expression Language		
Basic Arithmetic	 Execute	 Source
Basic Comparisons	 Execute	 Source
Implicit Objects	 Execute	 Source

○ WebSocket Examples

Apache Tomcat WebSocket Examples

- [Echo example](#)
- [Chat example](#)
- [Multiplayer snake example](#)
- [Multiplayer drawboard example](#)