

Configuring Tomcat5 and Apache 2.2 with Virtual Hosts using mod_jk

Table of Contents

1	Overview.....	1
2	System Details.....	1
3	Installing Apache.....	1
4	Installing JDK.....	1
5	Installing Tomcat.....	1
6	Installing mod_jk.....	2
7	Configure Tomcat.....	2
8	Configure Apache.....	3
9	Conclusion.....	6
10	Resources/Links.....	6

1 Overview

This tutorial explains how to setup a web server in order to support Java Server Pages (JSP) and Servlets using virtually hosted websites. It is my intention that this tutorial will help anyone that has attempted to install such a system without success. If you find any inconsistencies within this tutorial, please notify me.

2 System Details

Operating System: SLES 10 SP2

Webserver: Apache 2.2.3

Servlet Container: Tomcat 5.0.30

Tomcat Connector: mod_jk 4.1.30

JDK: Java 1.5.0

3 Installing Apache

Install Apache with YaST > Software > Software Management either by selecting the Web Server Pattern or from the command line with `yast -i apache2`

4 Installing JDK

Install the JDK with YaST > Software > Software Management either by searching for java and installing the java-1_5_0-sun package along with the -devel package as well, or from the command line with `yast -i java-1_5_0-sun`

5 Installing Tomcat

Install Tomcat with YaST > Software > Software Management either by searching for tomcat and installing the tomcat5 package, or from the command line with `yast -i tomcat5`

6 Installing mod_jk

In order to make the connection between Tomcat and Apache, we will need to download and install mod_jk connector. Many outdated resources recommend installing the mod_jk2 connector, but I have found that it has been deprecated and although mod_jk was developed before mod_jk2, it is still fully supported and is very stable. Mike Millson gave some good reasoning behind using mod_jk for connecting Tomcat to Apache here:

http://www.meritonlinesystems.com/docs/apache_tomcat_redhat.html

Here is what he had to say:

"At this point, Apache and Tomcat should be working separately in standalone mode. You can run Tomcat in standalone mode as an alternative to Apache. In fact, in some cases, it is said that Tomcat standalone is faster than serving static content from Apache and dynamic content from Tomcat. However, there are the following compelling reasons to use Apache as the front end:

- 1. You can use Apache to buffer slow connections. Tomcat uses java.io, which uses a thread for each request, so Tomcat can run out of connections as the number of slow requests grows. This could be an issue if your application supports a large number of dial-up users.*
- 2. You can use a connector such as mod_jk to load balance amongst several Tomcat instances.*
- 3. You can take advantage of Apache features such as cgi and PHP.*
- 4. You can take advantage of Apache modules such as mod_rewrite, mod_headers, and mod_expire.*
- 5. You can isolate virtual hosts in their own Tomcat instances.*

The increased functionality obtained by using Apache on the front end can outweigh the effort required to install and configure a connector."

Novell does not package mod_jk with apache on SLES 10, but does make available through the openSUSE Build Service several modules which do not come supported with the distribution. You can find those packages here

http://download.opensuse.org/repositories/Apache:/Modules/Apache_SLE_10/

You will find that this URL can be used as an Installation Source in YaST by opening YaST > Software > Installation Source > Click Add > Select Specify URL > Click Next > Paste in the URL > Click Next > Click Finish. Once you have the Installation Source added we can open YaST > Software > Software Management > Search for mod_jk and install it. You could also download it directly from the URL by clicking on the correct link for your architecture and finding mod_jk in the list. Once you have it downloaded to the server then you need to execute `rpm -ivh apache2-mod_jk*`

Now you are ready to move to the next stage of configuring Apache and Tomcat.

7 Configure Tomcat

Create the workers.properties file located in /etc/tomcat5/base

I like to copy the annotated version from /usr/share/doc/packages/apache2-mod_jk/workers.properties to give me something to work with. Execute the following:

```
cp /usr/share/doc/packages/apache2-mod_jk/workers.properties /etc/tomcat5/base
```

The workers.properties file contains the details about how each process is linked to Tomcat by defining workers that communicate through the ajpv13 protocol. Refer to the [Workers HowTo](#) for more detail.

Now lets open the workers.properties file and edit the following:

```
workers.tomcat_home=/srv/www/tomcat5/base
workers.java_home=$JAVA_HOME
```

Save and Close the file.

Everything else in the file can stay at the default for the scope of this document. If you need more detailed configuration then refer to the documentation above.

Now lets open the server.xml file in the same location.

You will have a line similar to the following:

```
<Host name="localhost" debug="0" appBase="webapps"
    unpackWARs="true" autoDeploy="true"
    xmlValidation="false" xmlNamespaceAware="false">
```

Add a line below it with the following directive:

```
<Alias>your_host.some_domain</Alias>
```

Save and Close the file.

Everything else in the file can stay at the default for the scope of this document.

For Single Web App Instance

If your wanting to point Tomcat's default host to a single webapp instead of having multiple webapps deployed on the server then you will want to add this file.

Open the location /etc/tomcat5/base/Catalina/Some_Hostname/ and create a file called context.xml with the following one liner in it

```
<Context path="" docBase="<webapp_dir>" debug="0"/>
fill in <webapp_dir> with the name of the directory for the single webapp instance
you have deployed at /srv/www/tomcat5/base/webapps/
```

Save and Close the file.

Set the Group and Owner to tomcat by executing the following:

```
chown tomcat:tomcat context.xml
```

8 Configure Apache

We now need to add the mod_jk and jk.conf to the apache configuration. I like to copy the annotated version from /usr/share/doc/packages/apache2-mod_jk/jk.conf

execute the following:

```
cp /usr/share/doc/packages/apache2-mod_jk/jk.conf /etc/apache2/conf.d/
```

Once it is copied we will need to edit the file and add some information in for SSL abstraction if your going to be using SSL for your connections.

Open /etc/apache2/conf.d/jk.conf and add the following lines at the bottom of the file before the line </IfModule>

```
JkExtractSSL On
JkHTTPSIndicator HTTPS
JkSESSIONIndicator SSL_SESSION_ID
JkCIPHERIndicator SSL_CIPHER
JkCERTSIndicator SSL_CLIENT_CERT
```

Save and Close the file.

We will keep the other defaults in the file alone for the scope of this document. You may want to change or remove some at some point.

Now lets add mod_jk as a module to the configuration with the following command:

```
#a2enmod jk
```

This will add the jk module to the APACHE_MODULES line in /etc/sysconfig/apache2

Also if you are deploying with SSL then you will want to add the SSL Flag with the following command:

```
a2enflag SSL
```

This will add the SSL flag to the APACHE_SERVER_FLAGS line in /etc/sysconfig/apache2 which in turn is used to initialize SSL when apache server is started.

Now lets setup the virtual hosts. You will only have one virtual host file if your setting this up for a single Web App Instance, and of course multiples if you have multiple Web App instances.

Vhost For port 80 No SSL

Change your directory over to /etc/apache2/vhosts.d/

This folder has a few template files you can use for SSL and No SSL. I like to copy the vhost.template to my webapp_name.conf for ease of identification. So lets do that.

```
cp vhost.template webapp_name.conf
```

Open the file we just copied for editing and make it looks similar to the following.

```
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.com
    ServerName somehost.some_domain:8080
    ServerAlias somehost.somedomain
    # if not specified, the global error log is used
    ErrorLog /var/log/apache2/some_webapp-error_log
    CustomLog /var/log/apache2/some_webapp-access_log combined
    # don't loose time with IP address lookups
    HostnameLookups Off
    # needed for named virtual hosts
    UseCanonicalName Off
    # Add JK mount rule - cseader
    JkMount <web_app_name>/* ajpl3
</VirtualHost>
```

You will notice right off the bat you will need to delete several directives which are not needed. For ServerName you will need to put in the port used for the tomcat connection which by default is 8080. ServerAlias is going to be the alias hostname your using for this Web App Instance. Change the noticeable parts in ErrorLog and CustomLog. JkMount can be set for /* which for a single instance setup would point to the root of the instance context set in your context.xml file for tomcat. If you are not doing a single instance then you will want to point this to the directory of your Web App under webapps folder at the location /srv/www/tomcat5/base/webapps/ .

Save and Close the file.

Vhost For port 443 SSL

There is another template file called vhost-ssl.template which we will need to copy. I like to copy the vhost-ssl.template to my webapp_name-ssl.conf for ease of identification. So lets do that.

```
cp vhost-ssl.template webapp_name-ssl.conf
```

Open the file we just copied for editing and make it looks similar to the following.

```
<IfDefine SSL>
<IfDefine !NOSSL>
<VirtualHost _default_:443>
ServerName  somehost.some_domain:8443
ServerAlias somehost.somedomain
ServerAdmin webmaster@dummy-host.example.com
ErrorLog /var/log/apache2/some_webapp-ssl-error_log
TransferLog /var/log/apache2/some_webapp-ssl-access_log
SSLEngine on
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
SSLCertificateFile /etc/ssl/servercerts/somedomain-cert.pem
SSLCertificateKeyFile /etc/ssl/servercerts/somedomain-key.pem
<Files ~ "\.(cgi|shtml|phtml|php3?)">
    SSLOptions +StdEnvVars
</Files>
SetEnvIf User-Agent ".*MSIE.*" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
CustomLog /var/log/apache2/ssl_request_log    ssl_combined
JkMount <web_app_name>/* ajp13
</VirtualHost>
</IfDefine>
</IfDefine>
```

You will already notice there are quite a few lines that you can keep without modifying so if you want to have the comments in your file with those then that is fine. Some of the main parts that need to be

modified are the ServerName, ServerAlias, ServerAdmin, ErrorLog, TransferLog, SSLCertificateFile, SSLCertificateKeyFile, and JkMount. For ServerName you will need to put in the port used for the tomcat connection which by default is 8080. ServerAlias is going to be the alias hostname you are using for this Web App Instance. Change the noticeable parts in ErrorLog and CustomLog. JkMount can be set for /* which for a single instance setup would point to the root of the instance context set in your context.xml file for tomcat. If you are not doing a single instance then you will want to point this to the directory of your Web App under webapps folder at the location /srv/www/tomcat5/base/webapps/. The SSLCertificate Lines will need to be updated with the correct certificate name generated for your configuration. SSL certificate generation is outside the scope of this document.

Save and Close the file.

9 Conclusion

To Finalize and test this configuration you will need to deploy your Tomcat Instance into the webapps directory and start tomcat and apache2.

Start Tomcat:

```
rcat5 start
```

Start Apache Server:

```
rcapache2 start
```

Once everything is started without error then you can test your configuration by pointing your browser to http or https://somehost.somedomain

When you are satisfied and it is working to your expectations you may add tomcat and apache to your init process with the following.

```
chkconfig apache2 on
```

```
chkconfig tomcat5 on
```

Congratulations! Your server should be ready now to serve JSP.

If you need to add more Web Apps to this server then all you will need to do is create another vhosts file and your off and running.

10 Resources/Links

http://www.meritonlinesystems.com/docs/apache_tomcat_redhat.html

[Apache website](#)

[Tomcat website](#)

[Workers HowTo](#)

<http://tomcat.apache.org/connectors-doc-archive/jk2/jk2/vhosthowto.html>

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