By philipp
Published: 2008-08-22 20:30

How To Set Up Apache, Tomcat (mod_jk), SSO (CAS, mod_auth_cas)

This article describes how you can set up Apache and Tomcat, linked with mod_jk. It also explains how you set up the SSO (single sign on) solution JA-SIG CAS to protect servlets (provided by tomcat) and static content (provided by Apache). I worked with OpenSuse 10.2 and 11, Apache2, Tomcat 5.5 and 6. It should work on other distributions as well.

1. Install Apache, Tomcat and mod_jk

Goto Yast, Software, Software Management and search and install Apache (with devel package), Tomcat (with webapps package) and apache2-mod_jk. Next go to System, Runlevel Editor and start both. You can check Apache by pointing your browser at localhost (you should get a access denied) and Tomcat by pointing your browser at localhost:8080 (you should get the default start page).

2. Configure mod_jk

Next, edit `/etc/apache2/httpd.conf` and add:

```
LoadModule jk_module /usr/lib/apache2/mod_jk.so
JkWorkersFile /etc/apache2/workers.properties
JkMount /*.jsp worker1
JkMount /servlets-examples/*
JkMount /cas/* worker1
```

You can do this alternatively in your vhost. This configuration will send all jsp's and all in the path /servlets-examples/* to Tomcat. If you know the exact path to your servlet, you can write:

```
JkMount /trn-webapp-0.8.1/map worker1
```

for example, where map is the servlet

Next, create */etc/apache2/workers.properties* with the following content:

```
worker.list=worker1
worker.worker1.port=8009
worker.worker1.host=localhost
worker.worker1.type=ajp13
```

Then, goto */etc/tomcat5/base/* and check your server.xml. You should find something like this:

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

Make sure, it is enabled (without <!-- ... -->).

At this point, you can edit also your */etc/tomcat5/base/tomcat-users.xml*. You can replace it with this:

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <role rolename="manager"/>
  <role rolename="admin"/>
  <user username="root" password="password" roles="admin,manager"/>
</tomcat-users>
```

You should replace root and password with your own settings

For testing purposes, edit */etc/apache2/default-server.conf* and change *DocumentRoot*, *Directory* and *Options*:

```
[...]
DocumentRoot "/srv/www/tomcat5/base/webapps/"
[...]
<Directory "/srv/www/tomcat5/base/webapps/">
[...]
        Options All
[...]
```

In Yast, Runlevel Editor first restart Tomcat, then Apache. (If you change something in Tomcat, everytime restart Tomcat first and then restart Apache, too!)

Now point your browser again at localhost. Go into the servlets-examples dir and check mod_jk by clicking on the execute links. If everything is fine, go on. Otherwise, troubleshoot with the Apache log (*/var/log/apache2/error.log*) and Tomcat log (*/var/log/tomcat5/base/catalina.out*).

3. Install and configure JA-SIG CAS

Dowload the CAS Server from **http://www.ja-sig.org/products/cas/index.html** and extract it somewhere. Rename the *cas-server-webapp-x.x.war* in the modules dir to *cas.war*.Go to *localhost:8080* with your browser and open the manager (with "root" and "password"). You must select this cas.war to deploy from file.Now, you can check CAS by clicking on the cas link below "Applications". You can authenticate with any equal username/password.

If you are successful, you can secure your servlets with CAS. But you have tho change the authentication method (to authenticate against an LDAP server for example). Read the howto's and wiki at **http://www.ja-sig.org/products/cas/** how to accomplish that!

4. Install and Configure mod_auth_cas

Download the sources from **https://www.ja-sig.org/svn/cas-clients/mod_auth_cas/tags/mod_auth_cas-1.0.7/src/**

Now you can compile mod_auth_cas with apxs2:

```
apxs2 -i -c mod_auth_cas.c
```

If mod_auth_cas is in the folder */usr/lib/apache2*, then everything is fine.Now create a folder *cas* in the */tmp* directory.It's time to edit */etc/apache2/httpd.conf* again (or your vhost). Add:

```
LoadModule auth_cas_module /usr/lib/apache2/mod_auth_cas.so
CASCookiePath /tmp/cas/
CASloginURL https://localhost/cas/login
CASValidateURL https://localhost/cas/serviceValidate
CASCertificatePath /root/Desktop/exported-pem.crt
<Directory "/srv/tomcat6/webapps/">
AuthType CAS
Require valid-user
</Directory">
```

You have to replace localhost with your FQDN! The configuration secures the */srv/tomcat6/webapps/* dir.

You will need a Tomcat SSL Connector! Edit your *server.xml*:

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true" maxThreads="150" scheme="https" secure="true" clientAuth="false" sslProtocol="TLS" connectionTimeout="20000" />
```

Now, it's time to test your configuration: Restart Tomcat and then Apache and point to localhost. If you can login, you win! If not, troubleshooting starts.

I had problems with the SSL handshake: "Unable to perform SSL handshake with (cas server)". Here is the way, I solved it.

First, check the *.keystore* file: It should be in your home directory or in */usr/share/tomcat5/*. If there is no, generate one:

```
linux-3pjy:~ # keytool -genkey -alias tomcat -keyalg RSA -keystore /usr/share/tomcat6/.keystore
```

```
Geben Sie das Keystore-Passwort ein:
 Geben Sie das Passwort erneut ein:
Wie lautet Ihr Vor- und Nachname?
 [Unknown]:  pm
Wie lautet der Name Ihrer organisatorischen Einheit?
 [Unknown]:  ivi
Wie lautet der Name Ihrer Organisation?
 [Unknown]:  fhg
Wie lautet der Name Ihrer Stadt oder Gemeinde?
 [Unknown]:  dd
Wie lautet der Name Ihres Bundeslandes oder Ihrer Provinz?
 [Unknown]:  sn
Wie lautet der Landescode (zwei Buchstaben) fÃ¼r diese Einheit?
 [Unknown]:  de
Ist CN=pm, OU=ivi, O=fhg, L=dd, ST=sn, C=de richtig?
 [Nein]:  ja
Geben Sie das Passwort fÃ¼r <tomcat> ein.
 (EINGABETASTE, wenn Passwort dasselbe wie fÃ¼r Keystore):
```

You need the certificate from this keystore:

```
linux-3pjy:~/Desktop # keytool -export -alias tomcat -keystore /usr/share/tomcat6/.keystore -file exported-der.crt
```

```
Geben Sie das Keystore-Passwort ein:
 Zertifikat in Datei <exported-der.crt> gespeichert.
```

```
linux-3pjy:~/Desktop # openssl
OpenSSL> x509 -out /root/Desktop/exported-pem.crt -outform pem -in /root/Desktop/exported-der.crt -inform der
```

Or you can define your own keysore in your Tomcat Connector in the *server.xml*:

```
<Connector port="8443" protocol="HTTP/1.1"
SSLEnabled="true" maxThreads="150"
scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
connectionTimeout="20000"
keystoreFile="${catalina.home}/conf/server.jks"
keystoreType="JKS" keystorePass="password"
truststoreFile="${catalina.home}/conf/server.jks"
truststoreType="JKS" truststorePass="password"/>
```