

Serveur Tomcat avec Apache2 et ssl

Introduction

Si vous l'avez déjà installé (sinon, passez à la section [Tomcat](#)), vous savez que Tomcat gère tout les requêtes http provenant du port 8180 (ou d'un autre de votre précédent choix). Il serait plus pratique de laisser apache gérer les requêtes http et que celui-ci délègue uniquement celles pour les Servlets et JSP à Tomcat. De plus, apache2 permettra de gérer la connexion sécurisée (https).

Attention, le tuto ne mentionne pas comment réaliser ses vhosts. Il concerne uniquement la mise en place basique de ce mode de fonctionnement apache2-ssl/tomcat !

Prérequis

- Un Lan (ici, en 192.168.0.0/24)
- Un Pc/Mac client
- Debian Etch (installation Minimale, ici en adresse 192.168.0.10)
- [Tomcat](#)

Bien sûr, Tomcat, JDK et votre application (Servlet/JSP) Java fonctionne parfaitement sur le port 8180. On appellera votre application *vosre_servlet*. Ce dernier est stocké dans */usr/share/tomcat5.5-webapps* OU dans */var/lib/tomcat5.5/webapps*. Pour le premier cas, il soit faire un lien vers le second répertoire, soit mentionner l'existence de votre servlets via un fichier xml à mettre dans */usr/share/tomcat5.5/conf/Catalina/localhost* (ce n'est pas mentionné ici ...) Vous connaissez un minimum Apache2. Comme je l'ai mentionné plus haut, je donne uniquement les éléments permettant une config basique Apache2-SSL/Tomcat/Le_Servlet ...

Installation

Installez le serveur Apache2 ainsi que le nécessaire pour Tomcat :

```
aptitude install apache2 libapache2-mod-jk
```

Normalement, les modules ssl et jk sont fonctionnel, faites un :

```
ls /etc/apache2/mods-enabled
```

Vérifiez la présence de jk.load, ssl.conf et ssl.load

Si il en manque, activer les modules JDK et SSL d'Apache2 :

```
a2enmod ssl  
a2enmod jk
```

Configuration

Apache2 seul

Vérifier le fichier `/etc/default/apache2` : La ligne doit être

```
NO_START=0
```

Et non `NO_START=1`

Modifier le fichier `/etc/apache2/ports.conf` et ajouter la ligne Pour le port d'écoute par défaut en mode ssl : `Listen 443`

Relancez apache2 : `/etc/init.d/apache2 restart`

Testez votre serveur a l'adresse <http://192.168.0.10> , ça devrait retourner un joli *It works!*.

Mod jk Apache2

Créer un fichier `/etc/apache2/worker.properties` avec ceci dedans :

```
ps=/
worker.list=default
worker.default.port=8009
worker.default.host=127.0.0.1
worker.default.type=ajp13
```

`port=8009` -> Le port par lequel Tomcat "communique" avec Apache2 `host=127.0.0.1` -> L'ip du serveur Tomcat

Editer le fichier `/etc/apache2/mods-available/jk.load` en ajoutant ceci :

```
#
# Mod_jk settings
#
JkWorkersFile "/etc/apache2/worker.properties"
JkLogFile "/var/log/apache2/mod-jk.log"
JkLogLevel warning
JkMount /votre_servlet default
JkMount /votre_servlet/* default
```

`JkWorkersFile "/etc/apache2/worker.properties"` → le fichier créé plus haut

`JkLogFile "/var/log/apache2/mod-jk.log"` → Le fichier log, vous pouvez le créer a l'endroit indiqué comme ça :

```
touch /var/log/apache2/mod-jk.log && chmod 777 /var/log/apache2/mod-jk.log
```

```
JkMount /votre_servlet default
```

JkMount /votre_servlet/ default* C'est le répertoire de l'application votre Servlet ... (la racine ici c'est */var/lib/tomcat5.5*).

Après un redémarrage de Tomcat5.5 et de Apache2, l'application devrait être visible en allant sur http://192.168.0.10/votre_servlet

Ici, l'application passe par Apache !

Vhost Apache2 pour Votre_Servlet

Modifiez le fichier de configuration */etc/apache2/sites-available/default* pour le vhost et indiquez ceci :

```
NameVirtualHost *
<VirtualHost *>
ServerAdmin admin@mon-domaine.org
#ServerName mon-servlet.mon-domaine.org
DocumentRoot /var/lib/tomcat5.5/
RedirectMatch ^/$ /votre_servlet/
ErrorLog /var/log/apache2/error.log
LogLevel warn
CustomLog /var/log/apache2/access.log combined
ServerSignature On
SetEnvIf User-Agent ".*MSIE.*" \
nokeepalive ssl-unclean-shutdown \
downgrade-1.0 force-response-1.0
</VirtualHost>
```

Testez à présent la même page après avoir relancé le serveur ...

<http://192.168.0.10>

Ici, l'application passe toujours par apache mais cette fois sans mentionner le dossier *votre_servlet* (voir le *RedirectMatch*).

Mode SSL

Créez le répertoire ssl et générez la clef

Allez dans le répertoire d'apache

```
cd /etc/apache2/
```

Créez le dossier ssl

```
mkdir ssl && cd ssl
```

Générez la clef

```
openssl req -x509 -nodes -days 365 -newkey rsa:1024 -out server.crt -keyout server.key
```

-days 365 -> durée de la validité de la clef *rsa:1024* -> Cryptage en 1024 bits (je crois que ça ira ...)

Remplissez le formulaire avec vos données correspondantes...

Retour sur la configuration de Apache2

Pour activer le mode ssl sur son vhost il faut ajouter une modification.

Ajouter ces trois lignes juste après *DocumentRoot /var/lib/tomcat5.5* :

```
SSLEngine on
SSLCertificateFile /etc/apache2/ssl/server.crt
SSLCertificateKeyFile /etc/apache2/ssl/server.key
```

Test Final

Relancez apache2 et testez la page <https://192.168.0.10>

Cette fois tout va bien : ça passe en ssl !

Et maintenant ?

Si vous avez une seconde application (appelée pour l'exemple *votre_second_servlet*, contentez-vous d'ajouter dans le fichier */etc/apache2/mods-available/jk.load* à la suite :

```
JkMount /votre_second_servlet default
JkMount /votre_second_servlet/* default
```

Ensuite, plusieurs configurations sont possibles :

- Il faudra par la suite soit ajouter un vhost pour apache
- Ou supprimer la ligne *RedirectMatch ^/\$ /votre_servlet*. Cela dit, il ne sera plus possible d'accéder aux deux servlets sans les indiquer en plus de l'url. Il faudra donc y accéder a ces adresses http://192.168.0.10/votre_servlet et http://192.168.0.10/votre_second_servlet
- Enfin, vous pouvez très bien laisser le *RedirectMatch ^/\$ /votre_servlet*. Dans ce cas *votre_second_servlet* sera accessible via http://192.168.0.10/votre_second_servlet et le premier via <http://192.168.0.10>

Cela dit, vous avez tout intérêt a vous familiariser avec Apache2 car c'est pas fini pour lui...

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:obsolete:tomcat-ssl>



Last update: **05/07/2014 17:28**