

Apache como Proxy de Tomcat

Podemos hacer que *Apache* se convierta en *proxy* de otras aplicaciones, de tal forma que llamadas al servidor *Apache* puedan ser desviadas a segundas aplicaciones. Esto puede ser especialmente interesante cuando esa segunda aplicación es *Tomcat*, pues podemos de esta forma dar servicio por el puerto 80 (e incluso por el 443) tanto a un servidor web funcionando con *PHP* como un servidor de aplicaciones funcionando con aplicaciones *Java*.

Vamos a configurar este proxy para funcionar por el puerto 80. Para ello debemos partir de un *Apache* funcionando y corriendo por el puerto 80 y un *Tomcat* igualmente funcionando y corriendo por el puerto 8080, siendo ambas las configuraciones por defecto de las aplicaciones.

Los pasos que seguiremos a continuación son los siguientes:

- Cargar los módulos de *Apache* correspondientes con *proxy* y *proxy_http*:

```
sudo a2enmod proxy
sudo a2enmod proxy_http
```

Es posible que el segundo módulo se instale automáticamente al instalar el primero, depende de la versión de *Apache*.

- Configurar el Host Virtual de *Apache* correspondiente, por ejemplo `/etc/apache2/sites-available/000-default.conf` tal como se muestra a continuación:

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    ProxyPass    /proxy/    http://localhost:8080/
    ProxyPassReverse /proxy/    http://localhost:8080/

    <Location /proxy/>
        Order Allow,Deny
        Allow from all
    </Location>
</VirtualHost>
```

El texto escrito en rojo es el que realmente configura el *proxy*, el que está en azul es intrascendente. Con esto lo que estamos diciendo es que cuando en un navegador se escriba `http://localhost/proxy/` desvíe la petición a `http://localhost:8080/`, que está gestionada por *Tomcat*.

Si en *ProxyPass* y *ProxyPassReverse* sustituimos */proxy/* por */*, haremos que directamente al poner la dirección *localhost* nos desvíe a *localhost:8080*.

Si en lugar de querer utilizar como dirección *localhost* quisiéramos utilizar otra URL, necesitamos añadir la correspondiente línea *ServerName*.

Para finalizar la parte de *Apache* solo resta reiniciarlo:

```
sudo service apache2 restart
```

- Configurar Tomcat para que funcione bajo proxy por el puerto 80. En este caso solo tendremos que añadir la línea escrita en rojo en el conector de *server.xml*:

```
<Connector port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           URIEncoding="UTF-8"
           redirectPort="8443"
           proxyPort ="80" />
```

Y reiniciamos el servidor Tomcat:

```
sudo service tomcat9 restart
```

Por último indicar que si queremos que funcione un Host Virtual de *Tomcat* bajo el proxy de *Apache* tendremos que definir el Host Virtual tanto en *Tomcat* como en *Apache*.

Apache como Proxy de Tomcat mediante HTTPS

Ampliando el apartado anterior esta vez vamos a configurar el *Apache* para que funcione de *proxy* de *Tomcat* pero por el puerto 443, es decir, el puerto del protocolo de las conexiones seguras o *HTTPS*.

Comenzamos por activar el módulo *ssl* de *Apache*:

```
sudo a2enmod ssl
```

Pasamos a continuación a editar un Host Virtual de *Apache* para que funcione por el puerto 443, para ello podemos tomar de modelo, o incluso adaptar el archivo *default-ssl.conf*, colocando en él las siguientes líneas:

```
SSLEngine on
SSLProxyEngine On
SSLProxyVerify none
SSLProxyCheckPeerCN off
SSLProxyCheckPeerName off
SSLProxyCheckPeerExpire off

ProxyPass /      https://localhost:8443/
ProxyPassReverse /      https://localhost:8443/

<Location />
    Order Allow,Deny
    Allow from all
</Location>
```

Las líneas marcadas en rojo son necesarias para que Apache acepte el certificado de *Tomcat* y se produzca una correcta conexión *Apache-Tomcat*. La que está escrita en azul es la que realmente activa el *Proxy*. Tras haber realizado este proceso, tendremos la dirección indicada en el *ServerName* del Host Virtual (en el caso de no poner ninguna será <https://localhost>) redirigida hacia <https://localhost:8443/>, que es la dirección por defecto del *Tomcat*, siempre que esté configurado para funcionar por el puerto 8443. Por otro lado debemos asegurarnos que se han establecido correctamente las directivas referidas al certificado y su clave, puestas por defecto en:

```
SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
```

Finalizamos la parte de Apache reiniciándolo con el sabido:

```
sudo service apache2 restart
```

Nos resta ahora por configurar la parte concerniente a *Tomcat*. Para ello en primer lugar debemos generar el archivo con el almacén de claves que se ha visto anteriormente:

```
sudo keytool -genkey -alias miclave -keyalg RSA -keystore
/var/lib/tomcat9/mialmacen
```

Tras contestar a las diferentes preguntas que nos hace el comando anterior, se crea el deseado almacén. Vamos ahora a editar el conector correspondiente con el puerto 8443 en el archivo *server.xml* de configuración del *Tomcat*, escribiendo las siguientes líneas:

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

    keystoreFile="/var/lib/tomcat9/mialmacen"
    keystorePass="123456"
    keyAlias="miclave"
    keyPass="123456"
    proxyPort="443"/>
```

Por último reiniciamos *Tomcat* con lo que conseguiremos que funcione totalmente nuestro proxy.

Se debe recordar que en caso de que esté activado el puerto 8080 en *Tomcat*, para impedir que se pueda acceder a una aplicación por el 8080 hay que crear y editar el archivo *web.xml* de forma idéntica a como está explicado en el apartado “**Tomcat por el puerto 443**”.