

## **PRÁCTICA**

### **EL SERVIDOR WEB APACHE (LINUX) – SEGUNDA PARTE**

#### **Configuración del servidor web Apache como servidor proxy**

El servidor Apache se puede comportar también como un servidor proxy, es decir, un servidor que acepta peticiones de páginas web provenientes de otros ordenadores, gestiona estas peticiones recogiendo esas páginas y reenviándoselas al cliente que hizo inicialmente la petición. De esta forma sólo el ordenador en el que está funcionando el servidor proxy será visto "desde fuera" de la red en la que estén todas las máquinas.

El uso típico de un servidor proxy es el siguiente: Se tiene una red local con direcciones IP privadas. Hay un ordenador que tiene dos tarjetas de red en las que están configuradas dos direcciones IP: una privada, para comunicarse con los ordenadores de la "red local interior" y otra pública, para poder comunicarse con Internet. En este ordenador se instala un servicio proxy que permite a todos los demás ordenadores acceder a Internet a través de él. Internet sólo tiene constancia de la existencia de un ordenador haciendo peticiones por lo que, además, se consigue aislar a los ordenadores interiores de peligros potenciales.

El servidor proxy será cargado con todas las peticiones provenientes de los clientes, así que, dependiendo del número de ordenadores que lo utilicen y del tráfico que esos ordenadores generen la carga del servidor puede llegar a ser grande.

Una ventaja del servidor proxy es la posibilidad de mantener una caché de páginas web que ya han sido pedidas, de forma que ante una petición de una página repetida se enviará esta desde el servidor proxy local y no habrá que hacer la petición de nuevo al exterior.

En el servidor también se almacenará qué equipo hizo la petición de cada página, así como el momento en que esta petición se hizo. Esto puede ser un problema para ciertas comunicaciones de carácter "personal" que puedan realizar las personas que usan este servicio.

En el caso del Apache, la instalación predeterminada que se hace con las distribuciones tipo Red Hat incluyen un servidor proxy sencillo. En principio no se cargan los módulos que permiten el uso de la caché, aunque puede hacerse por separado. La configuración mínima necesaria para poner en funcionamiento el servidor proxy es la siguiente (directivas que deben ponerse en el archivo httpd.conf):

```
<IfModule mod_proxy.c>
```

```
ProxyRequests On
```

```
<Proxy      *:8080>  
    Order deny,allow  
    Deny from all
```

```
    Allow from 10.30.48
</Proxy>

</IfModule>
```

Nota: También sería necesario añadir otra directiva `Listen 8080` para que el servidor escuche en dicho puerto.

En el ejemplo anterior se ha activado el servicio proxy y se ha permitido su uso a los equipos cuya IP comience por 10.30.48. Se pueden indicar IP completas, nombres completos de equipos, nombres de dominio, IPs parciales y también cualquier usuario (indicando "all").

Una vez configurado el servidor proxy habrá que configurar adecuadamente todos los clientes (los navegadores web) para que utilicen ese servidor como proxy. Habrá que indicar la dirección IP del servidor así como el puerto utilizado para este servicio.

Editar -> Preferencias -> (pestaña) Avanzado -> (pestaña) Red -> (botón) Configuración

Hay que elegir "Configuración manual del proxy" y, en "Proxy HTTP" indicar la IP del servidor Apache y el puerto en el que escucha peticiones tipo proxy.

## Host virtuales

Hasta ahora se ha visto la configuración de un servidor web Apache en el que hay sólo un servicio, pero un mismo servidor web Apache puede hacerse cargo de varios sitios web diferentes si se configuran los denominados "host virtuales".

### Host virtual basado en nombre

En el caso de los host virtuales basados en el nombre, un mismo servidor con una sola dirección IP puede dar servicio a varios sitios web con diferentes nombres. Por ejemplo, en un ordenador con una sola dirección IP podrían estar los sitios web:

```
www.diariodeleon.com
www.diariodevigo.com
www.lavozdegalicia.com
```

Para que un host virtual basado en nombre pueda funcionar correctamente es necesario un servidor DNS que esté correctamente configurado, de forma que todos esos nombres de sitios web apunten a la misma dirección IP. Cuando las peticiones llegan al servidor es éste el que se encarga de discernir cuál de los sitios web debe servir, basándose para ello en el nombre del sitio web que se está pidiendo.

Para utilizar host virtuales basados en nombre se debe indicar la dirección IP (y también el puerto si se quiere) que aceptará las peticiones en el servidor. Esto se hace mediante la directiva `NameVirtualHost`. En el caso habitual en el que se quieren aceptar peticiones en todas las direcciones IP se puede indicar un asterisco (\*).

El siguiente paso es crear una sección con etiquetas `<VirtualHost>` para cada uno de los diferentes host virtuales que se quieran mantener. El argumento que acompañará a la etiqueta `VirtualHost` debe ser exactamente el mismo que el indicado en la directiva `NameVirtualHost` indicado anteriormente.

Dentro de cada bloque `<VirtualHost>.....</VirtualHost>` deberá incluirse como mínimo una directiva `ServerName` para indicar qué host (qué dirección web) se sirve y una directiva `DocumentRoot` para decir dónde están los archivos con las páginas que se van a servir en ese host.

Cuando se crean host virtuales se debe crear también uno más para el que hasta ahora era el host por defecto, las directivas `ServerName` y `DocumentRoot` deberían ser las mismas que las que se indicaron en la parte global del archivo de configuración.

En el caso descrito anteriormente, las directivas a incluir en un fichero de configuración posible serían:

```
NameVirtualHost *:80

<VirtualHost *:80>
ServerName www.diariodeleon.com
DocumentRoot /www/diariodeleon
</VirtualHost>

<VirtualHost *:80>
ServerName www.diariodevigo.com
DocumentRoot /www/diariodevigo
</VirtualHost>

<VirtualHost *:80>
ServerName www.lavozdiegal.cia.com
DocumentRoot /www/lavozdegalicia
</VirtualHost>
```

Cuando llega una petición web al servidor primero se mira si la IP coincide con alguna de las indicadas en la directiva `NameVirtualHost`. Si coincide, se mirará en cada sección `<VirtualHost>` en la que también coincida la IP y se tratará de encontrar una directiva `ServerName` que coincida con la que trae la petición. Si coincide, se servirán las páginas correspondientes, si no coincide ninguna se utilizará por defecto el primer host virtual que aparezca en el fichero de configuración (en el que coincida la IP con la etiqueta `VirtualHost`).

### Utilizar puertos diferentes

Utilizando un sólo nombre de sitio web y una sola dirección IP se pueden tener varios sitios web utilizando puertos diferentes. El puerto "bien conocido" para el servicio web es el 80, pero puede configurarse el servidor para que se utilicen otros puertos, como en este ejemplo, donde se utilizan hosts virtuales con IP:

```
Listen 80
Listen 8080
```

```
<VirtualHost *:80>
DocumentRoot /www/lavozdegalicia
</VirtualHost>
```

```
<VirtualHost *:8080>
DocumentRoot /www/diariodeleon
</VirtualHost>
```

### Configurar usuario y contraseña para acceder a un directorio web

Es posible asignar un usuario y una contraseña para acceder a un sitio web gracias al fichero **.htaccess** . De esta forma podemos hacer que dentro de /var/www/webpredet, si queremos crear una carpeta /admin/ tal que /var/www/webpredet/admin a la que solo tengan acceso aquellos que dispongan de un usuario y una contraseña, lo que deberemos hacer eso:

1. Añadir un fichero .htaccess dentro de /var/www/webpredet/admin que contenga el siguiente texto:

```
AuthType Basic
AuthName "Password Protected Area"
AuthUserFile /var/www/webpredet/admin/.htpasswd
Require valid-user
```

2. Acceder a la web <http://www.htaccesstools.com/htpasswd-generator/> y elegir un usuario y una contraseña, por ejemplo, usuario paco contraseña eslaclave, correspondería con:

```
paco:$apr1$cfMN0Ub/$o7z8MNxWp36MeTCRNKK580
```

3. Crear el fichero .htpasswd dentro del directorio /var/www/webpredet/admin con el texto

```
paco:$apr1$cfMN0Ub/$o7z8MNxWp36MeTCRNKK580
```

Una vez hecho esto, si accedemos desde nuestro navegador a <http://localhost/admin/> nos solicitará un usuario y una contraseña. Suponiendo que nuestra directiva DocumentRoot es /var/www/webpredet/.

### Directivas en .htaccess

El fichero .htaccess nos permite realizar una serie de acciones interesantes de cara a la mejora del posicionamiento SEO de nuestro servidor, como por ejemplo, mejorar las URLs haciéndolas amigables. Por ejemplo, si en lugar de acceder a nuestros ficheros .html de esta manera:

<http://localhost/nuestrofichero.html> queremos acceder desde <http://localhost/nuestrofichero> , nuestro fichero .htaccess en el directorio raíz de nuestro sitio web, debería ser de la siguiente forma:

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-D
RewriteCond %{REQUEST_FILENAME}.html -f
RewriteRule ^(.*)$ $1.html
```

Para que esto funcione, es necesario que esté activado el módulo `mod_rewrite`.

Una directiva de seguridad interesante que se puede añadir en el `.htaccess` es aquella mediante la cual denegamos el acceso a aquellas carpetas que **no contengan** un fichero `index` dentro:

```
Options All -Indexes
```

Si necesitamos que nuestro sitio web “fuerce” que todos aquellos accedan mediante el dominio completo con las `www` incluidas, la regla a indicar en el `.htaccess` sería la siguiente:

```
RewriteCond %{HTTP_HOST} .  
RewriteCond %{HTTP_HOST} !^www\.midominio\.com$  
RewriteRule (.*?) http://www.midominio.com/$1 [R=301,L]
```

Si por ejemplo nos mudamos de dominio y queremos redireccionar mediante `.htaccess` a los visitantes al nuevo dominio, podríamos hacerlo de la siguiente forma:

```
RewriteRule ^(.*)$ http://www.minuevodomino.com/$1 [R=301,L]
```

Algunas de las guías interesantes sobre este fichero se pueden encontrar en:

- <http://www.netandsoftware.es/articulos-blog/articulos-del-blog/art-para-webmasters-y-joomla/505-htaccess-en-profundidad-la-guia-completa>
- <http://www.htaccess-guide.com/>
- <https://www.whoishostingthis.com/resources/htaccess/>

### Asignar IP local a un dominio

Para asignar una IP local a un dominio, podemos modificar el fichero `/etc/hosts` de nuestro ordenador, de manera que, si queremos que [www.miweb.com](http://www.miweb.com) apunte a 127.0.0.1 el fichero quedaría de la siguiente forma:

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6  
127.0.0.1 www.miweb.com miweb.com
```

## EJERCICIOS

1.- Implementar la capacidad que posee Apache para alojar varios sitios web en un mismo equipo con una única IP o nombre, mediante el uso de diferentes puertos.

De manera que mediante el puerto 80 se acceda al sitio web por defecto alojado en el directorio local `/var/www/webpredet` que tendrá una página de inicio sencilla que puede ser la siguiente (guardar las siguientes líneas en un archivo de texto con nombre `inicio.html`):

```
<html>  
<body bgcolor="#FFEE00">  
Página principal del sitio web por defecto.  
</body>  
</html>
```

Y otro sitio web al que se accederá mediante el puerto 3221 alojado en el directorio /var/www/webpuerto3221 que posea una página de inicio sencilla que puede ser la siguiente (guardar las siguientes líneas en un archivo de texto con nombre inicio.html):

```
<html>
<body bgcolor="#CCEE00">
Página principal del sitio web PUERTO 3221
</body>
</html>
```

Para acceder ha dicho sitios web habrá que escribir en nuestro navegador las url siguientes:

Para acceder a la página principal del sitio por defecto:

`http://IPservidorWeb`

Para acceder a la pagina principal del sitio webpuerto3221

`http:/ IPservidorWeb:3221`

2.- Implementar la capacidad que posee Apache para alojar varios sitios web en un mismo equipo con una única IP pero con distintos nombres para crear:

Un sitio web por defecto con las características antes mencionadas, alojado en el directorio /var/www/webpredet. De manera que al escribir en la barra de dirección del navegador

`http://www.midominioof3.com`

Aparezca la página de inicio del sitio por defecto.

Otro sitio web alojado en el directorio /var/www/webdeperico con una página de inicio llamada inicio.html

```
<html>
<body bgcolor="#FFBB00">
Página principal del sitio web de PERICO.
</body>
</html>
```

de manera que al escribir en la barra de dirección del navegador

`http://perico.midominioof3.com`

Aparezca dicha página.

Nota: Para realizar este ejercicio es necesario tener un servidor DNS para el dominio midominioof3.com en el que se indique la traducción del nombre www.midominioof3.com a la IP del equipo en el que está el servidor web. Además, este equipo deberá tener como alias el

nombre perico.midominiof3.com para que, se ponga un nombre u otro, la IP devuelta sea siempre la del equipo que tiene los sitios web (será luego Apache quien discrimine qué sitio web debe servir).

3.- Editar el archivo de configuración para que el Sevidor Apache funcione como proxy escuchando peticiones por el puerto 8080. De manera que un cliente de la red donde este nuestro servidor pueda acceder a cualquier sitio web a través de él.

Nota: Es necesario configurar el navegador en el cliente en la sección proxies para comprobar que funciona.

Una vez finalizado este ejercicio quitar la configuración del proxy y borrar la caché del navegador para evitar problemas.