

Práctica 4 - Administración de servidores web: Apache

Jose Jesús Torronteras Hernández

1. Recorre las carpetas del servidor observando qué se guarda en ellas. Necesitarás acceder al archivo `httpd.conf` para modificar la configuración de Apache

```
i42tohej@NEMT5:~/httpd$ ls -l
total 104
drwxr-xr-x  2 i42tohej upi0 1024 abr 24 12:19 bin
drwxr-xr-x  2 i42tohej upi0 1024 abr 24 12:19 build
drwxr-xr-x  2 i42tohej upi0 1024 abr 24 12:19 cgi-bin
drwxr-xr-x  4 i42tohej upi0 1024 abr 24 13:48 conf
drwxr-xr-x  3 i42tohej upi0 2048 abr 24 12:19 error
drwx----- 2 i42tohej upi0 1024 dic 16 19:24 htdocs
drwxr-xr-x  3 i42tohej upi0 9192 abr 24 12:19 icons
drwxr-xr-x  2 i42tohej upi0 7168 abr 24 12:19 include
drwxr-xr-x  4 i42tohej upi0 1024 abr 24 12:18 lib
drwxr-xr-x  2 i42tohej upi0 1024 may  4 18:12 logs
drwxr-xr-x  4 i42tohej upi0   80 abr 24 12:19 man
drwx----- 14 i42tohej upi0 12288 dic 16 19:24 manual
drwxr-xr-x  2 i42tohej upi0  5120 abr 24 12:19 modules
i42tohej@NEMT5:~/httpd$
```

2. Por defecto, Apache escucha por el puerto 80. Comprueba si esto produce algún error e investiga cuál es el motivo. Modifica el puerto de escucha para que sea el 8080. Comprueba que funciona con un navegador, accediendo a `localhost:8080`.

No podemos ejecutar Apache en el puerto 80, ya que está reservado. Para hacer funcionar nuestro servidor modificaremos el puerto, para ello nos iremos al archivo de configuración de Apache que se encuentra en:

`httpd/conf/httpd.conf`

Modificaremos `Listen`, para que así escuche el puerto deseado. (Puerto 8083 ya que otro alumno lo tenía usándolo)

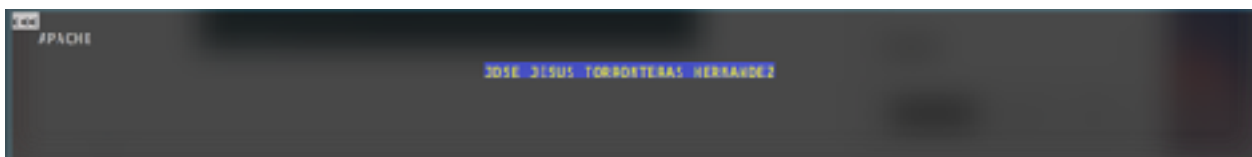
```
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the «VirtualHost»
# directive.
#
# Change this to listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 8083
```

3. Los archivos que sirve Apache están en una carpeta concreta. Cambia dicha carpeta (DocumentRoot), para que sea \$HOME/httpd-docs/. Crear la carpeta y copiar dentro un conjunto de archivos .html para tu servidor web, dándole como nombre a uno de ellos index.html e introduciendo en él tu nombre y apellidos.

Los archivos de Apache se encuentran en la carpeta por defecto htdocs. Para utilizar otra carpeta modificaremos de nuevo nuestro archivo de configuración pero ahora modificaremos el DocumentRoot. También crearemos una carpeta llamada http-docs donde se alojará los archivos HTML.

```
#  
# DocumentRoot: The directory out of which you will serve your  
# documents. By default, all requests are taken from this directory, but  
# symbolic links and aliases may be used to point to other locations.  
#  
DocumentRoot "/home/i42tohej/httpd-docs"
```

Ejecutamos lynx localhost:8083 para ver el resultado por consola.



4. Cambia el nombre de los archivos índice (archivos que servirá Apache cuando el cliente solo especifique un directorio). Normalmente es index.html, añadir como posibilidad index.htm y probar si funciona. ¿Qué prioridad se utiliza si tenemos más de un archivo índice?.

Si modificamos el archivo index.html a index.htm y comprobamos el funcionamiento veremos que no se carga ningún fichero y se nos muestra en pantalla los ficheros y directorios que tengamos en nuestra carpeta httpd-conf. Para modificar esto y que se nos cargue el fichero index.htm por defecto modificaremos la siguiente línea del archivo httpd.conf

```
#  
# DirectoryIndex: sets the file that Apache will serve if a directory  
# is requested.  
#  
<IfModule dir_module>  
    DirectoryIndex index.htm  
</IfModule>
```

5. ¿Qué opción de qué directiva es la encargada de permitir mostrar el contenido de un directorio aunque este no contenga ningún archivo índice? Crea un directorio nuevo \$HOME/httpd-docs/midirectorio ,incluye un archivo cualquiera y prueba a utilizar esta opción para prohibir o no el listado de archivos en dicha carpeta. ¿Cómo podríamos mejorar el aspecto visual del listado? (pistas, Indexes y httpd-autoindex.conf).

Para evitar el listado de un directorio con cualquier archivo que no sea nuestro HTML, tendemos que modificar del fichero httpd.conf la siguiente linea:

```
Options Indexes FollowSymLinks
```

```
Options FollowSymLinks
```

```
DocumentRoot "/home/i42tohej/httpd-docs"
<Directory "/home/i42tohej/httpd-docs">
    #
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
    #
    # Note that "MultiViews" must be named "explicitly" --- "Options All"
    # doesn't give it to you.
    #
    # The Options directive is both complicated and important. Please see
    # http://httpd.apache.org/docs/2.4/mod/core.html#options
    # for more information.
    #
    Options FollowSymLinks
```

Haremos lo mismo con:

```
<Directory "/home/i42toheh/httpd-docs/directorio">
```

Para mejorar el aspecto de nuestro listado, buscando por internet he encontrado que se puede implementar un código que carga una hoja de estilos.

```
IndexOptions FancyIndexing HTMLTable
IndexStyleSheet /css/autoindex.css
```

6. Encuentra la directiva que especifica el nombre del servidor e introduce su valor correcto.

```
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
#ServerName www.example.com:80
ServerName i42tohej:8080
```

7. Encuentra la directiva que especifica el usuario y grupo para el demonio httpd. ¿Sirve para algo cambiarla?.

Sirve para ejecutar el servidor con ese usuario y grupo

```
<IfModule unixd_module>
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# It is usually good practice to create a dedicated user and group for
# running httpd, as with most system services.
#
User daemon
Group daemon
```

8. Prueba a hacer un telnet al puerto del servidor (telnet localhost 8080) y a mandarle mensajes HTML (escribe “GET / HTTP/1.0” y pulsa dos veces intro). Prueba a escribir HOLA y pulsar dos veces a Intro. Explica qué sucede y los códigos de error que devuelve el servidor.

```
l42tohej@NEWTS:~$ telnet localhost 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^I'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Thu, 04 May 2017 18:32:53 GMT
Server: Apache/2.4.25 (Ubuntu)
Last-Modified: Thu, 04 May 2017 16:36:57 GMT
ETag: "77-54eb56198bcc2"
Accept-Ranges: bytes
Content-Length: 119
Connection: close
Content-Type: text/html

<html>
  <head>
    <title>APACHE</title>
  </head>
  <body>
    <h1>JOSE JESUS TORRONTERAS HERNANDEZ</h1>
  </body>
</html>
Connection closed by foreign host.
l42tohej@NEWTS:~$
```

Al introducir la palabra HOLA da error ya que no es ningún tipo de petición al servidor y por lo tanto no puede procesar nada, dándonos un error HTTP/1.1 400 Bad Request.

9. Encuentra la directiva que se utiliza para la visualización de páginas de error. Modifícala para personalizar el mensaje de error 404 que mostrará el servidor y que sea una cadena del tipo “El URI que has pedido no lo podemos servir”. ¿Se podría especificar un fichero .html de error?. Modifica el error 501 para que muestre “Metodo no implementado” y comprueba que funciona.

Para modificar el mensaje de error nos dirigimos a la carpeta error de nuestro directorio httpd.

En él encontraremos un archivo llamado HTTP_NOT_FOUND.html.var.

Buscaremos nuestro lenguaje y modificaremos el mensaje.

```
Content-language: es
Content-type: text/html
Body:-----es--
<!--#set var="TITLE" value="¡Excl!Objeto no localizado!" -->
<!--#include virtual="include/top.html" -->

El URI que has pedido no lo podemos servir.
```

También se puede tener un archivo para mostrar un mensaje de error personalizado, para ello antes debemos modificar el httpd.conf y establecer cual será el fichero a mostrar en caso de error 404.

```
ErrorDocument 404 /error.html
```

Si queremos modificar el error 501 añadimos al httpd.conf:

```
ErrorDocument 501 "Metodo no implementado"
```

```
i42tohej@NENTS:~/httpd$ telnet localhost 8083
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
|PEPE / HTTP/1.0

HTTP/1.1 501 Not Implemented
Date: Sat, 06 May 2017 15:35:42 GMT
Server: Apache/2.4.25 (Unix)
Allow: HEAD,HEAD,GET,HEAD,POST,OPTIONS,TRACE
Content-Length: 22
Connection: close
Content-Type: text/html; charset=iso-8859-1

Metodo no implementadoConnection closed by foreign host.
i42tohej@NENTS:~/httpd$
```

10. Identifica las directivas relacionadas con los archivos de logs de Apache. Haz un acceso normal y acceso erróneo (por ejemplo, un error de tipo 501). Comprueba los logs y muestra como se han modificado.

Para ver el log de nuestro servidor Apache, nos vamos a la carpeta logs. En el encontramos 3 ficheros: access_log, error_log, httpd.pid.

Si observamos `access_log`:

```
127.0.0.1 - - [06/May/2017:17:35:42 +0200] "PEPE / HTTP/1.0" 501 22
127.0.0.1 - - [06/May/2017:17:40:02 +0200] "GET / HTTP/1.0" 200 119
```

Comprobamos dos tipos de peticiones distintas. Una petición incorrecta con error 501, y una satisfactoria.

Al haber realizado una petición incorrecta, si nos dirigimos al fichero `error_log` observamos:

```
[Sat Nov 06 07:39:33.829172 2009] [error] [pid 7636] [client 127.0.0.1:42294] AH00135: Cannot load the module /usr/lib/httpd/modules/mod_ssl.so.
```

11. Redirecciona la dirección /uco a www.uco.es, de manera que al acceder a localhost:8080/uco aparezca la web de la Universidad de Córdoba.

Para hacer una redirección introduciremos lo siguiente en nuestro archivo de configuración de Apache, httpd.conf

```
Redirect /uco http://www.uco.es
```

Comprobaremos su funcionamiento introduciendo:

```
lynx localhost:8083/uc
```

Universidad de Córdoba [et al.]

Universidad de Córdoba

Investigación de Córdoba

Acceso directo a:

- Portal del profesorado
- Portal de estudiantes (PUE)
- Portal del Páid
- Registro de Instituciones
- Investigación y transformación
- Recursos de la Universidad
- Gestión y efectos académicos
- Servicios para estudiantes

12. Crea un Host virtual de manera que cuando un cliente se conecte al servidor usando localhost:8080 el servidor muestre la carpeta raíz original, y cuando se conecte usando IPMAQUINA:8080 muestre la subcarpeta /midirectorio. IPMAQUINA es la IP de la máquina en la que estás, puedes averiguarla con ifconfig. Haz que tengan ficheros de log independientes (local-access.log, local-error.log, ip-access.log y ip-error.log).

Modificamos httpd.conf y añadimos:

```
<VirtualHost 150.214.110.82:8083>
    ServerAdmin webmaster@localhost
    DocumentRoot /home/i42tohej/httpd-docs/directorio
    ErrorLog logs/ip-error.log
    CustomLog logs/ip-access.log combined
</VirtualHost>
```

Las siguientes líneas de código están creando un Host Virtual y permite que cuando pongamos en el navegador 150.214.110.82:8083 se nos muestre el directorio que hemos introducido (DocumentRoot), además establecemos los logs dentro de la carpeta de logs por defecto en Apache.

Probamos su funcionamiento:

```
Lynx 150.214.110.82:8083
```



Funciona correctamente.

```
i42tohej@VMTS:~/httpd$ cat logs/ip-access.log
150.214.110.82 - - [06/May/2017:18:09:37 +0200] "GET / HTTP/1.0" 403 200 "-" "Lynx/2.8.7rel.2 libwww-FM/2.14"
150.214.110.82 - - [06/May/2017:18:15:01 +0200] "GET / HTTP/1.0" 200 34 "-" "Lynx/2.8.7rel.2 libwww-FM/2.14"
```

13. Haz que el servidor web sirva toda la documentación de Apache. Para ello, habilita la configuración del manual de Apache (extra/httpd-manual.conf) y los módulos que necesite. Explica las distintas líneas que aparecen en dicha configuración, incluyendo el significado de las expresiones regulares.

```
i42tohej@NEWT5:~/httpd$ cat conf/extra/httpd-manual.conf
#
# Provide access to the documentation on your server as
# http://yourserver.example.com/manual/
# The documentation is always available at
# http://httpd.apache.org/docs/2.4/
#
# Required modules: mod_alias, mod_authz_core, mod_authz_host,
#                   mod_setenvif, mod_negotiation
#
```

Como se observa para que funcione el manual de Apache debemos activar los módulos que en el archivo httpd-manual.conf. Para comprobar su funcionamiento tendremos que introducir en el navegador localhost:8083/manual/

```
i42tohej@NEWT5:~/httpd$ ./bin/apachectl -M
Loaded Modules:
  core_module (static)
  so_module (static)
  http_module (static)
  mpm_event_module (static)
  authn_file_module (shared)
  authn_core_module (shared)
```

Comprobamos los
módulos cargados:

```
./bin/apachectl  
-M
```

NO TERMINADO

^/manual(?:/(?:daldelenleslfrljalkolpt-brlrultrzh-cn))?(/*.)?\$

14. Apache permite el acceso a recursos restringidos mediante la creación de usuarios y grupos que deberán autenticarse antes de acceder a dicho recurso protegido. Se debe: crear los usuarios y contraseñas correspondientes dentro de un archivo llamado `.htpasswd` (para esto, tendrás que utilizar el comando `./htpasswd -c .htpasswd usuario`, incluido en la carpeta `bin` de Apache, una vez por cada usuario a añadir y luego copiar el archivo resultante a un sitio protegido; ojo, el `-c` indica que el fichero se cree nuevo, por lo que solo debe utilizarse para el primer usuario) e incluir un archivo `.htaccess` (por defecto, aunque estos nombres de archivo pueden modificarse en `httpd.conf`) con los usuarios o grupos de usuarios que tendrán acceso. La sintaxis de ese archivo es la misma que el resto de configuraciones de directorios de Apache. Debes crear un directorio, que se llamará `carpetaSecreta`, de forma que, para acceder a él, habrá que autenticarse. Los usuarios que tendrán acceso a ese directorio serán: `usuario1`, `usuario2` y `usuario3`; y la contraseña, para todos será: `entra`. Explica los pasos realizados para conseguirlo, y el contenido de los archivos creados. NOTAS: para que el archivo `.htaccess` se interprete por Apache es necesario que, previamente, hayamos activado la directiva `AllowOverride All` sobre el directorio correspondiente (por defecto está a `None`) en `httpd.conf`. Consultar el ejemplo de autenticación en <http://www.cristalab.com/tutoriales/proteger-carpetas-con-.htaccess-y-.htpasswd-c213/>.

Para realizar la restricción primero ejecutaremos el comando en la carpeta `bin`:

```
i42tohej@NEWTs:~/httpd/bin$ ./htpasswd -c .htpasswd usuario
[New password:
[Re-type new password:
Adding password for user usuario
i42tohej@NEWTs:~/httpd/bin$
```

```
i42tohej@NEWTs:~/httpd/bin$ cat .htpasswd
usuario:$apr1$UNZ8bAYL$S4.FUJ.WimkptKabYU01N/
```

Haremos lo mismo para `usuario2`, `usuario3` pero sin la opción `-c`.

Creamos una carpeta llamada `carpetaSecreta` en nuestro `http-docs`.

Dentro de esta carpeta ejecutamos:

```
pico ./htaccess
```

```
UH FICO(tn) 4.10 File: httpd-docs/carpetaSecreta/.htaccess
AuthUserFile /home/i42tohej/httpd/bin/.htpasswd
AuthName Carpeta Secreta
AuthType Basic
require user usuario usuario2 usuario3
```

Es importante que pongamos bien la ruta de `AuthUserFile` que se ha generado antes.

Para activar este fichero debemos ir a nuestro fichero de configuración httpd.conf

Dentro de la directiva <Directory "/home/i42tohej/httpd-docs">

```
#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   AllowOverride FileInfo AuthConfig Limit
#
AllowOverride All
```

Modificamos el valor de AllowOverride.

```
Username for 'CarpetaSecreta' at server 'localhost:8083':
```

Comprobamos que funciona perfectamente.

15. Existe también la posibilidad de permitir o denegar el acceso a diferentes directorios o archivos dependiendo de la dirección IP del cliente. Para ello, disponemos de las directivas Allow from, Deny from y Order, que pueden utilizarse en el archivo de configuración httpd.conf o en cada uno de los directorios mediante el archivo .htaccess. Prueba esta opción de seguridad para permitir el acceso a nuestro servidor web, únicamente para direcciones IP de la subred de los equipos de sobremesa del laboratorio. Prueba también a denegar el acceso a todas las direcciones IP. NOTA: para especificar una subred se debe utilizar la siguiente notación 172.16.215.0/24, donde 24 es el número de bits de la subred (tres primeras cifras) y la subred sería 172.16.215. Averigua cuál es la subred del laboratorio con el comando ifconfig.

Lo primero es comprobar la dirección ip de la máquina. Para ello haremos un ifconfig.

```
eth0      Link encap:Ethernet  HWaddr E4:1F:13:18:A7:AC
          inet addr:150.214.110.82  Bcast:150.214.110.255  Mask:255.255.255.0
```

Observamos que nuestra dirección es 172.16.218.37. El número 37 hace referencia a mi máquina por lo tanto mi subred es 172.16.218.0

Ahora sabiendo la subred si puedo permitir o denegar a partir de esta subred únicamente.

Para ello vamos a modificar en httpd.conf y pondremos en él:

```
Order allow, deny
Require all granted
AllowOverride from 172.16.218.0/24
Deny from 172.16.218.38/24
```

Ahora permitimos el acceso a toda nuestra red y bloqueamos a nuestro compañero con Deny from. Order establece que primero permite y luego deniega. (Cambio de ip por problemas con el ssh).