

Práctica 4: Servidor Apache

1) Para crear un servidor Apache, primero debemos entrar en la carpeta httpd, la cual se encuentra en la carpeta de nuestro \$HOME. Una vez ahí, encontraremos varias carpetas, entre las que destacan:

- conf: Es donde se encuentra la configuración del servidor. Aquí se encuentra el archivo httpd.conf.

- bin: Los ejecutables de nuestro servidor. Gracias a esta carpeta podremos ejecutar, reiniciar y parar nuestro servidor Apache, todo esto con el ejecutable apachectl start/restart/stop.

2) Por defecto, el puerto por el que escucha Apache es el 80, para cambiarlo, tendremos que irnos al directorio httpd/conf y modificar el archivo httpd.conf. Este cambio se consigue con la directiva Listen, la cual se encarga de decir por qué puerto debe de escuchar nuestro servidor. Bastaría con añadir:

```
Listen 8080
```

3) La carpeta donde se encuentran los archivos que sirve Apache, se encuentran por defecto en la carpeta htdocs. Para cambiar esto, utilizaremos la directiva DocumentRoot, la cual se encarga de decirle a nuestro servidor de dónde leer los ficheros índice. Quedando de la siguiente manera:

```
DocumentRoot "/home/i42metoj/httpd-docs"
```

*httpd-docs: Carpeta que creamos en el directorio \$HOME con el comando mkdir.

4) Para añadir un archivo llamado index.htm, bastaría con abrir el archivo index.html con un editor (p.e: nano) y guardarlo con el nombre con la extensión .htm. Una vez hecho ésto, cuando ejecutamos el servidor, seguimos viendo el fichero index.html, ya que, gracias a la directiva DirectoryIndex, podemos decirle a nuestro servidor que elija primero los archivos índices (index.html) y después los que más se le parezcan. Para cambiar esta prioridad, bastaría con cambiar el orden en el que ponemos los archivos:

```
<IfModule dir_module>
    DirectoryIndex index.htm index.html
</IfModule>
```

5) La directiva encargada de mostrar el contenido es Options. Por defecto, viene con dos opciones, Indexes y FollowSymLinks, pues para no mostrar ningún archivo índice bastaría con dejarlo de la siguiente manera:

```
<Directory "/home/i42metoj/httpd-docs/midirectorio">
    Options FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

Aunque también se podría poner un un “-” delante de la directiva Indexes para así, poder prohibir el listado de los archivos de esta carpeta. Quedando de la siguiente manera:

```
<Directory "/home/i42metoj/httpd-docs/midirectorio">
  Options -Indexes FollowSymLinks
  AllowOverride None
  Require all granted
</Directory>
```

Finalmente, para mejorar el aspecto visual del listado, bastaría con añadir la directiva IndexOptions, junto con la opción que queramos (p.e: FancyIndexing).

6) La directiva que especifica el nombre del servicio es Servername, la cuál hay que escribirla de la siguiente manera:

```
Servername i42metoj:8080 (También se puede poner 127.0.0.1)
```

7) Las directivas que especifican el usuario y el grupo para el demonio httpd son:

```
<IfModule unixd_module>
  User daemon
  Group daemon
</IfModule>
```

8) Cuando realizamos un telnet hacia nuestro servidor, y escribimos una petición correcta, por ejemplo GET /HTTP/1.0, nos sale un código 200, lo que significa que todo ha salido bien.

```
i42metoj@NEWTs:~/httpd/bin$ telnet 127.0.0.1 8081
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 08 May 2017 15:41:27 GMT
Server: Apache/2.4.25 (Unix)
Content-Length: 571
Connection: close
Content-Type: text/html; charset=ISO-8859-1

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
<head>
  <title>Index of /</title>
</head>
<body>
<h1>Index of /</h1>
<pre>      <a href="?C=N;O=D">Name</a>                <a href="?C=M;O=A">Last modified</a>      <a href=?C=D;O=A>Size</a>
cription</a><hr>      <a href="index.htm">index.htm</a>                2017-05-04 18:03  130
      <a href="index.html">index.html</a>                2017-05-04 18:02  105
      <a href="midirectorio/">midirectorio/</a>          2017-05-04 20:06   -
<hr></pre>
</body></html>
Connection closed by foreign host.
i42metoj@NEWTs:~/httpd/bin$
```

En cambio, si realizamos una petición errónea, nos saldrá un código 404, el cuál nos dice que es fallo nuestro, es decir, del cliente.

9) Para cambiar el mensaje de error 404, bastaría con irnos a la carpeta /httpd/error/ y allí modificar

```
i42metoj@NEWS:~/httpd/bin$ telnet 127.0.0.1 8081
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
HOLA
HTTP/1.1 400 Bad Request
Date: Mon, 08 May 2017 15:49:00 GMT
Server: Apache/2.4.25 (Unix)
Content-Length: 226
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
</body></html>
Connection closed by foreign host.
i42metoj@NEWS:~/httpd/bin$
```

el archivo HTTP_NOT_FOUND.html.var, y el apartado en el que se muestra en español, habría que cambiarlo y dejarlo de la siguiente manera:

Content-language: es

Content-type: text/html

Body:-----es--

<!--#set var="TITLE" value="¡Objeto no localizado!" -->

<!--#include virtual="include/top.html" -->

El URI que has pedido no lo podemos servir

<!--#if expr="-n v('HTTP_REFERER')" →

Para modificar el mensaje en el error 501, Bastaría con cambiar una directiva llamada ErrorDocument [Tipo Error] [Mensaje], y ponerlo de la siguiente manera:

ErrorDocument 501 “Metodo no implementado”

Para ver un ejemplo:

```
i42metoj@NEWS:~/httpd/bin$ telnet 127.0.0.1 8081
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
asdf / HTTP/1.0

HTTP/1.1 501 Not Implemented
Date: Mon, 08 May 2017 10:40:22 GMT
Server: Apache/2.4.25 (Unix)
Allow: POST,OPTIONS,HEAD,HEAD,GET,HEAD,TRACE
Content-Length: 23
Connection: close
Content-Type: text/html; charset=iso-8859-1

Metodo no implementado
Connection closed by foreign host.
i42metoj@NEWS:~/httpd/bin$
```

10) Para encontrar las directivas relacionadas con los archivos logs, tenemos que irnos al archivo httpd.conf y buscar por ErrorLog. Esta directiva establece el fichero que guarda todas las peticiones que se han hecho al servidor.

Dentro del archivo logs/access_log, encontramos dichas peticiones. Una petición correcta podría verse tal que así:

```
127.0.0.1 - - [08/May/2017:12:31:54 +0200] "GET / HTTP/1.0" 200 571
```

Y una incorrecta así:

```
127.0.0.1 - - [08/May/2017:12:40:22 +0200] "asdf / HTTP/1.0" 501 23
```

11) La directiva que consigue redireccionar a una página es la directiva Redirect, la cuál sólo tendríamos que añadirla a nuestro httpd.conf y declararla tal que así:

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

Posteriormente, reiniciamos el servidor y una vez entramos a localhost:8081/uco entraríamos a la página de la uco.

12) Para crear un virtual host, tendremos que utilizar la directiva <VirtualHost> para poder crearlo. Para ello, hay que escribir lo siguiente:

```
<VirtualHost 150.214.110.82:8081>
    ServerAdmin i42metoj@uco.es
    ServerName 150.214.110.82:8081
    DocumentRoot "/home/i42metoj/httpd-docs/midirectorio"
    ErrorLog "logs/ip-error.log"
    CustomLog logs/ip-access.log combined
</VirtualHost>
```

Con las directivas ErrorLog y CustomLog, creamos los nuevos ficheros logs para este host virtual con el nombre que se pide. Además utilizamos el DocumentRoot para que sólo nos muestre el contenido de mi directorio cuando entramos con 150.214.110.82:8081.

13) Una vez entramos en la carpeta `httpd/conf/extra`, encontraremos un archivo llamado `httpd-manual.conf`, en el cuál veríamos lo siguiente:

```
i42metoj@NEWTS:~/httpd/conf/extra$ cat 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
#

AliasMatch ^/manual(?:/(?:da|de|en|es|fr|ja|ko|pt-br|ru|tr|zh-cn))?(/.*)?$ "/home/i42metoj/httpd/manual$1"

<Directory "/home/i42metoj/httpd/manual">
    Options Indexes
    AllowOverride None
    Require all granted

    <Files *.html>
        SetHandler type-map
    </Files>
    # .tr is text/troff in mime.types!
    <Files *.html.tr.utf8>
        ForceType "text/html; charset=utf-8"
    </Files>

    AddLanguage da .da

    SetEnvIf Request_URI ^/manual/(da|de|en|es|fr|ja|ko|pt-br|ru|tr|zh-cn)/ prefer-language=$1
    RedirectMatch 301 ^/manual(?:/(da|de|en|es|fr|ja|ko|pt-br|ru|tr|zh-cn)){2,}(/.*)?$ /manual/$1$2

    LanguagePriority en da de es fr ja ko pt-br ru tr
    ForceLanguagePriority Prefer Fallback
</Directory>
i42metoj@NEWTS:~/httpd/conf/extra$
```

Los módulos que necesita son los nombrados en la directiva `Required modules`, y para comprobar cuáles están activos, podemos conseguirlo ejecutando: `./apachectl -t -D DUMP_MODULES` Teniendo un resultado como en la siguiente imagen:

```
i42metoj@NEWTS:~/httpd/bin$ ./apachectl -t -D DUMP_MODULES
Loaded Modules:
core_module (static)
so_module (static)
http_module (static)
mpm_event_module (static)
authn_file_module (shared)
authn_core_module (shared)
authz_host_module (shared)
authz_groupfile_module (shared)
authz_user_module (shared)
authz_core_module (shared)
access_compat_module (shared)
auth_basic_module (shared)
reqtimeout_module (shared)
filter_module (shared)
mime_module (shared)
log_config_module (shared)
env_module (shared)
headers_module (shared)
setenvif_module (shared)
version_module (shared)
unixd_module (shared)
status_module (shared)
autoindex_module (shared)
dir_module (shared)
alias_module (shared)
i42metoj@NEWTS:~/httpd/bin$
```

Con la directiva `AliasMatch`, hacemos que, con la ayuda de expresiones regulares, consigamos todos los manuales en distintos idiomas, ya que con la expresión regular `^/manual(?:/(?:da|de|en|es|fr|ja|ko|pt-br|ru|tr|zh-cn))?(/*)?$` `"/home/i42metoj/httpd/manual$1"`, creamos nuestro manual a través de que encuentre un archivo que empiece por `manual` y que, seguido de una barra, encuentre entre las distintas posibilidades recogiendo en un primer argumento. Además, recoge todo lo que termina hasta el final en un segundo argumento. Con el primer argumento escribiremos el idioma de nuestro manual.

14) Para crear los usuarios, nos iremos a la carpeta `httpd/bin` y ejecutaremos el comando `./htpasswd .htpasswd usuario`, poniendo como contraseña `entra`. El resultado sería el siguiente:

```
i42metoj@NEWTs:~/httpd/bin$ cat .htpasswd
usuario1:$apr1$meVyN2R0$S70A2JrdbPC/Ng/Ccs1.v1
usuario2:$apr1$cdgQNve8$XeBhpGkAg0GJD56Qhna1a0
usuario3:$apr1$I54ZmG1h$PQ6ZWc/NQ2f.kI19Z4B0t.
i42metoj@NEWTs:~/httpd/bin$
```

El archivo `.htpasswd`, para una mayor confianza, lo he movido hacia la carpeta `/home/i42metoj/`. Seguidamente, en la carpeta `httpd-docs`, creamos una carpeta con el comando `mkdir CarpetaSecreta`, una vez ahí, creamos un archivo llamado `.htaccess` en el cuál escribimos lo siguiente:

```
GNU nano 1.2.5                                Fichero: htaccess
AuthUserFile /home/i42metoj/.htpasswd
AuthName CarpetaSecreta
AuthType Basic
require user usuario1 usuario2 usuario3
```

`AuthUserFile` de donde nuestro servidor recogerá las contraseñas y usuarios del servidor. El `AuthName` es el nombre que le ponemos, podemos ponerle el que queramos. `AuthType Basic` significa que es de autenticación básica, y por último tenemos el `require user`, que es una directiva que sólo pueden entrar los usuarios especificados, es decir, que aunque sepas la contraseña, si no tienes usuario, no podrás entrar.

Una vez hecho esto, hay que irse al archivo `httpd.conf` y crear la nueva directiva:

```
<Directory "/home/i42metoj/httpd-docs/CarpetaSecreta">
    Options Indexes
    AllowOverride All
    Require all granted
</Directory>
```

`AllowOverride` toma el valor `All` para que esta carpeta en especial, no pueda entrar aquel que no cuente con un nombre de usuario y una contraseña.

15) Para saber cuál es la dirección de red de nuestro servidor, haremos un `ifconfig`, de donde sacamos, a través del `eth0`, que nuestra dirección ip es `172.16.218.36`, con lo que la dirección de red será `172.16.218.0`, y cuya máscara sería `255.255.255.0`, con lo que quedaría: `172.16.218.0/24`.

Una vez sabemos esto, para poder permitir el acceso a nuestro servidor, tendremos que irnos a la directiva de los archivos (`httpd-docs`) y dejarla de la siguiente manera:

```
<Directory "/home/i42metoj/httpd-docs">
    Order allow,deny
    Require all granted
    Allow from 172.16.218.0/24
    AllowOverride None
    Options Indexes FollowSymLinks
    IndexOptions FancyIndexing
</Directory>
```

Lo que hacemos aquí es, hacer que nuestro servidor, si encuentra la ip, primero permita el servicio, y si no se encuentra, que lo deniege. Con Allow from, hacemos que todos los ordenadores cuya subred sea 172.16.218.x pueda entrar a nuestro servidor. Para denegar el servicio, bastaría con añadir una línea más:

Deny from 172.16.218.21/24