



P5.3 – Servidor Apache: Servidores Virtuales

Nicolás López Flores

HOST VIRTUAL BASADO EN NOMBRES	3
Estructura de carpetas y su contenido	3
Configuración Apache	4
Configuración servidor DNS	5
Configuración directorios	5
Comprobación	7
HOST VIRTUAL BASADO EN PUERTOS	8
Estructura de carpetas y su contenido	8
Configuración Apache y firewall	8
Configuración servidor DNS	10
Nueva zona	10
Configuración directorios	11
Comprobación	12

HOST VIRTUAL BASADO EN NOMBRES

Estructura de carpetas y su contenido

Para empezar, debemos crear las carpetas web1/web2 dentro de la ruta /home/alumno, siguiendo la siguiente estructura:

```
alumno@alumnomv:~$ pwd
/home/alumno
alumno@alumnomv:~$ ls -lR | grep web
drwxrwxr-x 3 alumno alumno 4096 nov 14 13:28 web1
drwxrwxr-x 3 alumno alumno 4096 nov 14 13:28 web2
./web1:
./web1/files:
./web2:
./web2/files:
alumno@alumnomv:~$
```

Ahora, **crearemos los archivos** “mainwebX.html” dentro de la carpeta “webX/” con el siguiente código HTML con un link haciendo referencia a un archivo(*fileX.html*) que crearemos en la carpeta “webX/files/” quedando una estructura de carpetas/archivos parecida a la mostrada abajo.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1>Nicolas Lopez Flores</h1>
  <a href="#">file1.html</a>
</body>
</html>
~
~
```

```
alumno@alumnomv:~/web1$ tree
.
├── files
│   └── file1.html
└── mainweb1.html

1 directory, 2 files
alumno@alumnomv:~/web1$
```

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1>Nicolas Lopez Flores</h1>
  <a href="">file2.html</a>
</body>
</html>
~
~

```

```
alumno@alumnov:~/web2$ tree
```

```

.
├── files
│   └── file2.html
└── mainweb2.html

```

```
1 directory, 2 files
```

```
alumno@alumnov:~/web2$
```

Configuración Apache

El siguiente paso será configurar nuestro DNS, crearemos dos copias del archivo `"/etc/bind/sites-available/000-default.conf"` en su misma ruta:

- **web1.conf**
- **web2.conf**

Y editaremos lo siguiente dentro de la variable **<VirtualHost>...</VirtualHost>** en ambos archivos:

ServerName www.webX.daw212.iesldv.com
<i>//Esta variable la encontraremos comentada de manera predeterminada y sirve para establecer el nombre del servidor</i>
ServerAdmin webmaster@localhost
<i>//Indicara cual es el servidor maestro</i>
DocumentRoot /home/alumno/webX
<i>//Indicara al servidor cual es la ruta de inicio</i>
DirectoryIndex webmain1.html
<i>//Establece cual va a ser el archivo principal(index) al que accederá el servidor</i>

Y quedar algo como la siguiente imagen:

```

<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName www.web1.daw212.iesldv.com

    ServerAdmin webmaster@localhost
    DocumentRoot /home/alumno/web1
    DirectoryIndex mainweb1.htm

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

```

Configuración servidor DNS

Después **actualizaremos nuestro servidor DNS** para que acepte los nombres que hemos establecido en el archivo de configuración y verificaremos que la zona este funcionando correctamente con “*named.checkzone*”

```

$TTL 1D

@ IN SOA daw212.iesldv.com. nicolas (
    1 ; serial
    1500 ; refresh
    600 ; retry
    172800 ; expiration
    360 ; TTL negative cache
);

@ IN NS lserver.daw12.leonardo.es.
apachelinux IN A 10.33.12.4
nicolas IN CNAME apachelinux.daw212.iesldv.com.

www.web1 IN CNAME apachelinux.daw212.iesldv.com.
www.web2 IN CNAME apachelinux.daw212.iesldv.com.

~

alumno@alumno:~/etc/bind$ sudo named-checkzone daw212.iesldv.com db.daw212.iesldv.com
zone daw212.iesldv.com/IN: loaded serial 1
OK

```

Configuración directorios

En este punto, las páginas web están creadas y podemos acceder por el nombre, pero no podemos visualizar su contenido porque no hemos creado las directivas correspondientes para sus directorios.

Para configurar correctamente los directorios iremos al archivo “*etc/apache2/apache2.conf*” y añadiremos lo siguiente al final en el apartado de las etiquetas **<directory>...</directory>**:

```
# Sets the default security model of the Apache2 HTTPD server. It does
# not allow access to the root filesystem outside of /usr/share and /var/www.
# The former is used by web applications packaged in Debian,
# the latter may be used for local directories served by the web server. If
# your system is serving content from a sub-directory in /srv you must allow
# access here, or in any related virtual host.
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /home/alumno/web1>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /home/alumno/web2>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

Ahora para **aplicar los cambios** en la configuración deberemos usar el siguiente comando:

```
sudo a2ensite web1.conf
```

a2ensite se utiliza para habilitar la configuración de un sitio web que está definido en un archivo dentro del directorio “*/etc/apache2/sites-available/*”.

Este comando nos pedirá recargar el servicio de apache2 y acto seguido nosotros **reiniciaremos los servicios** tanto de APACHE como de BIND para aplicar todos los cambios.

```

alumno@alumnov:/etc/apache2$ sudo a2ensite web1.conf
Enabling site web1.
To activate the new configuration, you need to run:
  systemctl reload apache2
alumno@alumnov:/etc/apache2$ sudo a2ensite web2.conf
Enabling site web2.
To activate the new configuration, you need to run:
  systemctl reload apache2
alumno@alumnov:/etc/apache2$ sudo systemctl reload apache2
alumno@alumnov:/etc/apache2$ sudo systemctl restart apache2
alumno@alumnov:/etc/apache2$ sudo systemctl restart bind9
alumno@alumnov:/etc/apache2$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2024-11-15 12:11:39 CET; 30s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 4314 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 4318 (apache2)
    Tasks: 6 (limit: 2246)
   Memory: 9.5M
   CGroup: /system.slice/apache2.service
           └─4318 /usr/sbin/apache2 -k start
             └─4319 /usr/sbin/apache2 -k start
               └─4320 /usr/sbin/apache2 -k start
                 └─4321 /usr/sbin/apache2 -k start
                   └─4322 /usr/sbin/apache2 -k start
                     └─4323 /usr/sbin/apache2 -k start

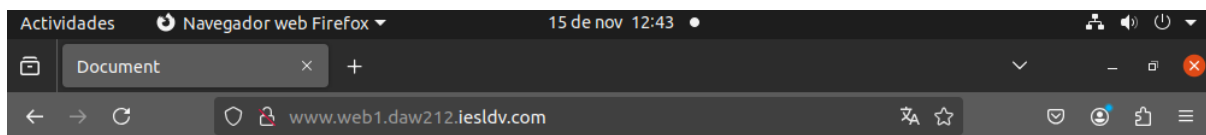
nov 15 12:11:39 alumnov systemd[1]: Starting The Apache HTTP Server...
nov 15 12:11:39 alumnov apachectl[4317]: AH00558: apache2: Could not reliably determine the server's fully c
nov 15 12:11:39 alumnov systemd[1]: Started The Apache HTTP Server.

alumno@alumnov:/etc/apache2$ sudo systemctl status bind9
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2024-11-15 12:11:55 CET; 23s ago
     Docs: man:named(8)
  Main PID: 4330 (named)
    Tasks: 4 (limit: 2246)
   Memory: 4.6M

```

Comprobación

Ahora en teoría si **buscamos en nuestro navegador con el nombre que establecimos en nuestro servidor DNS** (www.webX.daw212.iesldv.com) debería mostrarnos el contenido del archivo “*mainwebX.html*”.



Nicolas Lopez Flores

[file1.html](#)

HOST VIRTUAL BASADO EN PUERTOS

Estructura de carpetas y su contenido

El principio de la configuración (Estructura de carpetas y su contenido) es exactamente igual que en el ejemplo anterior de host virtual basado en nombres.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <a href="">wiki2p.html</a>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <a href="">blog2p.html</a>
</body>
</html>
```

```
alumno@alumnov:~$ tree wikip/ blogp/
wikip/
├── files
│   ├── wiki2p.html
│   └── wiki1p.html
blogp/
├── blog1p.html
├── files
│   └── blog2p.html
2 directories, 4 files
alumno@alumnov:~$
```

Configuración Apache y firewall

Creamos sus **archivos .conf** correspondientes y lo distinto que tiene respecto al host virtual basado en nombres será que hay que establecer con **“Listen”** el **puerto por el que van a escuchar**, cambiar el puerto predeterminado que establece la etiqueta `<VirtualHost* :80>` por el puerto correspondiente. Y por último **habilitar en nuestro firewall UFW los puertos que vamos a usar**.


```
Listen 5001
<VirtualHost *:5001>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName www.wiki.daw212.iesldv.com

    ServerAdmin webmaster@localhost
    DocumentRoot /home/alumno/wikip
    DirectoryIndex wikiip.html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

```
Listen 6001
<VirtualHost *:6001>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName www.blog.daw212.iesldv.com

    ServerAdmin webmaster@localhost
    DocumentRoot /home/alumno/blogp
    DirectoryIndex blogip.html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

```

alumno@alumnomv:~$ sudo ufw allow 5001/tcp
Regla añadida
Regla añadida (v6)
alumno@alumnomv:~$ sudo ufw allow 6001/tcp
Regla añadida
Regla añadida (v6)
alumno@alumnomv:~$ sudo ufw status
Estado: activo

Hasta                Acción              Desde
-----
Apache               ALLOW              Anywhere
Bind9                ALLOW              Anywhere
5001/tcp              ALLOW              Anywhere
6001/tcp              ALLOW              Anywhere
Apache (v6)          ALLOW              Anywhere (v6)
Bind9 (v6)           ALLOW              Anywhere (v6)
5001/tcp (v6)        ALLOW              Anywhere (v6)
6001/tcp (v6)        ALLOW              Anywhere (v6)

```

Configuración servidor DNS

Ahora configuraremos nuestro **DNS**, crearemos una **nueva zona** llamada “db.daw212p.iesldv.com” y añadiremos lo siguiente para **asociar el nombre a nuestra IP**:

```

$TTL 1D

@ IN SOA daw212p.iesldv.com. nicolas (
    1 ; serial
    1500 ; refresh
    600 ; retry
    172800 ; expiration
    360 ; TTL negative cache
);

@ IN NS lserver.daw12.leonardo.es.
apachelinux IN A 10.33.12.4
nicolas IN CNAME apachelinux.daw212.iesldv.com.

www.wiki IN A 10.33.12.4
www.blog IN A 10.33.12.4
~

```

Nueva zona

Editaremos el archivo “**named.conf.local**” y añadiremos la siguiente zona relacionada con el archivo que acabamos de crear.

```

zone "daw212p.iesldv.com" IN {
    type master;
    file "/etc/bind/db.daw212p.iesldv.com";
};

```

Comprobamos el funcionamiento de la zona creada.

```
alumno@alumnoov:/etc/bind$ sudo named-checkzone daw212p.iesldv.com db.daw212p.iesldv.com
zone daw212p.iesldv.com/IN: loaded serial 1
OK
```

Configuración directorios

Configuraremos las **directivas** de los **directorios** como hicimos en el servidor virtual basado en nombres en el archivo “*etc/apache2/apache2.conf*”

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /home/alumno/web1>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /home/alumno/web2>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /home/alumno/wikip>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /home/alumno/blogp>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

Para aplicar los cambios en la configuración usaremos el comando “**a2ensite**” con hicimos anteriormente, **resetearemos los servicios** tanto de APACHE como BIND

```
alumno@alumnov:/etc/apache2$ sudo a2ensite wiki.conf
Enabling site wiki.
To activate the new configuration, you need to run:
    systemctl reload apache2
alumno@alumnov:/etc/apache2$ sudo a2ensite blog.conf
Enabling site blog.
To activate the new configuration, you need to run:
    systemctl reload apache2
alumno@alumnov:/etc/apache2$ sudo systemctl reload apache2
alumno@alumnov:/etc/apache2$ sudo systemctl restart apache2
alumno@alumnov:/etc/apache2$ sudo systemctl restart bind2
Failed to restart bind2.service: Unit bind2.service not found.
alumno@alumnov:/etc/apache2$ sudo systemctl restart bind9
alumno@alumnov:/etc/apache2$ sudo systemctl restart named
alumno@alumnov:/etc/apache2$
```

Comprobación

Al introducir www.XXXX.daw212p.iesldv.com:5001 debería mostrarnos el contenido del archivo que hemos establecido como **DirectoryIndex** en cada IP.

