

Índice

Actividad 1 - Configuración de un servidor web - VirtualHost.....	1
1. Introducción.....	1
1.2. Gestión del servicio Apache.....	1
2. Arquitectura del Servidor apache2.....	2
3. Configuración del demonio httpd.....	2
4. Hosts Virtuales.....	3
4.1 Creación de un host Virtual.....	4
4.2 Verificación del servidor.....	7
5. Arquitectura a configurar.....	8
6. Bibliografía / Webgrafía.....	9

Actividad 1 - Configuración de un servidor web - VirtualHost

1. Introducción

El objetivo de esta práctica es entender cómo **configurar** un servidor web **Apache2**, más concretamente aprenderemos cómo configurar un servidor web para que pueda servir **diferentes sitios web bajo una misma IP**, a través de lo que denominamos Servidores Virtuales.

Se partirá de la máquina virtual configurada en la unidad 1.



1.2. Gestión del servicio Apache

Podemos gestionar el demonio httpd con los siguientes comandos:

```
$ sudo systemctl disable apache2 #Desactiva apache para que sea iniciado durante arranque el sistema operativo
```

```
$ sudo systemctl enable apache2 #Activa apache para que sea iniciado durante arranque el sistema operativo
```

```
$ sudo systemctl stop apache2 #Para el servicio de apache (nuestra web ya no estará disponible)
```

```
$ sudo systemctl start apache2 #Arranca el servicio de apache (nuestra web ja
```

no estará disponible)

```
$ sudo systemctl reload apache2 #Recarga posibles cambios en la configuración del servidor
```

2. Arquitectura del Servidor apache2

La configuración del servidor de apache se basa en una serie de directorios y ficheros que iremos tratando a lo largo de las diferentes prácticas de la unidad y que debemos conocer, a continuación se detallan los más importantes.

- `/var/www/html:` el **contenido web** real, que por defecto solo consta de la página predeterminada de Apache.
- `/etc/apache2:` el directorio que contiene los **archivos de configuración** de Apache.
- `/etc/apache2/apache2.conf:` el **archivo principal de configuración** de Apache. Este archivo administra la carga de muchos de los demás archivos del directorio de configuración.
- `/etc/apache2/ports.conf:` este archivo especifica los **puertos** en los que Apache escuchará. Por defecto, Apache escucha en el puerto 80. De forma adicional, lo hace en el 443 cuando se habilita un módulo que proporciona capacidades SSL.
- `/etc/apache2/sites-available/:` el directorio en el que se pueden almacenar los archivos para cada uno de los **host virtuales** que configuremos. Toda la configuración de bloques de servidor se realiza en este directorio y luego se habilita al vincularse al otro directorio con el comando **a2ensite**. (Lo veremos más adelante)
- `/etc/apache2/sites-enabled/:` el directorio donde se almacenan hosts virtuales habilitados **mediante el comando a2ensite**. Realmente este comando realiza un **enlace simbólico desde** el archivo correspondiente de la carpeta **sites-available** → **sites-enabled**.

3. Configuración del demonio httpd

Si visualizamos el archivo de configuración `/etc/apache2/apache2.conf` podemos ver que cada una de las líneas siguen el patrón: `directiva valor,` donde **directiva**

es el **nombre de la opción** que se quiere configurar, y **valor** será el **valor que se le asigna** a esta directiva. Es importante remarcar que **no hay un signo =** entre la directiva y el valor, tan **solo tenemos que dejar un espacio en blanco**. En caso contrario, no se asignará el valor de forma correcta, y la directiva tendrá un valor **nulo**.

A continuación se detallan algunas directivas útiles:

Directiva	configuración
User User _www	Permite especificar el usuario con el que se ejecutará cada una de las peticiones de apache y, por tanto, deberá tener permisos de acceso a la web.
Group Group _www	Permite especificar el grupo con el que se ejecutará cada una de las peticiones de apache y, por tanto, deberá tener permisos de acceso a la web.
ServerAdmin ServerAdmin admin@example.org	Permite especificar el email del administrador del servidor que aparecerá en los mensajes de error del servidor
ServerName ServerName www.tupaginaweb.es	Permite especificar el dominio que atenderá nuestro servidor.
DirectoryIndex DirectoryIndex index.html	Permite especificar el archivo que buscará el servidor cuando se accede a una url sin especificar la página/path del servidor

Para ver una relación completa de las directivas consulta este [enlace](#).

4. Hosts Virtuales

El servidor web apache2 permite la definición de **VirtualHost**. Los VirtualHost o **servidores virtuales** posibilitan tener **diferentes sitios web** con diferentes configuraciones atendiendo a diferentes nombres de dominio bajo una misma IP pública. La **instalación por defecto** crea un **host virtual** que atenderá todas las peticiones web que llegan al servidor y que se ubica en la siguiente ruta
`/etc/apache2/sites-available/000-default.conf`

```
/etc/apache2/sites-available/000-default.conf
```

```
<VirtualHost *:80> #IP:PUERTO - Cualquier ip por el puerto 80
    ServerName www.example.com # Dominio que atenderá el vhost
    ServerAdmin webmaster@localhost #email del administrador
    DocumentRoot /var/www/html #Document root específico
    ErrorLog ${APACHE_LOG_DIR}/error.log #Fichero de log específico
    CustomLog ${APACHE_LOG_DIR}/access.log combined #Fichero de log específico
</VirtualHost>
```

4.1 Creación de un host Virtual

En primer lugar crearemos un directorio donde situar los archivos de la nueva web que queremos desplegar, para mantenerlo más ordenado, **utilizaremos el nombre de dominio** al que responderá el navegador (para nuestro caso **example.com**).

```
sudo mkdir /var/www/example.com
```

A continuación, asignaremos como propietario del directorio la variable de entorno \$USER que contiene el usuario con el que estamos registrados actualmente y que será con el que llevamos a cabo el despliegue de la web.

```
sudo chown -R $USER:$USER /var/www/example.com
```

Seguidamente **asignamos los permisos** al directorio, de forma que apache pueda leer los archivos y nuestro usuario de configuración tenga acceso total sobre los mismos.

```
sudo chmod -R 755 /var/www/example.com
```

A continuación, **creamos una página de ejemplo** `index.html` utilizando **vi**, **nano**, **gedit** o tu **editor favorito** (incluso mediante Webmin):

```
sudo nano /var/www/example.com
```

Dentro de ella **creamos una página de bienvenida**

```
/var/www/example.com/index.html
```

```
<html>
  <head>
    <title>Bienvenido a tu nuevo Virtual Host!</title>
  </head>
  <body>
    <h1>Hecho! El dominio 'example.com' funciona correctamente!</h1>
  </body>
</html>
```

Para que **Apache sirva este contenido**, es necesario **crear un archivo de host virtual** con las **directivas correctas**. En lugar de modificar el archivo de configuración predeterminado situado en `/etc/apache2/sites-available/000-default.conf` directamente, vamos a crear uno nuevo en:

```
/etc/apache2/sites-available/com.example.conf
```

```
$ sudo nano /etc/apache2/sites-available/com.example.conf
```

Pegamos el siguiente bloque de configuración, similar al predeterminado, pero actualizado para nuestro nuevo directorio y nombre de dominio:

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName example.com
    ServerAlias www.example.com
    DocumentRoot /var/www/example.com
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

El uso del asterisco en VirtualHost se utiliza para especificar que se escucharán todas las

peticiones y no se filtrarán por dirección IP. Para filtrar por dirección IP, se pueden consultar estos [ejemplos](#) de la documentación de Apache.

Ten en cuenta que deberás cambiar **DocumentRoot** por el nuevo directorio y **ServerAdmin** por un correo electrónico al que pueda acceder el administrador del sitio example.com. También agregamos dos directivas: **ServerName**, que establece el dominio de base que debería coincidir para esta definición de host virtual, y **ServerAlias**, que **define más nombres que deberían coincidir con el nombre dominio con el que accederemos al vhost**.



Es una buena opción **utilizar el archivo que configura el virtualhost por defecto** (`/etc/apache2/sites-available/000-default.conf`) como **una plantilla** para crear los archivos de configuración de los siguientes vhost que necesitemos configurar.

```
$ sudo cp sites-available/000-default.conf /etc/apache2/sites-available/nuevo_dominio.conf
```

Habilitaremos el nuevo vhost con la herramienta **a2ensite**, realmente se trata de un script que crea un enlace simbólico entre las carpetas `/etc/apache2/sites-available/` y `/etc/apache2/sites-enabled/`.

```
$ sudo a2ensite com.example.conf
```

A continuación, **realizaremos una prueba** para ver que no haya errores de sintaxis **en la configuración del vhost** realizada:

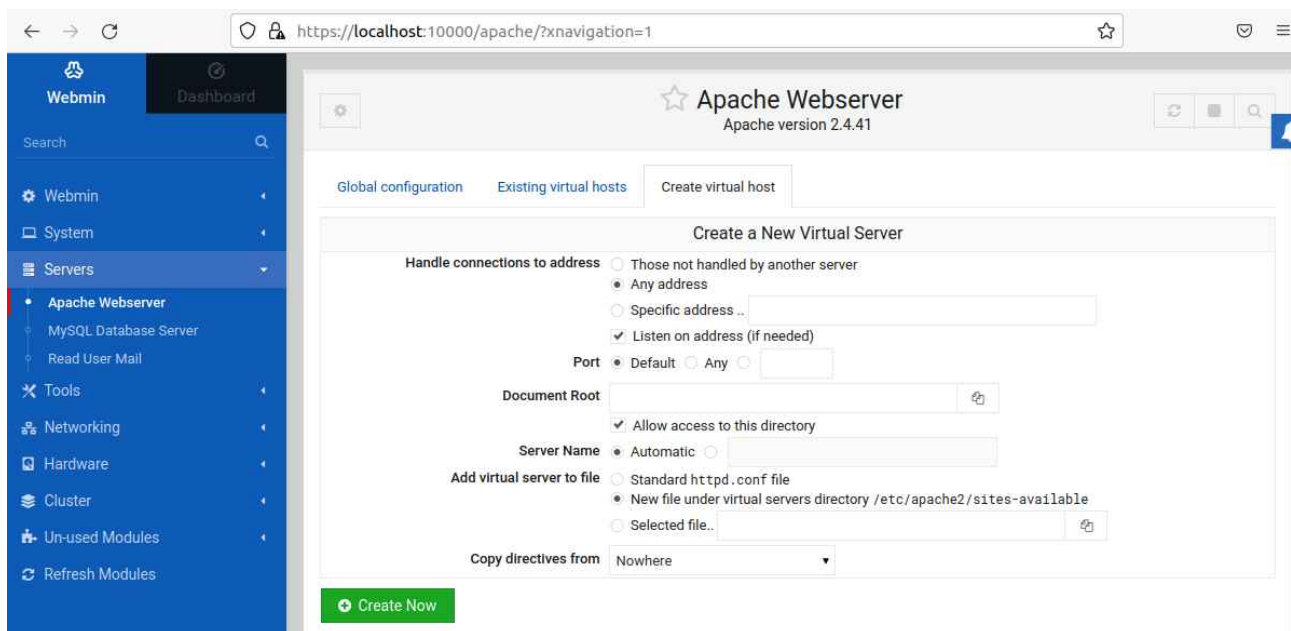
```
$ sudo apache2ctl configtest
```

Por último, reiniciamos apache y cargamos la nueva configuración.

```
$ sudo systemctl restart apache2
```

Con esto, **Apache debería ser el servidor de tu nombre de dominio**

NOTA: También es posible crear un nuevo host virtual de forma gráfica mediante Webmin:



4.2 Verificación del servidor

Nuestro **servidor apache** está listo para servir el contenido del nuevo sitio web, el problema ahora radica en que no tenemos un nombre de dominio configurado para que los clientes accedan a él. Este **nombre de dominio vendrá dado por el servidor DNS** que todavía no podemos utilizar.

Es muy común **en entornos de desarrollo** utilizar el fichero `/etc/hosts` de la máquina cliente (sistema operativo en el que se ejecutará el navegador con el que vamos a probar la máquina. Este fichero contienen un conjunto de **entradas clave-valor** con los **nombres de dominio y las ip asignadas** a cada nombre de dominio.

<code>/etc/hosts</code>	
192.168.1.113	backoffice.ddaw.es
172.20.10.7	moodle.ddaw.es

Introduciremos una nueva entrada con la **IP de la máquina** donde hemos instalado apache y el **nombre de dominio** asignado al **vhost creado**. Para saber la IP de nuestra máquina, utilizaremos el comando `ifconfig` (es posible que hayamos de instalar la librería

net-tools).

<i>/etc/hosts</i>	
192.168.1.113	backoffice.ddaw.es
172.20.10.7	moodle.ddaw.es
192.168.1.1	example.com

Tan solo nos quedaría visitar la URL `http://example.com`, desde el navegador debería ver algo como esto:

Hecho! El dominio 'example.com' funciona correctamente!

NOTA: En caso de que no aparezca el nuevo html, sino el que viene por defecto en Apache, se ha de deshabilitar el virtual host por defecto que viene configurado en Apache (000-default).

5. Arquitectura a configurar

Actividad 1 (1.5 puntos)

5.1 Crea 3 servidores virtuales con las siguientes características:

Nombre de dominio: intranet.ddaw.es	
DocumentRoot	/var/www/01-es-ddaw-intranet
Log de Errores	\${APACHE_LOG_DIR}/001-es-ddaw-intranet-error.log
Log de Acceso	\${APACHE_LOG_DIR}/001-es-ddaw-intranet-access.log
Archivo de configuración	/etc/apache2/sites-available/001-es-ddaw-intranet.conf
web.ddaw.es	
DocumentRoot	/var/www/01-es-ddaw-web
Log de Errores	\${APACHE_LOG_DIR}/002-es-ddaw-web-error.log
Log de Acceso	\${APACHE_LOG_DIR}/002-es-ddaw-web-access.log
Archivo de configuración	/etc/apache2/sites-available/002-es-ddaw-web.conf

backoffice.ddaw.es	
DocumentRoot	/var/www/01-es-ddaw-backoffice
Log de Errores	\${APACHE_LOG_DIR}/003-es-ddaw-backoffice-error.log
Log de Acceso	\${APACHE_LOG_DIR}/003-es-ddaw-backoffice-access.log
Archivo de configuración	/etc/apache2/sites-available/003-es-ddaw-backoffice.conf

5.2.- Crear una página web `index.html` de bienvenida en cada uno de los **Document root** de cada host virtual que indique qué dominio esta sirviendo cada uno de ellas. Seguidamente **accede a cada uno de lo sitios web configurados**. *Deberás demostrar su correcto funcionamiento pegando una captura de pantalla tanto del navegador con la página cargada como del log de acceso del servidor.*

6. Bibliografía / Webgrafía

- Ubuntu Server Docs. Apache. <https://ubuntu.com/server/docs/web-servers-apache> **Canonical**
- Documentación oficial apache server. <https://httpd.apache.org/docs/current/>. **Apache Software Foundation**.
- Glass E. “Como instalar un servidor web apache en ubuntu 20.04” <https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-04-es>. **Digital Ocean**
- Jay laCroix. "Mastering Ubuntu Server. Second edition". (2018). **Packt Publishing**