

Servicio HTTP Apache

1. Presentación

Apache 2 es el servidor HTTP más utilizado actualmente en los servidores Web. Su configuración y su flexibilidad hacen que sea un servidor ineludible.

Cuando un servidor Apache recibe peticiones, puede distribuirlas a procesos hijos. La configuración permite iniciar procesos de manera anticipada y adaptar de manera dinámica este número según la carga.

Apache es modulable. Cada módulo permite añadir funcionalidades al servidor. El módulo más famoso es probablemente el que gestiona el lenguaje PHP, "mod_php". Cada módulo se añade mediante los archivos de configuración y no hace falta iniciar de nuevo el servidor Apache: sólo se le da la orden de volver a leer su configuración.

Apache puede gestionar varios sitios web a la vez, cada uno con su nombre, mediante los anfitriones virtuales.

2. Parada/Reinicio

El nombre del servicio depende de la distribución. A menudo se denomina **apache** o **httpd**. Según la distribución, inicie el servicio mediante el comando **service** o directamente por su nombre, `/etc/init.d/apache`.

- ~ **/etc/init.d/httpd start**: inicia;
- ~ **/etc/init.d/httpd stop**: detiene;
- ~ **/etc/init.d/httpd restart**: reinicia;
- ~ **/etc/init.d/httpd reload**: pide a Apache que vuelva a leer su configuración sin reiniciarse.

Con `systemd`, el servicio se ejecuta de la misma manera:

- ~ `systemctl start httpd`

- ~ `systemctl reload httpd`
- ~ `systemctl restart httpd`
- ~ `systemctl stop httpd`

Apache va acompañado de la herramienta **apachectl**, que retoma los parámetros (lista no exhaustiva) `start`, `stop`, `status`, `reload`, y sobre todo **configtest**, que valida o no el contenido del archivo de configuración de Apache.

3. Configuración

Se almacena la configuración principal en `/etc/httpd/conf/httpd.conf`: controla los parámetros generales del servidor web, los anfitriones virtuales y los accesos. La configuración de los diferentes módulos está ubicada en `/etc/httpd/conf.d`. Los módulos están presentes en `/etc/httpd/modules/`. Por defecto la raíz del servidor, donde se colocan las páginas del sitio, está en `/var/www` o `/srv/www`. Esta posición depende de la directiva **DocumentRoot** en los archivos de configuración.

4. Directivas generales

Es imposible listar todas las directivas del archivo `httpd.conf`, pero algunas son importantes.

- ~ **ServerRoot**: directorio que contiene los archivos del servidor (configuración y módulos). Suele ser `/etc/httpd`.
- ~ **Listen**: puertos en los cuales escucha el servidor Apache. Por defecto 80 (443 en https). Se pueden especificar varios puertos con varias directivas `Listen`. Si el servidor dispone de varias direcciones IP, se puede añadir la IP al puerto asociado: **Listen 192.168.1.3:80**.
- ~ **User**: usuario de los procesos Apache. Nunca se utiliza root, sino una cuenta creada para la ocasión, en general `apache`.
- ~ **Group**: ídem, pero para el grupo. Lleva generalmente el nombre `apache`.
- ~ **ServerAdmin**: dirección de correo electrónico del administrador.

- ✧ **ServerName:** nombre de anfitrión (y puerto) del servidor. No corresponde obligatoriamente al nombre de anfitrión de la máquina. En cambio, es válido. **ServerName** `www.midominio.org`
- ✧ **UseCanonicalName:** si vale **on**, Apache va a responder utilizando la información de ServerName y Port, y no la información mandada por el cliente. Por ejemplo, un `http://192.168.1.3` se transforma en `http://www.midominio.org`.
- ✧ **UserDir:** nombre de un subdirectorío donde cada usuario puede ubicar sus archivos HTML personales. En general, `public_html`. Se accede a él con: **http://www.midominio.org/~login/page.html**
- ✧ **ErrorLog:** archivo donde se colocan los logs de error del servidor. `/var/log/httpd/error_log`.
- ✧ **CustomLog:** registro de Apache. `/var/log/httpd/access_log`.

5. Los directorios, alias y ubicaciones

a. Directory

Las etiquetas **<Directory ruta>** y **</Directory>** permiten agrupar directivas que no sólo se aplicarán a la ruta (y a sus subdirectorios) de los datos. Se aconseja encarecidamente la directiva **Options**.

```
<Directory /var/www/html/images>
Options +Indexes +FollowSymLinks
DirectoryIndex index.php index.html
Order allow, deny
Allow from All
</Directory>

<Directory /var/www/html/cgi-bin>
Options +ExecCGI
</Directory>
```

La directiva **Options** acepta los valores siguientes precedidos de + o - y separados por espacios:

- ✓ **All**: todas las opciones salvo MultiViews;
- ✓ **Indexes**: si el directorio no contiene archivos HTML por defecto (vea DirectoryIndex), se muestra el contenido del directorio en forma de listing;
- ✓ **ExecCGI**: se autoriza la ejecución de scripts CGI;
- ✓ **FollowSymLinks**: el servidor sigue los vínculos simbólicos.

La directiva **DirectoryIndex** especifica los archivos html o cgi por defecto durante la carga de una URL.

`DirectoryIndex index.php index.html`

Durante la carga, sin precisar el nombre del archivo html, el servidor intentará cargar index.php. Si no está, entonces index.html. En el caso contrario, es la opción **Indexes** la que determina si el contenido debe mostrarse en forma de directorio.

La directiva **AllowOverride** permite autorizar o no la interpretación de todas o algunas directivas contenidas en un archivo .htaccess en la raíz del directorio.

Solo en la versión 2.2 la directiva **Allow** indica qué clientes tienen autorización para acceder al directorio. Puede ser **all**, un dominio, una IP, una IP truncada (subred), un par red/subred, etc. La directiva **Deny** prohíbe el acceso y se utiliza de la misma manera. Se determina el orden por la directiva **Order**.

A partir de la versión 2.4 las directivas Order, Allow y Deny son remplazadas por **Require**. Las líneas asociadas del ejemplo son remplazadas por:

`Require all granted`

b. Alias

La directiva **Alias** permite crear un atajo entre el árbol lógico del sitio Web y una ruta del sistema de archivos.

`Alias /help "/usr/share/doc/html"`

En este caso, la URL `http://www.misitio.org/help` no buscará en el directorio `/var/www/html/help`, sino en `/usr/share/doc/html`.

A diferencia de las etiquetas **<Directory>**, las etiquetas **<Location>** y **</Location>** permiten aplicar directivas basadas en la URL (y no los directorios).

```
<Location /help>
  Options +All -FollowSymLinks
  Order deny, allow
  Deny from all
  Allow from .midominio.org
  # En 2.4 las tres líneas son remplazadas por:
  # Require host midominio.org
</Location>
```

6. Hosts virtuales

Un servidor Apache es capaz de gestionar varios sitios Web en un mismo servidor. Existen varios métodos. El primero se basa en los nombres (varios sitios Web para un servidor); el otro, en las direcciones IP (una dirección IP para cada sitio web). Vamos a considerar la primera versión.

La directiva **NameVirtualHost** especifica la dirección IP en la cual el servidor va a recibir las peticiones de acceso a los anfitriones virtuales.

Las etiquetas **<VirtualHost>** y **</Virtualhost>** permiten definir un anfitrión virtual.

```
NameVirtualHost 192.168.1.3

<VirtualHost 192.168.1.3>
  ServerName www2.midominio.org
  ServerAdmin webmaster@www2.midominio.org
  DocumentRoot /var/www/www2.midominio.org/
  ErrorLog logs/www2_error_log
  CustomLog logs/www2_access_log
</VirtualHost>
```

Vuelva a iniciar Apache. Con un navegador (p. ej.: Firefox) se verifica si nuestro anfitrión virtual contesta con la URL `http://www2.midominio.org` (esta dirección debe declararse en `/etc/hosts` o bien ser conocida por el servidor de nombres y apuntar al servidor correcto).

Observe, sin embargo, que si pasa por `http://www.midominio.org` ¡ya no obtiene el sitio por defecto! En efecto, cuando se declaran hosts virtuales, el primero de la lista se convierte en el anfitrión por defecto y prioritario.



Cuando se accede a un servidor, Apache busca primero una correspondencia entre el nombre de anfitrión especificado por la URL y cada `ServerName` de los anfitriones virtuales. Si no se encuentra ninguna correspondencia exacta, se elige por defecto el primer anfitrión virtual haciendo abstracción de los parámetros globales.

Añada un anfitrión virtual para el sitio principal.

```
<VirtualHost 192.168.1.3>
  ServerName    www.midominio.org
  ServerAdmin   webmaster@www.midominio.org
  DocumentRoot  /var/www/html
  ErrorLog      logs/error_log
  CustomLog     logs/access_log
</VirtualHost>
```

Cuidado, la regla anterior se aplica también con un nombre corto. Si inscribe `http://www` o `http://www2` se topará siempre con el anfitrión virtual por defecto. Hay que requerir: `http://www.midominio.org` y `http://www2.midominio.org`

Se pueden colocar en un anfitrión virtual todas las directivas deseadas (o casi).