

Maquinas virtuales para la puesta en marcha de un portal educativo. Servidor Web Apache 2.0

José Domingo Muñoz Rodríguez <josedom24@gmail.com>

28 de agosto de 2006

Resumen

En este documento vamos a explicar los pasos para instalar, configurar y poner en funcionamiento el servidor web Apache 2.0 en el servidor de la DMZ. Vamos a utilizar la técnica de *Virtual Hosting* para tener dos sitios web en el mismo servidor. Además vamos a poner en funcionamiento awstats y webalizer que son dos programas que nos permiten generar estadísticas analizando los ficheros logs de accesos a los sitios.

Índice

1. Introducción a Apache 2.0	3
2. Instalación del servidor web Apache 2.0	3
3. Configuración del servidor web Apache 2.0	3
4. Virtual Hosting en Apache 2.0	5
4.1. ¿Qué son los hosts virtuales?	5
4.2. Ejemplo de virtual hosting	6
4.3. Configuración y puesta en marcha de los sitios virtuales . . .	7
5. Analizadores de registros	9
5.1. Awstats	9
5.1.1. Instalación de awstats	10
5.1.2. Configuración de awstats	10
5.1.3. Puesta en marcha de awstats	12
5.2. Webalizer	13
5.2.1. Instalación de Webalizer	13
5.2.2. Configuración de webalizer	13
5.2.3. Puesta en marcha de webalizer	14
5.2.4. Configuración de webalizer para dominios virtuales .	14
6. Conclusión	16
7. Bibliografía	16

1. Introducción a Apache 2.0

Un servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición.

El servidor HTTP Apache es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 (RFC 2616) y la noción de sitio virtual.

La versión 2.x de Apache tiene varias mejoras clave sobre el núcleo de Apache 1.x. Estas mejoras incluyen threads de UNIX, mejor soporte para plataformas no Unix (como Windows), un nuevo API, y soporte de IPv6.

2. Instalación del servidor web Apache 2.0

Vamos a instalar el servidor Web Apache 2.0 en Debian Sarge, para ello necesitamos una conexión a internet y utilizaremos el comando `apt-get` para bajar e instalar los paquete necesarios. De esta manera el comando que debemos escribir para la instalación es el siguiente, teniendo en cuenta que lo debemos hacer como root:

```
apt-get install apache2
```

Además del paquete `apache2`, se instalarán los siguientes paquetes:

Se instalarán los siguientes paquetes extras:

```
apache2-common apache2-mpm-worker apache2-utils libapr0 ssl-cert
```

3. Configuración del servidor web Apache 2.0

En Debian Sarge los ficheros de configuración los tenemos en `/etc/apache2` siendo `apache2.conf` el fichero principal. El primer problema que nos encontramos lo observamos al reiniciar el servicio con la siguiente instrucción:

```
/etc/init.d/apache2 restart
```

El aviso que nos da al volver a iniciar el servidor es el siguiente:

```
apache2: Could not determine the server's fully qualified domain name,  
using 127.0.1.1 for ServerName
```

Es decir, Apache2 no es capaz de determinar cual es el nombre de dominio totalmente cualificado¹ con el que va a trabajar. Para solucionar esto tenemos que poner la directiva `ServerName` en el fichero `/etc/apache2/apache2.conf` con el nombre del dominio que vamos a utilizar².

Después de cualquier cambio en la configuración debemos volver a iniciar el servicio:

```
/etc/init.d/apache2 restart
```

Bueno, continuando con la configuración de Apache2 veamos los ficheros y directorios que nosotros vamos a utilizar para la configuración del servidor con Virtual Hosting:

- `/etc/apache2/apache2.conf`: Como hemos visto anteriormente es el fichero de configuración principal del servidor. Por medio de directivas incluye en este fichero los módulos necesarios y el resto de la configuración de los restantes ficheros de configuración.
- `/etc/apache2/httpd.conf`: Este fichero se mantiene por razones de compatibilidad con la primera versión de Apache. Además se recomienda que la configuración de los módulos de terceros vayan en este fichero.
- `/etc/apache2/sites-available`: Es el directorio donde se tienen los ficheros de configuración de los diferentes sitios virtuales. El fichero *default* contiene la configuración del sitio por defecto.
- `/etc/apache2/sites-enabled`: Es un directorio donde existe enlaces simbólicos a los ficheros de configuración de los sitios virtuales (en el directorio `/etc/apache2/sites-available`) que están activos.

¹El FQDN es el nombre completo de un sistema, incluyendo el 'nombre propio' del sistema. Por ejemplo, `ati` es un nombre de sistema y `ati.es` es un FQDN.

²Este nombre de dominio no tiene porque estar registrado, podemos coger cualquier nombre de dominio de ejemplo. En un próximo capítulo estudiaremos como registrar un dominio para poder acceder desde internet a nuestra página. En este documento vamos a escoger `midominio.com` como ejemplo de nombre de dominio.

- `/etc/apache2/ports.conf`: Es el fichero de configuración donde se indica el puerto por el que escucha el servidor, por defecto el 80.
- `/etc/apache2/mods-available` y `/etc/apache2/mods-enabled`: Ficheros de configuración de los módulos de apache. Estos ficheros lo veremos más detenidamente cuando instalemos PHP.

En los ficheros de configuración las líneas que comienzan con el símbolo `#` son comentarios y el servidor no las tiene en cuenta.

La configuración del servidor por defecto define un host virtual *default* cuyo directorio de trabajo es `/var/www`, por lo tanto para probar que el servidor está instalado podemos poner en un navegador `http://localhost`, con lo que se visualizará la página web guardada en el directorio `/var/www/apache2-default`³:



4. Virtual Hosting en Apache 2.0

4.1. ¿Qué son los hosts virtuales?

El término *Virtual Hosting* se refiere a hacer funcionar más de un sitio web (tales como `www.company1.com` y `www.company2.com`) en una sola máquina. Los sitios web virtuales pueden estar "basados en direcciones IP", lo que significa que cada sitio web tiene una dirección IP diferente, o "basados en nombres diferentes", lo que significa que con una sola dirección IP están funcionando sitios

³Este es el directorio por defecto configurado en el fichero `/etc/apache2/sites-available/default`, que contiene la página web principal, si quieres cambiar este directorio o simplemente no te sale la página de prueba comprueba la directiva *RedirectMatch* en el fichero de configuración.

web con diferentes nombres (de dominio). El hecho de que estén funcionando en la misma máquina física pasa completamente desapercibido para el usuario que visita esos sitios web.

4.2. Ejemplo de virtual hosting

Para explicar la técnica de Virtual Hosting vamos a trabajar con un ejemplo. Este ejemplo nos va a ayudar a comprender la configuración de Apache 2.0 para trabajar con diferentes sitios web virtuales basados en nombres diferentes, posteriormente este mismo ejemplo lo vamos a utilizar para explicar como acceder a estos sitios webs desde un servidor FTP con usuarios virtuales y además vamos a instalar dos CMS⁴ en cada uno de los sitios webs: joomla, que nos permite montar un portal y moodle que nos permite configurar una plataforma educativa. El ejemplo que vamos a implementar es el siguiente:

- Vamos a definir por virtual hosting dos sitios webs, `www.pagina1.com` y `www.pagina2.com` que van a ser servida por el mismo servidor web.
- Como hemos dicho anteriormente el directorio por defecto donde se encuentran las páginas web es `/var/www`. Nosotros no vamos a utilizar el directorio `/var`, vamos usar el directorio `/srv`⁵, donde crearemos la siguiente estructura de directorio:
 - `/srv/www/pagina1`: Donde se encontrarán los ficheros del sitio web correspondiente a `www.pagina1.com`.
 - `/srv/www/pagina2`: Donde se encontrarán los ficheros del sitio web correspondiente a `www.pagina2.com`.
- En este primer paso vamos a hacer que el propietario de estos directorios sea el usuario definido por Apache2 `www-data`. Posteriormente, cuando configuremos el servidor FTP para acceder a estos directorios crearemos dos usuarios virtuales que gestionarán cada una de las páginas, y por tanto, serán los propietarios de cada uno de los directorios.

⁴Un Sistema de gestión de contenido (Content Management System, en inglés, abreviado CMS) permite la creación y administración de contenidos principalmente en páginas web.

⁵Según el Estándar Jerarquía de Sistemas de Ficheros, donde se define los principales directorios y sus contenidos en GNU/Linux y otros sistemas operativos similares a Unix, el directorio `/srv` contiene los datos específicos de sitio servidos por el sistema.

4.3. Configuración y puesta en marcha de los sitios virtuales

Lo primero que vamos a construir es la estructura de directorios donde vamos a almacenar los sitios web. Para ello, y como root nos dirigimos al directorio /srv:

```
cd /srv
```

A continuación creamos el directorio www, cuyos subdirectorios (pagina1 y pagina2) almacenarán los ficheros de los dos sitios web:

```
mkdir www
cd www
mkdir pagina1
mkdir pagina2
cd ..
```

Por último hacemos propietario del directorio www y todos sus subdirectorios al usuario www-data que pertenece al grupo www-data:

```
chown -R www-data:www-data www
```

Veamos ahora la configuración de Apache2 para implementar los dos sitios virtuales. Nos colocamos en el directorio /etc/apache2/sites-available, hacemos una copia del fichero de configuración default que nos sirva como modelo para realizar la configuración del primer sitio virtual:

```
cd /etc/apache2/sites-available
cp default pagina1
nano pagina1
```

Los cambios que hay que realizar en este fichero son los siguientes:

1. La primera línea *NameVirtualHost* * hay que eliminarla, ya que el nombre del host virtual sólo debe aparecer en un fichero de configuración.
2. En la directiva *ServerAdmin* ponemos el usuario que administra dicha página. En nuestro caso será *usuario1@pagina1.com*.
3. En la directiva *ServerName* ponemos el nombre del dominio, en nuestro caso *www.pagina1.com*

4. En la directiva `DocumentRoot` ponemos el directorio donde se almacena este sitio web, en nuestro caso `/srv/www/paginal`
5. En la segunda etiqueta `Directory` vamos a definir la configuración de nuestro directorio por lo tanto escribimos `<Directory /srv/www/paginal/>`, la directiva `RedirectMatch` la podemos comentar para que no redirija la llamada a otro directorio.

El resto del fichero no es necesario modificarlo, por lo que el principio del fichero de configuración de la `paginal` quedaría:

```
<VirtualHost *>
    ServerAdmin usuarios1@paginal.com
    ServerName www.paginal.com
    DocumentRoot /srv/www/paginal
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /srv/www/paginal>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
        # Uncomment this directive if you want to see apache2's
        # default start page (in /apache2-default) when you go to /
        #RedirectMatch ^/$ /apache2-default/
    </Directory>
```

Una vez realizado los cambios nos queda hacer dos cosas para que funcione el nuevo sitio virtual. Primero, crear un enlace dentro del directorio `/etc/apache2/sites-enabled`, que apunte al fichero de configuración que acabamos de crear:

```
cd /etc/apache2/sites-enabled
ln -s /etc/apache2/sites-available/paginal paginal
```

A continuación volvemos a iniciar el servidor web:

```
/etc/init.d/apache2 restart
```


Antes de probar si funciona el sitio que acabamos de definir vamos a crear una pequeña página web de prueba dentro del directorio /srv/www/pagina1, esta página se debe llamar index.html:

```
<html>
<body>
<h1> Página www.pagina1.com </h1>
</body>
</html>
```

Como todavía no tenemos instalado ningún servidor DNS para hacer la resolución de nombres de dominio a dirección IP, lo tendremos que hacer en modo local, por lo que habrá que incluir en el fichero /etc/hosts la siguiente línea:

```
127.0.0.1      www.pagina1.com
```

Ya lo único que queda es probar nuestro sitio escribiendo en un navegador `http://www.pagina1.com`. Habrá que repetir el proceso para configurar el dominio `www.pagina2.com`.

5. Analizadores de registros

5.1. Awstats

AWStats (Advanced Web Statistics) es un visualizador avanzado de logs, que genera estadísticas gráficas a partir de los logs del sistema. Es software libre que se distribuye bajo la licencia GNU GPL (General Public License) y su mantenedor es Jonas Smedegaard.

Y digo, logs en general, porque AWStat es capaz de visualizar logs generados por distintos servidores, entre los que merece la pena destacar:

- Web: Apache, Tomcat, IIS
- FTP: ProFTP
- Correo SMTP: Postfix, Sendmail
- Proxy Web: Squid

AWStats soporta cualquier sistema operativo, debido a que está escrito íntegramente en Perl, y por ello es suficiente con que el servidor que lo va a interpretar tenga el módulo correspondiente instalado. Siendo así, las estadísticas de los logs se pueden generar tanto desde la línea de comandos, como un CGI y mostrar la información en una o varias páginas Web.

La página oficial es <http://awstats.sourceforge.net/>

5.1.1. Instalación de awstats

Para instalar awstats utilizamos el comando apt-get de la siguiente manera:

```
debian:~# apt-get install awstats
```

Entre los archivos instalados, vamos a destacar los siguientes:

- `/usr/share/doc/awstats`: Directorio donde se guardan archivos de ejemplo y ayuda.
- `/usr/share/awstats/lang/awstats-es.txt`: Archivo de idioma castellano. Como podemos ver es un simple archivo de texto.
- `/usr/share/awstats/plugins`: Directorio en el que se instalan los plugins que viene por defecto con el paquete.
- `/usr/share/awstats/icon`: Directorio donde se instalan los iconos (imágenes), que se van a utilizar para mostrar las estadísticas.
- `/usr/lib/cgi-bin/awstats.pl`: Es el archivo que va a generar las estadísticas a partir de los logs del sistema. Escrito en Perl.
- `/etc/cron.d/awstats`: Entrada de cron para actualizar las estadísticas periódicamente.
- `/etc/awstats/awstats.conf`: Archivo de configuración de AWStats.

5.1.2. Configuración de awstats

Nosotros vamos a realizar una configuración para poder generar estadísticas para nuestro servidor Apache.

Lo primero que vamos a configurar es el idioma. AWStats viene preconfigurado para mostrar los logs y estadísticas en inglés, aunque bien es cierto que trae un montón de archivos de idioma. Cambiarlo es muy fácil, así que vamos a ello.

Los archivos de idioma han sido instalados en el directorio `/usr/share/awstats/lang/`. Antes de nada, comprobaremos que el archivo de configuración de AWStats (normalmente en `/etc/awstats/awstats.conf`) conoce que esa es la ruta en donde debe de buscar los archivos de idioma.

```
DirLang="/usr/share/awstats/lang"
```

Para que el idioma por defecto sea el castellano, tenemos que sustituir el valor de la directiva `Lang` 'en' por 'es' (también en `/etc/awstats/awstats.conf`).

```
Lang="es"
```

La página que se genera nos muestra en la parte superior izquierda unas banderas con las que podemos elegir en cualquier momento el idioma que queramos que muestre la página. Si queremos modificar los idiomas que aparecen, su orden, etc. utilizaremos la directiva `ShowFlagLinks`, con los identificativos de los idiomas que queramos que aparezcan separados por un espacio. Veamos un ejemplo:

```
ShowFlagLinks="en es es_cat"
```

Para que las imágenes que contiene la página que se va a generar se vean correctamente, hay que indicarle a AWStats donde las ubicamos. Para ello disponemos de la directiva `DirIcons` en el archivo de configuración (`/etc/awstats/awstats.conf`). Esta directiva se caracteriza porque la ruta que indiquemos va a resultar relativa a la ubicación de la página web.

Por ello NO podemos poner:

```
DirIcons="/usr/share/awstats/icon/"
```

Porque estaría intentando acceder a `'http://my.doma.in/usr/share/awstats/icon/icon_name.png'`

La solución a este inconveniente es bien sencilla. Utilizar los alias de nuestro servidor Apache. Solamente tenemos que añadir la siguiente línea al archivo de configuración del apache (`/etc/apache2/httpd.conf`):

```
Alias /awstats-icon/ /usr/share/awstats/icon/
```

De modo que el valor que tiene que tomar la directiva es 'awstats-icon', nombre del alias que acabamos de crear:

```
DirIcons="/awstats-icon"
```

Como en un mismo host pueden "cohabitar" distintos dominios, SiteDomain es la directiva que indica a cual de esos dominios nos estamos refiriendo. En caso de tener varios dominios en un mismo host (virtual host) es recomendable generar logs independientes, es decir, uno para cada dominio. En mi caso:

```
SiteDomain="paginal.com"
```

Por último hay que indicar donde se encuentra el log de Apache2, para ello modificamos la siguiente directiva:

```
LogFile="/var/log/apache2/access.log"
```

5.1.3. Puesta en marcha de awstats

Una vez llegado a este punto, deberíamos de ser capaces de visualizar la página web por medio del enlace <http://sitedomain/cgi-bin/awstats.pl>.

Enseguida te darás cuenta de que las estadísticas están vacías. Eso es porque el archivo no lee los datos de los logs directamente sino que genera un archivo de texto y muestra los datos a partir de dicho texto. Los archivos de texto generados se encuentran en:

```
/var/lib/awstats
```

Para generar estos archivos por primera vez o actualizarlos manualmente, ejecutamos el archivo awstats.pl con los siguientes parámetros: /usr/lib/cgi-bin/awstats.pl -config=myvirtualhostname -update, para hacer esto automáticamente cuando se instala awstats se crea un tarea cron en /etc/cron.d/awstats. Esta tarea hay que modificarla para indicar la ruta correcta de los log de Apache:

```
0,10,20,30,40,50 * * * * www-data [ -  
x /usr/lib/cgi-bin/awstats.pl -a -  
f /etc/awstats/awstats.conf -a -  
r /var/log/apache2/access.log ] && /usr/lib/cgi-bin/awstats.pl -  
config=awstats -update >/dev/null
```

Es decir cada 10 minutos se va actualizar la estadística utilizando el log de apache y el fichero de configuración de awstats.

5.2. Webalizer

The Webalizer es una herramienta de análisis de servidores rápida, fiable y fácil de usar. Genera informes con información detallada sobre todos los movimientos que se producen en un servidor web.

Estos informes están en formato HTML, por lo que se pueden visualizar con cualquier navegador web, y son sencillos de configurar y totalmente personalizables.

5.2.1. Instalación de Webalizer

Para la instalación de webalizer utilizamos apt-get de la siguiente forma:

```
apt-get install webalizer
```

5.2.2. Configuración de webalizer

El fichero de configuración de webalizer es /etc/webalizer.conf, veamos los cambios que hay que realizar para poner en marcha nuestra estadística:

- Fichero donde apache esta guardando sus LOGS.

```
LogFile /var/log/apache2/access.log
```

- Directorio donde se van a generar los ficheros html que muestran la estadística⁶.

```
OutputDir /srv/www/paginal/webalizer
```

- Indicar el nombre de dominio.

```
HostName www.paginal.com
```

⁶El directorio donde vamos a indicar que se guarde la estadística ha de ser creado con anterioridad.

5.2.3. Puesta en marcha de webalizer

Para generar las estadísticas sólo es necesario ejecutar el comando:

```
webalizer
```

Para ver las estadísticas sólo hay que indicar como dirección URL:

```
http://www.paginal.com/webalizer
```

Para automatizar esta tarea, podemos crear una tarea crontab que ejecute el comando periódicamente cada cierto tiempo, para ello editamos el crontab del root con la instrucción crontab -e y añadimos la siguiente línea:

```
0,10,20,30,40,50 * * * * webalizer > /dev/null
```

5.2.4. Configuración de webalizer para dominios virtuales

Lo que hemos explicado hasta ahora es la configuración de webalizer para un sólo dominio. En el ejemplo que estamos siguiendo tenemos dos hosts virtuales. Vamos a explicar la forma para que tengamos una estadística para cada dominio.

Lo primero que hay que hacer es guardar en fichero de logs distintos cada uno de los accesos a los distintos sitios virtuales, para ello hay que modificar la configuración de los sitios virtuales y añadir una nueva directiva. Por ejemplo para paginal.com habría que añadir la siguiente línea:

```
<VirtualHost *>
    ServerAdmin usuarios1@paginal.com
    ServerName www.paginal.com
    DocumentRoot /srv/www/paginal
    CustomLog /var/log/apache/paginal-access.log common
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /srv/www/paginal>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
```

```

        allow from all
        # Uncomment this directive if you want to see apache2's
        # default start page (in /apache2-default) when you go to /
        #RedirectMatch ^/$ /apache2-default/
    </Directory>

```

Este cambio hay que hacerlo también con el host virtual pagina2.com, y posteriormente reiniciar el servidor web.

A continuación vamos a crear un directorio donde vamos a guardar los ficheros de configuración de webalizer para cada uno de los dominios:

```
mkdir /srv/www/estadistica/
```

Del mismo modo creamos los directorios donde vamos a guardar los ficheros html generados que muestran las estadísticas:

```
mkdir /srv/www/pagina1/webalizer
mkdir /srv/www/pagina2/webalizer
```

A continuación copiamos el fichero de configuración de webalizer en el directorio que hemos creado anteriormente pero con nombre distintos para cada uno de los dominios:

```
cp /etc/webalizer.conf /srv/www/estadistica/pagina1.conf
cp /etc/webalizer.conf /srv/www/estadistica/pagina2.conf
```

Dentro de cada uno de ellos se deben hacer los siguientes cambios:

- LogFile /var/log/apache2/<FICHERO_LOG>
- OutputDir /srv/www/<DOMINIO>/webalizer
- HostName <DOMINIO>

Por ejemplo para el dominio pagina1.com, los cambios serían los siguientes:

```
LogFile /var/log/apache2/pagina1-access.log
OutputDir /srv/www/pagina1/webalizer
HostName pagina1.com
```

Para ejecutar webalizer con cada una de las configuraciones lo más fácil es hacer un script estadistica.sh:

```
#!/bin/bash
webalizer -c /srv/www/estadistica/pagina1.conf
webalizer -c /srv/www/estadistica/pagina2.conf
```

Por último, para ejecutar este script lo podemos meter en el crontab:

```
0,10,20,30,40,50 * * * * estadistica.sh > /dev/null
```

Utilizando los conceptos que hemos aprendido en este último punto podemos hacer un paralelismo y configurar awstats para varios virtual hosts.

6. Conclusión

En este capítulo hemos explicado la instalación y configuración del servidor web Apache2, fijándonos sobre todo en la configuración y puesta de marcha de sitios web virtuales mediante *Virtual Hosting*. En los dos últimos apartados hemos visto la instalación, configuración y puesta en marcha de los analizadores de registros más utilizados: awstats y webalizer.

7. Bibliografía

Referencias

- [1] Seminario de configuración de Apache 2:
www.informatica.us.es/~ramon/articulos/SeminarioApache2.pdf
- [2] Guía de Apache 2:
<http://www.osmosislatina.com/apache2/index.htm>
- [3] Documentación sobre Hosting Virtual en Apache:
<http://httpd.apache.org/docs/2.0/es/vhosts/>
- [4] Configurando hosts virtuales o como tener http://bulma.midominio.com:
<http://bulma.net/body.phtml?nIdNoticia=644>

- [5] AWStats:
<http://fermat.movimage.com/docs/awstats.html>
- [6] Estadísticas web con webalizer:
<http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=196>