



Autor: Kitai Fecha: 16-1-2007

<http://www.apache-es.org>

Cómo instalar Apache, primeros pasos para montar un WebSite

Este documento trata sobre los primeros pasos que necesitamos seguir para montar un Sitio Web con Apache. Vamos a cubrir la instalación, compilando o desde paquete, y las directivas necesarias para dar servicio a varios sites.

La razón de este documento, es que la mayoría de las preguntas que me realizan por IRC o en los foros, son de instalación básica y conceptos muy básicos de Apache. Así que pensé que sería una buena manera de continuar, antes de meternos en configuraciones más complicadas.

La versión a la que vamos a hacer referencia, es la 2.2.4 de Apache.

1- Instalación

Existen varias maneras de instalar Apache, dependiendo del Sistema Operativo que utilizemos.

- Compilando desde el código fuente. Con este tipo de instalación se pueden conseguir pequeñas mejoras de rendimiento respecto a las versiones de las distribuciones. Consiste en generar unos ficheros ejecutables “compilando” el código fuente de Apache.
- Instalación desde paquete. Las diferentes distribuciones de Linux proporcionan paquetes con versiones operativas de Apache. Tan sólo hace falta seguir las instrucciones de instalación de cada fabricante, y en muy poco tiempo se puede tener Apache funcionando.

Personalmente, recomiendo compilar Apache. Es algo más engorroso a la hora de actualizarlo, debido a que cuando sale una nueva versión hay que andar recompilando de nuevo. Pero más o menos, vais a tener los mismos problemas en las actualizaciones tanto si compiláis, como si utilizáis un paquete RPM o DEB.

Compilación de Apache en Linux / Solaris

Vamos a explicar las diferentes opciones de compilación, así como consejos para sobrellevar las actualizaciones posteriores.

Antes de nada, en Linux y en solaris se necesitan los siguientes paquetes para compilar:

- GCC (el compilador de C de GNU)
- AutoMake
- AutoConf
- Tar
- Gzip o Gunzip

En Linux las descargáis de vuestro repositorio de Software favorito, aunque es bien posible que lo tengáis todo ya. En Solaris, de la web de <http://www.sunfreeware.com/>.

- Descargamos las fuentes de Apache.
 - o <http://apache.rediris.es/httpd/httpd-2.2.4.tar.gz>
- Creamos un directorio donde queramos, para descomprimirlo y ejecutamos:
 - o `tar zxf httpd-2.2.4.tar.gz`
(z para descomprimir, x para Extraer, F porque estamos ejecutando Tar sobre un fichero)

El script para compilar Apache se llama “configure” y está dentro del nuevo directorio recién creado al descomprimir.

Tiene MUCHAS opciones diferentes, como se puede ver al hacer un “./configure --help”.

Lo mejor es crear un fichero con la línea que utilizamos para compilar Apache. Por ejemplo, creamos un ficherillo llamado COMPILAR, que será el que luego ejecutemos para compilar Apache:

```
./configure --enable-so --with-mpm=worker --prefix=/usr/local/apache
```

Con esa línea tendremos un Apache **listo** para ser compilado:

- en el que podremos cargar otros módulos posteriormente si queremos, con el “--enable-so”
- instalado en el directorio /usr/local/apache, por la opción “--prefix”
- y con soporte para hilos para mejorar el rendimiento si tenemos dos o más procesadores. (--with-mpm=worker). Si queremos que Apache no utilice hilos y se comporte como las anteriores versiones, basta con poner aquí --with-mpm=prefork.

¿Cómo compilamos?

Ejecutamos “make”.

¿Cómo lo instalamos después de compilarlo?

Ejecutamos “make install”.

Es posible que de errores de compilación. Suelen ser por problemas de librerías que faltan, sobre todo si instalamos algún módulo no incluido en la distribución original.

Cualquier duda de compilación la podéis plantear en los foros de <http://www.apache-es.org>.

Instalar desde paquete (RPM, DEB, o lo que sea)

La mayoría de las distribuciones de Linux tienen su propia instalación, y cada una a su manera. Normalmente, suelen dejar los ficheros de configuración en /etc/apache o /etc/httpd, y los ficheros de log en /var/log/httpd o /var/log/apache.

El directorio base de contenido suelen meterlo en /var/www/html.

Debian

Entramos como root, y ejecutamos:

“apt-get install apache-common apache”

Si queremos instalar MySQL o PHP:

```
“apt-get install mysql-common mysql-client mysql-server”
```

```
“apt-get install php4 php4-cgi php4-mysql”
```

(No he mirado a ver si existe ya PHP Versión 5 en Debian, pero algún día de estos deberían actualizar).

Ficheros de configuración: /etc/apache2

Ficheros de log: /var/log/apache

Ficheros de módulos: /etc/apache2/mods-available y /etc/apache2/mods-enabled

Para habilitar un módulo usamos el comando “a2enmod”, y para deshabilitarlo “a2dismod”.

Estos comandos nos sacan una salida por pantalla tal que así:

```
Which module would you like to enable?

Your choices are: actions asis auth_anon auth_dbm auth_digest
auth_ldap cache cern_meta cgi cgid dav dav_fs deflate disk_cache
expires ext_filter file_cache geoip headers imap include info ldap
mem_cache mime_magic php4 proxy proxy_connect proxy_ftp proxy_http
rewrite rpaf ruby spelling ssl suexec unique_id userdir usertrack
vhost_alias
Module name?
```

Para instalar módulos extra, también se hace con apt-get:

```
“apt-get install mod-jk”
```

Lo siguiente ya es configurar Apache, dando los primeros pasos que describiré en el siguiente capítulo de esta guía.

RedHat, CentOS, Fedora

Más o menos es similar en estas tres distribuciones.

No está mal empezar por mirarse el manual, que, por una vez, lo hay, en la Web de RedHat.
<http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/es/ref-guide/ch-httpd.html>

Se puede instalar utilizando el comando “up2date” de Red Hat Enterprise y CentOS. Personalmente, siempre utilizo el gestor de paquetes “Yum” para estas instalaciones.

```
“yum install httpd httpd-devel httpd-suexec”
```

```
“yum install php php-mysql”
```

```
“yum install mysql mysql-server mysql-devel “
```

Los ficheros de configuración están en : /etc/httpd

Los ficheros de log en : /var/log/httpd

Los ficheros de módulos en : /usr/lib/httpd/modules

En RedHat Enterprise, la versión de apache que instala es la 2.0. En Fedora, el PHP que te instala es la versión 5.

Posteriormente, hay que activar el servicio para que se inicie al arrancarse el sistema. Esto se hace con el comando chkconfig.

```
“chkconfig httpd on”
```

```
“chkconfig mysql on”
```

Y luego ya seguimos con la configuración de Apache , que detallo en el siguiente capítulo.

2- Configuración

Vamos a proceder ahora a explicar la configuración base de cualquier Apache, y esos primeros pasos que hay que dar para configurar un Apache.

Definiciones

Host, nombre de host: www.pepitogrillo.com, www.miservidor.com, etc...

IP: el número que identifica un interfaz de red. 192.168.1.100, 62.151.2.8, etc...

Log : es un fichero de registro en donde se almacena texto.

Módulo: una extensión para Apache. Sirven para aumentar las capacidades del servidor Web. Ejemplos de módulos, ...PHP, LDAP, mod_security...

HTTP: el nombre del protocolo que utiliza Apache para servir página Web.

HTTPS: lo mismo, pero va cifrado para mejorar la seguridad de la transmisión.

El fichero de configuración

El fichero base en donde está la configuración principal de Apache, se llama httpd.conf.

La configuración de Apache tiene varios bloques. Parte se refieren a la configuración de los módulos que extienden las capacidades de Apache, y parte se refieren a la definición de los host que vamos a servir.

Voy a partir de una configuración BASE en un servidor CentOS (suele ser más o menos igual esta parte en toda las distribuciones). Y os voy a indicar qué es lo que se ha de retocar nada más instalarse un servidor Web. No voy a entrar a temas de seguridad serios y configuración de módulos extra, voy a ir directo a explicar el típico tema de “necesito un apache funcionando en 5 minutos”.

Y para abreviar, también elimino los comentarios.

Este es un fichero httpd.conf de ejemplo:

```
ServerTokens Prod

# Lo queremos a Prod, para impedir que el resto de la gente sepa que
# versión de apache estamos utilizando

ServerRoot "/etc/httpd"

PidFile run/httpd.pid

Timeout 120

KeepAlive On

MaxKeepAliveRequests 10000

KeepAliveTimeout 4

# Activamos el KeepAlive, y le damos unos valores sensatos para un
# rendimiento decente

<IfModule prefork.c>
StartServers      8
MinSpareServers   5
MaxSpareServers   20
ServerLimit       256
MaxClients        256
MaxRequestsPerChild 4000
</IfModule>

<IfModule worker.c>
StartServers      2
MaxClients        150
MinSpareThreads   25
MaxSpareThreads   75
ThreadsPerChild   25
MaxRequestsPerChild 0
</IfModule>

Listen 192.168.1.100:80

Listen 192.168.1.101:81

# Ponemos Apache a escuchar por la IP en la que vamos a servir el contenido

LoadModule access_module modules/mod_access.so
LoadModule auth_module modules/mod_auth.so
#LoadModule auth_anon_module modules/mod_auth_anon.so
#LoadModule auth_dbm_module modules/mod_auth_dbm.so
#LoadModule auth_digest_module modules/mod_auth_digest.so
#LoadModule ldap_module modules/mod_ldap.so
#LoadModule auth_ldap_module modules/mod_auth_ldap.so
LoadModule include_module modules/mod_include.so
LoadModule log_config_module modules/mod_log_config.so
```

```
LoadModule env_module modules/mod_env.so
LoadModule mime_magic_module modules/mod_mime_magic.so
LoadModule cern_meta_module modules/mod_cern_meta.so
LoadModule expires_module modules/mod_expires.so
#LoadModule deflate_module modules/mod_deflate.so
LoadModule headers_module modules/mod_headers.so
#LoadModule usertrack_module modules/mod_usertrack.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule mime_module modules/mod_mime.so
#LoadModule dav_module modules/mod_dav.so
#LoadModule status_module modules/mod_status.so
#LoadModule autoindex_module modules/mod_autoindex.so
#LoadModule asis_module modules/mod_asis.so
#LoadModule info_module modules/mod_info.so
#LoadModule dav_fs_module modules/mod_dav_fs.so
#LoadModule vhost_alias_module modules/mod_vhost_alias.so
LoadModule negotiation_module modules/mod_negotiation.so
LoadModule dir_module modules/mod_dir.so
#LoadModule imap_module modules/mod_imap.so
LoadModule actions_module modules/mod_actions.so
#LoadModule speling_module modules/mod_speling.so
#LoadModule userdir_module modules/mod_userdir.so
LoadModule alias_module modules/mod_alias.so
LoadModule rewrite_module modules/mod_rewrite.so
#LoadModule proxy_module modules/mod_proxy.so
#LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
#LoadModule proxy_http_module modules/mod_proxy_http.so
#LoadModule proxy_connect_module modules/mod_proxy_connect.so
#LoadModule cache_module modules/mod_cache.so
#LoadModule suexec_module modules/mod_suexec.so
#LoadModule disk_cache_module modules/mod_disk_cache.so
#LoadModule file_cache_module modules/mod_file_cache.so
#LoadModule mem_cache_module modules/mod_mem_cache.so
LoadModule cgi_module modules/mod_cgi.so

# Eliminamos módulos que no vamos a utilizar.

Include conf.d/*.conf

User apache

Group apache

ServerAdmin miemail@miservidordecorreo.com

UseCanonicalName Off

DocumentRoot "/var/www/html"

<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>

DirectoryIndex index.html index.php index.htm

# El orden de búsqueda de documentos de apache Primero buscará el
```

index.html, luego el index.php y luego el index.htm

```
AccessFileName .htaccess
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>
TypesConfig /etc/mime.types
DefaultType text/plain
<IfModule mod_mime_magic.c>
#   MIMEMagicFile /usr/share/magic.mime
    MIMEMagicFile conf/magic
</IfModule>
HostnameLookups Off
ErrorLog logs/error_log
LogLevel warn
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
CustomLog logs/access_log combined
ServerSignature OFF
# Para evitar mostrar la versión de apache en los ficheros de Error
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>
AddDefaultCharset UTF-8
AddCharset ISO-8859-1 .iso8859-1 .latin1
AddCharset ISO-8859-2 .iso8859-2 .latin2 .cen
AddCharset ISO-8859-3 .iso8859-3 .latin3
AddCharset ISO-8859-4 .iso8859-4 .latin4
AddCharset ISO-8859-5 .iso8859-5 .latin5 .cyr .iso-ru
AddCharset ISO-8859-6 .iso8859-6 .latin6 .arb
AddCharset ISO-8859-7 .iso8859-7 .latin7 .grk
AddCharset ISO-8859-8 .iso8859-8 .latin8 .heb
AddCharset ISO-8859-9 .iso8859-9 .latin9 .trk
AddCharset ISO-2022-JP .iso2022-jp .jis
AddCharset ISO-2022-KR .iso2022-kr .kis
AddCharset ISO-2022-CN .iso2022-cn .cis
AddCharset Big5 .Big5 .big5
# For russian, more than one charset is used (depends on client, mostly):
AddCharset WINDOWS-1251 .cp-1251 .win-1251
AddCharset CP866 .cp866
AddCharset KOI8-r .koi8-r .koi8-ru
AddCharset KOI8-ru .koi8-uk .ua
AddCharset ISO-10646-UCS-2 .ucs2
AddCharset ISO-10646-UCS-4 .ucs4
AddCharset UTF-8 .utf8
AddCharset GB2312 .gb2312 .gb
AddCharset utf-7 .utf7
AddCharset utf-8 .utf8
AddCharset big5 .big5 .b5
AddCharset EUC-TW .euc-tw
AddCharset EUC-JP .euc-jp
AddCharset EUC-KR .euc-kr
AddCharset shift_jis .sjis
```

```
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
AddType text/html .shtml
AddOutputFilter INCLUDES .shtml
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0
BrowserMatch "Microsoft Data Access Internet Publishing Provider"
redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
BrowserMatch "^WebDAVFS/1.[012]" redirect-carefully
BrowserMatch "^gnome-vfs" redirect-carefully
```

Ahora viene la parte de definición de los VirtualHost. Cada bloque de Virtualhost define un host que vamos a servir. Es decir, si queremos servir dos páginas web diferentes, una con el nombre de host “pepito.com” y otra con el nombre de host “jose.com”, tendremos un virtualhost para pepito.com, y otro para jose.com.

Un mismo VirtualHost puede servir para varios nombres de host, de manera que pepito.com, pudiera servir pepe.com, pepitogrillo.com, pepaza.org, etc...

Es importante asociar los virtualhost a la dirección IP por la cual lo vamos a servir. ¿Por qué? Porque los VirtualHost han de ir colocados de la siguiente manera.

```
NameVirtualHost ip.ip.ip.ip:puerto
```

```
<VirtualHost ip.ip.ip.ip:puerto>
```

```
...
```

```
</VirtualHost>
```

```
NameVirtualHost ip2.ip2.ip2.ip2:puerto
```

```
<VirtualHost ip2.ip2.ip2.ip2:Puerto>
```

```
</VirtualHost>
```

Vamos agrupando los VirtualHost por IP. Si sólo tenemos una IP, tendremos una directiva NamevirtualHost, y debajo, todos los Virtualhosts que vayan en esa IP.

Apache funciona de la siguiente manera a la hora de decidir qué VirtualHost sirve para responder a la petición del cliente:

- Empezará por el VirtualHost que esté más abajo, y avanzará hacia arriba. De manera que el VirtualHost que definamos PRIMERO, será el VirtualHost por defecto de nuestra lista (ver <http://www.apache-es.org/index.php/2006/11/16/truco-creacion-de-virtualhost-falso-para-mejorar-la-seguridad/>)
- Irá comparando, primero el nombre de host que haya en la directiva SERVERNAME, y luego en las directivas SERVERALIAS, que sirven para definir nombres de hosts alternativos.
- Si no encuentra ninguno, servirá el primer VirtualHost o uno que tengamos definido con el nombre _default.

Vamos a poner un ejemplo. Queremos servir 2 páginas Web. Una llamada apache-es.org, y otra llamada mysql-es.org. Cada una escucha por su IP.

```
NameVirtualHost 192.168.1.101:80

<VirtualHost 192.168.1.101:80>

ServerName apache-es.org

ServerAlias *.apache-es.org

DocumentRoot /usr/local/apache-es/docs

ErrorLog logs/apache-es.error.log

CustomLog logs/apache-es.access.log combined

ServerAdmin admin@apache-es.org

</VirtualHost>

NameVirtualHost 192.168.1.102:80

<VirtualHost 192.168.1.102:80>

ServerName mysql-es.org

ServerAlias www.mysql-es.org www2.mysql-es.org admin-mysql-es.org

DocumentRoot /usr/local/mysql-es/docs

ErrorLog logs/mysql-es.error.log

CustomLog logs/mysql-es.access.log combined

ServerAdmin admin@mysql-es.org

</VirtualHost>
```

Cada directiva sirve para lo siguiente:

- NameVirtualHost: define un grupo de virtualhost bajo una IP.

- VirtualHost: inicia o termina un bloque para definir un virtualhost
- ServerName: el nombre del host que vamos a servir, www.mipaginaweb.com, servidor.com, etc...
- ServerAlias: para que ese virtualhost responda a varios nombres de hosts. www2.mipaginaweb.com, o *.mipaginaweb.com.
- DocumentRoot : indica el directorio adonde Apache ha de ir a buscar los ficheros para servirlos.
- ErrorLog: define el fichero en donde vamos a almacenar los errores que de Apache al servir páginas.
- CustomLog: define un fichero en donde vamos a almacenar los datos de los clientes que acceden a nuestro site.
- ServerAdmin: un e-mail que aparece en la página de Error indicativo de quién administra ese website.

La opción B

Existe otra opción para definir los VirtualHost. En muchas distribuciones, encontraremos un directorio llamado /etc/httpd/conf/vhosts.d. Dentro hay ficheros en los cuales están definidos los VirtualHost. Cada fichero, contiene bloques de configuración de Apache.

Esto lo detectaréis si en vuestro fichero httpd.conf, si al final veis una directiva del tipo

```
Include conf/vhosts.d/*.conf
```

Apache buscará en el directorio vhosts.d, ficheros que terminen en .conf para incluirlos en la configuración.

¿Cuál carga apache primero?

Alfabéticamente. Si tenemos dos ficheros:

[a.com.conf](#)

[b.com.conf](#)

Cargará antes [a.com.conf](#).

Dentro de esos ficheros van bloques de configuración <VirtualHost...> de la misma manera que los definimos antes.

Conclusión

Una vez hemos configurado el fichero httpd.conf, y hemos metido los virtualHost que vamos a servir, tan sólo nos resta configurar los módulos extra, si vamos a servir PHP, Java, etc...

Meter los documentos en los directorios definidos por la directiva DocumentRoot.

Y arrancar el Apache con /etc/init.d/apache start o /etc/init.d/httpd start.

Necesitamos ser usuario root para arrancarlo.

¿Crees que falta algo en este documento?

¿Crees que sobra algo?

Por favor, enviádmelo a través de mi página web: <http://www.apache-es.org>

Cientos de usuarios os estarán agradecidos, si podemos mejorar este documento.

Kitai