

## Servidor web con Debian 6.0 : Apache 2, Suexec, Fastcgi y Virtual Host

Hoy vamos con un pequeño tutorial de como montar un **servidor web** básico basado en **Debian 6.0 squeeze**. Hace unos meses expliqué como crear un servidor para Ubuntu, el proceso es bastante similiar aunque esta vez he cambiado algunas cosas y añadido otras. La idea del otro tutorial era crear un pequeño servidor para usar en un entorno local, esta vez me centro en un servidor remoto para alojar páginas web "reales". Partiremos de una máquina con Debian 6.0 ya instalado. Lo primero que tenemos que hacer, es iniciar un **terminal**, bien localmente o bien remotamente, a través de **SSH** (podemos usar un terminal de Linux para conectarnos mediante SSH a la máquina remota o bien usar [Putty/Kitty](#) en Windows).



Lo primero, vamos a **actualizar** todos los paquetes de nuestro Debian 6.0 squeeze:

```
apt-get update && apt-get upgrade
```

Con esto actualizaremos la lista de paquetes de los repositorios e instalaremos las actualizaciones necesarias.

### Instalando Apache 2:

Como servidor web instalaremos **Apache 2**. Habilitaremos así mismo los módulos **php-cgi**, el módulo **fast-cgi** para debian y el módulo **suexec**. Con esto conseguiremos tener en la misma máquina varios usuarios ejecutando php, obteniendo más seguridad y versatilidad que si todo el

código php es lanzado unicamente por el usuario de apache. Para instalar el servidor Apache junto con los módulos necesarios, escribimos:

```
apt-get install apache2-mpm-worker libapache2-mod-fcgid apache2-suexec-custom php5-cgi
```

Gracias a que el módulo apache2-suexec-custom incorpora un archivo de **configuración** a diferencia del suexec original, podremos cambiar ciertas opciones sin necesidad de recompilarlo. Por defecto nos cogerá como raíz `"/var/www"`. En mi caso, prefiero tener todo en **/home**, ya que tengo una partición donde la tengo montada, siendo más fácil un posterior traslado o respaldo del servidor.

La carpeta de cada usuario, contará con un subdirectorío en su interior que será donde se aloje la página web. Por lo tanto, la ruta sería `"/home/usuario/www"`, pudiendo substituir el directorio `www` por el que cada uno prefiera. Para evitar tener que crear este directorio cada vez que añadamos un usuario, podemos crear uno del mismo nombre en **`"/etc/skel/"`**, ya que todo el contenido de esta carpeta se añadirá a la de cada usuario cuando lo creamos (debe añadirse con el comando **`"adduser"`** y no `"useradd"`).

```
mkdir /etc/skel/www
```

Ahora deberemos habilitar el módulo suexec

```
a2enmod fcgid suexec actions
```

También hay que editar el archivo de **configuración** de este **módulo** para especificar los directorios a usar.

```
nano /etc/apache2/suexec/www-data
```

y cambiamos las dos primeras líneas para que quede así:

```
/home  
www
```

Hay que crear un archivo de configuración para decirle al módulo fast-cgi como debe manejar los archivos `.php`. Dicho archivo irá en `/etc/apache2/conf.d/` con cualquier nombre que queramos acabado en `.conf`. Para hacerlo más descriptivo, le llamaremos `mod-fast-cgi-php5.conf`:

```
nano /etc/apache2/conf.d/mod-fast-cgi-php5.conf
```

En el interior añadimos las siguientes líneas y guardamos:

```
AddType application/x-httpd-php .phpAddHandler php-fcgi .php  
Action php-fcgi /fcgi-bin/php5-fcgi
```

```
Alias /fcgi-bin/ /home/www-data/
```

```
SetHandler fcgid-script  
Options +ExecCGI
```

Si te has fijado en las líneas de configuración anteriores, la carpeta `www-data` no existe dentro de `/home`. Debemos crearla y hacer `chown` con el usuario y grupo `www-data`.

```
cd /home  
mkdir www-data  
chown www-data:www-data www-data
```

Ahora debemos crear dentro de esta carpeta el archivo `php5-fcgi`:

```
nano /home/www-data/php5-fcgi
```

Dentro pegamos el contenido siguiente:

```
#!/bin/sh  
exec /usr/bin/php5-cgi
```

Ahora solo nos queda hacerlo ejecutable y hacer `chown` con usuario y grupo `www-data`:

```
chmod +x /home/www-data/php5-fcgi  
chown www-data:www-data /home/www-data/php5-fcgi
```

Con todo esto ya tenemos listo nuestro **Apache** para añadir hosts virtuales. A continuación os pongo un ejemplo de como añadir un host virtual y una configuración básica para el mismo. Creamos un nuevo archivo de configuración de host virtual en `/etc/apache2/sites-available`. Partimos de que hemos creado un usuario con el nombre "dominio"

```
nano /etc/apache2/sites-available/dominio
```

Dentro de este archivo escribimos lo siguiente:

```
ServerAdmin admin@dominio.com
ServerName dominio.com
ServerAlias www.dominio.comAlias /fcgi-bin/ /home/www-data/dominio/
SuexecUserGroup dominio dominio
```

```
DocumentRoot /home/dominio/www/
```

```
Options Indexes FollowSymLinks
AllowOverride None
Order allow,deny
allow from all
```

```
ErrorLog /var/log/apache2/error.log
LogLevel warn
CustomLog /var/log/apache2/access.log combined
```

Ya solo nos queda añadir una carpeta a `/home/www-data` con el nombre del usuario, copiar dentro el archivo `php5-fcgi` que creamos antes, y hacer `chown` con el usuario en cuestión:

```
mkdir /home/www-data/dominio
cp /home/www-data/php5-cgi /home/www-data/dominio/
chown -R dominio:dominio /home/www-data/dominio
```

Finalmente tenemos que habilitar el virtual host. Para ello usaremos `a2ensite`.

```
a2ensite dominio
```

Reiniciamos apache:

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

Y con esto tendríamos nuestro servidor Apache 2 funcionando con suexec, fastcgi y un host virtual. Los archivos de la web de este usuario irían en /home/dominio/www.

En el próximo artículo, explicaré como instalar y configurar el servidor dns Bind9 en nuestro servidor Debian 6.0.