

# ProFTPD - un servidor FTP para profesionales

Daniel Esteban Coletti

## 1. Introducción

El servicio de ftp es uno de los más viejos y utilizados para la transferencia de archivos, la mayoría de los servidores webs lo necesitan y lo utilizan de alguna u otra forma, para actualizar sus páginas estáticas, subir imágenes o para permitir la descarga de archivos a sus visitantes. Sea cual fuere el verdadero requerimiento, un servidor ftp siempre es necesario o en el peor de los casos, útil.

ProFTPD debe ser uno de los mejores servidores ftp que existen para Linux (también se lo puede utilizar en otros sistemas operativos). Es seguro, sumamente flexible, modular y fácil de configurar. Entre todas las características que tiene, las más rescatables son: permite autenticar usuarios con casi cualquier cosa, se pueden utilizar servidores virtuales de ftp, se pueden tener múltiples servidores brindando servicio de ftp anónimo, es modular (lo que permite extender su funcionalidad ampliamente), un usuario con acceso por ftp únicamente no requiere de una configuración especial, y su código es libre (esta licenciado bajo GPL).

## 2. Instalación

La instalación es igual a todas las instalaciones comunes.

RPMS:

```
#rpm -Uvh proftpd-1.2.4-i386.rpm
```

DEBS:

```
#apt-get install proftpd
```

FUENTES:

```
#tar xzvf proftpd-1.2.4.tar.gz
#cd proftpd-1.2.4
#./configure
#make && make install
```

Como siempre, se recomienda hacer la instalación con paquetes que contengan binarios, de esta forma no es necesario llevar adelante la compilación de los fuentes y es mucho más fácil mantener estas aplicaciones actualizadas en el futuro (punto extremadamente importante si el servicio de FTP va a brindarse en internet o cualquier red insegura).

### 3. Configuración

ProFTPD trabaja con un único archivo de configuración `/etc/proftpd.conf`, la sintaxis que utiliza es parecida a la utilizada en apache, por lo que el usuario acostumbrado a configurar apache le resultará familiar.

Lo primero que el administrador deberá determinar es cómo va a ejecutarse este servicio, si lo hará bajo `inetd` o en forma independiente (como un proceso más). La diferencia entre ambos métodos es mayormente la velocidad de ejecución que se obtendrá del servicio y la carga que se le dará al equipo. Utilizando `inetd` el equipo iniciará un proceso nuevo por cada pedido de ftp que llegue al equipo, mientras que ejecutándose en forma independiente, el proceso ya está inicializado por lo que no hay necesidad ejecutar un nuevo programa (simplemente se clona el proceso, generando una nueva instancia del mismo para atender al cliente ftp que inició el requerimiento).

El parámetro dentro del archivo de configuración que determina el método con que va a ejecutarse el `proftpd` es "ServerType", si este está en "standalone" el servicio correrá como un proceso independiente.

Por otro lado si se especifica "inetd" se deberá agregar al `inetd.conf` una configuración extra y será el `inetd` el proceso encargado de inicializar al `proftpd` cada vez que llegue un requerimiento de ftp al equipo.

En el caso de que se elija ejecutar `proftpd` vía `inetd`, la siguiente línea de configuración se deberá especificar en el archivo `/etc/inetd.conf`:

```
ftp      stream  tcp      nowait  root    /usr/sbin/tcpd  /usr/sbin/proftpd
```

El siguiente es un ejemplo del archivo de configuración `/etc/proftpd.conf`, más adelante continuaremos con configuraciones más divertidas, que suelen ser muy útiles pero que no siempre están documentadas.

```
ServerName "Mi servidor FTP"
ServerType standalone
DeferWelcome off

DefaultServer on
ShowSymlinks on
AllowOverwrite on

Port 21
MaxInstances 30

User nobody
Group nogroup

<Directory /*>
<Anonymous /home/ftp/>
</Anonymous>
Umask 022
AllowOverwrite on
</Directory>
```

Si bien muchas de las opciones descriptas hablan por sí solas, hay algunas que no tanto. "DeferWelcome" por ejemplo, determina si el servidor muestra el mensaje de bienvenida antes o después de que el usuario se haya

autenticado. "ShowSymlinks" determinará si se muestran los enlaces simbólicos como tales o no (en el caso de que no, se los despliega como archivos comunes o directorios comunes, sin poner flechas ni nada semejante que los delate como tales).

Como en apache, el proftpd puede definir opciones por directorio (el bloque encerrado por

```
<Directory [directorio]> </Directory>
```

determina la opciones que recibirá este directorio), incluso hasta se pueden poner archivos tipo .htaccess de apache (que en proftpd se llaman .ftpassess). El bloque comenzado con

```
<Anonymous [directorio]>
```

le indica a proftpd que este directorio deberá ser tratado como un directorio de acceso anónimo, esto en la práctica podrá ser verdad o no, pero proftpd no permitirá que el cliente suba archivos dentro de este directorio o sus subdirectorios (a menos que se especifique lo contrario).

La opción "Umask 022" determina con qué permisos se creará un archivo o directorio (en caso de que se configure el servidor para que deje a los usuarios subir archivos vía ftp). La opción "MaxInstances [cantidad]" limita la cantidad de instancias que podrá tener este servidor ftp, esto es importante limitarlo ya que sino el servidor podrá verse afectado por un ataque de negación de servicio (DOS). Por último la opción "AllowOverwrite" permitirá o no sobreescribir archivos existentes.

## 4. Armandos servidores virtuales de ftp

Los servidores virtuales son aquellos que están definidos dentro de un equipo real. La idea es tener varios servidores, que compartan la dirección de IP pública, y que estén todos definidos en un mismo servidor. Desgraciadamente, el protocolo ftp no fue pensado para hacer esto, lo que hace imposible que el proftpd detecte que "nombre" escribió el usuario que esta accediendo a este servidor (p.e.: ftp.xtech.com.ar). Por lo tanto se pueden hacer dos cosas únicamente, utilizar varias direcciones de IP o escuchar en la misma dirección de IP, pero en diferentes puertos. Esta última opción solo puede utilizarse si el cliente sabe de antemano a qué puerto debe comunicarse.

El siguiente es un ejemplo de como definir servidores virtuales de ftp:

```
<Global>
DisplayLogin /home/ftp/welcome.msg
MaxClients 30
MaxClientsPerHost 3 "Lo siento, solamente se permite acceder de a %m clientes a la vez"
</Global>

<VirtualHost ftp.xtech.com.ar>
ServerName "Servidor FTP de XTech"
MaxClients 15
DefaultRoot /var/www/xtech
<Anonymous /var/www/xtech/pub>
User ftp
Group ftp
UserAlias anonymous ftp
</Anonymous>
</VirtualHost>
```

```
<VirtualHost ftp.linux.org.ar>
ServerName "Servidor FTP del LUGAr"
MaxClients 50
DefaultRoot /var/www/LUGAr
<Anonymous /var/www/LUGAr/pub>
User ftp
Group ftp
UserAlias anonymous ftp
</Anonymous>
</VirtualHost>

<VirtualHost ftp.cafelug.org.ar>
ServerName "Servidor FTP del CaFeLUG"
Port 2121
DefaultRoot /var/www/cafelug
<Anonymous /var/www/cafelug/pub>
User ftp
Group ftp
UserAlias anonymous ftp
</Anonymous>
</VirtualHost>
```

#### La sección

```
<Global>
```

se utiliza para definir opciones globales, esto se va a aplicar a todos los servidores virtuales a menos que redefinan la opción. La opción "MaxClientsPerHost" es sumamente útil para evitar los abusos de los aceleradores de descargas (o "Download Accelerators"), utilizando esta opción evitamos que desde una misma dirección de internet accedan múltiples conexiones. Por otro lado esta opción puede generarnos problemas si hay varios usuarios que acceden al servidor ftp y están detrás de un proxy, dependiendo de los usos de cada servidor su administrador sabrá si utiliza esta opción o no.

Al igual que en apache, cada "VirtualHost" puede tener su directorio raíz propio y en este ejemplo también definimos que cada servidor virtual tenga un directorio para accesos anónimos diferente. Lo ligeramente peculiar de este ejemplo es que si bien los usuarios anónimos entran a diferentes servidores virtuales y diferentes directorios dentro del mismo equipo, están todos utilizando el mismo usuario de sistema "ftp" (en este caso).

El último de los "VirtualHost" muestra el caso de la utilización de un puerto diferente al 21, esto se genera con la opción "Port [número\_de\_puerto]".

## 5. Conclusiones

Este servidor ftp es sumamente práctico para configurar y lo detallado en este artículo es un pequeño porcentaje de todas sus posibilidades, vale la pena investigar más las funcionalidades que tiene este software, especialmente las diferentes opciones de autenticación ya que con este sistema se pueden alojar los usuarios en bases de datos o directorios X.500 tipo LDAP.

## **6. Referencias**

<http://www.ptoftpd.org> (sitio oficial del ProFTPD) (<http://www.proftpd.org>)