# ProFTPD tutorial en GNU/Linux Debian/Ubuntu

Muchas veces necesitamos copiar archivos de una computadora a otra de manera rápida y para ello podemos utilizar el Protocolo de Transferencia de Archivo (**FTP** por sus iniciales en inglés File Transfer Protocol). Pero para ellos debemos utilizar (sin importar el tipo  sistema operativo en ambas máquinas) un programa servidor y un programa cliente. Para el programa servidor en esta oportunidad presentamos ProFTPD un programa de código abierto -software libre- sobre el sistema operativo GNU/Linux. Para este tutorial utilizaremos Ubuntu como máquina servidora de archivos: bajoe ste ambiente instalaremos el ProFTPD.

## Breve historia del FTP.

En 1971, mucho antes de inventarse el protocolo TCP/IP, ya había la necesidad de copiar archivos de un ordenador a otro, el problema es que ambas máquinas tenían sus propios "sistemas operativos" (eran los albores de la infomática: en realidad era un software que servía única y exclusivamente a cada modelo de máquina en particular, hoy día le decimos "firmware"). Para resolver este problema el programador indio Abhay Bhushan propuso la norma en el documento RFC 114 donde esbozó cómo hablarían las máquinas por medio de una *conexión indirecta.* Este método de conexión indirecta significaba que no se necesitaba un nombre de usuario o identificación (o incluso contraseña alguna) para acceder a los archivos de una computadora destinada a alojar datos (archivos).

Para aquella época la seguridad era muy simple: ¿quién tiene dinero para comprar una computadora y de paso tenerla en linea? "La nube" -como se empeñan en llamarla hoy- era muy pequeña, todos se conocían. Por ello no se pretendía ir más allá de resolver el problema de pasar datos por medio del  **Programa de Control de Red de trabajo** (NCP por sus siglas en inglés de "Network Control Program") la primigenia y precursora capa de red para entonces. Aquí el sr. Bhushan propuso utilizar los caracteres ASCII siguientes para controlar los paquetes al transmitirlos: SOH, STX, ETX, DC1, DC2, DC3, US, RS, GS y FS. También se definió los comandos a utilizar definidos por un solo caracter ASCII: recordad que el hardware era lentísimo y cada bit ahorrado era tiempo ganado. Esto fue un denominador común en el nacimiento de TODOS los protocolos de esa época: debido a la limitación de hardware existente MENOS ES MÁS -y cuya principal consecuencia fue el "bug" del milenio en el año 2000, pero esa es otra historia-.

https://twitter.com/ks7000/status/760205678061481984

Así, por ejemplo, el comando "Retrieve" (descargar un archivo) se iniciaba con la letra "R" a lo que el servidor respondía con un "ready-to-send (rs)" enviando el caracter ">" y finalizaba de enviar el archivo con el comando "complete_file" enviando el caracter "*" (todo lo anterior "encapsulado" por los códigos STX y ETX, entre otros). Si quereís ahondar más en el uso de caracteres ASCII como "protocolo" de intercambio de información podeís hacer click en este enlace.

Ya para 1980 esta norma fue reemplazada por la RFC 765 donde soportaba los entonces nuevos protocolos TCP y TELNET (el señor Jon Postel, proponente, así aclara en la introducción donde asume el conocimiento previo de dichas normas). Además incluye una terminología donde incluye conceptos nuevos acorde con la tecnología y hardware exitentes y los comandos dejan de ser de un solo caracter a varios, iniciando así un lenguaje de alto nivel (lo comprendemos los humanos, por ejemplo, desconectarse o "logout" se enviaba un comando "**QUIT**").

En 1985 de nuevo el señor Jon Postel -junto con J. Reynolds- lanza la RFC 959 donde reconoce los nuevos artefactos de computación y los define, aparte de agregar comandos opcionales como el famoso **MKD** -"make directory", crear un directorio- sumando funcionalidad práctica los programas clientes. Esta sigue siendo la norma vigente hoy en día y la razón de su longevidad es que es muy parecida a todas las cosas que podemos hacer hoy día por medio de una linea de comandos en nuestras propias computadoras: copiar, renombrar, mover archivos, etc.

---

> ***Nota curiosa:*** Jon Postel es el creador de 8 servidores de dominio raíz DNS agrupados bajo la figura de **Internet Assigned Numbers Authority** (IANA), lo cual hizo para separarse de la ARPANET (4 servidores raíz en control del ejército de los EE.UU.) *sin perder conexión nunca -ni en ningún momento- ambas redes entre sí*. A pesar de sus detractores, fue una acción heróica y por represión el gobierno hizo que se detractara en sus acciones y una semana después se adueñó de la IANA -incluso la Wikipedia apunta que la fundó el gobierno lo cual no es así, el autor intelectual es Postel- y el incidente quedó registrado en el RFC 2468. Postel murió 9 meses después del asunto debido a un ataque al corazón -¿quién no se preocuparía si tu propio gobierno te reprime en tus pensamientos?- y con el pasar de los años el triunfo fue para Jon Postel: hoy la IANA es apenas un departamento de la ICANN donde los "civiles" somos en realidad quienes controlamos los nombres de dominios en internet -insisto: siguen habiendo detractores de que no tenemos ese control pero esa discusión no la detallaremos en esta entrada-.

---

En 1994 en la RFC 1579 se agrega se **Firewall-Friendly FTP** (passive mode), en 1997 se hace un enmienda para agregarle seguridad en la RFC 2228 y en 1998 en la RFC 2428 hace soporte para IPv6 y define un *nuevo modo pasivo de conexión*. Este modo pasivo se agrega para dar soporte a servidores FTP que están en una red de área local regida por un NAT (un enrutador utilizado para compatir una conexión a internet entre varios ordenadores o artefactos) o un "Firewall" (filtro de conexiones entrantes utilizado para evitar entradas no autorizadas en una

computadora).

Pero para que conozcaís cual es la diferencia entre modo activo y modo pasivo, vamos a explicarlo *corriendo el riesgo de ser demasiados simplistas* -catedráticos, PhD, licenciados, etc. perdonadme por adelantado debido a lo que os voy a decir-. El esquema de conexión (conocido hoy como conexión en modo activo) es el siguiente:

## Conexión modo activo:

1. Un servidor de archivos ejecuta constantemente (da servicio) un programa que "escucha" en el puerto 21 y "habla" por el puerto 20 (imaginemos una carretera doble vía: así evitamos colisiones de paquetes).
2. Un cliente dado conoce la dirección IP del servidor y abre un puerto de escucha **N**.
3. El cliente envia el comando "**PORT"** acompañado de su dirección IP y el puerto **N** abierto en el paso 2 al puerto 21 del servidor.
4. El servidor envia respuesta por su puerto 20 al puerto N del cliente.
5. Se ejecutan más comandos, los que se necesiten o deseen.
6. Se cierra la conexión por cualquiera de las dos partes tras lo cual quien recibe el aviso de cierre contesta indicando que también cierra la conexión.

## Conexión modo pasivo:

1. Un servidor de archivos ejecuta constantemente un programa que "escucha" en el puerto 21 y "habla" por el puerto 20.
2. Un cliente dado conoce la dirección IP del servidor y simplemnte envia al puerto 21 del servidor el comando "**PASV**" (está prohibido enviar algo más, es decir se vnia SIN parámetros).
3. El servidor recibe por el puerto 21 y procede a abrir un puerto "de salida" **N**.
4. El servidor contesta por el puerto 20 enviando un código "227" más un texto explicativo más su propia dirección IP acompañado del puerto **N** abierto en el paso 3.
5. El cliente recibe el mensaje del punto anterior y cuyo formato es el siguiente: "227 Entering Passive Mode. A1,A2,A3,A4,a1,a2" donde A1~A4 es la dirección IP y a1*256+a2=puerto **N** abierto en el servidor.
6. Se ejecutan más comandos, los que se necesiten o deseen.
7. Se cierra la conexión por cualquiera de las dos partes tras lo cual quien recibe el aviso de cierre contesta indicando que también cierra la conexión.

## Comandos de respuesta numerados y agrupados.

En la sección anterior pudieron observar que el servidor responde con un comando numerado, 227 en el caso del comando **PASV**. En realidad todos los comandos que comienzan con el número 2 son comandos de respuesta exitosa y que se espera por un nuevo comando que no tiene que ver

con el comando previamente enviado. El resto de las numeraciones de respuesta son las siguientes:

```
 1XX Comando exitoso y se espera un hilo de comandos (mantener orden estr
icto).  2XX Comando exitoso y se esperan nuevos comandos.  3XX El coman
do ha sido aceptado pero necesita más datos para completarlo.  4XX El co
mando no se aceptó de manera temporal, se puede intentar de nuevo más tar
de.  5XX El comando se negó de plano, no reintentar.  6XX Son utilizado
s en protocolos seguros de transferencia.  X0X Errores de sintaxis o com
andos superfluos.  X1X Mensajes informativos.  X2X Referentes a conexió
n.  X3X Referentes a ingreso a cuentas y autenticación.  X4X No especif
icados en la norma.  X5X Referentes a sistemas de archivos.
```

Como podéis observar hay diversas combinaciones posibles, no obstante ya hay unos mensajes predefinidos que podéis observar en este enlace. No obstante no quiere decir que no hay más mensajes, por ejemplo Microsoft -empresa que gusta de hacer las cosas a su manera apalancados por muchísimo dinero de por medio- creó su propio mensaje **no normado** e identificado con el número 234 ("2" comando exitoso pero "3" se necesitan más datos para acceder a la cuenta y "4" un mensaje solo para nosotros los seres humanos). Así que si creaís vuestro propio sofware libre de servidor FTP no dudéis de informar a vuestro clientes con el código 234 cuando tengan problemas para autenticarse con vosotros y se lo explicáis bien amablemente. ;-)

---

## Breve historia del ProFTPD.

ProFTPD nace de la necesidad de tener un software con mayor seguridad y con miras a complementar al servidor web Apache. Sus autores invirtieron mucho tiempo y esfuerzo en WU-FTPD (**wuarchive-ftpd**), un servidor FTP para Unix, al cual le corrigieron muchos errores de seguridad. Pronto se dieron cuenta que lo mejor era comenzar un proyecto totalmente nuevo al cual denominaron ProFTPD. Su creador fue Jesse Sipprell (hoy retirado de su desarrollodo) y es mantenido a la fecha por TJ Saunders -y su equipo-.

## Instalación del ProFTPD.

Teniendo nuestros repositorios bien configurados, abrimos una ventana terminal y ganamos acceso con usuario raíz **root** e introducimos los siguientes comandos:

```
apt-get update  apt-get install proftpd
```

Veremos algo similar a esto por pantalla:

Durante su instalación nos hará una sola pregunta, si lo queremos ejecutar sobre **inetd** (un demonio "**daemon**" que escucha y recibe las solicitudes y ejecuta el programa adecuado) o si lo queremos ejecutar de manera completa "**standalone**". La decisión en este caso depende de si queremos descargar archivos de vez en cuando o si queremos un servir múltiples conexiones. *Con la capacidad de hardware de hoy en día el cual es más que suficiente para la mayoría de los usuarios, de manera predeterminada está seleccionada la opción "**standalone**".*

Y listo ***¿A que fue fácil, verdad o no?*** Ahora para probarlo nos podemos conectar desde cualquier otro ordenador o dispositivo en nuestra red de área local con la dirección IP del servidor y con nuestro nombre de usuario -y contraseña- en nuestra distribución Ubuntu. *Por defecto, todos los usuarios registrados en el sistema operativos del servidor tienen acceso vía ftp.* Desde cualquier distribución GNU/Linux podemos conectarnos rápidamente por medio de la linea de comandos gracias a la utilidad **ftp** desarrollada para Unix en 4.2 BSD, acá ponemos una captura de pantalla hecha en Debian:

## Configuración del ProFTPD.

Una vez probado os daréis cuenta que ProFTPD honra sobradamente los deseos del señor Abhay Bhushan. Aunque no podemos conectarnos de manera indirecta podemos habilitar la conexión anónima a nuestro servidor de la siguiente manera: se acostumbra utilizar como nombre de usuario la palabra anónimo en inglés "**anonymous**" y como contraseña nuestro correo electrónico con las restricciones de solo lectura en todos los directorios disponibles y como solo escritura en la carpeta "incoming". Para ello debemos modificar al archivo "/etc/proftpd/proftpd.conf". Primero debemos respaldar dicho archivo por asia caso cometemos cualqueir error podremos restaurar rapidamente los valores:

```
cp /etc/proftpd/proftpd.conf /etc/proftpd/proftpd.conf.bak  nano /etc/pro
ftpd/proftpd.conf
```

E introducimos el contenido siguiente, tal como aconsejan en la propia página web de ProFTPD:

```
   # After anonymous login, daemon runs as user/group ftp.   User ftp   G
```

```
roup ftp    # The client login 'anonymous' is aliased to the "real" user
'ftp'.   UserAlias anonymous ftp    # Deny write operations to all direct
ories, except for 'incoming' where    # 'STOR' is allowed (but 'READ' ope
rations are prohibited)          DenyAll                DenyAll         A
llowAll
```

Luego ejecutamos el reinicio del servicio ftp:

```
sudo service proftpd restart
```

Unav vez reiniciado el servicio procedemos a comprobar de nuevo la conexión.

También es aconsejable cambiar, según nuestras necesidades, los siguientes parámetros de igual manera como acabamos de hacer con las conexiones anónimas:

- **UseIPv6**: podemos colocarlo en "off" mientras no usemos IPv6, por defecto viene "on".
- **ServerName**: por defecto al instalarse le coloca el nombre dado al ordenador por medio del sistema operativo, pero tal vez necesitemos colocarle el nombre de nuestra empresa u organización.
-

## Enlaces consultados.

### En idioma castellano:

### En idioma inglés:


- «https://ftptest.net/Help»
- WU-FTPD Development Group (at web.archive.org year 2008).
- WU-FTPD (at Wikipedia).
- ProFTPD: Project goals (at official site).
- Proftpd: a simple proftp tutorial (at calomel.org).
- Securizing the FTP (proftpd) Server (at Pandora FMS's Wiki).
- How to set up an FTP server on Windows (at Serv-U).
- How To Set Up ProFTPD on Ubuntu 12.04 (at DigitalOcean).
- How To Configure ProFTPd To Use SFTP Instead of FTP (at DigitalOcean).
- Installing and Configuring ProFTPD Server in Ubuntu/Debian (by Hanny Helal).
- How to use the Linux ftp command to up- and download files on the shell (at

HowToForge.com)

## En idioma castellano:

- «Que es y como instalar un servicio FTP en Ubuntu» por Lluis Espert.