# Instalación y configuración del servidor ftp PureFTP


## Luis Llorente Campo
**Universidad de León, España**

**luisllorente@luisllorente.com**



Este documento muestra cómo instalar y configurar el servidor ftp PureFTP (http://www.pureftpd.org/). Todo el proceso está pensado para la distribución Debian GNU/Linux. Pretende ser una guía que muestre el proceso de una forma genérica, teniendo que ser adaptado el proceso para cada situación específica.


## Introducción

Se ha elegido el servidor ftp PureFTP (http://www.pureftpd.org/) por ser un servidor muy seguro (no ha habido nunca ningún fallo grave conocido), eficiente y con muchas características. Dispone de abundante documentación y satisface con creces los requerimientos exigidos.

Algunas de sus características más notables son:

- Usuarios virtuales
- Gestión del ancho de banda y de espacio en disco por usuario
- Directorios personales Chroot()
- Estadísticas en tiempo real en Texto, HTML ó XML
- Autentificación de usuarios con MySQL, PostgreSQL, LDAP, ...
- Opciones avanzadas de seguridad

Como prueba de su potencia, mencionar que el servidor ftp de RedIris (ftp://ftp.rediris.es) ha migrado a PureFTP hace varias semanas.

## Instalación

Necesitamos instalar el paquete pure-ftpd

Debido a que no es un paquete oficial de la distribución Debian (aún), deberemos añadir al fichero /etc/apt/sources.list las siguientes líneas:

```
deb http://pureftpd.sourceforge.net/debian/potato/ ./
```

```
deb-src http://pureftpd.sourceforge.net/debian/potato/ ./
```

Una vez añadidas, deberemos actualizar la lista de paquetes e instalarlo mediante los comandos:

```
# apt-get update
# apt-get install pure-ftpd
```

Si queremos compilar nosotros mismos el código fuente para activar o desactivar alguna de las opciones con las que viene por defecto, deberemos bajar el código con el comando

```
# apt-get update
# apt-get source pure-ftpd
```

Y después compilarlo siguiendo las instrucciones que lo acompañan.

# Configuración

Los ficheros de configuración son:

- /etc/pure-ftpd.conf para la configuración del servidor
- /etc/pureftpd.passwd para la gestión de usuarios

Veamos la configuración del servidor y la gestión de los usuarios

## Servidor

Veremos las directivas de configuración más importantes que hemos utilizado. Una de las características es que queremos que los usuarios no puedan salir de su directorio:

```
ChrootEveryone              yes
```

Otra de las características que debe poseer el servidor ftp es que disponga de una zona de descarga pública (sin necesitar usuario autentificado). Para ello debemos aceptar conexiones anónimas, con lo que deberá existir la siguiente línea:

```
NoAnonymous                 no
```

Debido a que los usuarios serán virtuales, debemos asegurarnos que la línea a continuación existe y no está comentada. Nos indica la localización del fichero de usuarios.

```
PureDB                        /etc/pureftpd.pdb
```

Por último, para evitar que algún usuario por descuido (o maliciosamente) nos pueda llenar el servidor de ficheros, activaremos el límite. En este caso es de 1000 ficheros y 50 MB:

```
Quota                       1000:50
```

### Usuarios

Los usuarios virtuales son aquellos que sólo existen para el servicio ftp (no existen en el sistema), con lo que no tenemos por qué crear usuarios de sistema si sólo van a utilizar el ftp. De esta forma aumentamos la seguridad del equipo (al haber menos riesgos).

La gestión de los usuarios se realiza con el comando **pure-pw**. Éste nos permite crear, modificar, borrar y mostrar los usuarios virtuales. También se puede hacer lo mismo editando directamente el fichero /etc/pureftpd.passwd, pero se recomienda el uso del comando por su mayor sencillez.

Para ver todos los parámetros disponibles, ejecutaremos:

```
# pure-pw --help
```

Veamos por ejemplo cómo se crearía un usuario llamado "manolo" cuyo directorio fuese /home/manolo :

```
# pure-pw useradd manolo -u manolo -d /home/manolo
```

Hay que recordar que después de haber realizado cualquier cambio relativo a los usuarios, deberemos rehacer la base de datos. Para ello ejecutaremos:

```
# pure-pw mkdb
```

## Ejemplo de fichero de configuración de PureFTP

Este es el fichero completo de configuración del servidor ftp que hemos configurado. Se incluyen los comentarios originales para una mejor comprensión del significado de algunas de las directivas de configuración:

```
############################################################
#                                                          #
#         Configuration file for pure-ftpd wrappers        #
#                                                          #
############################################################

# If you want to run Pure-FTPd with this configuration
# instead of command-line options, please run the
# following command :
#
# /usr/sbin/pure-config.pl /usr/etc/pure-ftpd.conf
#
# RPM binary files use another configuration file by default :
# /etc/sysconfig/pure-ftpd
#
# Please don't forget to have a look at documentation at
# http://www.pureftpd.org/documentation.html for a complete list of
# options.

# Cage in every user in his home directory

ChrootEveryone              yes
```

```
# If the previous option is set to "no", members of the following group
# won't be caged. Others will be. If you don't want chroot()ing anyone,
# just comment out ChrootEveryone and TrustedGID.

# TrustedGID                    50


# Turn on compatibility hacks for broken clients

BrokenClientsCompatibility  no


# Maximum number of simultaneous users

MaxClientsNumber            10


# Fork in background

Daemonize                   yes


# Maximum number of sim clients with the same IP address

MaxClientsPerIP             6


# If you want to log all client commands, set this to "yes".
# This directive can be duplicated to also log server responses.

VerboseLog                  no


# List dot-files even when the client doesn't send "-a".

DisplayDotFiles             no


# Don't allow authenticated users - have a public anonymous FTP only.

AnonymousOnly               no
```

```
# Disallow anonymous connections. Only allow authenticated users.

NoAnonymous                 no



# Syslog facility (auth, authpriv, daemon, ftp, security, user, local*)
# The default facility is "ftp".

SyslogFacility              ftp



# Display fortune cookies

# FortunesFile              /usr/share/fortune/zippy



# Don't resolve host names in log files. Logs are less verbose, but
# it uses less bandwidth. Set this to "yes" on very busy servers or
# if you don't have a working DNS.

DontResolve                 yes



# Maximum idle time in minutes (default = 15 minutes)

MaxIdleTime                 15



# LDAP configuration file (see README.LDAP)

# LDAPConfigFile                /etc/pureftp-ldap.conf



# MySQL configuration file (see README.MySQL)

# MySQLConfigFile               /etc/pureftp-mysql.conf


# Postgres configuration file (see README.PGSQL)

# PGSQLConfigFile               /etc/pureftp-pgsql.conf


# PureDB user database (see README.Virtual-Users)

PureDB                        /etc/pureftpd.pdb
```

```
# Path to pure-authd socket (see README.Authentication-Modules)

# ExtAuth                       /var/run/ftpd.sock


# If you want to enable PAM authentication, uncomment the following line

# PAMAuthentication             yes


# If you want simple Unix (/etc/passwd) authentication, uncomment this

# UnixAuthentication            yes


# Please note that LDAPConfigFile, MySQLConfigFile, PAMAuthentication and
# UnixAuthentication can be used only once, but they can be combined
# together. For instance, if you use MySQLConfigFile, then UnixAuthentication,
# the SQL server will be asked. If the SQL authentication fails because the
# user wasn't found, another try # will be done with /etc/passwd and
# /etc/shadow. If the SQL authentication fails because the password was wrong,
# the authentication chain stops here. Authentication methods are chained in
# the order they are given.


# 'ls' recursion limits. The first argument is the maximum number of
# files to be displayed. The second one is the max subdirectories depth

LimitRecursion              2000 5


# Are anonymous users allowed to create new directories ?

AnonymousCanCreateDirs      no


# If the system is more loaded than the following value,
# anonymous users aren't allowed to download.

MaxLoad                     4


# Port range for passive connections replies. - for firewalling.
```

```
PassivePortRange          30000 50000


# Force an IP address in PASV/EPSV/SPSV replies. - for NAT.

ForcePassiveIP               192.168.2.2


# Upload/download ratio for anonymous users.

# AnonymousRatio               1 10


# Upload/download ratio for all users.
# This directive superscedes the previous one.

# UserRatio                1 10


# Disallow downloading of files owned by "ftp", ie.
# files that were uploaded but not validated by a local admin.

AntiWarez                  yes


# IP address/port to listen to (default=all IP and port 21).

Bind                      litio,21


# Maximum bandwidth for anonymous users in Kb/s

# AnonymousBandwidth           8


# Maximum bandwidth for *all* users (including anonymous) in Kb/s
# Use AnonymousBandwidth *or* UserBandwidth, both makes no sense.

# UserBandwidth            8


# File creation mask. <umask for files>:<umask for dirs> .
# 177:077 if you feel paranoid.

Umask                      133:022
```

```
# Minimum UID for an authenticated user to log in.

MinUID                      100


# Allow FXP transfers for authenticated users only.

#AllowUserFXP               yes


# Allow anonymous FXP for anonymous and non-anonymous users.

#AllowAnonymousFXP          no


# Users can't delete/write files beginning with a dot ('.')
# even if they own them. If TrustedGID is enabled, this group
# will have access to dot-files, though.

ProhibitDotFilesWrite       yes


# Prohibit *reading* of files beginning with a dot (.history, .ssh...)

ProhibitDotFilesRead        yes


# Never overwrite files. When a file whoose name already exist is uploaded,
# it get automatically renamed to file.1, file.2, file.3, ...

#AutoRename                 no


# Disallow anonymous users to upload new files (no = upload is allowed)

#AnonymousCantUpload        no


# Only connections to this specific IP address are allowed to be
# non-anonymous. You can use this directive to open several public IPs for
# anonymous FTP, and keep a private firewalled IP for remote administration.
# You can also only allow a non-routable local IP (like 10.x.x.x) to
# authenticate, and keep a public anon-only FTP server on another IP.
```

```
#TrustedIP                  10.1.1.1


# If you want to add the PID to every logged line, uncomment the following
# line.

#LogPID                     yes


# Create an additional log file with transfers logged in a Apache-like format :
# fw.c9x.org - jedi [13/Dec/1975:19:36:39] "GET /ftp/linux.tar.bz2" 200 21809338
# This log file can then be processed by www traffic analyzers.

AltLog                     clf:/var/log/pureftpd.log


# Create an additional log file with transfers logged in a format optimized
# for statistic reports, as done with ftpStats
# (http://www.shagged.org/ftpstats) .

# AltLog                     stats:/var/log/pureftpd.log


# Create an additional log file with transfers logged in the standard W3C
# format (compatible with most commercial log analyzers)

# AltLog                     w3c:/var/log/pureftpd.log


# Disallow the CHMOD command. Users can’t change perms of their files.

#NoChmod                    yes


# Allow users to resume and upload files, but *NOT* to delete them.

#KeepAllFiles               yes


# Automatically create home directories if they are missing

#CreateHomeDir              yes
```

```
# Enable virtual quotas. The first number is the max number of files.
# The second number is the max size of megabytes.
# So 1000:10 limits every user to 1000 files and 10 Mb.

#Quota                      1000:10



# If your pure-ftpd has been compiled with standalone support, you can change
# the location of the pid file. The default is /var/run/pure-ftpd.pid

#PIDFile                    /usr/local/var/pure-ftpd.pid



# If your pure-ftpd has been compiled with pure-uploadscript support,
# this will make pure-ftpd write info about new uploads to
# /var/run/pure-ftpd.upload.pipe so pure-uploadscript can read it and
# spawn a script to handle the upload.

#CallUploadScript yes



# This option is usefull with servers where anonymous upload is
# allowed. As /var/ftp is in /var, it save some space and protect
# the log files. When the partition is more that X percent full,
# new uploads are disallowed.

MaxDiskUsage               90



# Set to 'yes' if you don't want your users to rename files.

#NoRename yes



# Be 'customer proof' : workaround against common customer mistakes like
# 'chmod 0 public_html', that are valid, but that could cause ignorant
# customers to lock their files, and then keep your technical support busy
# with silly issues. If you're sure all your users have some basic Unix
# knowledge, this feature is useless. If you're a hosting service, enable it.

CustomerProof yes
```

Para una explicación detallada de todas las directivas, se ruega consultar la documentación que acompaña al servidor.

## Prueba

Una vez esté todo configurado, pondremos en marcha el servidor mediante el comando:

```
# /etc/init.d/pure-ftpd restart
```

A partir de este momento, podremos entrar en el servidor ftp con cualquier cliente ftp, tanto con un usuario anónimo como con uno creado anteriormente con el comando **pure-pw useradd**. En ese caso, deberemos introducir el nombre de usuario y la contraseña que utilizamos al crearlo.

```
ftp://nombre.del.servidor/
```

Si queremos ver qué usuarios están conectados al servidor ftp en un determinado momento, lo haremos mediante el comando **pure-ftpwho**:

```
# pure-ftpwho
```

Nos aparecerá una tabla en la que veremos los usuarios conectados, el fichero que están subiendo o bajando y la velocidad de transmisión.

Hay que recordar que si se hace alguna modificación al fichero de configuración del servidor, se deberá arrancar de nuevo el servidor para que lea la nueva configuración. Si se hace al fichero de usuarios, no es necesario hacerlo.

## Más información

Se recomienda la lectura de toda la documentación que acompaña al servidor, ya que lo aquí visto es sólo una pequeña parte de las enormes posibilidades que posee. Como ejemplo, se puede mencionar la posibilidad de almacenar los usuarios en una base de datos MySQL, o de utilizar un método de autentificación propio. Se puede también gestionar el ancho de banda que puede utilizar cada usuario (tanto de subida como de bajada), así como el horario en el que se puede conectar al servidor y las direcciones IP desde las que se permite conectar. También se puede hacer que al subir algún fichero al servidor se ejecute un script que realice algo con él (avisar al administrador, comprobar si tiene virus, etc...).

Como se puede ver, las posibilidades son casi infinitas.

Esta documentación se encuentra en el directorio /usr/share/doc/pure-ftpd, y está compuesta principalmente por los ficheros:

- README
- README.Authentication-Modules
- README.Contrib
- README.LDAP
- README.MySQL
- README.PGSQL
- README.Netfilter
- README.Virtual-Users

- README.Debian

Otra fuente de información puede ser la página web oficial, en la que siempre se encuentra la última versión disponible de la documentación: http://www.pureftpd.org (http://www.pureftpd.org/), así como la última versión del servidor, junto con noticias de interés e información de cómo conseguir la versión en desarrollo del mismo a través de CVS (se recomienda utilizar la versión de desarrollo sólo para hacer pruebas y ayudar en el desarrollo del proyecto).

## Sobre este documento



Este documento ha sido escrito en formato XML utilizando la DTD de DocBook (http://www.docbook.org). Mediante este sistema, puede ser fácilmente transformado a múltiples formatos (HTML, TXT, PDF, PostScript, LaTeX, DVI, ...). Se recomienda su utilización como herramienta de documentación potente y libre.