SERVICE FTP

Serveur Linux CentOS

Table des matières

1 Introduction

File Transfer Protocol, ou FTP, est un protocole de transfert de fichiers sur un réseau TCP/IP. Il permet, depuis un client ftp, de transférer des fichiers de ou vers un serveur ftp.

vsftpd signifie "**Very Secure File Transfert Protocol deamon**". C'est le service ftp le plus utilisé sur **Linux**.

2 Installation du client FTP

Vérifier si le client FTP est installé :

```
[root@localhost ~]# rpm -q ftp
package ftp is not installed
```

Installer le client FTP :

```
root@localhost ~]# yum install ftp
```

3 Installation du service FTP

Vérifier si le service FTP est installé :

```
[root@localhost ~]# rpm -q vsftpd
package vsftpd is not installed
```

Installer le service FTP :

```
[root@localhost ~]# yum install vsftpd
```

4 Démarrer/Arrêter le service FTP

Démarrer le service FTP :

```
[root@localhost ~]# systemctl start vsftpd
```

Vérifier le statut du service FTP:

```
[root@localhost ~]# systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded
(/usr/lib/systemd/system/vsftpd.service; disabled;
vendor preset: disabled)
   Active: active (running) since Wed 2019-06-05
00:15:45 EDT; 1min 30s ago
  Process: 8196 ExecStart=/usr/sbin/vsftpd
/etc/vsftpd/vsftpd.conf (code=exited, status=0/SUCCESS)
 Main PID: 8197 (vsftpd)
   CGroup: /system.slice/vsftpd.service
           └─8197 /usr/sbin/vsftpd
/etc/vsftpd/vsftpd.conf
```

Arrêter le service FTP:

```
 [root@localhost ~]# systemctl stop vsftpd
```

5 Port FTP

Par défaut, le service ftp écoute sur le port 21 de toutes les interfaces disponibles :

```
[root@localhost ~]# nmap 127.0.0.1

Starting Nmap 6.40 ( http://nmap.org ) at 2019-06-12
11:26 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000020s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
25/tcp open  smtp
```

```
[root@localhost ~]# nmap 192.168.17.158

Starting Nmap 6.40 ( http://nmap.org ) at 2019-06-12
11:50 EDT
Nmap scan report for 192.168.17.158
Host is up (0.0000020s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
```

6 Ouvrir une session FTP

Par défaut, il n'est pas permis d'ouvrir une session ftp avec l'utilisateur *root*.

```
[root@localhost ~]# ftp 192.168.17.158
Connected to 192.168.17.158 (192.168.17.158).
220 (vsFTPd 3.0.2)
Name (192.168.17.158:root):
530 Permission denied.
Login failed.
ftp>
```

Il faut donc créer un utilisateur sur le serveur ftp pour pouvoir faire un test.

```
[root@localhost ~]# useradd hakimb

[root@localhost ~]# passwd hakimb
Changing password for user hakimb.
New password:
BAD PASSWORD: The password fails the dictionary check -
it is based on a dictionary word
Retype new password:
passwd: all authentication tokens updated successfully.
```

Ouvrir une session ftp avec l'utilisateur *hakimb* :

```
[root@localhost ~]# ftp 192.168.17.158
Connected to 192.168.17.158 (192.168.17.158).
220 (vsFTPd 3.0.2)
Name (192.168.17.158:root): hakimb
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

La commande *help* permet d'afficher les commandes disponibles :

```
ftp> help
Commands may be abbreviated.  Commands are:

!               debug           mdir            sendport        site
$               dir             mget            put             size
account         disconnect      mkdir           pwd             status
append          exit            mls             quit            struct
ascii           form            mode            quote           system
bell            get             modtime         recv            sunique
binary          glob            mput            reget           tenex
bye             hash            newer           rstatus         tick
case            help            nmap            rhelp           trace
cd              idle            nlist           rename          type
cdup            image           ntrans          reset           user
chmod           lcd             open            restart         umask
close           ls              prompt          rmdir           verbose
cr              macdef          passive         runique         ?
delete          mdelete         proxy           send
ftp>
```

Pour quitter une session ftp, on peut utiliser la commande *exit* ou *bye* ou *quit* :

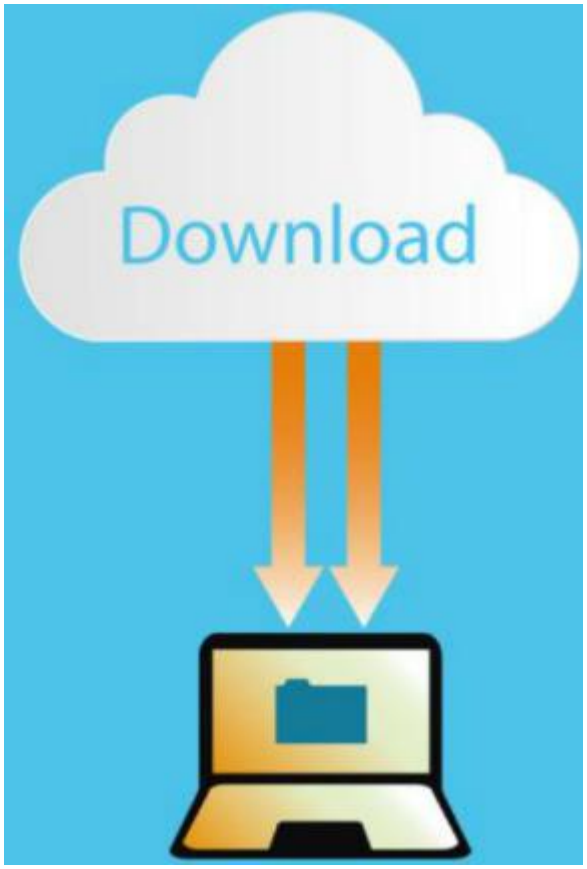
```
ftp> exit
221 Goodbye.
[root@localhost ~]#
```

```
ftp> quit
221 Goodbye.
[root@localhost ~]#
```

```
ftp> bye
221 Goodbye.
[root@localhost ~]#
```

7 Télécharger

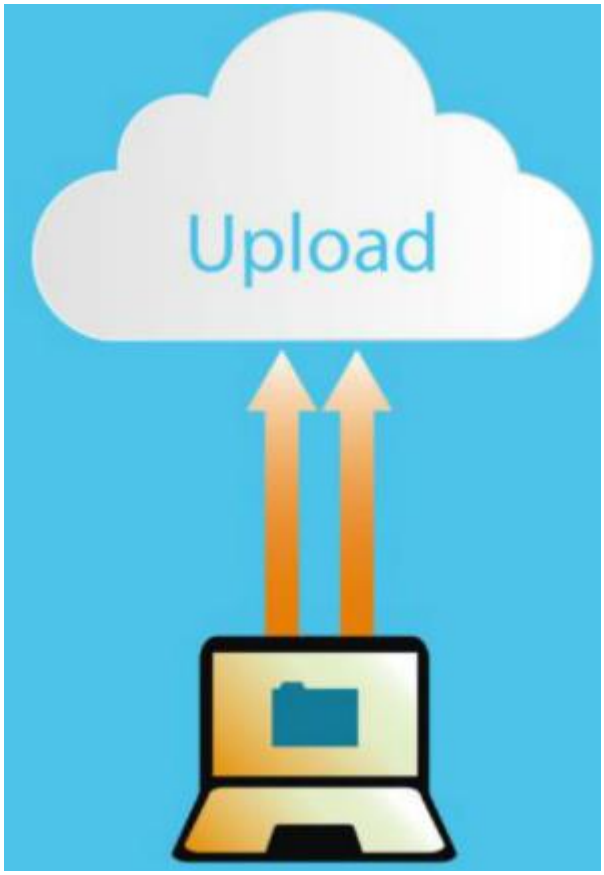
Télécharger (download) signifie transférer un fichier du serveur ftp vers le client ftp.



La commande *get* permet de télécharger un seul fichier alors que la commande *mget* permet de télécharger plusieurs fichiers.

8 Téléverser

Téléverser (upload) signifie transférer un fichier du client ftp vers le serveur ftp.



La commande *put* permet de téléverser un seul fichier alors que la commande *mput* permet de téléverser plusieurs fichiers.

Exemple

Dans cet exemple, on va transférer trois fichiers du client vers le serveur ftp.

On va commencer par créer les trois fichiers :

Se connecter sur le client avec l'utilisateur *hakimb* :

```
[root@localhost ~]# su - hakimb
```

Créer trois fichiers :

```
[hakimb@localhost ~]$ touch script1.sh script2.sh
script3.sh
[hakimb@localhost ~]$ ls -l
total 0
-rw-rw-r-- 1 hakimb hakimb 0 Jun 12 12:31 script1.sh
-rw-rw-r-- 1 hakimb hakimb 0 Jun 12 12:31 script2.sh
-rw-rw-r-- 1 hakimb hakimb 0 Jun 12 12:31 script3.sh
[hakimb@localhost ~]$
```

Ouvrir une session ftp au serveur ftp :

```
[hakimb@localhost ~]$ ftp 192.168.17.158
Connected to 192.168.17.158 (192.168.17.158).
220 (vsFTPd 3.0.2)
Name (192.168.17.158:hakimb):
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Transférer le fichier *script1.sh* vers le serveur ftp :

```
ftp> ls
227 Entering Passive Mode (192,168,17,158,230,67).
150 Here comes the directory listing.
226 Directory send OK.
```

```
ftp> put script1.sh
local: script1.sh remote: script1.sh
227 Entering Passive Mode (192,168,17,158,254,0).
150 Ok to send data.
226 Transfer complete.
```

```
ftp> ls
227 Entering Passive Mode (192,168,17,158,241,184).
150 Here comes the directory listing.
-rw-r--r--    1 1000     1000            0 Jun 12 16:38
script1.sh
226 Directory send OK.
```

Transférer tous les fichiers **.sh* vers le serveur ftp :

```
ftp> mput *.sh
mput script1.sh? y
227 Entering Passive Mode (192,168,17,158,227,24).
150 Ok to send data.
226 Transfer complete.
mput script2.sh? y
227 Entering Passive Mode (192,168,17,158,71,34).
150 Ok to send data.
226 Transfer complete.
mput script3.sh? y
227 Entering Passive Mode (192,168,17,158,171,7).
150 Ok to send data.
226 Transfer complete.
ftp>
```

Pour vérifier :

```
ftp> ls
227 Entering Passive Mode (192,168,17,158,251,80).
150 Here comes the directory listing.
-rw-r--r--    1 1000   1000     0 Jun 12 16:40 script1.sh
-rw-r--r--    1 1000   1000     0 Jun 12 16:40 script2.sh
-rw-r--r--    1 1000   1000     0 Jun 12 16:40 script3.sh
226 Directory send OK.
```

Pour transférer tous les fichiers **.sh* vers le serveur ftp **sans confirmation**, il suffit de désactiver le mode interactif avec commande prompt :

```
ftp> prompt
Interactive mode off.
```

Ensuite, faire le transfert :

```
ftp> mput *.sh
local: script1.sh remote: script1.sh
227 Entering Passive Mode (192,168,17,158,95,17).
150 Ok to send data.
226 Transfer complete.
local: script2.sh remote: script2.sh
227 Entering Passive Mode (192,168,17,158,185,157).
150 Ok to send data.
226 Transfer complete.
local: script3.sh remote: script3.sh
227 Entering Passive Mode (192,168,17,158,205,41).
150 Ok to send data.
226 Transfer complete.
ftp>
```

9 Fichiers de configuration

Les fichiers de configuration sont groupés dans le répertoire /etc/vsftpd. Le fichier de configuration principal est vsftpd.conf.

```
[root@localhost ~]# ls -l /etc/vsftpd/
total 20
-rw------- 1 root root  125 Oct 30  2018 ftpusers
-rw------- 1 root root  361 Oct 30  2018 user_list
-rw------- 1 root root 5116 Oct 30  2018 vsftpd.conf
-rwxr--r-- 1 root root  338 Oct 30  2018
vsftpd_conf_migrate.sh
[root@localhost ~]#
```

9.1 /etc/vsftpd/ftpusers

```
[root@localhost ~]# cat /etc/vsftpd/ftpusers

# Users that are not allowed to login via ftp
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
[root@localhost ~]#
```

9.2 /etc/vsftpd/user_list

```
[root@localhost ~]# cat /etc/vsftpd/user_list

# vsftpd userlist
# If userlist_deny=NO, only allow users in this file
# If userlist_deny=YES (default), never allow users in this file, and
# do not even prompt for a password.
# Note that the default vsftpd pam config also checks /etc/vsftpd/ftpusers
# for users that are denied.
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
[root@localhost ~]#
```

9.3 /etc/vsftpd/vsftpd_conf_migrate.sh

```
[root@localhost ~]# cat /etc/vsftpd/vsftpd_conf_migrate.sh

#!/bin/bash
#move old config files and symlink them
#shipped with vsftpd-2.0.1-6
PREFIX="vsftpd"
for file in $( ls /etc/${PREFIX}.* ); do
    if [ ! -L $file ]; then
        new=`echo $file | sed
s/${PREFIX}\./${PREFIX}\\\\//g | sed s/\.rpmsave//g`
        mv -f ${file} ${new}
        ln -s ${new} ${file}
        echo $file moved to $new
    fi
done
[root@localhost ~]#
```

9.4 /etc/vsftpd/vsftpd.conf

```
# Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd
options.
# Please read the vsftpd.conf.5 manual page to get a full idea of
vsftpd's
# capabilities.
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this
out).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
# When SELinux is enforcing check for SE bool ftp_home_dir
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to
022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This
only
# has an effect if the above global write enable is activated. Also, you
will
# obviously need to create a directory writable by the FTP user.
# When SELinux is enforcing check for SE bool allow_ftpd_anon_write,
allow_ftpd_full_access
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# Activate logging of uploads/downloads.
xferlog_enable=YES
#
# Make sure PORT transfer connections originate from port 20 (ftp-data).
connect_from_port_20=YES
#
# If you want, you can arrange for uploaded anonymous files to be owned
by
# a different user. Note! Using "root" for uploaded files is not
```

```
# recommended!
#chown_uploads=YES
#chown_username=whoever
#
# You may override where the log file goes if you like. The default is
shown
# below.
#xferlog_file=/var/log/xferlog
#
# If you want, you can have your log file in standard ftpd xferlog
format.
# Note that the default log file location is /var/log/xferlog in this
case.
xferlog_std_format=YES
#
# You may change the default value for timing out an idle session.
#idle_session_timeout=600
#
# You may change the default value for timing out a data connection.
#data_connection_timeout=120
#
# It is recommended that you define on your system a unique user which
the
# ftp server can use as a totally isolated and unprivileged user.
#nopriv_user=ftpsecure
#
# Enable this and the server will recognise asynchronous ABOR requests.
Not
# recommended for security (the code is non-trivial). Not enabling it,
# however, may confuse older FTP clients.
#async_abor_enable=YES
#
# By default the server will pretend to allow ASCII mode but in fact
ignore
# the request. Turn on the below options to have the server actually do
ASCII
# mangling on files when in ASCII mode. The vsftpd.conf(5) man page
explains
# the behaviour when these options are disabled.
# Beware that on some FTP servers, ASCII support allows a denial of
service
# attack (DoS) via the command "SIZE /big/file" in ASCII mode. vsftpd
# predicted this attack and has always been safe, reporting the size of
the
# raw file.
# ASCII mangling is a horrible feature of the protocol.
#ascii_upload_enable=YES
#ascii_download_enable=YES
#
# You may fully customise the login banner string:
#ftpd_banner=Welcome to blah FTP service.
#
# You may specify a file of disallowed anonymous e-mail addresses.
Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# (default follows)
```

```
#banned_email_file=/etc/vsftpd/banned_emails
#
# You may specify an explicit list of local users to chroot() to their
home
# directory. If chroot_local_user is YES, then this list becomes a list
of
# users to NOT chroot().
# (Warning! chroot'ing can be very dangerous. If using chroot, make sure
that
# the user does not have write access to the top level directory within
the
# chroot)
#chroot_local_user=YES
#chroot_list_enable=YES
# (default follows)
#chroot_list_file=/etc/vsftpd/chroot_list
#
# You may activate the "-R" option to the builtin ls. This is disabled
by
# default to avoid remote users being able to cause excessive I/O on
large
# sites. However, some broken FTP clients such as "ncftp" and "mirror"
assume
# the presence of the "-R" option, so there is a strong case for
enabling it.
#ls_recurse_enable=YES
#
# When "listen" directive is enabled, vsftpd runs in standalone mode and
# listens on IPv4 sockets. This directive cannot be used in conjunction
# with the listen_ipv6 directive.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default,
listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and
IPv6
# sockets. If you want that (perhaps because you want to listen on
specific
# addresses) then you must run two copies of vsftpd with two
configuration
# files.
# Make sure, that one of the listen options is commented !!
listen_ipv6=YES

pam_service_name=vsftpd
userlist_enable=YES
tcp_wrappers=YES
```