

Mise en place d'un serveur FTP

Projet m2l.org



DUMAS Lucie

Table des matières

Qu'est-ce qu'un serveur FTP ?	3
Création du serveur FTP	4
Configuration des utilisateurs anonymes	7



Qu'est-ce qu'un serveur FTP ?

Un serveur FTP, ou File Transfer Protocol server en anglais, est un serveur informatique qui permet le transfert de fichiers entre des ordinateurs via le protocole FTP. Le protocole FTP est un protocole de communication standard utilisé pour le transfert de fichiers sur un réseau TCP/IP, comme Internet.

Les serveurs FTP peuvent fonctionner de différentes manières. Il existe deux modes principaux de connexion FTP : le mode actif et le mode passif.

Dans le mode actif, le client ouvre une connexion de commande (command connection) au serveur. Lorsqu'il est prêt à recevoir des données, le client ouvre une deuxième connexion appelée connexion de données (data connection) vers le serveur. C'est le client qui choisit le port de la connexion de données. Le serveur accepte la connexion de données du client et envoie les données sur cette connexion. Dans le mode actif, le client est responsable d'ouvrir la connexion de données, et le serveur doit accepter cette connexion entrante.

Dans le mode passif, le client ouvre une connexion de commande au serveur comme d'habitude. Lorsqu'il est prêt à recevoir des données, le serveur ouvre une connexion de données et informe le client de l'adresse IP et du port sur lesquels il peut se connecter. Le client se connecte ensuite à la connexion de données ouverte par le serveur. Dans le mode passif, c'est le serveur qui est responsable d'ouvrir la connexion de données, et le client se connecte à cette connexion sortante.

Les serveurs FTP sont souvent utilisés pour permettre aux utilisateurs d'accéder et de partager des fichiers à distance. Cependant, en raison de problèmes de sécurité liés à la transmission de mots de passe en texte clair et d'autres vulnérabilités, il est recommandé d'utiliser des variantes plus sécurisées du protocole FTP, telles que FTPS (FTP sécurisé) ou SFTP (SSH File Transfer Protocol), qui ajoutent une couche de sécurité aux communications FTP standard.



Création du serveur FTP

Pour créer notre conteneur qui accueillera notre serveur FTP, nous commençons par dupliquer notre conteneur template :

```
lxc-copy -n template -N ftp
```

Nous entrons ensuite la directive suivante dans le fichier `/var/lib/lxc/ftp/config` afin que le conteneur se lance au démarrage de la machine :

```
lxc.start.auto = 1
```

Nous entrons ensuite dans notre conteneur FTP afin de lui attribuer ses nouvelles adresses IP. Contrairement aux autres conteneurs créés précédemment, le conteneur FTP aura besoin de 2 adresses IP différentes. Ces adresses IP permettront la configuration des Virtual Hosts plus tard dans la mission. Nous entrons les nouvelles IP dans le fichier `/etc/rc.local` :

```
ifconfig eth0:0 10.31.96.21/20 up  
ifconfig eth0:1 10.31.96.22/20 up
```

Définir l'interface réseau comme ceci nous permet d'attribuer 2 adresses IP en créant 2 interfaces virtuelles. Nous donnons ensuite les droits d'exécution sur le fichier `/etc/rc.local` :

```
chmod +x /etc/rc.local
```

Nous changeons également l'adresse principale dans le fichier `/etc/network/interfaces` :

```
address 10.31.96.20/20
```

Nous modifions le nom d'hôte de la machine en entrant la ligne ci-dessous dans le fichier `/etc/hosts` :

```
10.31.96.20    ftp
```



Nous redémarrons le conteneur :

```
reboot
```

Nous installons le paquet ProFTPD :

```
apt update  
apt install proftpd
```

Nous redémarrons le service puis nous regardons sur quel port écoute l'outil ProFTPD :

```
systemctl restart proftpd  
netstat -natp
```

```
root@ftp:/var/log/proftpd# netstat -natp  
Connexions Internet actives (serveurs et établies)  
Proto Recv-Q Send-Q Adresse locale Adresse distante Etat PID/Program name  
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 92/sshd: /usr/sbin/  
tcp 0 0 127.0.0.1:25 0.0.0.0:* LISTEN 341/exim4  
tcp6 0 0 :::21 :::* LISTEN 751/proftpd: (accep  
tcp6 0 0 :::22 :::* LISTEN 92/sshd: /usr/sbin/  
tcp6 0 0 :::1:25 :::* LISTEN 341/exim4
```

Nous remarquons que le protocole FTP utilise le port 21. Nous modifions ensuite le fichier `/etc/proftpd/proftpd.conf` en décommentant la ligne `DefaultRoot ~` (il se peut qu'il faille rajouter un espace entre la directive et le tilde « ~ »). Cette instruction permet de bloquer l'utilisateur dans son répertoire personnel.

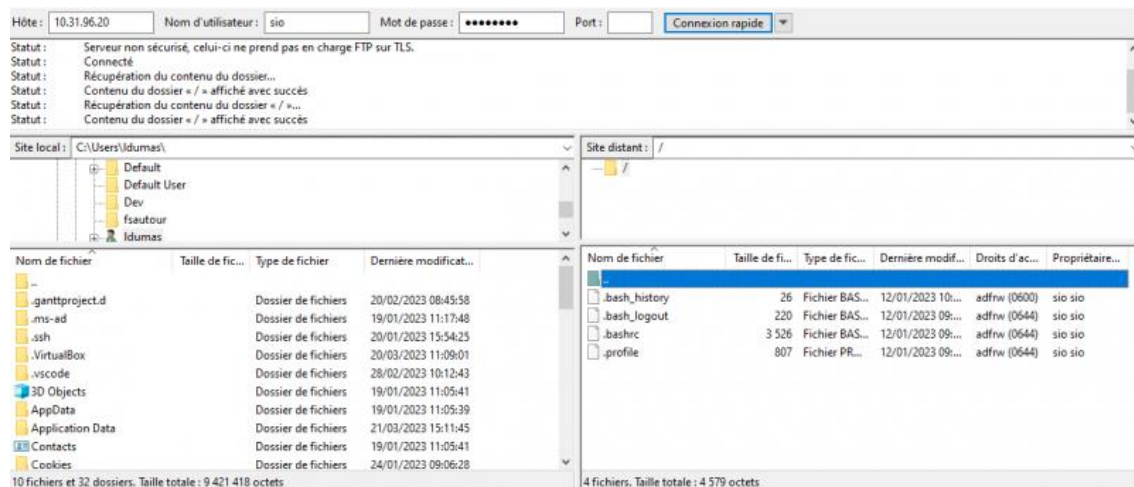
```
# Use this to jail all users in their homes  
DefaultRoot ~
```

Nous redémarrons proftpd afin que les changements soient effectifs :

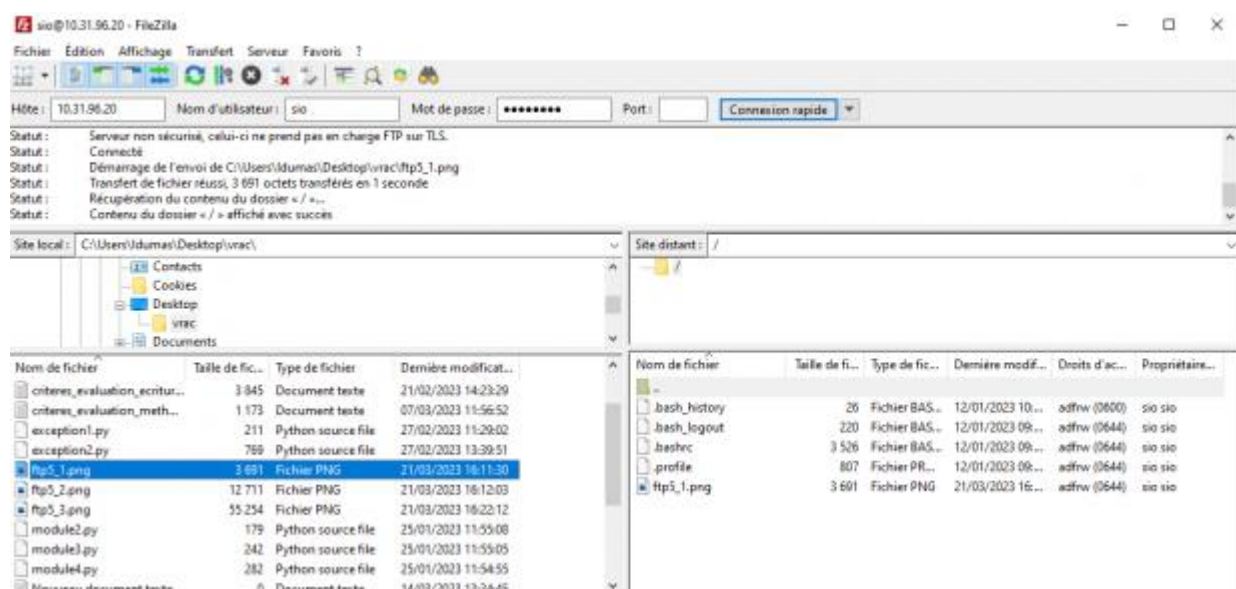
```
systemctl restart proftpd
```



En utilisant le logiciel FileZilla (solution FTP open source), nous nous connectons avec le compte sio sur notre conteneur FTP afin de nous assurer du bon fonctionnement de ce dernier :



Nous vérifions également que nous pouvons transférer des fichiers de notre client à notre serveur (et inversement) et que nous sommes bloqués dans notre répertoire personnel, c'est-à-dire que nous ne possédons pas les droits pour remonter l'arborescence du système de notre serveur FTP depuis notre répertoire personnel :



Configuration des utilisateurs anonymes

Maintenant que nous avons accès à notre serveur FTP avec le compte SIO, nous devons configurer les utilisateurs anonymes afin que les utilisateurs ne possédant pas de compte puissent déposer des fichiers sur le serveur FTP. Pour cela, nous décommentons le bloc « Anonymous » à la fin du fichier `/etc/proftpd/proftpd.conf` :

```
<Anonymous /home/ftpdocs>
    User ftp
    Group nogroup
    UserAlias anonymous ftp
    DirFakeUser on ftp
    DirFakeGroup on ftp
    RequireValidShell off
    MaxClients 10
    DisplayLogin welcome.msg
    DisplayChdir .message

    <Directory *>
        <Limit WRITE>
            DenyAll
        </Limit>
    </Directory>
</Anonymous>
```

```
<Anonymous /home/ftpdocs>
    User ftp
    Group nogroup
    # We want clients to be able to login with "anonymous" as well as "ftp"
    UserAlias anonymous ftp
    # Cosmetic changes, all files belongs to ftp user
    DirFakeUser on ftp
    DirFakeGroup on ftp

    RequireValidShell off

    # limit the maximum number of anonymous logins
    MaxClients 10

    # We want 'welcome.msg' displayed at login, and '.message' displayed
    # in each newly chdired directory.
    DisplayLogin welcome.msg
    DisplayChdir .message

    # limit WRITE everywhere in the anonymous chroot
    <Directory *>
        <Limit WRITE>
            DenyAll
        </Limit>
    </Directory>

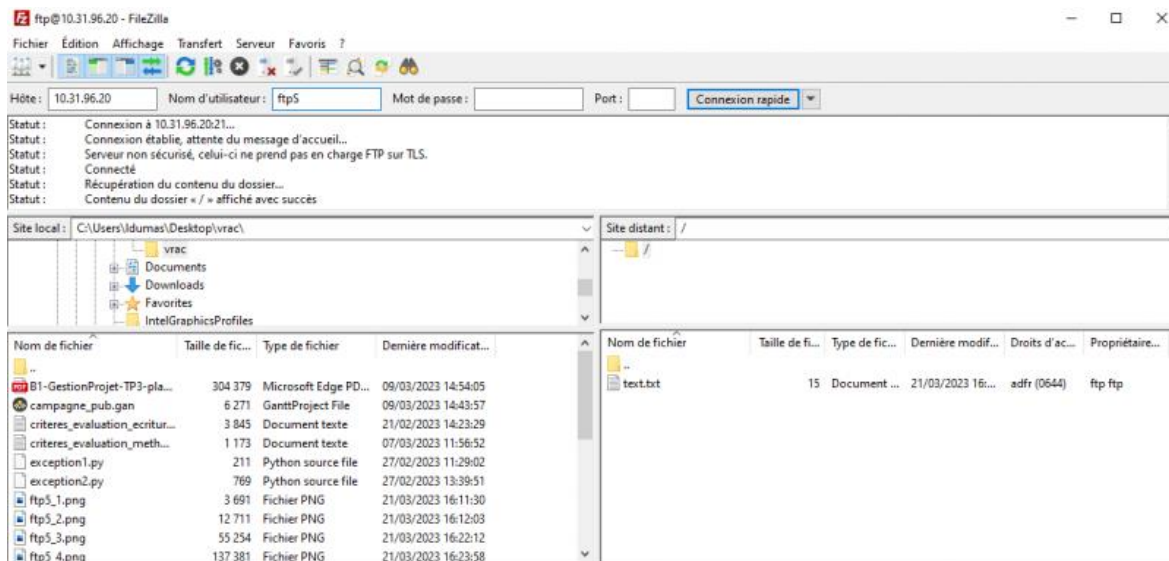
    # Uncomment this if you're brave.
    # <Directory incoming>
    #     # Umask 022 is a good standard umask to prevent new files and dirs
    #     # (second parm) from being group and world writable.
    #     Umask022 022
    #     <Limit READ WRITE>
    #         DenyAll
    #     </Limit>
    #     <Limit STOR>
    #         AllowAll
    #     </Limit>
    # </Directory>

</Anonymous>
```



Afin de rediriger les utilisateurs sur le dossier /home/ftpdocs, nous rajoutons le chemin de ce dossier dans la balise ouvrante du bloc « Anonymous » en effaçant l'ancien chemin.

Pour accéder au serveur ftp avec un compte anonyme, nous utilisons l'identifiant ftp sans mot de passe :



Configuration des Virtual Hosts

Nous avons configuré précédemment les adresses IP des différents Virtual Hosts. Nous allons maintenant ajouter deux nouveaux utilisateurs qui seront capables d'interagir avec le Virtual Host correspondant :

```
useradd intra  
passwd intra  
useradd extra  
passwd extra
```

Nous créons pour chaque utilisateur un répertoire personnel :

```
mkdir -p /srv/ftp/intranet  
mkdir -p /srv/ftp/extranet
```

Nous attribuons chaque nouvel utilisateur à son répertoire personnel :

```
chown -R intra /srv/ftp/intranet  
chown -R extra /srv/ftp/extranet
```

Nous vérifions que les dossiers possèdent les droits 755 et les modifions si nécessaire.

```
drwxr-xr-x 2 extra root 4096 17 nov. 07:52 extranet  
drwxr-xr-x 2 intra root 4096 16 nov. 10:09 intranet
```



Nous devons à présent créer les Virtual Hosts dans le fichier `/etc/proftpd/virtuals.conf`.

```
# VirtualHost intranet
<VirtualHost intra.m2l.org>

ServerAdmin      admin@m2l.org
ServerName       "FTP INTRANET"
User             intra
Group            intra

<Limit LOGIN>
    Order Allow,Deny
    Allowgroup intra
    Deny from all
</Limit>

Umask            022
TransferLog       /var/log/proftpd/xfer/ftp-intranet.m2l.org
MaxLoginAttempts  10
DefaultRoot       /srv/ftp/intranet
AllowOverwrite    yes

</VirtualHost>
```



```
# VirtualHost extranet
<VirtualHost extra.m2l.org>

ServerAdmin      admin@m2l.org
ServerName       "FTP EXTRANET"
User             extra
Group            extra

<Limit LOGIN>
    Order Allow,Deny
    Allowgroup extra
    Deny from all
</Limit>

Umask            022
TransferLog       /var/log/proftpd/xfer/ftp-extranet.m2l.org
MaxLoginAttempts  10
DefaultRoot       /srv/ftp/extranet
AllowOverwrite    yes

</VirtualHost>
```

Nous activons maintenant le fichier des Virtual Hosts en décommentant la ligne suivante dans le fichier `/etc/proftpd/proftpd.conf` :

```
Include /etc/proftpd/virtuals.conf
```

```
#
# Useful to keep VirtualHost/VirtualRoot directives separated
#
Include /etc/proftpd/virtuals.conf
```

Nous pouvons redémarrer notre service :

```
systemctl restart proftpd
```

Grâce à l'outil FileZilla, nous vérifions que nous puissions nous connecter avec les différents utilisateurs dans leur répertoire personnel.

