

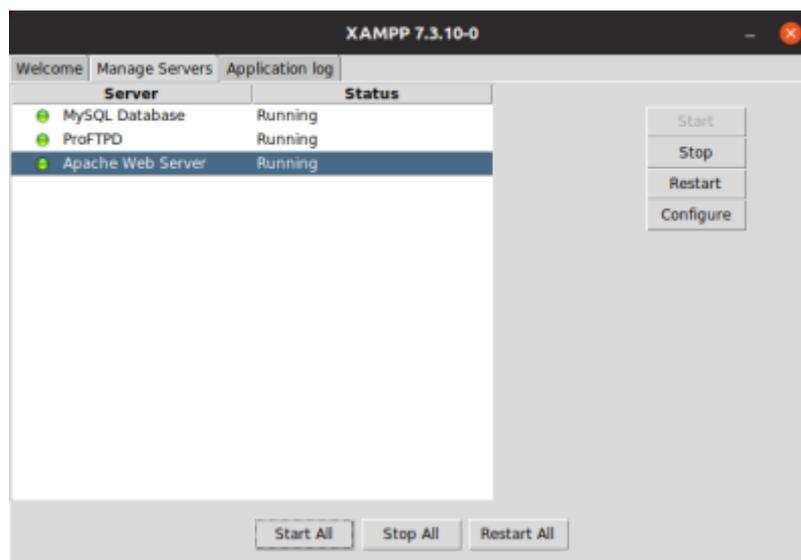
Logiciel



# XAMPP : un serveur LAMP (Apache + MariaDB + PHP + Perl)

Le paquet open source **XAMPP** est un serveur LAMP complet et préconfiguré. Il contient :

- un serveur HTTP **Apache**
- une base de données **MariaDB**
- **PHP**
- **Perl**
- et un serveur FTP : **ProFTPd**.



XAMPP est très pratique pour créer un serveur local de développement : il fournit au néophyte un serveur de test en quelques clics, sans avoir à choisir ni à connaître les différents services à installer.

Il est multiplateforme : il existe pour Linux, Windows et OS X

Tout est installé de façon conjointe avec les emplacements et autorisations qui vont bien.



**Remarque préliminaire importante** : ne demandez pas à XAMPP ce pour quoi il n'est pas fait !

XAMPP n'est **pas un serveur de production** mais un petit serveur local perso. Pour cet usage, il est parfait.

**Pour un serveur de production définitif, mieux vaut**

**LAMP** qui s'installe via les dépôts, plus riche de possibilités et de réglages, en particulier pour la sécurité.

Cependant, le serveur XAMPP installe des valeurs sûres :

- [Apache 2](#) : un serveur web HTTP,
- [MariaDB](#) avec [phpMyAdmin](#),
- [SQLite](#) : une base de données SQL
- [PHP](#),
- Perl,
- [ProFTPD](#)
- et de nombreuses bibliothèques



Les sites qui fonctionnent sur XAMPP fonctionneront donc sur LAMP : il suffit de copier le répertoire et éventuellement de déplacer la base de données. A partir de là, vous pourrez mettre en place les réglages.

Vous pouvez cependant modifier XAMPP moyennant quelques "bidouilles" et une bonne connaissance de sa structure.

## Pré-requis

- Une connexion internet

## Installation

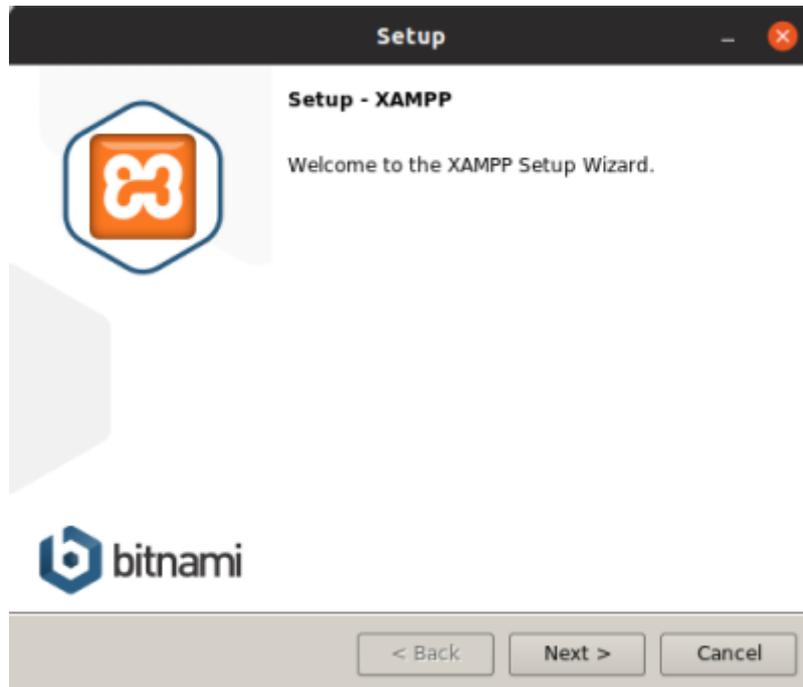
Tout XAMPP tient en un seul répertoire : **/opt/lampp**.

Son installation est donc d'une extrême simplicité (de même que sa désinstallation = supprimer le répertoire **/opt/lampp**).

Sur la page <https://www.apachefriends.org/fr/index.html>, choisissez la version 32 ou 64 bits selon votre système d'exploitation Linux.

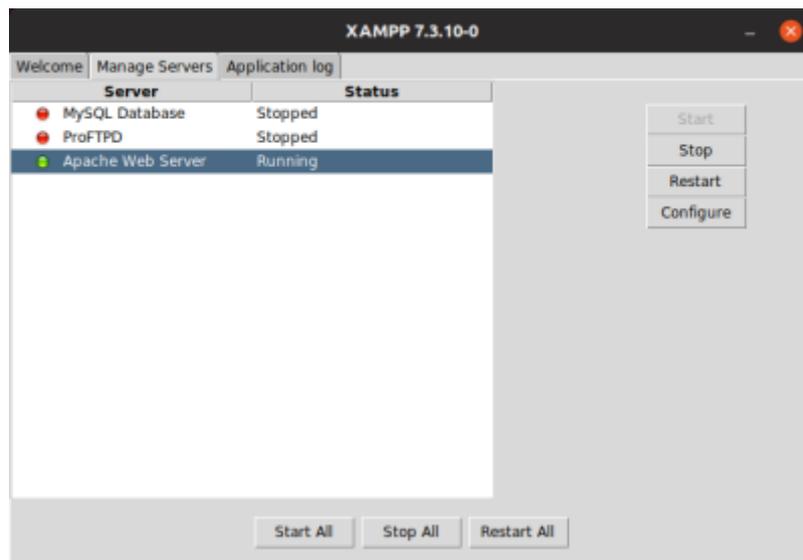
Modifiez les autorisations du programme d'installation et lancez-le en mode super-utilisateur :

```
$ cd /répertoire/de/téléchargement/  
$ sudo chmod +x xampp-linux-*-installer.run  
$ sudo ./xampp-linux-*-installer.run
```

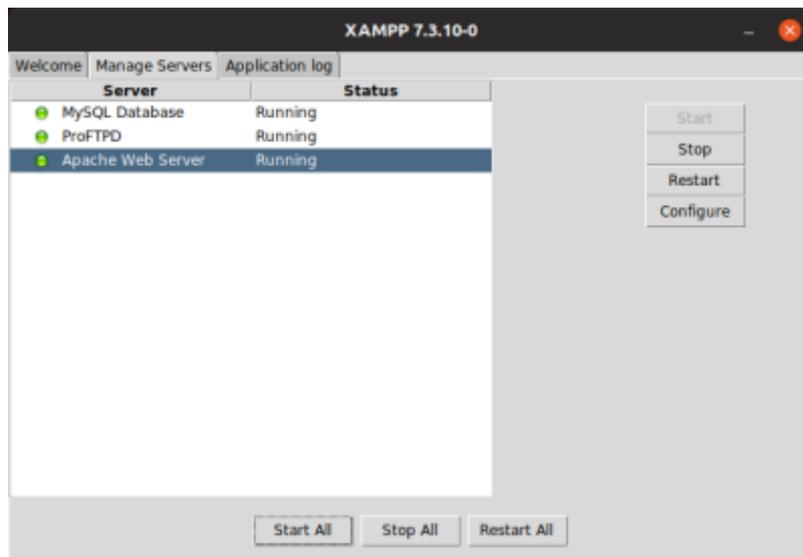


Acceptez toutes les étapes en cliquant sur Next.

Pour la dernière fenêtre, cliquez sur Finish → le panneau de contrôle de XAMPP démarre :



Ouvrez l'onglet **Manage Servers** et cliquez sur Start All pour démarrer les serveurs :



Pour que Xampp se lance au démarrage, lancez dans un terminal :



```
$ sudo ln -s /opt/lampp/lampp  
/etc/init.d/lampp  
$ sudo update-rc.d lampp defaults
```

Au redémarrage, XAMPP sera démarré.

## Aliases à créer

Créez les aliases en ajoutant au fichier `~/.bash_aliases` les lignes :

`~/.bash_aliases`

```
...  
# Alias pour permettre sudo [un_alias]  
alias sudo='sudo '  
  
# Aliases pour Xampp  
alias lampp='/opt/lampp/lampp'  
alias ctlampp='/opt/lampp/manager-linux-x64.run'  
  
# Aliases pour proftpd  
alias ftpasswd='/opt/lampp/bin/ftpasswd'
```



L'alias `sudo='sudo '` permet d'utiliser les autres aliases en `sudo`, comme avec



```
$ sudo lampp start
```

Activez les alias par un

```
$ source ~/.bashrc
```

## Arborescence créée par l'installation

L'installation a créé l'arborescence :

```
$ tree /opt/lampp/ -d -L 1
/opt/lampp/
├── apache2
├── bin
├── build
├── cgi-bin
├── docs
├── error
├── etc
├── htdocs
├── icons
├── img
├── include
├── info
├── lib
├── libexec
├── licenses
├── logs
├── man
├── manual
├── modules
├── mysql
├── pear
├── php
├── phpmyadmin
├── proftpd
├── sbin
├── share
├── temp
└── var
```

28 directories

## Configuration



### Une suggestion

Pour vous simplifier la vie, créez des alias en éditant ou créant le fichier `~/.bash_aliases` pour y écrire :

`~/.bash_aliases`

```
# Alias pour permettre sudo
[un_alias]
alias sudo='sudo '

# Aliases pour Xampp
alias lampp='/opt/lampp/lampp'
alias ctlampp='/opt/lampp/manager-
linux-x64.run'

# Aliases pour proftpd
alias
ftpasswd='/opt/lampp/bin/ftpasswd'
```



On obtient les raccourcis suivants :

Description	Raccourci	Effet
<b>lampp</b>	sudo lampp ...	→ sudo /opt/lampp/lampp ...
<b>panneau de contrôle</b>	sudo ctlampp	→ sudo /opt/lampp/manager-linux-x64.run
<b>gérer les utilisateurs de proftpd</b>	sudo ftpasswd ...	→ sudo /opt/lampp/bin/ftpasswd ...

L'alias `sudo='sudo '` permet d'utiliser les autres alias en `sudo`, comme avec

```
$ sudo lampp start
```

Activez les alias par :

```
$ source ~/.bashrc
```

## Accès FTP : proFTPD

XAMPP inclut **proFTPD**, un serveur FTP open source pour faciliter le transfert de fichiers.

Voir :

- [Pro-ftp](#) : un serveur FTP open source pour Linux

- [gadmin-proftpd](#)

Le fichier de configuration est **/opt/lampp/etc/proftpd.conf**.

### Fichier de la distribution

#### [/opt/lampp/etc/proftpd.conf](#)



```
# This is a basic ProFTPD
configuration file (rename it to
# 'proftpd.conf' for actual use.
It establishes a single server
# and a single anonymous login. It
assumes that you have a user/group
# "daemon" and "ftp" for normal
operation and anon.

ServerName          "ProFTPD"
ServerType          standalone
DefaultServer       on

# Port 21 is the standard FTP port.
Port                21
# Umask 022 is a good standard
umask to prevent new dirs and files
# from being group and world
writable.
Umask               022

# To prevent DoS attacks, set the
maximum number of child processes
# to 30. If you need to allow more
than 30 concurrent connections
# at once, simply increase this
value. Note that this ONLY works
# in standalone mode, in inetd mode
you should use an inetd server
# that allows you to limit maximum
number of processes per service
# (such as xinetd)
MaxInstances        30

# Set the user and group that the
server normally runs at.
User daemon
#Group daemon

# Normally, we want files to be
```



```
overwriteable.  
<Directory /opt/lampp/htdocs/*>  
  AllowOverride          on  
</Directory>  
  
# only for the web servers content  
DefaultRoot /opt/lampp/htdocs  
  
<Limit SITE_CHMOD>  
  DenyAll  
</Limit>  
  
# daemon gets the password "xampp"  
UserPassword daemon 2TgxE8g184G9c  
  
# daemon is no normal user so we  
have to allow users with no real  
shell  
RequireValidShell off  
  
# daemon may be in /etc/ftpusers so  
we also have to ignore this file  
UseFtpUsers off
```

Nous choisissons de ne pas toucher au fichier principal et d'utiliser des fichiers de configuration que nous stockerons dans un répertoire **/opt/lampp/etc/conf.d/** qui n'est pas affecté par les mises à jour.

### Mise en place du répertoire **/etc/proftpd/conf.d/**

Donc :

1. Créez les répertoires **/opt/lampp/etc/conf.d/** :

```
$ sudo mkdir /opt/lampp/etc/conf.d/  
$ sudo mkdir /opt/lampp/etc/proftpd/
```

2. ajoutez la ligne suivante à la fin du fichier principal :

[/opt/lampp/etc/proftpd.conf](#)

```
(...)  
# Inclure les fichiers de configuration personnalisés  
Include /opt/lampp/etc/conf.d/
```

Tous les fichiers contenus dans le répertoire **/etc/proftpd/conf.d/** seront prises en compte.

- avec les droits d'administration, placez les fichiers de configuration dans le répertoire **/opt/lampp/etc/conf.d/** (basic.conf, anonymous.conf, mysql.conf, virtual.conf, etc.) :

```
$ sudo cp <fichier.conf> /opt/lampp/etc/conf.d
```

## Présentation

Supposons que vous ayez dans votre répertoire de base un script PHP simple, exemple.php, contenant le code suivant, que vous souhaitez transférer sur le serveur XAMPP.

~/exemple.php

```
<!-- exemple.php -->
<html>
<head></head>
<body>
  <h2><?php echo "Bonjour. Nous sommes le " . date('l'); ?>.</h2>
</body>
</html>
```

Pour transférer des fichiers via proFTPD, vous devez d'abord configurer les règles d'accès FTP. Suivez ces étapes.

- Ouvrez un terminal.
- Créez un groupe ftp qui contiendra les utilisateurs autorisés à télécharger des fichiers via FTP:

```
$ sudo groupadd ftp
```

- Ajoutez votre compte (moi dans cet exemple) au nouveau groupe. Ajoutez d'autres utilisateurs si nécessaire.

```
$ sudo usermod -a -G ftp moi
```

- Modifiez le propriétaire et les permissions du sous-répertoire htdocs/ du répertoire d'installation de XAMPP (généralement, /opt/lampp) pour le rendre accessible en écriture au groupe ftp.

```
$ cd /opt/lampp
$ sudo chown root:ftp htdocs
$ sudo chmod 775 htdocs
```



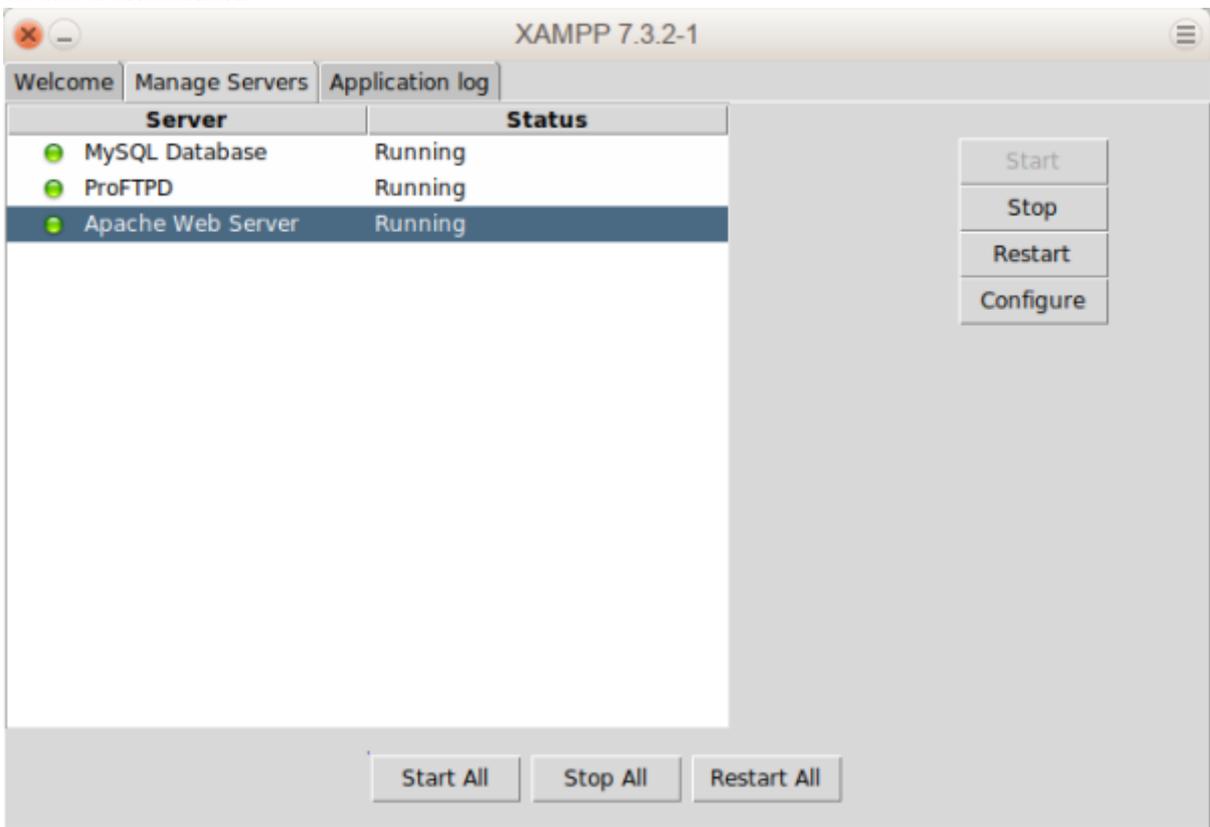
Si vous utilisez XAMPP avec un seul utilisateur qui est seul à transférer des fichiers via FTP, vous pouvez ignorer la création d'un nouveau groupe et simplement rendre cet utilisateur propriétaire du répertoire htdocs/. Pour

cela, utilisez la commande



```
$ sudo chown moi:moi htdocs.
```

5. Vérifiez dans le panneau de configuration de XAMPP, onglet Manage servers, que proFTPD est en cours d'exécution.



Vous pouvez maintenant transférer des fichiers sur le serveur XAMPP en procédant comme suit :

- Démarrez un client FTP et entrez les détails de la connexion comme ci-dessous.
  - adresse de l'hôte :
    - 127.0.0.1 si vous vous connectez au serveur à partir du même système
    - le nom d'hôte ou l'adresse IP du serveur XAMPP si vous vous connectez depuis un autre système.
  - port : 21
  - votre nom d'utilisateur et votre mot de passe Linux comme informations d'identification FTP.  
Votre client FTP se connecte maintenant au serveur dans le répertoire /opt/lampp/htdocs/, qui est la racine par défaut du serveur Web.
- Transférez les fichiers depuis votre ordinateur vers le serveur comme d'habitude. Si vous utilisez un client FTP graphique, vous pouvez généralement faire glisser le fichier d'un répertoire à l'autre. Si vous utilisez un client FTP de ligne de commande, vous pouvez utiliser la commande FTP PUT.
- Une fois le fichier transféré, il apparaît en naviguant sur <http://localhost/example.php>, comme indiqué ci-dessous :

## Utilisateurs virtuels

Créez le répertoire :

```
$ sudo mkdir /opt/lampp/etc/proftpd/
```

éditez avec les droits d'administration le fichier **/opt/lampp/etc/conf.d/global.conf** pour y ajouter les directives :

[/opt/lampp/etc/conf.d/global.conf](#)

```
DefaultRoot ~  
  
AuthUserFile /opt/lampp/etc/proftpd/ftpd.passwd  
AuthGroupFile /opt/lampp/etc/proftpd/ftpd.group
```

Vérifiez l'existence de l'utilisateur **www-data** et de son groupe :

```
$ id www-data  
uid=33(www-data) gid=33(www-data) groupes=33(www-data)
```

[Sinon, créez-le](#)

```
$ sudo groupadd www-data  
$ sudo useradd -g www-data -d /var/www -s /bin/false www-data
```

On constate que l'identifiant du groupe www-data est 33.

Créons l'utilisateur virtuel admiweb, de home /var/www/html, avec les uid et gid de www-data en lançant les commandes (le mot de passe du compte vous sera demandé) :

```
$ cd /opt/lampp/etc/proftpd  
$ sudo ftpasswd --passwd --name=admiweb --uid=33 --gid=33 --  
home=/var/www/html --shell=/bin/false  
ftpasswd: creating passwd entry for user admiweb  
  
ftpasswd: /bin/false is not among the valid system shells. Use of  
ftpasswd: "RequireValidShell off" may be required, and the PAM  
ftpasswd: module configuration may need to be adjusted.  
  
Password:  
Re-type password:  
  
ftpasswd: entry created
```

Vérifiez que l'utilisateur courant peut se connecter :

```
$ ftp localhost
```

```
Connected to localhost.localdomain.
220 ProFTPD 1.3.4c Server (ProFTPD) [::ffff:127.0.0.1]
Name (localhost:xxxxxxx):
331 Password required for xxxxxxx
Password:
230 User xxxxxxx logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful
150 Opening ASCII mode data connection for file list
-rw-r--r--  1 root    root      3607 Feb 27  2017 applications.html
-rw-r--r--  1 root    root      177 Feb 27  2017 bitnami.css
drwxr-xr-x 21 root    root      4096 Apr  4 11:48 dashboard
-rw-r--r--  1 root    root     30894 May 11  2007 favicon.ico
drwxr-xr-x  2 root    root      4096 Apr  4 11:48 img
-rw-r--r--  1 root    root      260 Jul  9  2015 index.php
drwxr-xr-x  2 daemon daemon    4096 Apr  4 11:48 webalizer
226 Transfer complete
ftp> bye
221 Goodbye.
```

De même pour admiweb :

```
$ ftp localhost
Connected to localhost.localdomain.
220 ProFTPD 1.3.4c Server (ProFTPD) [::ffff:127.0.0.1]
Name (localhost:xxxxxxx): admiweb
331 Password required for admiweb
Password:
230 User admiweb logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful
150 Opening ASCII mode data connection for file list
-rw-r--r--  1 root    root      3607 Feb 27  2017 applications.html
-rw-r--r--  1 root    root      177 Feb 27  2017 bitnami.css
drwxr-xr-x 21 root    root      4096 Apr  4 11:48 dashboard
-rw-r--r--  1 root    root     30894 May 11  2007 favicon.ico
drwxr-xr-x  2 root    root      4096 Apr  4 11:48 img
-rw-r--r--  1 root    root      260 Jul  9  2015 index.php
drwxr-xr-x  2 daemon daemon    4096 Apr  4 11:48 webalizer
226 Transfer complete
ftp> bye
221 Goodbye.
```

Vous pouvez maintenant créer tous les utilisateurs nécessaires en répétant cette commande.

## Configuration

La configuration de proftpd se fait dans **/opt/lampp/etc/** <sup>1)</sup>.

Pour faciliter la configuration, ajouter la répertoire **/opt/lampp/etc/conf.d** :

```
$ sudo mkdir /opt/lampp/etc/conf.d
```

Nous y placerons nos fichiers de configuration en les appelant par des Include. Ainsi, ils ne seront pas affectés par les mises à jour.

Liste des directives : Voir <http://www.proftpd.org/docs/directives/linked/by-name.html>

## Contextes

Contextes :

- configuration du serveur

<Global>

<Global>

Directives à appliquer dans l'ensemble de démon

Contexte :

- configuration du serveur
- <VirtualHost>



++++ <VirtualHost>

<VirtualHost>...</VirtualHost>

Définit un serveur FTP virtuel

Contexte :

- configuration du serveur

Le bloc VirtualHost crée un ensemble de directives qui s'appliquent à un nom d'hôte ou à une adresse IP particulière.

Souvent utilisé avec des IP ou des interfaces réseau factices pour établir un ou plusieurs serveurs virtuels qui s'exécutent tous sur le

même ordinateur physique.

Avec la directive `Port` dans un bloc `VirtualHost`, on peut créer un serveur virtuel à la même adresse que le serveur maître, mais sur un port TCP particulier.

Au démarrage de `proftpd`, les connexions au serveur virtuel sont gérées de deux façons selon la paramètre `ServerType` :

- **inetd** : Le démon examine l'adresse de destination et le port de la connexion entrante transmise par `inetd`.  
Si la connexion correspond à un hôte virtuel configuré, la connexion est desservie selon la configuration appropriée.  
Si aucun hôte virtuel ne correspond et que le serveur principal ne correspond pas, le client est informé qu'aucun serveur n'est disponible pour traiter la demande et est déconnecté.
- **standalone** : Après avoir analysé le fichier de configuration, le démon commence à écouter les connexions sur tous les ports configurés, en générant les processus enfants nécessaires pour gérer les connexions du serveur principal ou de tous les serveurs virtuels.  
En raison de la méthode utilisée par le démon pour écouter les connexions en mode autonome, il est possible de prendre en charge un nombre extrêmement important de serveurs virtuels, dépassant potentiellement le nombre de descripteurs de fichiers par processus.  
Cela est dû au fait qu'un seul descripteur de fichier est utilisé pour écouter chaque port configuré, quel que soit le nombre d'adresses surveillées.  
Notez qu'il peut être nécessaire d'augmenter la valeur `tcpBackLog` sur des serveurs fortement chargés afin d'éviter les connexions client rejetées par le noyau.



++++ <Anonymous>

<Anonymous> ... </Anonymous>

Définit un serveur anonyme

Contexte :

- configuration du serveur
- <VirtualHost>
- <Global>

Le bloc de configuration Anonymous crée une connexion FTP anonyme.

Les paramètres du répertoire racine spécifient le répertoire dans lequel le démon va d'abord aller (chdir), puis y rester enfermé après la connexion.

Une fois l'opération chroot terminée, les répertoires de niveau supérieur ne sont plus accessibles au démon en cours d'exécution (et donc à l'utilisateur connecté).

Par défaut, proftpd suppose une connexion anonyme si le client distant tente de se connecter en tant qu'utilisateur en cours, (sauf si c'est root, auquel cas les connexions anonymes sont interdites)

Pour forcer les connexions anonymes à être liées à un utilisateur autre que l'utilisateur en cours, voir les directives User et Group.

De plus, si une directive User ou Group est présente dans un bloc <Anonymous>, le démon bascule de manière permanente sur l'uid / gid spécifié avant l'opération de chroot.

Normalement, les connexions anonymes ne sont pas obligées de s'authentifier avec un mot de passe, mais sont supposées entrer une adresse électronique valide au lieu d'un mot de passe enregistré.

Si ce comportement n'est pas souhaitable pour un bloc de configuration <Anonymous> donné, vous pouvez le remplacer via la directive AnonRequirePassword.



+++ <Directory>

<Directory> ... </Directory>

Directives limitées à un répertoire

Contexte :

- configuration du serveur

- <Global>
- <VirtualHost>
- <Anonymous>

Bloc de directives pour le répertoire spécifié et ses sous-répertoires.

Si le nom du répertoire se termine par `"/*`, le bloc ne s'applique qu'au contenu et non au répertoire lui-même.

Les blocs `<Directory>` ne peuvent pas être imbriqués.

Les chemins doivent toujours être absolus (sauf dans un bloc `<Anonymous>`) et ne doivent pas faire référence à des liens symboliques.

Si les chemins commencent par `'~'` non immédiatement suivi d'un nom d'utilisateur, le caractère `'~'` est remplacé par le répertoire de base de l'utilisateur quand il s'authentifie.

Cette fonction est pas prise en charge dans un bloc `<Anonymous>`.

Exemple :



```
# Utilisation par défaut de la
directive directory
<Directory /users/robroy/private>
  HideNoAccess on
</Directory>
```

```
# Exemple avec extension de nom
d'utilisateur
<Directory ~/anon-ftp>
  <Limit WRITE>
    DenyAll
  </Limit>
</Directory>
```

++++ <Limit>

<Limit>

Définit les commandes / actions à contrôler

Contexte :

- configuration du serveur
- <VirtualHost>
- <Directory>

- <Anonymous>
- <Global>
- .ftpass

Le bloc de configuration Limit permet d'imposer des restrictions d'accès à une ou plusieurs commandes FTP, dans un contexte donné.

Les limites s'étendent vers le bas : une configuration Limit dans le contexte de configuration du serveur s'applique à tous les blocs <Directory> et <Anonymous> qui sont également dans la configuration, jusqu'à ce qu'il soit remplacé par un bloc <Limit> de niveau inférieur.

Vous pouvez spécifier n'importe quel nombre de paramètres de commande auxquels le contenu du bloc <Limit> sera appliqué.

La commande peut être n'importe quelle commande FTP valide, mais est généralement l'une des suivantes :

; CWD (Change Working Directory)



Envoyé par le client lors du changement de répertoire.

; MKD / XMKD (MaKe Directory)

Envoyé par le client pour créer un nouveau répertoire.

; RNFR (ReName FRom), RNTD (ReName TD)

Envoyé en paire par le client pour renommer une entrée de répertoire.

; DELE (DELEte)

Envoyé par le client pour supprimer un fichier.

; RMD / XRMD (ReMove Directory)

Envoyé par le client pour supprimer un répertoire.

; RETR (RETRieve)

Transférer un fichier du serveur vers le client.

; STOR (STORe)

Transférer un fichier du client vers le serveur.

De plus, les groupes de commandes suivants sont acceptés. Leur priorité est inférieure à celle des commandes réelles, ce qui signifie qu'une limite de commande réelle aura priorité sur le groupe de commandes.

- READ Commandes FTP traitant de la lecture de fichiers (liste de répertoires non incluse) :
  - RETR
  - SITE
  - SIZE
  - STAT
- WRITE Commandes FTP traitant de l'écriture / création / suppression de fichiers ou de répertoires :
  - APPE
  - DELE
  - MKD
  - RMD
  - RNT0
  - STOR
  - XMKD
- XRMD DIRS Commandes FTP traitant du listing des répertoires :
  - CDUP
  - CWD
  - LIST
  - MDTM
  - NLST
  - PWD
  - RNFR
  - XCUP
  - XCWD
- XPWD ALL Commandes FTP (identiques à READ WRITE DIRS). Notez que ce groupe a la priorité la plus basse de tous ; il ne remplacera pas une limite imposée par un autre groupe de commandes (par exemple, DIRS).
- Enfin, une commande spéciale qui peut contrôler l'accès à la connexion : LOGIN Connexion ou connexion au serveur.  
On peut appliquer une <Limit> à cette pseudo-commande pour autoriser ou refuser la connexion



initiale ou la connexion au contexte.

Il n'a aucun effet et est ignoré lorsqu'il est utilisé dans un contexte autre que la configuration du serveur, <VirtualHost> ou <Anonymous> (c'est-à-dire que son utilisation dans un contexte <Directory> n'a pas de sens).

Les restrictions de la commande <Limit> ne doivent pas être confondues avec les droits d'accès aux fichiers / répertoires.

Bien que les limites puissent être utilisées pour restreindre une commande à un certain répertoire, elles ne peuvent pas remplacer les autorisations de fichiers inhérentes au système de fichiers / d'exploitation de base.

Les commandes FTP suivantes ne peuvent pas être restreintes via <Limit> :

- ABOR
- HELP
- MODE (non implémenté, toujours S)
- NOOP PASS (utiliser <Limit LOGIN>)
- PASV
- PORT
- QUIT
- REST (utiliser AllowRetrieveRestart, AllowStoreRestart)
- STRU (non implémenté, toujours F)
- SYST
- TYPE
- USER (utiliser <Limit LOGIN>)



++++ [.ftppass](#)

.ftppass

</WRAP> == Directives utiles == <WRAP center round tip 60%> Quelques directives utiles : ++++ UseIPv6

- Active ou désactive le support IPv6
- Syntaxe :

```
UseIPv6 [ "on"|"off" ]
```

- Valeur par défaut :

```
UseIPv6 on
```

- Contexte :
  - configuration du serveur
- exemple :

```
UseIPv6 on
```

## User

- Définit l'utilisateur sous lequel le démon s'exécutera  
Dans un bloc <VirtualServer>, proftpd s'exécutera en tant qu'utilisateur spécifié sur les connexions vers l'adresse ou le port du serveur virtuel.  
Dans un bloc <Anonymous>, proftpd établira une connexion anonyme lorsqu'un utilisateur essaiera de se connecter avec l'ID utilisateur spécifié  
Note : Lorsqu'un utilisateur unix autorisé se connecte, tous les privilèges antérieurs sont libérés, le démon bascule de manière permanente vers l'uid de l'utilisateur connecté et ne peut plus revenir à la racine ou à tout autre utilisateur.
- Syntaxe :



```
User [ User userid]
```

- Valeur par défaut : aucune
- Contexte :
  - configuration du serveur
  - <VirtualHost>
  - <Anonymous>
  - <Global>
- exemple :

```
User ftpuser
```

## Group

- Définit le groupe sous lequel le démon s'exécutera  
Dans un bloc <VirtualServer>, proftpd s'exécutera en tant que groupe spécifié sur les connexions vers l'adresse ou le port du serveur virtuel.

Dans un bloc `<Anonymous>`, `proftpd` établira une connexion anonyme lorsqu'un utilisateur essaiera de se connecter avec l'ID groupe spécifié

Note : Lorsqu'un utilisateur unix autorisé se connecte, tous les privilèges antérieurs sont libérés, le démon bascule de manière permanente vers le gid de l'utilisateur connecté et ne peut plus revenir à la racine ou à tout autre utilisateur.

- Syntaxe :

```
Group [ Group groupid]
```

- Valeur par défaut : aucune
- Contexte :
  - configuration du serveur
  - `<VirtualHost>`
  - `<Anonymous>`
  - `<Global>`

- exemple :

```
Group                                ftpgroup
```



### UseFtpUsers

- Bloc basé sur `/etc/ftpusers`  
Le serveur FTP vérifie un fichier d'autorisation (généralement `/etc/ftpusers`) lorsqu'un client tente de s'authentifier.  
Si le nom de l'utilisateur est trouvé dans ce fichier, l'accès FTP est refusé.

- Syntaxe :

```
UseFtpUsers [ UseFtpUsers on|off]
```

- Valeur par défaut :

```
UseFtpUsers on
```

- Contexte :
  - configuration du serveur
  - `<Anonymous>`
  - `<VirtualHost>`
  - `<Global>`

- exemple :

```
UseFtpUsers on
```

### RequireValidShell

- Autoriser les connexions basées sur /etc/shells  
La directive RequireValidShell configure le serveur, l'hôte virtuel ou la connexion anonyme pour autoriser ou refuser les connexions qui ne disposent pas d'un fichier shell répertorié dans /etc/shells.
- Syntaxe :

```
RequireValidShell [ RequireValidShell  
on|off]
```

- Valeur par défaut :

```
RequireValidShell on
```



- Contexte :
  - configuration du serveur
  - <VirtualHost>
  - <Anonymous>
  - <Global>
- exemple :

```
RequireValidShell on
```

### DefaultRoot

- Contrôle le répertoire racine attribué à un utilisateur lors de la connexion.  
Si DefaultRoot est défini sur un répertoire autre que "/", une opération de chroot est effectuée immédiatement après l'authentification du client. Cela permet d'isoler le client dans une zone de l'espace de fichiers du système hôte.  
Le répertoire racine spécifié doit commencer par un / ou peut être le caractère magique '~' qui signifie que le client est emprisonné dans son répertoire personnel. DefaultRoot ne peut pas être utilisé dans les blocs de configuration <Anonymous> car la directive

<Anonymous> contient explicitement un répertoire racine pour les connexions anonymes.  
L'argument facultatif group-expression limite la directive DefaultRoot à un groupe unix. L'expression a le format :

```
[!]group-name1[, [!]group-name2[, ...]]
```

L'expression est un ET logique booléen, de sorte que chaque membre de l'expression doit avoir la valeur TRUE logique pour que la directive DefaultRoot s'applique.  
Le caractère spécial '!' nie l'appartenance à un groupe.

- Syntaxe :

```
DefaultRoot [ directory [group-expression]]
```

- Valeur par défaut :

```
DefaultRoot /
```



- Contexte :
  - configuration du serveur
  - <VirtualHost>
  - <Global>
- exemple :

```
DefaultRoot ~ !pfoo
```

Utilisateur isomé dans son home, sauf pour le groupe pfoo

## umask

- Définit le masque des autorisations pour les fichiers et répertoires nouvellement créés dans un contexte donné.  
Les arguments doivent être un nombre octal, au format 0xxx.  
Un deuxième argument facultatif peut spécifier un Umask pour la création de répertoires.  
Si aucun second argument n'est spécifié, les répertoires sont créés en utilisant l'Umask du premier

argument.

Les autorisations du fichier téléchargé peuvent être modifiées par une commande SITE CHMOD qui modifie le mode du fichier téléchargé. La syntaxe de la commande est :

```
SITE CHMOD <mode> <file>
```

.

- Syntaxe :

```
Umask [ Umask file octal-mask [directory octal-mask]]
```

- Valeur par défaut : Aucune

- Contexte :

- configuration du serveur
- <Anonymous>
- <VirtualHost>
- <Directory>
- <Global>
- .ftppass



- exemple :

```
umask 0027 0027
```

## AuthOrder

- La directive AuthOrder configure les noms des modules auth et l'ordre dans lequel ils seront vérifiés lors de l'authentification d'un utilisateur.

Au moins un nom de module doit être donné; il n'y a pas de nombre maximum de modules pouvant être listés.

Les noms de module répertoriés doivent correspondre au nom complet du fichier source, par exemple.

“mod\_auth\_unix.c”.

Pour voir une liste complète des noms de modules, utilisez “

```
proftpd -l
```

”.\N'utilisez pas “mod\_auth.c”, car ce module est le module frontal d'authentification et est nécessaire. Pour qu'un module d'authentification fasse autorité,

ajouter un astérisque (\*) après le nom du module. C'est généralement fait pour le module "mod\_auth\_pam.c", pour que la connexion échoue en cas d'échec de la vérification PAM.

- Syntaxe :

```
AuthOrder [ module-name...]
```

- Valeur par défaut : Aucune
- Contexte :
  - configuration du serveur
  - <VirtualHost>
  - <Global>
- exemple :

```
AuthOrder  
mod_auth_pam.c* mod_auth_unix.c
```

</WRAP>



### Serveurs virtuels

Les serveurs virtuels, <sup>2)</sup>, servent plusieurs adresses ou sites sur un même ordinateur hôte. Ils sont basés sur des IP et des noms.

Les mots "démon" et "serveur" ne sont pas identiques.

- Un "démon" est un processus de longue durée sur une machine hôte
- Un "serveur" est un processus qui fournit un service



Un processus proftpd autonome, peut être considéré à la fois comme un "démon" et un "serveur". Un "serveur virtuel" n'est pas un processus séparé qui semble unique aux clients distants.

Le fichier de configuration proftpd.conf comporte trois contextes serveur (ou sections) :

- **<VirtualHost>**
- **<Global>**,
- et **configuration du serveur.**

Le contexte <VirtualHost> définit un hôte virtuel lié à une adresse IP ou un nom DNS.

- Par exemple :

```
<VirtualHost 1.2.3.4>
...
</VirtualHost>
```

définit un serveur virtuel à utiliser quand un client distant se connecte à l'adresse IP 1.2.3.4.

- ou avec un nom DNS :

```
<VirtualHost ftp.mydomain.com>
...
</VirtualHost>
```



Au démarrage, proftpd résout le nom DNS fourni en son adresse IP et utilise cette adresse IP. ProFTPd crée automatiquement une directive ServerAlias pour ce nom DNS, ce qui permet de définir plusieurs sections <VirtualHost> avec le même nom DNS.

Pour utiliser le même vhost pour plusieurs noms, répertoriez ces noms avec la directive ServerAlias comme suit :

```
<VirtualHost ftp.mydomain.com>
# Utilisez cet hôte aussi pour d'autres
noms
ServerAlias ftp.mydomain.org
ftp2.mydomain.com ftp.otherdomain.org
...
</VirtualHost>
```

Exemple de Virtualhost fourni par proftpd à placer dans conf.d |

</opt/lampp/etc/conf.d/virtuals.conf>

```
# Exemple de configuration Proftpd
pour les hôtes virtuels et les
```

```
racines virtuelles.

# Notez que le protocole FTP
nécessite un hôte virtuel basé sur
IP, et non sur un nom.

# Un exemple d'hôte virtuel
générique.
<VirtualHost ftp.server.com>
    # adresse électronique de
l'administrateur de l'hôte virtuel
    ServerAdmin
ftpmaster@server.com

    # chaîne qui s'affiche à la
connexion d'un utilisateur au
serveur virtuel
    ServerName                "Grande
archive FTP"

    # chemin vers le journal de
transfert
    TransferLog
/var/log/proftpd/xfer/ftp.server.co
m
    # nombre de tentatives de mot
de passe autorisées avant
déconnexion
    MaxLoginAttempts          3
    # Autoriser les connexions
basées sur /etc/shells
    RequireValidShell          no
    # répertoire racine attribué à
un utilisateur lors de la connexion
    DefaultRoot
/srv/ftp_root
    # Permettre le remplacement des
fichiers
    AllowOverwrite             yes
</VirtualHost>

# Le module vroot n'est pas
obligatoire, mais peut être utile
pour les répertoires partagés.

#<IfModule mod_vroot.c>
    #VRootEngine on
    #DefaultRoot ~
    #VRootAlias upload
/var/ftp/upload
```



```
#<VirtualHost a.b.c.d>  
  #VRootEngine on  
  #VRootServerRoot  
/etc/ftpd/a.b.c.d/  
  #VRootOptions allowSymlinks  
  #DefaultRoot ~  
#</VirtualHost>  
#</IfModule>
```

++++

### section <Global>

Supposons de nombreuses sections <VirtualHost> avec la même configuration pour chaque hôte virtuel (sections <Directory> communes, les paramètres DefaultRoot, etc.) Au lieu d'inclure la même configuration plusieurs fois, on peut utiliser la section <Global> :

```
<Global>  
  ...  
</Global>
```



Le contenu d'une section <Global> s'applique à chaque configuration de serveur et à chaque <VirtualHost>.

### Configuration du serveur

Le contexte "configuration serveur" s'applique à tout ce qui n'est pas dans une section <VirtualHost> ou <Global>.



Proftpd démarre même avec un fichier proftpd.conf complètement vide : essayez ! Le démon utilisera tous les paramètres par défaut, ce qui n'est généralement pas souhaitable, mais possible. Le contexte de serveur par défaut est appelé "configuration de serveur".

Les directives de configuration de la section "configuration du serveur" ne s'appliquent pas en dehors du contexte. C'est ce à quoi sert la section <Global>.

La directive **DefaultAddress** spécifie l'adresse IP que l'hôte de la "configuration serveur" doit écouter.

Par défaut, chaque serveur écoute le port 21, le port standard IANA pour FTP. Pour changer le port, utilisez la directive `Port`. Comme mentionné ailleurs, si vous avez plusieurs sections `<VirtualHost>` différentes utilisant la même adresse mais des ports différents, vous devez vous assurer de laisser vide chaque numéro de `Port-1`. La RFC 959 spécifie que le port source pour un transfert de données actif (à lire ici) doit être `L-1`, `L` étant le port sur lequel votre serveur écoute. En outre, comme indiqué dans la documentation du port, Dans n'importe quel contexte de serveur,

#### Port 0

désactive ce serveur. C'est parfois utilisé pour désactiver la "configuration serveur".

Lorsqu'un client contacte `proftpd`, le serveur cherche quelle configuration utiliser dans la liste des `vhosts` configurés, en recherchant un `vhost` dont l'adresse IP correspond. Si un `vhost` correspond, on utilise cette configuration. Sinon, `proftpd` utilise la configuration qui porte la directive **DefaultServer**. En l'absence de directive `DefaultServer` et si aucune configuration correspondante ne peut être trouvée, le client reçoit un message "Aucun serveur disponible pour traiter votre demande". `DefaultServer` peut indiquer qu'un `<VirtualHost>` est le contexte par défaut, au lieu du contexte "configuration serveur" habituel.



Il y a une dernière directive de configuration qu'un administrateur devrait connaître : `SocketBindTight`. Par défaut, le démon `proftpd` écoutera toutes les adresses sur le port 21, pour les demandes de connexion des clients distants. Parfois, l'administrateur peut souhaiter que le démon `proftpd` écoute uniquement les adresses IP pour lesquelles il a été configuré, et non toutes les adresses. Pour ce faire, utilisez simplement la directive de configuration `SocketBindTight` :

#### SocketBindTight on

Ceci configure le démon pour qu'il "lie étroitement" aux adresses IP pour lesquelles il a été configuré pour écouter, plutôt qu'à toutes les adresses. Par défaut, le démon `proftpd` écoute toutes les adresses de la machine hôte.

## Utilisation

### Commandes

Avec ces aliases, voici les commandes :

- **Démarrer** le serveur :

```
$ sudo lampp start
Starting XAMPP for Linux 7.3.10-0...
XAMPP: Starting Apache...ok.
XAMPP: Starting MySQL...ok.
XAMPP: Starting ProFTPD...ok.
```

- **Redémarrer** le serveur :

```
$ sudo lampp restart
Restarting XAMPP for Linux 7.3.10-0...
XAMPP: Stopping Apache...ok.
XAMPP: Stopping MySQL...ok.
XAMPP: Stopping ProFTPD...ok.
XAMPP: Starting Apache...ok.
XAMPP: Starting MySQL...ok.
XAMPP: Starting ProFTPD...ok.
```

- **Sécuriser** le serveur :

```
$ sudo lampp security
```



- **Arrêter** le serveur :

```
$ sudo lampp stop
Stopping XAMPP for Linux 7.3.10-0...
XAMPP: Stopping Apache...ok.
XAMPP: Stopping MySQL...ok.
XAMPP: Stopping ProFTPD...ok.
```

- **Liste des options** possibles :

```
$ sudo lampp
```

- Lancer le **panneau de contrôle** :

```
$ sudo ctlampp
```

Bien sur, vous pourrez toujours utiliser ces commandes avec `/opt/lampp/lampp`.

## Test

Pour tester en ligne de commande :

```
$ ftp localhost
Connected to localhost.localdomain.
220 ProFTPD 1.3.4c Server (ProFTPD)
[::ffff:127.0.0.1]
Name (localhost:xxxxxxx):
331 Password required for xxxxxxx
Password:
230 User xxxxxxx logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
200 PORT command successful
150 Opening ASCII mode data connection for
file list
-rw-r--r--  1 root    root      3607 Feb
27 2017 applications.html
-rw-r--r--  1 root    root      177 Feb
27 2017 bitnami.css
drwxr-xr-x 21 root    root      4096 Mar
21 11:19 dashboard
-rw-r--r--  1 root    root    30894 May
11 2007 favicon.ico
drwxr-xr-x  2 root    root      4096 Mar
21 11:19 img
-rw-r--r--  1 root    root      260 Jul
9 2015 index.php
drwxr-xr-x  2 daemon  daemon    4096 Mar
21 11:19 webalizer
226 Transfer complete
ftp> bye
221 Goodbye.
$
```



## Désinstallation

Lancez en ligne de commande :

```
$ sudo rm -rf /opt/lampp
```

## Voir aussi

- **(fr)** <https://www.apachefriends.org/fr/index.html>

---

Basé sur « [Article](#) » par Apachefriends.

1)

au lieu de /etc/proftpd/ pour proftpd installé isolément

2)

ou hôte virtuel, en abrégé vhost



From:

<https://www.nfrappe.fr/doc/> - **Documentation du Dr Nicolas Frappé**

Permanent link:

<https://www.nfrappe.fr/doc/doku.php?id=logiciel:internet:xampp:linux:start>

Last update: **2022/11/08 19:28**