

[tutoriel](#)

LLMP : un serveur HTTP sous Linux, Lighty, PHP et MySQL

Pré-requis

Première étape

Autres étapes

Configuration

La définition d'un hôte virtuel permet de redéfinir les valeurs par défaut du fichier principal.

public_html

Lors de la création d'un compte utilisateur, les droits par défaut interdisent à lighttpd de traverser le dossier. Pour utiliser les sites personnels (dossier public_html), il faut donc donner aux "autres" (et donc à l'utilisateur lighttpd) le droit de traverser le répertoire d'accueil. Chaque utilisateur fera :

```
chmod o+x $HOME
```

L'ajout d'hôte supplémentaire est très simple et permet de varier les options de configuration.

Ajout du nom du site dans le DNS ou, pour test, dans le fichier **/etc/hosts**

```
127.0.0.1      nommachine site1 site2 site3  localhost.localdomain
localhost
```

Exemple de définition d'un hôte dans notre fichier de configuration (messites.conf) :

[messites.conf](#)

```
server.modules += ( "mod_dirlisting" )

$http["host"] == "site1" { # hôte du site http://site1/
    server.document-root = "/var/www/lighttpd/site1"
    dir-listing.activate = "enable" # affiche la liste des
    fichiers d'un répertoire
    dir-listing.hide-dotfiles = "enable" # bloque l'accès aux
```

```
fichier cachés
    accesslog.filename = "/var/log/lighttpd/site1/access_log" #un
fichier de log dédié
}
```

Il est possible d'utiliser les expressions régulières dans les nom d'hôtes en utilisant `=~` au lieu de `==` : `$HTTP["host"] =~ "(www.|)site1"`. Remarque: `server.modules =` assigne le contenu à `server.modules`, `server.modules +=` ajoute le contenu en conservant l'existant à `server.modules`.

En l'état actuel, seules les pages statiques fonctionnent. Pour du PHP, il est nécessaire d'installer l'extension **fast-cgi** et **php-cli** ([Installation_et_configuration_de_PHP](#)).

Extensions

Il existe un grand nombre d'extensions dont la liste est ici : <http://redmine.lighttpd.net/projects/lighttpd/wiki/Docs> (en).

Déplacement de la racine du site (Apache)

Apache est démarré à son installation. On peut le vérifier depuis un pc du réseau en ouvrant <http://framboise.home> dans un navigateur. Il s'affiche :

It works! → le serveur fonctionne.

Sauvegarder l'original du fichier **/etc/apache2/sites-available/default** :

- `sudo cp /etc/apache2/sites-available/default /etc/apache2/sites-available/default.original`

Ouvrir avec les droits d'administration le fichier **/etc/apache2/sites-available/default** et y remplacer les occurrences de **/var/www** par **/le/chemin/voulu** (par exemple **/media/Reservoirs/www** :

remplacer	par
DocumentRoot /var/www	DocumentRoot /media/Reservoirs/www
<Directory /var/www/>	<Directory /media/Reservoirs/www/>

Redémarrer le serveur apache par :

- `sudo service apache2 restart`

Désormais, en tapant l'adresse <http://framboise>, on aboutit dans le répertoire **/media/Reservoirs/www**.

Création de sous-domaines automatisés

Il faut d'abord installer le module `vhost_alias` :

- `a2enmod vhost_alias`
`/etc/init.d/apache2 reload`

Tout se passe dans le répertoire **`/etc/apache2/sites-available/`**. Ce répertoire contient un fichier de configuration par serveur. Le serveur par défaut est défini par le fichier **`default`**.

Nous prendrons comme exemple le sous-domaine **`doc.framboise.home`** qui correspond à un dokuwiki de racine **`/media/Reservoirs/www/doc`**.

Les sous-domaines sont déduits de l'appel :

- sous-répertoires de domaine :
 - **`xxx`**.domaine.tld → répertoire `www/domaine/xxx`
 - **`yyy`**.domaine.tld → répertoire `www/domaine/yyy`
- mais on peut aussi faire des répertoires plus élaborés :
 - **`aaa`**.domaine.tld → répertoire `www/domaine/aaa/un_répertoire`
 - **`bbb`**.domaine.tld → répertoire `www/domaine/bbb/un_répertoire`

Créer une copie de **`/etc/apache2/sites-available/default`** → **`/etc/apache2/sites-available/framboise.home`**

Ajouter une directive `VirtualDocumentRoot` et `ServerAlias` dans la configuration de votre `VirtualHost` comme dans l'ex suivant :

[framboise.home](#)

```
# obtenir le nom du serveur à partir de l'entête "Host:"
UseCanonicalName Off

<VirtualHost *:80>
    DocumentRoot /media/Reservoirs/www
    ServerName framboise.home
    ServerAlias *.framboise.home
    VirtualDocumentRoot /media/Reservoirs/%-3
    # format de journal avec l'appel
    LogFormat "%V %h %l %u %t \"%r\" %s %b" vcommon
    CustomLog access.log vcommon
</VirtualHost>

# réglages pour les répertoires du dokuwiki
<Directory /media/Reservoirs/www/doc>
    Order deny,allow
    allow from all
</Directory>
```

```
<LocationMatch "/doc/(data|conf|bin|inc)/">
    Order allow,deny
    Deny from all
    Satisfy All
</LocationMatch>
```

- variable %0 = le sous-domaine déduit de l'adresse appelée. (xxx.domaine.com)
- variable %1 = premier élément de l'adresse appelée. (xxx)
- variable %2 = 2e élément de l'adresse appelée. (domaine)
- variable %-1 = dernier élément de l'adresse appelée. (com)
- variable %-2 = avant-dernier élément de l'adresse appelée. (domaine)
- variable %-3 = avant-avant-dernier élément de l'adresse appelée. (xxx, yyy, ...)

Ici, %-3 permet d'éviter les problèmes dûs à www.

Enregistrer le fichier framboise.home et relancer apache :

- ```
sudo a2ensite framboise.home
sudo service apache2 restart
```

## Méthode simple pour un domaine

Pour créer un sous-domaine (par ex. doc.framboise), nous allons créer un fichier doc.framboise dans ce répertoire. Chaque sous-domaine doit être répertorié dans le dns ou dans le fichier hosts.

Éditer le fichier /etc/hosts et relancer Apache :

```
sh fredit.sh /etc/hosts \
&& ssh framboise "sudo /etc/init.d/apache2 restart"
```

Ajouter au fichier /etc/hosts la ligne :

```
192.168.0.100 doc.framboise
```

Enregistrer : apache se relance.

Copier le fichier /etc/apache2/sites-available/default → doc.framboise :

```
scp framboise:/etc/apache2/sites-available/default /etc/apache2/sites-
available/doc.framboise
```

, l'éditer, enregistrer le site et relancer apache :

```
sh fredit.sh /etc/apache2/sites-available/default \
&& ssh framboise "sudo a2ensite doc.framboise; \
sudo /etc/init.d/apache2 restart"
```

et y ajouter les lignes :

[doc.framboise](#)

```
NameVirtualHost doc.framboise

<VirtualHost doc.framboise>
 ServerAdmin webmaster@domaine.fr
 ServerName www.doc.framboise
 ServerAlias doc.framboise
 DocumentRoot /media/Reservoirs/www/dokuwiki

 # réglages pour le répertoire
 <Directory /media/Reservoirs/www/dokuwiki>
 Order deny,allow
 allow from all
 </Directory>
 <LocationMatch "/dokuwiki/(data|conf|bin|inc)/">
 Order allow,deny
 Deny from all
 Satisfy All
 </LocationMatch>
</VirtualHost>
```

Pour pouvoir accéder depuis deux adresses (par exemple une en local et une sur internet) :

<note important>Les adresses doc.framboise et 1.2.3.4 doivent exister dans le DNS ou le hosts</note>

[doc.framboise](#)

```
NameVirtualHost doc.framboise
NameVirtualHost 1.2.3.4

<VirtualHost 1.2.3.4 doc.framboise>
 ServerAdmin webmaster@domaine.fr
 ServerName www.doc.framboise
 ServerAlias doc.framboise
 DocumentRoot /media/Reservoirs/www/dokuwiki

 # réglages pour le répertoire
 <Directory /media/Reservoirs/www/dokuwiki>
 Order deny,allow
 allow from all
 </Directory>
 <LocationMatch "/dokuwiki/(data|conf|bin|inc)/">
 Order allow,deny
 Deny from all
 Satisfy All
 </LocationMatch>
```

```
</VirtualHost>
```

## Autres exemples de sous-domaines

fichier de configuration apache /etc/apache2/sites-available/domaine.com  
(<http://forum.ubuntu-fr.org/viewtopic.php?id=361680>):

domaine.com :

```
NameVirtualHost *:80

<VirtualHost *:80>

 ServerAdmin admin@domaine.com
 ServerName www.domaine.com
 ServerAlias domaine.com

 UseCanonicalName Off

 DocumentRoot /websites/www.domaine.com
 <Directory /websites/www.domaine.com>
 Options Indexes FollowSymlinks
 AllowOverride none
 Order deny,allow
 Allow from all
 </Directory>

 ErrorLog /var/log/apache2/www_error.log
 LogLevel warn
 CustomLog /var/log/apache2/www_access.log combined
 ServerSignature On

</VirtualHost>
```

UseCanonicalName Off → le nom du serveur est déduit du contenu de l'entête Host: de la requête0.

sql.domaine.com :

```
<VirtualHost *:80>
 ServerName sql.domaine.com
 ServerAdmin "admin@domaine.com"

 DocumentRoot /websites/sql.domaine.com
 <Directory /websites/sql.domaine.com>
 Options Indexes FollowSymlinks
 AllowOverride none
 Order deny,allow
 Allow from all
 </Directory>
```

```
ErrorLog /var/log/apache2/sql_error.log
LogLevel warn
CustomLog /var/log/apache2/sql_access.log combined
</VirtualHost>
```

dev.domaine.com :

```
<VirtualHost *:80>
 ServerName dev.domaine.com
 ServerAdmin "admin@domaine.com"

 DocumentRoot /websites/dev.domaine.com
 <Directory /websites/dev.domaine.com>
 Options Indexes FollowSymlinks
 AllowOverride none
 Order deny,allow
 Allow from all
 </Directory>

 ErrorLog /var/log/apache2/dev_error.log
 LogLevel warn
 CustomLog /var/log/apache2/dev_access.log combined
</VirtualHost>
```

phpmyadmin.domaine.com :

```
<VirtualHost *:80>
 ServerName phpmyadmin.domaine.com
 ServerAdmin "admin@domaine.com"

 DocumentRoot /websites/phpmyadmin.domaine.com
 <Directory /websites/phpmyadmin.domaine.com>
 Options Indexes FollowSymlinks
 AllowOverride none
 Order deny,allow
 Allow from all
 </Directory>
 ErrorLog /var/log/apache2/phpmyadmin_error.log
 LogLevel warn
 CustomLog /var/log/apache2/phpmyadmin_access.log combined
</VirtualHost>
```

blog.domaine.com :

```
<VirtualHost *:80>
 ServerName blog.domaine.com
 ServerAdmin "admin@domaine.com"

 DocumentRoot /websites/blog.domaine.com
 <Directory /websites/blog.domaine.com>
```

```
Options Indexes FollowSymlinks
AllowOverride none
Order deny,allow
Allow from all
</Directory>
ErrorLog /var/log/apache2/blog_error.log
LogLevel warn
CustomLog /var/log/apache2/blog_access.log combined
</VirtualHost>
```

## Rendre le serveur disponible sur Internet

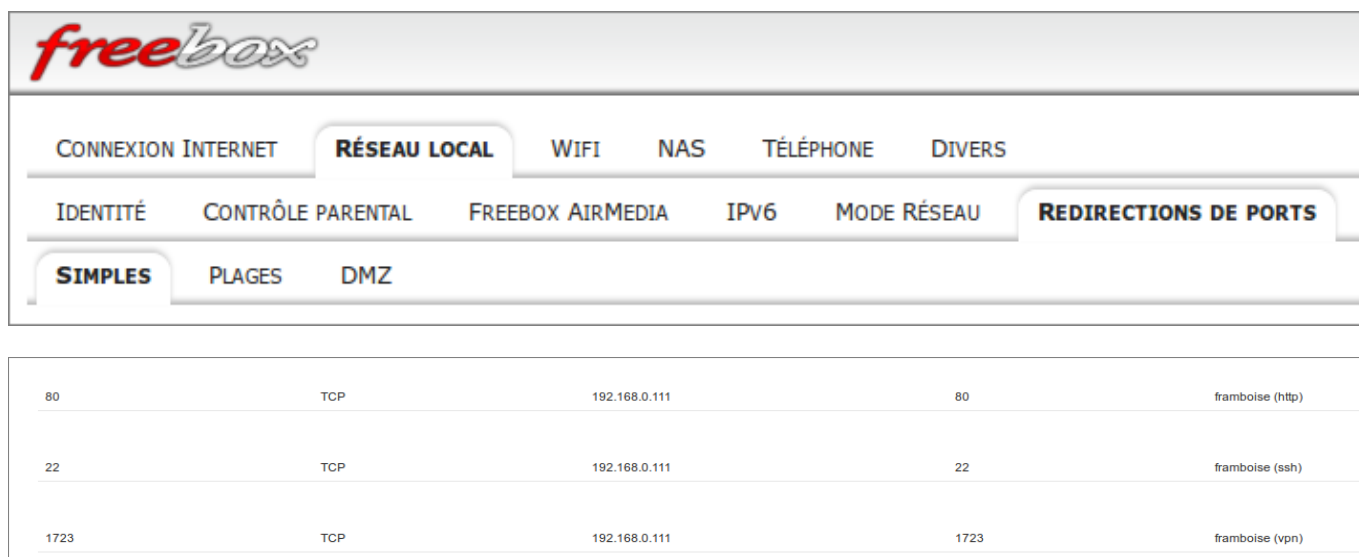
Nous allons configurer le routeur pour autoriser les connexions ssh et des trafics Web grâce à son pare-feu au Raspberry Pi.

Par sécurité, le mot de passe de l'utilisateur pi doit avoir été changé et éventuellement un nouvel utilisateur mis en place.

Il faut rediriger les adresses parvenant sur le routeur (la freebox) vers le Raspberry Pi.

Les ports à rediriger en TCP sont :

- le port 80 (pour http)
- le port 22 (pour ssh)
- le port 1723 (pour vpn)



80	TCP	192.168.0.111	80	framboise (http)
22	TCP	192.168.0.111	22	framboise (ssh)
1723	TCP	192.168.0.111	1723	framboise (vpn)

## Optimisation du Raspberry Pi pour Lighttpd Owncloud, WordPress et plus

source :

<http://c-mobberley.com/wordpress/index.php/2013/05/18/raspberry-pi-lighttpd-optimisation-for-wordpress-owncloud-and-more/>

Lighttpd, sont installés. Nous allons optimiser **lighttpd** pour le Raspberry Pi en installant PHP APC, en



activant fast cgi et en mettant en place la compression avec gzip.

Tout d'abord, [Ouvrir avec les droits d'administration](#) le fichier **/etc/lighttpd/lighttpd.conf** et remplir la section server.modules avec les modules suivants :

```
server.modules = (
 #"mod_rewrite",
 "mod_redirect",
 "mod_alias",
 "mod_access",
 #"mod_auth",
 #"mod_status",
 #"mod_simple_vhost",
 #"mod_evhost",
 #"mod_userdir",
 #"mod_secdownload",
 #"mod_fastcgi",
 #"mod_proxy",
 #"mod_cgi",
 #"mod_ssi",
 #"mod_compress",
 #"mod_usertrack",
 #"mod_expire",
 #"mod_rrdtool",
 #"mod_accesslog"
)
```

## PostgreSQL

Voir la page [PostgreSQL sur un Raspberry Pi : une base de données SQL](#)

## Conclusion

## Problèmes connus

## Voir aussi

- (fr) [http://Article](#)

---

Basé sur « [Article](#) » par Auteur.

From:

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

Permanent link:

<https://doc.nfrappe.fr/doku.php?id=tutoriel:nanopc:raspi:llmp:start>



Last update: **2022/11/08 19:41**