Prendre en main Apache2

- Objet : Mise en place d'un serveur web
- Niveau requis : Fix Me!
  débutant
- Commentaires : On verra l'installation et la configuration rapide d'un serveur LAMP, et on ajoutera quels trucs et astuces.
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !.

Introduction

Un serveur LAMP

est un serveur Web. Historiquement, l'acronyme « LAMP » fait référence aux quatre composantes d'un tel serveur Web :

- Linux : le système d'exploitation constituant la base du système.
- Apache : le serveur HTTP qui gère la communication avec le client.
- MySQL : le système de gestion de base de données.
- PHP : le langage de script utilisé pour générer les pages dynamiques.

Pré-requis

- Installation de Jessie : https://debian-facile.org/doc:install:install
- Mise à jour :

```
apt-get update && apt-get dist-upgrade
```

- Installer les librairies de base :

```
apt-get install build-essential
```

```
apt-get install apache2
```

Dans ce wiki : la prise en main d'apache2 qui est le serveur web !

Créer son site web rapidement

Création dossier pour les fichiers de code du site

```
cd /var/www/
```

```
mkdir hypathie.net
```

```
chown -R hypathie:www-data hypathie.net/
```

```
chmod -R 750 hypathie.net/
```

```
vim hypathie.net/index.html
```

```
<!DOCTYPE html>
<html>
 <head>
  <meta charset="UTF-8">
  <title>hypathie.net _ index.html</title>
 </head>

 <body>
    <h1>Bienvenue sur le site hypathie ! </h1>
 </body>
</html>
```

A savoir !

Pour régler définitivement les problèmes d'encodage, au cas où l'on oublie la balise `<meta charset="UTF-8">` du head, on édite le fichier `/etc/apache2/conf-availables/charset.conf`

```
vim /etc/apache2/conf-available/charset.conf
```

Et on dé-commente la ligne :

```
AddDefaultCharset UTF-8
```

```
systemctl restart apache2
```

Création d'un virtualhost en local pour le site dans /etc/apache2/sites-available/

```
cd /etc/apache2/sites-available
```

```
vim hypathie.net.conf
```

```
<VirtualHost *:80>
        ServerName hypathie.net

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/hypathie.net/

        <Directory /var/www/hypathie.net>
```

```
          require all granted
          AllowOverride All
        </Directory>

        LogLevel info

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

Explications

- **VirtualHost *:80**
  L'étoile signifie toutes les adresses écoutées par apache et auxquelles il répond. Ce sont les adresses des clients qui consultent les sites servis par apache, sur **le port 80**.
- **ServerName and ServerAlias** :
  Let Apache know the domain to match to this virtual host by setting the ServerName. Optionally also use ServerAlias to tell apache to listen to other domains and point them to this virtual host as well, such as a “www” subdomain.
- **DocumentRoot**:
  Change to suit your needs. I often have a “public_html” or “public” directory which is the web root. Then I can encapsulate related files which stay behind the web-root within the sites directory. (site.com directory, with site.com/public_html directory as the web-root). This is how Laravel works by default.

Options -Indexes:: -Indexes stops people from being able to go to a directory and see files listed in there. Instead they see a Forbidden error. This can stops users view all your files in your /images directory, for instance.

- **AllowOverride**:
  Set to “all” to allow .htaccess files in your virtual host (And sub-directories)

ErrorLog, CustomLog: Create log files specifically for your domain, so they don't get mixed in with traffic / errors from other sites running on the server.

Activer le site (cela crée un lien symbolique dans /etc/apache2/sites-enabled/)

```
a2ensite hypathie.net.conf
```

```
systemctl restart apache2
```

Ajouter son domaine dans /etc/hosts

Le domaine n'a pas à être loué forcément, mais le nom du site pour Apache2.

```
vim /etc/hosts
```

```
127.0.0.1       localhost
127.0.1.1       db
127.0.1.1       hypathie.net


# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

- `db` : c'est le hostname de la machine sur laquelle tourne apache2 et pour laquelle on vient de modifier le fichier `/etc/hosts`.

Vérifications

- En local : Dans le navigateur : **`localhost`**

Si l'on tombe sur sur la page d'accueil du site de test d'apache2
dont le code source est placé dans `/var/www/html/index.html`

On désactive ce site :

```
cd /etc/apache2/sites-enabled/ && a2dissite 000-default.conf &&
systemctl restart apache2
```

Création de plusieurs virtualhosts accessibles depuis une autre machine du réseau local

IP fixe

Au préalable, il faut aller configurer son routeur (box_machin ADSL) pour qu'il attribut une IP fixe à la machine sur laquel tourne apache.
Si l'on veut aussi accéder au site depuis l'extérieur du réseau local, il faut aussi configurer son routeur

pour qu'il fasse une redirection de port vers l'IP fixe en question.

Plusieurs fichiers pour chaque site

- On a créé plusieurs dossiers pour ses sites contenant chacun un index.html différent.

(voir plus haut si besoin)

```
ls /var/www/
```

```
essai  html  hypathie.net  test
```

Créations de plusieurs virtualhosts

Comme plus haut :

```
vim essai.conf
```

```
<VirtualHost *:80>
        ServerName essai.local
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/essai/public

        <Directory /var/www/essai/public/>
          require all granted
          AllowOverride All
        </Directory>

        LogLevel info

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

```
vim test.conf
```

```
<VirtualHost *:80>
        ServerName test.local
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/test/public

        <Directory /var/www/test/public/>
          require all granted
          AllowOverride All
        </Directory>

        LogLevel info
```

```
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

- On active ces deux sites :

```
a2ensite essai.conf
```

```
a2ensite test.conf
```

- On redémarre apache :

```
systemctl restart apache2
```

Configurer le fichier /etc/hosts

On configure le fichier hosts des ordinateurs distants du réseau local susceptibles de se connecter aux sites `test.local` et `essai.local` :

L'ip 192.168.0.17 est celle du réseau local sur lequel tourne apache2. Elle a été fixé au niveau du routeur.

```
vim /etc/hosts
```

```
127.0.0.1       localhost
127.0.1.1       debian
192.168.0.17    db
192.168.0.17    test.local
192.168.0.17    essai.local


# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

- `debian` : c'est le hostname de la machine cliente d'ip 192.168.0.14 qui va visiter les sites de la machine d'ip 192.168.0.17 sur laquelle tourne apache2 ;
- `db` : c'est la hostname de la machine qui fait office de serveur qui a l'ip 192.168.0.17 et sur laquelle tourne apache2.
- `test.local` : c'est le `servername`, le nom du serveur virtuel crée par apache2 et qui va permettre au client (distant) de demander à apache2 de visualiser ce site
- `essai.local` : idem

Le fichier hosts sur windows

On édite le fichier hosts qui se trouve : `C:\Windows\System32\drivers\etc`

On y ajoute l'ip du serveur debian qui se trouve sur le sous-réseau :

```
# localhost name resolution is handled within DNS itself.
    127.0.0.1       localhost
    ::1             localhost

    192.168.0.17    db
    192.168.0.17    test.local
    192.168.0.17    essai.local
```

On teste

- Dans le navigateur depuis une autre machine du réseau local, par exemple une d'ip 192.168.0.14 : **`essai.local`**
- Puis idem pour : **`test.local`**
- Pour tester depuis l'extérieur de son réseau local :**`http://ip_du_routeur_de_son_fai/essai.local`**

Installer son site dans son répertoire personnel

Attention de modifier `user` par l'utilisateur qui est propriétaire du dit répertoire !

On crée un dossier Web

```
mkdir /home/user/www/site1
```

On donne propriétés et droits

```
chown -R user:www-data /home/user/www/
```

```
chmod -R 755 /home/user/www/
```

Si on veut que www-data puisse écrire (log ou autre) il faut ajouter les droits d'écriture pour le groupe :
`chmod -R 775 /home/user/www/`

- 700 : 4 (lecture) + 2 (écriture) + 1 (exécution) pour l'utilisateur
- 070 : idem pour le groupe

- 005 : 2 (écriture) + 1 (exécution) pour les autres

Avec -R pour récursif dans les répertoires, il faut le droit d'éxécution pour traverser un répertoire.

On crée un dossier pour le site1 et un index.html pour tester

```
cd /home/user/www/ && mkdir site1/
```

```
vim site1/index.html/
```

On crée un lien symbolique vers le répertoire utilisateur dans www

```
ln -s /home/user/www/site1/ /var/www/site1
```

On lui crée un virtualhost

On donne le chemin du dossier de son répertoire personnel dans le virtualhost.

```
vim /etc/apache2/sites-available/site1.conf
```

```
<VirtualHost *:80>
        ServerName site1.local
        ServerAdmin webmaster@localhost
        DocumentRoot /home/user/www/site1/

        <Directory /home/user/www/site1/
          require all granted
          AllowOverride All
        </Directory>

        LogLevel info

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

```
a2ensite site1.conf
```

```
systemctl reload apache2
```

On teste :

* Par exemple, dans le navigateur : **`site1.local`**

Le module php7

On installe les paquets nécessaires

```
apt install php7.3-cli php7.3-fpm php7.3-json php7.3-pdo php7.3-mysql
php7.3-zip php7.3-gd  php7.3-mbstring php7.3-curl php7.3-xml php7.3-bcmath
php7.3-json
```

```
apt -y install libapache2-mod-php
```

On active le mode php

```
a2enmod php7.3
```

```
systemctl restart apache2
```

Fichiers intéressants

php.ini

```
/etc/php/7.[version]/apache2/php.ini
```

```
file_uploads = On
allow_url_fopen = On
short_open_tag = On
memory_limit = 256M
upload_max_filesize = 100M
max_execution_time = 360
```

La directive active directory

```
more /etc/apache2/mods-available/dir.conf
```

```
DirectoryIndex DirectoryIndex index.html index.cgi index.pl index.php
index.xhtml index.htm
```

Le fichier php7.3.conf

```
vim php7.3.conf
```

```
more /etc/apache2/mods-enabled/php7.3.conf
```

```
<FilesMatch ".+\.ph(ar|p|tml)$">
    SetHandler application/x-httpd-php
</FilesMatch>
```

Les premières lignes que l'application x-httpd-php interprète les fichier qui se termine soit par `.phar`, soit par `.php` soit par `.phtml`.

Pour résumer

Fichiers de log d'apache

* Erreurs

```
more /var/log/apache2/error.log
```

Par exemple erreur suivante indique un problème de droit sur le fichier index.php

```
[Mon Dec 07 19:18:33.354048 2020] [php7:error] [pid 5220] [client ::1:38114]
PHP Fatal error:  Unknown: Failed opening required
'/home/user/www/mon_site.com/index.php' (include_path='.:/usr/share/php') in
Unknown on line 0
```

* Accès

```
more /var/log/apache2/access.log
```

Propriétés et droits

- **On donne les propriétés** : utilisateur du site et au groupe www-data :
  `chown -R utilisateur:www-data /chemin/du/site/`
- **On donne récursivement les droits 2750 sur le dossier du site** :
  `chmod -R 750 /home/utilisateur/site/`
  - SGID sur un fichier ou un dossier, Valeur octale : 2000, valeur symbolique : s
  - les droits de lecture + écriture + exécution pour l'utilisateur (7)
  - les droits de lecture + exécution pour le groupe www-data (5)
  - aucun droit pour les autres (0)

Pour les *paranoïdes* de la sécurité.

```
cd /home/nathalie/www/ && find . -type d -exec chmod 2775 {} \;
```

```
cd /home/nathalie/www/ && find . -type f -exec chmod 0664 {} \;
```

Fichiers de configuration d'apache

- Les fichiers de code source du site web sont placés sur le système dans : `/var/www/`
- Il est conseillé d'approprier le dossiers www et à ses sous dossiers à l'utilisateur courant du système et au groupe www-data, ainsi que les droits 755 avec chmod.
- Les fichiers de configuration d'un virtualhost qui permettent de servir le code source d'un site web se trouve dans : `/etc/apache2/site-available`. Lorsqu'on crée un fichier on lui donne l'extension `.conf` sur Jessie.
- Dans le dossiers /etc/apache2/site-enabled/ on trouve des liens symboliques dont la présence prouve que le site correspondant a été mis en service avec la commande `a2ensite le-site`. On peut aussi crée manuellement un lien symbolique avec la commande linux ln
- Le fichier de configuration d'apache qui gère le mapping des noms et de leurs extensions possibles pour le premier fichier présentant le site web (en général `index.html` ou `index.php`) se modifie à la dernière ligne :

```
vim /etc/apache2/apache2.conf
```

```
DirectoryIndex index.html default.html index.php default.php
```

- Le fichier de démarrage du site web (par exemple `index.php`) doit se trouver dans le dossier déclaré dans la configuration du virtualhost dans la balise `Directory`, par exemple :

```
[...]
        <Directory /var/www/essai/public/>
[...]
```

- Lister les modules d'apache :

```
ls /etc/apache2/mods-available/
```

- Lister les modules d'apache activés :

```
ls /etc/apache2/mods-enabled/
```

Commandes de gestion d'apache

- Relancer le service d'apache après une modification d'un ou plusieurs fichiers de configuration

```
systemctl reload apache2
```

Info sur systemctl :

https://www.linuxtricks.fr/wiki/systemd-les-commandes-essentielles

- Mettre en service un site déjà configuré :

```
a2ensite le_site
```

- Désactiver un site :

```
a2dissite le_site
```

- Activer un module :

```
a2enmod nomModule
```

- Désactiver un module :

```
a2dismod nomModule
```

Pour un projet symfony5

Apache2 et php7 sont installés.

On crée un virtualhost de même nom que le projet symfony

```
cd /etc/apache2/sites-available && vim TestSymfony
```

```
<VirtualHost *:80>
   ServerName TestSymfony

   DocumentRoot /var/www/TestSymfony/public/
   <Directory /var/www/TestSymfony/public/>
        Options FollowSymLinks MultiViews
        AllowOverride All
        Order allow,deny
        Allow from All
   </Directory>

   ErrorLog /var/log/apache2/project_error.log
   CustomLog /var/log/apache2/projec_access.log combined
</VirtualHost>
```

On donne les bons droits au dossier du projet

Symfony a besoin du droit d’exécution de certains fichiers.

```
chown -R www-data:www-data /var/www/TestSymfony/
```

```
chmod -R 755 /var/www/TestSymfony/
```

On crée un lien symbolique vers "site enable"

```
a2ensite TestSymfony && systemctl restart apache2.service
```

Discuter avec Apache

```
systemctl [stop|start|enable] apache2.service
```

On installe symfony

```
mkdir InstallSymfonyComposer
cd InstallSymfonyComposer/
wget https://getcomposer.org/installer

php installer

cp composer.phar composer.phar_copy

sudo mv composer.phar /usr/local/bin/composer

sudo chmod +x /usr/local/bin/composer

cd /var/www/
```

```
composer create-project symfony/skeleton TestSymfony
```

Le dossier web /var/www appartient à root :

```
chmod -R 777 /var/www/
```

Pour créer une application plus légère comme un microservice, une app console ou une API dans une certaine version du framework :

```
composer create-project symfony/skeleton TestSymfony [^version_voulue]
```

ou

```
symfony new my_project [--version=lts|stable|4.4]
```

Pour créer une application web traditionnelle :

```
composer create-project symfony/website-skeleton TestSymfony
```

ou

```
symfony new --full my_project [--version=lts|stable|4.4]
```

On n'oublie de remettre les droits d'origines à /var/www

```
chmod -R 755 /var/www/
```

Avec un virualhost qui pointe dans un répertoire personnel pour symfony :

```
chmod -R 2775 /var/www/
```

Pour un projet symfony5

1. À la racine du projet symfony on ajoute un fichier dans public/.htaccess

```
# Use the front controller as index file. It serves as a fallback solution
when
# every other rewrite/redirect fails (e.g. in an aliased environment without
# mod_rewrite). Additionally, this reduces the matching process for the
# start page (path "/") because otherwise Apache will apply the rewriting
rules
# to each configured DirectoryIndex file (e.g. index.php, index.html,
index.pl).
DirectoryIndex index.php

# By default, Apache does not evaluate symbolic links if you did not enable
this
# feature in your server configuration. Uncomment the following line if you
# install assets as symlinks or if you experience problems related to
symlinks
# when compiling LESS/Sass/CoffeScript assets.
# Options +FollowSymlinks

# Disabling MultiViews prevents unwanted negotiation, e.g. "/index" should
not resolve
# to the front controller "/index.php" but be rewritten to
"/index.php/index".
<IfModule mod_negotiation.c>
    Options -MultiViews
</IfModule>

<IfModule mod_rewrite.c>
    # This Option needs to be enabled for RewriteRule, otherwise it will
show an error like
    # 'Options FollowSymLinks or SymLinksIfOwnerMatch is off which implies
that RewriteRule directive is forbidden'
    Options +FollowSymlinks

    RewriteEngine On
```

```
    # Determine the RewriteBase automatically and set it as environment
variable.
    # If you are using Apache aliases to do mass virtual hosting or
installed the
    # project in a subdirectory, the base path will be prepended to allow
proper
    # resolution of the index.php file and to redirect to the correct URI.
It will
    # work in environments without path prefix as well, providing a safe,
one-size
    # fits all solution. But as you do not need it in this case, you can
comment
    # the following 2 lines to eliminate the overhead.
    RewriteCond %{REQUEST_URI}::$0 ^(/.+)/(.*)::\2$
    RewriteRule .* - [E=BASE:%1]

    # Sets the HTTP_AUTHORIZATION header removed by Apache
    RewriteCond %{HTTP:Authorization} .+
    RewriteRule ^ - [E=HTTP_AUTHORIZATION:%0]

    # Redirect to URI without front controller to prevent duplicate content
    # (with and without `/index.php`). Only do this redirect on the initial
    # rewrite by Apache and not on subsequent cycles. Otherwise we would get
an
    # endless redirect loop (request -> rewrite to front controller ->
    # redirect -> request -> ...).
    # So in case you get a "too many redirects" error or you always get
redirected
    # to the start page because your Apache does not expose the
REDIRECT_STATUS
    # environment variable, you have 2 choices:
    # - disable this feature by commenting the following 2 lines or
    # - use Apache >= 2.3.9 and replace all L flags by END flags and remove
the
    #   following RewriteCond (best solution)
    RewriteCond %{ENV:REDIRECT_STATUS} =""
    RewriteRule ^index\.php(?:/(.*)|$) %{ENV:BASE}/$1 [R=301,L]

    # If the requested filename exists, simply serve it.
    # We only want to let Apache serve files and not directories.
    # Rewrite all other queries to the front controller.
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteRule ^ %{ENV:BASE}/index.php [L]
</IfModule>

<IfModule !mod_rewrite.c>
    <IfModule mod_alias.c>
        # When mod_rewrite is not available, we instruct a temporary
redirect of
        # the start page to the front controller explicitly so that the
```

```
website
        # and the generated links can still be used.
        RedirectMatch 307 ^/$ /index.php/
        # RedirectTemp cannot be used instead
    </IfModule>
</IfModule>
```

2. On active le mode rewrite d'apache

```
a2enmod rewrite && systemctl restart apache2
```

From:
http://debian-facile.org/ - **Documentation - Wiki**

Permanent link:
http://debian-facile.org/utilisateurs:hypathie:tutos:creer-son-site-web-en-php-_-prendre-en-main-apache2-sur-jessie

Last update: **17/09/2023 10:36**