

Apache

Master 2 I2L

Année 2009/2010

D'après la version de Jean-Christophe Soulié (2007-2008)

D. Duvivier
LIL – Université du Littoral Côte d'Opale
duvivier@lil.univ-littoral.fr

1 Préparer son serveur

```
# aptitude update
# aptitude upgrade
# aptitude clean
```

Attention : Ne rendez visible que le strict nécessaire sur votre machine, protégez en mode 700 /root, \$home et /usr/src/...

2 Installer apache2

```
# aptitude install apache2 apache2-doc apache2-utils apachetop
```

Il peut être intéressant de charger le paquet contenant la documentation d'apache (paquet apache2-doc) ☺ !
→ Installée dans /usr/share/doc/apache2-doc/, notamment /usr/share/doc/apache2-doc/manual/fr/index.html

3 Installer php5

Vérifiez si le paquet est installé en utilisant l'une des commandes suivantes :

```
# aptitude search ^php5$
# aptitude show php5
# dpkg -l php5
```

S'il n'est pas installé, installez-le :

```
# aptitude install php5
```

Il peut être intéressant également d'installer la documentation de php4&5 (paquet php-doc) ☺ !

4 Installer la base de données MySql et PhpMyAdmin

PHP est très très souvent couplé à un Système de Gestion de Base de Données : MySql. Nous installons ici MySql-server → **ATTENTION : ne pas le réinstaller si déjà installé !**

```
# aptitude install mysql-server
```

Définir le mot de passe root de MySql :

- mysqladmin password <monbopasswordvisiblepartoutlemonde>
→ Permet de figer le mot de passe
- mysqladmin password <monbonouveaupasswordvisiblepartoutlemonde>
→ Permet de changer le mot de passe

```
# mysqladmin password toto
```

On vérifie que MySql fonctionne bien :

```
# mysql -p
entrer le mot de passe
mysql> status
mysql> exit
```

Installer les librairies php5-mysql :

```
# aptitude install php5-mysql
```

Installer PhpMyAdmin :

```
# aptitude install phpmyadmin
Choix du serveur web à reconfigurer automatiquement : [*] apache2
Faut-il configurer la BD de phpmyadmin avec dbconfig-common ? <Oui>
Si un mot de passe est demandé, entrez : toto
On redémarre apache si/quand proposé
```

Dans un navigateur, testez l'URL suivante : <http://127.0.0.1>. Le serveur apache doit répondre "It works!".

On se connecte par l'adresse IP (remplacez xxx.xxx.xxx.xxx par votre adresse IP ci-après) : <http://xxx.xxx.xxx.xxx> pour vérifier que apache fonctionne correctement et on utilise phpmyadmin : <http://xxx.xxx.xxx.xxx/phpmyadmin>.

L'utilisateur est « root » et le mot de passe « toto » en salle de TP. **ATTENTION** : sur une machine en production, surtout évitez ce type de connexion (i.e. « root » avec le mot de passe initialisé à « toto »). Naviguez parmi les onglets (Bases de données, SQL, État, Variables...).

5 Organisation d'apache

Après cette installation, nous allons maintenant décrire un petit peu la structure des fichiers de configuration de apache2.

L'ensemble de ces fichiers se trouvent dans le répertoire `/etc/apache2/` qui est d'ailleurs aussi la racine de l'application.

Il est possible de visualiser l'arborescence effective (mise à jour ...) des répertoires & fichiers de `/etc/apache2` en récupérant le paquet `tree` (`aptitude install tree`) puis en tapant `tree /etc/apache2`

Ce répertoire est organisé de cette manière :

```
/etc/apache2
|_____apache2.conf
|_____conf.d
|           |_____charset
|           |_____etc .....
|_____envvars
|_____httpd.conf
|_____magic
|_____mods-available
|           |_____headers.load
|           |_____include.load
|           |_____mime_magic.conf
|           |_____mime_magic.load
|           |_____php5.conf
|           |_____php5.load
|           |_____etc .....
|_____mods-enabled
|           |_____mime.conf (lien symbolique)
|           |_____mime.load (lien symbolique)
|           |_____php5.conf (lien symbolique)
|           |_____php5.load (lien symbolique)
|           |_____etc .....
|_____ports.conf
|_____sites-available
|           |_____default
|_____sites-enabled
|           |_____000-default (lien symbolique)
```

- **apache2.conf** : fichier principal de configuration, incluant la configuration de base générale du serveur.
- **conf.d** : répertoire dans lequel on mettra les configurations supplémentaires générales au serveur.
- **envvars** : fichier pour les variables d'environnement, inutilisé dans notre cas.
- **httpd.conf** : ce fichier était à l'origine le fichier principal de configuration de apache, il n'est pas utilisé ici (en tout cas sur Debian) et n'est présent que par souci de compatibilité.
- **magic** : données de configuration du module `mime_magic`.
- **mods-available** : ce répertoire contient tous les fichiers de configuration et de chargement des modules apache installés. Lorsque vous installez un module via `apt-get` (tel que `libapache2-mod-php5`), les fichiers de `load` et de `conf` correspondants sont automatiquement créés ici.
- **mods-enabled** : voilà la subtilité de la configuration `apache2` de Debian (et peut-être d'autres distrib) ; lorsque vous voulez activer un module `apache2`, il vous suffit de créer un lien symbolique du fichier de `conf` et/ou de `load` du module désiré ici même. Le module sera alors chargé lors du prochain redémarrage du serveur apache. Pour cela, soit vous créez ce lien à la main, soit vous pouvez utiliser l'utilitaire `a2enmod` (par exemple pour `php` : `a2enmod php5`). Une fois les liens créés vous pouvez relancer `apache` pour les activer (`/etc/init.d/apache2 restart`).
- **ports.conf** : liste simplement les ports internet que le serveur doit écouter (dans mon cas les ports 80 et 443).
- **sites-available** : comme pour les modules, ce répertoire contient la configuration spécifique à chaque site web que vous voudrez installer sur votre serveur. Pour le moment seul le site « `default` » est créé; à vous de vous inspirer de ce fichier pour créer le votre ou les vôtres.
- **sites-enabled** : là aussi, comme pour les modules, ce répertoire contient les liens symboliques vers les fichiers de configuration des sites web que vous désirerez activer ou non. De la même manière, vous pouvez créer ces liens manuellement ou plus simplement à l'aide de la commande `a2ensite` `votre-fichier-de-site-web`.

6 Configuration de base d'apache

Vérifions si apache fonctionne :

```
# apache2 -v
# apache2 -V
```

Dans un navigateur, testez l'URL suivante : <http://127.0.0.1>. Le serveur apache doit répondre “It works!”.

Pour un redémarrage “en douceur” (regardez les pages de man de la commande utilisée ci-dessous), tapez :

```
# apache2ctl graceful
```

Quelques variables d'environnement sont définies dans le fichier `/etc/apache2/envvars`, notamment :

<code>APACHE_RUN_USER</code>	→ Utilisateur « <code>apache</code> » (<code>www-data</code> par défaut)
<code>APACHE_RUN_GROUP</code>	→ Groupe associé à <code>apache</code> (<code>www-data</code> par défaut)
<code>LANG</code>	→ Langue locale (« <code>C</code> » par défaut)

Quelques options utiles dans `/etc/apache2/apache2.conf` :

<code>HostnameLookups</code>	→ Utilise adresse ip pour accéder au serveur si placé sur <code>Off</code> , le nom (via DNS) si placé sur <code>On</code>
------------------------------	--

Configuration du site par défaut (pour l'instant) via le fichier `/etc/apache2/sites-available/default`, notez que ce document est organisé en sections (à l'image du fichier de configuration de proftpd) :

<code>DocumentRoot</code>	→ C'est ce qui sera vu comme la racine de votre serveur (i.e. http://127.0.0.1/)
<code><VirtualHost></code>	→ Début de section de directives pour un serveur virtuel (plusieurs sites peuvent ainsi être hébergés par un seul serveur apache)
<code><Directory></code>	→ Début de section de directives pour un répertoire

Pour vérifier si vos fichiers de configuration sont OK, utilisez :

```
apache2ctl configtest
```

7 Création d'un « VirtualHost »

Pour fonctionner correctement/complètement, le site « `m2i2l.org` » associé au « VirtualHost » créé ci-dessous nécessite que la suite des TP soit réalisée (notamment iptables et DNS). Cependant, nous pouvons déjà configurer Apache pour la suite ... Créez un répertoire `/home/m2i2l.org`, assurez vous que les droits soient bien en 755. Créez un fichier « `/home/m2i2l.org/index.html` » contenant – pour l'instant – uniquement une phrase du genre « LINUX VAINCRA ! »

Depuis `/etc/apache2/sites-available` nous allons créer un site associé à un « VirtualHost ». Vous pouvez vous baser sur le site par défaut (cf. fichier `default`) pour créer le fichier `m2i2l_org` dont le contenu est le suivant :

```
<VirtualHost 172.16.0.1:80>
    ServerAdmin webmaster@m2i2l.org
    DocumentRoot "/home/m2i2l.org"
    ServerName www.m2i2l.org
    ServerAlias www.m2i2l.org
    DocumentRoot /home/m2i2l.org
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /home/m2i2l.org/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>
    ErrorLog "/var/log/apache2/m2i2l.org-error_log"
    # Possible values include: debug, info, notice, warn, error, crit, alert, emerg.
    LogLevel warn
    CustomLog "/var/log/apache2/m2i2l.org-access_log" common
</VirtualHost>
```

Nous avons pour ce TP choisi un « VirtualHost » identifié par son adresse IP, c'est-à-dire qu'Apache va distinguer ce site par rapport à son adresse IP (172.16.0.1). Nous aurions pu également demander à Apache d'identifier le site par rapport à son nom. Selon votre configuration (présence ou non de plusieurs sous-réseaux ou adresses IP) vous pouvez sélectionner l'une de ces possibilités.

Le «site » `m2i2l_org` est disponible mais non activé (i.e. non visible) pour le rendre visible, il suffit de taper la commande suivante (qui va créer notamment un lien depuis le répertoire `/etc/apache2/site-enabled`) :

```
# a2ensite m2i2l_org
```

Dans un navigateur, testez l'URL suivante : <http://172.16.0.1>. Le serveur apache doit répondre «LINUX VAINCRA !».