

CHAPITRE 6

PHP

Maintenant que notre serveur web est installé, il serait bon de lui fournir un peu de dynamisme et rien de tel pour cela que d'installer PHP que je ne présenterai pas =).

Commençons par installer « php5 » et ce qu'il faut pour l'allier à « apache » :

```
ALBAN@bebeserv:~$ sudo -i
Password:
bebeserv:~# apt-get update
[...]
bebeserv:~# apt-get install apache2-mpm-prefork apache2-mod-php5
php5-common php5 php5-cli php5-mysql
[...]
```

Maintenant comme à l'habitude coupons « apache2 » le temps de le reconfigurer. Comme vous avez pu l'observer, nous avons changé la manière dont « apache2 » gère ses processus. Nous sommes passés de « apache2-mpm-worker » à « apache2-mpm-prefork ». Bien que « worker » soit plus rapide et offre une meilleure méthode de travail à « apache2 », « php5 » requiert que « apache2 » travaille en « prefork » encore aujourd'hui.

De même nous avons installé le minimum pour faire fonctionner notre configuration, c'est à dire « php5 » lui-même, « php5 » en ligne de commande : « cli » et « mysql » pour « php5 », bien que « php5 » offre une couche d'abstraction totale pour l'accès au SGBD beaucoup de sites sont encore construits autour de l'API mysql directement.

Retournons donc à notre configuration. Configurons « mpm-prefork » :

```
bebeserv:~# cd /etc/apache2
bebeserv:/etc/apache2# nano apache2.conf
```

Changeons notre configuration en ceci :

```
# prefork MPM
<IfModule mpm_prefork_module>
    StartServers      5
    MinSpareServers   5
    MaxSpareServers   10
    MaxClients        150
    MaxRequestsPerChild 500
```

```
</IfModule>

# worker MPM
#<IfModule mpm_worker_module>
#   StartServers      2
#   MaxClients       150
#   MinSpareThreads  25
#   MaxSpareThreads  75
#   ThreadsPerChild  25
#   MaxRequestsPerChild 500
#</IfModule>
```

Les commentaires sont facultatifs, ils me servent à me rappeler que mon apache tourne en « prefork », comme je l'avais fais dans le chapitre précédent.

Enregistrez et quittez.

Maintenant mettons à jour quelques points comme la configuration de « mod_dir ».

```
bebeserv:/etc/apache2# nano mods-avalable/dir.conf
```

Modifier la configuration comme suit :

```
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl
    index.php index.xhtml
</IfModule>
```

Enregistrez et quittez.

Maintenant configurons « php5 » :

```
bebeserv:/etc/apache2# cd /etc/php5/apache2
bebeserv:/etc/php5/apache2# nano php.ini
```

Modifier comme suit :

```
;;;;;;;;;;;;;
; Language Options ;
;;;;;;;;;;;;;

short_open_tag = Off
allow_call_time_pass_reference = Off
open_basedir = /home/nobody/default_apache
ignore_user_abort = On
realpath_cache_size=16k
realpath_cache_ttl=120
expose_php = Off
```

```

;;;;;;;;;;;;;;;;;;;;;;;;;
; Resource Limits ;
;;;;;;;;;;;;;;;;;;;;;;;;;

max_execution_time = 10
max_input_time = 10

;;;;;;;;;;;;;;;;;;;;;;;;;
; Error handling and logging ;
;;;;;;;;;;;;;;;;;;;;;;;;;

error_reporting = E_ALL | E_STRICT
display_errors = On ; A passé à Off quand vous rendez
votre site web public
log_errors = On
ignore_repeated_errors = On
error_log = /var/log/php5.log

;;;;;;;;;;;;;;;;;;;;;;;;;
; Data Handling ;
;;;;;;;;;;;;;;;;;;;;;;;;;

arg_separator.output = "&"
register_long_arrays = Off
register_argc_argv = Off
magic_quotes_gpc = Off
default_mimetype = "text/html"
default_charset = "iso-8859-15"

;;;;;;;;;;;;;;;;;;;;;;;;;
; Fopen wrappers ;
;;;;;;;;;;;;;;;;;;;;;;;;;

allow_url_fopen = Off
default_socket_timeout = 10

;;;;;;;;;;;;;;;;;;;;;;;;;
; Module Settings ;
;;;;;;;;;;;;;;;;;;;;;;;;;

[date]
date.timezone = Europe/Paris

[mail function]
;SMTP = localhost
;smtp_port = 25
;sendmail_from = me@example.com
sendmail_path = sendmail -t -f apache2@bebeserv.local

[mysql]
mysql.connect_timeout = 10

[session]
session.use_only_cookies = 1
session.name = SESSID
session.cache_limiter =
session.hash_function = 1

```

Enregistrez et quittez. Maintenant PHP est configuré pour travailler avec Apache mais pas pour fonctionner en ligne de commande.

Refaites donc la même chose pour le CLI :

```
bebeserv:/etc/apache2# cd ../cli
bebeserv:/etc/php5/cli# nano php.ini
```

Mettez exactement les même paramètre à la différence de ceux-ci :

```
;open_basedir =
; ignore_user_abort = On
max_execution_time = 60
max_input_time = 10
memory_limit = 32M
log_errors = Off
file_uploads = Off
```

Enregistrez et quittez.

Il nous reste à créer le fichier de log de PHP et de lui créer sa rotation :

```
bebeserv:/home/nobody/default_apache# cd /var/log
bebeserv:/var/log# touch php5.log
bebeserv:/var/log# cd /etc/logrotate.d/
bebeserv:/etc/logrotate.d# nano php5
```

Créez le fichier suivant :

```
/var/log/php5.log {
    weekly
    missingok
    rotate 4
    compress
    delaycompress
    notifempty
    create 640 root root
}
```

Voilà c'est terminer pour la configuration de PHP.

A l'heure actuelle avec la directive « open_basedir », PHP est enfermé dans « /home/nobody/default_apache » pour chaque site que nous mettrons en place (et donc chaque virtual host) nous mettrons une directive « open_basedir » dans le virtual host afin que PHP soit ouvert uniquement pour ce site. Ainsi nous mettrons d'autre élément de configuration PHP dans la configuration des vhosts. Cette méthode offre le grand avantage d'offrir au dessus de la configuration générale de PHP beaucoup de finesse pour ajuster certain paramètre notamment de sécurité.

C'est ce que nous allons voir tout de suite en installant Dotclear 2. Un exemple simple et facile afin d'illustrer ce que nous venons de mettre en place durant les 4 derniers chapitres.

Mais avant de nous lancer, vérifions le bon fonctionnement de PHP et pour ceci dans notre vhost « default_apache » créons un fichier « php.php » qui nous donnera le phpinfo.

```
bebeserv:/etc/logrotate.d# cd /home/nobody/default_apache
bebeserv:/home/nobody/default_apache# nano php.php
```

Mettez ceci dans votre fichier :

```
<?php
phpinfo();
?>
```

Enregistrez et quittez.

Donnez les bons droits à votre fichier :

```
bebeserv:/home/nobody/default_apache# chown nobody php.php
```

Rechargeons la configuration « apache2 » ainsi que « php5 ».

```
bebeserv:/home/nobody/default_apache# apachectl graceful
[...]
```

Maintenant observons notre phpinfo :

Depuis un poste du réseau local allez à cette adresse : « <http://10.0.0.2/php.php> ».

Passons à la partie facultative de ce chapitre :

L'installation de Dotclear 2

Sachant que vous avez votre propre nom de domaine ici nous ferons une installation pour le réseau locale. il faut savoir que Dotclear ne requiert pas mais serait mieux avec la librairie GD2. ! Nous allons donc l'installer.

```
bebeserv:/home/nobody/default_apache# apt-get install php5-gd
```

```
[...]
```

Rechargeons la configuration de PHP pour intégrer GD :

```
bebeserv:/home/nobody/default_apache# apace2ctl graceful  
[...]
```

Vous pouvez à nouveau regarder le phpinfo pour observer que GD est bien intégré dans sa version 2.

Maintenant nous allons préparer l'arborescence pour notre futur site.

Afin de ne pas refaire la même manipulation à chaque fois, nous allons générer un squelette qu'il suffira de recopier :

```
bebeserv:/home/nobody/default_apache# cd ..  
bebeserv:/home/nobody# mkdir skeleton  
bebeserv:/home/nobody# cd skeleton  
bebeserv:/home/nobody/skeleton# mkdir backup html includes logs  
sessions tmp  
bebeserv:/home/nobody/skeleton# cd logs  
bebeserv:/home/nobody/skeleton/logs# touch apache2.log php5.log  
bebeserv:/home/nobody/skeleton/logs# cd ../../  
bebeserv:/home/nobody# chown -R nobody skeleton
```

Voilà une bonne chose de faite, maintenant créons la véritable arborescence de notre site que j'appellerai ici « `www.bebeserv.local` » :

```
bebeserv:/home/nobody# cp -ra skeleton www.bebeserv.local
```

Nous devons dorénavant nous créer un vhost pour pouvoir exploiter tout ceci.

```
bebeserv:/home/nobody# cd /etc/apache2/site-available  
bebeserv:/etc/apache2/site-available# nano www.bebeserv.local
```

Créez le fichier suivant :

```
<VirtualHost *>  
    ServerSignature Off  
    ServerAdmin webmaster@bebeserv.local  
    ServerName www.bebeserv.local  
    ServerAlias bebeserv.local
```

```

ServerAlias 10.0.0.2

DocumentRoot /home/nobody/www.bebeserv.local/html/

<Directory /home/nobody/www.bebeserv.local/html/>
    AllowOverride All
    Order allow,deny
    allow from all
</Directory>

<IfModule mod_php5.c>
    php_admin_value open_basedir
/home/nobody/www.bebeserv.local
    php_admin_value error_log
/home/nobody/www.bebeserv.local/logs/php5.log
    php_admin_value upload_tmp_dir
/home/nobody/www.paradoxal.org/tmp
    php_admin_value include_path
./:/home/nobody/www.bebeserv.local/includes
    php_admin_value session.save_path
/home/nobody/www.bebeserv.local/sessions
    php_admin_value sendmail_path "sendmail -t -f
www@bebeserv.local"
</IfModule>

CustomLog
/home/nobody/www.bebeserv.local/logs/apache2.log combined
</VirtualHost>

```

Enregistrez et quittez.

J'ai mis en alias de serveur l'adresse IP de la machine afin qu'on puisse accéder au site localement pour le tester.

Autorisons ce vhost :

```

bebeserv:/etc/apache2/site-available# cd ../site-enabled
bebeserv:/etc/apache2/site-enabled# ln -s ../site-
available/www.bebeserv.local 100-www.bebeserv.local

```

Je garde une numérotation devant les nom des vhosts afin de définir l'ordre de chargement de ces derniers et ainsi leur priorité dans apache.

Puisque nous y sommes, ouvrons nous un compte FTP afin de compléter l'ouverture de notre futur blog :

```

bebeserv:/etc/apache2/site-enabled# cd /etc/vsftpd/user_conf
bebeserv:/etc/vsftpd/user_conf# cp example www.bebeserv.local

```

Dans le fichier « www.bebeserv.local » mettez les options qui vous plaisent et surtout :

```
local_root=/home/nobody/www.bebeserv.local
```

Nous pouvons ajouter notre utilisateur « `www.bebeserv.local` » à la base de donnée de « `vsftpd` » :

```
bebeserv:/etc/vsftpd/user_conf# cd ..  
bebeserv:/etc/vsftpd# nano userdb
```

Entrez les lignes :

```
www.bebeserv.local  
votre-mot-de-passe
```

Enregistrez et quittez puis rechargez la base de données de « `vsftpd` » :

```
bebeserv:/etc/vsftpd# vsftpd-makedb
```

Tout est OK, rechargeons aussi la configuration de apache :

```
bebeserv:/etc/vsftpd# apache2ctl graceful
```

Maintenant téléchargeons Dotclear 2 :

```
bebeserv:/etc/vsftpd# cd /home/nobody/www.bebeserv.local/html  
bebeserv:/home/nobody/www.bebeserv.local/html# wget  
http://download.dotclear.net/latest-2.0.tar.gz  
[...]
```

Décompressez l'archive :

```
bebeserv:/home/nobody/www.bebeserv.local/html# tar -xvzf dotclear-  
2.0-beta6.tar.gz  
[...]
```

Un dossier « `dotclear` » est créé, donnons lui les bons droits (ainsi qu'à tout ce qu'il contient) :

```
bebeserv:/home/nobody/www.bebeserv.local/html# chown -R
nobody.nogroup dotclear
```

Supprimons le tarball :

```
bebeserv:/home/nobody/www.bebeserv.local/html# rm dotclear-2.0-
beta6.tar.gz
```

Nous pouvons changer le nom du répertoire « dotclear » par « blog » par exemple :

```
bebeserv:/home/nobody/www.bebeserv.local/html# mv dotclear blog
```

Voilà c'est presque fini. Dotclear aura besoin d'une base de données, créons donc un utilisateur et une base pour Dotclear.

```
bebeserv:/home/nobody/www.bebeserv.local/html# mysql -u bebeserv -p
Entre password:
[...]

mysql> create database bebeserv_dc2;
Query OK, 1 row affected (0.00 sec)

mysql> grant create, delete, insert, update, select, index, lock
tables, alter, drop on bebeserv_dc2.* to 'dotclear2'@'localhost'
identified by 'un-mot-de-passe';
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> quit
Bye
bebeserv:/home/nobody/www.bebeserv.local/html#
```

Voilà c'est terminé, rendez vous sur la page « <http://10.0.0.2/blog/admin/install/> » pour terminer l'installation.

Pour le wizard de configuration Dotclear voici ce que vous aurez à entrer :

```
hote de la base : localhost
base de donnée bebeserv_dc2
utilisateurs de la base de donnée : dotclear2
```

Mémorisez bien le mot de passe qui vous sera fourni !

Voilà c'est terminé pour ce chapitre, vous avez intérêt à produire de bon billet maintenant !

```
bebeserv:/home/nobody/www.bebeserv.local/html# exit  
logout  
ALBAN@bebeserv:~$
```