Apache 1.3.37 et 2.2.4 + PHP5.2.1 + MySQL 5.0.33

Olivier Hoarau (olivier.hoarau@funix.org)

V2.8, 8 mars 2007

1 Historique

08/03/07 V2.8 Passage à php 5.2.1, Apache 2.2.4, MySQL 5.0.33, awstats 6.6 et phpMyAdmin 2.10.0.2

02/11/06 V2.7 passage à apache 1.3.37, php 5.1.6, MySQL 5.0.27 et phpMyAdmin 2.9.0.2 et apache 2.2.3 adaptation pour installation sur (k)ubuntu 6.10

06/08/05 V2.6 passage à PHP 5.0.4, MySQL 5.0.27, phpMyAdmin 2.6.3-pl1, awstat 6.4 et apache 2.0.54 rajout d'un mot pour le lancement automatique d'Apache 2, correction compilation webalizer

28/01/05 V2.5 analyser les logs d'Apache, modification de la configuration de webalizer et awstats pour ne pas prendre en compte ses propres accès, rajout d'un commentaire sur un message de webalizer, rajout d'un paragraphe pour voir les stats d'un site hébergé non localement avec webalizer et awstats

07/01/05 V2.4 passage à apache 1.33, PHP 5.0.3, MySQL 4.1.8a, phpMyAdmin 2.6.0-pl3, awstats 6.2, grosse modification au niveau de la configuration et de la méthode d'utilisation et modification de la configuration de webalizer pour qu'on puisse voir les noms des machines plutôt que leur adresse IP

02/10/04 V2.3 passage à apache 2.0.52, PHP 5.0.2, phpMyAdmin 2.6.0 pl1 et MySQL 4.0.21

31/05/04 V2.2 passage à apache 1.3.31 et MySQL 4.0.20

18/04/04 V2.1 un mot sur la variable register_globals, passage à http 2.0.49, php 4.3.6 et phpMyAdmin 2.5.6.

20/02/04 V2.0 Passage à Apache 1.3.29, Apache 2.0.48, MySQL 4.0.18, PHP 4.3.4, phpMyAdmin 2.5.5-pl1, rajout d'un paragraphe sur la sécurisation d'Apache et PHP, un mot sur l'utilisation simultanée d'Apache 1.3 et 2.

06/04/03 V1.9 Passage à MySQL 4.0.12

01/03/03 V1.8 Passage à Apache 2.0.44, MySQL 3.23.55, PHP 4.3.1, phpMyAdmin 2.4.0 et awstats 5.4

24/12/02 V1.7 Petites modifs dans le fichier de conf d'Apache suite passage Mdk 9.0, passage à awstats 5.2

13/10/02 V1.6 Passage à Apache 1.3.27, Apache 2.0.43, PHP 4.2.3, MySQL 3.23.52, phpMyAdmin 2.3.2, awstats 5.0 et HtDig 3.1.6

07/07/02 V1.5 Passage à Apache 1.3.26, PHP 4.2.1, MySQL 3.23.51, phpMyAdmin 2.2.6, Webalizer 2.01-10, awstats 4.1, rajout d'un paragraphe sur l'installation d'un moteur de recherche avec Ht://dig et sur l'installation d'Apache 2.0.36

27/05/01 V1.4 Passage à Apache 1.3.20, PHP 4.0.5 et MySQL 3.23.38

04/03/01 V1.3 Passage à PHP4.0.4pl1, Apache 1.3.19, MySQL 3.23.33, GD 1.8.4 et awstat 2.23

03/12/00 V1.2 Rajout du paragraphe PHP et LDAP, mise à jour de webalizer (v2.01.06)

22/10/00 V1.1 Passage à Apache 1.3.14, PHP 4.0.3pl1 et phpMyAdmin 2.1.0

25/07/00 V1.0 Création du document

2 Préambule

Ce document présente l'installation, la configuration et l'utilisation d'Apache 1.3 et 2, MySQL et PHP sous Linux.
La dernière version de ce document est téléchargeable à l'URL http://www.funix.org. Ce document peut être reproduit et distribué librement dès lors qu'il n'est pas modifié et qu'il soit toujours fait mention de son origine et de son auteur, si vous avez l'intention de le modifier ou d'y apporter des rajouts, contactez l'auteur pour en faire profiter tout le monde.
Ce document ne peut pas être utilisé dans un but commercial sans le consentement de son auteur. Ce document vous est fourni « dans l'état » sans aucune garantie de toute sorte, l'auteur ne saurait être tenu responsable des quelconques misères qui pourraient vous arriver lors des manipulations décrites dans ce document.

3 Présentation

Cette page présente l'installation et la configuration d'**Apache 1.3.37** avec gestion de **PHP 5.2.1** et **MySQL 5.0.33** , elle est basée sur la compilation complète de ces applications par conséquent elle devrait convenir pour toutes les distributions de linux.

Vous devez récupérer préalablement les sources d' **Apache** à l'URL www.apache.org , on récupère ensuite **PHP** à l'URL www.php.net, et enfin **MySQL** , à l'URL www.mysql.com.

Pour l'analyser des logs d'Apache, reportez vous au paragraphe correspondant.

Pour mettre en place un moteur de recherche, reportez vous au paragraphe correspondant.

4 Installation

4.1 Installation de MySQL

4.1.1 Compilation et installation

Dans le répertoire de travail, on décompresse l'archive:

tar xvfz mysql-5.0.33.tar.gz

Cela crée le répertoire **mysql-5.0.33**, sur ma Mandriva j'ai du installer les packages **libncurses5-devel** et **orbit-devel** et installer le lien suivant

ln -s /usr/bin/orbit2-config /usr/bin/orbit-config

Sous ubuntu ce n'est pas nécessaire. Dans le répertoire de **MySQL**, on tape alors:

./configure

Puis

make

La compilation est relativement longue même sur une machine puissante. On tape maintenant en temps que **root**:

make install

Les exécutables sont installés par défaut sous **/usr/local/bin** et le serveur **mysqld** sous **/usr/local/libexec**. Les bibliothèques vont se trouver sous **/usr/local/lib/mysql**, rajoutez cette ligne dans le fichier **/etc/ld.so.conf** et tapez

ldconfig

Si c'est la première fois que vous installez **MySQL** vous devez taper la commande suivante, vous pouvez passer à l'étape de configuration si vous upgradez **MySQL** :

./mysql-5.0.33/scripts/mysql_install_db

Voilà le résultat obtenu

Installing all prepared tables
Fill help tables

To start mysqld at boot time you have to copy support-files/mysql.server
to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:
/usr/local/bin/mysqladmin -u root password 'new-password'
/usr/local/bin/mysqladmin -u root -h mana password 'new-password'
See the manual for more instructions.

NOTE: If you are upgrading from a MySQL <= 3.22.10 you should run
the /usr/local/bin/mysql_fix_privilege_tables. Otherwise you will not be
able to use the new GRANT command!

You can start the MySQL daemon with:
cd /usr/local ; /usr/local/bin/mysqld_safe &

You can test the MySQL daemon with the benchmarks in the 'sql-bench' directory:
cd sql-bench ; perl run-all-tests

Please report any problems with the /usr/local/bin/mysqlbug script!

The latest information about MySQL is available on the web at
http://www.mysql.com
Support MySQL by buying support/licenses at http://shop.mysql.com

Pour info les bases sont installées par défaut sous **/usr/local/var**. Si ce n'est pas déjà fait créer le groupe **mysql**

groupadd mysql

Puis l'utilisateur **mysql** du groupe **mysql**

useradd -g mysql mysql

Les bases se trouvent par défaut sous **/usr/local/var**, **mysql** doit en être propriétaire

chown -R mysql /usr/local/var

idem pour le groupe

chgrp -R mysql /usr/local/var

Puis on lance le serveur en tapant:

scripts/mysqld_safe &

Ce qui va donner comme résultat

Starting mysqld daemon with databases from /usr/local/var

Avec cette dernière version **mysql** refusait de se lancer, dans le fichier d'erreur (sous **/usr/local/var** par défaut) j'avais

041002 12:18:13 mysqld started
/usr/local/libexec/mysqld: Can't read dir of '/root/tmp/' (Errcode: 13)
/usr/local/libexec/mysqld: Can't create/write to file '/root/tmp/ibqQdaUA' (Errcode: 13)
041002 12:18:13 InnoDB: Error: unable to create temporary file; errno: 13
041002 12:18:13 Can't init databases
041002 12:18:13 Aborting

j'ai résolu cela en tapant avant de lancer **mysqld_safe**

export TMPDIR=/tmp

4.1.2 Lancement automatique

Pour que **MySQL** soit lancé automatiquement au boot, on modifiera le fichier **mysql.server.sh**. Dans ce fichier au lieu de **@HOSTNAME@** on écrit **hostname**, on modifie également la variable **basedir** (au lieu d'avoir **basedir=** on écrit **basedir=/usr/local**) et pour finir la variable **datadir** (au lieu de **datadir=** on écrit **datadir=/usr/local/var**).

En admettant que votre poste soit connecté sur internet et que vous vouliez pas que n'importe qui puisse accéder à votre serveur internet, vous pouvez préciser l'adresse IP locale de votre serveur pour que seuls les postes de votre réseau puisse y accéder, on modifie la ligne suivante comme ceci

$bindir/mysqld_safe --datadir=$datadir --pid-file=$server_pid_file --bind-address=192.168.26.110 $other_args >/dev/null 2>&1 &

Vous mettre l'adresse IP local de votre serveur (127.0.0.1 si poste isolé). Par ailleurs si vous avez l'erreur mentionné plus haut rajouté (à la suite de la déclaration de **PATH**)

TMPDIR=/tmp
export TMPDIR

Maintenant sous Mandriva

on copie le fichier sous **/etc/rc.d/init.d** et on le renommera **mysql**

cp ./mysql-5.0.33/support-files/mysql.server.sh /etc/rc.d/init.d/mysql

On donne les droits d'exécution à ce fichier

chmod 755 /etc/rc.d/init.d/mysql

Pour un lancement automatique de **MySQL** à l'état de marche 3, 4 et 5, on tapera ensuite

chkconfig --level 345 mysql on

Et un arrêt à l'état de marche 0, 1, 2 et 6

chkconfig --level 0126 mysql off

Sous ubuntu

On copie le fichier sous **/etc/init.d** et on le nomme **mysql**, on doit dans ce fichier faire une dernière modif à la place de source on met un . (un point). On donne les droits d'exécution à ce fichier

chmod 755 /etc/init.d/mysql

Pour un lancement automatique on tape

update-rc.d mysql defaults

4.1.3 Mise en place des utilisateurs

La première chose à faire est de mettre un mot de passe pour **root** pour l'accès à l'administration des bases de données. La commande à taper en tant que root est:

mysql_secure_installation

voilà le résultat

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MySQL to secure it, we'll need the current password for the root user. If you've just installed MySQL, and you haven't set the root password yet, the password will be blank, so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MySQL root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n] n
... skipping.

By default, a MySQL installation has an anonymous user, allowing anyone to log into MySQL without having to have a user account created for

them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

Remove anonymous users? [Y/n] Y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y
... Success!

By default, MySQL comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

Remove test database and access to it? [Y/n] Y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? [Y/n] Y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MySQL installation should now be secure.

Thanks for using MySQL!

Le mot de passe peut être différent de celui du login.

Maintenant on va créer un compte utilisateur (**olivier** dans mon exemple), pour cela on doit se connecter en tant qu'administrateur de la base à la base de donnée **mysql** contenant les infos sur les utilisateurs et leurs droits.

[root@asterix RPMS]# mysql -u root -p mysql
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 4 to server version: 5.0.33

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>

A présent on va entrer l'utilisateur **olivier** qui sera un super utilisateur avec les mêmes droits que root:

```
mysql> INSERT INTO user
    -> VALUES('localhost','olivier',PASSWORD('mot-de-passe'),
    -> 'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y',' ',' ',' ',' ','0','0','0');
Query OK, 1 row affected (0.00 sec)
```

NOTE Il n'est pas obligatoire de rentrer le login pour le nom d'utilisateur et le mot de passe de login.

Pour voir si la saisie s'est bien passée:

```
+-----------+---------+-------------------+-------------+-------------+-------------+-------------+-----------+-------------+-------------+-------------+-----------+-----------+-----------------+-----------+-----------
+-------------+-----------+-------------------+-----------------+-------------+---------------+---------------+---------+-----------+-----------+-------------+-----------+-----------+-----------------+
| Host      | User    | Password          | Select_priv | Insert_priv | Update_priv | Delete_priv | Create_priv | Drop_priv | Reload_priv | Shutdown_priv | Process_priv | File_priv | Grant_priv | References_priv |
Index_priv | Alter_priv | Show_db_priv | Super_priv | Create_tmp_table_priv | Lock_tables_priv | Execute_priv | Repl_slave_priv | Repl_client_priv | ssl_type | ssl_cipher | x509_issuer | x509_subject |
max_questions | max_updates | max_connections |
+-----------+---------+-------------------+-------------+-------------+-------------+-------------+-----------+-------------+-------------+-------------+-----------+-----------+-----------------+-----------+-----------
+-------------+-----------+-------------------+-----------------+-------------+---------------+---------------+---------+-----------+-----------+-------------+-----------+-----------+-----------------+
| localhost | root    | 6bf57a02084c345b | Y         | Y         | Y         | Y         | Y         | Y       | Y         | Y         | Y           | Y         | Y         | Y         | Y         | Y         | Y         | Y         |
Y               | Y             | Y         | Y           | Y             |       |       |       |       |         0 |       0 |         0 |

| localhost | olivier | 588c797a754bd00a | Y         | Y         | Y         | Y         | Y         | Y       | Y         | Y         | Y           | Y         | Y         | Y         | Y         | Y         | Y         | Y         |
Y               | Y             | Y         | Y           | Y             |       |       |       |       |         0 |       0 |         0 |
+-----------+---------+-------------------+-------------+-------------+-------------+-------------+-----------+-------------+-------------+-------------+-----------+-----------+-----------------+-----------+-----------
+-------------+-----------+-------------------+-----------------+-------------+---------------+---------------+---------+-----------+-----------+-------------+-----------+-----------+-----------------+
5 rows in set (0.00 sec)
```

Maintenant pour prendre tout ça en compte.

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.08 sec)
```

Pour quitter

```
mysql>quit
```

4.1.4 Création de tables

Maintenant notre utilisateur olivier va créer une table qui nous servira plus tard pour nos expérimentations avec **Apache**. Il doit d'abord se connecter:

```
mysql -u olivier -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5 to server version: 5.0.33

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Pour voir la liste des bases de données disponibles, on tapera:

```
mysql> SHOW DATABASES;
+----------------+
| Database       |
+----------------+
| mysql          |
+----------------+
4 rows in set (0.00 sec)
```

On va maintenant créer une base de données **essai**:

```
mysql> CREATE DATABASE essai;
Query OK, 1 row affected (0.00 sec)
```

On va utiliser maintenant cette base de donnée

```
mysql> USE essai
Database changed
```

Comme la base vient d'être créée, elle ne contient aucune table, pour s'en convaincre il suffit de taper:

```
mysql> SHOW TABLES;
Empty set (0.00 sec)
```

Pour notre première exemple **Apache+PHP+MySQL**, on va créer la table suivante:

```
mysql> CREATE TABLE coord (
   -> nom VARCHAR(20),
   -> prenom VARCHAR(20),
   -> email VARCHAR(30)
   -> );
Query OK, 0 rows affected (0.03 sec)
```

Jetons un coup d'œil maintenant sur les tables disponibles:

```
mysql>  SHOW TABLES;
+--------------------+
| Tables in essai    |
+--------------------+
| coord              |
```

```
+--------------------+
1 row in set (0.00 sec)
```

La table nouvellement crée apparaît bien. Pour avoir le détail de cette table, on tapera:

```
mysql> DESCRIBE coord;
+-----------+-------------+--------+------+-----------+------+
| Field     | Type        | Null   | Key  | Default |Extra |
+-----------+-------------+--------+------+-----------+------+
| nom       | varchar(20) | YES |        | NULL   |       |
| prenom  | varchar(20) | YES |         | NULL   |       |
| email     | varchar(30) | YES |        | NULL   |       |
+-----------+-------------+--------+------+-----------+------+
3 rows in set (0.00 sec)
```

Pour notre deuxième exemple **Apache+PHP+MySQL**, on créera la table suivante:

```
 mysql> CREATE TABLE ref (
       -> date VARCHAR(20),
       -> host VARCHAR(20),
      -> ip VARCHAR(15),
      -> os VARCHAR(20),
      -> page VARCHAR(30)
      -> );
Query OK, 0 rows affected (0.05 sec)
```

Elle contiendra les informations sur les visiteurs du site. A présent pour quitter tapez simplement **quit**.

4.2 Installation d'Apache

Pour **Apache**, on désarchive en tapant:

tar xvfz apache_1.3.37.tar.gz

Cela va créer le répertoire **apache_1.3.37**. Dans ce répertoire, on tape alors:

./configure --prefix=/usr/local/apache --enable-module=most --enable-shared=max

Par **prefix** on indique que le fichier de conf se trouvera sous **/usr/local/apache/conf**. Sous Mandriva j'ai du préalablement installer le package **libgdbm2-devel**. Sous ubuntu il faudra installer le package **libgdbm-dev** puis créer le lien suivant

ln -s /usr/include/gdbm-ndbm.h /usr/include/ndbm.h

sous ubuntu j'obtiens l'erreur suivante

```
 + Warning: Your 'echo' command is slightly broken.
 + It interprets escape sequences per default. We already
 + tried 'echo -E' but had no real success. If errors occur
 + please set the SEO variable in 'configure' manually to
 + the required 'echo' options, i.e. those which force your
 + 'echo' to not interpret escape sequences per default.
 + using installation path layout: Apache (config.layout)
```

pour la corriger on édite le fichier **configure** et à la première ligne au lieu de **#!/bin/sh** on écrit **#!/bin/bash** on retape ensuite **configure** avec toutes ses options puis

make

et en tant que root:

make install

Il faudra modifier quelque peu le fichier **httpd.conf** se trouvant sous **/usr/local/apache/conf** notamment

ServerName nomdevotremachine

et

Port 80

Pour le reste on verra plus tard pour améliorer. Pour lancer maintenant **Apache**, il faut taper :

/usr/local/apache/bin/apachectl start

si vous rencontrez l'erreur suivante

Cannot load /usr/local/apache/libexec/mod_auth_dbm.so into server: /usr/local/apache/libexec/mod_auth_dbm.so: undefined symbol: dbm_fetch

Dans le répertoire d'**Apache** précisément sous **src/modules/standard** on efface **mod_auth_dbm.so** et on tape

gcc -shared -o mod_auth_dbm.so mod_auth_dbm.lo -lgdbm -lgdbm_compat

on copie maintenant le module dans le répertoire qui va bien

cp mod_auth_dbm.so /usr/local/apache/libexec

et on peut maintenant lancer **apache.** A présent dans le répertoire **/usr/local/apache/htdocs** tapez

mv index.html.fr index.html

Maintenant dans votre navigateur préféré dans le champ URL taper **http://localhost** ou **http://nomdelamachine** et là la page d'accueil d'**Apache** apparaît, pour info celle-ci se trouve sous **/usr/local/apache/htdocs**.

NOTE Si vous upgradez d'une ancienne version, vos fichiers de conf ne seront pas écrasés.

4.3 Installation de PHP

Pour **php5**, on tapera d'abord

tar xvfj php-5.2.1.tar.bz2

Cela va créer un répertoire **php-5.2.1**. A présent dans le répertoire de **php**, vous taperez

./configure --with-apxs=/usr/local/apache/bin/apxs --with-config-file-path=/usr/local/apache/conf --enable-versioning --with-mysql --with-ftp --enable-bcmath=yes --enable-debug=no --enable-memory-limit=yes --enable-tracks-vars --with-gd --with-zlib --with-kerberos --enable-mbstring --with-mcrypt

Notes - Vous avez besoin du package **flex** contenant **lex** ainsi que du package **libxml2-devel**

- les options **--with-gd --with-zlib --with-kerberos** sont utiles pour l'utilisation des mods freeplayer de la freebox

- les options **--enable-mbstring --with-mcrypt** sont nécessaires pour **phpMyAdmin** elles requièrent l'installation des packages **libmcrypt-dev et mcrypt**

Tapez maintenant

make

Puis en tant que root

make install

Cette dernière commande va installer le module **PHP** sous **/usr/local/apache/libexec** et modifier automatiquement le fichier **httpd.conf** pour qu'**Apache** prenne en compte **PHP**, et rajouter les lignes suivantes :

LoadModule php5_module libexec/libphp5.so

Supprimez les lignes qui font référence au chargement du module pour **PHP4** dans le cas d'une ancienne installation.

Au niveau de

AddType application/x-tar .tgz

On rajoutera

AddType application/x-httpd-php .php .php .php4 .php5 .phtml
AddType application/x-httpd-php-source .phps

Par ailleurs à la ligne

DirectoryIndex index.html

On rajoutera

DirectoryIndex index.html index.htm index.php index.php index.php4 index.php5

A présent on va copier le fichier **php.ini-dist** se trouvant dans le répertoire de **PHP ./php-5.2.1** pour le mettre sous **/usr/local/apache/conf** et en le renommant **php.ini**

cp ./php-5.2.1/php.ini-dist /usr/local/apache/conf/php.ini

Dans le cas d'une ancienne installation de **php**, vous avez tout intérêt à repartir du nouveau fichier **php-dist.ini** fourni et de le remodifier.
On relance **Apache**

/usr/local/apache/bin/apachectl restart

Créer maintenant le fichier **infophp.php** contenant

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

Que vous placerez sous **/usr/local/apache/htdocs**, dans l'URL de votre navigateur préféré, taper **http://localhost/infophp.php** ou **http://nommachine/infophp.php** et là magique devrait s'afficher des info sur la configuration de **PHP** sur votre système, dont le tableau suivant:

PHP Version 5.2.1

System	Linux mana 2.6.15-26-386 #1 PREEMPT Mon Jul 17 19:52:53 UTC 2006 i686
Build Date	Mar 7 2007 16:58:38
Configure Command	'./configure' '--with-apxs=/usr/local/apache/bin/apxs' '--with-config-file-path=/usr/local/apache/conf' '--enable-versioning' '--with-mysql' '--with-ftp' '--enable-bcmath=yes' '--enable-debug=no' '--enable-memory-limit=yes' '--enable-tracks-vars' '--with-gdbm'
Server API	Apache
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/apache/conf/php.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, data, http, ftp
Registered Stream Socket Transports	tcp, udp, unix, udg
Registered Stream Filters	string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v2.2.0, Copyright (c) 1998-2007 Zend Technologies

Powered By Zend Engine 2

ATTENTION Si comme moi vous utilisez des URL du style **http://www.funix.org/fr/linux/main-linux.php3?ref=apache&page=menu** c'est à dire avec passage d'arguments, à partir de la version 4.2.0 de **php** cela risque de ne pas fonctionner. Le passage d'argument est régi entre autres par la variable **register_globals** qui a été désactivée dans le fichier **php.ini** pour des raisons de sécurité. Pour la réactiver éditez **/usr/local/apache/conf/php.ini** et au lieu de

register_globals = Off

Mettez

register_globals = On

Relancez **Apache**. Cela dit vous avez intérêt à laisser cette variable à Off et procéder comme suit. Voilà le bout de code php des pages html de ce site pour faire l'inclusion du menu à gauche ou pas. Voilà la code avec **register_globals = On**

```
<?
    if ($page=="menu")
    {
        $menu="menu.htm";
        include($menu);
    }
?>
```

Et le même code avec **register_globals = Off**

```
<?
    if ($_GET["page"]=="menu")
    {
        $menu="menu.htm";
        include($menu);
    }
?>
```

Pour plus d'info sur cette fonctionnalité désactivée et sur la manière de faire autrement http://www.php.net/manual/en/security.registerglobals.php

5 Configuration basique

5.1 Présentation de l'arborescence d'Apache

L'installation va créer un répertoire **/usr/local/apache** contenant:
- répertoire **bin** contient les exécutables d'Apache
- répertoire **cgi-bin** contient les scripts CGI
- répertoire **conf** contient les fichiers de configuration d'Apache
- répertoire **htdocs** contient la page d'accueil d'Apache

Dans le répertoire **/usr/local/apache/htdocs** (répertoire d'accueil par défaut) vous devez taper pour avoir la page d'accueil en français:

mv index.html.fr index.html

- répertoire **icons** contient des icônes qui servent notamment pour identifier les types de fichier.
- répertoire **include** contient les includes d'Apache
- répertoire **libexec** contient les modules d'Apache
- répertoire **logs** contient les fichiers de log d'Apache
- répertoire **man** contient les mans d'Apache
- répertoire **proxy** Apache peut être utilisé comme un proxy, c'est le répertoire cache pour cette fonction

Le répertoire de log contient essentiellement deux fichiers:
- **access_log** listant les accès au serveur
- **error_log** listant les erreurs en tout genre

Le répertoire de modules **libexec** contient tous les modules utilisables par Apache, pour info un module est une extension logicielle à Apache, lui permettant par exemple d'interpréter le PHP (module **libphp5.so**), ou alors de créer des hôtes virtuels (module **mod_vhost_alias.so**).

Le répertoire **/usr/local/apache/conf** contient:
- le fichier de configuration d'Apache **http.conf**
- **mime.types** fixe le type de fichier suivant l'extension du dit fichier (.**doc**=msword, .**ps**=postscript, ...), ça permet au client qui se connecte sur le serveur, de savoir comment interpréter le fichier suivant son extension.
- **magic** sert pour le module **mod_mime_magic**
- **php.ini** pour contrôler certains aspects de PHP

5.2 Le fichier de configuration d'Apache

Le fichier de conf d'**Apache** se trouve sous **/usr/local/apache** et se nomme **httpd.conf**, voici les points que je juge important dans le fichier:

(...)

 # ---------------------- Server Configuration ----------------------------

 # ServerType is either inetd, or standalone.
 # vous pouvez soit lancer Apache par le super daemon inetd (config dans /etc/inetd.conf) soit tout seul avec un script dans /etc/rc.d/init.d, à noter qu'avec la première méthode vous pouvez contrôler l'accès avec les TCP_Wrappers (hosts.deny, hosts.allow)
 # voir la note en bas du paragraphe pour les avantages de l'un ou de l'autre
 # moi j'ai choisi lancement en standalone (voir paragraphe suivant)

ServerType standalone

(...)

 # Do NOT add a slash at the end of the directory path.
 # Répertoire racine d'Apache
ServerRoot "/usr/local/apache"

(...)

 # PidFile: The file in which the server should record its process
 # identification number when it starts.
 # C'est dans ce fichier qu'on trouvera l'identification du process d'Apache
 # c'est utile pour ne pas lancer deux fois Apache par exemple
PidFile /usr/local/apache/logs/httpd.pid

(...)

 # Limit on total number of servers running, i.e., limit on the number
# of clients who can simultaneously connect --- if this limit is ever
 # reached, clients will be LOCKED OUT, so it should NOT BE SET TOO LOW.
 # It is intended mainly as a brake to keep a runaway server from taking
 # Unix with it as it spirals down...
 # ici on fixe le nombre de clients max qui peuvent se connecter à un moment donné

MaxClients 150

(...) suit ensuite une liste de modules, ceux avec un **#** devant ne seront pas inclus

LoadModule vhost_alias_module libexec/mod_vhost_alias.so
LoadModule env_module libexec/mod_env.so

LoadModule config_log_module libexec/mod_log_config.so
LoadModule mime_magic_module libexec/mod_mime_magic.so
LoadModule mime_module libexec/mod_mime.so
LoadModule negotiation_module libexec/mod_negotiation.so
LoadModule status_module libexec/mod_status.so
LoadModule info_module libexec/mod_info.so

....

(...) Pour chaque module on doit maintenant la routine **AddModule**

ClearModuleList
AddModule mod_vhost_alias.c
AddModule mod_env.c
AddModule mod_log_config.c
AddModule mod_mime_magic.c
AddModule mod_mime.c
AddModule mod_negotiation.c
AddModule mod_status.c
AddModule mod_info.c
AddModule mod_include.c
AddModule mod_autoindex.c
AddModule mod_dir.c
AddModule mod_cgi.c

(...)

 # Port: The port the standalone listens to. For ports < 1023, you will
 # need httpd to be run as root initially.
 # numéro de port pour le serveur, traditionnellement à 80

Port 80

(...)

 # On lance initialement httpd en tant que root, puis immédiatement
 # c'est l'utilisateur nobody (groupe nobody) qui en devient le proprio
 # ainsi s'il y a une faille dans Apache, le hacker au lieu de devenir root
 # devient nobody avec les droits qui vont avec
 # pour vérifier que nobody est bien le proprio
 # ps aux | grep httpd
User nobody
Group nogroup

 # ServerAdmin: Your address, where problems with the server should be
 # e-mailed. This address appears on some server-generated pages, such
 # as error documents.
 # En cas de problème un email sera envoyé au webmaster, mettez donc
 # ici l'adresse email du webmaster
ServerAdmin olivier@asterix.kervao.fr

 # DocumentRoot: The directory out of which you will serve your
 # documents. By default, all requests are taken from this directory, but
 # symbolic links and aliases may be used to point to other locations.
 # C'est dans ce répertoire qu'on va trouver la page d'accueil d'Apache
DocumentRoot "/usr/local/apache/htdocs"

(...)

```
# UserDir: The name of the directory which is appended onto a user's home
# directory if a ~user request is received.
# Chaque utilisateur pourra mettre ses pages dans sa homedirectory
# dans un répertoire public_html, les pages seront accessibles à l'URL
# http://localhost/~user
<IfModule mod_userdir.c>
   UserDir public_html
</IfModule>
```

(...)

```
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
# L'emplacement du fichier de log peut ne pas vous convenir
# si vous voulez le placer dans /var/log/httpd il faut modifier ici
ErrorLog /usr/local/apache/logs/error_log
```

(...)

```
# The location and format of the access logfile (Common Logfile Format).
# If you do not define any access logfiles within a <VirtualHost>
# container, they will be logged here.  Contrariwise, if you *do*
# define per-<VirtualHost> access logfiles, transactions will be
# logged therein and *not* in this file.
# De même pour l'emplacement du fichier de log des accès à apache
# c'est cette variable qu'il faut modifier
CustomLog /usr/local/apache/logs/access_log common
```

(...)

```
# DirectoryIndex: Name of the file or files to use as a pre-written HTML
# directory index.  Separate multiple entries with spaces.
# liste des fichiers d'entrée des pages
<IfModule mod_dir.c>
   DirectoryIndex index.html index.htm index.php3 index.php index.php4
</IfModule>
```

(...)

```
# priorité dans les langages à utiliser, mettez fr en premier
   <IfModule mod_negotiation.c>
      LanguagePriority fr en da nl et de el it ja pl pt pt-br ltz ca es sv
   </IfModule>
```

(...)

```
# Apache peut se comporter comme un proxy comme squid
# par défaut cette fonction n'est pas activée
# Proxy Server directives. Uncomment the following line to
# enable the proxy server:

# ProxyRequests On

# To enable the cache as well, edit and uncomment the following lines:

# CacheRoot /var/cache/httpd
# CacheSize 5
```

 # CacheGcInterval 4
 # CacheMaxExpire 24
 # CacheLastModifiedFactor 0.1
 # CacheDefaultExpire 1
 # NoCache a_domain.com another_domain.edu joes.garage_sale.com

NOTE "Anciennement" on trouvait un fichier **srm.conf** et **access.conf**, ils peuvent maintenant complètement être intégrés dans **httpd.conf**

5.3 Lancement automatique de l'application

5.3.1 Sous Mandriva

On prendra le fichier **apachectl** se trouvant sous **/usr/local/apache/bin** et on le placera sous **/etc/rc.d/init.d**, et on le renommera **httpd**

cp /usr/local/apache/bin/apachectl /etc/rc.d/init.d/httpd

On rajoutera en début de fichier juste après **#!/bin/sh**

#!/bin/sh

 # Startup script for the Apache Web Server
#
 # chkconfig: 345 85 15
**# description: Apache is a World Wide Web server. It is used to serve **
 # HTML files and CGI.
 # processname: httpd

Pour un lancement automatique à l'état de marche 3,4 et 5 on doit normalement taper:

chkconfig --level 345 httpd on

Et pour un arrêt à l'état de marche 0, 1, 2 et 6

chkconfig --level 0126 httpd off

Pour lancer le serveur, il suffit maintenant de taper:

/etc/rc.d/init.d/httpd start

5.3.2 Sous ubuntu

Voilà un script de lancement, il faut l'appeler **httpd** et le placer sous **/etc/init.d**

#!/bin/bash
#
 # apache Start the apache HTTP server.
#
 # The variables below are NOT to be changed. They are there to make the
 # script more readable.

NAME=apache
DAEMON=/usr/local/apache/bin/httpd
PIDFILE=/usr/local/apache/logs/httpd.pid
CONF=/usr/local/apache/conf/httpd.conf
APACHECTL=/usr/local/apache/bin/${NAME}ctl
 # note: SSD is required only at startup of the daemon.
SSD=`which start-stop-daemon`

```
ENV="env -i LANG=C PATH=/bin:/usr/bin:/usr/local/bin"

trap "" 1

# Check that we're not being started by inetd
if egrep -q -i "^[[:space:]]*ServerType[[:space:]]+inet" $CONF
then
   exit 0
fi

test_config() {
   if [ ! -x $APACHECTL ]; then
   echo "$APACHECTL is not executable, exiting"
   exit 0
   fi

   # ensure we don't leak environment vars into apachectl
   APACHECTL="$ENV $APACHECTL"

   if ! $APACHECTL configtest 2> /dev/null
   then
      printf "Configuration syntax error detected. Not reloading.\n\n"
      $APACHECTL configtest
      exit 1
   fi
}

should_start() {
   if [ ! -x $DAEMON ]; then
   echo "apache is not executable, not starting"
   exit 0
   fi
}

case "$1" in
  start)
   should_start
   test_config
   echo -n "Starting web server: $NAME"
   $ENV $SSD --start --pidfile $PIDFILE --exec $DAEMON > /dev/null
   ;;

  stop)
   echo -n "Stopping web server: $NAME"
   start-stop-daemon --stop --pidfile $PIDFILE --oknodo
   rm -rf /var/lib/apache/mod-bandwidth/link/*
   ;;

  reload | force-reload)
   test_config
   echo -n "Reloading $NAME configuration"
   start-stop-daemon --stop --pidfile $PIDFILE --signal USR1
```

```
    ;;

  restart)
    test_config
    echo -n "Restarting $NAME"
    if ! start-stop-daemon -q --stop --pidfile $PIDFILE --signal HUP; then
    $ENV $SSD --start --pidfile $PIDFILE --exec $DAEMON > /dev/null
    fi
    ;;

  *)
    echo "Usage: /etc/init.d/$NAME {start|stop|reload|force-reload|restart}"
    exit 1
    ;;
esac

if [ $? -eq 0 ]; then
    echo .
    exit 0
else
    echo " failed"
    exit 1
fi
```

On donne les droits d'exécution

chmod 755 /etc/init.d/httpd

Pour le lancer automatique au démarrage il faudra taper

update-rc.d httpd defaults

5.4 Les pages webs utilisateurs

Le problème avec le répertoire **public_html** des utilisateurs et qu'il faut mettre 755 au niveau de la home directory, ce qui est particulièrement génant au niveau sécurité. Vous pouvez spécifier que chaque utilisateur doit créer ses pages sous **/home/http/login-utilisateur** en écrivant pour la variable **UserDir**

UserDir /home/httpd

Ainsi pour l'utilisateur toto quand vous taperez comme URL **http://serveur-apache/~toto** , apache ira chercher le fichier **index.htm** sous **/home/httpd/toto** . On peut aller plus loin en spécifiant un répertoire particulier, **/home/httpd/toto/html** par exemple, en écrivant:

UserDir /home/httpd/*/html

5.5 Les alias

Si vous ne voulez pas mettre en place un serveur DNS, vous avez un moyen plus simple, les alias. Concrètement, votre serveur s'appelle **obelix** , vous voulez rendre accessible les fichiers html se trouvant sous **/usr/doc/html** , les utilisateurs devront taper dans leur navigateur préféré: **http://obelix/doc**. Pour cela dans votre fichier **/etc/httpd/conf/httpd.conf** , vous allez rajouter:

```
Alias /icons/ "/usr/local/apache/icons/"
Alias /doc "/usr/doc/html/"
```

NOTE Si vous mettre **/doc/** à la place de **/doc** dans l'URL il faudra taper **http://obelix/doc/**, si vous omettez le dernier **/,** vous aurez une erreur.

6 Configuration avancée

6.1 Protection d'une page

La protection d'une page pour l'utilisateur **olivier** se fait de manière très simple, tous les fichiers à accès limité doivent être concentré dans un même répertoire **/home/olivier/public_html/reserve** par exemple, il suffit de créer dans celui-ci un fichier qu'on devra nommer **.htaccess** contenant:

AuthUserFile auth/olivier.users
AuthGroupFile auth/olivier.group
AuthName "Acces Restreint"
AuthType Basic
require group autorise

Le fichier **olivier.users** va contenir une liste d'utilisateurs, il va se trouver sous le répertoire **/usr/local/apache/auth** (éventuellement à créer), pour info vous pouvez changer le chemin **/usr/local/apache** en modifiant la valeur de la variable **ServerRoot** qu'on trouve dans le fichier **httpd.conf**. Le fichier **olivier.group** correspond à une liste des groupes de personnes, ces mêmes personnes ayant été défini dans le fichier **olivier.users**. Le principe consiste à créer un groupe de personnes autorisées et à leur attribuer un mot de passe à chacune, seul ce groupe pourra accéder à la section réservée.

Pour créer ces fichiers il suffit, en tant que root, d'une part de créer le répertoire **/usr/local/apache/auth**, puis de taper:

htpasswd -c /usr/local/apache/auth/olivier.users olivier

L'option **-c** correspond à la création du fichier **olivier.users**. On va alors avoir à rentrer un mot de passe pour l'utilisateur **olivier**.

New password:

On confirme

Re-type new password:
Adding password for user olivier

Pour créer un autre utilisateur **veronique** vous taperez la même commande sans l'option de création :

htpasswd /etc/httpd/auth/olivier.users veronique

Toujours en tant que root, créer le fichier **/usr/local/apache/auth/olivier.group** qui contiendra pour autoriser par exemple olivier et veronique à accéder aux pages réservées :

autorise: olivier veronique

Il faut décommenter maintenant les lignes suivantes dans le fichier **httpd.conf**:

```
<Directory /home/*/public_html>
   AllowOverride FileInfo AuthConfig Limit
   Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
   <Limit GET POST OPTIONS PROPFIND>
      Order allow,deny
      Allow from all
   </Limit>
   <LimitExcept GET POST OPTIONS PROPFIND>
      Order deny,allow
      Deny from all
```

</LimitExcept>
</Directory>

Ces lignes autorisent l'emploi du fichier **.htaccess** dans les pages de vos utilisateurs (répertoire **/home/*/public_html**)

Maintenant quand à partir de votre navigateur préféré quand vous allez rentrer comme URL **http://obelix/~olivier/reserve** , vous aurez une fenêtre popup qui va s'ouvrir vous demandant de rentrer votre nom d'utilisateur et le mot de passe préalablement rentré.

Notez que pour que quelqu'un ne puisse jeter un coup d'œil dans les fichiers **.htaccess** de vos utilisateurs**,** le fichier **httpd.conf** , contient la directive suivante:

<Files ~ "^\ht">
Order allow,deny
Deny from all
Satisfy All
</Files>

6.2 Les hôtes virtuels

On peut mettre en place des hôtes virtuels, en d'autres termes un utilisateur pour un même serveur Apache croira en voir plusieurs. Exemple, soit votre serveur Apache **obelix** (adresse IP **192.168.13.11**), votre domaine **breizland.bz**, on va créer les hôtes virtuels **www.asterix.breizland.bz** et **www.idefix.breizland.bz** qui vont pointer chacun vers un endroit différent du disque (respectivement **/usr/local/asterix** et **/usr/local/idefix** chacun contenant des pages html).
Dans le fichier **httpd.conf** le module est déjà chargé :

LoadModule vhost_alias_module libexec/mod_vhost_alias.so
AddModule mod_vhost_alias.c

On va rajouter tout à la fin du fichier:

NameVirtualHost 192.168.13.11

<VirtualHost 192.168.13.11>
ServerName obelix.breizland.bz
DocumentRoot /usr/local/apache/htdocs
ErrorLog logs/obelix-error_log
TransferLog logs/obelix-access_log
</VirtualHost>

<VirtualHost 192.168.13.11>
ServerName www.asterix.breizland.bz
DocumentRoot /usr/local/asterix
ErrorLog logs/asterix-error_log
TransferLog logs/asterix-access_log
</VirtualHost>

<VirtualHost 192.168.13.11>
ServerName www.idefix.breizland.bz
DocumentRoot /usr/local/idefix
ErrorLog logs/idefix-error_log
TransferLog logs/idefix-access_log
</VirtualHost>

Relancez **Apache** en tapant:

/etc/rc.d/init.d/httpd restart

Maintenant nous allons créer nos hôtes **asterix** et **idefix**, pour cela vous avez deux méthodes:

- rajouter **www.asterix.breizland.bz** et **www.idefix.breizland.bz** dans **/etc/hosts** sur la même ligne que votre serveur Apache (**obelix** dans notre exemple).

192.168.13.11 obelix obelix.breizland.bz www.asterix.breizland.bz www.idefix.breizland.bz

Normalement si vous faites un ping sur **www.idefix.breizland.bz** ça devrait marcher, pour les postes clients il faudra rajouter la même ligne dans le fichier **hosts** (non nécessaire).

- si vous disposez d'un serveur DNS sur votre machine , au niveau de votre config DNS dans votre fichier **breizland.bz** qui se trouve sous **/var/named** vous devez rajouter tout à la fin:

www.asterix A 192.168.13.11
www.idefix A 192.168.13.11

Relancez le DNS en tapant:

/etc/rc.d/init.d/named restart

Pour tester tapez dans un shell:

ping www.asterix.breizland.bz

Maintenant dans le champ URL de votre navigateur préféré:

http://www.asterix.breizland.bz

Et là, normalement vous devriez voir s'afficher la page que vous avez placé sous **/usr/local/asterix**

7 Gestion de bases de données avec MySQL

7.1 Tests de fonctionnement avec MySQL

On suppose que vous avez installé, configuré **MySQL** et créé les deux exemples du paragraphe installation **MySQL**. On suppose aussi que le serveur s'appelle **obelix** et l'utilisateur **olivier**.
Voici une page écrite en **PHP** qui va accéder à la base de donnée **essai** et à sa table **coord**.

```
<?
$serveur="localhost";
$login="olivier";
$pass="mot-de-passe";
$base="essai";
$table="coord";

$id=MYSQL_CONNECT($serveur,$login,$pass);
mysql_select_db($base);
$nom="hoarau";
$prenom="olivier";
$email="olivier.hoarau@fnac.net";
$query="INSERT INTO $table VALUES('$nom','$prenom','$email')";
$result=mysql_query($query,$id);
echo "Saisie terminée";
?>
```

Placer ce script dans **~/public_html** et appeler le **bd1.php**

Dans votre navigateur préféré, dans le champ URL saisissez :

http://obelix/~olivier/bd1.php

A priori y a pas grand chose qui s'est passé, maintenant connecter vous à votre base **essai** dans un shell

```
[olivier@obelix olivier]$ mysql -u olivier -p essai
Enter password:
```

Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10 to server version: 5.0.33

Type 'help' for help.

mysql> SELECT * FROM coord;
+-----+-------+-------------------+
| nom | prenom | email |
+-----+-------+-------------------+
| hoarau | olivier | olivier.hoarau@fnac.net |
+-----+-------+-------------------+
1 row in set (0.00 sec)

C'est bon ça fonctionne. Passons à un exemple plus pointu, on va entrer les informations concernant vos visiteurs dans une base **MySQL**, créer la table telle que décrite dans l'exemple 2 du paragraphe installation de **MySQL**, créer maintenant le script **PHP**.

```
<?

$page=getenv("HTTP_REFERER");
$ip=getenv( "REMOTE_ADDR");
$host=gethostbyaddr($ip);
$d = date("d/m/Y H:i:s");
$expl=getenv("HTTP_USER_AGENT");

$serveur="localhost";
$login="olivier";
$pass="mot-de-passe";
$base="essai";
$table="ref";

$id=MYSQL_CONNECT($serveur,$login,$pass);
mysql_select_db($base);

$query="INSERT INTO $table VALUES('$d','$host','$ip','$expl','$page')";
$result=mysql_query($query,$id);

echo "$d $host($ip) $expl $page";

?>
```

Nommez ce script **bd2.php** et placez le dans **~/public_html** . Dans votre navigateur préféré tapez dans le champ URL

http://obelix/~olivier/bd2.php

Vous devriez voir la date, le nom de votre machine avec son adresse IP et des infos sur votre OS et votre navigateur. A présent connectons nous à la base:

[olivier@obelix olivier]$ mysql -u olivier -p essai
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10 to server version: 5.0.33

Type 'help' for help.

mysql> SELECT * FROM ref;
+-------------------------+----------------------+-----------------+--------------------------+---------+
| date | host | ip | os | page |
+-------------------------+----------------------+-----------------+--------------------------+---------+
| 24/04/2000 08:34:05 | asterix.armoric.bz | 192.168.13.11 | Mozilla/4.61 [en] (X | |

+-------------------------+----------------------+-----------------+--------------------------+---------+

1 row in set (0.00 sec)

C'est bon le visiteur a bien été pris en compte.

Maintenant que vous savez comment **Apache** fonctionne avec **MySQL** et **PHP**, laissez libre cours à votre imagination.

7.2 Administration des bases MySQL avec phpMyAdmin

phpMyAdmin est un ensemble de scripts PHP qui permet d'administrer des bases **MySQL** à partir d'un navigateur. Vous pouvez le récupérer à l'URL www.phpmyadmin.net. En détail phpMyAdmin permet de:
- créer et supprimer des bases de données,
- éditer, ajouter ou supprimer des champs,
- taper des commandes SQL,
- gérer les clés de champs,
- ...

L'archive se présente sous la forme d'un tarball qu'on décompresse en tapant :

tar xvfz phpMyAdmin-2.10.0.2-all-languages-utf-8-only.tar.gz

Cela va créer dans le répertoire de travail un répertoire **phpMyAdmin-2.10.0.2-all-languages-utf-8-only**. Dans ce répertoire vous avez un fichier **config.inc.sample.php**, il faut le renommer en fichier **config.inc.php**, dans ce fichier il faut modifier les champs suivants :

Pour cette variable on peut mettre un peu n'importe quoi, ça sert ensuite pour chiffrer de manière aléatoire.

$cfg['blowfish_secret'] = 'je suis toto'; /* YOU MUST FILL IN THIS FOR COOKIE AUTH! */

pour ces deux variables on définit le compte de connexion à la base MySQL avec le mot de passe

$cfg['Servers'][$i]['controluser'] = 'olivier';
$cfg['Servers'][$i]['controlpass'] = 'mot-de-passe-en-clair';

pour le reste j'ai laissé les paramètres par défaut.

Maintenant on doit rendre accessible le répertoire **phpMyAdmin** d'une page web, pour cela deux solutions:
- (solution simple) placer **phpMyAdmin** dans **/usr/local/apache/htdocs** et au niveau de la page d'accueil d'apache faire un lien vers **/usr/local/apache/htdocs/phpMyAdmin-2.10.0.2-all-languages-utf-8-only/index.php**
- (solution préconisée), créer un hôte virtuel pointant vers **./phpMyAdmin-2.10.0.2-all-languages-utf-8-only** qu'on appelera **www.sql.breizland.bz** .

NOTE Si ça vous gène que n'importe qui d'un navigateur puisse aller dans le répertore **phpMyAdmin**, mettez y des restrictions d'accès avec un fichier **.htaccess**.

Avec la solution hôte virtuel, à partir d'un navigateur quand on sélectionne **www.sql.breizland.bz** on tombe d'abord sur une bannière de login, ensuite sur une fenêtre avec frame avec à gauche la liste des bases de données disponibles et à droite, le menu suivant:

Pour travailler sur une base de données particulières il suffit de la sélectionner dans le choix déroulant à gauche, on retrouve d'ailleurs notre base **essai**, pour en créer une autre il suffit de choisir **Bases de données** puis **Créer une base de données**.

Si on sélectionne **essai** par exemple on obtient

Vous pouvez donc créer des nouvelles tables, faire des requêtes **SQL**, etc.

8 Scripts CGI

Pour activer les scripts CGI, le fichier **httpd.conf** est déjà configuré pour, on y trouve notamment la ligne qui indique où trouver les scripts:

ScriptAlias /cgi-bin/ /usr/local/apache/cgi-bin/

Le but de l'exercice est de créer un script perl CGI qui va traiter un formulaire quelconque d'une page HTML. Vous allez créer votre script perl sous **/usr/local/apache/cgi-bin**, et le nommer **form.pl** , voici son contenu:

```
#!/usr/bin/perl
use CGI;
$html=new CGI;
print $html->header;

print "<HTML>\n";
print "<HEAD>\n";
print "<TITLE>Premier script CGI perl</TITLE>\n";
print "</HEAD>\n";
print "<BODY>\n";
print "<H1>Traitement du formulaire</H1>\n";
```

```
print "Nom :";
print $html->param('nom');
print "<p>\n";
print "Email :";
print $html->param('email');
print "<p>\n";
print "Commentaire:";
print $html->param('comment');

print "</BODY>\n";
print "</HTML>\n";
```

Donner les droits qui vont bien avec ce fichier:

chmod 755 form.pl

En tant qu'utilisateur standard (**olivier** dans notre exemple), créer maintenant le fichier HTML suivant que vous appellerez **formulaire.htm**

```
<html>
<body>
<h2>Formulaire</h2>
<form action="http://obelix/cgi-bin/form.pl" METHOD=GET>
Nom: <input type="text" name=nom size=20><br>
Email: <input type="text" name=email size=30><br>
Commentaire: <input type="text" name=comment size=100><br>
<input type=submit value="Envoyer"> <input type=reset value="remettre à zéro">
</form>
</body>
</html>
```

Voilà maintenant quand vous allez accéder à **http://obelix/~olivier/formulaire.htm** , vous allez avoir une page du style:

Haut du formulaire

Nom:

Email:

Commentaire:

Envoyer | remettre à zéro

Bas du formulaire

En appuyant sur **Envoyer** ça va déclencher l'exécution du script **CGI** perl, qui va provoquer l'affichage des valeurs précédemment saisies.

9 PHP et LDAP

On peut compiler **PHP** pour pouvoir utiliser des commandes gérant une base LDAP, pour cela au paragraphe de compilation de PHP on rajoutera aux options de **configure** l'option :

--with-ldap=/var/lib/ldap

/var/lib/ldap devant être remplacé par le chemin où se trouve votre base **LDAP**. Recompilez puis réinstallez le module qui va bien :

make

Et en tant que **root**

make install

Relancez **Apache**

/etc/rc.d/init.d/httpd restart

Voilà un petit programme qui va nous permettre de rajouter une entrée dans la base, libre à vous maintenant de créer des formulaires de saisie, de destruction, et de recherche:

```
<?
// nom du serveur LDAP
 $server="asterix";

// identification de l'administrateur de la base
 $rootdn="cn=Manager, dc=breizland, dc=bz";

//mot de passe administrateur
 $rootpw="secret";

//connexion à la base
 $result=ldap_connect($server);
 if ($result==1)
{
      ldap_bind($result,$rootdn,$rootpw);
}

//Enregistrement d'une entrée dans la base
echo"Enregistrement de Benjamin Hoarau\n";

$dn="dc=breizland, dc=bz";
$nom="Hoarau";
$prenom="Benjamin";
$mail="benjamin.hoarau@funix.org";

$info["cn"]=$nom." ".$prenom;
$info["mail"]=$mail;
$info["objectclass"]="person";

ldap_add($result,"cn=$nom $prenom,$dn",$info);
ldap_close($result);

echo "L'enregistement a réussi";
?>
```

Appelez ce fichier **ldap.php3**, vous pouvez le tester et vérifier que l'entrée a bien été saisie dans la base.

10 Sécuriser Apache et PHP

La compilation de **PHP** est identique à celle au chapitre de configuration de **PHP**, c'est au niveau du fichier de configuration qu'on modifiera certains points. Pour Apache, on désactivera par défaut tous les modules (statiques ou dynamiques), on installera le strict minimum, à vous de rajouter ensuite les modules qui sont nécessaires à votre site (ou d'adapter votre site).
Cette sécurisation est relativement basique à moins d'un trou de sécurité dans Apache ou PHP elle devrait vous mettre à l'abri de 99,99% des attaques. Pensez donc à visiter régulièrement les sites d'Apache ou de PHP pour vous tenir au courant des dernières informations en terme de sécurité, évitez aussi d'installer les dernières versions de ces logiciels, attendez qu'elles aient fait leur preuve.

On installera les modules suivants:

mod_access qui permet un contrôle d'accès (hôte, adresse IP et autres caractéristiques), il est nécessaire si on veut utiliser les options order, deny et allow entre autres.

mod_auth qui permet l'authentification des utilisateur basés sur le fichier .htaccess
mod_dir qui permet de définir l'index (index.php3, index.html, ...) d'un site
mod_log_config pour activer les logs
mod_mime qui permet de déterminer le type du fichier à partir de son extension
mod_vhost_alias pour pouvoir bénéfier des hôtes virtuels
mod_so pour pouvoir charger une biblio dynamique comme PHP

La compilation d'**Apache** commence d'abord en tapant:

make clean

Puis

./configure --prefix=/usr/local/apache --disable-module=all --enable-module=access --enable-module=log_config --enable-module=dir --enable-module=mime --enable-module=auth --enable-module=vhost_alias --enable-module=so

Puis **make** et en tant que root **make install**

Voilà le fichier de configuration **/usr/local/apache/conf/httpd.conf** réduit à sa plus simple expression

```
##
## httpd.conf -- Apache HTTP server configuration file
##

# parametres globaux

ServerType standalone
ServerRoot "/usr/local/apache"
PidFile /usr/local/apache/logs/httpd.pid
ScoreBoardFile /usr/local/apache/logs/httpd.scoreboard

# parametres de performances

Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15
MinSpareServers 5
MaxSpareServers 10
StartServers 5
MaxClients 150
MaxRequestsPerChild 0

# chargement des modules
LoadModule php5_module        libexec/libphp5.so

ClearModuleList
AddModule mod_so.c
AddModule mod_log_config.c
AddModule mod_mime.c
```

```
AddModule mod_dir.c
AddModule mod_access.c
AddModule mod_auth.c
AddModule mod_vhost_alias.c
AddModule mod_php5.c

# configuration generale

Port 80
User nobody
Group nogroup
ServerAdmin olivier@tosh.kervao.fr
DocumentRoot "/usr/local/apache/htdocs"
<IfModule mod_dir.c>
   DirectoryIndex index.php3 index.php
</IfModule>
UseCanonicalName Off
HostnameLookups Off
ServerSignature Off
ServerTokens Prod

# controle d'acces
# par défaut on interdit tout

<Directory />
   Options None
   AllowOverride None
   Order deny,allow
   Deny from all
</Directory>

# définition des droits d'acces pour les répertoires
# des hotes virtuels

<Directory "/www/homepage/online">
   php_value open_basedir /www/homepage/online
   Order allow,deny
   Allow from all
</Directory>

<Directory "/www/homepage/ohoarau">
   php_value open_basedir /www/homepage/ohoarau
   Order allow,deny
   Allow from all
</Directory>

# pour ne pas pouvoir voir les .htaccess
AccessFileName .htaccess

<Files ~ "^\.ht">
   Order allow,deny
```

```
    Deny from all
    Satisfy All
</Files>

# definition des types MIME

<IfModule mod_mime.c>
    TypesConfig /usr/local/apache/conf/mime.types
</IfModule>
DefaultType text/plain
<IfModule mod_mime.c>
    AddType application/x-tar .tgz
    AddType application/x-httpd-php .php .php3 .php4 .php5 .phtml
    AddType application/x-httpd-php-source .phps
    AddEncoding x-compress .Z
    AddEncoding x-gzip .gz .tgz
</IfModule>

# configuration des logs

ErrorLog /usr/local/apache/logs/error_log
LogLevel warn

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

CustomLog /usr/local/apache/logs/access_log combined

# définition des hôtes virtuels qui seront accessibles sur internet
# les noms de domaine auront été obtenus par des services comme DynDNS

# ici vous devez mettre votre adresse IP sur le net
NameVirtualHost 80.52.250.167

<VirtualHost funix.homelinux.org>
ServerName funix.homelinux.org
DocumentRoot /www/homepage/online
ErrorLog logs/online-error_log
TransferLog logs/online-access_log
CustomLog logs/online-access_log combined
</VirtualHost>

<VirtualHost ohoarau.homelinux.org>
ServerName ohoarau.homelinux.org
DocumentRoot /www/homepage/ohoarau
ErrorLog logs/ohoarau-errror_log
TransferLog logs/ohoarau-access_log
CustomLog logs/ohoarau-access_log combined
</VirtualHost>
```

Maintenant on va modifier certains points dans la configuration de PHP, cela se passe dans le fichier **/usr/local/apache/conf/php.ini**. On va modifier les paramètres suivants:

safe_mode=On	Les scripts PHP ne pourront modifier d'autres fichiers que si le propriétaire du dit script et aussi propriétaire des autres fichiers. En conséquence le script PHP d'un utilisateur ne pourra pas modifier les fichiers d'un autre utilisateur.
safe_mode_gid = Off	Avec safe_mode à On en mettant cette variable à Off, il faut non seulement que le propriétaire du script PHP soit aussi le propriétaire du fichier à modifier mais que le groupe (GID) soit le même également.
open_basedir=	Cette variable a été définie plus haut dans le fichier **httpd.conf**, PHP ne pourra accéder qu'aux fichiers se trouvant dans les répertoires (et sous répertoires) mentionnés.
expose_php=Off	PHP ne délivrera pas d'information en réponse aux requêtes faites par des clients.
display_errors = Off	PHP ne délivrera à l'affichage (dans le navigateur) aucune erreur et warning.
log_errors = On	Les erreurs et warnings seront inscrits dans un fichier de log
error_log = /usr/local/apache/logs/php.log	Définition du fichier de log
file_uploads = Off	Désactivation de l'upload
register_globals=Off	En mettant à On les variables d'environnement, GET, POST, etc. deviennent automatiquement des variables globales, ce qui est potentiellement dangereux, en conséquence il est fortement conseillé de désactiver ce paramètre

Pour faire propre il faudrait maintenant chrooter l'environnement de votre serveur **Apache**, supprimer les compilateurs sur votre serveur, ...

11 Analyser les logs d'Apache

11.1 Présentation

Cette page présente deux outils pour analyser les logs d'Apache. Le premier **webalizer** est sûrement le plus connu, le deuxième **awstats** est le nouveau venu dans le domaine, il gagne à être connu car il fournit une information plus riche que **webalizer**. Les deux outils ont la particularité d'afficher leurs résultats dans une page web.
Awstats fait appel à un script CGI, ce qui n'est pas le cas de **webalizer** qui pourra servir pour un site web sécurisé (CGI désactivé) accessible sur internet.

11.2 Analyser les logs d'Apache avec webalizer

11.2.1 Présentation

Webalizer est un outil qui permet de présenter de manière synthétique les logs d'**apache** sous forme de pages html avec graphique. Vous pouvez le récupérer à l'URL www.mrunix.net/webalizer/ sous la forme d'une archive tarball

webalizer-2.01-10-src.tgz . Pour info cette fonctionne ne gère plus le format GIF mais le format PNG qui le remplace pour une sombre histoire de droits, pour une utilisation de GIF rabattez vous vers la version 1.3.
Webalizer se base entre autres sur la bibliothèque **GD**.

11.2.2 Installation de GD

Dans le cas d'une mandrake on se contentera de taper

urpmi gd-devel

et on passe à l'étape suivante sinon vous pouvez toujours utiliser la méthode suivante

On récupérera **GD** qui est une bibliothèque d'outils utilisés pour faire des images dynamiques à l'adresse www.boutell.com/gd. On décompresse l'archive en tapant

tar xvfz gd-2.0.33.tar.gz

Cela va créer le répertoire **gd-2.0.33**. Sur une Mandriva, vous devez installer les packages **zlib1-devel**, **libpng3-devel**, **libjpeg62-devel**. Dans le répertoire de **GD**, taper maintenant:

./configure

Puis

make

Et en tant que root

make install

Si ce n'est déjà fait on rajoute la ligne **/usr/local/lib** dans le fichier **/etc/ld.so.conf** et on tape

ldconfig

11.2.3 Installation

Préalablement on installera le package suivant

urpmi db1-devel

Sous ubuntu il s'agit du package **libdb2-dev** et **libdb1-compat** . Dans un répertoire de travail pour décompresser l'archive, on tapera:

tar xvfz webalizer-2.01-10-src.tgz

Cela va nous créer un répertoire **webalizer-2.01-10** . Puis dans le répertoire **webalizer-2.01-10** en tant que simple utilisateur:

./configure --with-language=french --enable-dns

L'option **enable-dns** permet d'avoir le nom de la machine qui s'est connecté sur le site plutôt que son adresse IP, c'est quand même plus parlant. On tape maintenant

make

Gloups je tombe sur cette erreur

webalizer.c:70:16: db.h: No such file or directory

Dans le fichier **db.h** au lieu de

#include <db.h>

J'écris

#include <db1/db.h>

Je fais pareil pour le fichier **dns_resolv.c**. On retape **make** et enfin en tant que root

make install

Voilà le résultat

```
/usr//bin/install -c webalizer /usr/local/bin/webalizer
/usr//bin/install -c -m 644 webalizer.1 /usr/local/man/man1/webalizer.1
/usr//bin/install -c -m 644 sample.conf /etc/webalizer.conf.sample
rm -f /usr/local/bin/webazolver
ln -s /usr/local/bin/webalizer /usr/local/bin/webazolver
```

Comme toujours si les chemins ne vous plaisent pas:

configure --help

11.2.4 Configuration

Maintenant on crée le fichier de configuration en partant d'un fichier exemple qui a été installé sous **/etc**

cp /etc/webalizer.conf.sample /etc/webalizer.conf

Dans ce fichier on doit définir les paramètres suivants, les fichiers qui seront pris en compte dans les stats

```
PageType      htm*
PageType      cgi
#PageType      phtml
PageType      php3
#PageType      pl
```

Pour que vos propres requêtes ne soient pas prises en compte pour les visites (remplacez par votre nom de domaine)

HideSite   *kervao.fr

Pour que vos visites ne soient pas prises en compte pour le compte des "referers"

Pour le domaine local

HideReferrer kervao.fr

Pour les accès directs

HideReferrer Direct Request

pour cacher les liens à partir de votre site

HideReferrer www.funix.org

pour cacher les liens à partir de google

HideReferrer google

l'emplacement où se trouvera la base de donnée nécessaire à la résolution DNS

DNSCache /usr/local/apache/webalizer/dns_cache.db

On doit définir également ce paramètre, je l'ai fixé à 10 comme conseillé

DNSChildren 10

webalizer peut maintenant être appelé régulièrement avec **cron**, sur ma Mandriva, j'ai rajouté le fichier **webalizer** (avec des droits en exécution) sous **/etc/cron.daily**

```
#!/bin/bash
cd /usr/local/apache/webalizer/online
/usr/local/bin/webalizer /usr/local/apache/logs/online-access_log
cd /usr/local/apache/webalizer/ohoarau
/usr/local/bin/webalizer /usr/local/apache/logs/ohoarau-access_log
```

Les stats seront donc mis à jour tous les jours.

NOTE Si vous ne voulez pas que n'importe qui puisse lancer **webalizer** , je vous invite à mettre des droits restreints sur l'exécutable (limiter l'exécution à un utilisateur ou un groupe).

11.2.5 Utilisation

Pour lancer **webalizer** c'est très simple, placer vous dans un répertoire qui contiendra les résultats de la commande et taper:

webalizer /usr/local/apache/logs/access_log

access_log correspond au fichier de log d'**Apache** et **/usr/local/apache/logs** étant le chemin par défaut. Voilà le résultat:

```
Webalizer V2.01-10 (Linux 2.4.8-26mdk) French
Using logfile /usr/local/apache/logs/access_log (clf)
Creating output in current directory
Hostname for reports is 'asterix.kervao.fr'
History file not found...
Generating report for Juin 2002
```

Generating summary report
Saving history information...
3 records in 0.22 seconds

Si vous avez installé Apache avec les packages de la Mandrake, la commande à taper est

webalizer /var/log/httpd/access_log

Cela nous crée à l'endroit où vous avez tapé la commande les fichiers html suivants:

index.html
usage_200111.html

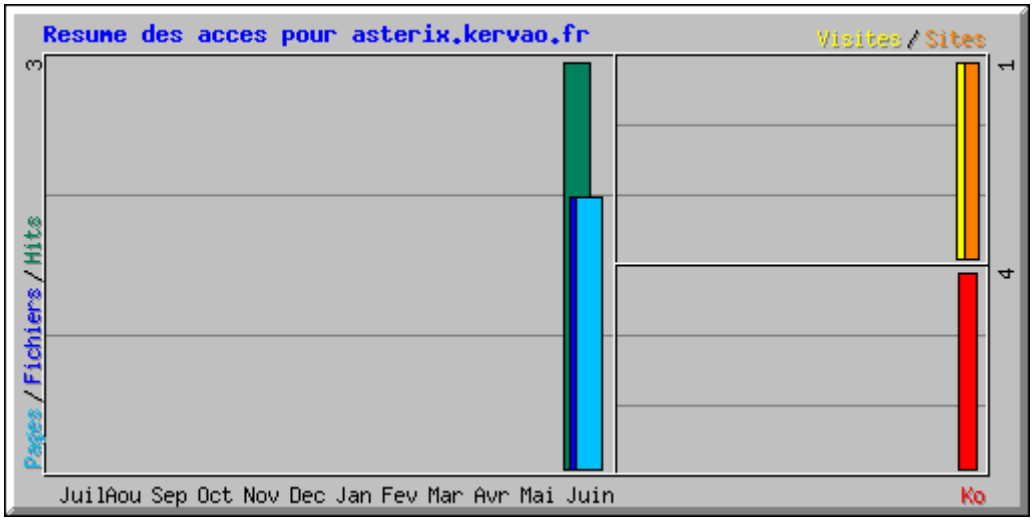
Et les graphiques suivants:

ctry_usage_200111.png
hourly_usage_200111.png
daily_usage_200111.png
usage.png

Voilà ce que ça donne

Statistiques d'accès à asterix.kervao.fr

Période du résumé: 12 derniers mois
Généré le 23-Jun-2002 07:43 TAHT



Résumé par mois

Mois	Moyenne journalière				Totaux mensuels					
	Hits	Fichiers	Pages	Visites	Sites	Ko	Visites	Pages	Fichiers	Hits
Juin 2002	3	2	2	1	1	4	1	2	2	3
Totaux						**4**	**1**	**2**	**2**	**3**

Generated by Webalizer Version 2.01

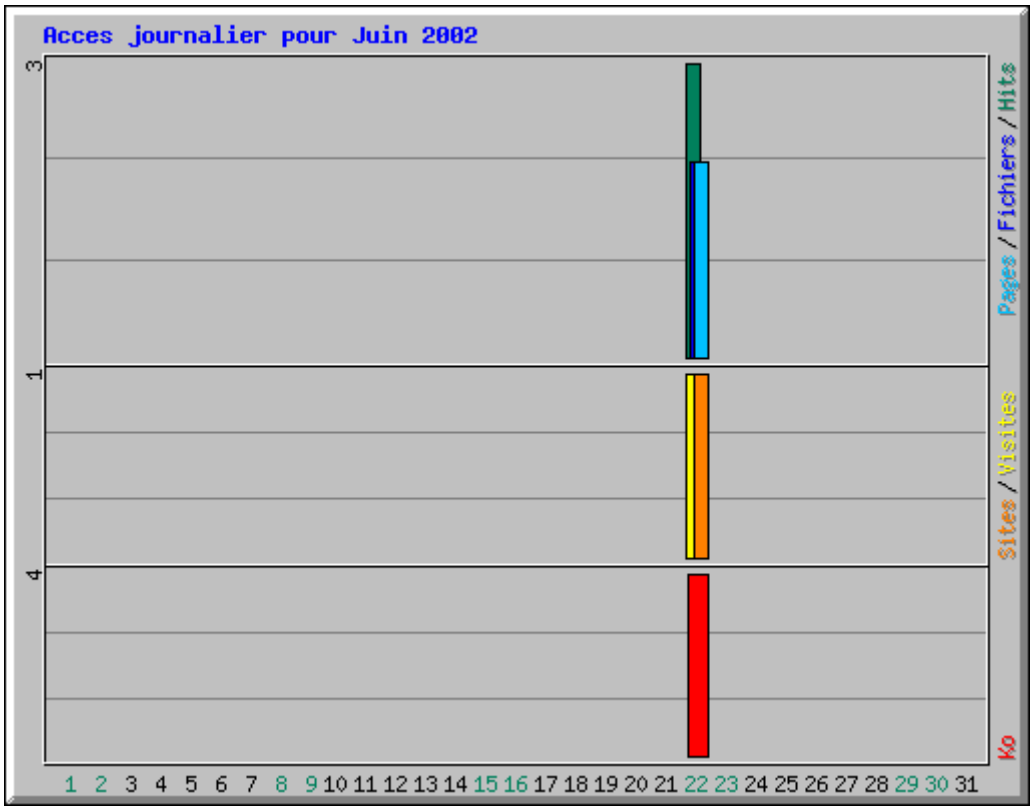
Et en cliquant sur Juin 2002 on obtient

Statistiques d'accès à asterix.kervao.fr

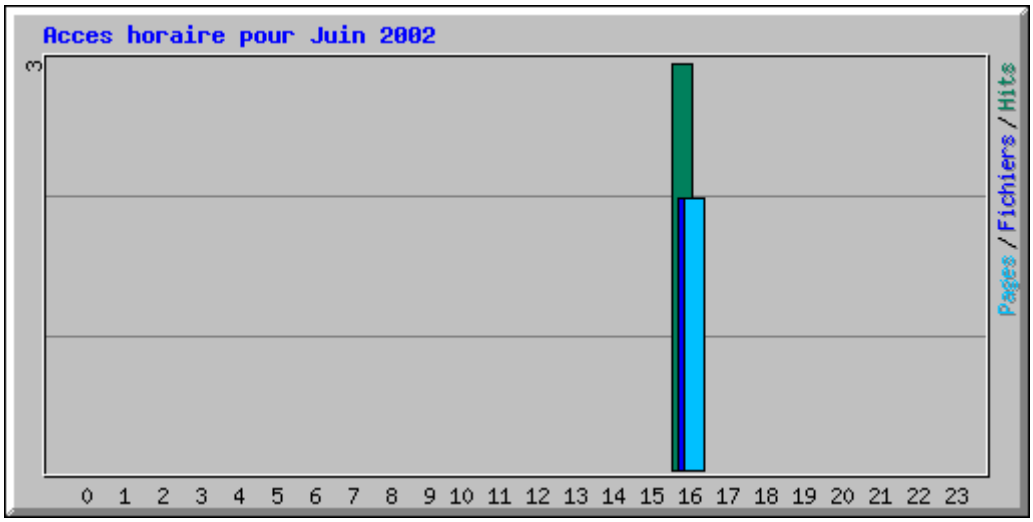
Période du résumé: Juin 2002
Généré le 23-Jun-2002 07:43 TAHT

[Statistiques journalières] [Statistiques horaires] [URLs] [Pages d'entrée] [Pages de sortie] [Sites] [Pays]

Statistiques mensuelles pour Juin 2002		
Total des Hits		**3**
Total des Fichiers		**2**
Total Pages		**2**
Total Visites		**1**
Total des Ko		**4**
Total des Sites uniques		**1**
Total des URLs uniques		**2**
	Moy.	**Max**
Hits par Heure	**0**	**3**
Hits par Jour	**3**	**3**
Fichiers par Jour	**2**	**2**
Pages par Jour	**2**	**2**
Visites par Jour	**1**	**1**
KOctets par Jour	**4**	**4**
Hits par code de réponse		
Code 200 - OK		**2**
Code 304 - Not Modified		**1**



Statistiques journalières pour Juin 2002												
Jour	Hits		Fichiers		Pages		Visites		Sites		Ko	
22	3	100.00%	2	100.00%	2	100.00%	1	100.00%	1	100.00%	4	100.00%



Statistiques horaires pour Juin 2002								
Heure	Hits		Fichiers		Pages		Ko	
	Moy.	Total	Moy.	Total	Moy.	Total	Moy.	Total

0	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
1	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
2	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
3	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
4	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
5	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
6	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
7	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
8	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
9	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
10	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
11	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
12	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
13	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
14	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
15	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
16	**3**	**3**	100.00%	**2**	**2**	100.00%	**2**	**2**	100.00%	**4**	**4**	100.00%
17	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
18	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
19	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
20	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
21	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
22	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%
23	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%	**0**	**0**	0.00%

Top 2 sur un total de 2 URLs

#	Hits		Ko		URL
1	**2**	66.67%	**1**	38.50%	/
2	**1**	33.33%	**2**	61.50%	/apache_pb.gif

Top 2 sur un total de 2 URLs par Ko

#	Hits		Ko		URL
1	**1**	33.33%	**2**	61.50%	/apache_pb.gif
2	**2**	66.67%	**1**	38.50%	/

Top 1 sur un total de 1 pages d'entrée

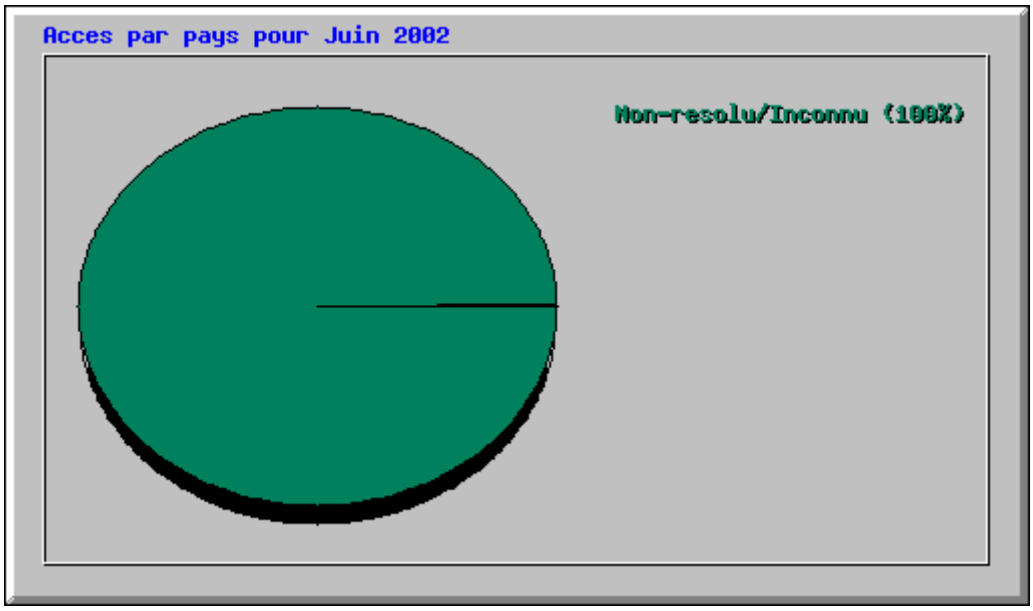
#	Hits		Visites		URL
1	2	66.67%	1	100.00%	/

Top 1 sur un total de 1 sites

#	Hits		Fichiers		Ko		Visites		Nom d'hôte
1	3	100.00%	2	100.00%	4	100.00%	1	100.00%	192.168.26.75

Top 1 sur un total de 1 sites par Ko

#	Hits		Fichiers		Ko		Visites		Nom d'hôte
1	3	100.00%	2	100.00%	4	100.00%	1	100.00%	192.168.26.75



Top 1 sur un total de 1 pays

#	Hits		Fichiers		Ko		Pays
1	3	100.00%	2	100.00%	4	100.00%	Non-resolu/Inconnu

Generated by Webalizer Version 2.01
Si ça ne vous convient pas, vous pouvez rentrer des options à **webalizer**, pour en avoir la liste:

webalizer -help

Par ailleurs vous pouvez définir une bonne fois pour toute ces options dans le fichier **webalizer.conf** qu'on placera sous **/etc**. Vous pouvez d'ailleurs trouver un **webalizer.conf.sample** sous **/etc** pour vous inspirer.

webalizer peut maintenant être appelé régulièrement avec cron, sur ma Mdk, j'ai rajouté le fichier **webalizer** (avec des droits en exécution) sous **/etc/cron.daily**

```
#!/bin/bash
cd /usr/local/apache/webalizer/online
/usr/local/bin/webalizer /usr/local/apache/logs/online-access_log
cd /usr/local/apache/webalizer/ohoarau
/usr/local/bin/webalizer /usr/local/apache/logs/ohoarau-access_log
```

Les stats seront donc mis à jour tous les jours.

NOTE Si vous ne voulez pas que n'importe qui puisse lancer **webalizer** , je vous invite à mettre des droits restreints sur l'exécutable (limiter l'exécution à un utilisateur ou un groupe).

NOTEBIS si au lancement de webalizer vous obtenez comme erreur

```
Reading history file... webalizer.hist
Error: Skipping oversized log record
Error: Skipping oversized log record
Error: Skipping oversized log record
Error: Skipping oversized log record
Error: Skipping oversized log record
Error: Skipping oversized log record
```

C'est normal c'est juste un petit malin qui a envoyé une requête hyper longue pour faire tomber votre serveur, il semblerait que IIS soit sensible à ce genre d'attaque. **Webalizer** se contente d'ignorer ce genre de stats.

11.3Analyser les logs d'Apache avec awstats

11.3.1 Présentation

Awstat est un outil pour analyser les logs d'**apache**, il est plus riche que **webalizer**, vous pouvez le trouver à l'URL awstats.sourceforge.net , c'est une archive au format tarball **awstats-6.5.tgz** qu'on récupérera.

11.3.2 Installation et configuration

On décompresse l'archive en tapant :

```
tar xvfz awstats-6.6.tgz
```

Cela va nous donner le répertoire **awstats-6.6**. Dans un premier temps on va devoir modifier légèrement le fichier de config d'**apache httpd.conf** dans le cas où vous utilisez des hôtes virtuels, voici un exemple de configuration

```
<VirtualHost 192.168.13.11:1234>
ServerName funix.homelinux.org
DocumentRoot /roger/homepage/online
ErrorLog logs/online-error_log
TransferLog logs/online-access_log
CustomLog logs/online-access_log combined
</VirtualHost>
```

On doit mettre **combined** pour le paramètre **CustomLog**. N'oubliez pas de relancer **apache** en cas de modification

/etc/rc.d/init.d/httpd restart

Maintenant on doit créer le répertoire **/etc/awstats**

mkdir /etc/awstats

Pour la configuration d'**awstats** on ira dans le répertoire **tools**

cd awstats-6.6/tools

Dans le cas d'une mise à jour on tapera

awstats_updateall.pl now -awstatsprog=/chemin-absolu/awstats-6.6/wwwroot/cgi-bin/awstats.pl -configdir= /etc/awstats/

/etc/awstats est le répertoire où se trouve les fichiers de configuration de vos serveurs. Voilà le résultat

Running '"/chemin-absolu/awstats-6.5/wwwroot/cgi-bin/awstats.pl" -update -config=funix .homelinux.org:1234 -configdir="/etc/awstats/"' to update config funix.homelinux.org:1234
Update for config "/etc/awstats/awstats.funix.homelinux.org:1234.conf"
With data in log file "/usr/local/apache/logs/online-access_log"...
Phase 1 : First bypass old records, searching new record...
Direct access after last parsed record (after line 26968)
Jumped lines in file: 26968
Found 26968 already parsed records.
Parsed lines in file: 130
Found 52 dropped records,
Found 65 corrupted records,
Found 0 old records,
Found 13 new qualified records.

Running '"/chemin-absolu/awstats-6.5/wwwroot/cgi-bin/awstats.pl" -update -config=ohoarau .homelinux.org:1234 -configdir="/etc/awstats/"' to update config ohoarau.homelinux.org:1234
Update for config "/etc/awstats/awstats.ohoarau.homelinux.org:1234.conf"
With data in log file "/usr/local/apache/logs/ohoarau-access_log"...
Phase 1 : First bypass old records, searching new record...
Direct access after last parsed record (after line 11746)
Jumped lines in file: 11746
Found 11746 already parsed records.
Parsed lines in file: 0
Found 0 dropped records,
Found 0 corrupted records,
Found 0 old records,
Found 0 new qualified records.

Running '"/chemin-absolu/awstats-6.5/wwwroot/cgi-bin/awstats.pl" -update -config=olivier.funix.org -configdir="/etc/awstats/"' to update config olivier.funix.org
Update for config "/etc/awstats/awstats.olivier.funix.org.conf"

With data in log file "/usr/local/apache/logs/olivier/olivier.funix.org.log"...
Phase 1 : First bypass old records, searching new record...
Direct access after last parsed record (after line 6034)
Jumped lines in file: 6034
Found 6034 already parsed records.
Parsed lines in file: 0
Found 0 dropped records,
Found 0 corrupted records,
Found 0 old records,
Found 0 new qualified records.

Running "'/chemin-absolu/awstats-6.5/wwwroot/cgi-bin/awstats.pl" -update -config=www.funix.org -configdir="/etc/awstats/"' to update config www.funix.org
Update for config "/etc/awstats/awstats.www.funix.org.conf"
With data in log file "/usr/local/apache/logs/funix/www.funix.org.log"...
Phase 1 : First bypass old records, searching new record...
Direct access after last parsed record (after line 13987)
Jumped lines in file: 13987
Found 13987 already parsed records.
Parsed lines in file: 0
Found 0 dropped records,
Found 0 corrupted records,
Found 0 old records,
Found 0 new qualified records

Dans le cas d'une première installation, on tapera (dans l'exemple ci-dessous, je dispose de deux serveurs Apache (version 1.3 et 2) avec les serveurs web virtuels **http://funix.homelinux.org:1234** et **http://ohoarau.homelinux.org:1234**) en tant que root

perl awstats_configure.pl

voilà le résultat

----- AWStats awstats_configure 1.0 (build 1.6) (c) Laurent Destailleur -----
This tool will help you to configure AWStats to analyze statistics for
one web server. You can try to use it to let it do all that is possible
in AWStats setup, however following the step by step manual setup
documentation (docs/index.html) is often a better idea. Above all if:
- You are not an administrator user,
- You want to analyze downloaded log files without web server,
- You want to analyze mail or ftp log files instead of web log files,
- You need to analyze load balanced servers log files,
- You want to 'understand' all possible ways to use AWStats...
Read the AWStats documentation (docs/index.html).

-----> Running OS detected: Linux, BSD or Unix
Warning: AWStats standard directory on Linux OS is '/usr/local/awstats'.
If you want to use standard directory, you should first move all content
of AWStats distribution from current directory:
/usr/local/linux/web/awstats-6.6
to standard directory:
/usr/local/awstats

And then, run configure.pl from this location.
Do you want to continue setup from this NON standard directory [yN] ? y

ici il trouve les fichiers de configuration de mes deux serveurs **Apache**

-----> Check for web server install
Found Web server Apache config file '/usr/local/apache/conf/httpd.conf'
Found Web server Apache config file '/usr/local/apache2/conf/httpd.conf'

Il modifie les deux fichiers de configuration, il modifie également les fichiers pour mettre à jour le format d'archivage des logs

-----> Check and complete web server config file '/usr/local/apache2/conf/httpd.conf'
Add '<Directory>' directive
AWStats directives added to Apache config file.

-----> Check and complete web server config file '/usr/local/apache/conf/httpd.conf'
Warning: You Apache config file contains directives to write 'common' log files
This means that some features can't work (os, browsers and keywords detection).
Do you want me to setup Apache to write 'combined' log files [y/N] ? y
Add 'Alias /awstatsclasses "/usr/local/linux/web/awstats-6.6/wwwroot/classes/"'
Add 'Alias /awstatscss "/usr/local/linux/web/awstats-6.6/wwwroot/css/"'
Add 'Alias /awstatsicons "/usr/local/linux/web/awstats-6.6/wwwroot/icon/"'
Add 'ScriptAlias /awstats/ "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/"'
Add '<Directory>' directive
AWStats directives added to Apache config file.

-----> Update model config file '/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.model.conf'
File awstats.model.conf updated.

Il met à jour son propre fichier de configuration

-----> Update model config file '/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.model.conf'
File awstats.model.conf updated.

-----> Need to create a new config file ?
Do you want me to build a new AWStats config/profile
file (required if first install) [y/N] ?

on indique ici l'URL du serveur web à analyser (adresse:port)

-----> Define config file name to create
What is the name of your web site or profile analysis ?
Example: www.mysite.com
Example: demo
Your web site, virtual server or profile name:
> funix.homelinux.org:1234

Définition du chemin pour stocker les fichiers de configuration pour chaque serveur à analyser

-----> Define config file path
In which directory do you plan to store your config file(s) ?

Default: /etc/awstats
Directory path to store config file(s) (Enter for default):
>

Création du fichier de configuration du serveur mentionné plus haut

-----> Create config file '/etc/awstats/awstats.funix.homelinux.org:1234.conf'
Config file /etc/awstats/awstats.funix.homelinux.org:1234.conf created.

Apache est relancé automatiquement

-----> Restart Web server with '/sbin/service httpd restart'
/etc/init.d/httpd restart: httpd restarted

-----> Add update process inside a scheduler
Sorry, configure.pl does not support automatic add to cron yet.
You can do it manually by adding the following command to your cron:
/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl -update -config=funix.homelinux.org:1234
Or if you have several config files and prefer having only one command:
/usr/local/linux/web/awstats-6.6/tools/awstats_updateall.pl now
Press ENTER to continue...

A SIMPLE config file has been created: /etc/awstats/awstats.funix.homelinux.org:1234.conf
You should have a look inside to check and change manually main parameters.
You can then manually update your statistics for 'funix.homelinux.org:1234' with command:
> perl awstats.pl -update -config=funix.homelinux.org:1234
You can also read your statistics for 'funix.homelinux.org:1234' with URL:
> http://localhost/awstats/awstats.pl?config=funix.homelinux.org:1234

Press ENTER to finish...

Il a créé un fichier de configuration pour le site concerné **/etc/awstats/awstats.funix.homelinux.org:1234.conf** dans ce fichier j'ai modifié les paramètres suivants:

- Celui qui indique le chemin du fichier de log brut à analyser

LogFile="/usr/local/apache/logs/online-access_log"

- pour activer la résolution de nom DNS

DNSLookup=1

- Celui qui contiendra les données obtenues

DirData="/usr/local/var/awstats"

- Pour indiquer le fichier index de votre site

DefaultFile="index.php"

- Pour ignorer les marchines de votre domaine

SkipHosts="127.0.0.1 REGEX[^192\.168\.13\.] REGEX[\.kervao\.fr$]"

- Si votre site est identifié par une autre URL il faut la rajouter ici (mettre un espace comme champ séparateur)

HostAliases="funix.homelinux.org:1234"

Pensez à créer préalablement ce répertoire

mkdir /usr/local/var/awstats

Pour voir si tout marche on se place maintenant en tant que root sous **awstats-6.6/wwwroot/cgi-bin** et on tape

./awstats.pl -config=funix.homelinux.org:1234 -update

Voilà le résultat

Update for config "/etc/awstats/awstats.funix.homelinux.org:1234.conf"
With data in log file "/usr/local/apache/logs/online-access_log"...
Phase 1 : First bypass old records, searching new record...
Searching new records from beginning of log file...
Phase 2 : Now process new records (Flush history on disk after 20000 hosts)...
Jumped lines in file: 0
Parsed lines in file: 4255
Found 4 dropped records,
Found 6 corrupted records,
Found 0 old records,
Found 4245 new qualified records.

Sur les versions précédentes il y avait un léger bug, voilà une ligne typique de log

192.168.0.11 - - [19/Feb/2004:10:58:17 +0100] "GET /images/journal.gif HTTP/1.1" 304 - "http://funix.homelinux.org:1234/fr/linux/main-linux.php?ref=tips&page=menu" "Mozilla/5.0 (X11; U; Linux i686; fr-FR; rv:1.4) Gecko/20
030630"

Or parfois **Apache** archive ceci

192.168.0.11 - - [19/Feb/2004:10:58:24 +0100] "GET / HTTP/1.1" 200 17493

A défaut d'information il laisse des champs vides et **awstats** n'apprécie guère (il n'analyse pas les logs...). Je n'ai pas le recul suffisant pour savoir si ça était corrigé avec la 6.6.

Si vous avez un deuxième site virtuel, créer un autre fichier **/etc/awstats/awstats.mon-domaine-virtuel.conf** sur le même modèle que précédemment. Dans mon cas il suffit de taper

cp /etc/awstats/awstats.funix.homelinux.org:1234.conf /etc/awstats/awstats.ohoarau.homelinux.org:1234.conf

Dans ce fichier pensez à modifier les paramètres pour l'URL et le fichier de logs d'Apache à analyser.

A noter que vu la méthode employée (voir chapitre suivant) pour utiliser **awstats**, j'ai supprimé les lignes qu'**awstats** à rajouter dans mes fichiers de configuration d'**Apache**.
Si vous voulez utiliser **awstats** en tant que cgi-bin accessible d'**Apache**, vous devez laisser ces lignes.

11.3.3 Utilisation

Voilà comment j'utilise **awstats**, j'ai créé un fichier **awstats** (droit exécutable) sous **/etc/cron.daily** qui s'exécute donc tous les jours et permet de réaliser les stats pour mes deux adresses web virtuelles

```
#!/bin/bash
/usr/local/linux/web/awstats-6.6/tools/awstats_buildstaticpages.pl -config=ohoarau.homelinux.org:1234 -update -lang=fr -dir=/usr/local/apache2/htdocs/awstats/ohoarau -awstatsprog=/usr/local/linux/web/awstats-6.5/wwwroot/cgi-bin/awstats.pl

/usr/local/linux/web/awstats-6.6/tools/awstats_buildstaticpages.pl -config=funix.homelinux.org:1234 -update -lang=fr -dir=/usr/local/apache2/htdocs/awstats/online -awstatsprog=/usr/local/linux/web/awstats-6.5/wwwroot/cgi-bin/awstats.pl
```

L'option **-dir** permet de sauvegarder sous forme html les statistiques obtenus. Pour terminer le répertoire icon d'**awstats** doit être accessible sur votre serveur web, pour cela dans le fichier de configuration d'**Apache**, j'ai rajouté les directives suivantes

```
Alias /awstatsicons/ "/usr/local/linux/web/awstats-6.6/wwwroot/icon/"

<Directory "/usr/local/linux/web/awstats-6.6/wwwroot/icon">
   Options Indexes MultiViews
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>
```

Vous remplacerez bien sûr par le chemin en absolu d'**awstats**. Par ailleurs j'ai mis les droits qui allaient bien pour le répertoire d'**awstats**

```
chmod 755 awstats-6.6/
```

Voilà ce que ça donne dans je lance mon script **/etc/cron.daily/awstats**

```
Launch update process : "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau.homelinux.org:1234 -update -configdir=
Build main page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output
Build alldomains page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=alldomains
Build allhosts page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=allhosts
Build lasthosts page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=lasthosts
Build unknownip page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=unknownip
Build allrobots page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=allrobots
Build lastrobots page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau
```

```
.homelinux.org:1234 -staticlinks -lang=fr -output=lastrobots
Build session page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau
.homelinux.org:1234 -staticlinks -lang=fr -output=session
Build urldetail page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau
.homelinux.org:1234 -staticlinks -lang=fr -output=urldetail
Build urlentry page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau
.homelinux.org:1234 -staticlinks -lang=fr -output=urlentry
Build urlexit page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau
.homelinux.org:1234 -staticlinks -lang=fr -output=urlexit
Build osdetail page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau
.homelinux.org:1234 -staticlinks -lang=fr -output=osdetail
Build unknownos page: "/usr/local/linux/web/awstats-6.5/wwwroot/cgi-bin/awstats.pl" -config=ohoarau
.homelinux.org:1234 -staticlinks -lang=fr -output=unknownos
Build browserdetail page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau
.homelinux.org:1234 -staticlinks -lang=fr -output=browserdetail
Build unknownbrowser page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl"
-config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=unknownbrowser
Build refererse page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau
.homelinux.org:1234 -staticlinks -lang=fr -output=refererse
Build refererpages page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau
.homelinux.org:1234 -staticlinks -lang=fr -output=refererpages
Build keyphrases page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau
.homelinux.org:1234 -staticlinks -lang=fr -output=keyphrases
Build keywords page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau
.homelinux.org:1234 -staticlinks -lang=fr -output=keywords
Build errors404 page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=ohoarau
.homelinux.org:1234 -staticlinks -lang=fr -output=errors404
20 files built.
Main HTML page is 'awstats.ohoarau.homelinux.org:1234.html'.
Launch update process : "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix
.homelinux.org:1234 -update -configdir=
Build main page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix
.homelinux.org:1234 -staticlinks -lang=fr -output
Build alldomains page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix
.homelinux.org:1234 -staticlinks -lang=fr -output=alldomains
Build allhosts page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix
.homelinux.org:1234 -staticlinks -lang=fr -output=allhosts
Build lasthosts page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix
.homelinux.org:1234 -staticlinks -lang=fr -output=lasthosts
Build unknownip page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix
.homelinux.org:1234 -staticlinks -lang=fr -output=unknownip
Build allrobots page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix
.homelinux.org:1234 -staticlinks -lang=fr -output=allrobots
Build lastrobots page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix
.homelinux.org:1234 -staticlinks -lang=fr -output=lastrobots
Build session page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix
.homelinux.org:1234 -staticlinks -lang=fr -output=session
Build urldetail page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix
.homelinux.org:1234 -staticlinks -lang=fr -output=urldetail
Build urlentry page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix
.homelinux.org:1234 -staticlinks -lang=fr -output=urlentry
Build urlexit page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix
```

.homelinux.org:1234 -staticlinks -lang=fr -output=urlexit
Build osdetail page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix .homelinux.org:1234 -staticlinks -lang=fr -output=osdetail
Build unknownos page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix .homelinux.org:1234 -staticlinks -lang=fr -output=unknownos
Build browserdetail page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix .homelinux.org:1234 -staticlinks -lang=fr -output=browserdetail
Build unknownbrowser page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix .homelinux.org:1234 -staticlinks -lang=fr -output=unknownbrowser
Build refererse page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix .homelinux.org:1234 -staticlinks -lang=fr -output=refererse
Build refererpages page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix .homelinux.org:1234 -staticlinks -lang=fr -output=refererpages
Build keyphrases page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix .homelinux.org:1234 -staticlinks -lang=fr -output=keyphrases
Build keywords page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix .homelinux.org:1234 -staticlinks -lang=fr -output=keywords
Build errors404 page: "/usr/local/linux/web/awstats-6.6/wwwroot/cgi-bin/awstats.pl" -config=funix .homelinux.org:1234 -staticlinks -lang=fr -output=errors404
20 files built.
Main HTML page is 'awstats.funix.homelinux.org:1234.html'.

Maintenant sous les répertoires **/usr/local/apache2/htdocs/awstats/online** et **/usr/local/apache2/htdocs/awstats/ohoarau** je me retrouve avec des fichiers html, pour le premier le fichier d'entrée est **awstats.funix.homelinux.org:1234.html** et pour l'autre **awstats.ohoarau.homelinux.org:1234.html**
A partir d'une page web accessible sur votre serveur web (page d'accueil par exemple), j'ai fait un lien vers ces pages de statistiques.
Il faudra éventuellement rajouter le répertoire **/usr/local/apache2/htdocs/awstats/online** dans une directive **Directory** dans le fichier **httpd.conf** si nécessaire.

Voilà quelques aperçus (non exhaustifs)

Dernière mise à jour:	06 Jan 2005 - 04:03
Période d'analyse:	Mois Jan 2005

Quand:	Historique mensuel Jours du mois Jours de la semaine Heures
Qui:	Pays Liste complète Hôtes Liste complète Dernière visite Adresses IP non résolues Visiteurs Robots/Spiders Liste complète Dernière visite
Navigation:	Durée des visites Types de fichiers Pages vues Liste complète Entrée Sortie Systèmes exploitation Versions Inconnu Navigateurs Versions Inconnu
Origine/Referer:	Origine de la connexion Moteurs de recherche Sites référenceurs Recherche Phrases clés Mots clés
Autres:	Divers Codes Status HTTP Pages non trouvées

Résumé

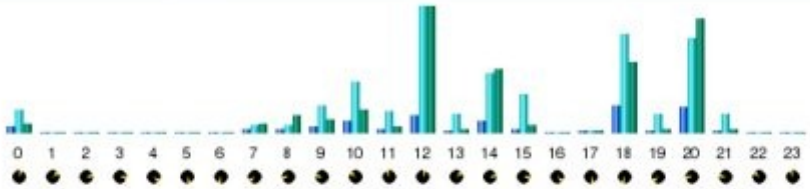
Période d'analyse	Mois Jan 2005
Première visite	03 Jan 2005 - 10:09
Dernière visite	06 Jan 2005 - 00:23

	Visiteurs différents	Visites	Pages	Hits	Bande passante
Trafic 'vu' *	**18**	**26** (1.44 visites/visiteur)	**66** (2.53 pages/visite)	**318** (12.23 hits/visite)	**3.33 Mo** (130.97 Ko/visite)
Trafic 'non vu' *			**189**	**189**	**2.51 Mo**

* Le trafic 'non vu' est le trafic généré par les robots, vers ou réponses HTTP avec code retour spécial.

Historique mensuel

Heures

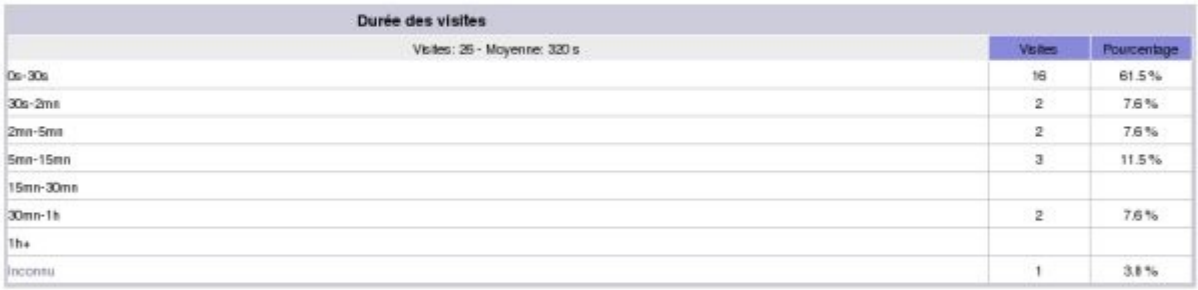


Heures	Pages	Hits	Bande passante	Heures	Pages	Hits	Bande passante
00	3	12	60.92 Ko	12	9	65	908.60 Ko
01	0	0	0	13	1	10	26.80 Ko
02	0	0	0	14	6	31	458.79 Ko
03	0	0	0	15	2	20	53.60 Ko
04	0	0	0	16	0	0	0
05	0	0	0	17	1	1	12.58 Ko
06	0	0	0	18	14	51	512.46 Ko
07	2	4	62.63 Ko	19	1	10	26.80 Ko
08	2	4	127.76 Ko	20	13	49	820.47 Ko
09	3	14	93.82 Ko	21	1	10	26.80 Ko
10	6	26	169.44 Ko	22	0	0	0
11	2	11	43.88 Ko	23	0	0	0

Visiteurs Robots/Spiders (Top 10) - Liste complète - Dernière visite

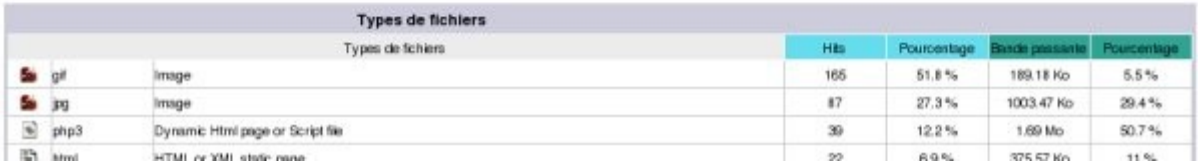
3 robots différents*	Hits	Bande passante	Dernière visite
Inktomi Slurp	92	2.43 Mo	05 Jan 2005 - 13:16
Googlebot	3	51.10 Ko	05 Jan 2005 - 09:08
MSNBot	1	17.03 Ko	05 Jan 2005 - 06:42

* Les robots présentés ici sont à l'origine de hits ou de trafic "non vus" par les visiteurs donc non représentés dans les autres tableaux.



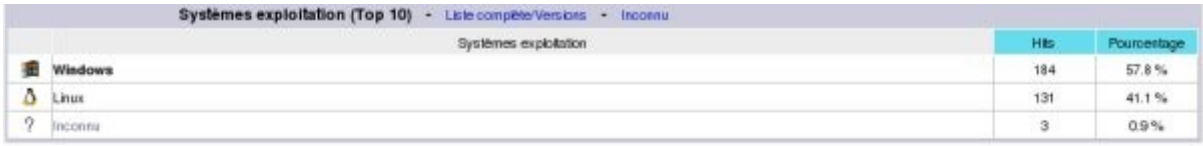
Durée des visites

Visites: 26 - Moyenne: 320 s	Visites	Pourcentage
0s-30s	16	61.5 %
30s-2mn	2	7.6 %
2mn-5mn	2	7.6 %
5mn-15mn	3	11.5 %
15mn-30mn		
30mn-1h	2	7.6 %
1h+		
Inconnu	1	3.8 %



Types de fichiers

Types de fichiers		Hits	Pourcentage	Bande passante	Pourcentage
gif	Image	165	51.8 %	189.18 Ko	5.5 %
jpg	Image	87	27.3 %	1003.47 Ko	29.4 %
php3	Dynamic Html page or Script file	39	12.2 %	1.69 Mo	50.7 %
html	HTML or XML static page	22	6.9 %	375.57 Ko	11 %



Systèmes exploitation (Top 10) - Liste complète/Versions - Inconnu

Systèmes exploitation	Hits	Pourcentage
Windows	184	57.8 %
Linux	131	41.1 %
Inconnu	3	0.9 %

Navigateurs (Top 10) - Liste complète/Versions - Inconnu

Navigateurs	Aspirateur	Hits	Pourcentage
MS Internet Explorer	Non	124	38.9 %
Mozilla	Non	109	34.2 %
Firefox	Non	71	22.3 %
Galeon	Non	10	3.1 %
W3C HTML Validator	Non	2	0.6 %
Konqueror	Non	1	0.3 %
Inconnu	?	1	0.3 %

Connexions au site par

Origine de la connexion	Pages	Pourcentage	Hits	Pourcentage
Adresse directe / Bookmarks	23	39.6 %	23	9.4 %
Lien depuis un NewsGroup				
Lien depuis un moteur de recherche Internet - Liste complète - Google 2 2 - Yahoo 2 2	4	6.8 %	4	1.6 %
Lien depuis une page externe (autres sites, hors moteurs) - Liste complète - http://funix.tomeilinux.org/fr/linux/main-linux.php3 13 80	31	53.4 %	217	88.9 %

11.4Analysez les logs d'un site hébergé non localement

Pour être concret, j'ai mon site funix.org qui est hébergé chez online, j'ai moyen de récupérer les logs d'**Apache** bruts, chaque jour un fichier log est créé et qui a pour syntaxe **www.funix.org_AAAAmmjj.log.gz** (AAAA année, mm mois, jj jour), ce fichier se trouve dans le répertoire **log_apache** qui se trouve au niveau de la racine. Tant que

la taille max n'est pas atteinte, les fichiers logs sont créés, c'est à moi de les supprimer localement. Si avec **cron**, je demande une mise à jour des stats quotidienne, je le demande pour la veille.
Voilà le script qui permet de récupérer les stats de la veille, et qui le place dans le répertoire **/usr/local/apache/logs/funix**

```
#!/bin/bash
# construction du format de la date
jour=`date +%e`
mois31="janvier mars mai juillet août octobre décembre"
mois30="avril juin septembre novembre"
if  [ $jour -eq "1" ];
then
  mois=`date +%B`
  # ca c'est le mois de février, à modifier éventuellement
  hier="28"
  for mois31
  do
     if [ $mois== "[$mois31]"]
     then
         hier=31
     fi
  done
  for mois30
  do
     if [ $mois== "[$mois30]"]
     then
          hier=30
     fi
  done
else
  hier=`expr $jour - 1`
fi
longueur=`expr length $hier`
if [ $longueur -eq "1" ];
then
           hier=0$hier
fi
anneemois=`date +%G%m`
veille=$anneemois$hier
# je me déplace dans le répertoire qui va bien et je supprime les vieux fichiers log
cd /usr/local/apache/logs/funix
rm -f /usr/local/apache/logs/funix/*.log
# récupération des logs par ftp
wget -c ftp://privftp.pro.proxad.net/log_apache/www.funix.org_$veille.log.gz --user=login --
password=motdepasse
# je décompresse l'archive et la renomme
gunzip www.funix.org_$veille.log.gz
mv www.funix.org_$veille.log www.funix.org.log
# pour ne pas avoir à  faire le ménage des fichiers logs sur le serveur ftp, on le réalise automatiquement
ftp -n <<lafin
open privftp.pro.proxad.net
user login motdepasse
```

cd log_apache
binary
delete www.funix.org_$veille.log.gz
bye
lafin

 # je lance l'analyse des logs par webalizer
 # les résultats se retrouveront dans /usr/local/apache/logs/funix
/usr/local/bin/webalizer /usr/local/apache/logs/funix/www.funix.org.log

Pour réaliser des stats avec **awstats**, j'ai créé un fichier **/etc/awstats/awstats.www.funix.org** dans lequel j'ai modifié les paramètres suivants

LogFile="/usr/local/apache/logs/funix/www.funix.org.log"

SiteDomain="www.funix.org"

HostAliases="www.funix.org"

Voilà la commande à rajouter à la fin du script pour lancer une analyse par **awstats**

/usr/local/linux/web/awstats-6.5/tools/awstats_buildstaticpages.pl -config=www.funix.org -update -lang=fr -dir=/usr/local/apache2/htdocs/awstats/funix -awstatsprog=/usr/local/linux/web/awstats-6.5/wwwroot/cgi-bin/awstats.pl

Les fichiers résultats vont se retrouver sous **/usr/local/apache2/htdocs/awstats/funix**

Placer maintenant ce script sous **/etc/cron.daily** pour qu'il soit appelé tous les jours (à 4h du mat pour une Mandrake par défaut).

12Installer un moteur de recherche avec Ht://dig

12.1Présentation

Ht://Dig est un moteur de recherche performant pour votre site intranet, il peut mettre en index vos sites web internes mais aussi ceux se trouvant sur le net. Il marche sur plusieurs plate formes, dont évidemment linux.
Dans la suite de ma page je présente un site intranet ayant pour adresse **www.funix.kervao.fr** et tournant sous Apache. On va réaliser un moteur de recherche pour ce site à l'aide de **Ht://Dig** accessible à partir de ce même site. La configuration par défaut est très satisfaisante, vous pouvez néanmoins consulter la doc livrée avec le package qui détaille toutes les subtilités de la configuration.

La dernière version stable est la 3.1.6 qu'on peut trouver sur le site officiel de **Ht://Dig**, la Mandrake 8.2 et la 9.0 fournissent la version 3.2.0 qui est une version instable. Cette page présente la première version.

12.2Installation avec tarball

On récupérera les sources à l'adresse http://www.htdig.org, elle se présente sous la forme d'un tarball **htdig-3.1.6.tar.gz** de 2Mo, qu'on décompressera en tapant:

tar xvfz htdig-3.1.6.tar.gz

Cela va créer un répertoire **htdig-3.1.6** dans lequel on tapera **configure** avec les options suivantes :

./configure --prefix=/chemin-d-install --with-cgi-bin-dir=chemin-cgi-apache --with-image-dir=répertoire-contenant-image --with-search-dir=répertoire-contenant-script-recherche

Voilà la commande que j'ai tapée chez moi après avoir créé préalablement le répertoire **/usr/local/apache/htdocs/htdig** :

./configure --prefix=/usr/local/htdig --with-cgi-bin-dir=/usr/local/apache/cgi-bin --with-image-dir=/usr/local/apache/htdocs/htdig --with-search-dir=/usr/local/apache/htdocs/htdig

Je ne passais pas cette étape avec la Mandrake 7.1 pour une sombre histoire de headers C++. Puis

make

Et enfin en tant que root

make install

Cela va mettre en place un fichier **htsearch** dans le répertoire **cgi-bin** d'**apache** et rajoutez un script de recherche sous le répertoire **htdocs/htdig** d'**apache** dont on verra l'utilité plus tard. Les binaires sont sous **/usr/local/htdig/bi**n, le fichier de conf sous **/usr/local/htdig/conf** et les bases de données sous **/usr/local/htdig/db**.

12.3Installation avec les rpm de la Mandriva

Vous devez installer les packages **htdig** et **htdig-web**. Les fichiers suivants vont être installés:
- **htsearch** et **qtest** sous **/var/www/cgi-bin**
- les exécutables (commençant par ht) sous **/usr/bin**
- le fichier de conf **htdig.conf** sous **/etc/htdig**
- des fichiers html pour l'affichage des résultats sous **/usr/share/htdig**
- les bases de données vont se trouver sous **/var/lib/htdig**
-un fichier **htdig** sous **/var/www/html** qui est en fait un lien vers le répertoire **/usr/share/htdi**

12.4Configuration

On trouvera le fichier de configuration sous **/usr/local/htdig/conf** (**/etc/htdig** pour une installation par rpm), il se nomme **htdig.conf**, voici les lignes à modifier éventuellement

```
# définition de l'emplacement où se trouveront les bases de données sur le site intranet
# attention, elles peuvent être assez grosses, pour info pour mon site intranet contenant mes pages
# www.funix.org les bases font un total de 12Mo !
# avec une install par rpm le chemin est /var/lib/htdig
database_dir:           /usr/local/htdig/db

# Définition de l'adresse à partir de laquelle htdig doit construire ses bases de données
# mettez ici l'URL de votre intranet
# vous pouvez éventuellement mettre l'URL de n'importe quel site sur internet
start_url:              http://www.funix.kervao.fr/

# votre site peut contenir des liens vers des sites extérieurs, cette variable permet de
# limiter la rechercher à des pages de votre domaine
limit_urls_to:)         ${start_url}

# Définition des pages à ne pas indexer, ce sont donc celles qui sont dans
# http://www.funix.kervao.fr/cgi-bin par exemple ici
exclude_urls:%)         /cgi-bin/ .cgi


# fichiers qui seront ignorés pendant l'indexation
bad_extensions:         .wav .gz .z .sit .au .zip .tar .hqx .exe .com .gif \
                .jpg .jpeg .aiff .class .map .ram .tgz .bin .rpm .mpg .mov .avi
```

```
# adresse email de l'administrateur
maintainer:            olivier@kervao.fr

# Par défaut un extrait de chaque page est archivé dans la base, vous pouvez limiter
# évidemment cette taille
max_head_length:       10000

# si vous récupérez des pages sur internet (indexation de sites sur internet) vous pouvez aussi
# limiter la taille des pages à récupérer
max_doc_size:%)        200000

#
# Most people expect some sort of excerpt in results. By default, if the
# search words aren't found in context in the stored excerpt, htsearch shows
# the text defined in the no_excerpt_text attribute:
# (None of the search words were found in the top of this document.)
# This attribute instead will show the top of the excerpt.
#
no_excerpt_show_top:    true

#
# Depending on your needs, you might want to enable some of the fuzzy search
# algorithms.  There are several to choose from and you can use them in any
# combination you feel comfortable with.  Each algorithm will get a weight
# assigned to it so that in combinations of algorithms, certain algorithms get
# preference over others.  Note that the weights only affect the ranking of
# the results, not the actual searching.
# The available algorithms are:
#       exact
#       endings
#       metaphone
#       prefix
#       soundex
#       synonyms
# By default only the "exact" algorithm is used with weight 1.
# Note that if you are going to use the endings, metaphone, soundex,
# or synonyms algorithms, you will need to run htfuzzy to generate
# the databases they use.
#
search_algorithm:       exact:1 synonyms:0.5 endings:0.1
#
# The following are the templates used in the builtin search results
# The default is to use compiled versions of these files, which produces
# slightly faster results. However, uncommenting these lines makes it
# very easy to change the format of search results.
# See <http://www.htdig.org/hts_templates.html for more details.
#
# template_map: Long long ${common_dir}/long.html \
#               Short short ${common_dir}/short.html
# template_name: long

#
# The following are used to change the text for the page index.
# The defaults are just boring text numbers.  These images spice
# up the result pages quite a bit.  (Feel free to do whatever, though)
```

```
#
next_page_text:         <img src="/htdig/buttonr.gif" border="0" align="middle"
width="30" height="30" alt="next">
no_next_page_text:
prev_page_text:         <img src="/htdig/buttonl.gif" border="0" align="middle"
width="30" height="30" alt="prev">
no_prev_page_text:
page_number_text:       '<img src="/htdig/button1.gif" border="0" align="middle" width="30"
height="30" alt="1">' \
                '<img src="/htdig/button2.gif" border="0" align="middle" width="30" height="30"
alt="2">' \
                '<img src="/htdig/button3.gif" border="0" align="middle" width="30" height="30"
alt="3">' \
                '<img src="/htdig/button4.gif" border="0" align="middle" width="30" height="30"
alt="4">' \
                '<img src="/htdig/button5.gif" border="0" align="middle" width="30" height="30"
alt="5">' \
                '<img src="/htdig/button6.gif" border="0" align="middle" width="30" height="30"
alt="6">' \
                '<img src="/htdig/button7.gif" border="0" align="middle" width="30" height="30"
alt="7">' \
                '<img src="/htdig/button8.gif" border="0" align="middle" width="30" height="30"
alt="8">' \
                '<img src="/htdig/button9.gif" border="0" align="middle" width="30" height="30"
alt="9">' \
                '<img src="/htdig/button10.gif" border="0" align="middle" width="30" height="30"
alt="10">'
# To make the current page stand out, we will put a border arround the
# image for that page.
#
no_page_number_text:    '<img src="/htdig/button1.gif" border="2" align="middle" width="30"
height="30" alt="1">' \
                '<img src="/htdig/button2.gif" border="2" align="middle" width="30" height="30"
alt="2">' \
                '<img src="/htdig/button3.gif" border="2" align="middle" width="30" height="30"
alt="3">' \
                '<img src="/htdig/button4.gif" border="2" align="middle" width="30" height="30"
alt="4">' \
                '<img src="/htdig/button5.gif" border="2" align="middle" width="30" height="30"
alt="5">' \
                '<img src="/htdig/button6.gif" border="2" align="middle" width="30" height="30"
alt="6">' \
                '<img src="/htdig/button7.gif" border="2" align="middle" width="30" height="30"
alt="7">' \
                '<img src="/htdig/button8.gif" border="2" align="middle" width="30" height="30"
alt="8">' \
                '<img src="/htdig/button9.gif" border="2" align="middle" width="30" height="30"
alt="9">' \
                '<img src="/htdig/button10.gif" border="2" align="middle" width="30" height="30"
alt="10">'
```

 # local variables:
 # mode: text
 # eval: (if (eq window-system 'x) (progn (setq font-lock-keywords (list '("^#.*" . font-lock-keyword-face) '("^[a-zA-Z][^ :]+" . font-lock-function-name-face) '("[+$]*:" . font-lock-comment-face))) (font-lock-mode)))

 # end:

A noter qu'avec une installation par rpm le fichier est automatiquement renseigné d'après le fichier de configuration d'**Apache**. On trouvera en fin de fichier

 # Automatically set up by htdig RPM, from your current Apache httpd.conf...
 # Verify and configure these, and set maintainer above, before running
 # /usr/bin/rundig.
 # See /usr/doc/htdig*/attrs.html for descriptions of attributes.

 # The URL(s) where htdig will start. See also limit_urls_to above.
start_url: http://asterix.kervao.fr

 # This makes sure that we don't spider the web
local_urls_only: true

 # These attributes allow indexing server via local filesystem rather than HTTP.
local_urls: http://asterix.kervao.fr/=/var/www/html/
local_user_urls: http://asterix.kervao.fr/=/home/,/public_html/

A modifier selon votre convenance. On va maintenant explorer le site web et récupérer les pages en tapant

/usr/local/htdig/bin/htdig -ivs

Note : Taper **/usr/bin/htdig -ivs** pour une install par rpm

Ne vous inquiétez pas si votre site est écrit en PHP, ça marche aussi très bien.
L'option **-i** permet d'effacer les recherches précédentes et de repartir de zéro, l'option **-v** est l'option "verbeuse". On crée à présent l'index en tapant :

/usr/local/htdig/bin/htmerge -vs

Pour info avec l'option **-m** les tables sont simplement mises à niveau.

Pour une installation avec rpm la syntaxe est un peu différente, vous devez taper:

/usr/bin/htmerge -v -m /etc/htdig/htdig.conf

Ce sera la même commande à taper pour une mise à jour.

12.5Utilisation

A partir de votre navigateur préféré au niveau de l'URL tapez:

URL: http://www.funix.kervao.fr/htdig/search.html

(Pour une install avec rpm, vous pouvez omettre le **search.html**)

Voilà le résultat de la recherche

Les pages sont classées suivant qu'elles collent plus ou moins avec la recherche, mais vous pouvez éventuellement modifier le type de tri (**Sort by**). Pour info on utilise un modèle de présentation des résultats se trouvant sous **/usr/local/htdig/common** (**/usr/share/htdig** pour une installation avec rpm) avec

header.html qui précède la liste des résultats
footer.html qui suit la liste des résultats
nomatch.html quand on ne trouve rien
syntax.html en cas d'erreur de syntaxe
wrapper.html entête et pied de page

13 Apache 2

13.1 Présentation

Ce paragraphe présente l'installation et la configuration d'**Apache 2.2.4** avec gestion de **PHP** et **MySQL**.

13.2 Installation d'Apache

13.2.1 Installation

Pour **Apache**, on désarchive en tapant:
tar xvfz httpd-2.2.4.tar.gz

Cela va créer le répertoire **httpd-2.2.4**. Avant d'aller plus loin, si **Apache** est déjà installé sur votre système on va le supprimer, on va d'abord supprimer les modules PHP, pour vérifier :

rpm -qa | grep -i php

Si vous obtenez par exemple

php-gd-version
php-ldap-version
mod_php-version
php-common-version
php-version

On supprime en tapant **rpm -e nom-du-package**

Maintenant on va supprimer **Apache**, pour vérifier s'il est présent :

rpm -qa | grep -i apache

Si on obtient par exemple

apache-version
apache-conf-version
apache-modules-version
apache-common-version

Supprimez les. Il est possible que pour des raisons de dépendances que vous ayez un refus de désinstallation, ce n'est pas grave, l'important est que le package **apache-version** soit au moins supprimé pour qu'il n'y ait pas de conflit.

Supprimez éventuellement les répertoires suivants **/var/www**, **/var/log/httpd** et **/etc/httpd**

Dans le répertoire d'**Apache httpd-2.2.4** , on tape alors:

./configure --prefix=/usr/local/apache2 --enable-module=most --enable-shared=max

Par **prefix** on indique que les répertoires d'**Apache** contenant entre autre le fichier de conf se trouveront sous **/usr/local/apache2** c'est utile dans le cas où vous voulez faire coexister deux versions d'**Apache** sur votre système. J'ai du préalablement installer le package **db2-devel**, on tape maintenant:

make

Et enfin en tant que root:

make install

Rajoutez la ligne **/usr/local/apache2/lib** dans le fichier **/etc/ld.so.conf** puis tapez

ldconfig

Pour lancer maintenant **Apache**, il faut taper:

/usr/local/apache2/bin/apachectl start

Maintenant votre navigateur préféré dans le champ URL taper **http://localhost** ou **http://nomdelamachine** et là la page d'accueil d'**Apache** apparaît, pour info celle-ci se trouve sous **/usr/local/apache2/htdocs**.

NOTE Si vous upgradez d'une ancienne version, vos fichiers de conf ne seront pas écrasés.

13.2.2 Pour un lancement automatique sous Mandriva

Pour un lancement automatique on prendra le fichier **apachectl** se trouvant sous **/usr/local/apache2/bin** et on le placera sous **/etc/rc.d/init.d**, et on le renommera **httpd2**

cp /usr/local/apache2/bin/apachectl /etc/rc.d/init.d/httpd2

On rajoutera en début de fichier juste après **#!/bin/sh**

#!/bin/sh

 # Startup script for the Apache Web Server
#
 # chkconfig: 345 85 15
**# description: Apache is a World Wide Web server. It is used to serve **
 # HTML files and CGI.
 # processname: httpd

Pour un lancement automatique à l'état de marche 3,4 et 5 on doit normalement taper:

chkconfig --level 345 httpd2 on

Et pour un arrêt à l'état de marche 0, 1, 2 et 6

chkconfig --level 0126 httpd2 off

Pour lancer le serveur, il suffira maintenant de taper:

/etc/rc.d/init.d/httpd2 start

13.2.3 Pour un lancement automatique sous Ubuntu

Voilà un script de lancement, il faut l'appeler **httpd2** et le placer sous **/etc/init.d**

#!/bin/bash
#

```
# apache    Start the apache2 HTTP server.
#
# The variables below are NOT to be changed.  They are there to make the
# script more readable.

NAME=apache
DAEMON=/usr/local/apache2/bin/httpd
PIDFILE=/usr/local/apache2/logs/httpd.pid
CONF=/usr/local/apache2/conf/httpd.conf
APACHECTL=/usr/local/apache2/bin/${NAME}ctl
# note: SSD is required only at startup of the daemon.
SSD=`which start-stop-daemon`
ENV="env -i LANG=C PATH=/bin:/usr/bin:/usr/local/bin"

trap "" 1

# Check that we're not being started by inetd
if egrep -q -i "^[[:space:]]*ServerType[[:space:]]+inet" $CONF
then
   exit 0
fi

test_config() {
   if [ ! -x $APACHECTL ]; then
   echo "$APACHECTL is not executable, exiting"
   exit 0
   fi

   # ensure we don't leak environment vars into apachectl
   APACHECTL="$ENV $APACHECTL"

   if ! $APACHECTL configtest 2> /dev/null
   then
      printf "Configuration syntax error detected. Not reloading.\n\n"
      $APACHECTL configtest
      exit 1
   fi
}

should_start() {
   if [ ! -x $DAEMON ]; then
   echo "apache is not executable, not starting"
   exit 0
   fi
}

case "$1" in
  start)
   should_start
   test_config
   echo -n "Starting web server: $NAME"
   $ENV $SSD --start --pidfile $PIDFILE --exec $DAEMON > /dev/null
```

```
    ;;

  stop)
    echo -n "Stopping web server: $NAME"
    start-stop-daemon --stop --pidfile $PIDFILE --oknodo
    rm -rf /var/lib/apache/mod-bandwidth/link/*
    ;;

  reload | force-reload)
    test_config
    echo -n "Reloading $NAME configuration"
    start-stop-daemon --stop --pidfile $PIDFILE --signal USR1
    ;;

  restart)
    test_config
    echo -n "Restarting $NAME"
    if ! start-stop-daemon -q --stop --pidfile $PIDFILE --signal HUP; then
    $ENV $SSD --start --pidfile $PIDFILE --exec $DAEMON > /dev/null
    fi
    ;;

  *)
    echo "Usage: /etc/init.d/$NAME {start|stop|reload|force-reload|restart}"
    exit 1
    ;;
esac

if [ $? -eq 0 ]; then
    echo .
    exit 0
else
    echo " failed"
    exit 1
fi
```

On donne les droits d'exécution

chmod 755 /etc/init.d/httpd2

Pour le lancer automatique au démarrage il faudra taper

update-rc.d httpd2 defaults

13.3Installation de PHP

Pour **PHP**, on tapera d'abord

tar xvfz php-5.2.1.tar.gz

Cela va créer un répertoire **php-5.2.1**. A présent dans ce répertoire on tape:

./configure --with-apxs2=/usr/local/apache2/bin/apxs --with-config-file-path=/usr/local/apache2/conf --enable-versioning --with-mysql --with-ftp --enable-bcmath=yes --enable-debug=no --enable-memory-limit=yes --enable-tracks-vars --with-gd --with-zlib --with-kerberos --enable-mbstring --with-mcrypt

Notes - Vous avez besoin du package **flex** contenant **lex**

- les options**--with-gd --with-zlib --with-kerberos** sont utiles pour l'utilisation des mods freeplayer de la freebox

- les options **--enable-mbstring --with-mcrypt** sont nécessaires pour **phpMyAdmin** elles requièrent l'installation des packages **libmcrypt-dev et mcrypt**

Taper ensuite:

make

Puis en tant que root

make install

Cette dernière commande va installer le module **PHP** sous **/usr/local/apache2/modules** et modifier le fichier **httpd.conf** en rajoutant

LoadModule php5_module modules/libphp5.so

Dans le cas d'une ancienne installation avec PHP4, pensez à mettre en commentaire la ligne suivante pour éviter les conflits

LoadModule php4_module modules/libphp4.so

On modifiera maintenant manuellement le fichier **httpd.conf** pour qu'**Apache** prenne en compte **PHP**, à la suite des lignes

AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz

On rajoute

AddType application/x-httpd-php .php .php .php4 .phtml
AddType application/x-httpd-php-source .phps

Par ailleurs à la ligne

DirectoryIndex index.html index.html.var

On rajoutera

DirectoryIndex index.html index.htm index.php index.php index.php4

A présent on va copier le fichier **ini-recommended** se trouvant dans le répertoire de **PHP php-5.2.1** pour le mettre sous **/usr/local/apache2/conf** et en le renommant **php.ini**

cp ./php-5.2.1/php.ini-recommended /usr/local/apache2/conf/php.ini

Dans le cas d'une ancienne installation de **php**, vous avez tout intérêt à repartir du nouveau fichier **php-dist.ini** fourni et de le remodifier.
Dans le répertoire **/usr/local/apache/htdocs** on trouve les fichiers qui seront accessibles à partir de la page d'accueil de votre serveur **Apache**. On doit cependant renommer le fichier suivant.

mv index.html.fr index.html

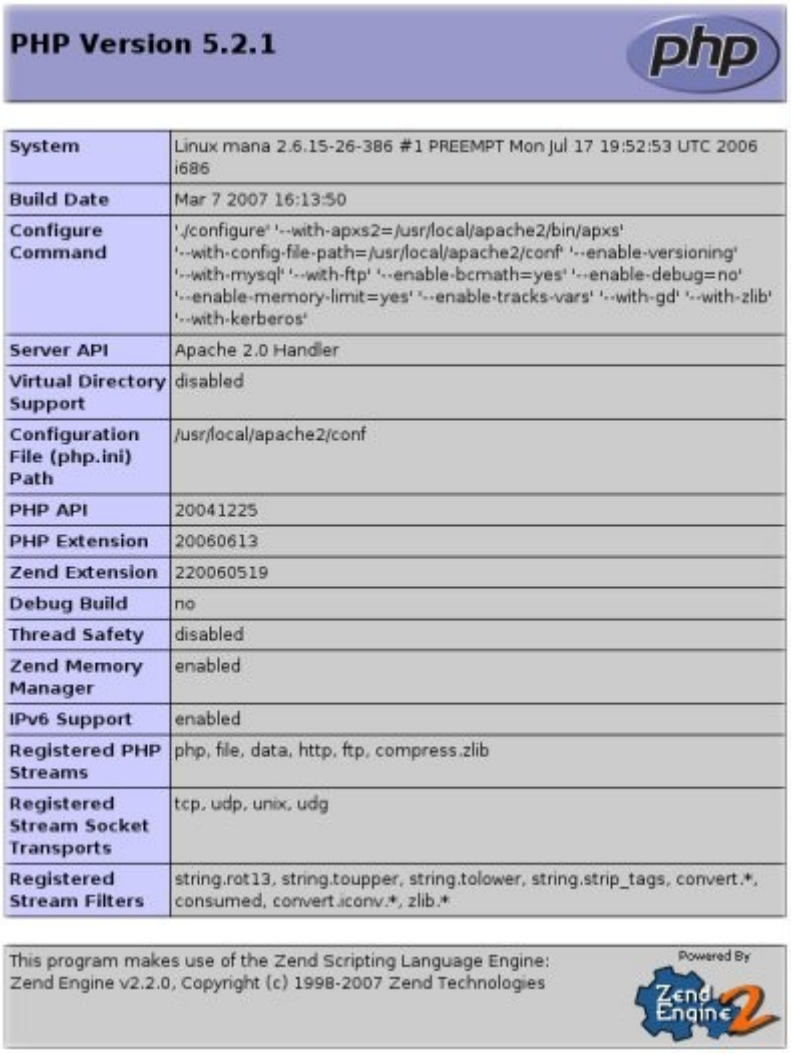
On relance **Apache**

/usr/local/apache2/bin/apachectl restart

Créer maintenant le fichier **infophp.php** contenant

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

Que vous placerez sous **/usr/local/apache2/htdocs**, dans l'URL de votre navigateur préféré, taper **http://localhost/infophp.php** ou **http://nommachine/infophp.php** et là magique devrait s'afficher des info sur la configuration de **PHP** sur votre système.



PHP Version 5.2.1

System	Linux mana 2.6.15-26-386 #1 PREEMPT Mon Jul 17 19:52:53 UTC 2006 i686
Build Date	Mar 7 2007 16:13:50
Configure Command	'./configure' '--with-apxs2=/usr/local/apache2/bin/apxs' '--with-config-file-path=/usr/local/apache2/conf' '--enable-versioning' '--with-mysql' '--with-ftp' '--enable-bcmath=yes' '--enable-debug=no' '--enable-memory-limit=yes' '--enable-tracks-vars' '--with-gd' '--with-zlib' '--with-kerberos'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/apache2/conf
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, data, http, ftp, compress.zlib
Registered Stream Socket Transports	tcp, udp, unix, udg
Registered Stream Filters	string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, convert.iconv.*, zlib.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v2.2.0, Copyright (c) 1998-2007 Zend Technologies

Powered By Zend Engine 2

13.4Présentation de l'arborescence d'Apache

L'installation va créer un répertoire **/usr/local/apache** contenant:
- répertoire **bin** contient les exécutables d'Apache
- répertoire **cgi-bin** contient les scripts CGI
- répertoire **error** contient les messages d'erreur dans de multiples langues, la langue est choisie en fonction de la configuration du navigateur. Les messages sont entièrement configurables.
- répertoire **lib** contient des bibliothèques
- répertoire **build**
- répertoire **conf** contient les fichiers de configuration d'Apache
- répertoire **htdocs** contient la page d'accueil d'Apache
- répertoire **icons** contient des icônes qui servent notamment pour identifier les types de fichier.
- répertoire **include** contient les includes d'Apache
- répertoire **modules** contient les modules d'Apache
- répertoire **logs** contient les fichiers de log d'Apache
- répertoire **man** contient les mans d'Apache

Le répertoire de log contient essentiellement deux fichiers:
- **access_log** listant les accès au serveur
- **error_log** listant les erreurs en tout genre

Le répertoire de modules **modules** contient les modules utilisables par Apache, pour info un module est une extension logicielle à Apache, lui permettant par exemple d'interpréter le PHP4 (module **libphp4.so**). Ce ne sont que les modules chargés dynamiquement qui sont dans ce répertoire.

Le répertoire **/etc/http/conf** contient:
- le fichier de configuration d'Apache **http.conf**
- **mime.types** fixe le type de fichier suivant l'extension du dit fichier (.**doc**=msword, .**ps**=postscript, ...), ça permet au client qui se connecte sur le serveur, de savoir comment interpréter le fichier suivant son extension.
- **magic** sert pour le module **mod_mime_magic**

Vous trouverz également le fichier de config en cas de serveur sécurisé SSL.
-**ssl.conf**
-**ssl-std.conf**

Et des fichiers de conf pour une configuration optimisée haute performance
- **highperformance.conf**
- **highperformance-std.conf**

13.5Le fichier de configuration

Le fichier de conf d'**Apache** se trouve sous **/usr/local/apache** et se nomme **httpd.conf**, voici les points que je juge important dans le fichier:

(...)

 # Répertoire racine d'Apache
ServerRoot "/usr/local/apache"

(...)

#
 # Timeout: The number of seconds before receives and sends time out.
#
 # définition du timeout
Timeout 300

 # nombre de serveurs à lancer au départ et d'instances à lancer
 # prefork MPM
 # StartServers: number of server processes to start
 # MinSpareServers: minimum number of server processes which are kept spare
 # MaxSpareServers: maximum number of server processes which are kept spare
 # MaxClients: maximum number of server processes allowed to start
 # MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule prefork.c>
StartServers 5
MinSpareServers 5
MaxSpareServers 10
MaxClients 150
MaxRequestsPerChild 0
</IfModule>

 # worker MPM
 # StartServers: initial number of server processes to start
 # MaxClients: maximum number of simultaneous client connections
 # MinSpareThreads: minimum number of worker threads which are kept spare
 # MaxSpareThreads: maximum number of worker threads which are kept spare
 # ThreadsPerChild: constant number of worker threads in each server process
 # MaxRequestsPerChild: maximum number of requests a server process serves

```
<IfModule worker.c>
StartServers         2
MaxClients         150
MinSpareThreads     25
MaxSpareThreads     75
ThreadsPerChild     25
MaxRequestsPerChild  0
</IfModule>
```

(...)

```
# définition de l'adresse IP du port du serveur
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, in addition to the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses (0.0.0.0)
#
#Listen 12.34.56.78:80
Listen 80
```

(...)

```
# On lance initialement httpd en tant que root, puis immédiatement
# c'est l'utilisateur nobody (groupe nobody) qui en devient le proprio
# ainsi s'il y a une faille dans Apache, le hacker au lieu de devenir root
# devient nobody avec les droits qui vont avec
# pour vérifier que nobody est bien le proprio
# ps aux | grep httpd

User nobody
Group #-1

# ServerAdmin: Your address, where problems with the server should be
# e-mailed.  This address appears on some server-generated pages, such
# as error documents.
# En cas de problème un email sera envoyé au webmaster, mettez donc
# ici l'adresse email du webmaster
ServerAdmin olivier@asterix.kervao.fr
```

(...)

```
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
# C'est dans ce répertoire qu'on va trouver la page d'accueil d'Apache
DocumentRoot "/usr/local/apache/htdocs"
```

(...)

```
# UserDir: The name of the directory which is appended onto a user's home
# directory if a ~user request is received.
# Chaque utilisateur pourra mettre ses pages dans sa homedirectory
# dans un répertoire public_html, les pages seront accessibles à l'URL

UserDir public_html
```

(...)

 # Définition des fichiers d'entrée
DirectoryIndex index.html index.html.var index.htm index.php index.php3 index.php4

 # défintion des fichiers de protection des pages
AccessFileName .htaccess

(...)

 # nom du fichier d'erreur
ErrorLog logs/error_log

 # niveau de log
 # LogLevel: Control the number of messages logged to the error_log.
 # Possible values include: debug, info, notice, warn, error, crit,
 # alert, emerg.
#
LogLevel warn

(...)

 # ordre de préférence des langues
 # LanguagePriority allows you to give precedence to some languages
 # in case of a tie during content negotiation.
#
 # Just list the languages in decreasing order of preference. We have
 # more or less alphabetized them here. You probably want to change this.
#
LanguagePriority fr en da nl et de el it ja kr no pl pt pt-br ltz ca es sv tw

Dans l'hypothèse où vous utilisez deux serveurs **Apache** (versions 1.3 et 2 par exemple), vous pouvez spécifier le port 80 pour **Apache** 1.3 et le port 8080 pour **Apache** 2 en fixant dans le fichier de configuration **Listen** à 8080. Au niveau de la déclaration des hôtes virtuels il faudra mettre quelque chose comme ça

NameVirtualHost 192.168.13.11:8080

<VirtualHost 192.168.13.11:8080>
ServerName tosh.kervao.fr
DocumentRoot /usr/local/apache2/htdocs
ErrorLog logs/tosh-error_log
TransferLog logs/tosh-access_log
</VirtualHost>