

# Monter son serveur web

# LAMP

Olivier Hoarau ([olivier.hoarau@funix.org](mailto:olivier.hoarau@funix.org))

V3.9 du 4 août 2022

1	Historique.....	2
2	Préambule.....	3
3	Présentation.....	3
4	Installation de MariaDB.....	4
4.1	Compilation de MariaDB.....	4
4.2	Configuration.....	5
4.3	Initialisation des bases de donnée.....	6
4.4	Lancement automatique par systemd.....	9
4.5	Mise à jour.....	10
4.6	Finalisation de l'installation.....	11
4.7	Perte du mot de passe d'administrateur.....	12
4.8	Mettre en place une connexion chiffrée avec MariaDB.....	13
5	Installation d'Apache.....	17
5.1	Compilation d'Apache.....	17
5.2	Présentation de l'arborescence d'Apache.....	18
5.3	Le fichier de configuration d'Apache.....	19
5.4	Lancement automatique de l'application avec systemd.....	21
6	Installation de PHP.....	21
6.1	Compilation de PHP.....	21
6.2	Configuration.....	22
7	Configuration avancée d'Apache.....	24
7.1	Les pages webs utilisateurs.....	24
7.2	Les alias.....	24
7.3	Protection d'une page.....	25
7.4	Les hôtes virtuels.....	26
7.5	Connexion sécurisée avec SSL.....	27
7.6	Scripts CGI.....	30
7.7	PHP et LDAP.....	31
8	Gestion de bases de données avec MariaDB.....	32
8.1	Création des bases de test.....	32
8.2	Tests de fonctionnement avec MariaDB.....	34
8.3	Administration des bases MySQL avec phpMyAdmin.....	36
8.4	Connexion sécurisée entre le serveur MariaDB et phpMyAdmin.....	39
9	Analyser les logs d'Apache avec awstats.....	39
9.1	Présentation.....	39
9.2	Installation et configuration.....	39
9.3	Utilisation.....	46
9.4	Analysez les logs d'un site hébergé non localement.....	51
10	Installer un moteur de recherche avec Hl://dig.....	54
10.1	Présentation.....	54
10.2	Installation.....	54
10.3	Configuration.....	55
10.4	Utilisation.....	59

# 1 Historique

04/08/2022	3.9	passage à MariaDB 10.8.3, Apache 2.4.54, PHP 8.1.18 et phpMyAdmin 5.2
05/11/2021	3.8	Passage à Apache 2.4.48, PHP 7.4.25, MariaDB 10.6.4, phpMyAdmin 5.1.1, rajout de Ht://Dig
11/04/2020	3.7	Passage à Apache 2.4.41, PHP 7.4.4, phpMyAdmin 5.0.2, awstats 7.8, installation de MariaDB 10.8.3 à la place de MySQL, rajout d'une connexion sécurisée à MariaDB. Abandon de Ht://Dig passage à Hl://Dig
23/12/17	3.61	passage à Apache 2.4.29, PHP 5.6.32, MySQL 5.7.20 et phpMyAdmin 4.7.8
3/6/2017	3.6	rajout d'un paragraphe chiffrer les connexions avec SSL
7/01/17	V3.51	filtrer les spams referrer avec awstats
25/12/16	V3.5	passage à MySQL 5.7.17, PHP 5.6.29, Apache 2.4.25, phpMyAdmin 4.6.5.2, awstats 7.8 et webalizer 2.23-08
27/02/13	V3.4	passage à Apache 2.4.3, php 5.4.11, phpMyAdmin 3.5.7 et MySQL 5.5.29 et rajout d'un paragraphe sur la perte du mot de passe administrateur de mysqld
03/10/10	V3.1	passage à Apache 2.2.16, php 5.3.3, MySQL 5.1.50, rajout d'un paragraphe sur la mise en place d'un serveur web dans un environnement chrooté passage à MySQL 5.1.50, Apache 2.2.16, PHP 5.3.3, phpMyAdmin 3.3.7 et webalizer 2.21-02
26/12/09	V3.0	passage à Apache 2.2.14, php 5.3.1, MySQL 5.1.41, phpMyAdmin 3.2.4 et awstats 6.95
3/11/08	V2.9	Passage à Apache 2.2.9, php 5.2.6, MySQL 5.0.67 et phpMyAdmin 2.11.8.1
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 5.4	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 awstat 2.23	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 (v2.01.06)	V1.2	Rajout du paragraphe PHP et LDAP, mise à jour de webalizer
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, MySQL/MariaDB et PHP sous Linux.

La dernière version de ce document est téléchargeable à l'URL <https://www.funix.org>.

Ce document est sous licence Creative Commons Attribution-ShareAlike 4.0 Unported, le détail de la licence se trouve sur le site <http://creativecommons.org/licenses/by-sa/4.0/legalcode>. Pour résumer, vous êtes libres

- de reproduire, distribuer et communiquer cette création au public
- de modifier cette création

suivant les conditions suivantes:

- **Paternité** — Vous devez citer le nom de l'auteur original de la manière indiquée par l'auteur de l'oeuvre ou le titulaire des droits qui vous confère cette autorisation (mais pas d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent votre utilisation de l'oeuvre).
- **Partage des Conditions Initiales à l'Identique** — Si vous transformez ou modifiez cette oeuvre pour en créer une nouvelle, vous devez la distribuer selon les termes du même contrat ou avec une licence similaire ou compatible.

Par ailleurs 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'un serveur Apache avec PHP et MariaDB (communément appelé LAMP pour **L**inux, **A**pache, **M**ySQL et **P**HP), elle est basée sur la compilation complète de ces applications et devrait convenir à toutes les distributions de Linux. Dans le cas présent Mageia 8 a été la distribution support.

Vous pouvez bien entendu choisir d'installer les packages fournis par votre distribution préférée, mais vous trouverez dans cette page des astuces de configuration qui pourront aussi vous être utiles.

Pour **MySQL** on choisira maintenant d'installer le fork de **MySQL**, **MariaDB** qui reste compatible avec les bases de données **MySQL** si vous comptez faire une migration. En synthèse **MySQL** a une approche beaucoup plus commercial alors que **MariaDB** reste totalement sous licence libre. Dans le détail vous pouvez vous référer à [cette page](#) qui présente les différences entre les deux. Par la suite on pourra mentionner indifféremment **MySQL** ou **MariaDB**.

Vous devez récupérer préalablement les sources d' **Apache** à l'URL <http://httpd.apache.org> on récupère ensuite **PHP** à l'URL [www.php.net](http://www.php.net) et enfin **MariaDB** à l'URL <https://mariadb.org/>

Pour mettre en place un serveur web sécurisé dans un environnement "chrooté" consulter cette [page](#) (non maintenue).

Pour installer un webmail basé sur **Roundcube mail** avec une configuration LAMP on ira voir [cette page](#).

## 4 Installation de MariaDB

### 4.1 Compilation de MariaDB

Pour éviter tout conflit avec les packages installés par défaut de la Mageia, j'ai supprimé le package **mariadb-client** et tous les packages associés qui me servent à rien. Dans le répertoire de travail, on décompresse l'archive:

```
tar xvfz mariadb-10.8.3.tar.gz
```

Cela crée le répertoire **mariadb-10.8.3**, sur ma Mageia j'ai dû installer les packages **lib64lz4-devel**, **lib64gnutls-devel**, **lib64curl-devel** et **lib64ncurses-devel**. Dans le répertoire de **MariaDB**, on tape alors:

```
mkdir build
cd build
cmake ..
```

voilà le résultat

```
-- The following OPTIONAL packages have been found:
```

```
* LibXml2
* CURL
* GSSAPI
* LZ4 (required version >= 1.6)
* LibLZMA
* BISON (required version >= 2.4)
```

```
-- The following RECOMMENDED packages have been found:
```

```
* OpenSSL
```

```
-- The following REQUIRED packages have been found:
```

```
* Curses
* Threads
```

```
-- The following features have been disabled:
```

```
* LIBWRAP, Support for tcp wrappers
* SYSTEMD, Systemd scripts and notification support
* CONNECT_ODBC, Support for ODBC in the CONNECT storage engine
```

- \* **CONNECT\_JDBC**, Support for JDBC in the **CONNECT** storage engine
- \* **CONNECT\_MONGODB**, Support for MongoDB in the **CONNECT** storage engine
- \* **NUMA**, NUMA memory allocation policy
- \* **INNODB\_EXTRA\_DEBUG**, Extra InnoDB debug checks
- \* **AWS\_KEY\_MANAGEMENT**, AWS Encryption Key Management Plugin
- \* **EMBEDDED\_SERVER**, Embedded MariaDB Server Library

-- The following **OPTIONAL** packages have not been found:

- \* **Git**
- \* **Java** (required version  $\geq 1.6$ )
- \* **JNI**
- \* **PMEM**
- \* **Boost** (required version  $\geq 1.40.0$ )  
Required for the **OQGraph** storage engine
- \* **BZip2**
- \* **LZO**
- \* **Snappy**

-- **Configuring done**

-- **Generating done**

-- **Build files have been written to: /usr/local/linux/web/mariadb-10.8.3/build**

on tape ensuite

**make**

et en tant que root **make install**

On rajoutera la ligne **/usr/local/lib/mariadb** dans le fichier **/etc/ld.so.conf** et on tapera **ldconfig**.

Les exécutable sont installés par défaut sous **/usr/local/mysql/bin**, les bibliothèques vont se trouver sous **/usr/local/mysql/lib**, maintenant pour que les headers **de mysql** soient accessibles pour d'autres softs qui en auraient l'usage, il faudra créer le lien suivant

**ln -s /usr/local/mysql/include/mysql /usr/include**

vous mettez bien sûr votre chemin absolu de **mysql**.

Si vous upgradez **MySQL** vous pouvez passer au [chapitre 4.5](#), sinon vous devez taper les commandes suivantes :

**groupadd mysql**

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

**useradd mysql -c "MySQL Server" -d /dev/null -g mysql -s /sbin/nologin**

On va créer préalablement le répertoire **/var/log/mysql**, et **mysql** doit en être propriétaire

**mkdir /var/log/mysql**

**chown mysql:mysql /var/log/mysql**

## 4.2 Configuration

On installera maintenant le fichier de configuration **/etc/my.cnf**, voilà à quoi il va ressembler

[client]

```
#emplacement de la socket  
socket = /var/run/mysql/mysql.sock
```

[mysqld]

```
# chemin par défaut d'installation de mariaDB  
basedir = /usr/local/mysql
```

```
# chemin de la base de donnée  
datadir = /usr/local/mysql/data/
```

```
# emplacement de la socket  
socket = /var/run/mysql/mysql.sock
```

```
#répertoire temporaire  
tmpdir = /tmp
```

```
# configuration diverse  
sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES
```

on donne les droits qui vont bien

```
chown root:sys /etc/my.cnf  
chmod 644 /etc/my.cnf
```

## 4.3 Initialisation des bases de donnée

On va d'abord initialiser les bases de données avec le script `mysql_install_db.sh` qui se trouve dans le répertoire `mariadb-10.8.3`, il faudra le modifier pour éviter les erreurs comme cela

**FATAL ERROR: Could not find mariadb**

**The following directories were searched:**

```
@mysql_locations@
```

ou bien encore

```
./mysql_install_db.sh: ligne 444: @HOSTNAME@ : commande introuvable
```

on l'édite et on modifie ainsi les lignes suivantes

```
resolveip=`find_in_dirs resolveip $bindir`
```

au lieu de `resolveip=`find_in_dirs resolveip @resolveip_locations@`` et

```
mysqld=`find_in_dirs mariadb $bindir`
```

```
langdir="/usr/local/mysql/share/french"
```

```
srcpkgdatadir="/usr/local/mysql/share"
```

ainsi qu'on remplace `hostname=@HOSTNAME@` par `hostname=hostname``

Une fois corrigé, on le lance

```
bash ./mysql_install_db.sh --user=mysql --basedir=/usr/local/mysql/ --datadir=/usr/local/mysql/data --  
buildldir=/chemin/mariadb-10.8.3/
```

et voilà le résultat

**Installing MariaDB/MySQL system tables in '/usr/local/mysql/data' ...  
OK**

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

**Two all-privilege accounts were created.  
One is root@localhost, it has no password, but you need to  
be system 'root' user to connect. Use, for example, sudo mysql  
The second is mysql@localhost, it has no password either, but  
you need to be the system 'mysql' user to connect.  
After connecting you can set the password, if you would need to be  
able to connect as any of these users with a password and without sudo**

**See the MariaDB Knowledgebase at <https://mariadb.com/kb>**

**You can start the MariaDB daemon with:  
cd '/usr/local/mysql/' ; /usr/local/mysql/bin/mysqld\_safe --datadir='/usr/local/mysql/data'**

**You can test the MariaDB daemon with mysql-test-run.pl  
cd '/usr/local/mysql/mysql-test' ; perl mysql-test-run.pl**

**Please report any problems at <https://mariadb.org/jira>**

**The latest information about MariaDB is available at <https://mariadb.org/>.**

**Consider joining MariaDB's strong and vibrant community:  
<https://mariadb.org/get-involved/>**

Dans la configuration présentée dans cette page les bases de données se trouvent sous **/usr/local/mysql/data**,  
**mysql** doit en être propriétaire

```
chown -R mysql:mysql /usr/local/mysql/data
```

on va également mettre les droits à 755 sinon seul root aura accès à la base **mysql**

```
chmod 755 /usr/local/mysql/data
```

On lance maintenant le serveur en tapant

```
/usr/local/mysql/bin/mysqld_safe --datadir='/usr/local/mysql/data'
```

voilà le résultat

```
220730 17:28:58 mysqld_safe Logging to '/usr/local/mysql/data/ultra.kervao.fr.err'.  
220730 17:28:58 mysqld_safe Starting mariadbd daemon with databases from /usr/local/mysql/data
```

On sécurise maintenant l'accès à la base de données en tapant **mysql\_secure\_installation** en tant que root, voilà  
le résultat

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

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

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

Setting the root password or using the unix\_socket ensures that nobody can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix\_socket authentication [Y/n] y  
Enabled successfully!  
Reloading privilege tables..  
... Success!

You already have your root account protected, so you can safely answer 'n'.

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

By default, a MariaDB installation has an anonymous user, allowing anyone to log into MariaDB 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, MariaDB 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]  
- 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]  
... Success!

Cleaning up...

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



Thanks for using MariaDB!

maintenant on peut se connecter à la base en tapant

```
mysql -u root -p
```

**Enter password:** on rentre le mot de passe défini plus haut, et voilà le résultat

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 82
Server version: 10.8.3-MariaDB Source distribution
```

```
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
MariaDB [(none)]>
```

## 4.4 Lancement automatique par systemd

Avec `systemd` créer le fichier `/usr/lib/systemd/system/mariadb.service` qui contient

```
[Unit]
Description=MariaDB Server
After=syslog.target
After=network.target

[Service]
Type=simple
PermissionsStartOnly=true
ExecStartPre=/bin/mkdir -p /var/run/mysql
ExecStartPre=/bin/chown mysql:mysql -R /var/run/mysql
ExecStart=/usr/local/mysql/bin/mysqld --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data --log-
error=/var/log/mysql/error.log --pid-file=/var/run/mysql/mysqld.pid
--socket=/var/run/mysql/mysqld.sock
TimeoutSec=300
PrivateTmp=true
User=mysql
Group=mysql
WorkingDirectory=/usr/local

[Install]
WantedBy=multi-user.target
```

on lance maintenant le service **MariaDB** en tapant

```
systemctl start mariadb.service
```

voilà le statut du daemon une fois lancé

```
systemctl status mariadb.service
```

voilà le résultat

- **mariadb.service - MariaDB Server**  
Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)  
Active: active (running) since Sat 2022-08-01 14:47:29 CEST; 8s ago  
Process: 45212 ExecStartPre=/bin/mkdir -p /var/run/mysql (code=exited, status=0/SUCCESS)  
Process: 45214 ExecStartPre=/bin/chown mysql:mysql -R /var/run/mysql (code=exited,

```
status=0/SUCCESS)
Main PID: 45215 (mysqld)
Tasks: 8 (limit: 4677)
Memory: 85.2M
CPU: 204ms
CGroup: /system.slice/mariadb.service
└─45215 /usr/local/mysql/bin/mysqld --basedir=/usr/local/mysql --datadir=/chroot/data --log-
error=/var/log/mysqld/error.log --pid-file=/var/run/mysqld/mysqld.pid
--socket=/var/run/mysqld/mysqld.sock
```

```
août 01 14:47:29 mana.kervao.fr systemd[1]: Starting MariaDB Server...
août 01 14:47:29 mana.kervao.fr systemd[1]: Started MariaDB Server.
août 01 14:47:29 mana.kervao.fr mysqld[45215]: 2022-08-01 14:47:29 0 [Note]
/usr/local/mysql/bin/mysqld (server 10.8.3-MariaDB) starting as process 45215 ...
```

Maintenant pour que le service soit lancé à chaque démarrage on tapera

```
systemctl enable mariadb.service
```

## 4.5 Mise à jour

Juste après avoir installé la nouvelle version de **MariaDB** la première chose à faire est de relancer le daemon

```
systemctl restart mariadb
```

il faudra faire ensuite un upgrade de la base de données en tapant

```
mysql_upgrade -u root -p
```

cela va donner

```
Enter password: Checking if update is needed.
Checking server version.
Running queries to upgrade MySQL server.
Checking system database.
mysql.columns_priv OK
mysql.db OK
mysql.engine_cost OK
mysql.event OK
mysql.func OK
mysql.general_log OK
mysql.gtid_executed OK
mysql.help_category OK
mysql.help_keyword OK
mysql.help_relation OK
mysql.help_topic OK
mysql.innodb_index_stats OK
mysql.innodb_table_stats OK
mysql.ndb_binlog_index OK
mysql.plugin OK
```

...

```
sys
sys.sys_config OK
Phase 7/7: Running 'FLUSH PRIVILEGES'
OK
```

La mise à jour de la base est terminée, en revanche si vous avez l'erreur suivante

```
Error  : Cannot load from mysql.proc. The table is probably corrupted  
error  : Corrupt  
sys.io_by_thread_by_latency  
Error  : Cannot load from mysql.proc. The table is probably corrupted  
error  : Corrupt  
sys.io_global_by_file_by_bytes  
Error  : Cannot load from mysql.proc. The table is probably corrupted  
error  : Corrupt  
sys.io_global_by_file_by_latency  
Error  : Cannot load from mysql.proc. The table is probably corrupted  
error  : Corrupt  
sys.io_global_by_wait_by_bytes  
Error  : Cannot load from mysql.proc. The table is probably corrupted  
error  : Corrupt
```

il faudra retaper la commande en rajoutant

```
mysql_upgrade -uroot -p --force
```

Il faudra ensuite relancer le serveur

```
systemctl restart mariadb
```

## 4.6 Finalisation de l'installation

Pour la suite des opérations 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.

```
/usr/local/mysql/bin/mysql -u root -p mysql  
Enter password:
```

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 25  
Server version: 10.8.3-MariaDB Source distribution
```

```
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
MariaDB [(mysql)]>
```

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

```
MariaDB [(mysql)]> GRANT ALL PRIVILEGES ON *.* TO 'olivier'@'localhost'  
> IDENTIFIED BY 'mot-de-passe' WITH GRANT OPTION;
```

pour créer un utilisateur toto qui n'aura accès qu'à la base test avec certaines fonctions citées ci-dessous à partir de la machine asterix.fr on tapera

```
MariaDB [(mysql)]> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP  
ON test.*  
TO toto@asterix.fr  
IDENTIFIED by 'mot-de-passe';
```

si vous tapez



```
MariaDB [(mysql)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,04 sec)
```

on relance ensuite le daemon **mariadb**

```
systemctl restart mariadb
```

## 4.8 Mettre en place une connexion chiffrée avec MariaDB

Maintenant pour créer une connexion sécurisée on va créer le répertoire `/etc/mysql` et `/etc/mysql/ssl`. On va sous `/etc/mysql/ssl` et on génère une clé RSA privée pour l'autorité de certification maison (CA pour Certificate Authority). Il est bien évident que les certificats et clé de votre CA ne seront valables que sur votre seul réseau interne, à moins que vous soyiez une CA reconnue et enregistrée sur internet ! Si votre site est accessible par internet je vous conseille de créer un certificat qui sera valable sur internet avec [let's encrypt](#).

```
openssl genrsa 2048 > ca-key.pem
```

voilà le résultat

```
Generating RSA private key, 2048 bit long modulus
.....+++++
.....+++++
e is 65537 (0x010001)
```

on crée maintenant le certificat du CA basé sur la clé précédemment créée en tapant

```
openssl req -new -x509 -nodes -days 365000 -key ca-key.pem -out ca-cert.pem
```

voilà le résultat

**You are about to be asked to enter information that will be incorporated into your certificate request.**

**What you are about to enter is what is called a Distinguished Name or a DN.**

**There are quite a few fields but you can leave some blank**

**For some fields there will be a default value,**

**If you enter '.', the field will be left blank.**

```
-----
Country Name (2 letter code) [XX]:FR
State or Province Name (full name) [Default Province]:
Locality Name (eg, city) [Default City]:
Organization Name (eg, company) [Default Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []: ca-mariadb
Email Address []:olivier.hoarau@funix.org
```

je me retrouve avec les fichiers **ca-cert.pem** (certificat) et **ca-key.pem** (la clé associée au certificat). Maintenant je crée un certificat pour le serveur en tapant

```
openssl req -newkey rsa:2048 -days 365000 -nodes -keyout server-key.pem -out server-req.pem
```

voilà le résultat

```
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'server-key.pem'
```

**You are about to be asked to enter information that will be incorporated**

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [XX]:FR  
State or Province Name (full name) [Default Province]:None  
Locality Name (eg, city) [Default City]: None  
Organization Name (eg, company) [Default Company Ltd]: None  
Organizational Unit Name (eg, section) []: None  
Common Name (e.g. server FQDN or YOUR name) []: serveur-mariadb  
Email Address []:olivier.hoarau@funix.org

Please enter the following 'extra' attributes  
to be sent with your certificate request

A challenge password []:

An optional company name []:

Attention pour le nom du serveur il faut mettre le nom de notre serveur tel qu'on le retrouve dans le fichier /etc/hosts. Maintenant on crée la clé RSA du serveur

**openssl rsa -in server-key.pem -out server-key.pem**

et voilà le résultat

**writing RSA key**

on signe le certificat du serveur

**openssl x509 -req -in server-req.pem -days 365000 -CA ca-cert.pem -CAkey ca-key.pem -set\_serial 01 -out server-cert.pem**

voilà le résultat

**Signature ok**

**subject=C = FR, ST = None, L = None, O = None, OU = None, CN = serveur-mariadb, emailAddress = olivier.hoarau@funix.org**

**Getting CA Private Key**

je me retrouve avec les fichiers supplémentaires suivants **server-cert.pem**, **server-key.pem** et **server-req.pem**.

On crée maintenant le certificat du client

**openssl req -newkey rsa:2048 -days 365000 -nodes -keyout client-key.pem -out client-req.pem**

voilà le résultat

**Generating a RSA private key**

.....+++++

.....+++++

**writing new private key to 'client-key.pem'**

-----

You are about to be asked to enter information that will be incorporated  
into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [XX]:FR  
State or Province Name (full name) [Default Province]:None

**Locality Name (eg, city) [Default City]:None**  
**Organization Name (eg, company) [Default Company Ltd]:None**  
**Organizational Unit Name (eg, section) []:None**  
**Common Name (e.g. server FQDN or YOUR name) []:client-mariadb**  
**Email Address []:olivier.hoarau@funix.org**

Please enter the following 'extra' attributes  
to be sent with your certificate request

A challenge password []:  
An optional company name []:None

on crée la clé RSA du client

```
openssl rsa -in client-key.pem -out client-key.pem
```

voilà le résultat

**writing RSA key**

et on signe le certificat du client

```
openssl x509 -req -in client-req.pem -days 365000 -CA ca-cert.pem -CAkey ca-key.pem -set_serial 01 -out client-cert.pem
```

voilà le résultat

**Signature ok**  
**subject=C = FR, ST = None, L = None, O = None, OU = None, CN = client-mariadb, emailAddress = olivier.hoarau@funix.org**  
**Getting CA Private Key**

on vérifie que tout est OK en tapant

```
openssl verify -CAfile ca-cert.pem server-cert.pem client-cert.pem
```

voilà le résultat

**server-cert.pem: OK**  
**client-cert.pem: OK**

si vous avez comme erreur

**error 18 at 0 depth lookup: self signed certificate**  
**error client-cert.pem: verification failed**

Ca vient d'un problème sur la ligne **Common Name (e.g. server FQDN or YOUR name) []** vous devez avoir un champ différent pour le CA, le serveur et le client, vous pouvez très bien utiliser des alias différents en les définissant dans **/etc/hosts**. Maintenant on va rendre **mysql** propriétaire du répertoire **ssl** pour qu'il puisse lire les fichiers.

```
chown -R mysql:mysql /etc/mysql/ssl/
```

on va éditer le fichier **/etc/my.cnf** et rajouter les lignes suivantes au niveau du serveur et du client

**[client]**

```
ssl-ca=/etc/mysql/ssl/ca-cert.pem  
ssl-cert=/etc/mysql/ssl/client-cert.pem  
ssl-key=/etc/mysql/ssl/client-key.pem
```

**[mysqld]**

```
ssl-ca=/etc/mysql/ssl/ca-cert.pem
ssl-cert=/etc/mysql/ssl/server-cert.pem
ssl-key=/etc/mysql/ssl/server-key.pem
```

On relance le serveur

```
systemctl restart mariadb
```

et on tape

```
mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.8.3-MariaDB Source distribution
```

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
MariaDB [(none)]>status
```

```
-----
mysql Ver 15.1 Distrib 10.8.3-MariaDB, for Linux (x86_64) using readline 5.1
```

```
Connection id:      4
Current database:
Current user:       root@localhost
SSL:                Cipher in use is Cipher in use is TLS_AES_256_GCM_SHA384
Current pager:     stdout
Using outfile:     ''
Using delimiter:   ;
Server:            MariaDB
Server version:    10.8.3-MariaDB Source distribution
Protocol version:  10
Connection:       Localhost via UNIX socket
Server characterset: latin1
Db characterset:  latin1
Client characterset: utf8mb3
Conn. characterset: utf8mb3
UNIX socket:      /var/run/mysqld/mysqld.sock
Uptime:           3 min 31 sec
```

```
Threads: 4 Questions: 53 Slow queries: 0 Opens: 44 Open tables: 38 Queries per second avg: 0.251
-----
```

et

```
MariaDB [(none)]> SHOW VARIABLES LIKE '%ssl%';
```

```
+-----+
| Variable_name | Value                               |
+-----+-----+
| have_openssl  | YES                                 |
| have_ssl      | YES                                 |
| ssl_ca        | /etc/mysql/ssl/ca-cert.pem        |
| ssl_capath    |                                     |
| ssl_cert      | /etc/mysql/ssl/server-cert.pem    |
| ssl_cipher    |                                     |
| ssl_crl       |                                     |
| ssl_crlpath   |                                     |
| ssl_key       | /etc/mysql/ssl/server-key.pem     |
+-----+-----+
```



```
| version_ssl_library | OpenSSL 1.1.1q 5 Jul 2022 |  
+-----+-----+  
10 rows in set (0.001 sec)
```

C'est bon vous êtes en connexion sécurisée.

Pour aller plus loin vous pouvez consulter [cette page](#) qui va vous indiquer (par exemple) comment forcer les connexions chiffrées par utilisateur, machine, ou autre.

## 5 Installation d'Apache

### 5.1 Compilation d'Apache

Pour Apache, on installera d'abord le package **lib64expat-devel**, on désarchive en tapant:

```
tar xvzf httpd-2.4.54.tar.gz
```

Cela va créer le répertoire **httpd-2.4.54**.

Il faudra préalablement installer les outils **apr** et **apr-util** qu'on téléchargera sur le site [apr.apache.org](http://apr.apache.org). On décompresse la première archive en tapant

```
tar xvzf apr-1.7.0.tar.gz
```

cela donne le répertoire **apr-1.7.0** dans lequel on tape successivement

```
./configure  
make
```

puis en tant que root

```
make install
```

on édite maintenant le fichier **/etc/ld.so.conf** et on rajoute la ligne suivante

```
/usr/local/apr/lib
```

on décompresse la deuxième archive en tapant

```
tar xvzf apr-util-1.6.1.tar.gz
```

cela donne le répertoire **apr-util-1.6.1** on y tape successivement

```
./configure --with-apr=/chemin-absolu/apr-1.7.0 --with-crypto --with-openssl=/usr/local/ssl/lib
```

```
make
```

puis en tant que root

```
make install  
ldconfig
```

Dans le répertoire d'Apache **httpd-2.4.54**, on tape alors:

**./configure --prefix=/usr/local/apache2 --enable-modules=most**

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. 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 (ou du moins un **It Works !!**), pour info celle-ci se trouve sous **/usr/local/apache2/htdocs**.

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

## 5.2 Présentation de l'arborescence d'Apache

L'installation va créer un répertoire **/usr/local/apache2** contenant:

- répertoire **auth** qui contient les mots de passe chiffrée pour les pages web avec accès par mot de passe
- 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** contient des paramètres divers de configuration de compilation
- 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 manuels 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 PHP (module **libphp7.so**). Ce ne sont que les modules chargés dynamiquement qui sont dans ce répertoire.

Le répertoire **/usr/local/apache2/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 trouverez également un tas d'autres fichiers de config fournis comme exemple dans les répertoire **extra** et **original**.

## 5.3 Le fichier de configuration d'Apache

Le fichier de conf d'Apache se trouve sous **/usr/local/apache2** et se nomme **httpd.conf**, voici les points importants du fichier:

(...)

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

(...)

```
# 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
# avec un deuxième serveur apache, vous pouvez indiquer un port 8080
# dans l'url du navigateur il faudra mettre http://url:8080
```

```
Listen 80
```

(...)

```
# chargement des modules complémentaires
```

```
LoadModule authn_file_module modules/mod_authn_file.so
LoadModule authn_core_module modules/mod_authn_core.so
LoadModule authz_host_module modules/mod_authz_host.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule authz_user_module modules/mod_authz_user.so
LoadModule authz_core_module modules/mod_authz_core.so
LoadModule access_compat_module modules/mod_access_compat.so
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
LoadModule reqtimeout_module modules/mod_reqtimeout.so
LoadModule filter_module modules/mod_filter.so
LoadModule mime_module modules/mod_mime.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule env_module modules/mod_env.so
LoadModule headers_module modules/mod_headers.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule version_module modules/mod_version.so
LoadModule ssl_module modules/mod_ssl.so
LoadModule unixd_module modules/mod_unixd.so
LoadModule status_module modules/mod_status.so
LoadModule autoindex_module modules/mod_autoindex.so
LoadModule cgid_module modules/mod_cgid.so
```

```

LoadModule dir_module modules/mod_dir.so
LoadModule alias_module modules/mod_alias.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule wsgi_module modules/mod_wsgi.so
LoadModule php_module    modules/libphp.so

# 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 daemon avec les droits qui vont avec
# pour vérifier que daemon est bien le proprio
# ps aux | grep httpd

User daemon
Group dameon

# 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"

(...)

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

(...)

# 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

```

En l'état s'est largement suffisant comme configuration pour faire tourner un serveur **Apache**, on verra plus loin comment le personnaliser et aller plus loin.

## 5.4 Lancement automatique de l'application avec systemd

Pour un lancement automatique il faudra créer préalablement le fichier `/usr/lib/systemd/system/httpd.service`

```
[Unit]
Description=Apache Web Server
After=network.target remote-fs.target nss-lookup.target
```

```
[Service]
Type=forking
PIDFile=/var/run/httpd/httpd.pid
ExecStart=/usr/local/apache2/bin/apachectl start
ExecStop=/usr/local/apache2/bin/apachectl graceful-stop
ExecReload=/usr/local/apache2/bin/apachectl graceful
PrivateTmp=true
LimitNOFILE=infinity
```

```
[Install]
WantedBy=multi-user.target
```

on lance apache en tapant `systemctl start httpd`

voilà le résultat en tapant `systemctl status httpd`

```
● httpd.service - Apache Web Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled)
   Active: active (running) since sam. 2015-10-17 13:48:52 CEST; 11s ago
   Process: 1219 ExecStart=/usr/local/apache2/bin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 1222 (httpd)
   CGroup: /system.slice/httpd.service
           └─1222 /usr/local/apache2/bin/httpd -k start
             └─1223 /usr/local/apache2/bin/httpd -k start
               └─1224 /usr/local/apache2/bin/httpd -k start
                 └─1225 /usr/local/apache2/bin/httpd -k start
```

pour qu'apache soit lancé automatique à chaque démarrage on tapera `systemctl enable httpd.service`

# 6 Installation de PHP

## 6.1 Compilation de PHP

Pour **php**, on aura besoin préalablement de **d'oniguruma** qu'on récupérera ici <https://github.com/kkos/oniguruma/releases> on décompresse l'archive en tapant :

```
tar xvfz onig-6.9.8.tar.gz
```

cela donne le répertoire **onig-6.9.8** dans lequel on tape

```
./configure
make
```

puis en tant que root

```
make install
```

on installera également les packages **argon2**, **argon2-devel**, **lib64gd-devel**, **lib64zip-devel** et **lib64sqlite3-devel** on revient à **php** qu'on décompresse en tapant

```
tar xvfj php-8.1.8.tar.bz2
```

Cela va créer un répertoire **php-8.1.8**. A présent dans le répertoire de **php**, vous taperez

```
./configure --with-apxs2=/usr/local/apache2/bin/apxs --with-config-file-path=/usr/local/apache2/conf --enable-bcmath=yes --enable-debug=no --with-zlib --with-kerberos --enable-mbstring --enable-ftp --with-mysqli --with-openssl --with-curl --enable-intl --with-ldap --enable-gd --with-zip --with-pdo-mysql --with-password-argon2 --with-freetype --with-external-gd --with-pear
```

Bizarrement il ne m'a pas trouvé **ldap** j'ai dû créer le lien suivant

```
ln -s /usr/lib64/libldap-2.4.so.2.11.5 /usr/lib/libldap.so
```

Tapez maintenant

```
make
```

si vous rencontrez l'erreur suivante

```
ldap.o: undefined reference to symbol 'ber_memfree'
```

il faudra éditer le fichier **Makefile** et au niveau de la ligne définissant **EXTRA\_LIBS** on rajoutera **-llber** on retape **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 php_module modules/libphp.so
```

## 6.2 Configuration

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 .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

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

### cp php.ini-production /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 relance **Apache**

### systemctl restart httpd

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

```
<?php
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 magie devrait s'afficher des info sur la configuration de **PHP** sur votre système.

PHP Version 8.1.8 	
System	Linux ultra.kervao.fr 5.15.50-desktop-1.mga8 #1 SMP Sat Jun 25 13:56:51 UTC 2022 x86_64
Build Date	Jul 30 2022 22:58:58
Build System	Linux ultra.kervao.fr 5.15.50-desktop-1.mga8 #1 SMP Sat Jun 25 13:56:51 UTC 2022 x86_64 x86_64 GNU/Linux
Configure Command	'./configure' '--with-apxs2=/usr/local/apache2/bin/apxs' '--with-config-file-path=/usr/local/apache2/conf' '--enable-bcmath=yes' '--enable-debug=no' '--with-zlib' '--with-kerberos' '--enable-mbstring' '--enable-ftp' '--with-mysql' '--with-openssl' '--with-curl' '--enable-intl' '--with-ldap' '--enable-gd' '--with-zip' '--with-pdo-mysql' '--with-password-argon2' '--with-freetype' '--with-external-gd' '--with-pear' 'PKG_CONFIG_PATH=/usr/lib/pkgconfig:/usr/local/lib/pkgconfig:/usr/lib64/pkgconfig'
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	/usr/local/apache2/conf
Loaded Configuration File	(none)
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902.TS
PHP Extension Build	API20210902.TS
Debug Build	no
Thread Safety	enabled
Thread API	POSIX Threads
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk

ATTENTION pour un fonctionnement optimal de **mysql** avec **php**, il faudra rajouter préciser la position du fichier socket de **mysql** dans le fichier **php.ini**

```
pdo_mysql.default_socket=/var/run/mysqld/mysqld.sock
```

sinon avec **phpMyAdmin** vous pourriez avoir une erreur du style

#2002 - Le serveur ne répond pas. (ou l'interface de connexion ("socket") vers le serveur MySQL local n'est pas correctement configurée)

## 7 Configuration avancée d'Apache

### 7.1 Les pages webs utilisateurs

Dans le fichier **httpd.conf** on doit rajouter

```
UserDir public_html
```

```
<Directory "/home/*/public_html">
  Options Indexes FollowSymLinks Includes ExecCGI
  AllowOverride All
  Require all granted
</Directory>
```

et relancez **httpd**, les utilisateurs qui veulent avoir une homepage perso devront placer leurs fichiers dans un répertoire **public\_html** placé dans leur homedirectory. Les pages seront accessibles à partir de l'URL `http://serveur-apache/~utilisateur`

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 gênant 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
```

### 7.2 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/"
```

on rajoute ensuite pour chacun des répertoires

```
<Directory "/usr/doc/html">
  Options Indexes FollowSymLinks Includes ExecCGI
  AllowOverride All
  Require all granted
</Directory>
```



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

## 7.3 Protection d'une page

La protection d'une page pour l'utilisateur **olivier** se fait de manière très simple, tout d'abord dans le fichier de configuration d'apache on doit avoir la ligne

```
# définition des fichiers de protection des pages
AccessFileName .htaccess
```

il faut bien évidemment relancer apache.

Tous les fichiers à accès limité doivent être concentrés 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/apache2/auth` (éventuellement à créer), pour info vous pouvez changer le chemin `/usr/local/apache2` 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/apache2/auth`, puis de taper:

```
htpasswd -c /usr/local/apache2/auth/olivier.users olivier
```

Pour info cet exécutable se trouve dans le répertoire d'installation d'apache précisément sous `httpd-2.4.48/support`. 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/apache2/auth/olivier.group` qui contiendra la liste des personnes autorisées à accéder aux pages réservées :

```
autorise: olivier veronique
```

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'oeil dans les fichiers **.htaccess** de vos utilisateurs, le fichier **httpd.conf** doit contenir la directive suivante:

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

## 7.4 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).

On va rajouter tout à la fin du fichier:

```
<VirtualHost 192.168.13.11>  
  ServerName obelix.breizland.bz  
  DocumentRoot /usr/local/apache2/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>
```

pour chacun des répertoires on doit créer ensuite les instructions suivantes

```
<Directory "/usr/local/asterix">  
  Options Indexes FollowSymLinks Includes ExecCGI  
  AllowOverride All  
  Require all granted  
</Directory>
```

Relancez Apache en tapant:

```
systemctl restart httpd
```

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:

```
systemctl restart dns
```

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**

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

```
<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>
```

## 7.5 Connexion sécurisée avec SSL

Si votre site est accessible par internet je vous conseille de créer un certificat qui sera valable sur internet avec [let's encrypt](#). Dans le cas présent on va créer un certificat perso auto signé qui ne marchera que pour mon réseau privé comme j'ai pu le faire pour [sendmail](#) et [dovecot](#). On veillera à ce que les répertoires **/etc/ssl/apache** et **/etc/ssl/public** soient préalablement créés.

```
openssl req -new -x509 -nodes -out /etc/ssl/public/apache.crt -keyout /etc/ssl/apache/apache.key
```

voilà le résultat

```
Generating a 2048 bit RSA private key
...+++
.....+++
writing new private key to '/etc/ssl/apache/apache.key'
-----
You are about to be asked to enter information that will be incorporated into your certificate request.
```

What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.

```
-----  
Country Name (2 letter code) [AU]:FR  
State or Province Name (full name) [Some-State]:Bretagne  
Locality Name (eg, city) []:Brest  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:None  
Organizational Unit Name (eg, section) []:None  
Common Name (e.g. server FQDN or YOUR name) []:www.asterix.breizland.bz
```

Email Address []:olivier.hoarau@funix.org

On va modifier maintenant le fichier de configuration d'**apache /usr/local/apache2/conf/httpd.conf** en activant les modules suivants

```
LoadModule ssl_module modules/mod_ssl.so  
LoadModule log_config_module modules/mod_log_config.so  
LoadModule setenvif_module modules/mod_setenvif.so  
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
```

j'ai décommenté également l'inclusion

```
# Secure (SSL/TLS) connections  
Include conf/extra/httpd-ssl.conf  
#  
# Note: The following must must be present to support  
# starting without SSL on platforms with no /dev/random equivalent  
# but a statically compiled-in mod_ssl.  
#
```

plus loin dans le fichier au niveau de la déclaration d'un hôte virtuel on écrira

```
<VirtualHost 192.168.0.13:443>  
ServerName funix.kervao.fr  
DocumentRoot /data/homepage/www.funix.org  
ErrorLog logs/funix-error_log  
TransferLog logs/funix-access_log  
  
SSLEngine on  
# on pointe vers le certificat et la clé privée  
SSLCertificateFile /etc/ssl/public/apache.crt  
SSLCertificateKeyFile /etc/ssl/apache/apache.key  
</VirtualHost>
```

si vous voulez rediriger l'ensemble du flux 80 vers le port 443 de votre hôte, il faudra rajouter la directive suivante

```
<VirtualHost 192.168.0.13:80>  
ServerName funix.kervao.fr  
# on redirige les connexions sur le port classique 80 vers le port 443 utilisé par SSL  
Redirect / https://funix.kervao.fr/  
</VirtualHost>
```

tout en pensant à ce que la redirection vers le port 443 soit bien en place

```
<VirtualHost 192.168.0.13:443>  
ServerName funix.kervao.fr
```

```
DocumentRoot /data/homepage/www.funix.org
ErrorLog logs/funix-error_log
TransferLog logs/funix-access_log
```

```
SSLEngine on
# on pointe vers le certificat et la clé privée
SSLCertificateFile /etc/ssl/public/apache.crt
SSLCertificateKeyFile /etc/ssl/apache/apache.key
</VirtualHost>
```

Dans le fichier `/usr/local/apache2/conf/extra/httpd-ssl.conf` on y trouve

```
#
# When we also provide SSL we have to listen to the
# standard HTTP port (see above) and to the HTTPS port
Listen 443

# SSL Cipher Suite:
# List the ciphers that the client is permitted to negotiate,
# and that httpd will negotiate as the client of a proxied server.
# See the OpenSSL documentation for a complete list of ciphers, and
# ensure these follow appropriate best practices for this deployment.
# httpd 2.2.30, 2.4.13 and later force-disable aNULL, eNULL and EXP ciphers,
# while OpenSSL disabled these by default in 0.9.8zf/1.0.0r/1.0.1m/1.0.2a.
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4
SSLProxyCipherSuite HIGH:MEDIUM:!MD5:!RC4

# User agents such as web browsers are not configured for the user's
# own preference of either security or performance, therefore this
# must be the prerogative of the web server administrator who manages
# cpu load versus confidentiality, so enforce the server's cipher order.
SSLHonorCipherOrder on

# SSL Protocol support:
# List the protocol versions which clients are allowed to connect with.
# Disable SSLv3 by default (cf. RFC 7525 3.1.1). TLSv1 (1.0) should be
# disabled as quickly as practical. By the end of 2016, only the TLSv1.2
# protocol or later should remain in use.
SSLProtocol all -SSLv3
SSLProxyProtocol all -SSLv3

# Pass Phrase Dialog:
# Configure the pass phrase gathering process.
# The filtering dialog program ('builtin' is an internal
# terminal dialog) has to provide the pass phrase on stdout.
SSLPassPhraseDialog builtin

# Inter-Process Session Cache:
# Configure the SSL Session Cache: First the mechanism
# to use and second the expiring timeout (in seconds).
#SSLSessionCache "dbm:/usr/local/apache2/logs/ssl_cache"
SSLSessionCache "shmcb:/usr/local/apache2/logs/ssl_cache(512000)"
SSLSessionCacheTimeout 300
```

tout le reste est en commentaire, on relance **apache**

```
systemctl stop httpd.service
systemctl start httpd.service
```

on pense à ouvrir le port 443 sur le firewall et on se connecte normalement sur le site. La première fois avec un certificat perso auto signé non reconnu par une autorité de certification il faudra accepter l'exception de sécurité, en cliquant sur **Accepter le risque et poursuivre**



## Attention : risque probable de sécurité

Firefox a détecté une menace de sécurité potentielle et n'a pas poursuivi vers ultra.kervao.fr. Si vous accédez à ce site, des attaquants pourraient dérober des informations comme vos mots de passe, courriels, ou données de carte bancaire.

[En savoir plus...](#)

**Retour (recommandé)**

Avancé...

ultra.kervao.fr utilise un certificat de sécurité invalide.  
Le certificat n'est pas sûr car il est auto-signé.  
Code d'erreur : MOZILLA\_PKIX\_ERROR\_SELF\_SIGNED\_CERT  
[Afficher le certificat](#)

**Retour (recommandé)**

**Accepter le risque et poursuivre**

On surfe maintenant de manière sécurisée sur le site.  
Pour aller plus loin sur le sujet, voir la [documentation](#) très complète d'**Apache** à ce sujet.

## 7.6 Scripts CGI

Les scripts **CGI** ne sont généralement pas activés par défaut, dans le fichier **httpd.conf** il faudra sans doute décommenter les lignes

**LoadModule cgid\_module modules/mod\_cgid.so**

**AddHandler cgi-script .cgi .pl**

j'ai rajouté l'extension **.pl** pour les scripts écrits en **perl**.

on y trouve notamment la ligne qui indique où trouver les scripts:

**ScriptAlias /cgi-bin/ /usr/local/apache2/cgi-bin/**

Pour que les scripts **CGI** soient exécutés dans les répertoires accessibles via **HTTP**, il faudra rajouter l'option **ExecCGI** comme ceci

```
<Directory "/monrepertoire">  
  Options Indexes FollowSymLinks Includes ExecCGI  
  AllowOverride All  
  Require all granted  
</Directory>
```

on relance bien sûr **apache** en cas de modification du fichier **httpd.conf**

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/apache2/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:

Nom:

Email:

Commentaire:

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

## 7.7 PHP et LDAP

On peut compiler **PHP** pour pouvoir utiliser des commandes gérant une base **LDAP**, pour cela à la compil 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**. Recompiliez puis réinstallez le module qui va bien :

**make**

Et en tant que **root**

**make install**

Relancez **Apache**

**systemctl restart httpd**

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:

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

## 8 Gestion de bases de données avec MariaDB

### 8.1 Création des bases de test

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

```
/usr/local/mysql/bin/mysql -u olivier -p  
Enter password:
```

```
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 25  
Server version: 10.8.3-MariaDB Source distribution
```

```
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
```

```
Reading table information for completion of table and column names
```

```
You can turn off this feature to get a quicker startup with -A
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
MariaDB [(none)]>
```

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

```
MariaDB [(none)]> SHOW DATABASES;
```

```
+-----+  
| Database      |  
+-----+  
| information_schema |  
| mysql         |
```



```
| performance_schema |
| test                |
+-----+
4 rows in set (0,00 sec)
```

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

```
MariaDB [(none)]> CREATE DATABASE essai;
Query OK, 1 row affected (0.00 sec)
```

On va utiliser maintenant cette base de donnée

```
MariaDB [(none)]> USE essai
Database changed
```

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

```
MariaDB [(essai)]> SHOW TABLES;
Empty set (0.00 sec)
```

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

```
MariaDB [(essai)]> 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:

```
MariaDB [(essai)]> SHOW TABLES;
+-----+
| Tables in essai |
+-----+
| coord           |
+-----+
1 row in set (0.00 sec)
```

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

```
MariaDB [(essai)]> 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+MariaDB**, on créera la table suivante:

```
MariaDB [(essai)]> 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**.

## 8.2 Tests de fonctionnement avec MariaDB

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**.

```

<?php
$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 MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 91
Server version: 10.8.3-MariaDB Source distribution

```

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```

MariaDB [(essai)]> 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 maintenant le script **PHP**.

```

<?php

$page=getenv("HTTP_REFERER");
$ip=getenv( "REMOTE_ADDR");
$host=gethostbyaddr($ip);
$d = date("d/m/Y H:i:s");
$sexpl=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','$sexpl','$page')";
$result=mysql_query($query,$id);

echo "$d $host($ip) $sexpl $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 MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 91
Server version: 10.8.3-MariaDB Source distribution

```

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```

MariaDB [(essai)]> 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 **MariaDB** et **PHP**, laissez libre cours à votre imagination.

## 8.3 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](http://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 zip qu'on décompresse en tapant :

```
unzip phpMyAdmin-5.2-all-languages.zip
```

Cela va créer dans le répertoire de travail un répertoire **phpMyAdmin-5.2-all-languages**. Dans ce répertoire vous avez un fichier **config.inc.sample.php**, il faut le copier en **config.inc.php**, dans ce fichier il faut modifier les champs suivants :

Pour cette variable on peut mettre un peu n'importe quoi pourvu que ça fasse 32 octets ! En mettant une ligne de 32 caractères ça semble passer, ça sert ensuite pour chiffrer de manière aléatoire.

```
$cfg['blowfish_secret'] = 'je suis toto et il faut que la phrase soit suffisamment longue'; /* YOU MUST FILL IN THIS FOR COOKIE AUTH! */
```

j'ai défini également cette variable

```
$cfg['DefaultLang'] = 'fr';
```

pour le reste j'ai laissé les valeurs par défaut à savoir

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-5.2-all-languages/index.php**
- (solution préconisée), créer un hôte virtuel pointant vers **./phpMyAdmin-5.2-all-languages** qu'on appellera **www.sql.breizland.bz**.

Si vous avez l'erreur

```
mysqli_real_connect(): (HY000/2002): No such file or directory
```

dans le fichier de configuration il faudra remplacer localhost par 127.0.0.1 comme ceci

```
$cfg['Servers'][$i]['host'] = '127.0.0.1';
```

si vous avez maintenant l'erreur

```
$cfg['TempDir'] (/tmp/) n'est pas accessible. phpMyAdmin est incapable de mettre en cache les modèles et de ce fait sera lent.
```

dans le fichier **config.inc.php** il faudra définir la variable

```
$cfg['TempDir'] = '/tmp';
```

Pour cette erreur

Créer une base de données nommée « phpmyadmin » et la configuration du stockage de phpMyAdmin dans cette base.

dans le fichier de configuration **config.inc.php** on décommentera les lignes suivantes

```
/* Storage database and tables */
$cfg['Servers'][$i]['pmadb'] = 'phpmyadmin';
$cfg['Servers'][$i]['bookmarktable'] = 'pma__bookmark';
$cfg['Servers'][$i]['relation'] = 'pma__relation';
$cfg['Servers'][$i]['table_info'] = 'pma__table_info';
$cfg['Servers'][$i]['table_coords'] = 'pma__table_coords';
$cfg['Servers'][$i]['pdf_pages'] = 'pma__pdf_pages';
$cfg['Servers'][$i]['column_info'] = 'pma__column_info';
$cfg['Servers'][$i]['history'] = 'pma__history';
$cfg['Servers'][$i]['table_uiprefs'] = 'pma__table_uiprefs';
$cfg['Servers'][$i]['tracking'] = 'pma__tracking';
$cfg['Servers'][$i]['userconfig'] = 'pma__userconfig';
$cfg['Servers'][$i]['recent'] = 'pma__recent';
$cfg['Servers'][$i]['favorite'] = 'pma__favorite';
$cfg['Servers'][$i]['users'] = 'pma__users';
$cfg['Servers'][$i]['usergroups'] = 'pma__usergroups';
$cfg['Servers'][$i]['navigationhiding'] = 'pma__navigationhiding';
$cfg['Servers'][$i]['savedsearches'] = 'pma__savedsearches';
$cfg['Servers'][$i]['central_columns'] = 'pma__central_columns';
$cfg['Servers'][$i]['designer_settings'] = 'pma__designer_settings';
$cfg['Servers'][$i]['export_templates'] = 'pma__export_templates';
```

Dans la console SQL en bas de l'écran on créera la base phpmyadmin

```
CREATE DATABASE IF NOT EXISTS `phpmyadmin`
  DEFAULT CHARACTER SET utf8 COLLATE utf8_bin;
USE phpmyadmin;
```

CTRL+enter pour exécuter

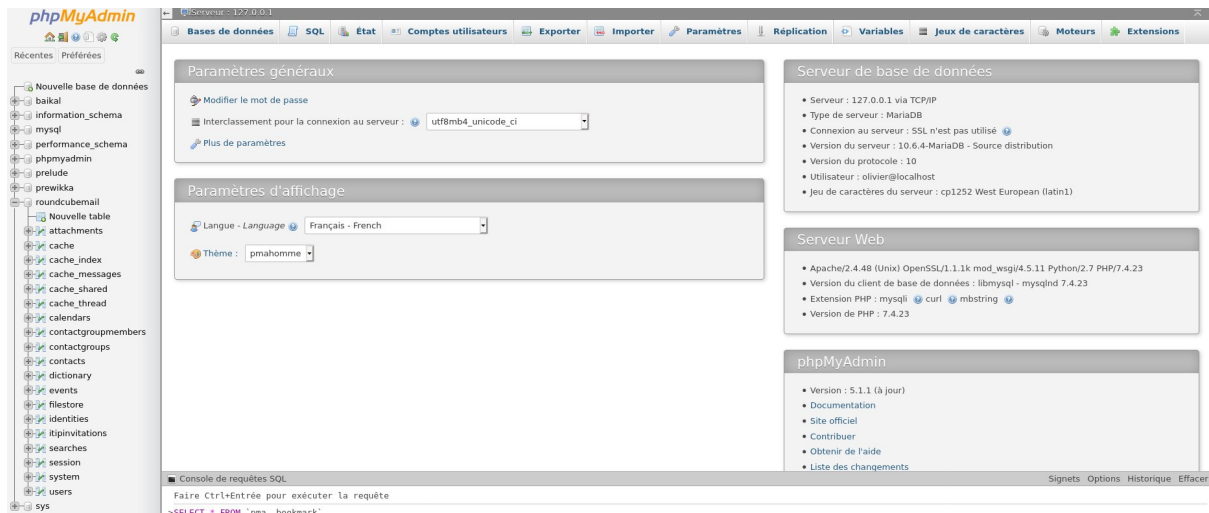
On sélectionne la base **phpmyadmin** à gauche et toujours dans la console on créera les tables qu'on trouvera dans le fichier **phpMyAdmin-5.2-all-languages/sql/create\_tables.sql**

```
CREATE TABLE IF NOT EXISTS `pma__bookmark` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `dbase` varchar(255) NOT NULL default "",
  `user` varchar(255) NOT NULL default "",
  `label` varchar(255) COLLATE utf8_general_ci NOT NULL default "",
  `query` text NOT NULL,
  PRIMARY KEY (`id`)
)
  COMMENT='Bookmarks'
  DEFAULT CHARACTER SET utf8 COLLATE utf8_bin;
```

CTRL+enter pour exécuter, etc.

**NOTE** Si ça vous gêne que n'importe qui d'un navigateur puisse aller dans le répertoire **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.

A noter que si vous avez une erreur de chargement de **phpMyAdmin** du style

**Error during session start; please check your PHP and/or webserver log file and configure your PHP installation properly. Also ensure that cookies are enabled in your browser.**

**session\_start(): open(SESSION\_FILE, O\_RDWR) failed: No such file or directory (2)**

**session\_start(): Failed to read session data: files (path: )**

il faudra éditer le fichier **/usr/local/apache2/conf/php.ini** et modifier la ligne suivante

**session.save\_path = "/tmp"**

vous mettez le répertoire temporaire que vous voulez.

## 8.4 Connexion sécurisée entre le serveur MariaDB et phpMyAdmin

Maintenant pour une connexion sécurisée on rajoutera dans le fichier `config.inc.php`

```
$cfg['Servers'][$i]['ssl'] = true;
$cfg['Servers'][$i]['ssl_key'] = '/etc/mysql/ssl/client-key.pem';
$cfg['Servers'][$i]['ssl_cert'] = '/etc/mysql/ssl/client-cert.pem';
$cfg['Servers'][$i]['ssl_ca'] = '/etc/mysql/ssl/ca-cert.pem';
$cfg['Servers'][$i]['ssl_ca_path'] = '/etc/mysql/ssl/';
$cfg['Servers'][$i]['ssl_verify'] = 'true';
$cfg['Servers'][$i]['ssl_ciphers'] = 'ECDHE-RSA-AES256-GCM-SHA384';
```

il faudra sans doute rajouter également le nom du serveur tel que défini plus haut.

```
$cfg['Servers'][$i]['host'] = '127.0.0.1';
```

```
$cfg['Servers'][$i]['host'] = 'mariadb-serveur';
```

ça évitera ce genre d'erreur

```
mysql::real_connect(): Peer certificate CN='mariadb-serveur' did not match expected CN='127.0.0.1'
```

et voilà quand on va se connecter, voilà ce que ça va donner dans les propriétés de **MariaDB**



Il faudra éviter également un autre piège avec les connexions sécurisées si votre utilisateur est déclaré comme étant `user@localhost` dans la base `mysql`, vous n'arriverez pas à vous connecter avec `user@mariadb-serveur`, il faudra créer un nouvel utilisateur `user@mariadb-serveur`.

## 9 Analyser les logs d'Apache avec awstats

### 9.1 Présentation

**Awstat** est un outil pour analyser les logs d'**apache**, il est plus riche que **webalizer**, qui est un projet mort vous pouvez le trouver à l'URL [awstats.sourceforge.net](http://awstats.sourceforge.net).

### 9.2 Installation et configuration

On décompresse l'archive en tapant :

```
tar xvzf awstats-7.8.tgz
```

Cela va nous donner le répertoire **awstats-7.8**. 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

```
systemctl restart httpd.service
```

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

```
mkdir /etc/awstats
```

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

```
cd awstats-7.8/tools
```

Dans le cas d'une mise à jour on tapera

```
perl awstats_updateall.pl now -awstatsprog=/chemin-absolu/awstats-7.8/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-7.8/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-7.8/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-7.8/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-7.8/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, avec 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 20140126) (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-7.8`

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'
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-7.8/wwwroot/classes/"'
Add 'Alias /awstatscss "/usr/local/linux/web/awstats-7.8/wwwroot/css/"'
Add 'Alias /awstatsicons "/usr/local/linux/web/awstats-7.8/wwwroot/icon/"'
Add 'ScriptAlias /awstats/ "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/"'
Add '<Directory>' directive
AWStats directives added to Apache config file.
```

Il met à jour son propre fichier de configuration

```
----> Update model config file '/usr/local/linux/web/awstats-7.8/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] ? y
```

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'
Redirecting to /bin/systemctl restart httpd.service
```

----> 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-7.8/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-7.8/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 machines 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"
```

Pour lutter contre les [spam referrers](#), vous devez activer la variable suivante

```
SkipReferrersBlackList="/etc/awstats/blacklist.txt"
```

avec un fichier `blacklist.txt` qu'on récupérera sous `awstats-7.8/wwwroot/cgi-bin/lib`

Par ici on trouvera une blacklist nettement plus récente <https://perishablepress.com/blacklist/ultimate-referrer-blacklist.txt>

dans ce fichier d'après mes tests il semblerait que la première partie qui commence par des `RewriteCond` ne serve à rien pour `awstats`, elle n'est utile que si vous filtrez les SPAM referrers directement à l'accès sur votre

site avec un **.htaccess** attention cette méthode aura pour conséquence de ralentir l'accès à votre site. Voir par [ici](#) une discussion intéressante sur le sujet.

Ce n'est que la seconde partie qui est réellement utile et qui fonctionne avec **awstats**, elle commence par

```
# This is the URL blacklist from the chongqed.org database
# it is available from http://blacklist.chongqed.org/
# You can use each line below as a regular expression
# that can be tested against URLs on your wiki.
# The last spammer was added on 2008-09-11 10:14:51
# Check http://blacklist.chongqed.org/ for updates
```

J'ai commencé à compléter la liste par les lignes suivantes

```
https?:\V([^\V]*\.)?proxtrail\.ru
https?:\V([^\V]*\.)?denterum\.ru
https?:\V([^\V]*\.)?profeservice\.ru
https?:\V([^\V]*\.)?bazakanstovarov\.com
https?:\V([^\V]*\.)?balkanfarma\.ru
https?:\V([^\V]*\.)?sobervoditel\.ru
https?:\V([^\V]*\.)?avtokor-23\.ru
https?:\V([^\V]*\.)?avtokor23\.ru
https?:\V([^\V]*\.)?xn--j1at1a.xn--p1ai
https?:\V([^\V]*\.)?rupolitshow\.ru
https?:\V([^\V]*\.)?vyezd-viezd\.ru
https?:\V([^\V]*\.)?igru-2015\.ru
https?:\V([^\V]*\.)?jeribetejewu\.c0\.pl
https?:\V([^\V]*\.)?creditservice\.ru
https?:\V([^\V]*\.)?megamashiny\.com
https?:\V([^\V]*\.)?seoxbeep\.ru
https?:\V([^\V]*\.)?woman3050\.ru
https?:\V([^\V]*\.)?www.vselgoty\.ru
https?:\V([^\V]*\.)?playpokeronline\.dk
https?:\V([^\V]*\.)?medical\.in\.ua
https?:\V([^\V]*\.)?viagralevitradzheneriki\.ru
https?:\V([^\V]*\.)?\.\com\.ua
https?:\V([^\V]*\.)?tasgroup\.it
https?:\V([^\V]*\.)?\.\co\.ua
https?:\V([^\V]*\.)?kiev\.ua
https?:\V([^\V]*\.)?bringtwo\.net
https?:\V([^\V]*\.)?cleaningservices\.kiev\.ua
https?:\V([^\V]*\.)?studio-topkapi\.eu
https?:\V([^\V]*\.)?kruchen\.com\.ua
https?:\V([^\V]*\.)?freedom\.co\.ua
https?:\V([^\V]*\.)?bringtwo\.net
https?:\V([^\V]*\.)?studio-topkapi\.eu
https?:\V([^\V]*\.)?c0\.pl
https?:\V([^\V]*\.)?bazakanstovarov\.com
https?:\V([^\V]*\.)?xikiz\.com
https?:\V([^\V]*\.)?lesbianmilf\.xblog\.in
https?:\V([^\V]*\.)?pamyatniki-in-kiev\.com\.ua
https?:\V([^\V]*\.)?carivka\.com\.ua
https?:\V([^\V]*\.)?agent-05\.su
https?:\V([^\V]*\.)?websolution\.com\.ua
https?:\V([^\V]*\.)?pamjatnik\.com\.ua
https?:\V([^\V]*\.)?startimes\.com
https?:\V([^\V]*\.)?arktech\.co
https?:\V([^\V]*\.)?goohey\.com
https?:\V([^\V]*\.)?himalayan-imports\.com
https?:\V([^\V]*\.)?pizza-imperia\.com
```

```
https?:\V{[^V]*\.}?nowellgroup\.com
https?:\V{[^V]*\.}?ddrgame\.com
https?:\V{[^V]*\.}?torrinomedica\.it
https?:\V{[^V]*\.}?oliveriobalcells\.com
https?:\V{[^V]*\.}?giocagiocagioca\.com
```

Ça m'a vite saoulé parce que tous les jours il y a des nouveaux sites qui font leur apparition, le plus simple est de bloquer carrément certaines extensions !

```
^https?://[^\.]+\.\ru
^https?://[^\.]+\.\ua
^https?://[^\.]+\.\su
^https?://[^\.]+\.\link
^https?://[^\.]+\.\cc
^https?://[^\.]+\.\in
```

là on va bloquer tous les sites en **.ru**, en **.ua**, etc. Tant qu'à faire comme ce n'est pas forcément intéressant de voir les referrers de son propre domaine, on les bloque également

```
https?:\V{[^V]*\.}?hoarau\.org
https?:\V{[^V]*\.}?funix\.org
```

la liste des referrers est maintenant assainie et enfin exploitable.

Pour que ça soit pris en compte avec **awstats** il faut effacer ce qu'il y a sous **/usr/local/var/awstats/** et relancer **awstats** pour tous les fichiers de log hebdo

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-7.8/wwwroot/cgi-bin** et on tape

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

Voilà le résultat

```
Create/Update database for config "/etc/awstats/awstats.www.online.kervao.fr.conf" by AWStats version
7.8 (build 20161204)
From data in log file "/usr/local/apache2/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/index.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...).

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.

## 9.3 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-7.8/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-7.8/wwwroot/cgi-bin/awstats.pl
```

```
/usr/local/linux/web/awstats-7.8/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-
7.8/wwwroot/cgi-bin/awstats.pl
```

L'option **-dir** permet de sauvegarder sous forme html les statistiques obtenus dans le répertoire qui va bien. 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-7.8/wwwroot/icon/"
```

```
<Directory "/usr/local/linux/web/awstats-7.8/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-7.8/
```

Voilà ce que ça donne quand on lance le script **/etc/cron.daily/awstats**

```
Launch update process : "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -update -configdir=
Build main page: "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output
Build alldomains page: "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=alldomains
Build allhosts page: "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=allhosts
```

```

Build    lasthosts    page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=lasthosts
Build    unknownnip    page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=unknownnip
Build    allrobots     page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=allrobots
Build    lastrobots    page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=lastrobots
Build    session       page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=session
Build    urldetail     page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=urldetail
Build    urlentry      page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=urlentry
Build    urlexit       page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=urlexit
Build    osdetail      page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=osdetail
Build    unknownos    page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=unknownos
Build    browserdetail page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=browserdetail
Build    unknownbrowser page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=unknownbrowser
Build    referer       page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=referer
Build    refererpages  page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=refererpages
Build    keyphrases   page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=keyphrases
Build    keywords     page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=ohoarau.homelinux.org:1234 -staticlinks -lang=fr -output=keywords
Build    errors404    page:    "/usr/local/linux/web/awstats-7.8/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-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -update -configdir=
Build   main       page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output
Build   alldomains page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=alldomains
Build   allhosts   page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=allhosts
Build   lasthosts  page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=lasthosts
Build   unknownnip page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=unknownnip
Build   allrobots  page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=allrobots
Build   lastrobots page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=lastrobots
Build   session    page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=session
Build   urldetail  page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=urldetail
Build   urlentry   page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=urlentry
Build   urlexit    page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=urlexit
Build   osdetail   page:    "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -

```

```

config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=osdetail
Build unknownnos page: "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=unknownnos
Build browserdetail page: "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=browserdetail
Build unknownbrowser page: "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=unknownbrowser
Build refererse page: "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=refererse
Build refererpages page: "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=refererpages
Build keyphrases page: "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=keyphrases
Build keywords page: "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
config=funix.homelinux.org:1234 -staticlinks -lang=fr -output=keywords
Build errors404 page: "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl" -
20 files built.
Main HTML page is 'awstats.funix.homelinux.org:1234.html'.

```

si vous avez une erreur du style

```

bash: /usr/local/linux/web/awstats-7.8/tools/awstats_buildstaticpages.pl: /usr/bin/perl^M: bad interpreter:
Aucun fichier ou répertoire de ce type

```

il suffit d'éditer le fichier `awstats_buildstaticpages.pl` et de virer le `^M` après `perl` (avec `mcedit` par exemple).

maintenant si vous avez l'erreur suivante `awstats.pl permission non accordée`, c'est que les droits d'`awstats` ne sont pas bon. Il faut le mettre à 755 comme ceci

```

chmod 755 /usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl

```

Maintenant sous les répertoires `/usr/local/apache2/htdocs/awstats/online` et `/usr/local/apache2/htdocs/awstats/ohoarau` on se 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. J'ai dû créer également la directive suivante pour les icônes soient visibles

```

<Directory "/usr/local/linux/web/awstats-7.8/wwwroot/icon">
  Options Indexes FollowSymLinks Includes ExecCGI
  AllowOverride All
  Require all granted
</Directory>

```

Voilà quelques aperçus (non exhaustifs)



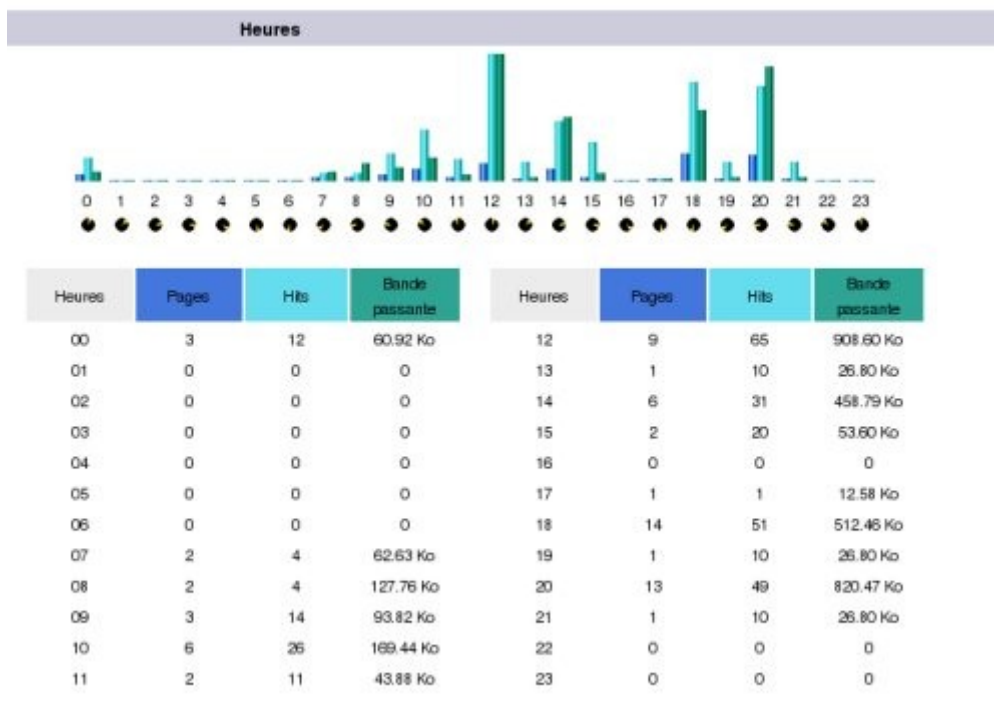
**Dernière mise à jour:** 06 Jan 2005 - 04:03  
**Période d'analyse:** Mois Jan 2005



**Quand:** Historique mensuel Jours du mois Jours de la semaine Heures  
**Où:** 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/Réfere:** Origine de la connexion Moteurs de recherche Sites référenciers Recherche Phrases clés Mots clés  
**Autres:** Divers Codes Status HTTP Pages non trouvées

Résumé					
<b>Période d'analyse</b>	Mois Jan 2005				
<b>Première visite</b>	03 Jan 2005 - 10:09				
<b>Dernière visite</b>	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.



Visiteurs Robots/Spiders (Top 10) - Liste complète - Dernière visite			
3 robots différents*			
	Hits	Bandwidth	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: 25 - Moyenne: 330 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	Bandwidth	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	99	30.9%	976.67 Ko	28.9%

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			
	Aspateur	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	4	6.8%	4	1.6%
- Google 2 2				
- Yahoo 2 2				
Lien depuis une page externe (autres sites, hors moteurs) - Liste complète	31	53.4%	217	88.9%
- http://snix.hornelinux.org/fr/linux/main-linux.php3 13 80				

Pour pouvoir faire les stats d'un site que vous n'hébergez pas localement, c'est par [ici](#).

Inconvénient de cette méthode les pages construites sont statiques et on ne peut pas accéder à l'historique des mois précédents. Il faudra passer par l'extension CGI d'Apache qu'il faudra activer, puis on rajoute les lignes suivantes dans le fichier **httpd.conf**

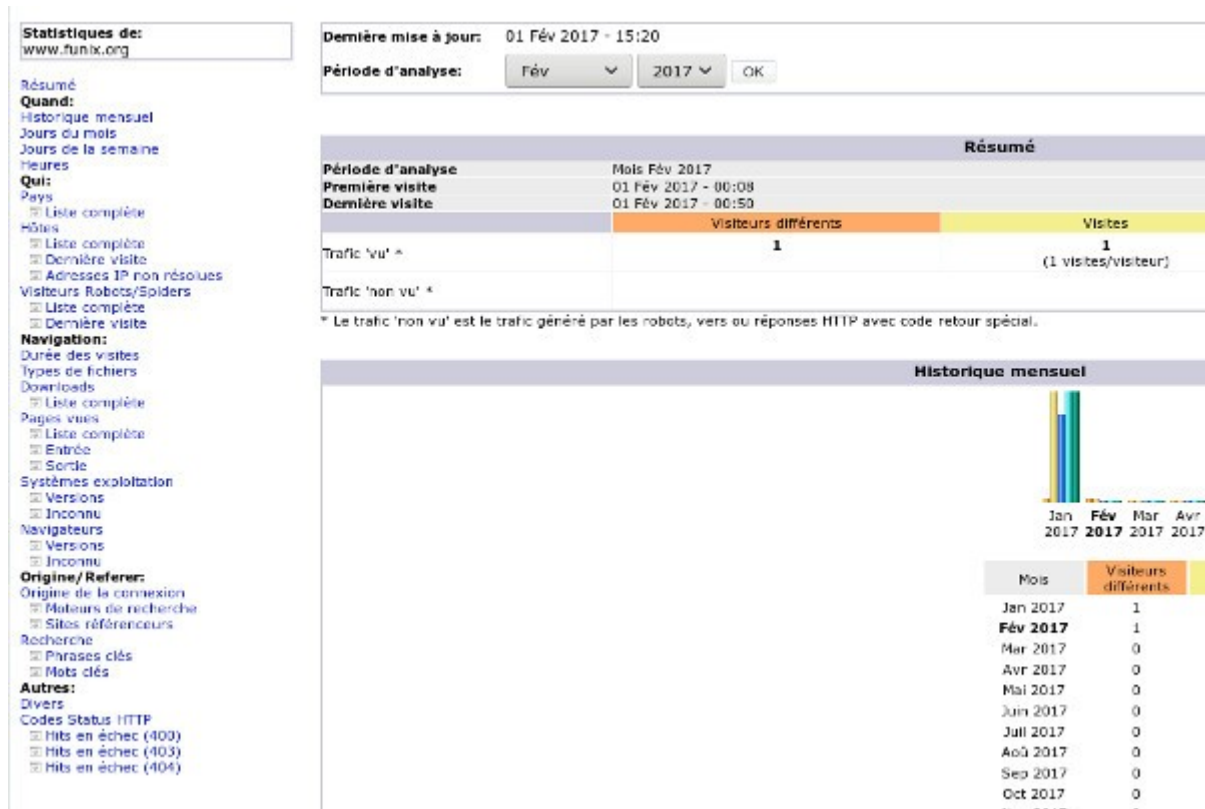
**Alias /awstatsclasses "/usr/local/linux/web/awstats-7.8/wwwroot/classes/"**

```
Alias /awstatscss "/usr/local/linux/web/awstats-7.8/wwwroot/css/"
Alias /awstatsicons "/usr/local/linux/web/awstats-7.8/wwwroot/icon/"
ScriptAlias /awstats/ "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/"
```

```
<Directory "/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin">
  Options ExecCGI
  AllowOverride None
  Order allow,deny
  Allow from all
</Directory>
```

```
<Directory "/usr/local/linux/web/awstats-7.8/wwwroot">
  Options Indexes FollowSymLinks Includes ExecCGI
  AllowOverride All
  Require all granted
</Directory>
```

Le site sera accessible en tapant comme URL <http://monserveurweb/awstats/awstats.pl?config=funix.homelinux.org:1234> et voilà le résultat



on retrouve un bandeau de menu à gauche et on peut choisir le mois d'analyse avec un menu type drop down.

## 9.4 Analysez les logs d'un site hébergé non localement

Pour être concret, j'ai mon site [funix.org](http://funix.org) qui est hébergé chez [online](http://online), j'ai moyen de récupérer les logs d'Apache bruts, chaque jour un fichier log est créé et se retrouve dans une arborescence année/mois/jour\_access.log.gz. 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/apache2/logs/funix.org
```

```
#!/bin/bash
# construction du format de la date
jour=`date +%e`
```

```

mois=`date +%m`
annee=`date +%Y`
mois31="mai juillet octobre décembre"
mois30="février avril juin septembre novembre"
# ce if est là quand on bascule au mois suivant, on doit prendre le dernier jour du mois précédent
if [ $jour -eq "1" ];
then
    moisb=`date +%B`
    for i in $mois31
    do
        if [ $moisb == $i ]
        then
            hier=30
            mois=`expr $mois - 1`
        fi
    done
    for i in $mois30
    do
        if [ $moisb == $i ]
        then
            hier=31
            mois=`expr $mois - 1`
        fi
    done
    if [ $moisb == "août" ]
    then
        hier=31
        mois=07
    fi
    if [ $moisb == "mars" ]
    then
        hier=28
        mois=02
    fi
    if [ $moisb == "janvier" ]
    then
        hier=31
        mois=12
        annee=`expr $annee - 1`
    fi
else
    hier=`expr $jour - 1`
fi
# le mois et le jour doivent être en 2 chiffres, je rajoute un zéro s'il le faut
if [ "${#mois}" = 1 ]
then mois="0"$mois
fi
if [ "${#hier}" = 1 ]
then hier="0"$hier
fi
# je me déplace dans le répertoire qui va bien et je supprime les vieux fichiers log
cd /usr/local/apache2/logs/funix.org
rm -f /usr/local/apache2/logs/funix.org/*.log
# récupération des logs par ftp
wget -c ftp://stats.online.net/archives/$annee/$mois/"$hier"_access.log.gz --user=webmaster@funix.org --
password=motdepasse
# je décompresse l'archive et la renomme
gunzip "$hier"_access.log.gz
mv "$hier"_access.log www.funix.org.log

```

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

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

```
SiteDomain="www.funix.org"
```

```
HostAliases="www.funix.org"
```

Mon fichier brut de log ressemble à ça

```
www.funix.org:80 109.8.158.150 - - [01/Jan/2017:01:02:29 +0100] "GET /fr/unix/awk.htm HTTP/1.0" 200 8711 "https://www.google.fr/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0"
www.funix.org:80 109.8.158.150 - - [01/Jan/2017:01:02:29 +0100] "GET /favicon.ico HTTP/1.0" 404 468 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0"
www.funix.org:80 109.8.158.150 - - [01/Jan/2017:01:02:29 +0100] "GET /favicon.ico HTTP/1.0" 404 468 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:50.0) Gecko/20100101 Firefox/50.0"
www.funix.org:80 163.172.66.155 - - [01/Jan/2017:01:02:52 +0100] "GET /fr/linux/main-linux.php?ref=xinetd&page=full HTTP/1.0" 200 5751 "-" "Mozilla/5.0 (compatible; AhrefsBot/5.2; +http://ahrefs.com/robot)"
www.funix.org:80 86.252.235.82 - - [01/Jan/2017:01:04:29 +0100] "GET /fr/linux/main-linux.php?ref=video&page=menu HTTP/1.0" 200 14052 "https://www.google.fr/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36"
www.funix.org:80 86.252.235.82 - - [01/Jan/2017:01:04:29 +0100] "GET /images/funix-icone.gif HTTP/1.0" 200 4320 "http://www.funix.org/fr/linux/main-linux.php?ref=video&page=menu" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36"
www.funix.org:80 86.252.235.82 - - [01/Jan/2017:01:04:29 +0100] "GET /images/journal.gif HTTP/1.0" 200 354 "http://www.funix.org/fr/linux/main-linux.php?ref=video&page=menu" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36"
```

par défaut il y aura toujours qu'un visiteur unique par jour ! J'ai donc dû adapter le format de log avec la variable **LogFormat**

```
LogFormat="%other %host - - %time1 %methodurl %code %bytesd %refererquot %uaquot"
```

Cela signifie qu'on laisse tomber le premier champ et on prend en compte que les suivants.

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

```
/usr/local/linux/web/awstats-7.8/tools/awstats_buildstaticpages.pl -config=www.funix.org -update -lang=fr -dir=/usr/local/apache2/htdocs/awstats/funix.org -awstatsprog=/usr/local/linux/web/awstats-7.8/wwwroot/cgi-bin/awstats.pl
```

Les fichiers résultats vont se retrouver sous `/usr/local/apache2/htdocs/awstats/funix.org`, il m'a semblé utile dans ce répertoire de créer le lien suivant

```
ln -s awstats.www.funix.org.html index.html
```

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

Pour que tout ça soit visible de mon serveur **Apache** local, j'ai rajouté les alias suivants dans le fichier `/usr/local/apache2/conf/httpd.conf`

```
Alias /awstats-funix-org "/usr/local/apache2/logs/funix.org/"
```

et pour que les répertoires ci-dessus soient accessibles

```
<Directory "/usr/local/apache2/logs/funix.org/">  
  Options Indexes FollowSymLinks Includes ExecCGI  
  AllowOverride All  
  Require all granted  
</Directory>
```

```
<Directory "/usr/local/apache2/htdocs/awstats/funix.org/">  
  Options Indexes FollowSymLinks Includes ExecCGI  
  AllowOverride All  
  Require all granted  
</Directory>
```

on relance Apache dans la foulée

```
systemctl restart httpd.service
```

avec la méthode par CGI mes stats sont accessibles à partir de l'URL <http://monserveur/awstats/awstats.pl?config=www.funix.org>

# 10 Installer un moteur de recherche avec Ht://dig

## 10.1 Présentation

**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. C'est un fork de l'outil **Ht://Dig** qui n'est plus maintenu.

Dans la suite de ma page je présente un site intranet ayant pour adresse [asterix.kervao.fr](http://asterix.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.

## 10.2 Installation

On récupérera les sources à l'adresse <https://github.com/solbu/hldig>, elle se présente sous la forme d'un tarball qu'on décompressera en tapant:

```
tar xvfz hldig-1.0.2.tar.gz
```

Cela va créer un répertoire **hldig-1.0.2/** dans lequel on tapera **configure** avec les options suivantes :

```
./configure --with-apache=chemin-apache --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/apache2/htdocs/htdig** :

```
./configure --prefix=/usr/local/htdig --with-apache=/usr/local/apache2  
--with-cgi-bin-dir=/usr/local/apache2/cgi-bin --with-image-dir=/usr/local/apache2/htdocs/hldig --with-  
search-dir=/usr/local/apache2/htdocs/hldig
```

Puis

**make**

Et enfin en tant que root

**make install**

On rajoutera la ligne

**/usr/local/htdig/lib/hldig**

dans le fichier **/etc/ld.so.conf** et on tapera **ldconfig**

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

## 10.3 Configuration

On trouvera le fichier de configuration sous **/usr/local/hldig/etc/hldig**, il se nomme **hldig.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 !
database_dir:      /usr/local/hldig/var/hldig
```

```
# 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://asterix.kervao.fr/
```

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

```
# This attribute is used for compressing the database. The default is to
# set it to the same as the limit_urls_to above, plus some common endings.
#
# Keep in mind that this list should be short. If your start_url is a very
# long list of URLs, it may be wise to replace it with something like
# http://www. or comment this out and use the compiled-in default.
#
common_url_parts:  ${limit_urls_to} .html .htm .shtml
```

```
# 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:    
no_next_page_text:
prev_page_text:    
no_prev_page_text:
page_number_text:  '<img src="/www/button1.gif" border="0" align="middle" width="30"

```







Connections closed : 9  
Changes of server : 0  
HTTP Requests : 722  
HTTP KBytes requested : 4948.29  
HTTP Average request time : 0.0138504 secs  
HTTP Average speed : 494.829 KBytes/secs

**hldig End Time: Sat Apr 11 12:02:59 2020**

On crée à présent l'index en tapant :

```
/usr/local/hldig/bin/hlmerge -v
```

Suivant la taille du site, la commande peut être plus ou moins longue. Dans le fichier de configuration d'**Apache**, j'ai rajouté la ligne suivante

```
Alias /www "/usr/local/apache2/htdocs/hldig"
```

Pour que les images s'affichent correctement. Maintenant il faut créer un script lancé régulièrement par **cron** qui met à jour régulièrement la base de donnée et l'index.

```
#!/bin/bash
```

```
/usr/local/hldig/bin/hldig  
/usr/local/hldig/bin/hlmerge
```

## 10.4 Utilisation

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

**URL: <http://asterix.kervao.fr/hldig/search.html>**

---

 **WWW Site Search**

---

This search will allow you to search the contents of all the publicly available WWW documents at this site.

Match:  Format:  Sort by:

Search:

---

[Powered by hldig \\$\(VERSION\)](#)

Voilà le résultat de la recherche



## Search results for 'mariadb'

Match:  Format:  Sort by:   
 Refine search:

Documents 1 - 10 of 12 matches. More ★'s indicate a better match.

### [FUNIX: Section Linux](#)★★★★

... - Piloter une webcam Divers - Chercher les mariens avec SETI@home - Piloter un PDA pocket PC Apache + PHP + MySQL [Présentation | Installation de MariaDB ( Installation par compilation , configuration , initialisation des bases de données , lancement automatique par systemd , finalisation de la ...  
[/linux/index.php?ref=apache2](#) 04/11/20, 115113 bytes

### [Apache + PHP + MySQL](#)★★★★

[Présentation | Installation de MariaDB ( Installation par compilation , configuration , initialisation des bases de données , lancement automatique par systemd , finalisation de la configuration , installation d'une connexion chiffée avec MariaDB ) | Installation d'Apache ( Compilation d ...  
[/linux/apache2.htm](#) 04/11/20, 103538 bytes

### [FUNIX: Section Linux](#)★★★

... | Mise en place des utilisateurs | Création d'une base de données | Perte du mot de passe administrateur | Dernière modif 11 avril 2020 MARIADB Manips diverses Présentations Cette n'a pas pour but de vous présenter ce qu'est un SGBD et encore moins de vous expliquer le langage SQL mais de ...  
[/linux/index.php?ref=mysql](#) 04/11/20, 26325 bytes

### [MySQL](#)★

... | Mise en place des utilisateurs | Création d'une base de données | Perte du mot de passe administrateur | Dernière modif 11 avril 2020 MARIADB Manips diverses Présentations Cette n'a pas pour but de vous présenter ce qu'est un SGBD et encore moins de vous expliquer le langage SQL mais de ...  
[/linux/mysql.htm](#) 04/10/20, 14750 bytes

### [FUNIX: Section Linux](#)★

... interne de ffmpeg pour éviter les erreurs de compilation avec une version trop récente . J'ai activé l'option MySQL en installant le package mariadb-devel afin que la base de données soit commune et visible de tous partout sur le réseau . Même si le serveur est distant et ne se trouve pas forc ...  
[/linux/index.php?ref=mediacenter](#) 04/11/20, 68182 bytes

### [FUNIX: Section Linux](#)★

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/hldig/share/templates`.