

Table des matières

TABLE DES MATIERES	1
INSTALLATION MARIADB	2
GESTION DES BASES DE DONNEES	3
SECURISER LA MARIADB	5
SAUVEGARDER ET RESTAURER LES BASES DE DONNEES	6

Avant-Propos

E6 :

Elaboration de documents relatifs à la production et à la fourniture de services

A1.1.1 , Analyse du cahier des charges d'un service à produire

A1.2.4 , Détermination des tests nécessaires à la validation d'un service

A1.3.4 , Déploiement d'un service

A4.1.3 , Conception ou adaptation d'une base de données

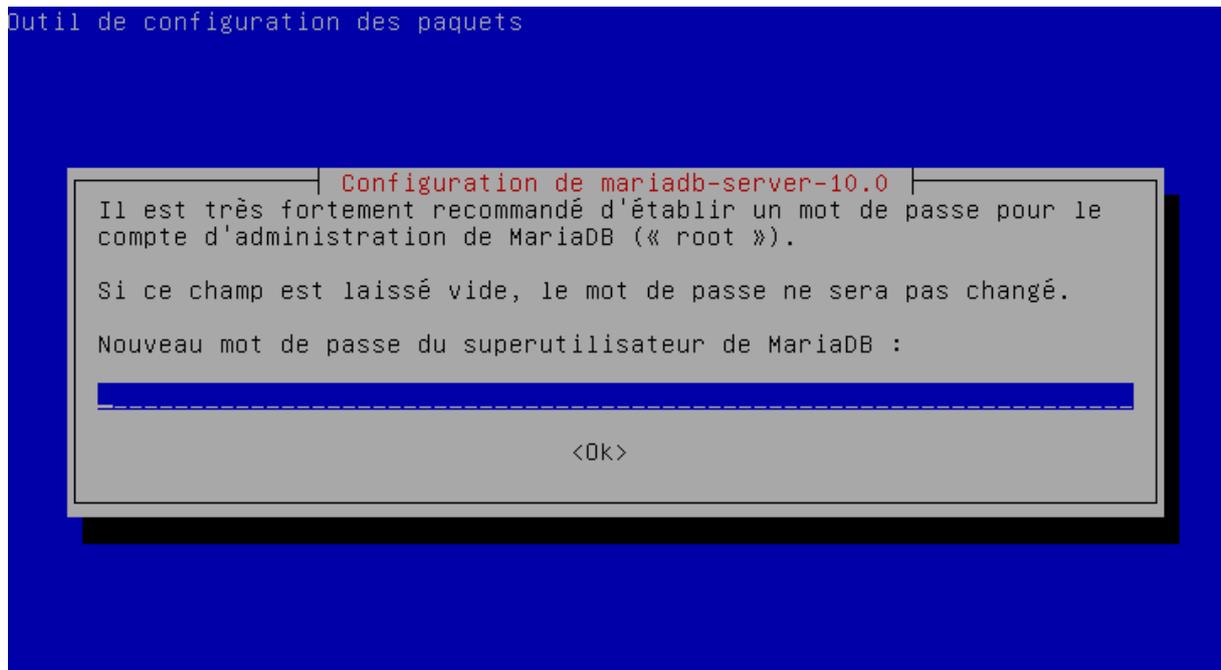
A4.1.9 , Rédaction d'une documentation technique

Installation MariaDB

Pour installer MariaDB, il faut taper la commande apt-get install mariadb-server

```
root@debianDB1:~# apt-get install mariadb-server_
```

Pendant l'installation, on doit rentrer le mot de passe pour l'utilisateur root de la base de données qui aura tous les droits d'accès.



J'ai mis comme mot de passe root.

Une fois que l'installation est terminée, il faut installer Apache2.

```
root@debianDB1:~# apt install apache2_
```

Puis PHP 5.

```
root@debianDB1:~# apt install php5_
```

Maintenant, nous pouvons aller sur un navigateur et rentrer l'adresse IP du serveur, pour vérifier que le service apache et php fonctionnent correctement.



Pour tester le PHP, il faut créer un fichier nommé info.php

```
GNU nano 2.2.6 Fichier : info.php
<?php
phpinfo();
?>
```

Lorsque l'on veut y accéder depuis un navigateur, voilà ce qui s'affiche

PHP Version 5.6.24-0+deb8u1 

System	Linux debianDB1 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt25-2+deb8u3 (2016-07-02) x86_64
Build Date	Jul 26 2016 08:17:13
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-json.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, 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
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine:
 Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies
 with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2016, by Zend Technologies



Si ces deux pages fonctionnent, alors nous pouvons passer à la suite.

Gestion des bases de données

Adminer est une application Web offrant une interface graphique pour le système de gestion de base de données MySQL, réalisée en PHP et distribuée sous licence Apache. Il se présente comme une alternative légère à phpMyAdmin et a pour particularité d'être entièrement contenu dans un seul fichier php. On peut toutefois ajouter un fichier CSS, pour modifier la présentation ; il y en a de nombreux à télécharger gratuitement sur le site.

Pour l'installer, il suffit de taper la commande suivante.

```
root@debianDB1:~# wget https://www.adminer.org/static/download/4.2.5/adminer-4.2.5-mysql-en.php_
```

```
root@debianDB1:~# ls
adminer-4.2.5-mysql-en.php
```

Maintenant, il faut déplacer ce fichier à la racine du serveur web, donc avec les autres pages web. On peut en profiter pour le renommer en adminer.php, c'est beaucoup plus pratique de rentrer ce nom-là dans le navigateur.

On peut maintenant se connecter sur la page web à partir de l'URL suivant

192.168.1.128/adminer.php

Maintenant, on arrive sur une page, pour le moment il suffit de se connecter en root root.

System	MySQL ▾
Server	localhost
Username	root
Password
Database	

Permanent login

Une erreur apparait, il manque un paquet, il faut donc l'installer.

MySQL » Server

No extension

None of the supported PHP extensions (MySQLi, MySQL, PDO_MySQL) are available.

Pour l'installer, il suffit de taper la commande suivante.

```
root@debianDB1:/var/www/html# apt install php5-mysql_
```

Une fois que le paquet est installé, nous devons redémarrer le service apache2, puis tenter de nous reconnecter.

```
root@ftp:/var/www/html# apt-get install php5-mysql_
```

MySQL » Server

Adminer 4.2.5

DB:

SQL command Import Export

Select database

Create new database Privileges Process list Variables Status

MySQL version: 5.5.5-10.0.27-MariaDB-0+deb8u1 through PHP extension MySQLi

Logged as: root@localhost

	Database - Refresh	Collation	Tables	Size - Compute
<input type="checkbox"/>	information_schema	utf8_general_ci	?	?
<input type="checkbox"/>	mysql	latin1_swedish_ci	?	?
<input type="checkbox"/>	performance_schema	utf8_general_ci	?	?

Selected (0)

Drop

Et voilà, nous sommes maintenant connectés sur l'interface.

Sécuriser la MariaDB

Maintenant, nous allons sécuriser MariaDB, il faut rentrer la commande suivante, puis répondre à quelques questions.

```
root@debianDB1:/var/www/html# mysql_secure_installation_
```

La première question nous demande notre mot de passe, celui que nous avons rentré lors de l'installation.

```
Enter current password for root (enter for none): _
```

J'ai donc mis root.

```
You already have a root password set, so you can safely answer 'n'.
Change the root password? [Y/n] _
```

Il nous demande si on veut changer de mot de passe, sachant que l'on en a déjà mis un, il nous conseille de répondre non, c'est ce que nous allons faire.

```
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] _
```

Maintenant on nous demande si on veut retirer les utilisateurs anonymes. Il faut le faire, car on ne doit pas pouvoir se connecter sans mot de passe sur notre base. On répond donc y.

```
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] _
```

Pour désactiver l'accès à distance par l'utilisateur root. On n'y accède qu'en local. Il faut répondre y.

```
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] _
```

Pour supprimer la base de test. Il faut car tout le monde peut y accéder. Il faut répondre y.

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

```
Reload privilege tables now? [Y/n] _
```

Maintenant, il faut encore répondre y. C'est pour que les changements soient pris en compte.

Sauvegarder et restaurer les bases de données

Pour la suite de cette partie, nous allons importer la base de données ProFTPD du test précédent.

Nous allons nous connecter sur la base de données afin de créer la base de données.

```
root@debianDB1:/home/anthony# mysql -u root -p
```

Il faut rentrer la commande suivante afin de la créer.

```
MariaDB [(none)]> create database proftpd  
-> ;  
Query OK, 1 row affected (0.00 sec)
```

Pour la suite, voici une capture d'écran de mon ancienne procédure :

Une fois que c'est fait on peut se connecter sur la base pour voir que tout est bien fait :

```
root@debian8:~# mysql -u root -p
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0.00 sec)
```

Il nous faut maintenant créer la table, qui va contenir nos users.

```
mysql> create database proftpd;
Query OK, 1 row affected (0.01 sec)
```

Il faut ensuite utiliser la commande suivante pour pouvoir l'utiliser :

```
mysql> grant select, insert, update, delete on proftpd.* to 'proftpd'@'localhost'
  identified by 'root';
Query OK, 0 rows affected (0.00 sec)
```

Il faut utiliser ensuite la commande suivante :

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

Maintenant, nous pouvons quitter mysql. Il faut maintenant remplir la table précédemment créée des champs, c'est là que nous allons utiliser le script de notre professeur. Il faut rentrer la commande suivant afin de remplir les champs.

```
root@debian8:~# mysql -u proftpd -p proftpd < /home/antoineh/proftpd_mysql.txt
Enter password:
```

Maintenant, pour vérifier que la base est bien importée, on peut se connecter avec l'utilisateur proftpd root sur adminer et voir le contenu de la base.

```
root@debianDB1:/home/anthony# mysql -u root -p proftpd < /home/anthony/saveproftpd2.sql
Enter password:
```

Login

System	MySQL ▾
Server	localhost
Username	proftpd
Password
Database	

Permanent login

Une fois connecté, on peut vérifier le contenu de la base voir si elle s'est bien remplie.

MySQL » Server » Database: proftpd

Database: proftpd

[Alter database](#) [Database schema](#) [Privileges](#)

Tables and views

Search data in tables (4)

<input type="checkbox"/>	Table	Engine?	Collation?	Data Length?	Index Length?	Data Free?	Auto Increment?	Rows?	Comment?
<input type="checkbox"/>	ftpgroup	MyISAM	latin1_swedish_ci	0	1,024	0		0	Table des groupes ProFTPD
<input type="checkbox"/>	ftpquotalimits	MyISAM	latin1_swedish_ci	0	1,024	0		0	Table des quotas ProFTPD
<input type="checkbox"/>	ftpquotatotal	MyISAM	latin1_swedish_ci	0	1,024	0		0	Table des compteurs des quotas ProFTPD
<input type="checkbox"/>	ftpuser	MyISAM	latin1_swedish_ci	0	1,024	0		1	Table des utilisateurs ProFTPD
	4 in total	InnoDB	latin1_swedish_ci	0	4,096	0			

Selected (0)

Move to other database:

[Create table](#) [Create view](#) [Create materialized view](#)

Maintenant que la base de données est exportée, on peut passer à la suite.

Pour faire une sauvegarde de la base de données, on utilise la commande suivante, ce qui va créer un fichier en .sql

```
root@debianDB1:/home/anthony# mysqldump -u root -proot proftpd > saveproftpd.sql
root@debianDB1:/home/anthony# ls
proftpd_mysql.txt saveproftpd.sql
```

Pour compresser un fichier en .gz il faut utiliser la commande suivante :

```
root@debianDB1:/home/anthony# gzip -9 saveproftpd.sql
root@debianDB1:/home/anthony# ls -l
total 8
-rw-r--r-- 1 anthony anthony 2132 mars 10 2014 proftpd_mysql.txt
-rw-r--r-- 1 root root 1221 oct. 3 11:11 saveproftpd.sql.gz
```

Si l'on veut dézipper le fichier, on peut faire la commande

```
root@debianDB1:/home/anthony# gunzip saveproftpd2.sql.gz _
```

Si l'on veut faire les deux commandes en une, il suffit de mettre en pipe, comme cela :

```
root@debianDB1:/home/anthony# mysqldump -u root -p proftpd > saveproftpd2.sql_1 | gzip -9 saveproftpd2.sql_
```

Il existe aussi une autre syntaxe plus courte : `mysqldump -u root -p proftpd | gzip > proftpd.sql.gz`

Maintenant, nous allons tester la restauration de la base. Pour ce faire, nous devons supprimer la base de données. Il faut se connecter sur mysql et rentrer la commande suivante, ce qui va la supprimer.

```
MariaDB [(none)]> drop database proftpd
-> ;
Query OK, 4 rows affected (0.00 sec)
```

Maintenant, la base de données n'existe plus.

Il faut maintenant la recréer, et oui, ce que l'on a sauvegardé est le contenu de la base. On ne peut pas la recréer directement, on peut juste la remplir avec ce que l'on a sauvegardé. Du coup il faut recréer la base :

```
MariaDB [(none)]> create database proftpd
-> ;
Query OK, 1 row affected (0.00 sec)
```

Puis une fois que cela est fait on injecte le contenu de la sauvegarde dans la base :

```
root@debianDB1:~# mysql -u root -p proftpd < /home/anthony/saveproftpd2.sql_
```

Une fois cela fait, on peut retourner sur le site d'adminer pour voir si la base est remplie.

Select database

[Create new database](#) [Privileges](#) [Process list](#) [Variables](#) [Status](#)

MySQL version: **5.5.5-10.0.27-MariaDB-0+deb8u1** through PHP extension **MySQLi**

Logged as: **root@localhost**

	Database - Refresh	Collation	Tables	Size - Compute
<input type="checkbox"/>	information_schema	utf8_general_ci	?	?
<input type="checkbox"/>	mysql	latin1_swedish_ci	?	?
<input type="checkbox"/>	performance_schema	utf8_general_ci	?	?
<input type="checkbox"/>	proftpd	latin1_swedish_ci	?	?

Selected (0)

Drop

Et quand l'on regarde le contenu de la base :

Database: proftpd

[Alter database](#) [Database schema](#) [Privileges](#)

Tables and views

Search data in tables (4)

<input type="checkbox"/>	Table	Engine?	Collation?	Data Length?	Index Length?	Data Free?	Auto Increment?	Rows?	Comment?
<input type="checkbox"/>	ftpgroup	MyISAM	latin1_swedish_ci	0	1,024	0		0	Table des groupes ProFTPD
<input type="checkbox"/>	ftpquotalimits	MyISAM	latin1_swedish_ci	0	1,024	0		0	Table des quotas ProFTPD
<input type="checkbox"/>	ftpquotatotal	MyISAM	latin1_swedish_ci	0	1,024	0		0	Table des compteurs des quotas ProFTPD
<input type="checkbox"/>	ftpuser	MyISAM	latin1_swedish_ci	0	1,024	0	1	0	Table des utilisateurs ProFTPD
<input type="checkbox"/>	4 in total	InnoDB	latin1_swedish_ci	0	4,096	0			

Selected (0)

Move to other database:

[Create table](#) [Create view](#) [Create materialized view](#)

Routines

[Create procedure](#) [Create function](#)

Events

[Create event](#)

Et voilà, le contenu de la base est rempli grâce au script !