

# MariaDB

## Généralités

MariaDB est un fork de MySQL créé par l'un des cofondateurs de MySQL suite aux différents rachat de SUN et d'Oracle. Il est sous licence libre et protégé par une fondation qui interdit son rachat par une société privé. Ce mécanisme permet de s'assurer de la pérennité du projet libre

- <https://mariadb.org>

## Installation

Pour configurer les dépôts en fonction de son système, se rendre sur la page officielle : <https://downloads.mariadb.org/mariadb/repositories/>

Procédure d'installation sous Debian Wheezy avec MariaDB 5.5 qui est compatible avec MySQL 5.5. Attention à partir de MariaDB 10 stable, les projets divergent mais MariaDB propose à présent plus de fonctionnalités que MySQL.

```
aptitude install python-software-properties
sudo apt-key adv --recv-keys --keyserver keyserver.ubuntu.com
0xcbc082a1bb943db
sudo add-apt-repository 'deb
http://ftp.igh.cnrs.fr/pub/mariadb/repo/5.5/debian wheezy main'
```

Le dépôt précédent était tombé. Il a fallu supprimer le précédent et le remplacer par celui qui est décrit dans cette page. Pour supprimer le dépôt.

```
add-apt-repository --remove 'deb
http://mirrors.linsrv.net/mariadb/repo/5.5/debian wheezy main'
```

```
aptitude update
aptitude install mariadb-server
```

Le mot de passe root est demandé à l'installation pour tout de suite le positionner.

```
chkconfig --list mysql
/etc/init.d/mysql status
```

## Configuration

### Configuration générale

A l'installation de MariaDB, la configuration se situe dans `/etc/mysql/my.cnf`.

## Configuration spécifique

Lancer le script de sécurisation qui permet au passage d'attribuer un mot de passe à root parce que par défaut, il n'y en a pas. Très clairement, si rien n'est fait ariaDB est une passoire par défaut!

```
root@<nameserver>:~# mysql_secure_installation
/usr/bin/mysql_secure_installation: line 379: find_mysql_client: command not
found
```

```
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
you haven't set the root password yet, the password will be blank,
so you should just press enter here.
```

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

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

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

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

```
By default, a 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] y
- Dropping test database...
ERROR 1008 (HY000) at line 1: Can't drop database 'test'; database doesn't
```

```
exist
... Failed! Not critical, keep moving...
- Removing privileges on test database...
... Success!
```

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

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

Cleaning up...

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

Thanks for using MariaDB!

Redémarrer la base de données après (/etc/init.d/mysqld restart).

## Création initiale base et utilisateur

Création d'une base de données avec un utilisateur qui ne peut que se connecter en local et qui dispose de tous les droits sur la base mais pas la possibilité de donner des permissions à d'autres comptes. Se loguer en root et entrer les commandes suivantes.

```
mysql -p
CREATE DATABASE <databasename>;
CREATE USER '<user>'@'localhost' IDENTIFIED BY 'mot_de_passe';
GRANT ALL PRIVILEGES ON <databasename>.* TO '<user>'@'localhost';
FLUSH PRIVILEGES;
```

La clause WITH GRANT OPTION de la commande GRANT donne à l'utilisateur le droit de donner les droits qu'il possède à d'autres utilisateurs. La plus grande prudence est recommandée pour cette commande, c'est ce qu'on ne veut surtout pas dans notre cas. Pour conserver un utilisateur admin mais uniquement de sa base de données, pas celle des autres.

```
show grants for '<user>'@'localhost';
```

Sortir de la connexion et réessayer de se loguer à notre nouvelle base de données.

```
mysql -u <user> -p <databasename>
```

On doit être connecté à notre base qui est actuellement vide.

Commande inverse de suppression de droits.

```
REVOKE ALL PRIVILEGES, GRANT OPTION FROM '<user>'@'localhost';
```

# Tuning

```
aptitude install mysqltuner
```

Lancer la commande et entrer le login/mot de passe de l'administrateur.

```
mysqltuner
```

```
root@<name_server># mysqltuner

>> MySQLTuner 1.1.1 - Major Hayden <major@mhtx.net>
>> Bug reports, feature requests, and downloads at http://mysqltuner.com/
>> Run with '--help' for additional options and output filtering
Please enter your MySQL administrative login: root
Please enter your MySQL administrative password:

----- General Statistics -----
--
[--] Skipped version check for MySQLTuner script
[OK] Currently running supported MySQL version 5.5.37-MariaDB-1~wheezy
[OK] Operating on 64-bit architecture

----- Storage Engine Statistics -----
--
[--] Status: +Archive -BDB +Federated +InnoDB -ISAM -NDBCluster
[--] Data in MyISAM tables: 33K (Tables: 35)
[--] Data in InnoDB tables: 16K (Tables: 1)
[--] Data in PERFORMANCE_SCHEMA tables: 0B (Tables: 17)
[!!!] Total fragmented tables: 7

----- Security Recommendations -----
--
[OK] All database users have passwords assigned

----- Performance Metrics -----
--
[--] Up for: 5d 21h 3m 19s (21K q [0.042 qps], 1K conn, TX: 15M, RX: 2M)
[--] Reads / Writes: 86% / 14%
[--] Total buffers: 192.0M global + 2.7M per thread (151 max threads)
[OK] Maximum possible memory usage: 597.8M (30% of installed RAM)
[OK] Slow queries: 0% (2/21K)
[OK] Highest usage of available connections: 3% (6/151)
[OK] Key buffer size / total MyISAM indexes: 16.0M/203.0K
[OK] Key buffer hit rate: 99.3% (28K cached / 202 reads)
[OK] Query cache efficiency: 41.6% (9K cached / 23K selects)
[OK] Query cache prunes per day: 0
[OK] Sorts requiring temporary tables: 0% (0 temp sorts / 418 sorts)
[OK] Temporary tables created on disk: 16% (196 on disk / 1K total)
[OK] Thread cache hit rate: 99% (6 created / 1K connections)
[!!!] Table cache hit rate: 17% (133 open / 772 opened)
```

```
[OK] Open file limit used: 16% (174/1K)
[OK] Table locks acquired immediately: 99% (7K immediate / 7K locks)
[OK] InnoDB data size / buffer pool: 16.0K/128.0M
```

```
----- Recommendations -----
--
General recommendations:
  Run OPTIMIZE TABLE to defragment tables for better performance
  Enable the slow query log to troubleshoot bad queries
  Increase table_cache gradually to avoid file descriptor limits
Variables to adjust:
  table_cache (> 400)
```

## Dump

Création de dump à chaud avec la commande `mysqldump`. Elle convient pour les tables MyISAM et InnoDB.

```
mysqldump --databases OPTIONS DB1 DB2...
mysqldump --all-databases OPTIONS DB1 DB2...

mysqldump -u root -p nombd > /tmp/nombd.sql
```

## Sauvegarde

Sauvegarde directe des fichiers.

Commande à exécuter connecté avec le client `mysql`.

- `backup table` : convient uniquement aux tables myisam.
- `restore table` : convient uniquement aux tables myisam.
- `mysqlhotcopy` copie les fichiers de tables myisam à chaud.

Pour effectuer la même chose avec des bases innodb, il n'existe pas d'outil gratuit stable pour le moment. Il faut suivre cet outil lorsqu'il aura bien mûri :

<http://www.percona.com/docs/wiki/percona-xtrabackup:start>.

## Maintenance des bases

Il existe 4 maintenances.

- `check`
- `analyze`
- `repair`
- `optimise`

En MyISAM, les 4 fonctionnent. En Innodb, seul le check et l'optimize ont une utilité.

En Innodb, analyze n'a pas d'intérêt et repair n'est pas supporté et peut entraîner des pertes de données.

Exemple avec le check.

```
mysqlcheck -u root -pPASSWORD --all-databases -s
```

Exemple avec l'optimize

```
mysqlcheck -u root -pPASSWORD --all-databases -optimize
```

Suppression manuelle des logs binaires Voir les logs en cours.

```
mysql> show master logs;
```

```
mysql> PURGE MASTER LOGS TO 'mysql-bin.000012';
```

Signifie supprimer tout les logs précédents le log 12. Cela supprime les logs de 1 à 11.

From:  
<https://wiki.ouieuhtoutca.eu/> - **kilsufi de noter**

Permanent link:  
<https://wiki.ouieuhtoutca.eu/mariadb>

Last update: **2021/01/21 21:42**

