

# Configuration d'un serveur de bases de données

Projet gsb.org



DUMAS Lucie

## Table des matières

---

Qu'est-ce que MariaDB.....	3
Installation de MariaDB.....	4
Installation .....	4
Conformité du cahier des charges .....	6



## Qu'est-ce que MariaDB

---

MariaDB est un système de gestion de base de données open source qui est une alternative compatible avec MySQL.

Il offre un stockage et une récupération efficace des données, la prise en charge du langage SQL, une grande extensibilité et une communauté active de développeurs.

MariaDB est utilisé pour stocker, organiser et récupérer des données dans de nombreuses applications, des sites web aux applications d'entreprise. Configuration de GPG



# Installation de MariaDB

---

## Installation

Dans un premier temps, nous commençons par cloner notre machine virtuelle template pour créer une nouvelle machine (priv-db1 ou priv-db2). Nous changeons le nom d'hôte ainsi que l'adresse IP (10.31.177.33 pour priv-db1 et 10.31.178.33) :

```
# Change le nom d'hôte à l'aide d'une commande
hostnamectl set-hostname priv-db1 # Ou priv-db2

# Change le nom d'hôte directement depuis le fichier de configuration
nano /etc/hosts
```

Nous modifions dans le fichier `/etc/hosts` le nom d'hôte de la machine et nous nous déconnectons pour actualiser l'affichage du nom.

Nous pouvons maintenant procéder à l'installation sécurisée de MariaDB. Nous entrons donc la commande suivante :

```
mysql_secure_installation
```

En ce qui concerne l'installation de MariaDB, nous laisserons les paramètres par défaut de l'outil. Nous changerons cependant le mot de passe du compte root :

```
root@priv-db2:~# mysql_secure_installation

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...
```

```
You already have your root account protected, so you can safely answer 'n'.
Switch to unix_socket authentication [Y/n] |
```



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

Change the root password? [Y/n]
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```

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



```
root@priv-db2:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 41
Server version: 10.11.3-MariaDB-1 Debian 12

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)]> |
```

## Conformité du cahier des charges

Dans un premier temps, nous commençons par nous connecter à l'interface en ligne de commande de MariaDB, puis créons notre base de données GSB :

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

```
CREATE DATABASE gsb ;
```

```
MariaDB [(none)]> create database gsb;
Query OK, 1 row affected (0,000 sec)
```

Nous vérifions que notre base de données soit bien créée à l'aide de la commande suivante :

```
SHOW DATABASES ;
```

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| gsb      |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
5 rows in set (0,000 sec)
```



Nous devons maintenant créer l'utilisateur admin, qui aura pour mot de passe « password » et qui possèdera tous les droits sur toutes les bases de données ainsi que la permission de modifier les droits des autres utilisateurs. Nous appliquerons les commandes suivantes :

```
-- Création de l'utilisateur
create user 'admin'@'%' identified by 'password';

-- Affectation des droits et de la permission de changement de droits des autres utilisateurs
grant all privileges on *.* to 'admin'@'%' with grant option;

-- Sauvegarde des modifications
flush privileges;
```

```
MariaDB [(none)]> create user 'admin'@'%' identified by 'password';
Query OK, 0 rows affected (0,001 sec)
```

```
MariaDB [(none)]> grant all privileges on *.* to 'admin'@'%' with grant option;
Query OK, 0 rows affected (0,001 sec)
```

```
MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0,000 sec)
```

Nous créons ensuite l'utilisateur gsb, qui aura pour mot de passe « password » et qui possèdera tous les droits sur la base de données gsb uniquement :

```
-- Création de l'utilisateur
create user 'gsb'@'%' identified by 'password';

-- Affectation des droits
grant all privileges on gsb.* to 'gsb'@'%' with grant option;

-- Sauvegarde des modifications
flush privileges;
```

```
MariaDB [(none)]> create user 'gsb'@'%' identified by 'password';
Query OK, 0 rows affected (0,001 sec)
```



```
MariaDB [(none)]> grant all privileges on gsb.* to 'gsb'@'%';  
Query OK, 0 rows affected (0,008 sec)
```

```
MariaDB [(none)]> flush privileges;  
Query OK, 0 rows affected (0,000 sec)
```

Nous ferons ensuite un nouvel utilisateur « test » sur lequel nous utiliserons le plugin « auth\_ed25519 » permettant de hash les mots de passe. Nous commençons donc à installer le plugin :

```
INSTALL SONAME 'auth_ed25519'
```

Enfin, nous créons l'utilisateur test, qui aura pour mot de passe hashé « password » :

```
CREATE USER 'test'@'localhost' IDENTIFIED VIA ed25519 USING PASSWORD('password');
```

```
MariaDB [(none)]> CREATE USER 'test'@'localhost' IDENTIFIED VIA ed25519 USING PASSWORD('password');  
Query OK, 0 rows affected (0,001 sec)
```

Pour que nos nouveaux comptes utilisateurs puissent se connecter à la base de données depuis une machine distante, nous devons modifier le fichier de configuration /etc/mysql/mariadb.conf.d/50-server.cnf en changeant la bind address locale par 0.0.0.0 :

```
# Instead of skip-networking the default is now to listen only on  
# localhost which is more compatible and is not less secure.  
bind-address            = 0.0.0.0|
```

