

# MYSQL REPLICATION ON DEBIAN 11 (Master ↔ Slave)

NOTE THAT : I used Debian 11 for this exercise and the procedure is explained in french but remains indifference for any other language

## Installation et configuration de MySQL/MariaDB

Avant d'installer le serveur MariaDB, exécutez la commande " apt update && apt install" ci-dessous pour installer certaines bases du package

**apt install gnupg2 apt-transport-https software-properties-common**

Exécutez la commande suivante pour importer la clé GPG pour le référentiel MariaDB. L'empreinte complète de la clé est "1993 69E5 404B D5FC 7D2F E43B CBCB 082A 1BB9 43DB".

**apt-key adv --recv-keys --keyserver hkps://keyserver.ubuntu.com:80 0xF1656F24C74CD1D8**

Exécutez la commande suivante pour ajouter le référentiel MariaDB pour le serveur Debian 11\*

Ensuite, mettez à jour et actualisez l'index des packages sur vos serveurs.

**add-apt-repository 'deb [arch=amd64,arm64,ppc64el]**

**http://sfo1.mirrors.digitalocean.com/mariadb/repo/10.8.3/debian bullseye main'**

**apt update**

Exécutez la commande ci-dessous pour installer mariadb

**apt install mariadb-server -y**

Le service MariaDB sera démarré et activé automatiquement. Exécutez la commande "systemctl" ci-dessous pour vérifier et vérifier le service MariaDB

**systemctl is-enabled mariadb**

**systemctl status mariadb**

Suite à l'installation, on exécute le script "mariadb-secure-installation" afin de sécuriser un minimum installation de MariaDB.

**mariadb-secure-installation**

Définir un mot de passe pour le compte "root" de MariaDB, empêcher les connexions distantes sur votre instance à l'aide du compte "root", empêcher les connexions anonymes et supprimer la base de test.

### a. Configuration du MariaDB réplication MASTER (web1)

Exécutez la commande "**systemctl stop mariadb**" pour arrêter le service MariaDB. Ceci est nécessaire avant de configurer le serveur MariaDB

On modifie le fichier de configuration du serveur MariaDB "**/etc/mysql/mariadb.conf.d/50-server.cnf**" en utilisant la commande

**nano /etc/mysql/mariadb.conf.d/50-server.cnf**

[mysqld]

bind-address = 192.168.100.251

server-id = 1

report\_host = web1

log\_bin = /var/log/mysql/mariadb-bin

log\_bin\_index = /var/log/mysql/mariadb-bin.index

relay\_log = /var/log/mysql/relay-bin

relay\_log\_index = /var/log/mysql/relay-bin.index

**systemctl start mariadb**

**ss -plnt ==> pour les ports et les service associés**

Créer un nouvel utilisateur MariaDB qui sera utilisé pour répliquer les données entre les serveurs MariaDB.

**mysql -u root -p**

**CREATE USER 'eva'@'%' IDENTIFIED BY 'poseidon';**

**GRANT REPLICATION SLAVE ON \*.\* TO 'eva'@'%';**

**FLUSH PRIVILEGES;**

Ensuite, exécutez la requête suivante pour vérifier le journal binaire actuel et sa position exacte. Dans mon cas, le fichier journal binaire du serveur MariaDB est "mariadb-bin.000007" avec la position "2669". Ces sorties seront utilisées dans l'étape suivante pour la mise en place du serveur "Slave".

**SHOW MASTER STATUS ;**

```
MariaDB [(none)]> SHOW MASTER STATUS;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mariadb-bin.000007 |      2669 |              |                  |
+-----+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [(none)]>
```

## **b. Configuration du MariaDB répllication Slave (web2)**

Exécutez la commande "systemctl stop mariadb» pour arrêter le service MariaDB. Ceci est nécessaire avant de configurer le serveur MariaDB

On modifie le fichier de configuration du serveur MariaDB ***"/etc/mysql/mariadb.conf.d/50-server.cnf"*** en utilisant la commande

**nano /etc/mysql/mariadb.conf.d/50-server.cnf**

[mysqld]

bind-address = 192.168.100.252

server-id = 2

report\_host = web2

log\_bin = /var/log/mysql/mariadb-bin

log\_bin\_index = /var/log/mysql/mariadb-bin.index

relay\_log = /var/log/mysql/relay-bin

relay\_log\_index = /var/log/mysql/relay-bin.index

**systemctl start mariadb**

**ss -plnt ==> pour les ports et les service associés**

Créer un nouvel utilisateur MariaDB qui sera utilisé pour répliquer les données entre les serveurs MariaDB.

**mysql -u root -p**

**CREATE USER 'rex'@'%' IDENTIFIED BY 'poseidon';**

**GRANT REPLICATION SLAVE ON \*. \* TO 'rex'@'%';**

**FLUSH PRIVILEGES;**

Arrêter l'esclave. Ensuite, ajoutez des informations détaillées sur le serveur MariaDB "master". Ici, le serveur MariaDB "master" utilise le nom d'hôte "web1", l'utilisateur et le mot de passe pour la réplication des données est "rex", et enfin le fichier binlog est "mariadb-bin.000007" avec la position "2669".

**STOP SLAVE;**

**CHANGE MASTER TO MASTER\_HOST='web1', MASTER\_USER='rex',**

**MASTER\_PASSWORD='poseidon', MASTER\_LOG\_FILE='mariadb-bin.000007',**

**MASTER\_LOG\_POS=2669;**

**START SLAVE;**

**SHOW SLAVE STATUS\G**

## SHOW SLAVE STATUS\G

```
MariaDB [(none)]>
MariaDB [(none)]> SHOW SLAVE STATUS\G
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: web1
Master_User: rex
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mariadb-bin.000007
Read_Master_Log_Pos: 3578
Relay_Log_File: relay-bin.000018
Relay_Log_Pos: 3879
Relay_Master_Log_File: mariadb-bin.000007
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 3578
Relay_Log_Space: 4483
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
```

### c. Configuration du pare-feu UFW

Il est recommandé d'utiliser le pare-feu entre tous les serveurs MariaDB. Cela garantira que sauf les réseaux autorisés pourront accéder et atteindre les ports de service MariaDB. Et dans ce tp, j'ai utilisé le pare-feu UFW.

**apt install ufw== > pour installer**

Seuls les réseaux "192.168.100.0/24" seront autorisés à accéder aux ports TCP du serveur MariaDB

ufw allow from 192.168.100.0/24 to 192.168.100.251 port 3306 ,4567,4568,443,80 proto tcp

====>MASTER

ufw allow from 192.168.100.0/24 to 192.168.100.252 port 3306 ,4567,4568,4444,443,80 proto tcp

====>SLAVE

Fonctions des ports :

3306==Connexion mysql

4567==trafic de replication de cluster

4568== transfert d'état incrémentiel

4444==copie complete des données du donneur

ufw enable === > pour l'activer

ufw status===> pour voir les réseaux autorisés sur le pare-feu

#### **d. Test de la réplication de la base de données**

**Sur le master ;**

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

Créez une nouvelle base de données "test1"

```
CREATE DATABASE test1 ;
```

```
USE test1 ;
```

Créez maintenant des tables dans la base de données « test1 »

```
CREATE TABLE users (id INT AUTO_INCREMENT,  
firstname VARCHAR(30),  
lastname VARCHAR(30),  
date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
PRIMARY KEY(id));
```

INSERT INTO users(firstname,lastname) VALUES ('Alice','Wonders') ; ====> Pour créer utilisateur Alice Wonders dans la table users

```
INSERT INTO users(firstname,lastname) VALUES ('Bob','Burgers');
```

```
SELECT * FROM users;
```

```
select @@hostname;
```

```
SELECT * FROM users;
```

OpenSSH SSH client

```
MariaDB [phpmyadmin]> select @@hostname;
```

@@hostname
web1

1 row in set (0,000 sec)

```
MariaDB [phpmyadmin]> SELECT * FROM users;
```

id	firstname	lastname	date
1	Alice	Wonders	2023-07-03 12:49:35
2	Bob	Burgers	2023-07-03 12:51:25

2 rows in set (0,000 sec)

```
MariaDB [phpmyadmin]>
```

OpenSSH SSH client

```
MariaDB [phpmyadmin]> select @@hostname;
```

@@hostname
web2

1 row in set (0,000 sec)

```
MariaDB [phpmyadmin]> SELECT * FROM users;
```

id	firstname	lastname	date
1	Alice	Wonders	2023-07-03 12:49:35
2	Bob	Burgers	2023-07-03 12:51:25

2 rows in set (0,000 sec)

```
MariaDB [phpmyadmin]>
```