

MYDIGITALSCHOOL ANGERS — BTS SIO SISR

Réplication de Base de Données MariaDB

Procédure de réplication Master-Slave

Rédacteur	Paul Barrault-Gautier
Établissement	MyDigitalSchool Angers
Formation	BTS SIO — option SISR (Session 2026)
Cadre	Travaux pratiques / Projet de formation
Date	Avril 2026
Version	1.0

1. Contexte et objectifs

Dans le cadre de la disponibilité des données chez Loste Tradi-France, une réplication MariaDB en mode Master-Slave a été mise en place. Ce mécanisme permet de maintenir un réplica synchronisé de la base de données principale, assurant ainsi la redondance des données et la possibilité de basculement en cas de panne du serveur maître.

La réplication MariaDB est basée sur les binary logs (binlog) : le serveur maître enregistre toutes les modifications dans un journal binaire, que le serveur esclave lit et rejoue pour rester synchronisé.

1.1 Architecture de réplication

Rôle	Hostname	Adresse IP	OS / Version MariaDB
Maître (Master)	db-master	192.168.20.10	Debian 12 / MariaDB 10.11 LTS
Esclave (Slave)	db-slave	192.168.20.11	Debian 12 / MariaDB 10.11 LTS

1.2 Flux de réplication

- Le Master écrit toutes les modifications dans le Binary Log (binlog)
- L'I/O Thread du Slave se connecte au Master et copie les événements binlog dans le Relay Log
- Le SQL Thread du Slave rejoue les événements du Relay Log sur sa propre base
- La réplication est asynchrone par défaut (semi-synchrone possible avec plugin)

1.3 Prérequis

- MariaDB 10.11 LTS installé sur les deux serveurs
- Connectivité réseau entre db-master et db-slave sur le port 3306
- Accès root MariaDB sur les deux nœuds
- Synchronisation NTP entre les deux serveurs (chrony ou systemd-timesyncd)

2. Configuration du serveur Maître

2.1 Éditer le fichier my.cnf

Sur db-master, éditer `/etc/mysql/mariadb.conf.d/50-server.cnf` (ou `/etc/mysql/my.cnf` selon l'installation) :

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

2.2 Paramètres de réplication du Master

```
[mysqld]
# Identifiant unique du serveur dans la topologie de réplication
# OBLIGATOIRE – chaque nœud doit avoir un server-id différent
server-id = 1

# Activer le Binary Log (journalisation des modifications)
log_bin = /var/log/mysql/mysql-bin.log

# Format des binlogs : ROW = réplication ligne par ligne (le plus fiable)
binlog_format = ROW

# Durée de conservation des binlogs (en jours)
expire_logs_days = 7

# Base(s) de données à répliquer (supprimer pour tout répliquer)
binlog_do_db = nom_base_production

# Adresse d'écoute : s'assurer que le Master écoute sur l'interface réseau
bind-address = 192.168.20.10
```

2.3 Redémarrer MariaDB et vérifier

```
sudo systemctl restart mariadb
sudo systemctl status mariadb

# Vérifier que le binlog est actif
sudo mysql -e "SHOW MASTER STATUS;"
```

2.4 Créer l'utilisateur de réplication

```
sudo mysql -u root -p

-- Créer un utilisateur dédié à la réplication
-- Remplacer 'MotDePasseRepl!' par un mot de passe fort
CREATE USER 'replicator'@'192.168.20.11' IDENTIFIED BY 'MotDePasseRepl!';

-- Accorder uniquement le privilège REPLICATION SLAVE
GRANT REPLICATION SLAVE ON *.* TO 'replicator'@'192.168.20.11';

FLUSH PRIVILEGES;
```

2.5 Prendre un snapshot initial de la base

Pour initialiser l'esclave avec les données du maître, il faut effectuer un dump cohérent en verrouillant les tables :

```
# Se connecter à MariaDB
sudo mysql -u root -p

-- Verrouiller les tables pour garantir la cohérence du dump
FLUSH TABLES WITH READ LOCK;

-- Relever la position exacte du binlog (à noter pour la config du Slave)
SHOW MASTER STATUS;
-- +-----+-----+-----+-----+
-- | File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
-- +-----+-----+-----+-----+
-- | mysql-bin.000003 |      154 | ma_base      |                   |
-- +-----+-----+-----+-----+

-- Ouvrir un AUTRE terminal pour effectuer le dump
-- NE PAS fermer cette session (le verrou serait libéré)
```

Dans un second terminal :

```
# Dump de la base de données
sudo mysqldump -u root -p --single-transaction --master-data=2 \
  nom_base_production > /tmp/dump_master.sql

# Copier le dump vers le Slave
scp /tmp/dump_master.sql paul@192.168.20.11:/tmp/
```

Puis dans le premier terminal, déverrouiller :

```
-- Libérer le verrou
UNLOCK TABLES;
EXIT;
```

3. Configuration du serveur Esclave

3.1 Éditer le fichier my.cnf sur db-slave

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

[mysqld]
# server-id DIFFÉRENT du Master (obligatoire)
```

```
server-id = 2

# Activer le relay log
relay-log = /var/log/mysql/mysql-relay-bin.log

# Lecture seule : empêche les écritures directes sur le Slave
read_only = 1

# Optionnel : répliquer uniquement la base de production
replicate_do_db = nom_base_production

# Adresse d'écoute du Slave
bind-address = 192.168.20.11
```

3.2 Importer le dump initial

```
sudo mysql -u root -p
CREATE DATABASE IF NOT EXISTS nom_base_production;
EXIT;

sudo mysql -u root -p nom_base_production < /tmp/dump_master.sql
```

3.3 Configurer la connexion au Master

```
sudo mysql -u root -p

-- Configurer la source de réplication
-- Utiliser les valeurs relevées dans SHOW MASTER STATUS
CHANGE MASTER TO
  MASTER_HOST='192.168.20.10',
  MASTER_USER='replicator',
  MASTER_PASSWORD='MotDePasseRepl!',
  MASTER_LOG_FILE='mysql-bin.000003',
  MASTER_LOG_POS=154;

-- Démarrer la réplication
START SLAVE;

-- Vérifier l'état de la réplication
SHOW SLAVE STATUS\G
```

3.4 Vérification de la réplication

Dans la sortie de SHOW SLAVE STATUS, vérifier :

Variable	Valeur attendue	Signification
Slave_IO_Running	Yes	Thread I/O connecté au Master
Slave_SQL_Running	Yes	Thread SQL rejoue les events
Seconds_Behind_Master	0	Slave synchronisé avec le Master
Last_Error	(vide)	Aucune erreur de réplication

4. Test de la réplication

```
# Sur le MASTER : créer une table de test
sudo mysql -u root -p nom_base_production
CREATE TABLE test_replication (id INT PRIMARY KEY, message VARCHAR(100));
INSERT INTO test_replication VALUES (1, 'Replication OK - Loste');

# Sur le SLAVE : vérifier que la table existe
```

```
sudo mysql -u root -p nom_base_production
SELECT * FROM test_replication;
-- Doit retourner : 1 | Replication OK - Leste

# Nettoyer après le test
-- Sur le MASTER :
DROP TABLE test_replication;
```

5. Surveillance de la réplication

- Vérifier quotidiennement Seconds_Behind_Master (doit rester < 30s en production)
- Mettre en place une alerte si Slave_IO_Running ou Slave_SQL_Running passe à No
- Superviser la taille des binlogs sur le Master (espace disque)
- Intégrer la surveillance dans Zabbix ou Nagios via le check MariaDB

```
# Commande de vérification rapide (depuis le Slave)
mysql -u root -p -e "SHOW SLAVE STATUS\G" | grep -E 'Running|Behind|Error'
```