

MYDIGITALSCHOOL ANGERS — BTS SIO SISR

Réplication de Base de Données MariaDB

Procédure de failover — Plan de Continuité d'Activité (PCA)

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. Objectif et périmètre

Ce document décrit la procédure de basculement (failover) de la base de données MariaDB en cas de défaillance du serveur maître (db-master). Il constitue la partie base de données du Plan de Continuité d'Activité (PCA) de Loste Tradi-France.

Le basculement consiste à promouvoir le serveur esclave (db-slave) au rôle de nouveau maître, de manière à rétablir rapidement les écritures en base de données.

1.1 Critères de déclenchement

Situation	Délai max avant action	Type de failover
Panne matérielle du serveur Master	15 minutes	Failover non planifié
Corruption de données sur le Master	Selon impact métier	Failover non planifié
Maintenance planifiée du Master	Planifié (J-7)	Switchover planifié
Retard de réplication > 5 minutes	Analyse immédiate	Alerte — pas de failover

1.2 Acteurs et responsabilités

Rôle	Responsable	Action
Administrateur BDD	Paul Barrault-Gautier	Exécute la procédure technique
Responsable IT	Responsable infrastructure	Décide du déclenchement
Métiers / Applications	Référents applicatifs	Valident la reprise fonctionnelle

2. Vérifications préalables au failover

2.1 Évaluer l'état de la réplication

```
# Sur db-slave – vérifier le statut de la réplication
mysql -u root -p -e "SHOW SLAVE STATUS\G" | grep -E 'Running|Behind|Error|File|Pos'
```

Points à vérifier :

- Slave_IO_Running et Slave_SQL_Running doivent être à Yes (ou No si le Master est déjà inaccessible)
- Seconds_Behind_Master : noter la valeur — si > 0, des données récentes peuvent être perdues
- Relay_Log_File et Relay_Log_Pos : noter pour estimer le dernier event reçu

2.2 Attendre la fin de l'application des relay logs

Si le Slave est encore en train d'appliquer des events reçus du Master avant la panne :

```
# Attendre que le Slave ait appliqué tous les relay logs
# (Seconds_Behind_Master doit atteindre 0 ou SQL_Remaining_Delay = NULL)
mysql -u root -p -e "STOP SLAVE IO_THREAD;"
-- Arrêter seulement le thread I/O (arrêt des nouvelles réceptions)
-- Laisser le thread SQL finir d'appliquer les events en attente

-- Attendre que Relay_Master_Log_File et Exec_Master_Log_Pos
-- correspondent aux dernières valeurs connues
SHOW SLAVE STATUS\G
```

3. Procédure de failover — Promotion du Slave en Master

IMPORTANT : Cette procédure provoque une interruption des écritures. Prévenir les équipes métier avant de commencer.

3.1 Arrêter la réplication sur le Slave

```
# Sur db-slave
mysql -u root -p

-- Arrêter complètement la réplication
STOP SLAVE;

-- Réinitialiser les informations de connexion au Master
RESET SLAVE ALL;

-- Désactiver le mode lecture seule : le Slave devient Master
SET GLOBAL read_only = 0;

-- Vérifier
SHOW VARIABLES LIKE 'read_only';
-- Doit retourner : OFF
```

3.2 Activer le Binary Log sur le nouveau Master

Si log_bin n'était pas activé sur le Slave (cas minimal), l'ajouter dans my.cnf :

```
# Éditer la configuration sur db-slave (nouveau Master)
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf

# Ajouter / décommenter :
log_bin = /var/log/mysql/mysql-bin.log
binlog_format = ROW
expire_logs_days = 7

sudo systemctl restart mariadb
```

3.3 Reconfigurer les applications

- Mettre à jour la chaîne de connexion des applications pour pointer vers 192.168.20.11
- Si un VIP (Virtual IP) est utilisé avec Keepalived, mettre à jour la configuration Keepalived
- Vider les caches applicatifs si nécessaire
- Redémarrer les services applicatifs si la reconnexion automatique n'est pas configurée

3.4 Créer l'utilisateur de réplication sur le nouveau Master

```
# Sur db-slave (devenu Master)
mysql -u root -p

CREATE USER 'replicator'@'192.168.20.10' IDENTIFIED BY 'MotDePasseRepl!';
GRANT REPLICATION SLAVE ON *.* TO 'replicator'@'192.168.20.10';
FLUSH PRIVILEGES;

-- Relever la position binlog pour le futur Slave (ancien Master)
SHOW MASTER STATUS;
```

4. Validation post-failover

4.1 Tests fonctionnels

Étape	Action	Résultat attendu	Statut
-------	--------	------------------	--------

1	Connexion applicative au nouveau Master (192.168.20.11)	Connexion OK	[] OK / [] KO
2	Test d'écriture en base (INSERT, UPDATE)	Écriture réussie	[] OK / [] KO
3	Test de lecture (SELECT sur tables principales)	Données cohérentes	[] OK / [] KO
4	Vérification de la cohérence des données métier	Validation métier	[] OK / [] KO
5	Contrôle des logs applicatifs (absence d'erreurs BDD)	0 erreur BDD	[] OK / [] KO

5. Retour à la topologie initiale (après rétablissement du Master)

5.1 Rétablir l'ancien Master en Slave

Une fois le serveur db-master rétabli, le reconfigurer comme Slave du nouveau Master (db-slave) :

```
# Sur db-master (rétabli)
mysql -u root -p

-- Configurer db-master comme Slave de db-slave
CHANGE MASTER TO
  MASTER_HOST='192.168.20.11',
  MASTER_USER='replicator',
  MASTER_PASSWORD='MotDePasseRepl!',
  MASTER_LOG_FILE='mysql-bin.000001', -- Valeur relevée à l'étape 3.4
  MASTER_LOG_POS=154;

START SLAVE;
SHOW SLAVE STATUS\G
```

5.2 Switchover planifié (retour à la topologie d'origine)

Si souhaité, effectuer un switchover planifié pour revenir à db-master comme Master :

1. Arrêter les écritures applicatives (maintenance applicative courte)
2. Attendre que db-master (Slave temporaire) soit synchronisé (Seconds_Behind_Master = 0)
3. Répéter la procédure de failover en sens inverse (promouvoir db-master)
4. Reconfigurer db-slave comme Slave de db-master
5. Reprendre les écritures applicatives

6. Annexe — Registre des interventions

Date	Type	Intervenant	Observations
__/__/__	Failover / Switchover		
__/__/__			
__/__/__			