

# Rapport sur le Partage et la Synchronisation des Données MySQL

Auteur : Ambinintsoa RANDRIANARISOA

2 décembre 2025

## Résumé

Ce second rapport documente l'étape de vérification de la synchronisation des données entre deux conteneurs MySQL s'exécutant simultanément, tous deux liés au même volume nommé (`mysql_data`). Cette procédure valide que les opérations d'écriture effectuées par un conteneur sont immédiatement et complètement accessibles par l'autre, démontrant l'efficacité du volume pour le partage des données en temps réel.

## 1 Mise en Place de l'Environnement de Synchronisation

Après avoir lancé le premier conteneur (`mysql-partage`) pour récupérer les données persistantes, il faut lancer un second conteneur pour simuler un environnement multi-service utilisant la même source de données.

### Instruction 6 : Lancement du Conteneur Synchronisé

Il faut lancer un nouveau conteneur, nommé `mysql-synchro`, en s'assurant qu'il utilise le même volume nommé : `mysql_data`.

```
1 docker run -d \
2   --name mysql-synchro \
3   -e MYSQL_ROOT_PASSWORD=pass \
4   -v mysql_data:/var/lib/mysql \
5   mysql:8
```

**Résultat Attendu :** Les deux conteneurs (`mysql-partage` et `mysql-synchro`) sont maintenant actifs et lisent/écrivent sur le même ensemble de fichiers.

## 2 Test d'Écriture et de Synchronisation en Temps Réel

Pour prouver la synchronisation, il faut insérer un nouvel enregistrement via le conteneur nouvellement créé, puis vérifier sa présence immédiate via le premier conteneur.

### Instruction 7 : Insertion de Données par le Second Conteneur

Il faut insérer un nouvel utilisateur (`Carla`) dans la table `utilisateurs` via le conteneur `mysql-synchro`.

---

```
1 docker exec -it mysql-synchro mysql -u root -ppass -e "
2   USE mon_projet;
3   INSERT INTO utilisateurs (nom, email) VALUES
4     ('Carla', 'carla@example.com');
5   SELECT * FROM utilisateurs;
6 "
```

---

**Vérification locale (Conteneur 2) :** Le résultat doit montrer les trois utilisateurs : Alice (ancienne donnée), Bob (ancienne donnée) et Carla (nouvelle donnée).

---

```
1 +---+-----+
2 | id | nom    | email          |
3 +---+-----+
4 | 1  | Alice   | alice@example.com |
5 | 2  | Bob     | bob@example.com  |
6 | 3  | Carla   | carla@example.com |
7 +---+-----+
```

---

### Instruction 8 : Vérification par le Premier Conteneur

Il faut interroger la base de données via le premier conteneur (`mysql-partage`) pour vérifier si l'enregistrement de Carla est visible, sans que ce conteneur n'ait effectué d'opération d'écriture.

---

```
1 docker exec -it mysql-partage mysql -u root -ppass -e "
2   SELECT * FROM mon_projet.utilisateurs;
3 "
```

---

**Résultat de la Synchronisation :** Le résultat doit être strictement identique au précédent, incluant la ligne de Carla :

---

```
1 +---+-----+
2 | id | nom    | email          |
3 +---+-----+
4 | 1  | Alice   | alice@example.com |
5 | 2  | Bob     | bob@example.com  |
6 | 3  | Carla   | carla@example.com |
7 +---+-----+
```

---

## 3 Conclusion

La réussite de l'Instruction 8 prouve que les deux conteneurs, bien qu'exécutant des processus MySQL séparés, partagent le même espace de stockage sur disque grâce au volume `mysql_data`. Cela confirme que les volumes nommés sont la méthode standard pour assurer à la fois la persistance et la synchronisation des données d'état (comme une base de données) entre différentes instances de conteneurs.