

# TP Docker : Persistance des données et Volumes

Compte Rendu

2 décembre 2025

## Introduction

Dans ce TP, on a utilisé Docker pour lancer une base de données MySQL. Le but principal était de comprendre comment garder nos données (les noms et emails) même si le conteneur s'arrête ou est supprimé. Pour ça, on a utilisé un **Volume**.

## 1 Partie 1 : Test sur un seul conteneur (Arrêt et Redémarrage)

### 1.1 1. Lancement et création du Volume

On a lancé le conteneur avec une option spéciale `-v` pour créer un volume nommé `mysql_data`. Ce volume stocke les fichiers sur l'ordinateur et pas juste dans le conteneur.

```
docker run -d --name mysql -e MYSQL_ROOT_PASSWORD=pass -v mysql_data:/var/lib/
mysql mysql:8
```

### 1.2 2. Insertion des données (Bash et SQL)

Pour ajouter des données, on a dû entrer dans le conteneur. On a fait ça en deux étapes :

- **Bash** : D'abord, on utilise `docker exec` avec `bash`. Bash est le langage de commande (le shell) qui nous permet de nous déplacer dans le conteneur.
- **SQL** : Une fois dedans, on lance le logiciel MySQL. Lui, il ne parle que SQL. C'est là qu'on crée la table.

```
# On entre dans le conteneur
docker exec -it mysql bash

# On se connecte a la base
mysql -u root -ppass
```

Ensuite, on a tapé les commandes SQL :

```
CREATE DATABASE mon_projet;
USE mon_projet;
CREATE TABLE utilisateurs (id INT AUTO_INCREMENT PRIMARY KEY, nom VARCHAR(50),
email VARCHAR(100));
INSERT INTO utilisateurs (nom, email) VALUES ('Alice', 'alice@test.com'), ('Bob'
, 'bob@test.com');
```

### 1.3 3. Le test de l'arrêt

On a stoppé le conteneur avec `docker stop mysql`, puis on l'a redémarré avec `docker start mysql`. Les données étaient toujours là car on n'a pas supprimé le conteneur, on l'a juste mis en pause.

## 2 Partie 2 : Test avec deux conteneurs différents

C'est la partie la plus importante. On voulait voir si on pouvait récupérer nos données (Alice et Bob) en utilisant un **tout nouveau conteneur**, comme si on avait changé de serveur.

### 2.1 1. Suppression de l'ancien conteneur

On a complètement supprimé le premier conteneur. À ce stade, le logiciel MySQL n'existait plus, mais le volume `mysql_data` était toujours sur le disque dur.

```
docker stop mysql
docker rm mysql
```

### 2.2 2. Lancement d'un nouveau conteneur

On a recréé un conteneur tout neuf. On a dû ajouter une option spéciale (`mysql_native_password`) parce qu'on avait des erreurs de connexion à cause de la sécurité de MySQL 8.

```
docker run -d \
  --name mysql \
  -e MYSQL_ROOT_PASSWORD=pass \
  -e MYSQL_AUTHENTICATION_PLUGIN=mysql_native_password \
  -v mysql_data:/var/lib/mysql \
  mysql:8
```

L'important ici, c'est qu'on a remis `-v mysql_data:/var/lib/mysql`. On a "rebranché" le vieux disque dur sur le nouveau conteneur.

### 2.3 3. Vérification finale

Pour prouver que ça marche, on a utilisé un autre conteneur temporaire juste pour lire les données :

```
docker run --rm mysql:8 mysql -h mysql -u root -ppass mon_projet -e "SELECT *
  FROM utilisateurs;"
```

**Résultat :** La commande a affiché Alice et Bob.

## Conclusion

Ce TP montre que les conteneurs sont éphémères (on peut les jeter et en refaire), mais les **Volumes** sont durables. Tant qu'on garde le volume, on peut retrouver nos données n'importe où, même avec un conteneur différent.