

# Docker : Dump et restauration

## Lancement MariaDB avec Docker

Pour lancer une base de données MariaDB avec Docker, utilisez la commande suivante :

```
docker run -d \  
  --name mariadb-server \  
  -e MYSQL_ROOT_PASSWORD=rootpass \  
  -e MYSQL_DATABASE=starwars \  
  -e MYSQL_USER=jedi \  
  -e MYSQL_PASSWORD=forcepass \  
  -v mariadb-data:/var/lib/mysql \  
  -p 3306:3306 \  
  mariadb:11.4
```

Explication des paramètres :

- `-d` : Exécute en arrière-plan
- `--name` : Nom du conteneur
- `-e MYSQL_ROOT_PASSWORD` : Mot de passe root
- `-e MYSQL_DATABASE` : Crée automatiquement une base de données
- `-e MYSQL_USER / MYSQL_PASSWORD` : Crée un utilisateur avec accès à la base
- `-v mariadb-data:/var/lib/mysql` : Volume pour persister les données
- `-p 3306:3306` : Expose le port 3306
- `mariadb:11.4` : Version fixe de MariaDB

## Exemple : Création des tables Star Wars

Connexion au conteneur pour créer les tables :

```
docker exec -it mariadb-server mysql -ujedi -pforcepass starwars
```

Création des tables :

```
CREATE TABLE planetes (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  nom VARCHAR(100),  
  climat VARCHAR(50),  
  population BIGINT  
);  
  
CREATE TABLE personnages (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  nom VARCHAR(100),  
  espece VARCHAR(50),  
  planete_id INT,  
  cote VARCHAR(20),  
  FOREIGN KEY (planete_id) REFERENCES planetes(id)  
);  
  
INSERT INTO planetes (nom, climat, population) VALUES  
( 'Tatooine', 'aride', 200000),  
( 'Naboo', 'tempéré', 4500000);  
  
INSERT INTO personnages (nom, espece, planete_id, cote) VALUES  
( 'Luke Skywalker', 'humain', 1, 'lumineux'),  
( 'Leia Organa', 'humain', 2, 'lumineux');
```

# Dump et restauration MariaDB

## Script de dump (sauvegarde)

Créez un fichier `dump_mariadb.sh` :

```
#!/bin/bash  
docker exec mariadb-server mysqldump -ujedi -pforcepass starwars > backup_starwars_$(date  
+%Y%m%d_%H%M%S).sql  
echo "Backup MariaDB créé : backup_starwars_$(date +%Y%m%d_%H%M%S).sql"
```

Rendre le script exécutable :

```
chmod +x dump_mariadb.sh
```

Exécution :

```
.\dump_mariadb.sh
```

Cela crée un fichier de sauvegarde avec horodatage, par exemple :

```
backup_starwars_20260101_143022.sql
```

## Script de restauration

Créez un fichier `restore_mariadb.sh` :

```
#!/bin/bash
docker exec -i mariadb-server mysql -ujedi -pforcepass starwars <
backup_starwars_20260101_143022.sql
echo "Restauration MariaDB terminée"
```

Rendre le script exécutable et exécuter :

```
chmod +x restore_mariadb.sh
.\restore_mariadb.sh
```

Le dump contient toute la structure (tables) et les données de la base `starwars`. Il peut être utilisé pour restaurer la base sur un autre serveur.

## Automatisation avec cron

Pour automatiser la sauvegarde quotidienne, ajoutez une tâche cron :

```
crontab -e
```

Ajoutez la ligne suivante pour une sauvegarde tous les jours à 2h du matin :

```
0 2 * * * /home/user/dump_mariadb.sh
```

## Lancement PostgreSQL avec Docker

Pour lancer une base de données PostgreSQL avec Docker, utilisez la commande suivante :

```
docker run -d \  
  --name postgres-server \  
  -e POSTGRES_PASSWORD=rootpass \  
  -e POSTGRES_USER=jedi \  
  -e POSTGRES_DB=starwars \  
  -v postgres-data:/var/lib/postgresql/data \  
  -p 5432:5432 \  
  postgres:14
```

Explication des paramètres :

- `-d` : Exécute en arrière-plan
- `--name` : Nom du conteneur
- `-e POSTGRES_PASSWORD` : Mot de passe de l'utilisateur
- `-e POSTGRES_USER` : Nom d'utilisateur principal
- `-e POSTGRES_DB` : Crée automatiquement une base de données
- `-v postgres-data:/var/lib/postgresql/data` : Volume pour persister les données
- `-p 5432:5432` : Expose le port 5432
- `postgres:14` : Version fixe de PostgreSQL

## Exemple : Création des tables Star Wars

Connexion au conteneur pour créer les tables :

```
docker exec -it postgres-server psql -U jedi -d starwars
```

Création des tables :

```
CREATE TABLE planetes (  
  id SERIAL PRIMARY KEY,  
  nom VARCHAR(100),  
  climat VARCHAR(50),  
  population BIGINT  
);  
  
CREATE TABLE personnages (  
  id SERIAL PRIMARY KEY,  
  nom VARCHAR(100),  
  espece VARCHAR(50),  
  planete_id INT,  
  cote VARCHAR(20),
```

```
FOREIGN KEY (planete_id) REFERENCES planetes(id)
);

INSERT INTO planetes (nom, climat, population) VALUES
('Tatooine', 'aride', 200000),
('Naboo', 'tempéré', 4500000);

INSERT INTO personnages (nom, espece, planete_id, cote) VALUES
('Luke Skywalker', 'humain', 1, 'lumineux'),
('Leia Organa', 'humain', 2, 'lumineux');
```

PostgreSQL utilise `SERIAL` pour les auto-incréments au lieu de `AUTO_INCREMENT` de MySQL/MariaDB.

# Dump et restauration PostgreSQL

## Script de dump (sauvegarde)

Créez un fichier `dump_postgres.sh` :

```
#!/bin/bash
docker exec postgres-server pg_dump -U jedi starwars > backup_starwars_$(date
+%Y%m%d_%H%M%S).sql
echo "Backup PostgreSQL créé : backup_starwars_$(date +%Y%m%d_%H%M%S).sql"
```

Rendre le script exécutable :

```
chmod +x dump_postgres.sh
```

Exécution :

```
./dump_postgres.sh
```

## Script de restauration

Créez un fichier `restore_postgres.sh` :

```
#!/bin/bash
docker exec -i postgres-server psql -U jedi starwars < backup_starwars_20260101_143022.sql
```

```
echo "Restauration PostgreSQL terminée"
```

Rendre le script exécutable et exécuter :

```
chmod +x restore_postgres.sh
.\restore_postgres.sh
```

## Automatisation avec cron

Pour automatiser la sauvegarde quotidienne :

```
crontab -e
```

Ajoutez la ligne suivante pour une sauvegarde tous les jours à 2h du matin :

```
0 2 * * * /home/user/dump_postgres.sh
```

## Comparaison des commandes

Opération	MariaDB	PostgreSQL
Lancement Docker	<code>mariadb:11.4</code>	<code>postgres:14</code>
Port par défaut	3306	5432
Connexion CLI	<code>mysql -u -p</code>	<code>psql -U -d</code>
Dump	<code>mysqldump</code>	<code>pg_dump</code>
Restauration	<code>mysql &lt; fichier.sql</code>	<code>psql &lt; fichier.sql</code>
Volume de données	<code>/var/lib/mysql</code>	<code>/var/lib/postgresql/data</code>

## Bonnes pratiques de sauvegarde

- **Fréquence** : Automatisez les sauvegardes quotidiennes avec cron
- **Rétention** : Conservez plusieurs sauvegardes (7 derniers jours, 4 dernières semaines, etc.)
- **Stockage externe** : Copiez les dumps vers un stockage distant (NAS, cloud, etc.)
- **Test de restauration** : Testez régulièrement la restauration pour vérifier l'intégrité des backups
- **Compression** : Comprimez les dumps pour économiser de l'espace : `gzip backup.sql`

Un backup non testé n'est pas un vrai backup ! Testez régulièrement la restauration de vos sauvegardes.

# Nettoyage des anciennes sauvegardes

Pour éviter de saturer le disque, supprimez automatiquement les sauvegardes de plus de 7 jours :

```
find ./ -name "backup_starwars_*.sql" -mtime +7 -delete
```

Ajoutez cette ligne à vos scripts de dump pour un nettoyage automatique.

---

Revision #2

Created 1 January 2026 16:39:47 by Thibaud FRICHET

Updated 14 January 2026 15:53:51 by Thibaud FRICHET