🚀 SQL Bootcamp – Mise en place et restauration AdventureWorks avec Docker + Adminer

Télécahrger les fichiers nécessaires

- [Docker Desktop](MOVE 'AdventureWorksLT2022_log' TO
  '/var/opt/mssql/data/AdventureWorksLT2022_log.ldf',
  REPLACE;
  GO

-- AdventureWorksLT2022
RESTORE DATABASE AdventureWorksLT2022
FROM DISK = '/var/backups/AdventureWorksLT2022.bak'
WITH MOVE 'AdventureWorksLT2022_Data' TO '/var/opt/mssql/data/AdventureWorksLT2022.mdf',
MOVE 'AdventureWorksLT2022_Log' TO '/var/opt/mssql/data/AdventureWorksLT2022_log.ldf',
REPLACE;
GO

~~~

🗄️ Installation de la base de données Northwind

Créer et installer Northwind (méthode recommandée)

```powershell
# 1. Créer la base de données Northwind
docker exec -i sql-bootcamp-server /opt/mssql-tools18/bin/sqlcmd -S localhost -U SA -P "YourStrong@Passw0rd" -C -Q "CREATE DATABASE Northwind"

# 2. Exécuter le script d'installation
docker exec -i sql-bootcamp-server /opt/mssql-tools18/bin/sqlcmd -S localhost -U SA -P "YourStrong@Passw0rd" -C -d Northwind -i /scripts/northwind/instnwnd.sql
~~~

### Ou en une seule ligne (PowerShell)

```
docker exec -i sql-bootcamp-server /opt/mssql-tools18/bin/sqlcmd -S localhost -U SA -P "YourStrong@Passw0rd" -C -Q "CREATE DATABASE Northwind" ; docker exec -i sql-bootcamp-server /opt/mssql-tools18/bin/sqlcmd -S localhost -U SA -P "YourStrong@Passw0rd" -C -d Northwind -i /scripts/northwind/instnwnd.sql
```

Vérification de l'installation Northwind

```
# Vérifier que Northwind existe
docker exec -it sql-bootcamp-server /opt/mssql-tools18/bin/sqlcmd -S localhost -U SA -P "YourStrong@Passw0rd" -C -Q "SELECT name FROM sys.databases WHERE name = 'Northwind'"

# Lister les tables créées dans Northwind
docker exec -it sql-bootcamp-server /opt/mssql-tools18/bin/sqlcmd -S localhost -U SA -P "YourStrong@Passw0rd" -C -d Northwind -Q "SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES ORDER BY TABLE_NAME"
```

Résultat attendu pour les tables :

```
Categories
CustomerCustomerDemo
CustomerDemographics
Customers
Employees
EmployeeTerritories
Order Details
Orders
Products
Region
Shippers
Suppliers
Territories
```

🗐 Installation de la base de données Pubs (optionnel)

```
# 1. Créer la base de données Pubs
docker exec -i sql-bootcamp-server /opt/mssql-tools18/bin/sqlcmd -S localhost -U SA -P "YourStrong@Passw0rd" -C -Q "CREATE DATABASE Pubs"

# 2. Exécuter le script d'installation
docker exec -i sql-bootcamp-server /opt/mssql-tools18/bin/sqlcmd -S localhost -U SA -P "YourStrong@Passw0rd" -C -d Pubs -i /scripts/northwind/instpubs.sql
```

☑ Vérification des bases restaurées (mise à jour)

```
docker exec -it sql-bootcamp-server /opt/mssql-tools18/bin/sqlcmd -S localhost -U SA -P "YourStrong@Passw0rd" -C -Q "SELECT name FROM sys.databases;"
```

Résultat attendu :

````
master
tempdb
model
msdb
SQLBootcamp
AdventureWorks2022
AdventureWorksDW2022
AdventureWorksLT2022
Northwind
Pubswww.docker.com/products/docker-desktop/)
- [Fichiers de backup AdventureWorks 2022](https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks)
- [AdventureWorks Backups](https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks)
- [Documentation](https://learn.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver16&tabs=ssms)
- [Adminer](https://www.adminer.org/)
- [Adminer Docker Hub](https://hub.docker.com/_/adminer)
---

- https://chatgpt.com/c/68c2c654-cc3c-8320-947c-31a076aeeae3



## 📂 Structure de l’environnement
````

```
C:\Users\awounfouet\formation\sql\sql-bootcamp
│
├── backups/ # Contient les fichiers .bak
│ ├── AdventureWorks2022.bak
│ ├── AdventureWorksDW2022.bak
│ └── AdventureWorksLT2022.bak
│
├── scripts/ # Contient nos scripts SQL
│ └── restore-databases.sql
│
├── docker-compose.yml # Version simple
└── docker-compose.advanced.yml # Version avancée avec Adminer
```

~~~

🐳 Docker Compose (avancé)

```yaml
~~~

```
services:
  sqlserver:
    image: mcr.microsoft.com/mssql/server:2022-latest
    container_name: sql-bootcamp-server
    environment:
      - ACCEPT_EULA=Y
      - SA_PASSWORD=YourStrong@Passw0rd
      - MSSQL_PID=Express
    ports:
      - "1433:1433"
    volumes:
      - sqlserver_data:/var/opt/mssql
      - ./scripts:/scripts
      - ./backups:/var/backups
    networks:
      - sql-network
    restart: unless-stopped
    healthcheck:
      test: ["CMD-SHELL", "/opt/mssql-tools18/bin/sqlcmd -S localhost -U sa -P YourStrong@Passw0rd -Q 'SELECT 1' -C"]
      interval: 30s
      timeout: 10s
      retries: 5
      start_period: 60s

  adminer:
    image: adminer:latest
    container_name: sql-adminer
    ports:
      - "8080:8080"
    networks:
      - sql-network
    restart: unless-stopped
    environment:
      - ADMINER_DEFAULT_SERVER=sqlserver
    depends_on:
      sqlserver:
        condition: service_healthy

volumes:
  sqlserver_data:

networks:
  sql-network:
    driver: bridge
```

📥 Vérification des backups

```
docker-compose -f docker-compose.advanced.yml down
docker-compose -f docker-compose.advanced.yml up -d



docker exec -it sql-bootcamp-server ls /var/backups
```

Résultat attendu :

```
AdventureWorks2022.bak
AdventureWorksDW2022.bak
AdventureWorksLT2022.bak
```

🔎 Identification des LogicalName

Avant de restaurer, on doit identifier les **LogicalName** de chaque fichier `.bak` :

```
docker exec -it sql-bootcamp-server /opt/mssql-tools18/bin/sqlcmd \
  -S localhost -U SA -P "YourStrong@Passw0rd" -C \
  -Q "RESTORE FILELISTONLY FROM DISK = '/var/backups/AdventureWorks2019.bak';"


# Version PowerShell
docker exec -it sql-bootcamp-server /opt/mssql-tools18/bin/sqlcmd -S localhost
-U SA -P "YourStrong@Passw0rd" -C -Q "RESTORE FILELISTONLY FROM DISK =
'/var/backups/AdventureWorks2019.bak';"

# ou en utilisant le caractère d'échappement `
docker exec -it sql-bootcamp-server /opt/mssql-tools18/bin/sqlcmd `
  -S localhost -U SA -P "YourStrong@Passw0rd" -C `
  -Q "RESTORE FILELISTONLY FROM DISK = '/var/backups/AdventureWorks2019.bak';"
```

Résultats :

- **AdventureWorks2022** → `AdventureWorks2022` + `AdventureWorks2022_log`
- **AdventureWorksDW2022** → `AdventureWorksDW2022` + `AdventureWorksDW2022_log`
- **AdventureWorksLT2022** → `AdventureWorksLT2022_Data` + `AdventureWorksLT2022_Log`

📜 Script de restauration (`scripts/restore-databases.sql`)

```
-- AdventureWorks2022
RESTORE DATABASE AdventureWorks2022
```

```
FROM DISK = '/var/backups/AdventureWorks2022.bak'
WITH MOVE 'AdventureWorks2022' TO '/var/opt/mssql/data/AdventureWorks2022.mdf',
     MOVE 'AdventureWorks2022_log' TO '/var/opt/mssql/data/AdventureWorks2022_log.ldf',
     REPLACE;
GO

-- AdventureWorksDW2022
RESTORE DATABASE AdventureWorksDW2022
FROM DISK = '/var/backups/AdventureWorksDW2022.bak'
WITH MOVE 'AdventureWorksDW2022' TO '/var/opt/mssql/data/AdventureWorksDW2022.mdf',
     MOVE 'AdventureWorksDW2022_log' TO '/var/opt/mssql/data/AdventureWorksDW2022_log.ldf',
     REPLACE;
GO

-- AdventureWorksLT2022
RESTORE DATABASE AdventureWorksLT2022
FROM DISK = '/var/backups/AdventureWorksLT2022.bak'
WITH MOVE 'AdventureWorksLT2022_Data' TO '/var/opt/mssql/data/AdventureWorksLT2022.mdf',
     MOVE 'AdventureWorksLT2022_Log' TO '/var/opt/mssql/data/AdventureWorksLT2022_log.ldf',
     REPLACE;
GO
```

▶️ Exécution du script

```
docker exec -i sql-bootcamp-server /opt/mssql-tools18/bin/sqlcmd \
  -S localhost -U SA -P "YourStrong@Passw0rd" -C \
  -i /scripts/restore-databases.sql


docker exec -i sql-bootcamp-server /opt/mssql-tools18/bin/sqlcmd -S localhost -U SA -P "YourStrong@Passw0rd" -C -i /scripts/restore-databases.sql
```

Logs attendus :

```
RESTORE DATABASE successfully processed ...
```

☑️ Vérification des bases restaurées

```
docker exec -it sql-bootcamp-server /opt/mssql-tools18/bin/sqlcmd \
  -S localhost -U SA -P "YourStrong@Passw0rd" -C \
  -Q "SELECT name FROM sys.databases;"
```

Résultat attendu :

```
master
tempdb
model
msdb
SQLBootcamp
AdventureWorks2022
AdventureWorksDW2022
AdventureWorksLT2022
```

🌐 Connexion via Adminer

- URL : http://localhost:8080
- SGBD : **MS SQL**
- Serveur : `sqlserver` (nom du service dans docker-compose) ou `localhost`
- Utilisateur : `SA`
- Mot de passe : `YourStrong@Passw0rd`

📤 Exporter les données (vers MySQL/Postgres)

Dans **Adminer** :

1. Sélectionner une base
2. Onglet **Export**
3. Format **SQL**
4. Télécharger le dump et l'importer dans le SGBD cible

🚀 Conclusion

Tu as maintenant un environnement complet de **SQL Server sous Docker**, avec **Adminer** comme interface web et les 3 bases **AdventureWorks 2022** restaurées.
Tu peux :

- Explorer les données via Adminer
- Faire des exports `.sql` ou `.csv`
- Connecter un client externe (Azure Data Studio, DBeaver, etc.)