

# Utiliser IBM Db2 sur macOS

Utiliser IBM Db2 sur un Mac nécessite une certaine configuration, car IBM ne fournit pas d'installation native de serveur Db2 pour macOS. Cependant, il existe des moyens pratiques d'exécuter et d'interagir avec Db2 sur un Mac, en fonction de vos besoins—que vous souhaitez exécuter une instance Db2 locale pour le développement/test ou vous connecter à une base de données Db2 distante. Ci-dessous, je vais décrire les approches les plus courantes et efficaces à partir du 28 février 2025.

---

## Option 1: Exécution de Db2 localement sur macOS en utilisant Docker

Comme Db2 n'est pas pris en charge de manière native en tant que serveur sur macOS, la méthode la plus simple pour exécuter une instance Db2 locale consiste à utiliser Docker. Cela fonctionne sur les Mac Intel et Apple Silicon (M1/M2/M3), en utilisant l'émulation si nécessaire.

### Étapes:

#### 1. Installer Docker:

- Téléchargez et installez Docker Desktop pour Mac depuis docker.com. Suivez les instructions d'installation pour votre Mac (Intel ou Apple Silicon).

#### 2. Télécharger l'image de la Db2 Community Edition:

- Ouvrez un terminal et exécutez :

```
docker pull icr.io/db2_community/db2
```

- Cela télécharge l'image de conteneur officielle de la Db2 Community Edition depuis le registre de conteneurs IBM.

#### 3. Exécuter le conteneur Db2:

- Démarrez une instance Db2 avec la commande suivante (personnalisez selon vos besoins) :

```
docker run -itd --name db2server --privileged=true -p 50000:50000 -e LICENSE=accept -e DB2INST1_PASSWORD=your_password
```

- Explication :

- \* `-itd` : Exécute le conteneur en mode interactif, détaché.
- \* `--name db2server` : Nomme le conteneur.
- \* `--privileged=true` : Nécessaire pour que Db2 fonctionne correctement.
- \* `-p 50000:50000` : Mappe le port 50000 (par défaut de Db2) à votre Mac.
- \* `-e LICENSE=accept` : Accepte l'accord de licence.
- \* `-e DB2INST1_PASSWORD=your_password` : Définit le mot de passe pour l'utilisateur `db2inst1` (remplacez `your_password`).

\* -e DBNAME=testdb : Crée une base de données nommée testdb.

#### 4. Vérifier que le conteneur est en cours d'exécution :

- Vérifiez avec :

```
docker ps
```

- Vous devriez voir db2server listé.

#### 5. Accéder à l'instance Db2 :

- Entrez dans le shell du conteneur :

```
docker exec -ti db2server bash -c "su - db2inst1"
```

- Une fois à l'intérieur, vous pouvez utiliser les commandes Db2 comme :

```
db2 connect to testdb
```

```
db2 "SELECT * FROM SYSIBM.SYSDUMMY1"
```

#### 6. Remarques pour les Mac Apple Silicon :

- L'image Db2 est construite pour x86\_64 (Intel), mais Docker Desktop sur Apple Silicon utilise Rosetta 2 pour émuler x86\_64, donc cela devrait fonctionner sans problème. Si vous rencontrez des problèmes de performance, envisagez d'utiliser des outils de virtualisation alternatifs comme Lima (voir Option 3).
- 

### Option 2: Connexion à une base de données Db2 distante en utilisant le pilote IBM Data Server

Si vous n'avez pas besoin d'un serveur Db2 local et que vous souhaitez simplement vous connecter à une base de données Db2 distante existante (par exemple, sur un serveur ou dans le cloud), vous pouvez installer le pilote IBM Data Server pour ODBC et CLI sur votre Mac.

#### Étapes:

##### 1. Télécharger le pilote :

- Pour les Mac Intel : Utilisez le pilote Db2 11.5 (x86\_64).
- Pour les Mac Apple Silicon : Utilisez le pilote Db2 12.1 (ARM64), sorti en novembre 2024.
- Obtenez-le depuis IBM Fix Central : IBM Support Fix Central. Recherchez « IBM Data Server Driver for ODBC and CLI » et sélectionnez la version appropriée pour votre architecture :
  - Intel : macos64\_odbc\_cli.tar.gz (11.5).
  - Apple Silicon : Recherchez le package de pilote ARM64 12.1 (par exemple, clidriver pour macOS ARM).

## 2. Installer le pilote :

- Extrayez le fichier téléchargé :

```
tar -xzf macos64_odbc_cli.tar.gz
```

- Déplacez-le dans un répertoire comme /usr/local/db2\_clidriver :

```
mv clidriver /usr/local/db2_clidriver
```

## 3. Configurer unixODBC :

- Installez unixODBC s'il n'est pas déjà présent (en utilisant Homebrew) :

```
brew install unixodbc
```

- Éditez /usr/local/etc/odbcinst.ini pour enregistrer le pilote :

```
[IBM DB2 ODBC DRIVER]
Description = IBM DB2 ODBC Driver
Driver = /usr/local/db2_clidriver/lib/libdb2.dylib
```

## 4. Configurer la connexion :

- Créez ou éditez /usr/local/etc/odbc.ini (ou ~/.odbc.ini pour les paramètres spécifiques à l'utilisateur) :

```
[MYDB2]
Driver = IBM DB2 ODBC DRIVER
Database = your_db_name
Hostname = your_db_host
Port = 50000
Protocol = TCPIP
UID = your_username
PWD = your_password
```

- Remplacez your\_db\_name, your\_db\_host, your\_username et your\_password par les détails de votre base de données distante.

## 5. Tester la connexion :

- Utilisez isql pour vérifier :

```
isql -v MYDB2
```

- Si cela réussit, vous vous connecterez à la base de données Db2 distante.

## 6. Utilisation avec des applications :

- Vous pouvez maintenant utiliser cette configuration ODBC avec des outils comme Python (pyodbc), Java (JDBC via le pont ODBC), ou toute application compatible ODBC.

## **Option 3: Configuration avancée avec Lima (Optimisation Apple Silicon)**

Pour les utilisateurs d'Apple Silicon cherchant une meilleure performance que l'émulation Docker, Lima fournit une machine virtuelle Linux légère avec le support « Fast Mode » de Rosetta 2. Cela est plus complexe mais offre une expérience similaire à celle d'un environnement natif.

### **Étapes:**

#### **1. Installer Lima et Podman :**

- Installez Homebrew si ce n'est pas déjà fait :

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

- Installez Lima et Podman :

```
brew install lima podman
```

#### **2. Démarrer une VM Lima :**

- Lancez une VM avec le support Rosetta :

```
limactl start --name=db2-vm --rosetta
```

#### **3. Exécuter Db2 dans Podman :**

- À l'intérieur de la VM, utilisez Podman pour exécuter le conteneur Db2 (similaire aux étapes Docker ci-dessus) :

```
podman run -itd --name db2server --privileged=true -p 50000:50000 -e LICENSE=accept -e DB2INST1_PASSWORD=yourpassword
```

#### **4. Accéder depuis macOS :**

- Connectez-vous à l'instance Db2 depuis votre Mac en utilisant l'IP de la VM Lima (par exemple, 192.168.5.15:50000) avec un outil client ou le pilote ODBC.

---

### **Conseils supplémentaires**

- **Considérations de licence :** La Db2 Community Edition est gratuite pour le développement et les tests. Pour une utilisation en production ou pour se connecter à Db2 sur z/OS ou iSeries, vous pourriez avoir besoin d'une licence Db2 Connect.
- **Performance :** Sur Apple Silicon, l'émulation fonctionne mais peut être plus lente. Le support ARM64 natif (pilote Db2 12.1) est limité au client, pas au serveur, pour l'instant.
- **Tests automatisés :** Si vous utilisez unixODBC pour des tests automatisés, assurez-vous que vos scripts de test font référence au bon DSN de odbc.ini.

## Quelle option choisir ?

- **Développement/local/test** : Utilisez l'Option 1 (Docker) pour la simplicité.
- **Accès à une base de données distante** : Utilisez l'Option 2 (pilote ODBC) pour vous connecter aux instances Db2 existantes.
- **Performance Apple Silicon** : Utilisez l'Option 3 (Lima) si vous avez besoin d'une configuration locale à haute performance.

Faites-moi savoir si vous avez besoin d'aide pour une étape spécifique !