

# Guide Complet : Installation Data Lakehouse Local (Dremio + MinIO) sur Docker Windows

Ce guide explique comment monter un environnement Big Data local. Nous utiliserons le dépôt Git d'Alex Merced. **Objectif :** Connecter **Dremio** (Moteur de requête) à **MinIO** (Stockage) en utilisant le connecteur **Amazon S3**.

## 1. Prérequis et Installation

### A. Installer Docker Desktop pour Windows

Si ce n'est pas déjà fait :

1. Téléchargez l'installateur : [Lien Officiel Docker Desktop](#)
2. Installez-le en gardant les options par défaut (WSL 2 recommandé).
3. **Redémarrez votre ordinateur.**

### B. Récupérer le projet

1. Créez un dossier pour vos projets.
2. Ouvrez un terminal (PowerShell ou Invite de commandes).
3. Clonez le dépôt :

```
git clone https://github.com/bkablam11/demo-minio-dremio.git
```

4. **Entrez dans le dossier** (Étape critique pour éviter l'erreur "configuration file not found") :

```
cd demo-minio-dremio
```

## 2. Préparation des Images et Données (Anti-Erreurs)

### A. Téléchargement manuel des images

Pour éviter l'erreur `TLS handshake timeout` due à une connexion lente ou instable, téléchargez les images principales une par une avant de lancer le tout :

```
# Image Dremio (avec Superset intégré)
docker pull alexmerced/dremio-superset

# Image MinIO (Stockage)
docker pull minio/minio
```

## B. Placement des données (Seed Data)

Le projet est configuré pour charger automatiquement des fichiers dans MinIO si vous les placez au bon endroit sur votre Windows.

- **Pour MinIO :** Mettez vos fichiers (CSV, Parquet, JSON) dans le dossier : `./minio-data` (situé dans le dossier que vous avez cloné).

---

## 3. Lancement des Services

Une fois les images téléchargées, Allez dans Docker et lancer l'orchestration des containers.

```
# Orchestration des containers
docker-compose up -d minio dremio
# Verification des containers allumés
docker ps
# Arret des containers
docker-compose down
```

**Vérification :** Ouvrez Docker Desktop. Vous devriez voir un groupe `dremio-demo-env` avec les conteneurs `minio`, `dremio`, etc. allumés en vert.

---

## 4. Configuration de MinIO (Le "Lake")

1. Accédez à l'interface : <http://localhost:9001>

2. Identifiants :

- User : `admin`
- Password : `password`

3. Vérifiez que vos fichiers placés dans le dossier `./minio-data` apparaissent bien dans le bucket. Sinon, créez un bucket nommé `datalake` et uploadez un fichier manuellement.

The screenshot shows the MinIO Object Browser interface. On the left, there's a sidebar with options like 'Create Bucket', 'Filter Buckets', 'Buckets' (listing 'datalake' and 'datalakehouse'), 'Documentation', 'License', and 'Sign Out'. The main area is titled 'datalake' and shows a single file named 'iris.csv' with a size of 3.9 KIB. The file was created on Sun, Feb 15 2026 15:02:56 (UTC) and last modified Today, 15:04.

## 5. Configuration de Dremio (Le "House") - ÉTAPE CRUCIALE

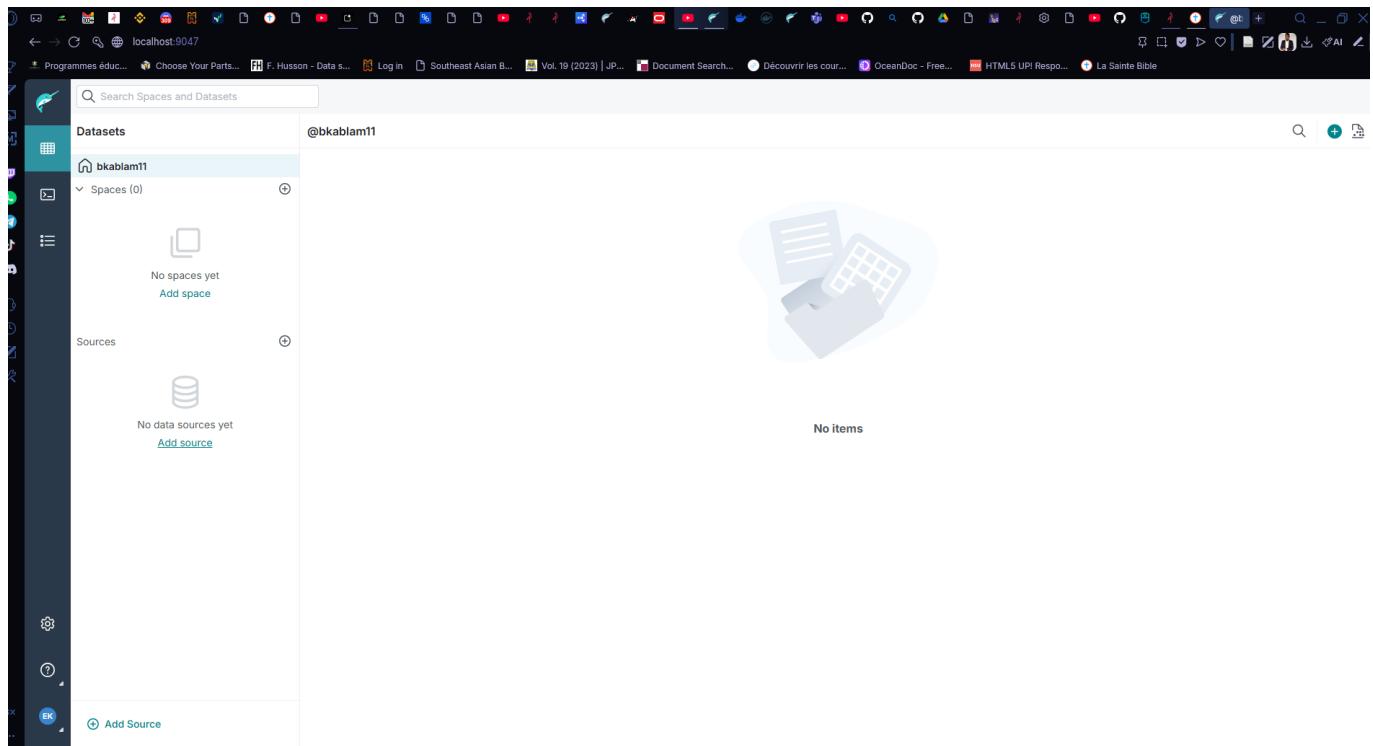
C'est ici que nous connectons Dremio à MinIO en utilisant le protocole S3.

1. Accédez à l'interface : <http://localhost:9047>
2. Créez votre compte administrateur.
3. Cliquez sur le bouton **+ Add Source** (en bas à gauche).
4. Sélectionnez **Amazon S3**.

The screenshot shows the 'Create Admin Account' page of the Dremio sign-up process. It features a cartoon dolphin logo and a welcome message: 'We are excited to have you on board! The first thing you need to do is set up an administrator account.' The form fields are as follows:

First Name	Last Name
Edjabrou Ulrich Blanchard	KABLAM
Username	Email
bkablam11	bkablam11@gmail.com
Password	(password field)
Confirm Password	(password field)

At the bottom, there are 'Next' and 'Privacy' buttons.



### 3. Connexion de Dremio à MinIO (S3)

Pour connecter Dremio à votre stockage MinIO local, nous allons utiliser le connecteur **Amazon S3** en mode compatibilité.

1. Cliquez sur le bouton **Add Source** (le + bleu) et sélectionnez **Amazon S3**.
2. Remplissez le formulaire comme suit :

#### A. Onglet "General" (Authentification)

Remplissez les informations d'identification définies dans votre fichier `docker-compose.yml` :

- **Name :** `MinioData` (Ce sera le nom de la source dans Dremio).
- **Authentication :** Laissez sur `AWS Access Key`.
- **AWS Access Key :** `admin`
- **AWS Access Secret :** `password`
- **Encrypt connection :**  **DÉCOCHEZ OBLIGATOIREMENT CETTE CASE.**
  - Note : MinIO tourne en HTTP local sans certificat SSL. Si vous laissez cochée, la connexion échouera.

New Amazon S3 Source

**General**



Amazon S3 Source

Advanced Options

Name  
MinioData

Reflection Refresh

Metadata

**Authentication**

AWS Access Key  EC2 Metadata  AWS Profile  No Authentication

All or allowlisted (if specified) buckets associated with this access key or IAM role to assume (if specified) will be available.

AWS Access Key  
admin

AWS Access Secret  
.....

IAM Role to Assume

Encrypt connection

**Public Buckets**

Buckets

**Cancel** **Save**

## B. Onglet "Advanced Options" (Configuration Réseau)

C'est ici que nous disons à Dremio de ne pas aller sur le vrai Amazon AWS, mais sur notre conteneur Docker.

1. Cochez la case **Enable compatibility mode** (si disponible).
2. Dans la section **Connection Properties**, ajoutez les propriétés suivantes une par une :

Name (Nom)	Value (Valeur)	Explication Technique
fs.s3a.endpoint	minio:9000	<b>Crucial :</b> Indique l'adresse du conteneur dans le réseau Docker.
fs.s3a.path.style.access	true	Force l'URL sous la forme <b>domaine/bucket</b> (requis par MinIO).
dremio.s3.compat	true	Active les correctifs de compatibilité S3 pour Dremio.

**⚠️ Attention au piège :** Ne mettez pas **localhost:9000** dans le endpoint ! Dremio tourne dans un conteneur. Pour lui, **localhost** c'est lui-même. Il doit contacter le conteneur **minio** via son nom de service Docker.

### 3. Cliquez sur **Save**.

The screenshot shows the 'Source Settings' dialog box for a new source. The 'Advanced Options' tab is selected. Configuration includes:

- General:** 'Enable compatibility mode' (checked), 'Apply requester-pays to S3 requests' (unchecked).
- Advanced Options:** 'Enable file status check' (checked), 'Enable partition column inference' (unchecked).
- Reflection Refresh:** 'Metadata' tab selected.
- Root Path:** '/'
- Server side encryption key ARN:** Empty field.
- Default CTAS Format:** 'ICEBERG' (selected from dropdown).
- Connection Properties:**

Name	Value
fs.s3a.endpoint	minio:9000
fs.s3a.path.style.access	true
dremio.s3.compat	true

At the bottom right are 'Cancel' and 'Save' buttons.

The screenshot shows the Dremio Data Explorer interface. On the left, the sidebar shows a dataset named 'bkablam11' with a single space named 'MinioData'. The main area displays the 'MinioData' dataset with two sub-folders: 'datalake' and 'datalakehouse'. Both have a 'Jobs' count of 0.

## 6. Récapitulatif des Accès (URLs)

Voici les adresses pour accéder à tous vos services locaux une fois lancés :

- **Dremio (Requêtes SQL) :** <http://localhost:9047>
- **MinIO (Stockage) :** <http://localhost:9001>

## 🛠️ Dépannage des erreurs fréquentes

Erreur	Cause probable	Solution
<code>TLS handshake timeout</code>	Connexion internet saturée lors du téléchargement des images.	Faire les <code>docker pull</code> manuellement un par un (voir Étape 2).
<code>no configuration file provided</code>	Vous n'êtes pas dans le bon dossier.	Faites <code>cd le_nom_du_dossier</code> avant de lancer docker-compose.
<code>Unsupported or unrecognized SSL message</code>	Dremio essaie de parler HTTPS à MinIO HTTP.	<b>Décochez</b> "Encrypt connection" dans la source Dremio.
<code>UnknownHostException</code>	Dremio ne trouve pas le serveur.	Dans <code>fs.s3a.endpoint</code> , mettez <code>minio:9000</code> au lieu de localhost.