



# LAB : Installation d'Odoo dans un environnement Docker sous Windows

# Objectif du LAB

L'objectif de ce lab est d'apprendre à **installer et exécuter Odoo dans un environnement Docker sous Windows**, en utilisant **Docker Desktop**, **Docker Compose** et **Git Bash** comme terminal principal.

À la fin de ce lab, vous serez capable de :

- Installer Docker Desktop sur Windows
- Utiliser Git Bash pour exécuter les commandes Docker
- Comprendre le rôle des conteneurs (Odoo et PostgreSQL)
- Déployer Odoo avec Docker Compose
- Accéder à l'interface web d'Odoo

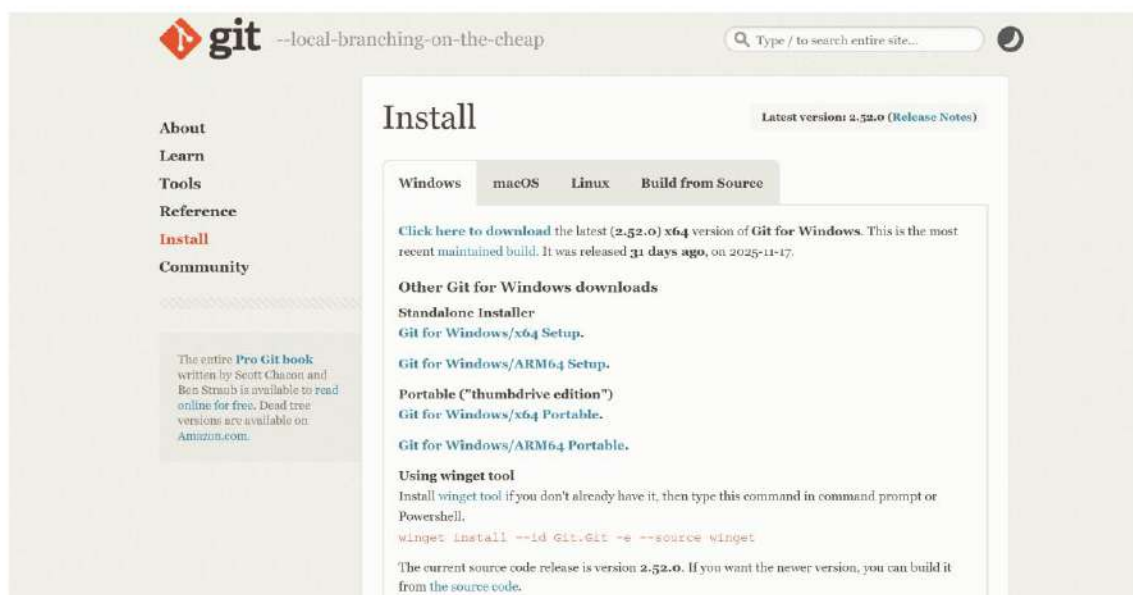
## Prérequis

- Windows 10 ou Windows 11 (64 bits)
- Virtualisation activée dans le BIOS
- Connexion Internet
- Droits administrateur
- Git for Windows installé (Git Bash inclus)

## Étape 1 : Installation de Git Bash

Ressource : <https://www.youtube.com/watch?v=hC0IRtdbV7o>

1. Accédez au site officiel :  
 <https://git-scm.com/download/win>
2. Téléchargez **Git for Windows**.
3. Lancez l'installation et laissez les options par défaut.
4. Une fois installé, ouvrez **Git Bash** depuis le menu Démarrer.

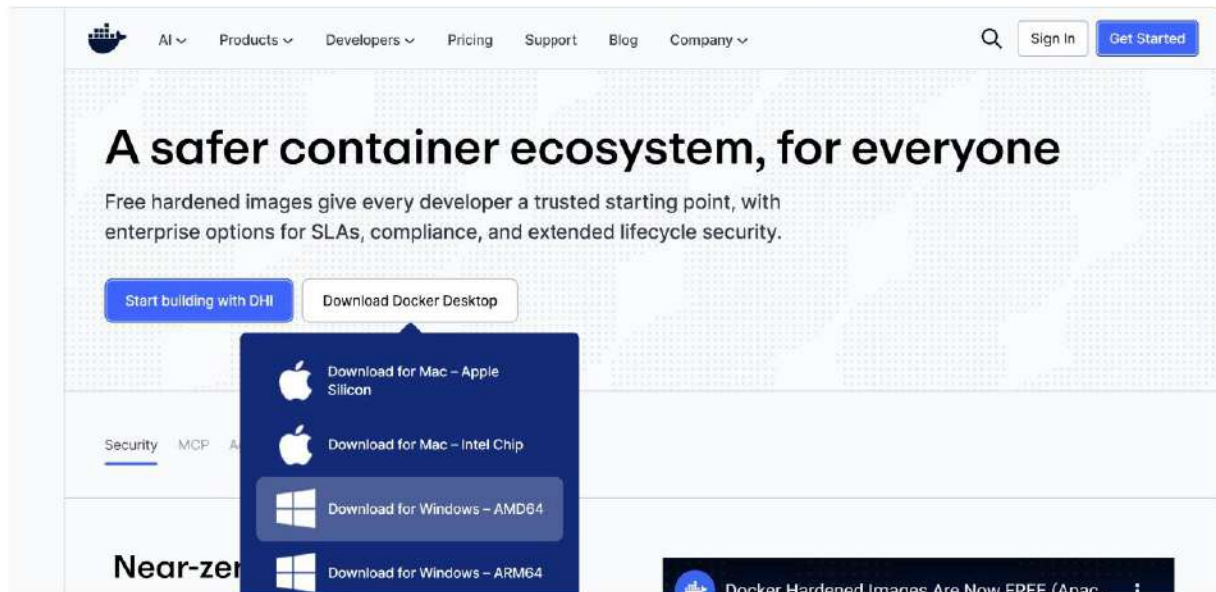


## Étape 2 : Installation de Docker Desktop

Ressource <https://www.youtube.com/watch?v=fsR8fj7iCNY>

1. Accédez au site officiel de Docker :

👉 <https://www.docker.com/products/docker-desktop/>



2. Téléchargez **Docker Desktop for Windows**.
3. Lancez l'installation.
4. Pendant l'installation vérifier l'installation de **WSL 2 dans votre machine**. Comme vous pouvez le voir sur l'image ci-dessus, WSL n'est pas installé.

```
Invite de commandes
Microsoft Windows [version 10.0.26100.6584]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\aitda>wsl
Sous-système Windows pour Linux devez être mis à jour vers la dernière version pour continuer. Vous pouvez effectuer une
mise à jour en exécutant « wsl.exe --update ».
Pour plus d'informations, visitez https://aka.ms/wslinstall

Appuyez sur une touche pour installer Sous-système Windows pour Linux.
Appuyez sur CTRL-C ou fermez cette fenêtre pour annuler.
Cette invite expirera dans 60 secondes.
Opération abandonnée
```

5. Pour l'installer, tapez la commande 'wsl.exe' puis appuyez sur la touche Entrée.

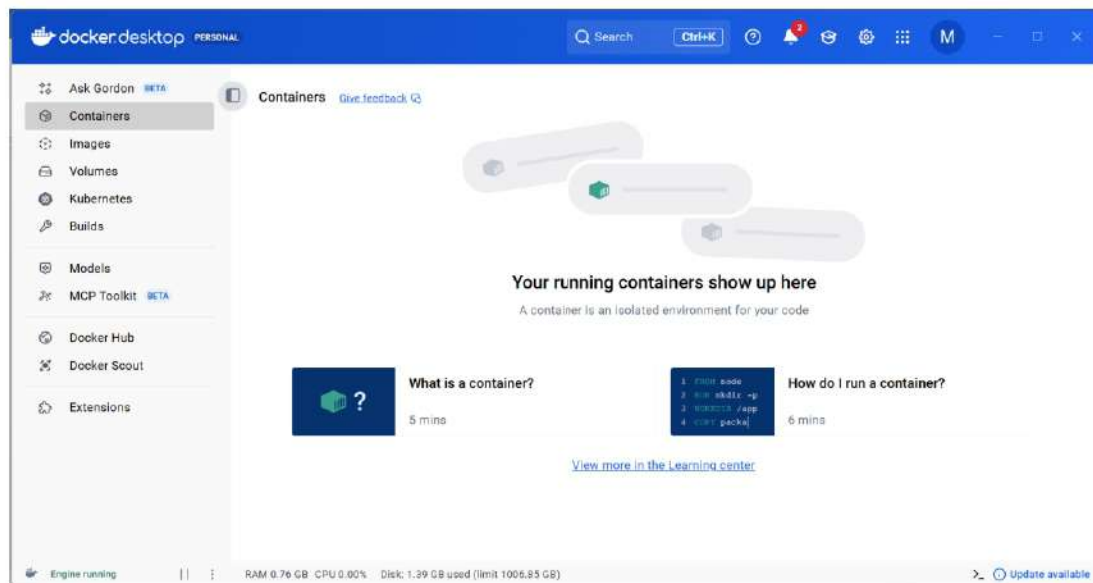
```
Invite de commandes
Pour plus d'informations, visitez https://aka.ms/wslinstall

Appuyez sur une touche pour installer Sous-système Windows pour Linux.
Appuyez sur CTRL-C ou fermez cette fenêtre pour annuler.
Cette invite expirera dans 60 secondes.
L'opération demandée nécessite une élévation.
Téléchargement en cours : Sous-système Windows pour Linux 2.6.3
Installation en cours : Sous-système Windows pour Linux 2.6.3
Sous-système Windows pour Linux 2.6.3 a été installé.
L'opération a réussi.
Sous-système Windows pour Linux n'a aucune distribution installée.
Vous pouvez résoudre ce problème en installant une distribution avec les instructions ci-dessous :

Utiliser wsl.exe --list --online' pour répertorier les distributions disponibles
et 'wsl.exe --install <Distro>' à installer.

C:\Users\aitda>
```

6. Redémarrez votre ordinateur.
7. Lancez **Docker Desktop** et assurez-vous qu'il est en état **Running**.

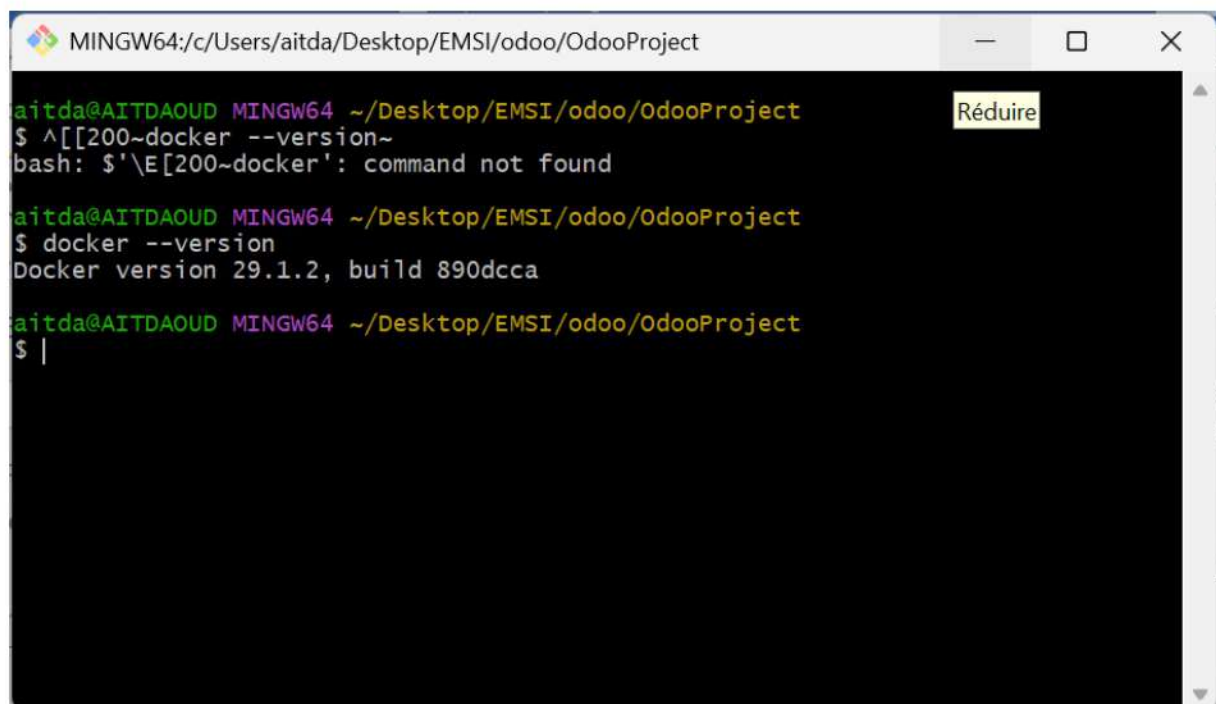


## Étape 3 : Vérification de Docker avec Git Bash

1. Ouvrez **Git Bash**.
2. Tapez la commande suivante :

```
docker --version
```

3. Vérifiez que la version de Docker s'affiche correctement.





```
C:\WINDOWS\system32\cmd. X + v
C:\Users\aitda>docker
Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:
  run          Create and run a new container from an image
  exec        Execute a command in a running container
  ps          List containers
  build       Build an image from a Dockerfile
  bake        Build from a file
  pull        Download an image from a registry
  push        Upload an image to a registry
  images      List images
  login        Authenticate to a registry
  logout       Log out from a registry
  search       Search Docker Hub for images
  version      Show the Docker version information
  info         Display system-wide information

Management Commands:
  ai*          Docker AI Agent - Ask Gordon
  builder      Manage builds
  buildx*      Docker Buildx
  compose*     Docker Compose
  container    Manage containers
  context       Manage contexts
  debug*       Get a shell into any image or container
  desktop*     Docker Desktop commands
  extension*   Manages Docker extensions
```

Ou bien : ouvrir l'invite de commandes (CMD) et taper la commande `wsl -l -v`.

```
C:\Windows\system32\cmd.e: X + v
Microsoft Windows [Version 10.0.26100.4349]
(c) Microsoft Corporation. All rights reserved.

C:\Users\TechjunkieAman>wsl -l -v
NAME                STATE              VERSION
* docker-desktop    Running            2
C:\Users\TechjunkieAman>
```

## Étape 4 : Création de la structure du projet Odoo

1. Créez un dossier principal et ouvrez le dans VS code :

```
odoo-docker
```

2. À l'intérieur, créez les dossiers suivants :

```
odoo-docker
├── addons
├── config
└── docker-compose.yml
```

3. Dans Git Bash, placez-vous dans votre dossier de travail (par exemple Documents) :

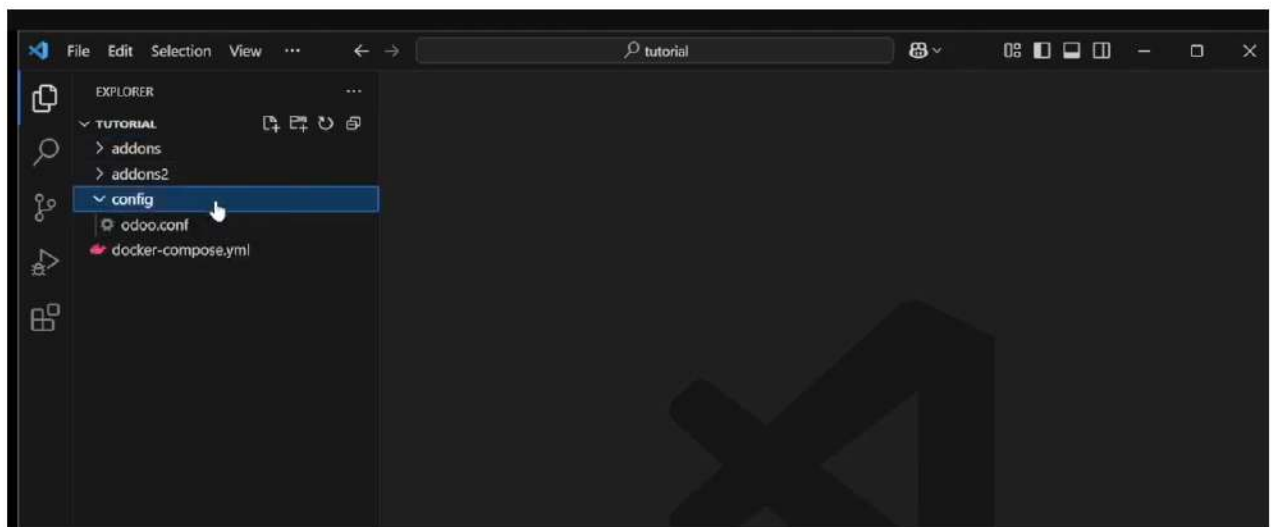
```
cd ~/Documents
```

4. Créez le dossier du projet :

```
mkdir odoo-docker
cd odoo-docker
```

3. Créez les sous-dossiers nécessaires :

```
mkdir addons config
```



## Étape 5 : Création du fichier docker-compose.yml

1. Dans le dossier odoo-docker, créez le fichier :

```
touch docker-compose.yml
```

2. Ouvrez-le avec un éditeur de texte (VS Code recommandé) :

```
docker-compose.yml
```

### 3. Collez le contenu suivant :

```
version: "3.8"

services:
  db:
    image: postgres:16
    container_name: odoo_db
    ports:
      - "5432:5432"
    environment:
      POSTGRES_USER: odoo
      POSTGRES_PASSWORD: odoo
      PGDATA: /var/lib/postgresql/data/pgdata
    volumes:
      - odoo-db-data:/var/lib/postgresql/data

  odoo17:
    image: odoo:17.0
    container_name: odoo_app
    depends_on:
      - db
    ports:
      - "8069:8069"
    volumes:
      - odoo-web-data:/var/lib/odoo
      - ./config:/etc/odoo
      - ./addons:/mnt/extra-addons
    command: >
      odoo -d odoo_db
      -i base
      --db_user=odoo
      --db_password=odoo
      --db_host=db

volumes:
  odoo-web-data:
  odoo-db-data:
```

### 4. Enregistrez le fichier.

## Étape 6 : Création du fichier odoo.conf

### 1. Dans le dossier config, créez le fichier :

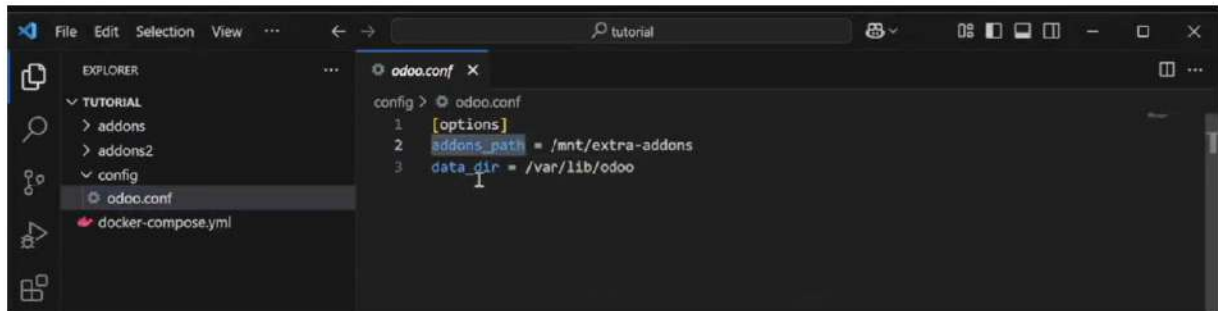
```
touch odoo.conf
```

### 2. Ouvrez-le avec un éditeur de texte (VS Code recommandé) :

```
odoo.conf
```

### 3. Collez le contenu suivant :

```
[options]
addons_path = /mnt/extra-addons
data_dir = /var/lib/odoo
```



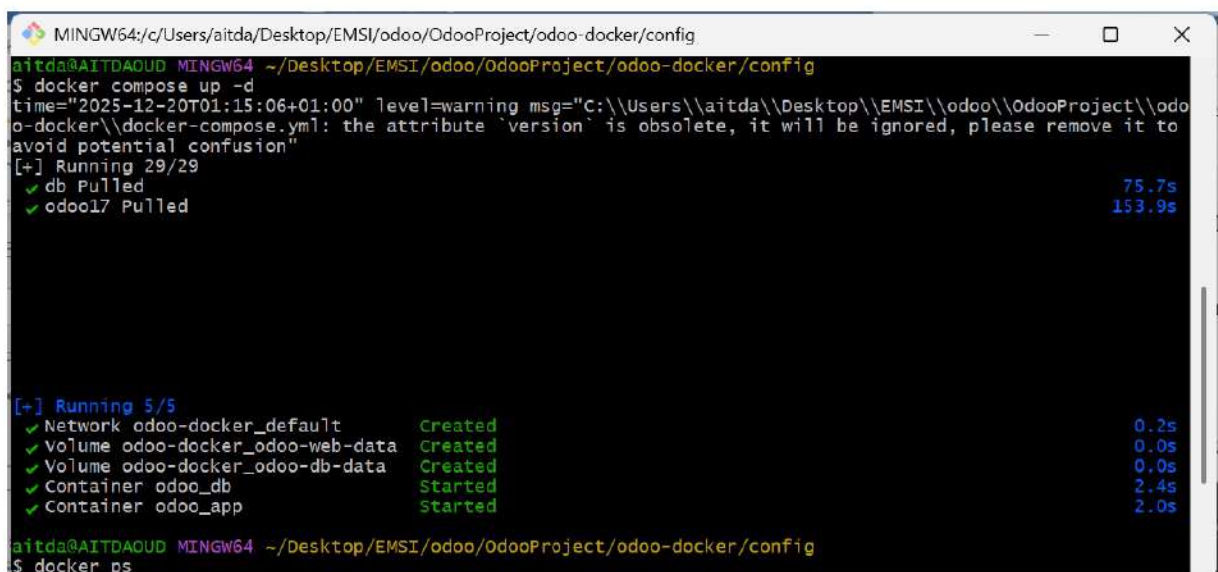
## Étape 7 : Lancement de Odoo avec Docker Compose (Git Bash)

Une fois les étapes précédentes terminées, nous sommes prêts à lancer Docker Compose pour exécuter nos conteneurs Docker. Assurez-vous que Docker Desktop est en cours d'exécution avant d'utiliser les commandes ci-dessous.

1. Ouvrez un terminal, PowerShell ou Git Bash dans le répertoire du projet et exécutez la commande suivante :

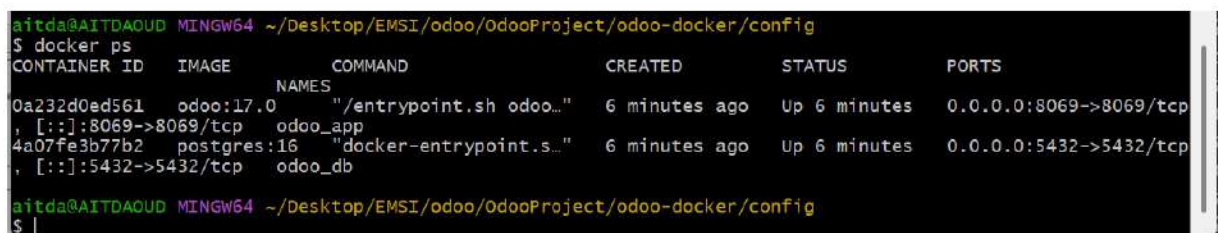
```
docker compose up -d
```

2. Patientez pendant le téléchargement des images.



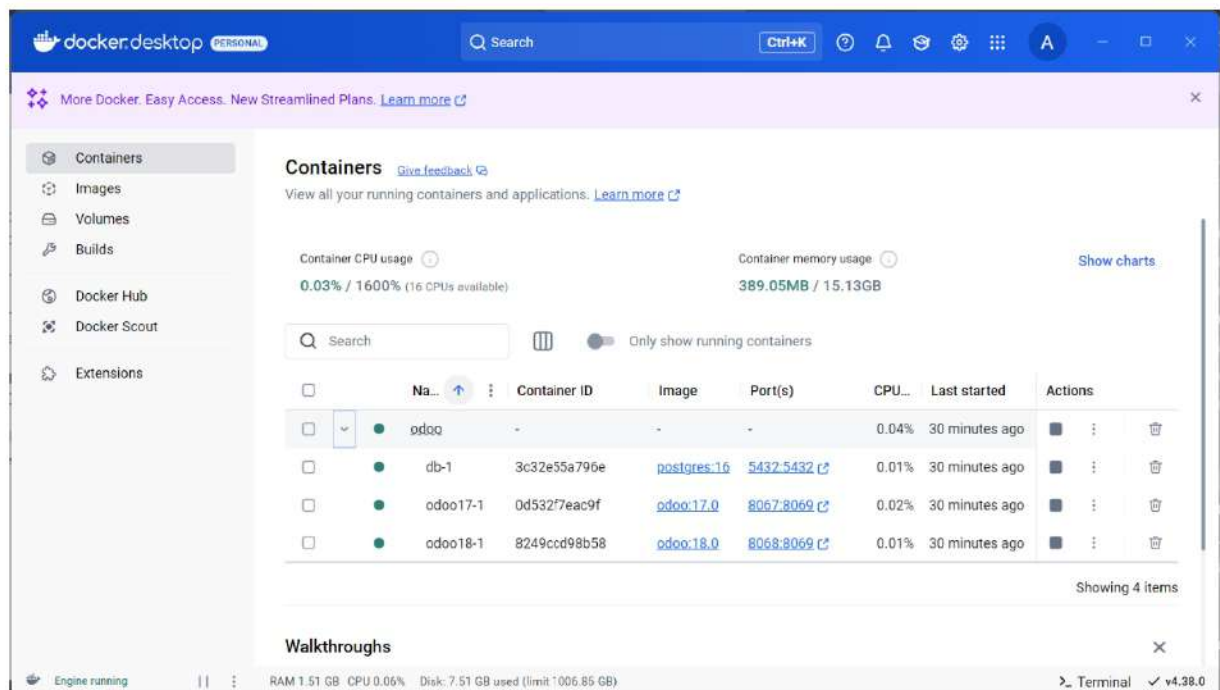
3. Vérifiez que les conteneurs sont actifs :

```
docker ps
```

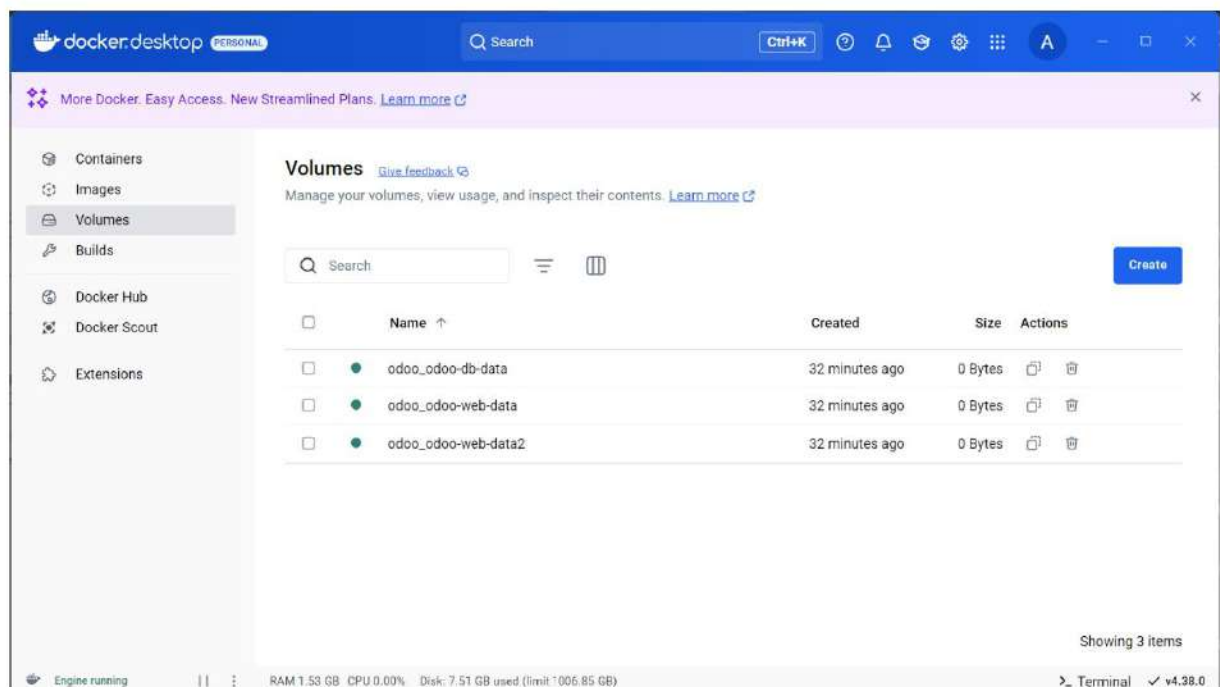




#### 4. Ouvrir docker desktop :



#### 5. Vérifiez que les volumes sont actifs :

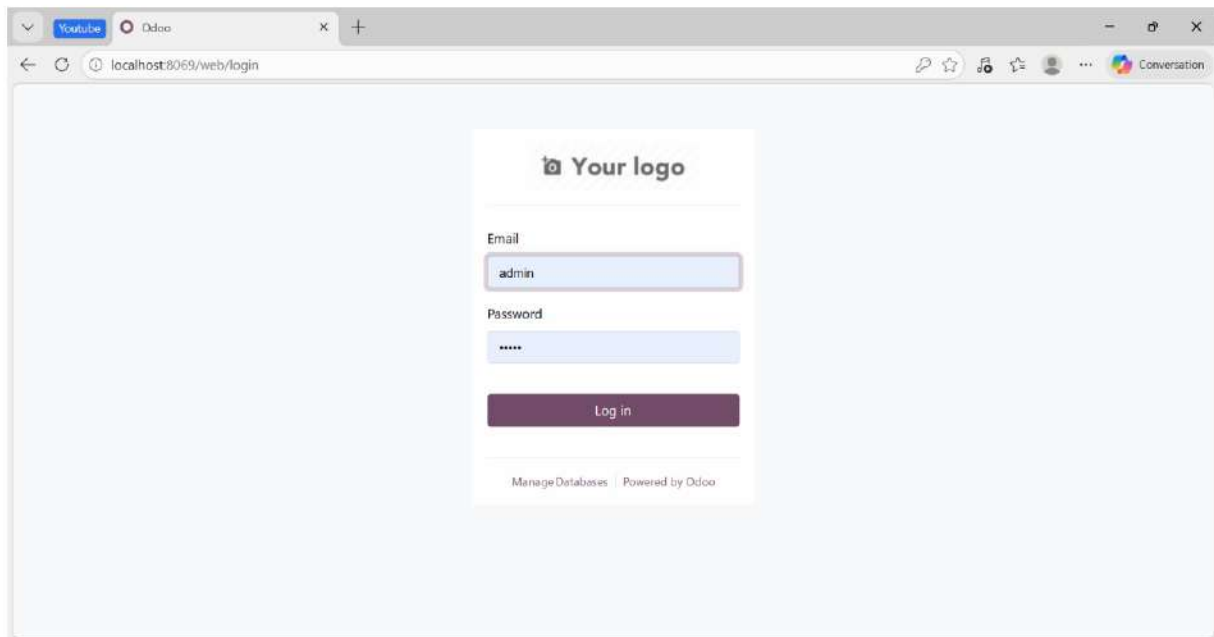


## Étape 8 : Accès à l'interface Odoo

1. Ouvrez un navigateur web.
2. Accédez à l'adresse :

`http://localhost:8069`

3. L'écran de création de base de données Odoo s'affiche.



3. Saisir le nom d'utilisateur et le mot de passe.

Email : admin  
Password : admin

