

Projet personnelle — Conteneurisation

Docker

Installation et prise en main de Docker Desktop

Docker Desktop — WSL 2

Auteur	Cyrille COUR
Type de projet	Personnel — socle technique (outil transverse)
Stack technique	Docker Desktop / WSL 2 / Windows
Objectif	Comprendre et exploiter la conteneurisation, en vue d'un profil DevOps
Usages associés	Pi-hole (perso), maintien d'applications PHP legacy (travail)

1. Contexte et objectif

Bien qu'étudiant en BTS SIO SLAM, je vise à terme un profil bien plus large qu'un « simple » développeur : je voudrais être DevOps, capable de développer mais aussi d'intervenir sur l'infrastructure réseau et matérielle d'une entreprise.

J'ai découvert Docker en effectuant des recherches sur le métier de DevOps, et ça m'a tout de suite plu.

Pourquoi Docker : c'est pour cette raison, et d'autres détaillées ci-dessous, que j'ai décidé de l'installer chez moi.

2. Qu'est-ce que Docker ?

Docker est un outil de conteneurisation.

- Il permet d'exécuter une application avec tout ce dont elle a besoin (code, dépendances, configuration).
- Chaque application tourne dans un conteneur isolé.
- Le comportement est identique sur tous les environnements (PC, serveur, cloud) — ça évite les « chez moi ça marche, chez toi non ».

2.1 Pourquoi Docker est utile pour un développeur

- Garantit un environnement de travail identique sur toutes les machines.
- Simplifie l'installation des projets.
- Permet de déployer les applications de manière rapide, fiable et sans mauvaises surprises.

3. Cas d'usage personnels et professionnels

Une fois installé, Docker s'est rapidement révélé utile dans deux contextes différents :

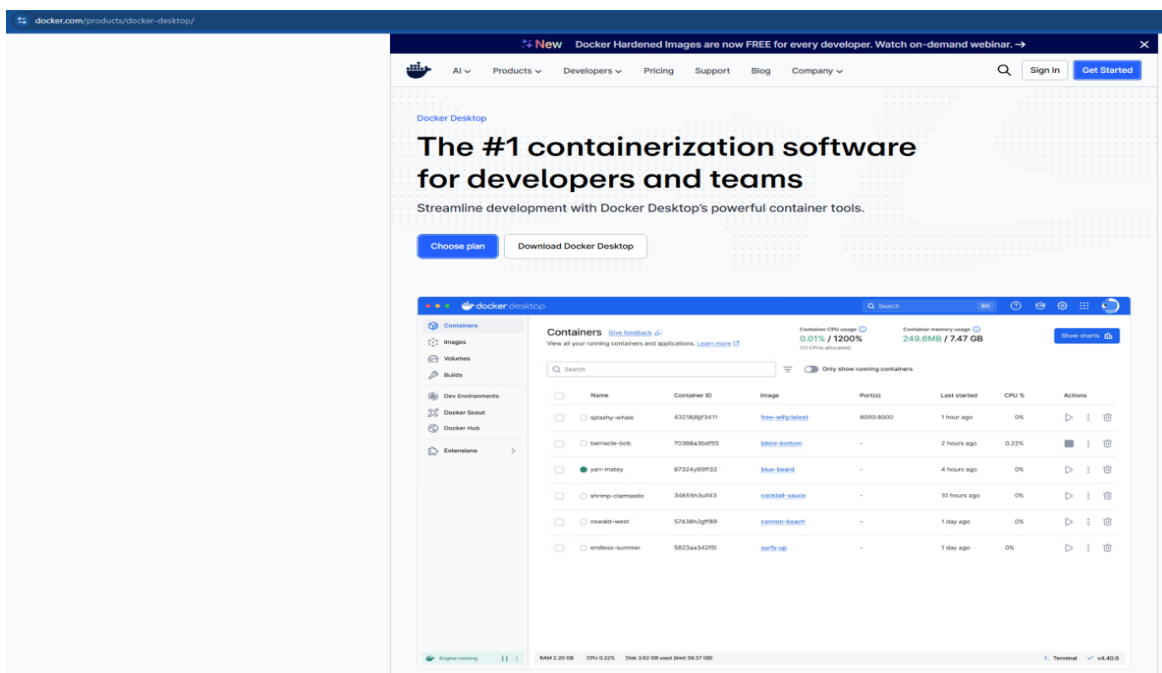
- **Usage personnel — Pi-hole** : premier conteneur lancé chez moi, via un fichier docker-compose.yml, pour héberger le bloqueur de publicités réseau Pi-hole sans avoir à gérer d'installation native ni de dépendances système.
- **Usage professionnel — maintien d'applications PHP legacy** : au travail, certaines applications tournent sous d'anciennes versions de PHP qu'il serait risqué ou impossible d'installer directement sur les postes actuels. Docker permet de faire tourner ces versions dans un conteneur isolé, sans polluer l'environnement de la machine hôte ni entrer en conflit avec d'autres projets utilisant des versions plus récentes.

Point commun des deux usages : dans les deux cas, Docker évite d'installer un service ou une version spécifique directement sur le système hôte — le conteneur encapsule tout, et peut être arrêté, recréé ou déplacé sans rien casser autour.

4. Installation de Docker Desktop

4.1 Téléchargement

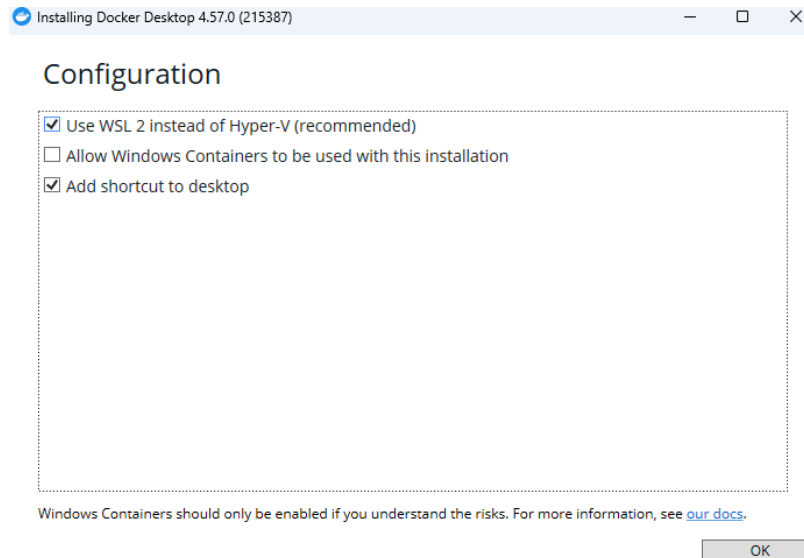
Page de téléchargement de Docker Desktop (docker.com/products/docker-desktop) :



4.2 Configuration de l'installation

Lors de l'installation, une fenêtre de configuration propose plusieurs options :

- **Use WSL 2 instead of Hyper-V** (cochée) : WSL 2 est beaucoup plus performant que Hyper-V pour faire tourner les conteneurs sous Windows.
- **Allow Windows Containers** (décochée) : non nécessaire ici, l'objectif étant de faire tourner des conteneurs Linux.
- **Add shortcut to desktop** (cochée) : ajoute une icône sur le bureau pour un accès rapide.



Fenêtre de configuration de l'installation — WSL 2 sélectionné

Après avoir validé sur OK, un redémarrage du PC est nécessaire pour terminer l'installation.

4.3 Mise à jour de WSL

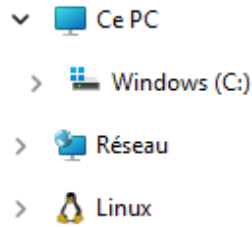
Au redémarrage, Docker demande de mettre à jour le sous-système Windows pour Linux. Commande à exécuter dans un PowerShell en administrateur :

```
wsl --update
```

```
PS C:\WINDOWS\system32> wsl --update
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.
Recherche des mises à jour.
La version la plus récente de Sous-système Windows pour Linux est déjà installée.
PS C:\WINDOWS\system32>
```

Mise à jour du sous-système Windows pour Linux via PowerShell

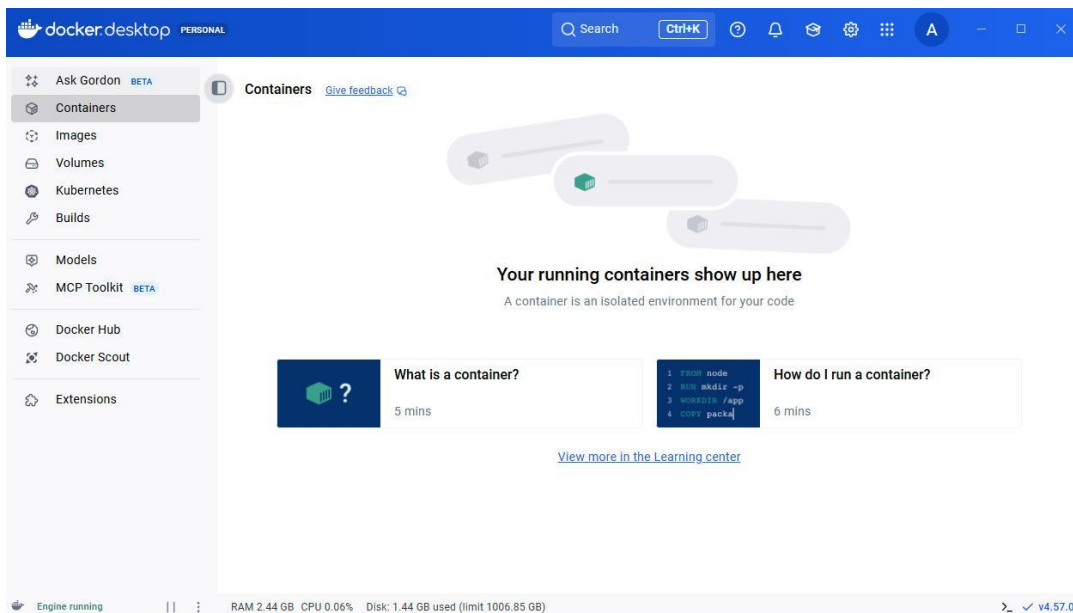
Une fois l'opération terminée, un système de fichiers Linux apparaît dans l'explorateur Windows, aux côtés du disque C: et du réseau :



Sous-système Linux visible dans l'explorateur de fichiers Windows

4.4 Docker Desktop prêt à l'emploi

Après un dernier redémarrage de Docker, l'application est officiellement prête à être utilisée :



Docker Desktop opérationnel — vue Containers, aucun conteneur encore lancé

5. Bilan

Docker est désormais installé et opérationnel. Ce socle technique a immédiatement servi de base à un premier projet personnel (Pi-hole) et s'est révélé tout aussi pertinent en entreprise pour isoler des applications tournant sous d'anciennes versions de PHP, sans risquer de perturber l'environnement des autres projets. Cette double utilisation — perso et pro — confirme l'intérêt de Docker comme outil transverse pour un profil orienté DevOps.