

# Docker - Comment et pourquoi l'installer ?

28 Novembre 2023

Version	2
Date	6 décembre 2023
Rédigé par	BINGINOT Etienne MONTAGNE Erwann

## Mises à jour du document

Version	Date	Modification réalisée
1	24 novembre 2023	Création du document
2	06 décembre 2023	Ajout de l'installation Ubuntu

# Table des matières

<b>1</b>	<b>Pourquoi utiliser Docker</b>	<b>4</b>
1.1	Qu'est-ce qu'un conteneur . . . . .	4
1.2	Comment fonctionne Docker . . . . .	4
1.3	Les différents fichiers de Docker . . . . .	4
1.3.1	Le Dockerfile . . . . .	4
1.3.2	Le fichier docker-compose.yml . . . . .	4
<b>2</b>	<b>Installer Docker sur Windows</b>	<b>6</b>
2.1	Installer l'application Docker . . . . .	6
2.2	OPTIONNEL (mais très recommandé) : Installation de WSL . . . . .	6
2.3	Utiliser Docker . . . . .	9
<b>3</b>	<b>Installer Docker sur Linux</b>	<b>9</b>
3.1	Préparation de l'installation . . . . .	9
3.2	Installation de Docker . . . . .	10
3.3	Premier lancement de Docker . . . . .	10

# 1 Pourquoi utiliser Docker

Docker est une application permettant de gérer des conteneurs, ayant pour but d'isoler efficacement une application d'une autre. Chaque personne utilisant Docker doit pouvoir avoir exactement la même configuration que son voisin : si un bug arrive dans le code de quelqu'un, il arrivera également dans le code de l'autre (la solution technique ne dépend donc pas de la plateforme, et les applications ne peuvent pas se gêner l'une l'autre).

## 1.1 Qu'est-ce qu'un conteneur

Un conteneur est un concept qui dépasse Docker : bien que celui-ci soit un gestionnaire de conteneurs, le concept en lui-même existe depuis longtemps. Le but d'un conteneur est de pouvoir isoler une application du système d'exploitation qui le fait tourner.

En somme, un conteneur est une machine virtuelle légère : celui-ci se base sur le système d'exploitation de la machine pour simuler un système d'exploitation sans perte de performance. Ceux-ci sont donc totalement indépendants les uns des autres.

## 1.2 Comment fonctionne Docker

Comme dit précédemment, Docker est donc un gestionnaire de conteneurs. Celui-ci fonctionne via un système d'images : une image est une représentation d'un conteneur à un moment donné. Il est donc possible à partir de cette image de la configurer à notre souhait en installant tout ce que nous souhaitons dessus, et ainsi de créer notre propre nouvelle image !

Ces images sont accessibles directement depuis des dépôts en ligne (Docker Hub) et sont régulièrement mis à jour. Il est ainsi possible de trouver une image correspondant à un serveur Linux vierge, à une installation de php dans la version de notre choix, à un serveur Windows, ou même Minecraft ! Il est possible de conteneuriser tout et n'importe quoi.

## 1.3 Les différents fichiers de Docker

### 1.3.1 Le Dockerfile

Le Dockerfile est le fichier principal de Docker : c'est lui qui va créer l'image de votre conteneur.

Celui-ci fonctionne d'une manière simple :

- Une première directive (FROM) permet d'indiquer l'image qu'on importe (on se base donc sur une image publique qui sera automatiquement téléchargée par Docker et que l'on personnalisera ensuite)
- On utilise ensuite des directives comme COPY (copier des fichiers dans le conteneur), RUN (exécuter une commande dans le conteneur, comme dans un bash), ... pour personnaliser à notre souhait notre image (copie des fichiers d'un projet, installation de dépendances (apt install), etc)
- Enfin, Docker va créer une nouvelle image à partir des directives que nous lui avons données, qui est maintenant accessible dans la liste des images docker (docker image ls, ou dans l'onglet images sur l'application)

La commande permettant de dire à Docker de lancer la création de l'image depuis un dockerfile est `docker build`.

### 1.3.2 Le fichier docker-compose.yml

Ce fichier permet de dire à Docker comment lancer vos conteneurs. En effet, depuis ce fichier, il est possible de lancer plusieurs conteneurs à la fois en spécifiant leurs images respectives (Docker va directement rechercher dans votre liste d'images). Il est ainsi possible de spécifier comment les conteneurs vont interagir entre eux (ouverture / fermeture de ports réseaux), qui doit démarrer en premier, les commandes à exécuter

au démarrage, etc.

Ainsi, le lancement d'une application Docker se résume bien souvent à une unique commande :  
`docker-compose up`

Cette commande permet à Docker de lancer la lecture du fichier `docker-compose` et donc de lancer les conteneurs et l'application.

## 2 Installer Docker sur Windows

### 2.1 Installer l'application Docker

La première étape de l'installation Docker correspond tout d'abord à télécharger l'application Docker pour Windows :

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

Il s'agit d'un exécutable, et l'application s'installera comme n'importe quelle autre application Windows.



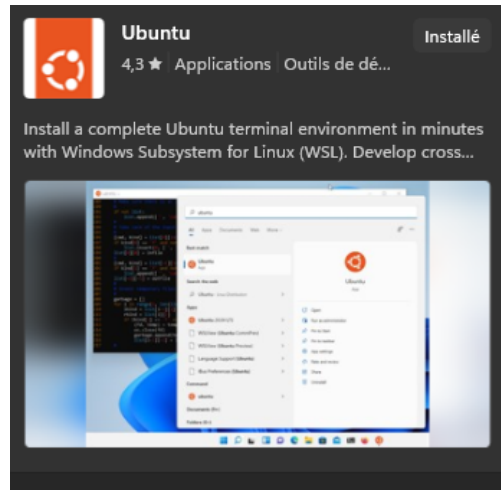
### 2.2 OPTIONNEL (mais très recommandé) : Installation de WSL

WSL est un sous-noyau Linux pour Windows (une forme de machine virtuelle légère pour Windows permettant de lancer des conteneurs sur une machine Linux).

Les conteneurs utiliseront donc un système Linux pour fonctionner ce qui leur apportera plus de fiabilité, plus d'efficacité, etc.

L'installation de WSL sur Windows est extrêmement simple :

- Lancer le Microsoft Store
- Rechercher Ubuntu
- Prendre la version de base d'Ubuntu (Ubuntu sans numéro de version derrière)
- Redémarrer votre ordinateur

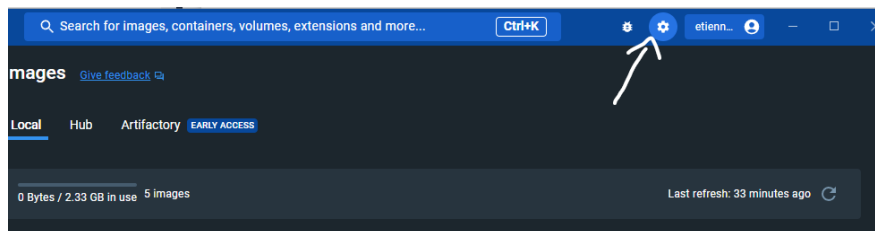


Il vous faut ensuite ouvrir un terminal sur votre système windows (rechercher le programme cmd), et exécuter la commande `wsl -install`. Si tout se déroule sans soucis, on vous demandera de créer un utilisateur pour votre système ubuntu (mettez les informations que vous voulez), et ensuite un bash s'ouvrira directement dans votre Ubuntu. Vous pouvez ensuite redémarrer votre ordinateur.

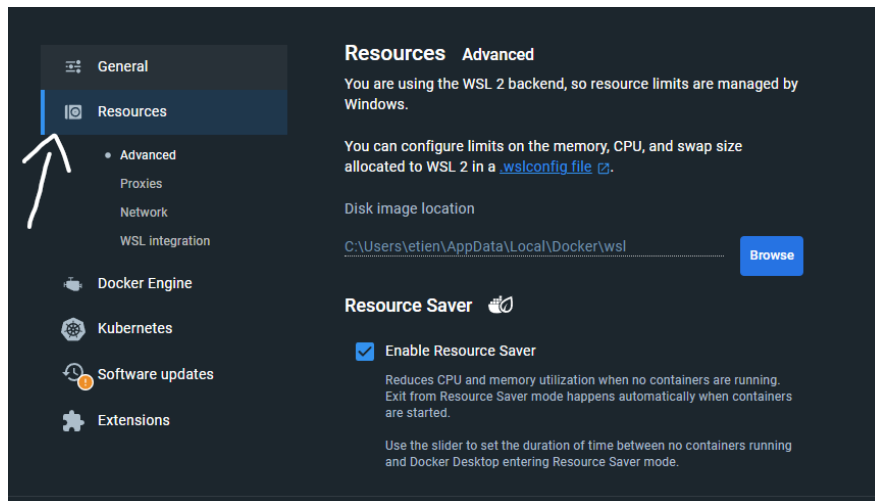
Maintenant, WSL est installé sur votre système Windows !

Il faut ensuite configurer Docker pour qu'il utilise WSL :

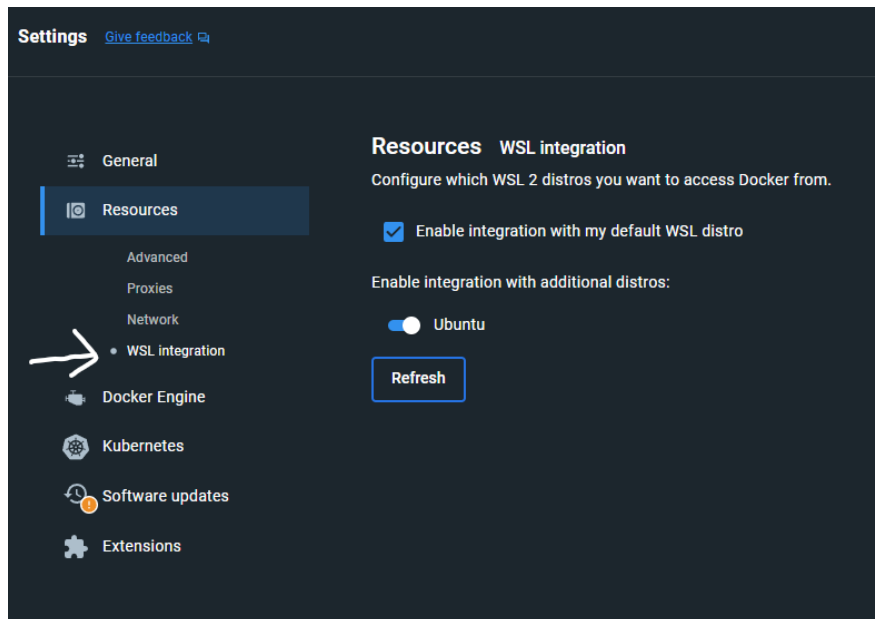
- Démarrer Docker
- Accéder aux paramètres de Docker



- Accéder à l'onglet ressources

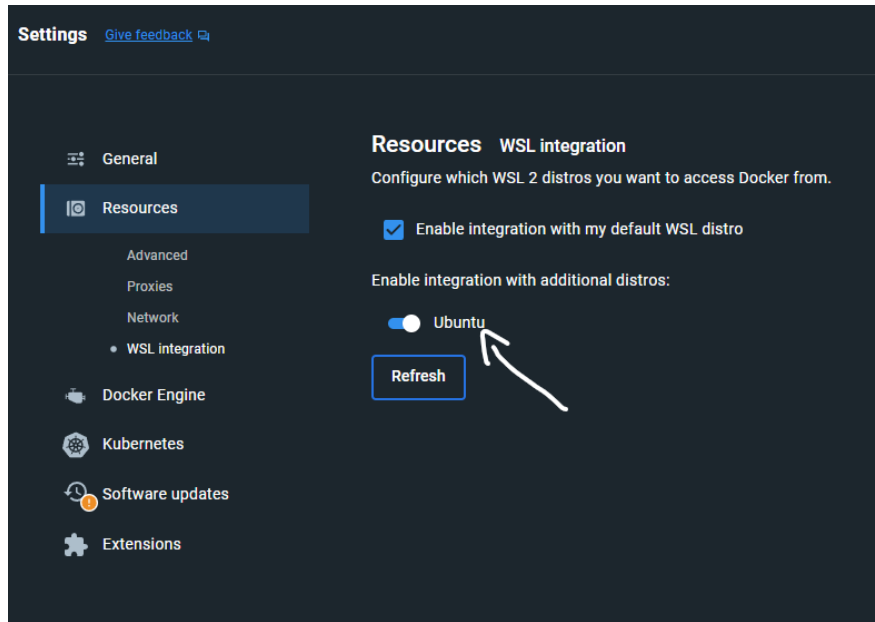


- Accéder à l'onglet WSL integration



- Activer l'intégration avec Ubuntu





## 2.3 Utiliser Docker

Docker est maintenant installé sur votre système !

Pour l'utiliser, vous pouvez, dans un terminal windows, réaliser les commandes docker suivantes :

- docker build : permet de créer l'image docker du dockerfile de votre dossier
- docker-compose up : permet de lancer les conteneurs définis dans votre fichier docker-compose.yml (permet souvent de lancer toute votre application)
- docker image ls : permet de lister vos images
- docker ps : permet de lister vos conteneurs
- docker run : permet de lancer un conteneur à partir d'une image

## 3 Installer Docker sur Linux

### 3.1 Préparation de l'installation

Afin de pouvoir installer Docker sur une machine Ubuntu, il va falloir installer quelques packages.

Pour ceux n'ayant pas un environnement gnome, il faut tout d'abord installer le package gnome-terminal :

```
sudo apt install gnome-terminal
```

Il faudra aussi supprimer les vieilles versions de Docker qui ont pu potentiellement être installées :

```
sudo apt remove docker-desktop
```

Pour supprimer proprement les anciennes versions de Docker, il faudra aussi supprimer les fichiers de configuration via les commandes suivantes :

```
rm -r $HOME/.docker/desktop  
sudo rm /usr/local/bin/com.docker.cli  
sudo apt purge docker-desktop
```

## 3.2 Installation de Docker

Pour installer Docker nous allons effectuer quelques commandes :  
`sudo apt-get update`

Il va maintenant falloir trouver la version de Docker qui vous convient en fonction de l'architecture de votre machine. Vous devriez pouvoir trouver celle-ci en suivant le lien suivant :

[Cliquez ici pour télécharger le .deb](#)

Une fois le fichier .deb téléchargé, exécutez la commande suivante :

`sudo apt-get install ./docker-desktop-"version"-"arch".deb`

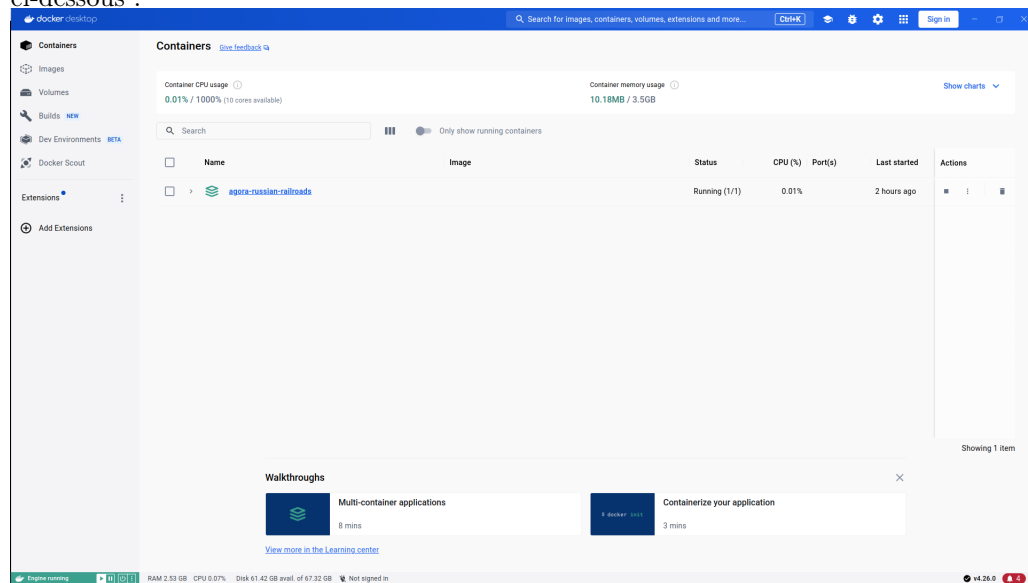
Remplacer version et arch par les valeurs données dans le fichier .deb téléchargé

Félicitations, Docker est maintenant installé sur votre machine

## 3.3 Premier lancement de Docker

Pour lancer Docker pour la première fois, utilisez la commande suivante :  
`systemctl --user start docker-desktop`

L'application Docker doit maintenant être démarrée et vous devriez avoir une fenêtre semblable à celle ci-dessous :



Normalement vous pouvez aussi lancer Docker depuis votre menu "Demarrer"