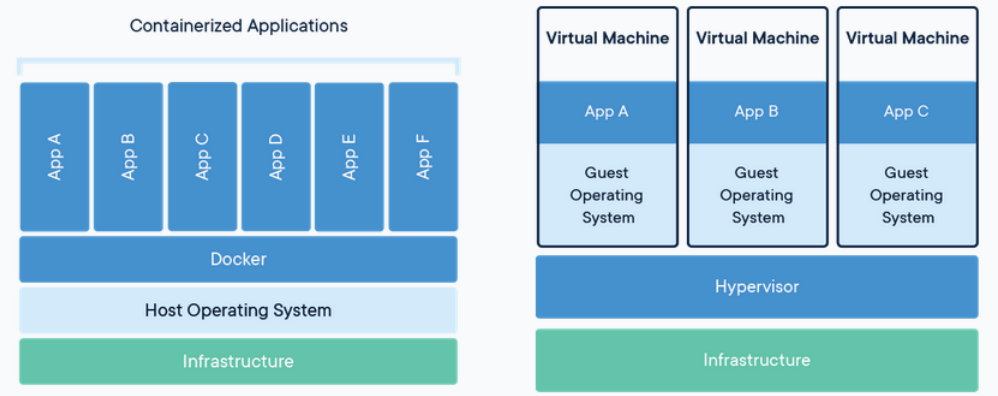
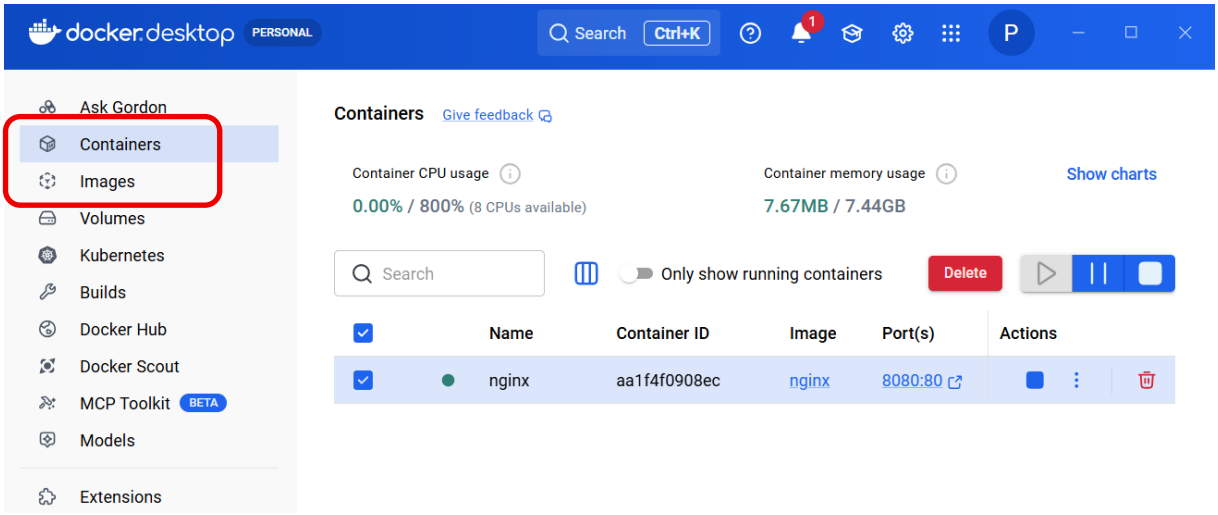
Comparaison entre conteneurs Docker et machines virtuelles



- Si **Docker Engine** s'installe nativement sur **Linux**, **Docker Desktop** est une application facile à installer qui permet de créer des conteneurs et donc d'exécuter des applications conteneurisées directement sur un ordinateur ou serveur **Windows**.

  Un conteneur est un environnement qui contient un service (règle de base : un processus par conteneur).

- **Docker Desktop** offre une interface graphique permettant d'interagir avec Docker afin de gérer **images** et **conteneurs** :



- **Docker Desktop** est un outil essentiel pour les développeurs ainsi que pour les administrateurs système et réseau travaillant dans des environnements hybrides DEVOPS.

  → Les développeurs peuvent dans un premier temps exprimer le besoin de monter rapidement un service. **Docker Desktop** leur permettra de créer très simplement un serveur afin de répondre à une problématique ponctuelle de développement. Outre la rapidité et la simplicité, l'intérêt sera de lancer plusieurs versions de PHP, de Python, de postgreSQL, de MySQL, de MariaDB… sur la même machine. Docker Desktop est donc un outil très pratique pour tester rapidement un projet, une idée.

  Nous pourrions utiliser éventuellement XAMPP ou WAMP pour faire ce genre de test mais nous serions limités, par exemple, à une version précise de PostgreSQL ou de MySQL.

  La force de **Docker Desktop** est donc de permettre de créer rapidement et tester des environnements de fonctionnement différents :

Apache version… + PHP version… + Mysql version…

Nginx version… + PHP version… + MariaDB version…

Caddy version… + PHP version… + PostgreSQL version…

→ Avec **Docker Desktop**, **Docker Compose** est inclus dans l'installation de base. Il n'est pas besoin de l'installer.

Docker-Compose permet de lancer plusieurs conteneurs Docker en même temps tout en facilitant la mise en réseau de ces conteneurs.

> Les services sont rarement indépendants les uns des autres. Par exemple Un serveur Web communique avec une base de données. Cet ensemble de services est appelé une stack applicative sous Docker (par exemple, Wordpress + MySQL + PHPMyAdmin).

- Avec Docker Desktop, deux architectures peuvent être utilisées :

→ **WSL 2 (Windows Subsystem for Linux 2) :**

La fonctionnalité WSL 2, lorsqu'elle est activée, permet d'exécuter un noyau Linux complet sur Windows et ainsi, de faire fonctionner des conteneurs Linux.

→ **Hyper-V :**

Hyper-V est le système de virtualisation de Windows. Il sera utilisé pour exécuter une machine virtuelle qui héberge Docker. Cette architecture sera plus particulièrement choisie par les utilisateurs qui ont pour objectif d'exécuter des conteneurs Windows.

➔ Dans le cadre de ce cours, nous utiliserons WSL 2.

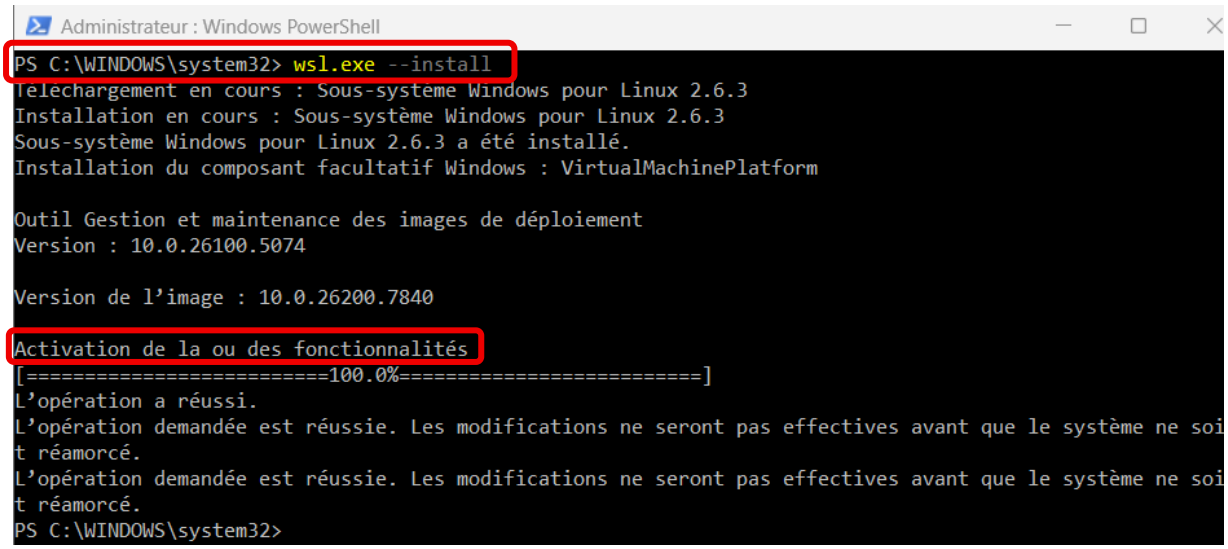
En résumé :

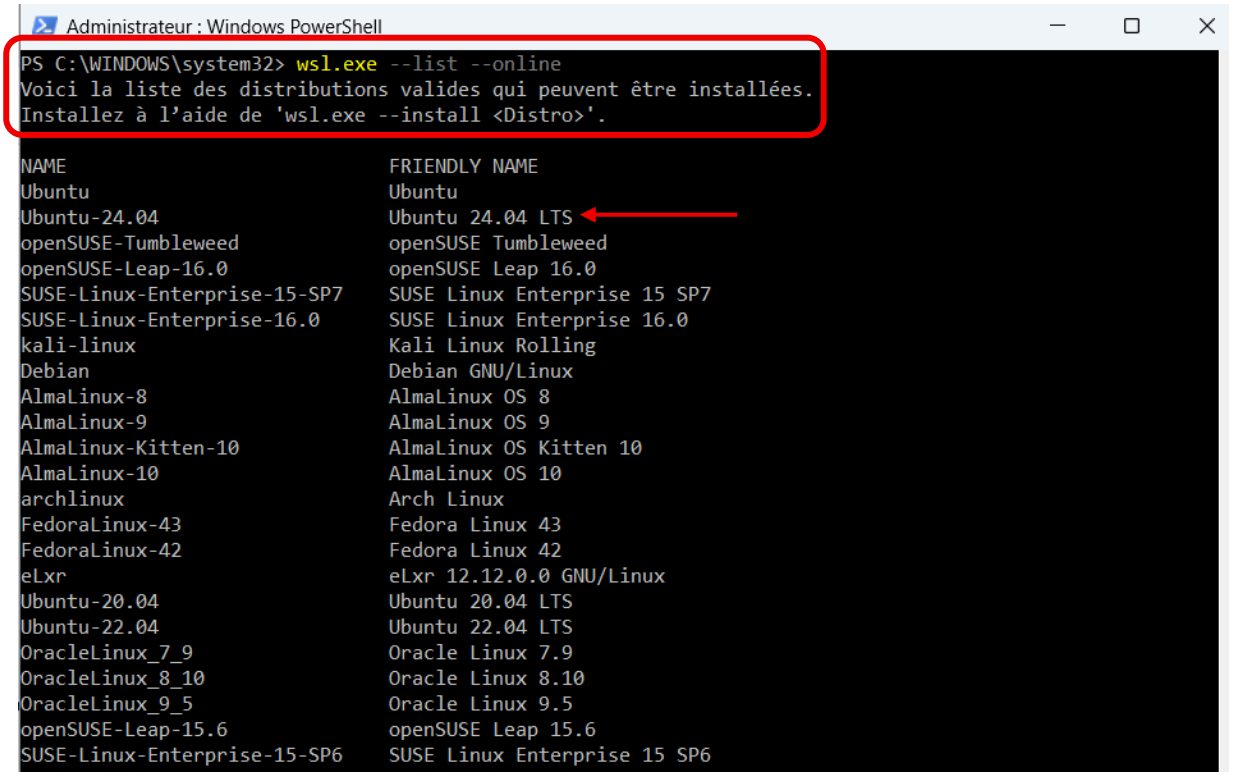
	Docker Engine	Docker Desktop
Plateforme	Linux natif	Windows, macOS
Interface	CLI uniquement	Interface graphique + CLI
Licence	Open source	Licence commerciale : gratuit pour un usage personnel et payant pour les entreprises de plus de 250 employés
Kubernetes	Non inclus	Inclus
Docker Compose	Installation séparée	Inclus
VM	Non (natif)	Oui (WSL2)

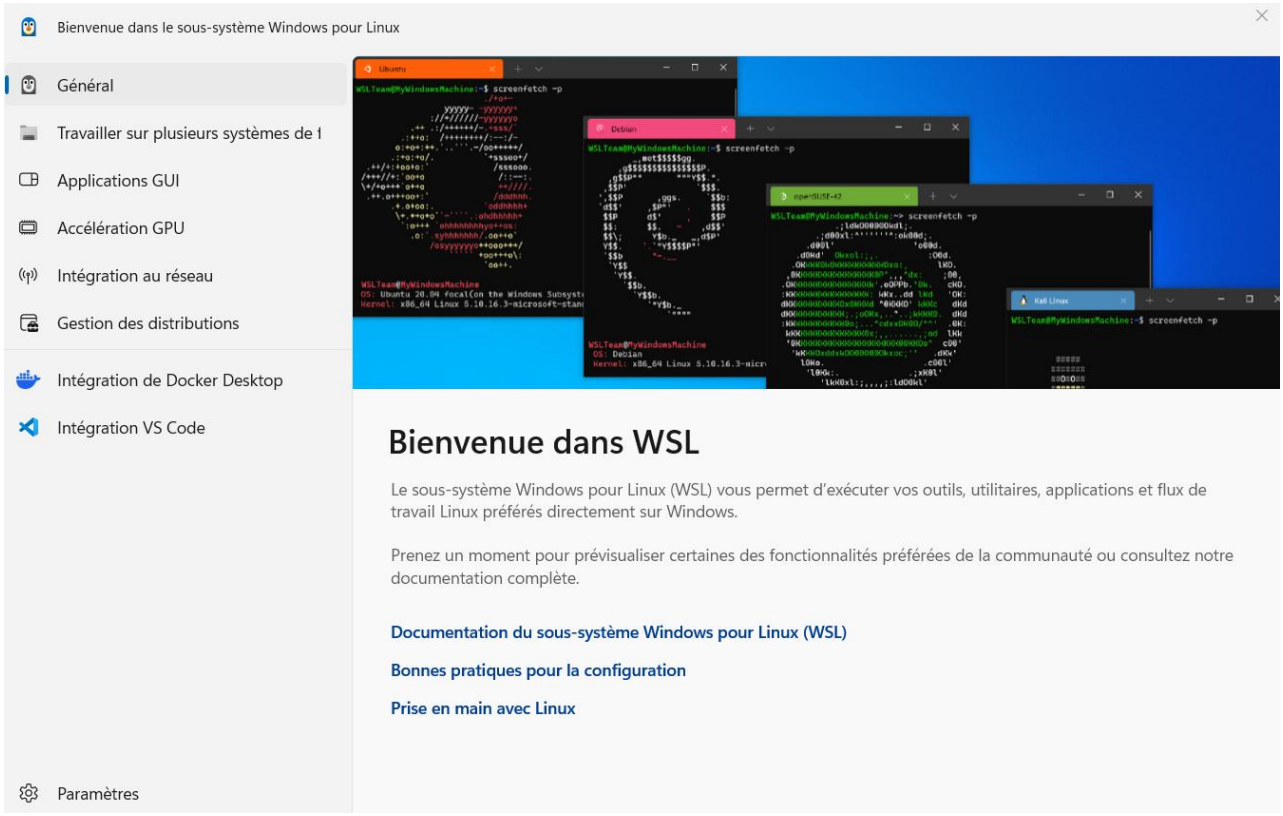
Infrastructure (serveur Linux) → Docker Engine

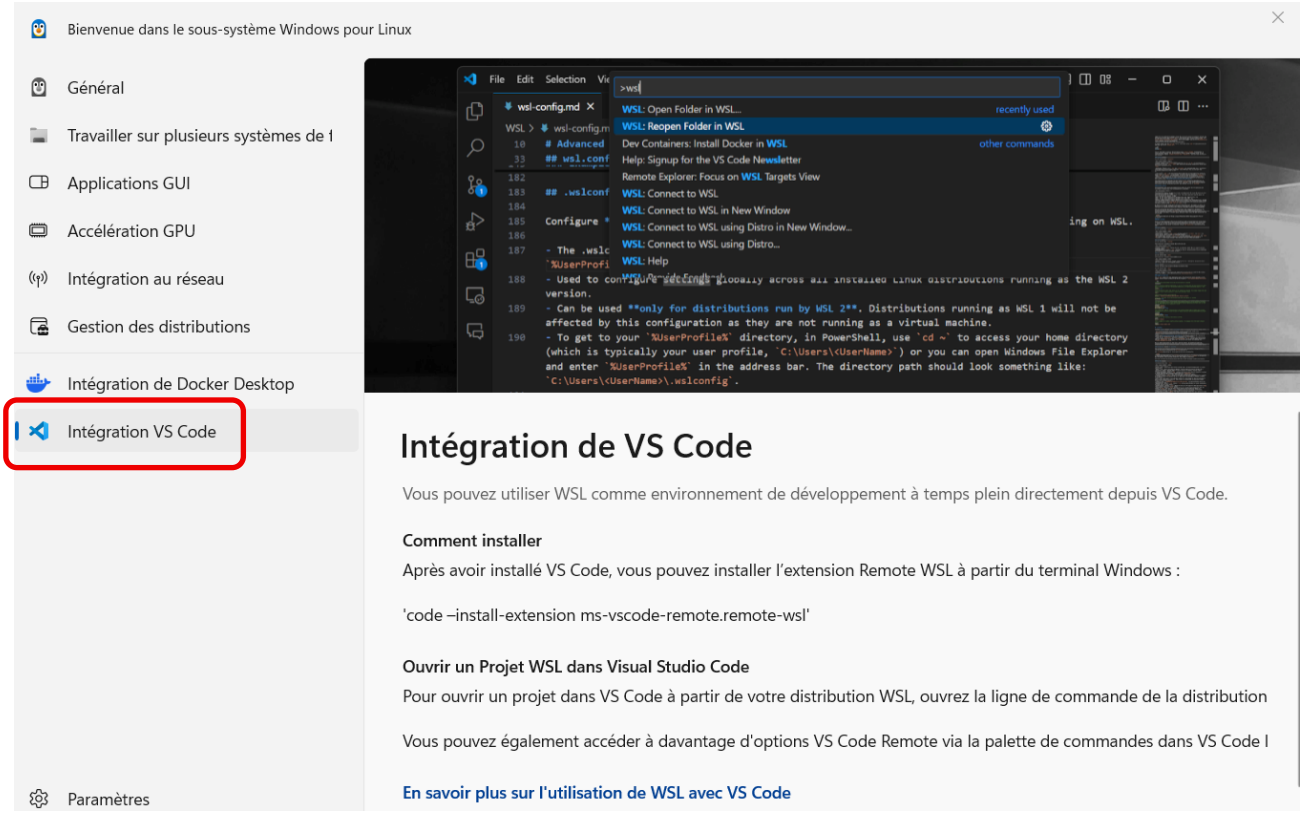
Développement Windows/macOS → Docker Desktop

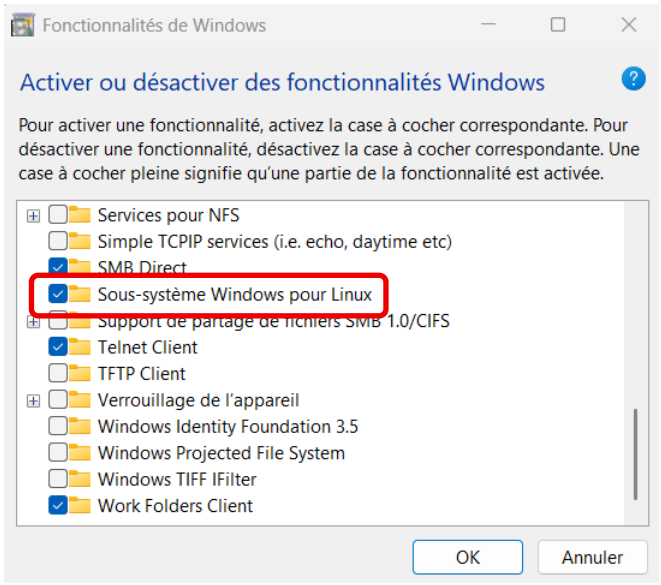
1. Installation / Activation de WSL.

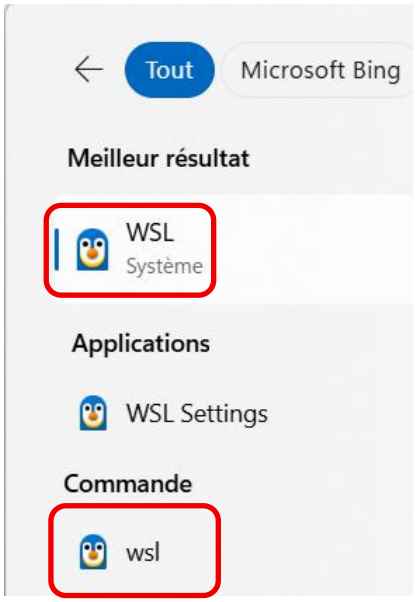


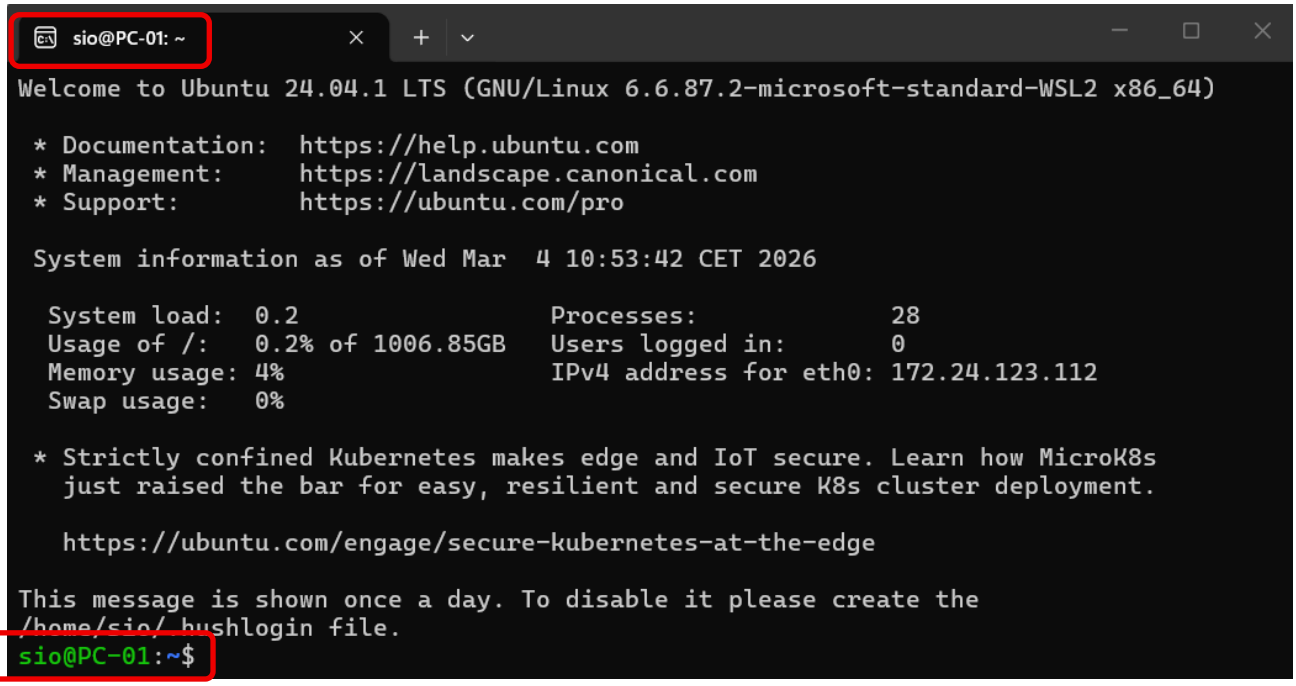



Bienvenue dans le sous-système Windows pour Linux
Général
Travailler sur plusieurs systèmes de f
Applications GUI
Accélération GPU
Intégration au réseau
Gestion des distributions
Intégration de Docker Desktop
Intégration VS Code
Paramètres
Bienvenue dans WSL
Le sous-système Windows pour Linux (WSL) vous permet d'exécuter vos outils, utilitaires, applications et flux de travail Linux préférés directement sur Windows.
Prenez un moment pour prévisualiser certaines des fonctionnalités préférées de la communauté ou consultez notre documentation complète.
Documentation du sous-système Windows pour Linux (WSL)
Bonnes pratiques pour la configuration
Prise en main avec Linux

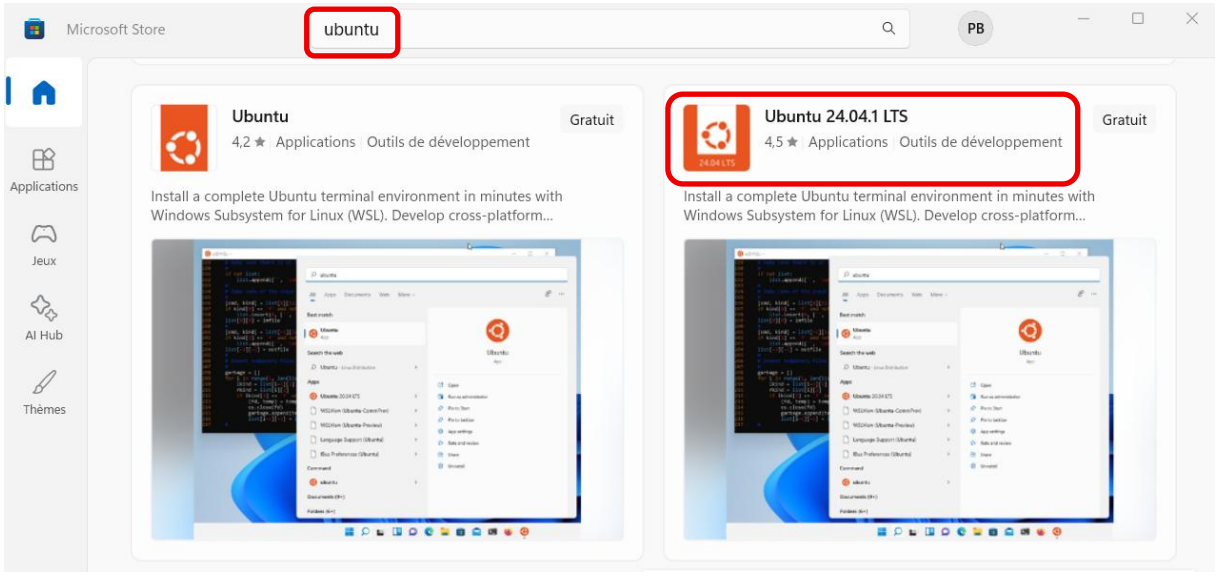

Bienvenue dans le sous-système Windows pour Linux
Général
Travailler sur plusieurs systèmes de f
Applications GUI
Accélération GPU
Intégration au réseau
Gestion des distributions
Intégration de Docker Desktop
Intégration VS Code
Paramètres
Intégration de VS Code
Vous pouvez utiliser WSL comme environnement de développement à temps plein directement depuis VS Code.
Comment installer
Après avoir installé VS Code, vous pouvez installer l'extension Remote WSL à partir du terminal Windows :
'code –install-extension ms-vscode-remote.remote-wsl'
Ouvrir un Projet WSL dans Visual Studio Code
Pour ouvrir un projet dans VS Code à partir de votre distribution WSL, ouvrez la ligne de commande de la distribution
Vous pouvez également accéder à davantage d'options VS Code Remote via la palette de commandes dans VS Code I
En savoir plus sur l'utilisation de WSL avec VS Code

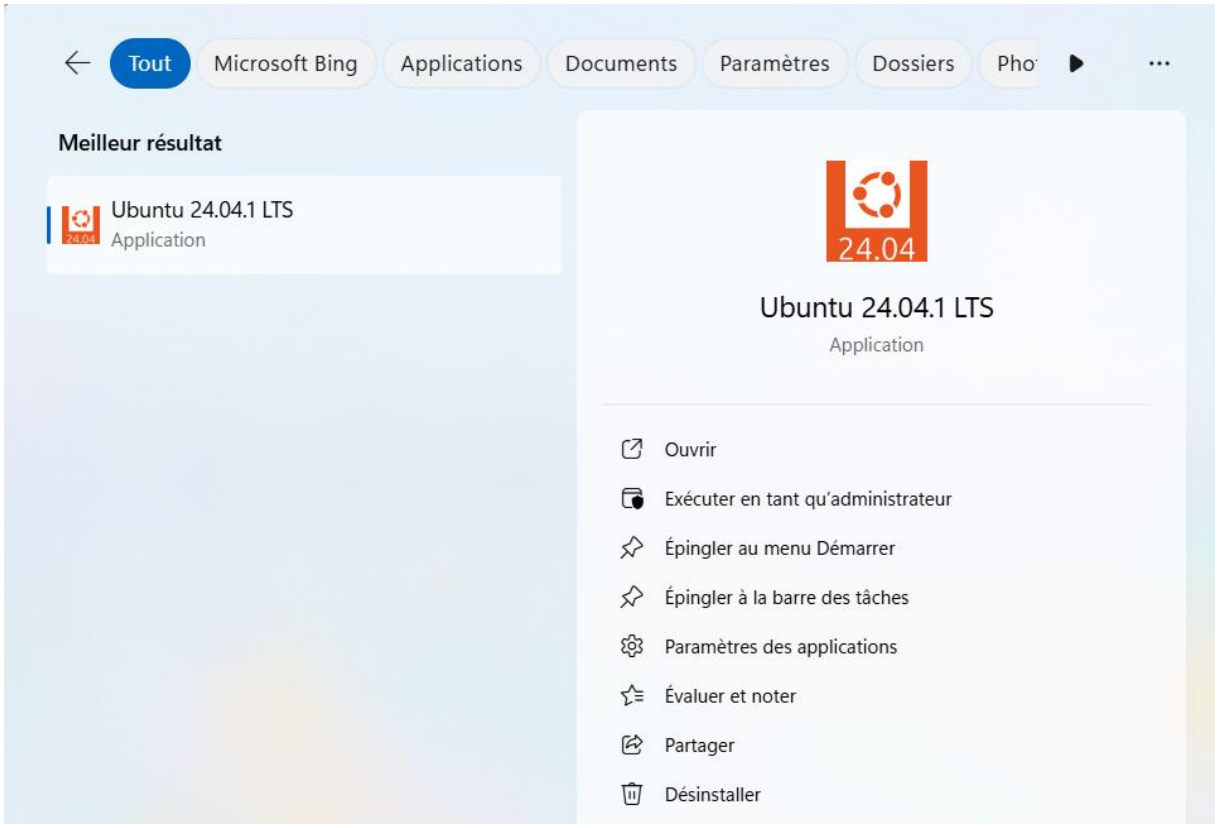

Fonctionnalités de Windows
Activer ou désactiver des fonctionnalités Windows
Pour activer une fonctionnalité, activez la case à cocher correspondante. Pour désactiver une fonctionnalité, désactivez la case à cocher correspondante. Une case à cocher pleine signifie qu'une partie de la fonctionnalité est activée.
Services pour NFS
Simple TCPIP services (i.e. echo, daytime etc)
SMB Direct
Sous-système Windows pour Linux
Support de partage de fichiers SMB 1.0/CIFS
Telnet Client
TFTP Client
Verrouillage de l'appareil
Windows Identity Foundation 3.5
Windows Projected File System
Windows TIFF IFilter
Work Folders Client
OK
Annuler

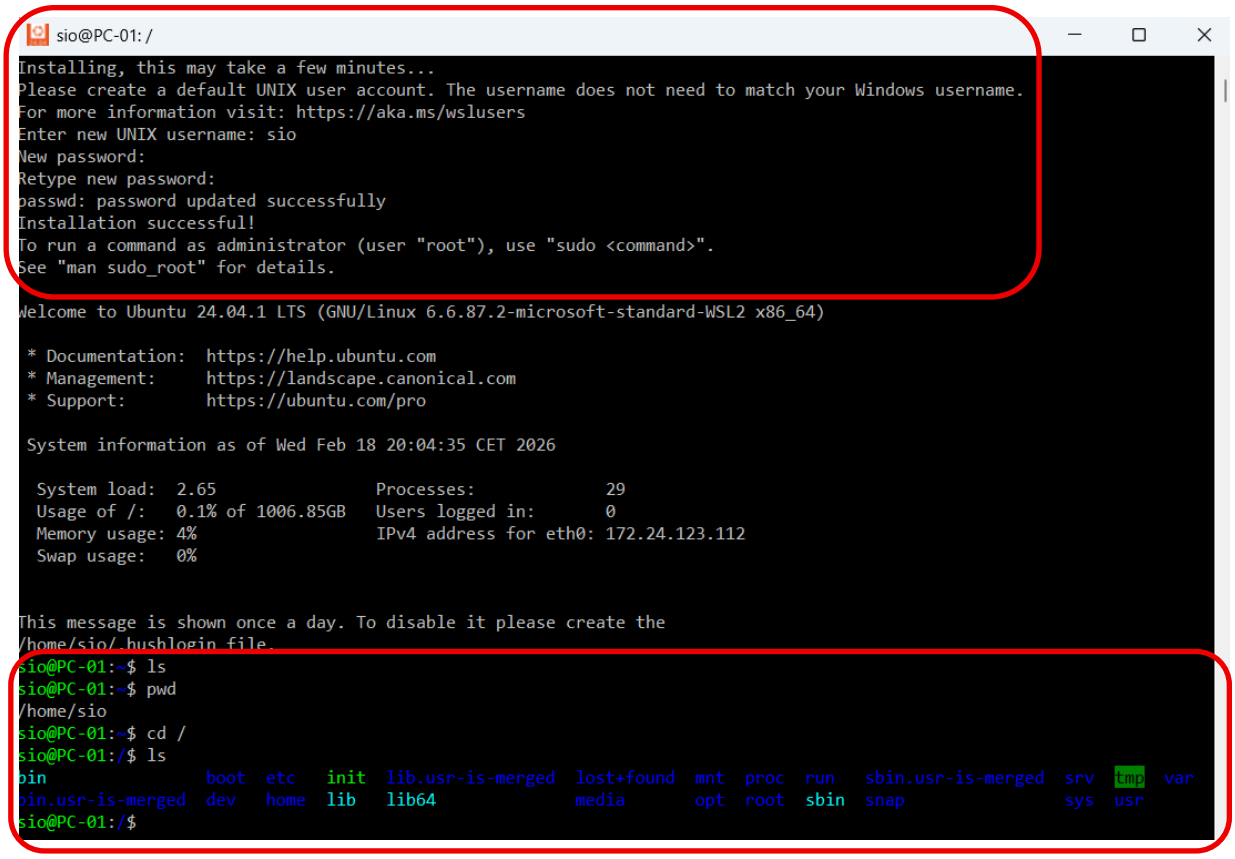

Tout
Microsoft Bing
Meilleur résultat
WSL
Système
Applications
WSL Settings
Commande
wsl


sio@PC-01: ~
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.6.87.2-microsoft-standard-WSL2 x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro
System information as of Wed Mar 4 10:53:42 CET 2026
System load: 0.2
Usage of /: 0.2% of 1006.85GB
Memory usage: 4%
Swap usage: 0%
Processes: 28
Users logged in: 0
IPv4 address for eth0: 172.24.123.112
* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s just raised the bar for easy, resilient and secure K8s cluster deployment.
https://ubuntu.com/engage/secure-kubernetes-at-the-edge
This message is shown once a day. To disable it please create the
/home/sio/.hushlogin file.
sio@PC-01:~$

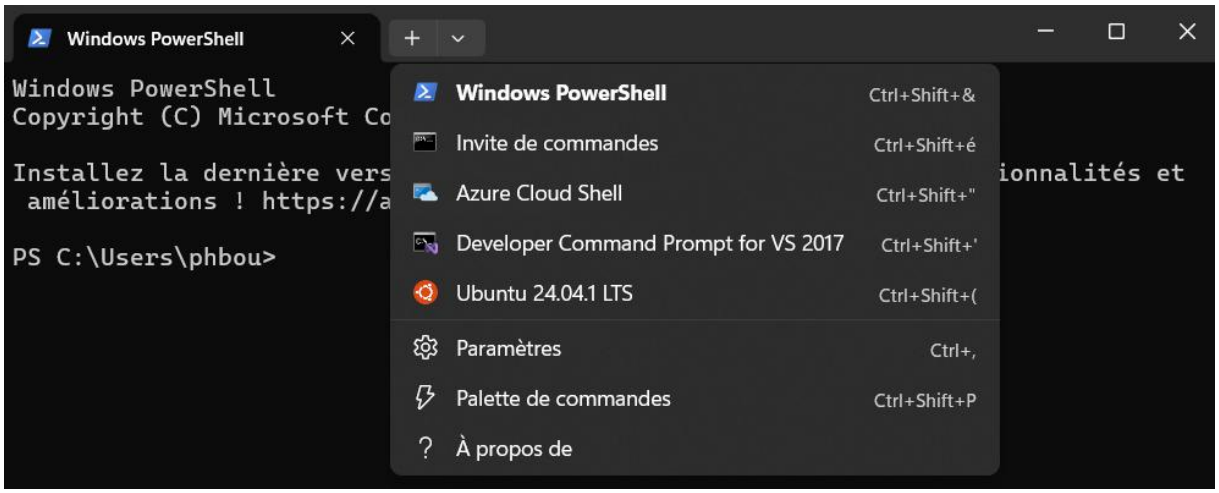
2. Installation Ubuntu.



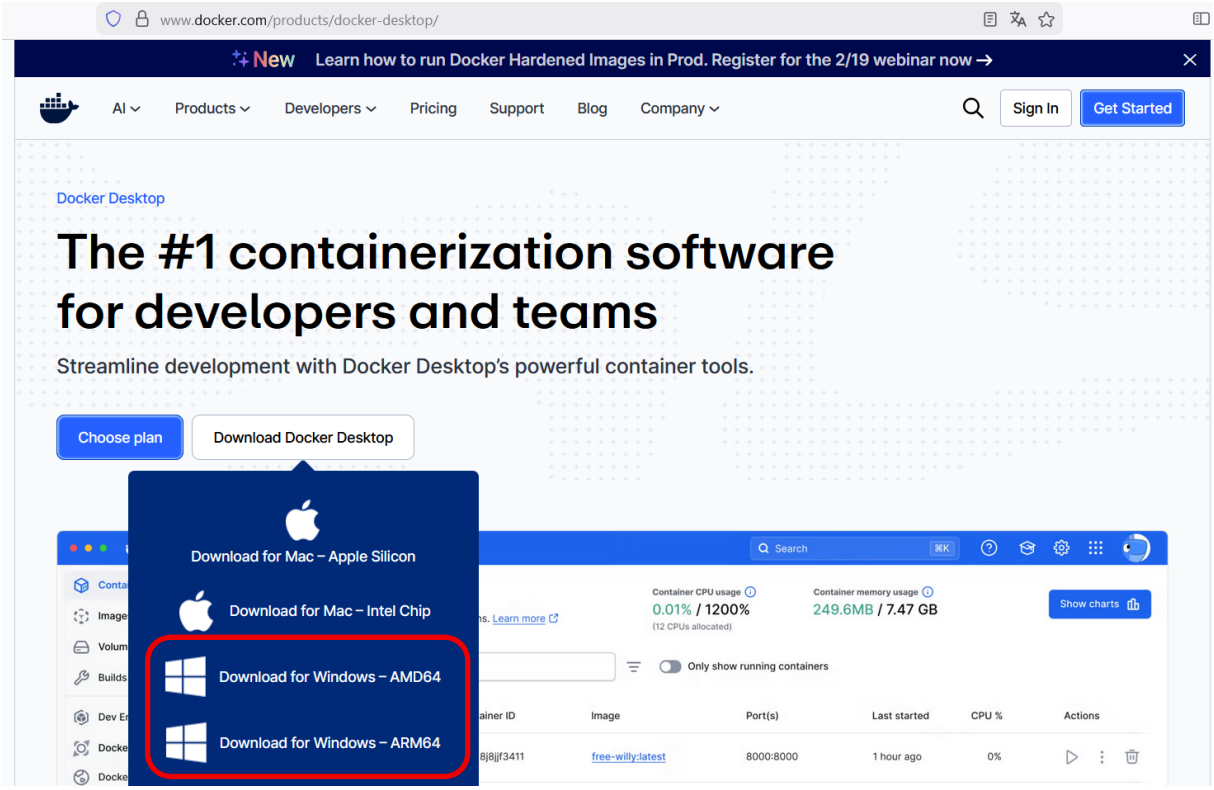


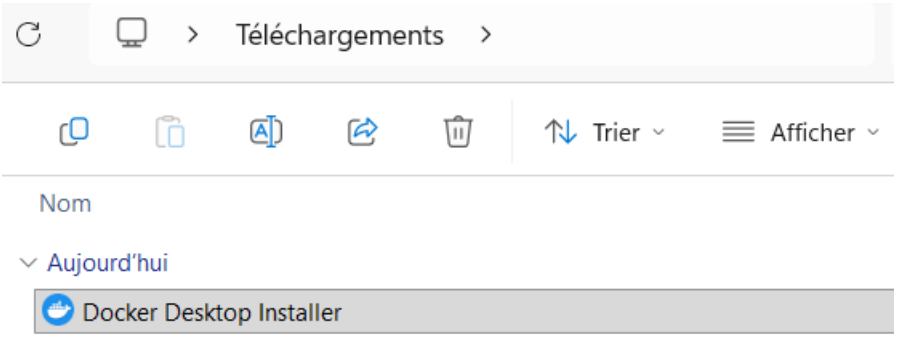


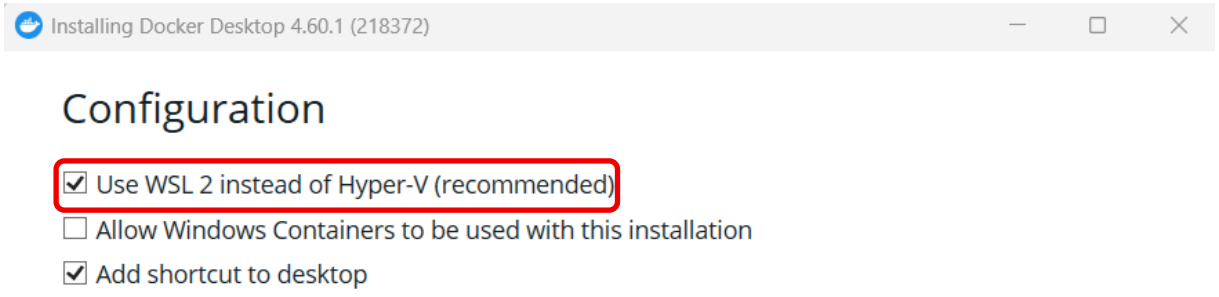
Terminal Windows :

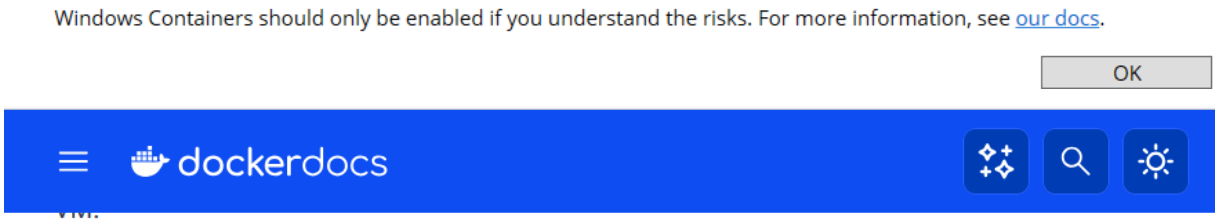


3. Installation Docker.





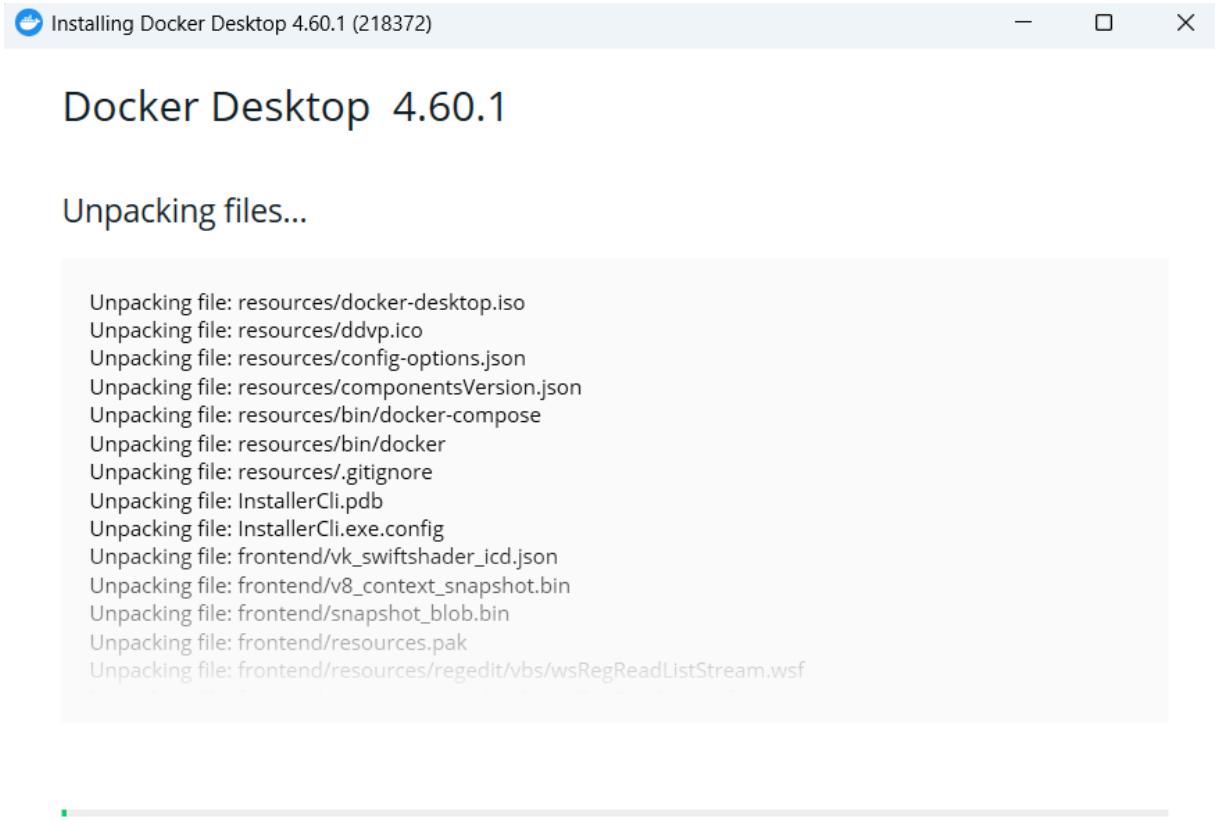


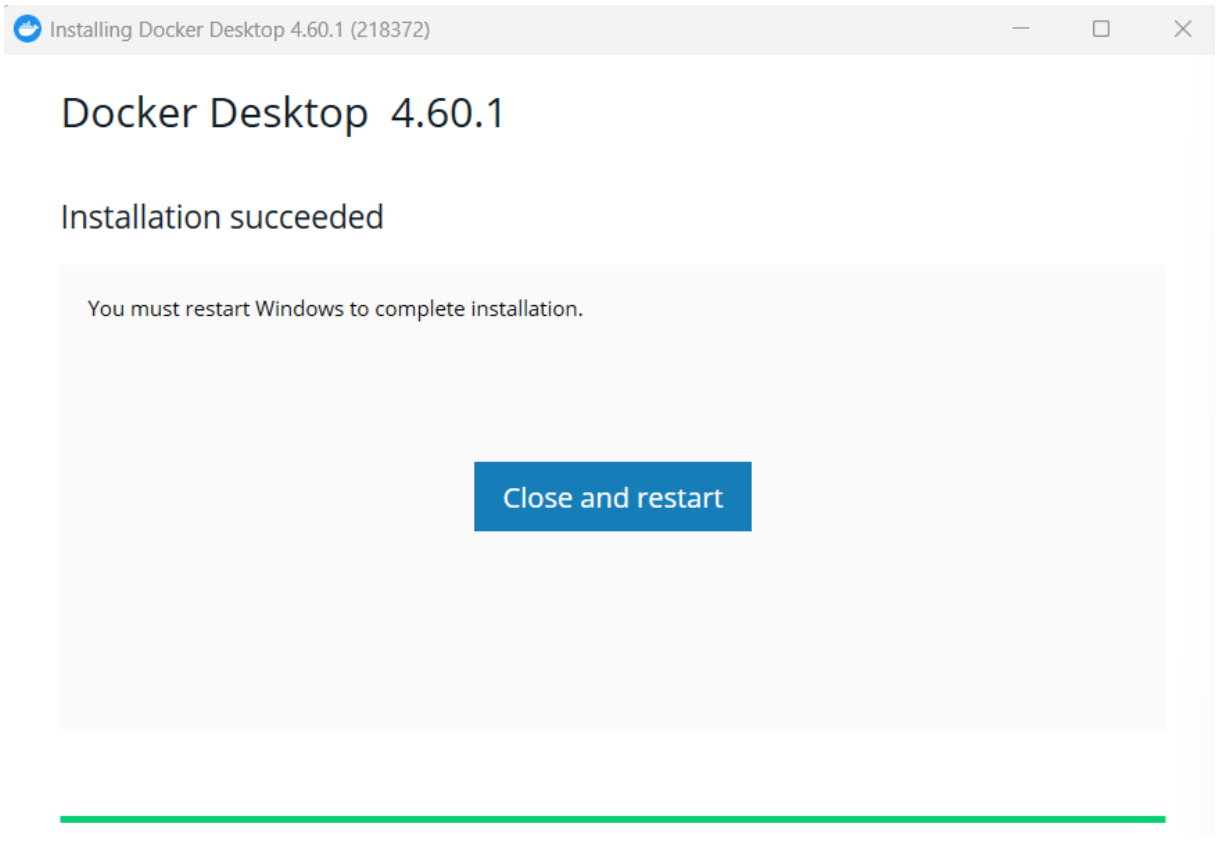


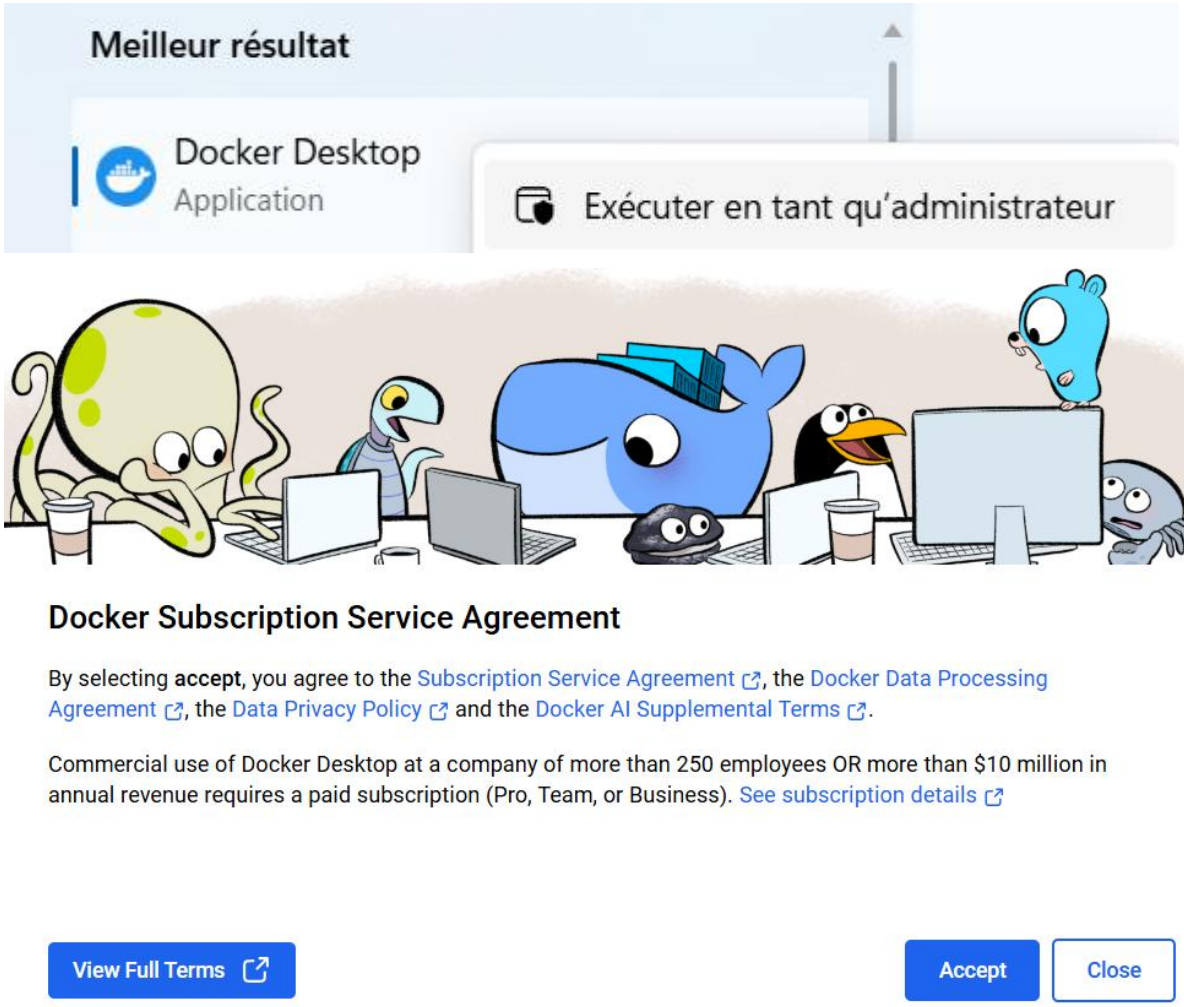
Windows containers

> **Warning**
>
> Enabling Windows containers has important security implications.

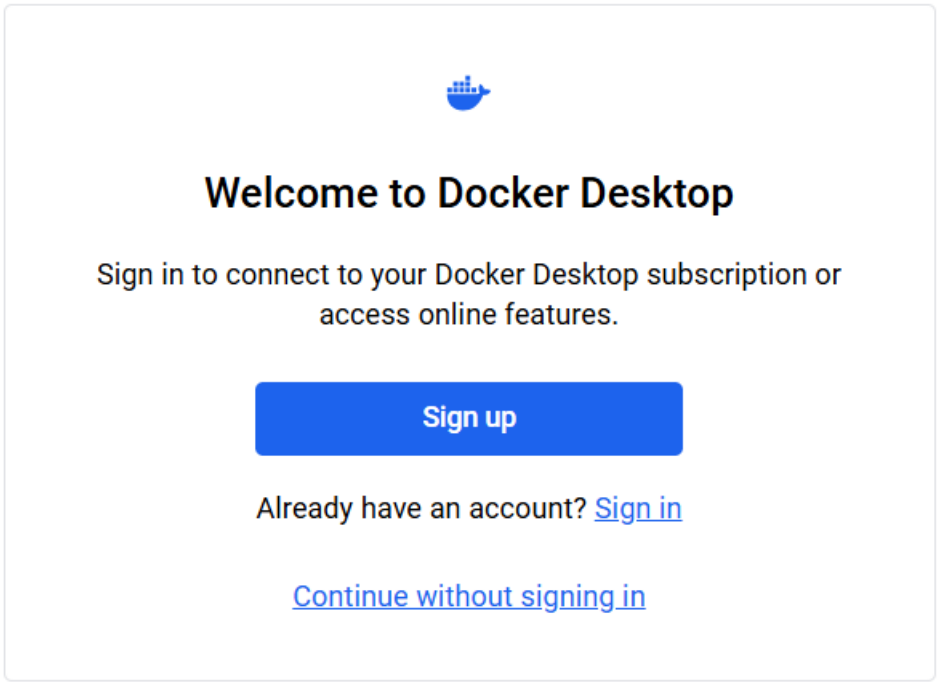
Unlike the Linux Docker Engine and containers which run in a VM, Windows containers are implemented using operating system features, and run directly on the Windows host. If you enable Windows containers during installation, the `ContainerAdministrator` user used for administration inside the container is a local administrator on the host machine. Enabling Windows containers during installation makes it so that members of the `docker-users` group are able to elevate to administrators on the host. For organizations who don't want their developers to run Windows containers, a `--no-windows-containers` installer flag is available to disable their use.

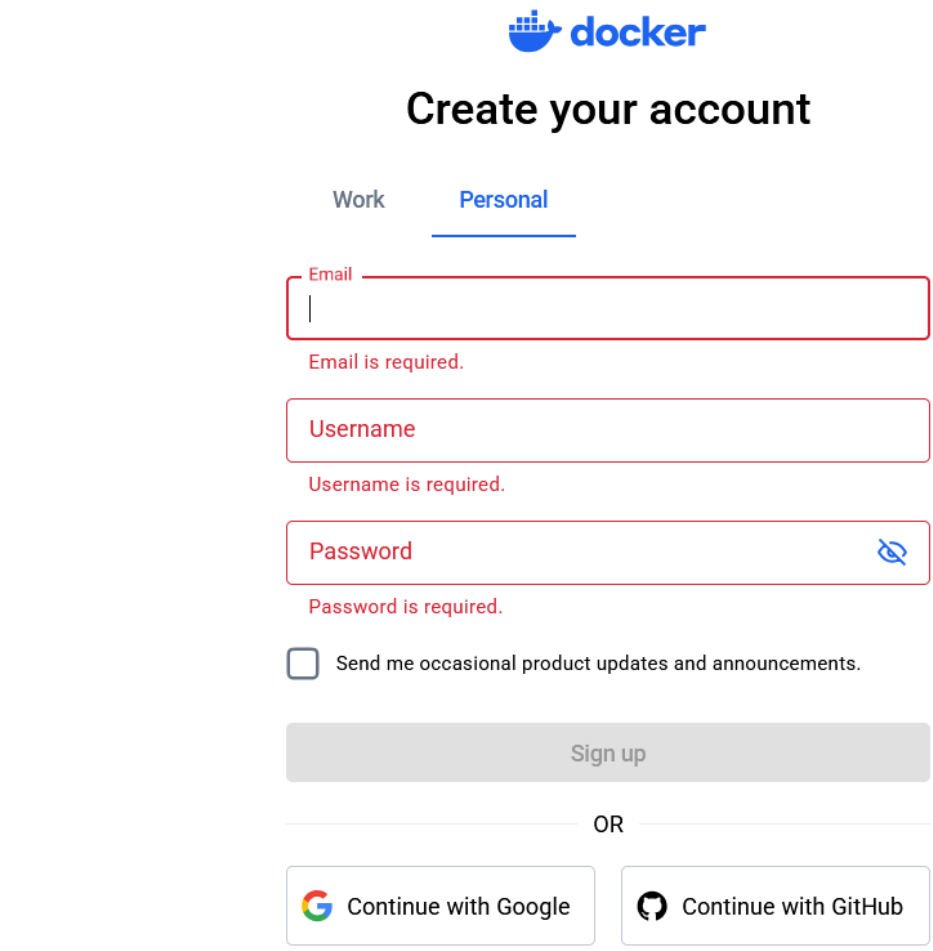

Installing Docker Desktop 4.60.1 (218372)
Docker Desktop 4.60.1
Unpacking files...
Unpacking file: resources/docker-desktop.iso
Unpacking file: resources/ddvp.ico
Unpacking file: resources/config-options.json
Unpacking file: resources/componentsVersion.json
Unpacking file: resources/bin/docker-compose
Unpacking file: resources/bin/docker
Unpacking file: resources/.gitignore
Unpacking file: InstallerCli.pdb
Unpacking file: InstallerCli.exe.config
Unpacking file: frontend/vk_swiftshader_icd.json
Unpacking file: frontend/v8_context_snapshot.bin
Unpacking file: frontend/snapshot_blob.bin
Unpacking file: frontend/resources.pak

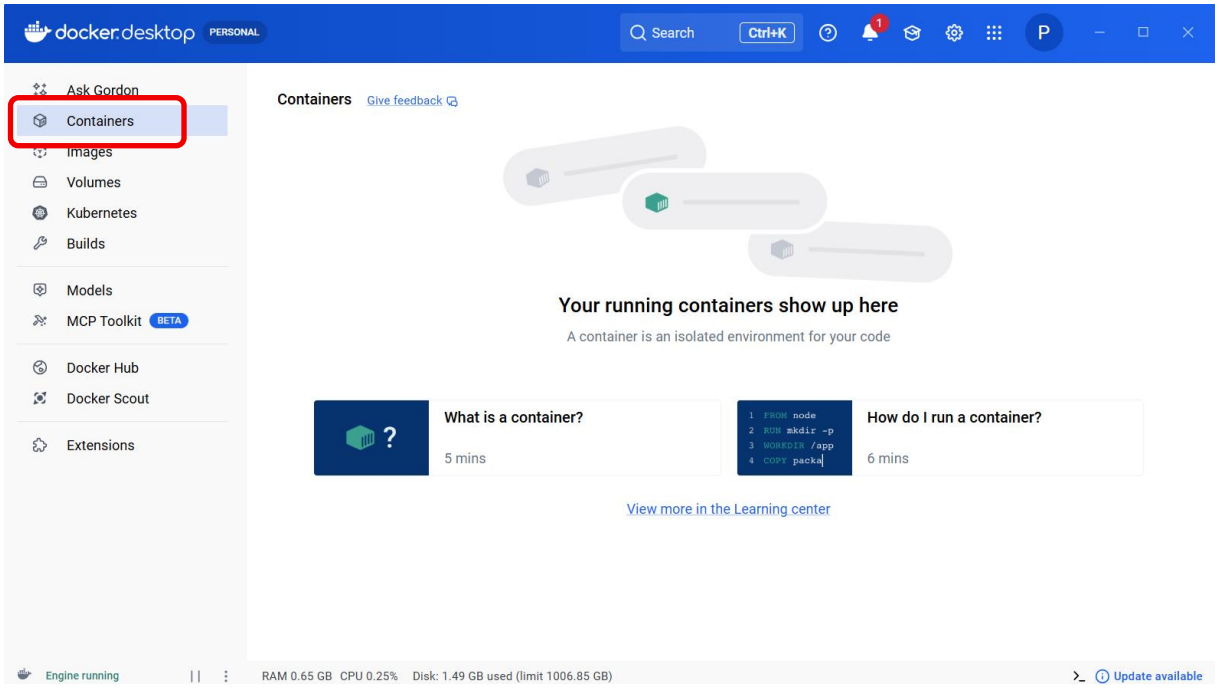

Installing Docker Desktop 4.60.1 (218372)
Docker Desktop 4.60.1
Installation succeeded
You must restart Windows to complete installation.
Close and restart

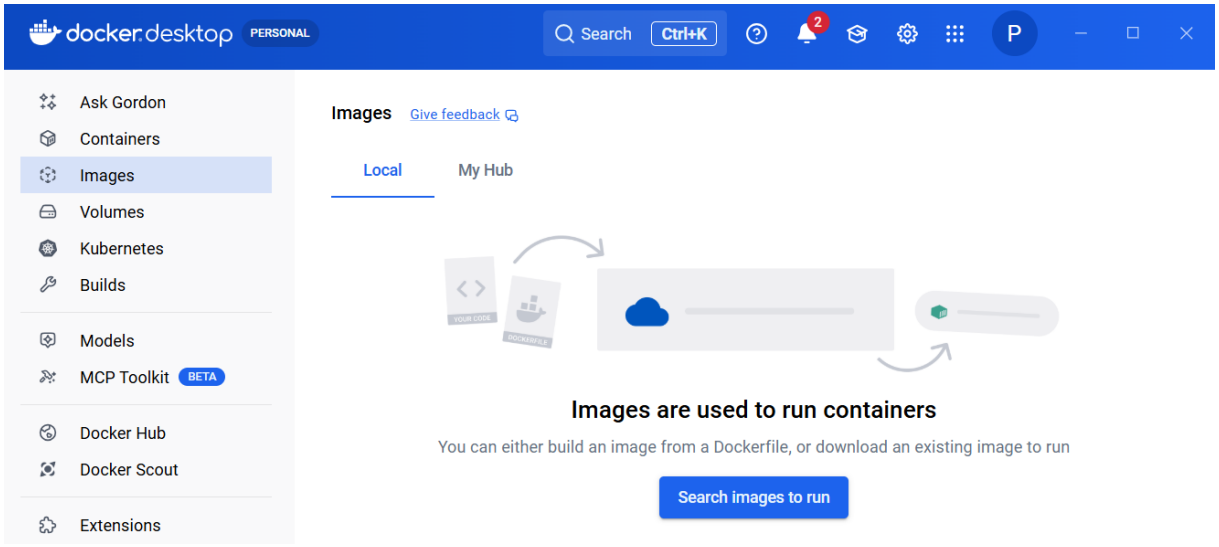


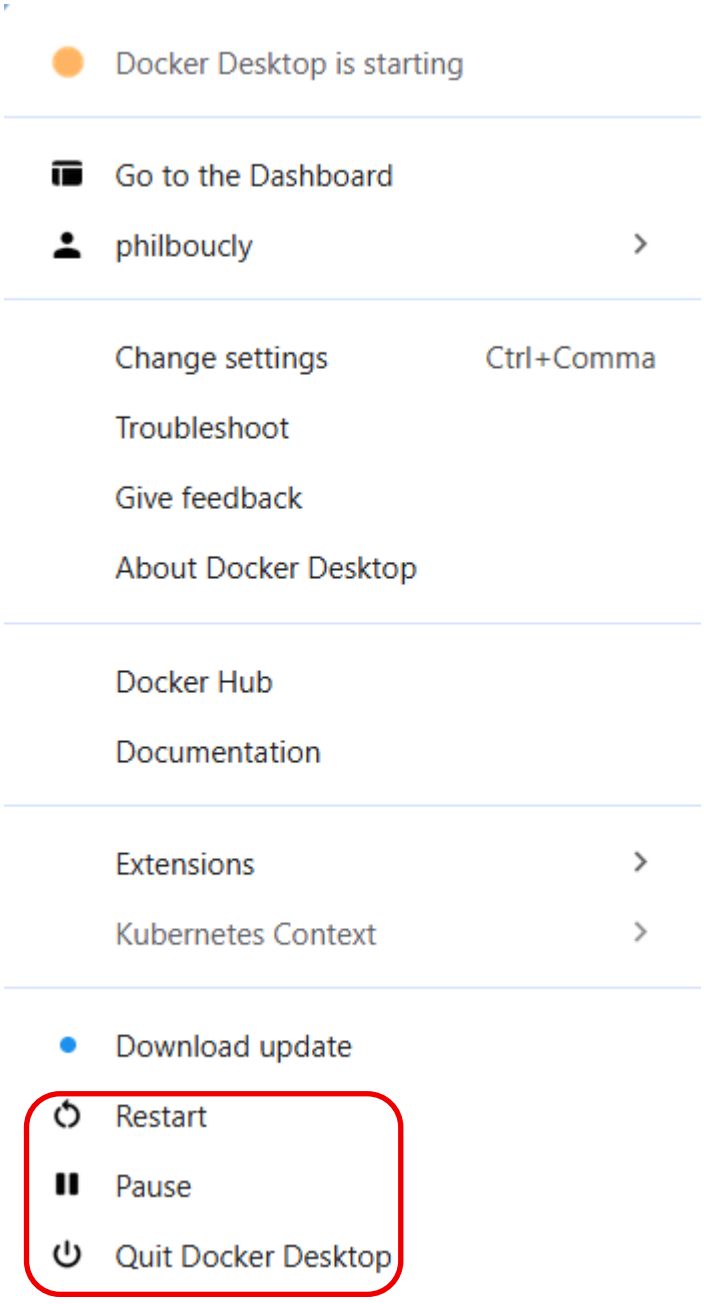
Créer un compte et se connecter :

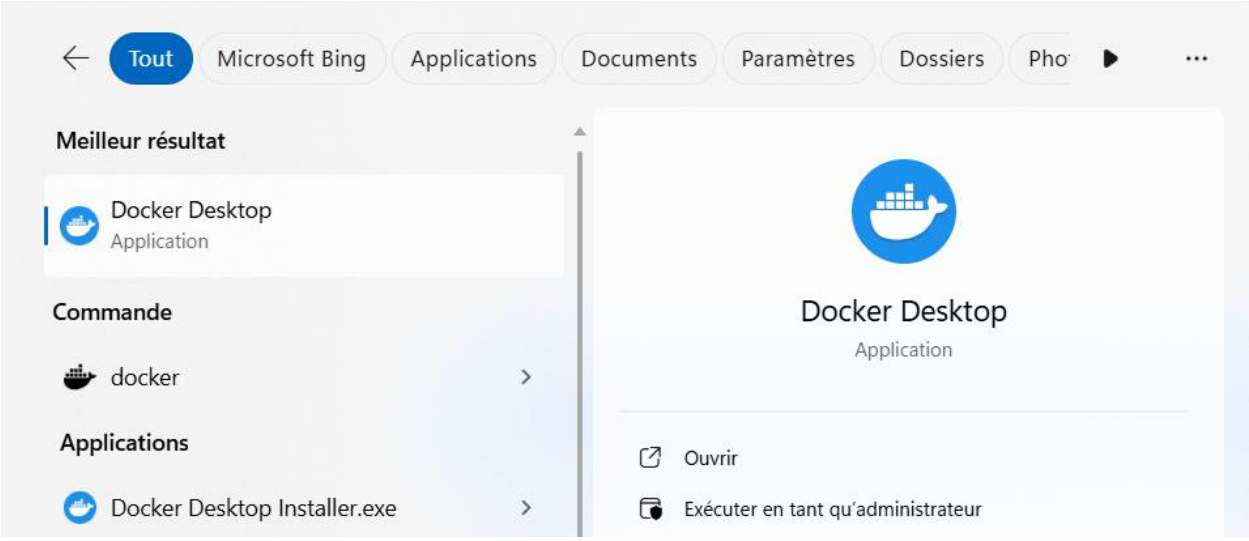



docker
Create your account
Work
Personal
Email
Email is required.
Username
Username is required.
Password
Password is required.
Send me occasional product updates and announcements.
Sign up
OR
Continue with Google
Continue with GitHub
Already have an account? Sign in

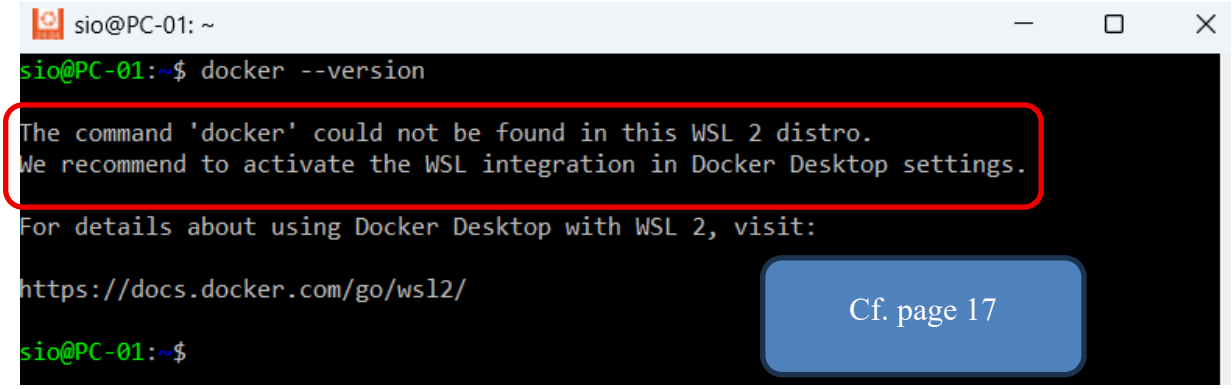

docker desktop
PERSONAL
Search
Ctrl+K
Ask Gordon
Containers
Images
Volumes
Kubernetes
Builds
Models
MCP Toolkit
BETA
Docker Hub
Docker Scout
Extensions
Containers
Give feedback
Your running containers show up here
A container is an isolated environment for your code
What is a container?
5 mins
How do I run a container?
6 mins
View more in the Learning center
Engine running
RAM 0.65 GB CPU 0.25%
Disk: 1.49 GB used (limit 1006.85 GB)
Update available


docker desktop
PERSONAL
Search
Ctrl+K
Ask Gordon
Containers
Images
Volumes
Kubernetes
Builds
Models
MCP Toolkit
BETA
Docker Hub
Docker Scout
Extensions
Images
Give feedback
Local
My Hub
Images are used to run containers
You can either build an image from a Dockerfile, or download an existing image to run
Search images to run

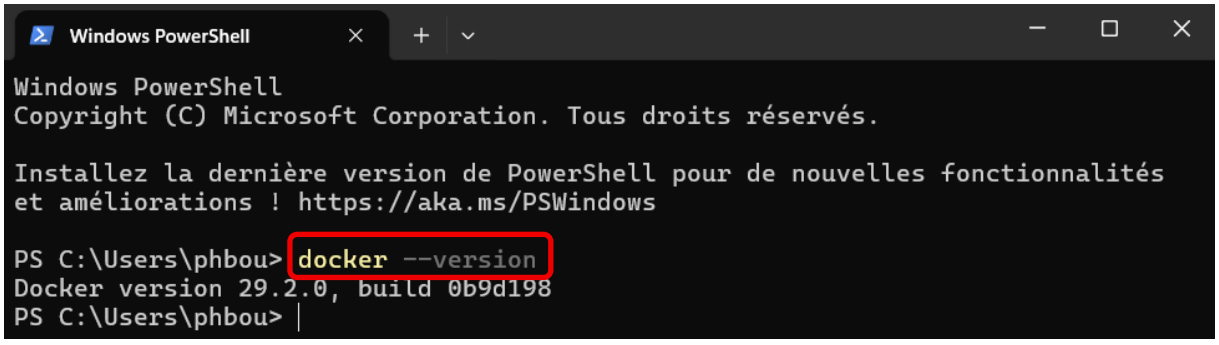

Docker Desktop is starting
Go to the Dashboard
philboucly
Change settings
Ctrl+Comma
Troubleshoot
Give feedback
About Docker Desktop
Docker Hub
Documentation
Extensions
Kubernetes Context
Download update
Restart
Pause
Quit Docker Desktop

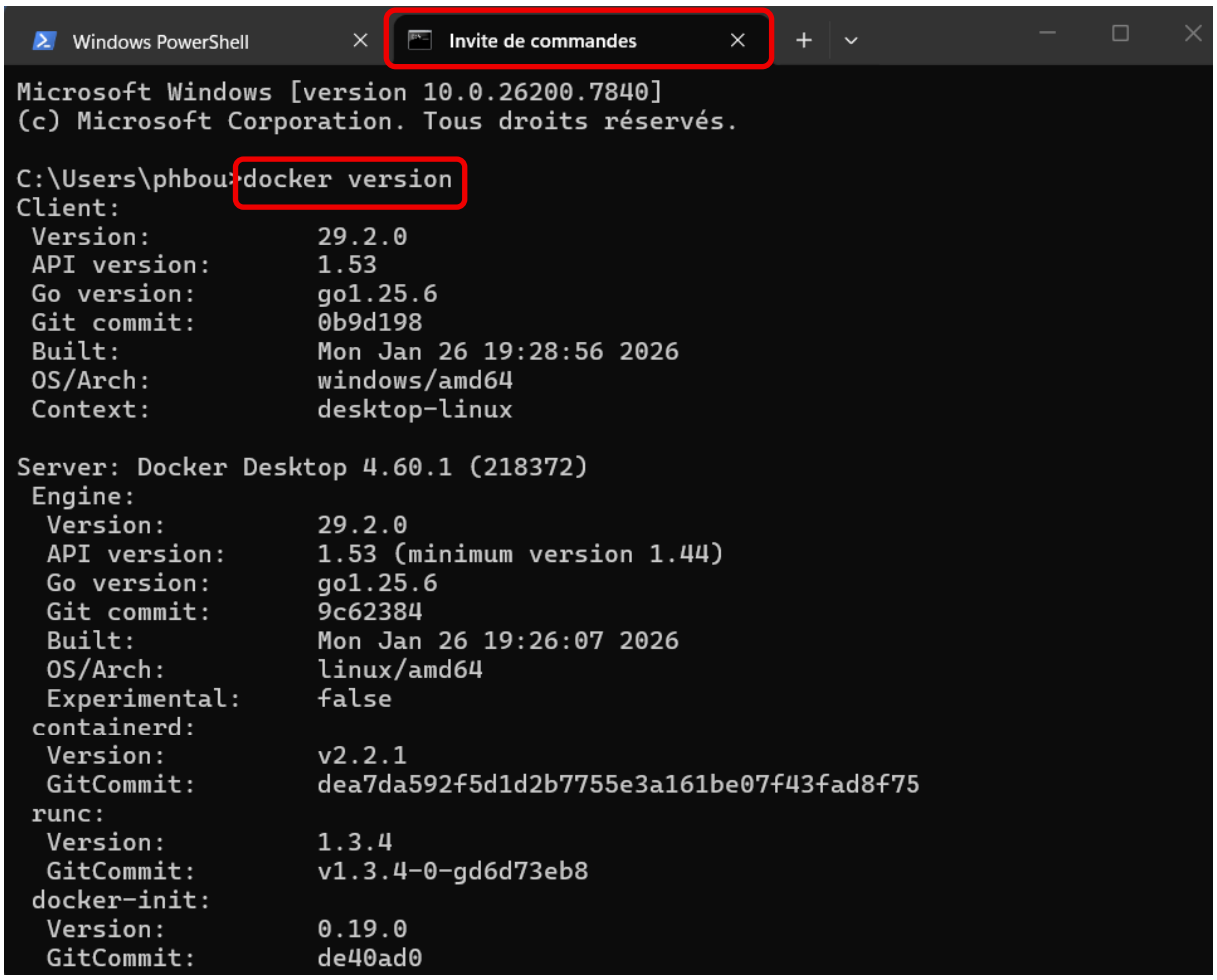


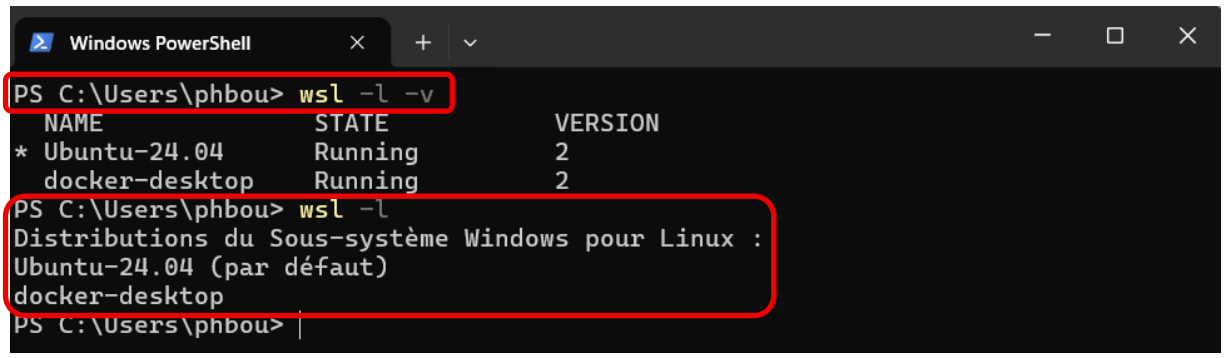
4. Premier test.

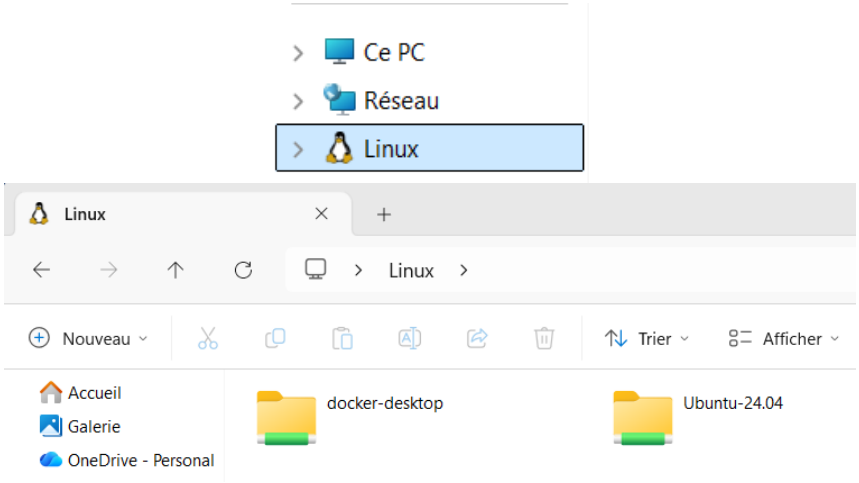


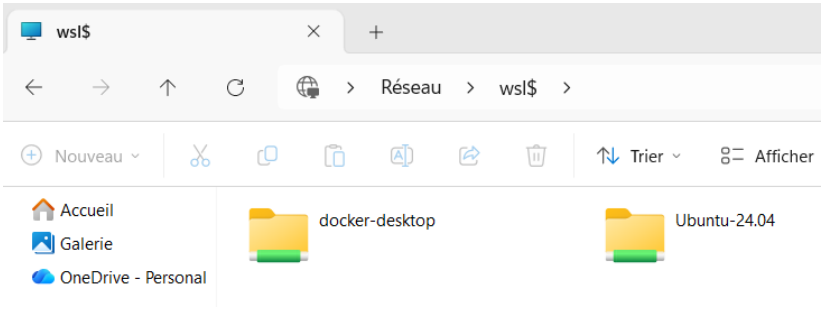
Terminal Windows :

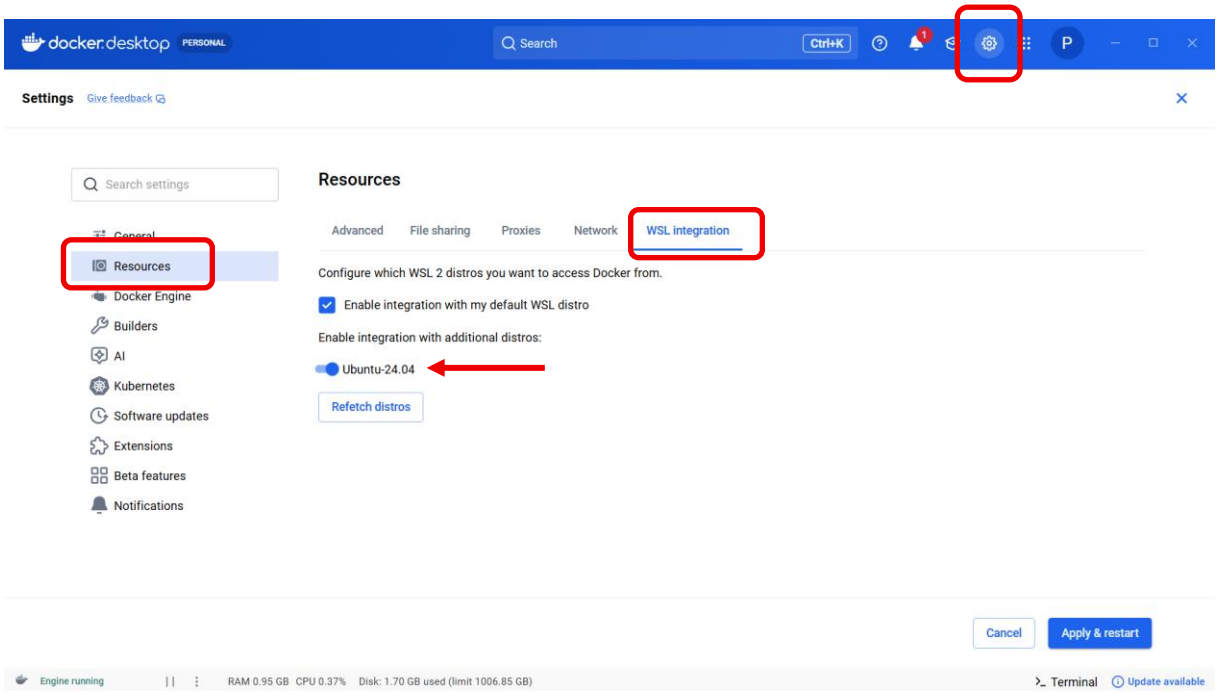



Windows PowerShell
Invite de commandes
Microsoft Windows [version 10.0.26200.7840]
(c) Microsoft Corporation. Tous droits réservés.
C:\Users\phbou>docker version
Client:
Version: 29.2.0
API version: 1.53
Go version: go1.25.6
Git commit: 0b9d198
Built: Mon Jan 26 19:28:56 2026
OS/Arch: windows/amd64
Context: desktop-linux
Server: Docker Desktop 4.60.1 (218372)
Engine:
Version: 29.2.0
API version: 1.53 (minimum version 1.44)
Go version: go1.25.6
Git commit: 9c62384
Built: Mon Jan 26 19:26:07 2026
OS/Arch: linux/amd64
Experimental: false
containerd:
Version: v2.2.1
GitCommit: dea7da592f5d1d2b7755e3a161be07f43fad8f75
runc:
Version: 1.3.4
GitCommit: v1.3.4-0-gd6d73eb8
docker-init:
Version: 0.19.0
GitCommit: de40ad0


Windows PowerShell
PS C:\Users\phbou> wsl -l -v
NAME STATE VERSION
* Ubuntu-24.04 Running 2
docker-desktop Running 2
PS C:\Users\phbou> wsl -l
Distributions du Sous-système Windows pour Linux :
Ubuntu-24.04 (par défaut)
docker-desktop
PS C:\Users\phbou>


Ce PC
Réseau
Linux
Linux
Linux
Nouveau
Trier
Afficher
Accueil
Galerie
OneDrive - Personal
docker-desktop
Ubuntu-24.04


wsl$
Réseau
wsl$
Nouveau
Trier
Afficher
Accueil
Galerie
OneDrive - Personal
docker-desktop
Ubuntu-24.04


docker desktop
PERSONAL
Search
Ctrl+K
Settings
Give feedback
Search settings
General
Resources
Docker Engine
Builders
AI
Kubernetes
Software updates
Extensions
Beta features
Notifications
Resources
Advanced
File sharing
Proxies
Network
WSL integration
Configure which WSL 2 distros you want to access Docker from.
Enable integration with my default WSL distro
Enable integration with additional distros:
Ubuntu-24.04
Refetch distros
Cancel
Apply & restart
Engine running
RAM 0.95 GB CPU 0.37% Disk: 1.70 GB used (limit 1006.85 GB)
Terminal
Update available

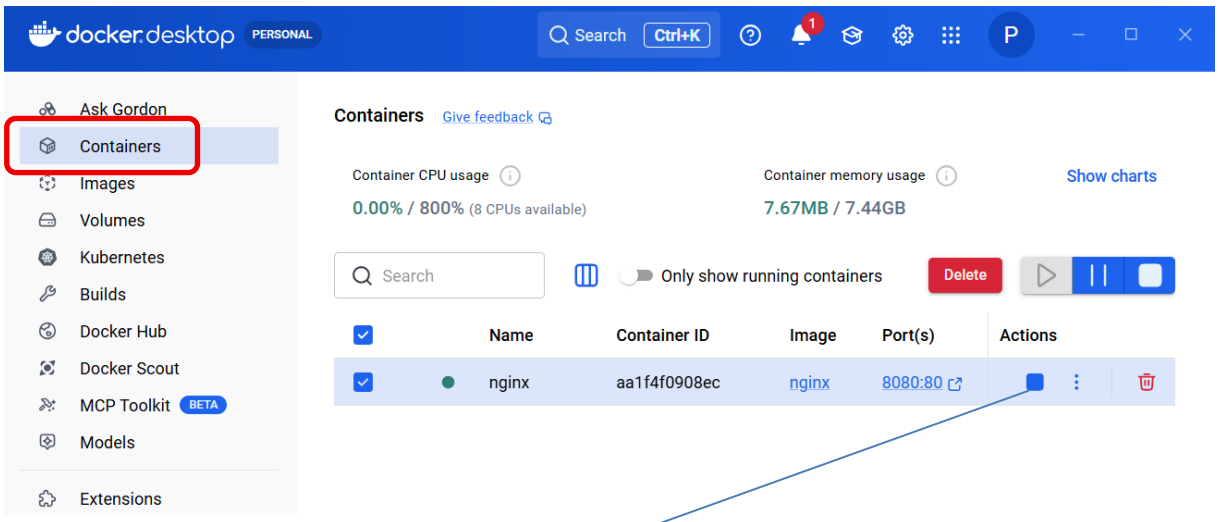
5. Deuxième test.

```
sio@PC-01: ~
sio@PC-01:~$ docker run --name nginx -p 8080:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
bae5a1799a80: Pull complete
46bf3a120c8e: Pull complete
47cd406a84ef: Pull complete
7b6cb8ccac7b: Pull complete
4f4efe02d542: Pull complete
0c8d55a45c0d: Pull complete
f73400a233fd: Pull complete
2e02dba24409: Download complete
a5d78d617315: Download complete
Digest: sha256:341bf0f3ce6c5277d6002cf6e1fb0319fa4252add24ab6a0e262e0056d313208
Status: Downloaded newer image for nginx:latest
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2026/02/23 09:46:21 [notice] 1#1: using the "epoll" event method
2026/02/23 09:46:21 [notice] 1#1: nginx/1.29.5
2026/02/23 09:46:21 [notice] 1#1: built by gcc 14.2.0 (Debian 14.2.0-19)
2026/02/23 09:46:21 [notice] 1#1: OS: Linux 6.6.87.2-microsoft-standard-WSL2
2026/02/23 09:46:21 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2026/02/23 09:46:21 [notice] 1#1: start worker processes
2026/02/23 09:46:21 [notice] 1#1: start worker process 29
2026/02/23 09:46:21 [notice] 1#1: start worker process 30
2026/02/23 09:46:21 [notice] 1#1: start worker process 31
2026/02/23 09:46:21 [notice] 1#1: start worker process 32
2026/02/23 09:46:21 [notice] 1#1: start worker process 33
2026/02/23 09:46:21 [notice] 1#1: start worker process 34
2026/02/23 09:46:21 [notice] 1#1: start worker process 35
2026/02/23 09:46:21 [notice] 1#1: start worker process 36
```

Nom du conteneur

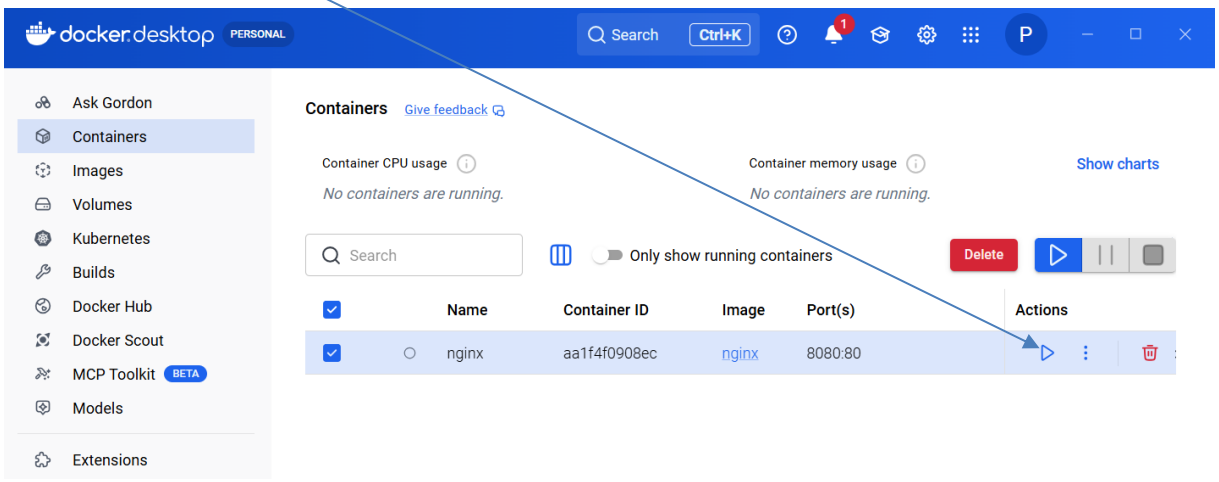
Nom de l'image

➔ Docker télécharge dans un premier temps l'image nginx. Il crée ensuite une instance de cette image qui est le conteneur.

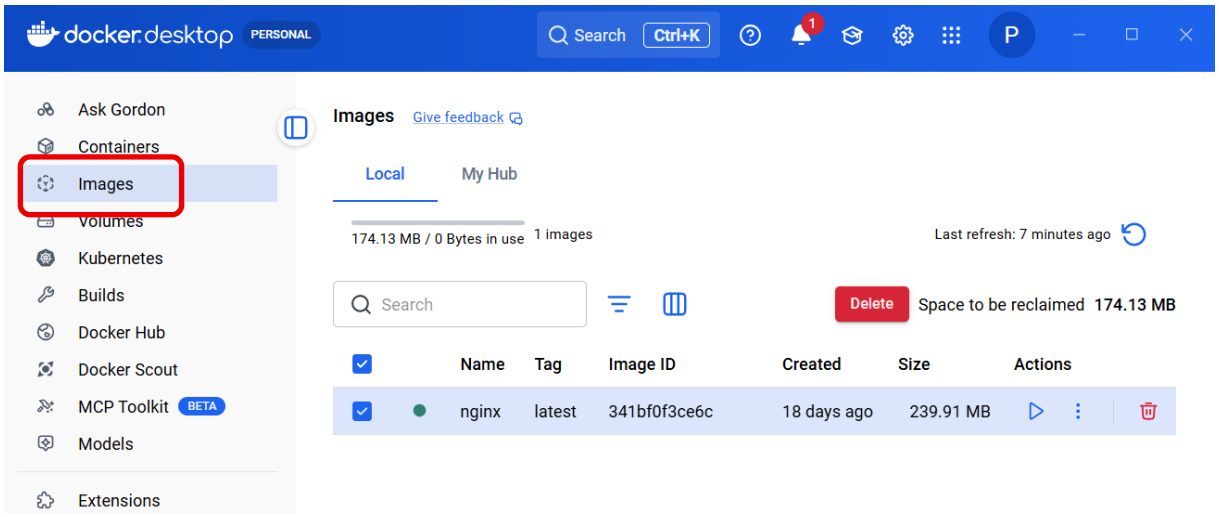


Si on relance la commande, l'image ne sera pas téléchargée de nouveau car elle existe déjà en local. Par contre, un nouveau conteneur sera créé.

- Action container : stop

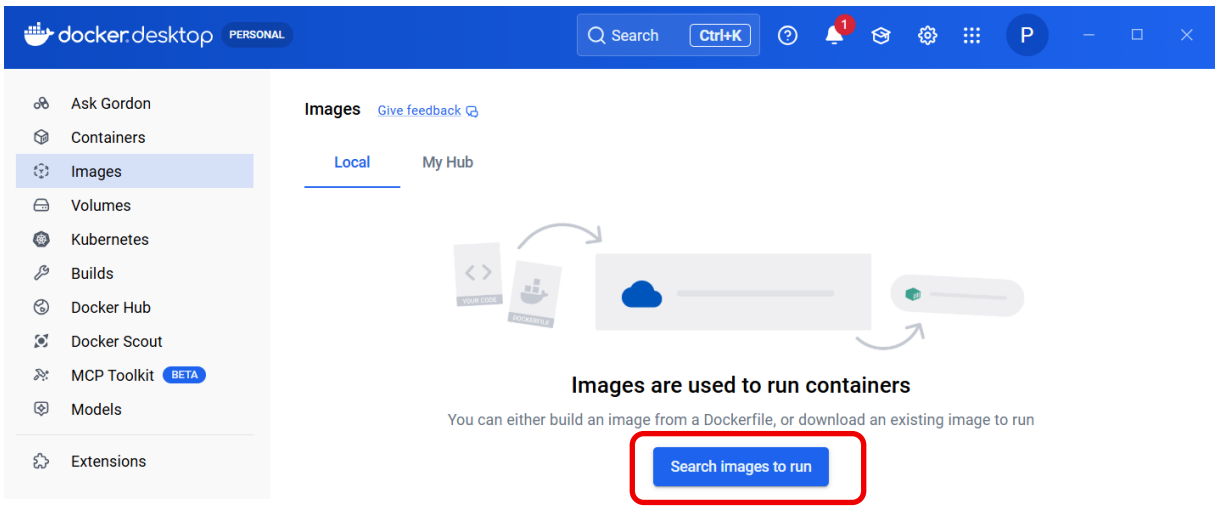


- Delete container
- Puis Delete image :

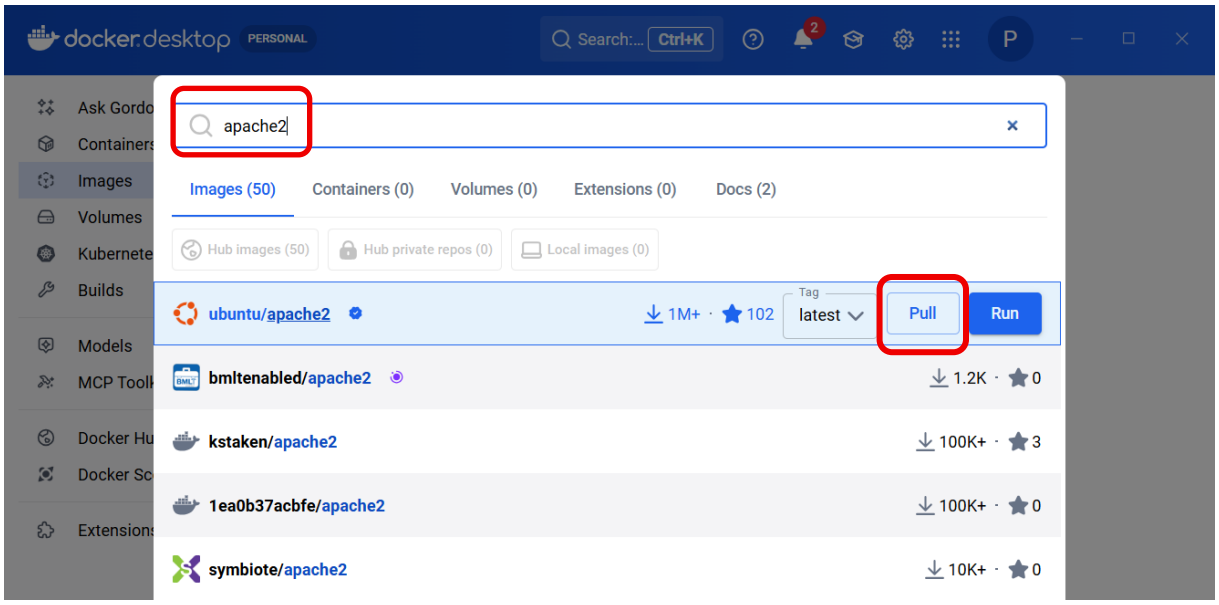


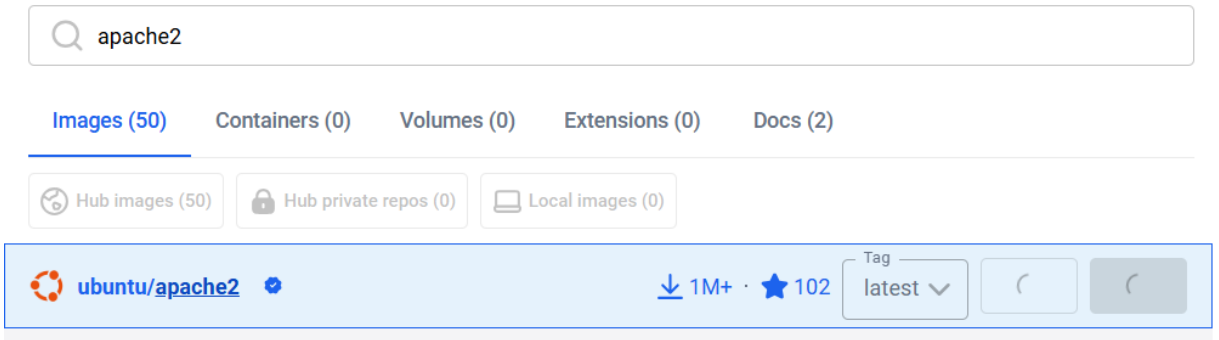
6. Troisième test.

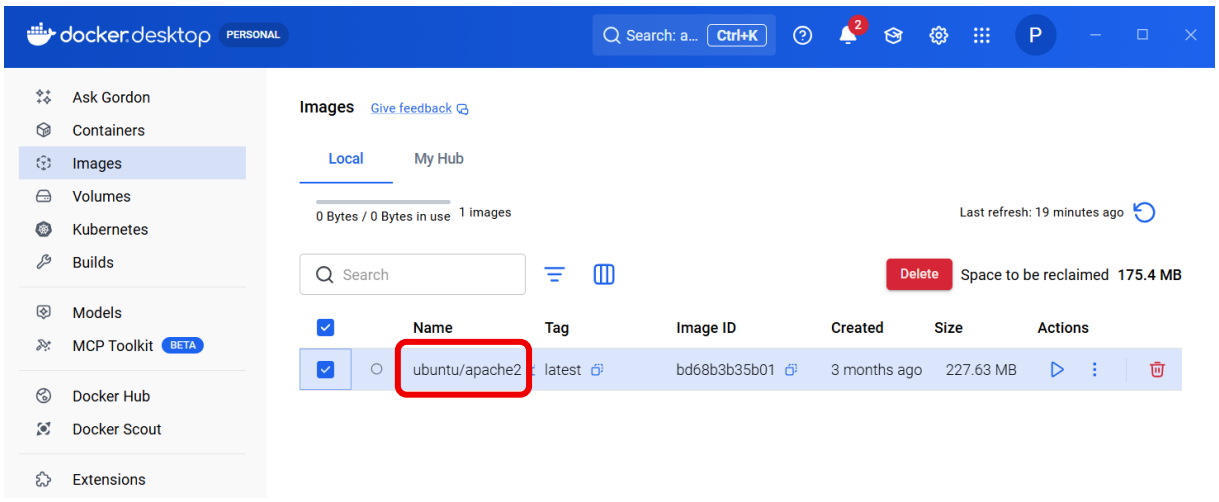
- Search images to run


docker desktop
PERSONAL
Search
Ctrl+K
Ask Gordon
Containers
Images
Volumes
Kubernetes
Builds
Docker Hub
Docker Scout
MCP Toolkit
BETA
Models
Extensions
Images
Give feedback
Local
My Hub
Images are used to run containers
You can either build an image from a Dockerfile, or download an existing image to run
Search images to run

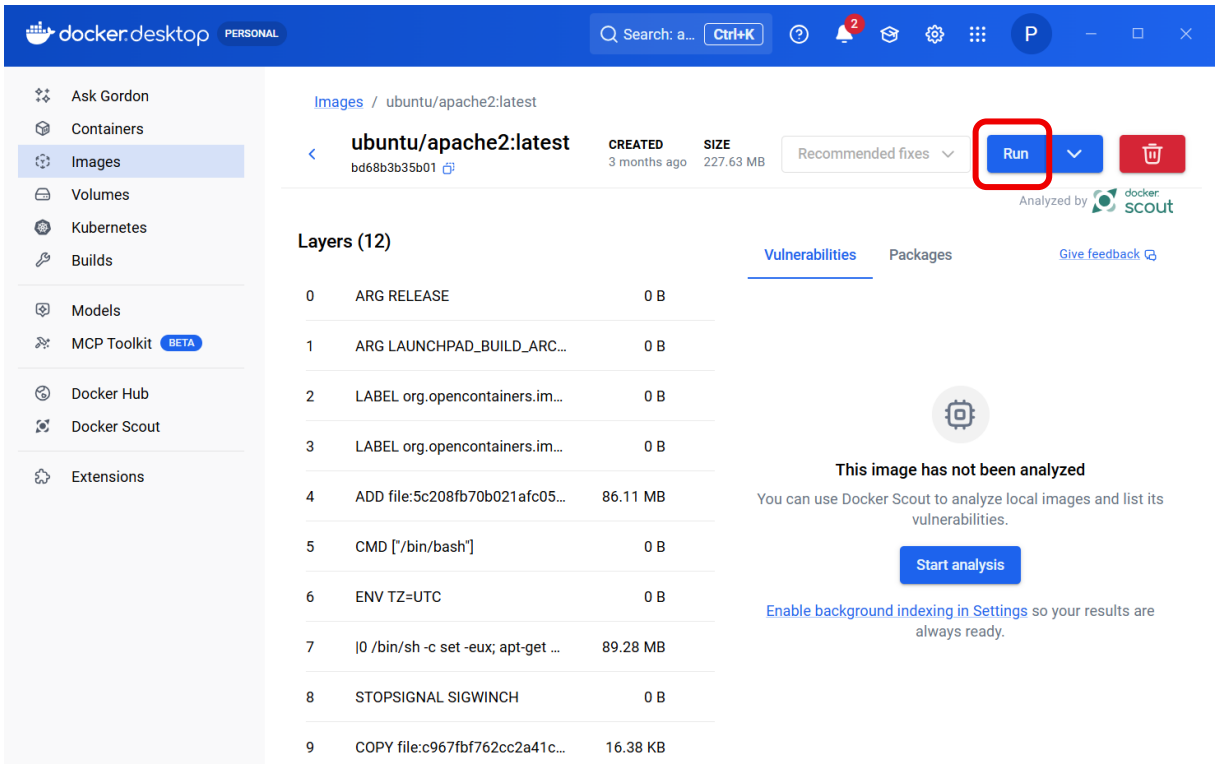
- Pull :

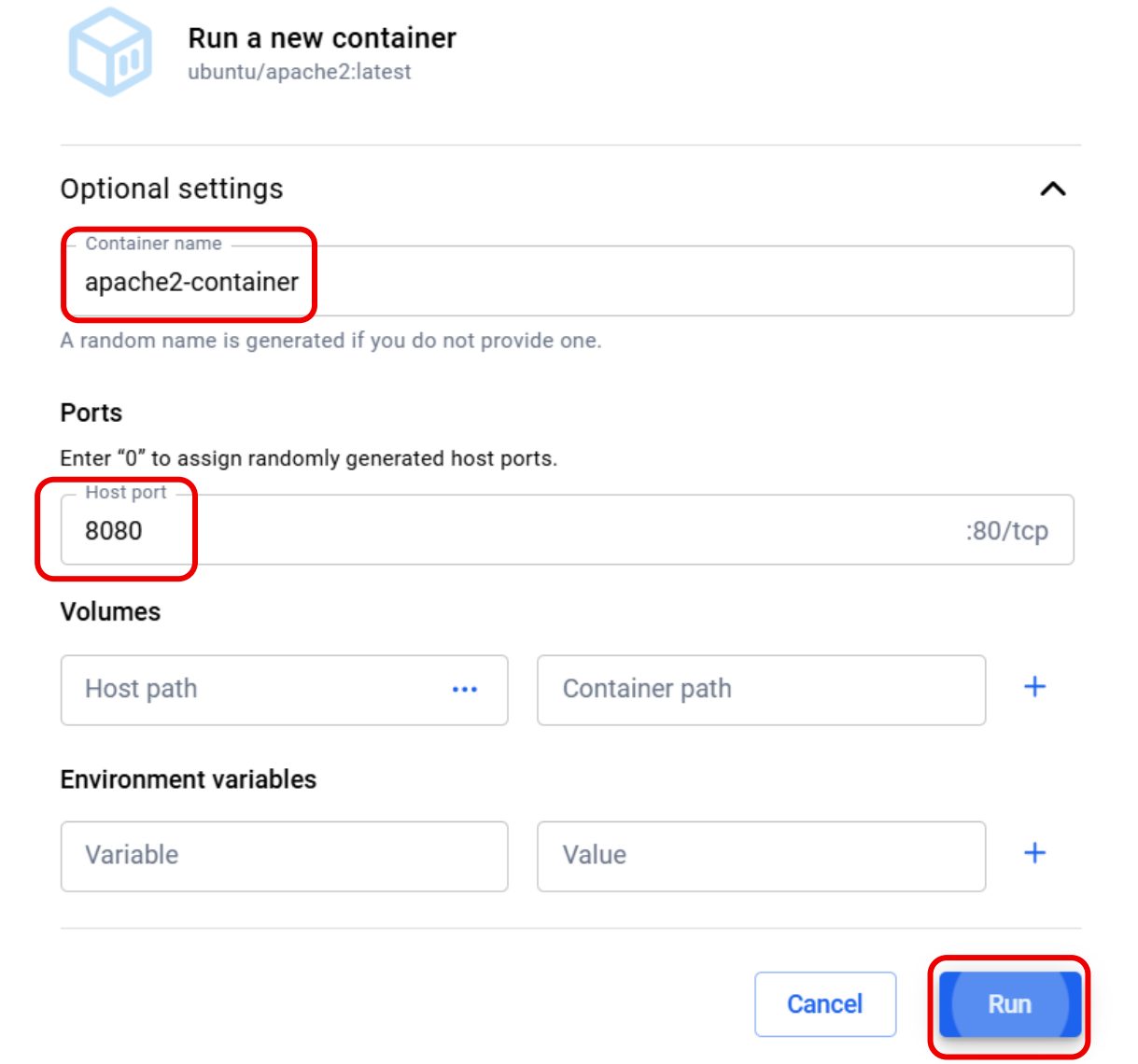

docker desktop
PERSONAL
Search:...
Ctrl+K
apache2
Images (50)
Containers (0)
Volumes (0)
Extensions (0)
Docs (2)
Hub images (50)
Hub private repos (0)
Local images (0)
ubuntu/apache2
1M+
102
Tag
latest
Pull
Run
bmltenabled/apache2
1.2K
0
kstaken/apache2
100K+
3
1ea0b37acbfe/apache2
100K+
0
symbiote/apache2
10K+
0

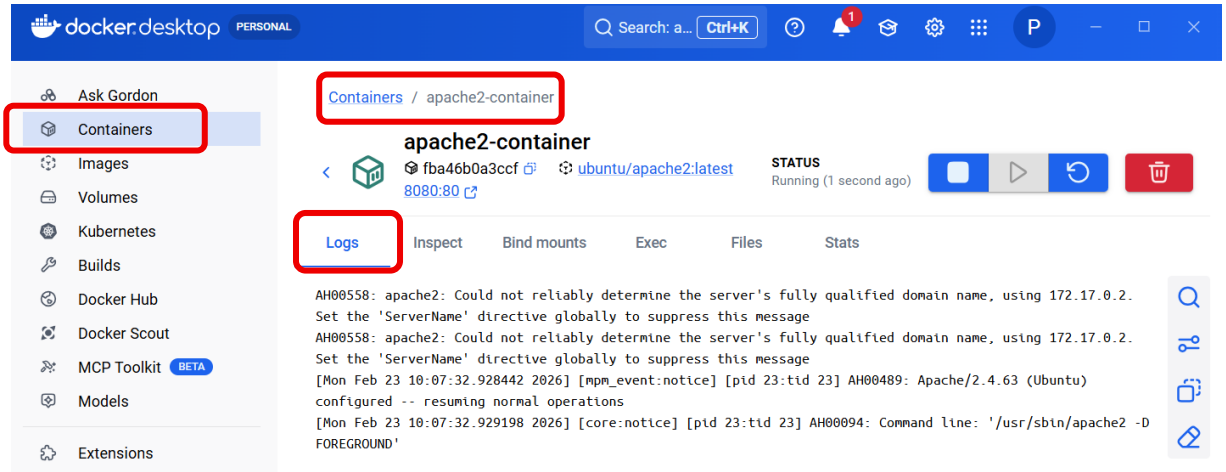

apache2
Images (50)
Containers (0)
Volumes (0)
Extensions (0)
Docs (2)
Hub images (50)
Hub private repos (0)
Local images (0)
ubuntu/apache2
1M+
102
Tag
latest

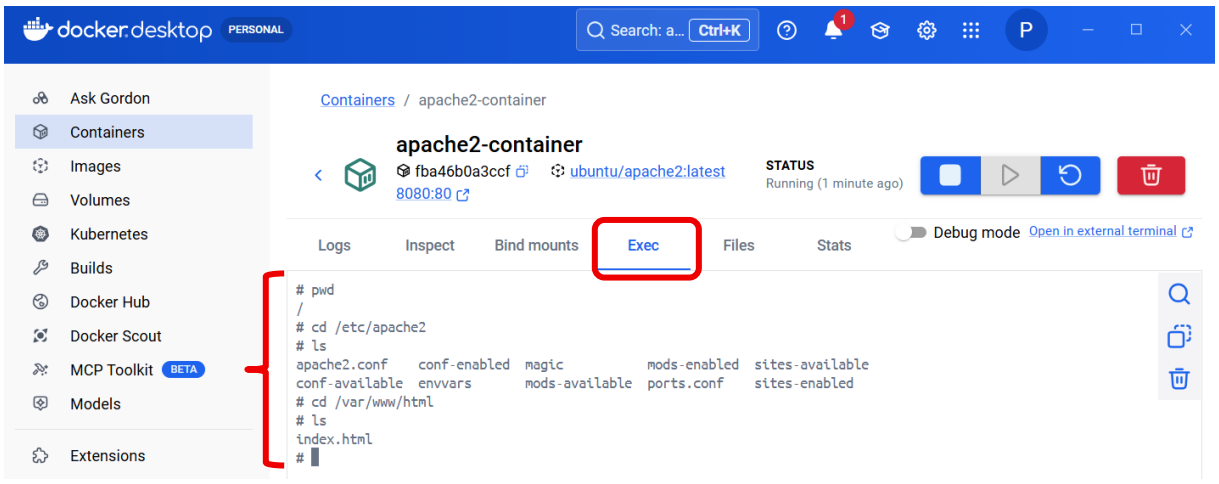

docker desktop
PERSONAL
Search: a...
Ctrl+K
Ask Gordon
Containers
Images
Volumes
Kubernetes
Builds
Models
MCP Toolkit
BETA
Docker Hub
Docker Scout
Extensions
Images
Give feedback
Local
My Hub
0 Bytes / 0 Bytes in use
1 images
Last refresh: 19 minutes ago
Search
Delete
Space to be reclaimed 175.4 MB
Name
Tag
Image ID
Created
Size
Actions
ubuntu/apache2
latest
bd68b3b35b01
3 months ago
227.63 MB

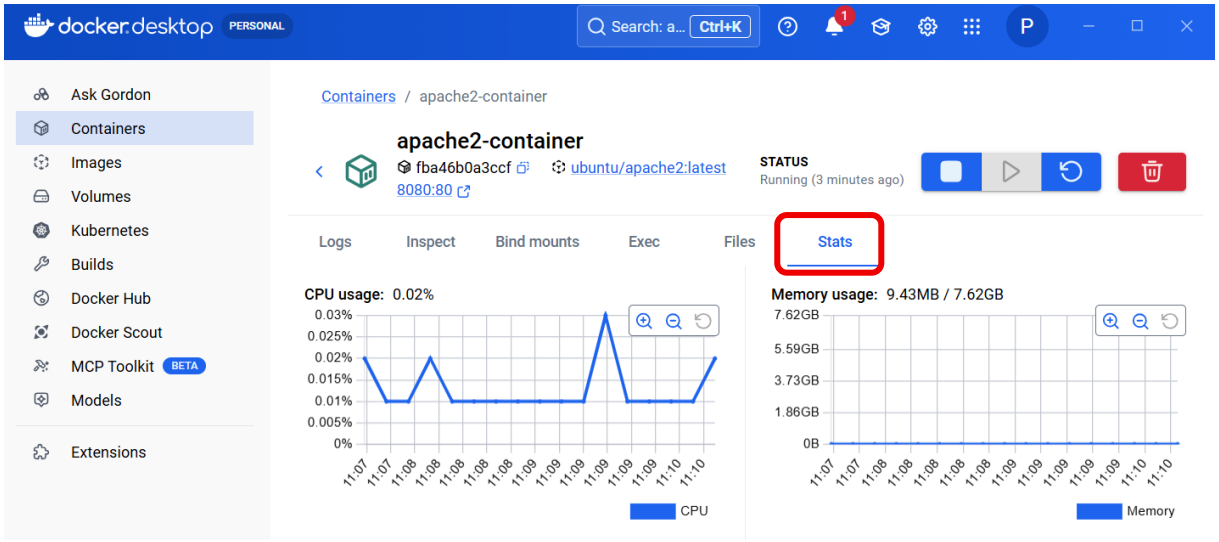
- Run :

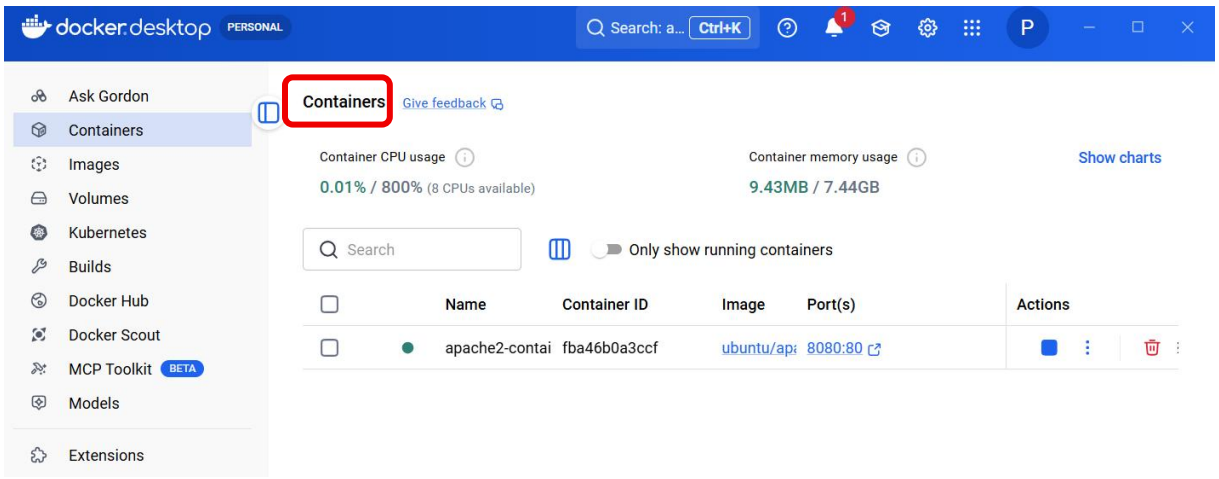

docker desktop
PERSONAL
Search: a...
Ctrl+K
Ask Gordon
Containers
Images
Volumes
Kubernetes
Builds
Models
MCP Toolkit
BETA
Docker Hub
Docker Scout
Extensions
Images / ubuntu/apache2:latest
ubuntu/apache2:latest
bd68b3b35b01
CREATED
3 months ago
SIZE
227.63 MB
Recommended fixes
Run
Analyzed by docker scout
Layers (12)
Vulnerabilities
Packages
Give feedback
0 ARG RELEASE 0 B
1 ARG LAUNCHPAD_BUILD_ARC... 0 B
2 LABEL org.opencontainers.im... 0 B
3 LABEL org.opencontainers.im... 0 B
4 ADD file:5c208fb70b021afc05... 86.11 MB
5 CMD ["/bin/bash"] 0 B
6 ENV TZ=UTC 0 B
7 |0 /bin/sh -c set -eux; apt-get ... 89.28 MB
8 STOPSIGNAL SIGWINCH 0 B
9 COPY file:c967fbf762cc2a41c... 16.38 KB
This image has not been analyzed
You can use Docker Scout to analyze local images and list its vulnerabilities.
Start analysis
Enable background indexing in Settings so your results are always ready.


Run a new container
ubuntu/apache2:latest
Optional settings
Container name
apache2-container
A random name is generated if you do not provide one.
Ports
Enter "0" to assign randomly generated host ports.
Host port
8080
:80/tcp
Volumes
Host path
Container path
Environment variables
Variable
Value
Cancel
Run


docker desktop
PERSONAL
Search: a...
Ctrl+K
Ask Gordon
Containers
Images
Volumes
Kubernetes
Builds
Docker Hub
Docker Scout
MCP Toolkit
BETA
Models
Extensions
Containers / apache2-container
apache2-container
fba46b0a3ccf
ubuntu/apache2:latest
8080:80
STATUS
Running (1 second ago)
Logs
Inspect
Bind mounts
Exec
Files
Stats
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2.
Set the 'ServerName' directive globally to suppress this message
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2.
Set the 'ServerName' directive globally to suppress this message
[Mon Feb 23 10:07:32.928442 2026] [mpm_event:notice] [pid 23:tid 23] AH00489: Apache/2.4.63 (Ubuntu)
configured -- resuming normal operations
[Mon Feb 23 10:07:32.929198 2026] [core:notice] [pid 23:tid 23] AH00094: Command line: '/usr/sbin/apache2 -D
FOREGROUND'

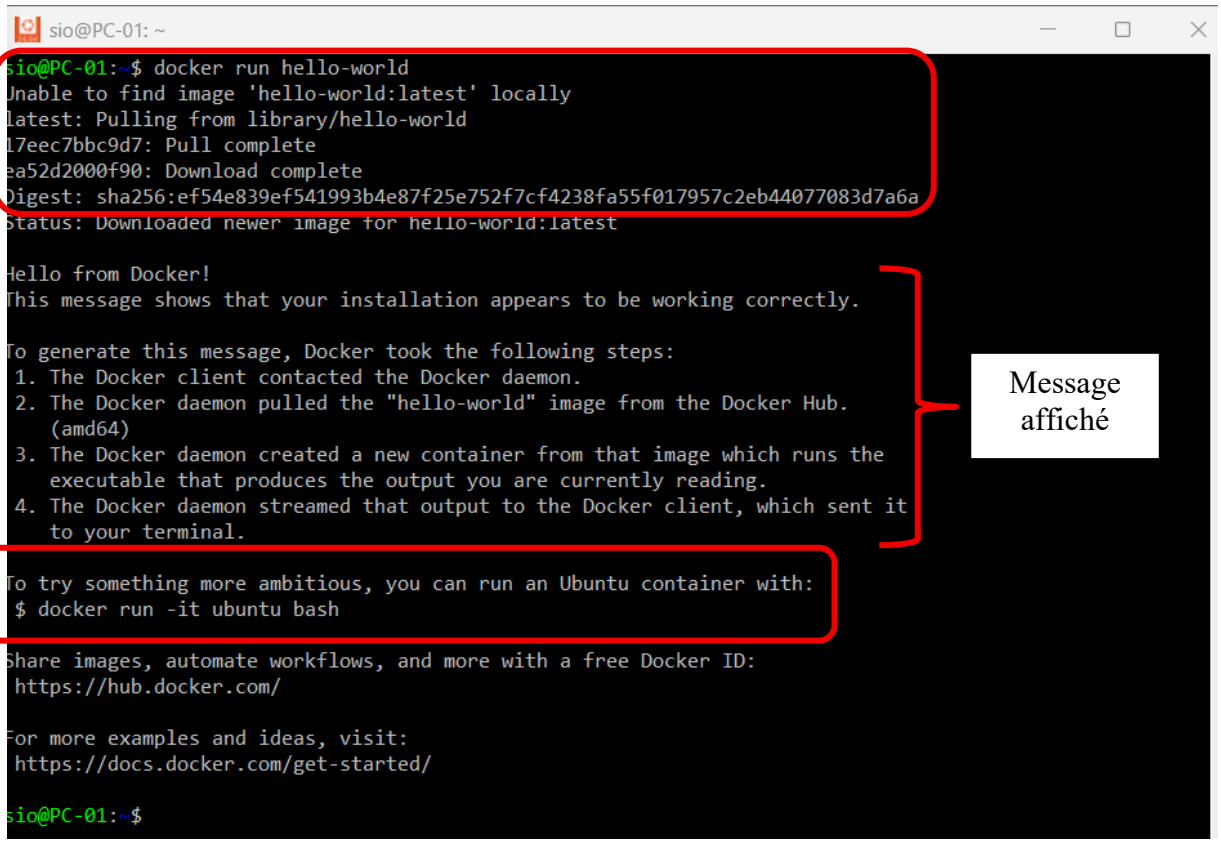

docker desktop
PERSONAL
Search: a...
Ctrl+K
Ask Gordon
Containers
Images
Volumes
Kubernetes
Builds
Docker Hub
Docker Scout
MCP Toolkit
BETA
Models
Extensions
Containers / apache2-container
apache2-container
fba46b0a3ccf
ubuntu/apache2:latest
8080:80
STATUS
Running (1 minute ago)
Logs
Inspect
Bind mounts
Exec
Files
Stats
Debug mode
Open in external terminal
pwd
/
cd /etc/apache2
ls
apache2.conf conf-enabled magic mods-enabled sites-available
conf-available envvars mods-available ports.conf sites-enabled
cd /var/www/html
ls
index.html
#


docker desktop
PERSONAL
Search: a...
Ctrl+K
Containers / apache2-container
apache2-container
fba46b0a3ccf
ubuntu/apache2:latest
8080:80
STATUS
Running (3 minutes ago)
Logs
Inspect
Bind mounts
Exec
Files
Stats
CPU usage: 0.02%
0.03%
0.025%
0.02%
0.015%
0.01%
0.005%
0%
CPU
Memory usage: 9.43MB / 7.62GB
7.62GB
5.59GB
3.73GB
1.86GB
0B
Memory

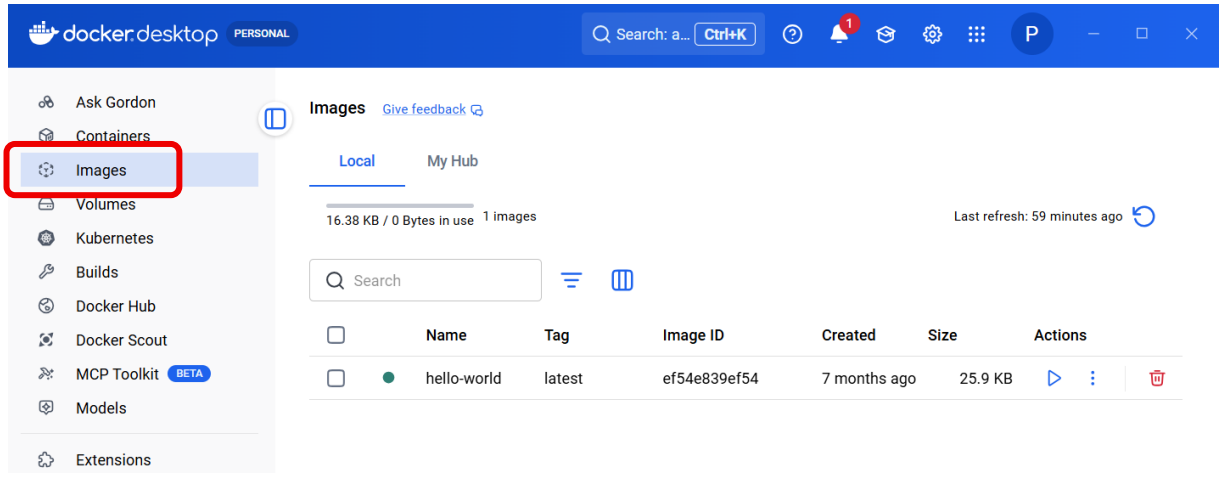

docker desktop
PERSONAL
Search: a...
Ctrl+K
Containers
Give feedback
Container CPU usage
0.01% / 800% (8 CPUs available)
Container memory usage
9.43MB / 7.44GB
Show charts
Search
Only show running containers
Name
Container ID
Image
Port(s)
Actions
apache2-contai
fba46b0a3ccf
ubuntu/apa
8080:80

- Arrêter et supprimer le container puis l'image.

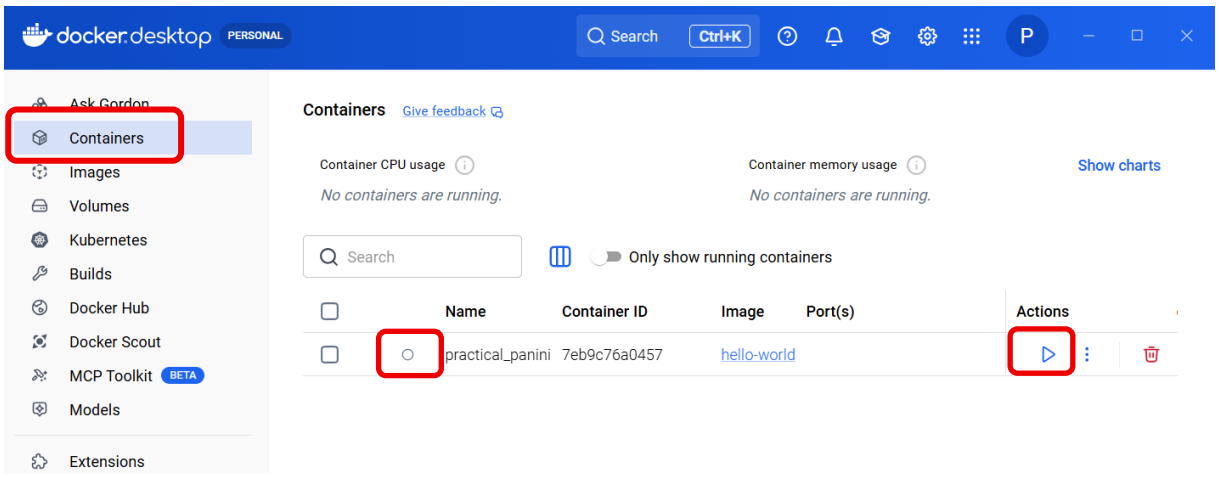
7. Quatrième test.

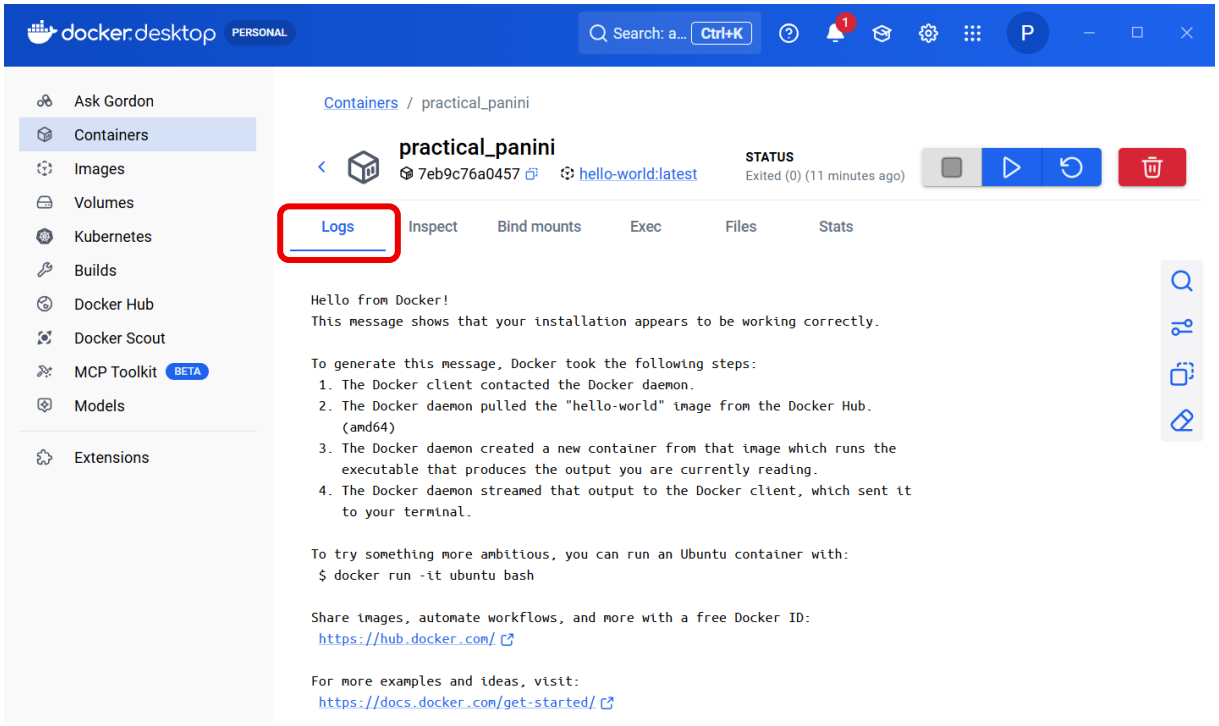


La commande docker run cherche l'image hello-world en local puis sur le dépôt en ligne (librairie en ligne cad le docker hub). Latest est le tag (étiquette).
Elle effectue 2 téléchargements car 2 couches image : une image scratch de base (vide : pas d'OS) et une destinée à exécuter le fichier hello codé en C.
Elle crée et démarre un conteneur (affichage du message) puis arrête le conteneur.
Digest : clé permettant de s'assurer que l'application est correctement téléchargée sans MITM.

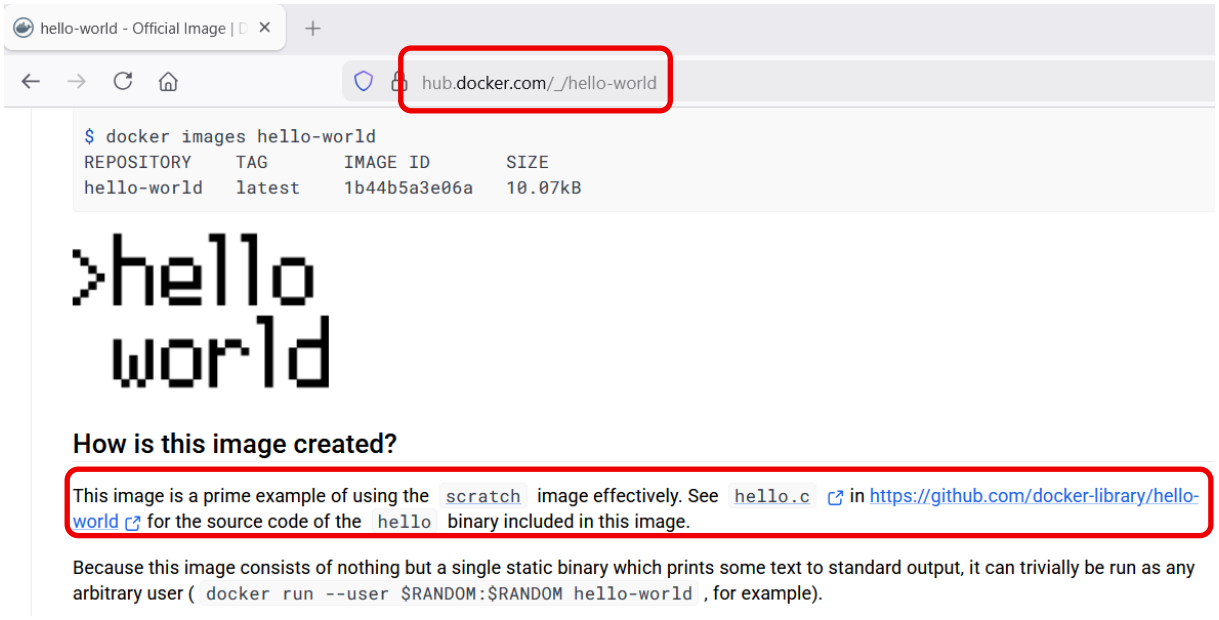


Le conteneur est arrêté après l'affichage du message :

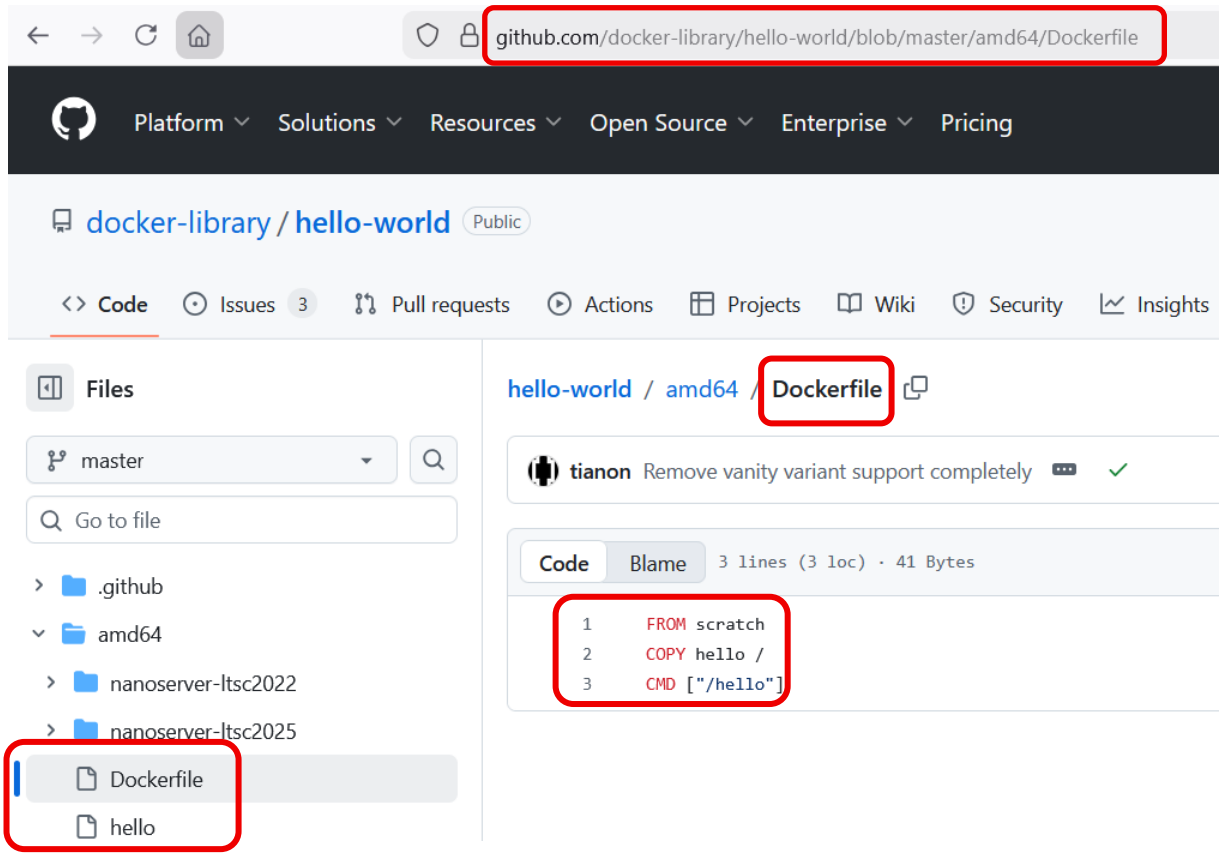




Les sources de l'image :



Les fichiers qui servent à créer l'image hello-word :

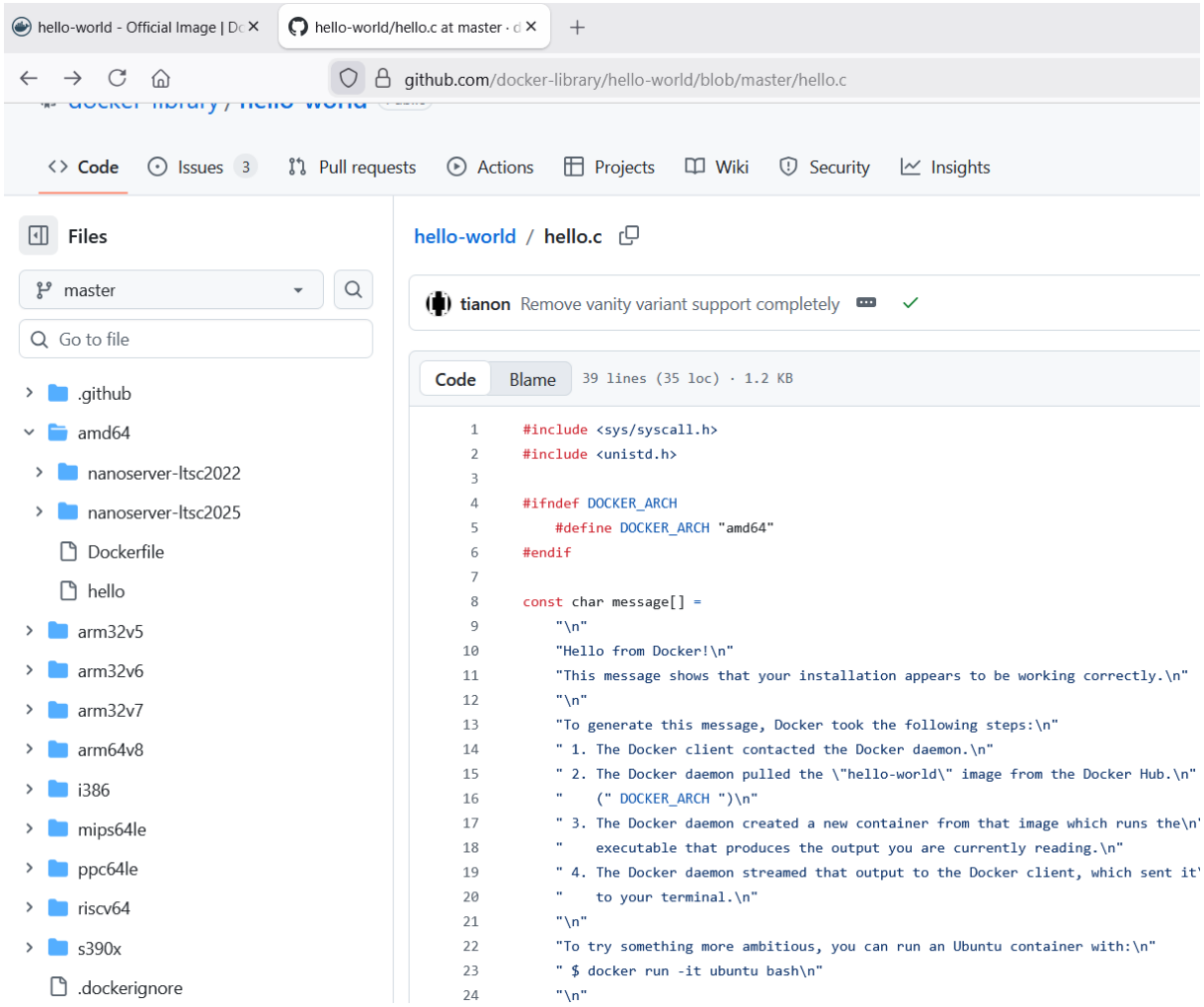


FROM scratch (image de base disposant elle aussi d'un DockerFile) : cela signifie que l'on part de rien c'est à dire pas d'image avec un OS.

COPY hello / : le fichier hello est copié à la racine de la machine.

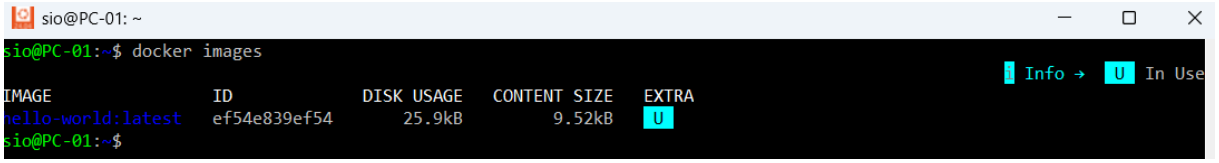
CMD [« /hello »] : commande lancée au démarrage de l'image et exécutée par l'OS Ubuntu

Code du fichier hello :

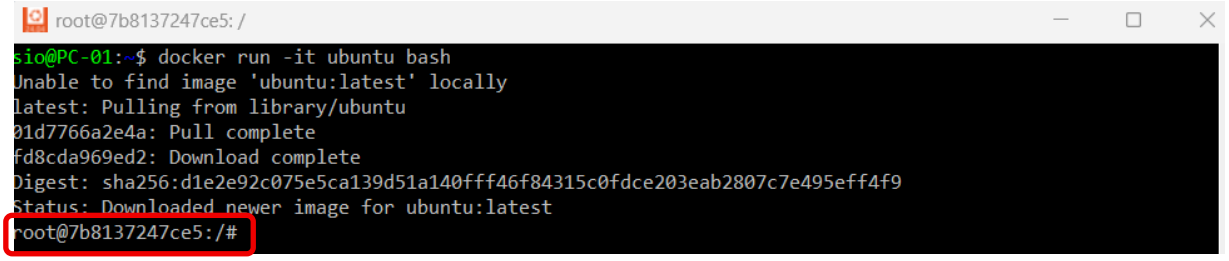


8. Cinquième test : mode interactif d'utilisation d'un conteneur.

- Commande **docker images** : liste toutes les images sur le cache local de la machine (identifiant = code hexadécimal)

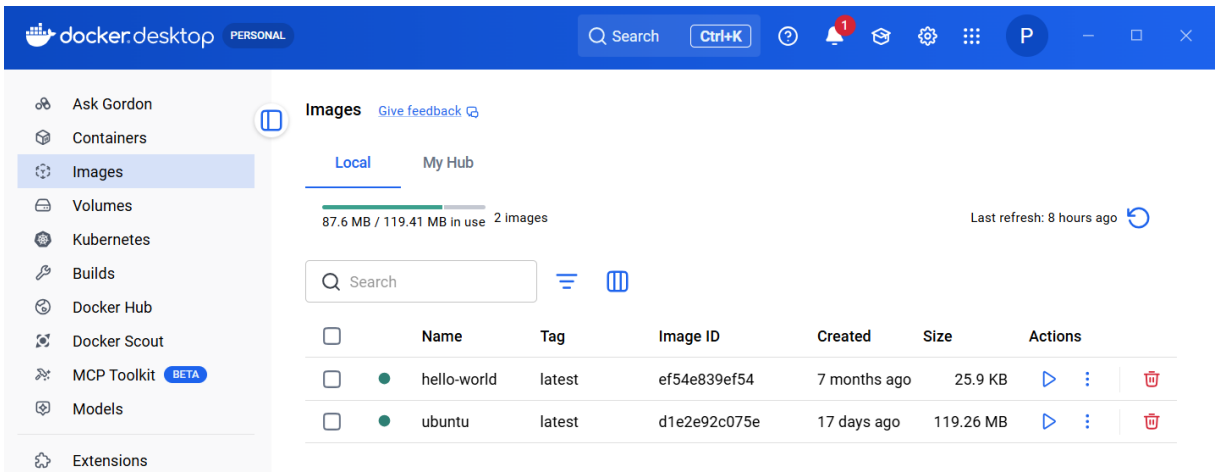


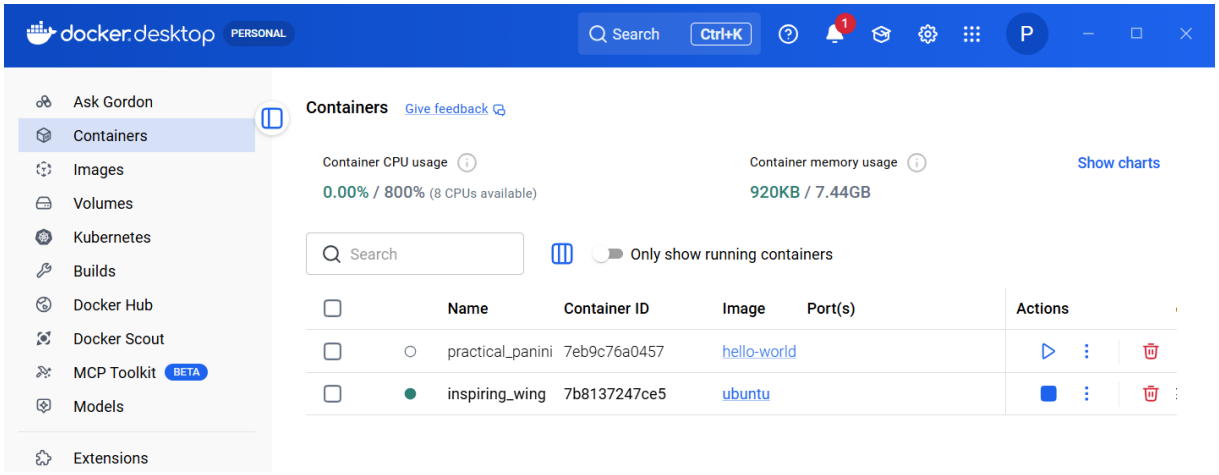
- Lancement d'un conteneur en mode interactif qui lancera un shell bash qui interprètera les commandes :



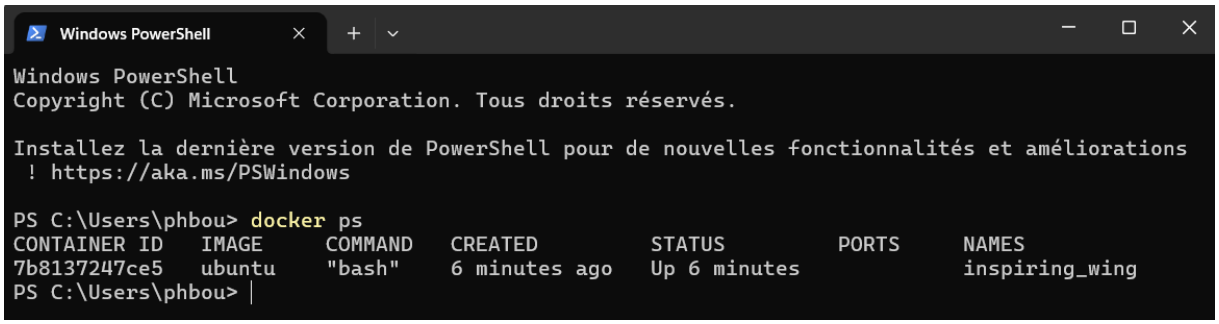
Option -i : permet d'interagir avec le conteneur.

Option -t : permet de saisir du texte (des commandes) dans une console

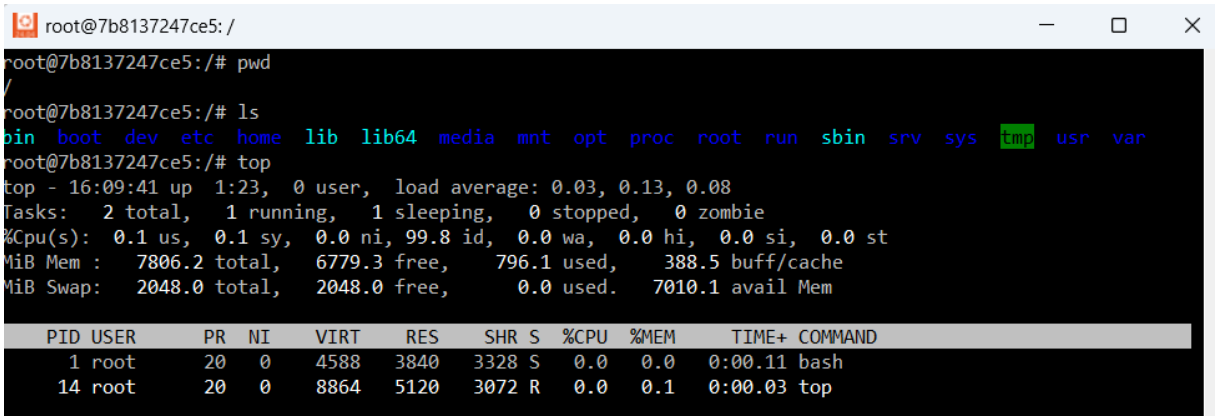




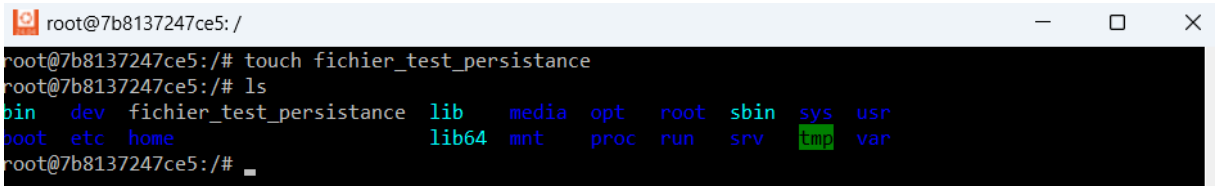
- La commande docker ps (depuis la console de la machine Ubuntu ou depuis le terminal Windows) permet de lister les containeurs en cours d'exécution :



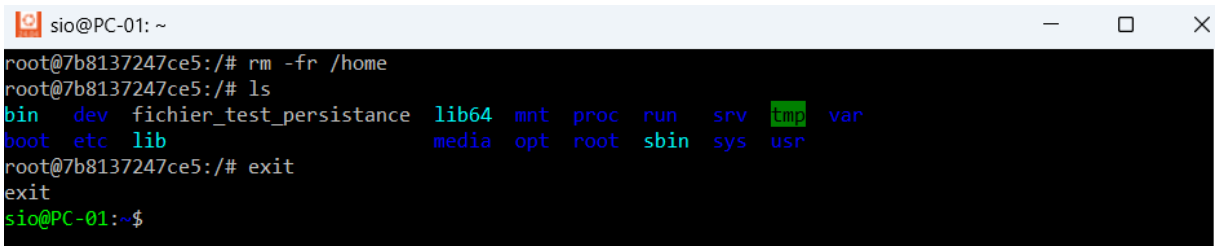
- Taper quelques commandes (ctrl+c pour quitter la commande top).

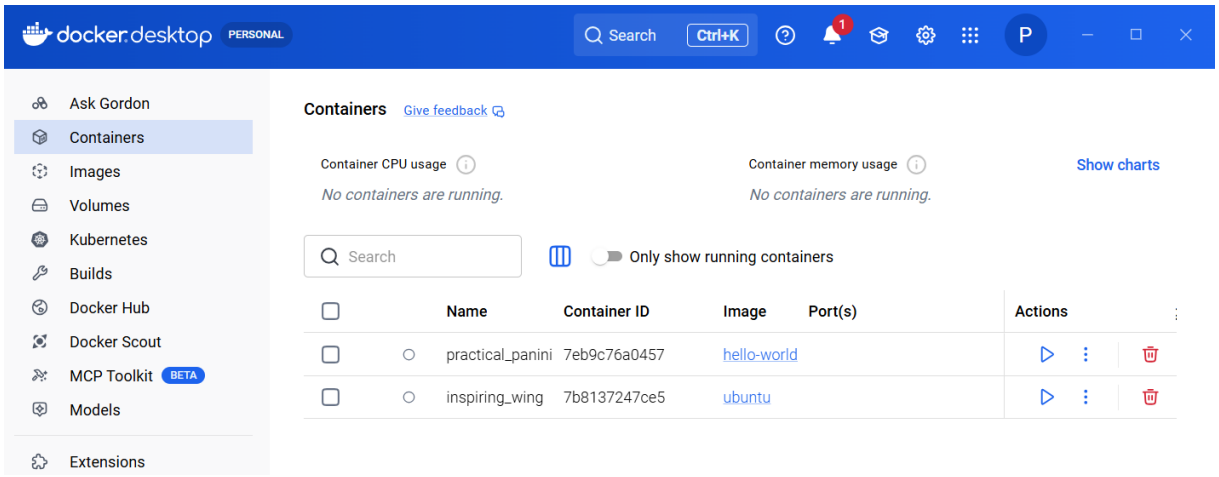


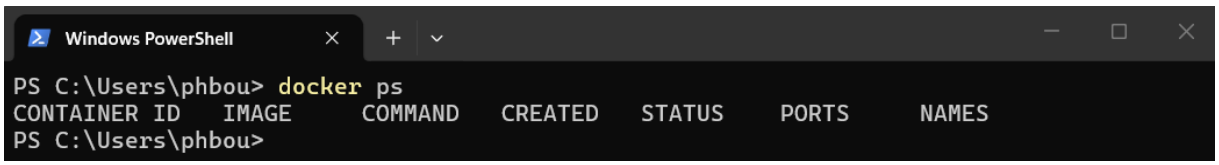
- Créer un fichier, vérifier sa présence :

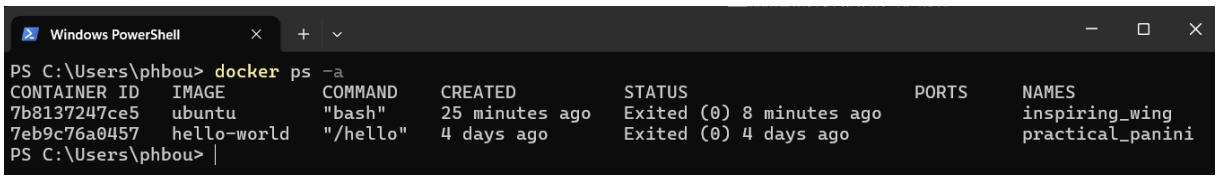


- Supprimer /home, vérifier sa suppression puis stopper le conteneur.

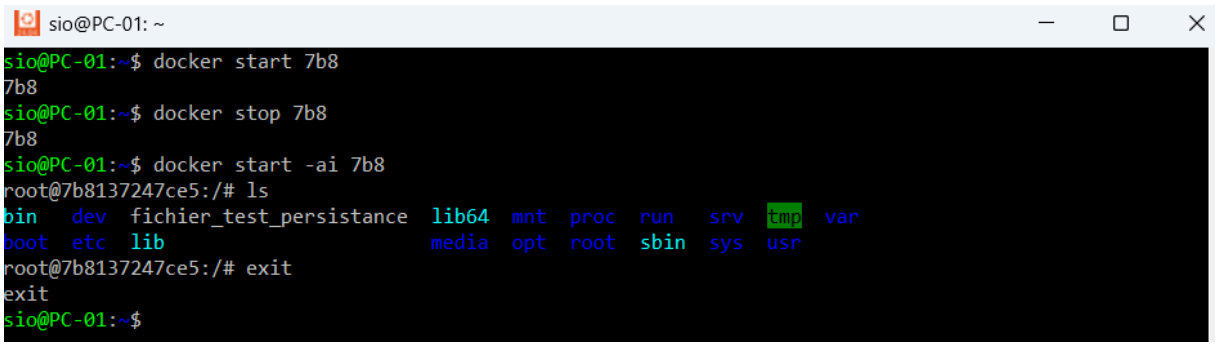






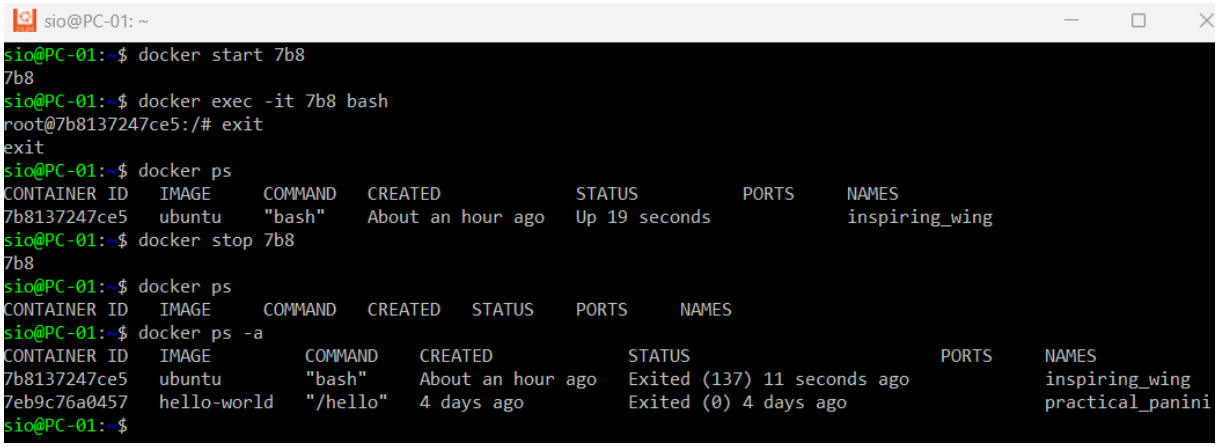


- Redémarrer le conteneur et y accéder (option -ai) :



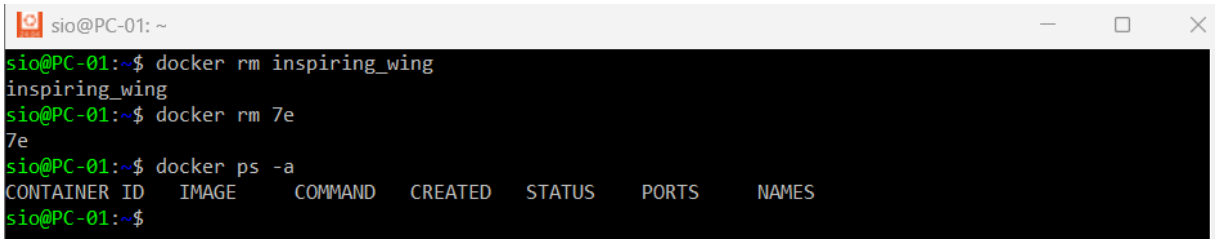
➔ Les modifications sont conservées tant que le conteneur n'est pas supprimé.

- Autre manière d'accéder au conteneur :

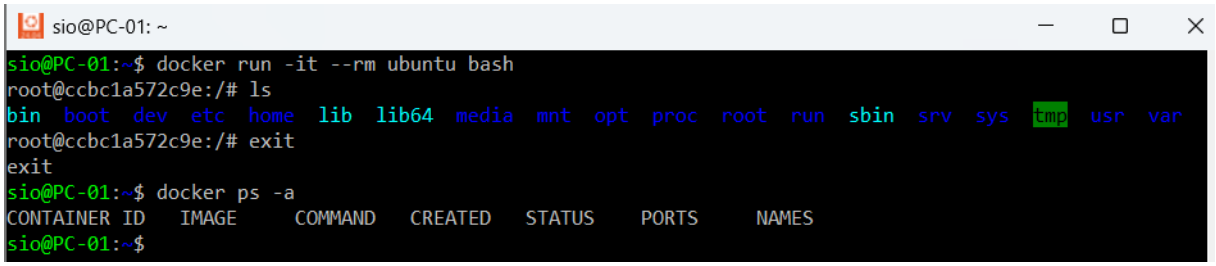


- Supprimer en ligne de commande les 2 containers

Un conteneur doit être arrêté avant d'être supprimé (option -f pour forcer l'arrêt du conteneur avant sa suppression ; ici, ce n'est pas nécessaire puisque les deux conteneurs sont déjà arrêtés).

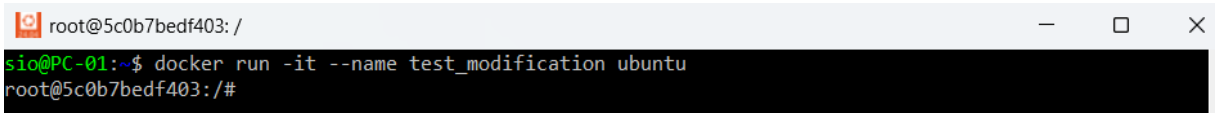


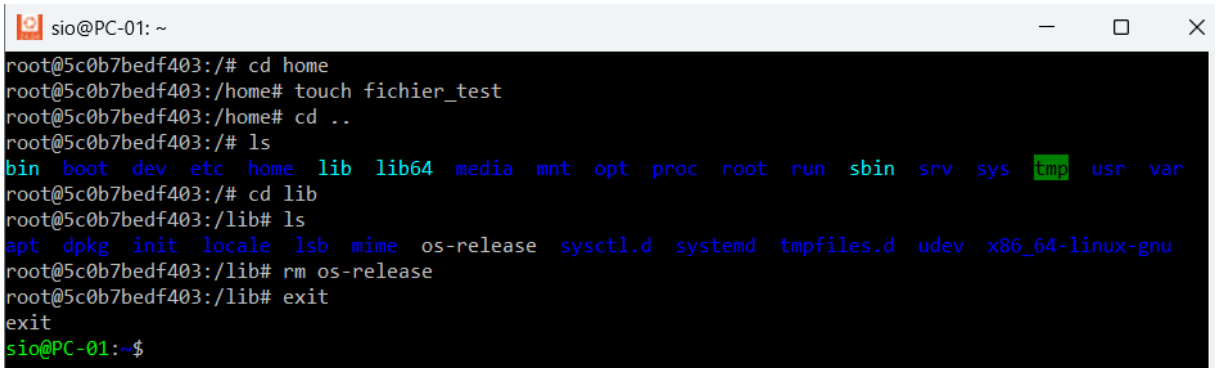
- Relancer l'image Ubuntu et constater la création d'un nouveau conteneur (évidemment, plus de fichier_test_persistance et de nouveau la présence de /home), prévoir la suppression automatique du conteneur après son utilisation avec l'option --rm :



9. Création d'une image à la main avec docker commit.

On souhaite conserver les modifications une fois le conteneur arrêté et supprimé.





```
sio@PC-01: ~
sio@PC-01:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
sio@PC-01:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND       CREATED         STATUS                     PORTS     NAMES
5c0b7bedf403   ubuntu    "/bin/bash"   5 minutes ago   Exited (0) 54 seconds ago            test_modifi
cation
sio@PC-01:~$ docker diff test_modification
C /usr
C /usr/lib
D /usr/lib/os-release
C /home
A /home/fichier_test
C /root
A /root/.bash_history
sio@PC-01:~$
```

```
sio@PC-01: ~
sio@PC-01:~$ docker commit --help
Usage:  docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]

Create a new image from a container's changes

Aliases:
  docker container commit, docker commit

Options:
  -a, --author string    Author (e.g., "John Hannibal Smith <hannibal@a-team.com>")
  -c, --change list      Apply Dockerfile instruction to the created image
  -m, --message string   Commit message
      --no-pause         Disable pausing container during commit
sio@PC-01:~$
```

```
sio@PC-01: ~
sio@PC-01:~$ docker commit test_modification ubuntumodif:1.0
sha256:c468643c70dd50171f451655c5db166188272a321a287ff4d5f271301c0632dd
sio@PC-01:~$
```

```
sio@PC-01: ~
sio@PC-01:~$ docker images
                                                                  i Info →  U  In Use
IMAGE                ID             DISK USAGE   CONTENT SIZE   EXTRA
hello-world:latest   ef54e839ef54       25.9kB         9.52kB
ubuntu:latest        d1e2e92c075e        119MB         31.7MB    U
ubuntumodif:1.0      c468643c70dd        117MB         29.7MB
sio@PC-01:~$
```

```
sio@PC-01: ~
sio@PC-01:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND       CREATED          STATUS                      PORTS     NAMES
5c0b7bedf403   ubuntu    "/bin/bash"   16 minutes ago   Exited (0) 11 minutes ago             test_modif
ication
sio@PC-01:~$ docker rm test_modification
test_modification
sio@PC-01:~$
```

```
sio@PC-01: ~
sio@PC-01:~$ docker run -it --rm --name test ubuntumodif:1.0
root@d5ccbd2f681e:/# ls /home
fichier_test  ubuntu
root@d5ccbd2f681e:/# exit
exit
sio@PC-01:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
sio@PC-01:~$
```

10. Première approche des volumes.

Les données sont sur le système local : création d'un répertoire web sur la machine ubuntu :

```
sio@PC-01: ~/web
sio@PC-01:~$ pwd
/home/sio
sio@PC-01:~$ mkdir web
sio@PC-01:~$ cd web
sio@PC-01:~/web$ nano index.html
```

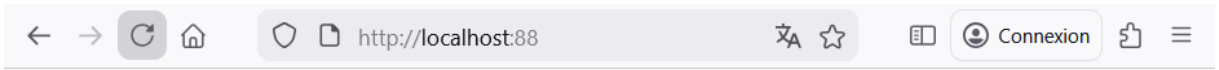
```
sio@PC-01: ~/web
 GNU nano 7.2                                  index.html
<html<>
<body>
<h1>Page web Nginx conteneur Docker</h1>
</body>
</html>
```

Création d'un conteneur qui va pointer sur le répertoire web du FS local :

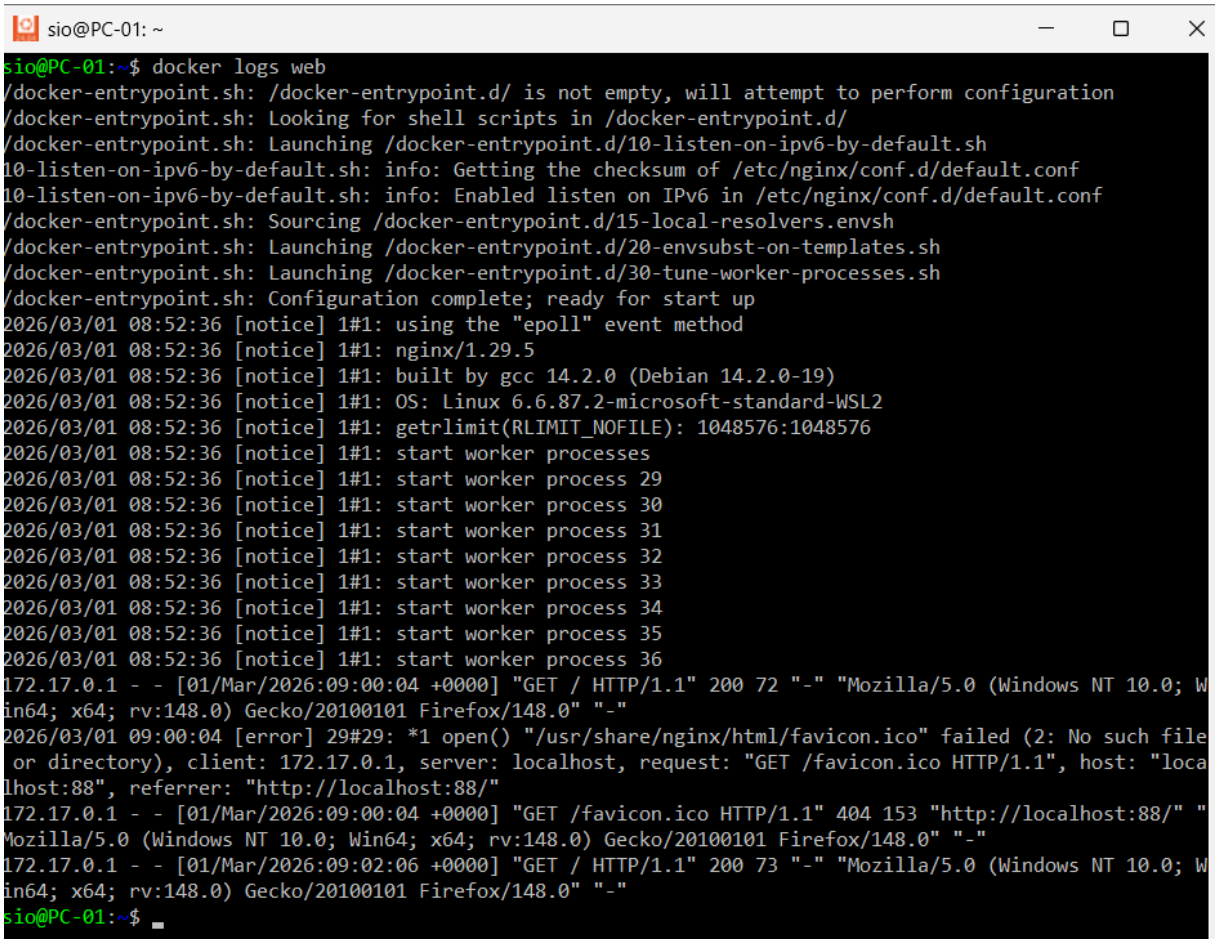
➔ Le conteneur n'est pas utilisé ici en mode interactif (option -d) mais en mode serveur, de manière détachée, en background. Il rend la main.

➔ On met en place un mappage de volume (option -v) : le répertoire local /home/sio prend le dessus sur le répertoire de publication /usr/share/nginx/html du conteneur (ce dernier contiendra le fichier index.html).

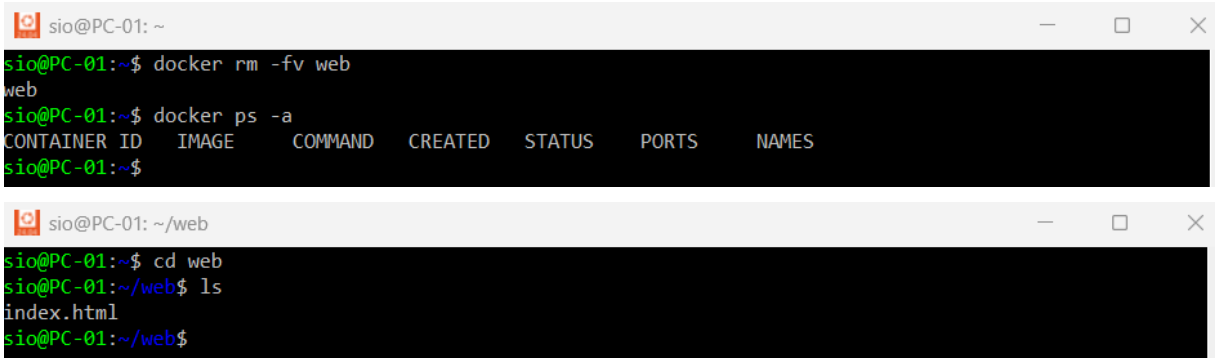
```
sio@PC-01: ~
sio@PC-01:~$ docker run -d --name web -v /home/sio/web:/usr/share/nginx/html:ro -p 88:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
df0b66c867e4: Pull complete
17d0911eaf62: Pull complete
eedda9fd8786: Pull complete
35ff83c394d6: Pull complete
b47f187216b6: Pull complete
1ad233904a11: Pull complete
206356c42440: Pull complete
0019bbecc572: Download complete
c13150b54a53: Download complete
Digest: sha256:0236ee02dcbce00b9bd83e0f5fbc51069e7e1161bd59d99885b3ae1734f3392e
Status: Downloaded newer image for nginx:latest
56f3c387f4b09e08565cf5509ef5f57bf518f96984440f794f8ed3a4d31fda15
sio@PC-01:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS
               NAMES
56f3c387f4b0   nginx     "/docker-entrypoint.…"   3 minutes ago   Up 3 minutes   0.0.0.0:88->80/tcp, [::
]:88->80/tcp   web
sio@PC-01:~$
```



Page web Nginx conteneur Docker



Arrêt et suppression du conteneur web :



11. Gestion des volumes en écriture.

Quand un conteneur est supprimé, ses fichiers et données disparaissent avec lui. Les **volumes Docker** permettent de contourner ce problème en offrant un mécanisme dit de **persistance des données** : contrairement aux fichiers du conteneur, le volume **survit à la suppression du conteneur**.

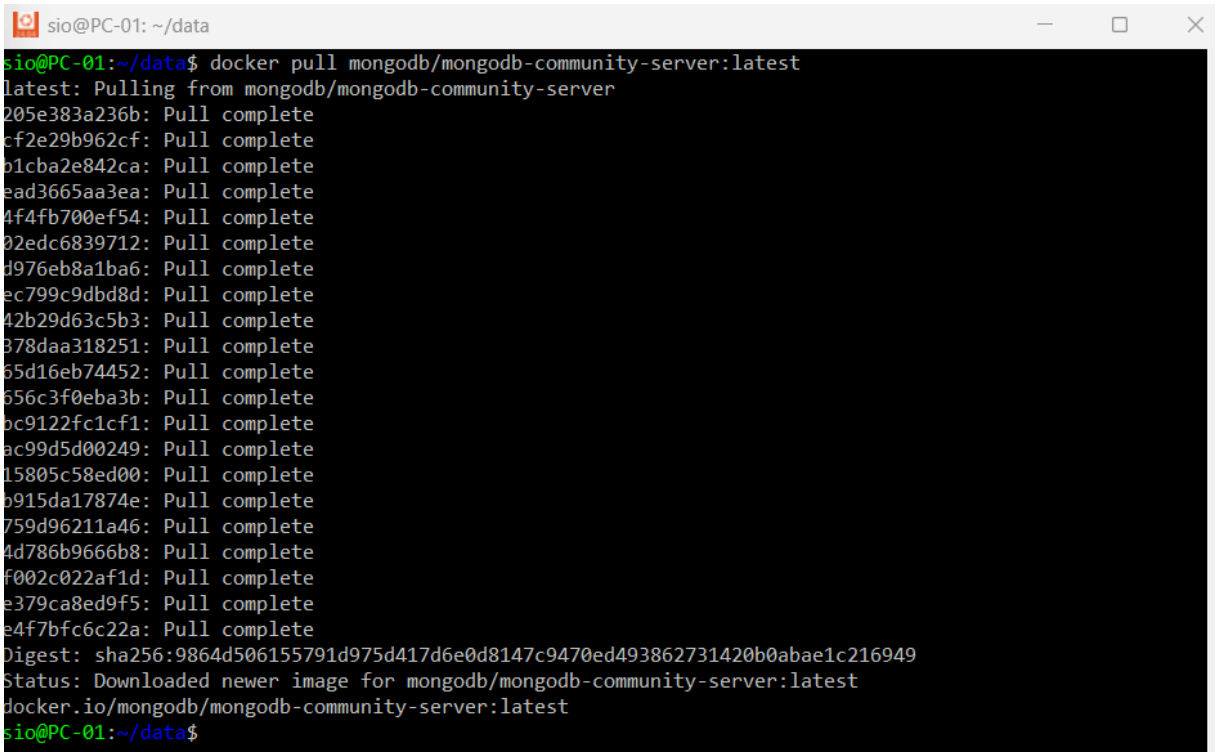
11.1. Premier type de volume : volume nommé

Il s'agit d'un **espace de stockage dédié**, créé et géré par Docker dans le répertoire **/var/lib/docker/volumes** de la machine hôte.

Il n'est pas nécessaire de connaître l'emplacement de ce répertoire, tout est géré automatiquement. Lorsqu'un conteneur est supprimé, le volume reste inchangé. Il suffit de démarrer un nouveau conteneur en y attachant le même volume pour retrouver toutes les données.

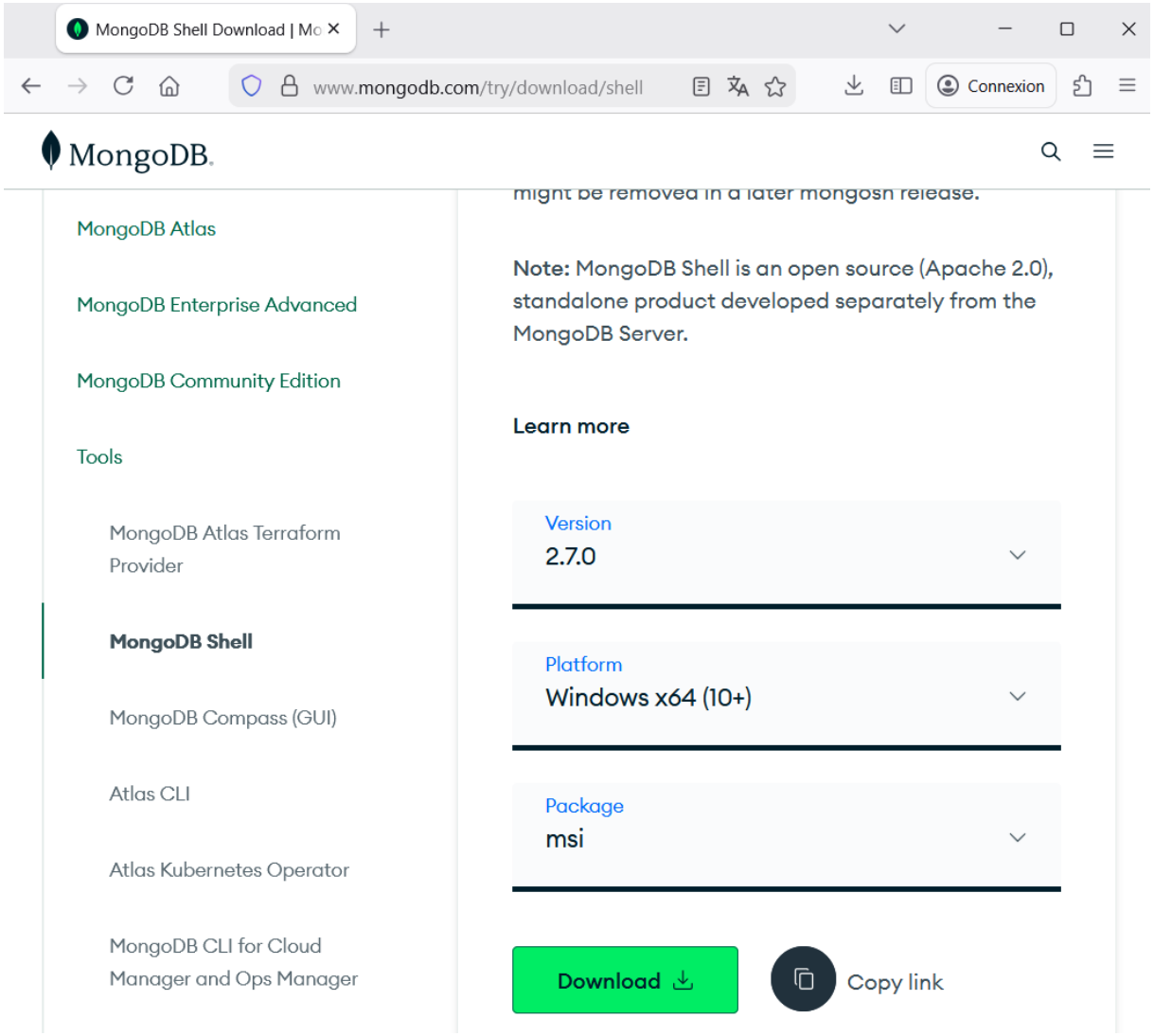
Application :

Lancement d'un conteneur qui va exécuter MongoDB qui est un système de gestion de bases de données NoSQL (ne nécessite pas de système de gestion de bases de données relationnelles). MongoDB manipule des objets structurés au format JSON. Les objets de données sont stockés dans des collections et des documents plutôt que dans des tables et des enregistrements.

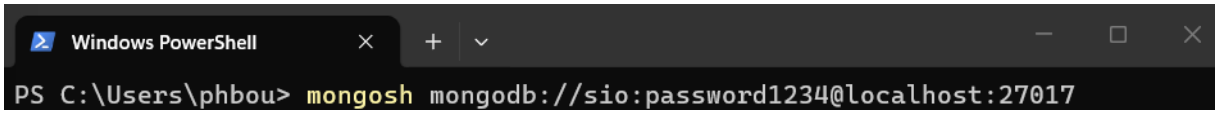


La commande ci-dessus prendra l'image téléchargé et la déploiera dans un conteneur nommé « MongoDB ». Le conteneur sera mappé sur un port, ce qui signifie que la machine Ubuntu pourra interagir avec lui sur le port « 27017 ».

Installez MongoDB Shell sur la machine physique pour pouvoir vous connecter au serveur MongoDB :



Connectez-vous au serveur MongoDB :



```
Current Mongosh Log ID: 69a48449e234379fa77c2906
Connecting to:          mongodb://<credentials>@localhost:27017/?directConnection=tr
ue&serverSelectionTimeoutMS=2000&appName=mongosh+2.7.0
Using MongoDB:          8.2.5
Using Mongosh:          2.7.0

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2026-03-01T17:10:16.909+00:00: Using the XFS filesystem is strongly recommended w
ith the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-file
system
   2026-03-01T17:10:17.413+00:00: For customers running the current memory allocator
, we suggest changing the contents of the following sysfsFile
   2026-03-01T17:10:17.413+00:00: For customers running the current memory allocator
, we suggest changing the contents of the following sysfsFile
   2026-03-01T17:10:17.413+00:00: We suggest setting the contents of sysfsFile to 0.
   2026-03-01T17:10:17.413+00:00: We suggest setting swappiness to 0 or 1, as swappi
ng can cause performance problems.
------

test> use("ma_bdd");
switched to db ma_bdd
ma_bdd> db.createCollection("etudiants");
{ ok: 1 }
ma_bdd> db.etudiants.insertOne({"nom":"smet", "prenom":"jp"});
{
  acknowledged: true,
  insertedId: ObjectId('69a48994e234379fa77c2907')
}
ma_bdd> db.etudiants.find();
[
  {
    _id: ObjectId('69a48994e234379fa77c2907'),
    nom: 'smet',
    prenom: 'jp'
  }
]
ma_bdd> exit
PS C:\Users\phbou>
```

Toutes les données sont stockées dans le conteneur. Les données MongoDB, qui comprennent les bases de données, les collections et même les documents, ne seront conservées que tant que le conteneur concerné existera.

```
sio@PC-01: ~
sio@PC-01:~$ docker inspect mongodb
[
    {
        "Id": "ba989cff4db83791507e15ee984a88cbd00e93575bc9067e4a70ad9f89fcfbda",
        "Created": "2026-03-01T17:10:12.887375053Z",
        "Path": "python3",
        "Args": [
            "/usr/local/bin/docker-entrypoint.py",
            "mongod"
        ],
        "State": {
            "Status": "running",
            "Running": true,
```

```
        },
        "Mounts": [
            {
                "Type": "volume",
                "Name": "7221ea7fe2bc45d46d3fdd7eb7950e460707c2bec6de4c39d7e143a0625d3097",
                "Source": "/var/lib/docker/volumes/7221ea7fe2bc45d46d3fdd7eb7950e460707c2bec6de4c39d7e14
3a0625d3097/_data",
                "Destination": "/data/configdb",
                "Driver": "local",
                "Mode": "",
                "RW": true,
                "Propagation": ""
            },
            {
                "Type": "volume",
                "Name": "027081e92124d73dc9b8112df4159c35dcb3395258783f78c88017858e8b2539",
                "Source": "/var/lib/docker/volumes/027081e92124d73dc9b8112df4159c35dcb3395258783f78c8801
7858e8b2539/_data",
                "Destination": "/data/db",
                "Driver": "local",
                "Mode": "",
                "RW": true,
                "Propagation": ""
            }
        ],
```

```
            "Env": [
                "MONGODB_INITDB_ROOT_USERNAME=sio",
                "MONGODB_INITDB_ROOT_PASSWORD=password1234",
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
                "HOME=/data/db",
                "GLIBC_TUNABLES=glibc.pthread.rseq=0"
            ],
            "Cmd": [
                "mongod"
            ],
            "Image": "mongodb/mongodb-community-server:latest",
            "Volumes": {
                "/data/configdb": {},
                "/data/db": {}
            },
```


```
sio@PC-01: ~
sio@PC-01:~$ docker exec -it mongodb bash
mongodb@ba989cff4db8:/$ cd /data
mongodb@ba989cff4db8:/data$ ls
configdb  db
mongodb@ba989cff4db8:/data$ cd db
mongodb@ba989cff4db8:~$ ls
WiredTiger                                    diagnostic.data
WiredTiger.lock                               index-02fa3b56-e171-4107-bcb4-52e0c99ee2fb.wt
WiredTiger.turtle                             index-4af79d0c-b92b-478c-b321-0a1bd3d54ebb.wt
WiredTiger.wt                                 index-8c7c8061-b106-4c86-b44c-82024ea60a7d.wt
WiredTigerHS.wt                               index-93ca7c7b-2297-4270-9288-0aa45fa38e10.wt
_mdb_catalog.wt                               index-cff671d3-7c14-4c86-9b98-8699d5e203d6.wt
_tmp                                          index-f738bd7a-e431-4caa-9898-3505f35992d7.wt
collection-21ab4c29-742e-4c6d-aa1e-be29d2d78dfb.wt  index-fc509acf-13a1-4c3d-a2ed-8793fe05ae0a.wt
collection-302ed8a6-7cb9-4964-aeac-dc66d76bdf63.wt  journal
collection-ad264049-bf97-4b67-ac53-751ed78f388c.wt  mongod.lock
collection-be904508-334a-452f-b4ad-ebbe38eca221.wt  sizeStorer.wt
collection-c0e4af4f-b0e8-4684-8789-885ebc36c8ed.wt  storage.bson
mongodb@ba989cff4db8:~$ exit
exit
sio@PC-01:~$
```

Arrêt et suppression du conteneur : les données vont être perdues.

```
sio@PC-01: ~
sio@PC-01:~$ docker rm -f mongodb
mongodb
sio@PC-01:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
sio@PC-01:~$
```

Nous souhaitons bénéficier d'une persistance des données MongoDB dans Docker. Créez un volume nommé à l'aide de la commande docker volume create :

```
sio@PC-01: ~
sio@PC-01:~$ docker volume create mon_volume_mongodb
mon_volume_mongodb
sio@PC-01:~$
```

```
sio@PC-01: ~
sio@PC-01:~$ docker volume ls
DRIVER    VOLUME NAME
local     182e20c52b7c382b971bdd80a6c074822a3af8fed2fbd328d84b46625b9fa797
local     7221ea7fe2bc45d46d3fdd7eb7950e460707c2bec6de4c39d7e143a0625d3097
local     027081e92124d73dc9b8112df4159c35dcb3395258783f78c88017858e8b2539
local     mon_volume_mongodb
sio@PC-01:~$
```

```
sio@PC-01: ~
sio@PC-01:~$ docker volume inspect mon_volume_mongodb
[
    {
        "CreatedAt": "2026-03-01T22:43:57Z",
        "Driver": "local",
        "Labels": null,
        "Mountpoint": "/var/lib/docker/volumes/mon_volume_mongodb/_data",
        "Name": "mon_volume_mongodb",
        "Options": null,
        "Scope": "local"
    }
]
sio@PC-01:~$
```

Sur **Linux**, les volumes sont directement accessibles dans /var/lib/docker/volumes/. Sur **macOS** et **Windows** (Docker Desktop), Docker tourne dans une machine virtuelle : les volumes ne sont pas directement accessibles depuis l'explorateur de fichiers de l'hôte.

```
sio@PC-01: ~
sio@PC-01:~$ docker run -d --name mongodb -p 27017:27017 -e MONGODB_INITDB_ROOT_USERNAME=sio -e MONGODB_
INITDB_ROOT_PASSWORD=password1234 -v mon_volume_mongodb:/data/db mongodb/mongodb-community-server:latest
66a434bcf6f72f28e0d8f2ffdfd69718ed7fb51c72786a13c1b93633c6913214
sio@PC-01:~$
```

```
Windows PowerShell
PS C:\Users\phbou> mongosh mongodb://sio:password1234@localhost:27017
```

```
Current Mongosh Log ID: 69a4c275e7ce1e99587c2906
Connecting to:          mongodb://<credentials>@localhost:27017/?directConne
ction=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.7.0
Using MongoDB:          8.2.5
Using Mongosh:          2.7.0

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2026-03-01T22:47:08.494+00:00: Using the XFS filesystem is strongly recom
mended with the WiredTiger storage engine. See http://dochub.mongodb.org/cor
e/prodnotes-filesystem
   2026-03-01T22:47:09.396+00:00: For customers running the current memory a
llocator, we suggest changing the contents of the following sysfsFile
   2026-03-01T22:47:09.396+00:00: For customers running the current memory a
llocator, we suggest changing the contents of the following sysfsFile
   2026-03-01T22:47:09.396+00:00: We suggest setting the contents of sysfsFi
le to 0.
   2026-03-01T22:47:09.397+00:00: We suggest setting swappiness to 0 or 1, a
s swapping can cause performance problems.
------

test> use("ma_bdd");
switched to db ma_bdd
ma_bdd> db.createCollection("etudiants");
{ ok: 1 }
ma_bdd> db.etudiants.insertOne({"nom":"smet", "prenom":"jp"});
{
  acknowledged: true,
  insertedId: ObjectId('69a4c386e7ce1e99587c2907')
}
ma_bdd> db.etudiants.find();
[
  {
    _id: ObjectId('69a4c386e7ce1e99587c2907'),
    nom: 'smet',
    prenom: 'jp'
  }
]
ma_bdd> exit
```

```
sio@PC-01: ~
sio@PC-01:~$ docker ps
CONTAINER ID   IMAGE                                     COMMAND                  CREATED          STATU
S          PORTS                                         NAMES
66a434bcf6f7   mongodb/mongodb-community-server:latest   "python3 /usr/local/…"   15 minutes ago   Up 15
 minutes   0.0.0.0:27017->27017/tcp, [::]:27017->27017/tcp   mongodb
sio@PC-01:~$
```

```
sio@PC-01: ~
sio@PC-01:~$ docker stop mongodb
mongodb
sio@PC-01:~$ docker rm -v mongodb
mongodb
sio@PC-01:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
sio@PC-01:~$
```

Après suppression du conteneur, si on redémarre un conteneur mongo :

```
sio@PC-01: ~
sio@PC-01:~$ docker run -d --name mongodb -p 27017:27017 -e MONGODB_INITDB_ROOT_USERNAME=sio -e MONGODB_
INITDB_ROOT_PASSWORD=password1234 -v mon_volume_mongodb:/data/db mongodb/mongodb-community-server:latest
d639aa988c06e520525b712e60295dbed6361c575f21ef5191d5b3c17891fff0
sio@PC-01:~$
```

La persistance a bien été gérée :

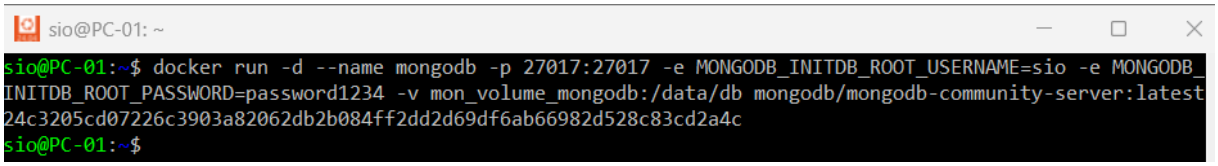
```
Windows PowerShell
PS C:\Users\phbou> mongosh mongodb://sio:password1234@localhost:27017
Current Mongosh Log ID: 69a4ce4ce0afdc898d7c2906
Connecting to:          mongodb://<credentials>@localhost:27017/?directConne
ction=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.7.0
Using MongoDB:          8.2.5
Using Mongosh:          2.7.0

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2026-03-01T23:38:56.990+00:00: Using the XFS filesystem is strongly recom
mended with the WiredTiger storage engine. See http://dochub.mongodb.org/cor
e/prodnotes-filesystem
   2026-03-01T23:38:57.775+00:00: For customers running the current memory a
llocator, we suggest changing the contents of the following sysfsFile
   2026-03-01T23:38:57.775+00:00: For customers running the current memory a
llocator, we suggest changing the contents of the following sysfsFile
   2026-03-01T23:38:57.775+00:00: We suggest setting the contents of sysfsFi
le to 0.
   2026-03-01T23:38:57.776+00:00: We suggest setting swappiness to 0 or 1, a
s swapping can cause performance problems.
------

test> use("ma_bdd");
switched to db ma_bdd
ma_bdd> db.etudiants.find();
[
  {
    _id: ObjectId('69a4c386e7ce1e99587c2907'),
    nom: 'smet',
    prenom: 'jp'
  }
]
ma_bdd> exit
PS C:\Users\phbou>
```

```
sio@PC-01: ~
sio@PC-01:~$ docker rm -fv mongodb
mongodb
sio@PC-01:~$
```

Ajoutez une deuxième ligne pour vérifier la persistance.

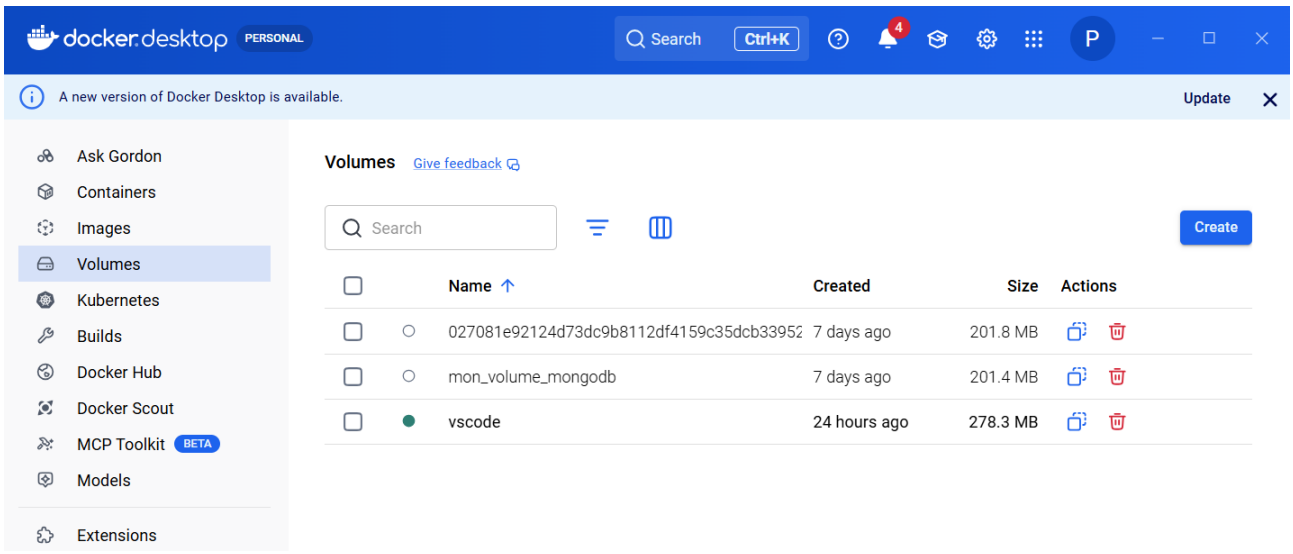
```
sio@PC-01: ~
sio@PC-01:~$ docker run -d --name mongodb -p 27017:27017 -e MONGODB_INITDB_ROOT_USERNAME=sio -e MONGODB_
INITDB_ROOT_PASSWORD=password1234 -v mon_volume_mongodb:/data/db mongodb/mongodb-community-server:latest
8db20fce2232bbfaec89d6d77dbde92246bdd80ece1074b9b4d6506422b02926
sio@PC-01:~$
```

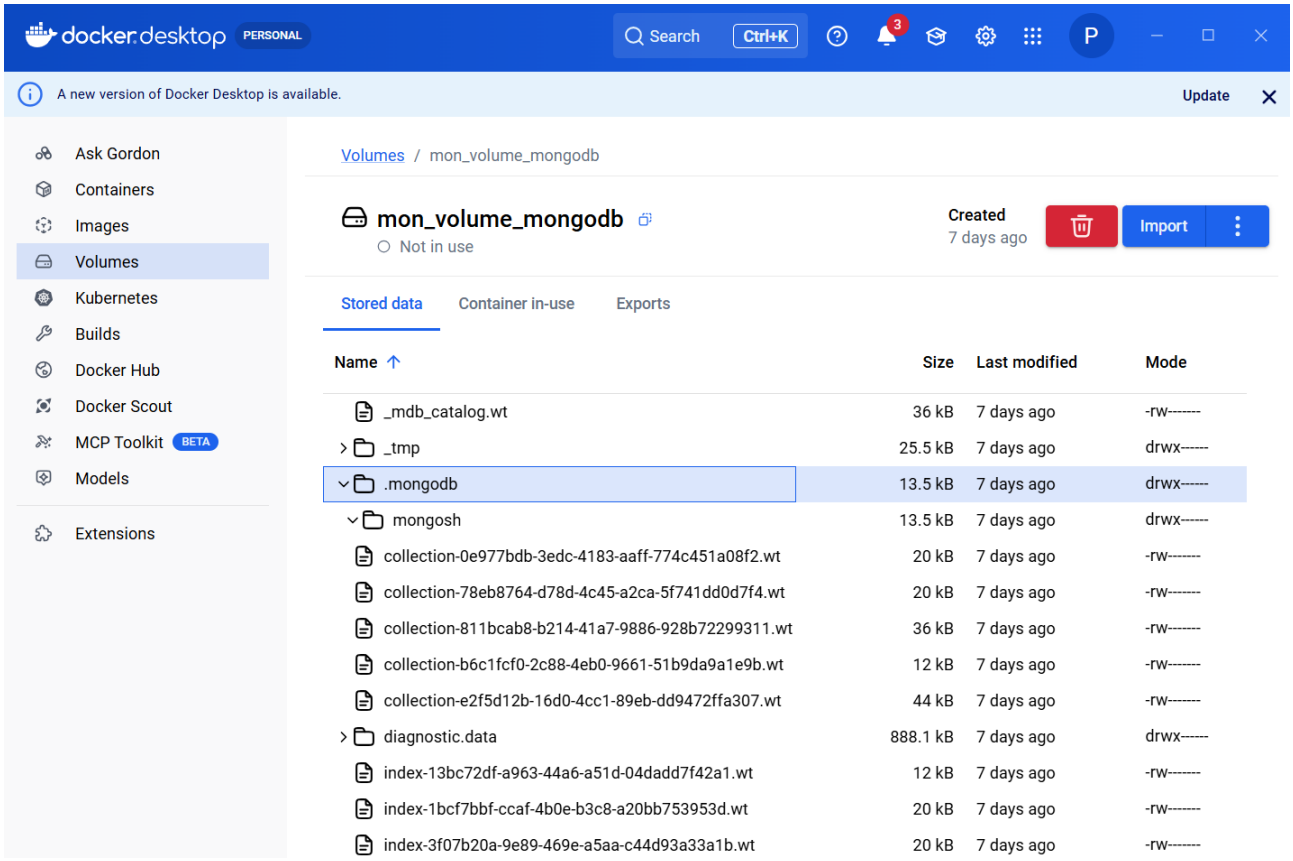
```
test> use("ma_bdd");
switched to db ma_bdd
ma_bdd> db.etudiants.insertOne({"nom":"hallyday", "prenom":"johnny"});
{
  acknowledged: true,
  insertedId: ObjectId('69a4d5d62bc0d675887c2907')
}
ma_bdd> db.etudiants.find();
[
  {
    _id: ObjectId('69a4c386e7ce1e99587c2907'),
    nom: 'smet',
    prenom: 'jp'
  },
  {
    _id: ObjectId('69a4d5d62bc0d675887c2907'),
    nom: 'hallyday',
    prenom: 'johnny'
  }
]
ma_bdd> exit
PS C:\Users\phbou>
```




Windows PowerShell
PS C:\Users\phbou> mongosh mongodb://sio:password1234@localhost:27017
Current Mongosh Log ID: 69a4d692b4d43e82a47c2906
Connecting to: mongodb://<credentials>@localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.7.0
Using MongoDB: 8.2.5
Using Mongosh: 2.7.0
For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/
The server generated these startup warnings when booting
2026-03-02T00:14:35.592+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2026-03-02T00:14:36.473+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
2026-03-02T00:14:36.474+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
2026-03-02T00:14:36.474+00:00: We suggest setting the contents of sysfsFile to 0.
2026-03-02T00:14:36.474+00:00: We suggest setting swappiness to 0 or 1, as swapping can cause performance problems.
test> use("ma_bdd");
switched to db ma_bdd
ma_bdd> db.etudiants.find();
[
{
_id: ObjectId('69a4c386e7ce1e99587c2907'),
nom: 'smet',
prenom: 'jp'
},
{
_id: ObjectId('69a4d5d62bc0d675887c2907'),
nom: 'hallyday',
prenom: 'johnny'
}
]
ma_bdd> exit
PS C:\Users\phbou>


sio@PC-01: ~
sio@PC-01:~$ docker rm -fv mongodb
mongodb
sio@PC-01:~$





11.1. Deuxième type de volume : montage lié (bind mounting)

Il s'agit du type de volume mis en place dans le paragraphe 10 mais cette fois-ci, le volume est accessible en écriture.

Les montages liés connectent un répertoire spécifique de la machine hôte à un répertoire d'un conteneur.

Si nous avons un dossier dans notre machine qui s'appelle /home/sio/ProjetB1 avec un fichier à l'intérieur dont le nom est index.html et que nous le mappons au dossier /var/www/html/web de notre conteneur Ubuntu (dans notre conteneur, le dossier web est vide initialement), le dossier web contiendra également le fichier index.html.

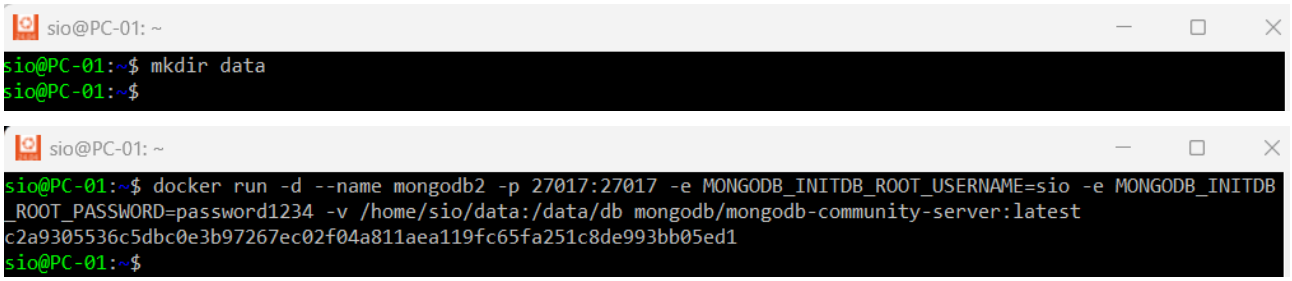
Si nous modifions le fichier index.html dans le conteneur, le contenu de ce fichier apparaîtra modifié dans le répertoire /home/sio/projetB1 de la machine locale.

Le contraire est également vrai. Si nous modifions ce fichier depuis le répertoire de la machine locale, il le sera également automatiquement dans le conteneur.

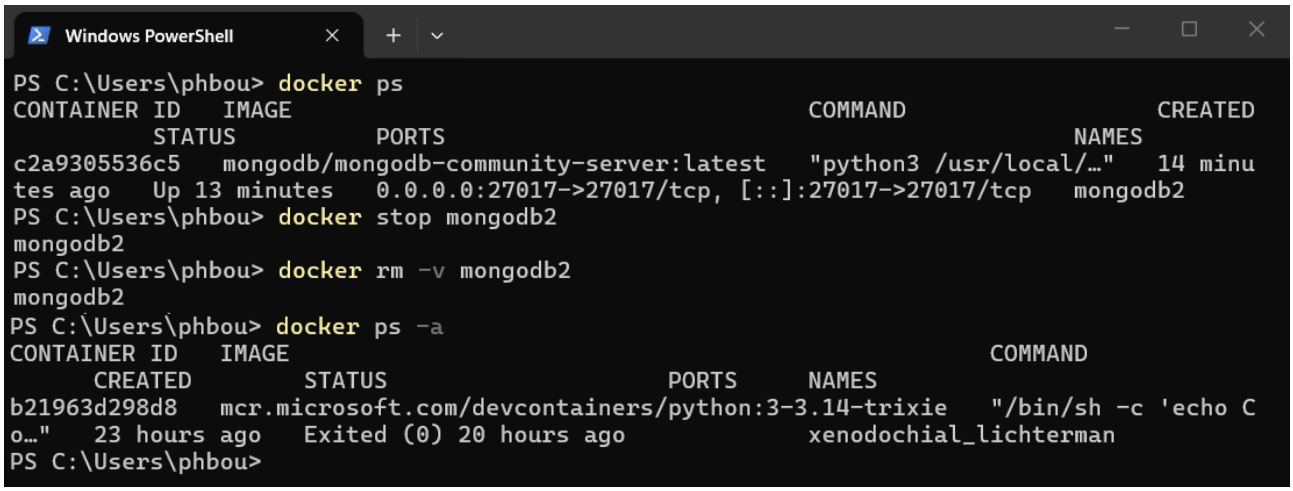
La particularité des volumes bind mounting est que le point de montage sur l'hôte est explicite plutôt que caché dans un répertoire de Docker comme dans le cas des volumes nommés.

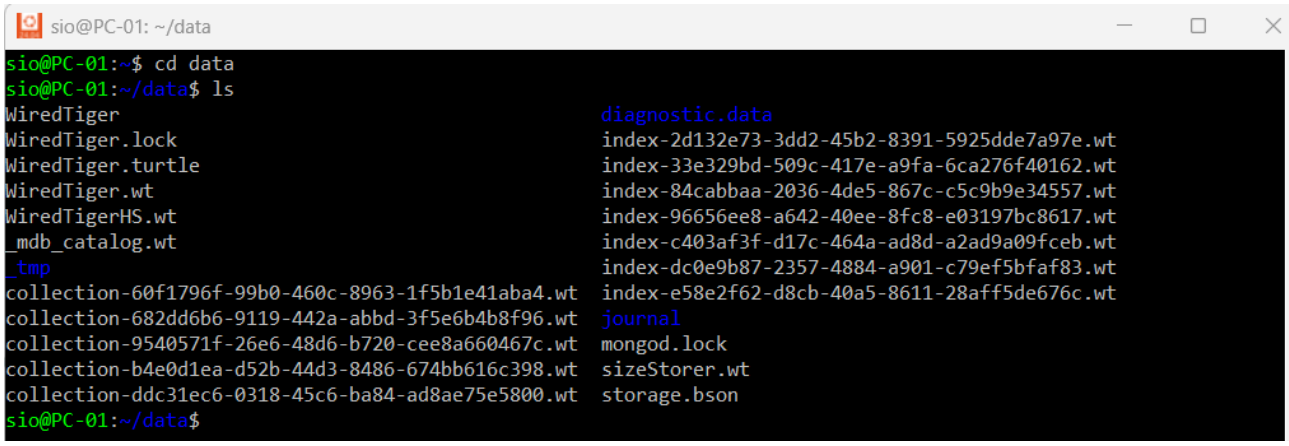
Cette **synchronisation en temps réel** est idéale pour les développeurs.

Les montages liés comportent cependant des risques car ils exposent les chemins d'accès hôtes aux conteneurs.

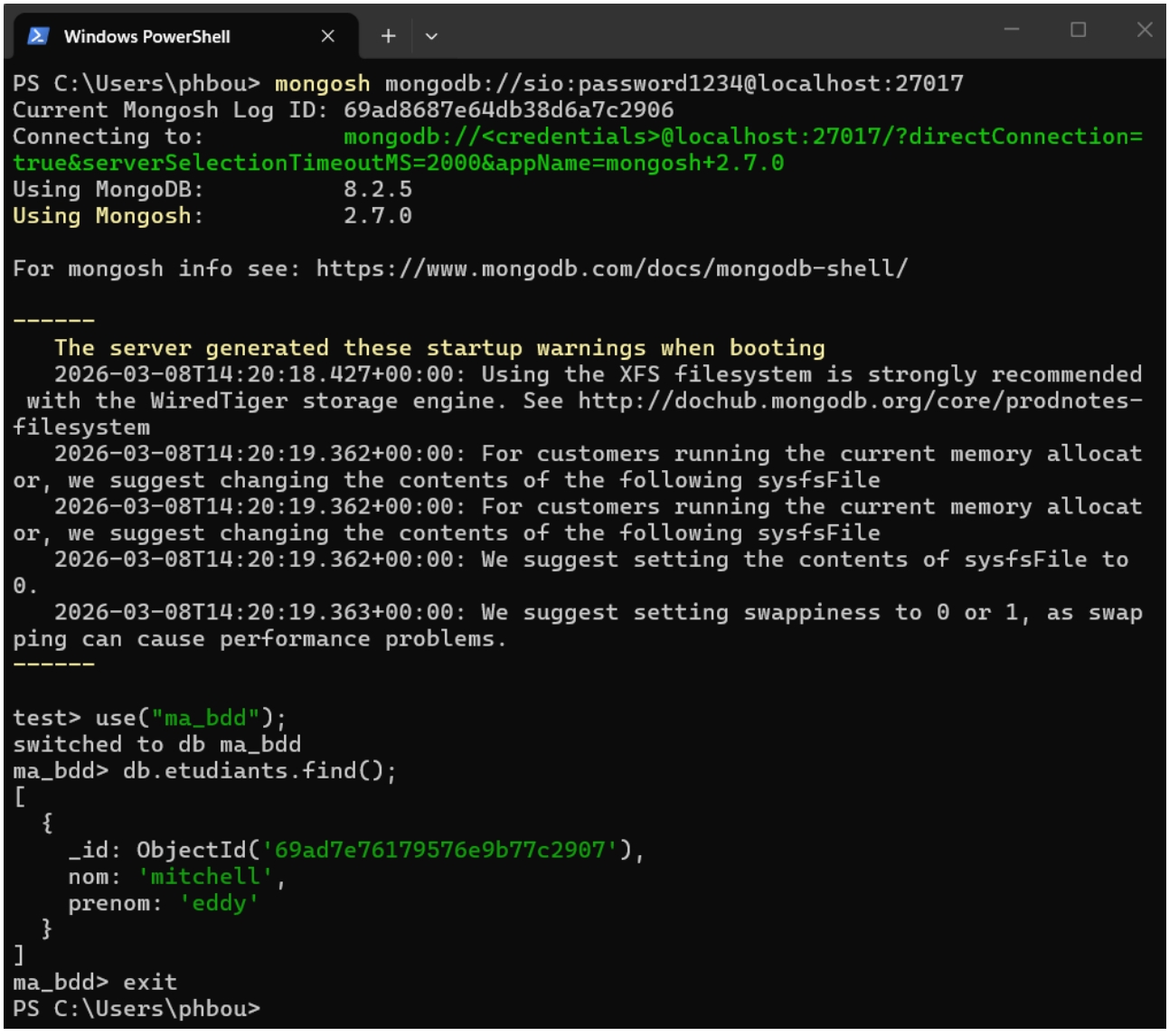



Windows PowerShell
PS C:\Users\phbou> mongosh mongodb://sio:password1234@localhost:27017
Current Mongosh Log ID: 69ad7d57179576e9b77c2906
Connecting to: mongodb://<credentials>@localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.7.0
Using MongoDB: 8.2.5
Using Mongosh: 2.7.0
For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/
The server generated these startup warnings when booting
2026-03-08T13:38:06.451+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2026-03-08T13:38:07.057+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
2026-03-08T13:38:07.057+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
2026-03-08T13:38:07.057+00:00: We suggest setting the contents of sysfsFile to 0.
2026-03-08T13:38:07.057+00:00: We suggest setting swappiness to 0 or 1, as swapping can cause performance problems.
test> use("ma_bdd");
switched to db ma_bdd
ma_bdd> db.etudiants.insertOne({"nom":"mitchell", "prenom":"eddy"});
{
acknowledged: true,
insertedId: ObjectId('69ad7e76179576e9b77c2907')
}
ma_bdd> db.etudiants.find();
[
{
_id: ObjectId('69ad7e76179576e9b77c2907'),
nom: 'mitchell',
prenom: 'eddy'
}
]
ma_bdd> exit
PS C:\Users\phbou>

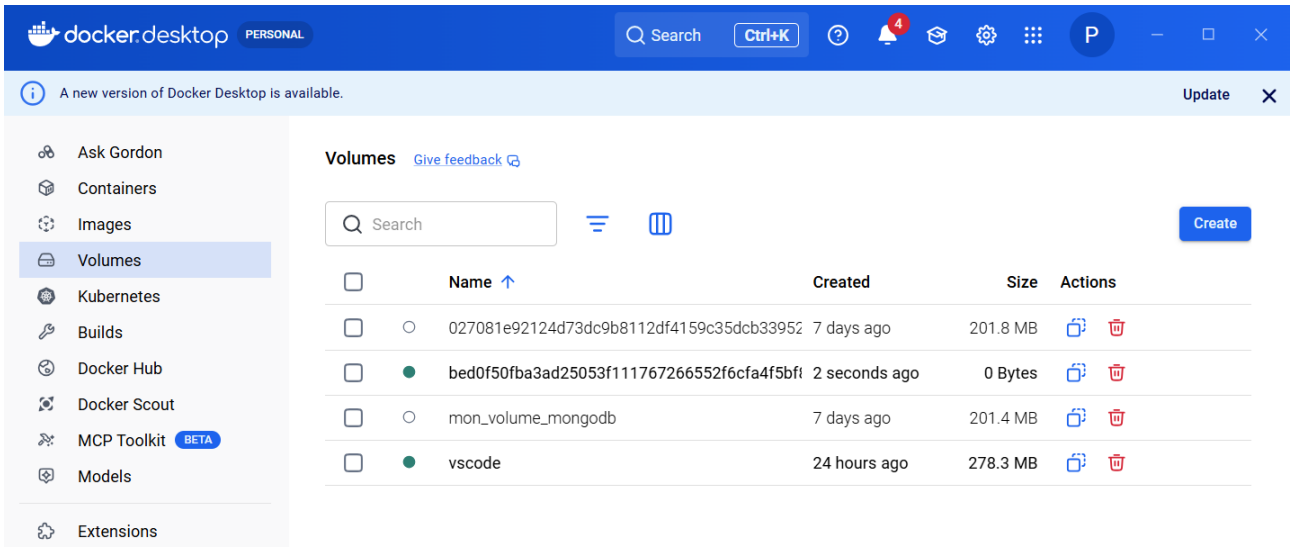

Windows PowerShell
PS C:\Users\phbou> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
c2a9305536c5 mongodb/mongodb-community-server:latest "python3 /usr/local/…" 14 minutes ago Up 13 minutes 0.0.0.0:27017->27017/tcp, [::]:27017->27017/tcp mongodb2
PS C:\Users\phbou> docker stop mongodb2
mongodb2
PS C:\Users\phbou> docker rm -v mongodb2
mongodb2
PS C:\Users\phbou> docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
b21963d298d8 mcr.microsoft.com/devcontainers/python:3-3.14-trixie "/bin/sh -c 'echo Co…" 23 hours ago Exited (0) 20 hours ago xenodochial_lichterman
PS C:\Users\phbou>

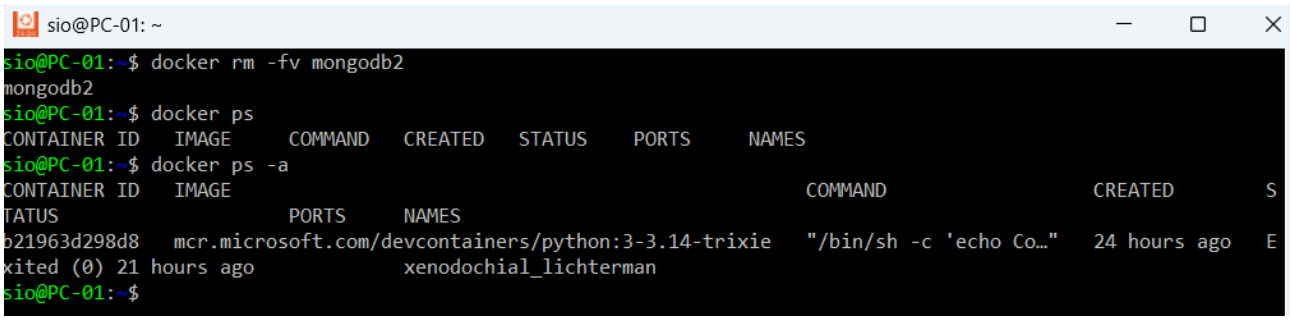


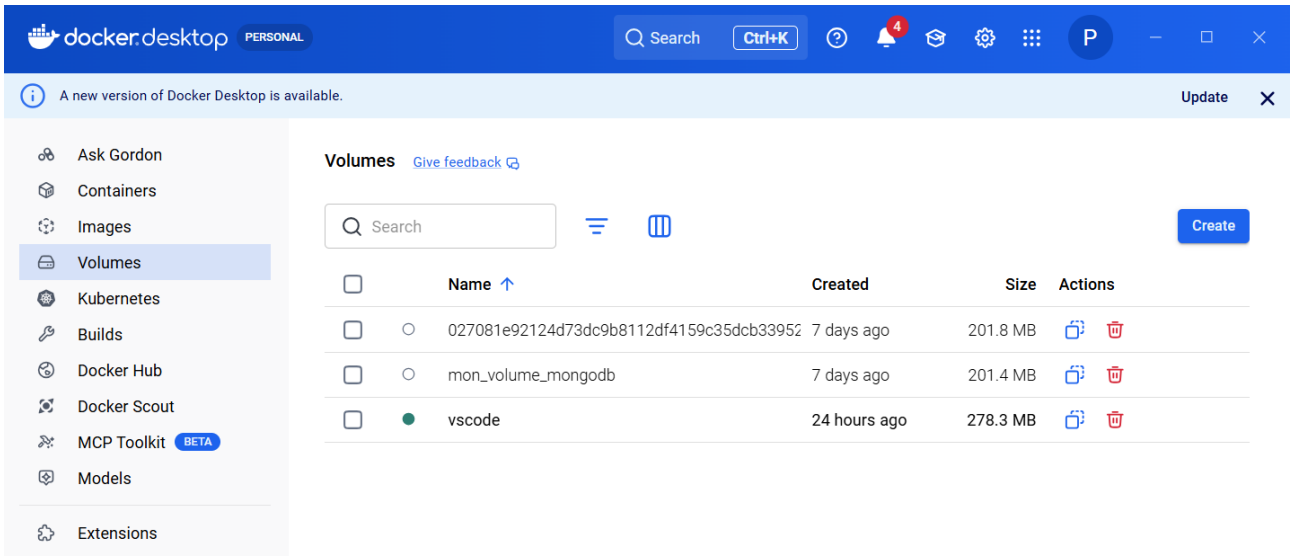
Après suppression du conteneur, on relance une nouvelle instance de l'image :



La persistance a bien été gérée.





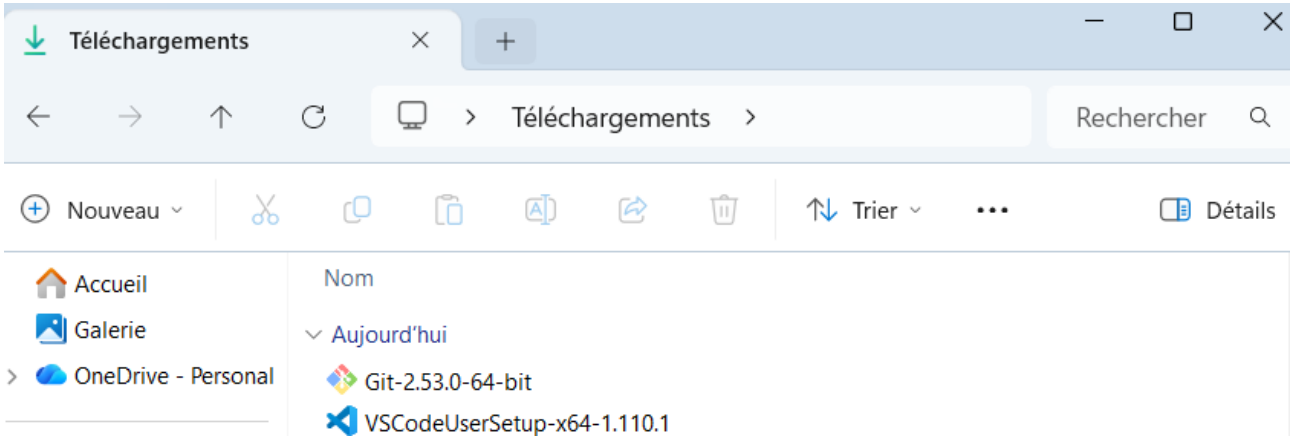


12. Utiliser un conteneur Docker et Visual Studio Code.

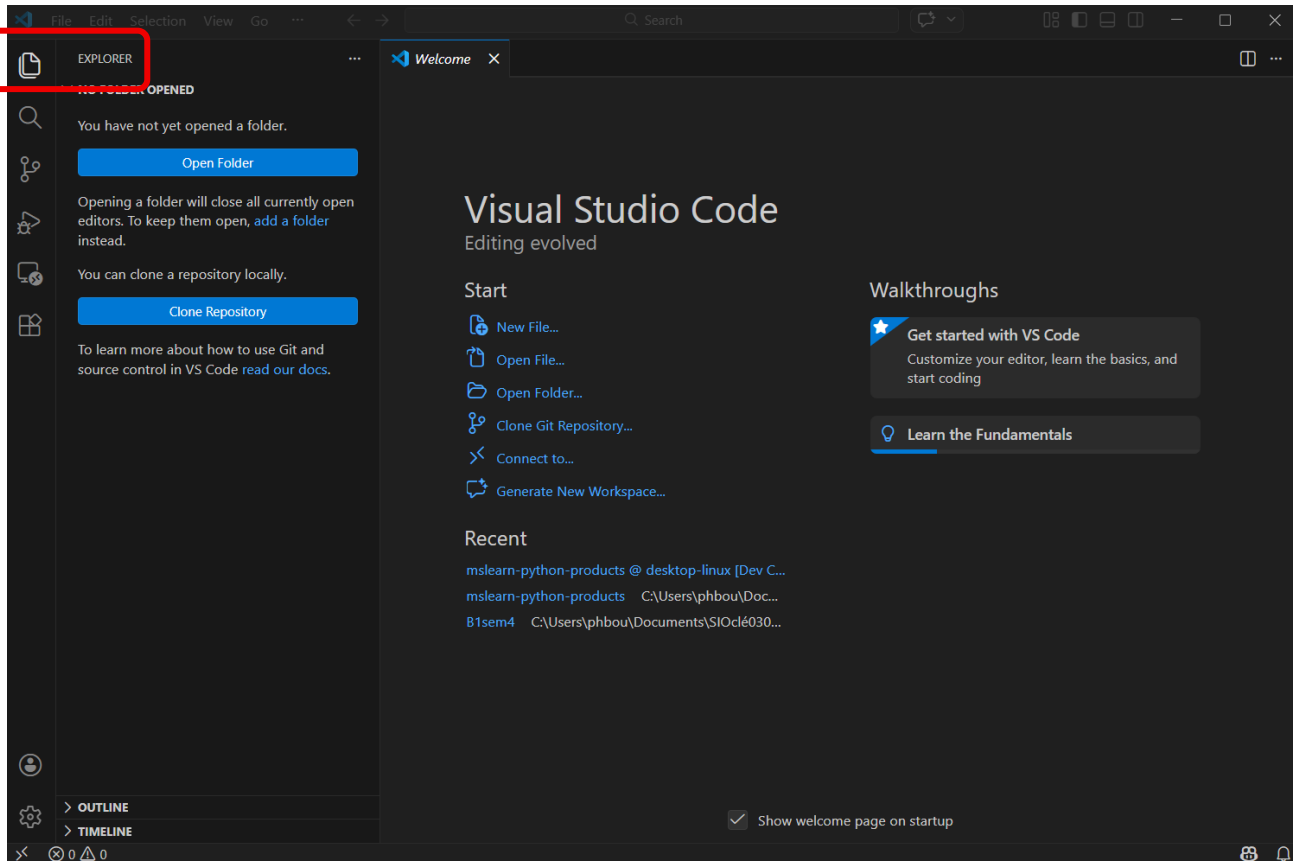
➔ L'objectif est d'utiliser **l'extension Dev Containers** de Visual Studio Code pour développer au sein d'un conteneur Docker car les développeurs doivent pouvoir travailler sur n'importe quel projet sans avoir à installer ou configurer leurs machines au préalable.

En l'occurrence, vous travaillez actuellement avec un projet Python qui doit afficher un tableau de produits.

- Installez Visual Studio Code si ce n'est pas déjà fait :

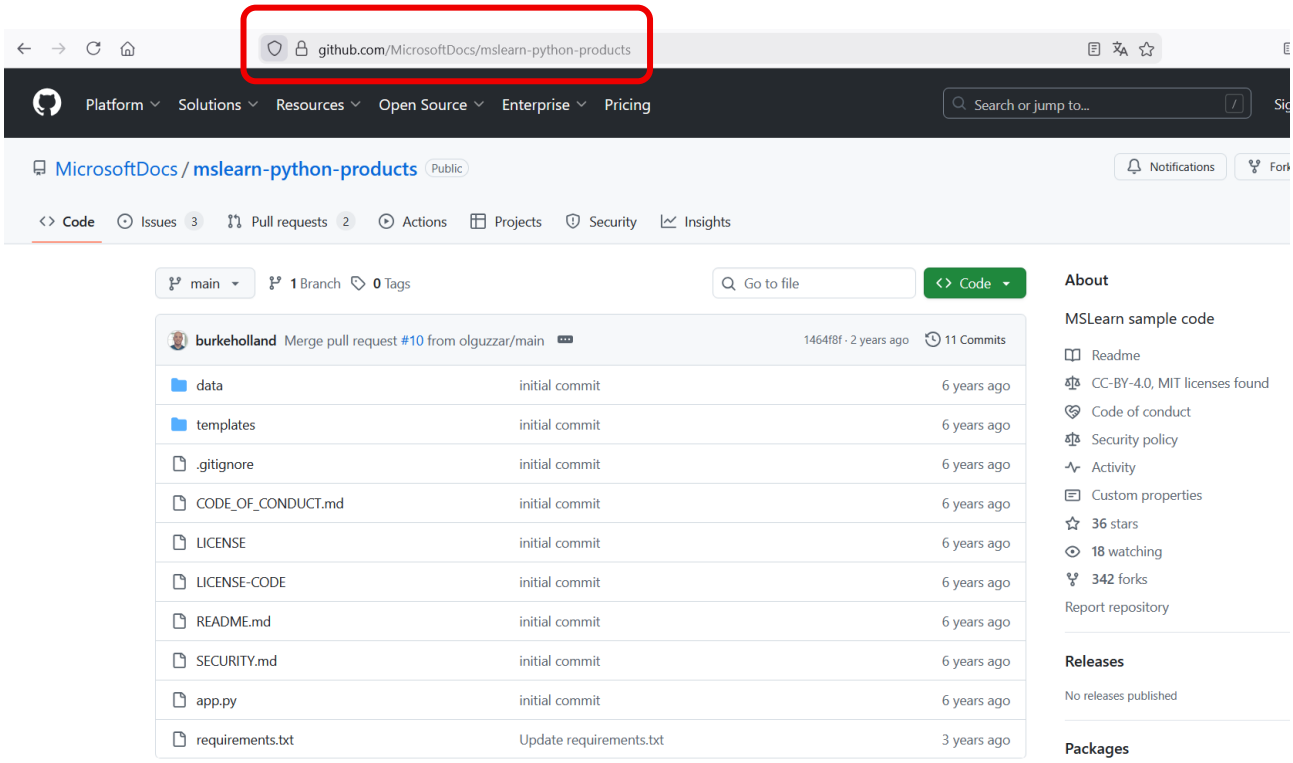


- Vérifiez que Docker Desktop soit bien en cours d'exécution sur votre ordinateur.
- Ouvrez une nouvelle instance de Visual Studio Code :

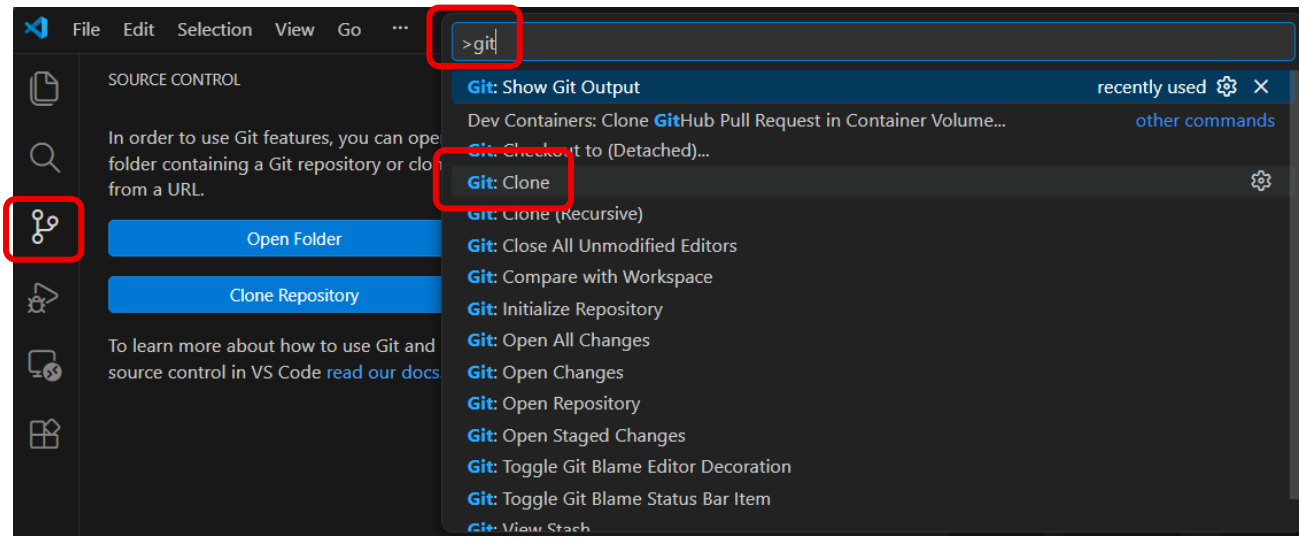


- Copiez l'URL du projet **mslearn-python-products** qui nous servira d'exemple :

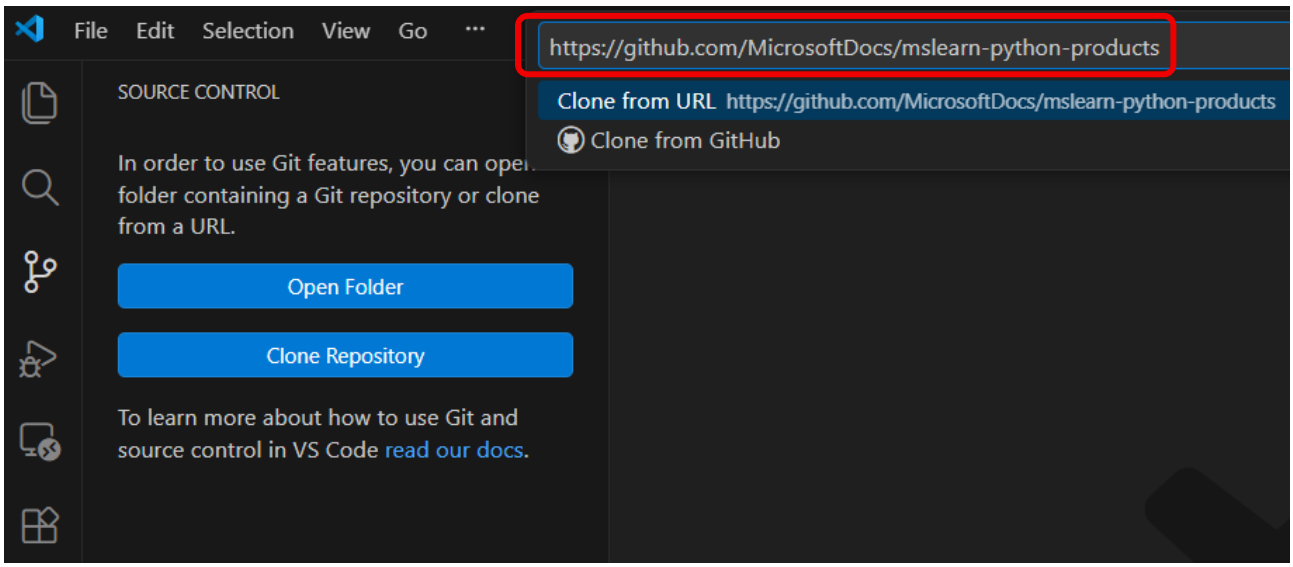
https://github.com/MicrosoftDocs/mslearn-python-products



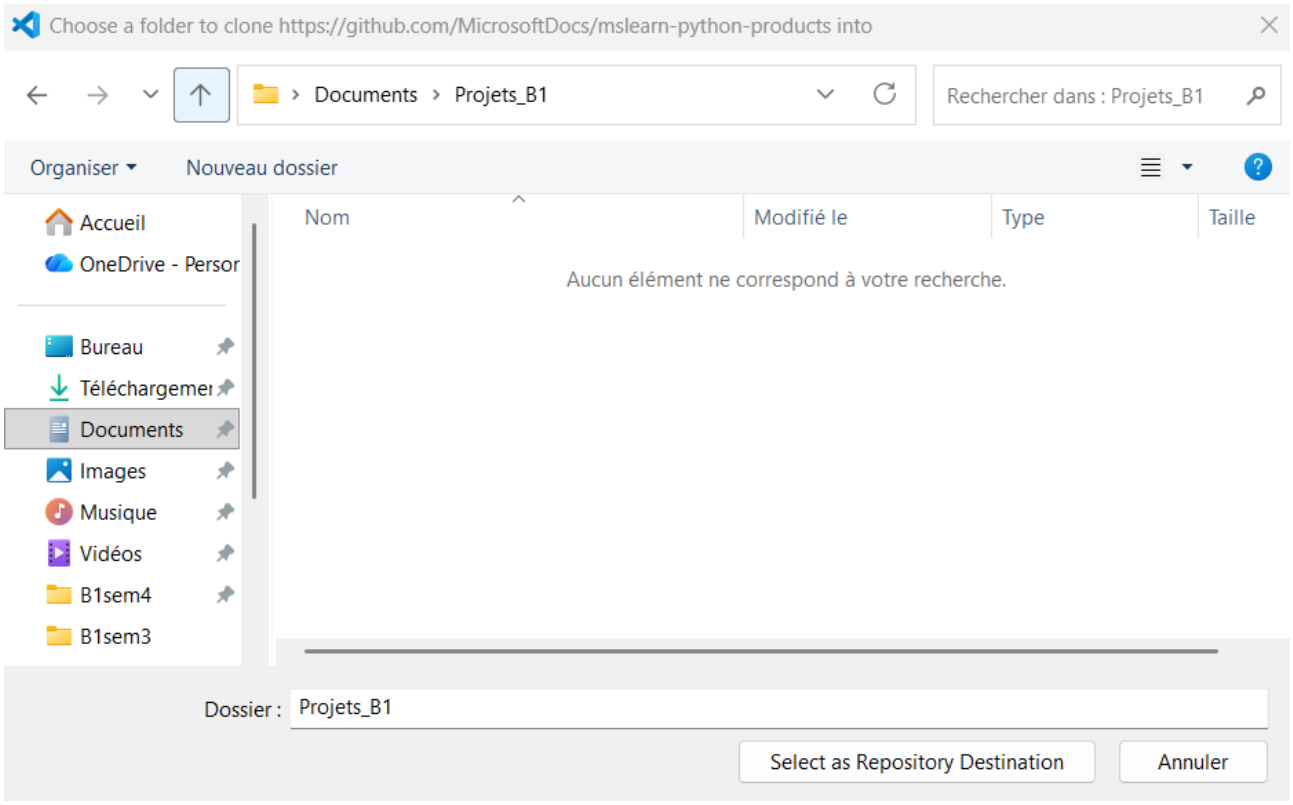
- Dans VS Code, sélectionnez **Clone Repository** ou appuyez sur F1 pour rechercher **Git : Clone :**

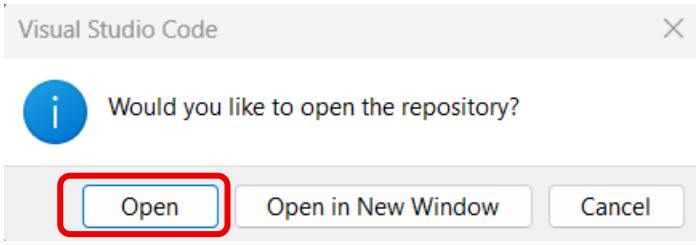


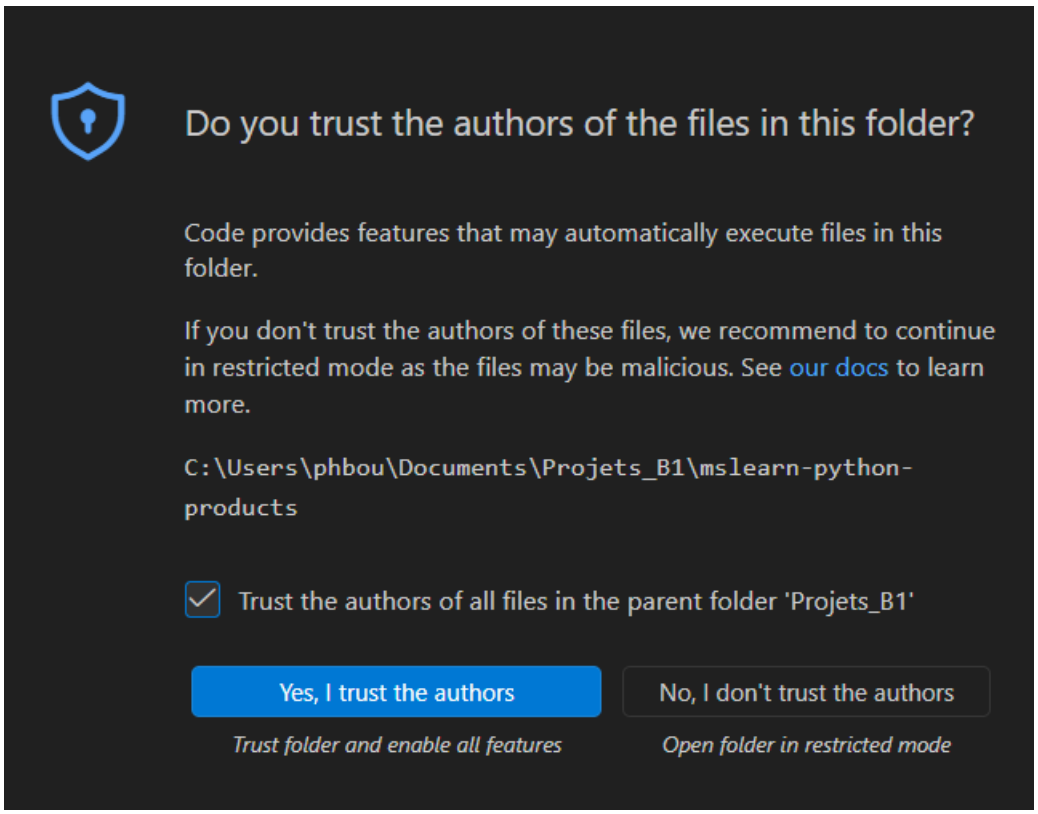
- Collez l'URL :



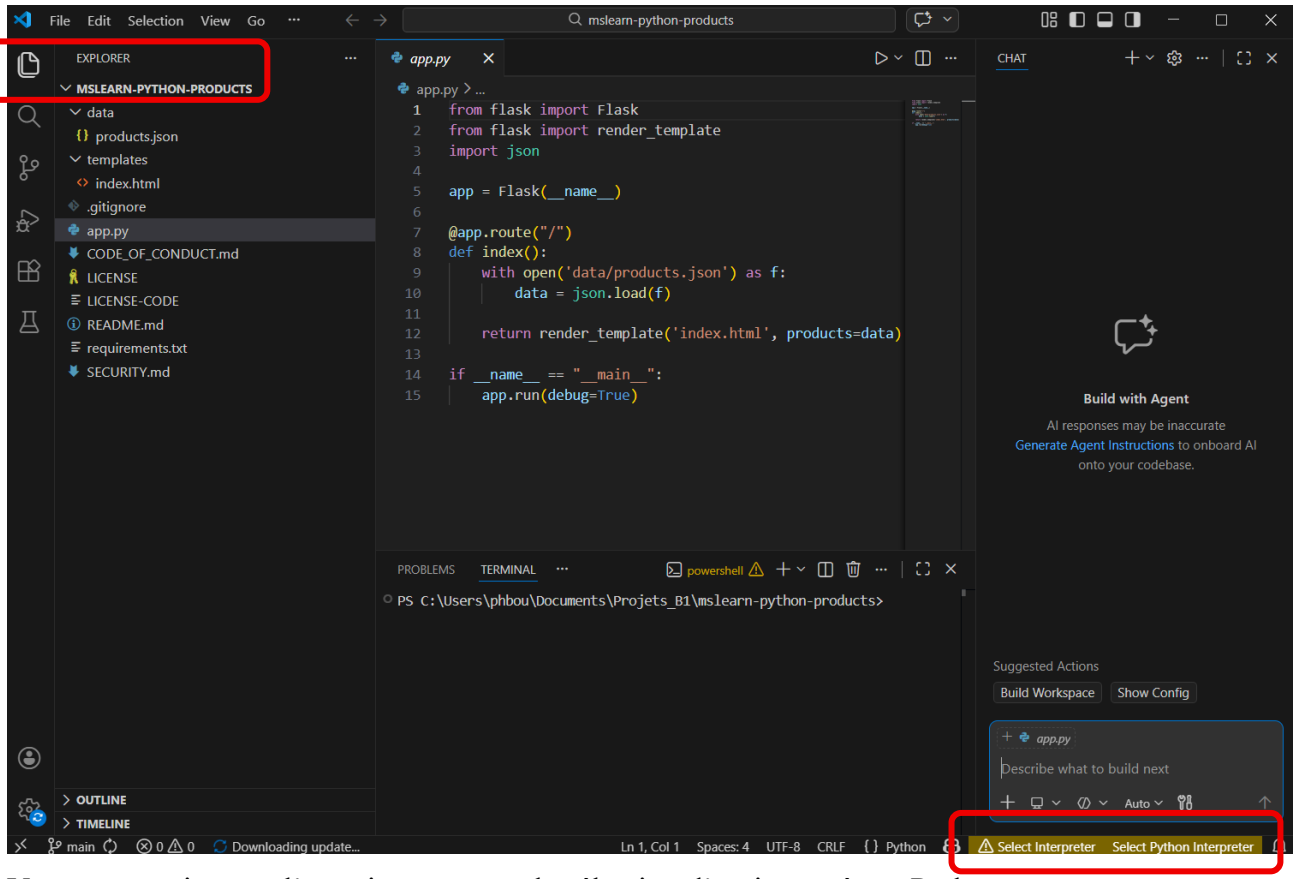
- Sélectionnez un emplacement sur votre disque où le projet va être cloné



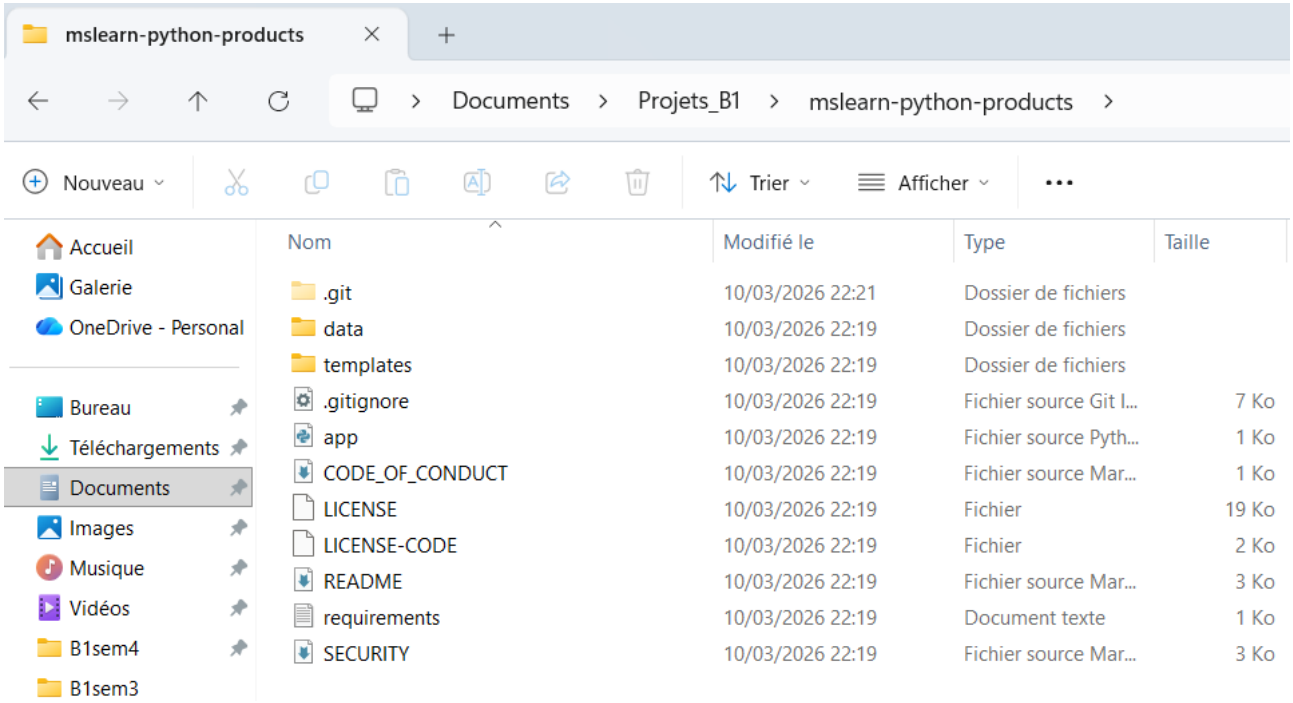




Le projet est maintenant ouvert dans Visual Studio Code :



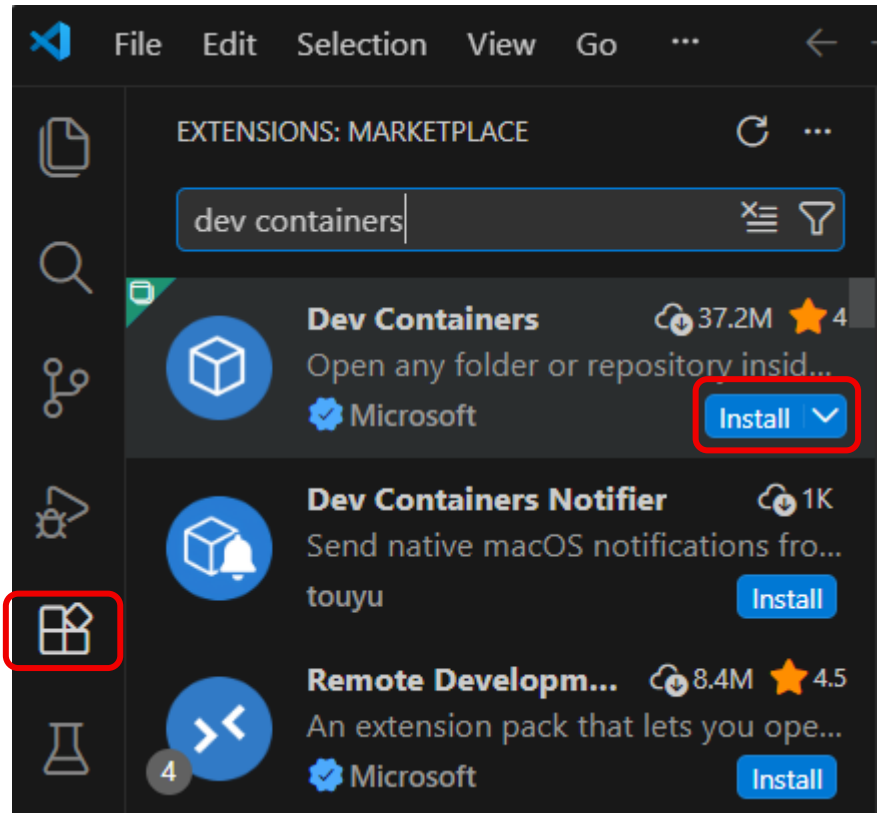
Vous pouvez ignorer l'avertissement sur la sélection d'un interpréteur Python.

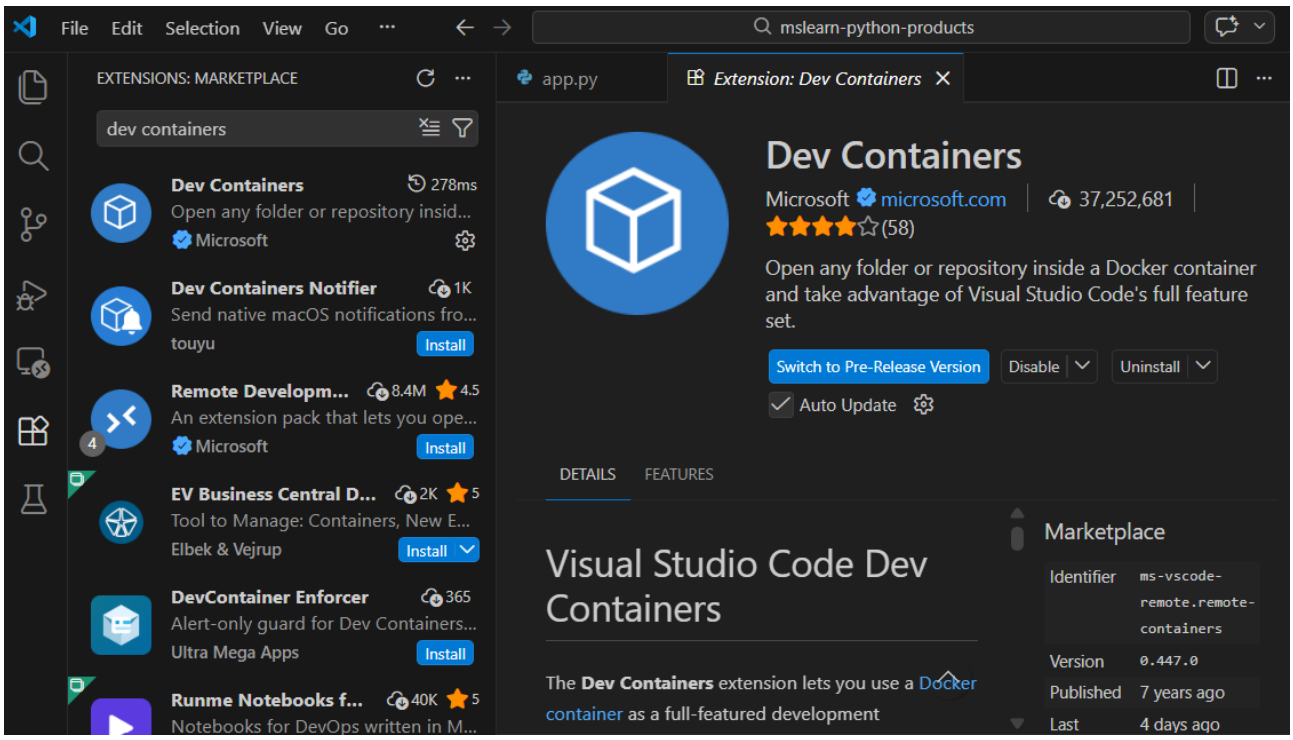


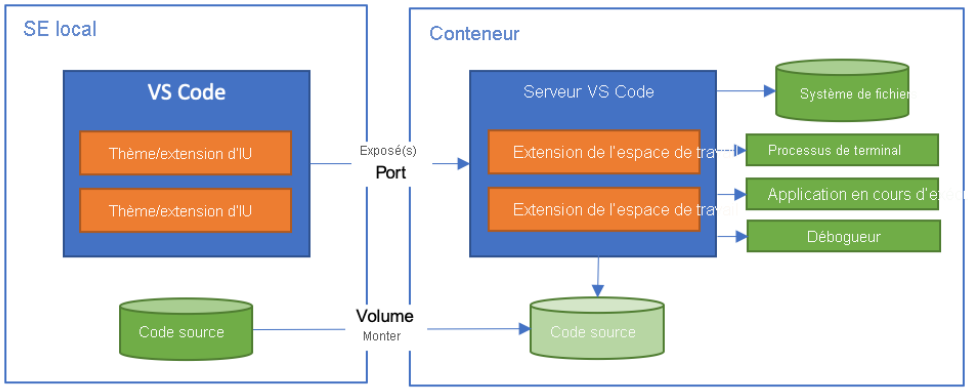
- Sélectionnez l'icône Extensions afin d'installer l'extension Dev Containers (« conteneurs de développement »).

➔ L'extension Dev Containers vous permettra d'exécuter Visual Studio Code à l'intérieur d'un conteneur Docker.

➔ Vous bénéficierez des commandes Dev Containers dans VS Code.

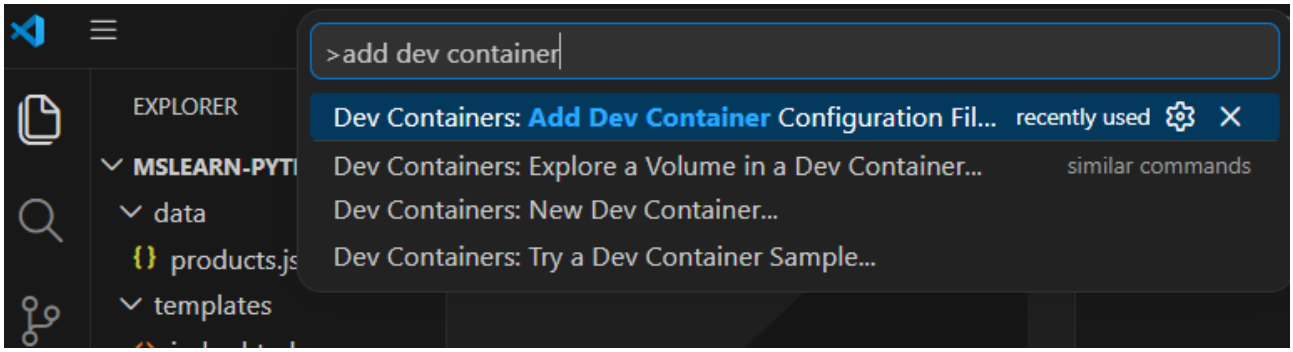






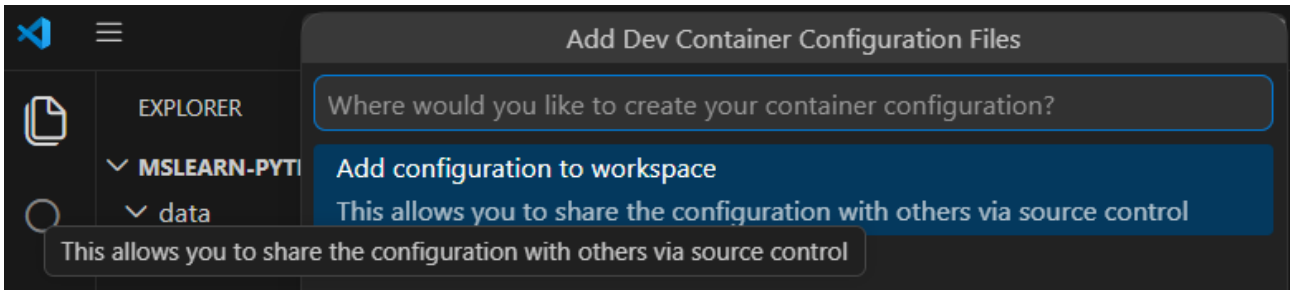
➔ Vous allez ajouter un conteneur de développement à un projet Python et l'exécuter sur votre propre ordinateur, même si Python n'est pas installé.

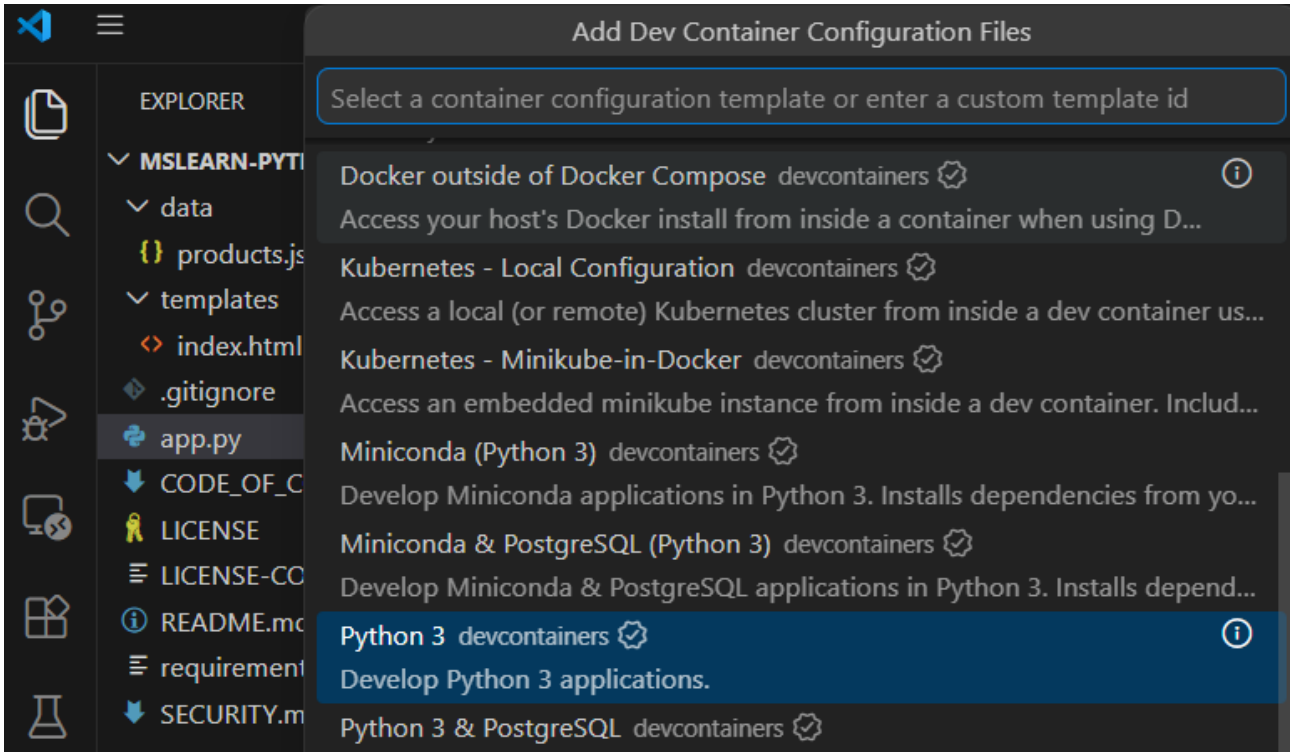
- Appuyez sur F1 pour ouvrir la palette de commandes
- Tapez **add dev container** et sélectionnez **Dev Containers: Add Dev Container Configuration Files**.

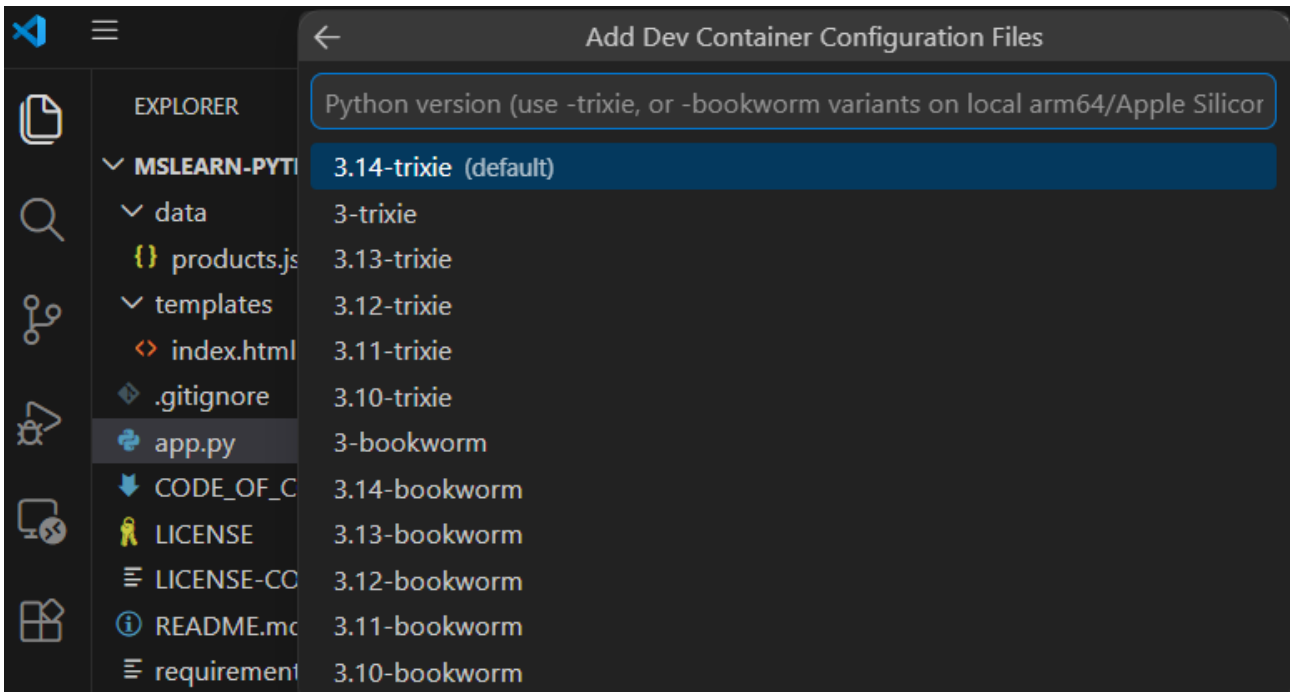


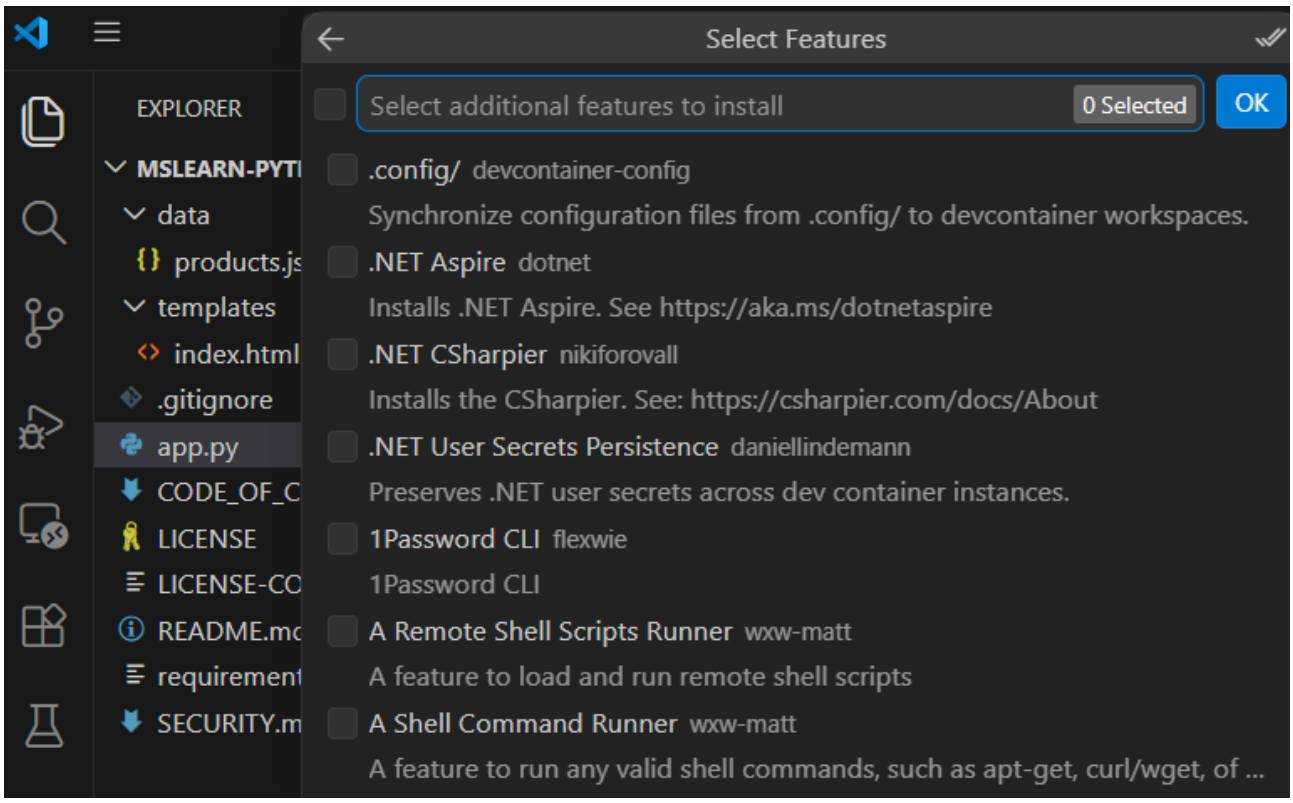
Ceci présente la liste de Modèles de conteneur de développement. Les modèles contiennent les fichiers sources nécessaires pour configurer un environnement de développement complet pour la pile technique spécifiée.

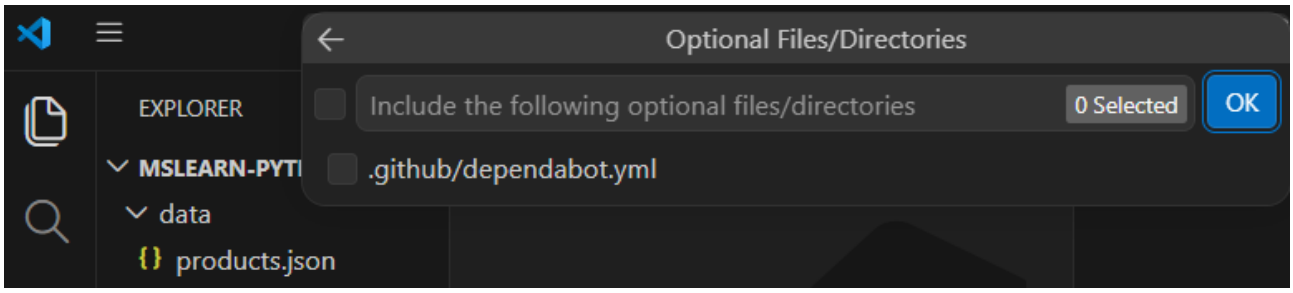
- Sélectionnez **Add configuration to workspace**.



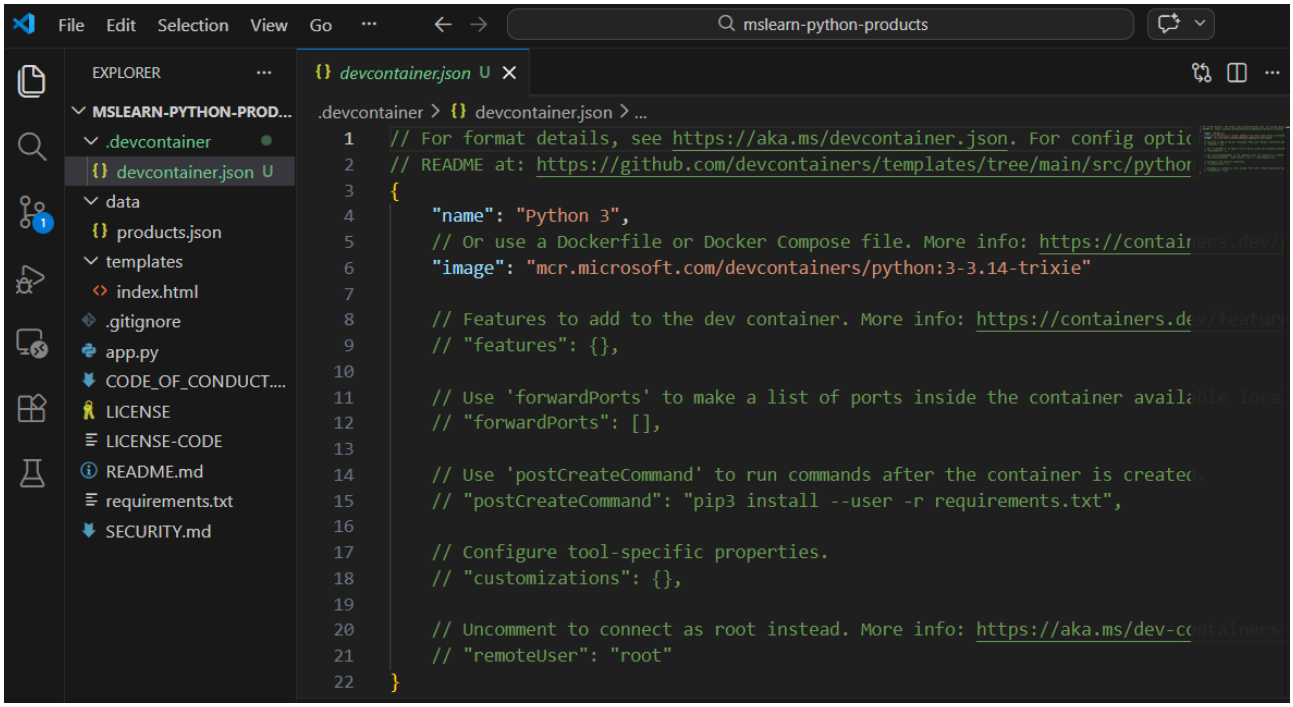






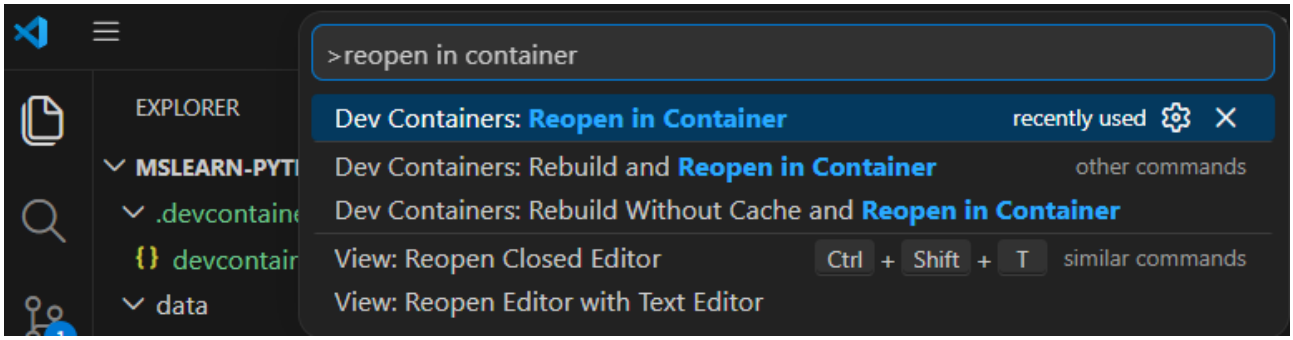


- Un nouveau dossier nommé **« .devcontainer »** a été ajouté au projet. Développez ce dossier : il contient un fichier **devcontainer.json**.



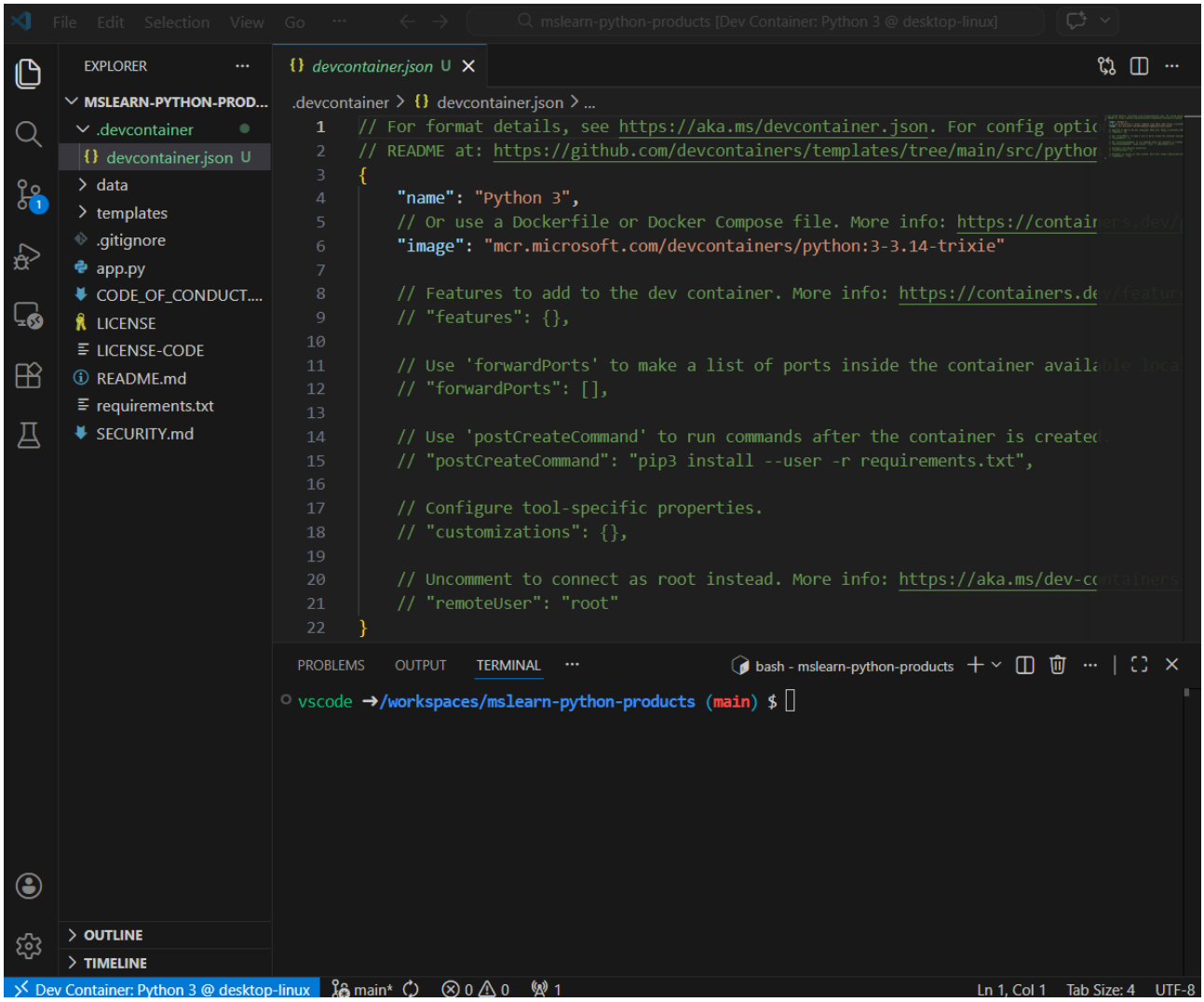
➔ Ouvrir le projet dans un conteneur

- Appuyez sur F1 pour ouvrir la palette de commandes, tapez **reopen in container** et sélectionnez **Dev Containers: Reopen in Container** dans la liste des options disponibles.



Le conteneur commence la génération. La génération initiale peut prendre quelques minutes, car une nouvelle image doit être extraite et générée sur votre machine.

- Examinez l'indicateur distant en bas à gauche de VS Code : il affiche à présent « Dev Container: Python 3 ».



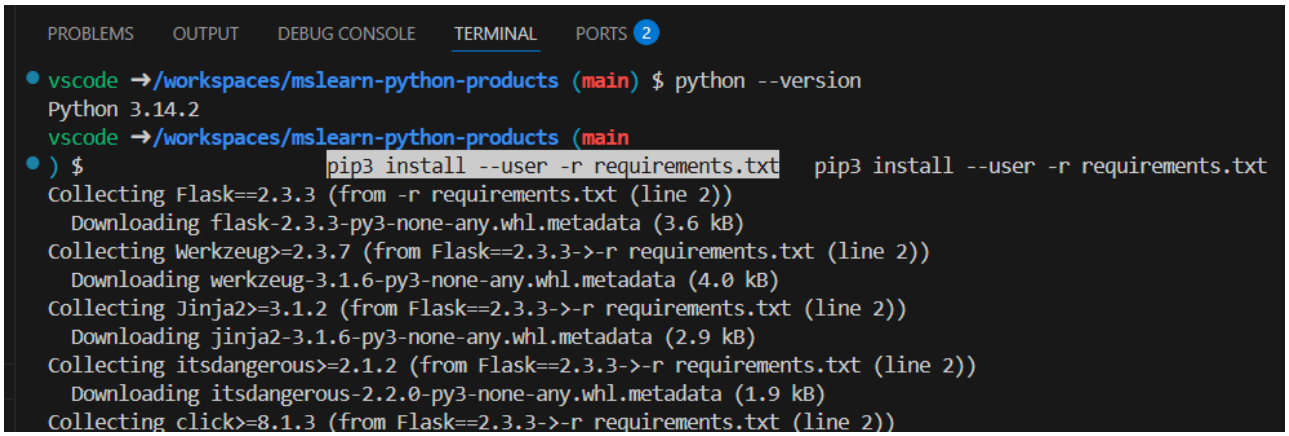
➔ Inspecter le conteneur

- Ouvrez le terminal intégré dans Visual Studio Code.
- Exécutez la commande suivante pour vérifier que Python est installé :

python –version

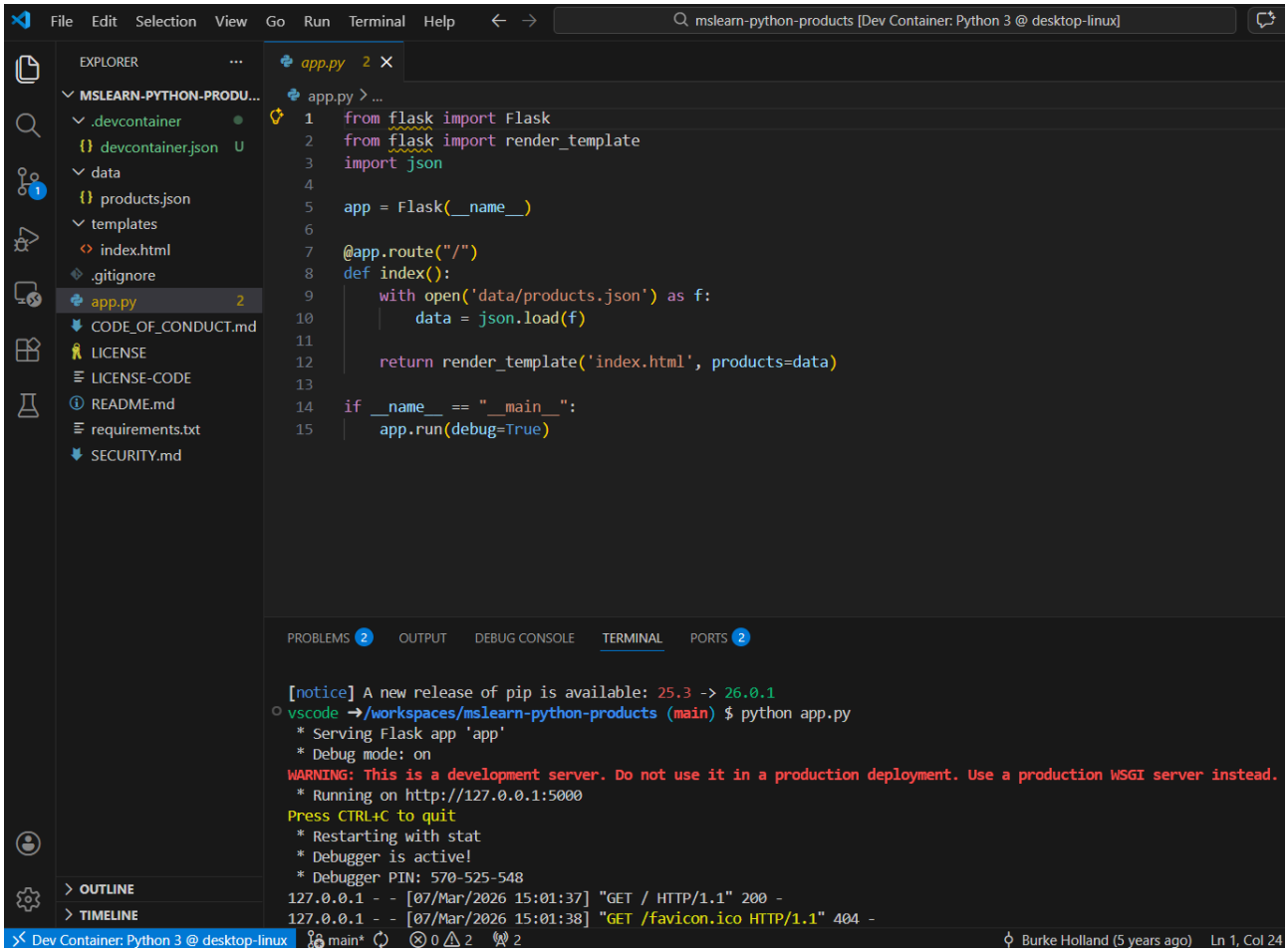
- Exécutez la commande suivante dans le terminal pour installer les dépendances Flask nécessaires à l'exécution du projet :

  pip3 install --user -r requirements.txt

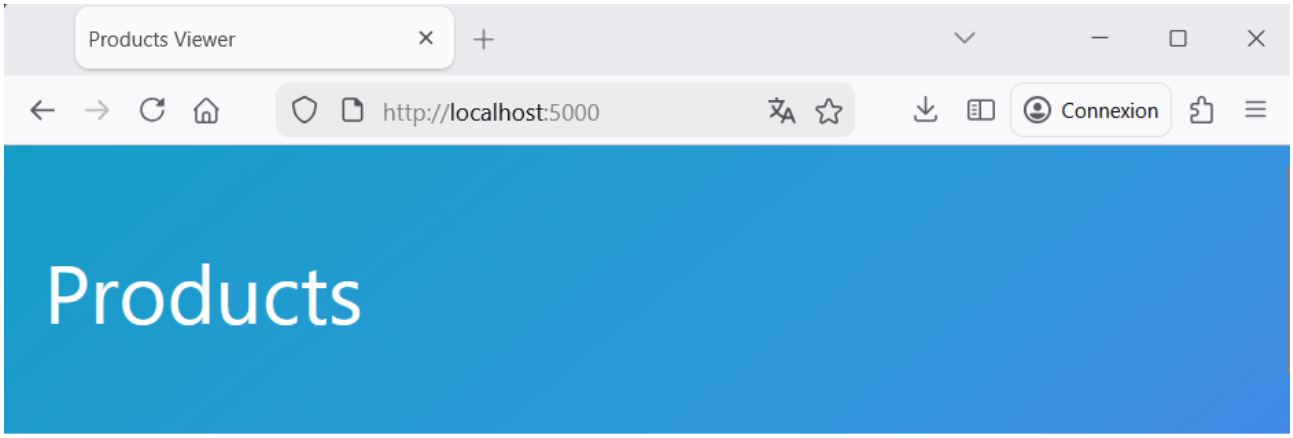


- Entrez la commande suivante dans le terminal pour exécuter le projet :

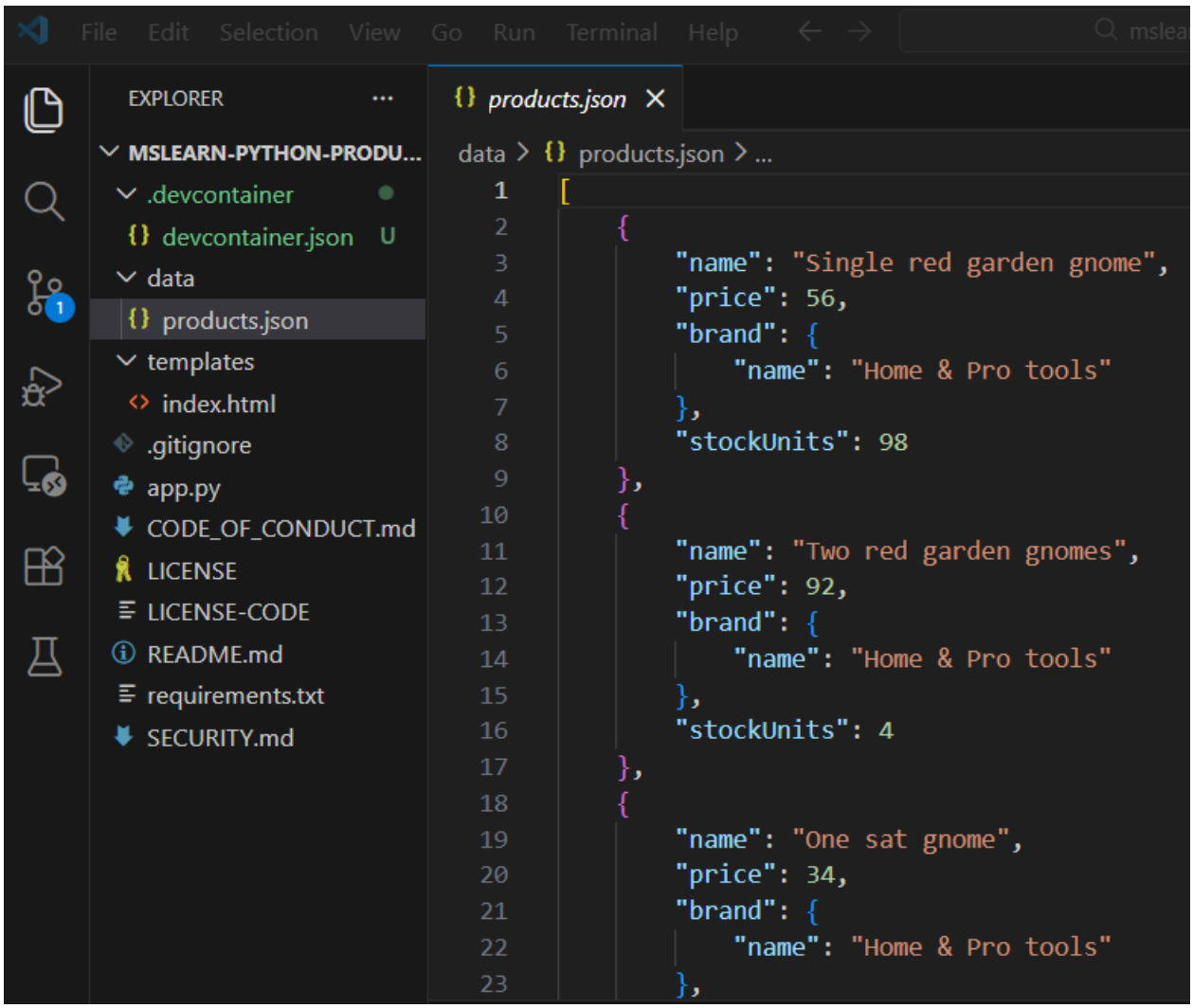
  python app.py

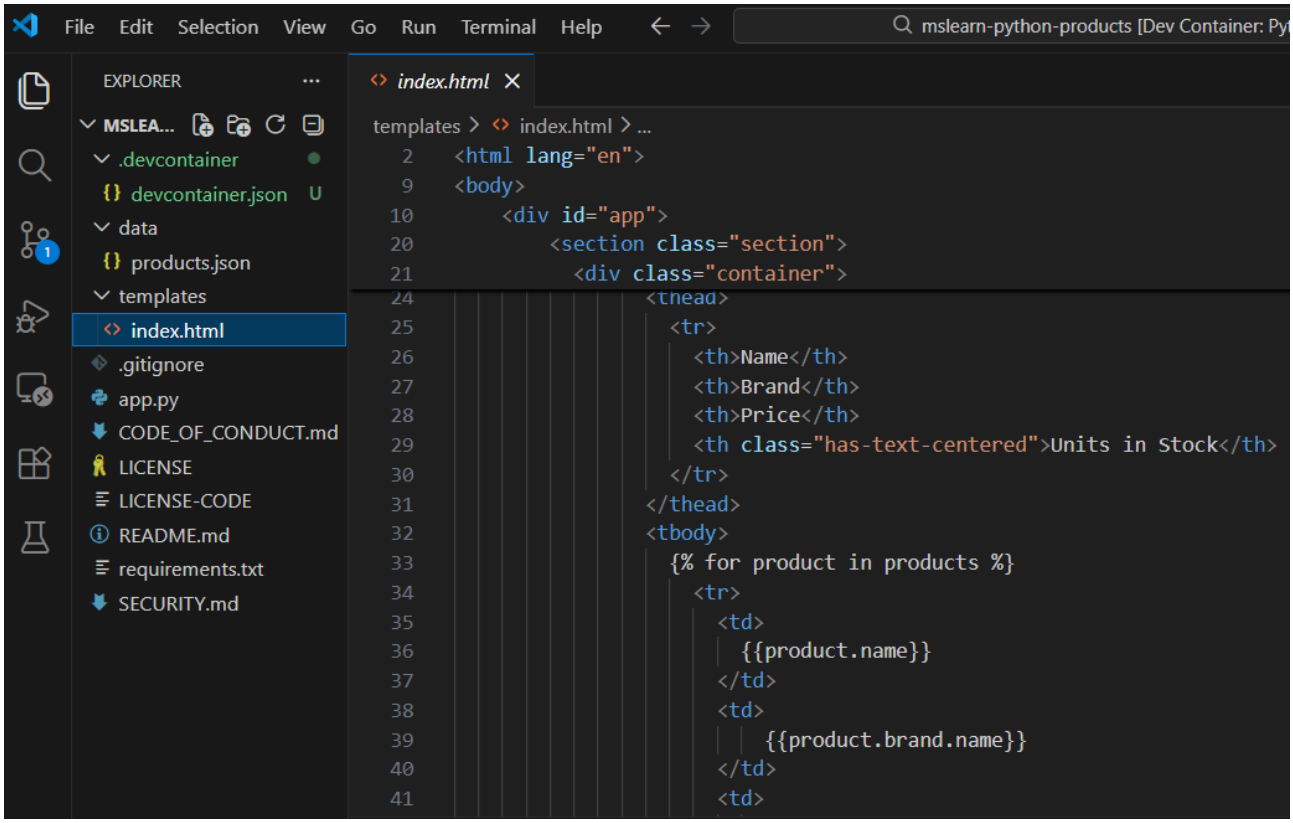


- Montrez que application web Python avec Flask s'est bien exécutée sur votre machine en saisissant http://127.0.0.1:5000 depuis votre navigateur :



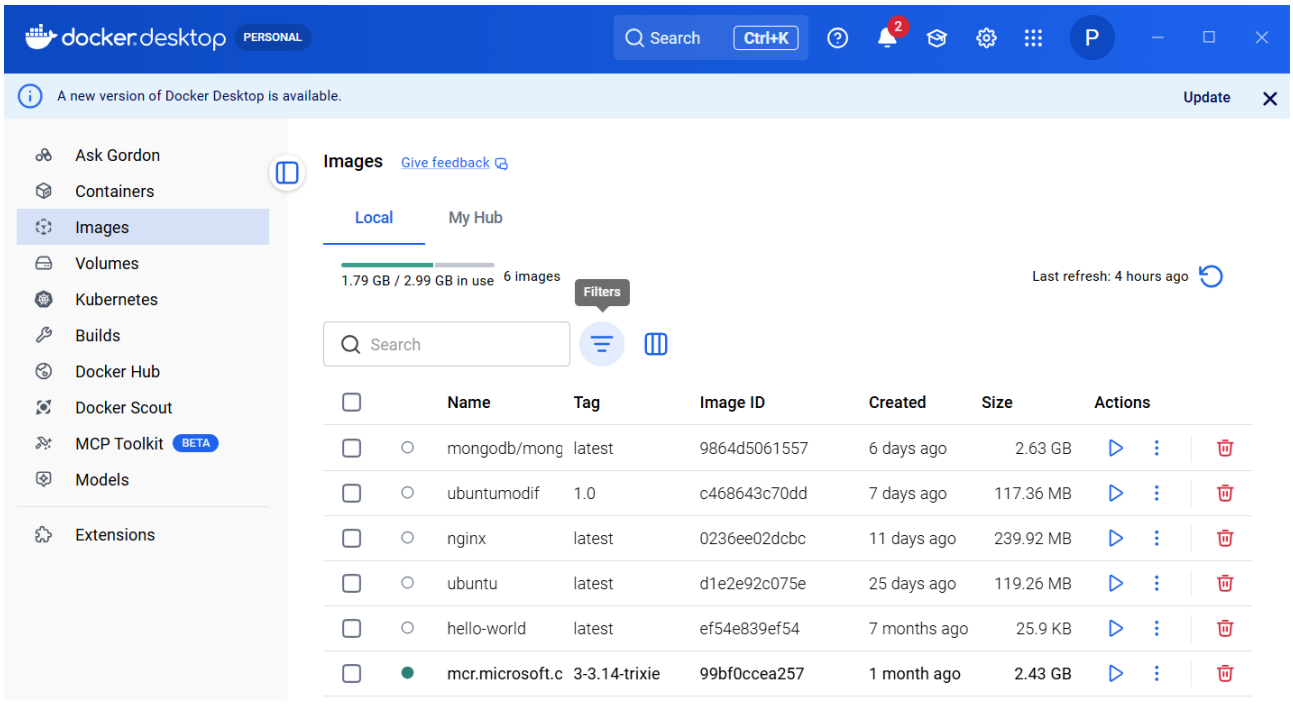
Name	Brand	Price	Units in Stock
Single red garden gnome	Home & Pro tools	56	98
Two red garden gnomes	Home & Pro tools	92	4
One sat gnome	Home & Pro tools	34	34
One sat on shoe gnome	Home & Pro tools	54	54
One barrow gnome	Home & Pro tools	29	23
One glasses gnome	Home & Pro tools	54	94
One smiling gnome	Home & Pro tools	43	45

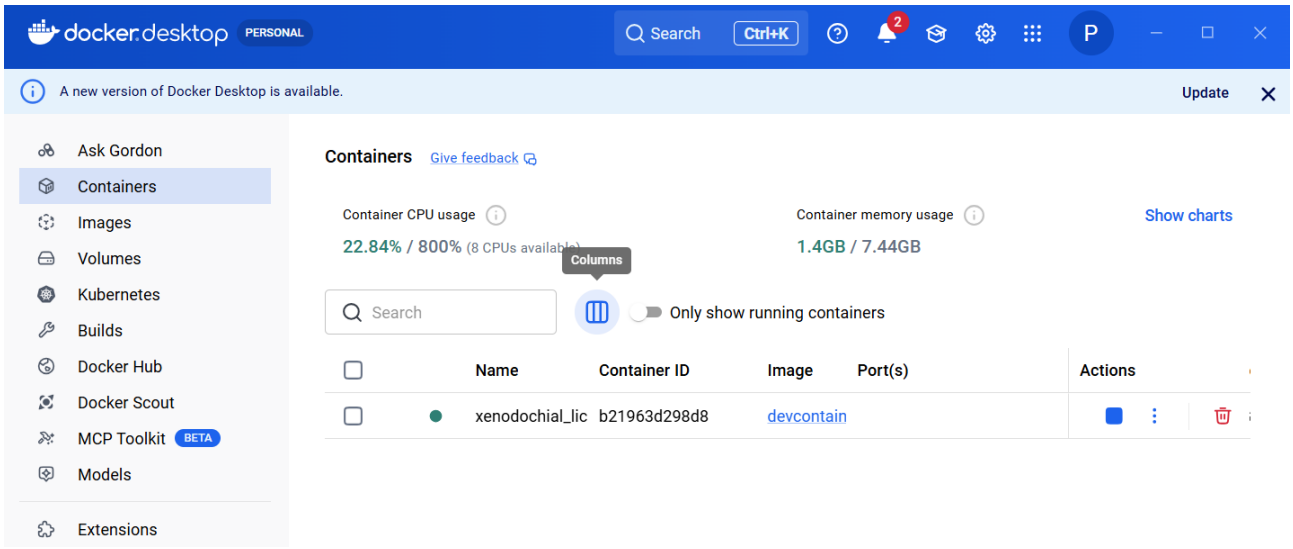

File Edit Selection View Go Run Terminal Help
EXPLORER
MSLEARN-PYTHON-PRODU...
.devcontainer
devcontainer.json U
data
products.json
templates
index.html
.gitignore
app.py
CODE_OF_CONDUCT.md
LICENSE
LICENSE-CODE
README.md
requirements.txt
SECURITY.md
products.json
data > products.json > ...
[
{
"name": "Single red garden gnome",
"price": 56,
"brand": {
"name": "Home & Pro tools"
},
"stockUnits": 98
},
{
"name": "Two red garden gnomes",
"price": 92,
"brand": {
"name": "Home & Pro tools"
},
"stockUnits": 4
},
{
"name": "One sat gnome",
"price": 34,
"brand": {
"name": "Home & Pro tools"
},


File Edit Selection View Go Run Terminal Help
mslearn-python-products [Dev Container: Py
EXPLORER
MSLEA...
.devcontainer
devcontainer.json U
data
products.json
templates
index.html
.gitignore
app.py
CODE_OF_CONDUCT.md
LICENSE
LICENSE-CODE
README.md
requirements.txt
SECURITY.md
index.html
templates > index.html > ...
<html lang="en">
<body>

<section class="section">

<thead>
<tr>
<th>Name</th>
<th>Brand</th>
<th>Price</th>
<th class="has-text-centered">Units in Stock</th>
</tr>
</thead>
<tbody>
{% for product in products %}
<tr>
<td>
{{product.name}}
</td>
<td>
{{product.brand.name}}
</td>
<td>


docker desktop
PERSONAL
Search
Ctrl+K
A new version of Docker Desktop is available.
Update
Ask Gordon
Containers
Images
Volumes
Kubernetes
Builds
Docker Hub
Docker Scout
MCP Toolkit
BETA
Models
Extensions
Images
Give feedback
Local
My Hub
1.79 GB / 2.99 GB in use
6 images
Last refresh: 4 hours ago
Filters
Search
Name
Tag
Image ID
Created
Size
Actions
mongodb/mong latest 9864d5061557 6 days ago 2.63 GB
ubuntumodif 1.0 c468643c70dd 7 days ago 117.36 MB
nginx latest 0236ee02dcbc 11 days ago 239.92 MB
ubuntu latest d1e2e92c075e 25 days ago 119.26 MB
hello-world latest ef54e839ef54 7 months ago 25.9 KB
mcr.microsoft.c 3-3.14-trixie 99bf0ccea257 1 month ago 2.43 GB


docker desktop
PERSONAL
Search
Ctrl+K
A new version of Docker Desktop is available.
Update
Ask Gordon
Containers
Images
Volumes
Kubernetes
Builds
Docker Hub
Docker Scout
MCP Toolkit
BETA
Models
Extensions
Containers
Give feedback
Container CPU usage
22.84% / 800% (8 CPUs available)
Container memory usage
1.4GB / 7.44GB
Show charts
Columns
Search
Only show running containers
Name
Container ID
Image
Port(s)
Actions
xenodochial_lic b21963d298d8 devcontain