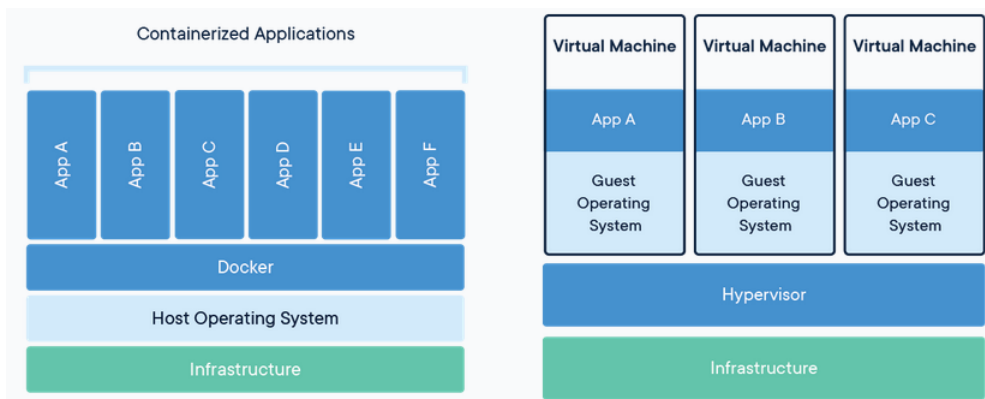


Chapitre 4 – Docker sous Windows (Docker Desktop et WSL 2)

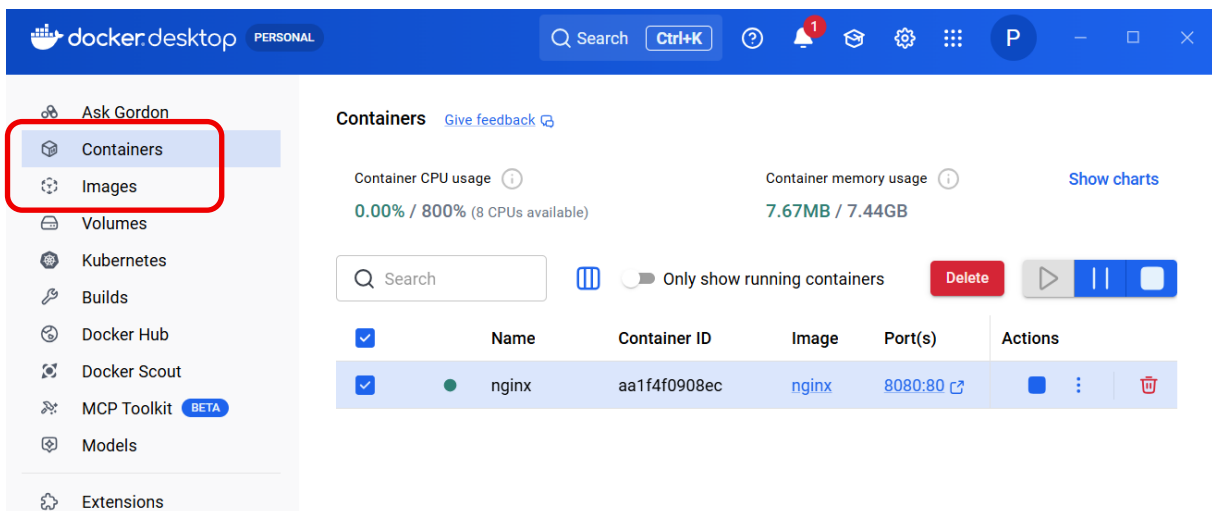
Comparaison entre conteneurs Docker et machines virtuelles



- 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.

```
Administrateur : Windows PowerShell
PS C:\WINDOWS\system32> wsl.exe --install
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.
Installation du composant facultatif Windows : VirtualMachinePlatform

Outil Gestion et maintenance des images de déploiement
Version : 10.0.26100.5074

Version de l'image : 10.0.26200.7840

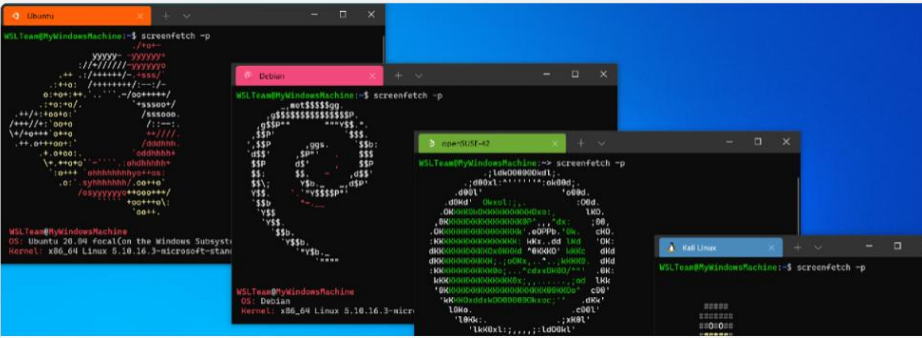
Activation de la ou des fonctionnalités
[=====100.0%=====]
L'opération a réussi.
L'opération demandée est réussie. Les modifications ne seront pas effectives avant que le système ne soit réamorcé.
L'opération demandée est réussie. Les modifications ne seront pas effectives avant que le système ne soit réamorcé.
PS C:\WINDOWS\system32>
```

```
Administrateur : Windows PowerShell
PS C:\WINDOWS\system32> wsl.exe --list --online
Voici la liste des distributions valides qui peuvent être installées.
Installez à l'aide de 'wsl.exe --install <Distro>'.

NAME                                FRIENDLY NAME
-----                                -
Ubuntu                               Ubuntu
Ubuntu-24.04                         Ubuntu 24.04 LTS ←
openSUSE-Tumbleweed                  openSUSE Tumbleweed
openSUSE-Leap-16.0                   openSUSE Leap 16.0
SUSE-Linux-Enterprise-15-SP7         SUSE Linux Enterprise 15 SP7
SUSE-Linux-Enterprise-16.0          SUSE Linux Enterprise 16.0
kali-linux                           Kali Linux Rolling
Debian                               Debian GNU/Linux
AlmaLinux-8                          AlmaLinux OS 8
AlmaLinux-9                          AlmaLinux OS 9
AlmaLinux-Kitten-10                 AlmaLinux OS Kitten 10
AlmaLinux-10                         AlmaLinux OS 10
archlinux                            Arch Linux
FedoraLinux-43                       Fedora Linux 43
FedoraLinux-42                       Fedora Linux 42
eLxr                                 eLxr 12.12.0.0 GNU/Linux
Ubuntu-20.04                         Ubuntu 20.04 LTS
Ubuntu-22.04                         Ubuntu 22.04 LTS
OracleLinux_7_9                      Oracle Linux 7.9
OracleLinux_8_10                     Oracle Linux 8.10
OracleLinux_9_5                      Oracle Linux 9.5
openSUSE-Leap-15.6                   openSUSE Leap 15.6
SUSE-Linux-Enterprise-15-SP6         SUSE Linux Enterprise 15 SP6
```

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.

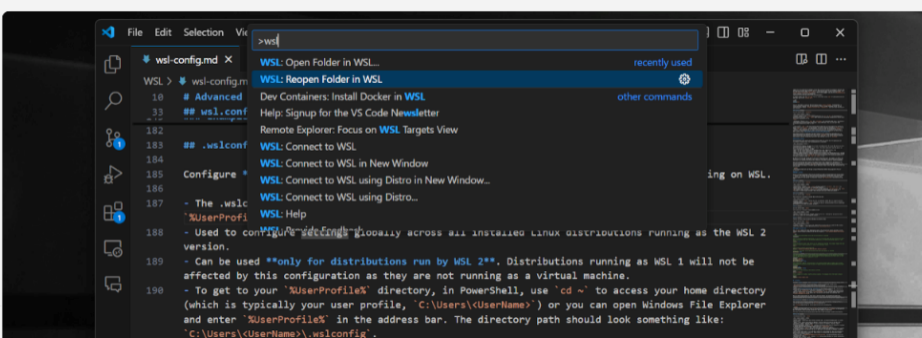
[Documentation du sous-système Windows pour Linux \(WSL\)](#)

[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 :

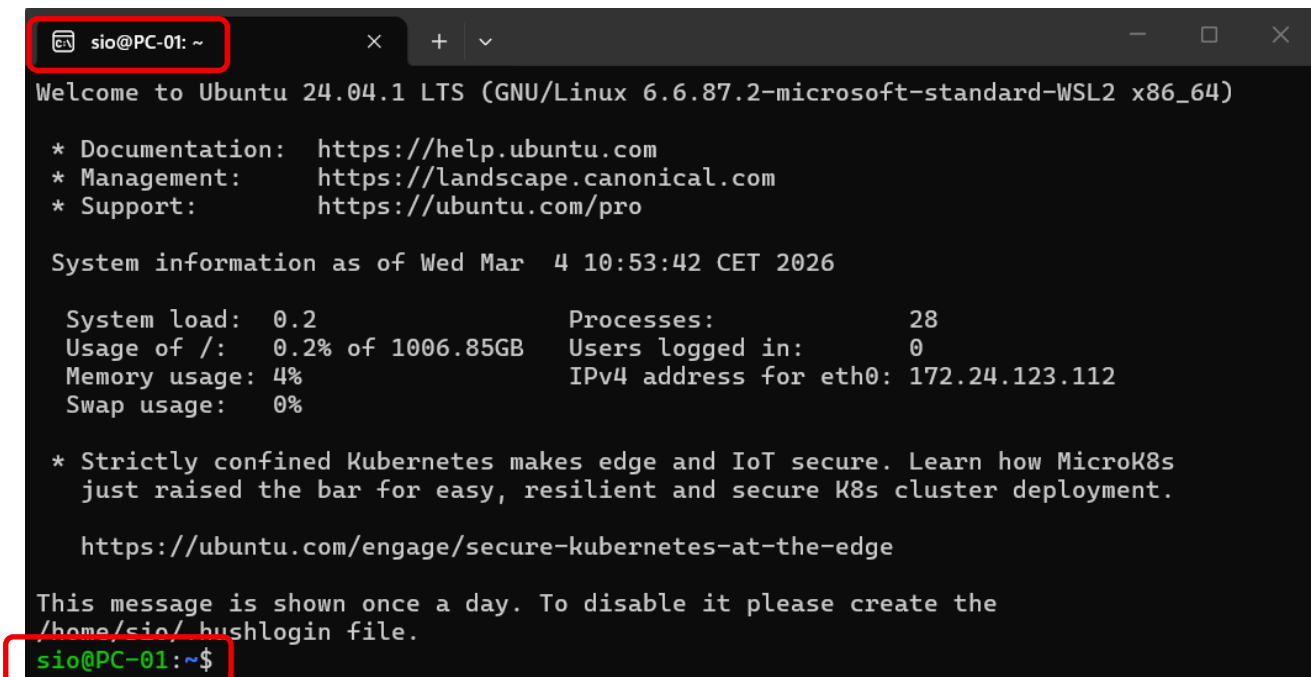
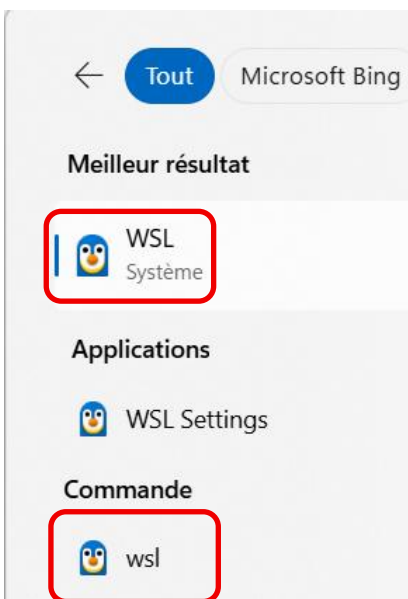
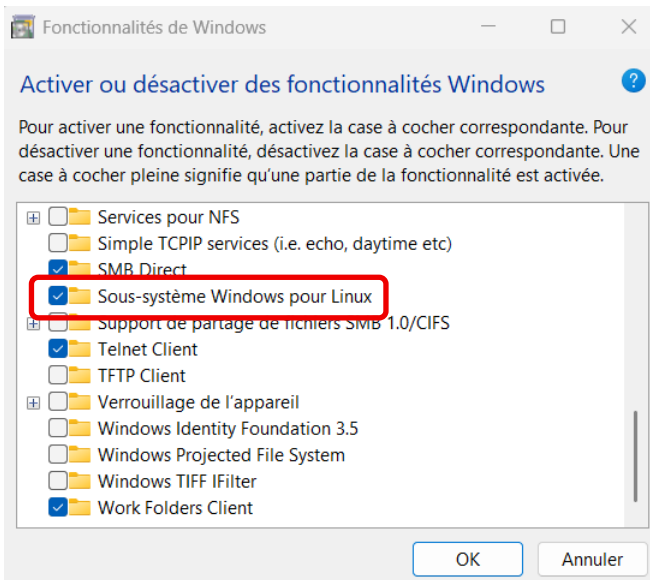
```
'code --install-extension ms-vscode-remote.remote-wsl'
```

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

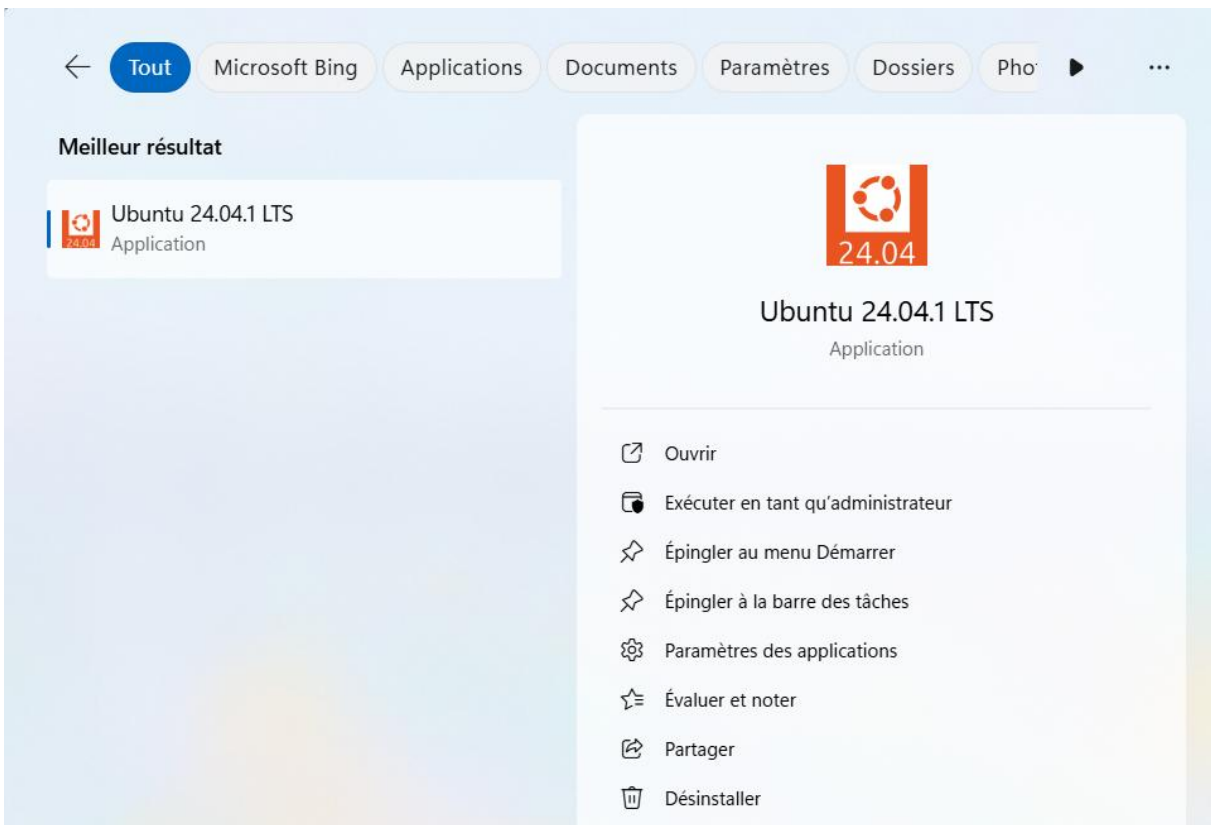
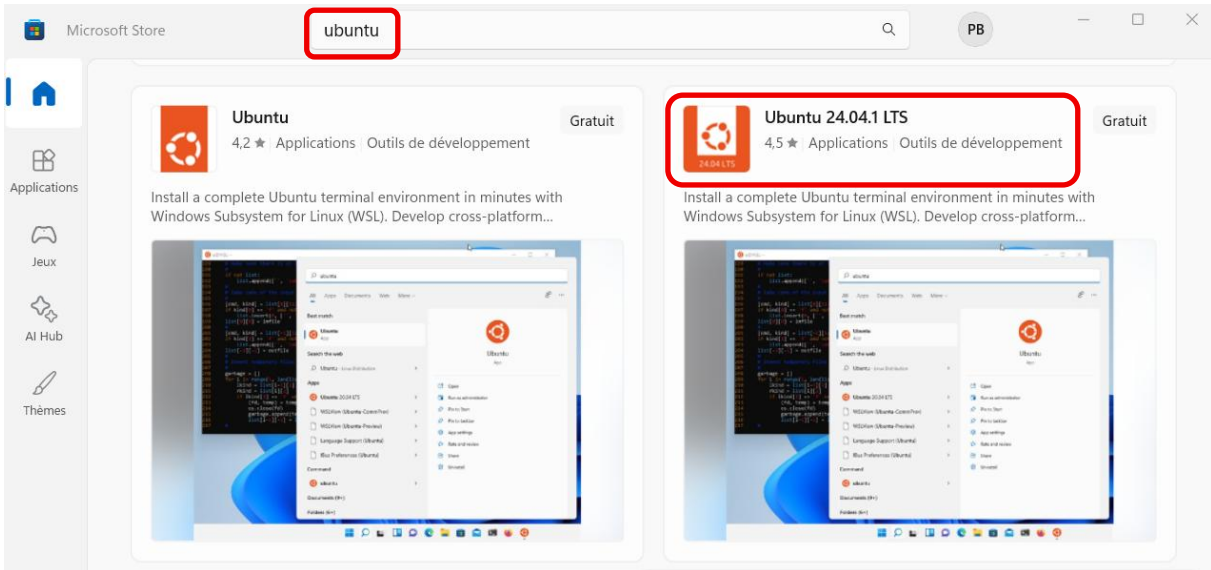
Vous pouvez également accéder à davantage d'options VS Code Remote via la palette de commandes dans VS Code

[En savoir plus sur l'utilisation de WSL avec VS Code](#)



```
sio@PC-01: /mnt/c/WINDOWS/ system32$
```

2. Installation Ubuntu.



```
sio@PC-01: /
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: sio
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

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

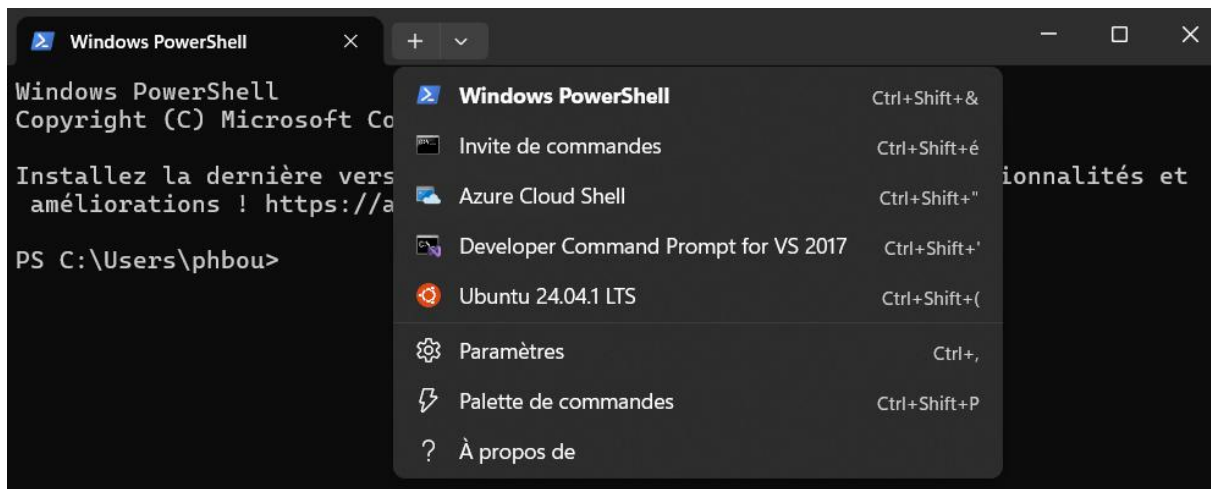
System information as of Wed Feb 18 20:04:35 CET 2026

System load:  2.65          Processes:      29
Usage of /:   0.1% of 1006.85GB  Users logged in:  0
Memory usage: 4%          IPv4 address for eth0: 172.24.123.112
Swap usage:   0%

This message is shown once a day. To disable it please create the
/home/sio/.hushlogin file.

sio@PC-01:~$ ls
sio@PC-01:~$ pwd
/home/sio
sio@PC-01:~$ cd /
sio@PC-01:/$ ls
bin          boot  etc  init  lib.usr-is-merged  lost+found  mnt  proc  run  sbin.usr-is-merged  srv  tmp  var
bin.usr-is-merged  dev  home  lib  lib64              media       opt  root  sbin  snap                sys  usr
sio@PC-01:/$
```

Terminal Windows :



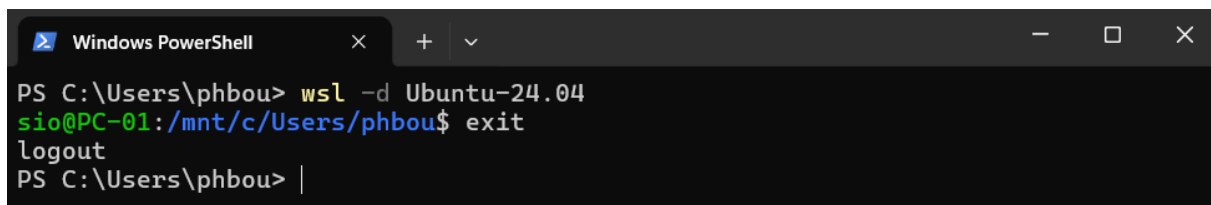
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Installez la dernière version de Windows 11 pour bénéficier des dernières améliorations ! https://aka.ms/windowsupdate

PS C:\Users\phbou>
```

The screenshot shows a Windows PowerShell window with a context menu open over it. The menu items are:

- Windows PowerShell (Ctrl+Shift+&)
- Invite de commandes (Ctrl+Shift+é)
- Azure Cloud Shell (Ctrl+Shift+")
- Developer Command Prompt for VS 2017 (Ctrl+Shift+')
- Ubuntu 24.04.1 LTS (Ctrl+Shift+(
- Paramètres (Ctrl+,)
- Palette de commandes (Ctrl+Shift+P)
- À propos de



```
PS C:\Users\phbou> wsl -d Ubuntu-24.04
sio@PC-01:/mnt/c/Users/phbou$ exit
logout
PS C:\Users\phbou>
```

3. Installation Docker.

www.docker.com/products/docker-desktop/

New Learn how to run Docker Hardened Images in Prod. Register for the 2/19 webinar now →

AI Products Developers Pricing Support Blog Company

Search Sign In Get Started

Docker Desktop

The #1 containerization software for developers and teams

Streamline development with Docker Desktop's powerful container tools.

Choose plan Download Docker Desktop

Download for Mac – Apple Silicon

Download for Mac – Intel Chip

Download for Windows – AMD64

Download for Windows – ARM64

Container CPU usage: 0.01% / 1200% (12 CPUs allocated)

Container memory usage: 249.6MB / 7.47 GB

| Container ID | Image | Port(s) | Last started | CPU % | Actions |
|--------------|-------------------|-----------|--------------|-------|---------|
| 8j8jpf3411 | free-willy:latest | 8000:8000 | 1 hour ago | 0% | ▶ ⋮ 🗑️ |

Téléchargements

Nom

▼ Aujourd'hui

Docker Desktop Installer

Configuration


- Use WSL 2 instead of Hyper-V (recommended)
- Allow Windows Containers to be used with this installation
- Add shortcut to desktop

Windows Containers should only be enabled if you understand the risks. For more information, see [our docs](#).

OK



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.

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
Unpacking file: frontend/resources/regedit/vbs/wsRegReadListStream.wsf
```

Docker Desktop 4.60.1

Installation succeeded

You must restart Windows to complete installation.

Close and restart

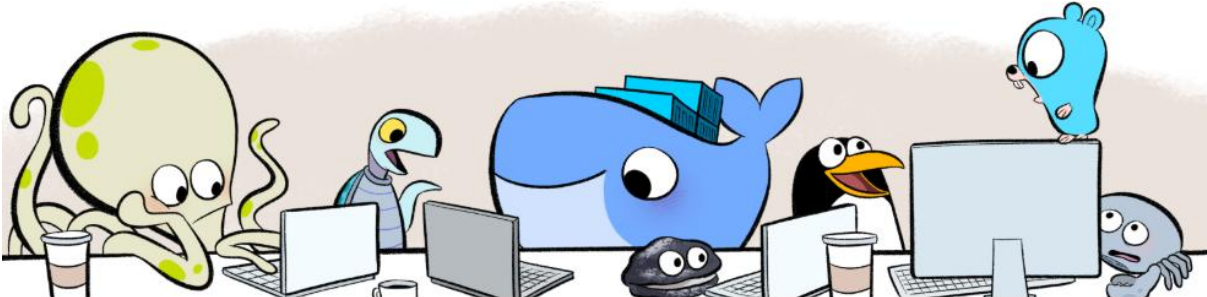
Meilleur résultat



Docker Desktop
Application



Exécuter en tant qu'administrateur



Docker Subscription Service Agreement

By selecting **accept**, you agree to the [Subscription Service Agreement](#), the [Docker Data Processing Agreement](#), the [Data Privacy Policy](#) and the [Docker AI Supplemental Terms](#).

Commercial use of Docker Desktop at a company of more than 250 employees OR more than \$10 million in annual revenue requires a paid subscription (Pro, Team, or Business). See [subscription details](#).

[View Full Terms](#)

Accept

Close

Créer un compte et se connecter :



Welcome to Docker Desktop

Sign in to connect to your Docker Desktop subscription or access online features.

Sign up

Already have an account? [Sign in](#)

[Continue without signing in](#)



Create your account

Work Personal

Email

Email is required.

Username

Username is required.

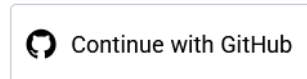
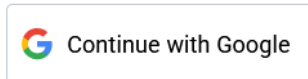
Password

Password is required.

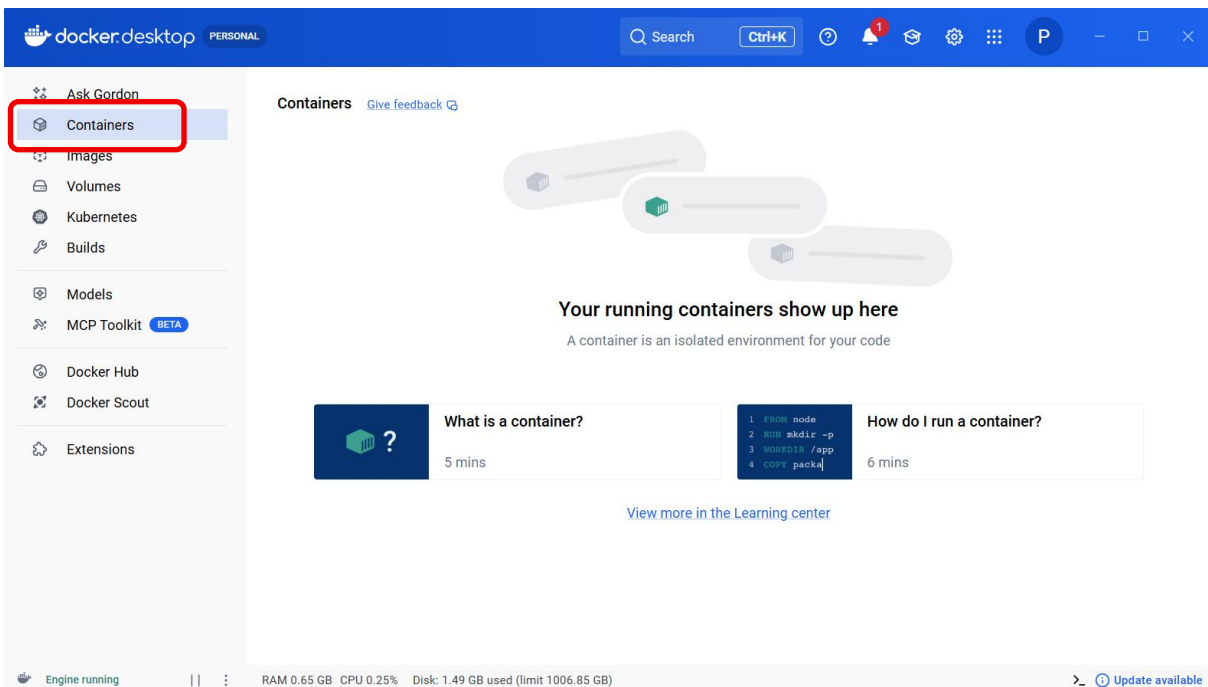
Send me occasional product updates and announcements.

Sign up


OR



Already have an account? [Sign in](#)



 Docker Desktop is starting

 Go to the Dashboard

 philboucey >

Change settings Ctrl+Comma

Troubleshoot

Give feedback


About Docker Desktop


Docker Hub

Documentation


Extensions >

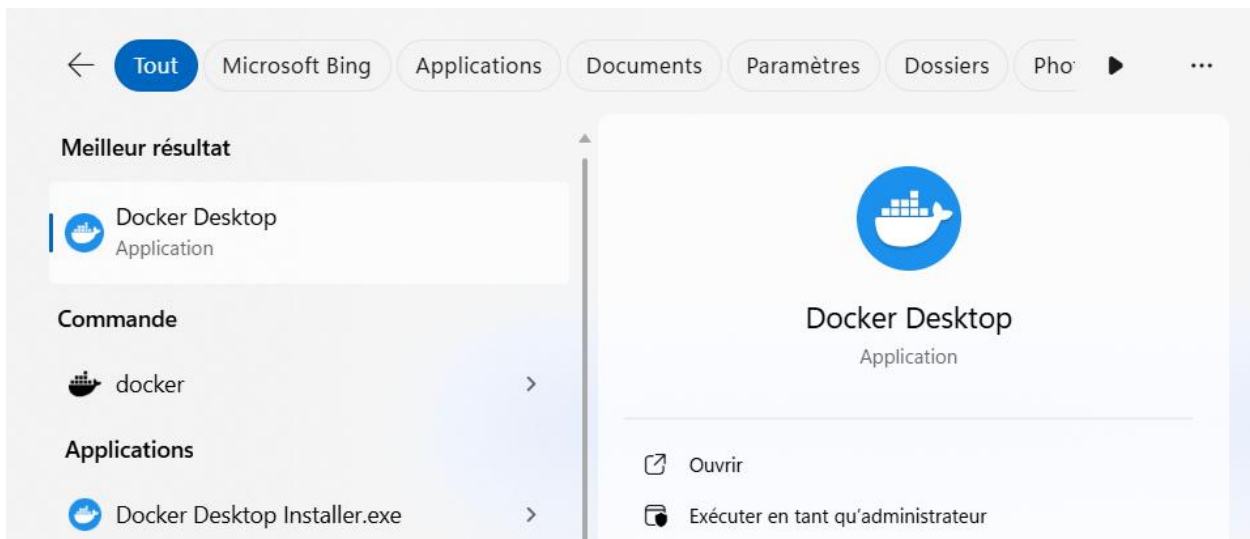
Kubernetes Context >

 Download update

 Restart

 Pause

 Quit Docker Desktop



4. Premier test.

```
sio@PC-01: ~  
sio@PC-01:~$ docker --version  
The command 'docker' could not be found in this WSL 2 distro.  
We recommend to activate the WSL integration in Docker Desktop settings.  
For details about using Docker Desktop with WSL 2, visit:  
https://docs.docker.com/go/wsl2/  
sio@PC-01:~$
```

Cf. page 17

Terminal Windows :

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. Tous droits réservés.  
Installez la dernière version de PowerShell pour de nouvelles fonctionnalités  
et améliorations ! https://aka.ms/PSWindows  
PS C:\Users\phbou> docker --version  
Docker version 29.2.0, build 0b9d198  
PS C:\Users\phbou>
```

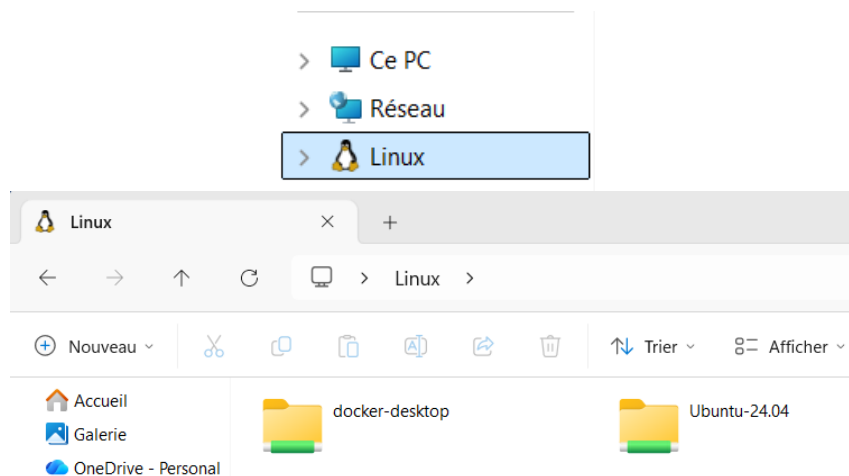
```
Windows PowerShell x Invite de commandes x + v - □ ×
Microsoft Windows [version 10.0.26200.7840]
(c) Microsoft Corporation. Tous droits réservés.

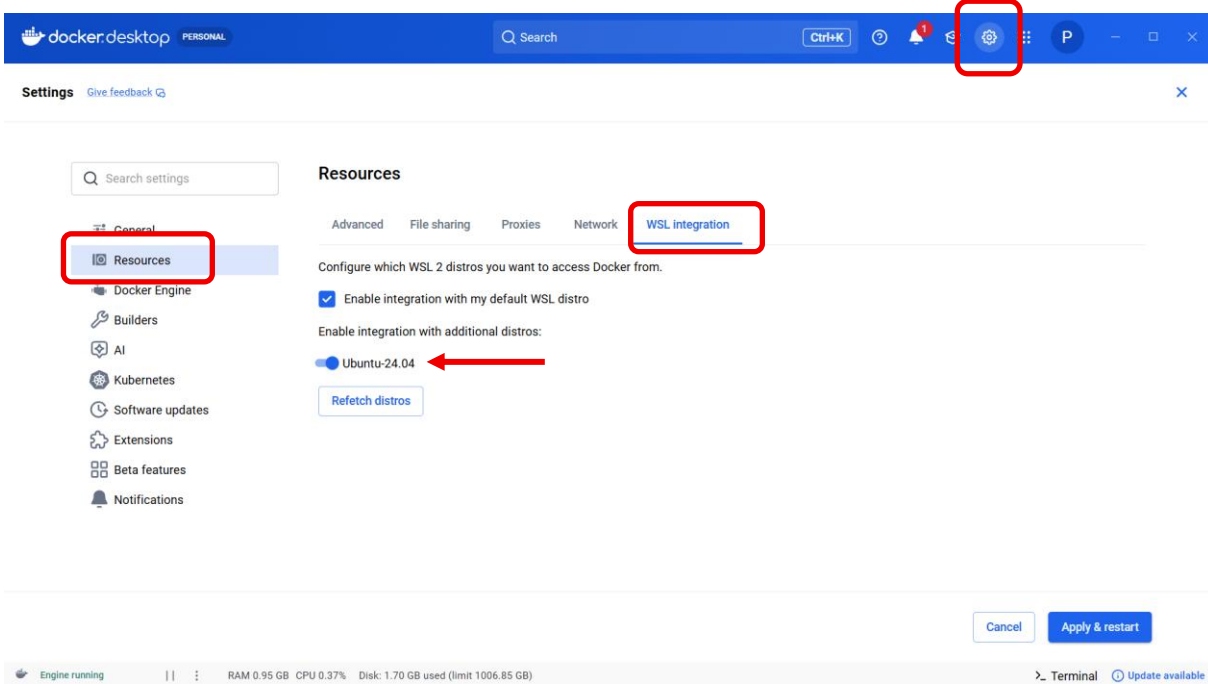
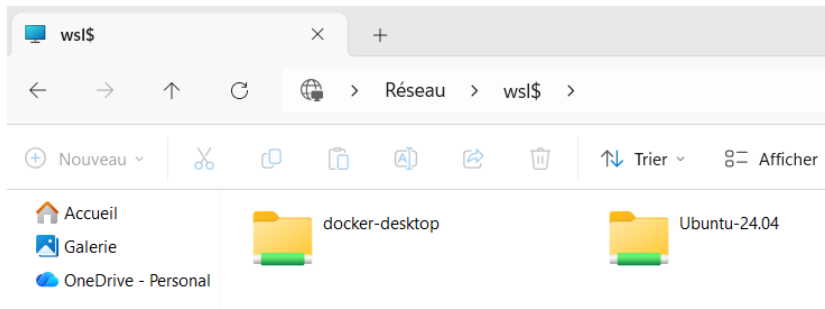
C:\Users\phbou> docker version
Client:
Version:          29.2.0
API version:      1.53
Go version:       go1.25.6
Git commit:       0b9d198
Built:            Mon Jan 26 19:28:56 2026
OS/Arch:          windows/amd64
Context:          desktop-linux

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 x + v - □ ×
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>
```





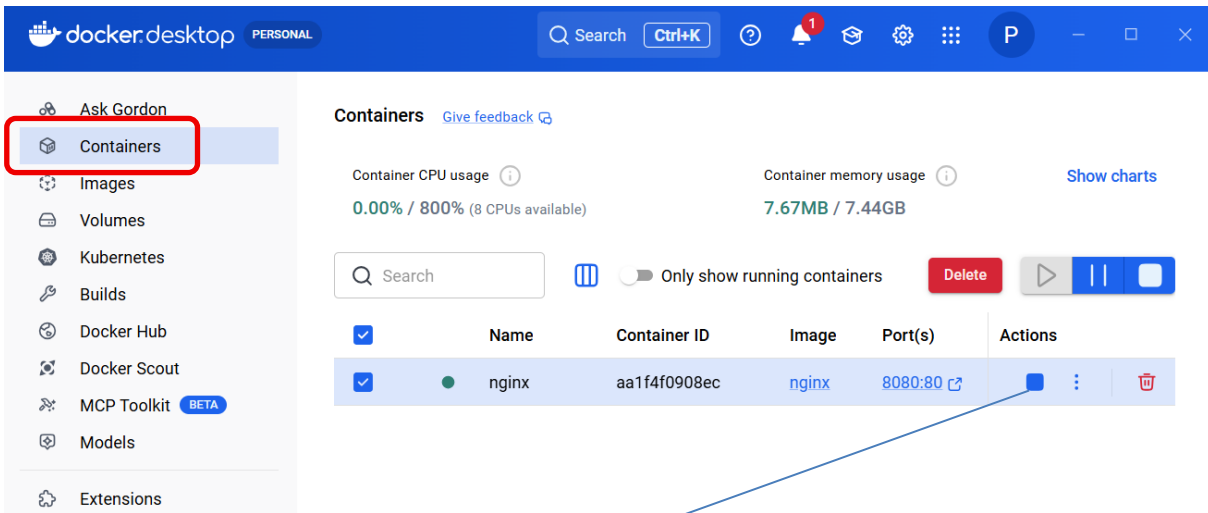
5. Deuxième test.

```
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
```



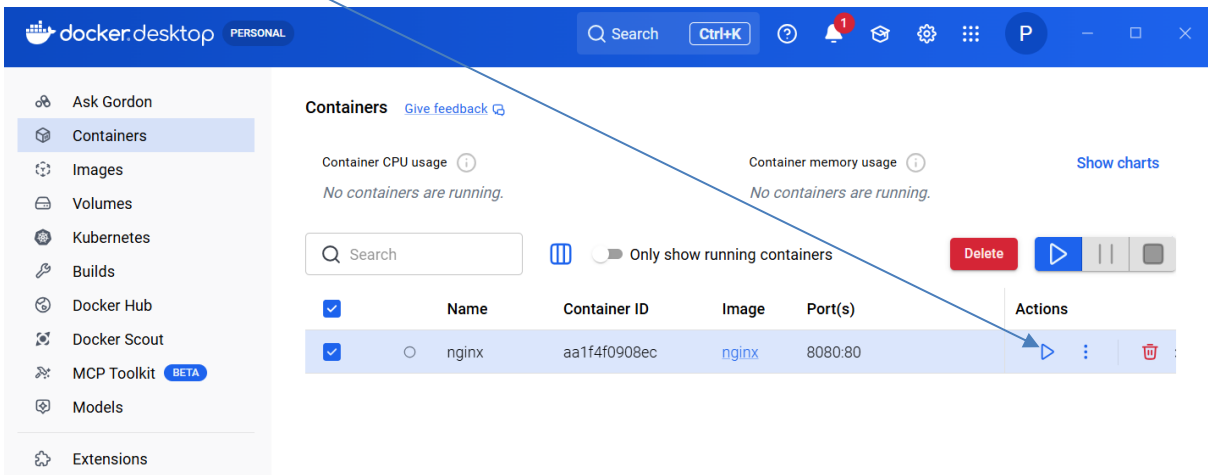
→ Docker télécharge dans un premier temps l'image nginx. Il crée ensuite une instance de cette image qui est le conteneur.



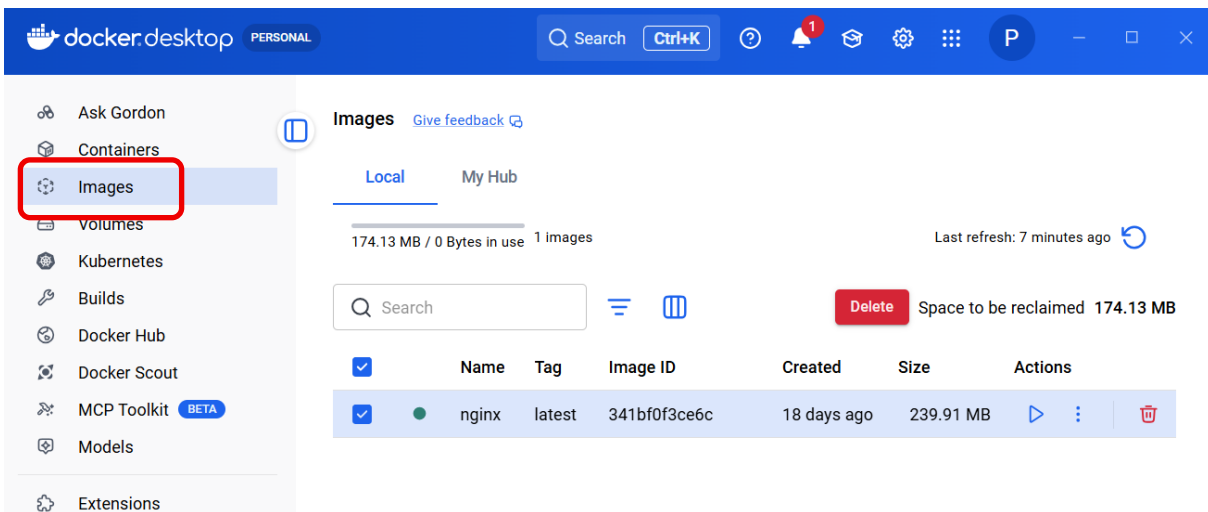


Si on relance la commande, l'image ne sera pas téléchargée de nouveau car elle existe déjà en local. Par contre, un nouveau conteneur sera créé.

- Action container : stop

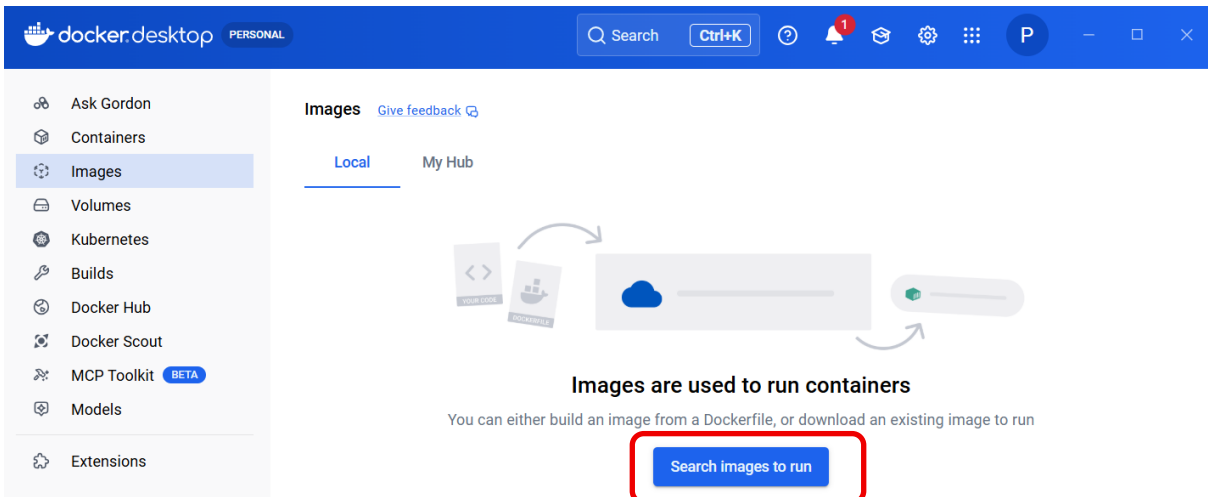


- Delete container
- Puis Delete image :

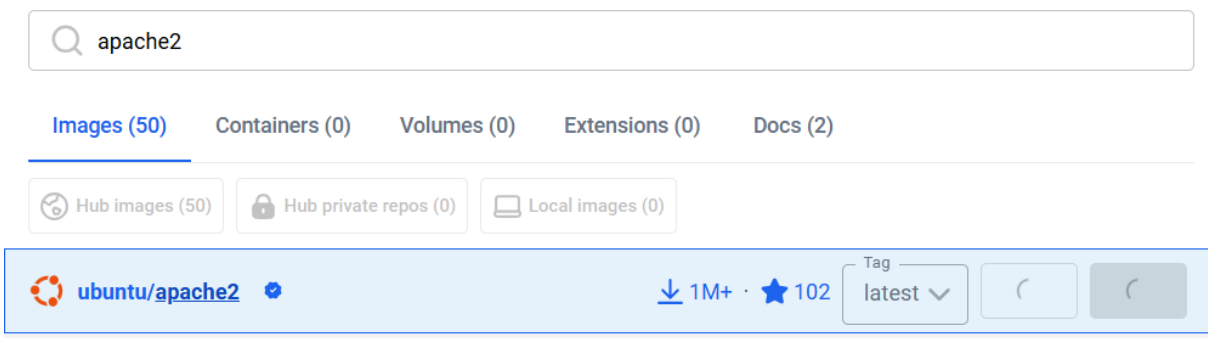
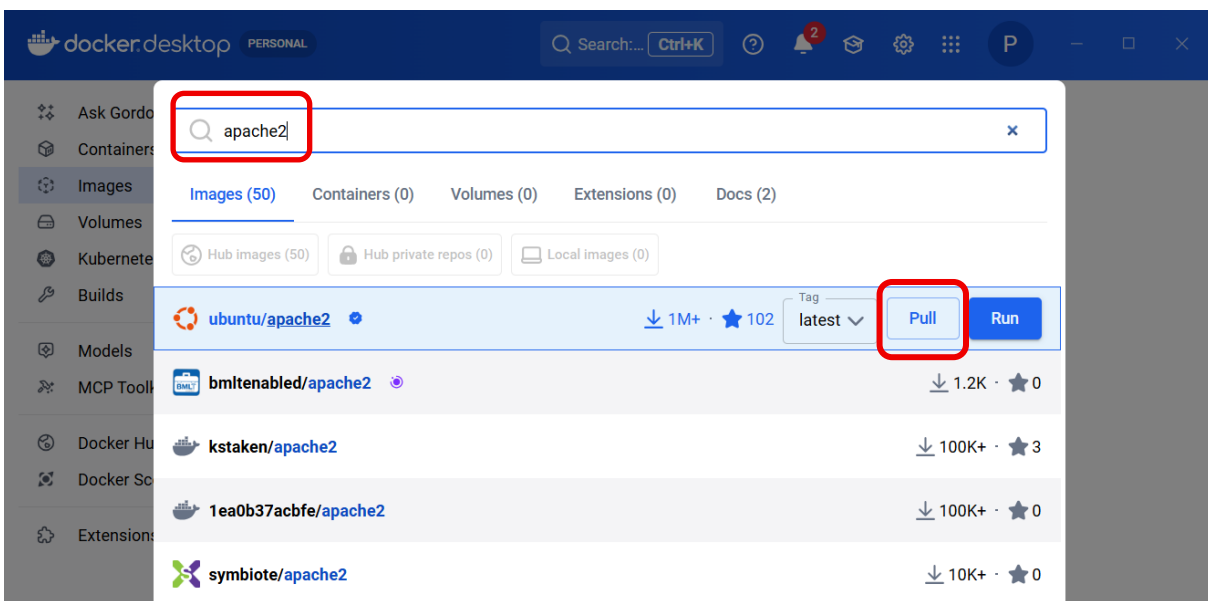


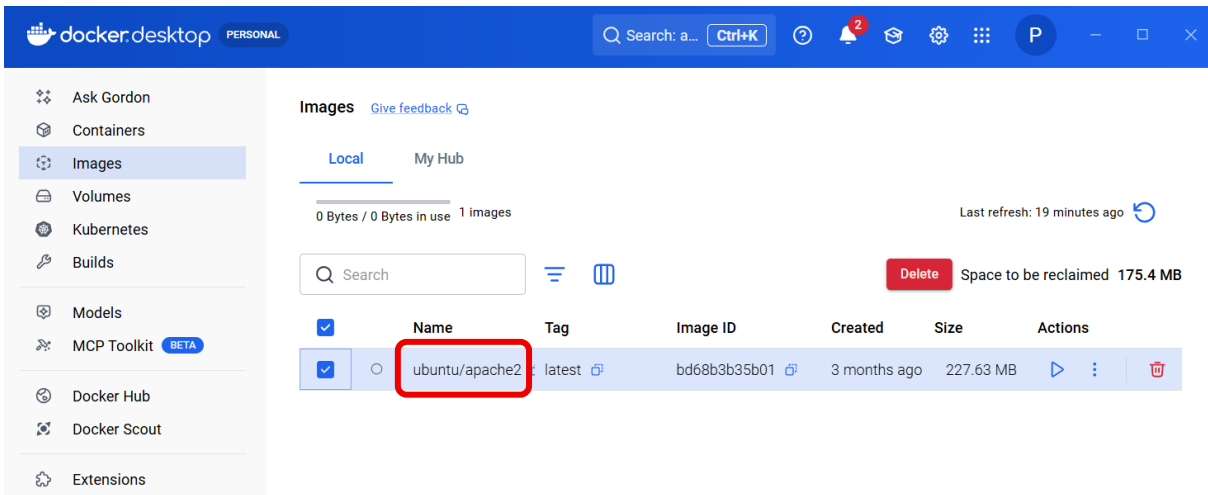
6. Troisième test.

- Search images to run

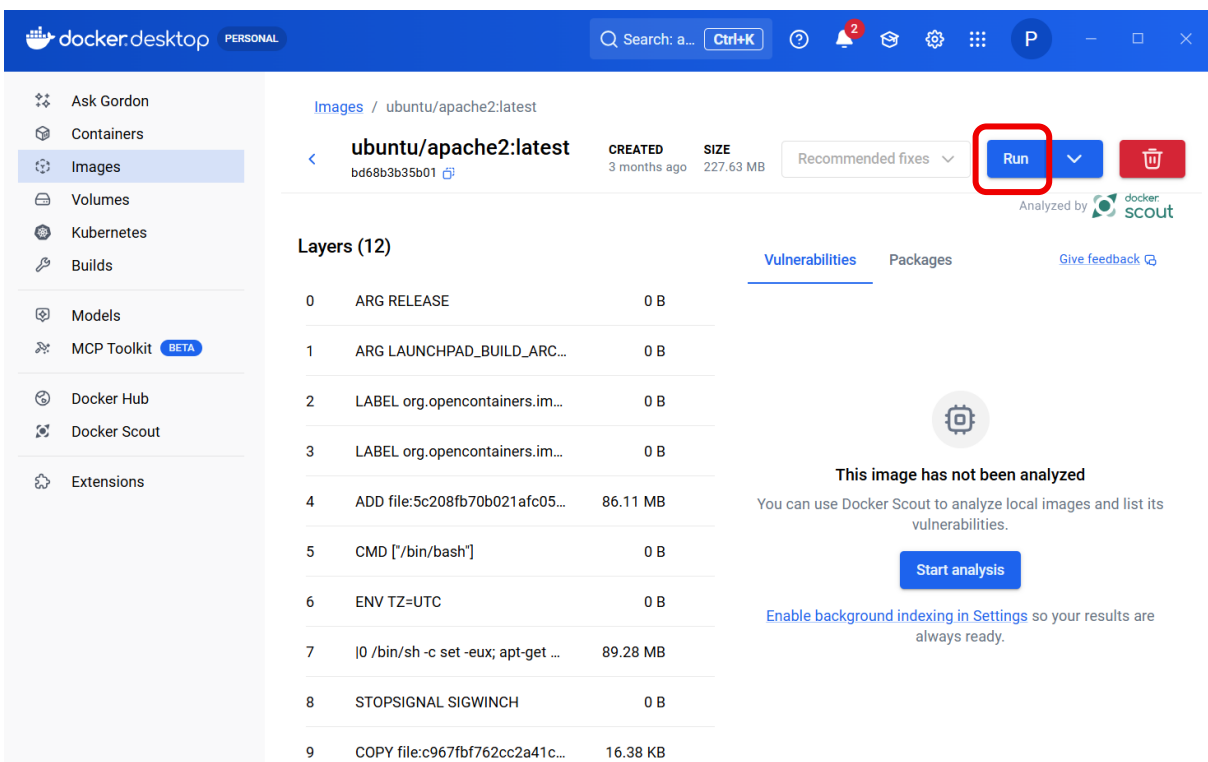


■ Pull :





■ Run :





Run a new container

ubuntu/apache2:latest

Optional settings

Container name

A random name is generated if you do not provide one.

Ports

Enter "0" to assign randomly generated host ports.

Host port :80/tcp

Volumes

Host path ... Container path +

Environment variables

Variable Value +

docker.desktop PERSONAL Search: a... Ctrl+K

Containers / apache2-container

apache2-container STATUS Running (1 second ago)

fb446b0a3ccf ubuntu/apache2:latest 8080:80

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

Containers / apache2-container

apache2-container
 fba46b0a3ccf ubuntu/apache2:latest
 8080:80

STATUS: Running (1 minute ago)

Logs Inspect Bind mounts **Exec** Files Stats

```
# 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%

Memory usage: 9.43MB / 7.62GB

docker.desktop PERSONAL Search: a... Ctrl+K

Containers Give feedback

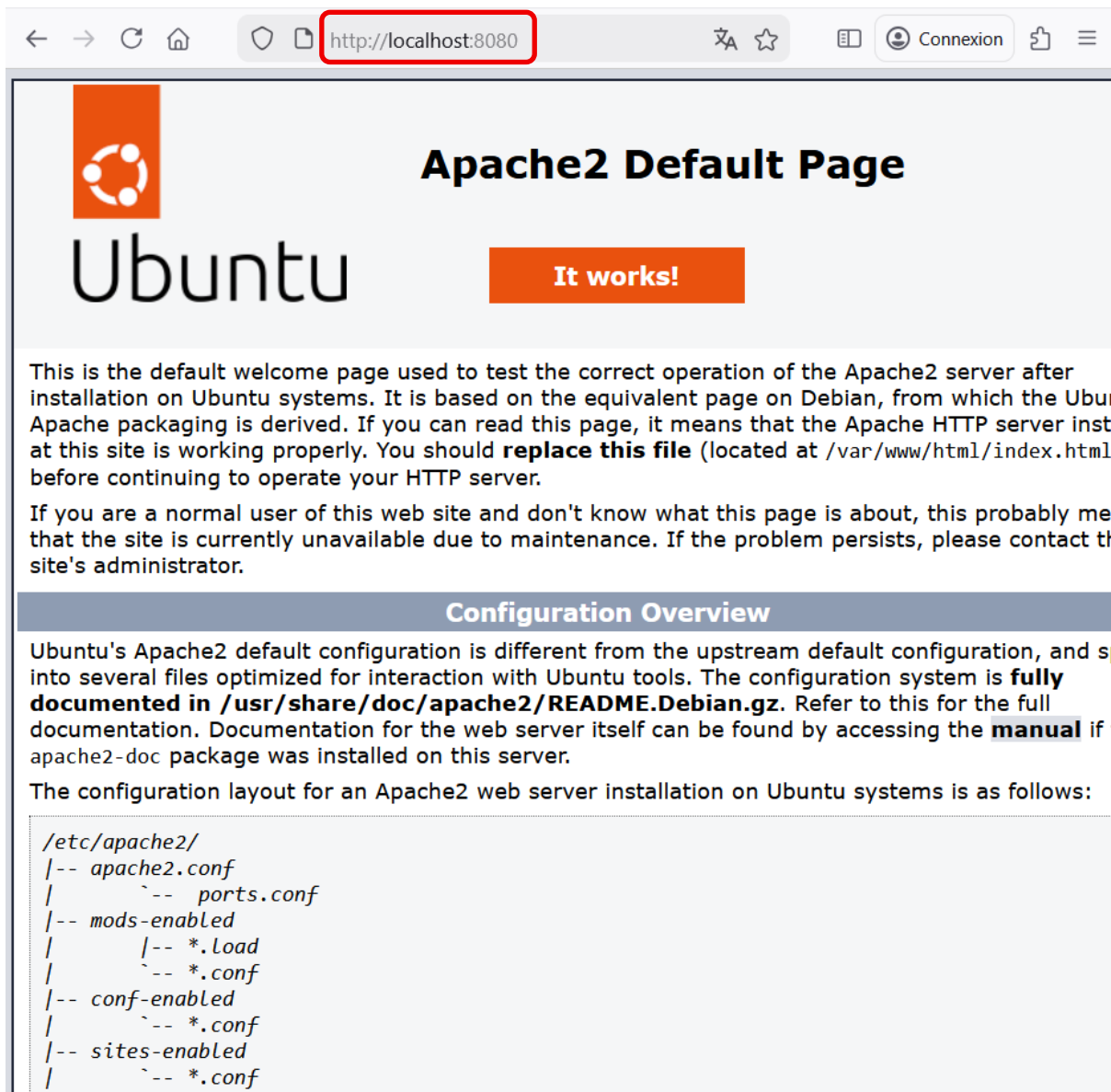
Container CPU usage: 0.01% / 800% (8 CPUs available)

Container memory usage: 9.43MB / 7.44GB

Show charts


Search Only show running containers

| | Name | Container ID | Image | Port(s) | Actions |
|--------------------------|----------------|--------------|--------------------|---------|--|
| <input type="checkbox"/> | apache2-contai | fba46b0a3ccf | ubuntu/ap: 8080:80 | | <input type="checkbox"/> ⋮ <input type="trash"/> |



← → ↻ 🏠 🔒 http://localhost:8080 🔍 ☆ 📄 Connexion 📄 ☰

Apache2 Default Page



Ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/  
|-- apache2.conf  
|   |-- ports.conf  
|-- mods-enabled  
|   |-- *.load  
|   |-- *.conf  
|-- conf-enabled  
|   |-- *.conf  
|-- sites-enabled  
|   |-- *.conf
```

- Arrêter et supprimer le container puis l'image.

7. Quatrième test.

```
sio@PC-01: ~  
sio@PC-01:~$ docker run hello-world  
Unable to find image 'hello-world:latest' locally  
latest: Pulling from library/hello-world  
17eec7bbc9d7: Pull complete  
ea52d2000f90: Download complete  
Digest: sha256:ef54e839ef541993b4e87f25e752f7cf4238fa55f017957c2eb44077083d7a6a  
Status: Downloaded newer image for hello-world:latest  
  
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
  
To generate this message, Docker took the following steps:  
1. The Docker client contacted the Docker daemon.  
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
   (amd64)  
3. The Docker daemon created a new container from that image which runs the  
   executable that produces the output you are currently reading.  
4. The Docker daemon streamed that output to the Docker client, which sent it  
   to your terminal.  
  
To try something more ambitious, you can run an Ubuntu container with:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/  
  
sio@PC-01:~$
```

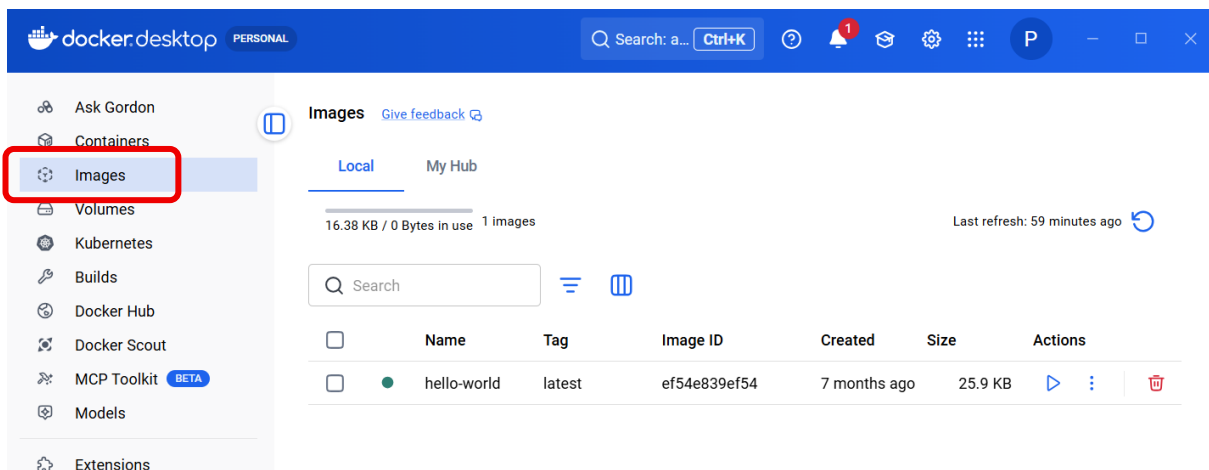
Message
affiché

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).

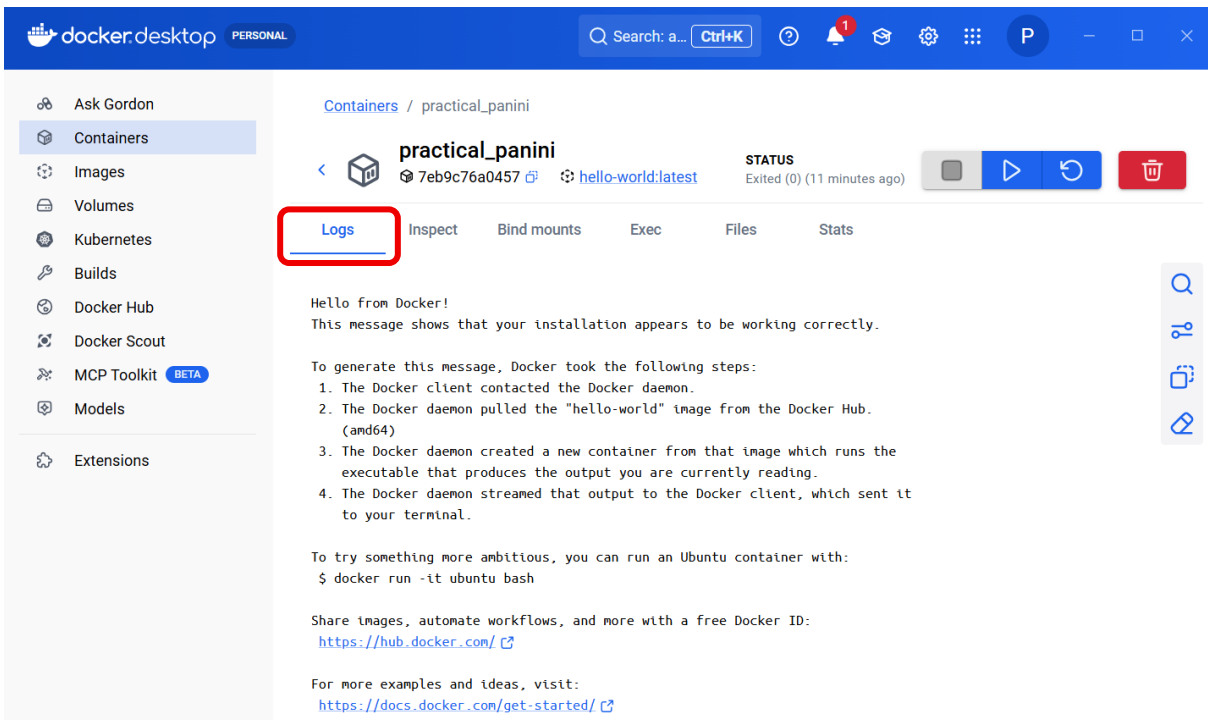
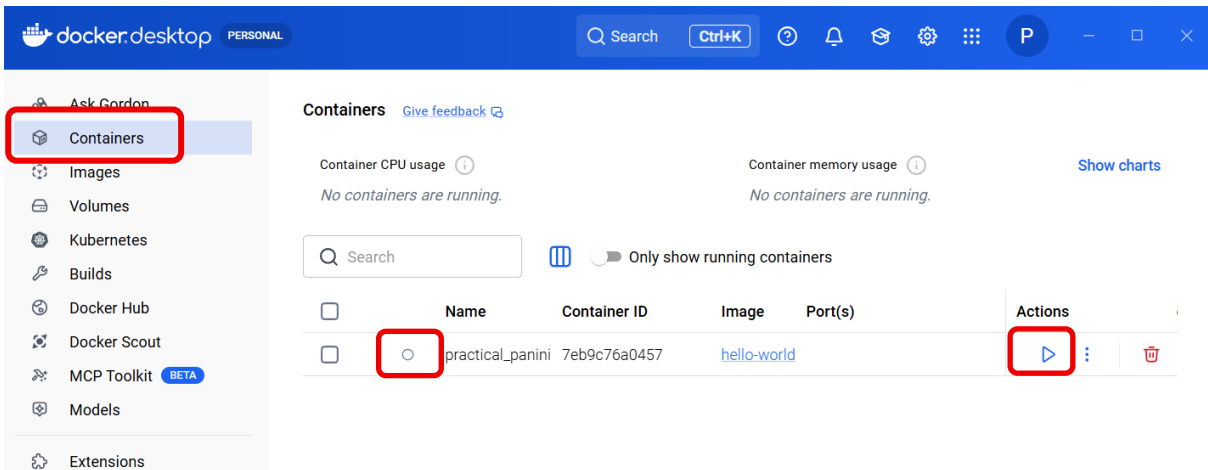
Elle effectue 2 téléchargements car 2 couches image : une image scratch de base (vide : pas d'OS) et une destinée à exécuter le fichier hello codé en C.

Elle crée et démarre un conteneur (affichage du message) puis arrête le conteneur.

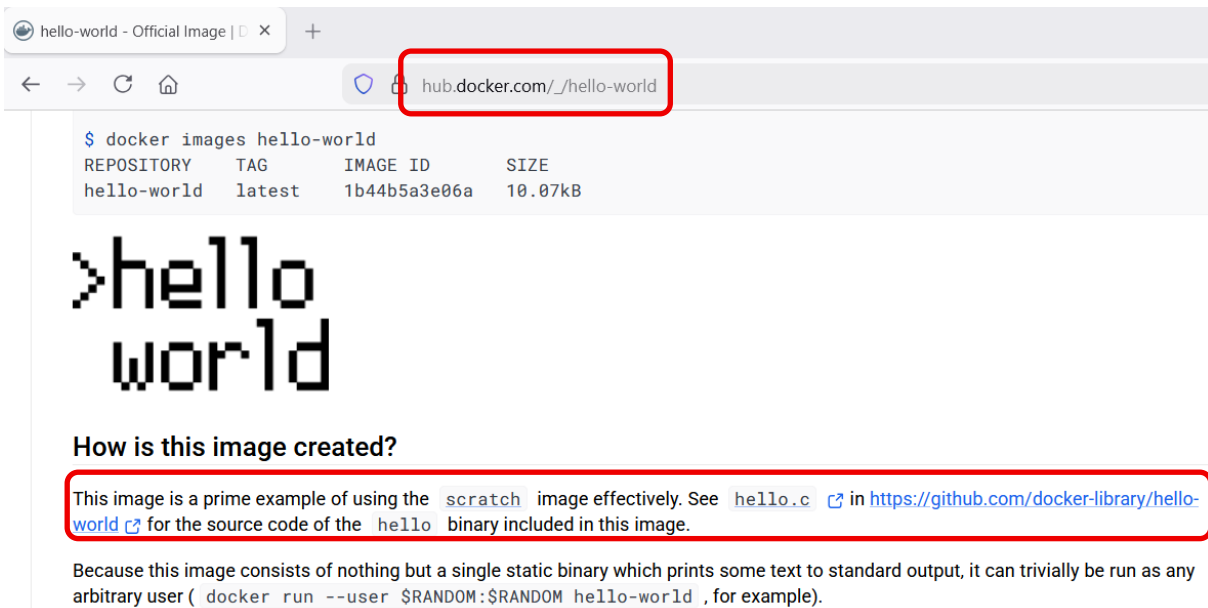
Digest : clé permettant de s'assurer que l'application est correctement téléchargée sans MITM.



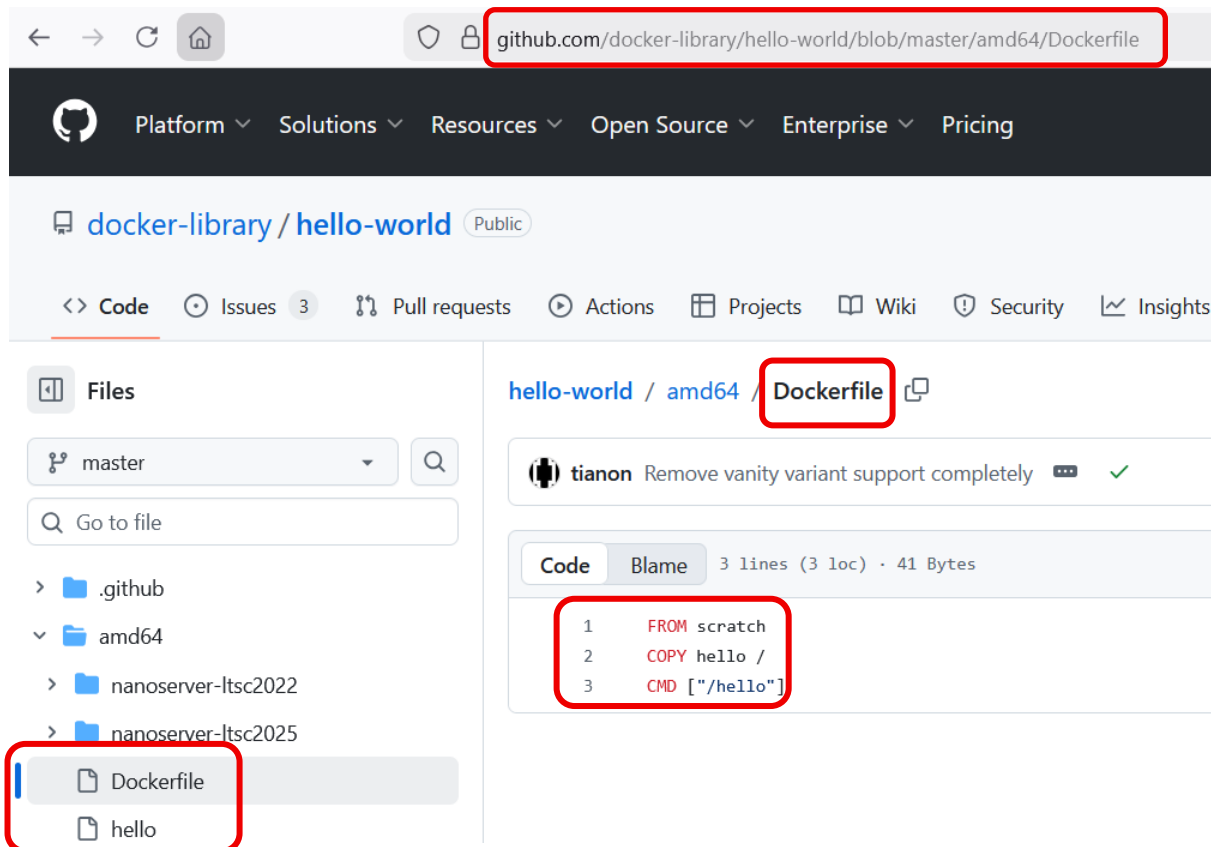
Le conteneur est arrêté après l'affichage du message :



Les sources de l'image :



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

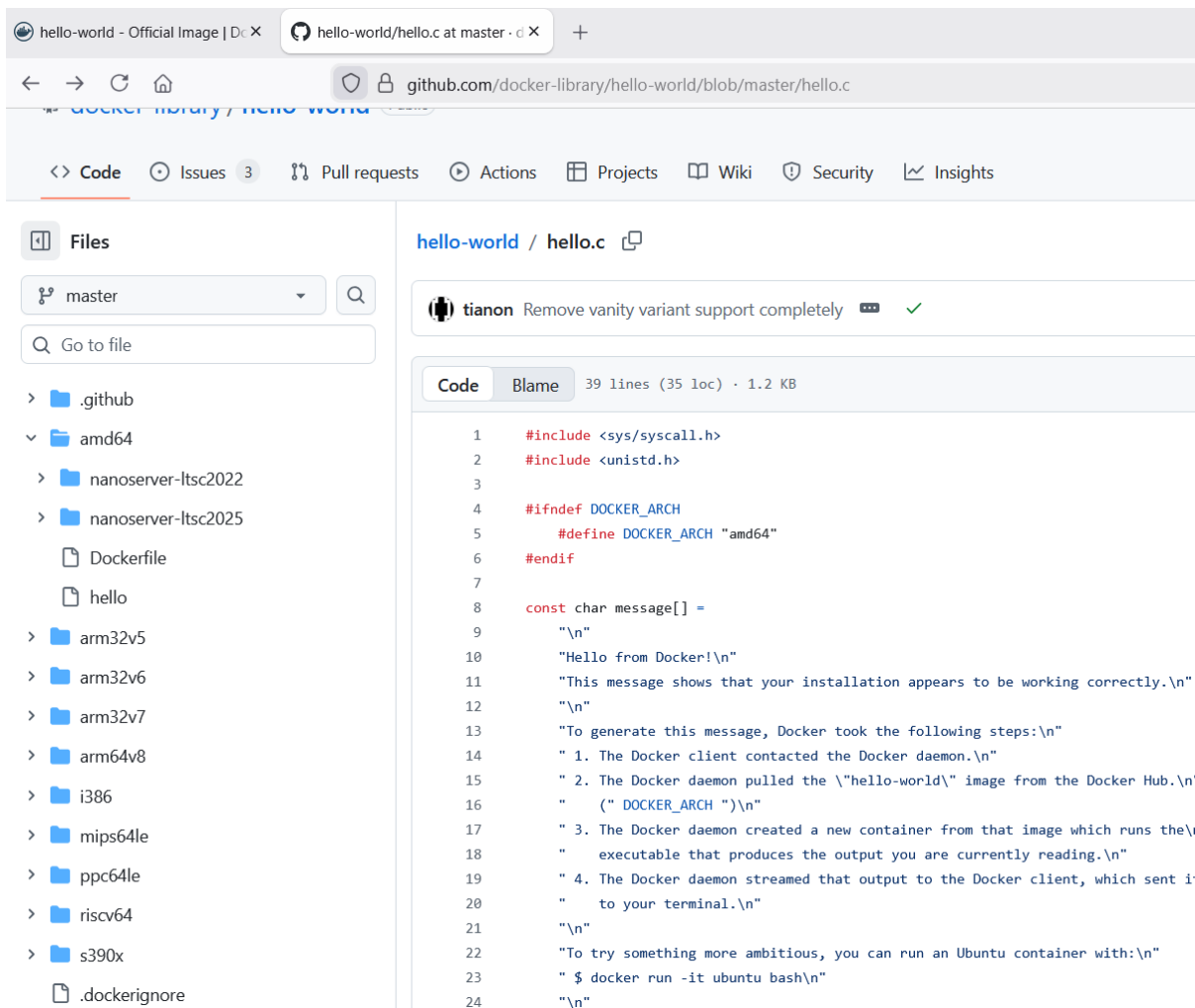


FROM scratch (image de base disposant elle aussi d'un DockerFile) : cela signifie que l'on part de rien c'est à dire pas d'image avec un OS.

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

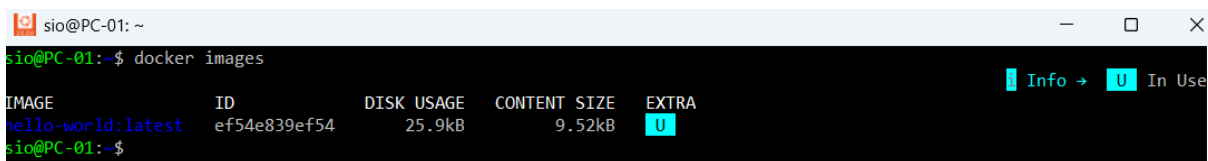
CMD [« /hello »] : commande lancée au démarrage de l'image et exécutée par l'OS Ubuntu

Code du fichier hello :

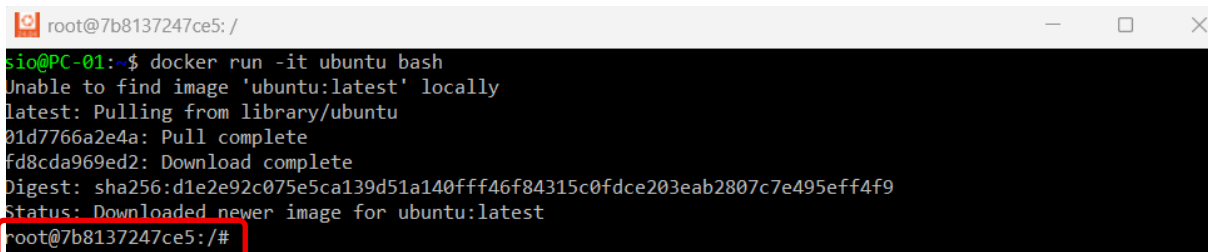


8. Cinquième test : mode interactif d'utilisation d'un conteneur.

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

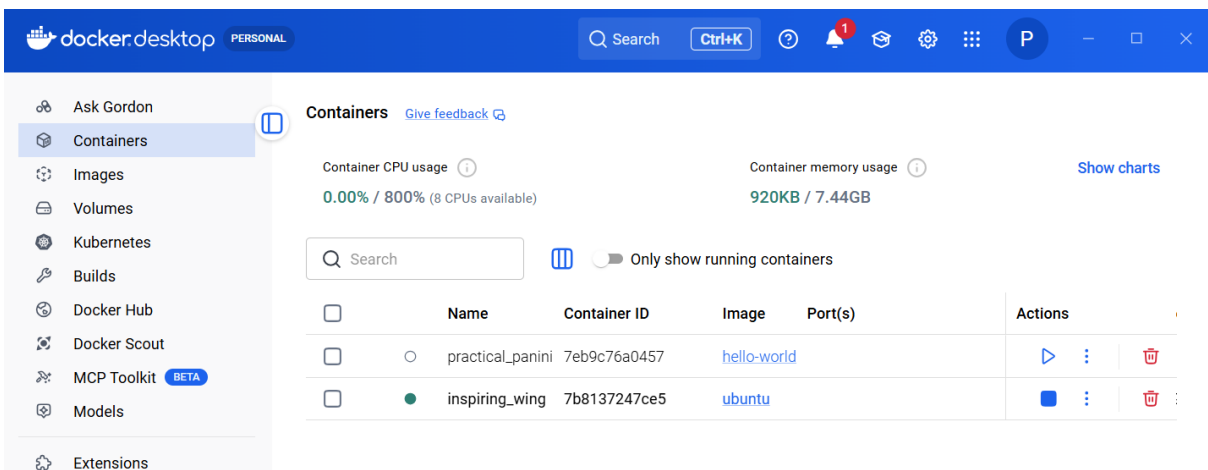
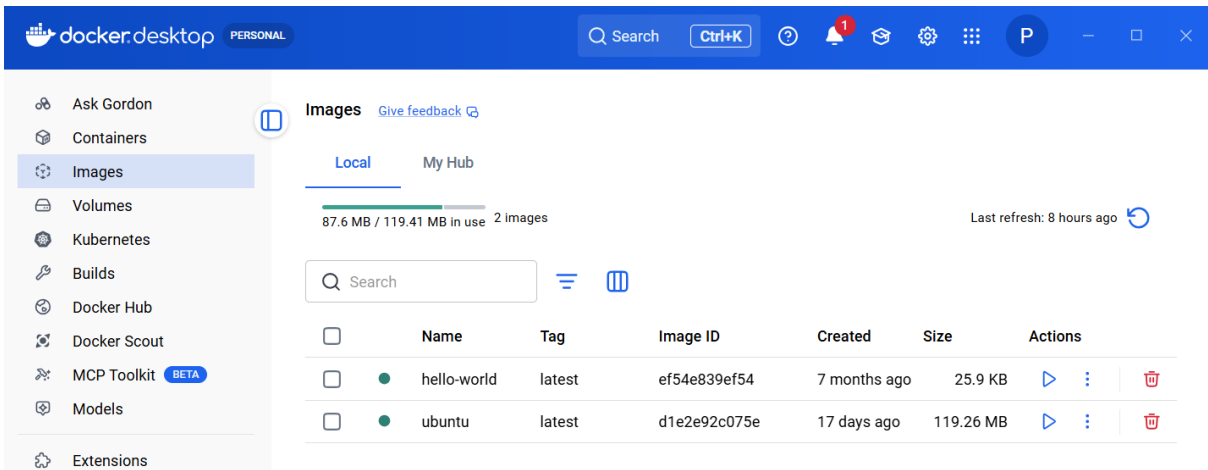


- Lancement d'un conteneur en mode interactif qui lancera un shell bash qui interprètera les commandes :

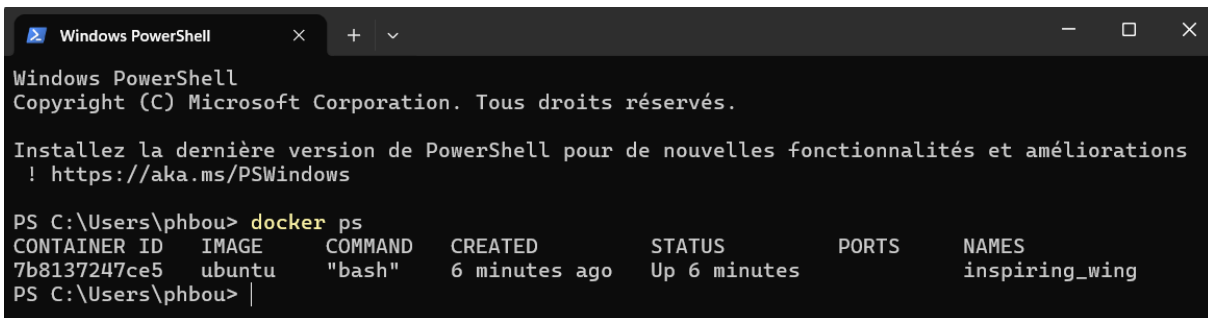


Option -i : permet d'interagir avec le conteneur.

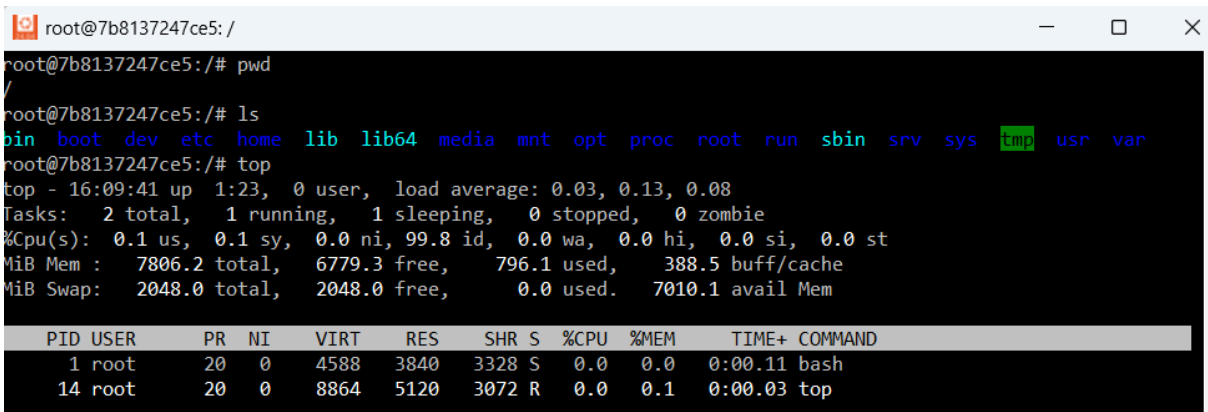
Option -t : permet de saisir du texte (des commandes) dans une console



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



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



- Créer un fichier, vérifier sa présence :

```

root@7b8137247ce5: /
root@7b8137247ce5:/# touch fichier_test_persistence
root@7b8137247ce5:/# ls
bin  dev  fichier_test_persistence  lib  media  opt  root  sbin  sys  usr
boot etc  home  lib64  mnt  proc  run  srv  tmp  var
root@7b8137247ce5:/#

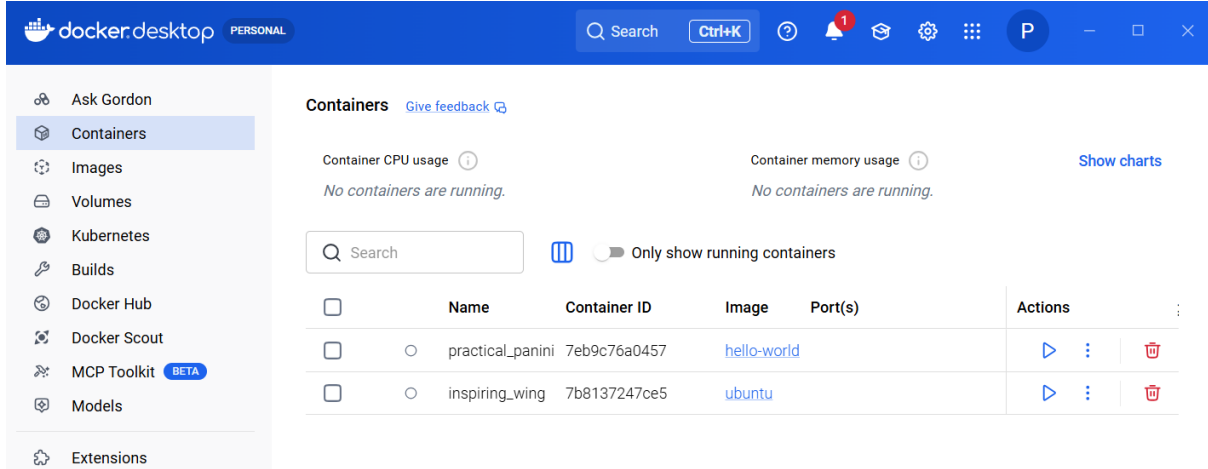
```

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

```

sio@PC-01: ~
root@7b8137247ce5:/# rm -fr /home
root@7b8137247ce5:/# ls
bin  dev  fichier_test_persistence  lib64  mnt  proc  run  srv  tmp  var
boot etc  lib  media  opt  root  sbin  sys  usr
root@7b8137247ce5:/# exit
exit
sio@PC-01:~$

```



```

Windows PowerShell
PS C:\Users\phbou> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
PS C:\Users\phbou>

```

```

Windows PowerShell
PS C:\Users\phbou> docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED        STATUS              PORTS   NAMES
7b8137247ce5  ubuntu   "bash"    25 minutes ago Exited (0) 8 minutes ago
7eb9c76a0457  hello-world "/hello"   4 days ago     Exited (0) 4 days ago
PS C:\Users\phbou>

```

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

```

sio@PC-01: ~
sio@PC-01:~$ docker start 7b8
7b8
sio@PC-01:~$ docker stop 7b8
7b8
sio@PC-01:~$ docker start -ai 7b8
root@7b8137247ce5:/# ls
bin  dev  fichier_test_persistence  lib64  mnt  proc  run  srv  tmp  var
boot etc  lib  media  opt  root  sbin  sys  usr
root@7b8137247ce5:/# exit
exit
sio@PC-01:~$

```

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

- Autre manière d'accéder au conteneur :

```

sio@PC-01: ~
sio@PC-01:~$ docker start 7b8
7b8
sio@PC-01:~$ docker exec -it 7b8 bash
root@7b8137247ce5:/# exit
exit
sio@PC-01:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
7b8137247ce5   ubuntu   "bash"   About an hour ago   Up 19 seconds   7b8137247ce5   inspiring_wing
sio@PC-01:~$ docker stop 7b8
7b8
sio@PC-01:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
7b8137247ce5   ubuntu   "bash"   About an hour ago   Exited (137) 11 seconds ago   7b8137247ce5   inspiring_wing
7eb9c76a0457   hello-world   "/hello"   4 days ago   Exited (0) 4 days ago   7eb9c76a0457   practical_panini
sio@PC-01:~$

```

- 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).

```

sio@PC-01: ~
sio@PC-01:~$ docker rm inspiring_wing
inspiring_wing
sio@PC-01:~$ docker rm 7e
7e
sio@PC-01:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
sio@PC-01:~$

```

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

```

sio@PC-01: ~
sio@PC-01:~$ docker run -it --rm ubuntu bash
root@ccbc1a572c9e:/# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@ccbc1a572c9e:/# exit
exit
sio@PC-01:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
sio@PC-01:~$

```

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

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

- Lancer un conteneur sur la base d’une image Ubuntu :

```

root@5c0b7bedf403: /
sio@PC-01:~$ docker run -it --name test_modification ubuntu
root@5c0b7bedf403:/#

```

- Procéder à quelques modifications et stopper l’exécution du conteneur :

```
sio@PC-01: ~  
root@5c0b7bedf403:/# cd home  
root@5c0b7bedf403:/home# touch fichier_test  
root@5c0b7bedf403:/home# cd ..  
root@5c0b7bedf403:/# ls  
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var  
root@5c0b7bedf403:/# cd lib  
root@5c0b7bedf403:/lib# ls  
apt dpkg init locale lsb mime os-release sysctl.d systemd tmpfiles.d udev x86_64-linux-gnu  
root@5c0b7bedf403:/lib# rm os-release  
root@5c0b7bedf403:/lib# exit  
exit  
sio@PC-01:~$
```

Commande docker diff pour voir les modifications :

```
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:~$
```

Commande docker commit --help pour prendre connaissance de la syntaxe de la commande docker commit :

```
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:~$
```

Créer l'image ubuntu-modif taguée 1.0 à partir du container test_modification :

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

Commande docker images pour lister les images présentes en local :

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

- Supprimer le conteneur :

```
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:~$
```

- Lancer un nouveau conteneur à partir de l'image créée avec la commande docker commit et vérifier que les modifications soient bien présentes :

```
sio@PC-01: ~  
sio@PC-01:~$ docker run -it --rm --name test ubuntu: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 : créer un répertoire web dans /home/sio de la machine ubuntu ainsi qu'une page html :

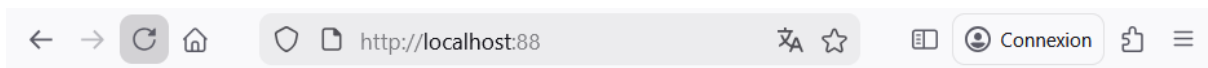
```
sio@PC-01: ~/web  
sio@PC-01:~$ pwd  
/home/sio  
sio@PC-01:~$ mkdir web  
sio@PC-01:~$ cd web  
sio@PC-01:~/web$ nano index.html
```

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

- Créer un conteneur qui va pointer sur le répertoire web du FS local accessible en lecture :
 - ➔ 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: sha256:0236ee02dbcce00b9bd83e0f5fbc51069e7e1161bd59d99885b3ae1734f3392e  
Status: Downloaded newer image for nginx:latest  
56f3c387f4b09e08565cf5509ef5f57bf518f96984440f794f8ed3a4d31fda15  
sio@PC-01:~$ docker ps  
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS  
56f3c387f4b0   nginx    "/docker-entrypoint...." 3 minutes ago  Up 3 minutes  0.0.0.0:88->80/tcp, [:::  
]:88->80/tcp  
sio@PC-01:~$
```

- Tester l'accès aux données :

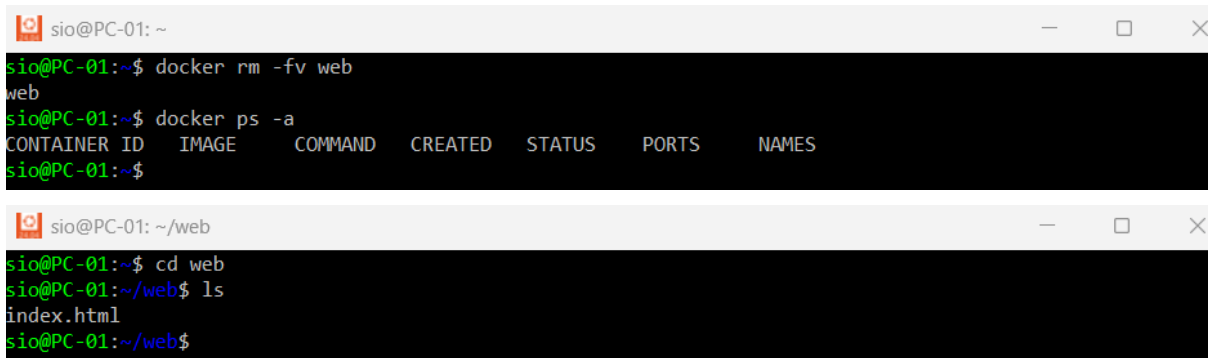


Page web Nginx conteneur Docker

- Afficher les logs du conteneur web :

```
sio@PC-01: ~  
sio@PC-01:~$ docker logs web  
/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  
/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  
2026/03/01 08:52:36 [notice] 1#1: using the "epoll" event method  
2026/03/01 08:52:36 [notice] 1#1: nginx/1.29.5  
2026/03/01 08:52:36 [notice] 1#1: built by gcc 14.2.0 (Debian 14.2.0-19)  
2026/03/01 08:52:36 [notice] 1#1: OS: Linux 6.6.87.2-microsoft-standard-WSL2  
2026/03/01 08:52:36 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576  
2026/03/01 08:52:36 [notice] 1#1: start worker processes  
2026/03/01 08:52:36 [notice] 1#1: start worker process 29  
2026/03/01 08:52:36 [notice] 1#1: start worker process 30  
2026/03/01 08:52:36 [notice] 1#1: start worker process 31  
2026/03/01 08:52:36 [notice] 1#1: start worker process 32  
2026/03/01 08:52:36 [notice] 1#1: start worker process 33  
2026/03/01 08:52:36 [notice] 1#1: start worker process 34  
2026/03/01 08:52:36 [notice] 1#1: start worker process 35  
2026/03/01 08:52:36 [notice] 1#1: start worker process 36  
172.17.0.1 - - [01/Mar/2026:09:00:04 +0000] "GET / HTTP/1.1" 200 72 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:148.0) Gecko/20100101 Firefox/148.0" "-"  
2026/03/01 09:00:04 [error] 29#29: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "localhost:88", referer: "http://localhost:88/"  
172.17.0.1 - - [01/Mar/2026:09:00:04 +0000] "GET /favicon.ico HTTP/1.1" 404 153 "http://localhost:88/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:148.0) Gecko/20100101 Firefox/148.0" "-"  
172.17.0.1 - - [01/Mar/2026:09:02:06 +0000] "GET / HTTP/1.1" 200 73 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:148.0) Gecko/20100101 Firefox/148.0" "-"  
sio@PC-01:~$
```

- Arrêter et supprimer le conteneur web ainsi que le mappage de volume :



```
sio@PC-01: ~  
sio@PC-01:~$ docker rm -fv web  
web  
sio@PC-01:~$ docker ps -a  
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES  
sio@PC-01:~$  
  
sio@PC-01: ~/web  
sio@PC-01:~$ cd web  
sio@PC-01:~/web$ ls  
index.html  
sio@PC-01:~/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 (ce n'est pas un 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.

- Récupération de l'image **mongodb-community-server** avec la commande **docker pull** :

```
sio@PC-01: ~/data
sio@PC-01:~/data$ docker pull mongodb/mongodb-community-server:latest
latest: Pulling from mongodb/mongodb-community-server
205e383a236b: Pull complete
cf2e29b962cf: Pull complete
b1cba2e842ca: Pull complete
ead3665aa3ea: Pull complete
4f4fb700ef54: Pull complete
02edc6839712: Pull complete
d976eb8a1ba6: Pull complete
ec799c9dbd8d: Pull complete
42b29d63c5b3: Pull complete
378daa318251: Pull complete
65d16eb74452: Pull complete
656c3f0eba3b: Pull complete
bc9122fc1cf1: Pull complete
ac99d5d00249: Pull complete
15805c58ed00: Pull complete
b915da17874e: Pull complete
759d96211a46: Pull complete
4d786b9666b8: Pull complete
f002c022af1d: Pull complete
e379ca8ed9f5: Pull complete
e4f7bfc6c22a: Pull complete
Digest: sha256:9864d506155791d975d417d6e0d8147c9470ed493862731420b0abae1c216949
Status: Downloaded newer image for mongodb/mongodb-community-server:latest
docker.io/mongodb/mongodb-community-server:latest
sio@PC-01:~/data$
```

- Lancer un conteneur en mode serveur à partir de l'image **mongodb-community-server** :

```
sio@PC-01: ~/data
sio@PC-01:~/data$ docker run -d -p 27017:27017 -e MONGODB_INITDB_ROOT_USERNAME=sio -e MONGODB_INITDB_ROOT_PASSWORD=password1234 --name mongodb mongodb/mongodb-community-server:latest
ba989cff4db83791507e15ee984a88cbd00e93575bc9067e4a70ad9f89fcfbda
sio@PC-01:~/data$
```

La commande ci-dessus prend l'image téléchargée et la déploie dans un conteneur nommé **mongodb**. Le conteneur est mappé sur un port ce qui signifie que la machine physique peut interagir avec lui sur le port 27017.

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

The screenshot shows a web browser window with the URL `www.mongodb.com/try/download/shell`. The page features a sidebar on the left with navigation links: MongoDB Atlas, MongoDB Enterprise Advanced, MongoDB Community Edition, Tools, MongoDB Atlas Terraform Provider, **MongoDB Shell** (highlighted), MongoDB Compass (GUI), Atlas CLI, Atlas Kubernetes Operator, and MongoDB CLI for Cloud Manager and Ops Manager. The main content area includes a note: "Note: MongoDB Shell is an open source (Apache 2.0), standalone product developed separately from the MongoDB Server." Below this is a "Learn more" section with three dropdown menus: "Version" set to "2.7.0", "Platform" set to "Windows x64 (10+)", and "Package" set to "msi". At the bottom of the main content area are two buttons: a green "Download" button with a download icon and a "Copy link" button with a copy icon.

- Connectez-vous au serveur MongoDB :

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

```
Current Mongosh Log ID: 69a48449e234379fa77c2906
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
2026-03-01T17:10:16.909+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
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 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('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 la base de données, les collections et les documents, ne sont conservées que tant que le conteneur concerné est en cours d'exécution.

```
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/7221ea7fe2bc45d46d3fdd7eb7950e460707c2bec6de4c39d7e143a0625d3097/_data",
        "Destination": "/data/configdb",
        "Driver": "local",
        "Mode": "",
        "RW": true,
        "Propagation": ""
      },
      {
        "Type": "volume",
        "Name": "027081e92124d73dc9b8112df4159c35dcb3395258783f78c88017858e8b2539",
        "Source": "/var/lib/docker/volumes/027081e92124d73dc9b8112df4159c35dcb3395258783f78c88017858e8b2539/_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": {}
    }
  },
],

```

- Accédez au conteneur puis aux données et arrêter le conteneur :

```

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:~$

```

- Supprimez le 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.

- Lancez un conteneur avec un mappage de volume (volume nommé:/volume du conteneur) :

```
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  
66a434bcf6f72f28e0d8f2ffdf69718ed7fb51c72786a13c1b93633c6913214  
sio@PC-01:~$
```

- Connectez-vous au serveur MongoDB :

```
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        STATUS
PORTS
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
```

- Arrêtez le conteneur et supprimez-le :

```
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, redémarrez 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  
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/?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  
2026-03-01T23:38:56.990+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem  
2026-03-01T23:38:57.775+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile  
2026-03-01T23:38:57.775+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile  
2026-03-01T23:38:57.775+00:00: We suggest setting the contents of sysfsFile to 0.  
2026-03-01T23:38:57.776+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'  
  }  
]  
ma_bdd> exit  
PS C:\Users\phbou> |
```

- Supprimez le conteneur :

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

- Relancez un conteneur et 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  
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> |

```

- Supprimez le conteneur :

```

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

```

- Relancez un conteneur et constatez la persistance des données :

```

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
24c3205cd07226c3903a82062db2b084ff2dd2d69df6ab66982d528c83cd2a4c
sio@PC-01:~$

```

```
Windows PowerShell
PS C:\Users\phbou> mongosh mongodb://sio:password1234@localhost:27017
Current Mongosh Log ID: 69a4d692b4d43e82a47c2906
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

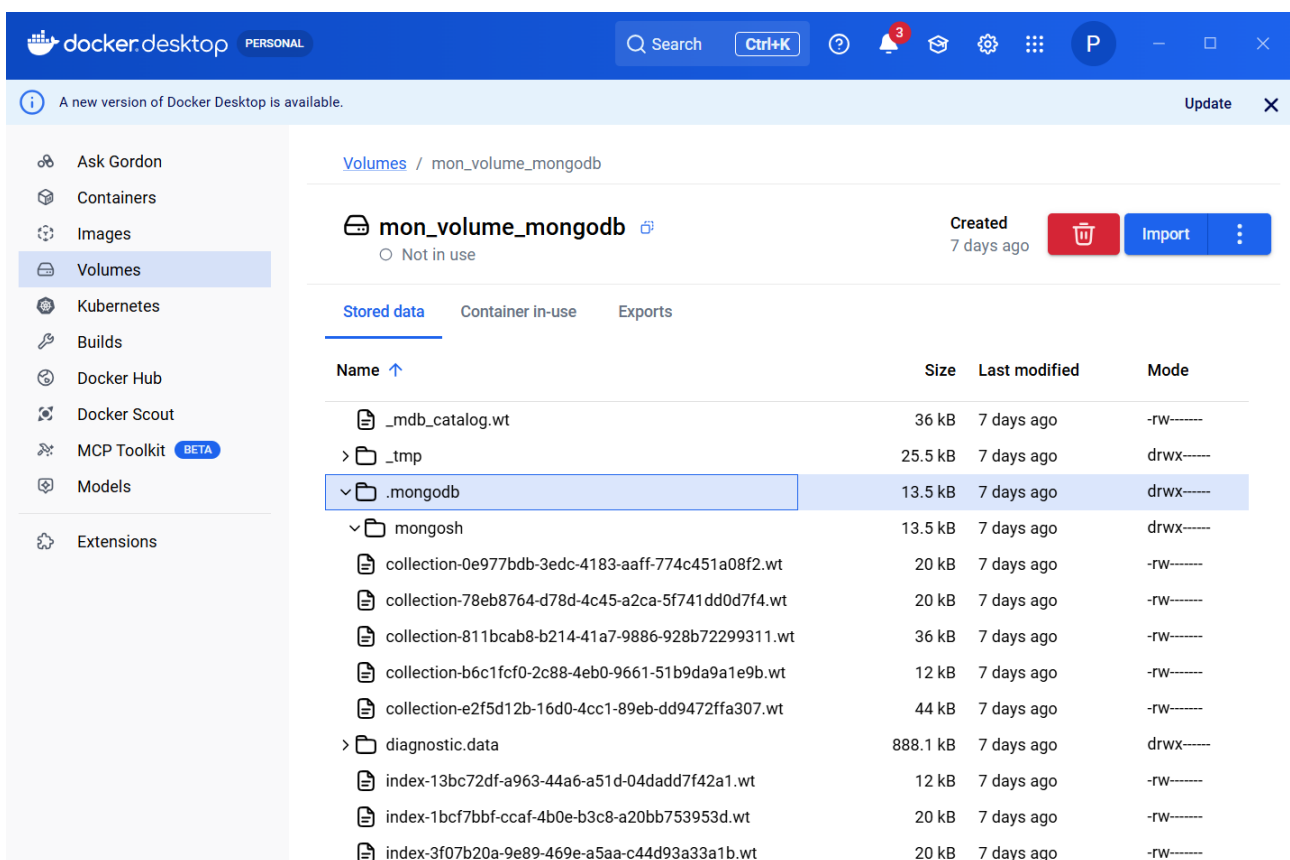
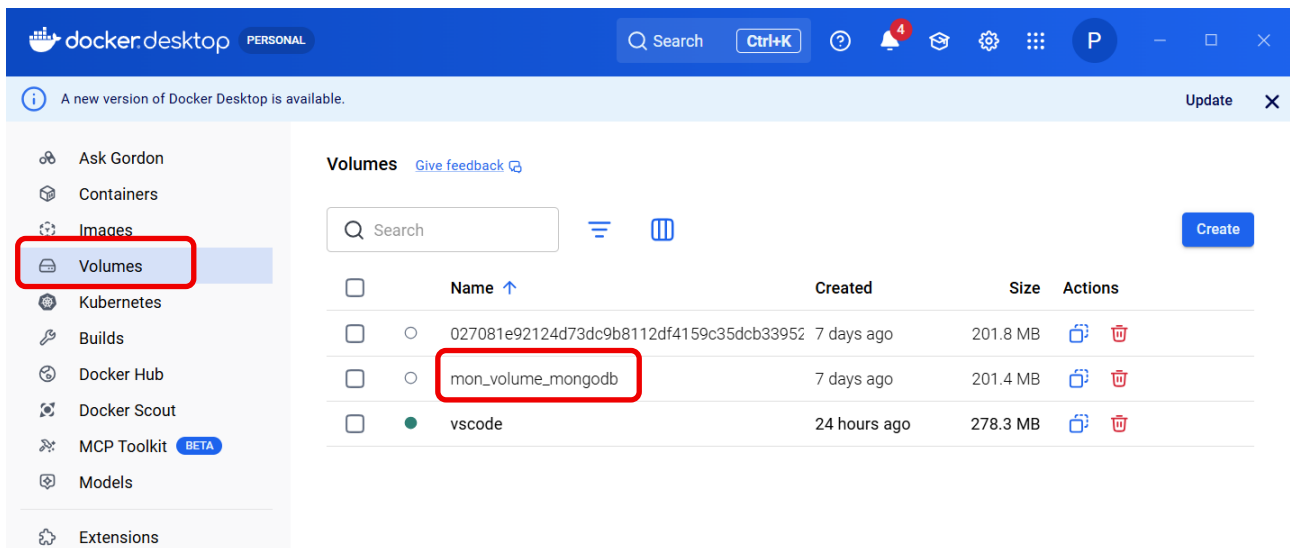
For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

-----
The server generated these startup warnings when booting
2026-03-02T00:14:35.592+00:00: Using the XFS filesystem is strongly recom
mended with the WiredTiger storage engine. See http://dochub.mongodb.org/cor
e/prodnotes-filesystem
2026-03-02T00:14:36.473+00:00: For customers running the current memory a
llocator, we suggest changing the contents of the following sysfsFile
2026-03-02T00:14:36.474+00:00: For customers running the current memory a
llocator, we suggest changing the contents of the following sysfsFile
2026-03-02T00:14:36.474+00:00: We suggest setting the contents of sysfsFi
le to 0.
2026-03-02T00:14:36.474+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'
  },
  {
    _id: ObjectId('69a4d5d62bc0d675887c2907'),
    nom: 'hallyday',
    prenom: 'johnny'
  }
]
ma_bdd> exit
PS C:\Users\phbou>
```

- Supprimez le conteneur :

```
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.



```
Terminal 1:
sio@PC-01: ~
sio@PC-01:~$ mkdir data
sio@PC-01:~$

Terminal 2:
sio@PC-01: ~
sio@PC-01:~$ docker run -d --name mongodb2 -p 27017:27017 -e MONGODB_INITDB_ROOT_USERNAME=sio -e MONGODB_INITDB_ROOT_PASSWORD=password1234 -v /home/sio/data:/data/db mongodb/mongodb-community-server:latest
c2a9305536c5dbc0e3b97267ec02f04a811aea119fc65fa251c8de993bb05ed1
sio@PC-01:~$
```

```
Windows PowerShell
PS C:\Users\phbou> mongosh mongodb://sio:password1234@localhost:27017
Current Mongosh Log ID: 69ad7d57179576e9b77c2906
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
2026-03-08T13:38:06.451+00:00: Using the XFS filesystem is strongly recom
mended with the WiredTiger storage engine. See http://dochub.mongodb.org/cor
e/prodnotes-filesystem
2026-03-08T13:38:07.057+00:00: For customers running the current memory a
llocator, we suggest changing the contents of the following sysfsFile
2026-03-08T13:38:07.057+00:00: For customers running the current memory a
llocator, we suggest changing the contents of the following sysfsFile
2026-03-08T13:38:07.057+00:00: We suggest setting the contents of sysfsFi
le to 0.
2026-03-08T13:38:07.057+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.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 minu
tes 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
```

```

sio@PC-01: ~/data
sio@PC-01:~$ cd data
sio@PC-01:~/data$ ls
WiredTiger                               diagnostic.data
WiredTiger.lock                          index-2d132e73-3dd2-45b2-8391-5925dde7a97e.wt
WiredTiger.turtle                        index-33e329bd-509c-417e-a9fa-6ca276f40162.wt
WiredTiger.wt                             index-84cabbba-2036-4de5-867c-c5c9b9e34557.wt
WiredTigerHS.wt                          index-96656ee8-a642-40ee-8fc8-e03197bc8617.wt
_mdb_catalog.wt                          index-c403af3f-d17c-464a-ad8d-a2ad9a09fceb.wt
_tmp                                       index-dc0e9b87-2357-4884-a901-c79ef5bfaf83.wt
collection-60f1796f-99b0-460c-8963-1f5b1e41aba4.wt index-e58e2f62-d8cb-40a5-8611-28aff5de676c.wt
collection-682dd6b6-9119-442a-abbd-3f5e6b4b8f96.wt journal
collection-9540571f-26e6-48d6-b720-cee8a660467c.wt mongod.lock
collection-b4e0d1ea-d52b-44d3-8486-674bb616c398.wt sizeStorer.wt
collection-ddc31ec6-0318-45c6-ba84-ad8ae75e5800.wt storage.bson
sio@PC-01:~/data$

```

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

```

sio@PC-01: ~
sio@PC-01:~$ docker run -d --name mongodb2 -p 27017:27017 -e MONGODB_INITDB_ROOT_USERNAME=sio -e MONGODB_INITDB_ROOT_PASSWORD=password1234 -v /home/sio/data:/data/db mongodb/mongod-community-server:latest
2e812022d73560279ce8a2f5edcac713bb45c6ed9a6e9cccbf89b183f69226af
sio@PC-01:~$

```

```

Windows PowerShell
PS C:\Users\phbou> mongosh mongodb://sio:password1234@localhost:27017
Current Mongosh Log ID: 69ad8687e64db38d6a7c2906
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

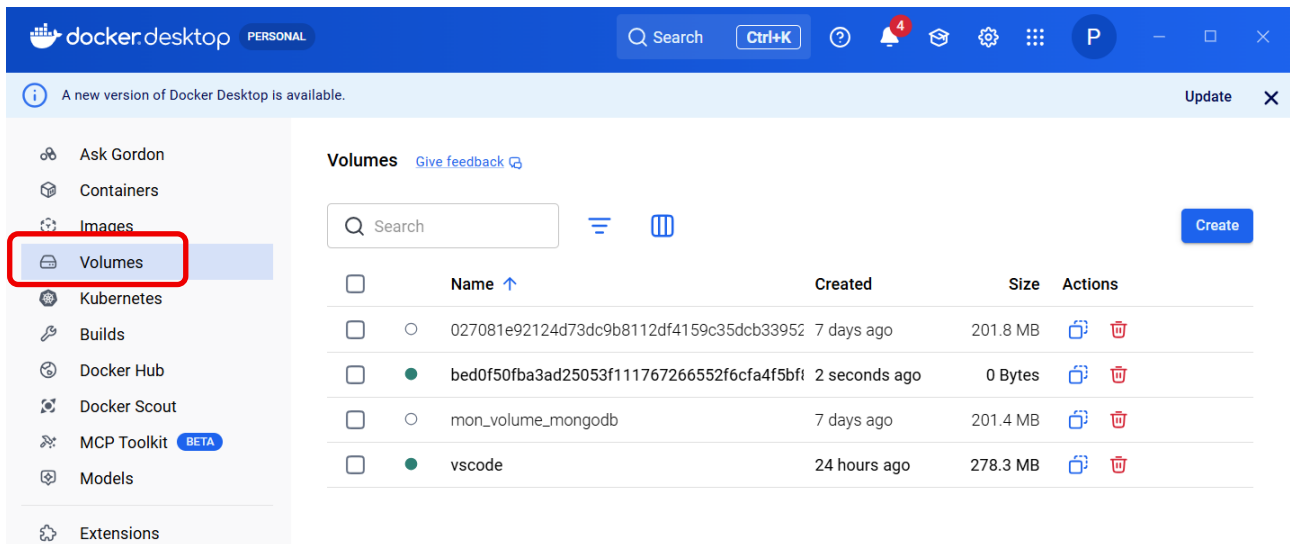
For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

-----
The server generated these startup warnings when booting
2026-03-08T14:20:18.427+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2026-03-08T14:20:19.362+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
2026-03-08T14:20:19.362+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
2026-03-08T14:20:19.362+00:00: We suggest setting the contents of sysfsFile to 0.
2026-03-08T14:20:19.363+00:00: We suggest setting swappiness to 0 or 1, as swapping can cause performance problems.
-----

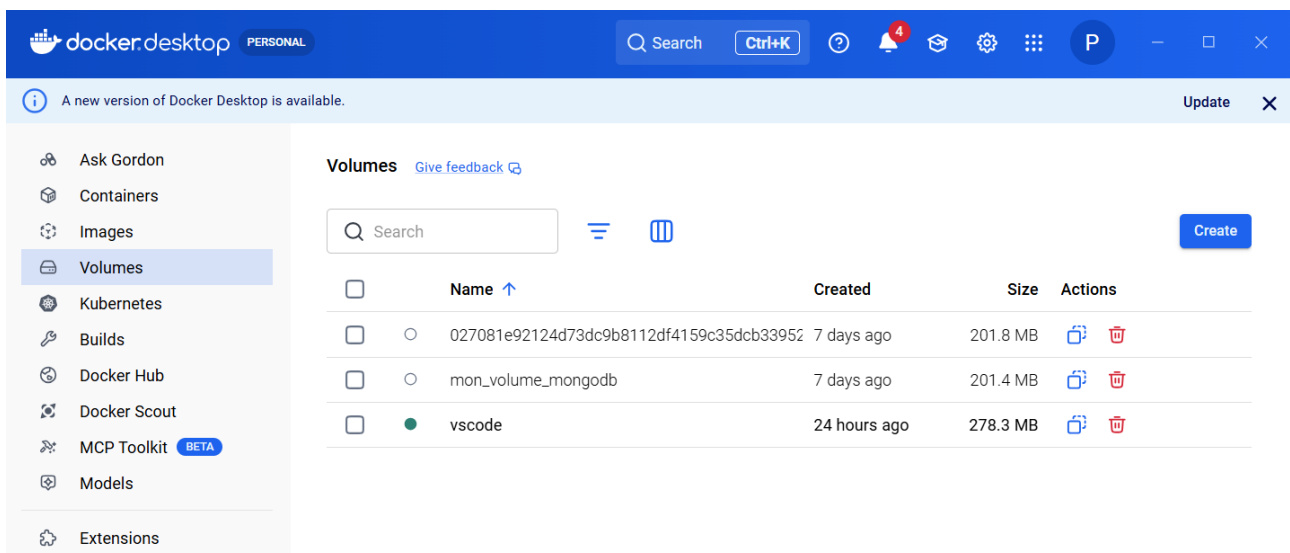
test> use("ma_bdd");
switched to db ma_bdd
ma_bdd> db.etudiants.find();
[
  {
    _id: ObjectId('69ad7e76179576e9b77c2907'),
    nom: 'mitchell',
    prenom: 'eddy'
  }
]
ma_bdd> exit
PS C:\Users\phbou>

```

La persistance a bien été gérée.



```
sio@PC-01: ~  
sio@PC-01:~$ docker rm -fv mongodb2  
mongodb2
```

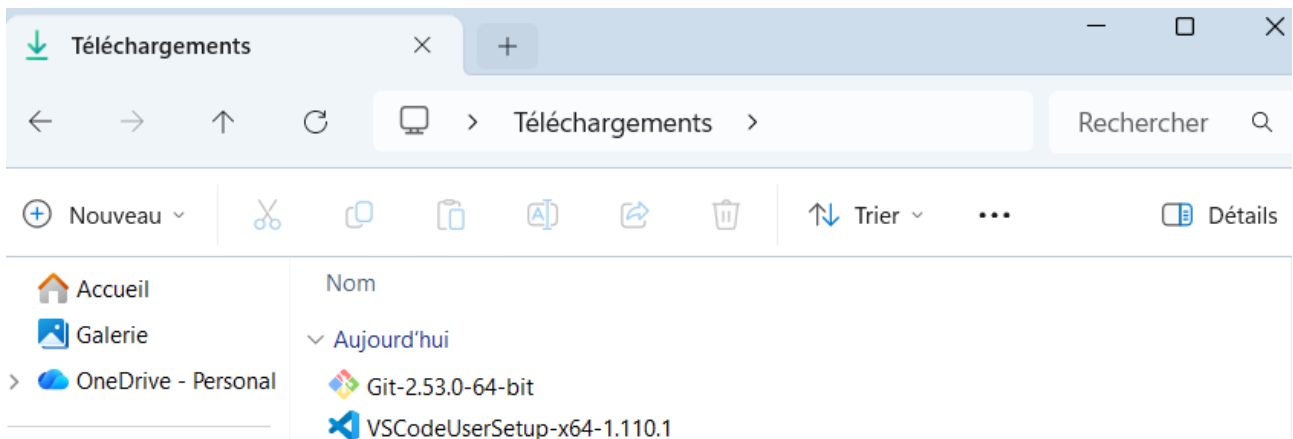


12. Utiliser un conteneur Docker et Visual Studio Code.

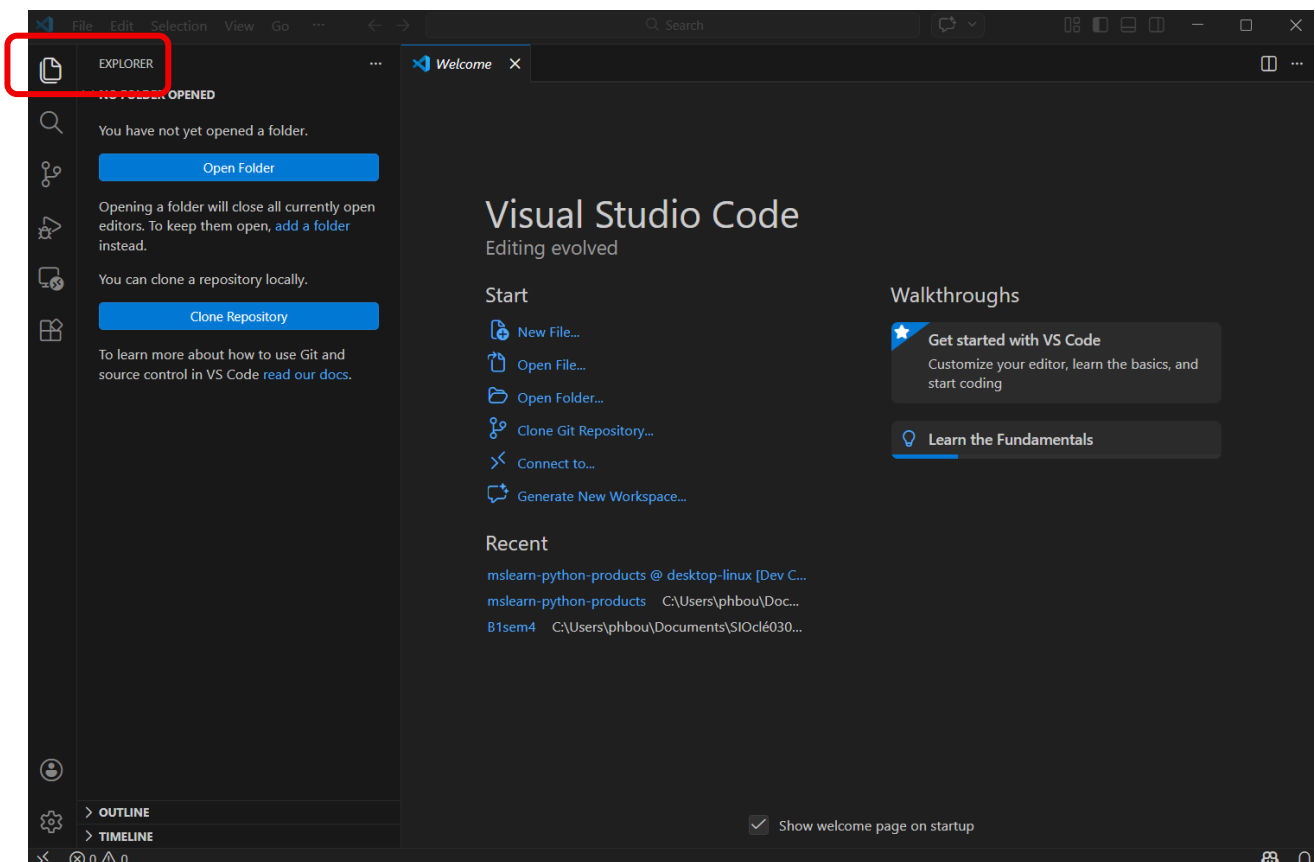
→ L'objectif est d'utiliser **l'extension Dev Containers** de Visual Studio Code pour développer au sein d'un conteneur Docker car les développeurs doivent pouvoir travailler sur n'importe quel projet sans avoir à installer ou configurer leurs machines au préalable.

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

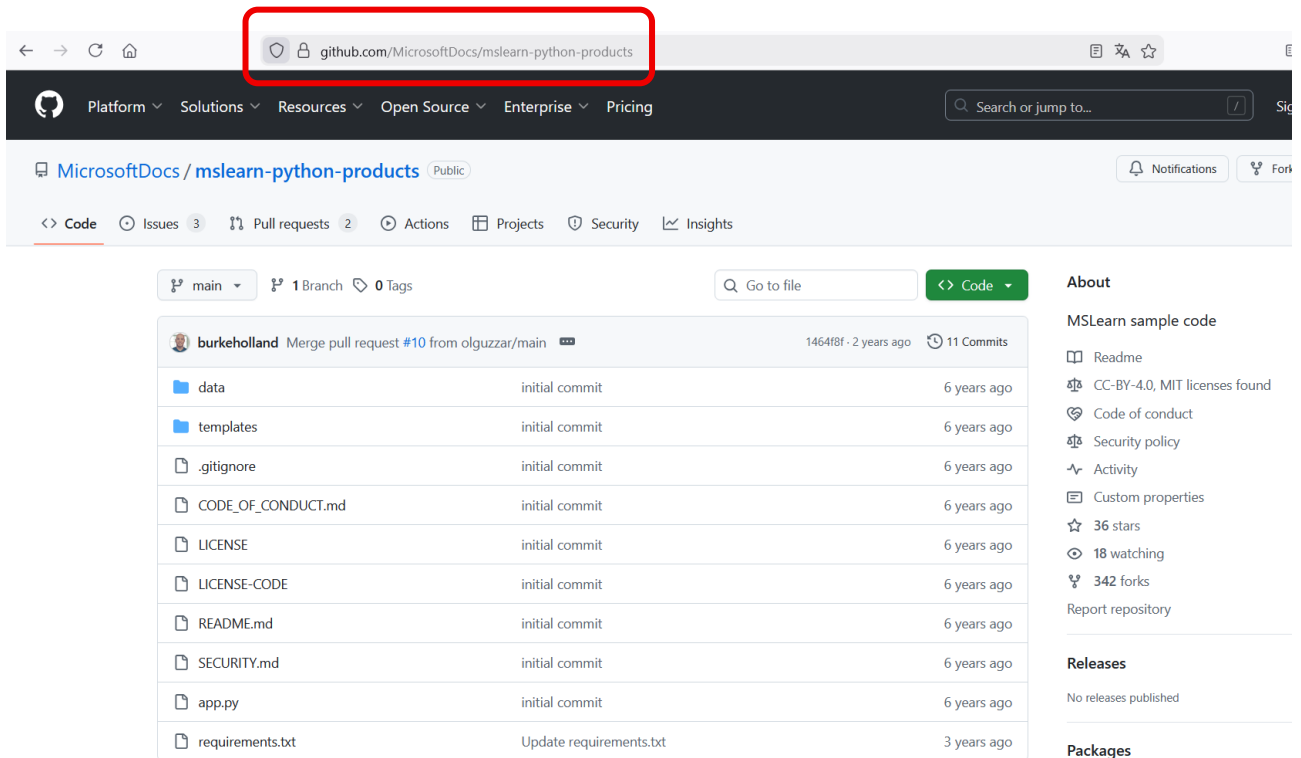
- Installez Visual Studio Code si ce n'est pas déjà fait :



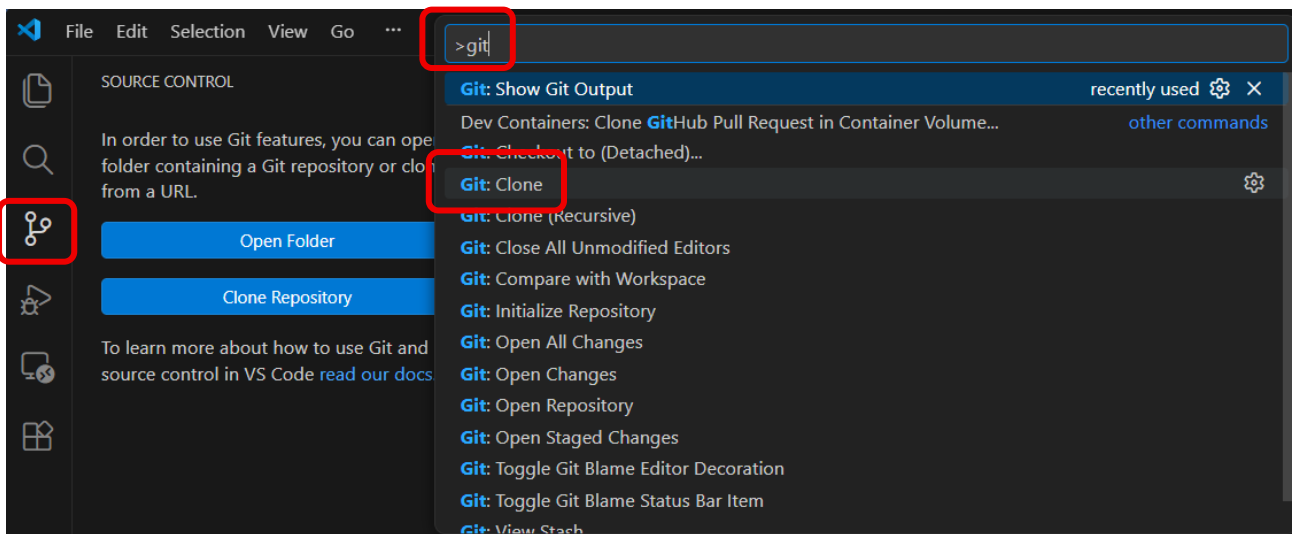
- Vérifiez que Docker Desktop soit bien en cours d'exécution sur votre ordinateur.
- Ouvrez une nouvelle instance de Visual Studio Code :



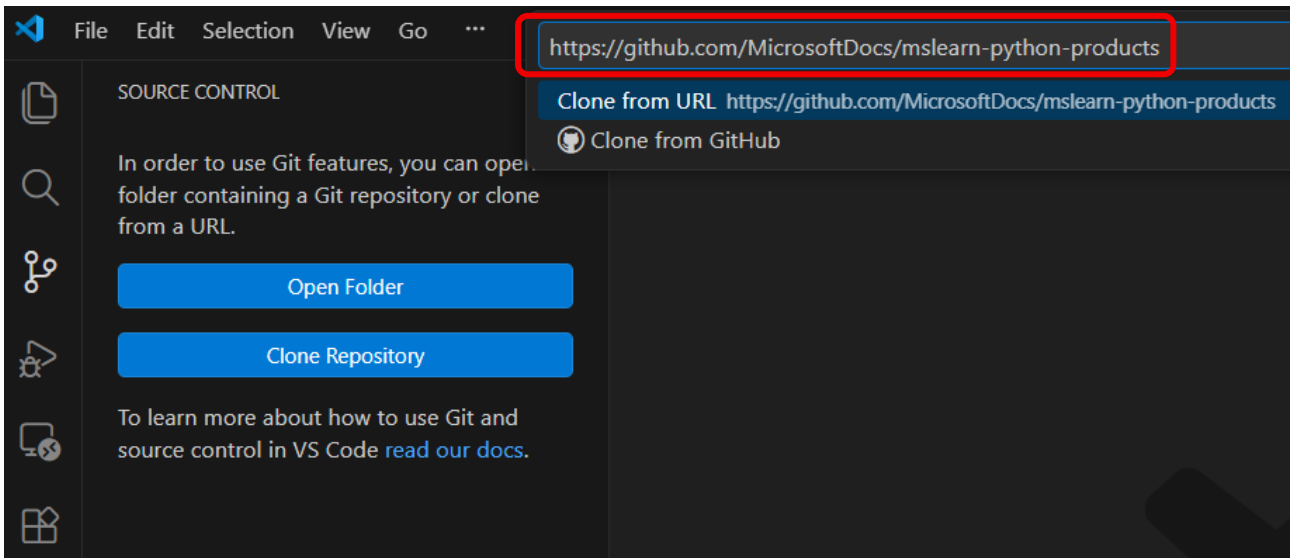
- Copiez l'URL du projet **mslearn-python-products** qui nous servira d'exemple :
<https://github.com/MicrosoftDocs/mslearn-python-products>



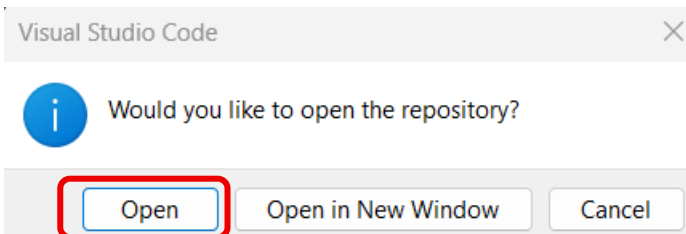
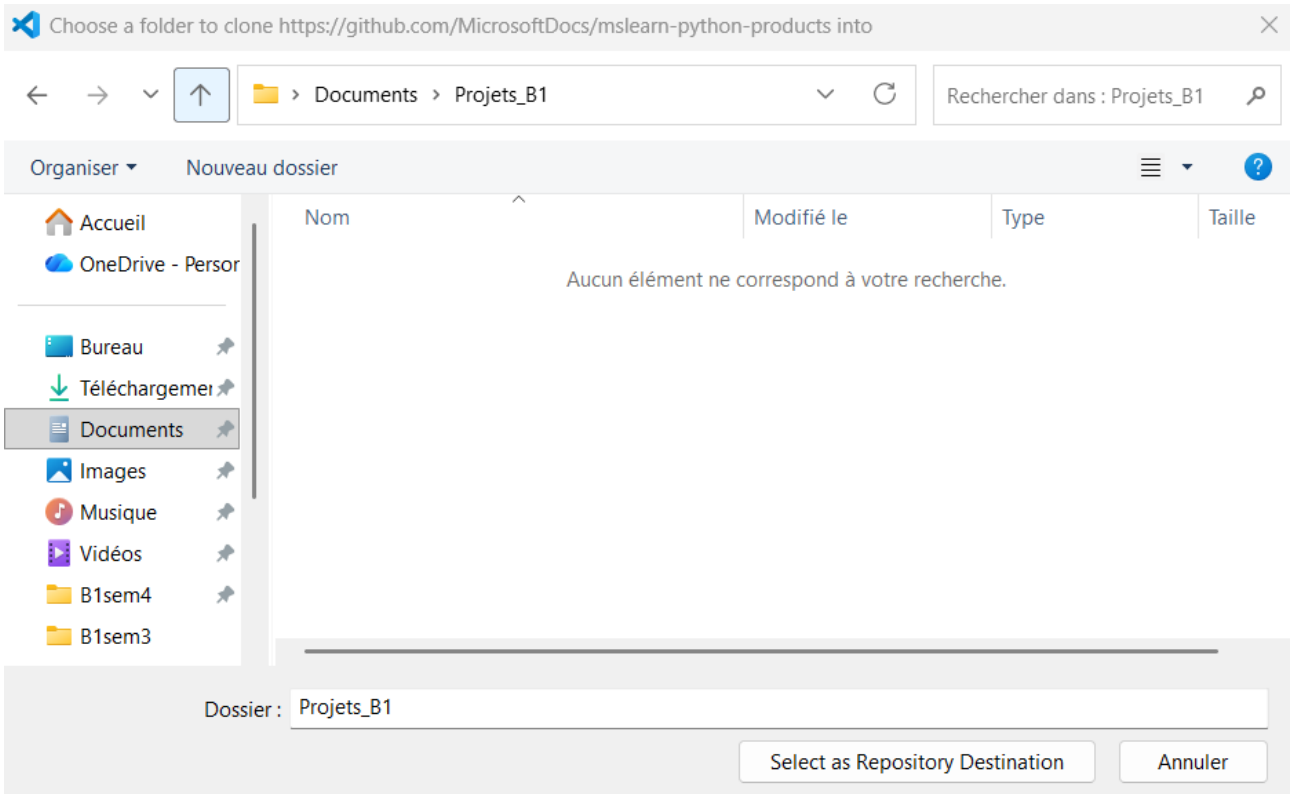
- Dans VS Code, sélectionnez **Clone Repository** ou appuyez sur F1 pour rechercher **Git** : **Clone** :

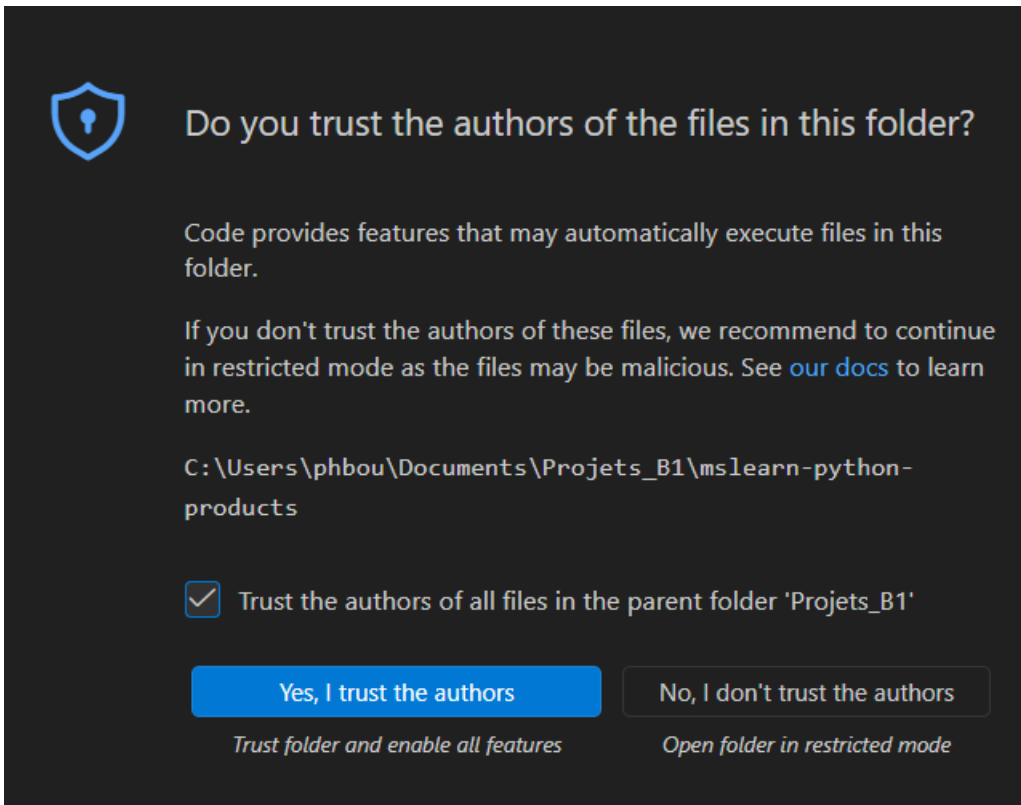


- Collez l'URL :

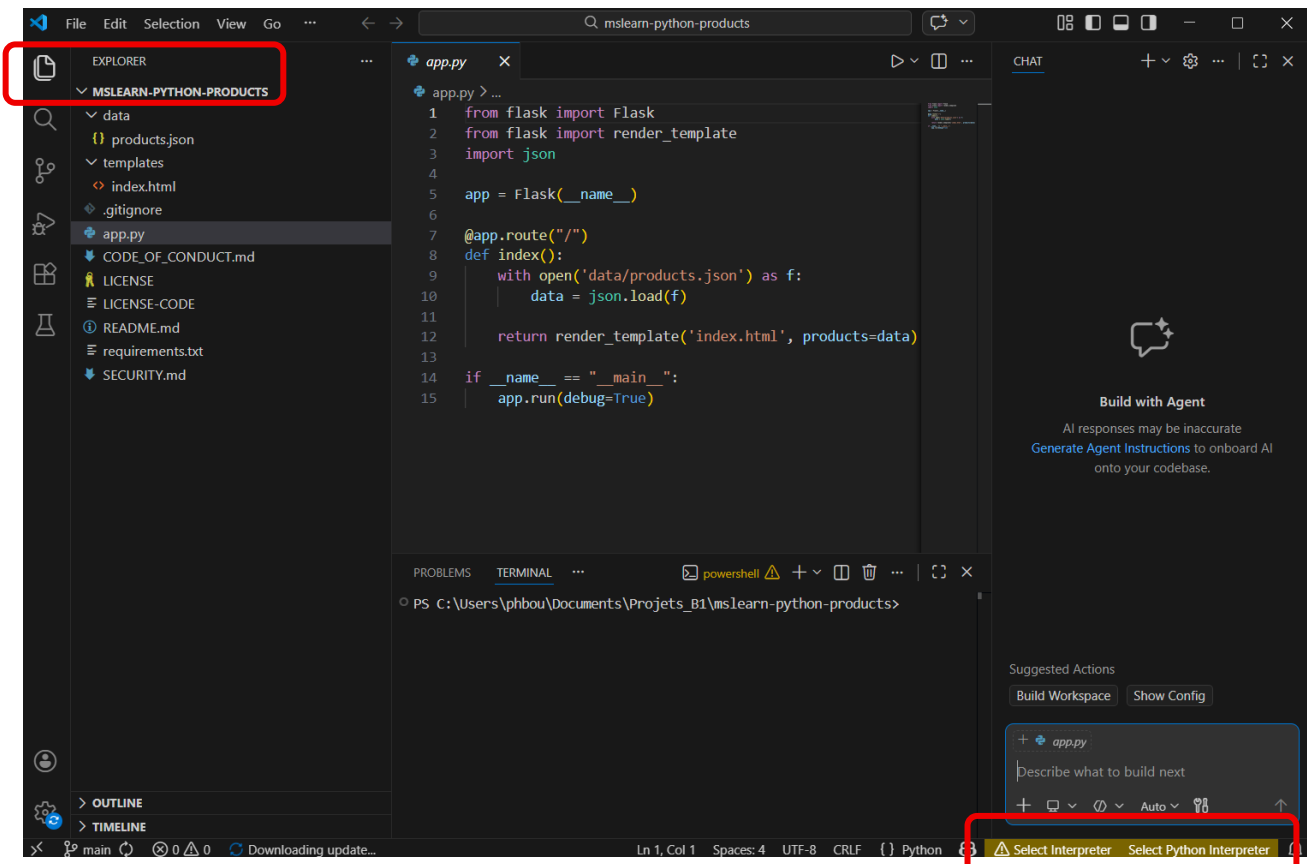


- Sélectionnez un emplacement sur votre disque où le projet va être cloné

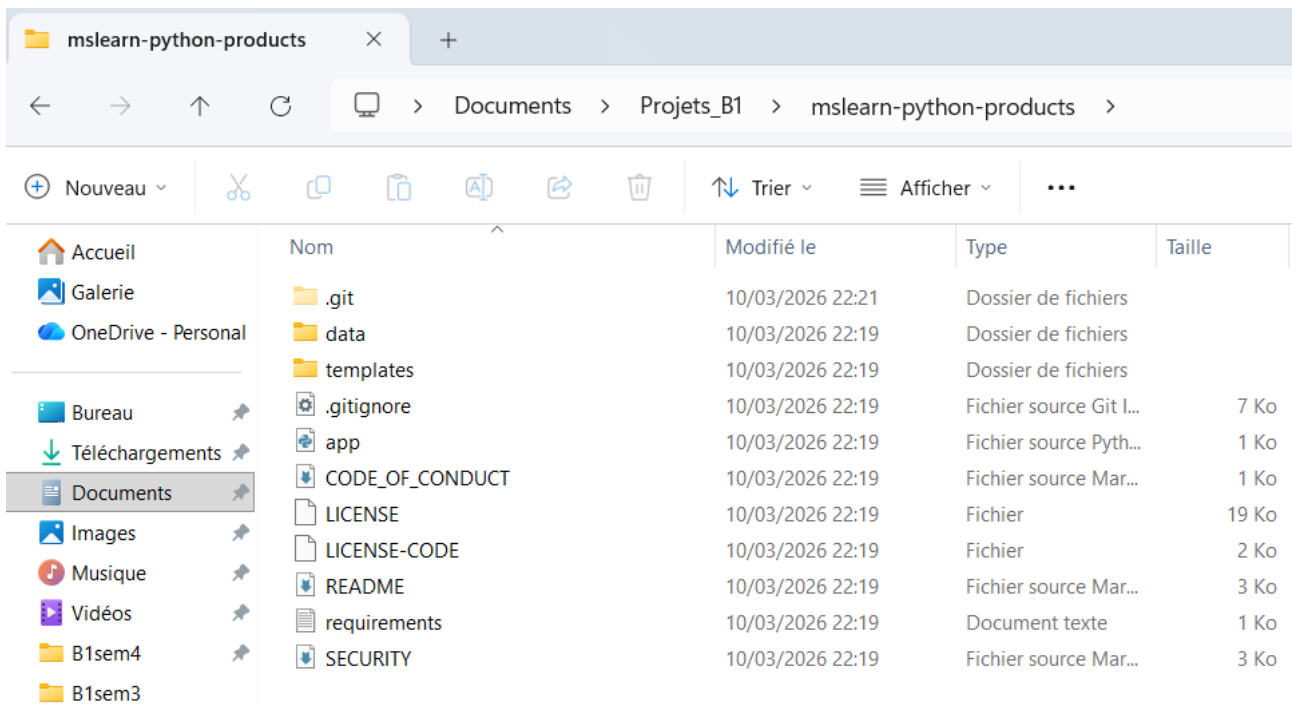




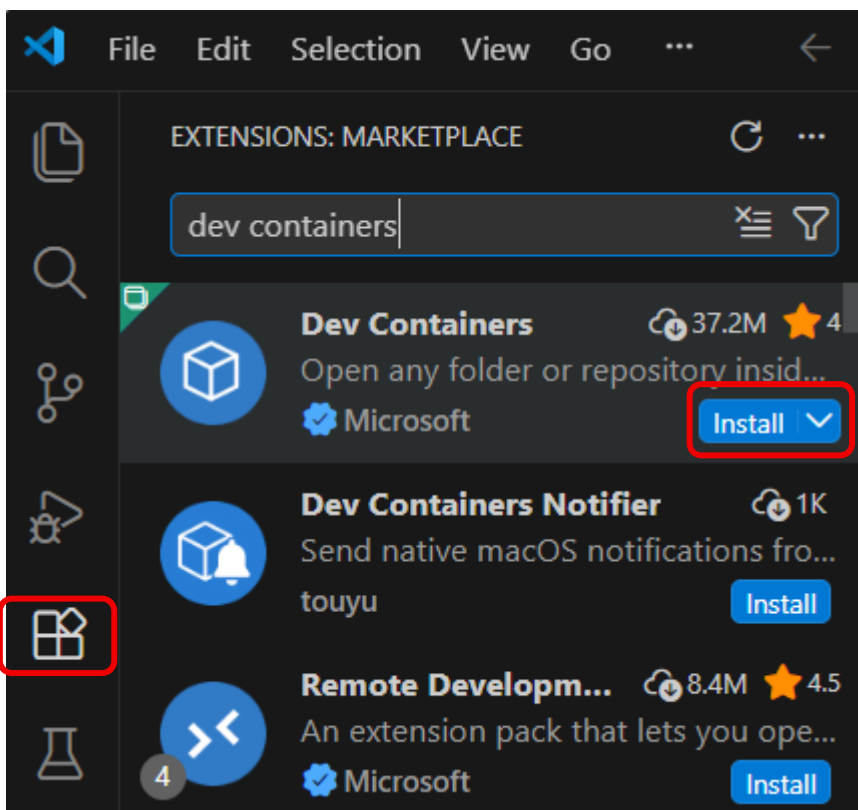
Le projet est maintenant ouvert dans Visual Studio Code :

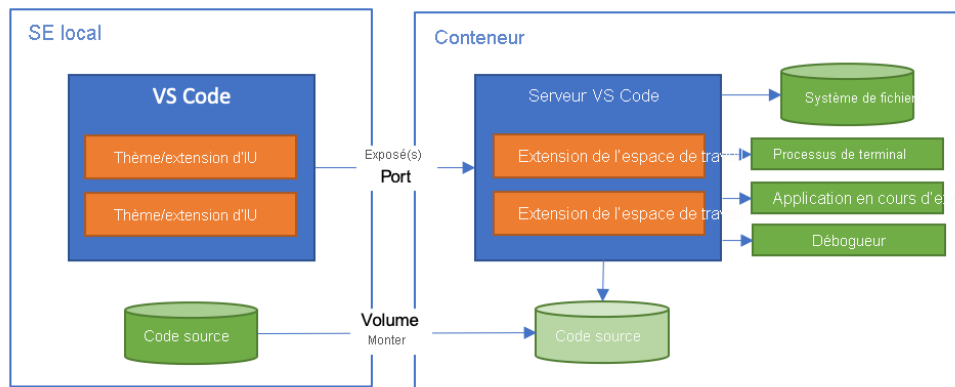
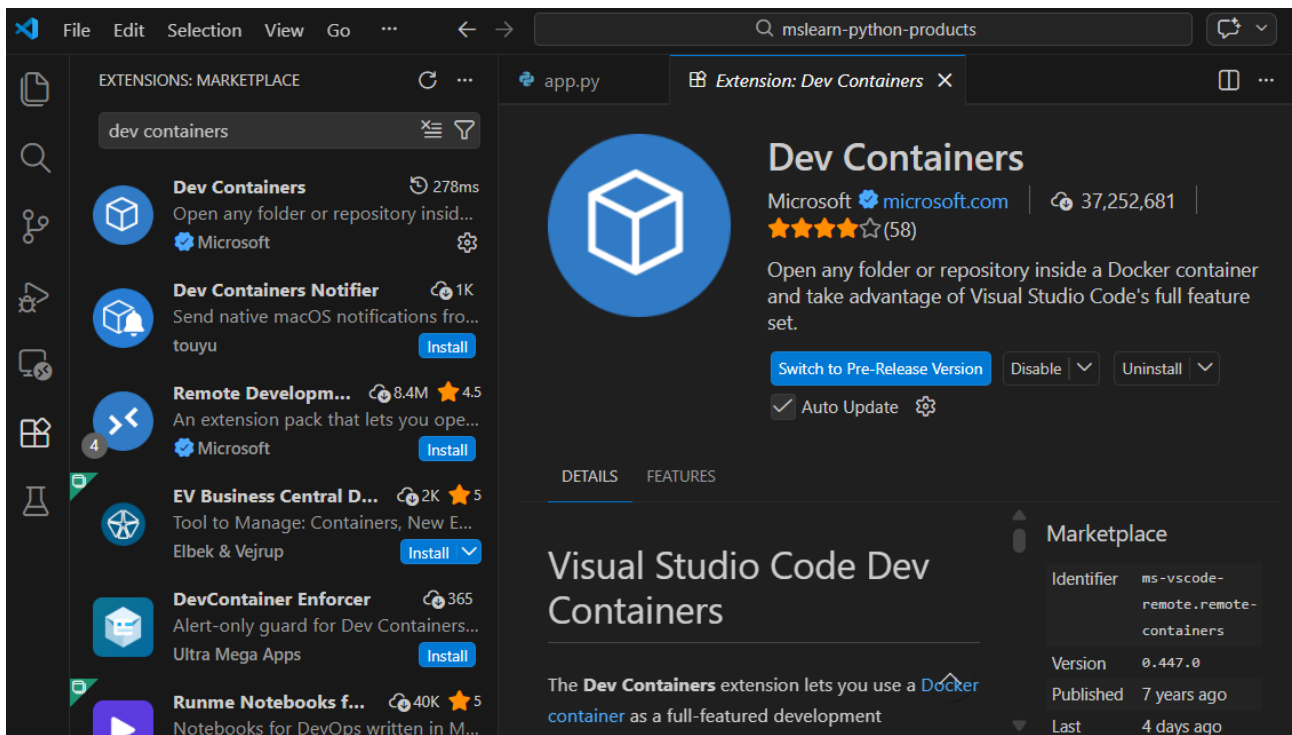


Vous pouvez ignorer l'avertissement sur la sélection d'un interpréteur Python.



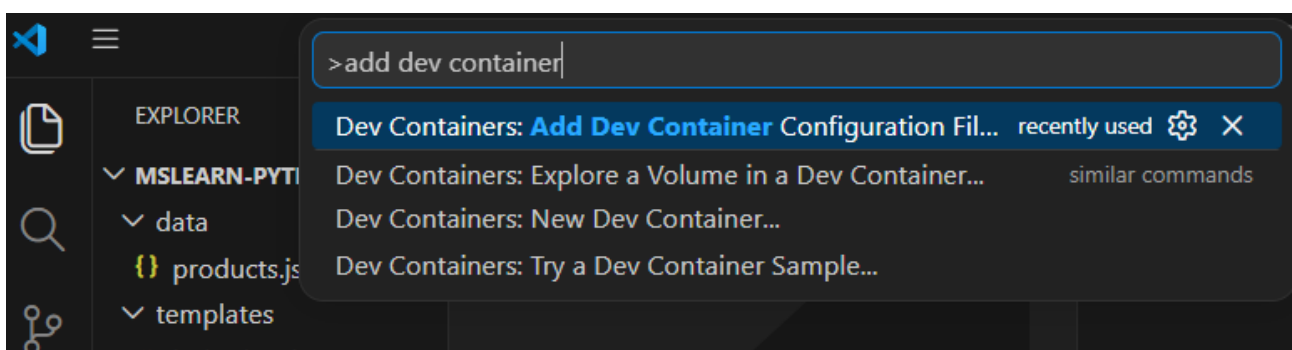
- Sélectionnez l'icône **Extensions** afin d'installer l'extension **Dev Containers** (« **conteneurs de développement** »).
- ➔ L'extension Dev Containers vous permettra d'exécuter Visual Studio Code à l'intérieur d'un conteneur Docker.
- ➔ Vous bénéficierez des commandes Dev Containers dans VS Code.





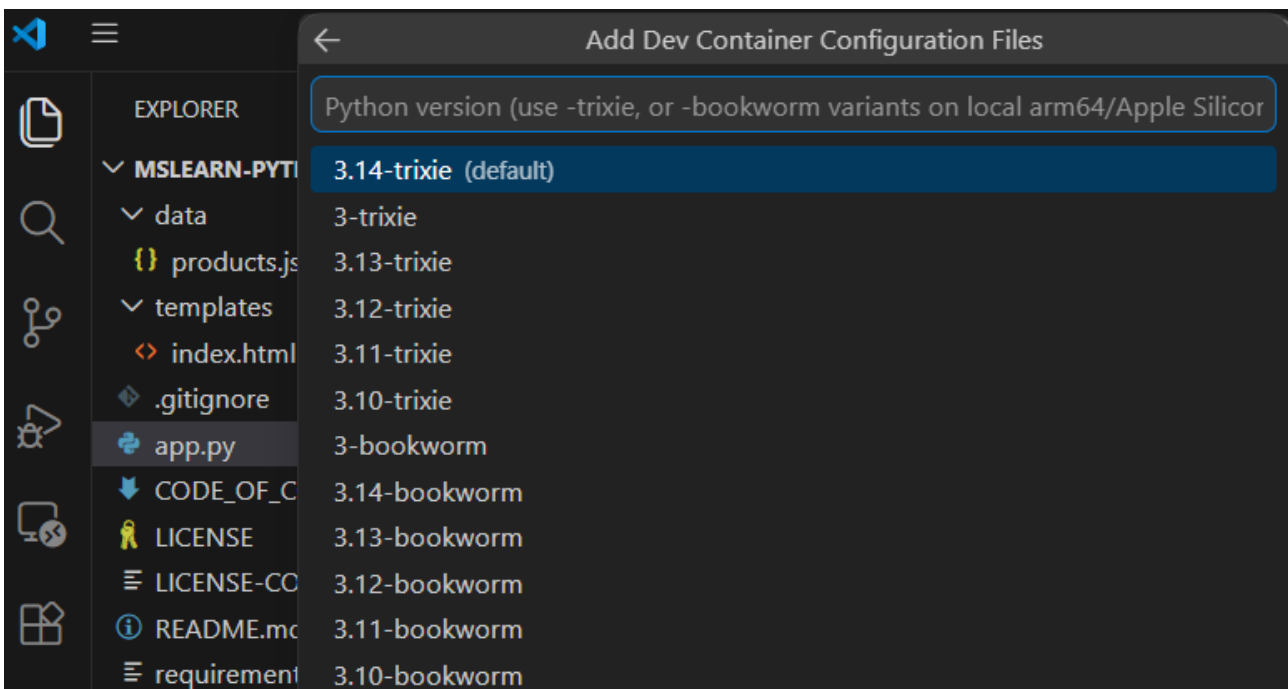
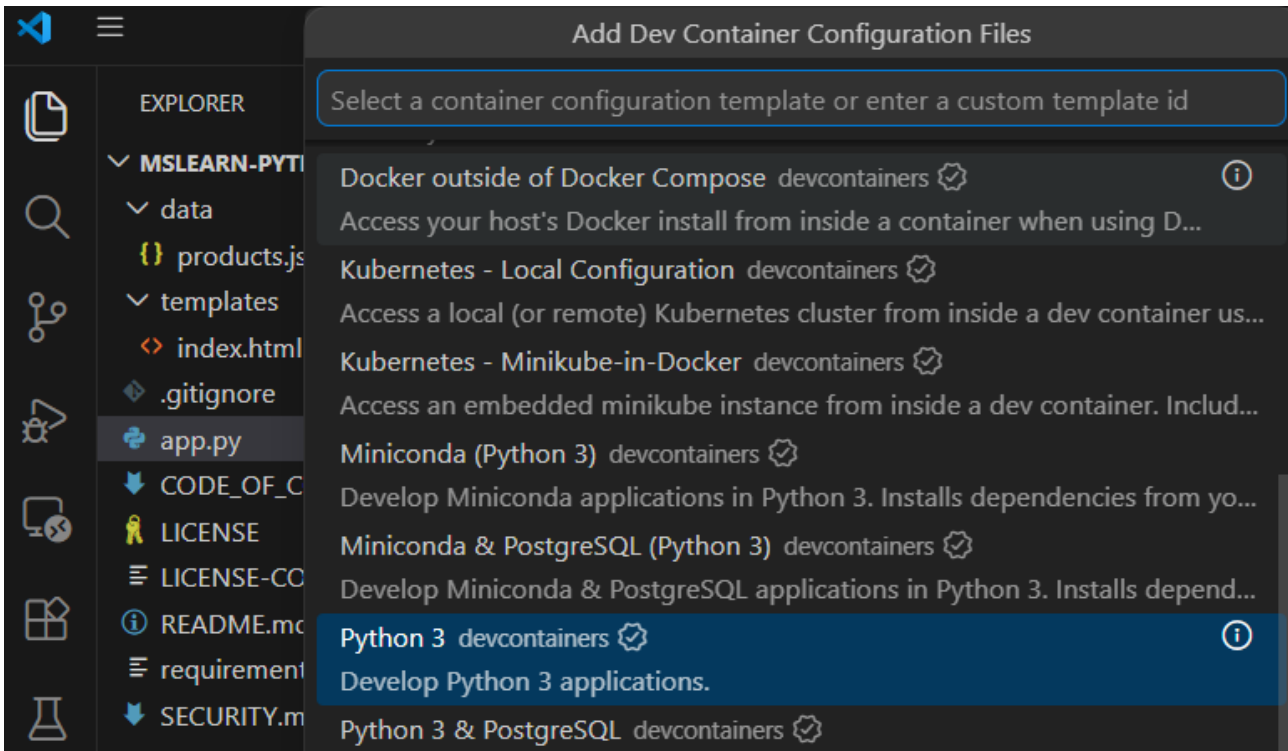
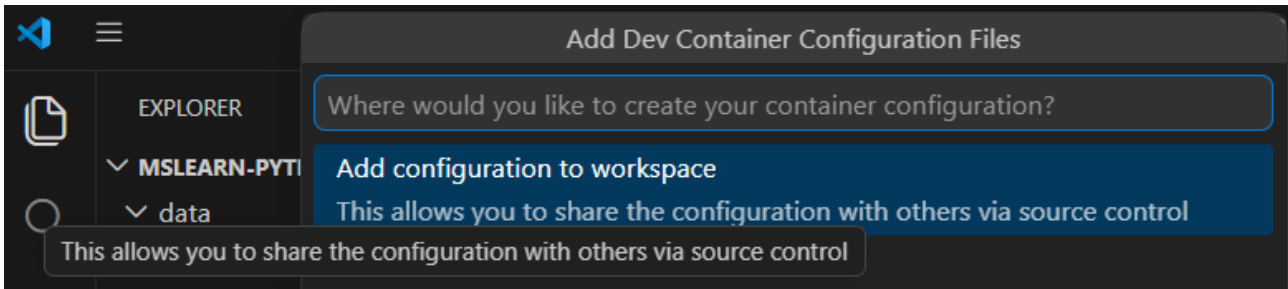
➔ Vous allez ajouter un conteneur de développement à un projet Python et l'exécuter sur votre propre ordinateur, même si Python n'est pas installé.

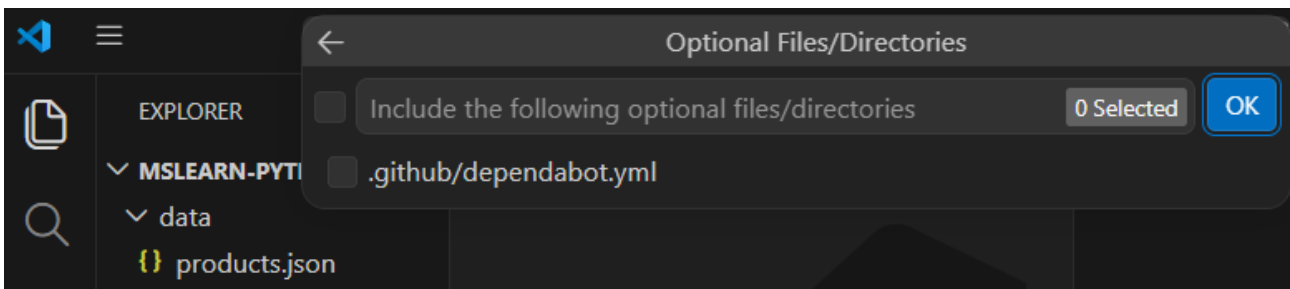
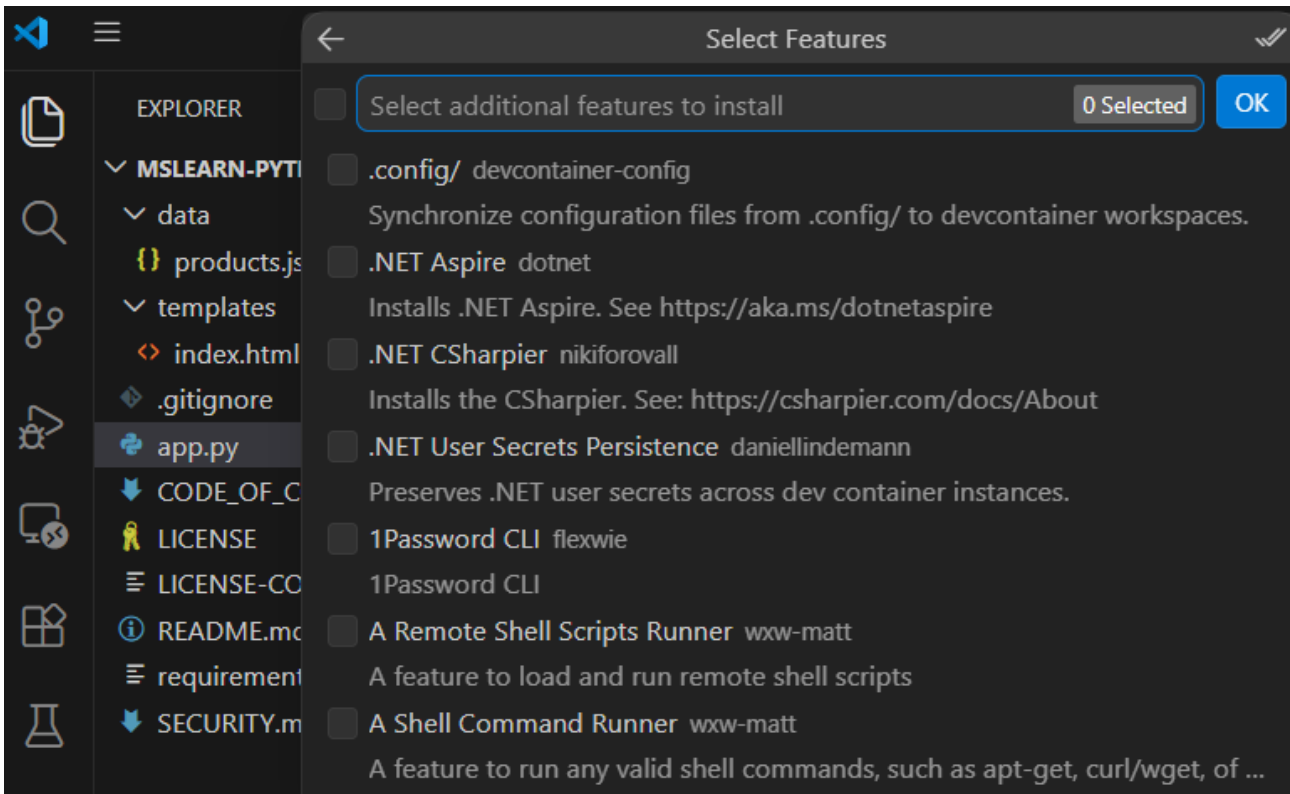
- Appuyez sur F1 pour ouvrir la palette de commandes
- Tapez **add dev container** et sélectionnez **Dev Containers: Add Dev Container Configuration Files**.



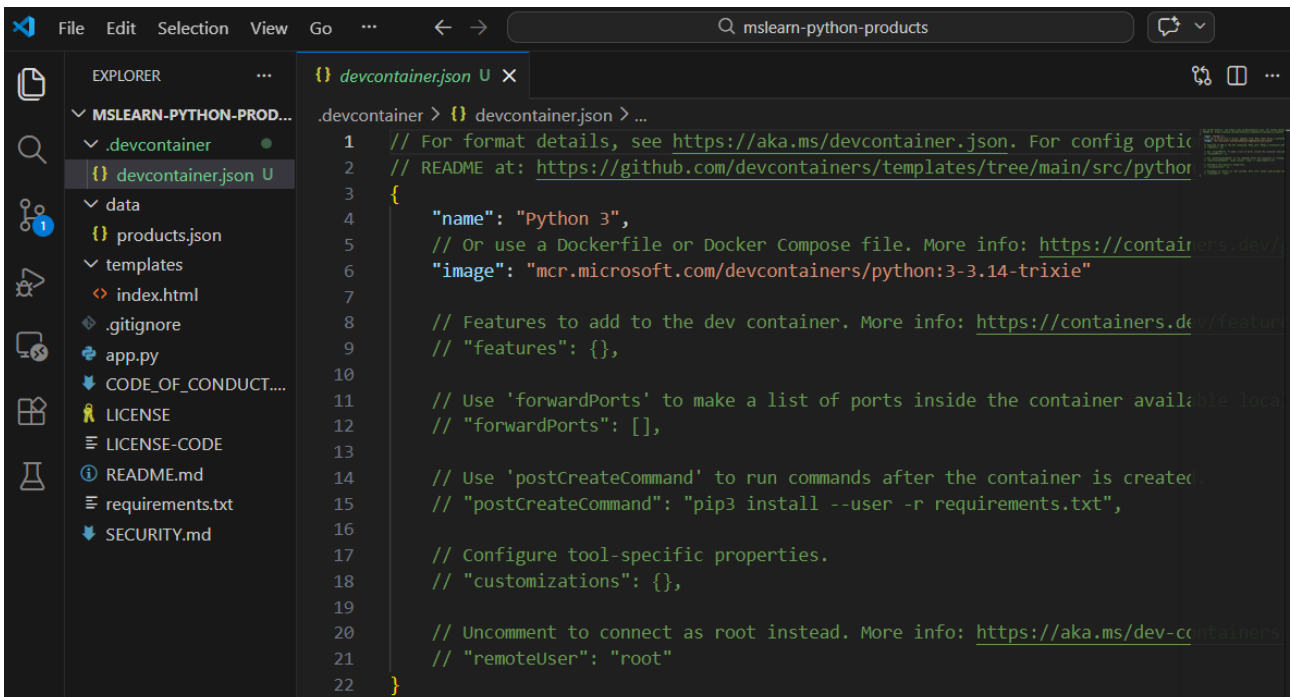
Ceci présente la liste de Modèles de conteneur de développement. Les modèles contiennent les fichiers sources nécessaires pour configurer un environnement de développement complet pour la pile technique spécifiée.

- Sélectionnez **Add configuration to workspace**.



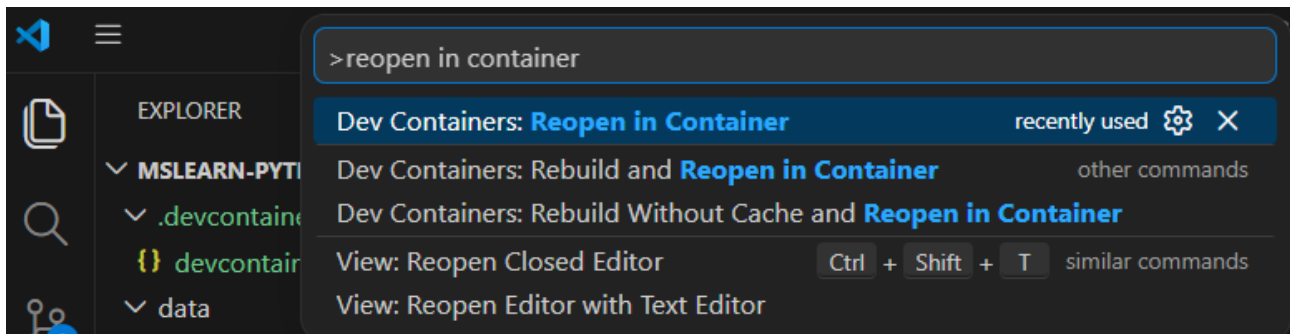


- Un nouveau dossier nommé « **.devcontainer** » a été ajouté au projet. Développez ce dossier : il contient un fichier **devcontainer.json**.



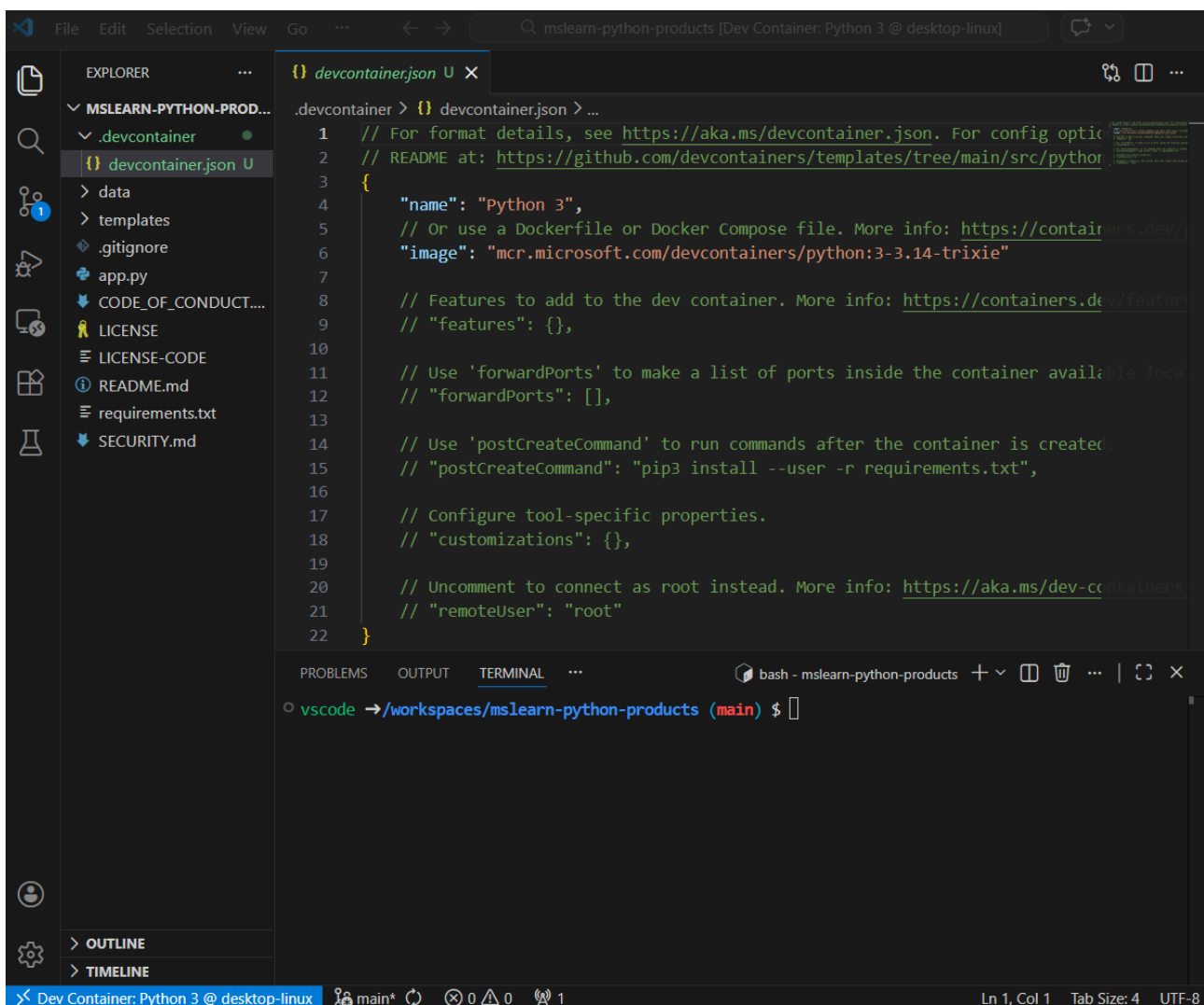
→ Ouvrir le projet dans un conteneur

- 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

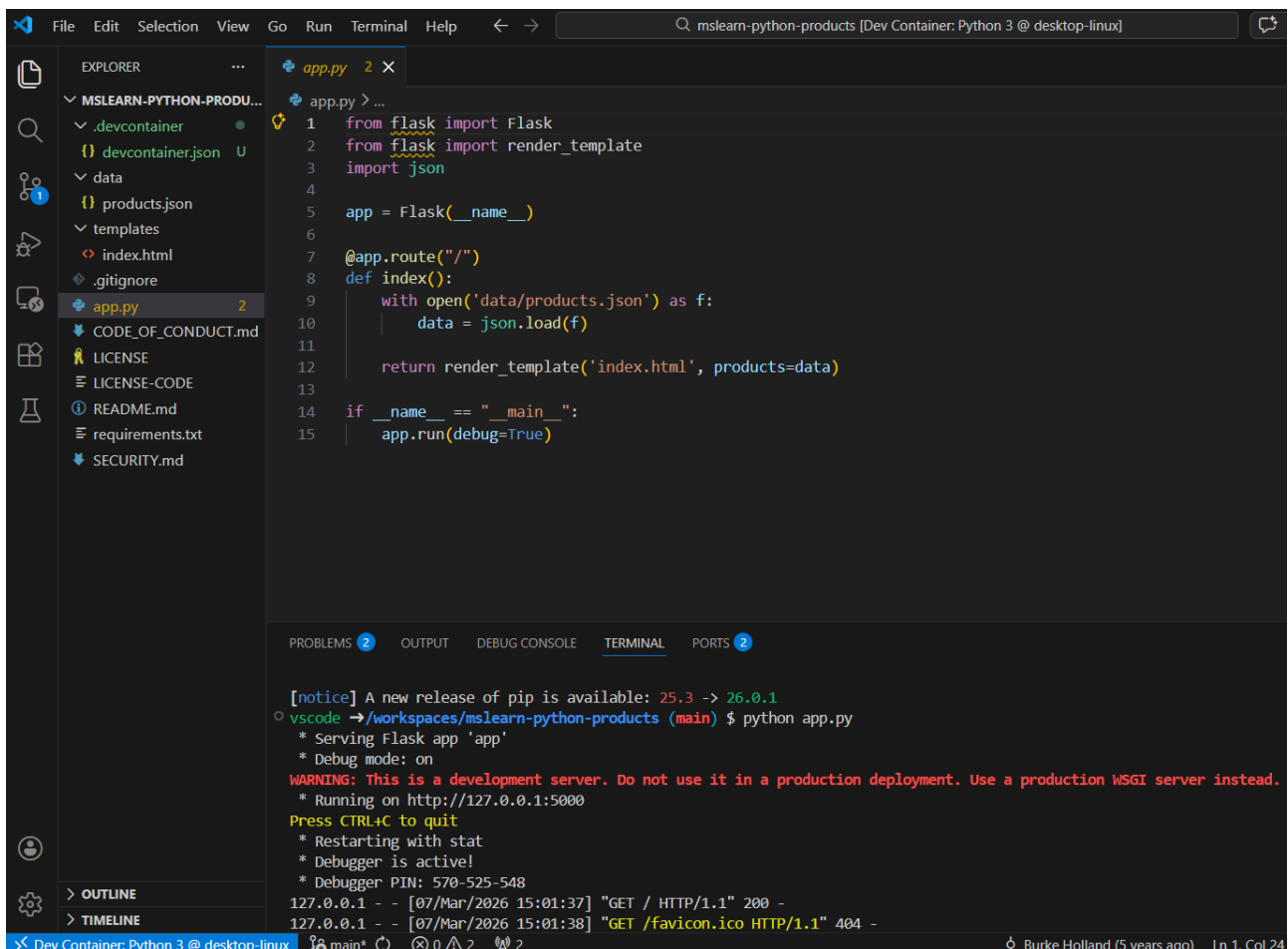
- Exécutez la commande suivante dans le terminal pour installer les **dépendances Flask** nécessaires à l'exécution du projet :

pip3 install --user -r requirements.txt

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 2
vscode →/workspaces/mslearn-python-products (main) $ python --version
Python 3.14.2
vscode →/workspaces/mslearn-python-products (main
) $ pip3 install --user -r requirements.txt pip3 install --user -r requirements.txt
Collecting Flask==2.3.3 (from -r requirements.txt (line 2))
  Downloading flask-2.3.3-py3-none-any.whl.metadata (3.6 kB)
Collecting Werkzeug>=2.3.7 (from Flask==2.3.3->-r requirements.txt (line 2))
  Downloading werkzeug-3.1.6-py3-none-any.whl.metadata (4.0 kB)
Collecting Jinja2>=3.1.2 (from Flask==2.3.3->-r requirements.txt (line 2))
  Downloading jinja2-3.1.6-py3-none-any.whl.metadata (2.9 kB)
Collecting itsdangerous>=2.1.2 (from Flask==2.3.3->-r requirements.txt (line 2))
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting click>=8.1.3 (from Flask==2.3.3->-r requirements.txt (line 2))
```

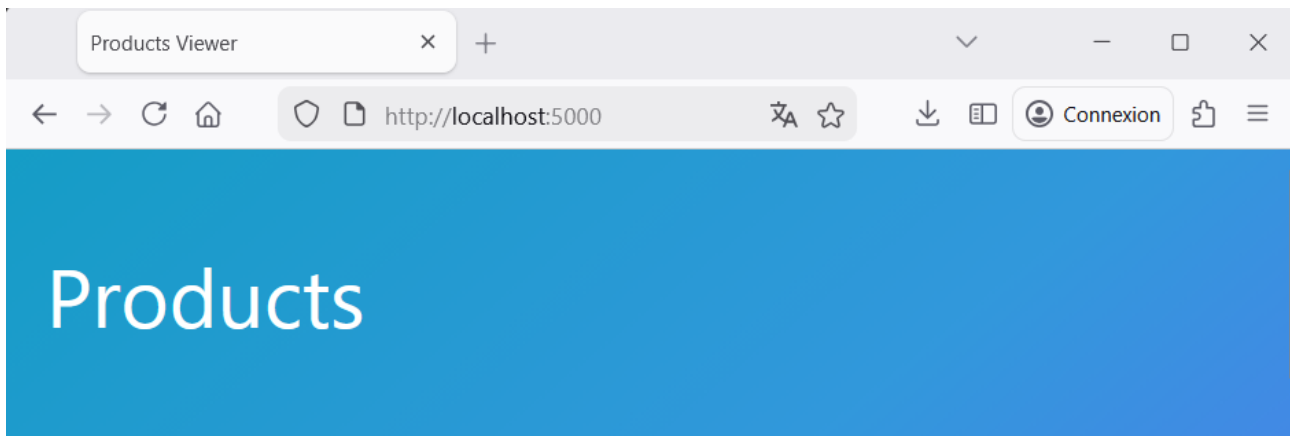
- Entrez la commande suivante dans le terminal pour exécuter le projet :

python app.py



```
File Edit Selection View Go Run Terminal Help ← → mslearn-python-products [Dev Container: Python 3 @ desktop-linux]
EXPLORER
MSLEARN-PYTHON-PRODU...
  .devcontainer
  devcontainer.json U
  data
  products.json
  templates
  index.html
  .gitignore
  app.py 2
  CODE_OF_CONDUCT.md
  LICENSE
  LICENSE-CODE
  README.md
  requirements.txt
  SECURITY.md
  app.py 2 X
1 from flask import Flask
2 from flask import render_template
3 import json
4
5 app = Flask(__name__)
6
7 @app.route("/")
8 def index():
9     with open('data/products.json') as f:
10         data = json.load(f)
11
12     return render_template('index.html', products=data)
13
14 if __name__ == "__main__":
15     app.run(debug=True)
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS 2
[notice] A new release of pip is available: 25.3 -> 26.0.1
vscode →/workspaces/mslearn-python-products (main) $ python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 570-525-548
127.0.0.1 - - [07/Mar/2026 15:01:37] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [07/Mar/2026 15:01:38] "GET /favicon.ico HTTP/1.1" 404 -
Dev Container: Python 3 @ desktop-linux main* 0 2 2 Burke Holland (5 years ago) Ln 1, Col 24
```

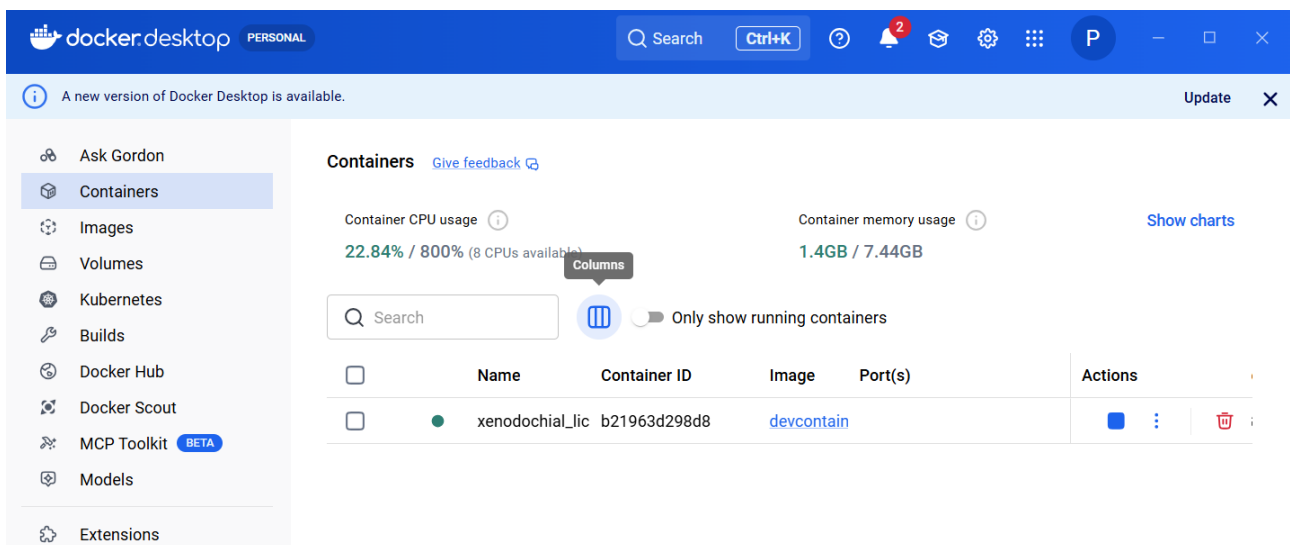
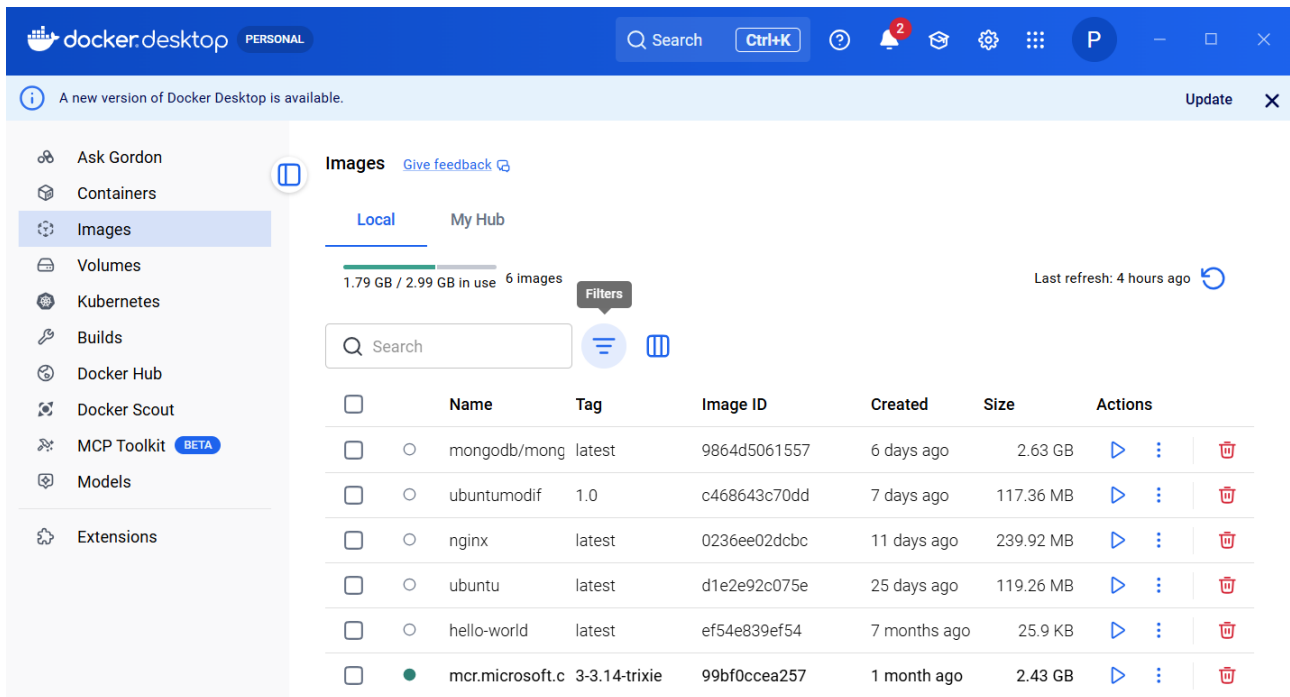
- Montrez que l'application web Python avec Flask s'est bien exécutée sur votre machine en saisissant `http://127.0.0.1:5000` depuis votre navigateur :



| 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 mslea  
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  
data > {} products.json > ...  
1  [  
2  
3  {  
4    "name": "Single red garden gnome",  
5    "price": 56,  
6    "brand": {  
7      "name": "Home & Pro tools"  
8    },  
9    "stockUnits": 98  
10  },  
11  {  
12    "name": "Two red garden gnomes",  
13    "price": 92,  
14    "brand": {  
15      "name": "Home & Pro tools"  
16    },  
17    "stockUnits": 4  
18  },  
19  {  
20    "name": "One sat gnome",  
21    "price": 34,  
22    "brand": {  
23      "name": "Home & Pro tools"  
24    },  
25  },  
26  ]  
27  ]
```

```
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  
templates > <> index.html > ...  
2 <html lang="en">  
9 <body>  
10   <div id="app">  
20     <section class="section">  
21       <div class="container">  
24         <thead>  
25           <tr>  
26             <th>Name</th>  
27             <th>Brand</th>  
28             <th>Price</th>  
29             <th class="has-text-centered">Units in Stock</th>  
30           </tr>  
31         </thead>  
32         <tbody>  
33           {% for product in products %}  
34             <tr>  
35               <td>  
36                 {{product.name}}  
37               </td>  
38               <td>  
39                 {{product.brand.name}}  
40               </td>  
41             </tr>
```



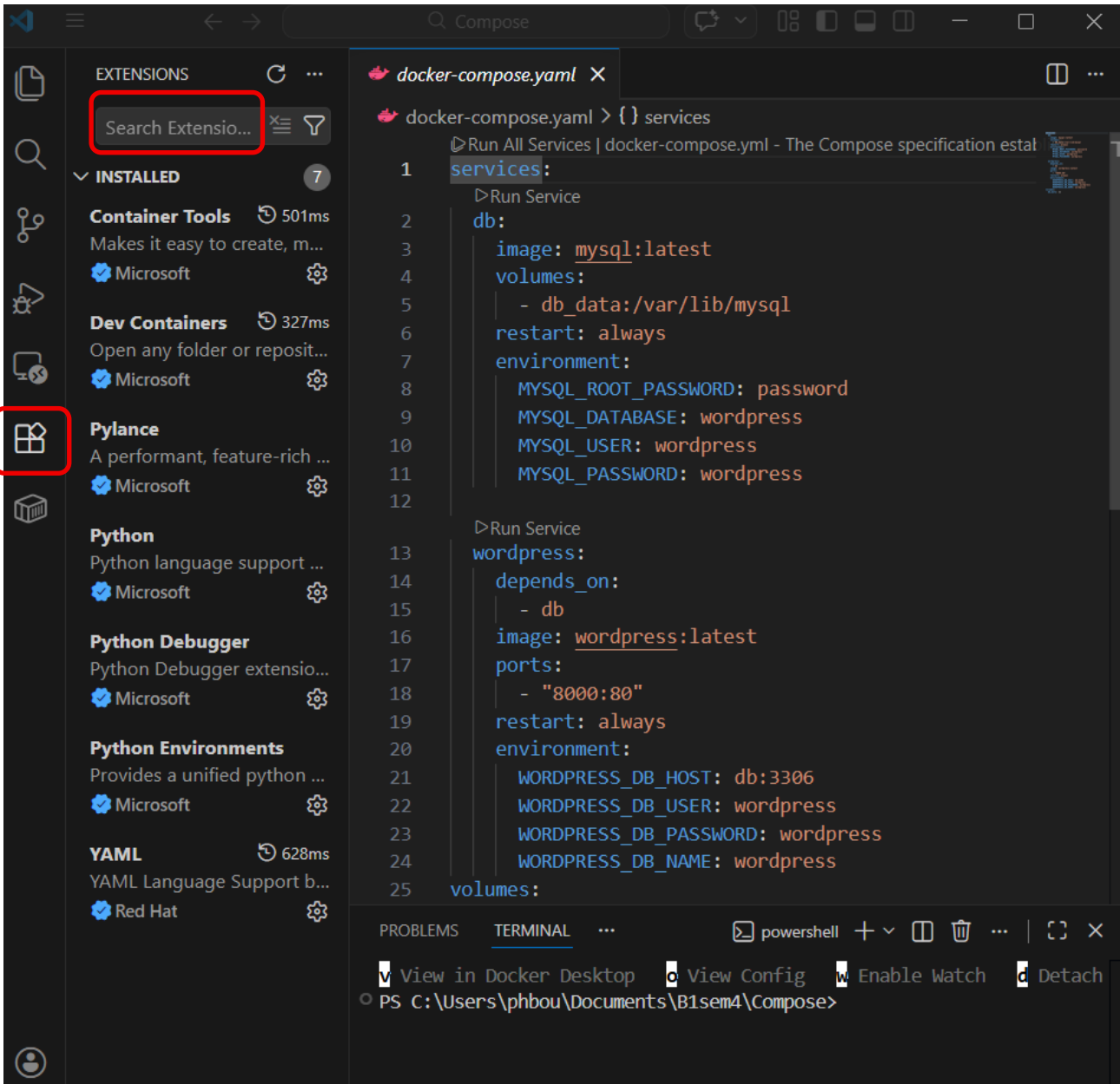
13. Monter une stack applicative avec Docker Compose.

Le lancement de plusieurs conteneurs liés peut devenir rapidement compliqué à gérer avec de nombreuses applications microservice. L'outil Docker Compose permet de définir des applications multi-conteneurs à l'aide d'un seul fichier de configuration YAML. La gestion des services, des réseaux et des volumes s'en trouve simplifiée.

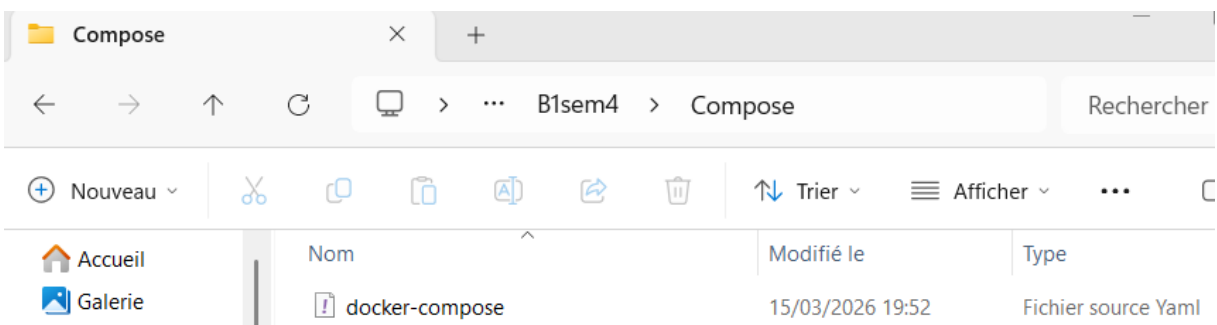
Etablir ce fichier revient à convertir en YAML les commandes docker. Par la suite, à l'aide d'une seule commande (**docker compose up**), les services sont démarrés à partir de cet unique fichier de configuration.

- ➔ Un fichier **docker-compose.yml** utilise la syntaxe YAML :
 - Indentation pour représenter les objets (2 espaces) : respecter l'alignement contrairement à JSON ;
 - Des paires **clé: valeur** (deux points pour séparer les paires clé-valeur) ;
 - Des listes (tirets suivi d'un espace) ;

- Installer l'extension YAML dans VS Code.



- Afin de monter un blog Wordpress avec sa base de données, créez dans un dossier le fichier **docker-compose.yml** suivant à l'aide de VS Code :



```
1 services:
2   db:
3     image: mysql:latest
4     volumes:
5       - db_data:/var/lib/mysql
6     restart: always
7     environment:
8       MYSQL_ROOT_PASSWORD: password
9       MYSQL_DATABASE: wordpress
10      MYSQL_USER: wordpress
11      MYSQL_PASSWORD: wordpress
12
13   wordpress:
14     depends_on:
15       - db
16     image: wordpress:latest
17     ports:
18       - "8000:80"
19     restart: always
20     environment:
21       WORDPRESS_DB_HOST: db:3306
22       WORDPRESS_DB_USER: wordpress
23       WORDPRESS_DB_PASSWORD: wordpress
24       WORDPRESS_DB_NAME: wordpress
25 volumes:
26   db_data: {}
27
```

- Depuis le terminal, lancez la stack applicative avec la commande **docker compose up** :

```
25 volumes:
26   db_data: {}
27
PROBLEMS  TERMINAL  ...
PS docker compose up
```

```
PROBLEMS  OUTPUT  TERMINAL  ...
- Image m... [████████████████████] 280.9MB / 281.9MB Pul
[+] up 37/39
- Image m... [████████████████████] 280.9MB / 281.9MB Pul
[+] up 37/39
- Image m... [████████████████████] 280.9MB / 281.9MB Pul
[+] up 37/39
- Image m... [████████████████████] 280.9MB / 281.9MB Pul
[+] up 37/39
Ln 4, Col 1  Spaces: 2  UTF-8  CRLF  {} Compose
```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- Image m... [██████████] 280.9MB / 281.9MB Pul[+] up 38/39
- Image m... [██████████] 280.9MB / 281.9MB Pul[+] up 43/43
✓ Image mysql:latest Pulled 314.3s
✓ Image wordpress:latest Pulled 291.5s
✓ Network compose_default Created 0.2s
✓ Volume compose_db_data Created 0.0s
✓ Container compose-db-1 Created 1.7s
✓ Container compose-wordpress-1 Created 0.3s
Attaching to db-1, wordpress-1
db-1 | 2026-03-15 18:35:45+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 9.6.0-1.el9 started.
wordpress-1 | WordPress not found in /var/www/html - copying now...
db-1 | 2026-03-15 18:35:46+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
db-1 | 2026-03-15 18:35:46+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 9.6.0-1.el9 started.
db-1 | 2026-03-15 18:35:46+00:00 [Note] [Entrypoint]: Initializing database files
db-1 | 2026-03-15T18:35:46.600265Z 0 [System] [MY-015017] [Server] MySQL Server Initialization - start.
db-1 | 2026-03-15T18:35:46.603865Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 9.6.0) initializing of server in progress as process 80
db-1 | 2026-03-15T18:35:46.634762Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
wordpress-1 | Complete! WordPress has been successfully copied to /var/www/html
wordpress-1 | No 'wp-config.php' found in /var/www/html, but 'WORDPRESS_...' variables supplied; copying 'wp-config-docker.php' (WORDPRESS_DB_HOST WORDPRESS_DB_NAME WORDPRESS_DB_PASSWORD WORDPRESS_DB_USER)
wordpress-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.19.0.3. Set the 'ServerName' directive globally to suppress this message
wordpress-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.19.0.3. Set the 'ServerName' directive glob

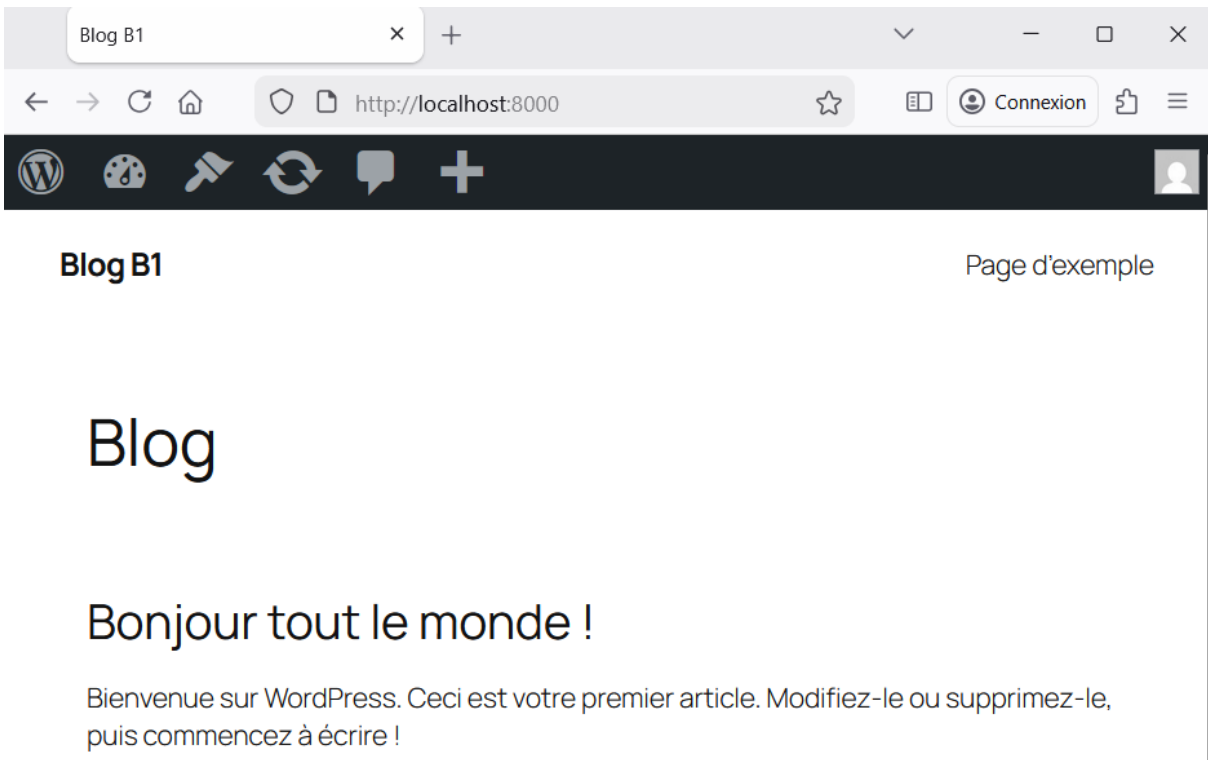
```

```

wordpress-1 | 172.19.0.1 - - [15/Mar/2026:18:51:39 +0000] "GET /wp-includes/js/wplink.min.js?ver=6.9.4 HTTP/1.1" 200 4241 "http://localhost:8000/wp-admin/"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:148.0) Gecko/20100101 Firefox/148.0"
wordpress-1 | 172.19.0.1 - - [15/Mar/2026:18:51:39 +0000] "GET /wp-includes/js/jquery/ui/menu.min.js?ver=1.13.3 HTTP/1.1" 200 3402 "http://localhost:8000/wp-admin/"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:148.0) Gecko/20100101 Firefox/148.0"
wordpress-1 | 172.19.0.1 - - [15/Mar/2026:18:51:39 +0000] "GET /wp-includes/js/jquery/ui/autocomplete.min.js?ver=1.13.3 HTTP/1.1" 200 3249 "http://localhost:8000/wp-admin/"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:148.0) Gecko/20100101 Firefox/148.0"
wordpress-1 | 172.19.0.1 - - [15/Mar/2026:18:51:39 +0000] "GET /wp-includes/js/dist/components.min.js?ver=0d72a218dfea8f6f442f HTTP/1.1" 200 253569 "http://localhost:8000/wp-admin/"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:148.0) Gecko/20100101 Firefox/148.0"
wordpress-1 | 172.19.0.1 - - [15/Mar/2026:18:51:39 +0000] "GET /wp-includes/js/dist/block-editor.min.js?ver=adf558a422e2ee57920 HTTP/1.1" 200 270522 "http://localhost:8000/wp-admin/"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:148.0) Gecko/20100101 Firefox/148.0"
wordpress-1 | 172.19.0.1 - - [15/Mar/2026:18:51:39 +0000] "GET /index.php?rest_route=%2Fwp%2Fv2%2Fusers%2Fme&context=edit&locale=user HTTP/1.1" 200 3070 "http://localhost:8000/wp-admin/"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:148.0) Gecko/20100101 Firefox/148.0"
wordpress-1 | 172.19.0.1 - - [15/Mar/2026:18:51:39 +0000] "GET /wp-admin/admin-ajax.php?action=wp-compression-test&test=1&ajax_nonce=738e69fc52&1773600699e40 HTTP/1.1" 200 1188 "http://localhost:8000/wp-admin/"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:148.0) Gecko/20100101 Firefox/148.0"
wordpress-1 | 172.19.0.1 - - [15/Mar/2026:18:51:39 +0000] "GET /favicon.ico HTTP/1.1" 404 529 "http://localhost:8000/wp-admin/"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:148.0) Gecko/20100101 Firefox/148.0"
wordpress-1 | 172.19.0.1 - - [15/Mar/2026:18:51:39 +0000] "GET /wp-includes/js/thickbox/loadingAnimation.gif HTTP/1.1" 200 15525 "http://localhost:8000/wp-admin/"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:148.0) Gecko/20100101 Firefox/148.0"
wordpress-1 | 172.19.0.1 - - [15/Mar/2026:18:51:39 +0000] "GET /wp-admin/admin-ajax.php?action=wp-compression-test&test=no&ajax_nonce=738e69fc52&1773600699e40 HTTP/1.1" 200 1188 "http://localhost:8000/wp-admin/"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:148.0) Gecko/20100101 Firefox/148.0"
wordpress-1 | 172.19.0.1 - - [15/Mar/2026:18:51:39 +0000] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 581 "http://localhost:8000/wp-admin/"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:148.0) Gecko/20100101 Firefox/148.0"
wordpress-1 | 172.19.0.1 - - [15/Mar/2026:18:51:39 +0000] "POST /wp-admin/admin-ajax.php HTTP/1.1" 200 581 "http://localhost:8000/wp-admin/"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:148.0) Gecko/20100101 Firefox/148.0"

```

- Accédez au blog en saisissant l'URL <http://localhost:8000> :



docker.desktop PERSONAL

Q Search **Ctrl+K**

Ask Gordon

Containers

Images

Volumes

Kubernetes

Builds

Docker Hub

Docker Scout

MCP Toolkit **BETA**

Models

Extensions

Manage

VS Code Web

Images Give feedback

Local My Hub

3.58 GB / 5.67 GB in use 10 images Last refresh: 2 hours ago

Q Search

| <input type="checkbox"/> | Name | Tag | Image ID | Created | Size | Actions |
|--------------------------|-----------------|---------------|--------------|--------------|-----------|--|
| <input type="checkbox"/> | mongodb/mong | latest | 9864d5061557 | 15 days ago | 2.63 GB | ▶ ⋮ 🗑️ |
| <input type="checkbox"/> | ubuntu/modif | 1.0 | c468643c70dd | 15 days ago | 117.36 MB | ▶ ⋮ 🗑️ |
| <input type="checkbox"/> | nginx | latest | 0236ee02dcbc | 19 days ago | 239.92 MB | ▶ ⋮ 🗑️ |
| <input type="checkbox"/> | ubuntu | latest | d1e2e92c075e | 1 month ago | 119.26 MB | ▶ ⋮ 🗑️ |
| <input type="checkbox"/> | mcr.microsoft.c | 3-3.14-trixie | 99bf0ccea257 | 1 month ago | 2.43 GB | ▶ ⋮ 🗑️ |
| <input type="checkbox"/> | hello-world | latest | ef54e839ef54 | 7 months ago | 25.9 KB | ▶ ⋮ 🗑️ |
| <input type="checkbox"/> | mochoa/coder-c | 4.106.2-39 | f5f3d7423df8 | 4 months ago | 20.82 MB | ▶ ⋮ 🗑️ |
| <input type="checkbox"/> | codercom/code | 4.106.2-39 | b1378d8d7cf5 | 4 months ago | 1.04 GB | ▶ ⋮ 🗑️ |
| <input type="checkbox"/> | wordpress | latest | 2a05ac562b5c | 4 days ago | 1.07 GB | ▶ ⋮ 🗑️ |
| <input type="checkbox"/> | mysql | latest | c1667ed43667 | 2 days ago | 1.26 GB | ▶ ⋮ 🗑️ |

docker.desktop PERSONAL

Q Search **Ctrl+K**

Ask Gordon

Containers

Images

Volumes

Kubernetes

Builds

Docker Hub

Docker Scout

MCP Toolkit **BETA**

Models

Extensions

Manage

VS Code Web

Containers Give feedback

Container CPU usage 0.60% / 800% (8 CPUs available)

Container memory usage 585MB / 7.44GB [Show charts](#)

Q Search Only show running containers

| <input type="checkbox"/> | Name | Container ID | Image | Port(s) | Actions |
|-------------------------------------|---------------|--------------|------------|---------|--|
| <input type="checkbox"/> | compose stack | f2d90b7fff94 | devcontain | | ▶ ⋮ 🗑️ |
| <input checked="" type="checkbox"/> | compose | | | | ▶ ⋮ 🗑️ |

Ask Gordon

Containers

Images

Volumes

Kubernetes

Builds

Docker Hub

Docker Scout

MCP Toolkit BETA

Models










Extensions ^

Manage

VS Code Web

 **compose**

C:\Users\phbou\Documents\B1sem4\Compose

| | | | | |
|---|---|--|---|---|
|  | db ● mysql:latest |  |  |  |
|  | wordpress ● wordpress:latest 8000:80  |  |  |  |