

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

Table des matières

Chapitre 4 – Docker sous Windows (Docker Desktop et WSL 2).....	1
1. Installation / Activation de WSL.....	2
2. Installation Ubuntu.....	4
3. Installation Docker.....	6
4. Premier test.....	11
5. Deuxième test.....	13
6. Troisième test.....	15
7. Quatrième test.....	19
8. Cinquième test : mode interactif d'utilisation d'un conteneur.....	22
9. Création d'une image à la main avec docker commit.....	25
10. Première approche des volumes.....	27
11. Gestion des volumes en écriture.....	28
11.1. Deuxième type de volume : montage lié.....	37
12. Utiliser un conteneur Docker et Visual Studio Code.....	40
13. Monter une stack applicative avec Docker Compose.....	48

1. Installation / Activation de WSL.

Nous installons WSL depuis PowerShell et affichons les OS disponibles.

```
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

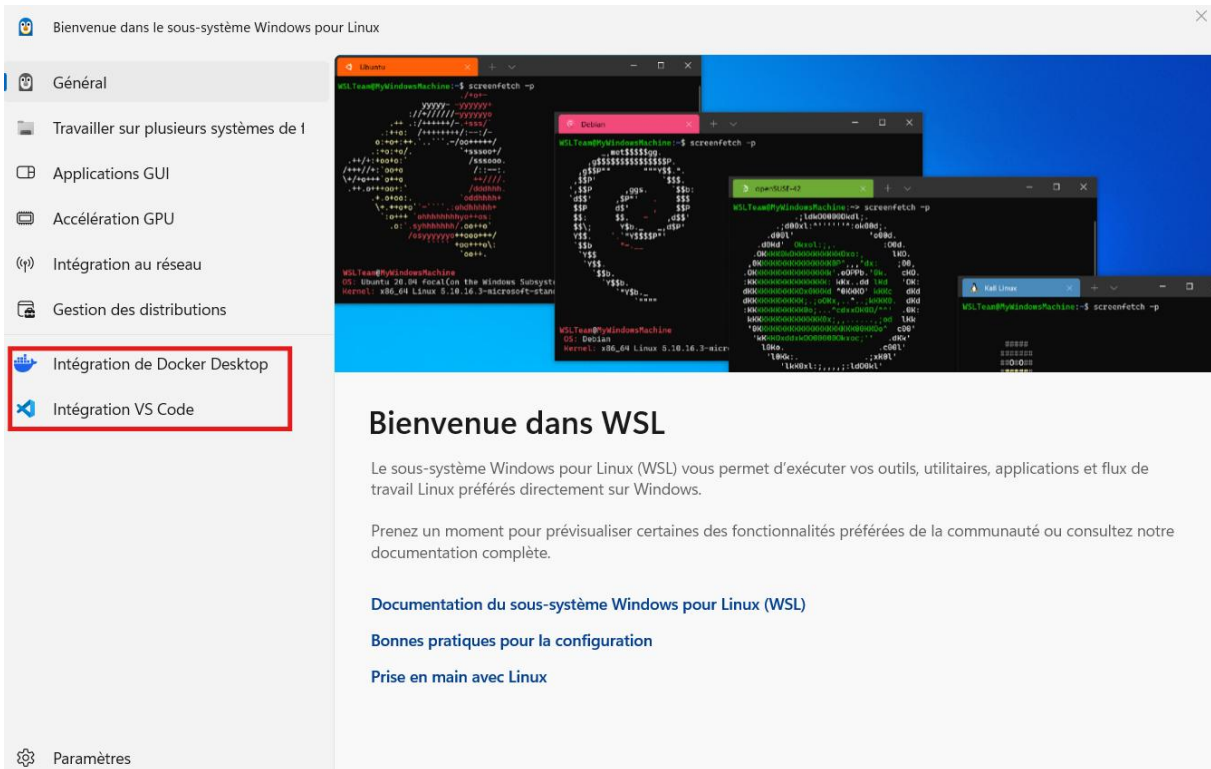
Version de l'image : 10.0.26200.7462

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>
```

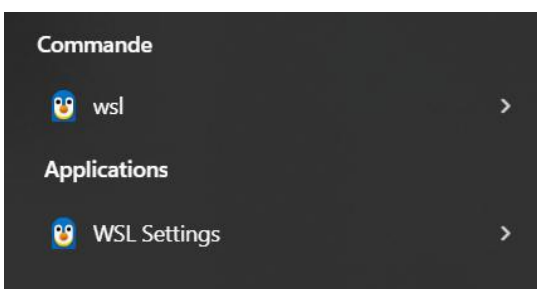
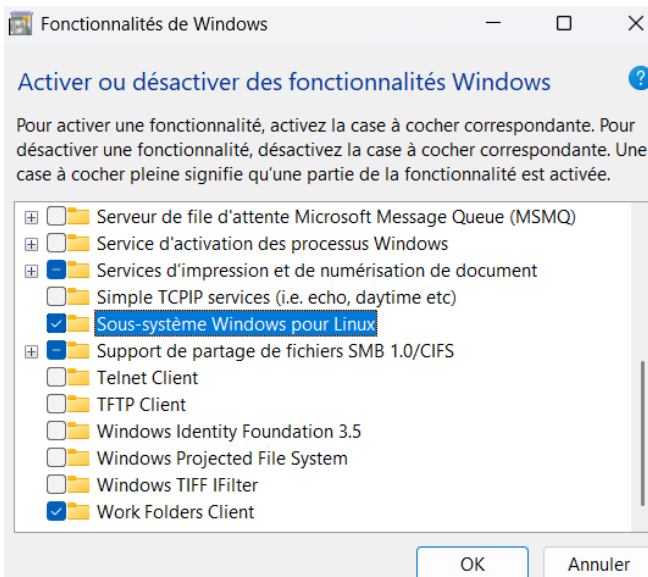
```
PS C:\WINDOWS\system32> wsl.exe --list --online
Voici la liste des distributions valides qui peuvent être installées.
Installez à l'aide de 'wsl.exe --install <Distro>'.

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
PS C:\WINDOWS\system32>
```

Nous lançons WSL et constatons que Docker ainsi que VS Code sont présentés comme pouvant être utilisés.



Nous activons le sous-système Windows pour Linux puis nous exécutons le WSL.

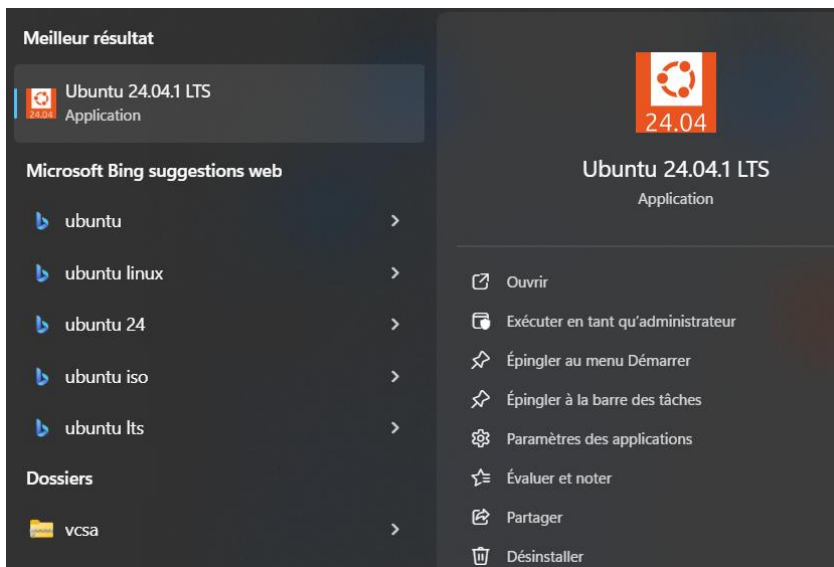
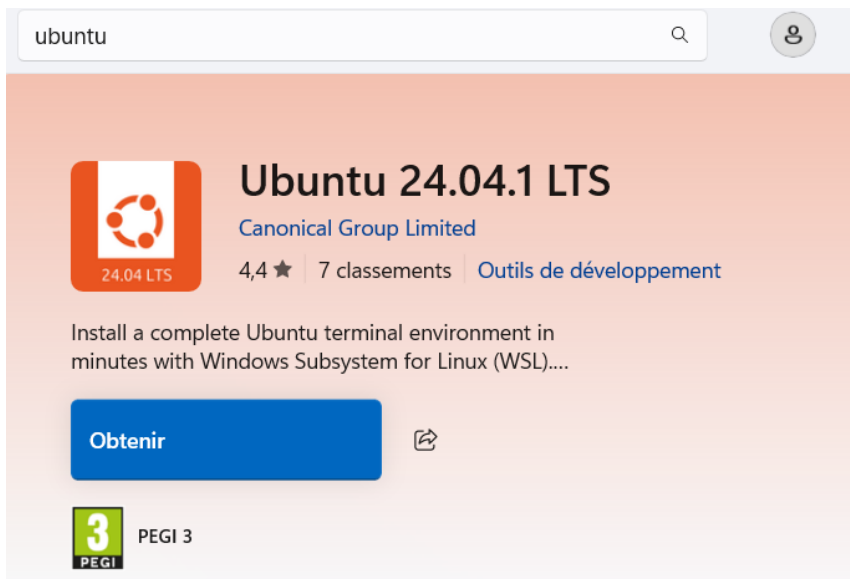


```
sio@Nicolau: /mnt/c/WINDO1 x + v
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

sio@Nicolau:~$ cd /mnt/c/WINDOWS/system32
sio@Nicolau:/mnt/c/WINDOWS/system32$
```

2. Installation Ubuntu.

Depuis Microsoft Store, nous téléchargeons la dernière version de Ubuntu TLS.



Nous créons le compte SIO et exécutons quelques commandes pour vérifier que nous sommes sur le Linux.

```
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to mat
ch 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 x8
6_64)

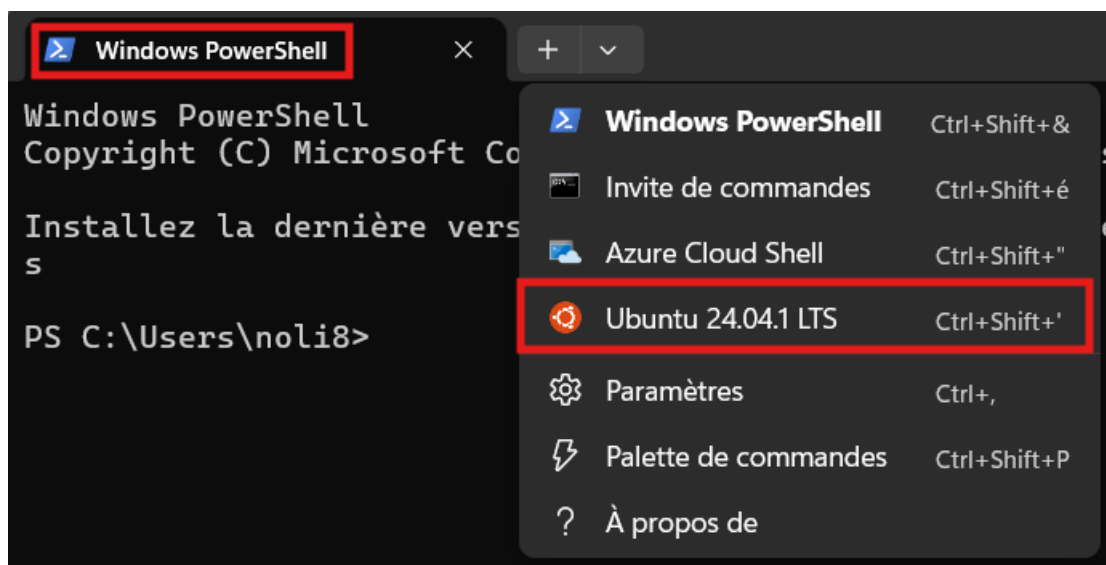
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Thu Mar  5 16:46:07 CET 2026

System load:  0.46          Processes:            58
Usage of /:   0.1% of 1006.85GB Users logged in:     0
Memory usage: 4%          IPv4 address for eth0: 172.22.149.8
Swap usage:   0%

This message is shown once a day. To disable it please create the
/home/sio/.hushlogin file.
sio@Nicolau:~$ pwd
/home/sio
sio@Nicolau:~$ cd /
sio@Nicolau:/$ ls
bin                home               lost+found         root               srv
bin.usr-is-merged  init              media              run                sys
boot              lib                mnt                sbin               tmp
dev               lib.usr-is-merged opt                 sbin.usr-is-merged usr
etc               lib64              proc                snap                var
sio@Nicolau:/$ |
```

Nous constatons que l'on peut utiliser plusieurs façons pour accéder au terminal d'Ubuntu.



```
Windows PowerShell
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

sio@Nicolau:~$
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

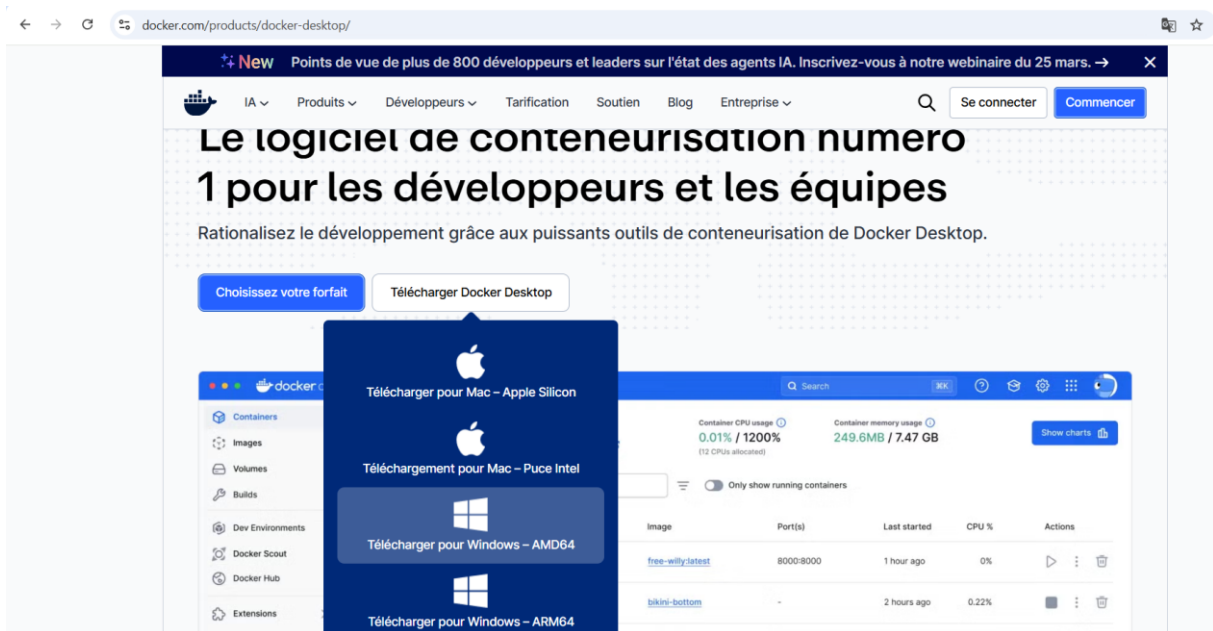
Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

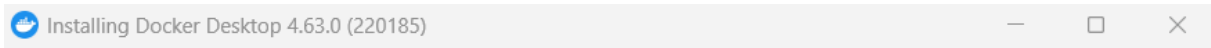
PS C:\Users\noli8> wsl -d Ubuntu-24.04
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

sio@Nicolau:/mnt/c/Users/noli8$ exit
logout
PS C:\Users\noli8>
```

3. Installation Docker.

Nous allons sur le site de Docker et installons Docker Desktop pour Windows.

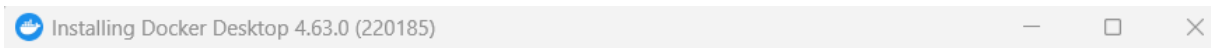




Configuration

Add shortcut to desktop

OK



Docker Desktop 4.63.0

Unpacking files...

```
Unpacking file: frontend/Docker Desktop.exe
Unpacking file: frontend/d3dcompiler_47.dll
Unpacking file: frontend/chrome_200_percent.pak
Unpacking file: frontend/chrome_100_percent.pak
Unpacking file: DockerCli.exe
Unpacking file: Docker.Core.pdb
Unpacking file: Docker.Core.dll
Unpacking file: Docker Desktop.exe
Unpacking file: Docker Desktop Installer.pdb
Unpacking file: Docker Desktop Installer.exe.config
Unpacking file: Docker Desktop Installer.exe
Unpacking file: courgette64.exe
Unpacking file: com.docker.service.pdb
Unpacking file: com.docker.service.config
```

Installing Docker Desktop 4.63.0 (220185)

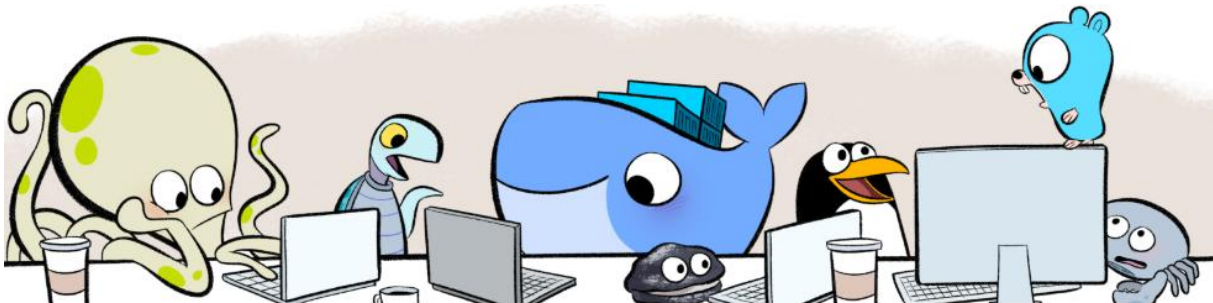
Docker Desktop 4.63.0

Installation succeeded

You must log out of Windows to complete installation.

Close and log out

Une fois l'installation finie, nous nous retrouvons sur Docker, celui-ci nous demande de nous connecter.



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



Welcome to Docker Skip

Work Personal

Using Docker for work?
We recommend signing in with your work email address.

Work email address

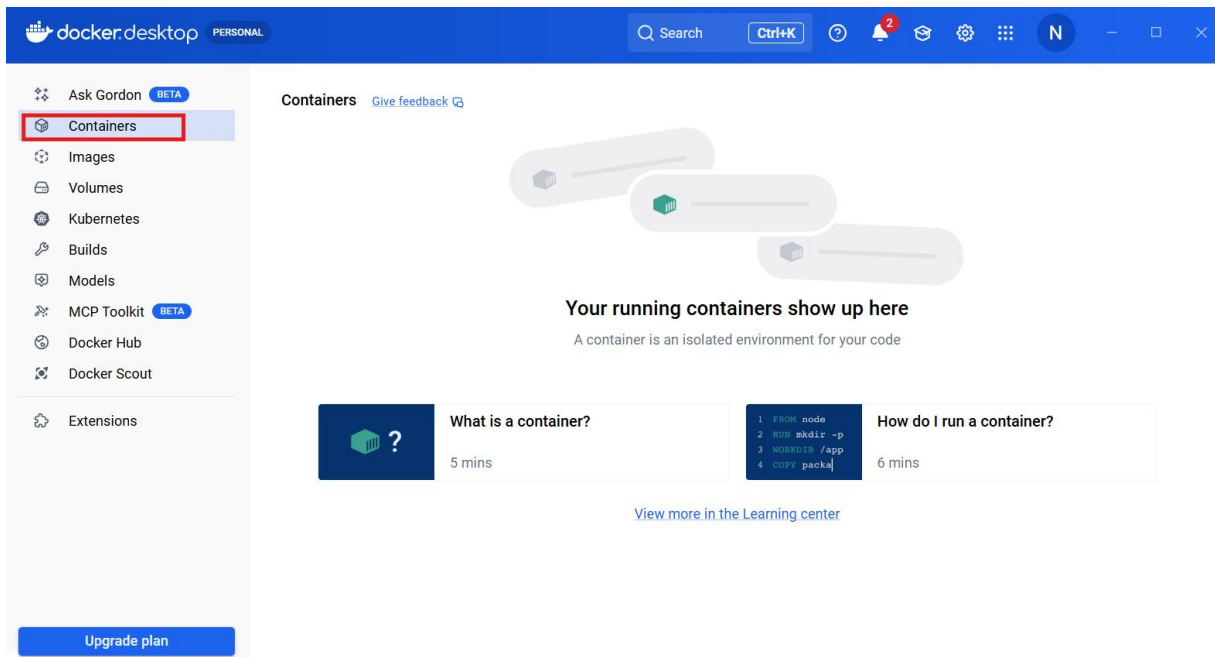
Continue

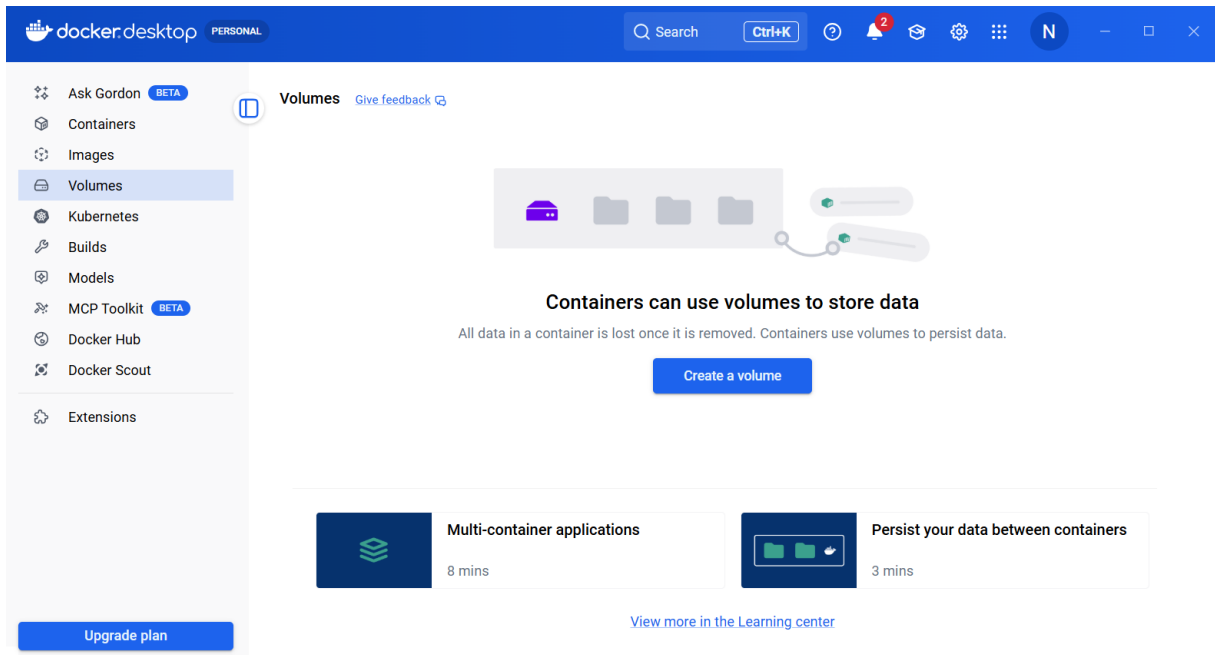
Or

[Create an account](#)

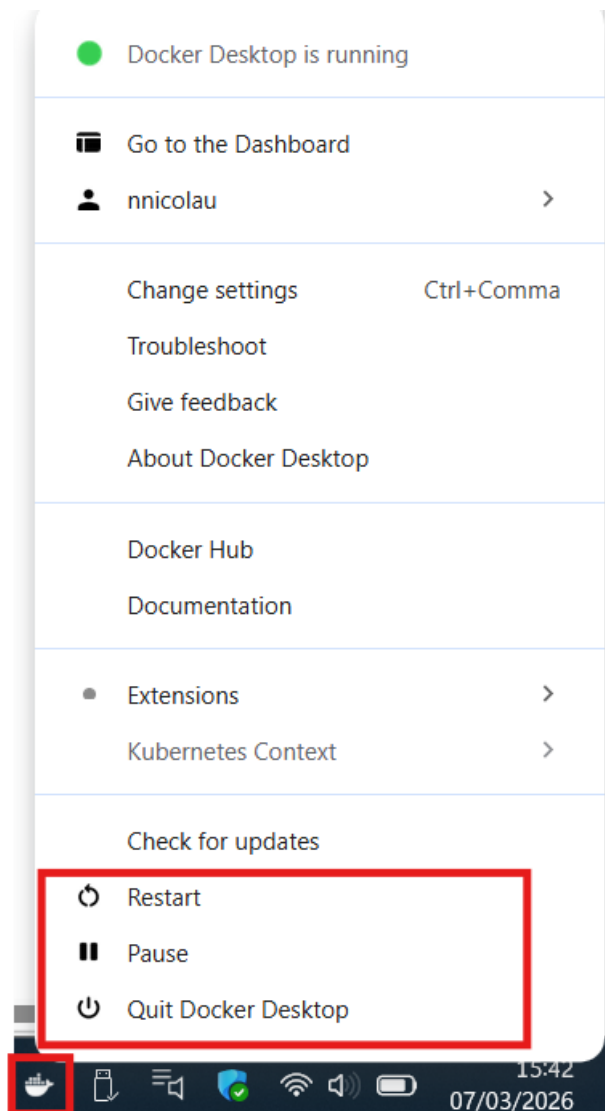
The form is a white card with rounded corners. It has a title 'Welcome to Docker' and a 'Skip' link. Below the title are two tabs: 'Work' (which is underlined) and 'Personal'. The main text asks if the user is using Docker for work and recommends signing in with a work email address. There is a text input field for the work email address, followed by a 'Continue' button. Below this is a horizontal line with the word 'Or' centered. At the bottom, there are three buttons: a Google logo, a GitHub logo, and a 'Create an account' button with a GitHub logo.

Une fois dans Docker, nous constatons qu'il y a différents onglets que nous utiliserons plus tard.





Nous pouvons agir sur Docker depuis la barre des tâches.



4. Premier test.

Nous testons si docker est bien reconnue sur les différents os.

```
sio@Nicolau: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
sio@Nicolau:~$ docker --version  
Docker version 29.2.1, build a5c7197  
sio@Nicolau:~$
```

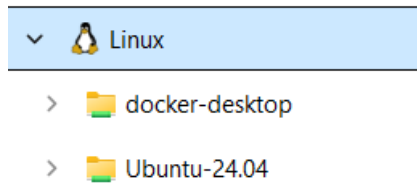
```
Windows PowerShell  
Copyright (C) Microsoft Corporation. Tous droits réservés.  
  
Installez la dernière version de PowerShell pour de nouvelles fonctionna-  
lités et améliorations ! https://aka.ms/PSWindows  
  
PS C:\Users\noli8> docker --version  
Docker version 29.2.1, build a5c7197  
PS C:\Users\noli8>
```

```
Windows PowerShell Invite de commandes  
Microsoft Windows [version 10.0.26200.7922]  
(c) Microsoft Corporation. Tous droits réservés.  
  
C:\Users\noli8>docker version  
Client:  
Version: 29.2.1  
API version: 1.53  
Go version: go1.25.6  
Git commit: a5c7197  
Built: Mon Feb 2 17:20:16 2026  
OS/Arch: windows/amd64  
Context: desktop-linux  
  
Server: Docker Desktop 4.63.0 (220185)  
Engine:  
Version: 29.2.1  
API version: 1.53 (minimum version 1.44)  
Go version: go1.25.6  
Git commit: 6bc6209  
Built: Mon Feb 2 17:17:24 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  
  
C:\Users\noli8>
```

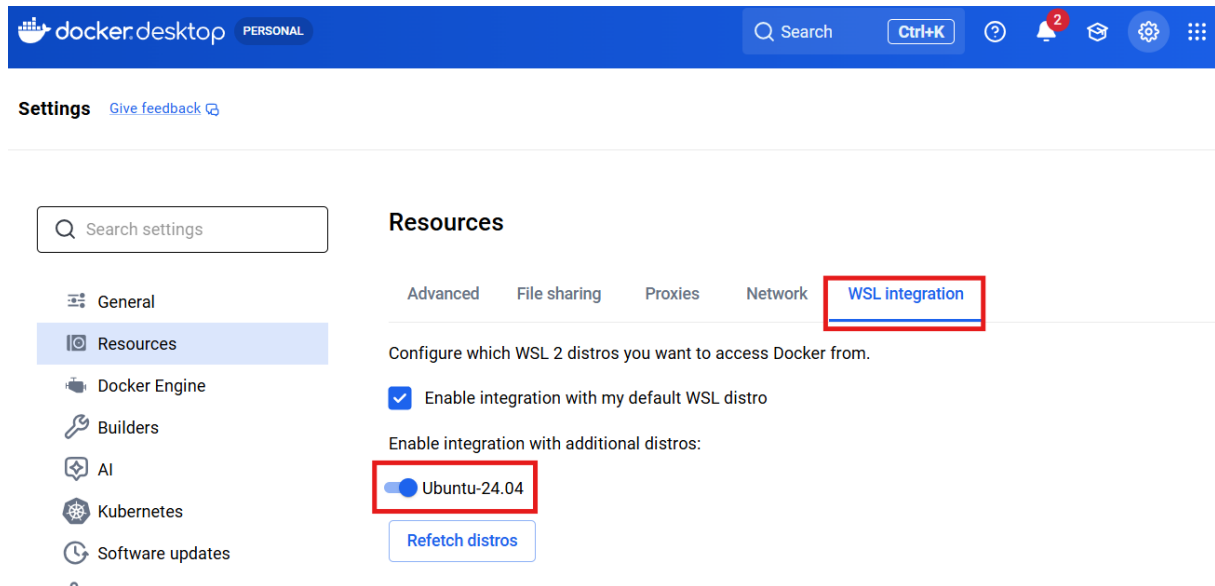
Nous vérifions que WSL tourne en arrière-plan et affichons sa fonctionnalité.

```
PS C:\Users\noli8> wsl -l -v
NAME                STATE              VERSION
* Ubuntu-24.04      Running            2
  docker-desktop    Running            2
PS C:\Users\noli8> wsl -l
Distributions du Sous-système Windows pour Linux :
Ubuntu-24.04 (par défaut)
docker-desktop
PS C:\Users\noli8> |
```

Nous constatons qu'un disque Linux est apparu sur notre machine avec à l'intérieur le docker ainsi que le Ubuntu téléchargé.



Dans les paramètres de Docker, nous activons la compatibilité WSL vers Ubuntu.

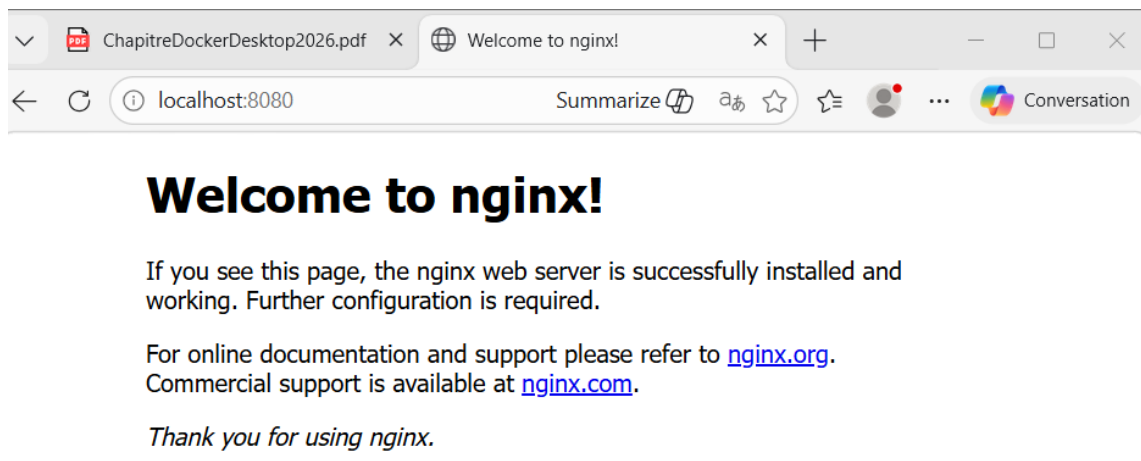


5. Deuxième test.

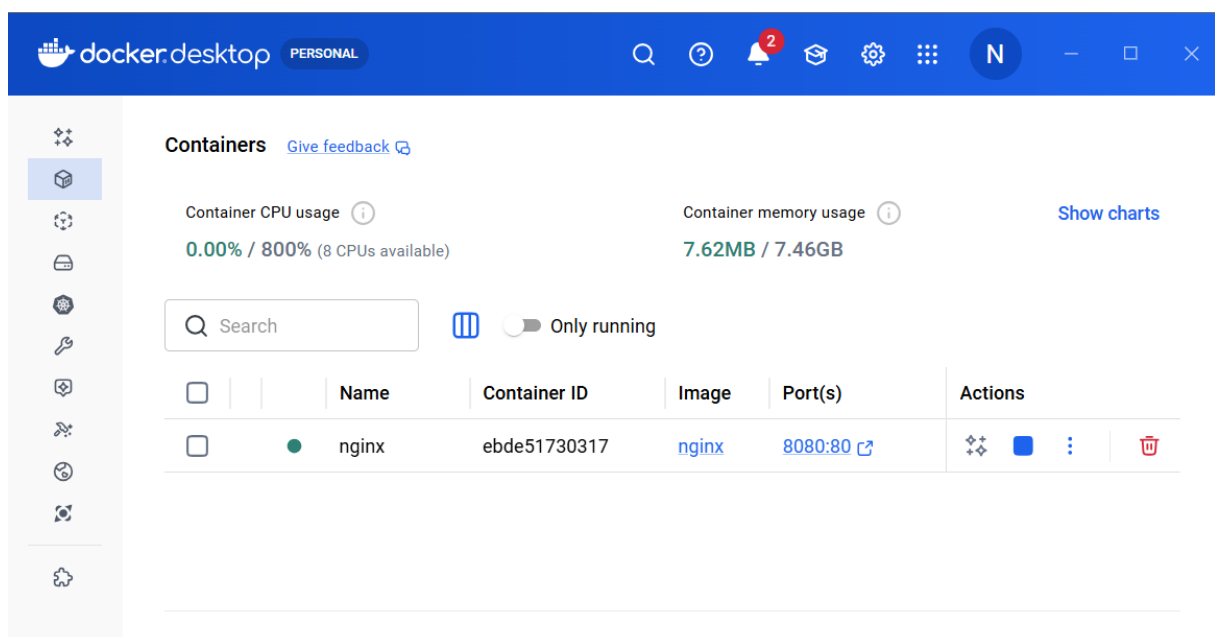
Depuis L'invite de commande Ubuntu, nous créons un conteneur et le nommons nginx en référence à l'image que nous utiliserons.

```
sio@Nicolau:~$ docker run --name nginx -p 8080:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
df0b66c867e4: Download complete
eedda9fd8786: Download complete
35ff83c394d6: Download complete
17d0911eaf62: Download complete
b47f187216b6: Downloading 10.49MB/33.14MB
1ad233904a11: Download complete
206356c42440: Downloading 7.34MB/29.78MB
0019bbecc572: Download complete
c13150b54a53: Download complete
```

Nous voici sur la page créée par Docker.

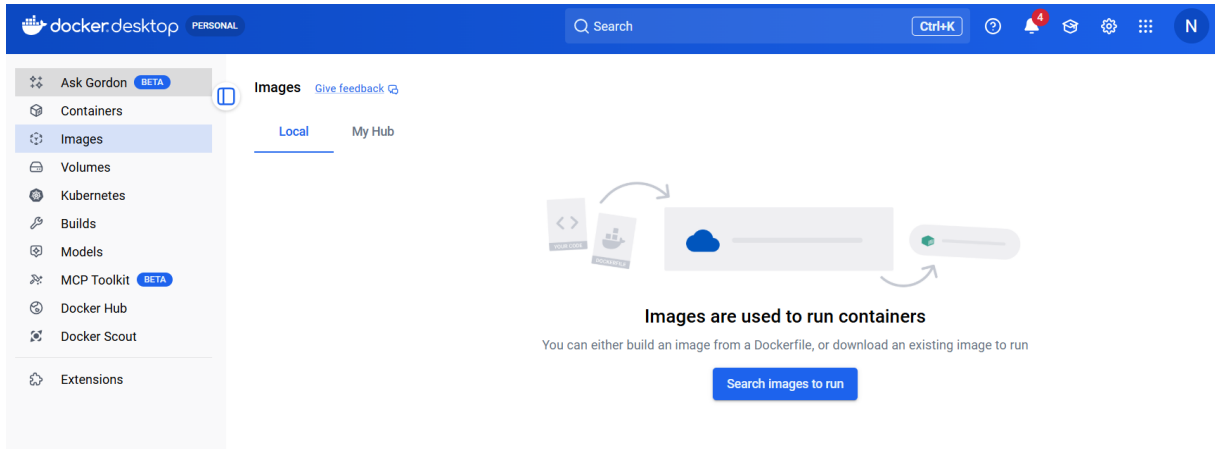


Nous allons docker pour constater que le centenaire s'est créé et qu'il tourne.

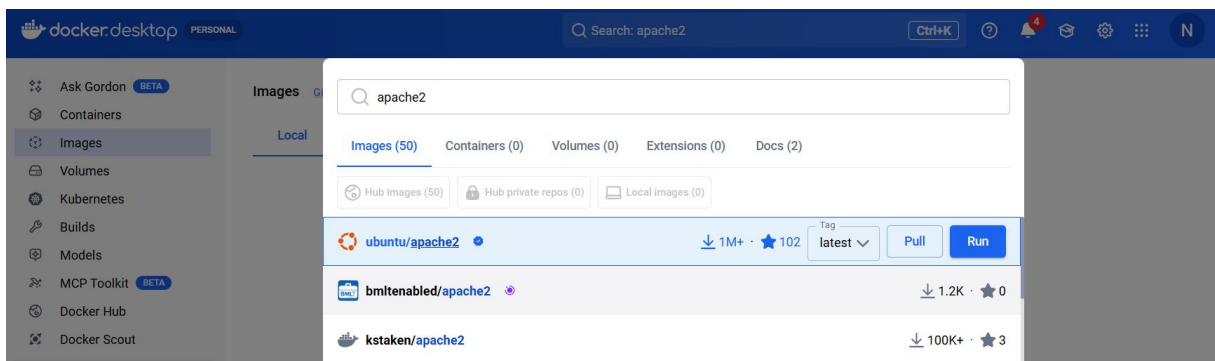


6. Troisième test.

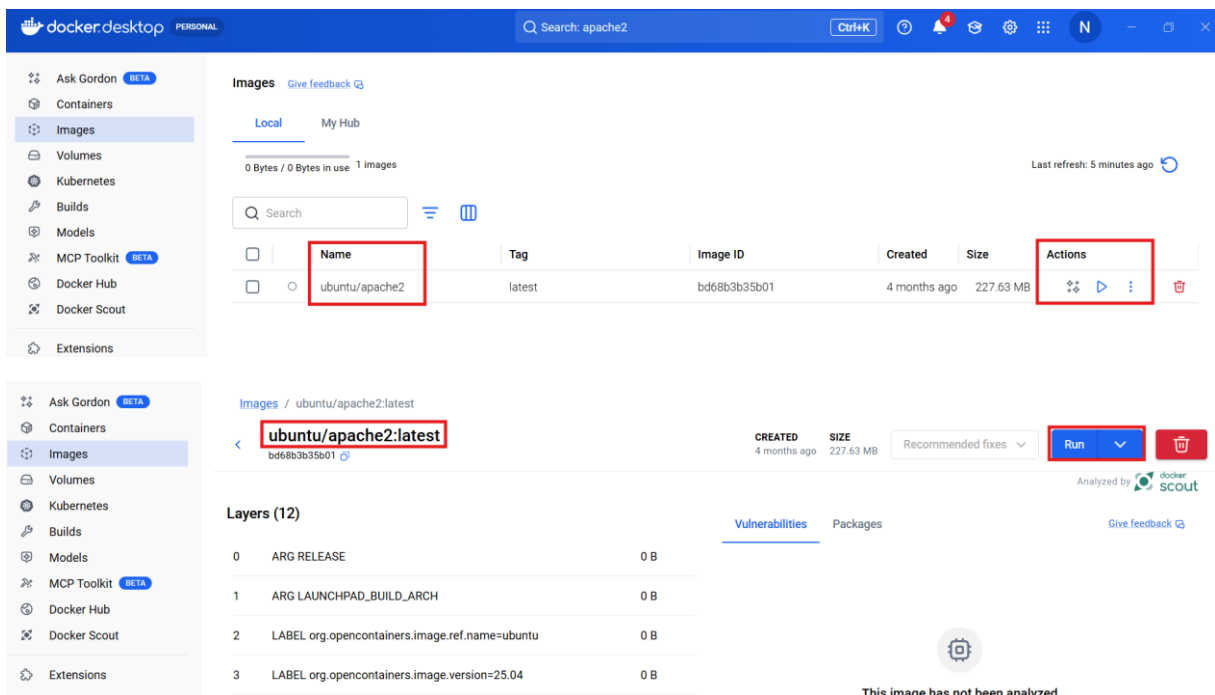
Nous cherchons une image depuis docker Desktop.



Nous installons apache2 en sélectionnant pull.



Une fois Apache installé, il apparaît dans les images.



Nous lançons un conteneur avec l'image que nous venons de télécharger en appuyant sur run.

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

Une fois lancé, nous avons accès aux logs/informations du conteneur.

Containers / apache2-container

apache2-container
7dd814a66414 ubuntu/apache2:latest
8080:80

STATUS
Running (0 seconds 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
[Wed Mar 11 13:39:17.911408 2026] [mpm_event:notice] [pid 23:tid 23] AH00489: Apache/2.4.63 (Ubuntu) configured -- resuming normal operations
[Wed Mar 11 13:39:17.912449 2026] [core:notice] [pid 23:tid 23] AH00094: Command line: '/usr/sbin/apache2 -D FOREGROUND'

Nous avons une invite de commande directement depuis le conteneur.

The screenshot shows the 'Exec' tab selected in the Docker Desktop interface for the 'apache2-container'. The terminal window displays the following commands and their outputs:

```
# pwd
/  
# cd /etc/apache2  
# ls  
apache2.conf  conf-available  conf-enabled  envvars  magic  mods-available  mods-enabled  ports.conf  sites-available  sites-enabled  
# cd /var/www/html  
# ls  
index.html  
#
```

Nous avons un statut pour vérifier l'état du conteneur.

The screenshot shows the 'Stats' tab in the Docker Desktop interface for the 'apache2-container'. It displays four performance graphs:

- CPU usage:** 0.01%. The graph shows a very low and stable CPU usage over time.
- Memory usage:** 8.38MB / 7.64GB. The graph shows a very low and stable memory usage over time.
- Disk read/write:** 7.3MB / 4.1KB. The graph shows a very low and stable disk usage over time.
- Network I/O:** 1.7KB / 126B. The graph shows a very low and stable network usage over time.

Le conteneur que nous venons de créer apparaît bien dans l'onglet Containers.

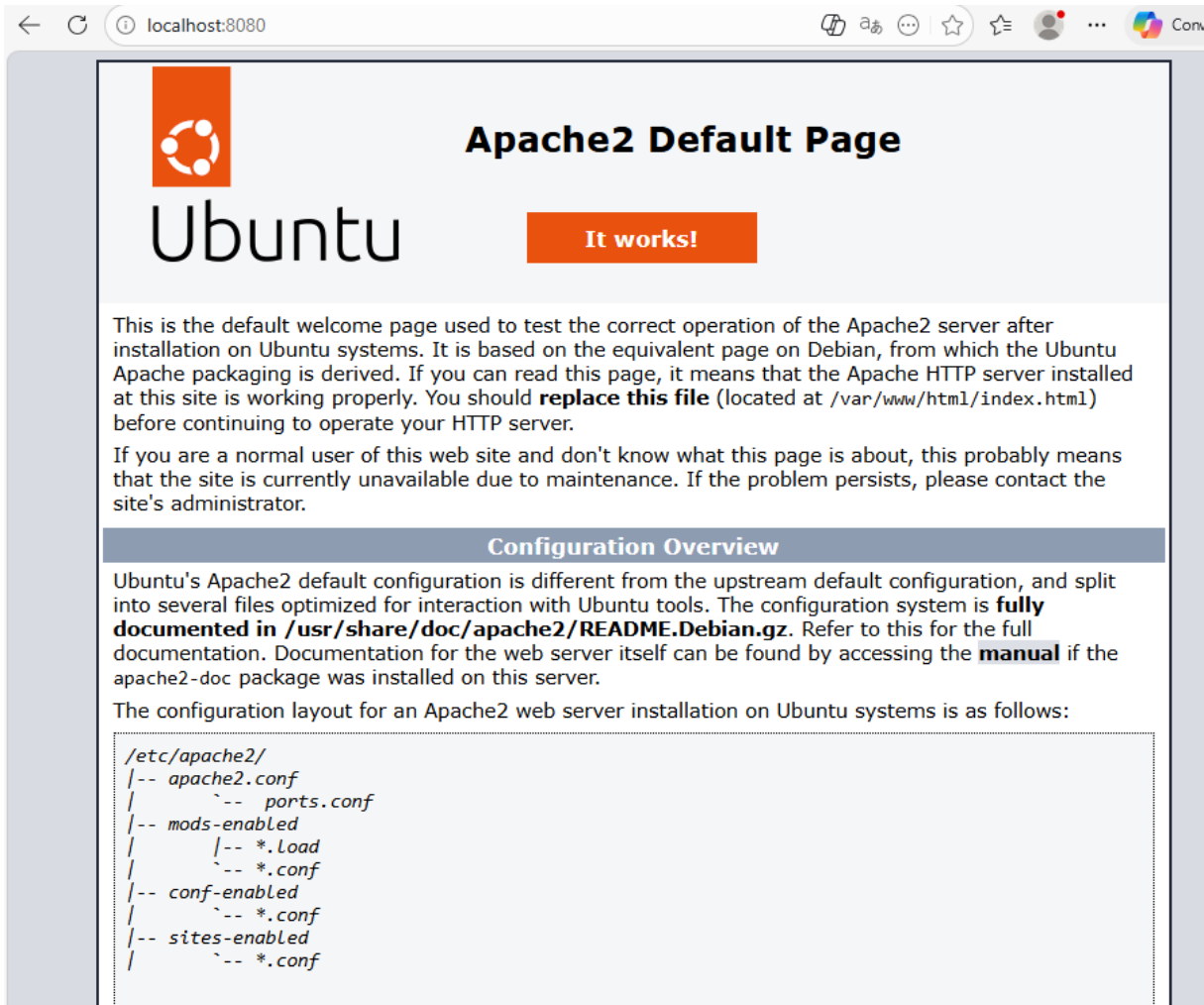
The screenshot shows the 'Containers' tab in the Docker Desktop interface. The 'Containers' menu item is highlighted in the left sidebar. The main area shows the following summary:

- Container CPU usage: 0.01% / 800% (8 CPUs available)
- Container memory usage: 8.38MB / 7.46GB

Below the summary is a table of containers:

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	● apache2-container	7dd814a66414	ubuntu/apache2:latest	8080:80	0.01%	10 minutes ago	

Nous lançons le container en localhost pour vérifier le bon fonctionnement de celui-ci.



Nous arrêtons et supprimons le container et l'image.

Delete container?

The 'apache2-container' container is selected for deletion. Any anonymous volumes associated with this container are also deleted.

Cancel

Delete forever

Delete image?

The 'ubuntu/apache2:latest' image is selected for deletion.

Cancel

Delete forever

7. Quatrième test.

Nous cherchons l'image hello-world depuis ubuntu.

```
sio@Nicolau:~$ 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: sha256:85404b3c53951c3ff5d40de0972b1bb21fafa2e8daa235355baf44f33db9dbdd
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.

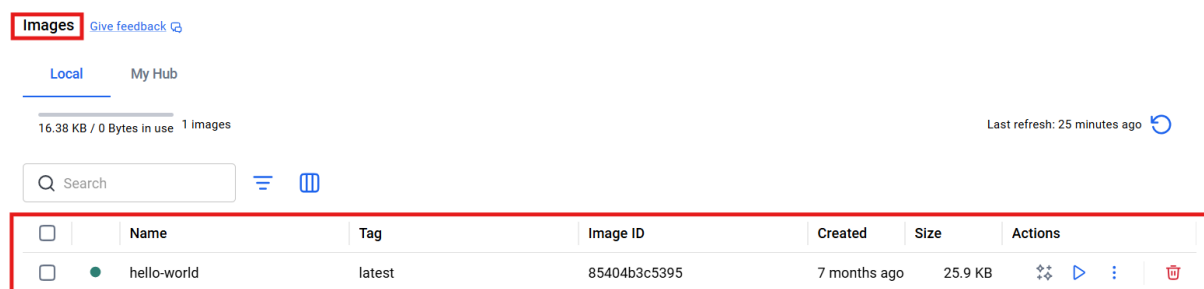
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

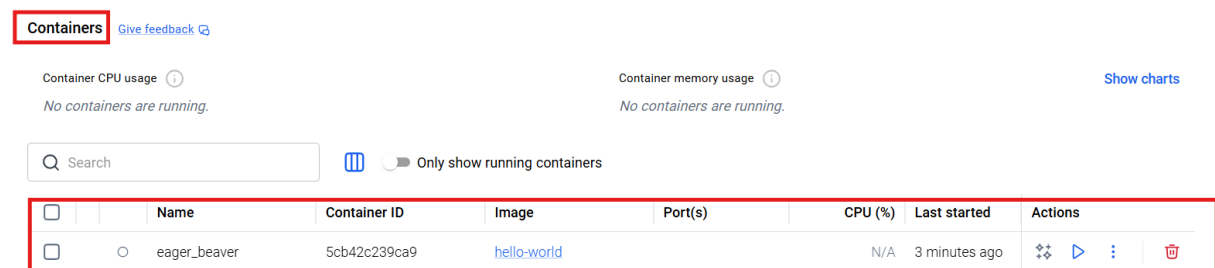
For more examples and ideas, visit:
https://docs.docker.com/get-started/

sio@Nicolau:~$
```

Une fois installé, il apparaît dans l'onglet Image de Docker.



Nous constatons que le container a lui aussi été créé depuis l'image.



Si nous allons vérifier dans les logs, celui-ci s'est éteint à l'affichage du message.

The screenshot shows the Docker Desktop interface for a container named 'eager_beaver'. The container ID is '5cb42c239ca9' and it is using the 'hello-world:latest' image. The 'Logs' tab is selected, showing the following output:

```
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/
```

Depuis le site de Docker, nous pouvons voir que l'image de hello-world a été récupérée depuis GitHub.

The screenshot shows the Docker Hub page for the 'hello-world' image. The URL in the browser is https://hub.docker.com/_/hello-world. The page displays the following information:

- Shared Tags:**
 - latest:
 - linux
 - nanoserver-ltsc2025
 - nanoserver-ltsc2022
 - nanoserver:
 - nanoserver-ltsc2025
 - nanoserver-ltsc2022
- Quick reference (cont.):**
 - Where to file issues:** <https://github.com/docker-library/hello-world/issues>
 - Supported architectures:** (more info) [amd64](#), [arm32v5](#), [arm32v6](#), [arm32v7](#), [arm64v8](#), [i386](#), [mips64le](#), [ppc64le](#), [riscv64](#), [s390x](#), [windows-amd64](#)
 - Published image artifact details:** [repo-info](#) [repo's](#) [repos/hello-world/](#) [directory](#) [\(history\)](#) (image metadata, transfer size, etc)

Additional sections on the right include 'This week's pulls' (2,116,845 pulls from Feb 23 to Mar 1) and 'About Official Images'.

Voici les fichiers qui servent à créer l'image hello.

https://github.com/docker-library/hello-world/tree/master/amd64/nanoserver-...
docker-library / hello-world Public
Notifications Fork 611 Star 538
Code Issues 3 Pull requests Actions Projects Wiki Security Insights
master hello-world / amd64 / nanoserver-ltsc2025 /
Go to file
tianon Remove vanity variant support completely 1f2052d · 4 months ago
Name Last commit message Last commit date
..
Dockerfile Remove vanity variant support completely 4 months ago
hello.txt Remove vanity variant support completely 4 months ago

https://github.com/docker-library/hello-world/blob/master/amd64/nanoserver-...
Code Issues 3 Pull requests Actions Projects Wiki Security Insights
master hello-world / amd64 / nanoserver-ltsc2025 / hello.txt
Go to file
tianon Remove vanity variant support completely 1f2052d · 4 months ago
22 lines (16 loc) · 894 Bytes
Code Blame Raw Copy Download Edit
1
2 Hello from Docker!
3 This message shows that your installation appears to be working correctly.
4
5 To generate this message, Docker took the following steps:
6 1. The Docker client contacted the Docker daemon.
7 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
8 (windows-amd64, nanoserver-ltsc2025)
9 3. The Docker daemon created a new container from that image which runs the
10 executable that produces the output you are currently reading.
11 4. The Docker daemon streamed that output to the Docker client, which sent it
12 to your terminal.
13
14 To try something more ambitious, you can run a Windows Server container with:
15 PS C:\> docker run -it mcr.microsoft.com/windows/servercore:ltsc2025 powershell
16
17 Share images, automate workflows, and more with a free Docker ID:
18 https://hub.docker.com/
19
20 For more examples and ideas, visit:
21 https://docs.docker.com/get-started/

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

Nous listons toutes les images que nous avons depuis Ubuntu.

```
sio@Nicolau:~$ docker images
```

IMAGE	ID	DISK USAGE	CONTENT SIZE	EXTRA
hello-world:latest	85404b3c5395	25.9kB	9.52kB	U

```
sio@Nicolau:~$
```

Nous lançons un conteneur en mode interactif, cela nous permettra de taper des commandes à l'intérieur.

```
sio@Nicolau:~$ docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
01d7766a2e4a: Pull complete
fd8cda969ed2: Download complete
Digest: sha256:d1e2e92c075e5ca139d51a140fff46f84315c0fdce203eab2807c7e495eff4f9
Status: Downloaded newer image for ubuntu:latest
root@dafdaf918553:/#
```

Nous constatons que l'image et le centenaire que nous venons de créer depuis Docker.

Containers [Give feedback](#)

Container CPU usage 0.00% / 800% (8 CPUs available) Container memory usage 924KB / 7.46GB [Show charts](#)

Search Only show running containers

Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
eager_beaver	5cb42c239ca9	hello-world		0%	20 minutes ago	↕ ▶ ⋮ 🗑
silly_einstein	dafdaf918553	ubuntu		0%	2 minutes ago	↕ ▶ ⋮ 🗑

Images [Give feedback](#)

Local My Hub

87.6 MB / 0 Bytes in use 2 Images Last refresh: 46 minutes ago [🔄](#)

Search

Name	Tag	Image ID	Created	Size	Actions
hello-world	latest	85404b3c5395	7 months ago	25.9 KB	↕ ▶ ⋮ 🗑
ubuntu	latest	d1e2e92c075e	29 days ago	119.26 MB	↕ ▶ ⋮ 🗑

Nous pouvons vérifier les conteneurs en cours d'exécution avec la commande

```
PS C:\Users\noli8> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
dafdaf918553	ubuntu	"bash"	5 minutes ago	Up 5 minutes		silly_einstein

```
PS C:\Users\noli8>
```

Noah NICOLAU

Nous tapons quelques commandes depuis la fenêtre.

```
root@dafdaf918553:/# pwd
/
root@dafdaf918553:/# ls
bin dev home lib64 mnt proc run srv tmp var
boot etc lib media opt root sbin sys usr
root@dafdaf918553:/# top
top - 14:19:58 up 50 min, 0 user, load average: 0.01, 0.00, 0.00
Tasks: 2 total, 1 running, 1 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7824.2 total, 6730.7 free, 815.3 used, 438.9 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 7008.9 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
   1 root        20   0   4588 3968 3456 S  0.0   0.0   0:00.02 bash
  10 root        20   0   8864 5120 3072 R  0.0   0.1   0:00.01 top
```

Nous créons un fichier et vérifions sa présence.

```
root@dafdaf918553:/# touch fichier_test_persistence
root@dafdaf918553:/# ls
bin dev fichier_test_persistence lib media opt root sbin sys usr
boot etc home lib64 mnt proc run srv tmp var
root@dafdaf918553:/#
```

Nous supprimons le dossier /home et arrêtons la machine.

```
root@dafdaf918553:/# rm -fr /home
root@dafdaf918553:/# ls
bin dev fichier_test_persistence lib64 mnt proc run srv tmp var
boot etc lib
root@dafdaf918553:/# exit
exit
sio@Nicolau:~$
```

Nous constatons qu'aucune machine n'est en train de tourner.

Containers [Give feedback](#)

Container CPU usage ⓘ No containers are running. Container memory usage ⓘ No containers are running. [Show charts](#)

Search Only show running containers

<input type="checkbox"/>	<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	<input type="radio"/>	eager_beaver	5cb42c239ca9	hello-world		N/A	32 minutes ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	<input type="radio"/>	silly_einstein	dafdaf918553	ubuntu		N/A	14 minutes ago	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

```
PS C:\Users\noli8> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
PS C:\Users\noli8> |
```

Nous pouvons voir tous les containers avec le -a.

```
PS C:\Users\noli8> docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
dafdaf918553  ubuntu   "bash"    16 minutes ago  Exited (0) 3 minutes ago
instein       5cb42c239ca9  hello-world  "/hello"    34 minutes ago  Exited (0) 34 minutes ago
eager_bea
PS C:\Users\noli8>
```

Nous redémarrons le container avec la commande docker start et son nom ou container ID et nous rentrons en ligne de commande -ai.

```
sio@Nicolau:~$ docker start daf
daf
sio@Nicolau:~$ docker stop daf
daf
```

```
sio@Nicolau:~$ docker start -ai daf
root@dafdaf918553:/# ls
bin  dev  fichier_test_persistence  lib64  mnt  proc  run  srv  tmp  var
boot  etc  lib                        media  opt  root  sbin  sys  usr
root@dafdaf918553:/# exit
exit
sio@Nicolau:~$
```

Une manière d’y accéder quand le container est lancé est avec la commande docker exec -it id bash.

```
sio@Nicolau:~$ docker start daf
daf
sio@Nicolau:~$ docker exec -it daf bash
root@dafdaf918553:/# exit
exit
sio@Nicolau:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
dafdaf918553  ubuntu   "bash"    27 minutes ago  Up 2 minutes
silly_einstein
sio@Nicolau:~$ docker stop daf
daf
sio@Nicolau:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
sio@Nicolau:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
dafdaf918553  ubuntu   "bash"    27 minutes ago  Exited (137) 15 seconds ago
silly_einstein
5cb42c239ca9  hello-world  "/hello"    46 minutes ago  Exited (0) 46 minutes ago
eager_bea
sio@Nicolau:~$
```

Nous pouvons supprimer les containers depuis une invite de commande avec les commandes :

```
sio@Nicolau:~$ docker rm eager_bea
eager_bea
sio@Nicolau:~$ docker rm daf
daf
sio@Nicolau:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
sio@Nicolau:~$ █
```

Nous créons un conteneur qui se supprime à la fermeture de celui-ci avec `-rm`.

```
sio@Nicolau:~$ docker run -it --rm ubuntu bash
root@62e66de08fc5:/# ls
bin  dev  home  lib64  mnt  proc  run  srv  tmp  var
boot  etc  lib  media  opt  root  sbin  sys  usr
root@62e66de08fc5:/# exit
exit
sio@Nicolau:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
sio@Nicolau:~$
```

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

Nous allons créer une image comportant les modifications faites depuis un conteneur. Pour ça, on recrée une image que nous nommons.

```
sio@Nicolau:~$ docker run -it --name test_modification ubuntu
root@d73ecc9e03df:/# cd home
root@d73ecc9e03df:/home# touch fichier_test
root@d73ecc9e03df:/home# cd ..
root@d73ecc9e03df:/# ls
bin  dev  home  lib64  mnt  proc  run  srv  tmp  var
boot  etc  lib  media  opt  root  sbin  sys  usr
root@d73ecc9e03df:/# cd lib
root@d73ecc9e03df:/lib# ls
apt  init  lsb  os-release  systemd  udev
dpkg  locale  mime  sysctl.d  tmpfiles.d  x86_64-linux-gnu
root@d73ecc9e03df:/lib# rm os-release
root@d73ecc9e03df:/lib# exit
exit
sio@Nicolau:~$ █
```

Nous affichons les endroits qui ont été modifiés.

```
sio@Nicolau:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
sio@Nicolau:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
d73ecc9e03df  ubuntu   "/bin/bash"             10 days ago   Exited (0) 10 day
s ago         test_modification
sio@Nicolau:~$ docker diff test_modification
C /root
A /root/.bash_history
C /usr
C /usr/lib
D /usr/lib/os-release
C /home
A /home/fichier_test
sio@Nicolau:~$ █
```

Nous nous renseignons sur la façon de créer une image avec la commande commit.

```
sio@Nicolau:~$ 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@Nicolau:~$
```

Puis nous créons une image du conteneur.

```
sio@Nicolau:~$ docker commit test_modification ubuntu:1.0
sha256:8088e35dd2e0e918ff7fd9073da69c23ff1c91ac5468582b60a97ccec44cd16
sio@Nicolau:~$
```

Nous constatons que cette image apparaît bien dans les images que nous avons téléchargées.

```
sio@Nicolau:~$ docker images
```

IMAGE	ID	DISK USAGE	CONTENT SIZE	In Use
hello-world:latest	85404b3c5395	25.9kB	9.52kB	
ubuntu:latest	d1e2e92c075e	119MB	31.7MB	U
ubuntumodif:1.0	8088e35dd2e0	117MB	29.7MB	

```
sio@Nicolau:~$
```

Nous détruisons le conteneur pour en recréer un nouveau avec l'image que nous avons créée depuis le précédent.

```
sio@Nicolau:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS   NAMES
d73ecc9e03df   ubuntu   "/bin/bash"            10 days ago    Exited (0) 10 day
s ago         test_modification
sio@Nicolau:~$ docker rm test_modification
test_modification
sio@Nicolau:~$ docker run -it --rm --name test ubuntumodif:1.0
root@de6127da9092:/# ls /home
fichier_test  ubuntu
root@de6127da9092:/# exit
exit
sio@Nicolau:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS   NAMES
sio@Nicolau:~$
```

10. Première approche des volumes.

Nous créons un répertoire web sur Ubuntu.

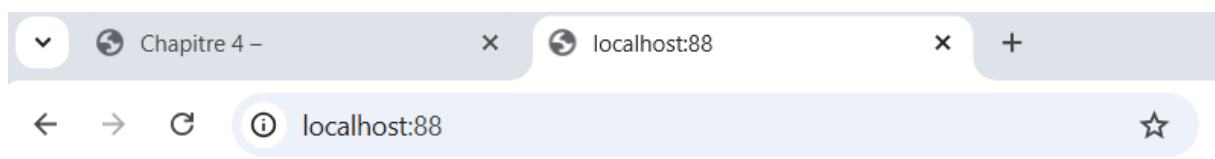
```
sio@Nicolau:~$ pwd
/home/sio
sio@Nicolau:~$ mkdir web
sio@Nicolau:~$ cd web
sio@Nicolau:~/web$ nano index.html
```

```
GNU nano 7.2 index.html
<html>
<body>
<h1> Page web Nginx conteneur Docker</h1>
</body>
</html>
```

Nous mettons en place un mappage de volume pour qu'un container puisse pointer directement sur le répertoire.

```
sio@Nicolau:~$ 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
9baba07a35b6: Pull complete
ec781dee3f47: Pull complete
980067d12da2: Pull complete
4174e33a2c9e: Pull complete
6b40784e4837: Pull complete
f0b77348d9b0: Pull complete
0289d65812c3: Pull complete
00238b7dc6b2: Download complete
c96ca1f4ddf7: Download complete
Digest: sha256:dec7a90bd0973b076832dc56933fe876bc014929e14b4ec49923951405370112
Status: Downloaded newer image for nginx:latest
f254449191b106ff66715cc0e5a34ddd98a755730f31b8408df2ddb67825b033
sio@Nicolau:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS      PORTS
f254449191b1   nginx    "/docker-entrypoint..." 24 seconds ago Up
23 seconds    0.0.0.0:88->80/tcp, [::]:88->80/tcp   web
```

Test depuis un navigateur :



Page web Nginx conteneur Docker

Noah NICOLAU

Nous affichons les logs pour montrer que c'est bien notre serveur local qui tourne.

```
sio@Nicolau:~$ 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
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
2026/03/22 12:52:15 [notice] 1#1: using the "spoll" event method
2026/03/22 12:52:15 [notice] 1#1: nginx/1.29.6
2026/03/22 12:52:15 [notice] 1#1: built by gcc 14.2.0 (Debian 14.2.0-19)
2026/03/22 12:52:15 [notice] 1#1: OS: Linux 6.6.87.2-microsoft-standard-WSL2
2026/03/22 12:52:15 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2026/03/22 12:52:15 [notice] 1#1: start worker processes
2026/03/22 12:52:15 [notice] 1#1: start worker process 29
2026/03/22 12:52:15 [notice] 1#1: start worker process 30
2026/03/22 12:52:15 [notice] 1#1: start worker process 31
2026/03/22 12:52:15 [notice] 1#1: start worker process 32
2026/03/22 12:52:15 [notice] 1#1: start worker process 33
2026/03/22 12:52:15 [notice] 1#1: start worker process 34
2026/03/22 12:52:15 [notice] 1#1: start worker process 35
2026/03/22 12:52:15 [notice] 1#1: start worker process 36
172.17.0.1 -- [22/Mar/2026:12:54:04 +0000] "GET / HTTP/1.1" 200 73 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/146.0.0.0 Safari/537.36" "-"
2026/03/22 12:54:04 [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 -- [22/Mar/2026:12:54:04 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://localhost:88/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/146.0.0.0 Safari/537.36" "-"
sio@Nicolau:~$
```

Nous arrêtons et supprimons le container.

```
sio@Nicolau:~$ docker rm -fv web
web
sio@Nicolau:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
sio@Nicolau:~$
```

Nous constatons que la suppression du container n'a pas supprimé le fichier index.

```
sio@Nicolau:~$ cd web
sio@Nicolau:~/web$ ls
index.html
sio@Nicolau:~/web$
```

11. Gestion des volumes en écriture.

Nous téléchargeons mongodb-community-server depuis Ubuntu.

```
sio@Nicolau:~$ docker pull mongodb/mongodb-community-server:latest
latest: Pulling from mongodb/mongodb-community-server
ee6c644e0c20: Pull complete
fe2fbbbc7ec0: Pull complete
4f4fb700ef54: Pull complete
568a1e397067: Pull complete
96c832531c38: Pull complete
f3e0282bbb14: Pull complete
2e4f7f4523be: Pull complete
a9af3f48621c: Pull complete
d68094177302: Pull complete
6989f79dc9b5: Pull complete
6ad2e513b847: Pull complete
Digest: sha256:7266392360c651bf99ad725eb78ee17d723e51a8f4cb7c3382aa89e7b2c3eee3
Status: Downloaded newer image for mongodb/mongodb-community-server:latest
docker.io/mongodb/mongodb-community-server:latest
sio@Nicolau:~$
```

Noah NICOLAU

Nous lançons un conteneur depuis cette image.

```
sio@Nicolau:~$ docker run -d -p 27017:27017 -e MONGODB_INITDB_ROOT_USERNAME=sio -e MONGODB_I
NITDB_ROOT_PASSWORD=password1234 --name mongodb mongodb/mongodb-community-server:latest
d19482ce8cb3f8a495867bb90d34afb64563ecfd274a897b0735ba667221694b
sio@Nicolau:~$
```

Nous téléchargeons MongoDB sur notre machine.

The screenshot shows the MongoDB website's download page for the MongoDB Shell. On the left is a sidebar with navigation links: Atlas MongoDB, MongoDB Enterprise Avancé, MongoDB Community Edition, Outils, Fournisseur Terraform MongoDB Atlas, **MongoDB Shell** (highlighted with a red box), MongoDB Compass (interface graphique), Atlas CLI, Opérateur Atlas Kubernetes, Interface de ligne de commande MongoDB pour Cloud Manager et Ops Manager, Mongosync, and Migrateur relationnel MongoDB. The main content area has a search icon and a menu icon. Below the navigation is a section titled 'Apprendre encore plus' (highlighted with a red box) containing three dropdown menus: 'Version' (2.8.1), 'Plate-Forme' (Windows x64 (10+)), and 'Emballer' (fermeture éclair). Below these are two buttons: 'Télécharger' (green) and 'Copier le lien' (black). At the bottom of the main content area is a 'Plus d'options' link with three dots.

Noah NICOLAU

Nous nous connectons au serveur et ajoutons une collection et un document.

```
PS C:\WINDOWS\system32> mongosh mongodb://sio:password1234@localhost:27017
Current Mongosh Log ID: 69c40ae797f8a5e5ba114029
Connecting to:      mongodb://<credentials>@localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.8.1
Using MongoDB:     8.2.6
Using Mongosh:     2.8.1

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

-----
The server generated these startup warnings when booting
2026-03-25T16:05:50.764+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2026-03-25T16:05:51.212+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
2026-03-25T16:05:51.212+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
2026-03-25T16:05:51.212+00:00: We suggest setting the contents of sysfsFile to 0.
2026-03-25T16:05:51.212+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('69c40c5897f8a5e5ba11402a')
}
ma_bdd> db.etudiants.find();
[
  {
    _id: ObjectId('69c40c5897f8a5e5ba11402a'),
    nom: 'smet',
    prenom: 'jp'
  }
]
ma_bdd> exit
PS C:\WINDOWS\system32>
```

Nous affichons la base depuis Ubuntu.

```
sio@nicolau:~$ docker inspect mongodb
[
  {
    "Id": "d19482ce8cb3f8a495867bb90d34afb64563ecfd274a897b0735ba667221694b",
    "Created": "2026-03-25T14:27:42.622842578Z",
    "Path": "python3",
    "Args": [
      "/usr/local/bin/docker-entrypoint.py",
      "mongod"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 420,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2026-03-25T16:05:49.354496557Z",
      "FinishedAt": "2026-03-25T16:05:23.960272899Z"
    },
    "Image": "sha256:7266392360c651bf99ad725eb78ee17d723e51a8f4cb7c3382aa89e7b2c3eee3",
    "ResolvConfPath": "/var/lib/docker/containers/d19482ce8cb3f8a495867bb90d34afb64563ecfd274a897b0735ba667221694b/resolv.conf",
    "HostnamePath": "/var/lib/docker/containers/d19482ce8cb3f8a495867bb90d34afb64563ecfd274a897b0735ba667221694b/hostname",
    "HostsPath": "/var/lib/docker/containers/d19482ce8cb3f8a495867bb90d34afb64563ecfd274a897b0735ba667221694b/hosts",
    "LogPath": "/var/lib/docker/containers/d19482ce8cb3f8a495867bb90d34afb64563ecfd274a897b0735ba667221694b/d19482ce8cb3f8a495867bb90d34afb64563ecfd274a897b0735ba667221694b-json.log",
    "Name": "/mongodb",
    "RestartCount": 0,
    "Driver": "overlayfs",
    "Platform": "linux",
    "MountLabel": "",
    "ProcessLabel": "",
    "AppArmorProfile": "",
    "ExecIDs": null,
    "HostConfig": {

```

```

"Mounts": [
  {
    "Type": "volume",
    "Name": "aec17443f29935afa5734e1b08903961daea1698a5550bf597b036ba01e8eb39",
    "Source": "/var/lib/docker/volumes/aec17443f29935afa5734e1b08903961daea1698a5550bf597b036ba01e8eb39/_data",
    "Destination": "/data/configdb",
    "Driver": "local",
    "Mode": "",
    "RW": true,
    "Propagation": ""
  },
  {
    "Type": "volume",
    "Name": "854f27786fa3348280035f308f8f7eaa5d128077c1cc98bf7b72f91c0128a3c4",
    "Source": "/var/lib/docker/volumes/854f27786fa3348280035f308f8f7eaa5d128077c1cc98bf7b72f91c0128a3c4/_data",
    "Destination": "/data/db",
    "Driver": "local",
    "Mode": "",
    "RW": true,
    "Propagation": ""
  }
]

```

Nous regardons les données du conteneur puis l'arrêtons.

```

sio@Nicolau:~$ docker exec -it mongodb bash
mongodb@d19482ce8cb3:/$ cd /data
mongodb@d19482ce8cb3:/data$ ls
configdb db
mongodb@d19482ce8cb3:/data$ cd db
mongodb@d19482ce8cb3:~$ ls
WiredTiger
WiredTiger.lock
WiredTiger.turtle
WiredTiger.wt
WiredTigerHS.wt
_mdb_catalog.wt
_tmp
collection-2edd4963-63f0-48c0-a92a-a8b58c9eca2d.wt
collection-4db06c63-6e47-4fc0-8491-a01b59369352.wt
collection-4fa0814d-0699-4f97-9644-85eb0fdd3476.wt
collection-6bf0eac4-098d-4541-ae39-599276e3b832.wt
collection-e6e640ce-589d-4995-af4a-3d7077f181cd.wt
diagnostic.data
index-0a1fc4bf-6679-445e-b4d7-fc6490ecc1d8.wt
index-61645cf6-21af-41c5-ab11-3b63626547f6.wt
index-64989ccc-4d13-4231-ad84-7624d58cd006.wt
index-734a042e-fb32-4313-b08c-da5923b27051.wt
index-7591c918-8544-4311-b2f6-c90a5322bec1.wt
index-ac810c3f-65f7-4af5-97bb-add47cf33fd0.wt
index-cf86aded-2a8b-4523-a0ec-7201b089ee8a.wt
journal
mongod.lock
sizeStorer.wt
storage.bson
mongodb@d19482ce8cb3:~$ exit
exit
sio@Nicolau:~$

```

Nous supprimons le conteneur.

```

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

```

Nous créons un volume que nous nommons mon_volume_mongodb.

```
sio@Nicolau:~$ docker volume create mon_volume_mongodb
mon_volume_mongodb
sio@Nicolau:~$ docker volume ls
DRIVER      VOLUME NAME
local       854f27786fa3348280035f308f8f7eaa5d128077c1cc98bf7b72f91c0128a3c4
local       aec17443f29935afa5734e1b08903961daea1698a5550bf597b036ba01e8eb39
local       mon_volume_mongodb
sio@Nicolau:~$
```

```
sio@Nicolau:~$ docker volume inspect mon_volume_mongodb
[
  {
    "CreatedAt": "2026-03-25T16:44:14Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/mon_volume_mongodb/_data"
    "Name": "mon_volume_mongodb",
    "Options": null,
    "Scope": "local"
  }
]
sio@Nicolau:~$
```

Nous recréons un container et ajoutons une table et des informations.

```
sio@Nicolau:~$ 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
fb2785f47f4014a5b3828c4be2424c93ab96e7c9f2da793825182e820baf8124
sio@Nicolau:~$
```

```
PS C:\WINDOWS\system32> mongosh mongodb://sio:password1234@localhost:27017
Current Mongosh Log ID: 69c798e38c86ac2dd7114029
Connecting to:  mongodb://<credentials>@localhost:27017/?directConnection=true&serverSelectionTi
meoutMS=2000&appName=mongosh+2.8.1
Using MongoDB:      8.2.6
Using Mongosh:      2.8.1

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

-----
The server generated these startup warnings when booting
2026-03-28T09:00:55.138+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger s
torage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2026-03-28T09:00:55.522+00:00: For customers running the current memory allocator, we suggest changin
g the contents of the following sysfsFile
2026-03-28T09:00:55.522+00:00: For customers running the current memory allocator, we suggest changin
g the contents of the following sysfsFile
2026-03-28T09:00:55.522+00:00: We suggest setting the contents of sysfsFile to 0.
2026-03-28T09:00:55.522+00:00: We suggest setting swappiness to 0 or 1, as swapping can cause perform
ance 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('69c799bb8c86ac2dd711402a')
}
ma_bdd>
```

```
sio@Nicolau:~$ docker ps
CONTAINER ID   IMAGE                                STATUS      PORTS                               COMMAND
CREATED          NAMES
fb2785f47f40   mongodb/mongodb-community-server:latest  Up 4 minutes   0.0.0.0:27017->27017/tcp, [
::]:27017->27017/tcp   mongodb
sio@Nicolau:~$
```

Nous arrêtons et supprimons le conteneur.

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

Une fois le conteneur supprimé, on en redémarre un nouveau pour vérifier si les modifications y sont toujours présentes.

```
sio@Nicolau:~$ 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
4bb45e0fc9ef50789a858c379e819d2b27f7edf1cac7377a12ea8285ebc70eda
sio@Nicolau:~$
```

```
PS C:\WINDOWS\system32> mongosh mongodb://sio:password1234@localhost:27017
Current Mongosh Log ID: 69c79c23c92a68dc05114029
Connecting to:   mongodb://<credentials>@localhost:27017/?directConnection=true&serverSelectionTi
meoutMS=2000&appName=mongosh+2.8.1
Using MongoDB:  8.2.6
Using Mongosh:  2.8.1
mongosh 2.8.2 is available for download: https://www.mongodb.com/try/download/shell

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

-----
The server generated these startup warnings when booting
  2026-03-28T09:14:47.385+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger s
torage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
  2026-03-28T09:14:47.807+00:00: For customers running the current memory allocator, we suggest changin
g the contents of the following sysfsFile
  2026-03-28T09:14:47.807+00:00: For customers running the current memory allocator, we suggest changin
g the contents of the following sysfsFile
  2026-03-28T09:14:47.807+00:00: We suggest setting the contents of sysfsFile to 0.
  2026-03-28T09:14:47.807+00:00: We suggest setting swappiness to 0 or 1, as swapping can cause perform
ance problems.
-----

test> use("ma_bdd");
switched to db ma_bdd
ma_bdd> db.etudiants.find();
[
  {
    _id: ObjectId('69c799bb8c86ac2dd711402a'),
    nom: 'smet',
    prenom: 'jp'
  }
]
ma_bdd>
```

Noah NICOLAU

Après avoir vérifié, nous supprimons la machine.

```
sio@Nicolau:~$ docker rm -fv mongodb
mongodb
sio@Nicolau:~$
```

Nous ajoutons une ligne pour la persistance.

```
ma_bdd> exit
PS C:\WINDOWS\system32> mongosh mongodb://sio:password1234@localhost:27017
Current Mongosh Log ID: 69c79ecfd3930dbf6c114029
Connecting to:   mongodb://<credentials>@localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.8.1
Using MongoDB:  8.2.6
Using Mongosh:  2.8.1
mongosh 2.8.2 is available for download: https://www.mongodb.com/try/download/shell

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

-----
The server generated these startup warnings when booting
 2026-03-28T09:25:05.968+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
 2026-03-28T09:25:06.377+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
 2026-03-28T09:25:06.377+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
 2026-03-28T09:25:06.377+00:00: We suggest setting the contents of sysfsFile to 0.
 2026-03-28T09:25:06.378+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":"hallyday", "prenom":"johnny"});
{
  acknowledged: true,
  insertedId: ObjectId('69c79f32d3930dbf6c11402a')
}
ma_bdd> db.etudiants.find();
[
  {
    _id: ObjectId('69c799bb8c86ac2dd711402a'),
    nom: 'smet',
    prenom: 'jp'
  },
  {
    _id: ObjectId('69c79f32d3930dbf6c11402a'),
    nom: 'hallyday',
    prenom: 'johnny'
  }
]
ma_bdd> exit
PS C:\WINDOWS\system32>
```

```
sio@Nicolau:~$ docker rm -fv mongodb
mongodb
sio@Nicolau:~$
```

Noah NICOLAU

Nous vérifions la persistance en recréant une machine.

```
sio@Nicolau:~$ 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 e3227d5f46b3fe350249a158690b08a647f1d56a12e5fce2a4ce8c91a7ee4180
sio@Nicolau:~$
```

```
PS C:\WINDOWS\system32> mongosh mongodb://sio:password1234@localhost:27017
Current Mongosh Log ID: 69c7a13b8440073fad114029
Connecting to:      mongodb://<credentials>@localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.8.1
Using MongoDB:      8.2.6
Using Mongosh:      2.8.1
mongosh 2.8.2 is available for download: https://www.mongodb.com/try/download/shell

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

-----
The server generated these startup warnings when booting
 2026-03-28T09:31:20.830+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
 2026-03-28T09:31:21.206+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
 2026-03-28T09:31:21.206+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
 2026-03-28T09:31:21.206+00:00: We suggest setting the contents of sysfsFile to 0.
 2026-03-28T09:31:21.206+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('69c799bb8c86ac2dd711402a'),
    nom: 'smet',
    prenom: 'jp'
  },
  {
    _id: ObjectId('69c79f32d3930dbf6c11402a'),
    nom: 'hallyday',
    prenom: 'johnny'
  }
]
ma_bdd> exit
PS C:\WINDOWS\system32>
```

```
sio@Nicolau:~$ docker rm -fv mongodb
mongodb
sio@Nicolau:~$
```

Nous constatons que le volume que nous avons créé se retrouve sur Docker.

The screenshot shows the Docker Desktop interface. At the top, there's a navigation bar with the Docker logo and 'docker.desktop PERSONAL'. Below that, the 'Volumes' section is active, showing a search bar and a 'Create' button. A table lists the following volumes:

Name	Created	Size	Actions
854f27786fa3348280035f308f8f7eaa5d12	3 days ago	201.1 MB	[Refresh] [Share] [Delete]
aec17443f29935afa5734e1b08903961dae	3 days ago	0 Bytes	[Refresh] [Share] [Delete]
mon_volume_mongodb	3 days ago	201.1 MB	[Refresh] [Share] [Delete]

Volumes / mon_volume_mongodb

This section shows the details for the 'mon_volume_mongodb' volume. It indicates that the volume is 'Not in use' and was 'Created 3 days ago'. There are buttons for 'Import' and a menu icon.

Stored data Container in-use Exports

Name	Size	Last modified	Mode
_mdb_catalog.wt	36 kB	12 minutes ago	-rw-----
> _tmp	25.5 kB	12 minutes ago	drwx----
▼ .mongodb	13.5 kB	42 minutes ago	drwx----
> mongosh	13.5 kB	42 minutes ago	drwx----
collection-0b2921f3-fc28-4806-b5de-58dff6c616	36 kB	5 minutes ago	-rw-----
collection-66dfa4b1-eca1-4c72-820e-0365003e8d	32 kB	5 minutes ago	-rw-----
collection-850a6714-3e87-46bb-b991-527785268	20 kB	12 minutes ago	-rw-----
collection-92128d5a-39ed-47fb-989e-e8368cf371	20 kB	6 minutes ago	-rw-----
collection-9c0b4c22-302e-4808-8532-bcd5d9ad5	40 kB	11 minutes ago	-rw-----
> diagnostic.data	605.2 kB	4 minutes ago	drwx----
index-2813afff-13a6-4d33-b9e4-66275f0903c3.wt	32 kB	5 minutes ago	-rw-----
index-2be7b86b-7b3d-43b3-bec9-97910b3509ed.1	20 kB	12 minutes ago	-rw-----
index-503d57c2-218a-4305-947e-2d738090ed03.1	20 kB	6 minutes ago	-rw-----
index-54500fa4-0b03-4d4e-aeef-e35f33fd09a1.wt	32 kB	5 minutes ago	-rw-----

11.1. Deuxième type de volume : montage lié

Nous créons un montage lié.

```
sio@Nicolau:~$ mkdir data
sio@Nicolau:~$ docker run -d --name mongodb -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
49f9f495ee587cbb5879809c1d49f57d2cfc8bab6f070768caeccabae6d1d3cd
sio@Nicolau:~$ █
```

```
PS C:\WINDOWS\system32> mongosh mongodb://sio:password1234@localhost:27017
Current Mongosh Log ID: 69c7a4cedc40b86e83114029
Connecting to:      mongodb://<credentials>@localhost:27017/?directConnection=true&serverSelectionTi
meoutMS=2000&appName=mongosh+2.8.1
Using MongoDB:      8.2.6
Using Mongosh:      2.8.1
mongosh 2.8.2 is available for download: https://www.mongodb.com/try/download/shell

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

-----
The server generated these startup warnings when booting
2026-03-28T09:51:22.790+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger s
torage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2026-03-28T09:51:23.168+00:00: For customers running the current memory allocator, we suggest changin
g the contents of the following sysfsFile
2026-03-28T09:51:23.168+00:00: For customers running the current memory allocator, we suggest changin
g the contents of the following sysfsFile
2026-03-28T09:51:23.168+00:00: We suggest setting the contents of sysfsFile to 0.
2026-03-28T09:51:23.168+00:00: We suggest setting swappiness to 0 or 1, as swapping can cause perform
ance problems.
-----

test> use("ma_bdd");
switched to db ma_bdd
ma_bdd> db.etudiants.insertOne({"nom":"michell", "prenom":"eddy"});
{
  acknowledged: true,
  insertedId: ObjectId('69c7a578dc40b86e8311402a')
}
ma_bdd> db.etudiants.find();
[
  {
    _id: ObjectId('69c7a578dc40b86e8311402a'),
    nom: 'michell',
    prenom: 'eddy'
  }
]
ma_bdd>
```

Nous arrêtons et supprimons le conteneur.

```
sio@Nicolau:~$ docker ps
CONTAINER ID   IMAGE                                COMMAND
CREATED          STATUS          PORTS          NAMES
49f9f495ee58   mongodb/mongodb-community-server:latest "python3 /usr
/local/..."    6 minutes ago  Up 6 minutes   0.0.0.0:27017->27017/tcp, [
::]:27017->27017/tcp   mongodb
sio@Nicolau:~$ docker stop mongodb
mongodb
sio@Nicolau:~$ docker rm -v mongodb
mongodb
sio@Nicolau:~$ docker ps -a
CONTAINER ID   IMAGE                                COMMAND          CREATED          STATUS          PORTS          NAMES
sio@Nicolau:~$
```

Nous vérifions que les modifications apparaissent dans l'onglet que nous avons créé précédemment.

```
sio@Nicolau:~$ cd data
sio@Nicolau:~/data$ ls
WiredTiger
WiredTiger.lock
WiredTiger.turtle
WiredTiger.wt
WiredTigerHS.wt
_mdb_catalog.wt
_tmp
collection-20592652-3b0e-48da-a61c-a460b0a14baa.wt
collection-88c63775-a8c6-44d8-a1c7-332460a713d0.wt
collection-d680cccc-b0e2-46e4-ad53-26cc3038e37c.wt
collection-deeb474c-a01f-476e-9513-b59e861008ce.wt
collection-ee7c797e-80a8-4eb4-b015-08accfe11e61.wt
diagnostic.data
index-1ec3a192-ae4c-4d06-bd2c-488fef66d096.wt
index-705b0090-8cef-4ef9-8587-92fb8b239321.wt
index-88398ed4-4cc7-4fcc-af1d-6bd8e6787fcb.wt
index-8a763cb7-167c-4d0b-993d-0dc283211e57.wt
index-be147646-a605-497e-9206-1bcca03e899e.wt
index-cddb7aca-904f-4e66-8ddd-e8883096fba7.wt
index-e8827089-c06c-450a-bf01-20238cd73746.wt
journal
mongod.lock
sizeStorer.wt
storage.bson
sio@Nicolau:~/data$
```

Nous relançons une nouvelle instance.

```
sio@Nicolau:~$ 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
3cadf3e13e22dfc235b376e151848ed8dac82060d809d7a030c3b41c239f5926
sio@Nicolau:~$
```

```

PS C:\WINDOWS\system32> mongosh mongodb://sio:password1234@localhost:27017
Current Mongosh Log ID: 69c7a92cc0eb02ff10114029
Connecting to:   mongodb://<credentials>@localhost:27017/?directConnection=true&serverSelectionTi
meoutMS=2000&appName=mongosh+2.8.1
Using MongoDB:  8.2.6
Using Mongosh:  2.8.1
mongosh 2.8.2 is available for download: https://www.mongodb.com/try/download/shell

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

-----
The server generated these startup warnings when booting
2026-03-28T10:09:54.640+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger s
torage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2026-03-28T10:09:55.046+00:00: For customers running the current memory allocator, we suggest changin
g the contents of the following sysfsFile
2026-03-28T10:09:55.046+00:00: For customers running the current memory allocator, we suggest changin
g the contents of the following sysfsFile
2026-03-28T10:09:55.046+00:00: We suggest setting the contents of sysfsFile to 0.
2026-03-28T10:09:55.046+00:00: We suggest setting swappiness to 0 or 1, as swapping can cause perform
ance problems.
-----

test> use("ma_bdd");
switched to db ma_bdd
ma_bdd> db.etudiants.find();
[
  {
    _id: ObjectId('69c7a578dc40b86e8311402a'),
    nom: 'michell',
    prenom: 'eddy'
  }
]
ma_bdd> exit
PS C:\WINDOWS\system32>

```

Nous vérifions si un nouveau volume a été créé.

Volumes [Give feedback](#)

Create

		Name ↑	Created	Size	Actions
<input type="checkbox"/>	<input type="radio"/>	854f27786fa3348280035f308f8f7eaa5d12	3 days ago	201.1 MB	
<input type="checkbox"/>	<input checked="" type="radio"/>	ad2bfeae81695a497fa21fe3f626e5c9e89	3 minutes ago	0 Bytes	
<input type="checkbox"/>	<input type="radio"/>	aec17443f29935afa5734e1b08903961dae	3 days ago	0 Bytes	
<input type="checkbox"/>	<input type="radio"/>	mon_volume_mongodb	3 days ago	201.1 MB	

Nous supprimons le conteneur ainsi que le volume.

```

sio@Nicolau:~$ docker rm -fv mongodb2
mongodb2
sio@Nicolau:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
sio@Nicolau:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
sio@Nicolau:~$

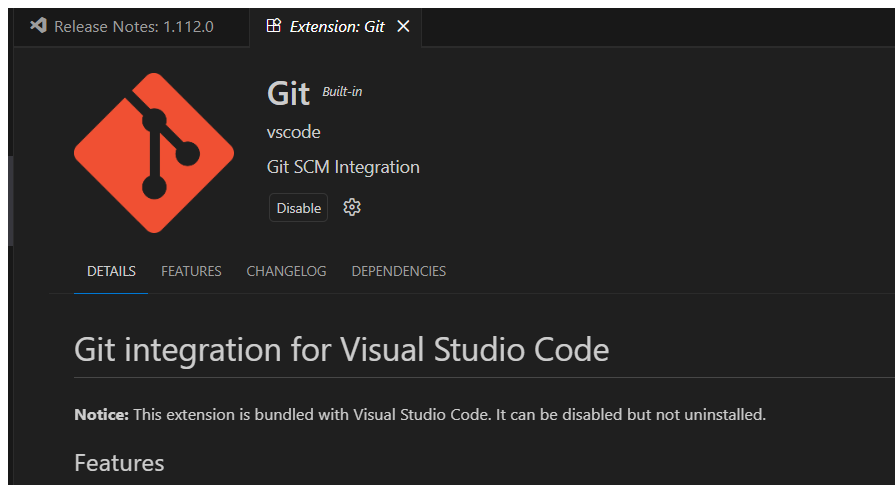
```

Volumes [Give feedback](#)

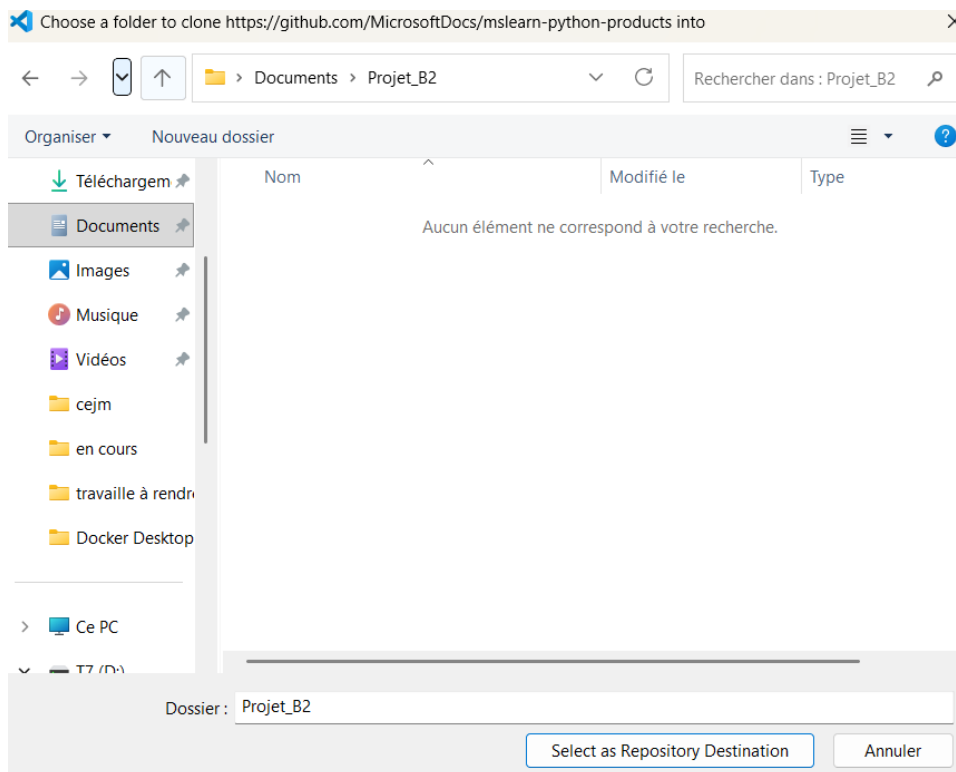
<input type="checkbox"/>	Name ↑	Created	Size	Actions
<input type="checkbox"/>	854f27786fa3348280035f308f8f7eaa5d12	3 days ago	201.1 MB	
<input type="checkbox"/>	aec17443f29935afa5734e1b08903961da6	3 days ago	0 Bytes	
<input type="checkbox"/>	mon_volume_mongodb	3 days ago	201.1 MB	

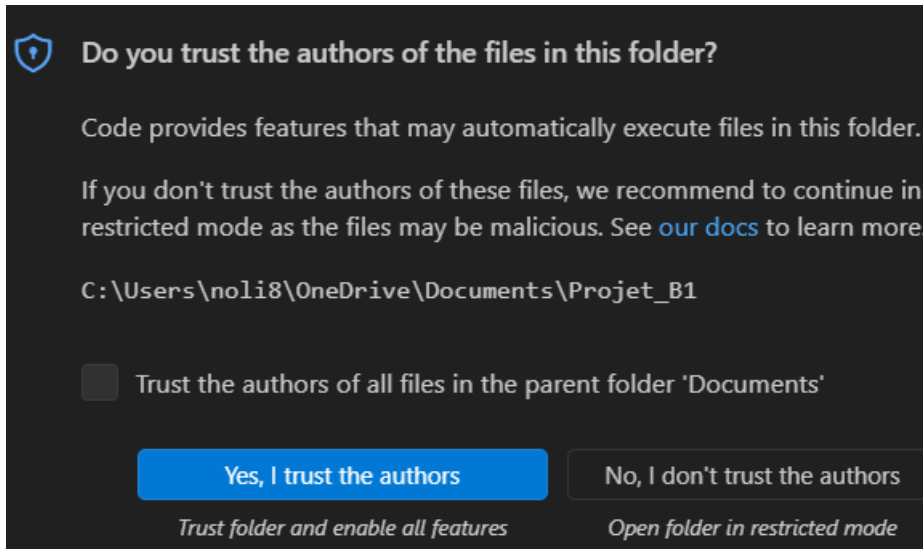
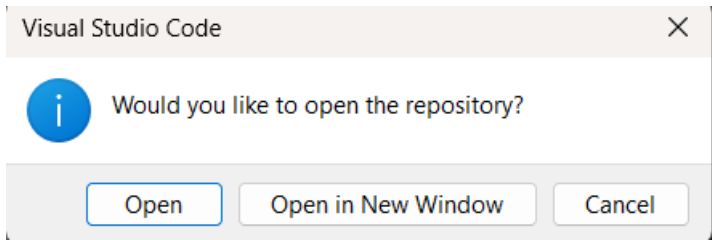
12. Utiliser un conteneur Docker et Visual Studio Code.

Depuis VSCode, nous téléchargeons l'extension GitHub.

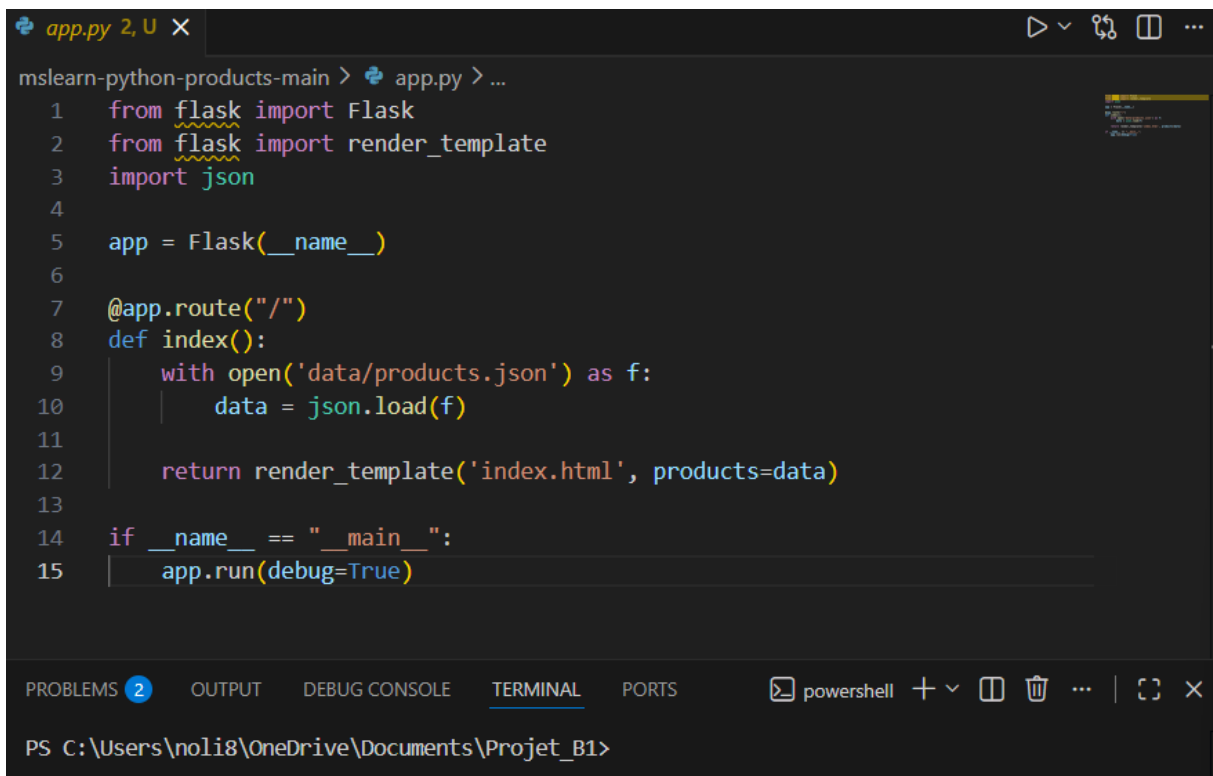


Nous sélectionnons un emplacement où le projet sera cloné.

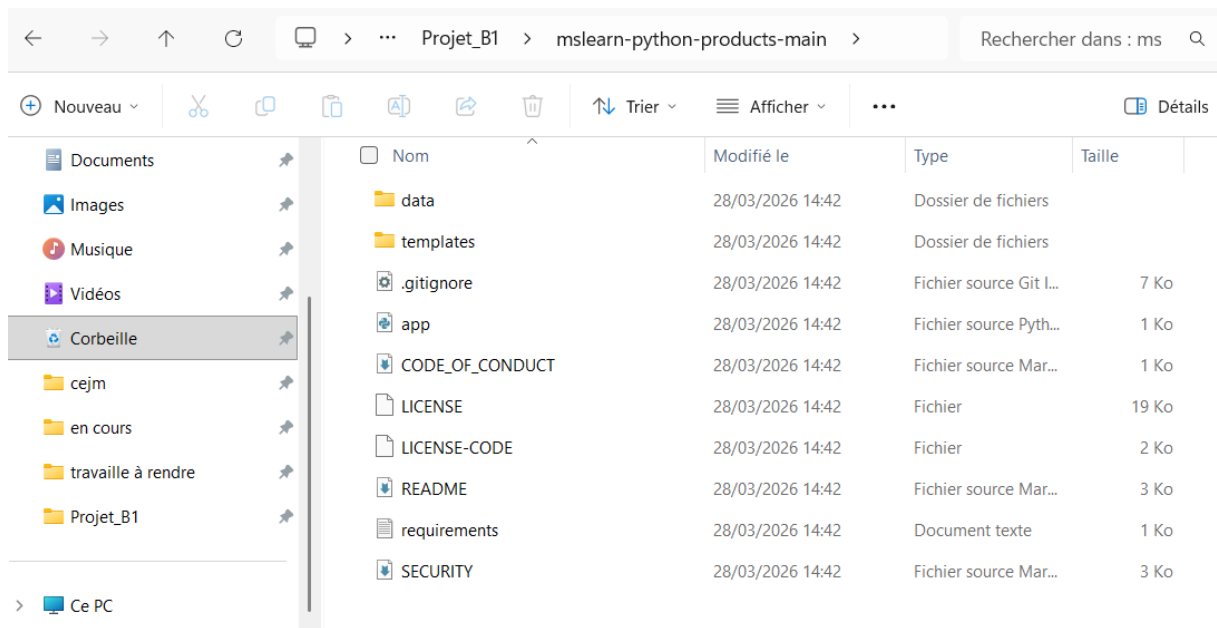




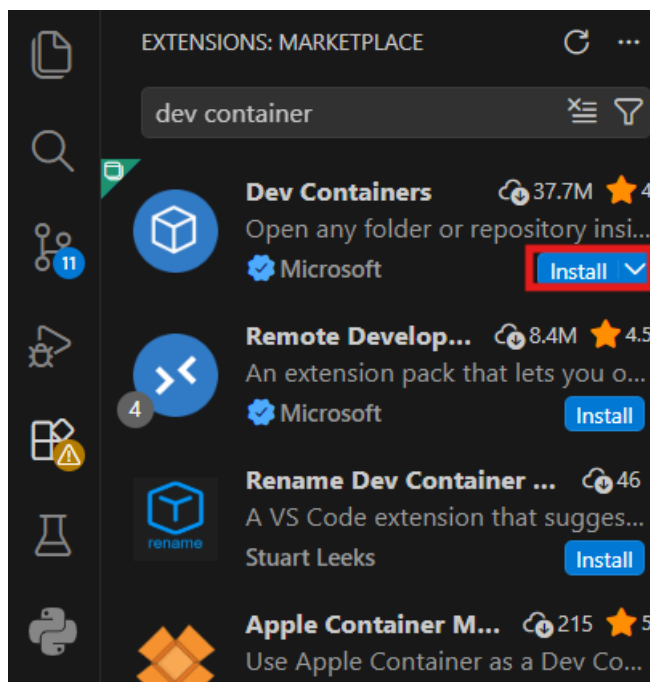
Le projet est maintenant ouvert dans VS Code.



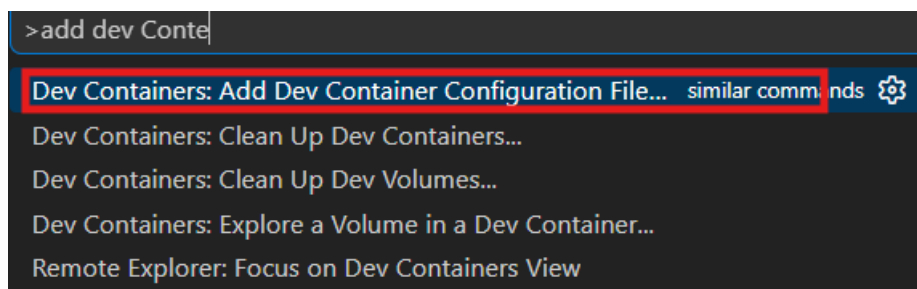
Et il apparait bien dans le fichier spécifié.

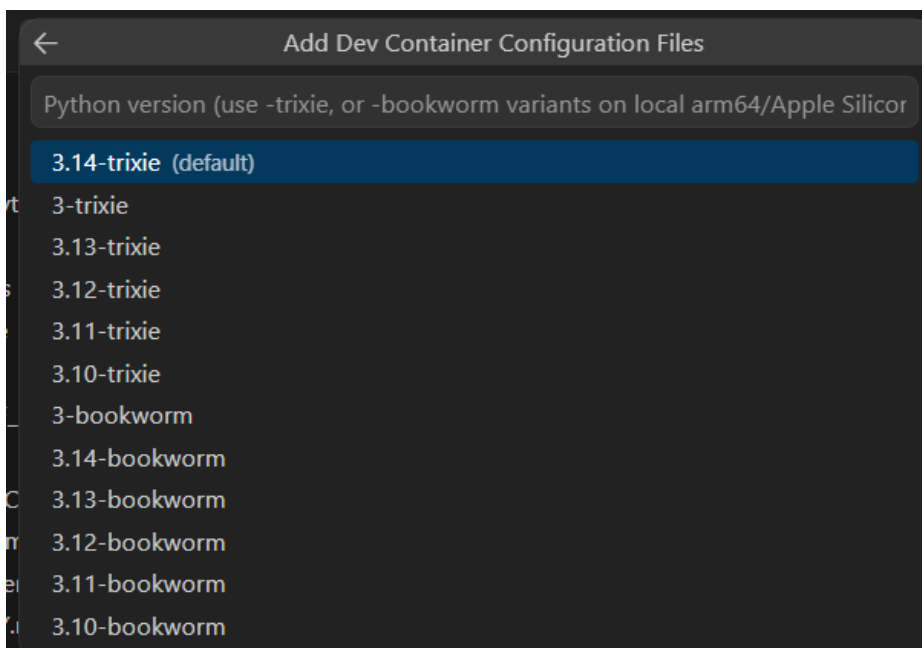
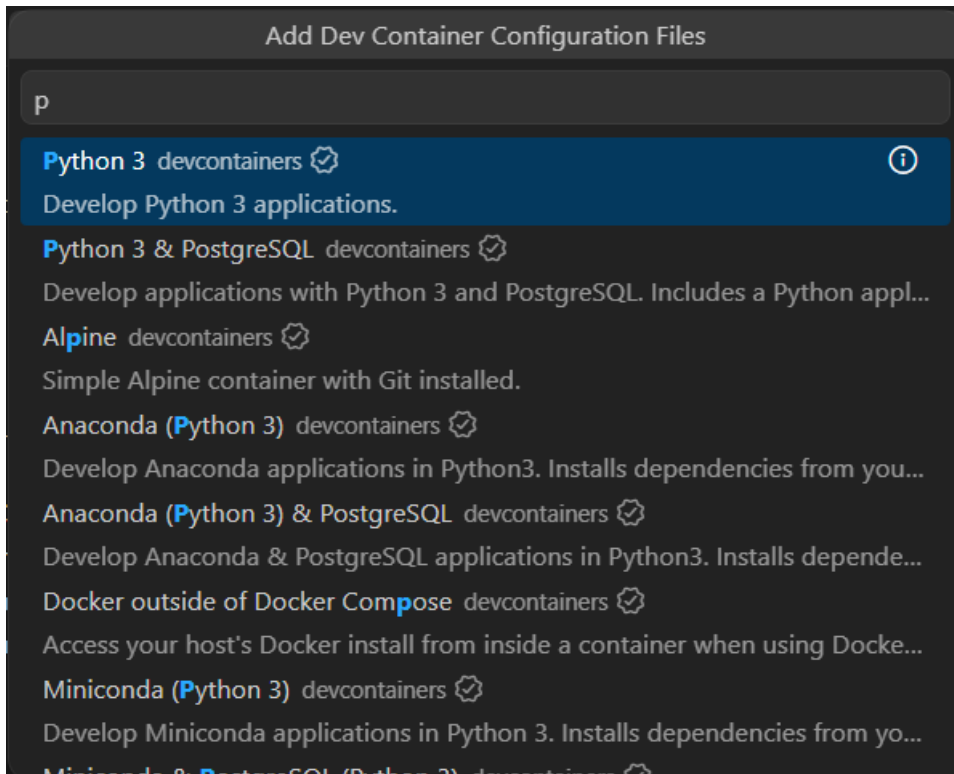
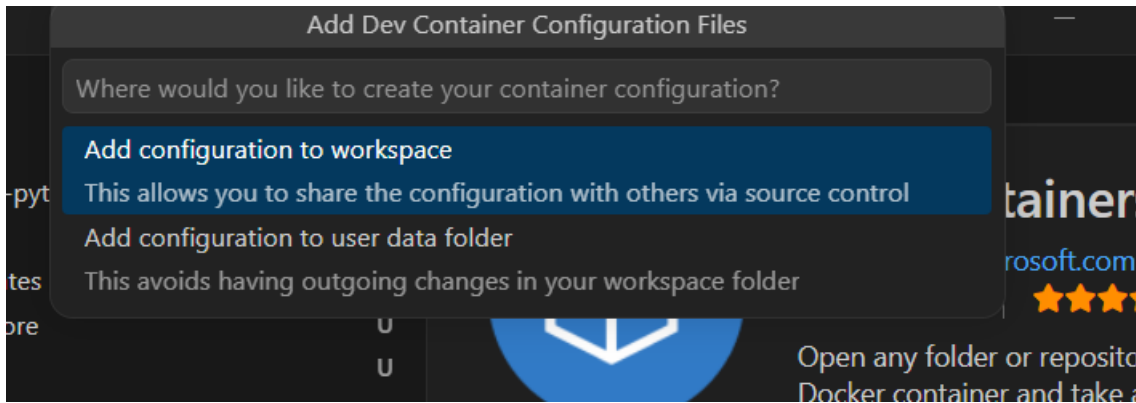


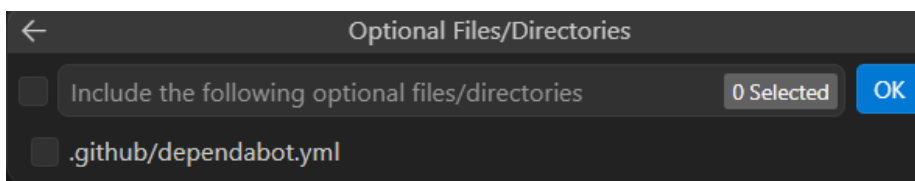
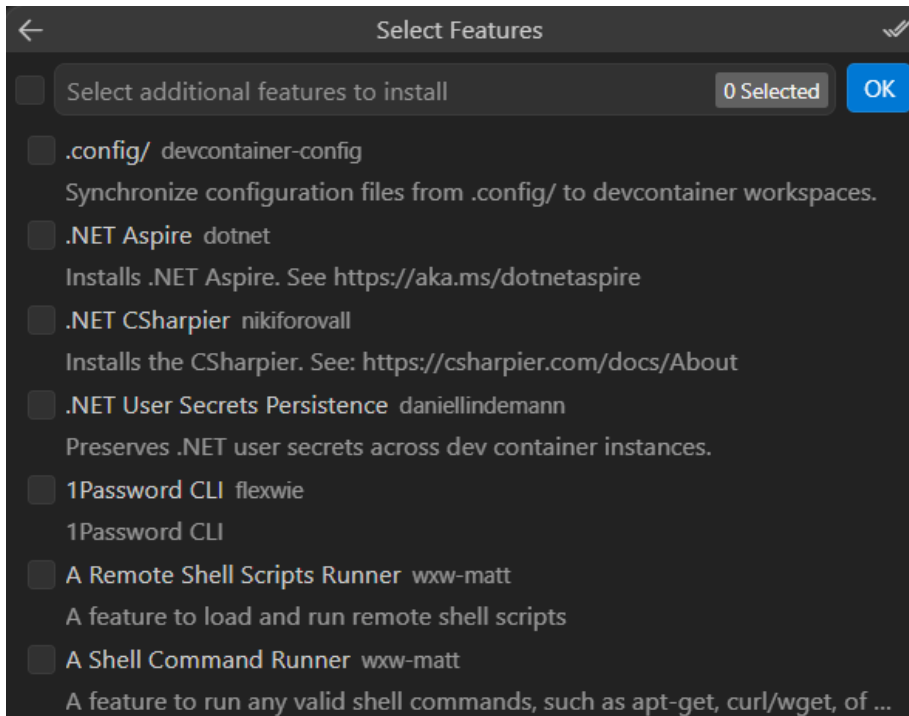
Nous installons l'extension Dev Containers.



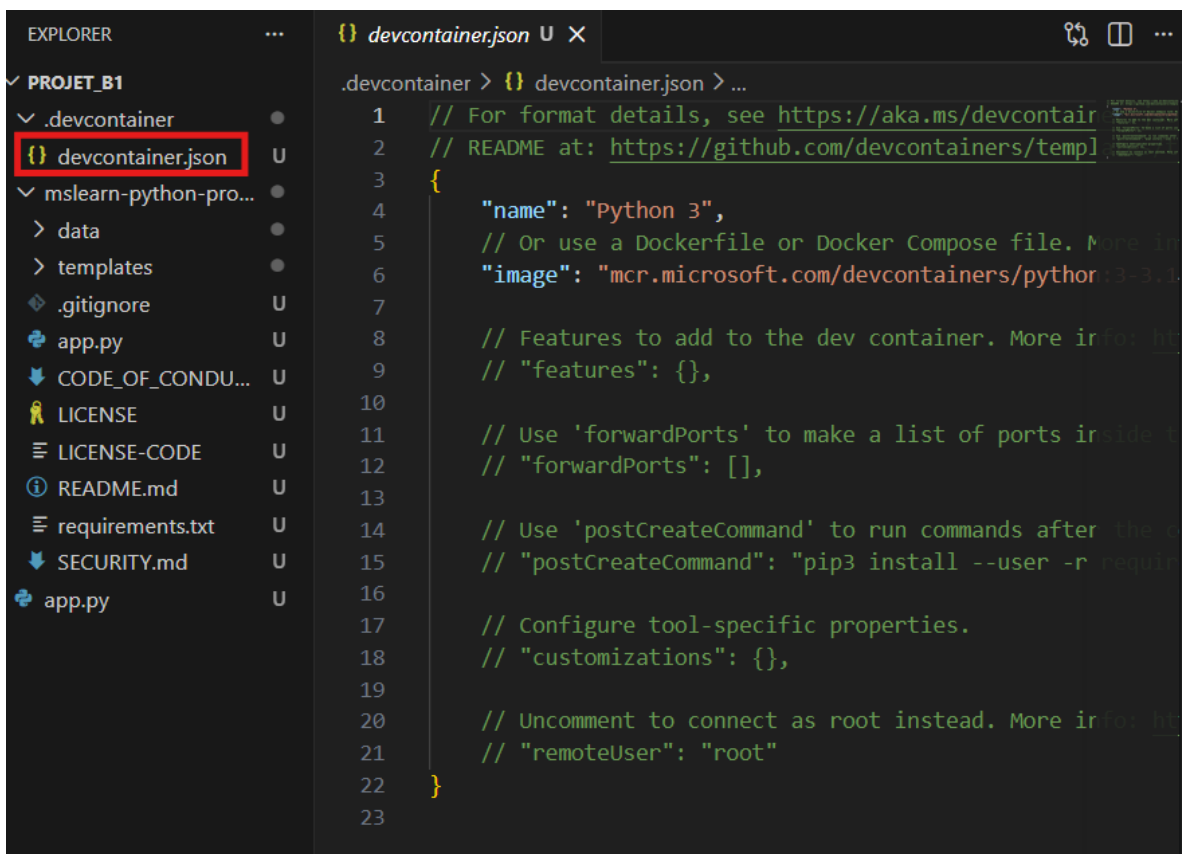
Nous sélectionnons dev container puis add configuration to workplace.



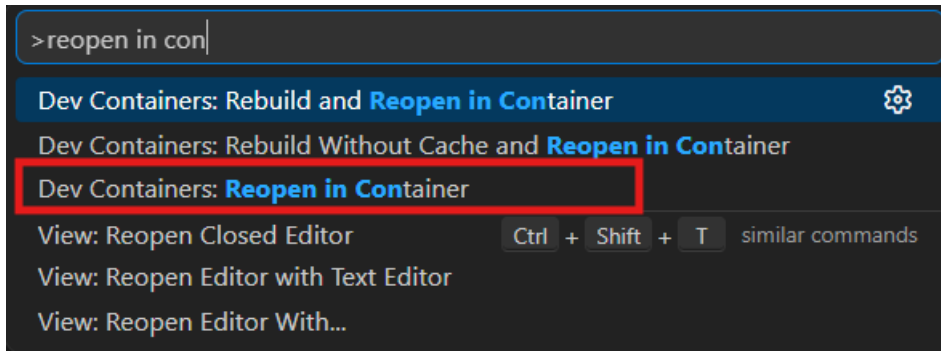




Un nouveau fichier devcontainer a été ajouté.



Nous ajoutons Dev Containers: Reopen in Container.



Nous installons les dépendances.

```

vscode →/workspaces/mslearn-python-products (main) $ python --version
Python 3.14.3
vscode →/workspaces/mslearn-python-products (main) $ 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-3.1.7-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))
  Downloading click-8.3.1-py3-none-any.whl.metadata (2.6 kB)
Collecting blinker>=1.6.2 (from Flask==2.3.3->-r requirements.txt (line 2))
  Downloading blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->Flask==2.3.3->-r requirements.txt (line 2))
  Downloading markupsafe-3.0.3-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl.metadata (2.7 kB)
Downloading flask-2.3.3-py3-none-any.whl (96 kB)
Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
Downloading click-8.3.1-py3-none-any.whl (108 kB)
Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)
Downloading markupsafe-3.0.3-cp314-cp314-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl (23 kB)
Downloading werkzeug-3.1.7-py3-none-any.whl (226 kB)
Installing collected packages: MarkupSafe, itsdangerous, click, blinker, Werkzeug, Jinja2, Flask
Successfully installed Flask-2.3.3 Jinja2-3.1.6 MarkupSafe-3.0.3 Werkzeug-3.1.7 blinker-1.9.0 click-8.3.1 itsdangerous-2.2.0

[notice] A new release of pip is available: 25.3 -> 26.0.1
[notice] To update, run: pip install --upgrade pip
vscode →/workspaces/mslearn-python-products (main) $

```

Nous exécutons le projet Python.

```

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 de
ployment. 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: 506-733-206
127.0.0.1 - - [28/Mar/2026 16:12:41] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/Mar/2026 16:12:41] "GET /favicon.ico HTTP/1.1" 404 -

```

Nous allons sur le site depuis le navigateur.

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000'. The page title is 'Produits'. Below the title is a table with the following columns: 'Nom', 'Marque', 'Prix', and 'Unités en stock'. The table contains 15 rows of product data.

Nom	Marque	Prix	Unités en stock
nain de jardin rouge solitaire	Outils pour la maison et les professionnels	56	98
Deux nains de jardin rouges	Outils pour la maison et les professionnels	92	4
Un gnome assis	Outils pour la maison et les professionnels	34	34
L'un était assis sur un gnome à chaussures	Outils pour la maison et les professionnels	54	54
Un gnome de terre	Outils pour la maison et les professionnels	29	23
Un gnome à lunettes	Outils pour la maison et les professionnels	54	94
Un gnome souriant	Outils pour la maison et les professionnels	43	45
Deux gnomes chanteurs	Outils pour la maison et les professionnels	65	78
Deux gnomes endormis	Outils pour la maison et les professionnels	32	12
gnomes en paquet de sept	Outils pour la maison et les professionnels	2543	45
Un gnome effrayé	Outils pour la maison et les professionnels	39	88
Un gnome de bienvenue	Outils pour la maison et les professionnels	28	65
Deux gnomes souriants	Outils pour la maison et les professionnels	76	87
Four à micro-ondes 33,9 pi² 92 325 W	Perceuse électrique	1740	58
Réfrigérateurur 1,7 pi³ 110 watts	Scies professionnelles	200	54

The screenshot shows a code editor with a file named 'products.json' open. The file contains a JSON array of product objects. The Explorer panel on the left shows the file structure, including a 'data' folder containing 'products.json'. The code in the editor is as follows:

```

1  [
2
3
4    {
5      "name": "Single red garden gnome",
6      "price": 56,
7      "brand": {
8        "name": "Home & Pro tools"
9      },
10     "stockUnits": 98
11   },
12   {
13     "name": "Two red garden gnomes",
14     "price": 92,
15     "brand": {
16       "name": "Home & Pro tools"
17     },
18     "stockUnits": 4
19   },
20   {
21     "name": "One sat gnome",
22     "price": 34,
23     "brand": {
24       "name": "Home & Pro tools"
25     },
26     "stockUnits": 34
27   },
28   {
29     "name": "One sat on shoe gnome",
30     "price": 54,
31     "brand": {
32       "name": "Home & Pro tools"
33     },
34     "stockUnits": 54
35   }
36 ]

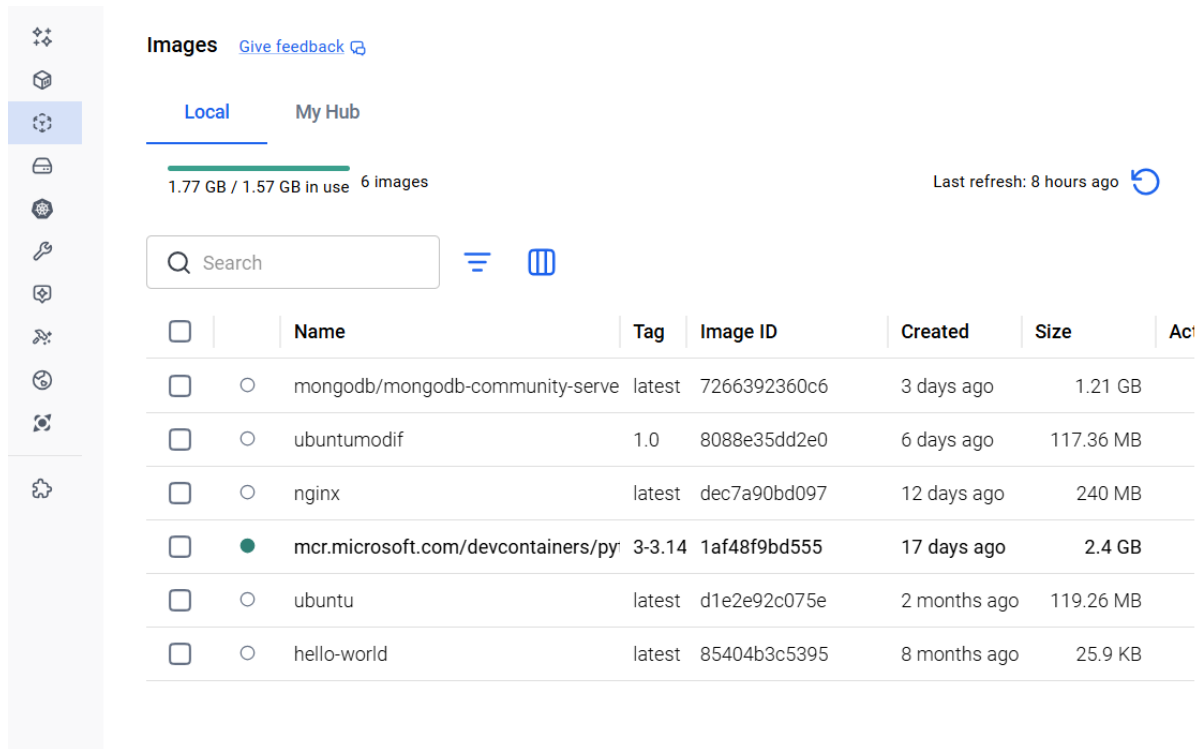
```

```
index.html X
templates > index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, height=device-height">
6   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css">
7   <title>Products Viewer</title>
8 </head>
9 <body>
10  <div id="app">
11    <header>
12      <div class="container">
13        <div class="hero is-info is-bold">
14          <div class="hero-body">
15            <h1 class="is-size-1">Products</h1>
16          </div>
17        </div>
18      </div>
19    </header>
20    <section class="section">
21      <div class="container">
22        <div>
23          <table class="table is-fullwidth is-striped">
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>
```

Nous constatons que le conteneur a bien été créé et apparaît dans Docker.

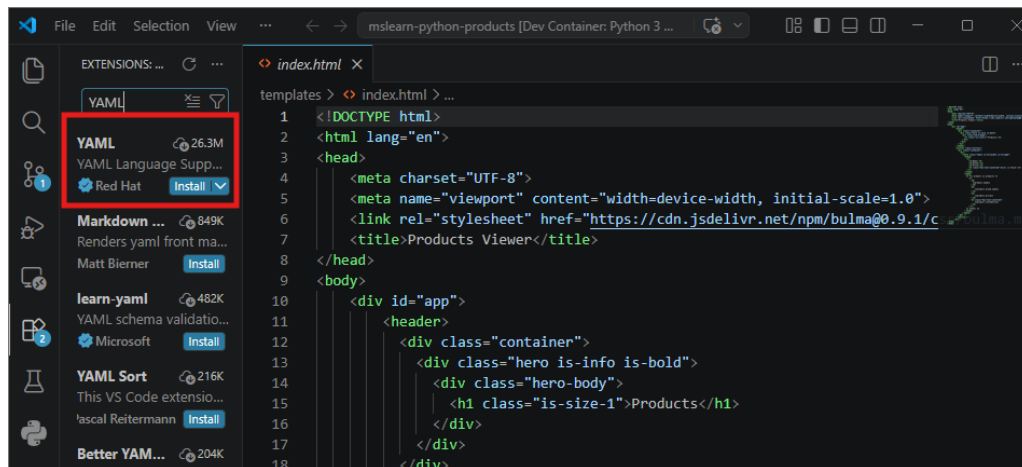
The screenshot shows the Docker Desktop interface. At the top, it displays 'Containers' with a 'Give feedback' link. Below this, there are two summary cards: 'Container CPU usage' showing 5.22% / 800% (8 CPUs available) and 'Container memory usage' showing 1.52GB / 7.46GB. A 'Show charts' link is also present. A search bar and a toggle for 'Only running' are visible. The main area contains a table with the following data:

Name	Container ID	Image	Port(s)	Actions
romantic_almeida	a1d57f6aa487	devcontain		Stop, Refresh, Restart, Delete

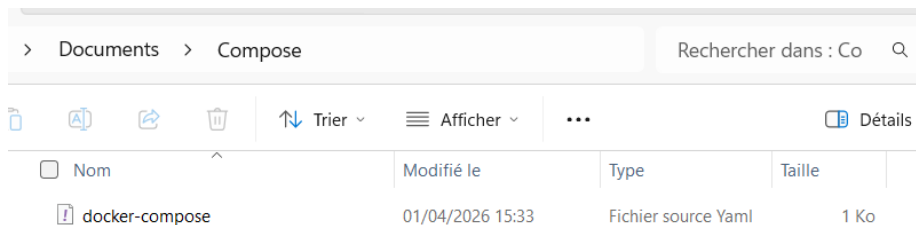


13. Monter une stack applicative avec Docker Compose.

Nous installons l'extension YAML depuis VSCode.



Nous créons un blog WordPress avec sa base de données.



```

docker-compose.yml X
C: > Users > noli8 > OneDrive > Documents > Compose > docker-compose.yml > {} volumes
  Run All Services | docker-compose.yml - The Compose specification establishes a standard for the definition
1  services:
   >Run Service
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
   >Run Service
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

```

Nous lançons le stack applicatif avec un docker-compose up.

```

PS C:\Users\noli8\OneDrive\Documents\Compose> docker compose up
[+] up 43/43
✓ Image wordpress:latest          Pulled          33.7s
✓ Image mysql:latest              Pulled          36.0s
✓ Network compose_default         Created         0.1s
✓ Volume compose_db_data         Created         0.0s
✓ Container compose-db-1         Created         0.8s
✓ Container compose-wordpress-1  Created         0.2s

```

```

Attaching to db-1, wordpress-1
db-1 | 2026-04-01 13:44:10+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 9.6.0-1.el9
started.
db-1 | 2026-04-01 13:44:10+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
db-1 | 2026-04-01 13:44:10+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 | 2026-04-01 13:44:11+00:00 [Note] [Entrypoint]: Initializing database files
db-1 | 2026-04-01T13:44:11.201637Z 0 [System] [MY-015017] [Server] MySQL Server Initializatio
n - start.
db-1 | 2026-04-01T13:44:11.203424Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 9
.6.0) initializing of server in progress as process 81
db-1 | 2026-04-01T13:44:11.228973Z 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; cop
ying 'wp-config-docker.php' (WORDPRESS_DB_HOST WORDPRESS_DB_USER)
wordpress-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain nam
e, using 172.18.0.3. Set the 'ServerName' directive globally to suppress this message
wordpress-1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain nam
e, using 172.18.0.3. Set the 'ServerName' directive globally to suppress this message
db-1 | 2026-04-01T13:44:11.678784Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has
ended.
wordpress-1 | [Wed Apr 01 13:44:11.683336 2026] [mpm_prefork:notice] [pid 1:tid 1] AH00163: Apache/2
.4.66 (Debian) PHP/8.3.30 configured -- resuming normal operations
wordpress-1 | [Wed Apr 01 13:44:11.683407 2026] [core:notice] [pid 1:tid 1] AH00094: Command line: '
apache2 -D FOREGROUND'
db-1 | 2026-04-01T13:44:13.128404Z 6 [Warning] [MY-010453] [Server] root@localhost is created
with an empty password ! Please consider switching off the --initialize-insecure option.

```

On nous informe qu'on s'est bien connecté au site web.

```

db-1 | 2026-04-01T13:44:21.363789Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready fo
r connections. Version: '9.6.0' socket: '/var/run/mysqld/mysqld.sock' port: 3306 MySQL Community S
erver - GPL.
wordpress-1 | 172.18.0.1 - - [01/Apr/2026:13:45:02 +0000] "GET / HTTP/1.1" 500 2757 "-" "Mozilla/5.0
 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/146.0.0.0 Safari/537.36"
wordpress-1 | 172.18.0.1 - - [01/Apr/2026:13:45:02 +0000] "GET /favicon.ico HTTP/1.1" 500 2757 "http
://localhost:8000/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/146.0.0.0 Safari/537.36"
wordpress-1 | 172.18.0.1 - - [01/Apr/2026:13:45:26 +0000] "GET / HTTP/1.1" 500 2757 "-" "Mozilla/5.0
 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/146.0.0.0 Safari/537.36"
wordpress-1 | 172.18.0.1 - - [01/Apr/2026:13:45:26 +0000] "GET /favicon.ico HTTP/1.1" 500 2757 "http
://localhost:8000/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/146.0.0.0 Safari/537.36"
wordpress-1 | 172.18.0.1 - - [01/Apr/2026:13:45:49 +0000] "GET / HTTP/1.1" 500 2757 "-" "Mozilla/5.0
 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/146.0.0.0 Safari/537.36"

```

Nous constatons que les containers et que l'image ont bien été créés.

Images [Give feedback](#)

Local My Hub

3.55 GB / 4.35 GB in use 8 images Last refresh: 25 minutes ago

Search

Name	Tag	Image ID	Created	Size	Act
mongodb/mongodb-community-serve	latest	7266392360c6	7 days ago	1.21 GB	
ubuntumodif	1.0	8088e35dd2e0	10 days ago	117.36 MB	
nginx	latest	dec7a90bd097	16 days ago	240 MB	
mcr.microsoft.com/devcontainers/py	3-3.14	1af48f9bd555	21 days ago	2.4 GB	
ubuntu	latest	d1e2e92c075e	2 months ago	119.26 MB	
hello-world	latest	85404b3c5395	8 months ago	25.9 KB	
wordpress	latest	ee74dc0ebbe7	16 days ago	1.07 GB	
mysql	latest	24e450bbd24f	13 days ago	1.26 GB	

Containers [Give feedback](#)

Container CPU usage 0.97% / 800% (8 CPUs available) Container memory usage 480.77MB / 7.46GB [Show charts](#)

Search Only running

Name	Container ID	Image	Port(s)	Actions
romantic_almeida	a1d57f6aa487	devcontain		
compose	-	-	-	

compose
C:\Users\noli8\OneDrive\Documents\Compose

db	mysql:latest			
wordpress	wordpress:latest	8000:80		