

Installation de Docker sur Ubuntu



Sommaire

Installation de Docker sur Ubuntu.....	1
Contexte.....	3
1) Installation de docker Desktop.....	4
2) Installation de Visual Studio Code.....	11
3) Test.....	13

Contexte

C'est quoi Docker et à quoi sert-il ?

Docker est tout simplement une plateforme qui permet de créer, déployer et exécuter des applications dans des conteneurs. Un conteneur est une unité logicielle qui regroupe tout ce dont une application a besoin pour fonctionner, y compris le code, les bibliothèques et les dépendances de manière isolée du système d'exploitation.

1) Installation de docker Desktop

Dans un premier temps, il faut mettre à jour le système avec les deux commandes suivantes :

- **sudo apt update**
- **sudo apt upgrade**

```
nolann@Ubuntu-Client-BTS2:~$ sudo apt update
[sudo] password for nolann:
Hit:1 http://fr.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:3 http://fr.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://fr.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
526 packages can be upgraded. Run 'apt list --upgradable' to see them.
nolann@Ubuntu-Client-BTS2:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  firefox libllvm15 ubuntu-pro-client ubuntu-pro-client-l10n
The following packages have been kept back:
  linux-generic-hwe-22.04 linux-headers-generic-hwe-22.04
  linux-image-generic-hwe-22.04
The following packages will be upgraded:
  accountsservice alsa-ucm-conf amd64-microcode apparmor apport apport-gtk apt
  apt-utils avahi-autoipd avahi-daemon avahi-utils base-files bash
  bind9-dnsutils bind9-host bind9-libs bluez bluez-cups bluez-obexd
  bsdxtrautils bsdutils bubblewrap busybox-initramfs busybox-static
  ca-certificates coreutils cpio cups cups-browsed cups-bsd cups-client
  cups-common cups-core-drivers cups-daemon cups-filters
  cups-filters-core-drivers cups-ipp-utils cups-ppdc cups-server-common dbus
  dbus-user-session distro-info distro-info-data dmidecode dmssetup
  dns-root-data dnsmasq-base dpkg espeak-ng-data evince evince-common
  evolution-data-server evolution-data-server-common fdisk file
  firmware-sof-signed fonts-noto-color-emoji fonts-opensymbol fprintd fwupd
  gdb gdm3 ghostscript ghostscript-x gir1.2-accountsservice-1.0 gir1.2-adw-1
  gir1.2-gdkpixbuf-2.0 gir1.2-gdm-1.0 gir1.2-gnomedesktop-3.0
  gir1.2-gst-plugins-base-1.0 gir1.2-gstreamer-1.0 gir1.2-gtk-3.0
  gir1.2-gtk-4.0 gir1.2-harfbuzz-0.0 gir1.2-javascriptcoregtk-4.0
```

Une fois les mises à jour effectuées, on va passer à l'installation de docker desktop. En premier, il faut installer l'environnement "gnome" terminal avec la commande "**sudo apt install gnome-terminal**".

```
nolann@Ubuntu-Client-BTS2:~$ sudo apt install gnome-terminal
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnome-terminal is already the newest version (3.44.0-1ubuntu1).
gnome-terminal set to manually installed.
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
nolann@Ubuntu-Client-BTS2:~$
```

Ensuite, il faut configurer le dépôt de paquets de Docker. Sur leur site, on pourra directement trouver la documentation pour [installer Docker sur Ubuntu](#).
Avec les commandes suivantes :

Add Docker's official GPG key:

```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

Add the repository to Apt sources:

```
echo \
"deb [arch=$(dpkg --print-architecture)
signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo
"${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

On pourra par la suite configurer le dépôt de paquets de Docker.

```
no1ann@Ubuntu-Client-BTS2:~$ # Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://fr.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://fr.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://fr.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203-22.04.1).
ca-certificates set to manually installed.
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 194 kB of archives.
After this operation, 455 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://fr.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64 7.81.0-1ubuntu1.20 [194 kB]
```

Maintenant, il faut installer le paquet de Docker.

Pour cela, il faut se rendre sur la documentation [Docker](https://docs.docker.com) et cliquer sur “**DEB package**”

The screenshot shows the Docker Desktop installation page for Ubuntu. The page includes a sidebar with navigation options like 'Manuals', 'Docker Engine', and 'Docker Desktop'. The main content area has a 'Prerequisites' section with a terminal command: `sudo apt install gnome-terminal`. Below this is the 'Install Docker Desktop' section, which lists three steps. Step 2, 'Download the latest DEB package', is highlighted with a red box and a red circle containing the number '1'. A download progress window is open in the top right corner, showing two files: 'docker-desktop-amd64(1).deb' (13 s remaining) and 'docker-desktop-amd64.deb' (Terminé). A 'Give feedback' button is located at the bottom right of the page.

Maintenant que le paquet est téléchargé, on va se rendre dans le dossier du paquet avec la commande “**cd Downloads**” et on va exécuter la commande “**ls**” pour vérifier que le fichier est bien là.

```
nolann@Ubuntu-Client-BTS2:~$ cd Downloads/  
nolann@Ubuntu-Client-BTS2:~/Downloads$ ls  
'docker-desktop-amd64(1).deb'  docker-desktop-amd64.deb  
nolann@Ubuntu-Client-BTS2:~/Downloads$
```

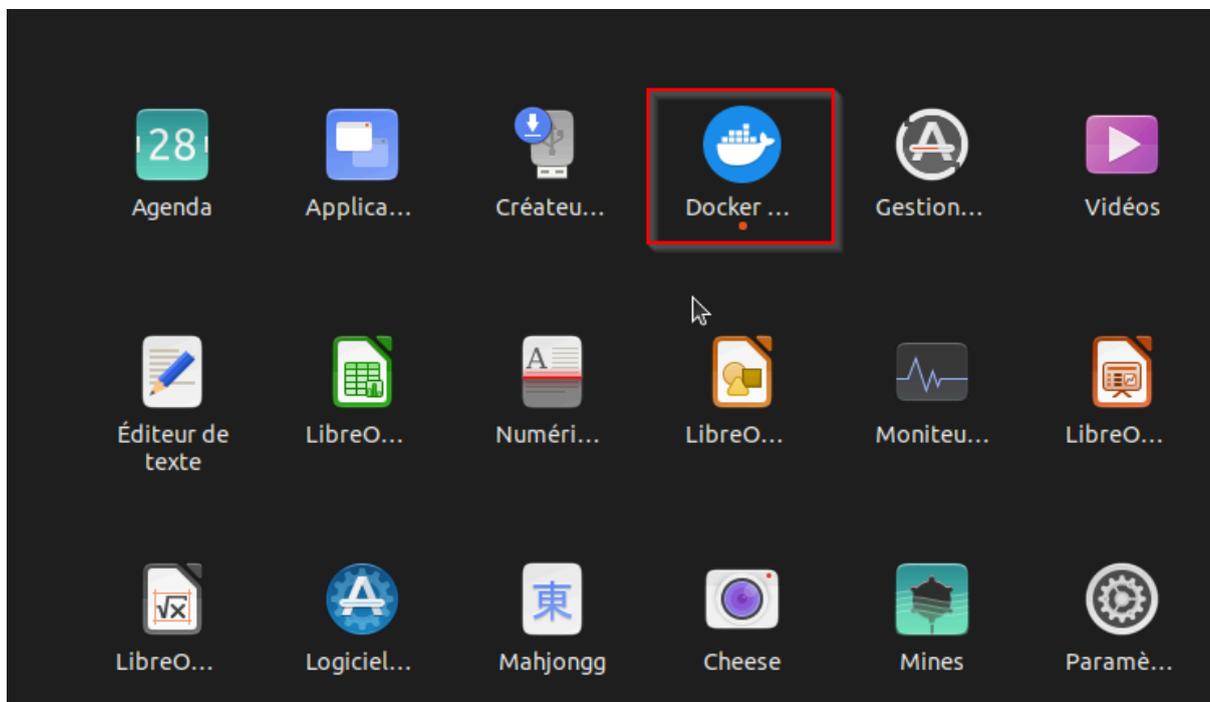
Ensuite, on va utiliser la commande “**sudo apt-get update**” pour mettre à jour tous les paquets.

```
nolann@Ubuntu-Client-BTS2:~/Downloads$ sudo apt-get update  
Hit:1 http://fr.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://fr.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:3 http://fr.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Hit:5 https://download.docker.com/linux/ubuntu jammy InRelease  
Reading package lists... Done  
nolann@Ubuntu-Client-BTS2:~/Downloads$
```

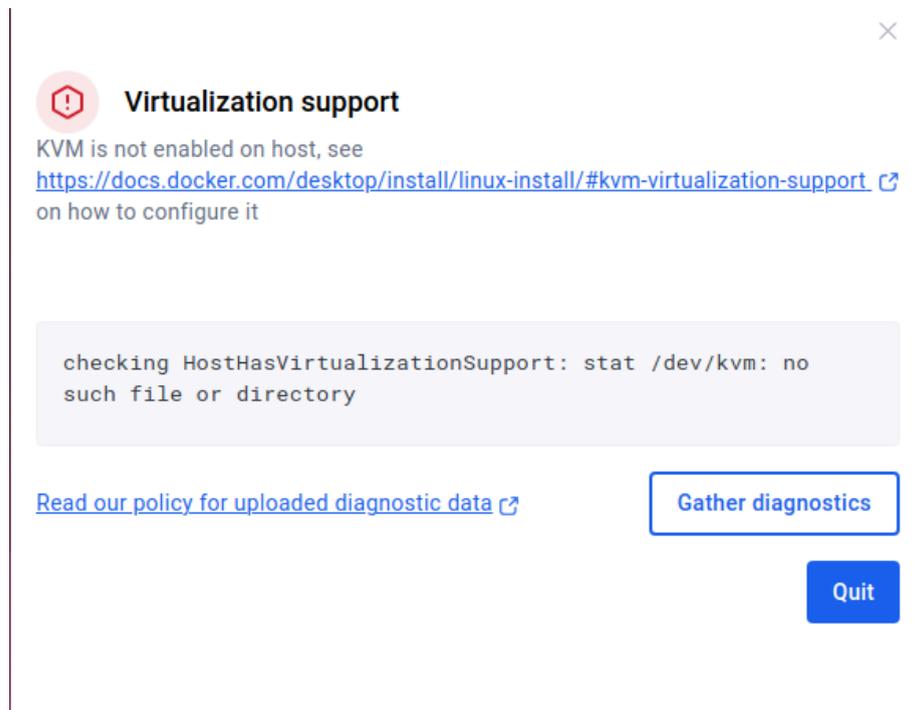
Pour finir, on va utiliser la commande “**sudo apt-get install ./docker-desktop-amd64.deb**”

```
no1ann@Ubuntu-Client-BTS2:~/Downloads$ sudo apt-get install ./docker-desktop-amd64.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'docker-desktop' instead of './docker-desktop-amd64.deb'
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  cpu-checker docker-buildx-plugin docker-ce-cli docker-compose-plugin git git-man ibverbs-provide
  ipxe-qemu-256k-compatible-efi-roms libaio1 libcacard0 libdaxctl1 libdecor-0-0 libdecor-0-plugin-1-ca
  libgfbapi0 libgfrpc0 libgfxdr0 libglusterfs0 libibverbs1 libiscsi7 libndctl6 libpmem1 libpmemobj1
  librdmacm1 libsd12-2.0-0 libslirp0 libspice-server1 liburing2 libusbredirparser1 libvirglirendere
  qemu-block-extra qemu-system-common qemu-system-data qemu-system-gui qemu-system-x86 qemu-utils
  xclip
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki
  gstreamer1.0-plugins-ugly libxml-simple-perl python ruby samba vde2 debootstrap
The following NEW packages will be installed:
  cpu-checker docker-buildx-plugin docker-ce-cli docker-compose-plugin docker-desktop git git-man
  ipxe-qemu-256k-compatible-efi-roms libaio1 libcacard0 libdaxctl1 libdecor-0-0 libdecor-0-plugin-1-ca
  libgfbapi0 libgfrpc0 libgfxdr0 libglusterfs0 libibverbs1 libiscsi7 libndctl6 libpmem1 libpmemobj1
  librdmacm1 libsd12-2.0-0 libslirp0 libspice-server1 liburing2 libusbredirparser1 libvirglirendere
  qemu-block-extra qemu-system-common qemu-system-data qemu-system-gui qemu-system-x86 qemu-utils
  xclip
0 upgraded, 50 newly installed, 0 to remove and 3 not upgraded.
Need to get 104 MB/496 MB of archives.
After this operation, 361 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://fr.archive.ubuntu.com/ubuntu jammy/main amd64 libaio1 amd64 0.3.112-13build1 [7 176 B]
Get:2 http://fr.archive.ubuntu.com/ubuntu jammy/main amd64 libfdt1 amd64 1.6.1-1 [20,0 kB]
Get:3 http://fr.archive.ubuntu.com/ubuntu jammy/main amd64 libibverbs1 amd64 39.0-1 [69,3 kB]
Get:4 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-cli amd64 5:28.0.1-1-u
Get:5 http://fr.archive.ubuntu.com/ubuntu jammy/main amd64 libdaxctl1 amd64 72.1-1 [19,8 kB]
Get:6 http://fr.archive.ubuntu.com/ubuntu jammy/main amd64 libndctl6 amd64 72.1-1 [57,7 kB]
Get:7 http://fr.archive.ubuntu.com/ubuntu jammy/main amd64 libpmem1 amd64 1.11.1-3build1 [81,4 kB]
Get:8 http://fr.archive.ubuntu.com/ubuntu jammy/main amd64 librdmacm1 amd64 39.0-1 [71,2 kB]
Get:9 http://fr.archive.ubuntu.com/ubuntu jammy/main amd64 libslirp0 amd64 4.6.1-1build1 [61,5 kB]
Get:10 http://fr.archive.ubuntu.com/ubuntu jammy/main amd64 liburing2 amd64 2.1-2build1 [10,3 kB]
```

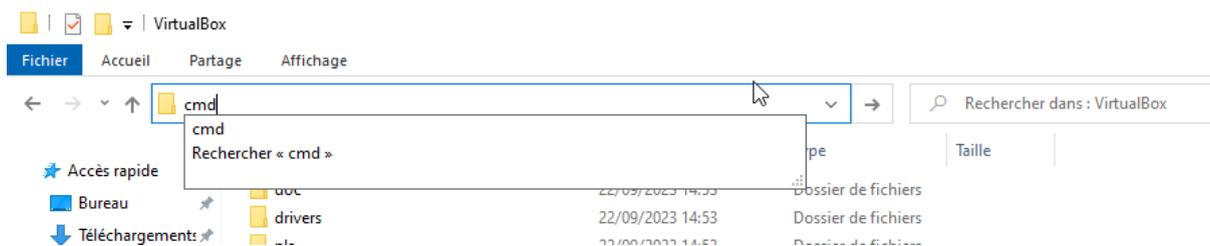
Maintenant, il faut attendre que le téléchargement soit terminé.
Ensuite, on peut démarrer Docker depuis le bureau.



On peut voir que j'ai un problème de virtualisation.



Pour corriger cette erreur, il faut ouvrir un explorateur de fichiers (sur votre poste), aller dans **“Ce PC - Disque Local - Programmes - Oracle - VirtualBox”**. Dans la barre d’adresse, tapez **“cmd”** pour ouvrir l’invite de commande.

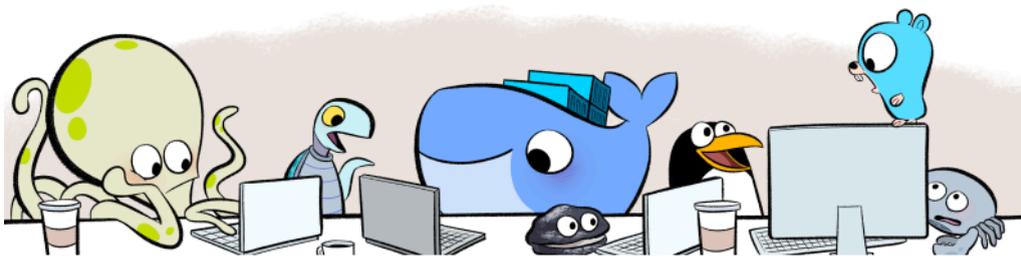


Dans l’invite de commande (cmd), tapez la commande **“VBoxManage modifyvm “Client_Ubuntu_BTS2” --nested-hw-virt on”**

Avant d’exécuter la commande, veuillez éteindre votre machine virtuelle.

```
C:\Program Files\Oracle\VirtualBox>VBoxManage modifyvm "Client_Ubuntu_BTS2" --nested-hw-virt on
C:\Program Files\Oracle\VirtualBox>
```

Une fois que la commande a été effectuée, démarrez votre machine virtuelle puis lancez Docker.



Docker Subscription Service Agreement

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

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](#)

Cliquez sur le bouton “**Accept**”.

Ensuite, ne vous connectez pas et ne créez pas de compte. À droite, il y a un petit bouton “**Skip**” .

Welcome to Docker

[Skip](#)

Work Personal

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

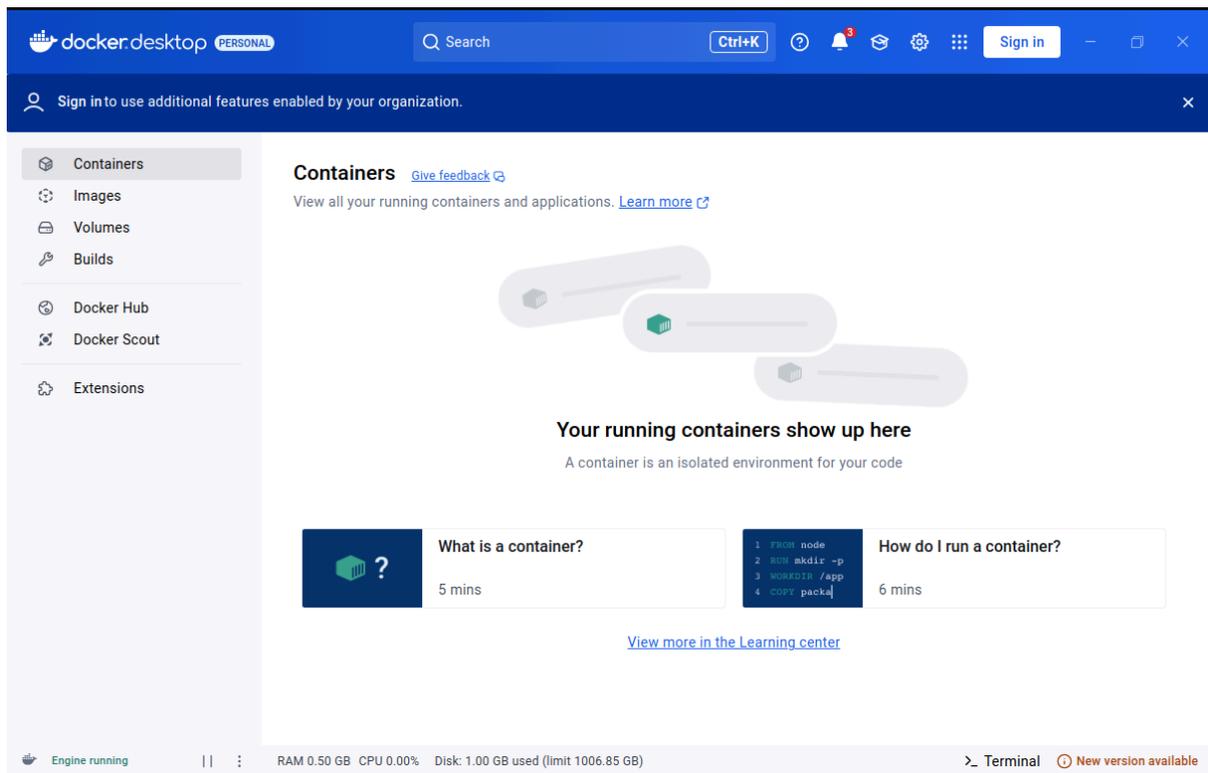
You must initialize pass before signing in to Docker Desktop.
[Learn more](#)

Continue

Or

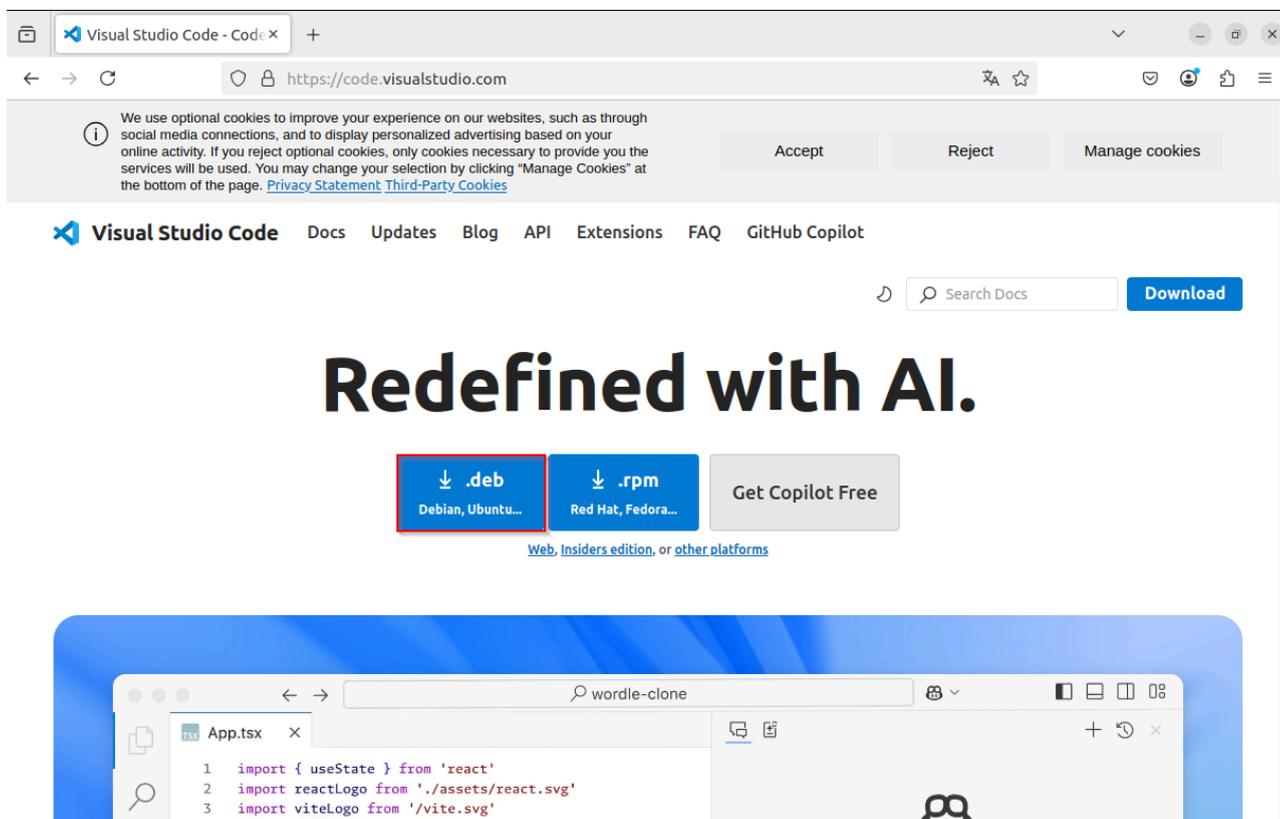
  [Create an account](#) 

Voici l'interface de Docker Desktop.



2) Installation de Visual Studio Code

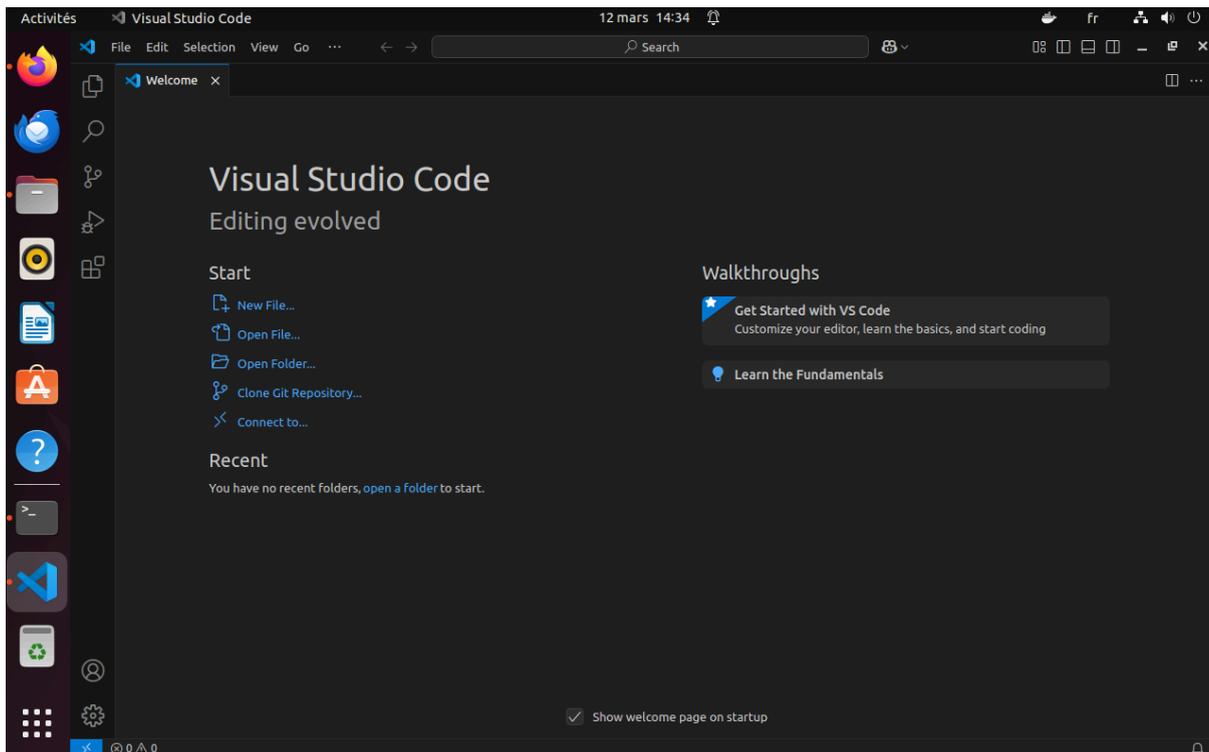
Pour pouvoir installer Visual Studio Code, allez sur le site de Visual Studio Code et téléchargez le fichier .deb .



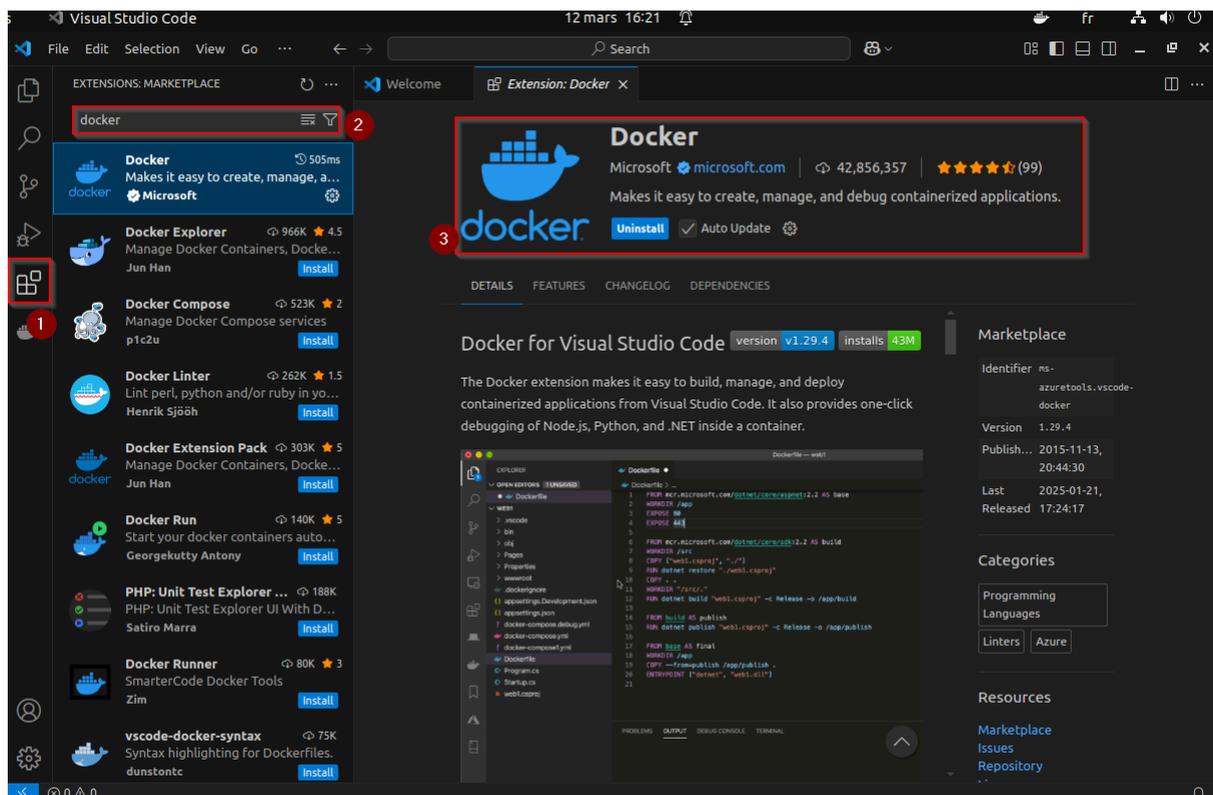
Ensuite, une fois le fichier téléchargé, ouvrez un terminal et tapez **“cd Downloads/”** et exécutez la commande **“sudo apt-get install ./code_1.98.1-1741624510_amd.deb”**

```
nolann@Ubuntu-Client-BTS2:~/Downloads$ sudo apt-get install ./code_1.98.1-1741624510_amd64.deb
[sudo] Mot de passe de nolann :
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Note : sélection de « code » au lieu de « ./code_1.98.1-1741624510_amd64.deb »
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  libflashrom1 libftdi1-2 libllvm13
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
Les NOUVEAUX paquets suivants seront installés :
  code
0 mis à jour, 1 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 0 o/104 Mo dans les archives.
Après cette opération, 422 Mo d'espace disque supplémentaires seront utilisés.
Réception de : :1 /home/nolann/Downloads/code_1.98.1-1741624510_amd64.deb code amd64 1.98.1-1741624510 [104 MB]
Préconfiguration des paquets...
Sélection du paquet code précédemment désélectionné.
(Lecture de la base de données... 211933 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../code_1.98.1-1741624510_amd64.deb ...
```

Une fois que l'installation est finie, vous pouvez lancer Visual Studio Code.



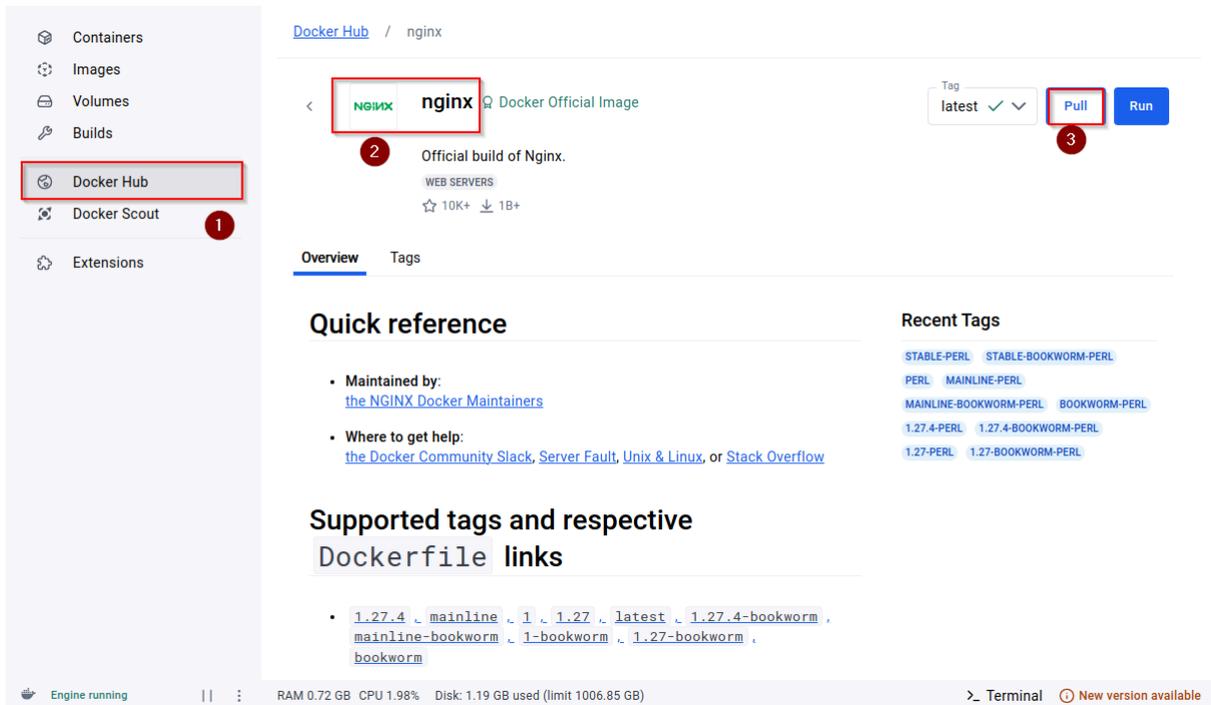
Maintenant que Visual Studio Code fonctionne, on va installer l'extension docker. Dans le menu de gauche, cliquez sur "Extensions", recherchez "Docker" puis installez.



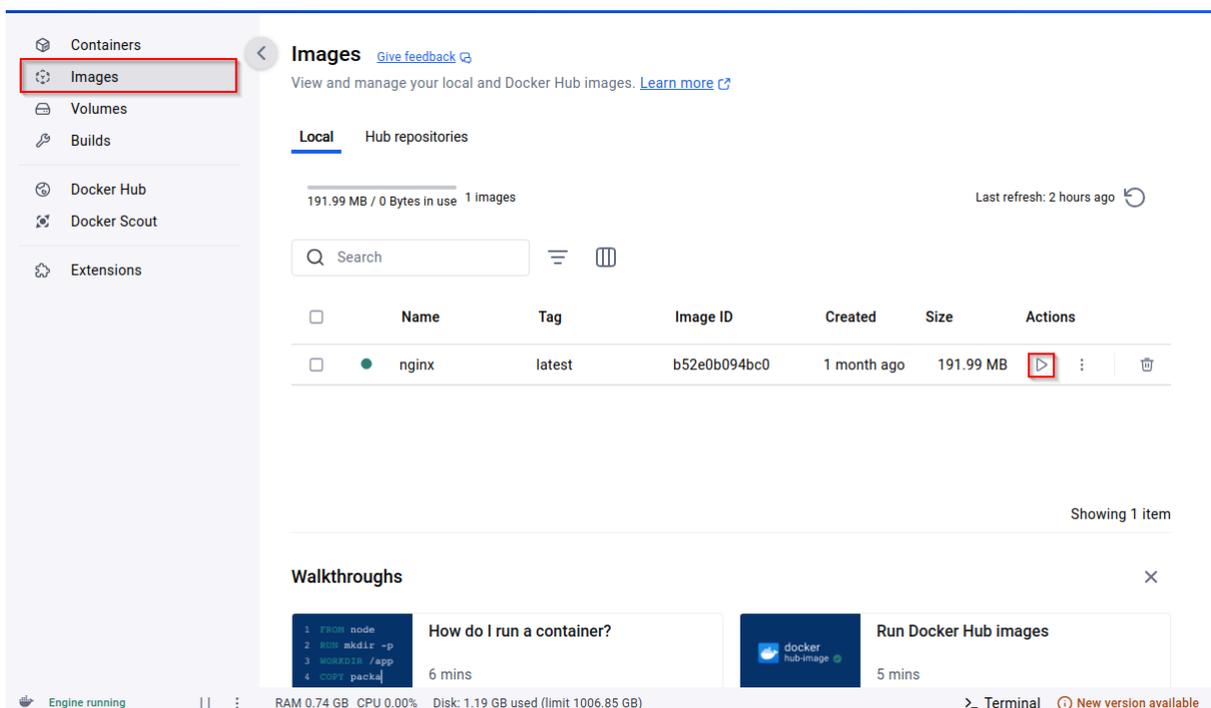
3) Test

Par la suite, il faudra installer “Nginx”. Nginx est un serveur web mais pour l’instant, il nous servira à voir si notre conteneur démarre.

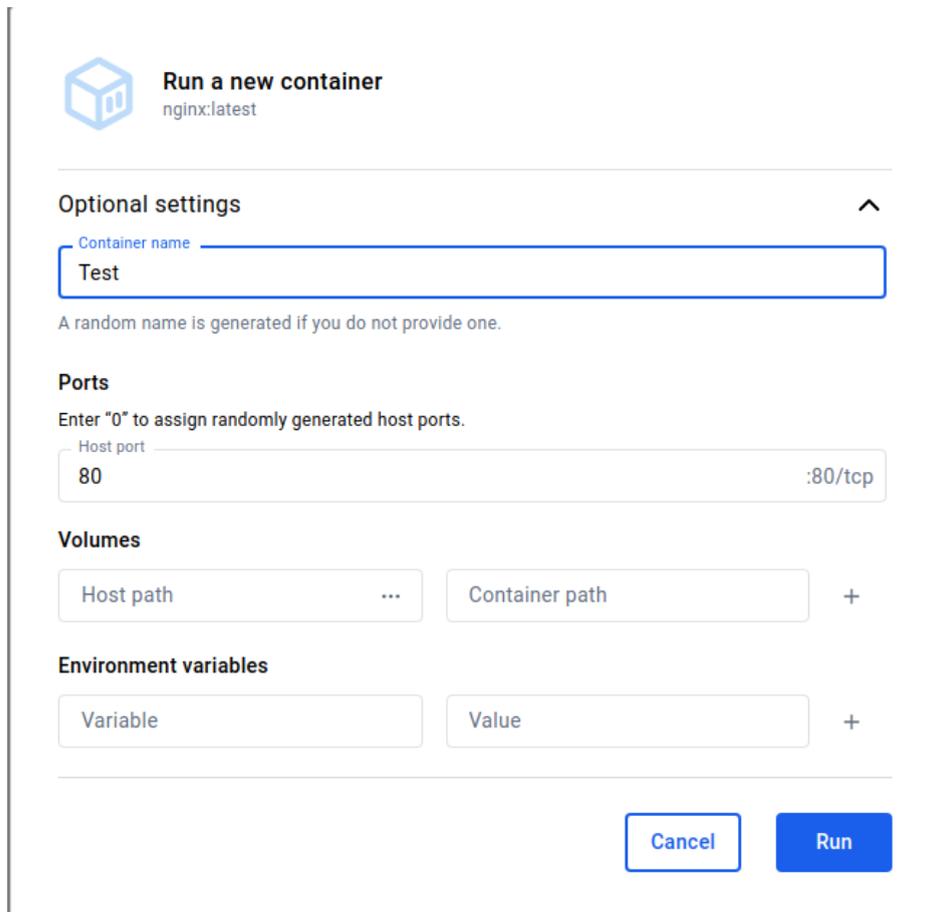
Pour l’installer, ouvrez Docker, allez dans “**Docker Hub**”, tapez “**Nginx**” et cliquez sur “**pull**”.



Quand l’installation s’est terminée, allez dans “**Images**” du menu à gauche, et cliquez sur l’icône en forme de triangle (Run).



Ensuite, il faudra configurer le conteneur. Comme c'est un test, je vais le nommer "Test" et définir le port "80".



Run a new container
nginx: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

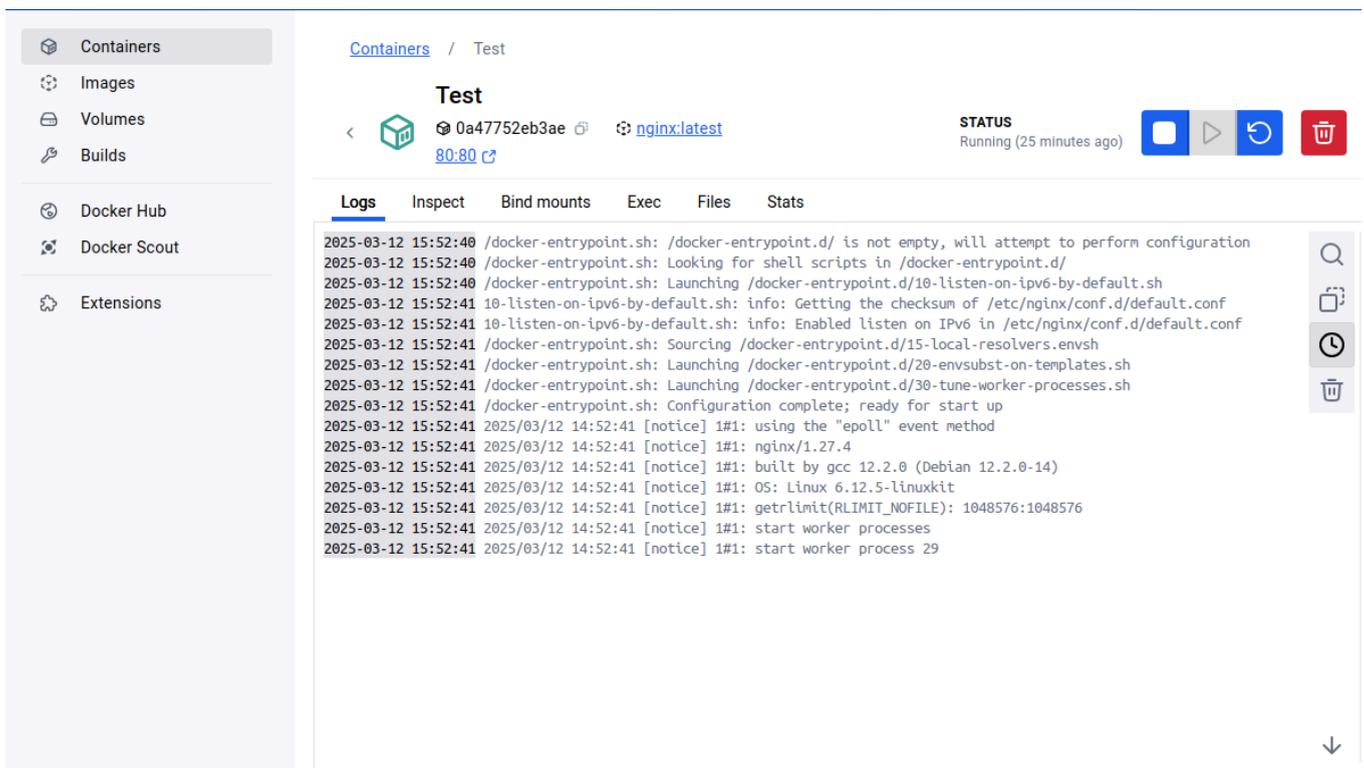
Volumes

Host path ... Container path +

Environment variables

Variable Value +

Une fois que le conteneur est configuré, normalement il démarrera automatiquement et dans la catégorie "Logs", on pourra voir s'il y a eu des erreurs au démarrage ou non .



Containers / Test

Test

0a47752eb3ae nginx:latest

STATUS Running (25 minutes ago)

Logs Inspect Bind mounts Exec Files Stats

```
2025-03-12 15:52:40 /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
2025-03-12 15:52:40 /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
2025-03-12 15:52:40 /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
2025-03-12 15:52:41 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
2025-03-12 15:52:41 10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
2025-03-12 15:52:41 /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
2025-03-12 15:52:41 /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
2025-03-12 15:52:41 /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
2025-03-12 15:52:41 /docker-entrypoint.sh: Configuration complete; ready for start up
2025-03-12 15:52:41 2025/03/12 14:52:41 [notice] 1#1: using the "epoll" event method
2025-03-12 15:52:41 2025/03/12 14:52:41 [notice] 1#1: nginx/1.27.4
2025-03-12 15:52:41 2025/03/12 14:52:41 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2025-03-12 15:52:41 2025/03/12 14:52:41 [notice] 1#1: OS: Linux 6.12.5-linuxkit
2025-03-12 15:52:41 2025/03/12 14:52:41 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025-03-12 15:52:41 2025/03/12 14:52:41 [notice] 1#1: start worker processes
2025-03-12 15:52:41 2025/03/12 14:52:41 [notice] 1#1: start worker process 29
```