

Docker

Docker est une plateforme présente sur Linux et Windows, permettant de lancer certaines applications dans des conteneurs. Ces conteneurs permettent de créer un micro-système isolé du système principal.

Et il se base sur des images que l'on peut créer soi-même ou récupérer sur le dépôt officiel de Docker.

Le dépôt est alimenté soit par des fans de certains programmes, soit par leurs créateurs originaux.

- [Installation de Docker sur debian](#)
- [Installation d'une web gui, Portainer](#)
- [Relier deux instance portainer différente](#)

Installation de Docker sur debian

Pour installé docker, j'utilise les commandes donner sur leur site [officiel](#) :

```
# 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/debian/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/debian \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

Dans un premier temps, on installe `ca-certificates` et `curl` pour pouvoir ajouter le certificat du dépôt Docker et récupérer la clé, rajoute la clé d'identification du dépôt de Docker en lui donnant les droits de lecture, puis on finit par rajouter le dépôt :

```

Après cette opération, 6 199 ko d'espace disque supplémentaires seront utilisés.
Réception de :1 http://deb.debian.org/debian bookworm/main amd64 sudo amd64 1.9.13p3-1+deb12u1 [1 889 kB]
1 889 ko réceptionnés en 0s (4 695 ko/s)
Sélection du paquet sudo précédemment désélectionné.
(Lecture de la base de données... 215973 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../sudo_1.9.13p3-1+deb12u1_amd64.deb ...
Dépaquetage de sudo (1.9.13p3-1+deb12u1) ...
Paramétrage de sudo (1.9.13p3-1+deb12u1) ...
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
debug2: client_check_window_change: changed
debug2: channel 0: request window-change confirm 0#####
debug3: send packet: type 98#####
debug2: client_check_window_change: changed
debug2: channel 0: request window-change confirm 0
debug3: send packet: type 98
Traitement des actions différées (« triggers ») pour libc-bin (2.36-9+deb12u10) ...
root@VM-Principal:/home/speed# # 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/debian/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/debian \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
Atteint :1 http://deb.debian.org/debian bookworm InRelease
Atteint :2 http://deb.debian.org/debian bookworm-updates InRelease
Atteint :3 http://security.debian.org/debian-security bookworm-security InRelease
Lecture des listes de paquets... Fait
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
ca-certificates est déjà la version la plus récente (20230311).
Les NOUVEAUX paquets suivants seront installés :
 curl
0 mis à jour, 1 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 315 ko dans les archives.
Après cette opération, 501 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] o
Réception de :1 http://deb.debian.org/debian bookworm/main amd64 curl amd64 7.88.1-10+deb12u12 [315 kB]
315 ko réceptionnés en 0s (3 961 ko/s)
Sélection du paquet curl précédemment désélectionné.
(Lecture de la base de données... 216106 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../curl_7.88.1-10+deb12u12_amd64.deb ...
Dépaquetage de curl (7.88.1-10+deb12u12) ...
Paramétrage de curl (7.88.1-10+deb12u12) ...

```

Après la fin du **"apt-get update"**, il nous reste plus qu'à faire :

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

On installe donc tous ce qui est en rapport avec docker, que ce soit son noyau et ces dépendances :

```

root@VM-Principal:/home/speed# sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
 docker-ce-rootless-extras git git-man liberror-perl libslirp0 patch pigz slirp4netns
Paquets suggérés :
 cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn ed diffutils-doc
Les NOUVEAUX paquets suivants seront installés :
 containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin git git-man liberror-perl libslirp0 patch pigz slirp4netns
0 mis à jour, 13 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 130 Mo dans les archives.
Après cette opération, 488 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] o
Réception de :1 http://deb.debian.org/debian bookworm/main amd64 pigz amd64 2.6-1 [64,0 kB]
Réception de :2 https://download.docker.com/linux/debian bookworm/stable amd64 containerd.io amd64 1.7.27-1 [30,5 MB]
Réception de :3 http://deb.debian.org/debian bookworm/main amd64 liberror-perl all 0.17029-2 [29,0 kB]
Réception de :4 http://deb.debian.org/debian bookworm/main amd64 git-man all 1:2.39.5-0+deb12u2 [2 053 kB]
Réception de :5 http://deb.debian.org/debian bookworm/main amd64 git amd64 1:2.39.5-0+deb12u2 [7 260 kB]
Réception de :6 http://deb.debian.org/debian bookworm/main amd64 libslirp0 amd64 4.7.0-1 [63,0 kB]
Réception de :7 http://deb.debian.org/debian bookworm/main amd64 patch amd64 2.7.6-7 [128 kB]
Réception de :8 http://deb.debian.org/debian bookworm/main amd64 slirp4netns amd64 1.2.0-1 [37,5 kB]
Réception de :9 https://download.docker.com/linux/debian bookworm/stable amd64 docker-buildx-plugin amd64 0.23.0-1-debian.12-bookworm [34,6 MB]
Réception de :10 https://download.docker.com/linux/debian bookworm/stable amd64 docker-ce-cli amd64 5:28.1.1-1-debian.12-bookworm [15,8 MB]
Réception de :11 https://download.docker.com/linux/debian bookworm/stable amd64 docker-ce amd64 5:28.1.1-1-debian.12-bookworm [19,2 MB]
Réception de :12 https://download.docker.com/linux/debian bookworm/stable amd64 docker-ce-rootless-extras amd64 5:28.1.1-1-debian.12-bookworm [6 095 kB]
Réception de :13 https://download.docker.com/linux/debian bookworm/stable amd64 docker-compose-plugin amd64 2.35.1-1-debian.12-bookworm [13,8 MB]
90% [13 docker-compose-plugin 45,1 kB/13,8 MB 0%]debug2: channel 0: window 999420 sent adjust 49156
130 Mo réceptionnés en 2s (59,1 Mo/s)
Sélection du paquet pigz précédemment désélectionné.
(Lecture de la base de données... 216113 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../00-pigz_2.6-1_amd64.deb ...
Dépaquetage de pigz (2.6-1) ...
Sélection du paquet containerd.io précédemment désélectionné.
Préparation du dépaquetage de .../01-containerd.io_1.7.27-1_amd64.deb ...
Dépaquetage de containerd.io (1.7.27-1) ...

```

Pour finir, nous allons faire la commande de base pour vérifier que docker fonctionne :

```
sudo docker run hello-world
```

Et on voit que docker fonctionne parfaitement bien :

```
root@VM-Principal:/home/speed# sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:c41088499908a59aae84b0a49c70e86f4731e588a737f1637e73c8c09d995654
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/

root@VM-Principal:/home/speed# |
```

Installation d'une web gui, Portainer

Portainer est une interface web pour Docker. Portainer tourne lui-même dans un conteneur et permet une configuration de Docker de façon intuitive.

Dans un premier temps, on va créer le volume persistant qui nous servira pour le conteneur avec cette commande :

```
docker volume create portainer_data
```

Voilà le volume créé :

```
root@VM-Principal:/home/speed# docker volume create portainer_data
portainer_data
root@VM-Principal:/home/speed#
```

Puis, on va taper cette commande :

```
docker run -d -p 8000:8000 -p 9443:9443 --name portainer --restart=always -v
/var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest
```

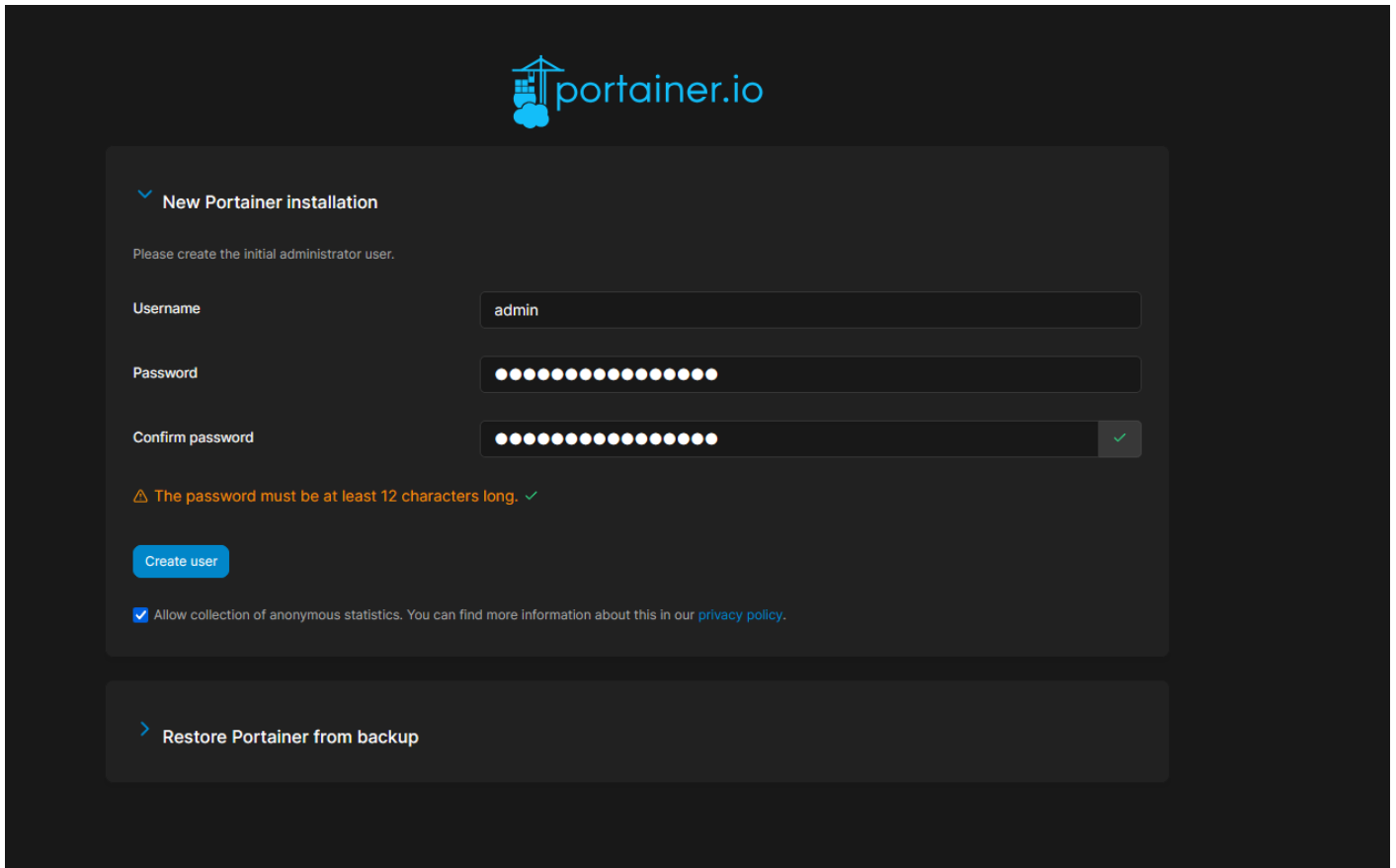
Cette commande permet à Docker de lancer la configuration du conteneur en autorisant les ports 8000 et 9443 à communiquer en dehors du conteneur (**-p**), de lui donner un nom (**--name**), de lui ordonner de redémarrer à chaque démarrage (**--restart=always**), de me définir où se trouveront les fichiers de configuration systèmes et utilisateur (**-v**) et le nom du dépôt qu'il doit utiliser pour configurer le conteneur (**portainer-ce:latest**) :

```
portainer_data
root@VM-Principal:/home/speed# docker run -d -p 8000:8000 -p 9443:9443 --name portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest
Unable to find image 'portainer/portainer-ce:latest' locally
Pulling from portainer/portainer-ce
970b305841e: Extracting [----->] 128.3kB/128.3kB
05d4fb47930: Download complete
04de093ad5ed: Downloading [----->] 192.7kB/18.85MB
a9ff7abff372: Waiting
009df2601140: Waiting
df254cde32a0: Waiting
5b34a21056c4: Waiting
faf3a09a336d: Waiting
06693e01e9e2: Waiting
4f4fb70ef54: Waiting
```

Un coup de "**docker ps**" pour vérifier que le conteneur soit bien créé et bien up :

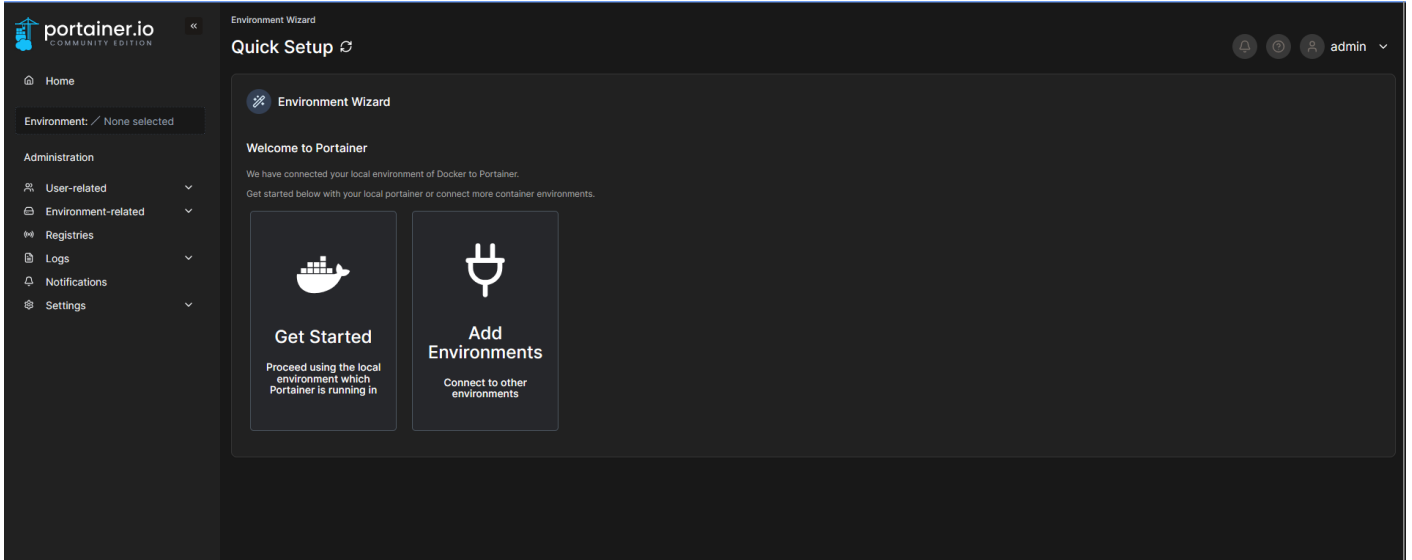
```
root@VM-Principal:/home/speed# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          PORTS                                                                 NAMES
676bb453bac3  portainer/portainer-ce:latest      "/portainer"           12 seconds ago  Up 8 seconds   0.0.0.0:8000->8000/tcp, [::]:8000->8000/tcp, 0.0.0.0:9443->9443/tcp, [::]:9443->9443/tcp, 9000/tcp  portainer
root@VM-Principal:/home/speed#
```

Pour finir, on accède à l'interface web en tapant `https://adresseip:9443` dans mon cas, `192.168.10.20:9443` :



Si jamais vous dépassez la limite de temps (environs 2min) pour la première configuration vous devrez éteindre le container **docker stop portainer** puis le redémarrer **docker start portainer**.

Vu que c'est la première fois que je me connecte à portainer, pour cela il propose comme "**Username**" admin que je vais laissé et demandé un **mot de passe** que je vais rentré puis faire "**Create user**" :



Me voilà connecté.

Relier deux instance portainer
différente