

Mise en place de Docker

Qu'est ce que Docker ?

Docker est une plateforme open-source conçue pour automatiser le déploiement, la mise à l'échelle et la gestion d'applications dans des conteneurs. Un conteneur est une unité standard de logiciel qui regroupe le code et toutes ses dépendances, de sorte que l'application s'exécute rapidement et de manière fiable d'un environnement informatique à un autre. Docker simplifie la création, le déploiement et l'exécution d'applications en utilisant des conteneurs qui peuvent fonctionner sur n'importe quel serveur, qu'il soit sur un cloud, sur site ou sur un ordinateur personnel.

Pourquoi choisir Docker ?

Docker offre plusieurs avantages majeurs qui expliquent son adoption massive par les développeurs et les équipes IT :

Portabilité : Les conteneurs Docker peuvent être exécutés sur n'importe quelle machine, quel que soit le système d'exploitation sous-jacent, du moment que Docker est installé.

Isolation : Chaque conteneur fonctionne de manière isolée, ce qui améliore la sécurité et évite les conflits entre les applications.

Rapidité : Le démarrage des conteneurs est quasi-instantané, ce qui permet des cycles de développement, de test et de déploiement plus rapides.

Efficacité : Docker utilise les ressources système de manière optimale, permettant de faire tourner plusieurs conteneurs sur un même hôte avec une surcharge minimale.

Gestion simplifiée des dépendances : Docker garantit que les applications contiennent toutes leurs dépendances, ce qui élimine les problèmes liés à des environnements non standardisés.

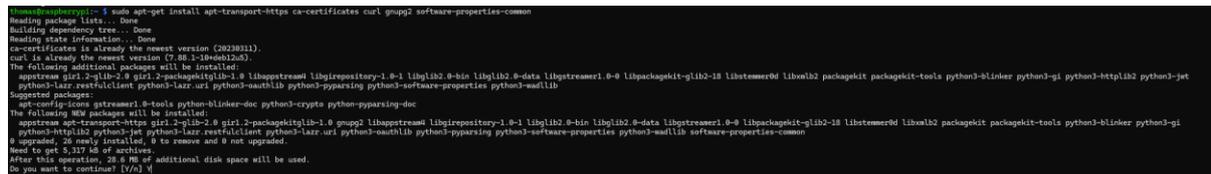
Qu'est ce que portainer ?

Portainer est une interface utilisateur graphique (GUI) open-source pour Docker qui simplifie la gestion de vos environnements Docker et Kubernetes. Portainer permet aux utilisateurs de visualiser, de gérer et de surveiller leurs conteneurs, images, réseaux et volumes Docker, offrant une alternative conviviale à l'utilisation de la ligne de commande Docker. Avec Portainer, les utilisateurs peuvent facilement déployer des applications, surveiller les performances et gérer les ressources, ce qui rend la gestion des conteneurs plus accessible, même pour ceux qui ne sont pas des experts en ligne de commande.

Installation Docker ?

Installation des dépendances :

```
sudo apt-get install apt-transport-https ca-certificates curl gnupg2 software-properties-common
```



```
root@kali:~# sudo apt-get install apt-transport-https ca-certificates curl gnupg2 software-properties-common
Building package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311).
curl is already the newest version (7.88.1-10+deb12u5).
The following additional packages will be installed:
  aptstream gir1.2-glib-2.0 gir1.2-packagekitglib-1.0 libappstream libgirepository-1.0-1 libglib2.0-bin libglib2.0-data libgststreamer1-0 libpackagekit-glib2-18 libstemmer0d libxwalb2 packagekit packagekit-tools python3-blinker python3-gi python3-httplib2 python3-jet
  python3-lazr restclient python3-lazr.uri python3-oauthlib python3-pyparsing python3-software-properties python3-wadllib
Suggested packages:
  apt-config-icons gstreamer1.0-tools python3-blinker-doc python3-crypts python-pyparsing-doc
The following NEW packages will be installed:
  aptstream apt-transport-https gir1.2-glib-2.0 gir1.2-packagekitglib-1.0 gnupg2 libappstream libgirepository-1.0-1 libglib2.0-bin libglib2.0-data libgststreamer1-0 libpackagekit-glib2-18 libstemmer0d libxwalb2 packagekit packagekit-tools python3-blinker python3-gi
  python3-httplib2 python3-jet python3-lazr restclient python3-lazr.uri python3-oauthlib python3-pyparsing python3-software-properties python3-wadllib software-properties-common
0 upgraded, 26 newly installed, 0 to remove and 0 not upgraded.
Need to get 5317 kB of archives.
After this operation, 28.6 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Ajout de la clé GPG du dépôt Docker : Importez la clé GPG officielle de Docker :

```
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

Ajout du dépôt Docker aux sources APT :

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/debian $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list
```

```
sudo apt-get update
```

```

thomas@raspberrypi:~$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
thomas@raspberrypi:~$ echo "deb [arch=arm64] https://download.docker.com/linux/debian $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list
deb [arch=arm64] https://download.docker.com/linux/debian $(lsb_release -cs) stable
thomas@raspberrypi:~$ sudo apt-get update
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 http://deb.debian.org/debian-security bookworm-security InRelease
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease
Hit:4 http://archive.raspberrypi.com/debian bookworm InRelease
Get:5 https://download.docker.com/linux/debian bookworm InRelease [43.3 kB]
Get:6 http://deb.debian.org/debian bookworm/main arm64 DEP-11 Metadata [6,711 kB]
Get:7 https://download.docker.com/linux/debian bookworm/stable arm64 Packages [19.4 kB]
Get:8 http://deb.debian.org/debian bookworm/contrib arm64 DEP-11 Metadata [12.7 kB]
Get:9 http://deb.debian.org/debian bookworm/non-free arm64 DEP-11 Metadata [2,048 B]
Get:10 http://deb.debian.org/debian bookworm/non-free-firmware arm64 DEP-11 Metadata [15.6 kB]
Fetched 95.6 MB in 10s (9,318 kB/s)
Reading package lists... Done

```

Installation de Docker Engine :

sudo apt-get install docker-ce docker-ce-cli containerd.io

```

thomas@raspberrypi:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-buildx-plugin docker-ce-rootless-extras docker-compose-plugin libltdl7 libsllp0 pigz slirp4netns
Suggested packages:
  cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libsllp0 pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 95.6 MB of archives.
After this operation, 369 MB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bookworm/main arm64 pigz arm64 2.6-1 [56.2 kB]
Get:2 http://deb.debian.org/debian bookworm/main arm64 libltdl7 arm64 2.4.7-5 [392 kB]
Get:3 http://deb.debian.org/debian bookworm/main arm64 libsllp0 arm64 4.7.0-1 [58.4 kB]
Get:4 https://download.docker.com/linux/debian bookworm/stable arm64 containerd.io arm64 1.6.28-2 [21.4 MB]
Get:5 http://deb.debian.org/debian bookworm/main arm64 slirp4netns arm64 1.2.0-1 [36.6 kB]
Get:6 https://download.docker.com/linux/debian bookworm/stable arm64 docker-buildx-plugin arm64 0.13.1-1-debian.12-bookworm [27.0 MB]
Get:7 https://download.docker.com/linux/debian bookworm/stable arm64 docker-ce-cli arm64 5:26.0-0-1-debian.12-bookworm [12.5 MB]
Get:8 https://download.docker.com/linux/debian bookworm/stable arm64 docker-ce arm64 5:26.0-0-1-debian.12-bookworm [15.3 MB]
Get:9 https://download.docker.com/linux/debian bookworm/stable arm64 docker-ce-rootless-extras arm64 5:26.0-0-1-debian.12-bookworm [8,403 kB]
Get:10 https://download.docker.com/linux/debian bookworm/stable arm64 docker-compose-plugin arm64 2.25.0-1-debian.12-bookworm [10.5 MB]
Fetched 95.6 MB in 10s (9,318 kB/s)

```

Activation et démarrage du service Docker :

sudo systemctl enable docker

sudo systemctl start docker

Vérification du statut du service Docker :

sudo systemctl status docker

docker --version

Installation de Portainer

Déploiement de Portainer

1. **Création d'un volume pour les données de Portainer** : Créez un volume Docker pour conserver les données de Portainer :

1. `docker volume create portainer_data`

Lancement de Portainer : Démarrez une instance de Portainer à l'aide de la commande suivante. Assurez-vous de spécifier la version la plus récente ou celle que vous souhaitez utiliser :

2. `sudo docker run -d -p 8000:8000 -p 9000:9000 -p 9443:9443 --name=portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest`

```
thomas@raspberrypi:~$ sudo docker volume create portainer_data
portainer_data
thomas@raspberrypi:~$ sudo docker run -d -p 8000:8000 -p 9000:9000 -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
latest: Pulling from portainer/portainer-ce
379538b6d68e: Pull complete
4ea3e2c3a39b: Pull complete
23f2184d3136: Pull complete
e21d017187f1: Pull complete
bfa9cfee4c8e: Pull complete
9d8366b4fa62: Pull complete
334f906d8c08: Pull complete
faa0c7e0dbf8: Pull complete
f82574fb5a84: Pull complete
b4e3f4546933: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:908d04d20e86f07a50b0f1a029a11b89aa3089b7fc7fdf68ec1c71b025f36cd
Status: Downloaded newer image for portainer/portainer-ce:latest
```

Les paramètres sont définis comme suit :

- **-d** : Exécute le conteneur en arrière-plan (mode détaché).
- **-p** : Publie les ports du conteneur vers l'hôte.
- **--name** : Nomme le conteneur.
- **--restart=always** : Assure que le conteneur redémarre automatiquement en cas de défaillance.
- **-v** : Monte des volumes pour la persistance des données et pour permettre à Portainer de communiquer avec le daemon Docker.

Accès à Portainer

- **Portainer UI** : Portainer sera accessible via un navigateur web à l'adresse **http://<adresse_ip_du_raspberry>:9000** ou **https://<adresse_ip_du_raspberry>:9443** pour une connexion sécurisée par SSL.

Une fois Portainer déployé, vous pourrez gérer vos conteneurs Docker via son interface utilisateur web. Vous pourrez y créer des conteneurs, des volumes, des réseaux, et surveiller l'utilisation de vos ressources par les conteneurs.

