

Introduction à Docker

Docker est un projet open source (Apache 2.0) écrit en GO et hébergé sur GitHub: <https://github.com/docker> (<https://github.com/docker>).

Initialement porté par la startup DotCloud (renommée depuis Docker) fondée par deux français anciens de l'Epitech.

Docker est composé de trois éléments :

- le daemon Docker qui s'exécute en arrière-plan et qui s'occupe de gérer les conteneurs (Containerd avec runC)
- une API de type REST qui permet de communiquer avec le daemon
- Le client en CLI (command line interface) : commande `docker`

Par défaut, le client communique avec le daemon Docker via un socket Unix (`/var/run/docker.sock`) mais il est possible d'utiliser un socket TCP.

Docker c'est aussi un dépôt d'images (aussi appelé registry) :

<https://store.docker.com> (<https://store.docker.com>)

Il contient les images officielles maintenues par Docker mais aussi celles mises à disposition par d'autres contributeurs.

Quelques concepts:

- une image est un ensemble de fichiers inertes en read-only.
- Un conteneur est une instance une active (started) ou inactive (stopped) d'une image. L'exécution d'un conteneur n'altère jamais une image.

Lexique

- **Conteneur** : Image exécutable d'un environnement complet incluant code, bibliothèques, outils et configuration
- **Image** : template de conteneur en read-only contenant un système de base et une application.
- **Docker HUB** : Dépôt public d'images mises à disposition par Docker
DockerHub (<https://store.docker.com>)
- **Dockerfile** : fichier texte de description d'une image

- **Docker Compose** : fichier texte (yaml) de description d'un ensemble de conteneurs
- **Docker Machine** : Outil de déploiement des hôtes Docker sur différentes plateformes (Mac, Windows) : <https://docs.docker.com/machine/overview/>
(<https://docs.docker.com/machine/overview/>)
- **Orchestrateur** : gère un pool de ressources serveurs (Swarm, Kubernetes, Mesos, Rancher...)
- **Registry** : Dépôt privé d'images Docker

Installation de Docker

Méthode d'installation officielle (<https://docs.docker.com/install/linux/docker-ce/centos/>)

Installer les prérequis (Centos 7.x amd64)

```
$ sudo yum install -y yum-utils device-mapper-persistent-data lvm2
```

Configurer le dépôt officiel et installer Docker en version CE (Community Edition)

```
sudo yum-config-manager --add-repo https://download.docker.com/linux/ce
sudo yum install docker-ce
```

Activer et démarrer le service

```
$ sudo systemctl enable docker
$ sudo systemctl start docker
```

Vérifier l'installation

```
$ sudo docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correct
```

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 run executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which 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/>

Détails des opérations réalisées par cette commande:

- 1: le client docker se connecte au daemon docker via le socket Unix
- 2: l'image "hello-world" n'étant pas présente localement, le daemon Docker la télécharge depuis la registry Docker Hub
- 3: le daemon Docker crée un nouveau conteneur depuis cette image dont la finalité est de produire le message ci-dessus.
- 4: le daemon Docker renvoi le message au client Docker pour afficher le résultat dans le terminal

Vérifier la version