
DOCKER

Principe

Docker est un logiciel libre destiné aux développeurs et administrateurs systèmes, dont l'objectif est de faciliter le développement, la diffusion et le déploiement d'applications web autonomes.

Son principal intérêt est **d'assembler les briques d'une application en conteneurs pouvant être partagés, sous forme d'images et de les exécuter quels que soient la plateforme et l'environnement.**

Grâce à Docker, **les applications deviennent portables et exécutables n'importe où** : sur des ordinateurs portables Mac ou Windows, des machines virtuelles, de serveurs de production, etc.

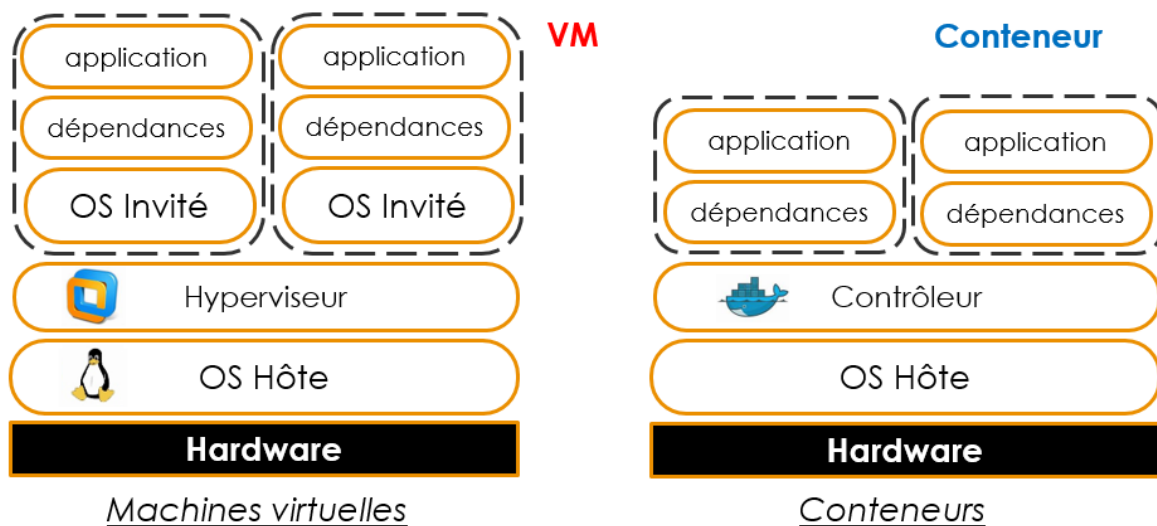
Origines

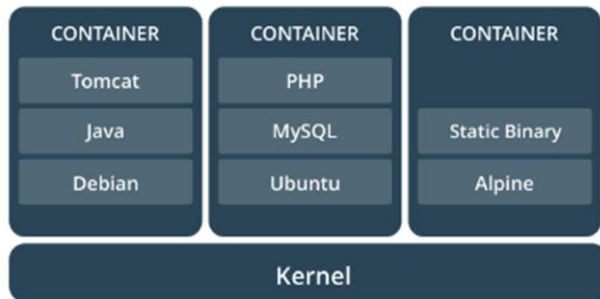
Docker a été initialement développé par Solomon Hykes pour un projet interne à la société de plateforme en tant que service (platform as a service - PaaS) qu'il a créée en 2008, dotCloud.

Docker a été distribué en tant que projet open source en 2013, projet qui a rapidement été très actif avec de nombreux contributeurs.

Fonctionnement

Docker est une technologie de virtualisation par conteneurs



Exemples d'utilisation

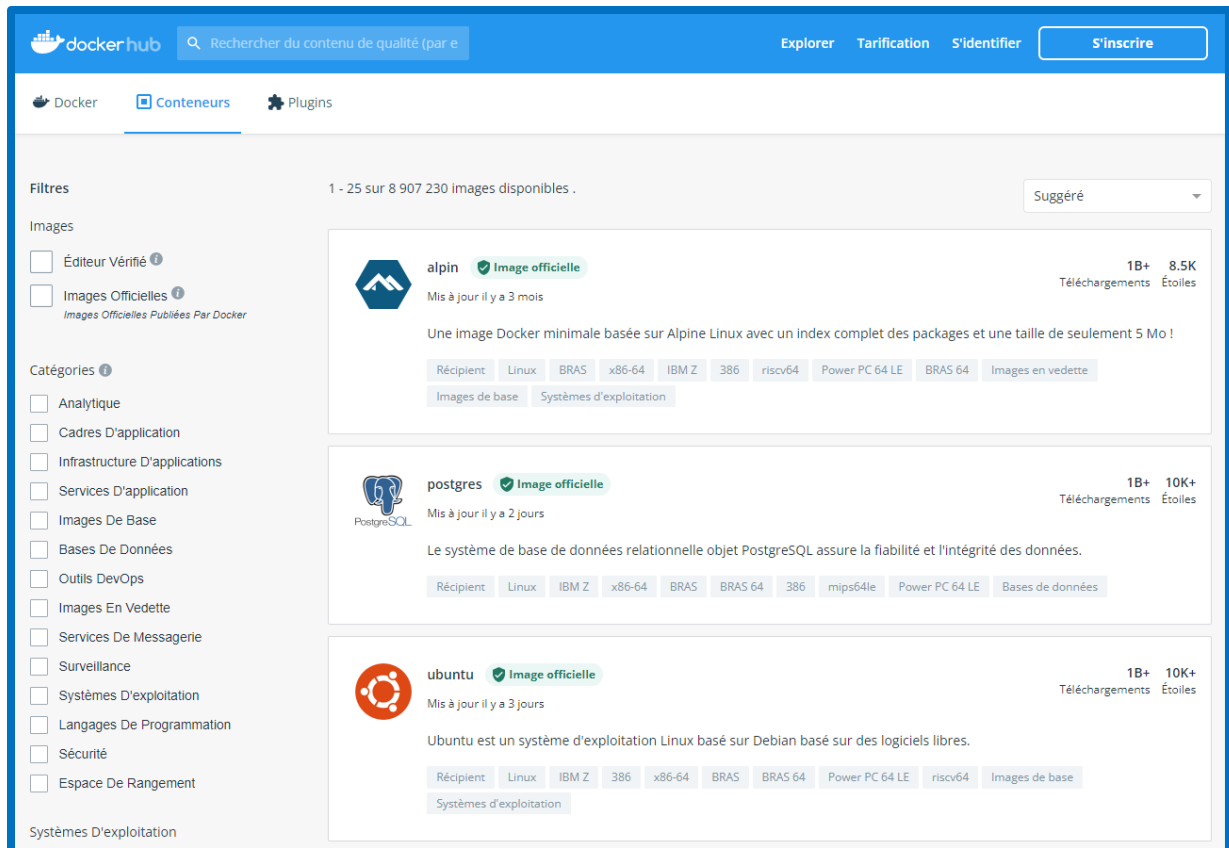
Les conteneurs Dockers offrent une grande flexibilité qui permet de les créer, déployer, mettre à jour, copier et déplacer d'un environnement à un autre.

Les conteneurs Docker sont actuellement pris en charge par les principaux acteurs du cloud. Ainsi, un conteneur Docker peut se déployer en de multiples instances sur toutes les principales distributions Linux, Microsoft Windows, et sur toutes les infrastructures.

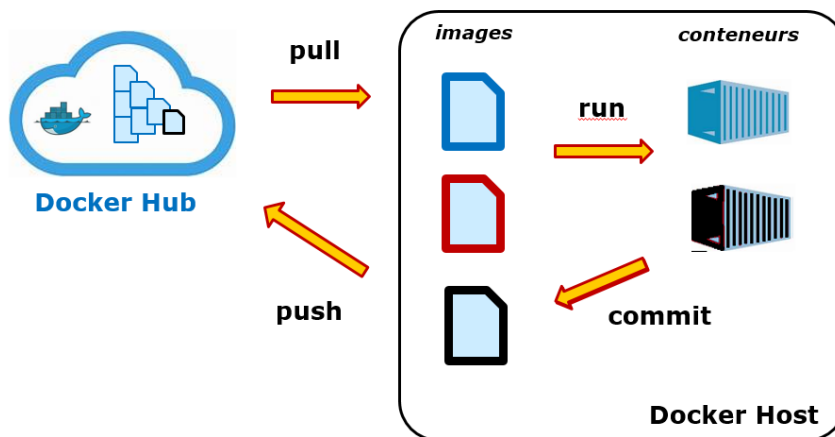
Où trouver les images ?

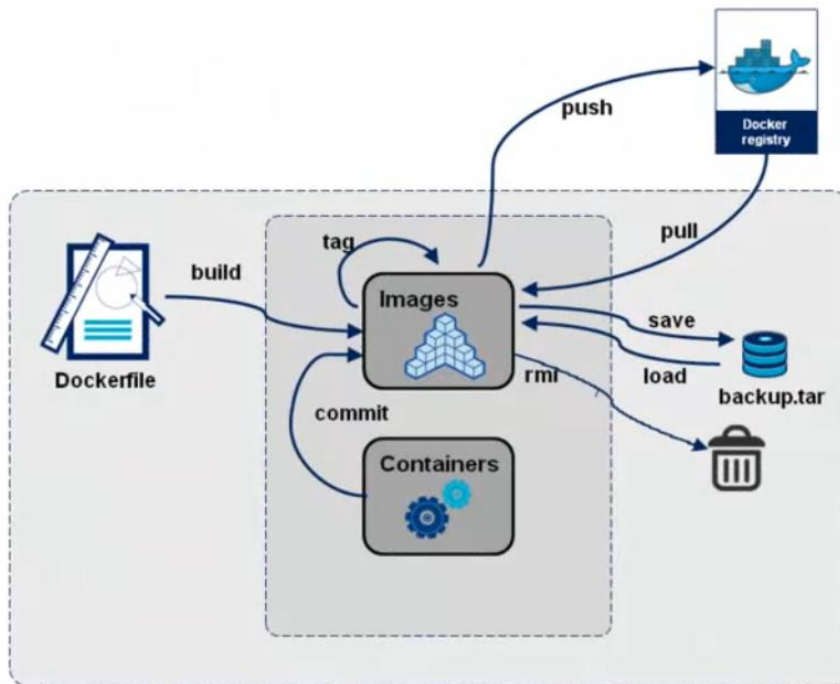
De nombreuses images allant de la simple application comme Nextcloud à un système d'exploitation complet comme Debian sont disponibles sur :

- Le registre officiel (appelé hub) : <https://registry.hub.docker.com> ;
- De nombreux dépôts initiés par de « simples » utilisateurs.
- Il est bien sûr possible de proposer des images, d'en modifier d'autres et de déposer la modification sur le dépôt officiel.



Gestion des images





Avantages

- Exécuter plusieurs machines virtuelles sur un même serveur demande des ressources suffisamment performantes pour assumer plusieurs machines virtuelles
- Les conteneurs Docker démarrent très rapidement et utilisent peu de ressources (processeur, mémoire vive, etc.)
- Les conteneurs sont légers : cela réduit l'utilisation du disque, ils se migrent plus facilement d'une machine physique à une autre et les téléchargements d'images sont beaucoup plus rapides

Inconvénients

- Les conteneurs peuvent être vulnérables, car ils partagent un noyau et des composants systèmes et leur fonctionnement exige généralement l'accès root : si toute l'architecture est basée sur Docker et si le système hôte est attaqué tous les services seront « accessibles » et exposés plus facilement

Installation

Avant d'installer Docker, vérifiez qu'une ancienne version ne soit pas déjà présente.

```
apt update
wget https://get.docker.com/
```

« Bash index.html »

Après cette commande, il faut patienter quelques minutes

Commandes à tester

```
## List Docker CLI commands
docker
docker container --help

## Display Docker version and info
docker --version
docker version
docker info

## Execute Docker image
docker run hello-world

## List Docker images
docker image ls

## List Docker containers (running, all, all in quiet mode)
docker container ls
docker container ls --all
docker container ls -a -q
```

Manipuler les conteneurs

https://librecours.net/module/picasoft/run/dk01/2-manipuler_conteneurs.xhtml#:~:text=Pour%20cela%20on%20utilise%20simplement,termin%C3%A9%20le%20conteneur%20s'arr%C3%AAta.

Systemctl status docker

```
For more help on how to use Docker, head to https://docs.docker.com/go/guides/
root@docker:~# systemctl status docker
• docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2023-03-28 11:00:53 CEST; 10min ago
  Docs: https://docs.docker.com
  Main PID: 1982 (dockerd)
  Tasks: 9
  Memory: 41.4M
  CGroup: /system.slice/docker.service
          └─1982 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

Création d'un docker

Docker run -d -p 8001 :80 --name wiki01 mprasil/dokuwiki

Afficher un docker

Docker ps

Entrer dans le docker

Docker exec -it wiki01 bash

Lancer un container

Docker start [NOM]

Docker pull Ubuntu

Rappel:

- Un conteneur est la version exécutée d'une image Docker. Il possède la copie du système de fichier de l'image, ainsi que la capacité de lancer des processus.
- Lorsqu'un conteneur est lancé à partir d'une image, Docker va monter le système de fichier de l'image en lecture seule et monter une couche accessible en lecture/écriture.
- Dans ce conteneur, il sera donc possible d'interagir avec les applications installées dans l'image, d'exécuter des scripts, de modifier des fichiers, d'installer d'autres applications, etc.

Docker ps -a

Docker run ubuntu cat /etc/lsb-release

Docker rm pour supprimer un container


Docker run -ti ubuntu /bin/bash

Pour mettre le serveur web apache2 sur ubuntu il faut juste créer un conteneur Ubuntu, rentrer dedans et installer le service apache2 et le lancer avec les commandes :

Docker exec -it ubuntu bash

Apt install apache2

Service apache2 start



Apache2 Default Page

Ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```

/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf

```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2` and is managed using `systemd`, so to start/stop the service use `systemctl start apache2` and `systemctl stop apache2`, and use `systemctl status apache2` and `journalctl -u apache2` to check status. `system` and `apache2ctl` can also be used for service management if desired. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file outside of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to their respective packages, not to the web server itself.

Par définition, le conteneur est éphémère

- La commande `docker run` va recréer un conteneur alors que la commande `exec -it` va rentrer dans un conteneur déjà existant.