

FONCTIONNEMENT ET MANIPULATION DES IMAGES DOCKER

Qu'est qu'une image Docker

Sur Docker, **un conteneur est lancé en exécutant une image**. "Attends mais **c'est quoi une image Docker**" ?

Une image est un package qui inclut tout ce qui est nécessaire à l'exécution d'une application, à savoir :

- **Le code**
- **L'exécution**
- **Les variables d'environnement**
- **Les bibliothèques**
- **Les fichiers de configuration**

Dans le chapitre suivant vous allez en savoir plus sur les conteneurs. Pour le moment tout ce que vous devez retenir, c'est qu'une image docker est créée à partir d'un fichier nommé le **Dockerfile**. Une image est un modèle **composé de plusieurs couches**, ces couches contiennent notre application ainsi que les fichiers binaires et les bibliothèques requises. Lorsqu'une image est **instanciée**, son nom est un conteneur, un conteneur est donc une image en cours d'exécution.

Pour mieux comprendre le système de couche, imaginons par exemple qu'on souhaite déployer notre application web dans un serveur LAMP (Linux Apache MySQL PHP) au moyen de Docker. Pour créer notre stack (pile en français), nous

aurons besoin de :

- Une couche OS pour exécuter notre Apache, MySQL et Php
- Une couche Apache pour démarrer notre serveur web et pourquoi pas la config qui va avec (.htaccess, apache2.conf, site-available/, etc ...)
- Une couche php qui contiendra un interpréteur Php mais aussi les bibliothèques qui vont avec (exemple : php-curl)
- Une couche Mysql qui contiendra notre système de gestion de bases de données Mysql

Au total, notre image docker sera composée de quatre couches, en schéma ceci nous donnerai :

Image docker : LAMP

Couche 4 (mysql)

Couche 3 (php, php-curl, etc ...)

Couche 2 (apache)

Couche 1 (OS Ubuntu)

Il est important de bien **différencier une image Docker d'un conteneur Docker** car ce sont deux choses distinctes, Sur le chapitre précédent nous avons téléchargé et exécuté l'image "hello-world", je vais m'appuyer au début sur cette image pour vous dévoiler quelques **commandes de manipulation d'images Docker** et par la suite nous téléchargerons d'autres images.

Quelques commandes

Récupérer des informations

Pour commencer on va d'abord récupérer la liste des commandes possible :

```
docker help
```

Résultat:

```
Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  -v, --version          Print version information and quit
  ...

Management Commands:
  builder      Manage builds
  ...

Commands:
  ...
  info         Display system-wide information
  ...
  Run 'docker COMMAND --help' for more information on a command.
```

Je n'ai pas listé toutes les commandes car il y en a beaucoup, mais sur votre terminal, vous verrez beaucoup plus d'options que moi.

Sur l'output de la commande help, nous avons une information super intéressante et croyez-moi elle vous sera d'une grande utilité et vous permettra de gagner beaucoup de temps . Je parle de la ligne suivante `Run 'docker COMMAND --help' for more information on a command`. Cette ligne nous informe qu'il est possible d'avoir de l'**auto-complétion** sur n'importe quelles sous-commandes/options de Docker, je m'explique, si vous tapez la commande suivante (avec un espace à la fin et sans la lancer) :

```
docker volumes
```

Si à ce moment vous appuyez sur la touche tabulation, alors ça vous affichera toutes les options possibles pour la sous-commande `volumes`, ainsi que leurs descriptions.

Résultat:

```
create  -- Create a volume
inspect -- Display detailed information on one or more volumes
ls      -- List volumes
prune   -- Remove all unused volumes
rm      -- Remove one or more volumes
```

Nice non ?

Sur le chapitre précédent, nous avons vu comment afficher la version du moteur Docker avec la commande suivante :

```
docker --version
```

Résultat:

```
Docker version 18.09.6, build 481bc77
```

C'est cool d'avoir une telle information, mais il existe une autre commande qui permet d'afficher encore plus de détails sur votre installation de Docker:

```
docker info
```

Cette commande, nous fournit plusieurs informations concernant les spécifications du moteur Docker. Elle nous transmet aussi d'autres informations telles que le nombre total de conteneurs tournant sur notre machine ainsi que leur état :

```
Containers: 3  
Running: 0  
Paused: 0  
Stopped: 3
```

Voir le nombre total d'images disponibles sur notre machine :

```
Images: 1
```

Hmm, par contre ça ne m'affiche que le nombre d'images disponibles sur ma machine, "Est-ce-que c'est possible d'avoir davantage d'informations sur mes images ?" La réponse est oui ! On va voir ça tout de suite .

Lister les images Docker téléchargées

Sur le chapitre précédent vous aviez lancé la commande `docker run hello-world`. Pour information cette commande télécharge une image depuis le [Docker Hub Registry](#) et l'exécute dans un conteneur. Lorsque le conteneur s'exécute, il vous affiche un petit message d'information et se ferme directement.

Voici la commande qui permet de répertorier les images Docker téléchargées sur votre ordinateur :

```
docker image ls
```

ou bien la commande :

```
docker images
```

Résultat:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hello-world	latest	fce289e99eb9	5 months ago	1.84M

Elle nous donne différentes informations dont :

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
Le titre REPOSITORY peut porter à confusion, c'est essentiellement le nom de l'image.	un tag ici est une façon de faire référence à votre image, ils sont utilisés principalement pour affecter une version à une image	L'identifiant de l'image (unique pour chaque image téléchargée)	Date de la dernière modification de l'image	Taille de l'image

Supprimer une image Docker

Maintenant si on souhaite supprimer une image Docker, on aura besoin soit de son IMAGE ID soit de son nom. Une fois qu'on aura récupéré ces informations, on peut passer à la suite en lançant la commande suivante :

avec l'id de l'image :

```
docker rmi fce289e99eb9
```

avec le nom de l'image :

```
docker rmi hello-world
```

Si vous lancez cette commande vous aurez le message d'erreur suivant :

```
Error response from daemon: conflict: unable to remove repository reference "hello-world" - container 3e444920f82d is using its referenced image fce289e99eb9
```

Ce message d'erreur nous explique, qu'on ne peut pas supprimer notre image Docker car des conteneurs ont été instanciés depuis notre image "hello-world". En gros si on jamais on supprime notre image "hello-world", ça va aussi supprimer nos conteneurs car ils se basent sur cette image. Dans notre cas ça ne pose aucun problème, d'ailleurs pour résoudre ce problème l'erreur nous informe qu'on doit **forcer la suppression** pour éliminer aussi les conteneurs liés à notre image **(must force)**.

Pour forcer la supprimer on va utiliser l'option **--force** ou **-f**.

```
docker rmi -f hello-world
```

Information

Si jamais vous possédez plusieurs images avec le même nom mais avec des tags différents alors vous devez préciser le tag dans votre commande **rmi**, pour ainsi être sûr de supprimer la bonne image. Pour information par défaut Docker prend le tag **latest**.

Bonus

Voici une commande qui permet de supprimer toutes les images disponibles sur notre machine :

```
docker rmi -f $(docker images -q)
```

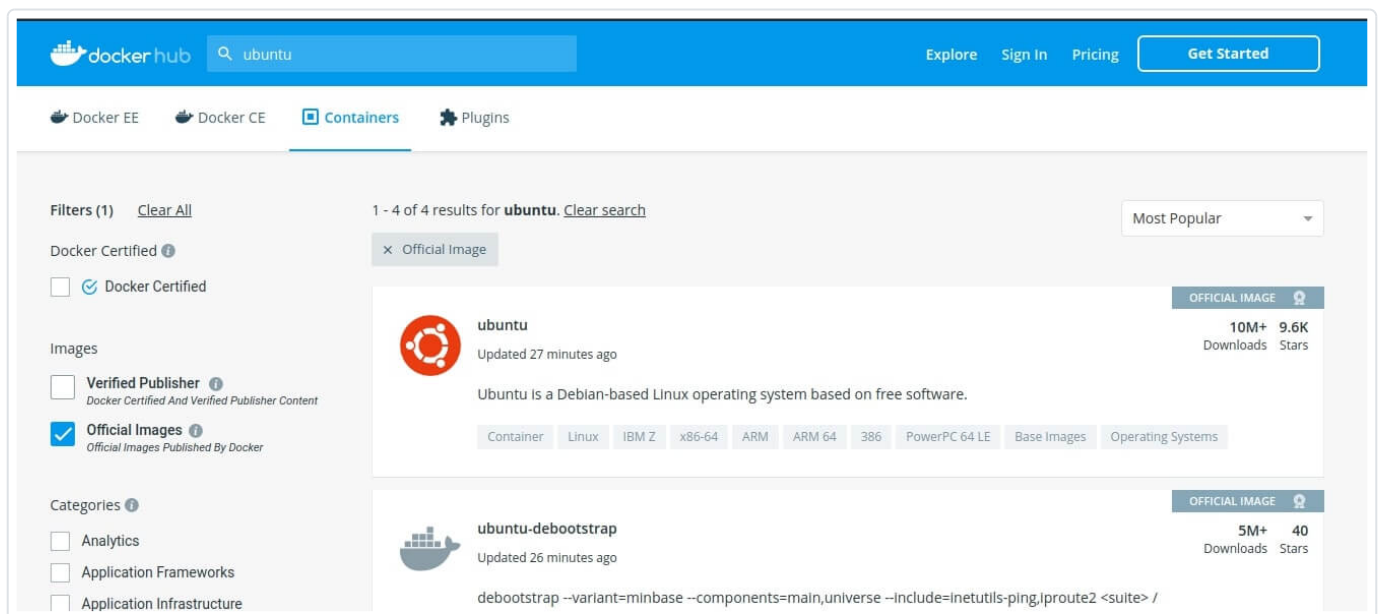
[Télécharger une image depuis le Docker Hub Registry](#)

[Lister les images disponibles sur le Docker Hub Registry](#)

Maintenant rentrons dans du concret et téléchargeons une image plus utile comme par exemple l'image officielle d'Ubuntu. "Oui mais où est-ce que je peux retrouver la liste des images disponibles ?"

Très bonne question au tout début je vous ai parlé du [Docker Hub Registry](#). Pour faire simple un Registry (registre en français) est une application côté serveur qui permet de stocker et de distribuer des images Docker, le Docker Hub Registry est le registre officiel de Docker.

Il existe deux façons pour voir si une image est disponible dans le Docker Hub Registry, la première consiste à visiter le [site web](#) et vous recherchez directement le nom de l'image dans la barre de recherche :



Vous remarquerez sur l'image que j'ai coché la case "Official Image" pour ne m'**afficher que les images officielles**. Ainsi je m'assure que l'image Ubuntu que je vais télécharger a bien été créée par l'équipe gérant la distribution Ubuntu.

Il faut bien faire attention aux images qu'on télécharge sur le net. Il faut toujours vérifier au préalable le code source de l'image plus précisément le fichier Dockerfile (ne vous inquiétez on verra le Dockerfile dans un prochain chapitre), car on est

jamais à l'abri d'avoir des images contenant des programmes vulnérables voire même des images malhonnêtes.

Je ne dis pas non plus qu'il ne faut télécharger que des images officielles mais juste faire attention à ce qu'on télécharge sur le net. Car d'un autre côté il ne faut pas oublier qu'il existe dans le Docker Hub Registry une multitude d'images créées par des utilisateurs expérimentés indépendants. Ces images sont parfois bien plus optimisées que les images officielles car certaines images n'utilisent que le strict minimum pour faire fonctionner leur application permettant ainsi de réduire la taille de l'image.

Bref revenons à nos moutons, si je clique sur l'image officielle d'ubuntu je tombe sur la page suivante :

 dockerhub

ubuntu

ExploreSign InPricingGet Started



ubuntu ☆

Docker Official Images

Ubuntu is a Debian-based Linux operating system based on free software.

10M+

ContainerLinux386PowerPC 64 LEIBM Zx86-64ARMARM 64Base ImagesOperating Systems

Official Image

Linux - x86-64 (latest)

Copy and paste to pull this image

docker pull ubuntu

View Available Tags

DESCRIPTIONREVIEWSTAGS

Supported tags and respective Dockerfile links

- 18.04, bionic-20190515, bionic, latest (bionic/Dockerfile)
- 18.10, cosmic-20190515, cosmic (cosmic/Dockerfile)
- 19.04, disco-20190515, disco, rolling (disco/Dockerfile)
- 19.10, eoan-20190508, eoan, devel (eoan/Dockerfile)
- 14.04, trusty-20190515, trusty (trusty/Dockerfile)
- 16.04, xenial-20190515, xenial (xenial/Dockerfile)

Quick reference

- Where to get help:**
the Docker Community Forums, the Docker Community Slack, or Stack Overflow
- Where to file issues:**
the cloud-images bug tracker (include the `docker` tag)
- Maintained by:**
Canonical and Tianon (Debian Developer)
- Supported architectures:** (more info)
amd64, arm32v7, arm64v8, i386, ppc64le, s390x

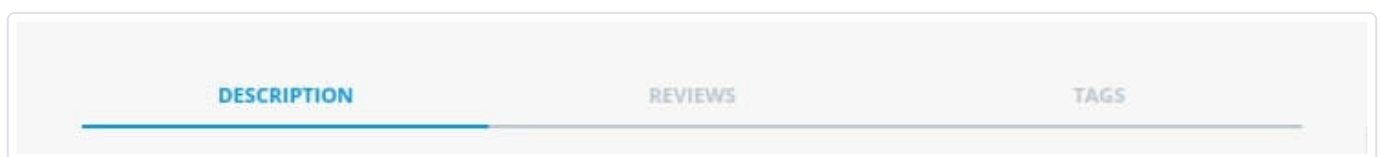
Please log in to write a review of this product.

en haut à droite on retrouve le nom de l'image avec une toute petite description



Plus bas on retrouve, un menu de navigation contenant :

- **DESCRIPTION** : Description de l'image, souvent on retrouve quelques tags, la configuration de votre conteneur (par exemple la config de votre base de données pour une image basé sur du mysql) et les liens github vers les sources du projet.
- **REVIEWS** : l'avis des utilisateurs
- **TAGS** : les différents tags disponible pour cette image



Enfin en haut à droit nous avons la commande à lancer permettant de télécharger l'image sur.

Copy and paste to pull this Image

```
docker pull ubuntu
```



[View Available Tags](#)

La deuxième manière pour lister les images disponibles dans le Docker hub Registry, c'est de passer par la ligne de commande :

```
docker search ubuntu
```

Résultat:

NAME	DESCRIPTION
ubuntu	Ubuntu is a Debian-based Linux operating system.
...	
pivotaldata/ubuntu-gpdb-dev	Ubuntu images for GPDB development.

Si vous souhaitez n'afficher que les images officielles, il est possible de filtrer le résultat avec la commande suivante :

```
docker search --filter "is-official=true" ubuntu
```

Vous aurez ainsi beaucoup moins de résultats.

Télécharger une images depuis le Docker Hub Registry

Je pense que vous l'aurez deviné, pour télécharger une image depuis le Docker hub Registry il faut utiliser la commande suivante (précisez le tag si vous souhaitez un tag différent de **latest**)

```
docker pull ubuntu
```

Pour télécharger une image ubuntu avec un autre tag différent de **latest** par exemple le tag **16.04**:

```
docker pull ubuntu:16.04
```

Je vais faire un petit coup de **docker images** pour si mon image s'est bien téléchargée.

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	16.04	2a697363a870	3 weeks ago	119MB
ubuntu	latest	7698f282e524	3 weeks ago	69.9MB
hello-world	latest	fce289e99eb9	5 months ago	1.84MB

Youpi !

Conclusion

Vous savez maintenant **lister, télécharger, supprimer et rechercher une image Docker**, je vous mets en bas un **aide-mémoire** des commandes vu ensemble :

```
## Afficher de l'aide
docker help
docker <sous-commande> --help

## Afficher des informations sur l'installation de Docker
docker --version
docker version
docker info

## Executer une image Docker
docker run hello-world

## Lister des images Docker
docker image ls
# ou
docker images

## Supprimer une image Docker
docker images rmi <IMAGE_ID ou IMAGE_NAME> # si c'est le nom de l'image qui est spécifié
-f ou --force : forcer la suppression
```

```
## Supprimer tous les images Docker
docker rmi -f $(docker images -q)

## Rechercher une image depuis le Docker hub Registry
docker search ubuntu
    --filter "is-official=true" : Afficher que les images officielles

## Télécharger une image depuis le Docker hub Registry
docker pull <IMAGE_NAME> # prendra par défaut le tag latest
docker pull ubuntu:16.04 # prendra le tag 16.04
```