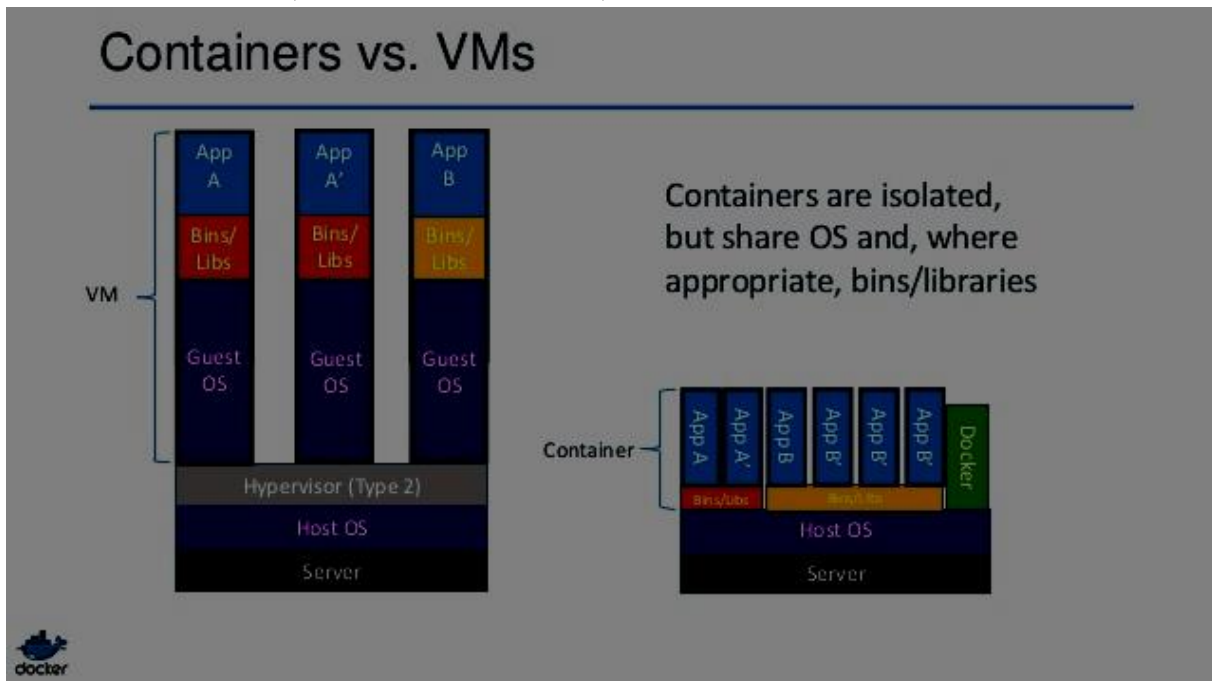


Table des matières

TABLE DES MATIERES	1
PREREQUIS	2
INSTALLATION	3
UTILISATION DE DOCKER	4

Avant-Propos

Docker permet de créer des environnements (appelées containers) de manière à isoler des applications. Docker repose sur le kernel Linux et sur une fonctionnalité : les containers. L'idée est de lancer du code (ou d'exécuter une tâche) dans un environnement isolé.



Liens utiles : -<https://docs.docker.com/engine/installation/linux/ubuntu/linux/>
 -<https://doc.ubuntu-fr.org/docker>
 -<https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-16-04>

Tutoriel complet : - <https://docs.docker.com/>

<https://serversforhackers.com/getting-started-with-docker>

https://docs.docker.com/engine/userguide/networking/default_network/custom-docker0/

<https://docs.docker.com/engine/userguide/networking/>

E6 :

Elaboration de documents relatifs à la production et à la fourniture de services
 A1.2.4 , Détermination des tests nécessaires à la validation d'un service
 A4.1.9 , Rédaction d'une documentation technique
 A5.2.2 , Veille technologique
 A5.2.4 , Étude d'une technologie, d'un composant, d'un outil ou d'une méthode

Prérequis

Docker fonctionne uniquement sur les distributions Linux 64bits et requiers la version 3.10 ou supérieure du Kernel Linux. Pour savoir la version du Kernel, on tape la commande suivante :

```
utilisateur@utilisateur-System-Product-Name:~$ uname -r
```

faut
 4.4.0-57-generic

ensuite mettre à jours les paquets avec un apt-get update

```
utilisateur@utilisateur-System-Product-Name:~$ sudo apt-get update
```

Ensuite on installe des paquets nécessaires au bon fonctionnement de Docker

```
utilisateur@utilisateur-System-Product-Name:~$ sudo apt-get install apt-transport-https ca-certificates
```

Ensuite il faut ajouter la clé GPG. La commande suivante télécharge la clé avec l'ID 58118E89F3A912897C070ADB76221572C52609D depuis le serveur hkp://ha.pool.sks-keyservers.net:80

```
utilisateur@utilisateur-System-Product-Name:~$ sudo apt-key adv \
> --keyserver hkp://ha.pool.sks-keyservers.net:80 \
> --recv-keys 58118E89F3A912897C070ADB76221572C52609D
```

Dans le tableau suivant, on regarde la version de notre Ubuntu, et on marque ce qui lui est associé entre les ‘ ‘ dans la commande

Ubuntu version	Repository
Precise 12.04 (LTS)	deb https://apt.dockerproject.org/repo ubuntu-precise main
Trusty 14.04 (LTS)	deb https://apt.dockerproject.org/repo ubuntu-trusty main
Wily 15.10	deb https://apt.dockerproject.org/repo ubuntu-wily main
Xenial 16.04 (LTS)	deb https://apt.dockerproject.org/repo ubuntu-xenial main

```
utilisateur@utilisateur-System-Product-Name:~$ echo "deb https://apt.dockerproject.org/repo ubuntu-xenial main" | sudo tee /etc/apt/sources.list.d/docker.list
deb https://apt.dockerproject.org/repo ubuntu-xenial main
```

Une fois cela fait, on fait un apt-get update :

```
utilisateur@utilisateur-System-Product-Name:~$ sudo apt-get update
```

Ensuite on effectue la commande suivante afin de savoir si APT arrive à récupérer les versions de Dockers. La version qui est installée est marquée avec *** devant, mais pour le moment Docker n'est pas installé, donc il n'y en a pas.

```
utilisateur@utilisateur-System-Product-Name:~$ apt-cache policy docker-engine
docker-engine:
  Installé : (aucun)
  Candidat : 1.12.5-0-ubuntu-xenial
  Table de version :
    1.12.5-0-ubuntu-xenial 500
      500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
    1.12.4-0-ubuntu-xenial 500
      500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
    1.12.3-0~xenial 500
      500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
    1.12.2-0~xenial 500
      500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
    1.12.1-0~xenial 500
      500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
    1.12.0-0~xenial 500
      500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
    1.11.2-0~xenial 500
      500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
    1.11.1-0~xenial 500
      500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
    1.11.0-0~xenial 500
      500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
```

Pour les versions 16.04 (Xenial), 15.10 (Wily), 14.04 (Trusty), il faut installer des paquets qui sont nécessaires au bon fonctionnement de Docker. Ils permettent d'utiliser le pilote de stockage aufs.

```
utilisateur@utilisateur-System-Product-Name:~$ sudo apt-get install linux-image-extra-$(uname -r) linux-image-extra-virtual
```

Une fois que tous ces prérequis sont correctement installés, nous pouvons passer à l'installation de Docker.

Installation

L'installation de Docker est rapide et facile à mettre en œuvre.

Pour commencer, on fait un apt-get update, pour être sûr que tout est correctement à jours.

```
utilisateur@utilisateur-System-Product-Name:~$ sudo apt-get update
```

Une fois cela fait, on installe Docker grâce à la commande suivante :

```
utilisateur@utilisateur-System-Product-Name:~$ sudo apt-get install docker-engine
```

Quand l'installation est terminée, on lance les démons de Docker :

```
utilisateur@utilisateur-System-Product-Name:~$ sudo service docker start
```

Pour vérifier que Dockers est correctement installé et fonctionne, on lance une image test nommée hello-world. Docker va alors la télécharger et l'afficher. Si il arrive à la télécharger et à l'afficher, alors Docker est correctement installé et fonctionnel :

```
utilisateur@utilisateur-System-Product-Name:~$ sudo docker run hello-world
```

```
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.
 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 Hub account:
https://hub.docker.com

For more examples and ideas, visit:
https://docs.docker.com/engine/userguide/
```

Maintenant que cela fonctionne, nous pouvons profiter des multiples fonctionnalités de Docker !

Utilisation de Docker

Pour utiliser les conteneurs Docker, il faut utiliser des images Docker. Les images Docker sont stockées depuis le Docker Hub, une sorte de base de donnée d'image déjà installées et configurée prêtes à l'emploi. On peut y accéder soit directement en ligne de commande en effectuant la recherche que l'on souhaite, soit à partir du site internet <https://hub.docker.com/> . Il faut néanmoins se créer un compte sur le site afin de pouvoir y accéder via navigateur.

Pour faire une recherche en ligne de commande, il faut utiliser la commande suivante, dans notre cas nous allons chercher une distribution Debian :

```
utilisateur@utilisateur-System-Product-Name:~$ sudo docker search debian
NAME                DESCRIPTION                               5
TARS                OFFICIAL    AUTOMATED
debian              Debian is a Linux distribution that's comp... 1
859                [OK]
neurodebian         NeuroDebian provides neuroscience research... 3
1                  [OK]
jesselang/debian-vagrant Stock Debian Images made Vagrant-friendly ... 9
[OK]
armbuild/debian     ARMHF port of debian                       8
[OK]
eboraas/debian     Debian base images, for all currently-avai... 6
[OK]
samueldebruyn/debian-git a minimal docker container with debian and... 6
[OK]
```

Les résultats de la recherche sur la capture d'écran ne sont qu'une toute petite partie de tous les résultats disponibles.

Pour télécharger l'image, il suffit de faire `docker pull` suivit du nom du paquet. Par exemple, si nous voulons télécharger Debian, il faut marquer :

/!\ il ne faut pas oublier de marquer `sudo` avant chaque commande /!\

```
utilisateur@utilisateur-System-Product-Name:~$ docker pull debian
```

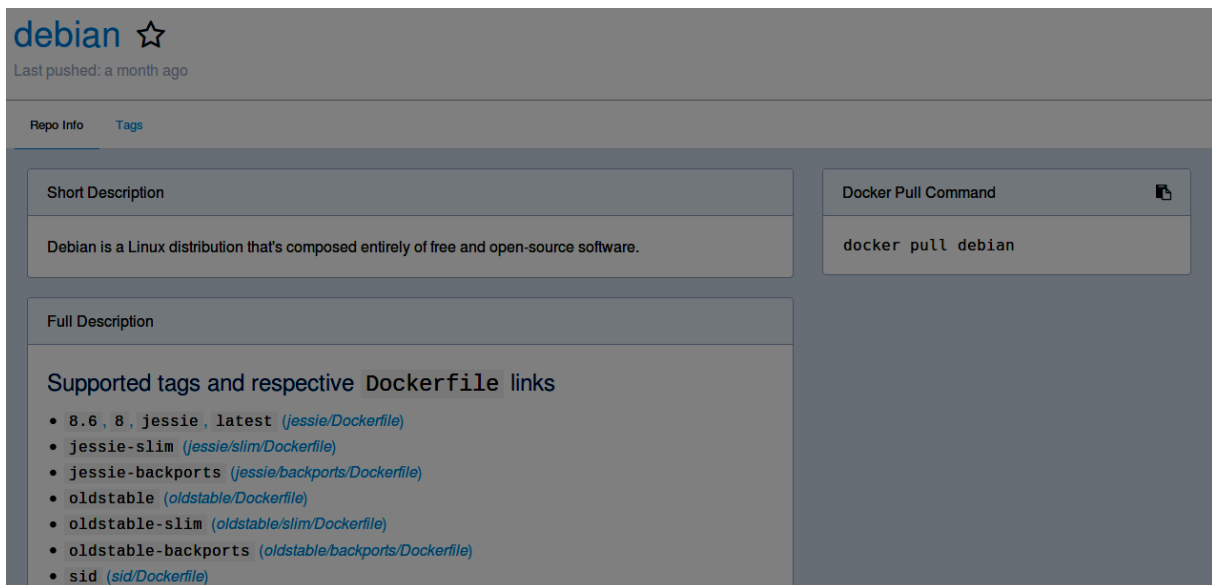
Mais si l'on veut télécharger la `armbuild/debian`, il faut alors marquer :

```
utilisateur@utilisateur-System-Product-Name:~$ docker pull armbuild/debian
```

Si l'on veut passer par le site internet, on fait alors la même recherche :

 debian official	1.9K STARS	10M+ PULLS	> DETAILS
 eboraas/debian public automated build	6 STARS	50K+ PULLS	> DETAILS
 reinblau/debian public automated build	2 STARS	3.2K PULLS	> DETAILS
 konstruktoid/debian public automated build	0 STARS	3.7K PULLS	> DETAILS

Si l'on veut télécharger la Debian de base, on clique alors sur détail, et on arrive sur la fiche de l'image.



debian ☆
Last pushed: a month ago

Repo Info Tags

Short Description

Debian is a Linux distribution that's composed entirely of free and open-source software.

Docker Pull Command

```
docker pull debian
```

Full Description

Supported tags and respective Dockerfile links

- 8.6, 8, jessie, latest ([jessie/Dockerfile](#))
- jessie-slim ([jessie/slim/Dockerfile](#))
- jessie-backports ([jessie/backports/Dockerfile](#))
- oldstable ([oldstable/Dockerfile](#))
- oldstable-slim ([oldstable/slim/Dockerfile](#))
- oldstable-backports ([oldstable/backports/Dockerfile](#))
- sid ([sid/Dockerfile](#))

On peut voir alors qu'il y a pleins d'information complémentaire, ainsi que la commande nécessaire afin de récupérer l'image.

Pour continuer, nous allons donc installer une Debian de base.

```

utilisateur@utilisateur-System-Product-Name:~$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
debian               latest             19134a8202e7       3 weeks ago
123.1 MB
hello-world         latest             c54a2cc56cbb       6 months ago
1.848 kB

```

Pour voir la liste des images téléchargées, on utilise la commande suivante :

```

utilisateur@utilisateur-System-Product-Name:~/nydockerbuild$ sudo docker images

```

Maintenant que l'on a notre image, nous pouvons la lancer dans un conteneur.

Pour lancer une image, on utilise la commande suivante :

```

utilisateur@utilisateur-System-Product-Name:~$ sudo docker run -it debian

```

Le -it permet d'avoir accès au shell en interactif, pour que l'on puisse l'utiliser et taper de commandes afin de s'en servir comme une vraie machine Linux.

La machine que l'on a installée est vierge, il faut donc tout installer dessus.

Déjà nous allons faire les mises-à-jours :

```

root@fb08f862f275:/# apt-get update

```

Puis une fois cela fait on installe nano :

```

root@fb08f862f275:/# apt-get install nano

```

On voit donc que la machine est indépendante de notre Ubuntu, et qu'elle fonctionne comme une machine virtuelle. Pour quitter la machine, on fait exit dans le terminal.

CEPENDANT le conteneur n'est pas sauvegarder, il ne faut pas oublier de le faire à chaque fois que l'on quitte un conteneur, sinon l'on perd tout ce que l'on a fait !

On quitte donc la debian avec la commande exit, puis, une fois que l'on est de retour sur notre Ubuntu, on tape la commande suivante pour savoir l'id du conteneur dans lequel est l'image que l'on veut sauvegarder :

```

utilisateur@utilisateur-System-Product-Name:~$ sudo docker ps -l
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS             PORTS              NAMES
5f7db4ed1239       apache2            "/bin/bash"        9 minutes ago
Up 3 minutes              sleepy_golick

```

Maintenant que l'on a l'ID du conteneur, on peut faire la commande suivante afin de sauvegarder. La sauvegarde recréera une nouvelle image, qu'il faudra lancer si l'on veut refaire des modifications :

```

utilisateur@utilisateur-System-Product-Name:~$ sudo docker commit 5f7db4ed1239 d
ebian/nano
sha256:e21cdf4c4610252a1619ee459f821d4079aeba478784b57d917dc94e11d513f9

```

Maintenant, quand on fait la commande pour lister les images qui sont présentes sur la machine :

```
utilisateur@utilisateur-System-Product-Name:~$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
debian/nano         latest             e21cdf4c4610       37 seconds ago
193.2 MB
```

On peut voir que notre image a bien été enregistrée. Si l'on veut, on peut même la déposer sur le Docker Hub afin de le partager, ou tout simplement le sauvegarder.

Pour lancer la machine, on lance alors la commande habituelle, mais en mettant le nom de l'image que l'on veut lancer :

```
utilisateur@utilisateur-System-Product-Name:~$ sudo docker run -it debian/nano
root@73f1807f5837:/# nano
```

Il ne faut pas oublier que dès que l'on effectue la commande docker run, un nouveau conteneur est créé, il faut penser à chaque fois à regarder le nouveau numéro afin d'effectuer une sauvegarde.