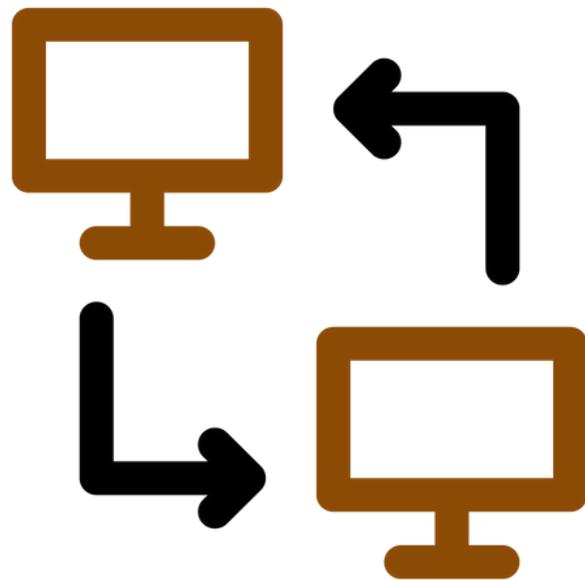


axel brugier

Fiche de procédure



iNFOTECH
SOLUTIONS



SOMMAIRE

- **Présentation de Docker**
 - **Installation de Docker sous Debian**
 - **Les commandes de bases**
 - **Comment créer une image docker**
 - **Automatiser une image docker**
- 

Présentation de docker:

Docker crée le 13 Mars 2013 est une plate-forme logicielle qui vous permet de concevoir, tester et déployer des applications rapidement.

Docker intègre les logiciels dans des unités normalisées appelées conteneurs, qui rassemblent tous les éléments nécessaires à leur fonctionnement, dont les bibliothèques, les outils système, le code et l'environnement d'exécution.

Avec Docker, vous pouvez facilement déployer et dimensionner des applications dans n'importe quel environnement, avec l'assurance que votre code s'exécutera correctement.

Installation de Docker sous Debian11:

Tous d'abord, il faut une machine (virtuelle ou non) avec l'os Debian d'installer. Sur la machine il faut passer en mode "super utilisateur" soit administrateur en tapent la commande "su - " .

Au début, il faut que l'on ajoute la clé GPG officiel de Docker. La clé GPG Docker est une clé cryptographique utilisée pour vérifier l'authenticité des images Docker provenant de registres Docker publics pour cela on utilise les commandes suivantes:

```
curl -fsSL https://download.docker.com/linux/debian/gpg -  
o/etc/apt/keyrings/docker.asc  
chmod a+r /etc/apt/keyrings/docker.asc
```

Par la suite on va ajouter le dépôt Docker dans le fichiers de conf "sources.list" afin de récupérer les paquets docker. On utilise les commandes suivantes:

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

Il faut ensuite installer les paquets Dockers suivants:

```
apt install docker-ce docker-ce-cli containerd.io docker-buildx-  
plugin docker-compose-plugin
```

On va ensuite lancer le conteneur Hello world qui confirmera que notre moteur docker est opérationnel sur notre ordinateur. Pour cela il suffit de taper la commande suivante:

```
docker run hello-world
```

Si votre installation s'est correctement déroulée, vous devriez lire un message tel que :

```
Hello from Docker!
```

```
This message shows that your installation appears to be working correctly.
```

La suite du message affiché explique en quelques lignes le fonctionnement de cette commande ainsi que, très basiquement, le fonctionnement de Docker.

Les commandes de base Docker:

Pour récupérer une image d'un service on utilise la commande:

Docker pull [nom du service]

Pour lancez un conteneur basé sur l'image du service, avec un partage du port interne 80 du conteneur sur le port externe 8080 de la machine on utilise :

Docker run -p 8080:80 [nom du service]

Pour accéder au service depuis le navigateur il suffit de d'aller sur un navigateur est de taper L'URL suivant:

http://[IP_VM]:8080

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Voici la page d'accueillie avec le service Nginx

Il est préférable de voir le statues de nos conteneur ainsi que leur ID avec les commandes suivantes:

Pour lister seulement les conteneurs actifs: **docker ps**

Pour lister tous les conteneurs: **docker ps -a**

Pour arrêter un conteneur il suffit de faire:

docker start <ID Container> => Démarrer le container

Puis pour démarrer le conteneur :

docker stop <ID Container> => Arrêter le container

Pour supprimer une image d'un service on utilise la commande suivante:

Docker rmi <ID image>

Pour supprimer un conteneur totalement on utilise:

Création d'une image docker

Il faut tous d'abord créer un fichier qu'on surnomme "dockerfile" dans lequel on va mettre

FROM debian:latest

CMD echo "Hello World!"

Puis on va construire l'image docker grace à la commande suivante:

docker build -t my-hello-world .

Attention: ne pas oublier l'espace et le point

par la suite on va créer un fichier index.html pour un service web comme nginx ou Apache. Dans ce fichier on notera :

echo "<h1>Hello from my custom page!</h1>" > index.html

Pour créer une image personnalisé incluant un fichier HTML il suffit de créer notre dockerfile :

FROM nginx:latest

COPY index.html /usr/share/nginx/html/index.html

Pour notre exemple nous avons pris le service Nginx

Puis il suffit de construire l'image grace à la commande suivantes:

docker build -t my-nginx .

il nous reste plus qu'à lancer notre conteneur:

```
docker run -d -p 8080:80 --name my-nginx my-nginx
```

Pour voir notre service Nginx il suffit d'aller sur mozilla et taper ceci:

```
http://[IP_Machine]:8080
```

Automatiser une image docker

Tous d'abord il faut créer un fichier "docker-compose.yml" avec dedans les informations suivantes:

```
1 version: '3'
2 services:
3   db:
4     image: mysql:5.7
5     volumes:
6       - db_data:/var/lib/mysql
7     restart: always
8     environment:
9       MYSQL_ROOT_PASSWORD: somewordpress
10      MYSQL_DATABASE: wordpress
11      MYSQL_USER: wordpress
12      MYSQL_PASSWORD: wordpress
13
14   wordpress:
15     depends_on:
16       - db
17     image: wordpress:latest
18     ports:
19       - "8000:80"
20     restart: always
21     environment:
22       WORDPRESS_DB_HOST: db:3306
23       WORDPRESS_DB_USER: wordpress
24       WORDPRESS_DB_PASSWORD: wordpress
25       WORDPRESS_DB_NAME: wordpress
26
27 volumes:
28   db_data: {}
```

Dans notre exemple nous utilisons Wordpress

Wordpress est un système de gestion de contenu gratuit, libre et open-source.

Ce logiciel écrit en PHP repose sur une base de données MySQL et MariaDB,

et est distribué par la fondation WordPress.org.

Pour lancer en arrière plan le conteneur on utilise : **docker-compose up -d**

Pour lancer au premier plan on utilise: **docker-compose up**

Pour vérifier les logs on utilise: **docker-compose log -f**

Pour lancer le projet il suffit de de taper la commande suivante:

docker-compose up -d

Tableaux des commandes de toute la fiche

Commandes	Signification
Docker pull [nom de service]	Permet de récupérer l'image d'un service
Docker ps -a	Récupérez la liste des conteneurs créés sur votre installation Docker. Notez le nom et l' <u>ID</u> du conteneur du service
Docker ps	Liste seulement les conteneurs actif
Docker <u>start</u> [ID Container]	Permet de démarrer le conteneur
Docker stop [ID Container]	Permet de stopper le conteneur
Docker <u>rm</u> -f [ID Container]	Permet de supprimer un conteneur
Docker <u>rmi</u> [ID image]	Supprime l'image d'un service
Docker <u>run</u> -p -d 8080	Permet de lancer le conteneur avec le service <u>nginx</u> au port 80
<u>touch</u>	Créer un fichier sous <u>debian</u>
<u>nano</u>	Pour modifier un fichier sous <u>debian</u>
Docker logs -f [<u>container-name</u>]	Permet de récupérer les logs du conteneur
Docker-compose <u>up</u>	Permet de lancer le conteneur en premier plan
Docker-compose <u>up</u> -d	Permet de lancer le conteneur en arrière plan
Docker build	Permet de construire une image