

Procédure Docker

Prérequis

Debian 9.5 ou supérieur
Connexion Internet

Installation

Nous reprenons ici les commandes trouvées sur [la documentation de Docker](https://docs.docker.com/install/linux/docker-ce/debian/).
<https://docs.docker.com/install/linux/docker-ce/debian/>

Puis utilisez les commandes suivantes:

```
sudo apt purge docker docker-engine docker.io -y
sudo apt update
sudo apt install -y apt-transport-https ca-certificates curl gnupg2 software-properties-common
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian
$(lsb_release -cs) stable"
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add
sudo apt-key fingerprint 0EBFCD88
sudo apt update && sudo apt install -y docker-ce
```

Première Utilisation

Après l'installation de Docker nous allons vérifier son fonctionnement en utilisant le conteneur Hello World.

```
$ sudo docker run hello-world
```

Nous allons maintenant installer et lancer le conteneur mongoDB et ouvrir le port par défaut pour y accéder.

```
$ sudo docker run -d --name mongodb -p 27017:27017 mongo
```

Puis pour vérifier que le conteneur soit bien démarré :

```
$ sudo docker ps
```

Vous aurez accès à un serveur mongodb sur le port 27017 de l'hôte debian.
Pour y accéder depuis un debian , installez mongodb-clients

```
$ sudo apt install -y mongodb-clients
```

```
$ mongo ipduserveur:port
```

Création de votre propre image docker

Pour créer votre propre image Docker, il faut déjà comprendre le principe du Dockerfile. Le Dockerfile peut être vu comme un livre de recettes. On se base sur une image existante, par exemple celle de Debian, sur laquelle nous allons installer le paquet cowsay et enfin paramétrer que lorsque le container est lancé la commande "cowsay Félicitations" est exécutée.

Le dockerfile possède une structure spéciale.

Vous pouvez inclure les instructions suivantes dans le Dockerfile :

- **FROM** – (Obligatoire en tant que première instruction du fichier) Spécifie l'image de base à partir de laquelle le conteneur Docker est créé.

L'image peut être hébergée dans un référentiel public, un référentiel privé hébergé par un registre tiers ou un référentiel que vous exécutez sur EC2.

- **EXPOSE** – (Obligatoire) Répertoire les ports à exposer sur le conteneur Docker.

Vous pouvez spécifier plusieurs ports de conteneurs

- **CMD** – Spécifie un paramètre exécutable et des paramètres par défaut, qui sont combinés dans une commande que le conteneur exécute lors du lancement. Utilisez le format suivant :

CMD ["executable","param1","param2"]

CMD permet également de fournir des paramètres par défaut pour une commande ENTRYPOINT en omettant l'argument exécutable.

Un exécutable doit être spécifié dans une instruction CMD ou ENTRYPOINT, mais pas dans les deux. Pour les scénarios de base, utilisez une instruction CMD et omettez ENTRYPOINT.

- **ENTRYPOINT** – Utilise le même format JSON que CMD et, comme CMD, spécifie une commande à exécuter lors du lancement du conteneur. Permet également à un conteneur d'être exécuté en tant qu'exécutable avec docker run.

Si vous définissez une instruction ENTRYPOINT, vous pouvez également utiliser une instruction CMD pour spécifier les paramètres par défaut qui peuvent être remplacés par l'option -d de docker run. La commande définie par une instruction ENTRYPOINT (incluant tous les paramètres) est combinée avec les paramètres de CMD bou docker run lorsque le conteneur est exécuté.

- **RUN** – Spécifie une ou plusieurs des commandes qui installent les packages et configurent votre application web dans l'image.

Si vous incluez des instructions RUN dans le Dockerfile, compressez le fichier et le contexte utilisés par les instructions RUN dans le Dockerfile sous la forme d'un fichier .zip.

Compressez les fichiers au niveau supérieur du répertoire.

Vous pouvez trouver simplement des exemples de Dockerfile sur Internet.

Commandes utiles

```
$ man docker
$ sudo docker ps (-a)
$ sudo docker run (-i -t -d) --name name image
$ sudo docker stop/start ID/NAME
$ sudo docker rm ID/NAME
$ sudo docker pull
$ sudo docker images
$ sudo docker image rm
```

D'autres documentations

<https://docs.docker.com/>

<https://docs.docker.com/get-started/>