Doc Job03

---
created : 24-06-02 21:28
modified : 24-06-02 21:28
tags : [note]
References :

Nous allons créer ce qu'on appelle un dockerfile.

Un Dockerfile est un fichier texte utilisé pour créer une image Docker. Il contient une série d'instructions que Docker utilise pour assembler une image. Voici les éléments de base et un exemple pour mieux comprendre.

> ⓘ **Dockerfile commandes :**
>
> - **FROM** permet de définir depuis quelle base votre image ve être créée
> - **MAINTAINER** permet de définir l'auteur de l'image et s'écrit de la manière suivante `Nom <email>`
> - **RUN** permet de lancer une commande, mais aura aussi pour effet de créer une image intermédiaire.
> - **ADD** permet de copier un fichier depuis la machine hôte ou depuis une URL
> - **EXPOSE** permet d'exposer un port du container vers l'extérieur
> - **CMD** détermine la commande qui sera exécutée lorsque le container démarrera

- **ENTRYPOINT** permet d'ajouter une commande qui sera exécutée par défaut, et ce, même si on choisit d'exécuter une commande différente de la commande standard
- **WORKDIR** permet de définir le dossier de travail pour toutes les autres commandes (par exemple RUN, CMD, ENTRYPOINT et ADD)
- **ENV** permet de définir des variables d'environnements qui pourront ensuite être modifiées grâce au paramètre de la commande `run --env <key>=<value>`.
- **VOLUMES** permet de créer un point de montage qui permettra de persister les données. On pourra alors choisir de monter ce volume dans un dossier spécifique en utilisant la commande `run -v <chemin hôte>:

Je créé mon fichier dockerfile (fichier texte qui va encadrer mon container):

```
nano dockerfile
```

puis:

```
FROM debian:latest

LABEL maintainer="user"

RUN apt-get update && apt-get install -y

ENTRYPOINT ["echo", "hello world"]
```

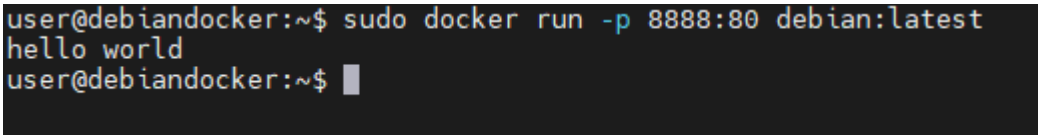
Je build ensuite mon image pour pouvoir être en possibilité de lancer le conteneur :

```
sudo docker build --tag "debian:latest" .
```

Une fois l'image build, je lance mon conteneur (qui ne contient qu'une seule commande à l'intérieur):

```
sudo docker run -p 8888:80 debian:latest
```

On voit bien que la commande s'exécute :



On peut même afficher la liste des images build avec la commande `sudo docker image list` :