

TP : Dockerfile

Exemple 1:

Créer une image affichant « Hello World »

Créez un fichier texte nommé Dockerfile, dans lequel on va mettre le texte suivant :

FROM busybox

ENTRYPOINT ["echo"]

CMD ["hello world"]

Ensuite dans votre invite de commande exécutez la commande suivante :

docker build -tag my_hello:latest .

docker run my_hello:latest

Exemple 2:

La première chose que vous devez faire est de créer un fichier nommé "Dockerfile", puis de définir dans celui-ci l'image que vous allez utiliser comme base, grâce à l'instruction FROM. Dans notre cas, nous allons utiliser une image de base Debian 9.

FROM debian:9

Ensuite, utilisez l'instruction RUN pour exécuter une commande dans votre conteneur.

**RUN apt-get update -yq **

**&& apt-get install curl gnupg -yq **

**&& curl -sL https://deb.nodesource.com/setup_10.x | bash **

**&& apt-get install nodejs -yq **

&& apt-get clean -y

Puis, utilisez l'instruction ADD afin de copier ou de télécharger des fichiers dans l'image. Dans notre cas, nous l'utilisons pour ajouter les sources de notre application locale dans le dossier /app/ de l'image.

ADD . /app/

Utilisez ensuite l'instruction WORKDIR qui permet de modifier le répertoire courant. La commande est équivalente à une commande cd en ligne de commande. L'ensemble des commandes qui suivront seront toutes exécutées depuis le répertoire défini.

WORKDIR /app



Maintenant que le code source et les dépendances sont bien présents dans votre conteneur, nous devons indiquer à notre image quelques dernières informations.

EXPOSE 2368

VOLUME /app/logs

Vous pouvez maintenant créer votre première image Docker.

L'argument `-t` permet de **donner un nom à votre image** Docker. Cela permet de retrouver plus facilement votre image par la suite.

Le `.` est le répertoire où se trouve le Dockerfile ; dans notre cas, à la racine de notre projet.

docker build -t debian_container .

Pour démarrer votre conteneur il suffit d'exécuter la commande :

docker run -it debian_container /bin/bash