

RÉCAPITULATIF DES PRINCIPALES INSTRUCTIONS DANS UN DOCKERFILE

Docker crée des images automatiquement en lisant les instructions d'un Dockerfile qui est un fichier texte contenant toutes les instructions, dans l'ordre, nécessaires pour créer une image donnée.

Ci-dessous, vous trouverez les principales instructions.

FROM : un fichier Dockerfile commence toujours par la commande FROM, qui permet de définir une image de base au-dessus de laquelle la nouvelle image sera construite.

ENV : permet de définir des variables d'environnement exploitables lors de la compilation du Dockerfile et dans les futurs conteneurs.

ARG définit une variable d'environnement uniquement disponible lors de la construction de l'image.

RUN : permet d'exécuter une ou plusieurs commandes à l'intérieur de l'image.

EXPOSE : indique le(s) port(s) sur lesquels un conteneur écoute les connexions (on utilise en règle générale le port « connu » pour l'application). Pour un accès externe, les utilisateurs devront exécuter « docker run » avec l'option « -p port_externe:port_interne » pour mapper le port ou l'instruction correspondante dans le docker compose.

WORKDIR 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 (utile si l'on doit se placer dans un répertoire particulier pour compiler un programme ou exécuter une application).

COPY permet de copier des fichiers/dossiers depuis l'hôte Docker vers l'image. Ces derniers doivent se trouver dans le contexte de construction c'est-à-dire dans le même dossier que le fichier Dockerfile.

Attention, lorsque l'on copie un dossier, pour que l'arborescence soit maintenue telle quelle il ne faut pas utiliser des métacaractères. Soit, par exemple, l'arborescence :

DOSSIER

```
sous_dossier1/
    fichier1
    fichier2
sous_dossier2/
    fichier3
    fichier4
```

RUN mkdir /chemin_dans_le_conteneur/DOSSIER (instruction obligatoire pour créer le dossier DOSSIER dans le conteneur)

COPY DOSSIER/* /chemin_dans_le_conteneur/DOSSIER/ ne crée pas l'arborescence mais ajoute les fichiers dans /chemin_dans_le_conteneur/DOSSIER/

COPY DOSSIER /chemin_dans_le_conteneur/DOSSIER/ crée l'arborescence.

USER désigne quel utilisateur lancera les instructions qui suivent (utile pour éviter d'utiliser l'utilisateur root si cela n'est pas nécessaire).

CMD définit le comportement par défaut au lancement d'un conteneur alors que tout ce qui précède permet de définir l'image.

S'il y a plusieurs CMD, seule la dernière commande sera exécutée.

La commande et les arguments fournis par CMD peuvent être remplacés depuis la ligne de commande lorsque le conteneur docker est lancé : on parle de « surcharge ». Si par exemple on veut seulement faire un « ls » : *docker run <image> ls* avec la commande « ls » qui remplacera le contenu de CMD.

ENTRYPOINT joue le même rôle que CMD mais ne peut pas être surchargé au « docker run ».

Ce mot clé **spécifie ainsi une commande qui sera toujours exécutée au démarrage du conteneur.**

Et dans ce cas le mot clé CMD (qui peut être utilisé avec ENTRYPOINT) spécifie les arguments qui seront envoyés par défaut à la commande spécifiée par ENTRYPOINT, arguments qui, eux, pourront donc être surchargés lors de l'exécution de « docker run ».