

# S04: Dockerfiles suite

## Recap

---

Ce matin nous avons commencé à travailler avec les Dockerfiles. Un Dockerfile contient une série d'instructions pour

- importer une image de base (FROM)
- installer des packages (RUN apt-get)
- copier des fichiers du host local vers le conteneur (COPY)
- exécuter des commandes (CMD)

Mais ce n'est que la partie immergée des dockerfiles.

Nous avons aussi vu comment lier ou mount un répertoire sur le host à un répertoire dans le container avec le flag `-v` dans la commande `docker run`.

## La suite

---

Dans cette session nous allons aller plus loin dans l'utilisation des Dockerfiles avec

- distinguer parmi les différents types d'images (slim, buster, ...)
- les nuances entre CMD et Entrypoint, ADD et COPY, ARG et ENV
- la gestion des variables d'environnement
- la gestion des mots de passe et clefs API
- les couches d'une image
- l'optimisation en temps de build et en volume d'une image et la notion de caching
- le build multi stage
- améliorer le Dockerfile avec HEALTHCHECK, MAINTAINER, Cleanup, Non-root User

## Pratique

---

Pour cela nous allons construire une application qui

- load un personnage
- saisie une question
- interroge l'API OpenAI (GPT 3.5) pour répondre à la question

La clef de l'API OpenAI est dispo dans le discord. Elle ne sera active que pour notre session de cette après midi.