Créer une image

Pour créer une image nous allons avoir besoin d'un nouveau fichier à la racine de notre projet.

Celui ci doit se nommer "**Dockerfile**" sans extension.

Stricte Minimum

Celui ci devra contenir au moins :

```
FROM php:8.2-apache
```

- **FROM** indique quelle autre image est utilisée comme source.
- **scratch** Si votre image n'a pas besoin d'une image source, vous pouvez utiliser le mot clef "scratch"

Commandes et Copies

```
FROM php:8.2-apache
COPY . /var/www/html
RUN apt update && apt upgrade
```

- **COPY** indique quel fichier ou dossier doit être copié et dans quel dossier de l'image il doit être copié.
- **RUN** permet de lancer des commandes sur le conteneur. Si on souhaite lancer plusieurs commandes, on peut au choix séparer chaque commande d'un "**&&**" ou bien remettre une instruction "**RUN**".

Pour plus de clarté, on peut échapper les sauts à la ligne avec un "****".

Ici on indique que l'on veut utilise l'image "**php:8.2-apache**" et coller dans son dossier "**/var/www/html**" tous les fichiers "**.**" et dossiers présent dans notre projet.

- **ADD** fonctionne comme **COPY** à la différence qu'il supporte en plus les URL.

Expose et Environnement

Il est aussi possible de définir un port ouvert par défaut ainsi que des variables d'environnement.

```
FROM mariadb
EXPOSE 3306
ENV MARIADB_ROOT_PASSWORD=root
```

- **EXPOSE** permet d'indiquer un port à exposer par défaut.
- **ENV** permet d'indiquer des variables d'environnement, si plusieurs sont utilisées, elles peuvent être séparé d'un espace ou alors d'un nouveau **ENV**.

Une fois le docker file créé, pour donner vie à notre image, nous pouvons utiliser la commande :

```
docker build -t nomImage:optionnelTag .
```

- **build** permet de construire l'image.
- **-t** indique que l'on va nommer l'image, on peut ajouter après le nom "**:**" pour ajouter une étiquette (tag).
- **.** le chemin vers le projet (ici "." signifie que nous somme déjà dans le projet)