

# TP 2 : Construction d'une image Docker et création d'un Dockerfile

## TP 2 : Construction d'une image Docker et création d'un Dockerfile

1. Construction d'une image Docker
  - 1.1. Récupération du code source
  - 1.2. Construction d'une image Docker
  - 1.3. Déployer l'image créée
2. Création d'un fichier Dockerfile
  - 2.1. Supprimer le Dockerfile existant
  - 2.2. Créer le fichier Dockerfile
3. (Bonus) Création d'un Dockerfile multi-stages

## 1. Construction d'une image Docker

Au cours de cette formation, nous allons travailler avec le code source d'une application web d'entraînement qui a déjà été développé. Cette application est composée d'un backend et d'un frontend : [gitlab.com/groupomania/express-ts](https://gitlab.com/groupomania/express-ts) et [gitlab.com/groupomania/vue3-ts](https://gitlab.com/groupomania/vue3-ts). Nous nous concentrerons ici principalement sur le backend.

### 1.1. Récupération du code source

Clonez le code source du backend à l'aide de git :

```
1 git clone git@gitlab.com:groupomania/express-ts
2 cd express-ts
```

### 1.2. Construction d'une image Docker

Le projet contient déjà un *Dockerfile*. Nous allons nous en servir pour tester la construction d'une image Docker :

```
1 docker build .
```

À la fin de l'opération, on pourra constater que l'image a bien été créée via la commande `docker image ls`.

On peut également lui attribuer un tag. Pour cela, lancer la commande :

```
1 docker build -t backend .
```

Le tag `backend` devrait se retrouver dans le résultat de la commande `docker image ls`.

## 1.3. Déployer l'image créée

Vous pouvez à présent déployer l'image *backend* via la commande `docker run`. Vous remarquerez qu'elle n'arrive pas à monter parce que le serveur attend une connexion à une base de données.

Supprimez le conteneur et l'image associée.

## 2. Création d'un fichier Dockerfile

Rappel des principales directives constitutives d'un Dockerfile :

- `FROM <image-name:tag>` : désigne la layer précédente sur laquelle se baser pour monter l'image.
- `WORKDIR <abspath>` : définit le point du système de fichier de l'image qui sera son répertoire de travail
- `COPY <hostpath> <imagepath>` : copie des fichiers depuis le système de fichiers de l'hôte vers celui de l'image
- `RUN <cmd>` : lance une commande depuis l'intérieur de l'image
- `CMD [<...cmdargs>]` : définit la commande que le conteneur exécutera à son lancement

### 2.1. Supprimer le Dockerfile existant

Commencez par supprimer le fichier *Dockerfile* existant déjà dans le projet via la commande :

```
1 rm Dockerfile
```

### 2.2. Créer le fichier Dockerfile

Tout d'abord, veillez à prendre connaissance du contenu du fichier `README.md` qui contient l'ensemble des instructions pour lancer le projet.

Ensuite, créez un nouveau fichier *Dockerfile* qui devra monter le backend en mode serveur de développement, sans lancer les migrations de la base de données.

Construisez ensuite l'image, montez un conteneur docker et tentez d'accéder à la documentation de l'API.

## 3. (Bonus) Création d'un Dockerfile multi-stages

Sur la base du même projet, créer une image dont le rôle sera de servir la version statique de la documentation en se basant sur une image nginx.