

# Documentation Docker pour Mountain Journey

---

## Table des matières

Introduction à Docker.....	2
Prérequis.....	2
Utilisation de Docker Hub .....	3
Créer un Compte Docker Hub (Facultatif).....	3
Connecter Docker à Docker Hub .....	3
Créer et Pousser une Image Docker .....	4
Configuration des Dockerfiles .....	4
Dockerfile pour NGINX.....	4
Dockerfile pour MYSQL.....	5
Configuration de Docker Compose.....	6
docker-compose.yml .....	6
Déploiement de l'Application .....	7
Construire l'image Docker .....	7
Démarrer les services Docker.....	7
Accès à l'Application.....	8
Schéma de l'Infrastructure .....	9

## Introduction à Docker

Docker est une plateforme open source qui permet de développer, expédier et exécuter des applications dans des conteneurs. Les conteneurs permettent de packager une application avec toutes ses dépendances afin qu'elle puisse s'exécuter de manière cohérente dans n'importe quel environnement. Docker simplifie le déploiement d'applications, l'isolation des processus et la gestion des environnements.

Cette documentation vous guidera à travers les étapes nécessaires pour configurer et déployer l'application Mountain Journey en utilisant Docker. L'application comprend une page d'accueil avec une carte interactive, des filtres de recherche et une liste d'itinéraires. Vous apprendrez à installer Docker, créer des images Docker, les pousser sur Docker Hub, configurer Docker Compose et déployer votre application.

## Prérequis

Avant de commencer, assurez-vous d'avoir les éléments suivants installés sur votre machine :

- Docker : [Installation de Docker](#)
- Docker [Compose Compose](#)
- Un compte sur Docker Hub : [Inscription à Docker Hub](#)



## Utilisation de Docker Hub

Docker Hub est un registre public pour stocker et partager des images Docker. Vous pouvez y pousser vos images pour les utiliser sur différents environnements.

## Créer un Compte Docker Hub (Facultatif)

1. Allez sur [Docker Hub](#) et cliquez sur "Sign Up".
2. Remplissez le formulaire d'inscription pour créer un compte.

Une fois le compte créé vous devrez créer un repo.

The screenshot shows the Docker Hub interface for a repository named 'duunky/montainjourney'. The page includes a breadcrumb trail at the top: 'duunky / Repositories / montainjourney / General'. On the right, it indicates 'Using 0 of 1 private repositories. [Get more](#)'. The main section is titled 'General' and shows the repository name 'duunky/montainjourney' with a 'Public View' button. Below the name, it says 'Updated 7 days ago' and 'projet final'. A note states 'This repository does not have a category' with an 'INCOMPLETE' status. To the right, under 'Docker commands', it provides the command 'docker push duunky/montainjourney:tagname'. Below this, the 'Tags' section shows two tags: 'BDD-MJ' and 'API-MJ', both pushed 7 days ago. The 'Automated Builds' section on the right explains how to connect GitHub or Bitbucket for automated builds and includes an 'Upgrade' button.

duunky / [Repositories](#) / [montainjourney](#) / [General](#) Using 0 of 1 private repositories. [Get more](#)

**General** Tags Builds Collaborators Webhooks Settings

**duunky/montainjourney**

Updated 7 days ago

projet final

This repository does not have a category INCOMPLETE

**Docker commands** [Public View](#)

To push a new tag to this repository:

```
docker push duunky/montainjourney:tagname
```

**Tags**

This repository contains 2 tag(s).

Tag	OS	Type	Pulled	Pushed
BDD-MJ		Image	20 hours ago	7 days ago
API-MJ		Image	20 hours ago	7 days ago

[See all](#)

**Automated Builds**

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#)

[Upgrade](#)

## Connecter Docker à Docker Hub

1. Ouvrez une invite de commande ou un terminal.
2. Exécutez `docker login` et saisissez vos identifiants Docker Hub.

## Créer et Pousser une Image Docker

1. Construisez l'image Docker avec `docker build -t yourusername/yourimagename` .
2. Poussez l'image sur Docker Hub avec `docker push yourusername/yourimagename` .

Sur votre cmd :

```
495 docker tag nginx duunky/montainjourney:API-MJ
496 docker tag mysql duunky/montainjourney:BDD-MJ
497 docker push duunky/montainjourney:API-MJ
498 docker push duunky/montainjourney:BDD-MJ
```

Ces commandes permettent de taguer vos images Docker locales et de les pousser vers Docker Hub sous des noms spécifiques.

## Configuration des Dockerfiles

Un `Dockerfile` est un fichier texte qui contient toutes les instructions nécessaires pour construire une image Docker.

### Dockerfile pour NGINX

Créez un fichier nommé `Dockerfile` avec le contenu suivant :

```
1 FROM duunky/montainjourney:API-MJ
2 WORKDIR /
3 COPY Front /usr/share/nginx/html
4 RUN mkdir /API
```

## Dockerfile pour MYSQL

Créez un fichier nommé `Dockerfile` avec le contenu suivant :

```
You, il y a 23 heures | 1 author (You)
1 FROM duunky/montainjourney:BDD-MJ
2 WORKDIR /app
3 COPY . /app
4 ADD lancement.sql /docker-entrypoint-initdb.d
5 EXPOSE 3306 You, il y a 23 heures • Implémenter
```

## Configuration de Docker Compose

Docker Compose est un outil permettant de définir et de gérer des applications multi-conteneurs.

### **docker-compose.yml**

Créez un fichier nommé `docker-compose.yml` avec le contenu suivant :

```
1  version: '3.8'
2
3  services:
4    web:
5      build: ./SITE_MJ
6      image: duunky/montainjourney:API-MJ
7      container_name: API-Mountain-Journey
8      ports:
9        - "8080:80"
10     networks:
11       my_network:
12         ipv4_address: 172.16.238.4
13
14     db:
15       build: ./BDD_MJ
16       image: duunky/montainjourney:BDD-MJ
17       container_name: BDD-Mountain-Journey
18       ports:
19         - "3306:3306"
20       restart: always
21       environment:
22         MYSQL_ROOT_PASSWORD: root
23         MYSQL_DATABASE: MountainJourney
24       networks:
25         my_network:
26           ipv4_address: 172.16.238.3
27
28     networks:
29       my_network:
30         driver: bridge
31         ipam:
32           config:
33             - subnet: 172.16.238.0/24
34
```

## Déploiement de l'Application

Pour déployer l'application, suivez les étapes ci-dessous :

### Construire l'image Docker

Exécutez la commande suivante pour construire l'image Docker :

docker-compose build

```
> docker-compose build
```

### Démarrer les services Docker

Exécutez la commande suivante pour démarrer les services Docker :

docker-compose up

```
> docker-compose up
```

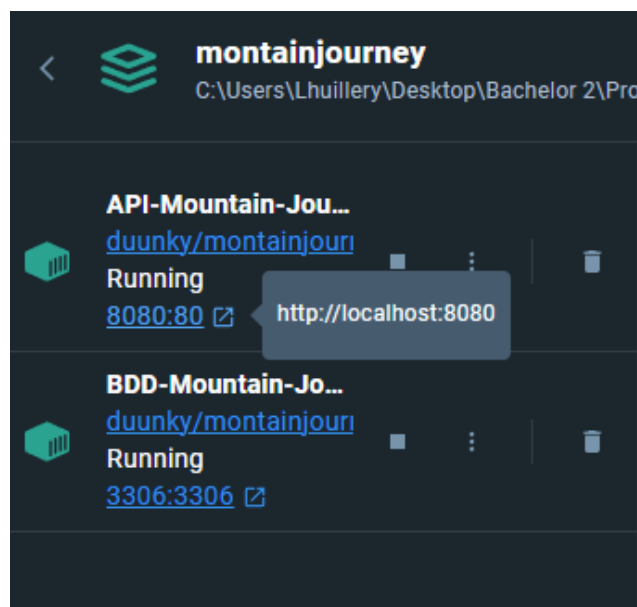
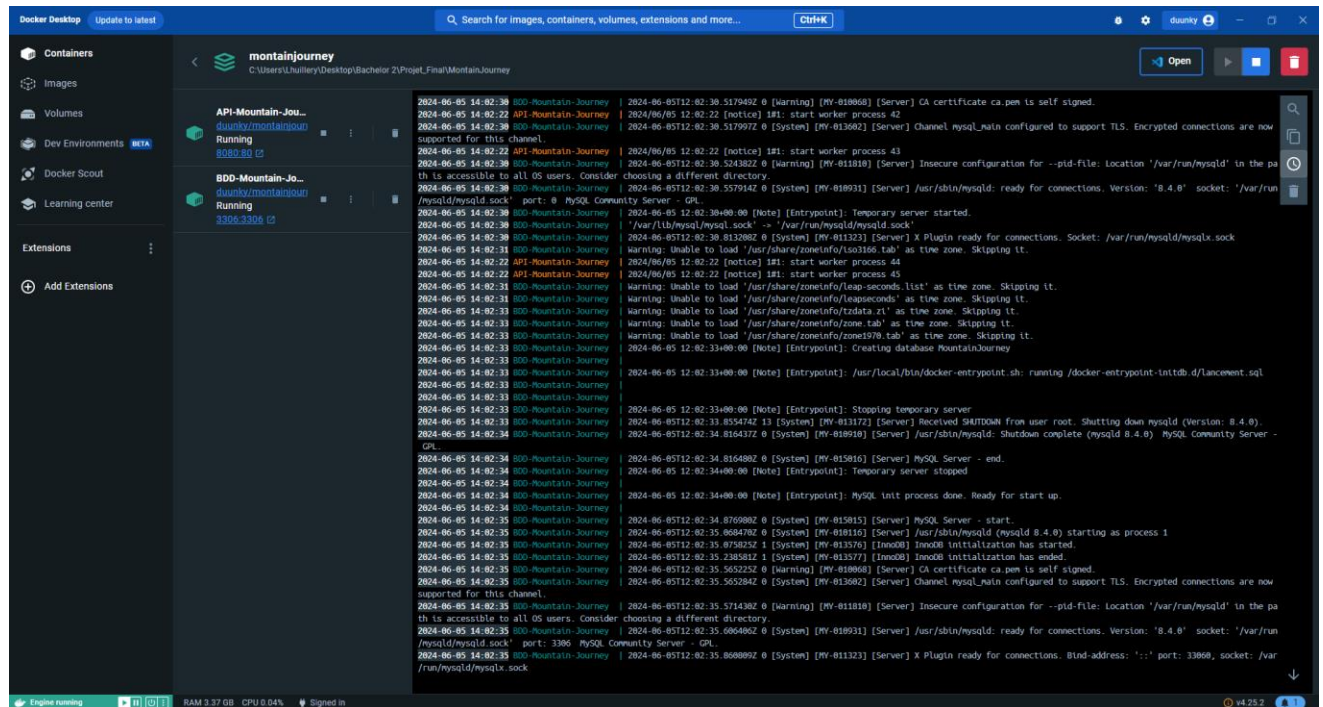
Ou faire les deux en même temps :

Docker-compose up --build

```
> docker-compose up --build
```

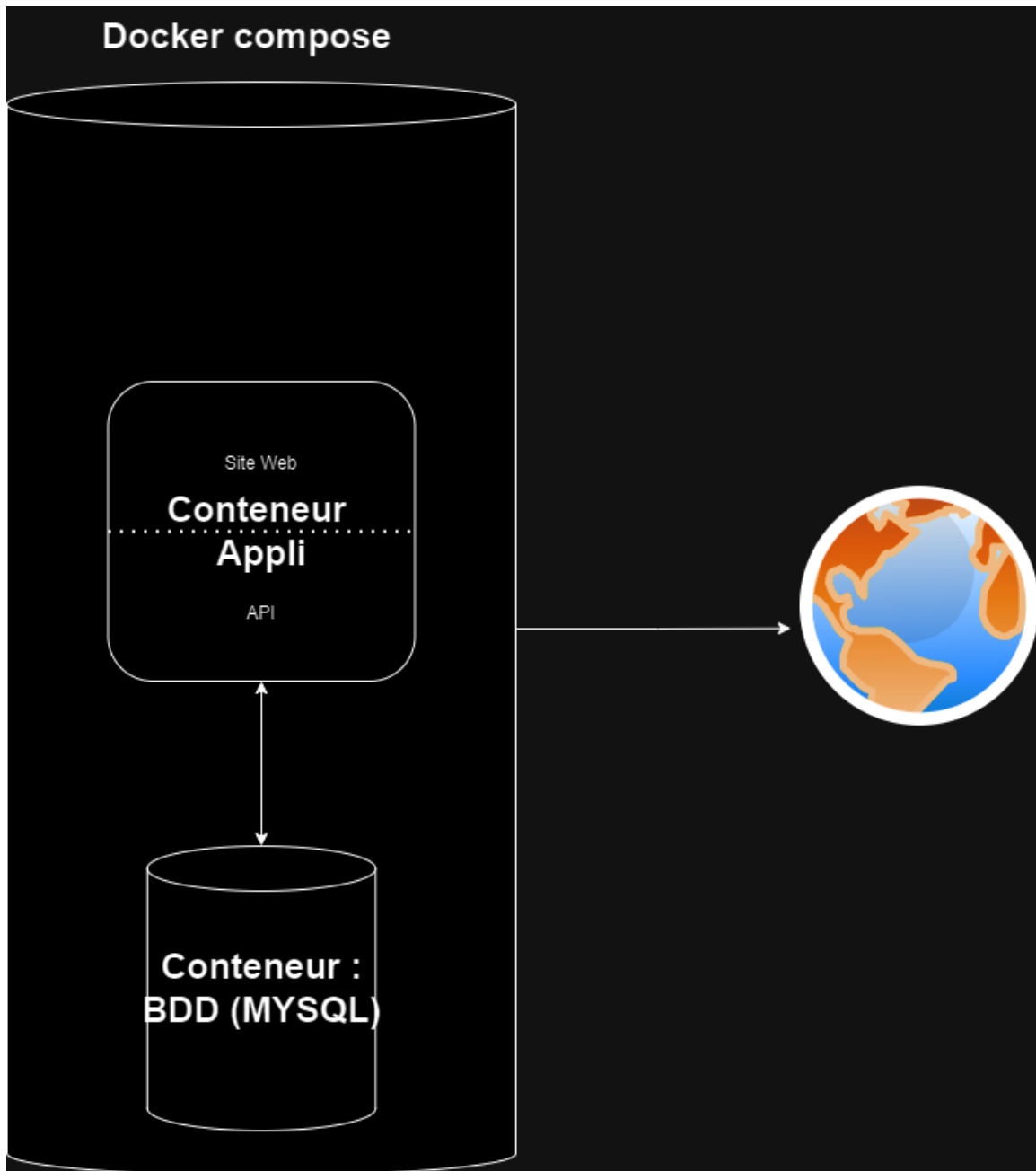
## Accès à l'Application

Une fois les services Docker démarrés, accédez à l'application en ouvrant votre navigateur et en allant à l'adresse suivante :





## Schéma de l'Infrastructure



L'infrastructure Docker pour l'application Mountain Journey se compose d'un conteneur NGINX qui sert les fichiers de l'application et d'un conteneur Mysql servant de BDD.