

# Compte Rendu de la Partie 1 : Conteneurisation d'une Application Web

## Objectif

L'objectif de cette partie était de conteneuriser une application web Node.js en utilisant Docker, depuis le fork du dépôt initial jusqu'à la création et au test de l'image Docker.

## Étapes et Commandes Suivies

### Étape 0 : Récupération du Code Source et Préparation

#### 1. Fork du Dépôt

- La première étape a consisté à réaliser un fork du dépôt source pour obtenir une copie personnelle et modifiable.
- Lien du dépôt source : <https://github.com/choucroute-volante/hetic-infra-2>.
- Nous avons cliqué sur **Fork** pour créer une copie dans notre propre compte GitHub.

#### 2. Cloner le Dépôt Forké

- Ensuite, nous avons cloné le dépôt forké pour travailler en local :
- Commande : `git clone https://github.com/YanisZedira/hetic-infra-2.git`
- Accès au dossier du projet : `cd hetic-infra-2`

#### 3. Création du Fichier `binome.md`

- Dans le dossier racine, nous avons créé un fichier `binome.md` pour documenter les noms et prénoms des membres de l'équipe.
- Voici le contenu du fichier `binome.md` :  
Yanis Zedira et Aymen Djerad (je vais déployer sur GitHub cet après-midi) DIA2
- Ensuite, nous avons ajouté, commité, et poussé ce fichier vers le dépôt GitHub :

- git add binome.md
- git commit -m "Ajout du fichier binome.md avec les noms des membres de l'équipe"
- git push origin master

## Étape 1 : Conteneurisation de l'Application Web

### 1.1 Vérification de l'Application en Local

Avant de conteneuriser l'application, nous avons vérifié son bon fonctionnement en local.

#### 1. Installation des Dépendances

- Nous nous sommes rendus dans le dossier `app` et avons installé les dépendances en utilisant npm.
- Commande : `cd app`, puis `npm install`

#### 2. Lancement de l'Application

- Nous avons ensuite lancé l'application pour nous assurer qu'elle fonctionnait correctement en local.
- Commande : `node app.js`
- Nous avons vérifié le rendu dans le navigateur à l'adresse `http://localhost:3000`.

### 1.2 Création du Dockerfile

Une fois l'application testée en local, nous avons créé un fichier `Dockerfile` dans le dossier racine ( `hetic-infra-2` ) pour conteneuriser l'application. Voici le contenu du Dockerfile :

```
FROM node:16.20.1-alpine3.17
LABEL maintainer="myanis.zedira@gmail.com"
ENV NODE_ENV=production
WORKDIR /app
COPY app/package*.json ./
RUN npm install
COPY app /app
```

```
RUN adduser -D appuser && chown -R appuser /app
```

```
USER appuser
```

```
EXPOSE 3000
```

```
CMD ["node", "app.js"]
```

## 1.3 Construction et Test de l'Image Docker

### 1. Construction de l'Image Docker

- Depuis le dossier `hetic-infra-2`, nous avons construit l'image Docker avec la commande suivante : `docker build -t image-infra-app`.

### 2. Test de l'Image Docker

- Nous avons ensuite lancé le conteneur pour nous assurer que l'application fonctionne correctement dans le conteneur : `docker run -p 3000:3000 image-infra-app`
- L'application était accessible sur `http://localhost:3000`, confirmant que l'image fonctionnait comme attendu.

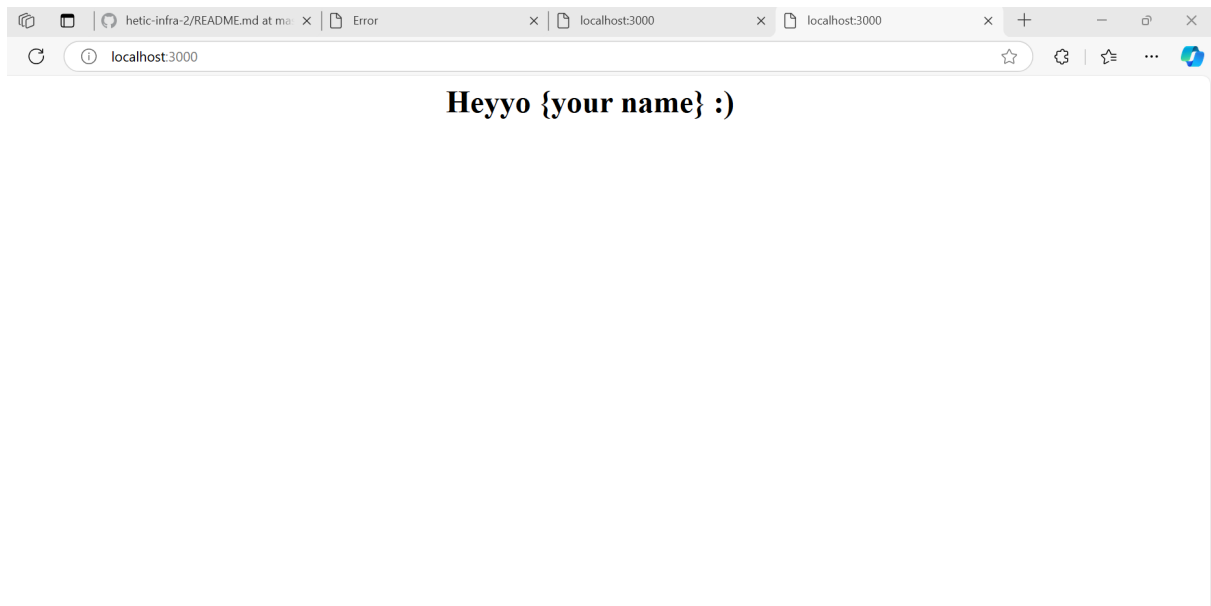
---

## Conclusion

Cette première partie a permis de :

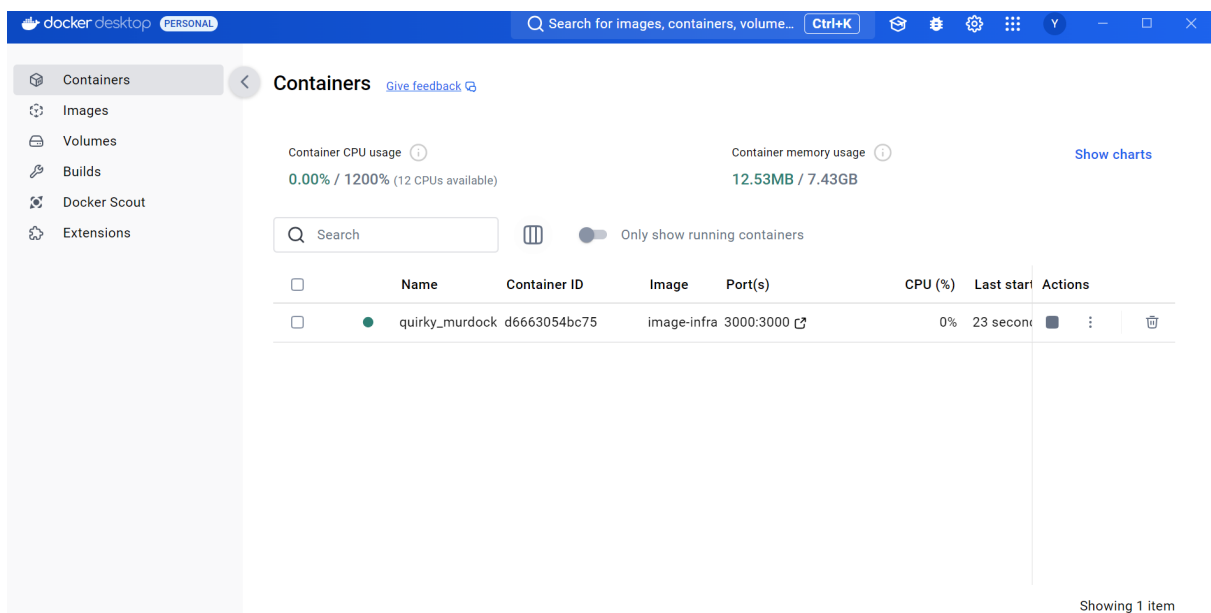
- Forker et cloner le dépôt GitHub initial,
- Tester l'application en local,
- Créer un `Dockerfile` respectant les bonnes pratiques de conteneurisation,
- Construire et tester l'image Docker avec succès.

En suivant ces étapes, nous avons pu conteneuriser l'application de manière simple et efficace. Nous allons maintenant ajouter des captures d'écran pour illustrer chaque étape du processus.

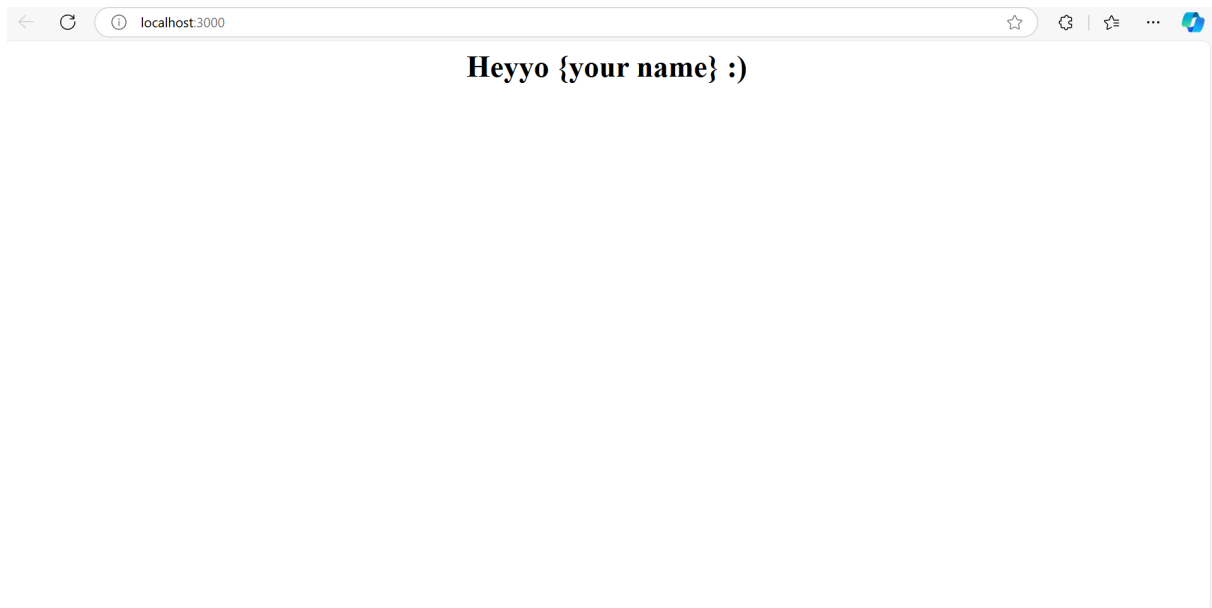


App screen

SCREEN PREUVES :



docker desktop screen



docker image screen