

# DOCUMENTATION

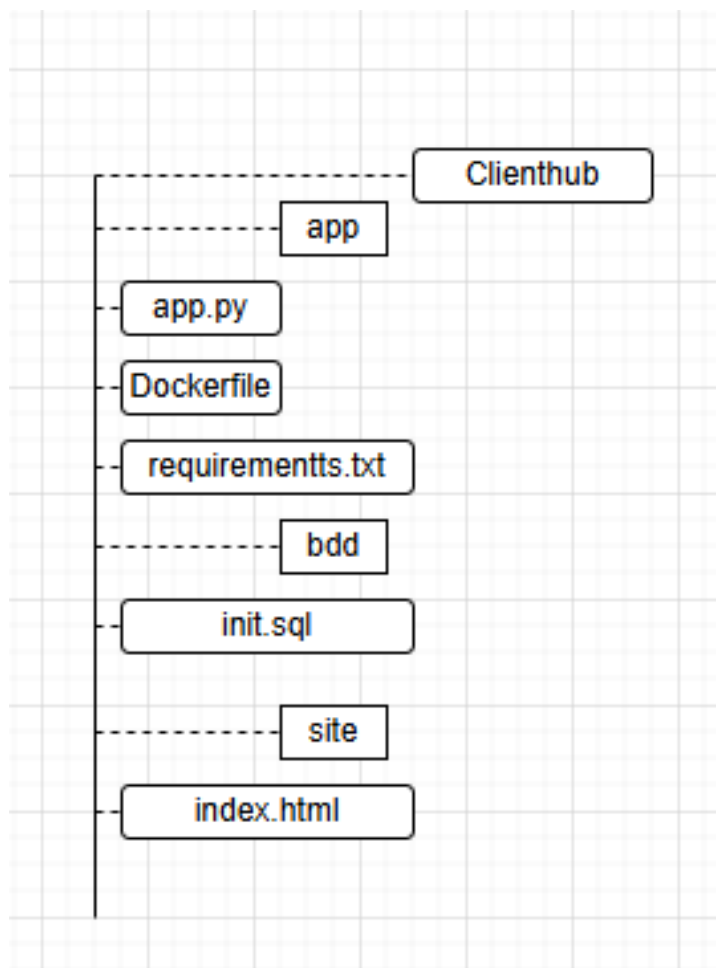
Docker file :

Ouvrir 

• Dockerfile  
~/clienthub/app

```
FROM python:3.9-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
EXPOSE 5000
CMD ["python", "app.py"]
```

Structure :



```

version: '3.8'

services:
  db:
    image: mysql:8.0
    container_name: clienthub-db
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
      MYSQL_DATABASE: clienthub_db
      MYSQL_USER: user
      MYSQL_PASSWORD: password
    volumes:
      - mysql_data:/var/lib/mysql
      - ./bdd/init.sql:/docker-entrypoint-initdb.d/init.sql
    networks:
      - clienthub-net
    healthcheck:
      test: ["CMD", "mysqladmin" ,"ping", "-h", "localhost"]
      timeout: 20s
      retries: 10

  api:
    build: ./app
    container_name: clienthub-api
    restart: always
    ports:
      - "5000:5000"
    environment:
      DB_HOST: db
      DB_NAME: clienthub_db
      DB_USER: user
      DB_PASSWORD: password
    depends_on:
      db:
        condition: service_healthy
    networks:
      - clienthub-net

  web:
    image: nginx:latest
    container_name: clienthub-web
    restart: always
    ports:
      - "8080:80"
    volumes:
      - ./site:/usr/share/nginx/html
    networks:
      - clienthub-net

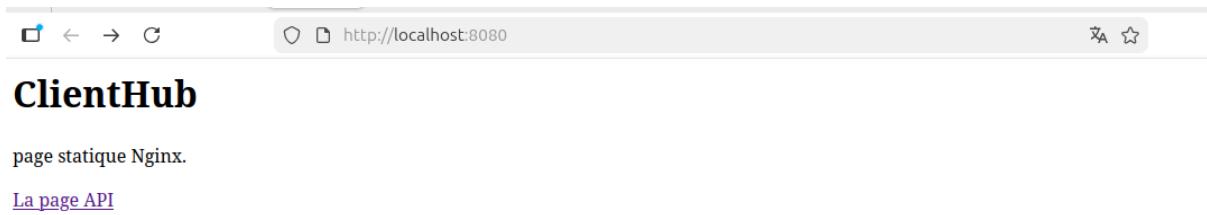
volumes:
  mysql_data:

networks:
  clienthub-net:
    driver: bridge

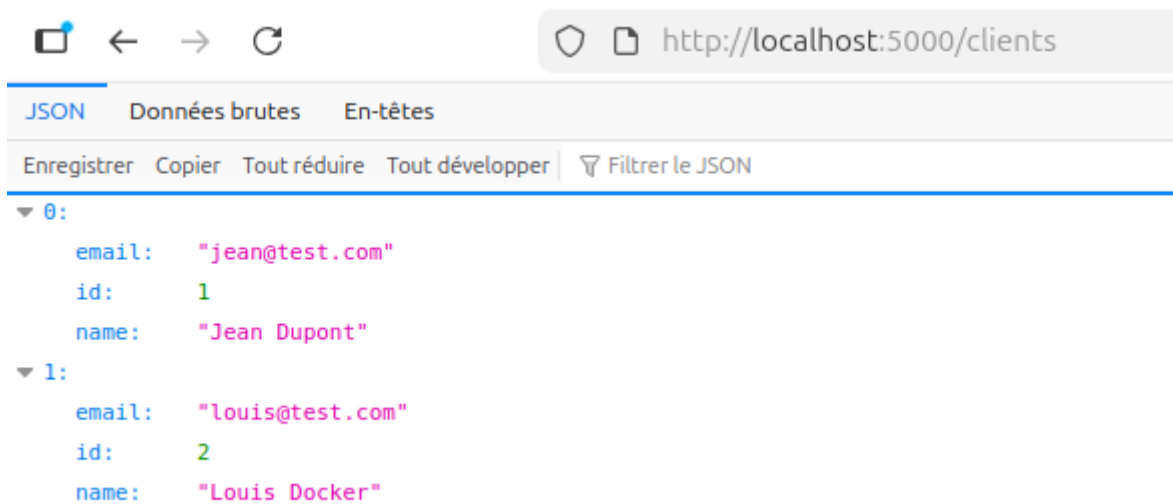
```

Docker compose.yml

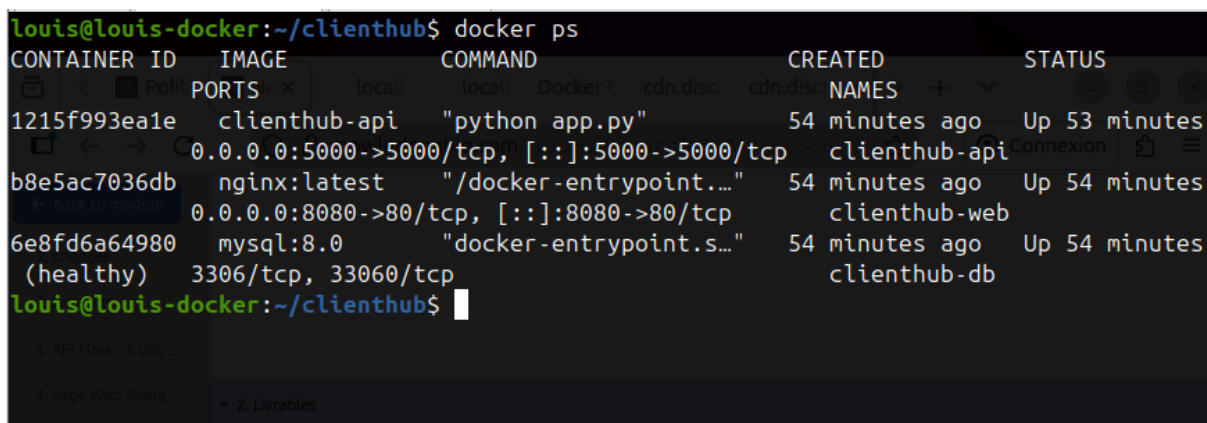
## Page Servie via Nginx:



## Page API:



## Les conteneur en fonctionnement (docker ps):



## QUESTION:

En quelques lignes, justifiez :

- le choix des ports exposés,
- 2)l'utilisation du réseau Docker,
- 3)la nécessité des volumes.

1)Le port 8080 évite les conflits avec le système hôte.  
3306 car mysql utilise ça par défaut  
80 pour nginx

2)Cela permet aux conteneur de communiquer entre eux tout en isolant l'application du reste de la machine.

3)Cela permet de ne pas perdre les données de la base quand on redémarre un conteneur

Mon lien git hub :

<https://github.com/kidloulou/docker>