Mini Projet Docker – Deploiement de l'application POZO

Lien de l'exercice et code source :

Tâches :

1. Build et test de l'api dans le dossier sample-api
2. Infrastructure as code docker compose de deploiement de la solution globale api + front end
3. Mise en place du registry local avec interface graphique
4. Rendu des livrables

Execution du Mini Projet

1. **Build et test de l'api dans le dossier sample-api**

DockerFile de build de l'api sample-api

```
FROM python:3.8-buster
LABEL name="Sidiki"
RUN apt update -y && apt install python-dev
python3-dev libsasl2-dev python-dev libldap2-
dev libssl-dev -y
RUN pip install flask==2.0.0
flask_httpauth==4.1.0 flask_simpleldap python-
dotenv==0.14.0
COPY ./requirements.txt /requirements.txt
COPY ./student_age.py /student_age.py
RUN pip3 install -r /requirements.txt
EXPOSE 5000
CMD [ "python", "./student_age.py" ]
```

Je me deplace dans le dossier simple-api et je fait la commande :

docker build -t simple-api . pour build l'image simple api

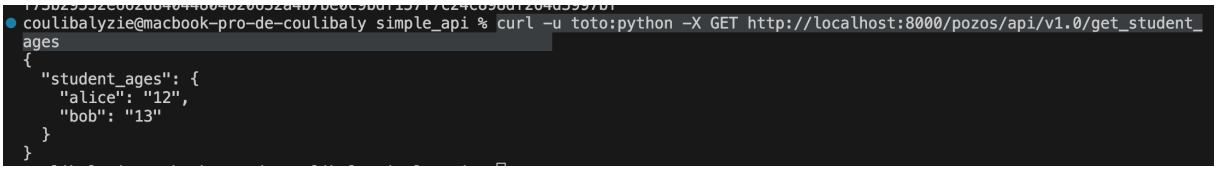
Ensuite je fais la commande

docker run -d -p 8000:5000 --name simple-api-test -v ./student_age.json:/data/student_age.json simple-api

Ce qui va demarrer un conteneur simple-api-test sur le port 8000 en local et 5000 sur le conteneur docker avec un volume de type bind monté sur conteneur dans le dossier /data .

Test de l'api avec la commande curl

curl -u toto:python -X GET http://localhost:8000/pozos/api/v1.0/get_student_ages



2. **Infrastructure as code docker compose de deploiement de la solution globale api + front end**

Docker-compose.yml à la racine du dossier

```yaml
version: '3.8'

services:

  website:

    image: php:apache

    environment:

      - USERNAME=toto

      - PASSWORD=python

      - API_PORT=5000

      - HOST_API=api

    volumes:

      - ./website:/var/www/html

    ports:

      - "8080:80"

    networks:

      - app_net
```

```yaml
  api:
    build: ./simple_api
    ports:
      - "81:5000"
    volumes:
      - ./simple_api/student_age.json:/data/student_age.json
    networks:
      - app_net
    container_name: api
networks:
  app_net:
    driver: bridge
```

Modification du fichier **index.php** dans le dossier webiste

```php
{
$username = getenv('USERNAME');
$password = getenv('PASSWORD');
$port = getenv('API_PORT');
$api_ip_or_name = getenv('HOST_API');
```

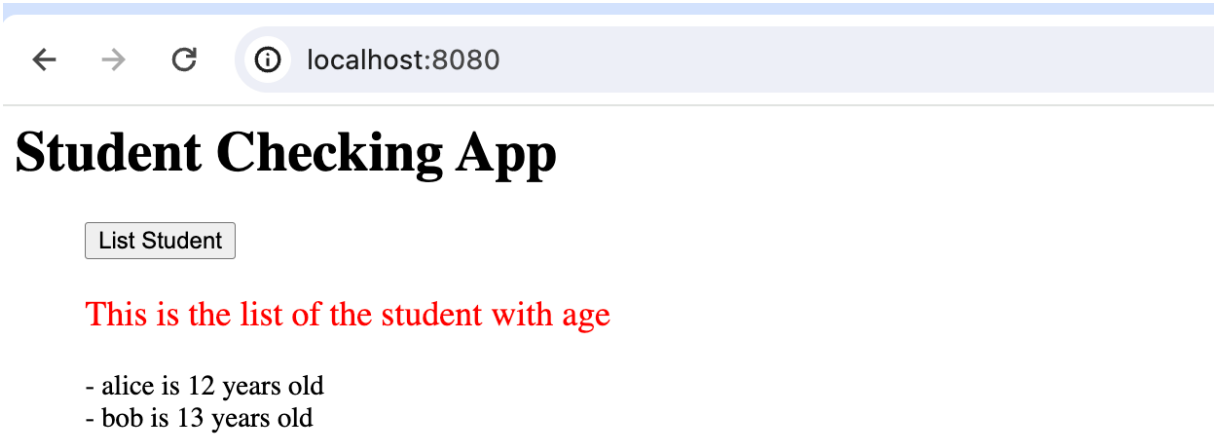
```php
$url = 'http://'.$api_ip_or_name.':'.$port.'/pozos/api/v1.0/get_student_ages';
```

Build du docker-compose avec la commande docker-compose up

Vérification des conteneurs en marche **docker ps**

```
coulibalyzie@macbook-pro-de-coulibaly student-list-master % docker ps
CONTAINER ID   IMAGE                      COMMAND                  CREATED         STATUS
 PORTS                    NAMES
c48f0e395254   php:apache                 "docker-php-entrypoi…"   2 minutes ago   Up 9 seconds
 0.0.0.0:8080->80/tcp     student-list-master-website-1
a4ed94eaf578   student-list-master-api    "python ./student_ag…"   2 minutes ago   Up 9 seconds
 0.0.0.0:81->5000/tcp     api
```

Verification et test de l'application deployée



3. Mise en place du registry local avec interface graphique

Creation d'un dossier docker registry et creation d'un docker-compose.yml donc le deployment du registry et de UI

```yaml
version: '3.8'



services:

  registry-ui:

    image: joxit/docker-registry-ui

    restart: always

    ports:

      - 8089:80

    environment:

      - REGISTRY_URL=http://localhost:5001

      - REGISTRY_TITLE=Docker Registry UI

      - DELETE_IMAGES=true
```

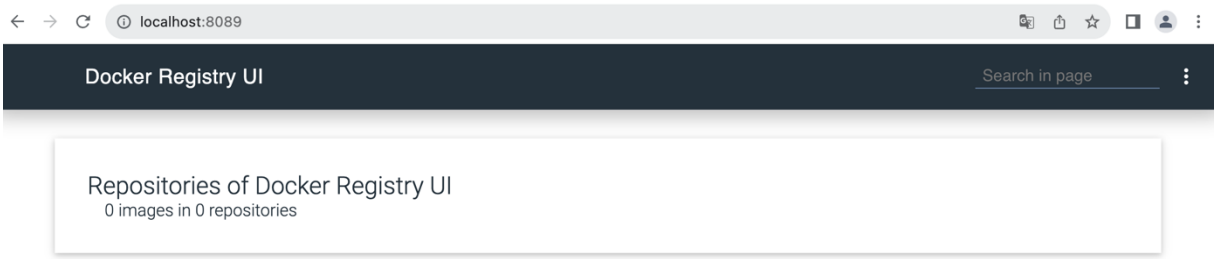
```
      - REGISTRY_HTTP_HEADERS_Access-Control-Allow-
Origin="['http://localhost:8089']"
    container_name: registry-ui
    networks:
      - registry-network


  registry-server:
    image: registry:2.8.2
    restart: always
    ports:
      - 5001:5000
    volumes:
      - ./registry/data:/var/lib/registry
    container_name: registry-server
    networks:
      - registry-network
networks:
  registry-network:
    driver: bridge
```

cd docker-registry && docker-compose up

docker ps

```
     PORTS                           NAMES
d07f3ed2e96e   joxit/docker-registry-ui                          "/docker-entrypoint.…"   38 seconds ago   Up 8 seconds
     0.0.0.0:8089->80/tcp            registry-ui
e8119c9d7dcf   registry:2.8.2                                    "/entrypoint.sh /etc…"   38 seconds ago   Up 8 seconds
     0.0.0.0:5001->5000/tcp          registry-server
```



Tag et push d’une image sur le docker hub local

```
coulibalyzie@macbook-pro-de-coulibaly docker-registry % docker tag student-list-master-api localhost:5001/student-list-master-a
pi
coulibalyzie@macbook-pro-de-coulibaly docker-registry %
```

```
coulibalyzie@macbook-pro-de-coulibaly docker-registry % docker push localhost:5001/student-list-master-api
Using default tag: latest
The push refers to repository [localhost:5001/student-list-master-api]
9101a90c84f4: Pushed
55bc1e170b1a: Pushed
8b9bd9c27998: Pushed
631ea3fc00df: Pushed
355e4758e95f: Pushing [==================================>              ]  115.7MB/161.6MB
5960de057cee: Pushed
fb221799d3d3: Pushed
21c43420a9a9: Pushed
01ed3e47c143: Pushed
867c03a4b370: Pushing [=======>                                         ]  66.51MB/465.1MB
7d9d4c2307c1: Pushing [=========================>                       ]  76.62MB/144.1MB
c401210c4968: Pushed
97551bf43823: Pushing [======================>                          ]  51.53MB/108.3MB
```