# Projet POZOS Dockerisation de l'application **student**_list

Ce projet vise à conteneuriser l'application **student_list** en utilisant Docker et Docker Compose.

1-Cloner le projet

Nous allons récupérer le projet avec la commande git clone

```
vagrant@debian12:~$ git clone https://github.com/guissepm/student-list.git
Cloning into 'student-list'...
remote: Enumerating objects: 54, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 54 (delta 10), reused 4 (delta 4), pack-reused 34 (from 3)
Receiving objects: 100% (54/54), 21.51 KiB | 1.54 MiB/s, done.
Resolving deltas: 100% (11/11), done.
vagrant@debian12:~$ ls
student-list
vagrant@debian12:~$
```

2-Fichier dockerfile

Nous allons maintenant éditer le fichier dockerfile

```
  GNU nano 7.2                                          Dockerfile *
# Utilisation de l'image Python 3.8 comme base
FROM python:3.8-buster

# Mainteneur de l'image
LABEL maintainer="Jean <zacharia.sidibe@univ-thies.sn>"

# Définition du répertoire de travail
WORKDIR /

# Copier le fichier requirements.txt pour installer les dépendances
COPY requirements.txt /

# Mise à jour du système et installation des prérequis
RUN apt update -y && apt install -y python-dev python3-dev libsasl2-dev libldap2-dev libssl-dev \
    && pip3 install -r /requirements.txt

# Copier le code source de l'API
COPY . /

# Définition du volume pour stocker les données persistantes
VOLUME /data

# Exposition du port de l'API
EXPOSE 5000

# Commande de démarrage de l'API
CMD ["python3", "./student_age.py"]
```

3-On va maintenant installer Docker compose et créer le fichier Docker compose .yml

```
vagrant@debian12:~/student-list/simple_api$ sudo curl -L "https://github.com/docker/compose/releases/latest/download/doc
ker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 71.4M  100 71.4M    0     0  3259k      0  0:00:22  0:00:22 --:--:-- 3147k
```

```yaml
version: '3.8'

services:
  api:
    build: ./simple_api
    container_name: student_api
    ports:
      - "5000:5000"
    volumes:
      - ./simple_api/student_age.json:/data/student_age.json
    networks:
      - student_network

  web:
    image: php:apache
    container_name: student_web
    ports:
      - "8080:80"
    volumes:
      - ./website:/var/www/html
    depends_on:
      - api
    networks:
      - student_network

networks:
  student_network:
```

4-Nous allons maintenant construire l'Image api_container

```
vagrant@debian12:~/student-list/simple_api$ sudo docker build -t student_api .
[+] Building 266.8s (6/8)                                                                                    docker:default
 => [1/5] FROM docker.io/library/python:3.8-buster@sha256:04c3f641c2254c229fd2f704c5199ff4bea57d26c1c29008ae3a4  212.4s
 => => resolve docker.io/library/python:3.8-buster@sha256:04c3f641c2254c229fd2f704c5199ff4bea57d26c1c29008ae3a4af  0.0s
 => => sha256:04c3f641c2254c229fd2f704c5199ff4bea57d26c1c29008ae3a4afddde98709 988B / 988B                      0.0s
 => => sha256:406c7153026b3565d5558f11c84fb77dff1f2a045c2e9b05dac78a8416050db7 7.56kB / 7.56kB                  0.0s
 => => sha256:b1e7e053c9f6f57c6d95002167a6d57aed6aacf04dd2f8e681cb4f74a7ca4381 51.87MB / 51.87MB               52.3s
 => => sha256:bc4b9fb034a871b285bea5418cedfcaa9d2ab5590fb5fb6f0c42aaebb2e2c911 2.01kB / 2.01kB                  0.0s
 => => sha256:ac8bb7e1a32398e26c129ce64e2ddc3e7ec6c34d93424b247f16049f5a91cff4 50.45MB / 50.45MB              103.7s
 => => sha256:3b1c264c0ad4598c25048a6dbd3030086cc5c74000e11d04ac27944cb116aabb 17.58MB / 17.58MB               14.9s
 => => sha256:a2e1e233599c00054fb839db78b4d42e6f12f36b64280aa62d482a3ad0ad7109 191.88MB / 191.88MB            196.7s
 => => sha256:0ebfe287e9761b9b7dd1703470ff3473a62fe75238f3de01282165f8725968af 6.15MB / 6.15MB                58.5s
 => => sha256:819d305a9b2210516eaf5929daf6caa015f1f0ffa8be96252cb05d0f283bfff5 19.29MB / 19.29MB              94.4s
 => => sha256:a11e135762f00d55e3d68410f1aab593ca9850d9c2ca07f69fb5ac353fc99e0a 243B / 243B                    96.1s
 => => sha256:5e4e21ea7d0468facdec867e65083ff3d90f16a69413e7c2a891b99c13ad8a55 2.85MB / 2.85MB              103.2s
 => => extracting sha256:ac8bb7e1a32398e26c129ce64e2ddc3e7ec6c34d93424b247f16049f5a91cff4                     4.0s
 => => extracting sha256:3b1c264c0ad4598c25048a6dbd3030086cc5c74000e11d04ac27944cb116aabb                     1.0s
 => => extracting sha256:b1e7e053c9f6f57c6d95002167a6d57aed6aacf04dd2f8e681cb4f74a7ca4381                     4.6s
 => => extracting sha256:a2e1e233599c00054fb839db78b4d42e6f12f36b64280aa62d482a3ad0ad7109                    12.3s
 => => extracting sha256:0ebfe287e9761b9b7dd1703470ff3473a62fe75238f3de01282165f8725968af                     0.7s
 => => extracting sha256:819d305a9b2210516eaf5929daf6caa015f1f0ffa8be96252cb05d0f283bfff5                     1.2s
 => => extracting sha256:a11e135762f00d55e3d68410f1aab593ca9850d9c2ca07f69fb5ac353fc99e0a                     0.0s
 => => extracting sha256:5e4e21ea7d0468facdec867e65083ff3d90f16a69413e7c2a891b99c13ad8a55                     0.3s
```

5-Démarrer l'ensemble des services avec Docker Compose avec la commande docker-compose up -d

```
vagrant@debian12:~/student-list$ sudo docker-compose up -d
WARN[0000] /home/vagrant/student-list/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, pleas
e remove it to avoid potential confusion
[+] Running 15/15
 ✔web Pulled                                                                                                        87.0s
   ✔6e909acdb790 Pull complete                                                                                      35.6s
   ✔a843a84fd431 Pull complete                                                                                      35.6s
   ✔54df264c65c0 Pull complete                                                                                      80.2s
   ✔e78f4a695473 Pull complete                                                                                      80.3s
   ✔dc325eebd405 Pull complete                                                                                      82.1s
   ✔b9e2f83d4e30 Pull complete                                                                                      82.1s
   ✔f01400cc6ea2 Pull complete                                                                                      82.1s
   ✔19883a5b6404 Pull complete                                                                                      82.3s
   ✔d49a5d4d78e5 Pull complete                                                                                      82.3s
   ✔9be9496ea2d8 Pull complete                                                                                      83.4s
   ✔dce5011ff25f Pull complete                                                                                      83.4s
   ✔51cb2ba39d1e Pull complete                                                                                      83.4s
   ✔7d6f3098fbfc Pull complete                                                                                      83.4s
   ✔4f4fb700ef54 Pull complete                                                                                      83.5s
Compose can now delegate builds to bake for better performance.
 To do so, set COMPOSE_BAKE=true.
[+] Building 1.8s (10/10) FINISHED                                                                        docker:default
```

```
[+] Running 4/4
 ✔api                                          Built
 ✔Network student-list_student_network         Created
 ✔Container student_api                        Started
 ✔Container student_web                        Started
```

6-nous allons maintenant tester l'API avec curl:

curl -u toto:python -X GET http://localhost:5000/pozos/api/v1.0/get_student_ages

```
vagrant@debian12:~/student-list$ curl -u toto:python -X GET http://localhost:5000/pozos/api/v1.0/get_student_ages
{
  "student_ages": {
    "alice": "12",
    "bob": "13"
  }
}
vagrant@debian12:~/student-list$
```



7-Nous allons Tester l'application Web

http://192.168.56.100:8080/index.php


L'application est bien dockoriser

8-Registre Docker

Exécutons la commande suivante pour démarrer un registre privé:

docker run -d -p 5000:5000 --name registry-pozos --network student-list_api-pozos registry:2

Cela démarre un registre privé sur le port 4000.

```
vagrant@uroot:~/student-list$ sudo docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED              STATUS              PORTS
     NAMES
dab6d6e627e2   registry:2         "/entrypoint.sh /etc…"   About a minute ago   Up About a minute   0.0.0.0:4000->5000/t
cp   registry-pozos
9d8ac0f2cade   php:apache         "docker-php-entrypoi…"   22 minutes ago       Up 22 minutes       0.0.0.0:8080->80/tcp
     student_web
fd88278dba1c   student-list-api   "python3 ./student_a…"   22 minutes ago       Up 22 minutes       0.0.0.0:5000->5000/t
cp   student_api
vagrant@uroot:~/student-list$ sudo docker ps
```

9-nous allons maintenant taguer notre image

docker tag student_api:latest localhost:5000/student_api:latest

```
ocalhost:4000/student_api   latest    930815ea89ef   12 hours ago    1.13GB
```

Nous allons maintenant pousser notre image dans notre registre avec la commande: docker push localhost:4000/student_api

```
vagrant@uroot:~/student-list$ sudo docker push localhost:4000/student_api
Using default tag: latest
The push refers to repository [localhost:4000/student_api]
ae0592d19d03: Pushed
9668352d59d5: Pushing [==================>                                ]  95.07MB/248.1MB
1837fcaba16f: Pushed
6e2f63b0cf05: Pushed
45359261cd7a: Pushed
ad312497d9a5: Pushed
474c7af10697: Pushed
dcc1cfeee1ab: Pushing [====>                                              ]  48.67MB/509.6MB
eccb9ed74974: Pushing [======================>                            ]  62.92MB/145.6MB
53d40515380c: Pushed
6af7a54a0a0d: Pushing [==================>                                ]  44.39MB/114.1MB
```

On va maintenant verifier si notre images est bien poussée.

curl -X GET http://localhost:4000/v2/_catalog

```
vagrant@uroot:~/student-list$ curl -X GET http://localhost:4000/v2/_catalog
{"repositories":["student_api"]}
vagrant@uroot:~/student-list$
```

10- Interface web

pour gérer le registre Nous allons maintenant ajouter une interface web pour visualiser les images. Exécutons la commande suivante pour déployer une interface web:

```
docker run -d \
 --name registry-pozos_UI \
 --network student-list_api-pozos \
 -p 4002:80 \
 -e REGISTRY_TITLE="POZOS REGISTRY" \
 -e REGISTRY_URL="http://registry-pozos:5000" \
 -e CATALOG_ELEMENTS_LIMIT=1000 \
 joxit/docker-registry-ui:static
```

```
vagrant@uroot:~/student-list$ sudo docker run -d \
  --name registry-pozos_UI \
  --network student-list_student_network \
  -p 4002:80 \
  -e REGISTRY_TITLE="POZOS REGISTRY" \
  -e REGISTRY_URL="http://registry-pozos:4000" \
  -e CATALOG_ELEMENTS_LIMIT=1000 \
  joxit/docker-registry-ui:static
Unable to find image 'joxit/docker-registry-ui:static' locally
static: Pulling from joxit/docker-registry-ui
540db60ca938: Pull complete
197dc8475a23: Pull complete
39ea657007e5: Pull complete
37afbf7d4c3d: Pull complete
0c01f42c3df7: Pull complete
d590d87c9181: Pull complete
3333c94ae44f: Pull complete
33d7cca6fc9f: Pull complete
076b2dd9bdd1: Pull complete
b70198f04ee7: Pull complete
1fb6c5acc953: Pull complete
Digest: sha256:b0657b6be748173583516e411bd71552e54cb7d5dda94964726297ce8774415c
Status: Downloaded newer image for joxit/docker-registry-ui:static
4b220668d7ee72f040587fd7a88864dbbfc5133db987f66f66fe80ba413712c5
vagrant@uroot:~/student-list$
```

On peut y acceder avec cette Url: http//:192.168.56.100:4002