

MINI-PROJET

1- La création du **Dockerfile**

```
mar. juil. 11 20:02:57
Dockerfile - vagrant [SSH: 192.168.56.12] - Visual Studio Code

Dockerfile M X docker-compose.yml M index.php M

student-list > simple_api > Dockerfile > ...
1 FROM python:2.7-buster AS base
2
3 LABEL MAINTAINER="kamgaingrodrigue3@gmail.com"
4 VOLUME [ "/data" ]
5 EXPOSE 5000
6 COPY student_age.py /
7
8 FROM base AS build
9 RUN apt update -y && apt install python-dev python3-dev libsasl2-dev python-dev libldap2-dev libssl-dev -y
10
11 RUN pip install flask==1.1.2 flask_httpauth==4.1.0 flask_simpleldap python-dotenv==0.14.0
12
13 #ADD git clone https://github.com/diranetafen/student-list.git /
14
15 #RUN cd student-list/
16 |
17 FROM build AS final
18 CMD ["python", "./student_age.py"]
```

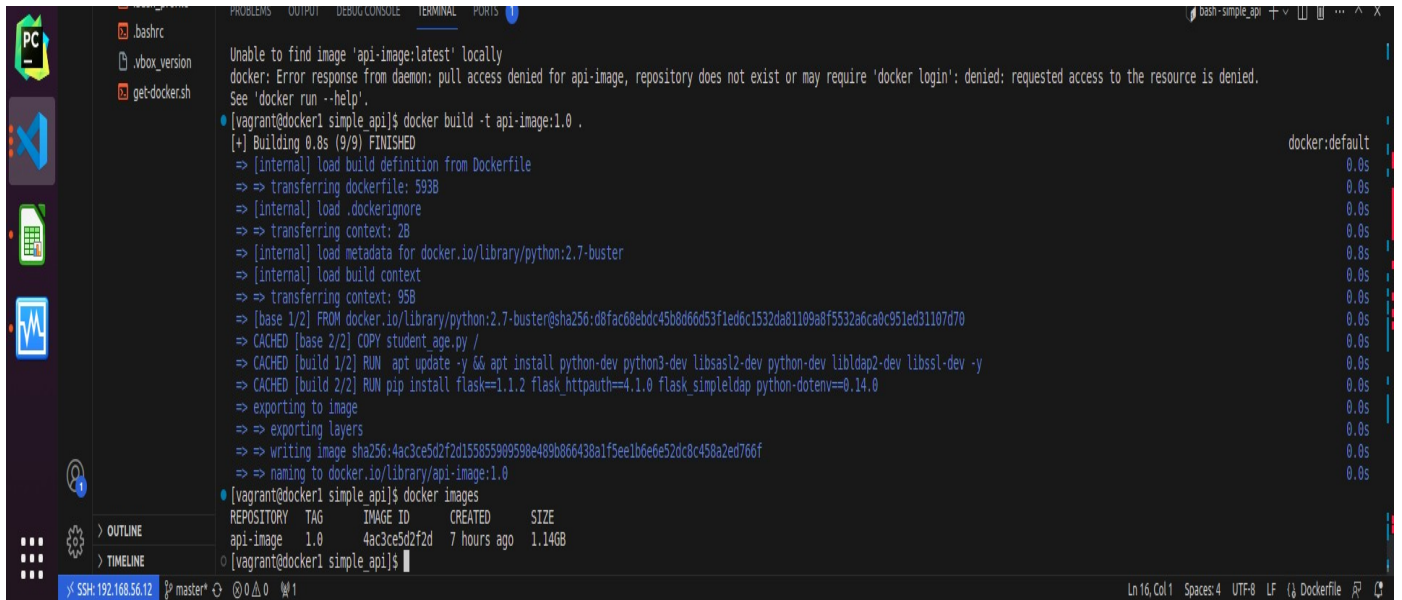
2- Création de l'image de l'api

docker build -t api-image:1.0 .

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1
bash - simple_api + v [ ] ... ^ X

[vagrant@docker1 simple_api]$ docker build -t api-image:1.0 .
[+] Building 0.8s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 593B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/python:2.7-buster
=> [internal] load build context
=> => transferring context: 95B
=> [base 1/2] FROM docker.io/library/python:2.7-buster@sha256:d8fac68ebdc45b8d66d53f1ed6c1532da81109a8f5532a6ca0c951ed31107d70
=> CACHED [base 2/2] COPY student_age.py /
=> CACHED [build 1/2] RUN apt update -y && apt install python-dev python3-dev libsasl2-dev python-dev libldap2-dev libssl-dev -y
=> CACHED [build 2/2] RUN pip install flask==1.1.2 flask_httpauth==4.1.0 flask_simpleldap python-dotenv==0.14.0
=> exporting to image
=> => exporting layers
=> => writing image sha256:4ac3ce5d2f2d155855909590e489b866438a1f5ee1b6e6e52dc8c458a2ed766f
=> => naming to docker.io/library/api-image:1.0
[vagrant@docker1 simple_api]$ docker build -t api-image:1.0 .

Ln 16, Col 1 Spaces: 4 UTF-8 LF Dockerfile
```



```
.bashrc
.vbox_version
get-docker.sh

Unable to find image 'api-image:latest' locally
docker: Error response from daemon: pull access denied for api-image, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.

[vagrant@docker1 simple_api]$ docker build -t api-image:1.0 .
[+] Building 0.8s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 593B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/python:2.7-buster
=> [internal] load build context
=> => transferring context: 95B
=> [base 1/2] FROM docker.io/library/python:2.7-buster@sha256:d8fac68ebdc45b8d66d53f1ed6c1532da81109a8f5532a6ca0c951ed31107d70
=> CACHED [base 2/2] COPY student_age.py /
=> CACHED [build 1/2] RUN apt update -y && apt install python-dev python3-dev libssl2-dev python-dev libldap2-dev libssl-dev -y
=> CACHED [build 2/2] RUN pip install flask==1.1.2 flask_httpauth==4.1.0 flask_simpleldap python-dotenv==0.14.0
=> exporting image
=> => exporting layers
=> => writing image sha256:4ac3ce5d2f2d155855909590e489b866438a1f5e1b6e6e52dc8c458a2ed766f
=> => naming to docker.io/library/api-image:1.0

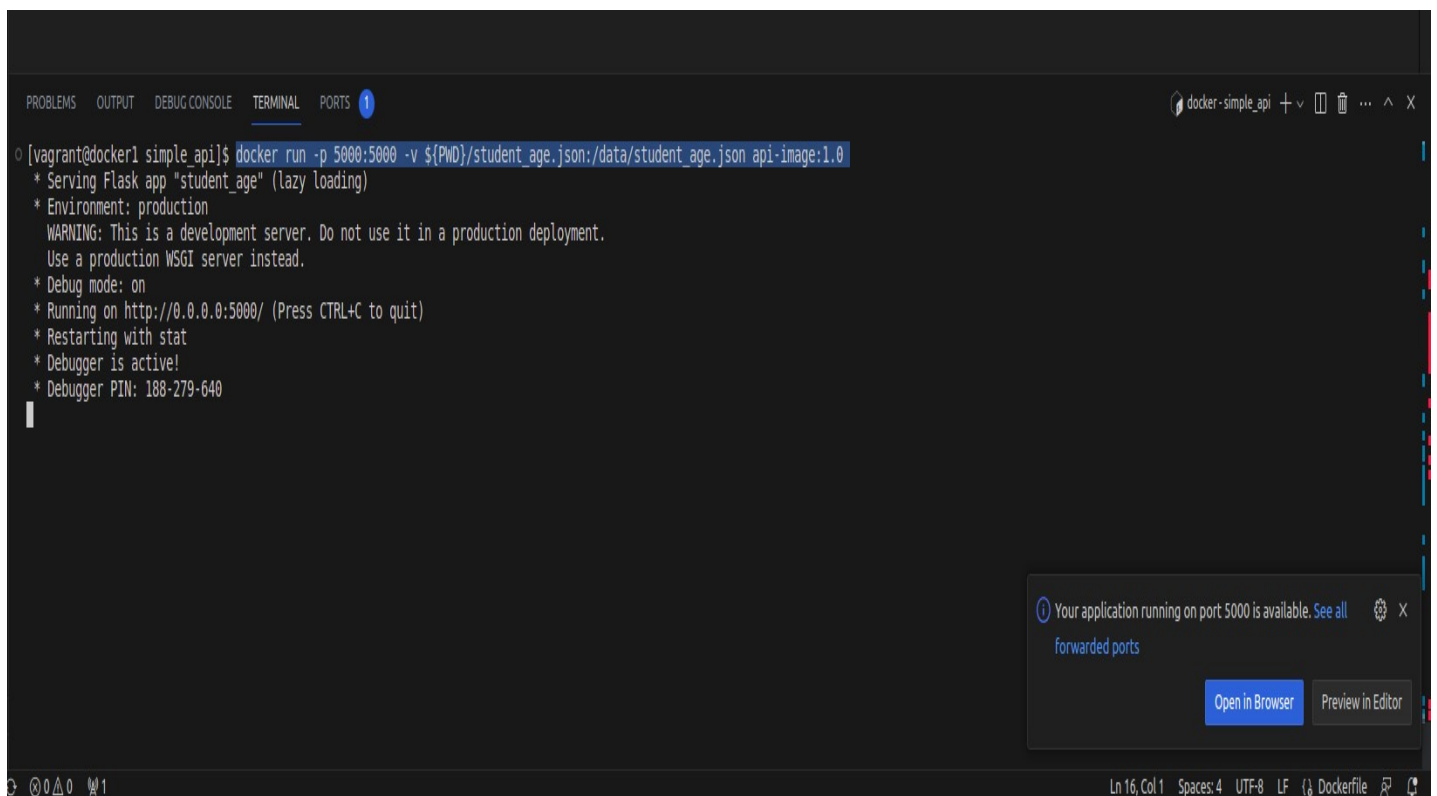
[vagrant@docker1 simple_api]$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
api-image 1.0 4ac3ce5d2f2d 7 hours ago 1.14GB

[vagrant@docker1 simple_api]$
```

Le nom de l'image créée est : **api-image**

3- Lancement du conteneur avec l'image précédente

docker run -p 5000:5000 -v \${PWD}/student_age.json:/data/student_age.json api-image:1.0

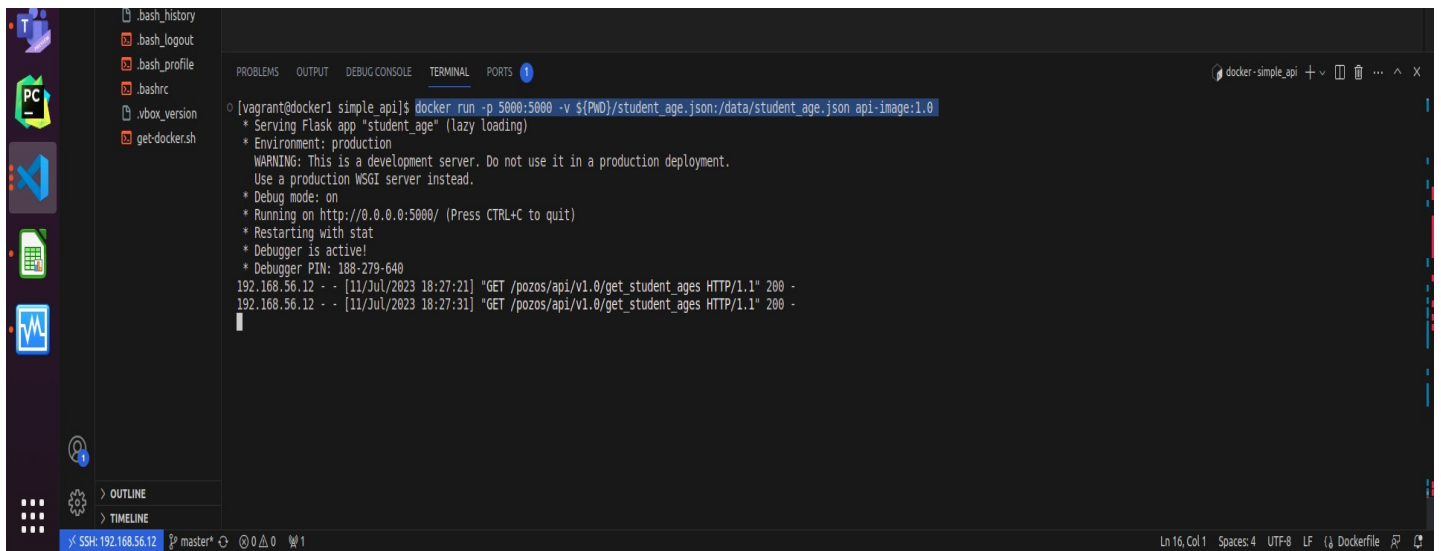


```
[vagrant@docker1 simple_api]$ docker run -p 5000:5000 -v ${PWD}/student_age.json:/data/student_age.json api-image:1.0
* Serving Flask app "student_age" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 188-279-640
```

4 – Test de l'api

L'adresse sur laquelle le conteneur est déployée : 192.168.56.12 sur le port 5000

curl -u toto:python -X GET http://192.168.56.12:5000/pozos/api/v1.0/get_student_ages



```
[vagrant@docker1 simple_api]$ docker run -p 5000:5000 -v ${PWD}/student_age.json:/data/student_age.json api-image:1.0
* Serving Flask app "student_age" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 188-279-640
192.168.56.12 - - [11/Jul/2023 18:27:21] "GET /pozos/api/v1.0/get_student_ages HTTP/1.1" 200 -
192.168.56.12 - - [11/Jul/2023 18:27:31] "GET /pozos/api/v1.0/get_student_ages HTTP/1.1" 200 -
```

Ici, on peut remarquer que l’api a reçu des requêtes



```
[vagrant@docker1 simple_api]$ curl -u toto:python -X GET http://192.168.56.12:5000/pozos/api/v1.0/get_student_ages
{
  "student_ages": {
    "alice": "12",
    "bob": "13"
  }
}
[vagrant@docker1 simple_api]$
```

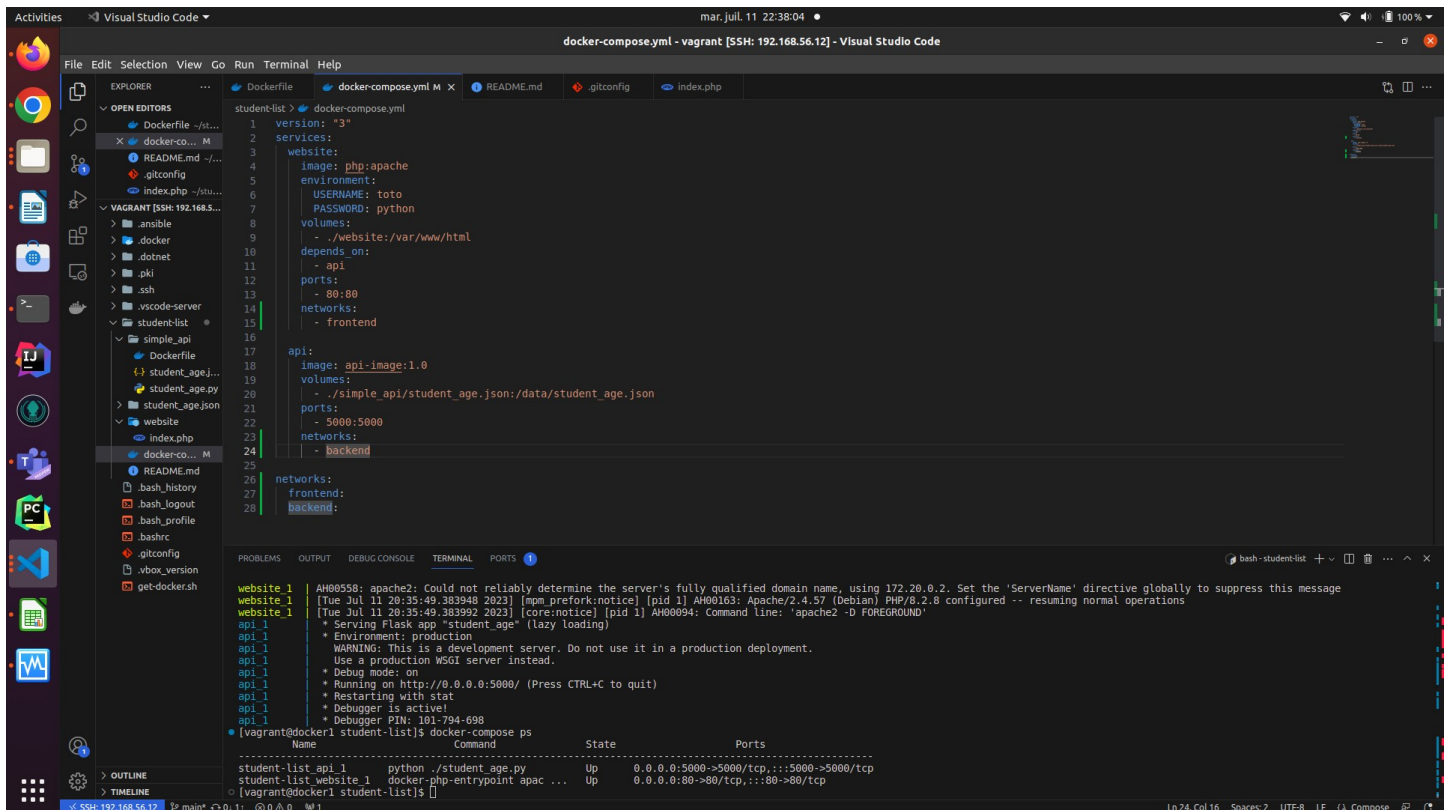
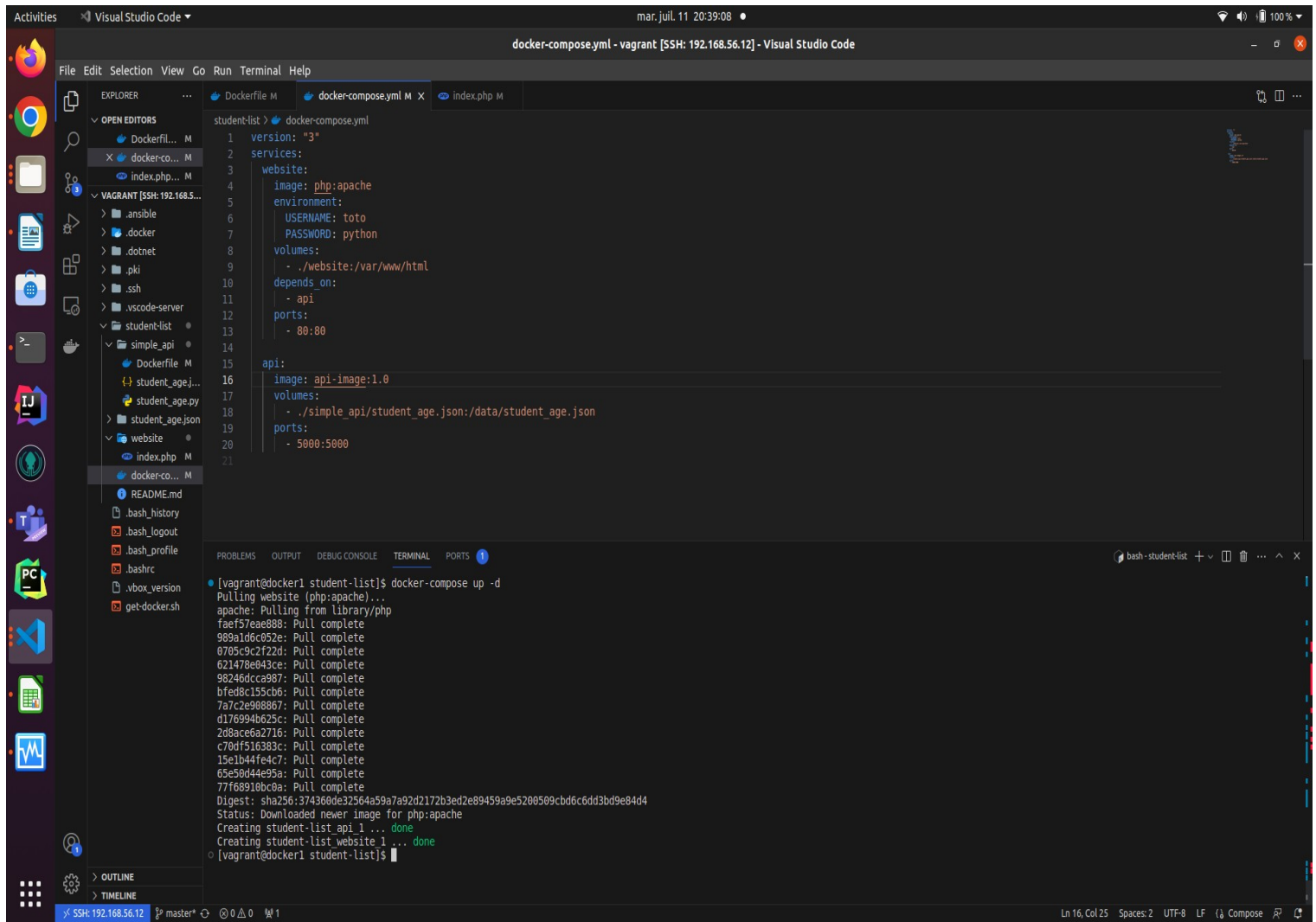
Ici cest la reponse venant de notre base de données : **student_age.json**

5 – Le **docker-compose.yml**

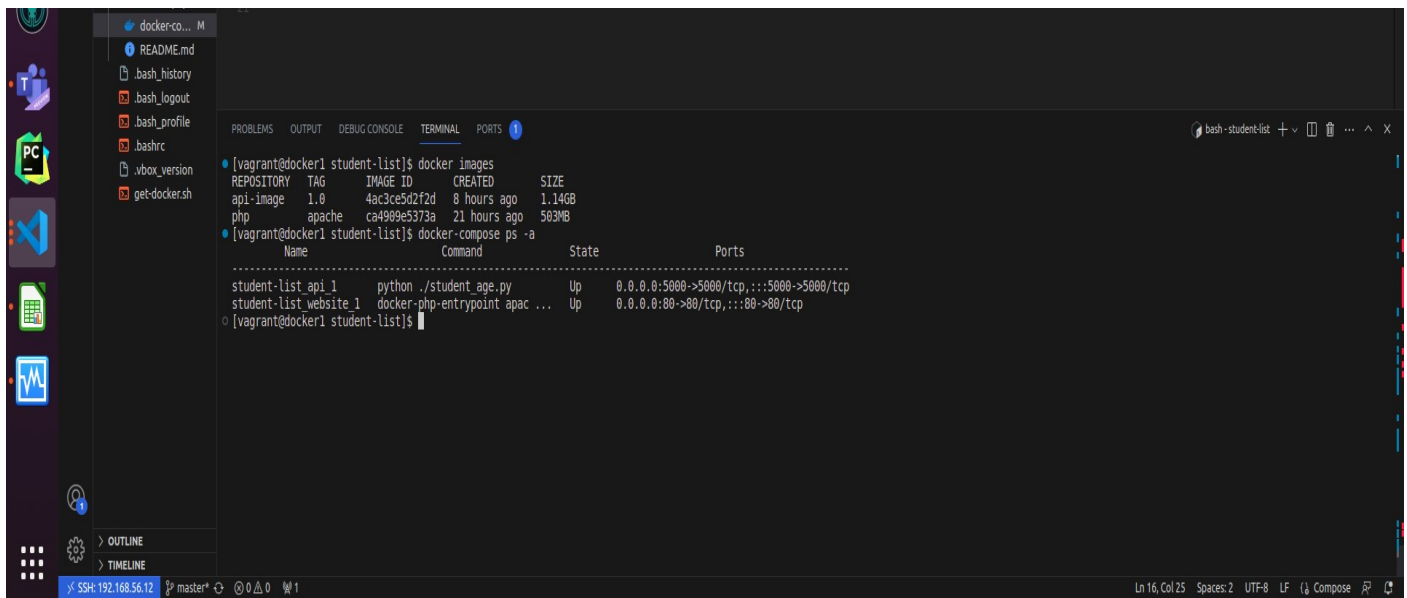
Après avoir supprimé le conteneur de l’image **api-image**, on crée le fichier **docker-compose.yml**.

Voici le contenu et le lancement :

Lancement: **docker-compose up -d**



- les differents services :



The screenshot shows a terminal window with the following commands and output:

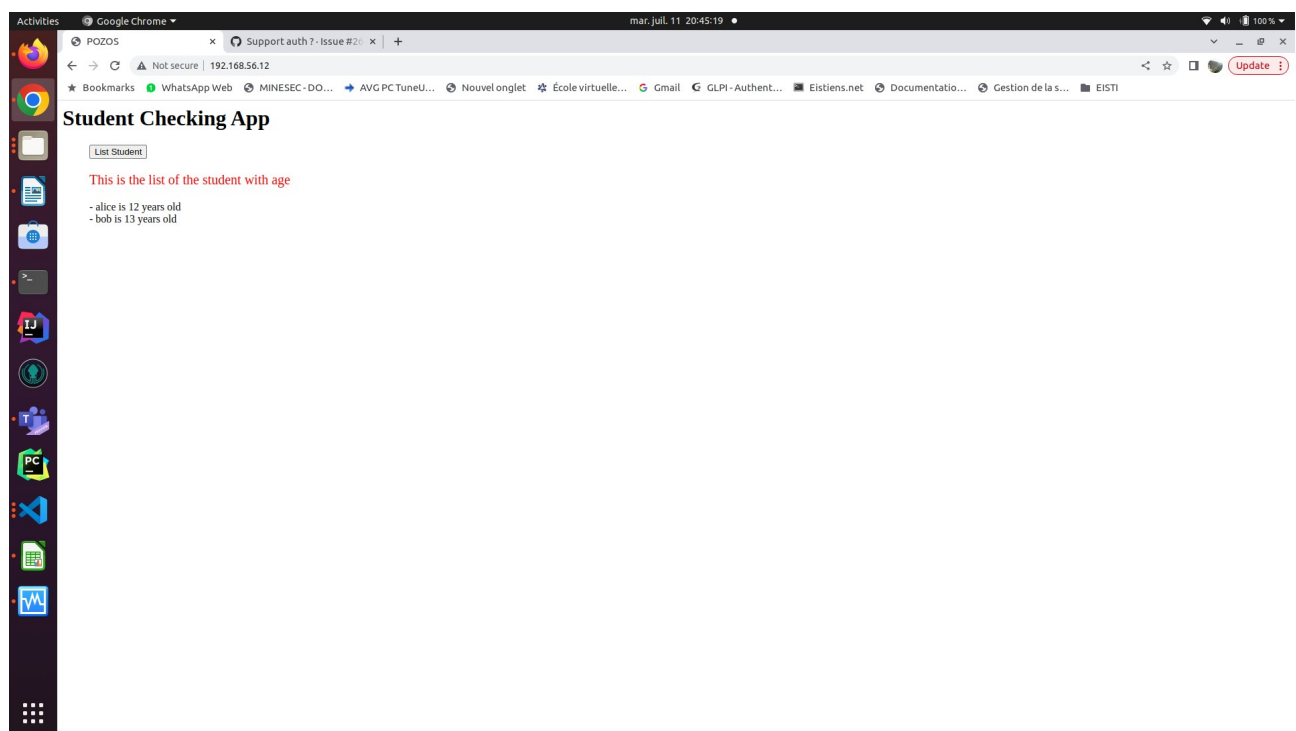
```
[vagrant@docker1 student-list]$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
api-image     1.0       4ac3ce502f2d   8 hours ago    1.14GB
php           apache    ca4909e5373a   21 hours ago    503MB
```

```
[vagrant@docker1 student-list]$ docker-compose ps -a
```

Name	Command	State	Ports
student-list api_1	python ./student_age.py	Up	0.0.0.0:5000->5000/tcp, :::5000->5000/tcp
student-list website_1	docker-php-entrypoint apac ...	Up	0.0.0.0:80->80/tcp, :::80->80/tcp

```
[vagrant@docker1 student-list]$
```

Test sur le navigateur: 192.168.56.12:80



Les sources sont disponibles ici : <https://github.com/Rodriguez001/dockerLabs.git/Mini-projet>