**Mini-Projet DOCKER**

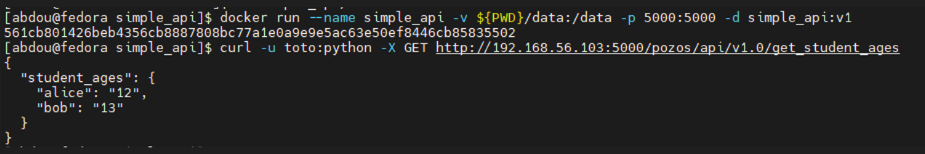
**Partie1: Build and test**

**Dans cette partie j’ai créé mon Dockerfile qui va me permettre de builder mon image selon les spécifications qui m’ont été données.**

**(Voir Dockerfile sur le delivery)**

**Une fois mon image buildée, j’ai pu la tester avec Curl**

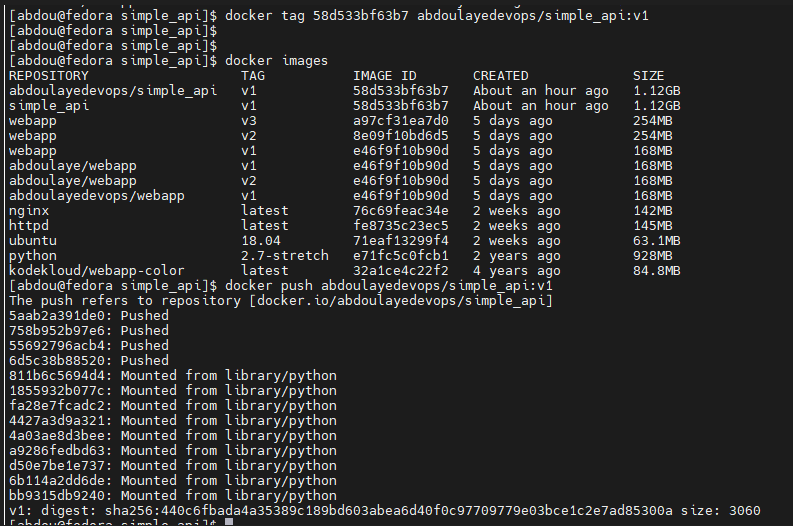
**curl -u toto:python -X GET <http://192.168.56.103:5000/pozos/api/v1.0/get_student_ages>**

****

**Partie 2: Infrastructure as code**

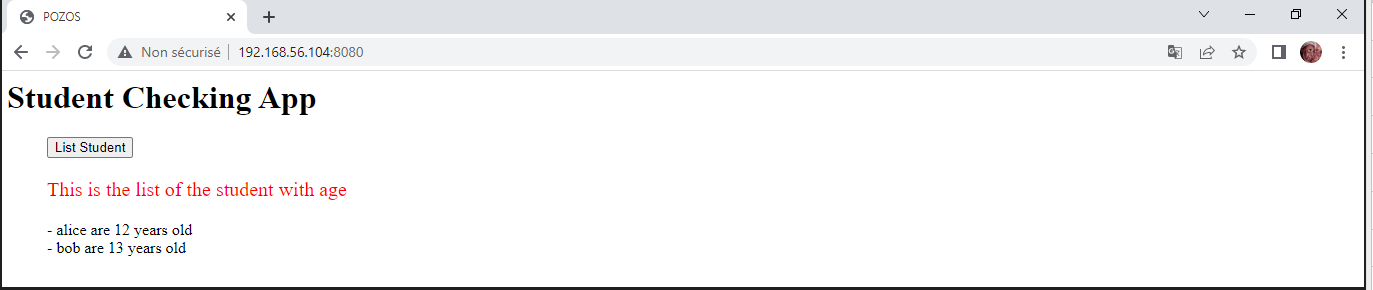
**Ici j’ai realisé mon docker-compose.yml en tenant compte de la dependance entre les conatainers, des ports à exposer et j’ai ajouté un reseau commun afin de permettre la communication entre les deux conteneurs.**

**Au préalable j’ai uploadé mon image déja buildée à la premiére étape sur le dockerhub afin de pouvoir l’appeler dans mon docker-compose pour creer un autre conteneur identique.**

****

**(Voir docker-compose.yml sur le delivery)**

**Test accés API depuis l’interface WEB**

****

**Partie 3: Creation Registre privé**

**Dans cette partie j’ai utilisé la solution fourni par docker: registry**

**Couplé avec le FrontEnd joxit/docker-registry-ui qui me permet de visualiser mon registre via une interface HTTP.**

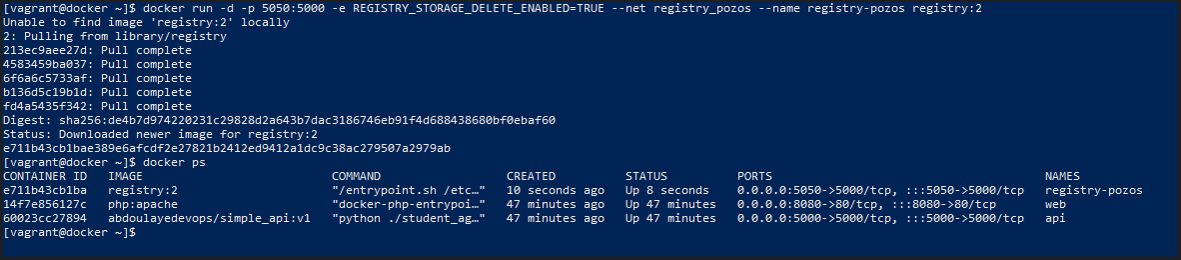
**Au préalable je créée mon réseau qui va acceuillir mes deux conteneurs le registre et le frontend:**

**\*docker network create registry\_pozos**

**-Installation registre privé**

**(Ici j’ai exposé le port 5050 en exterieur car le 5000 sur le hôte est deja associé à l’API POZOS)**

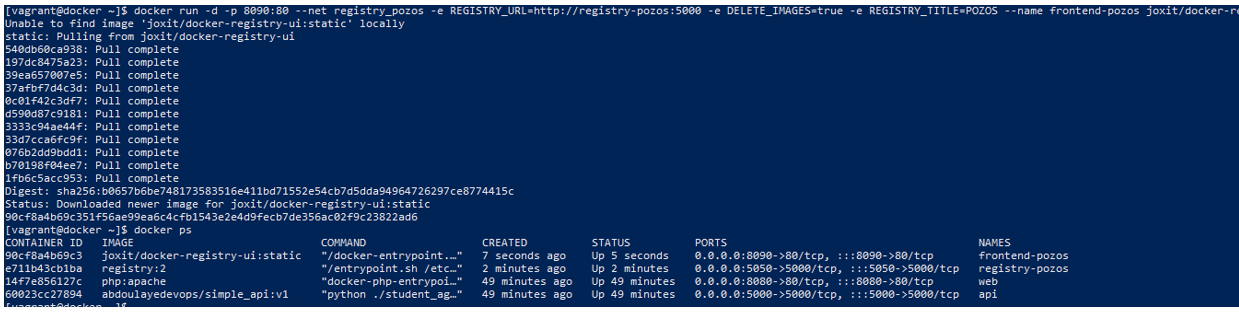
**docker run -d -p 5050:5000 -e REGISTRY\_STORAGE\_DELETE\_ENABLED=TRUE --net registry\_pozos --name registry-pozos registry:2**

****

**-Installation FrontEnd joxit**

**(En specifiant les variables d’environnement pour se connecter au registre)**

**docker run -d -p 8090:80 --net registry\_pozos -e REGISTRY\_URL=http://registry-pozos:5000 -e DELETE\_IMAGES=true -e REGISTRY\_TITLE=POZOS --name frontend-pozos joxit/docker-registry-ui:static**

****

**-Ajout des images api\_pozos et web\_pozos**

**\*Premiere etape création des tags pour l’ajout dans le registre privé local**

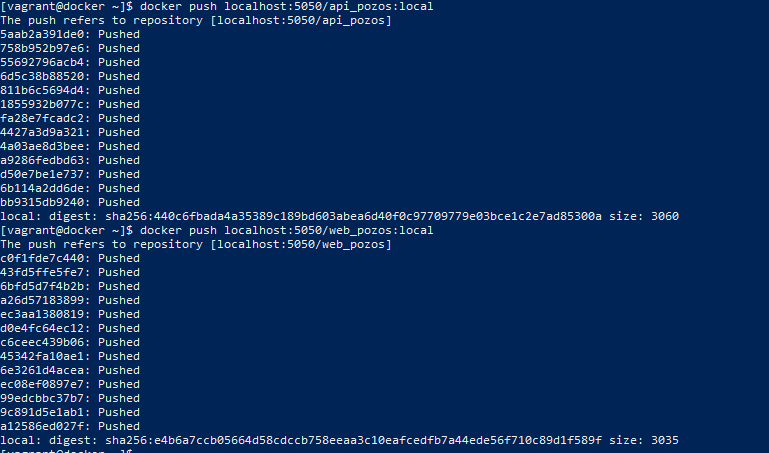
**docker tag 58d533bf63b7 localhost:5050/api\_pozos:local**

**docker tag b94cb75b58d5 localhost:5050/web\_pozos:local**

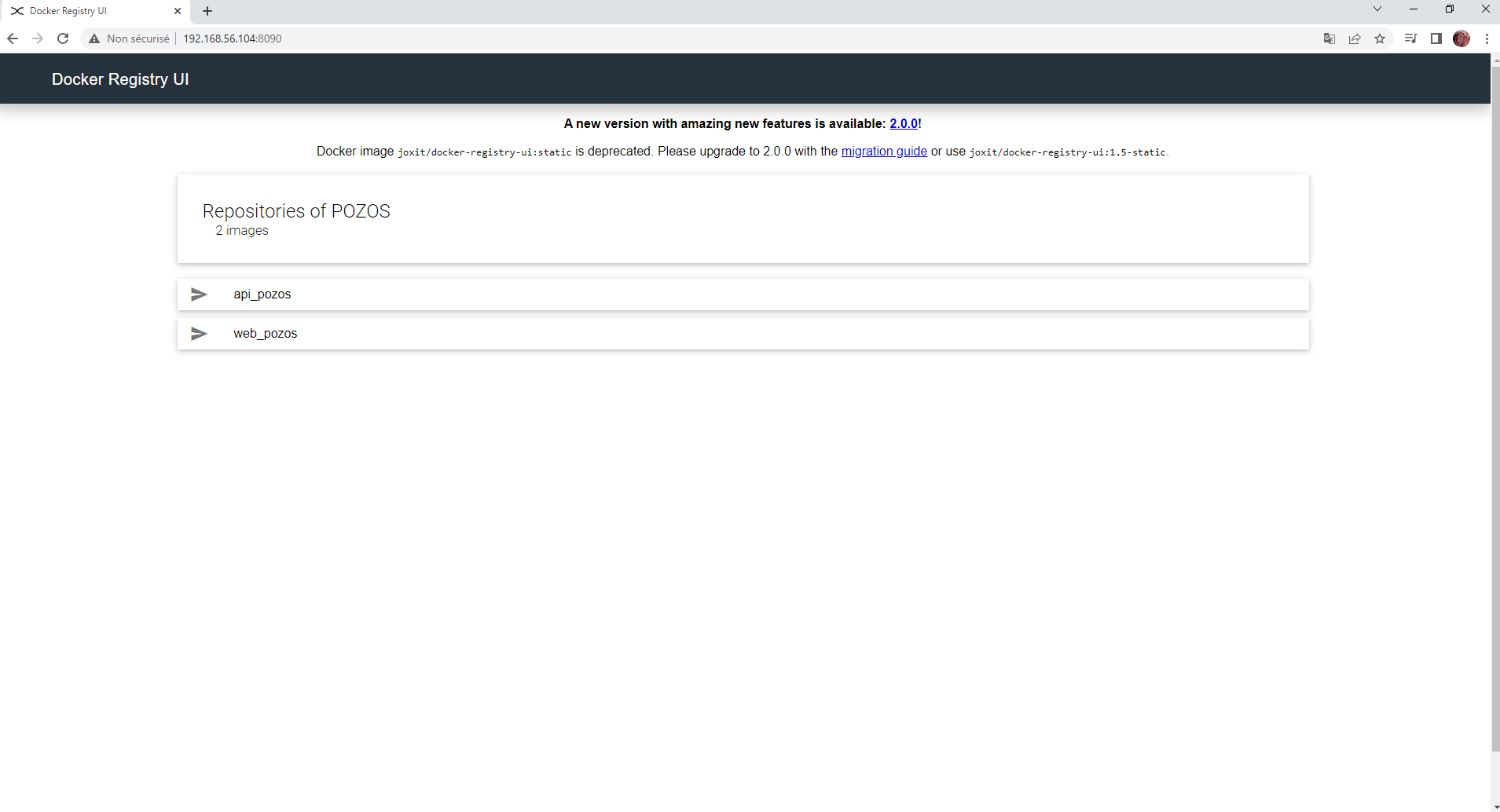
**\*Upload sur le registry**

**docker push localhost:5050/api\_pozos:local**

**docker push localhost:5050/web\_pozos:local**

****

**\*Visualition sur le frontEnd**

**ToutTou**

Toutes les captures d’ecran sont ajoutés dans le delivery pour plus de clarté.

**Merci**