



Ndeye Gnilane NDOUR

COMPTE RENDU DU PROJET DOCKER - POZOS

Introduction

POZOS est une société informatique qui cherche à moderniser son infrastructure en le rendant évolutive et automatisée. Ce projet vise à mettre en place un **<POC>** en utilisant Docker et Docker Compose pour conteneuriser une application existante.

L'application **student_list** est composée de deux modules :
- Une **API REST** qui fournit une liste d'étudiants à partir d'un fichier JSON.
- Une **application web en PHP** permettant d'afficher ces informations aux utilisateurs

L'objectif principal est de découpler ces services et d'assurer un déploiement simplifié grâce à Docker.

1. Installation de Docker et Docker Compose

➤ Installation de docker

Installer des outils utiles pour yum et ajout du dépôt officiel de Docker

vagrant@192:~

```
[vagrant@192 ~]$ sudo yum install -y yum-utils
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* epel: mirror.math.princeton.edu
Package yum-utils-1.1.31-54.el7_8.noarch already installed and latest version
Nothing to do
[vagrant@192 ~]$ sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
Loaded plugins: fastestmirror
adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yum.repos.d/docker-ce.repo
repo saved to /etc/yum.repos.d/docker-ce.repo
```

L'action suivante est d'installer docker

```
[vagrant@192 ~]$ sudo yum install -y docker-ce docker-ce-cli containerd.io
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* epel: mirror.math.princeton.edu
Package 3:docker-ce-26.1.4-1.el7.x86_64 already installed and latest version
Package 1:docker-ce-cli-26.1.4-1.el7.x86_64 already installed and latest version
Package containerd.io-1.6.33-3.1.el7.x86_64 already installed and latest version
Nothing to do
```

On va maintenant lancer docker et le configurer pour qu'il démarre automatiquement à chaque répétition du système.

```
[vagrant@192 ~]$ sudo systemctl start docker
[vagrant@192 ~]$ sudo systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service
to /usr/lib/systemd/system/docker.service.
[vagrant@192 ~]$ docker --version
Docker version 26.1.4, build 5650f9b
```

➤ Installation de Docker compose

Installons Docker Compose, un outil permettant de gérer plusieurs conteneurs avec un fichier docker-compose.yml.

```
[vagrant@192 ~]$ sudo curl -L "https://github.com/docker/compose/releases/latest
/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compo
se
% 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 70.2M  100 70.2M    0     0 1508k      0  0:00:47  0:00:47 --:--:-- 397k
[vagrant@192 ~]$ sudo chmod +x /usr/local/bin/docker-compose
```

2. Création et configuration des conteneurs

➤ Clonage d'un dépôt Git avec Docker

On va créer un dossier du nom de projet_pozos et se déplacer vers ce dossier

```
[vagrant@192 ~]$ mkdir projet_pozos && cd projet_pozos
```

La commande suivante exécute un conteneur Docker avec l'image **alpine/git**, monte le répertoire actuel de l'hôte dans le conteneur à **/git**, puis clone le dépôt Git **https://github.com/guissepmp/student-list.git** dans le répertoire **/git/student-list**, le rendant accessible dans le répertoire courant de l'hôte.

```
[vagrant@192 projet_pozos]$ sudo docker run --rm alpine/git clone https://github.com/guissepmp/student-list.git
Cloning into 'student-list'...
```

On a listé et on voit bien que le clonage a réussi

```
[vagrant@192 projet_pozos]$ ls -l
total 0
drwxr-xr-x. 5 root root 94 Mar 12 14:41 student-list
[vagrant@192 projet_pozos]$ cd student-list/
[vagrant@192 student-list]$ ls -l
total 8
-rw-r--r--. 1 root root 0 Mar 12 14:41 docker-compose.yml
-rw-r--r--. 1 root root 7535 Mar 12 14:41 README.md
drwxr-xr-x. 2 root root 94 Mar 12 14:41 simple_api
drwxr-xr-x. 2 root root 23 Mar 12 14:41 website
```

- Création d'un Dockerfile pour l'API

```
[vagrant@192 simple_api]$ sudo nano Dockerfile
```

Un **Dockerfile** a été créé pour construire une image Docker de l'API.

```
GNU nano 2.3.1 File: Dockerfile
# Utiliser l'image officielle de Python
FROM python:3.8-buster

# Mainteneur de l'image
LABEL maintainer="Ndeye Gnilane NDOUR"

# Définir le répertoire de travail
WORKDIR /

# Copier les fichiers nécessaires dans le conteneur
COPY student_age.py /student_age.py
COPY student_age.json /data/student_age.json
COPY requirements.txt /requirements.txt

# Installer les dépendances
RUN apt update -y && apt install -y \
    python-dev python3-dev libsasl2-dev libldap2-dev libssl-dev && \
    pip3 install -r /requirements.txt

# Exposer le port de l'API
EXPOSE 5000

# Définir le point d'entrée
CMD ["python3", "/student_age.py"]
```

Ce Dockerfile permet de créer une image Docker pour exécuter L'API REST, d'inclure les fichiers sources et dépendances, de lancer automatiquement l'API au démarrage du conteneur

- Construction et test de l'image

L'image Docker a été construite avec la commande suivante

```
[vagrant@192 student-list]$ sudo docker build -t student-api simple_api/
[*] Building 326.7s (10/10) FINISHED
docker:default
--> [internal] load build definition from Dockerfile
--> [internal] load metadata for docker.io/library/python:3.9-buster
--> [internal] load dockerignore
--> [internal] load .dockerignore
--> [internal] build context
--> [1/4] FROM docker.io/library/python:3.9-buster@sha256:74097641e2254c229f4d794e319174b0e7426c1c2299f0e04a2f2a08700
--> [2/4] WORKDIR /app
--> [3/4] COPY requirements.txt /app
--> [4/4] RUN pip install --no-cache-dir -r requirements.txt
--> [5/4] COPY . /app
--> [6/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [7/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [8/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [9/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [10/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [11/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [12/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [13/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [14/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [15/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [16/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [17/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [18/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [19/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [20/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [21/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [22/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [23/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [24/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [25/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [26/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [27/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [28/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [29/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [30/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [31/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [32/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [33/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [34/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [35/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [36/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [37/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [38/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [39/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [40/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [41/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [42/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [43/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [44/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [45/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [46/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [47/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [48/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [49/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [50/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [51/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [52/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [53/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [54/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [55/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [56/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [57/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [58/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [59/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [60/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [61/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [62/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [63/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [64/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [65/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [66/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [67/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [68/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [69/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [70/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [71/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [72/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [73/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [74/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [75/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [76/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [77/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [78/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [79/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [80/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [81/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [82/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [83/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [84/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [85/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [86/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [87/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [88/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [89/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [90/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [91/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [92/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [93/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [94/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [95/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [96/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [97/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [98/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [99/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
--> [100/4] RUN python3 -m pip install --no-cache-dir -r requirements.txt
```

Super ! L'image a été créée avec succès

```
[vagrant@192 student-list]$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
student-api         latest             b2d5efcb2331       About a minute ago 1.13GB
alpine/git          latest             98ee8546290d       10 days ago        91.5MB
hello-world         latest             74cc54e27dc4       7 weeks ago        10.1kB
```

Cette commande démarre un conteneur en arrière-plan (-d) nommé **api_student**, basé sur l'image **student_api**.

Elle expose l'API sur le port 5000, et monte un fichier JSON de l'hôte dans le conteneur pour qu'il puisse y accéder sous **/data/student_age.json**.

```
[vagrant@192 student-list]$ sudo docker run -d --name api-student -p 5000:5000 -v $(pwd)/simple_api/student_age.json:/data/student_age.json student-api
eb4f4307ec5dd451489d45ffed572921e5fdc79e51b5131308f4af9b1b9d7961
[vagrant@192 student-list]$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED             STATUS              PORTS                               NAMES
eb4f4307ec5d   student-api   "python3 /student_ag-"   About a minute ago   Up About a minute   0.0.0.0:5000->5000/tcp, :::5000->5000/tcp   api-student
[vagrant@192 student-list]$ sudo curl -u toto:python -X GET http://localhost:5000/pozos/api/v1.0/get_student_ages
{
  "student_ages": {
    "alice": "12",
    "bob": "13"
  }
}
```

3. Infrastructure avec Docker Compose

Un fichier **docker-compose.yml** a été créé pour orchestrer l'application. On va l'éditer

```
[vagrant@192 student-list]$ sudo nano docker-compose.yml
```

Ce script **déploie une API en Python et un site web PHP sous Apache**, les connecte dans un réseau Docker et permet au site d'interagir avec l'API via une architecture **backend-**

frontend

```
GNU nano 2.3.1 File: docker-compose.yml
Version: '3.8'

services:
  api:
    image: student-api # Utilisation de l'image créée précédemment
    container_name: api-student
    ports:
      - "5000:5000"
    volumes:
      - ./simple_api/student_age.json:/data/student_age.json
    networks:
      - student_network

  web:
    image: php:apache
    container_name: web-student
    ports:
      - "8080:80"
    volumes:
      - ./website:/var/www/html
    environment:
      API_USERNAME: "toto"
      API_PASSWORD: "python"
    depends_on:
      - api
    networks:
      - student_network

networks:
  student_network:
    driver: bridge
```

Docker compose up -d permet de démarrer tous les services définis dans docker-compose.yml

```
[vagrant@192 student-list]$ sudo docker compose up -d
WARN[0000] /home/vagrant/projet_pozos/student-list/docker-compose.yml: 'version' is obsolete
[+] Running 15/15
  ✓ web Pulled
  ✓ 7cf63256a31a Pull complete
  ✓ 4afdl72411dc Pull complete
  ✓ f45be8a9df33 Pull complete
  ✓ d7e8ec139b05 Pull complete
  ✓ 69bbbff9d9a6 Pull complete
  ✓ d0f79085f073 Pull complete
  ✓ 52f244cd5125 Pull complete
  ✓ d3d1c0ea2bbd Pull complete
  ✓ c6f46fa7475e Pull complete
  ✓ 42bbe1f66265 Pull complete
  ✓ effa821f91fe Pull complete
  ✓ 8c20370dec0d Pull complete
  ✓ 29501ed5d5fe Pull complete
  ✓ 4f4fb700ef54 Pull complete
[+] Running 3/3
  ✓ Network student-list_student_network Created
  ✓ Container api-student Started
  ✓ Container web-student Started
[vagrant@192 student-list]$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
26e53c358e72   php:apache    "docker-php-entrypoi..." 53 seconds ago Up 50 seconds 0.0.0.0:8080->80/tcp, :::8080->80/tcp web-student
b29d76882a7c   student-api   "python3 /student_ag..." 53 seconds ago Up 51 seconds 0.0.0.0:5000->5000/tcp, :::5000->5000/tcp api-student
[vagrant@192 student-list]$ sudo curl -u toto:python -X GET http://localhost:5000/posos/api/v1.0/get_student_ages
{
  "student_ages": {
    "alice": "12",
    "bob": "13"
  }
}
```

Dans website il y'a un index.php on va l'éditer

```
[vagrant@192 student-list]$ cd website/
[vagrant@192 website]$ ls
index.php
[vagrant@192 website]$ sudo nano index.php
```

```

<html>
<head>
  <title>POZOS</title>
</head>
<body>
  <h1>Student Checking App</h1>
  <ul>
    <form action="" method="POST">
      <!--<label>Enter student name:</label><br />
      <input type="text" name="" placeholder="Student Name" required/>
      <br /><br />
      <button type="submit" name="submit">List Student</button>
    </form>

    <?php
      if($_SERVER['REQUEST_METHOD'] == "POST" and isset($_POST['submit']))
      {
        $username = getenv('USERNAME');
        $password = getenv('PASSWORD');
        if ( empty($username) ) $username = 'fake_username';
        if ( empty($password) ) $password = 'fake_password';
        $context = stream_context_create(array(
          "http" => array(
            "header" => "Authorization: Basic " . base64_encode("$username:$password"),
          ));

        $url = 'http://192.168.1.85:5000/pozos/api/v1.0/get_student_ages';
        $username = 'toto';
        $password = 'python';

        // Crée un contexte de flux avec l'authentification de base
        $context = stream_context_create([
          "http" => [
            "header" => "Authorization: Basic " . base64_encode("$username:$password")
          ]
        ]);

        // Utilise le contexte pour faire la requête
        $response = file_get_contents($url, false, $context);
        $list = json_decode(file_get_contents($url, false, $context), true);
        echo "<p style='color:red;; font-size: 20px;'>This is the list of the student with age</p>";
        foreach($list["student_ages"] as $key => $value) {
          echo "- $key is $value years old <br>";
        }
      }
    </ul>
  </body>
</html>

```

L'objectif du code est de créer une page web qui permet d'afficher la liste des étudiants et leur âge en récupérant ces informations depuis une API REST sécurisée par une authentification Basic. Lorsqu'un utilisateur clique sur le bouton **"List Student"**, une requête est envoyée à l'API, et les données sont affichées dynamiquement sur la page.



L'image montre une page web, où un bouton **"List Student"** permet d'afficher une liste d'étudiants avec leur âge.

Après avoir cliqué sur le bouton, la page affiche le message **" Ceci est la liste des étudiants avec l'âge "**, suivi des noms et âges des étudiants récupérés depuis une API. Cela confirme que l'application fonctionne comme prévu en affichant les données reçues de l'API.

5. Mise en place d'un registre Docker privé

POZOS souhaite stocker les images Docker dans un registre privé. Voici comment il a été déployé :

La commande suivante démarre un registre privé où on peut stocker notre image Docker.

```
[vagrant@192 student-list]$ sudo docker run -d -p 5001:5000 --name registry registry:2  
5f6df805cbd4fadf6a218fb21ab5d47dd6fbd4e82e8e428413b4b08371e83b1e
```

Le docker tag associe un tag à l'image pour indiquer où elle sera poussée.

```
[vagrant@192 student-list]$ sudo docker tag student-api localhost:5001/student-api
```

Le docker push envoie l'image vers le registre privé.

```
[vagrant@192 student-list]$ sudo docker push localhost:5001/student-api  
Using default tag: latest  
The push refers to repository [localhost:5001/student-api]  
183507c491d6: Pushed  
f2e0a8ee2e3d: Pushed  
7e7b9231d715: Pushed  
edf20da9a815: Pushed  
6e2f63b0cf05: Pushed  
45359261cd7a: Pushed  
ad312497d9a5: Pushed  
474c7af10697: Pushed  
dcc1cfeee1ab: Pushed  
eccb9ed74974: Pushed  
53d40515380c: Pushed  
6af7a54a0a0d: Pushed  
latest: digest: sha256:0544f32577a359fb06c523856f350ecd36d4c11b5d134ae5af7d21993cb5d5c0 size: 2841
```

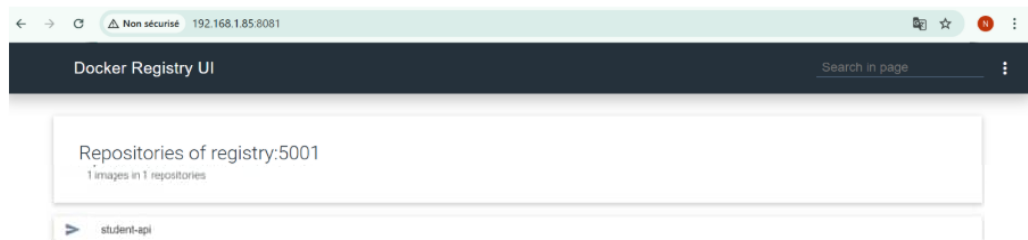
La commande suivante permet de vérifier les images stockées dans mon registre

```
[vagrant@192 student-list]$ curl http://localhost:5001/v2/student-api/tags/list  
{  
  "name": "student-api",  
  "tags": ["latest"]  
}  
[vagrant@192 student-list]$
```

Le résultat signifie que l'image student-api est bien enregistrée dans le registre privé.

```
[vagrant@192 student-list]$ sudo docker run -d -p 8081:80 --name registry-ui -e
REGISTRY_URL=http://registry:5001 --link registry:registry joxit/docker-registry
-ui
Unable to find image 'joxit/docker-registry-ui:latest' locally
latest: Pulling from joxit/docker-registry-ui
619be1103602: Pull complete
6c866301bd2c: Pull complete
9e03973bc803: Pull complete
a3a550dcd386: Pull complete
d18780149b81: Pull complete
4ea31a8fb875: Pull complete
e57ebb3e2067: Pull complete
4f4fb700ef54: Pull complete
b8f1be213d73: Pull complete
cf7a173b6dfb: Pull complete
6ace07744b36: Pull complete
c306db4532fb: Pull complete
5a4f7d72e702: Pull complete
Digest: sha256:5594b76bf8dd9de479648e28f38572d020d260568be40b7e52b9758b442275e1
Status: Downloaded newer image for joxit/docker-registry-ui:latest
9fbce46ed5e14c965063048b968996cf9ac89b3fd439c36941f03cf14f3bcc09
[vagrant@192 student-list]$ ^C
```

Cette commande permet de déployer une interface utilisateur pour notre registre Docker privé à l'aide de l'image `joxit/docker-registry-ui`.



L'image `student-api` est maintenant bien visible dans l'interface Docker Registry UI.

L'interface est accessible à `192.168.1.85:8081`.

Le registre utilisé est sur le port `5001`.

L'image `student-api` est bien présente dans le registre.

Conclusion

Ce projet a permis de conteneuriser l'application `student_list` pour POZOS, rendant son déploiement plus agile et modulaire. Grâce à Docker et Docker Compose, la mise à jour et la gestion des services (API et interface Web) sont simplifiées.

L'intégration d'un registre Docker privé améliore également la gestion des versions et la sécurité du déploiement.