TP Docker

Classe : 4A-41 / Groupe 9

Nom et Prénom : GUIZANI Ahmed, SHE Houyu

Partie 0 :

1- Rappeler le concept d'un Dockerfile

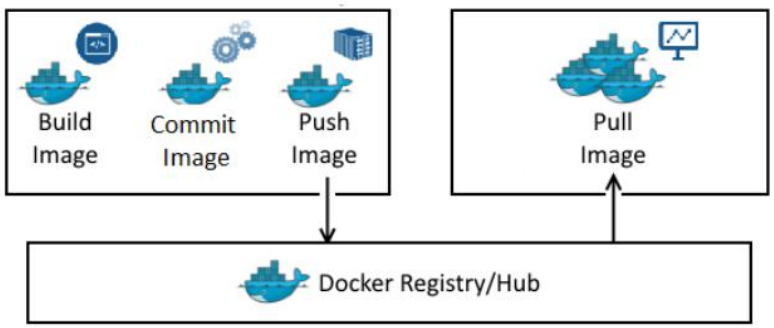
Dockerfile est un fichier contenant des instructions pour créer une image Docker

2- Rappeler le concept d'un docker-compose

Docker-Compose est un service d'orchestration de Docker. C'est un outil permettant de définir et d'exécuter des applications multi-conteneurs à l'aide de Docker, permettant aux utilisateurs de l'environnement de développement rapide et isolé de Docker. Docker-Compose résout le problème de la gestion de l'orchestration entre les conteneurs.

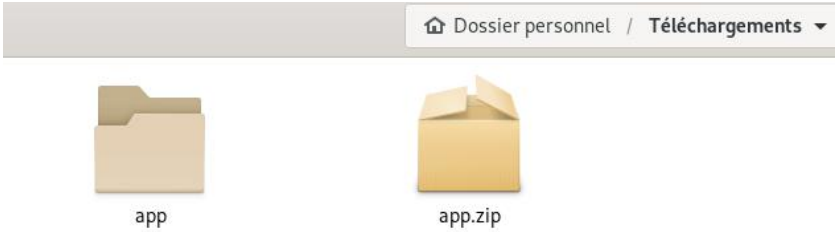
3- C'est quoi le docker registry ?

Le registre Docker stocke les images de Docker. On peut faire des pushs et des pulls images. Il fonctionne comme l'image ci-dessous :



Partie 1 : Dockerfile

1- Télécharger le fichier zip suivant: http://demo.amamou.com/app.zip



2- Vérifier que l'application tourne bien sur la machine pour cela vous devez :

- Installer postgresql sur votre machine linux ou sur la vm fournie

```
root@tp-docker:/home/tp-docker# apt install postgresql
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  libpq5 postgresql-11 postgresql-client-11 postgresql-client-common
  postgresql-common sysstat
Paquets suggérés :
  postgresql-doc postgresql-doc-11 libjson-perl isag
Les NOUVEAUX paquets suivants seront installés :
  libpq5 postgresql postgresql-11 postgresql-client-11 postgresql-client-common
  postgresql-common sysstat
```

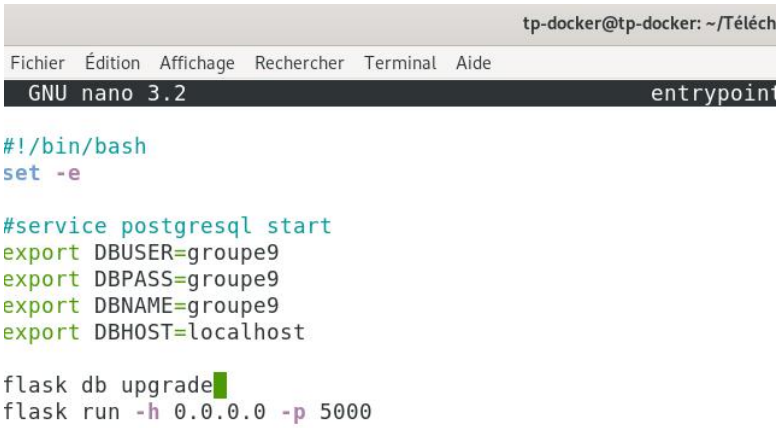
- Créer la base de donnée et le user :

```
postgres=# CREATE DATABASE groupe9;
CREATE DATABASE
postgres=#
postgres=# CREATE USER groupe9 WITH PASSWORD 'groupe9';
CREATE ROLE
postgres=#
postgres=# GRANT ALL ON DATABASE groupe9 TO groupe9;
GRANT
postgres=#
```

- Spécifier les variable d'environnement suivantes dans le fichier entrypoint.sh fournis avec l'application : DBUSER, DBPASS, DBHOST, DBNAME

```
tp-docker@tp-docker:~$ cd Téléchargements/app
tp-docker@tp-docker:~/Téléchargements/app$
tp-docker@tp-docker:~/Téléchargements/app$ ls
app.py      entrypoint.sh  models.py         templates
Dockerfile  migrations     requirements.txt
tp-docker@tp-docker:~/Téléchargements/app$
```

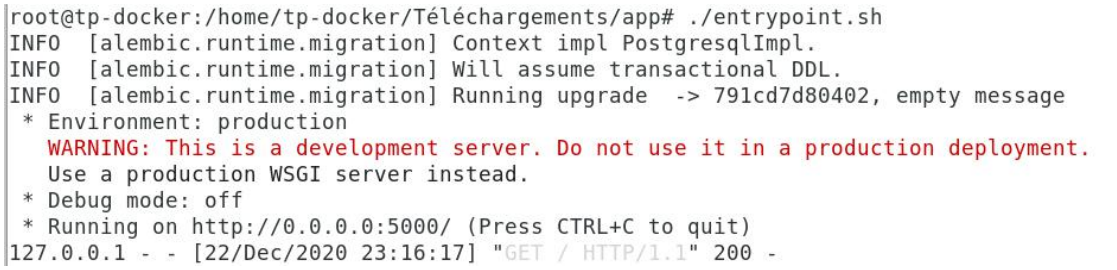
Dans le fichier entrypoint.sh, on spécifie les variables suivantes comme ci-dessous :



- Installer les requirements qui se trouvent dans le requirements.txt (**pip3 install -r requirements.txt**)

- Démarrer l'application en exécutant **entrypoint.sh**

Lorsqu'on tape **127.0.0.1:5000** sur Firefox, il répond "GET / HTTP/1.1" 200 - comme ci-dessous :



3- Créer une image docker a partir d'un Dockerfile fourni. Il faudra modifier les fichier de configuration de postgres au niveau du host pour autoriser les connexions depuis le réseau docker 172.17.0.0./16 et aussi il ne faut pas oublier de config votre postgres pour écouter sur toute les adresses 0.0.0.0

-D'abord, on efface les variables d'environnement dans le fichier **entrypoint.sh**

-Afficher les adresses IP de machine :

```
root@tp-docker:/home/tp-docker/Téléchargements/app# hostname -I
10.0.2.15 172.17.0.1
root@tp-docker:/home/tp-docker/Téléchargements/app#
```

-Configuration du fichier **postgresql.conf** :

on active le listennaire sur les addresses IP '0.0.0.0'

```
  GNU nano 3.2                     /etc/postgresql/11/main/postgresql.conf

external_pid_file = '/var/run/postgresql/11-main.pid'                   # writ
                                        # (change requires restart)


#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

listen_addresses = '0.0.0.0'            # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*' for
                                        # (change requires restart)
port = 5432                             # (change requires restart)
```

-Configuration du fichier **pg_hba.conf** :

Dans 'IPv4 local connections' on rajoute les addresses '172.17.0.0/16' et '10.0.2.15/32' associer respectivement aux réseau hôtes.

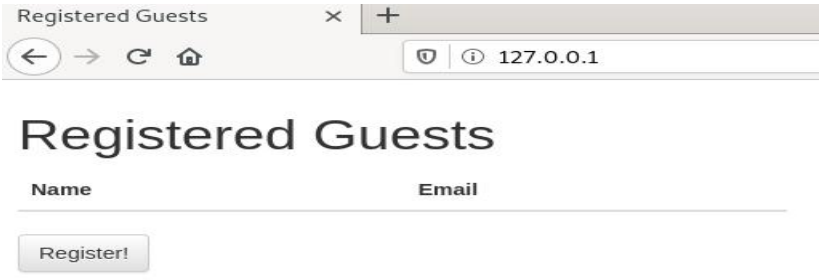
```
  GNU nano 3.2                       /etc/postgresql/11/main/pg_hba.conf

# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local   all             postgres                                peer

# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             172.17.0.0/16           md5
host    all             all             127.0.0.1/32            md5
host    all             all             10.0.2.15/32            md5
# IPv6 local connections:
host    all             all             ::1/128                 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     peer
host    replication     all             127.0.0.1/32            md5
host    replication     all             ::1/128                 md5
```

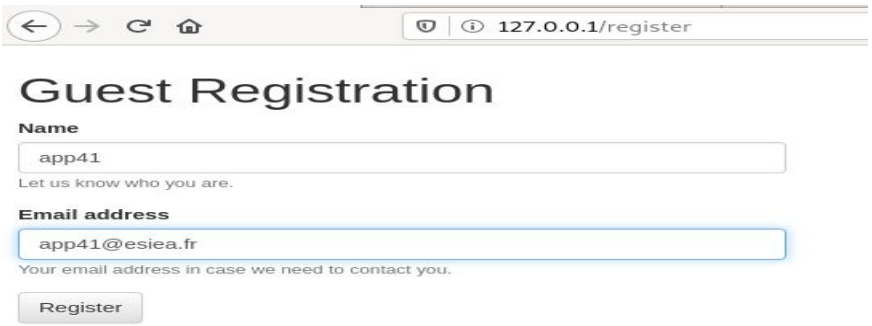
On redémarre le service de postgresql :

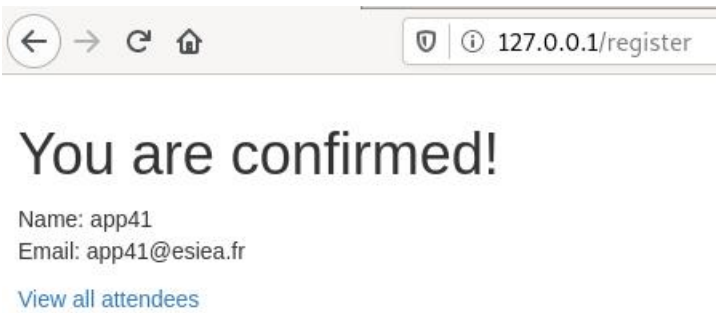
```
root@tp-docker:/home/tp-docker/Téléchargements/app# /etc/init.d/postgresql restart
[ ok ] Restarting postgresql (via systemctl): postgresql.service.
root@tp-docker:/home/tp-docker/Téléchargements/app#
```

Après Build et le Run du Dockerfile, lorsqu'on tape **127.0.0.1** sur Firefox, il répond "GET / HTTP/1.1" 200 - comme ci-dessous :

```
root@tp-docker:/home/tp-docker/Téléchargements/app# docker run -it -p 80:5000 app-41
INFO  [alembic.runtime.migration] Context impl PostgresqlImpl.
INFO  [alembic.runtime.migration] Will assume transactional DDL.
 * Serving Flask app "app.py"
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
172.17.0.1 - - [22/Dec/2020 22:45:24] "GET / HTTP/1.1" 200 -
```







4- Modification du Dockerfile pour avoir notre serveur postgres au niveau de l'image générée.

tp-docker@tp-docker: ~

Fichier Édition Affichage Rechercher Terminal Aide

GNU nano 3.2 Dockerfile

```
FROM tiangolo/uwsgi-nginx-flask:python3.6

WORKDIR /app
COPY . .

RUN apt-get update && apt-get install postgresql -y

RUN pip install -r requirements.txt
RUN cp entrypoint.sh /docker-entrypoint.sh
expose 5000

USER postgres

RUN /etc/init.d/postgresql start &&\
psql --command "CREATE USER groupe9 WITH SUPERUSER PASSWORD 'Groupe9';" &&\
createdb -O groupe9 groupe9


#ENV DBUSER=groupe9
#ENV DBNAME=groupe9
ENV DBHOST=10.0.2.15
#ENV DBPASS=groupe9


ENV FLASK_APP=app.py
#RUN chmod 750  /docker-entrypoint.sh
ENTRYPOINT ["/docker-entrypoint.sh"]
```

-Après Build et le Run du Dockerfile, lorsqu'on tape **127.0.0.1** sur Firefox, il répond "GET / HTTP/1.1" 200 et on remarque que le PostgreSQL 11 database server est bien démarré avec la statut **ok** - comme ci-dessous

```
root@tp-docker:/home/tp-docker/Téléchargements/app# docker run -it -p 80:5000 app-41
[ ok ] Starting PostgreSQL 11 database server: main.
INFO  [alembic.runtime.migration] Context impl PostgresqlImpl.
INFO  [alembic.runtime.migration] Will assume transactional DDL.
 * Serving Flask app "app.py"
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
172.17.0.1 - - [22/Dec/2020 23:23:56] "GET / HTTP/1.1" 200 -
```

Partie 2 : Docker-compose

1- Créer un fichier docker-compose permettant de mettre en place deux containers avec un pour l'application et un deuxième pour la base de données.

Voici Le code spécifique du fichier Docker-compose.yml :

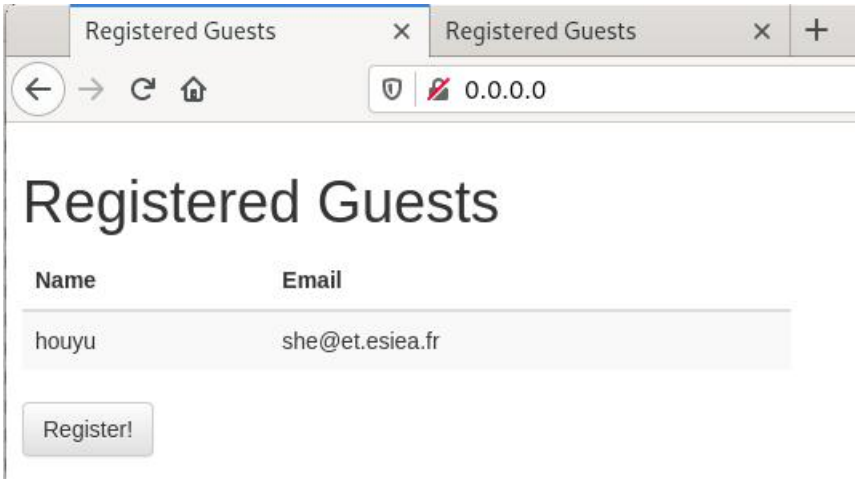
```
version: '3.3'

services:
    db:
       image: postgres
       restart: always

       environment:
         POSTGRES_USER: groupe9
         POSTGRES_PASSWORD: groupe9
         POSTGRES_DB: groupe9
       volumes:
         - /var/postgresql/data:/var/lib/postgresql

    app:
       build: .
       restart: always
       environment:
          DBNAME: groupe9
          DBUSER: groupe9
          DBPASS: groupe9
          DBHOST: db
          FLASK_APP: app.py
       ports:
          - 80:5000
       depends_on:
          - db
```

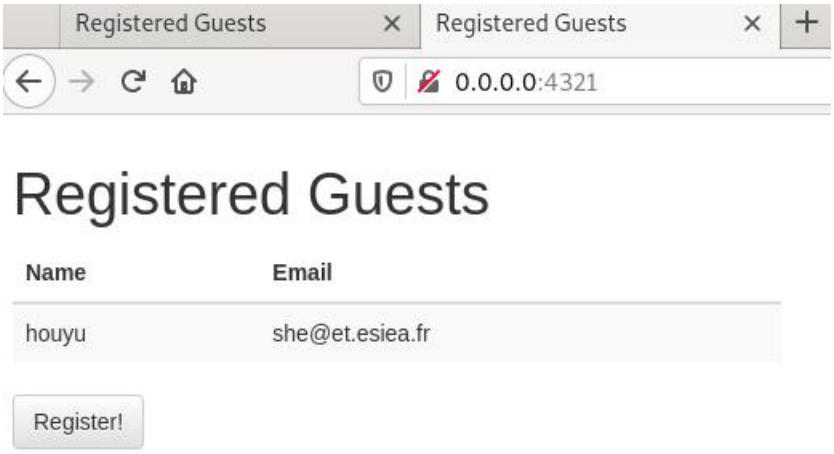
Voici le résultat d'exécution du docker-compose.yml avec l'exécution du db, app et le retour "GET / HTTP/1.1" 200

```
db_1   | 2020-12-23 15:05:14.376 UTC [63] LOG:  database system was shut down at 2020-12-23 15
4 UTC
db_1   | 2020-12-23 15:05:14.396 UTC [1] LOG:  database system is ready to accept connections
app_1  | Starting PostgreSQL 11 database server: main.
app_1  | INFO  [alembic.runtime.migration] Context impl PostgresqlImpl.
app_1  | INFO  [alembic.runtime.migration] Will assume transactional DDL.
app_1  | INFO  [alembic.runtime.migration] Running upgrade  -> 791cd7d80402, empty message
app_1  |  * Serving Flask app "app.py"
app_1  |  * Environment: production
app_1  |    WARNING: This is a development server. Do not use it in a production deployment.
app_1  |    Use a production WSGI server instead.
app_1  |  * Debug mode: off
app_1  |  * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
app_1  | 172.21.0.1 - - [23/Dec/2020 15:05:44] "GET / HTTP/1.1" 200 -
```

2- Créer un deuxième fichier docker compose permettant cette fois ci de générer deux container pour l'application avec deux accès différents au niveau de la db l'application A1 aura le droit d'écriture sur la DB l'application A2 copie de l'application A1 n'aura que les droits de lecture sur la DB

-Voici le nouveau fichier Docker-compose.yml permettant de générer deux container pour l'application avec deux accès différents au niveau de la db, l'application app1 aura le droit d'écriture sur la DB et l'application app2 n'aura que les droits de lecture sur la DB





Voici le résultat d'exécution du nouveau docker-compose.yml avec l'exécution du db, app1, app2 et le retour "GET / HTTP/1.1" 200 pour app1 et app2

```
app2_1  | Starting PostgreSQL 11 database server: main.
app1_1  | Starting PostgreSQL 11 database server: main.
app1_1  | INFO  [alembic.runtime.migration] Context impl PostgresqlImpl.
app1_1  | INFO  [alembic.runtime.migration] Will assume transactional DDL.
app1_1  | INFO  [alembic.runtime.migration] Running upgrade  -> 791cd7d80402, empty message
app2_1  | INFO  [alembic.runtime.migration] Context impl PostgresqlImpl.
app2_1  | INFO  [alembic.runtime.migration] Will assume transactional DDL.
app1_1  |  * Serving Flask app "app.py"
app1_1  |  * Environment: production
app1_1  |    WARNING: This is a development server. Do not use it in a production deployment.
app1_1  |    Use a production WSGI server instead.
app1_1  |  * Debug mode: off
app1_1  |  * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
app2_1  |  * Serving Flask app "app.py"
app2_1  |  * Environment: production
app2_1  |    WARNING: This is a development server. Do not use it in a production deployment.
app2_1  |    Use a production WSGI server instead.
app2_1  |  * Debug mode: off
app2_1  |  * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
app1_1  | 172.25.0.1 - - [23/Dec/2020 22:38:54] "GET / HTTP/1.1" 200 -
app1_1  | 172.25.0.1 - - [23/Dec/2020 22:38:55] "GET /favicon.ico HTTP/1.1" 404 -
app1_1  | 172.25.0.1 - - [23/Dec/2020 22:39:10] "GET /register HTTP/1.1" 200 -
app1_1  | 172.25.0.1 - - [23/Dec/2020 22:39:34] "POST /register HTTP/1.1" 200 -
app1_1  | 172.25.0.1 - - [23/Dec/2020 22:39:39] "GET / HTTP/1.1" 200 -
app2_1  | 172.25.0.1 - - [23/Dec/2020 22:39:54] "GET / HTTP/1.1" 200 -
```

2-1 Modification du code de App2 puisse démarrer car elle ne peut plus être utilisée vu que l'accès à la DB se fait uniquement en lecture

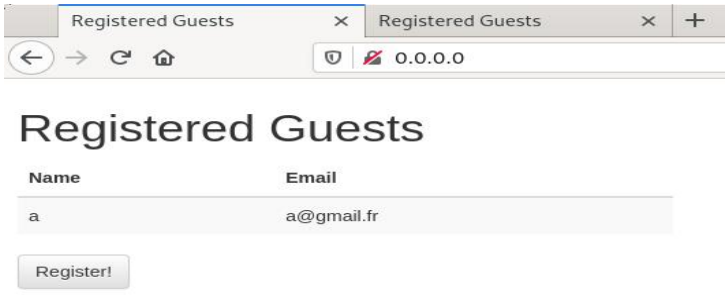
Changement du droit d'accès pour l'application app2

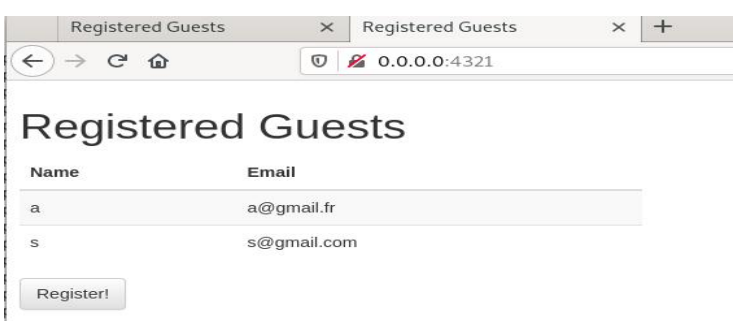
tp-docker@tp-docker: ~

Fichier Édition Affichage Rechercher Terminal Aide

GNU nano 3.2 Docker-compose.yml

```
version: '3.3'

services:
    db:
       image: postgres
       restart: always
       environment:
         POSTGRES_USER: groupe9
         POSTGRES_PASSWORD: groupe9
         POSTGRES_DB: groupe9
       volumes:
         - /var/postgresql/data:/var/lib/postgresql

    app1:
       build: .
       restart: always
       environment:
          DBNAME: groupe9
          DBUSER: groupe9
          DBPASS: groupe9
          DBHOST: db
          FLASK_APP: app.py
       depends_on:
          - db
       ports:
         - 80:5000
       command: chmod +w /var/postgresql/data:/var/lib/postgresql

    app2:
       build: .
       restart: always
       environment:
          DBNAME: groupe9
          DBUSER: groupe9
          DBPASS: groupe9
          DBHOST: db
          FLASK_APP: app.py
       depends_on:
          - db
       ports:
         - 4321:5000
       command: chmod +w /var/postgresql/data:/var/lib/postgresql
```





Voici le résultat d'exécution du nouveau docker-compose.yml avec l'exécution du db, app1, app2 et le retour "GET / HTTP/1.1" 200 pour app1 et app2

```
app2_1  | Starting PostgreSQL 11 database server: main.
app1_1  | Starting PostgreSQL 11 database server: main.
app2_1  | INFO  [alembic.runtime.migration] Context impl PostgresqlImpl.
app2_1  | INFO  [alembic.runtime.migration] Will assume transactional DDL.
app2_1  | INFO  [alembic.runtime.migration] Running upgrade  -> 791cd7d80402, empty message
app1_1  | INFO  [alembic.runtime.migration] Context impl PostgresqlImpl.
app1_1  | INFO  [alembic.runtime.migration] Will assume transactional DDL.
app2_1  |  * Serving Flask app "app.py"
app2_1  |  * Environment: production
app2_1  |    WARNING: This is a development server. Do not use it in a production deployment.
app2_1  |    Use a production WSGI server instead.
app2_1  |  * Debug mode: off
app1_1  |  * Serving Flask app "app.py"
app1_1  |  * Environment: production
app1_1  |    WARNING: This is a development server. Do not use it in a production deployment.
app1_1  |    Use a production WSGI server instead.
app1_1  |  * Debug mode: off
app2_1  |  * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
app1_1  |  * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
app1_1  | 172.26.0.1 - - [23/Dec/2020 23:18:38] "GET / HTTP/1.1" 200 -
app2_1  | 172.26.0.1 - - [23/Dec/2020 23:18:45] "GET / HTTP/1.1" 200 -
app1_1  | 172.26.0.1 - - [23/Dec/2020 23:19:11] "GET /register HTTP/1.1" 200 -
app1_1  | 172.26.0.1 - - [23/Dec/2020 23:19:32] "POST /register HTTP/1.1" 200 -
app1_1  | 172.26.0.1 - - [23/Dec/2020 23:19:36] "GET / HTTP/1.1" 200 -
app2_1  | 172.26.0.1 - - [23/Dec/2020 23:19:48] "GET /register HTTP/1.1" 200 -
app2_1  | 172.26.0.1 - - [23/Dec/2020 23:20:05] "POST /register HTTP/1.1" 200 -
app2_1  | 172.26.0.1 - - [23/Dec/2020 23:20:09] "GET / HTTP/1.1" 200 -
```

Partie 3: Docker-registry

1- Mettrez en place localement un docker registry

```
root@tp-docker:/home/tp-docker/Téléchargements/app# docker run -d -p 5000:5000 --restart=always --name registry-groupe9 app-41
af01f6fb1d5b52837a4b883d8abf35fec284d79c4416395e3e290777adff036e
```

2- Tager les image généré précédemment et envoyer les au niveau de votre registry local

```
root@tp-docker:/home/tp-docker/Téléchargements/app# docker tag app-41 localhost:5000/app-41
root@tp-docker:/home/tp-docker/Téléchargements/app#
root@tp-docker:/home/tp-docker/Téléchargements/app# docker push localhost:5000/app-41
The push refers to repository [localhost:5000/app-41]
Get http://localhost:5000/v2/: EOF
```

Vérification

```
root@tp-docker:/home/tp-docker/Téléchargements/app# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                          PORTS               NAMES
af01f6fb1d5b        app-41              "/docker-entrypoint.…"   19 minutes ago      Restarting (1) 29 seconds ago                       registry-groupe9
root@tp-docker:/home/tp-docker/Téléchargements/app#
```

```
root@tp-docker:/home/tp-docker/Téléchargements/app# docker tag app_app1 0.0.0.0:5000/app_app1
root@tp-docker:/home/tp-docker/Téléchargements/app# docker push 0.0.0.0:5000/app_app1
The push refers to repository [0.0.0.0:5000/app_app1]
Get https://0.0.0.0:5000/v2/: EOF
root@tp-docker:/home/tp-docker/Téléchargements/app#
root@tp-docker:/home/tp-docker/Téléchargements/app# docker tag app_app2 localhost:5000/app_app2
root@tp-docker:/home/tp-docker/Téléchargements/app# docker push localhost:5000/app_app2
The push refers to repository [localhost:5000/app_app2]
Get http://localhost:5000/v2/: read tcp [::1]:44062->[::1]:5000: read: connection reset by peer
root@tp-docker:/home/tp-docker/Téléchargements/app# docker push localhost:5000/app_app2
The push refers to repository [localhost:5000/app_app2]
Get http://localhost:5000/v2/: EOF
```

3- **Reconstruire les images après un légère modification du code et taguer les avec un nouveau tag permettant de les différencier de la première version**

4- Push les nouvelles image sur votre registry local

On a tagué et envoyé les images app_app1 et app_app2 avec des nouveaux tag respectivement v1 et v2 voir image ci-dessous

```
root@tp-docker:/home/tp-docker/Téléchargements/app# docker tag app_app1 localhost:5000/app_app1:v1
root@tp-docker:/home/tp-docker/Téléchargements/app# docker push localhost:5000/app_app1:v1
The push refers to repository [localhost:5000/app_app1]
Get http://localhost:5000/v2/: read tcp [::1]:44078->[::1]:5000: read: connection reset by peer
root@tp-docker:/home/tp-docker/Téléchargements/app# docker push localhost:5000/app_app1:v1
The push refers to repository [localhost:5000/app_app1]
Get http://localhost:5000/v2/: EOF
root@tp-docker:/home/tp-docker/Téléchargements/app# docker tag app_app2 localhost:5000/app_app2:v2
root@tp-docker:/home/tp-docker/Téléchargements/app# docker push localhost:5000/app_app2:v2
The push refers to repository [localhost:5000/app_app2]
Get http://localhost:5000/v2/: dial tcp [::1]:5000: connect: connection refused
root@tp-docker:/home/tp-docker/Téléchargements/app# docker push localhost:5000/app_app2:v2
The push refers to repository [localhost:5000/app_app2]
Get http://localhost:5000/v2/: EOF
```

5- Supprimer de votre registry local et les premières image tagués

-suppression du première image tagué app_app1 (TAG = latest)

```
root@tp-docker:/home/tp-docker/Téléchargements/app# docker image ls
REPOSITORY                    TAG                 IMAGE ID            CREATED             SIZE
0.0.0.0:5000/app_app1         latest              15151324e164        2 hours ago         1.26GB
0.0.0.0:5000/app_app2         latest              15151324e164        2 hours ago         1.26GB
80:5000/app_app1              latest              15151324e164        2 hours ago         1.26GB
app_app1                      latest              15151324e164        2 hours ago         1.26GB
app_app2                      latest              15151324e164        2 hours ago         1.26GB
localhost:5000/app_app1       latest              15151324e164        2 hours ago         1.26GB
localhost:5000/app_app1       v1                  15151324e164        2 hours ago         1.26GB
localhost:5000/app_app2       latest              15151324e164        2 hours ago         1.26GB
localhost:5000/app_app2       v2                  15151324e164        2 hours ago         1.26GB
app-41                        latest              d0da12b75b7b        4 hours ago         1.26GB
localhost:5000/app-41         latest              d0da12b75b7b        4 hours ago         1.26GB
5000:5000/app1                latest              290237f0c3b5        7 hours ago         1.26GB
80:5000/app1                  latest              290237f0c3b5        7 hours ago         1.26GB
app1                          latest              290237f0c3b5        7 hours ago         1.26GB
localhost:5000/app1           latest              290237f0c3b5        7 hours ago         1.26GB
app_app                       latest              b34d019d6002        8 hours ago         1.26GB
tiangolo/uwsgi-nginx-flask    python3.6           997d8bb5f569        4 days ago          944MB
registry                      2                   678dfa38fcfa        6 days ago          26.2MB
postgres                      latest              a6cd86e1dfce        12 days ago         314MB
root@tp-docker:/home/tp-docker/Téléchargements/app#
root@tp-docker:/home/tp-docker/Téléchargements/app# docker image rm app_app1:latest
Untagged: app_app1:latest
root@tp-docker:/home/tp-docker/Téléchargements/app# docker image ls
REPOSITORY                    TAG                 IMAGE ID            CREATED             SIZE
0.0.0.0:5000/app_app1         latest              15151324e164        2 hours ago         1.26GB
0.0.0.0:5000/app_app2         latest              15151324e164        2 hours ago         1.26GB
80:5000/app_app1              latest              15151324e164        2 hours ago         1.26GB
app_app2                      latest              15151324e164        2 hours ago         1.26GB
localhost:5000/app_app1       latest              15151324e164        2 hours ago         1.26GB
localhost:5000/app_app1       v1                  15151324e164        2 hours ago         1.26GB
localhost:5000/app_app2       latest              15151324e164        2 hours ago         1.26GB
localhost:5000/app_app2       v2                  15151324e164        2 hours ago         1.26GB
app-41                        latest              d0da12b75b7b        4 hours ago         1.26GB
localhost:5000/app-41         latest              d0da12b75b7b        4 hours ago         1.26GB
5000:5000/app1                latest              290237f0c3b5        7 hours ago         1.26GB
80:5000/app1                  latest              290237f0c3b5        7 hours ago         1.26GB
app1                          latest              290237f0c3b5        7 hours ago         1.26GB
localhost:5000/app1           latest              290237f0c3b5        7 hours ago         1.26GB
app_app                       latest              b34d019d6002        8 hours ago         1.26GB
tiangolo/uwsgi-nginx-flask    python3.6           997d8bb5f569        4 days ago          944MB
registry                      2                   678dfa38fcfa        6 days ago          26.2MB
postgres                      latest              a6cd86e1dfce        12 days ago         314MB
root@tp-docker:/home/tp-docker/Téléchargements/app#
```

-Stop et suppression du registry local nommé registry-groupe9

```
root@tp-docker:/home/tp-docker/Téléchargements/app# docker container ps

CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                          PORTS               NAMES
af01f6fb1d5b        app-41              "/docker-entrypoint.…"   About an hour ago   Restarting (1) 58 seconds ago                       registry-groupe9
root@tp-docker:/home/tp-docker/Téléchargements/app#
root@tp-docker:/home/tp-docker/Téléchargements/app# docker container stop registry-groupe9
registry-groupe9
root@tp-docker:/home/tp-docker/Téléchargements/app# docker container rm registry-groupe9
registry-groupe9
root@tp-docker:/home/tp-docker/Téléchargements/app# docker container ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
root@tp-docker:/home/tp-docker/Téléchargements/app#
```

6- déclencher le garbage collector au niveau de votre registry pour nettoyer les blob non référencés (suite à la suppression du tag)

```
root@tp-docker:/home/tp-docker/Téléchargements/app# docker system prune
WARNING! This will remove:
        - all stopped containers
        - all networks not used by at least one container
        - all dangling images
        - all dangling build cache
Are you sure you want to continue? [y/N] y
Deleted Containers:
a0b45ec5ab53bcefeb3ba153ba1c4791da0e3ec0c3fe8b3cfaeb514dc141632e
7a404708a796e3051b896f73f27ddca09e77a26de27a28210276ac579dec2933
1e33fdd418e11a2c73bd7724f3ba01a115f3ee954dec03629bf80bba202032f3
f16fc4380637ad7f7ee74920414537c1f58278802f74143e7cee4f0cacd0bcd2
6065f6a0f5014fcc63b02a99d3d71742f7513466ffccdb57d799d1104cbd2d53

Deleted Images:
deleted: sha256:e3aa1a213bbf5d3f5338233b794b2f1459bd9953f6d55fcf2cc130f4f21ecd11
deleted: sha256:2561c9f46e6a7c63573a89b81e6dd2a053076ea8a8b448b2c776cb5ae0cd4665
deleted: sha256:ae1dab5d031366c3c5382d3bffcf958b0a19ae20858d4a3be778e6234d946f46
deleted: sha256:34c90ea369d4ad4c39aedbb4f18d0250e17c08e7094ce3145c96d7e5f04d84f8
deleted: sha256:8b50aee35d347c1bfd80185e1f3425e0d0f5dd057d93a9c3e5d6981ddcbab7a3
deleted: sha256:c323db074b232aae2533a0d38d355afaf11cc03e9b76efdd8c9718a5b114c599
deleted: sha256:d21b4894000457302b8195c637a33dd6e727e610da51469e84fb14a978dfc954
deleted: sha256:01ee5901dd21bce73895be94416df48fd7187b8c7129b0c4bf2527da11eee472
deleted: sha256:64eb77804d51aecf181e338a78e28994211605b2354708e7e7e1079d939ad792
deleted: sha256:786ed1fd9b3b818e095e168c2a675ea0ff866826f840199303ec1298520b1655
deleted: sha256:8ba8cc05dab69558042c2849a98668ea3251f14feccf312b57726b4f2d4d9e8b
deleted: sha256:a83ccfdc7d5ec3ecc070bb1d38b7f0295f51c23fd045c12a2d9b8b62edec1594
deleted: sha256:a580ed80a0907e8e5879975f8ff0fa78aeb038aeb2ffd3862808f10cecc4e48d

Total reclaimed space: 9.202GB
root@tp-docker:/home/tp-docker/Téléchargements/app#
```