

Leo Bourse, Paco Hameline, Luca donneau, Kylian Boin

EXAMEN FINAL DOCKER

1. Conteneurisation de l'application web

1) Image de base : python:3.6-alpine

```
[vagrant@master opt]$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
httpd	latest	fe8735c23ec5	2 weeks ago	145MB
ubuntu	18.04	71eaf13299f4	2 weeks ago	63.1MB
python	3.6-alpine	3a9e80fa4606	11 months ago	40.7MB
ociotec/tetris	latest	456a4a9c606e	2 years ago	26MB
pengbai/docker-supermario	latest	077ed8c9236a	3 years ago	646MB

2) Définir le répertoire /opt comme répertoire de travail

```
[vagrant@master opt]$
```

3) Installer le module Flask version 1.1.2 à l'aide de pip install flask==1.1.2

```
(venv) [vagrant@master ~]$ pip install flask==1.1.2
Collecting flask==1.1.2
  Downloading https://files.pythonhosted.org/packages/f2/28/2a03252dfb9ebf377f40fba6a7841b47083260bf8bd8e737b0c6952df83f/Flask-1.1.2-py2.py3-none-any.whl (94kB)
    100% |#####| 102kB 3.0MB/s
Collecting Jinja2>=2.10.1 (from flask==1.1.2)
  Downloading https://files.pythonhosted.org/packages/20/9a/e5d9ec41927401e41aea8af6d16e78b5e612bca4699d417f646a9610a076/Jinja2-3.0.3-py3-none-any.whl (133kB)
    100% |#####| 143kB 2.8MB/s
Collecting itsdangerous>=0.24 (from flask==1.1.2)
```

4) Exposer le port 8080 qui est celui utilisé par défaut par l'application

```
(venv) [vagrant@master opt]$ docker run -p 8080:8069 --name odoo --link db:db -t odoo
```

5) Créer les variables d'environnement **ODOO_URL** et **PGADMIN_URL** afin de permettre la définition de ces url lors du lancement du container

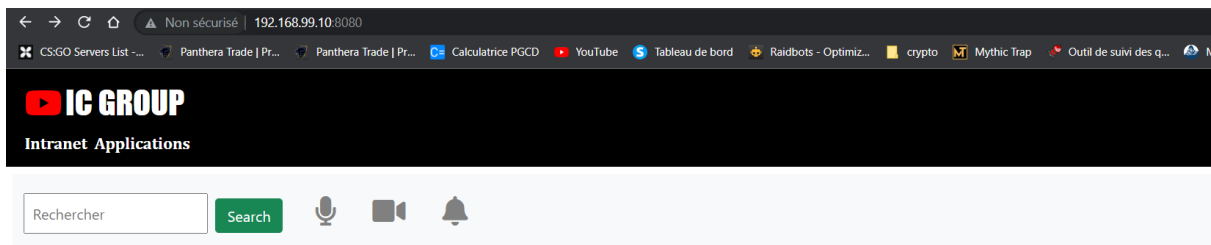
```
FROM python:3.6-alpine
LABEL maintainer="leotest"
WORKDIR /opt
RUN pip install flask
ADD . /opt/
VOLUME /opt
EXPOSE 8080
ENV ODOO_URL="http://192.168.99.10:8081/"
ENV PGADMIN_URL="http://192.168.99.10:10/"
ENTRYPOINT [ "python" ]
CMD [ "app.py" ]
```

Docker build -t ic-webapp:1.0 .

docker run -tif --name ic-webappoffi ic-webapp:1.0

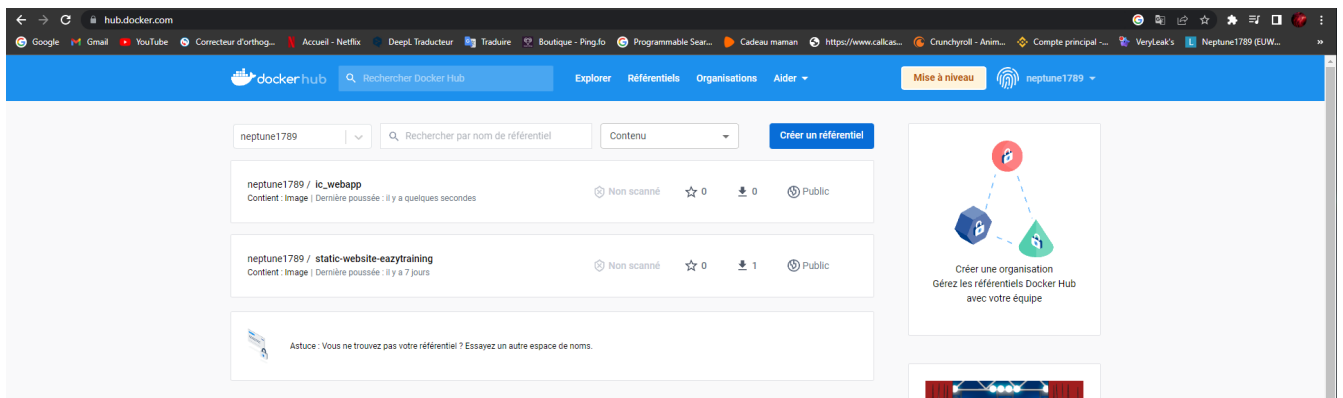
docker run -d -p 8080:8080 ic_webapp:1.0

6) Lancer l'application app.py dans le ENTRYPOINT grâce à la commande python.



2. Docker Registry

- 1) Mettez en place un registre privé avec [docker registry](#).
- 3) Une fois le test terminé, supprimez le container test et poussez votre image sur votre registre privé Docker, et aussi sur le registre [dockerhub](#).



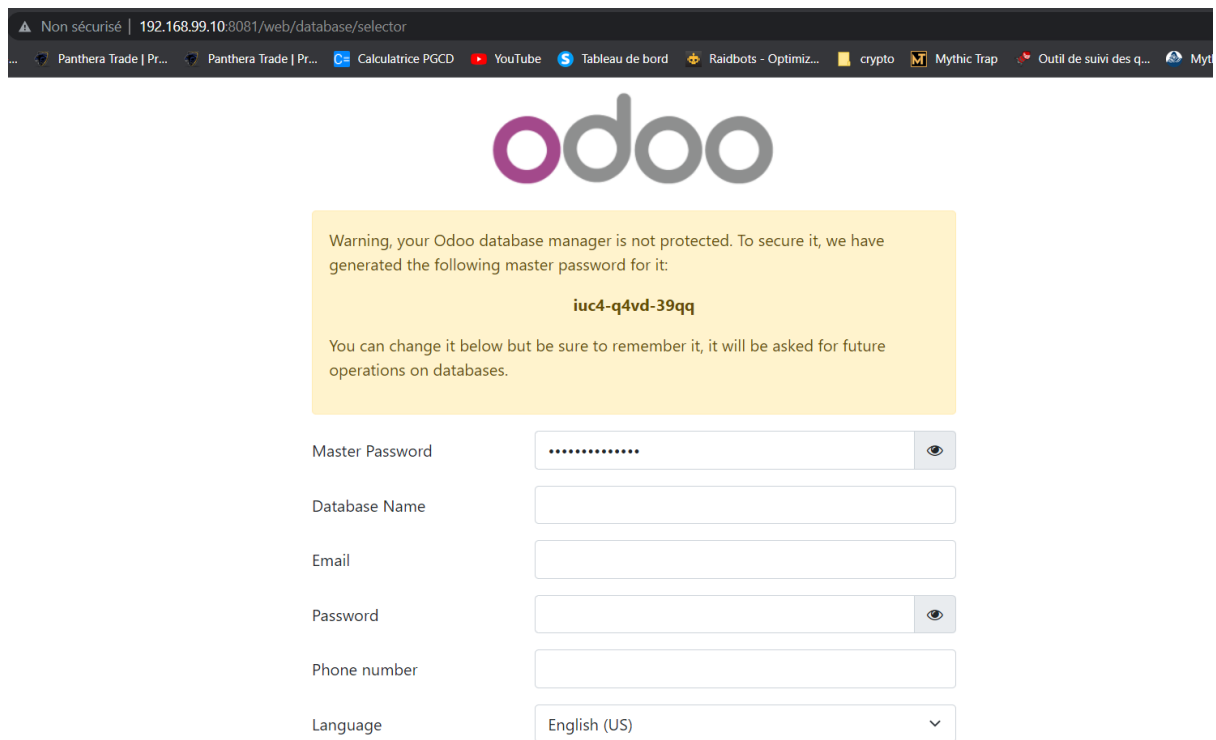
```
[vagrant@master docker-exams-1]$ docker tag ic-webapp:1.0 neptune1789/ic_webapp
```

```
[vagrant@master docker-exams-1]$ docker push neptune1789/ic-webapp
```

```
[vagrant@master docker-exams-1]$ docker pull neptune1789/ic_webapp
```

3. docker-compose

2) L'application **odoo** qui stocke ses informations dans la base postgres



Non sécurisé | 192.168.99.10:8081/web/database/selector

Panthera Trade | Pr... Panthera Trade | Pr... Calculatrice PGCD YouTube Tableau de bord Raidbots - Optimiz... crypto Mythic Trap Outil de suivi des q... Myt

odoo

Warning, your Odoo database manager is not protected. To secure it, we have generated the following master password for it:

iuc4-q4vd-39qq

You can change it below but be sure to remember it, it will be asked for future operations on databases.

Master Password

Database Name

Email

Password

Phone number

Language

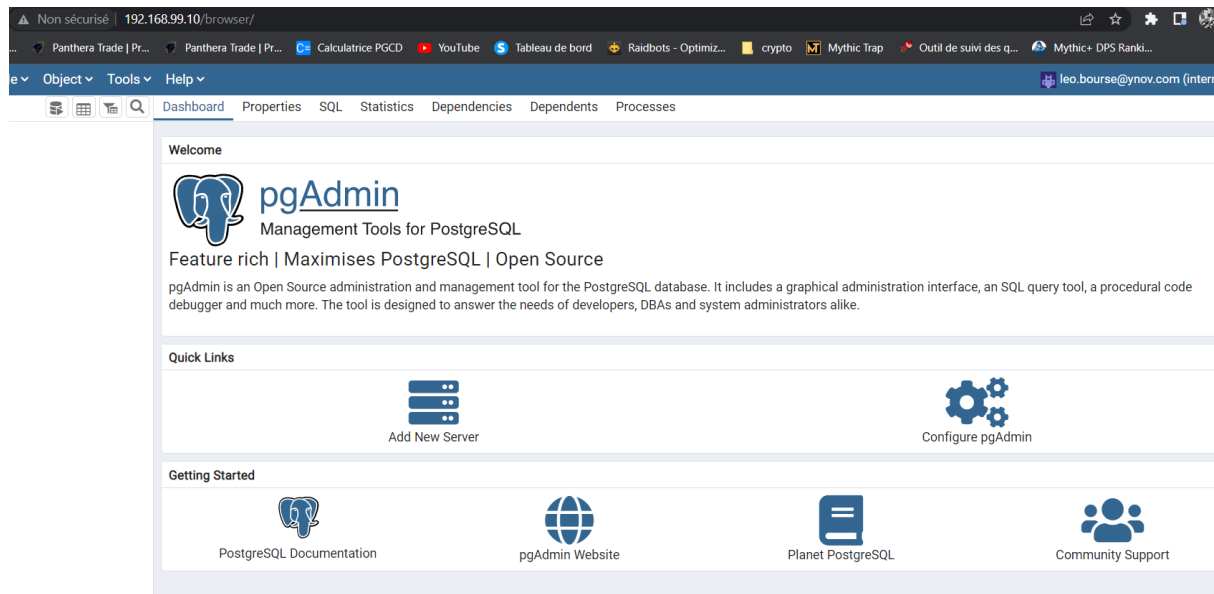
```
docker run -d -e POSTGRES_USER=odoo -e POSTGRES_PASSWORD=odoo -e POSTGRES_DB=postgres --name db postgres:13
```

```
docker run -p 8081:8069 --name odoo --link db:db -t odoo
```

```
docker start -a odoo
```

```
docker stop odoo
```

3) L'application [pgadmin](#) permettant de visualiser la base postgres dans une IHM.



- `docker pull dpage/pgadmin4`
- `docker run -p 80:80 \ -e`
- `'PGADMIN_DEFAULT_EMAIL=leo.bourse@ynov.com' \ -e`
- `'PGADMIN_DEFAULT_PASSWORD=123456' \ -d dpage/pgadmin4`