

8INF911: Mini-Projets-Docker

Ce dépôt contient les ressources des quatres mini-projets réalisés par Léo VANSIMAY et Gaël LECONTE dans le cadre du cours **Architecture Cloud et Méthodes DevOps, outils pour le Cloud-Gaming** pour le trimestre d'été 2022 à l'UQAC.

TODO List

- ☒ Sujet 1
- ☒ Sujet 2
- ☒ Sujet 3
- ☐ Sujet 4

Ressources Md

- [Cheatsheet](#)
- [Style en plus \(icônes réseaux et tout\)](#)

Table des matières

1. [VISUALISATION DE DONNEES IOT – FASTAPI / INFLUXDB](#)
2. [DISPONIBILITE DE SERVICE](#)
3. [HEBERGEMENT WEB \(VERSION NGINX\)](#)
4. [HEBERGEMENT WEB \(VERSION TRAEFIK\)](#)

1. VISUALISATION DE DONNEES IOT – FASTAPI / INFLUXDB

Notre idée initiale était de créer deux images, une pour utiliser l'API *pyflux*, et l'autre pour lancer *InfluxDB* et le configurer automatiquement. Mais nous avons remarqué qu'une image *InfluxDB* était disponible sur [DockerHub](#), et que si certaines variables d'environnements sont configurées, alors le service pourra se configurer tout seul. On fixe ainsi la variable `DOCKER_INFLUXDB_INIT_MODE` sur `setup` dans le fichier `docker-compose.yml`.

Pour des raisons de sécurité, aucune information confidentielle n'est renseignée dans ce fichier, nous utilisons à la place le fichier `.env` qui est par défaut chargé par docker-compose pour fixer les variables d'environnement.

N-B: Dans le contexte pédagogique, le `.env` est présent sur le dépôt git, mais dans une situation de production, ce fichier ne doit pas être communiqué, étant donné qu'il sert à configurer un produit pour un client.

Fonctionnement du projet

Une fois la commande `docker-compose up` lancée, le port 3000 du conteneur *pyflux* est exposé et mappé sur le port 3000 de la machine hôte, on peut donc joindre l'API en envoyant des requêtes à l'adresse <http://localhost:3000>. Pour populer la base de données, on se sert du script `script.sh`:

```
#!/bin/bash
while true
do
    v=$(shuf -i 5-20 -n 1)
    curl -X POST http://localhost:3000/climate/temp/$v
    sleep 1
done
```

```
server-9@server-9: /projects/IS ./script.sh
{"measure":["climate"],"field":["temp"],"value":19.0}{"measure":["climate"],"field":["temp"],"value":13.0}{"measure":["climate"],"field":["temp"],"value":20.0}{"measure":["climate"],"field":["temp"],"value":13.0}{"measure":["climate"],"field":["temp"],"value":7.0}{"measure":["climate"],"field":["temp"],"value":17.0}{"measure":["climate"],"field":["temp"],"value":14.0}{"measure":["climate"],"field":["temp"],"value":17.0}{"measure":["climate"],"field":["temp"],"value":11.0}{"measure":["climate"],"field":["temp"],"value":10.0}{"measure":["climate"],"field":["temp"],"value":15.0}{"measure":["climate"],"field":["temp"],"value":9
```

2. DISPONIBILITE DE SERVICE

```
server-9@server-9:~/projets/2/reverseproxy$ curl http://localhost
Hostname: whoami
IP: 127.0.0.1
IP: 192.168.160.7
RemoteAddr: 192.168.160.6:38214
GET / HTTP/1.1
Host: localhost
User-Agent: curl/7.81.0
Accept: */*
Connection: close
X-Forwarded-For: 192.168.160.1
X-Real-IP: 192.168.160.1

server-9@server-9:~/projets/2/reverseproxy$ curl http://localhost
Hostname: whoami
IP: 127.0.0.1
IP: 192.168.160.3
RemoteAddr: 192.168.160.6:56454
GET / HTTP/1.1
Host: localhost
User-Agent: curl/7.81.0
Accept: */*
Connection: close
X-Forwarded-For: 192.168.160.1
X-Real-IP: 192.168.160.1

server-9@server-9:~/projets/2/reverseproxy$ curl http://localhost
Hostname: whoami
IP: 127.0.0.1
IP: 192.168.160.5
RemoteAddr: 192.168.160.6:51188
GET / HTTP/1.1
Host: localhost
User-Agent: curl/7.81.0
Accept: */*
Connection: close
```

3. HEBERGEMENT WEB (VERSION NGINX)

```
root@server-9:/home/server-9/projets/3# docker-compose ps
```

Name	Command	State	Ports
3_certbot_1	/bin/sh -c while true; do ...	Up	
3_nginx_1	/docker-entrypoint.sh nginx ...	Up	0.0.0.0:443->443/tcp, :::443->443/tcp, 0.0.0.0:80->80/tcp, :::80->80/tcp

4. HEBERGEMENT WEB (VERSION TRAEFIK)