

Rapport Projet conteneurisation & virtualisation

Introduction :

Dans le cadre de notre cours sur la virtualisation et la conteneurisation, nous avons entrepris un projet visant à explorer l'utilisation de conteneurs pour le déploiement d'une application. L'application, intitulée "**HumansBestFriend App - CATs or DOGs?**", a été conçue pour illustrer les principes fondamentaux de la conteneurisation et la gestion d'applications multi-conteneurs avec Docker.

Infrastructure Utilisée :

Pour le développement de notre projet, nous avons opté pour l'utilisation d'ESXi, une solution de virtualisation. ESXi nous a offert la possibilité de créer une machine virtuelle basée sur l'image Ubuntu. Cette machine virtuelle a servi de terrain d'expérimentation pour nos travaux liés à la conteneurisation.

Logiciels Utilisées :

Docker :

Nous avons installé Docker sur notre machine virtuelle Ubuntu. Docker nous a permis d'encapsuler notre application et ses dépendances dans des conteneurs légers et portables.

Docker Compose :

Docker Compose a été intégré pour simplifier la gestion de notre application multi-conteneur. Ce gestionnaire d'orchestration a facilité le déploiement et la configuration de nos services, offrant une solution cohérente pour le développement, les tests et le déploiement.

Services de l'application :

Service Worker :

Dépendant de redis et db services.
Intégré au réseau back-tier.

Service Vote :

Un volume a été mappé à /usr/local/app à l'intérieur du conteneur.
Expose le port 80 (peut être accessible à l'extérieur via le port 5002).
Connecté aux réseaux front-tier et back-tier.
Health check configuré pour vérifier la disponibilité du service.

Service Seed-Data :

Intégré au réseau front-tier.
Utilise un Dockerfile spécifique pour générer des données de test.

Service Result :

Dépendant du service db.
Un volume mappé à /usr/local/app.
Expose le port 80 (accessible à l'extérieur via le port 5001).
Utilise un Dockerfile spécifique pour le développement avec nodemon.

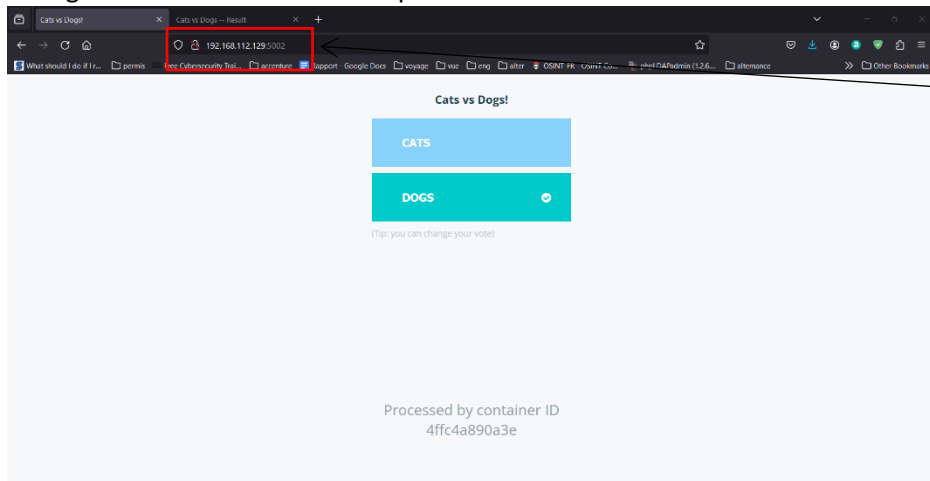
Tout d'abord il faut se connecter en SSH à ma VM :

```
matteo@k8s-master: ~/projet × + v
Microsoft Windows [version 10.0.22621.2861]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\matte>ssh matteo@192.168.112.129
matteo@192.168.112.129's password:
```

Déploiement des conteneurs :

Nous avons créé des Dockerfiles distincts pour chaque service afin de définir les étapes de construction des images Docker. Ces images encapsulent les dépendances, le code source, et les configurations nécessaires à chaque service.



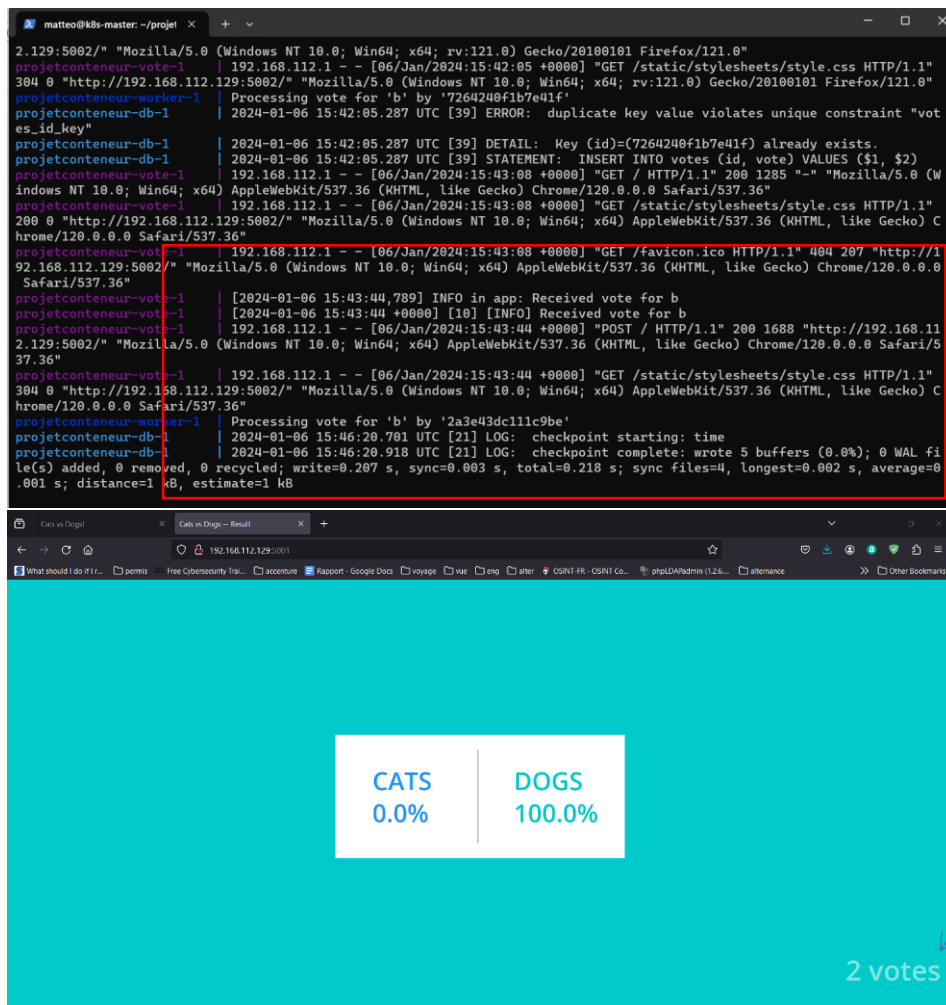
Comme on voit ici se connecte au service vote de l'app qui est exposé sur le port 5002

Partie front-end en python

```
matteo@k8s-master: ~/projet × + v
projctconteneur-vote-1 [2024-01-06 15:41:27 +0000] [8] [INFO] Booting worker with pid: 8
projctconteneur-vote-1 [2024-01-06 15:41:27 +0000] [10] [INFO] Booting worker with pid: 10
projctconteneur-vote-1 [2024-01-06 15:41:27 +0000] [9] [INFO] Booting worker with pid: 9
projctconteneur-worker-1 Connected to db
projctconteneur-worker-1 Found redis at 172.20.0.3
projctconteneur-worker-1 Connecting to redis
projctconteneur-result-1 Sat, 06 Jan 2024 15:41:28 GMT body-parser deprecated undefined extended: provide extended op
tion at server.js:67:17
projctconteneur-result-1 App running on port 80
projctconteneur-result-1 Connected to db
projctconteneur-vote-1 [2024-01-06 15:42:05.097] INFO in app: Received vote for b
projctconteneur-vote-1 [2024-01-06 15:42:05 +0000] [7] [INFO] Received vote for b
projctconteneur-vote-1 192.168.112.1 - - [06/Jan/2024:15:42:05 +0000] "POST / HTTP/1.1" 200 1688 "http://192.168.11
2.129:5002/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:121.0) Gecko/20100101 Firefox/121.0"
projctconteneur-vote-1 192.168.112.1 - - [06/Jan/2024:15:42:05 +0000] "GET /static/stylesheets/style.css HTTP/1.1"
304 0 "http://192.168.112.129:5002/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:121.0) Gecko/20100101 Firefox/121.0"
projctconteneur-worker-1 Processing vote for 'b' by '7264240f1b7e41f'
projctconteneur-db-1 [2024-01-06 15:42:05.287 UTC [39] ERROR: duplicate key value violates unique constraint "votes_id_key"
projctconteneur-db-1 [2024-01-06 15:42:05.287 UTC [39] DETAIL: Key (id)=(7264240f1b7e41f) already exists.
projctconteneur-db-1 [2024-01-06 15:42:05.287 UTC [39] STATEMENT: INSERT INTO votes (id, vote) VALUES ($1, $2)
projctconteneur-vote-1 192.168.112.1 - - [06/Jan/2024:15:43:08 +0000] "GET / HTTP/1.1" 200 1285 "-" "Mozilla/5.0 (W
indows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36"
projctconteneur-vote-1 192.168.112.1 - - [06/Jan/2024:15:43:08 +0000] "GET /static/stylesheets/style.css HTTP/1.1"
200 0 "http://192.168.112.129:5002/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) C
hrome/120.0.0.0 Safari/537.36"
projctconteneur-vote-1 192.168.112.1 - - [06/Jan/2024:15:43:08 +0000] "GET /favicon.ico HTTP/1.1" 404 207 "http://1
92.168.112.129:5002/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0
Safari/537.36"
```

LOGS

Nous avons une ERROR lorsqu'on essaie de revoter avec le même navigateur



Ici on voit qu'il enregistre bien le vote dans la base de données

Cette page sur le port 5001 nous montre le nombre de vote ainsi que le pourcentage pour les chats et les chiens c'est le service result

Afin de pouvoir gérer et exécuter les conteneurs correctement nous avons créé 2 fichiers YAML.

- docker-compose.build.yml -> contient nos services et les configurations de déploiements pour chacun d'entre eux avec notamment leur port où ils sont exposés, leurs dépendances ainsi que leur network.
- compose.yml -> Ce fichier contient les images pour les conteneurs que nous avons chargés, et notamment redis et postgres utilisés pour la base de données de notre application.

Conclusion :

Ce projet a été une opportunité enrichissante d'explorer les aspects avancés de la conteneurisation en mettant en œuvre une application réelle et complexe. L'utilisation d'outils tels que Docker et Docker Compose a permis un déploiement efficace et reproductible de notre application, renforçant ainsi nos compétences dans le domaine de la virtualisation et de la gestion de conteneurs.

Par Mattéo Picherit, Maxime Pensivy, Benjamin Perez