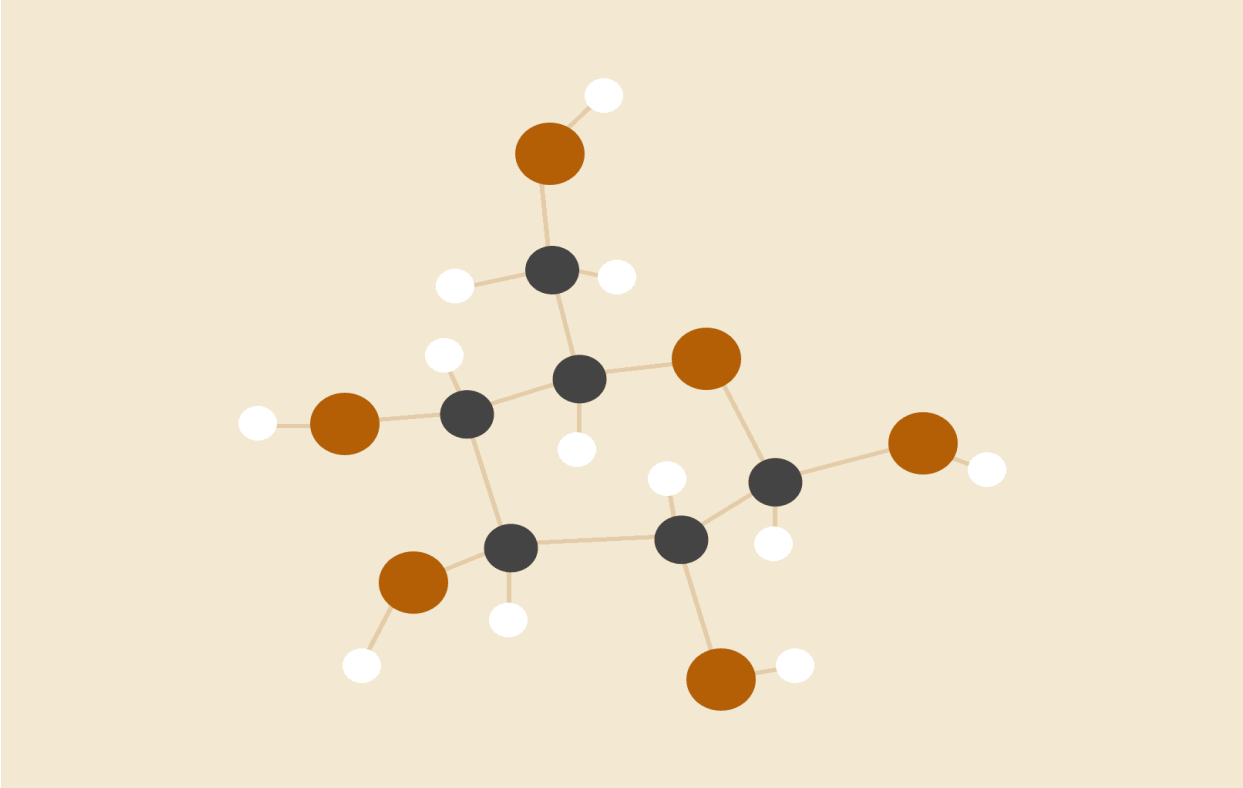
esiea

# RAPPORT PROJET DE VIRTUALISATION



**Audrey Jordan SOUOP**

2023-2024

TC3 - 4A - Classe 49

# INTRODUCTION

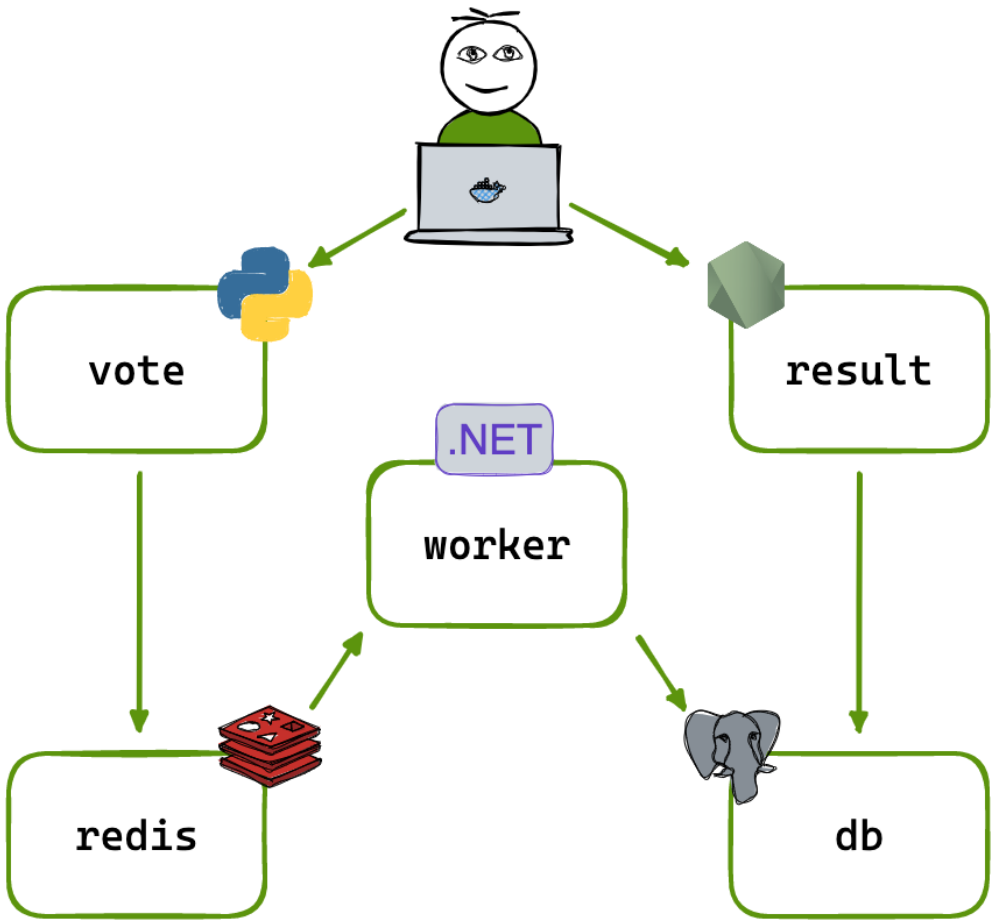
La virtualisation est une technologie informatique révolutionnaire qui permet de créer des environnements informatiques virtuels, distincts des systèmes physiques sous-jacents. Elle permet l'exécution simultanée de plusieurs systèmes d'exploitation et applications sur une seule machine physique. Cette approche offre une flexibilité inégalée, améliore l'utilisation des ressources matérielles et facilite la gestion et la sécurité des infrastructures informatiques.

## HYPOTHÈSE

L'application HumansBestFriend n'accepte qu'un seul vote par navigateur client. Elle n'enregistre pas de votes supplémentaires si un vote a déjà été soumis depuis un client.

Ceci n'est pas un exemple d'application distribuée parfaitement architecturée et conçue... C'est simplement un exemple simple des différents types de composants et de langages que l'on pourrait rencontrer (files d'attente, données persistantes, etc.) et de la manière de les gérer dans Docker à un niveau basique.

## ARCHITECTURE

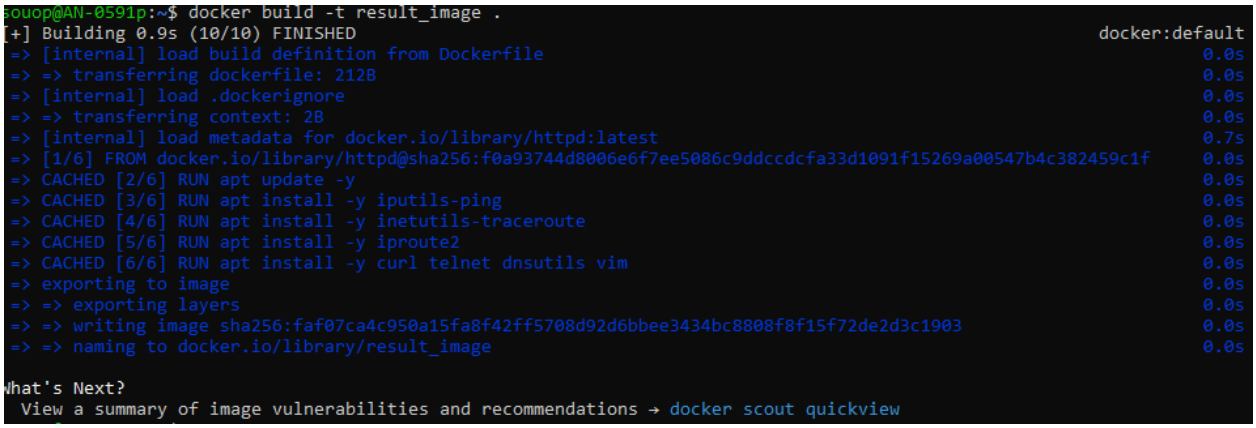


## PROCÉDURE

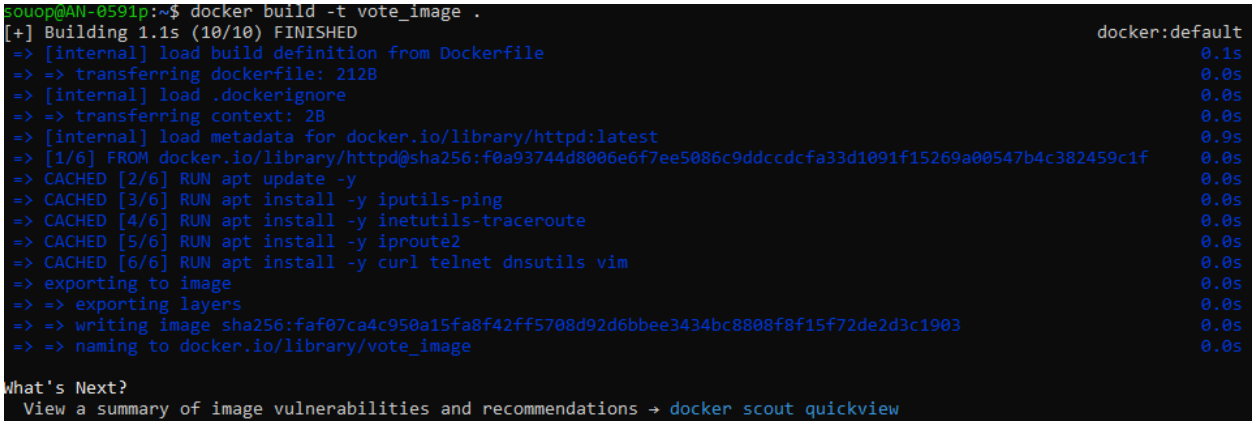
Afin de mener à bien notre projet et de bien construire nos fichiers d'exécution, nous aurons besoin de respecter méticuleusement un certain nombre de règles avec Docker afin de ne créer aucun bug.

Pour débuter, nous allons commencer par builder chacune des images de base fourni:

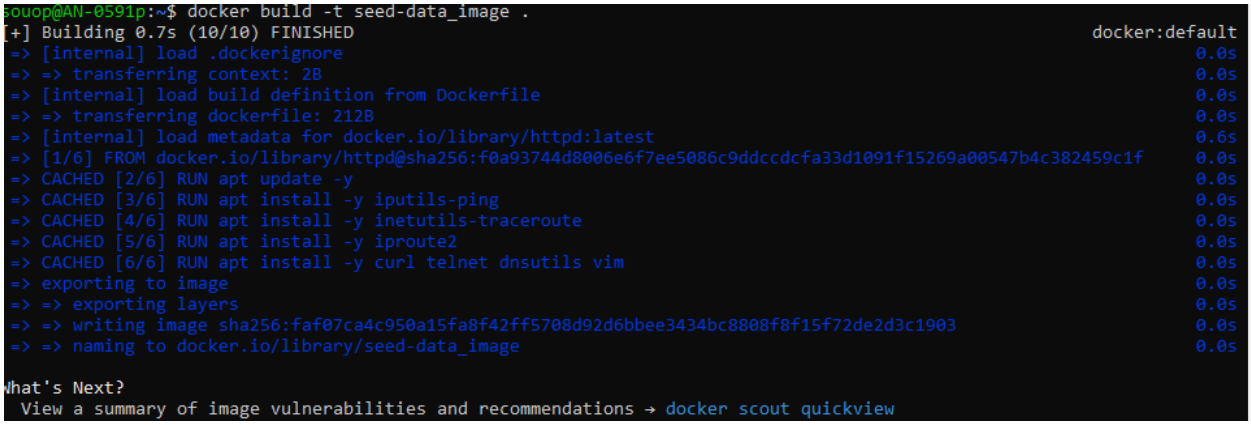
On commence par builder l'image de result avec la commande **docker build result_image.** On obtient le résultat suivant:



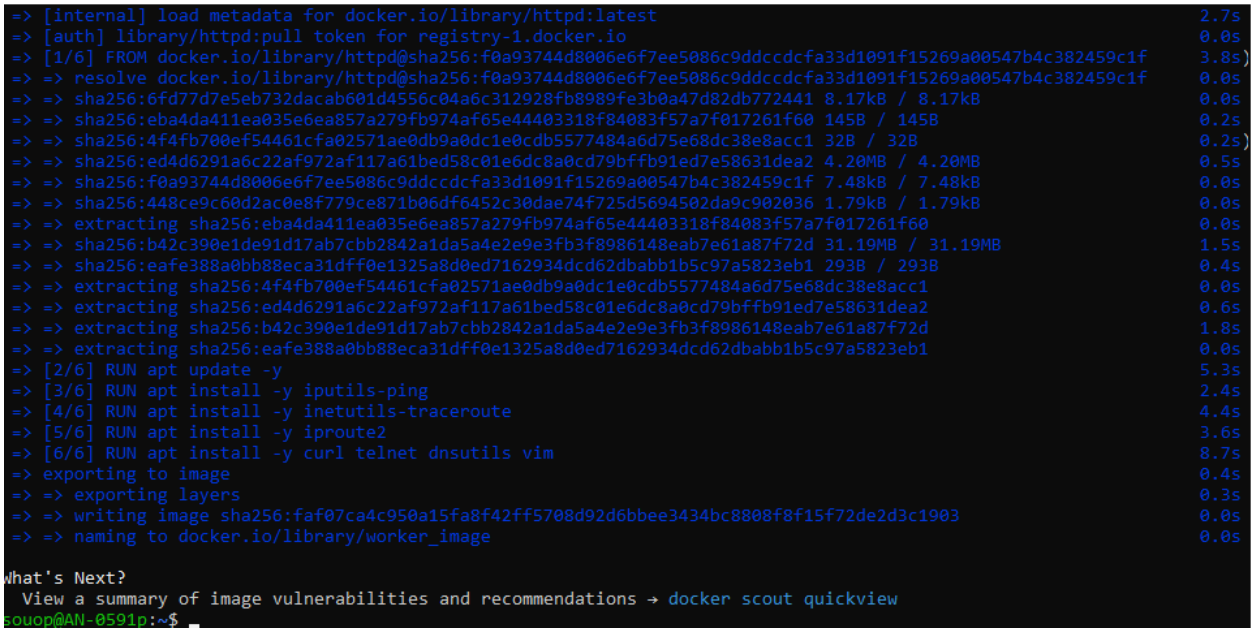

Après cela, on build l'image de vote avec la commande **docker build vote_image.** On obtient ceci:



Après cela, et pour que notre exécution se passe bien sans aucune interruption ni bug, nous avons apporté des modifications et lancé l'exécution de seed-data grâce à la commande **docker build seed-data_image.** le résultat est tel que vous pouvez voir ci dessous



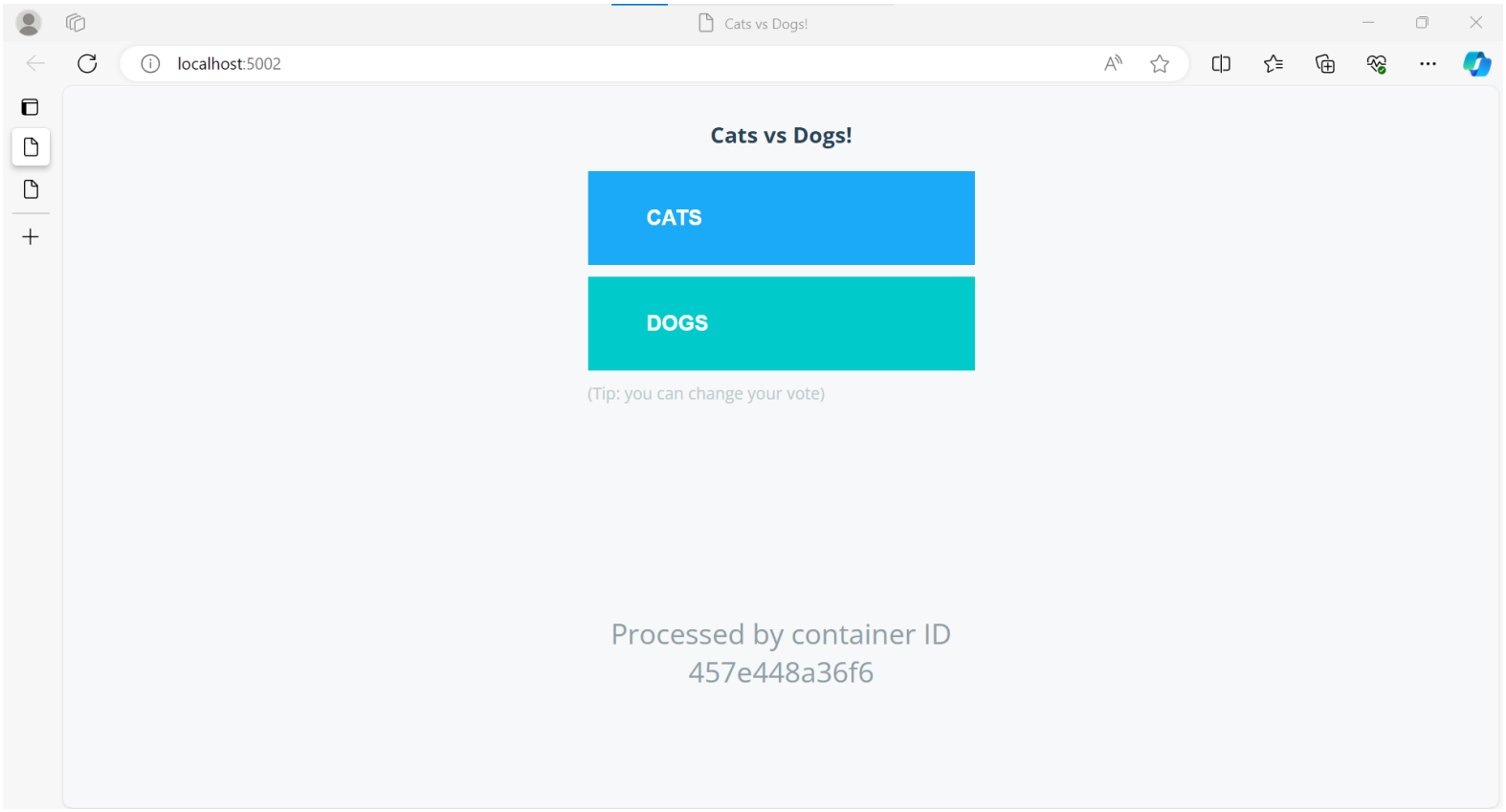

Et pour finir nous avons builder aussi l'image du worker avec la commande **docker build worker_image.** Nous avons obtenu ceci:



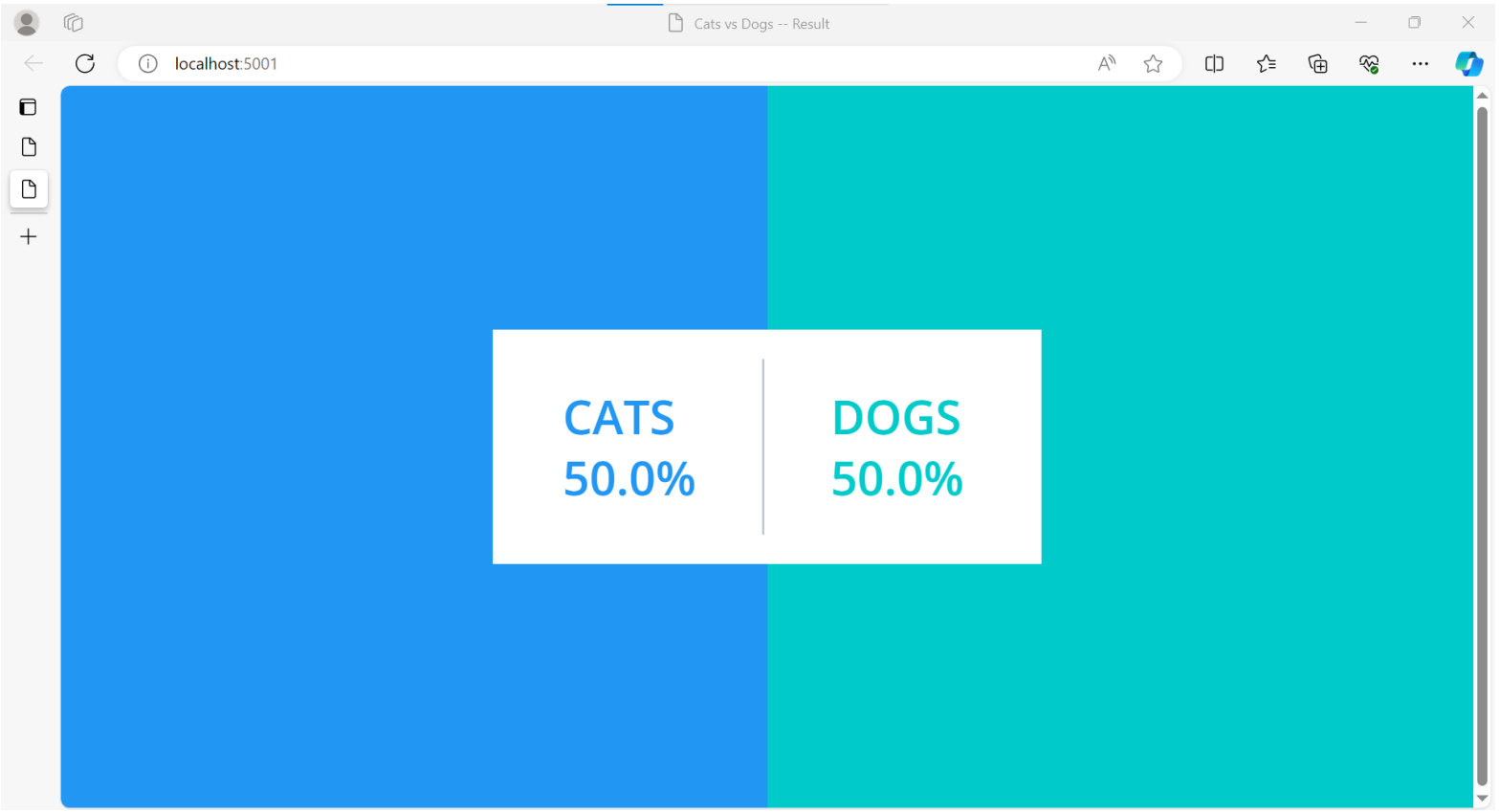
## RÉSULTATS

Après toutes ces différentes opérations, nous avons crée le fichier YAML compose.yml avec notre commande **docker-compose up** et le résultat était concédé dans compose.yml

En mettant en marche nos conteneurs, nous avons eu ces différentes sorties. au port 5002 nous avons eu ceci:



En seconde sortie sur le port 5001, nous avons eu ceci comme sortie:



## CONCLUSION

L’objectif ici n’était évidemment pas de faire une trop longue présentation sur le Projet de virtualisation, ce qui serait certes très intéressant, mais plutôt de poser les bases techniques et le vocabulaire employé dans ce système .

## RÉFÉRENCES


1. https://kubernetes.io/docs/setup/production-environment/tools/

2. https://docs.docker.com/engine/install/ubuntu/

3. https://docs.docker.com/compose/gettingstarted/

4. https://kubernetes.io/docs/reference/kubectl/cheatsheet/

5. https://github.com/JeffSouop/HUMAN-BEST-FRIENDS-APP-SOUOP-4A-49.git