

Projet Virtualisation et conteneurisation

Classe 49

SARGER Quentin
ELZEIN Fadi
LU William
MOUNGUENGUI Smol Kennedy



—
HumansBestFriend app ?

CATs or DOGs ?

—
Jean-Pascal MEWENEMESSE

Sommaire

Introduction	1
1) Présentation du projet	
2) Tâches	3
Architecture	4
1) Création du docker.....	5
2) Déploiement de l'application	6
Conclusion	6
1) Problèmes rencontrés	7
2) Récapitulatif du projet	8

Introduction

1) Présentation du projet

Ce projet vise concevoir une application distribuée, "HumansBestFriend app? CATs or DOGs?", à travers l'utilisation ingénieuse de multiples conteneurs Docker.

Pour atteindre cet objectif, nous plongerons au cœur d'une machine virtuelle dédiée, exécutant Docker et Docker Compose.

Nous aurons besoin des langages de programmation Python, Node.js, et .NET, accompagnées de Redis pour la messagerie et Postgres pour le stockage des données.

2) Taches

Nous commencerons par la création d'un fichier nommé `docker-compose.build.yml`, responsable exclusivement de la construction des images d'application à partir du contenu des fichiers Dockerfile fournis. Pour le service "worker", des dépendances clés sont établies envers Redis et la base de données, tandis que le service "vote" fait appel à des volumes pour stockage et écoute sur le port 80.

Chaque composant du projet, qu'il s'agisse du service de vote, du générateur de données de démarrage, ou des résultats, sera configuré à travers des fichiers Dockerfile et des paramètres spécifiques.

Le déploiement global du projet est orchestré à l'aide d'un fichier `compose.yml`, veillant à ce que les images référencées proviennent du Dockerfile privé.

Architecture

1) Création du docker

Après avoir récupéré tous les fichiers nécessaires, la phase suivante consistait à élaborer le fichier `docker-compose.yml`, un élément central permettant d'orchestrer de manière harmonieuse l'ensemble des services individuels du projet. Cela revient à définir comment ces différents services interagiront au sein de l'environnement de conteneurisation. Une pratique essentielle dans cette orchestration consiste à lier chaque service à des réseaux spécifiques, créant ainsi des environnements isolés et favorisant une communication fluide entre les conteneurs. Ces réseaux définissent des espaces virtuels où les services peuvent échanger des données de manière sécurisée et efficace.

En structurant correctement ces configurations, le fichier `docker-compose.yml` devient celui qui coordonne harmonieusement l'ensemble du projet, assurant une exécution cohérente et efficace de l'application distribuée à travers l'environnement Docker.

2) Déploiement de l'application

Une fois tous les fichiers et configurations prêts, le déploiement se simplifie grâce à la commande `'docker-compose up'`. Cette commande initie la création des conteneurs en se basant sur les directives du fichier `docker-compose.yml`.

Chaque service, comme le "Worker" pour les tâches asynchrones, le "Vote" pour la gestion des votes, le "Seed-Data" pour l'injection des données initiales, et le "Result" pour l'affichage des résultats, est instancié en tant que conteneur autonome. La configuration préalable des réseaux spécifiques garantit une communication fluide entre ces conteneurs, assurant ainsi une interaction sans accroc des services.

Le déploiement inclut également la gestion des dépendances. Les healthchecks veillent à ce que chaque service soit opérationnel avant d'être sollicité par d'autres parties de l'application. Les volumes partagés assurent un stockage persistant des données, même en cas de redémarrage des conteneurs, préservant ainsi la cohérence des informations.

Une fois le déploiement terminé, l'application est accessible à travers les ports spécifiés dans le fichier `docker-compose.yml`. Ces ports permettent l'accès aux différentes parties de l'application depuis l'extérieur des conteneurs Docker, rendant l'application fonctionnelle et prête à être utilisée.

Conclusion

1) Problèmes rencontrés

Au cours de la phase de compilation du fichier `docker-compose.build.yml`, notre équipe a rencontré des problèmes significatifs qui ont entravé la progression du projet.

Plus précisément, la complexité a émergé lors de la construction des images d'application à partir des directives du Dockerfile. Des erreurs dans la configuration et des incompatibilités entre les différentes parties du projet ont entravé le processus de compilation, conduisant à des échecs répétés dans la création des images Docker.

Ces difficultés ont généré des retards substantiels dans la mise en œuvre du projet, car la création réussie des images est une étape critique avant le déploiement. Il a été nécessaire de revisiter et de réajuster les instructions du Dockerfile, de résoudre les conflits de dépendances, et d'harmoniser les configurations entre les différents services de l'application distribuée.

2) Récapitulatif du projet

Ce projet nous a aidé à développer nos compétences en matière de conception, de déploiement, et de sécurité d'applications reposant sur des microservices. La maîtrise des interactions complexes entre les services ainsi que la gestion efficace des flux de données ont émergé comme des piliers essentiels de notre succès. Cela témoigne de notre compréhension approfondie des défis et des opportunités inhérents à la virtualisation et à la conteneurisation.

Ce travail a non seulement consolidé notre expertise technique, mais il a également ouvert des horizons vers une approche plus agile et innovante dans le domaine de la conception et de l'exploitation d'applications distribuées.

Bien que le projet n'ait pas été mis en œuvre dans son intégralité en raison des problèmes de compilation du `docker-compose.build.yml`, il a néanmoins servi de terrain d'apprentissage précieux. Ces défis nous ont permis d'affiner nos compétences dans le domaine de la conteneurisation, renforçant notre compréhension des complexités liées au déploiement d'applications distribuées à l'aide de Docker.

Cette expérience enrichissante ouvre la voie à des améliorations futures et souligne l'importance d'une approche agile et collaborative pour surmonter les obstacles rencontrés dans des projets technologiques complexes.