Rapport devOps

Table des matières

[Build de l’image 1](#_Toc211938010)

[On vérifie que ça run 1](#_Toc211938011)

[Question 1.1 1](#_Toc211938012)

[On lance Postgres 2](#_Toc211938013)

[On lance adminer 2](#_Toc211938014)

[Adminer et Postgres run 2](#_Toc211938015)

[On arrive sur Adminer 2](#_Toc211938016)

[Dockerfile Main.java 3](#_Toc211938017)

[On build puis on run Main.java 3](#_Toc211938018)

[On modifie le dockerfile 4](#_Toc211938019)

[On build le nouveau dockerfile 4](#_Toc211938020)

[On run et on accède à la nouvelle page 4](#_Toc211938021)

[On ajoute le docker compose 6](#_Toc211938022)

[Quand on fait Docker compose up -d 7](#_Toc211938023)

[Les containers se lancent et l’API tourne 7](#_Toc211938024)

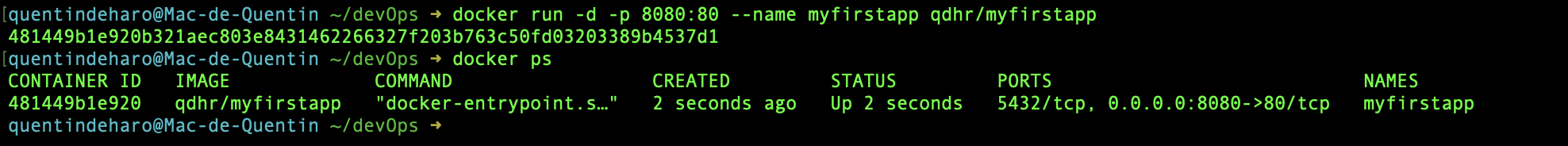
Docker

# Build de l’image

Une image contenant capture d’écran, texte

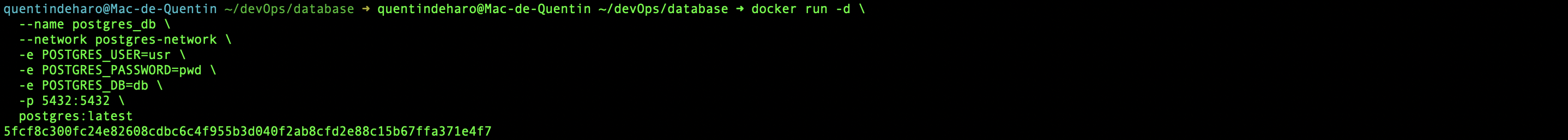
Le contenu généré par l’IA peut être incorrect.

# On vérifie que ça run

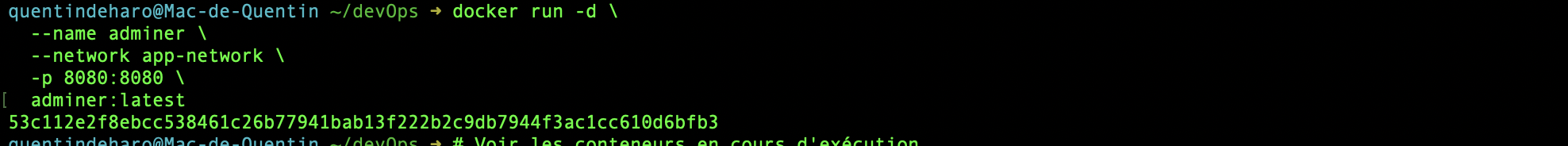


1-1)Le Dockerfile définit comment construire l'application. Les variables d'environnement définissent comment la configurer. Il faut séparer les deux.

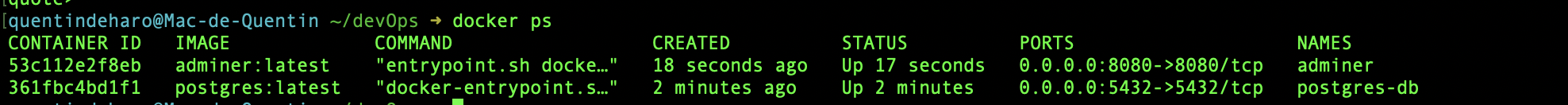
# On lance Postgres



# On lance adminer



# Adminer et Postgres run



# On arrive sur Adminer



1-2) Un volume permet à vos données de survivre aux redémarrages, aux reconstructions et même aux mises à jour d'images du conteneur.

1-3) Nous avons besoin d'un volume pour notre conteneur PostgreSQL afin de garantir la persistance des données de la base de données même si le conteneur est arrêté, supprimé ou reconstruit.

# Dockerfile Main.java

Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Le contenu généré par l’IA peut être incorrect.

# On build puis on run Main.java

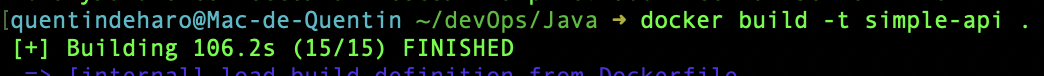
Une image contenant capture d’écran, texte

Le contenu généré par l’IA peut être incorrect.

# On modifie le dockerfile

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

On build le nouveau dockerfile   


# On run et on accède à la nouvelle page

Une image contenant texte, capture d’écran, horloge, Police

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, Police, logiciel

Le contenu généré par l’IA peut être incorrect.

1-4) On compile avec une grosse image mais on déploie avec une petite image en copiant juste le JAR compilé, ce qui réduit la taille finale et améliore la sécurité.

Une image contenant texte, Police, capture d’écran

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

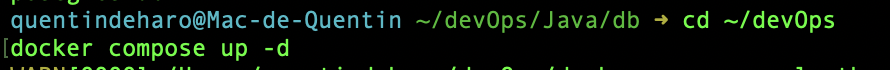
1-5) Un reverse proxy sert de "portier" : au lieu d'exposer directement ton application backend (port 8080), tout passe par le proxy (port 80), ce qui permet de sécuriser, cacher et gérer plus facilement l'accès à ton API.

# On ajoute le docker compose

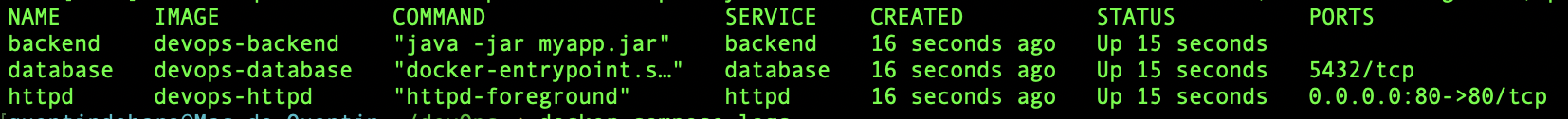
Une image contenant texte, capture d’écran, logiciel

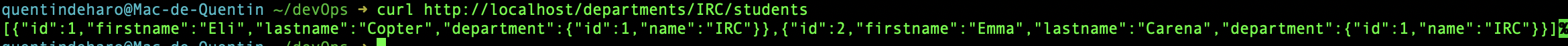
Le contenu généré par l’IA peut être incorrect.

# Quand on fait Docker compose up -d



# Les containers se lancent et l’API tourne



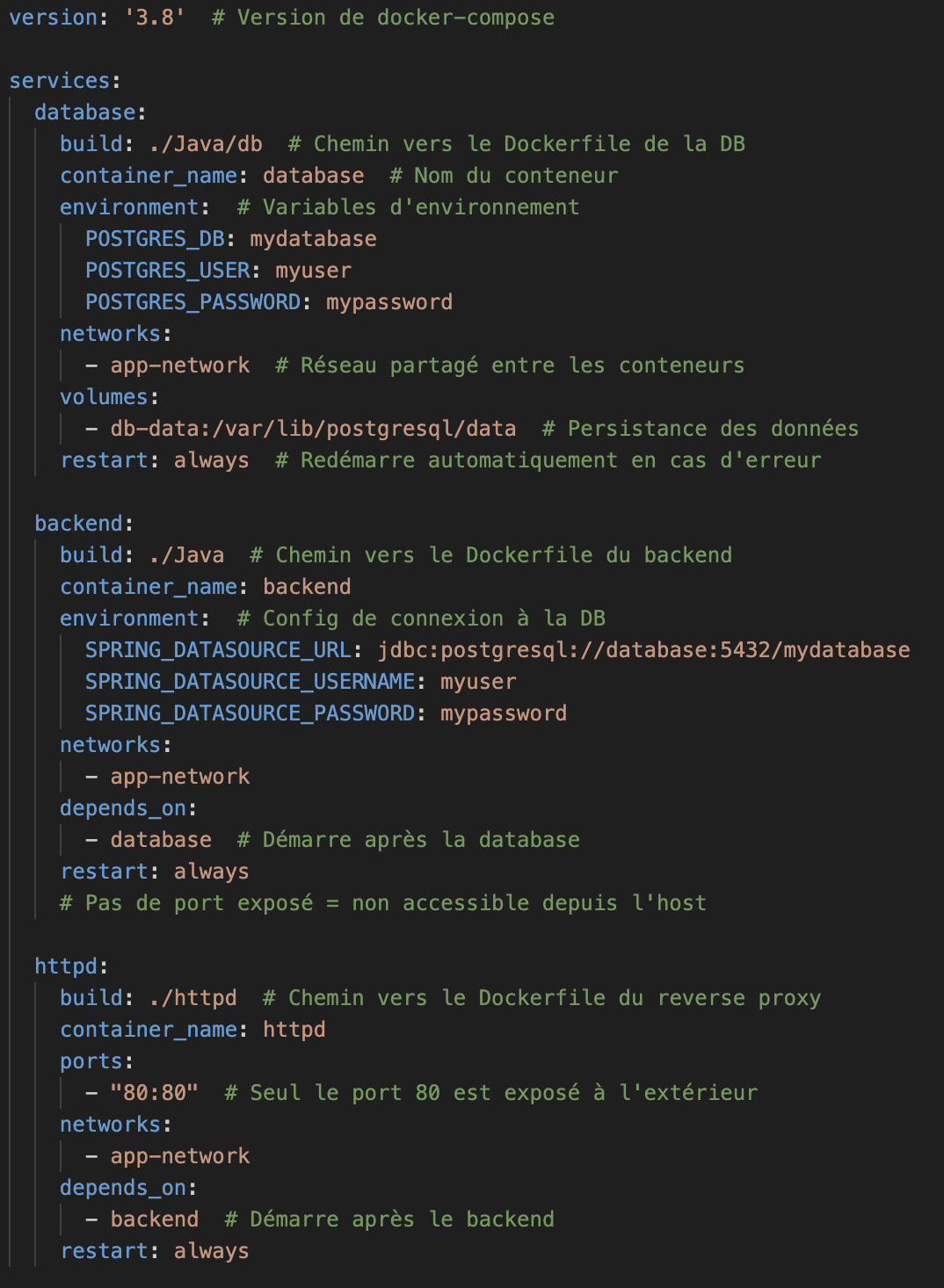


1-6) Docker-compose permet de démarrer, arrêter et gérer plusieurs conteneurs en une seule commande au lieu de les lancer un par un manuellement, ce qui simplifie énormément le développement et le déploiement.

1-7)

|  |  |
| --- | --- |
| Démarrer tous les services | docker compose up -d |
| Arrêter tous les services | docker compose down |
| Voir les logs | docker compose logs -f |
| Voir l'état des services | docker compose ps |
| Rebuild et redémarrer | docker compose up -d --build |
| Arrêter et supprimer tout (incluant les volumes) | docker compose down -v |
| Redémarrer un service spécifique | docker compose restart backend |
| Voir les logs d'un service spécifique | docker compose logs -f backend |

1-8)



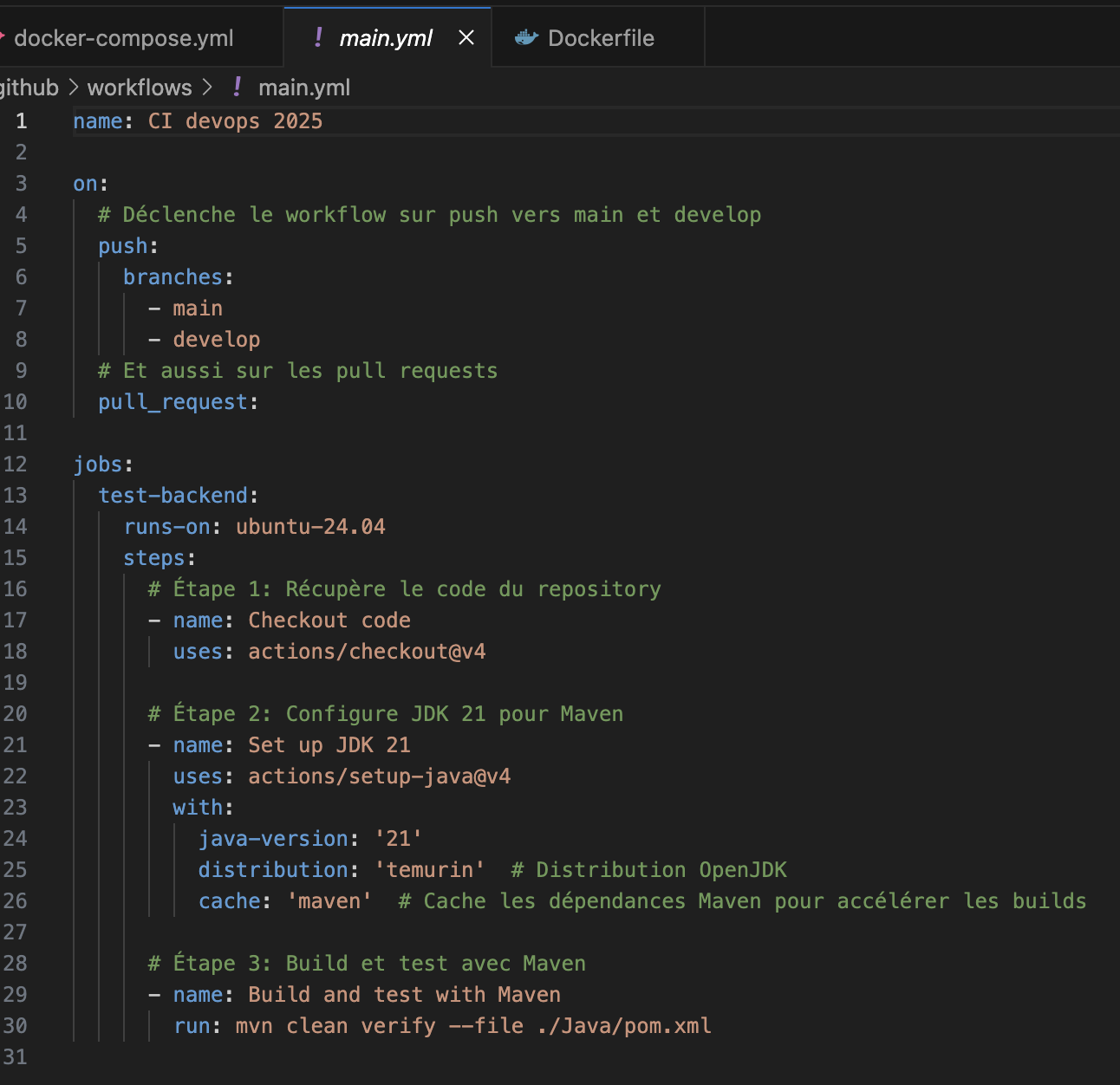
Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

GitHub Actions

2-1) Testcontainers est une librairie Java qui permet de lancer des conteneurs Docker temporaires pendant l'exécution des tests.

# On crée un main.yml pour gérer le github actions



# Lorsque l’on se rend sur le repository le GitHub actions se lance

Une image contenant capture d’écran, texte

Le contenu généré par l’IA peut être incorrect.

2-2) Les variables sécurisées (secrets) protègent les informations sensibles comme les mots de passe, tokens et credentials. Elles ne sont jamais affichées dans les logs et ne peuvent pas être lues par des personnes non autorisées, ce qui évite que quelqu'un vole tes identifiants pour publier du code malveillant sur ton Docker Hub.

2-3) needs: test-backend garantit que les images Docker ne sont buildées et pushées que si les tests passent. Sans cette ligne, les images pourraient être publiées même si le code est cassé ! Cela évite de déployer du code bugué en production.

2-4) Pusher les images Docker permet de les partager avec l'équipe, de les déployer sur différents environnements (dev, staging, production), et de garder un historique des versions. Sans ça, chaque machine devrait rebuilder les images localement, ce qui serait lent et inefficace.

Une image contenant texte, logiciel, Logiciel multimédia, Logiciel de graphisme

Le contenu généré par l’IA peut être incorrect.