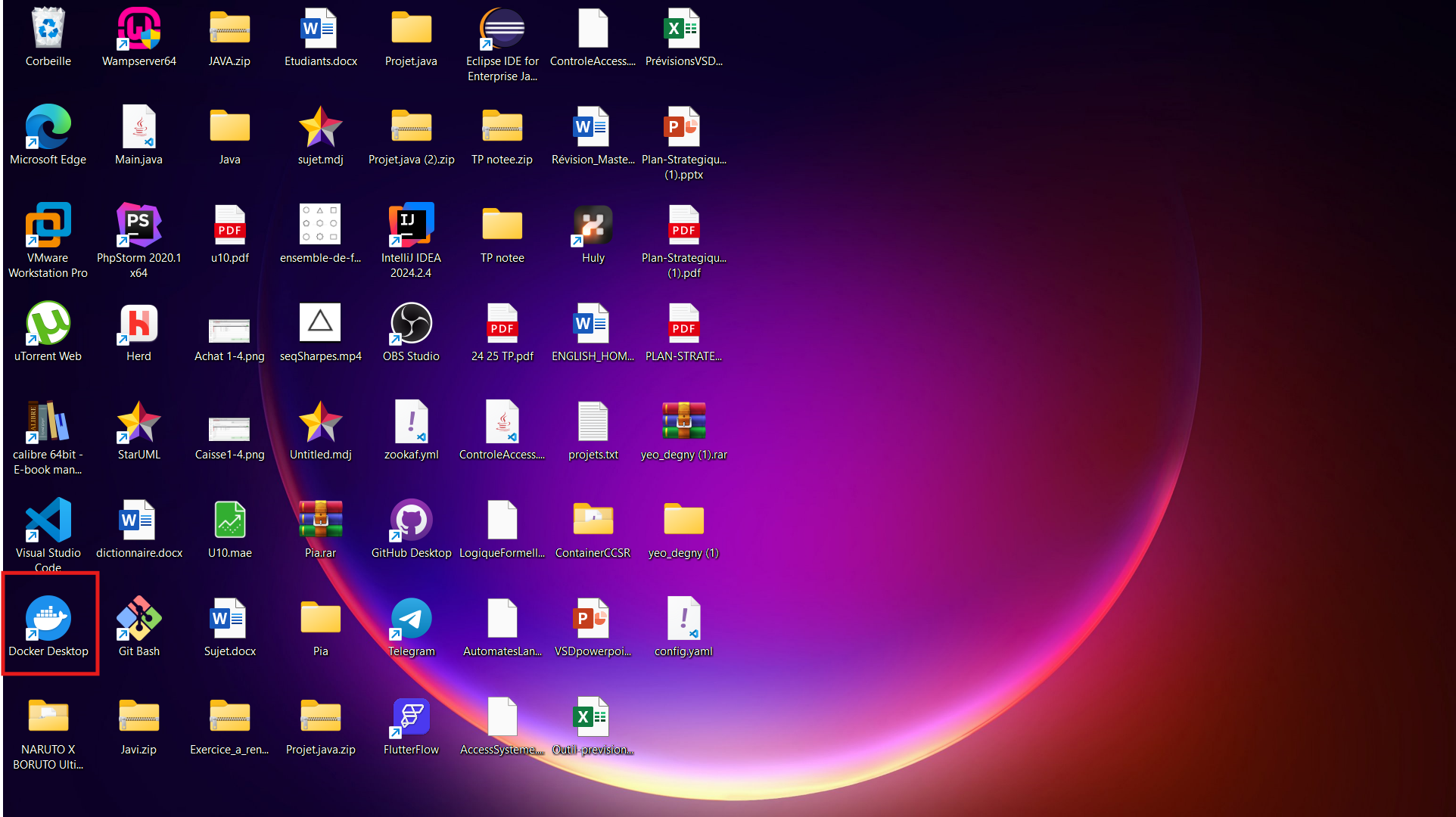
Guide de l’application

Version de Java Utilisée : Java22

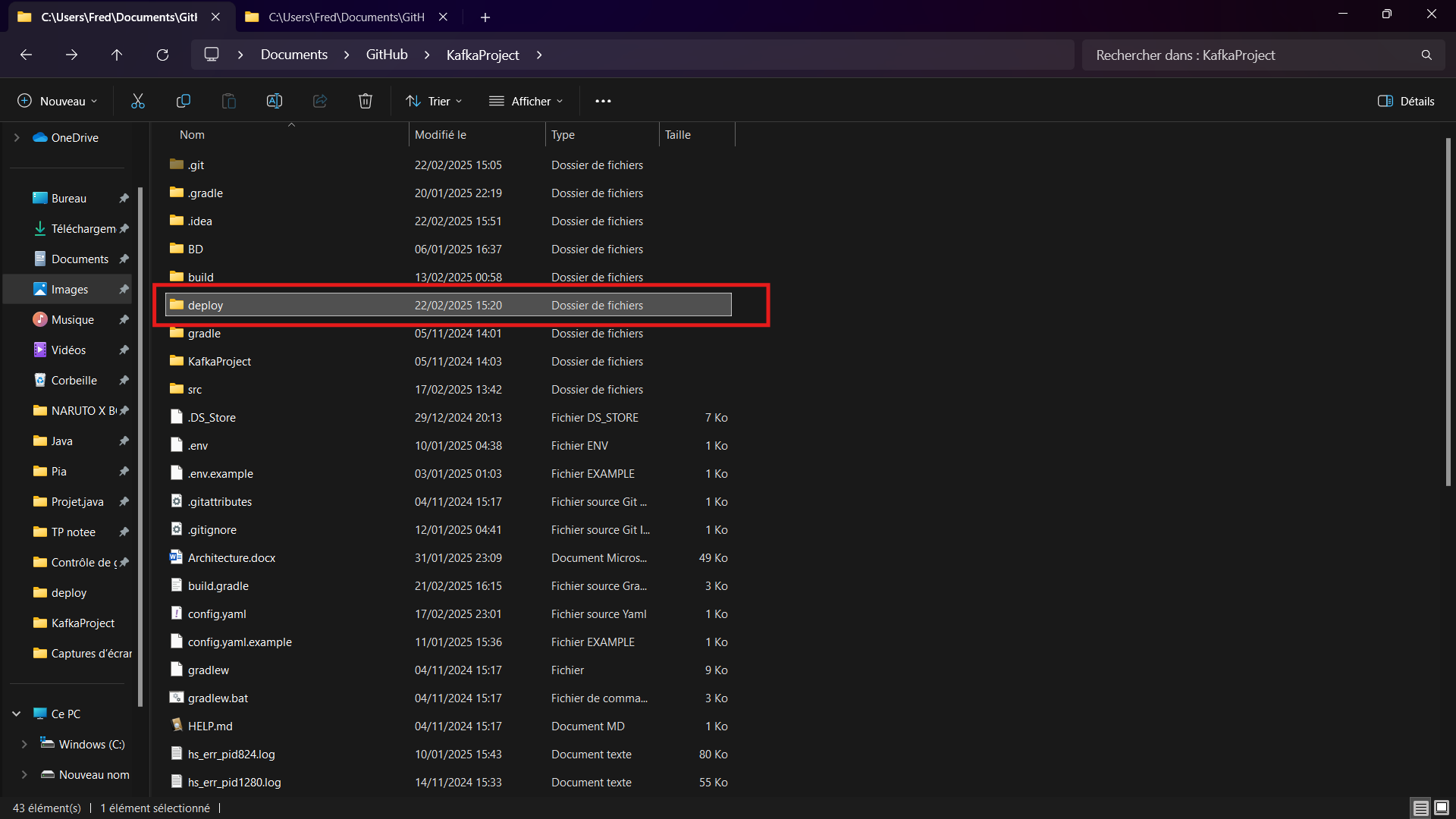
**----------Configuration pour déploiement----------**

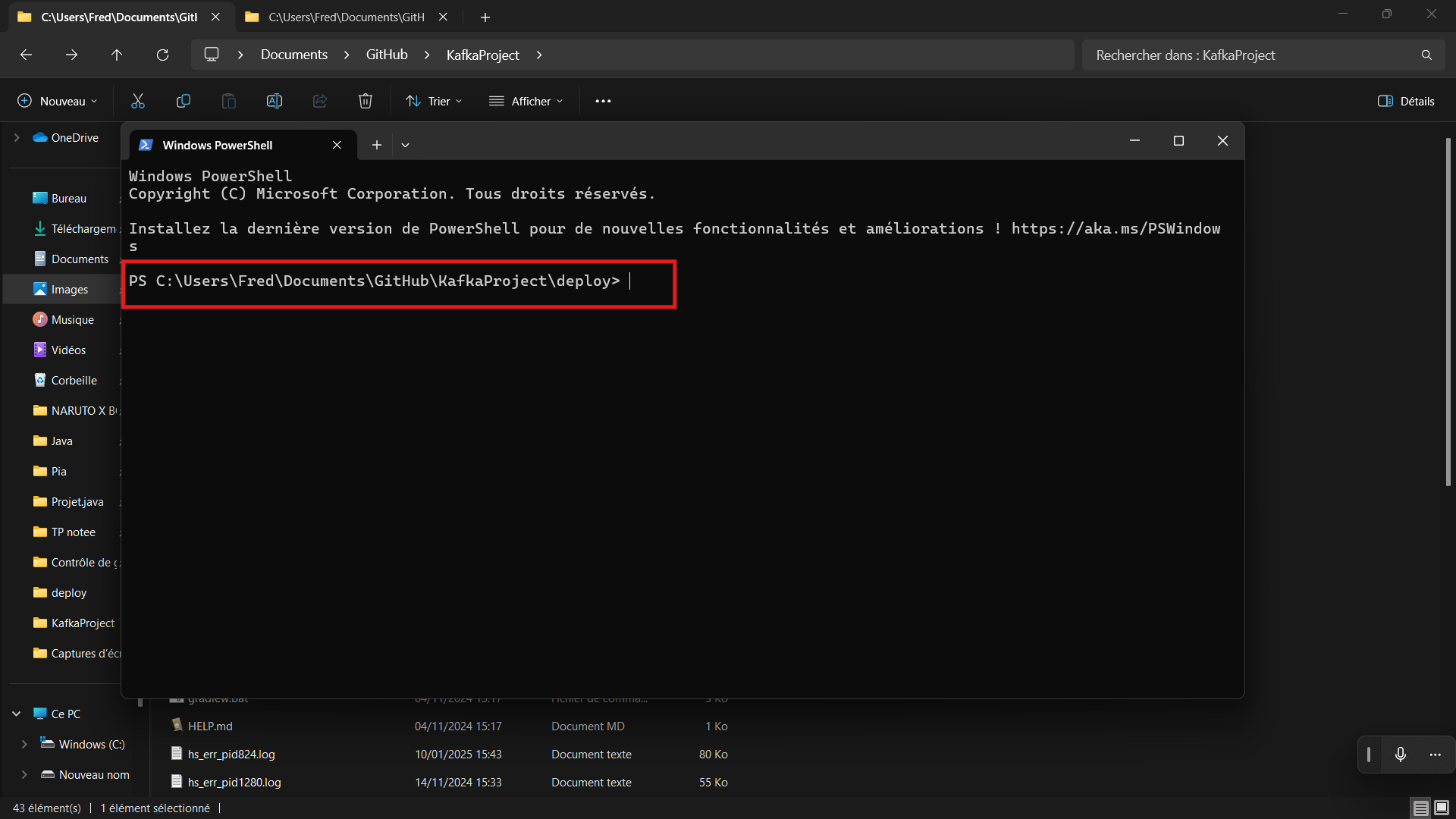
1. Télécharger le projet sur votre ordinateur

2. Activer docker

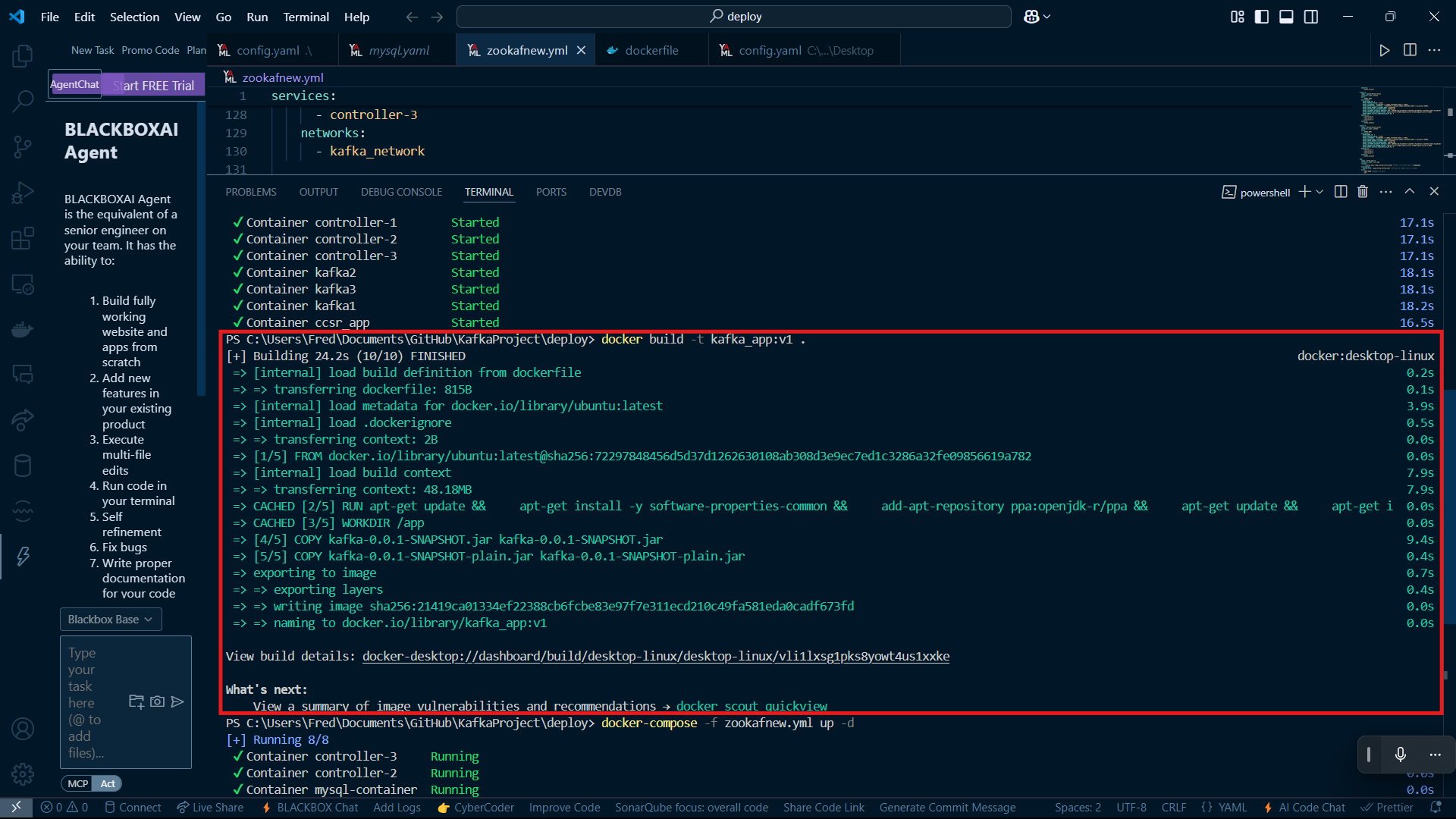


3. Ouvrir le dossier deploy en ligne de commande (‘’Assurez-vous que le fichier zookafnew.yml est présent dans le dossier deploy’’)

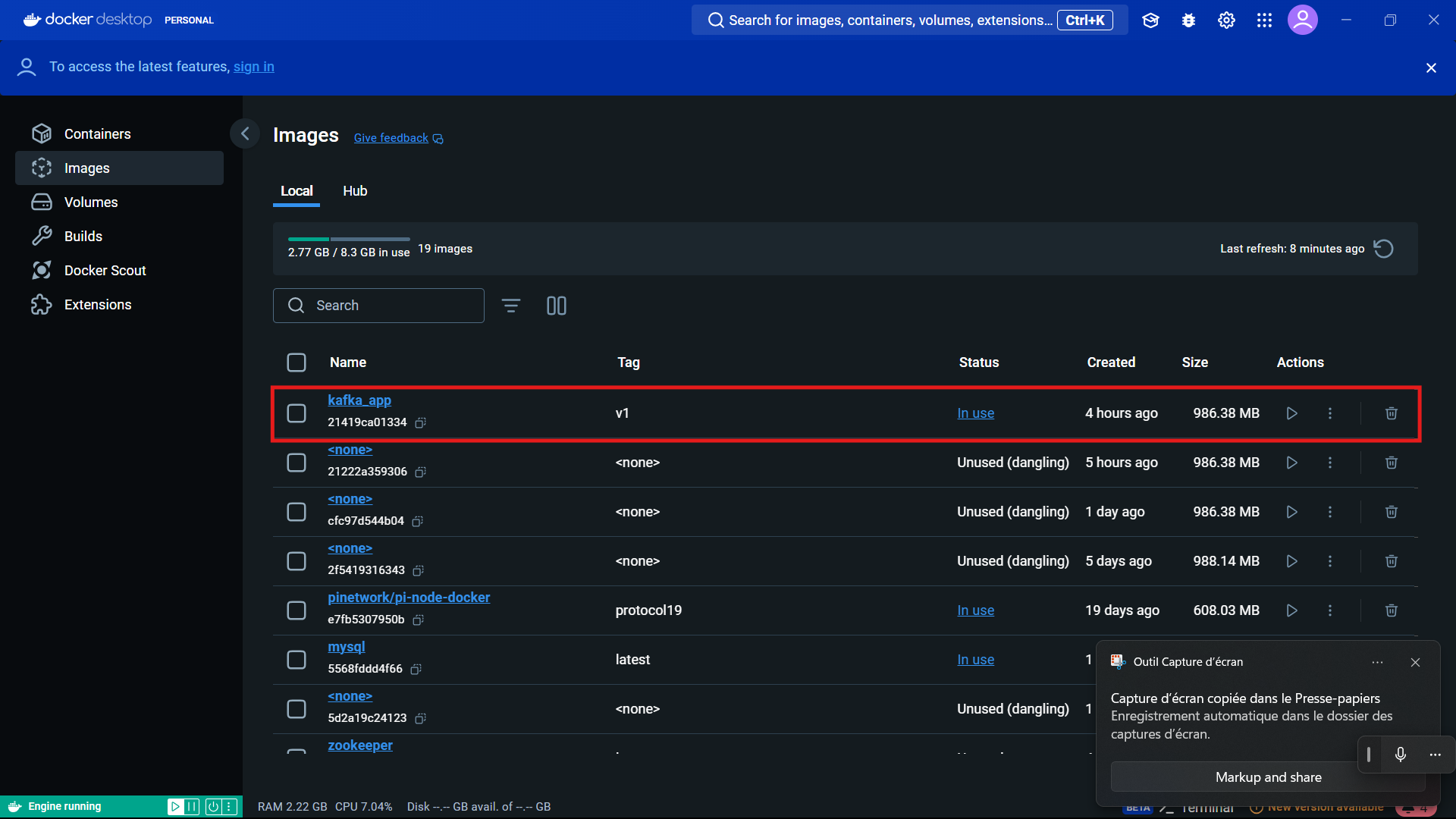




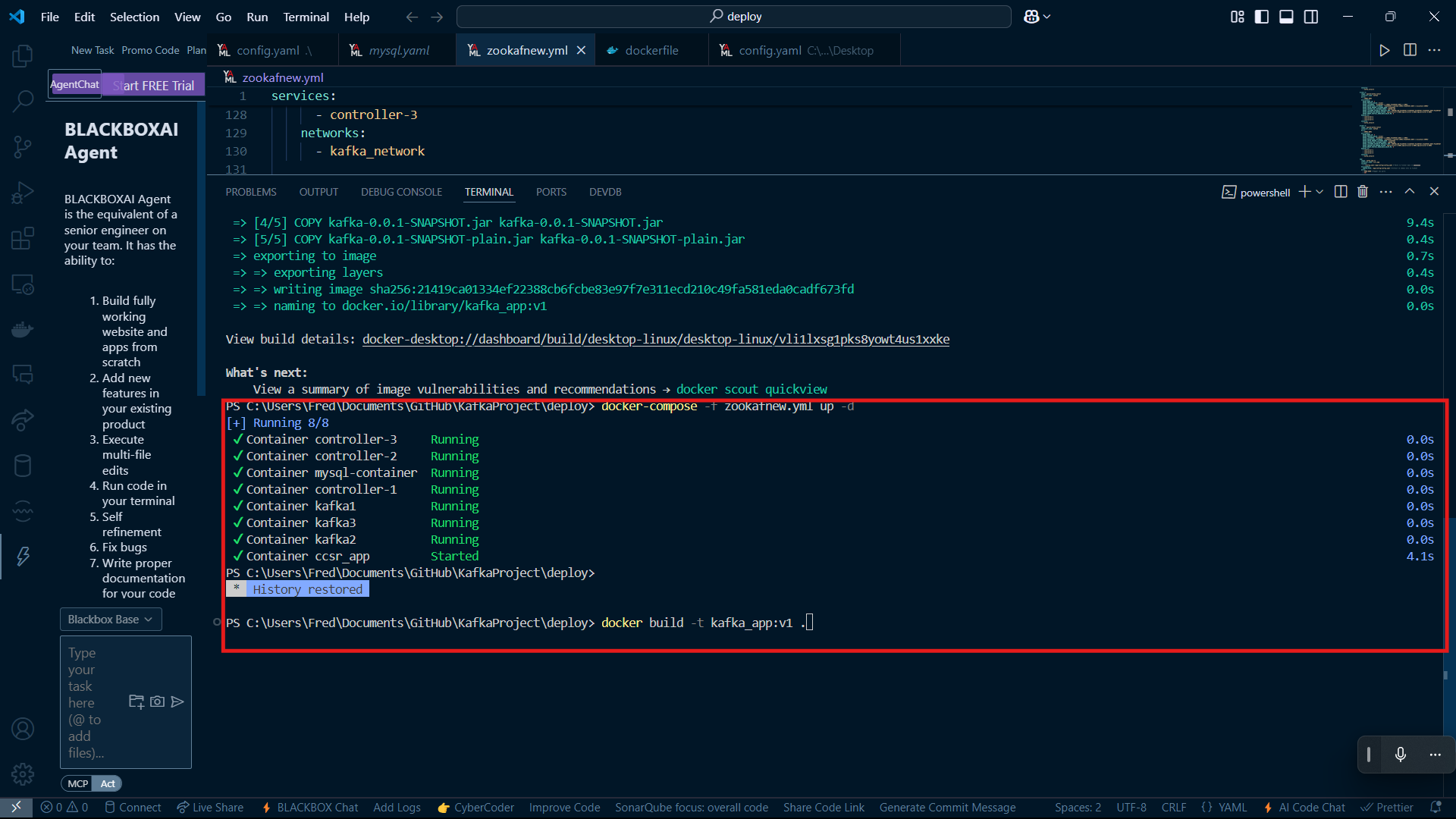
4. Entrer la commande pour construire l’image de l’application : **docker build -t kafka\_app:v1 .**

****

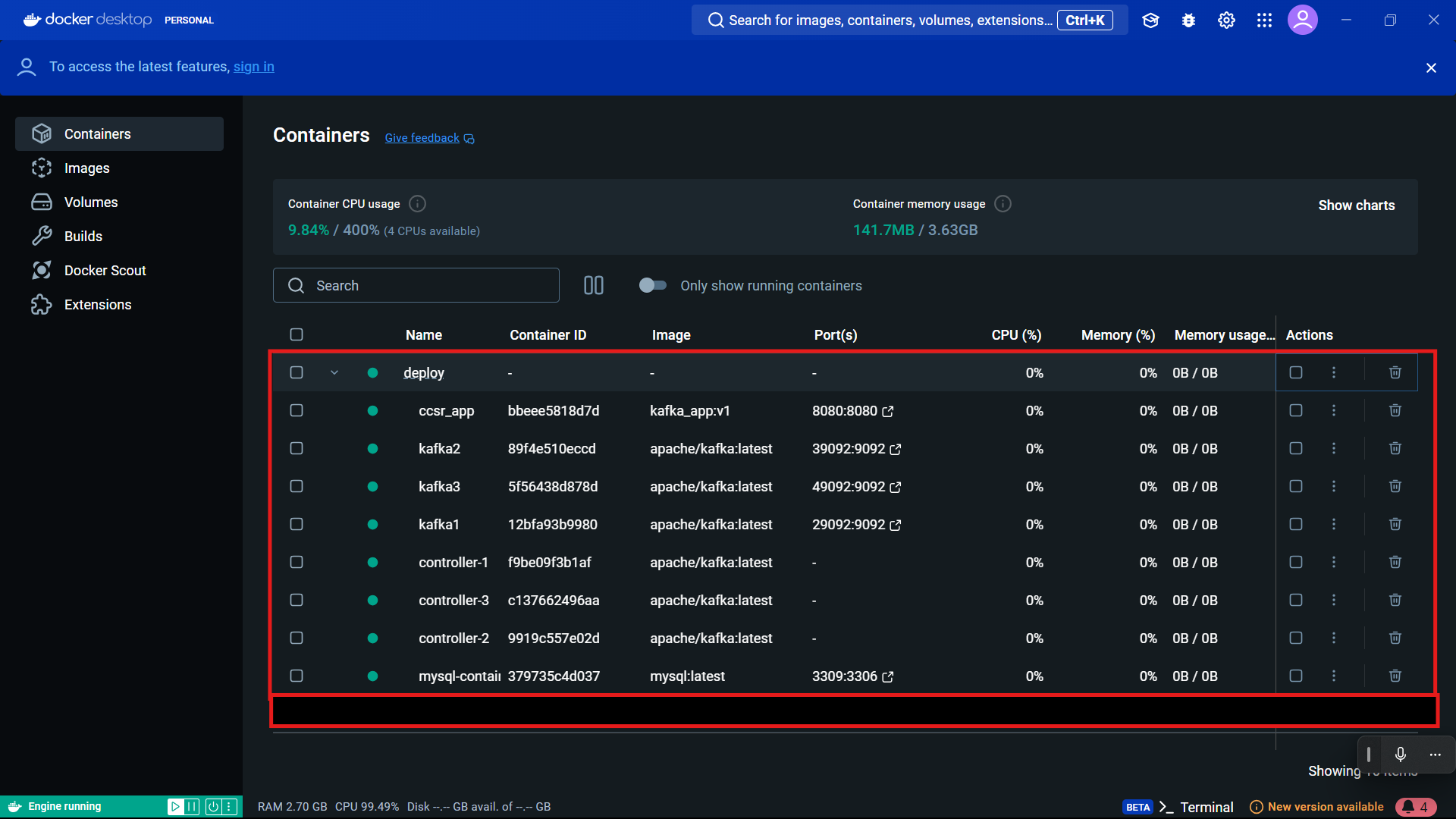
5. Allez vérifier dans kafka que vous avez bien cette image



6. Entrer la commande pour lancer les conteneurs : **docker-compose -f zookafnew.yml up -d**

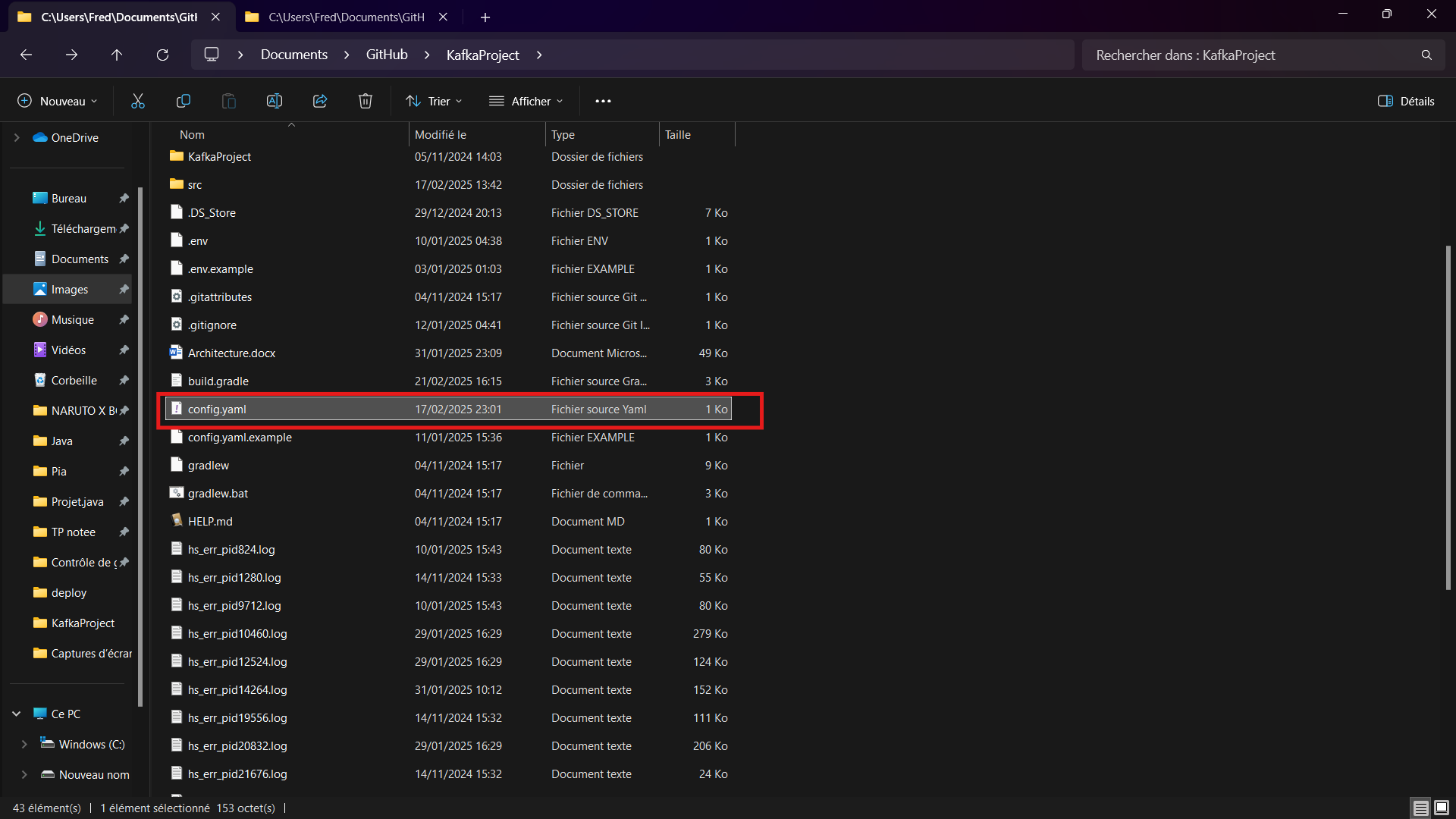
****

7. Les brockers devraient se mettre en marche dans docker

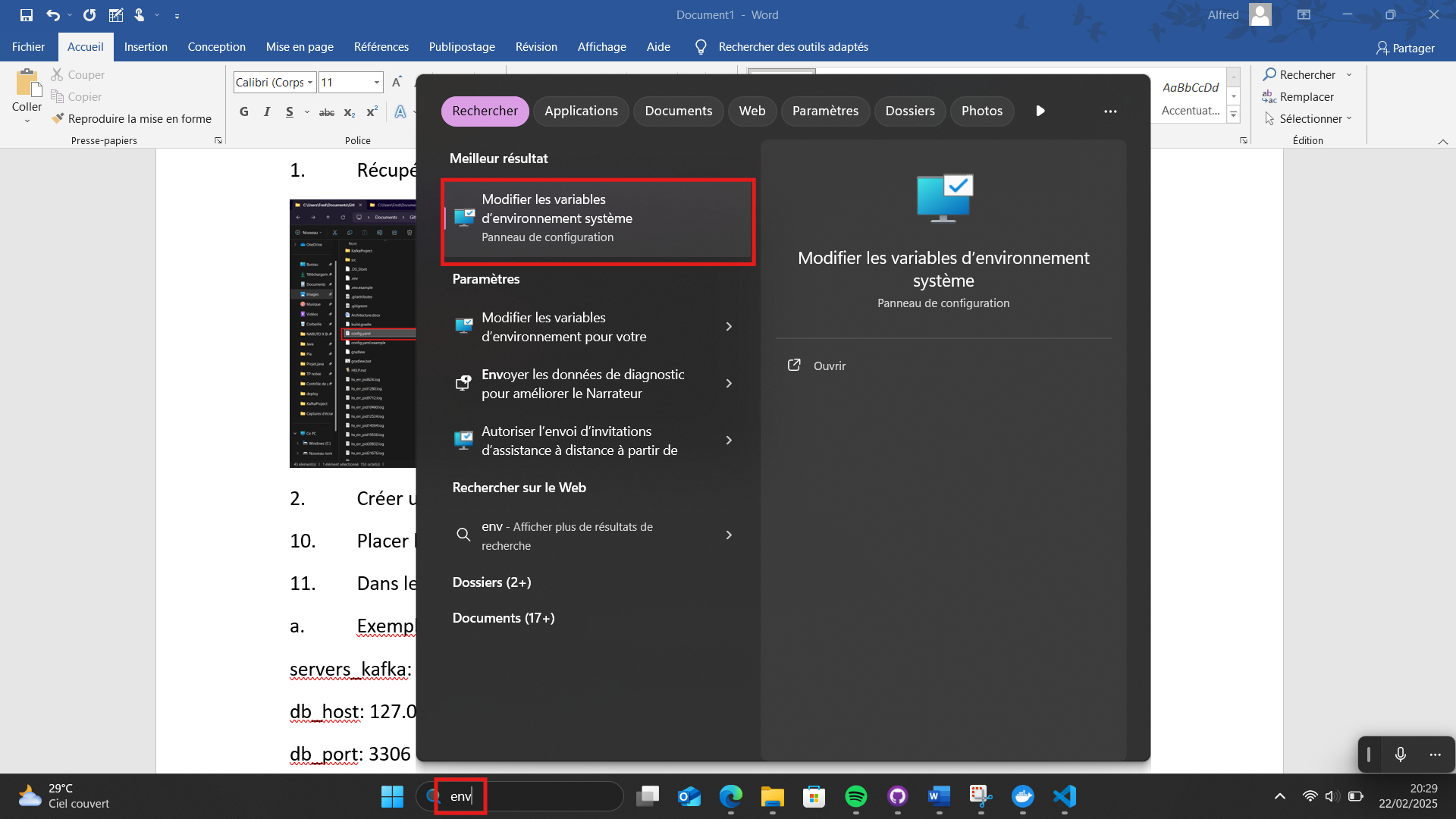


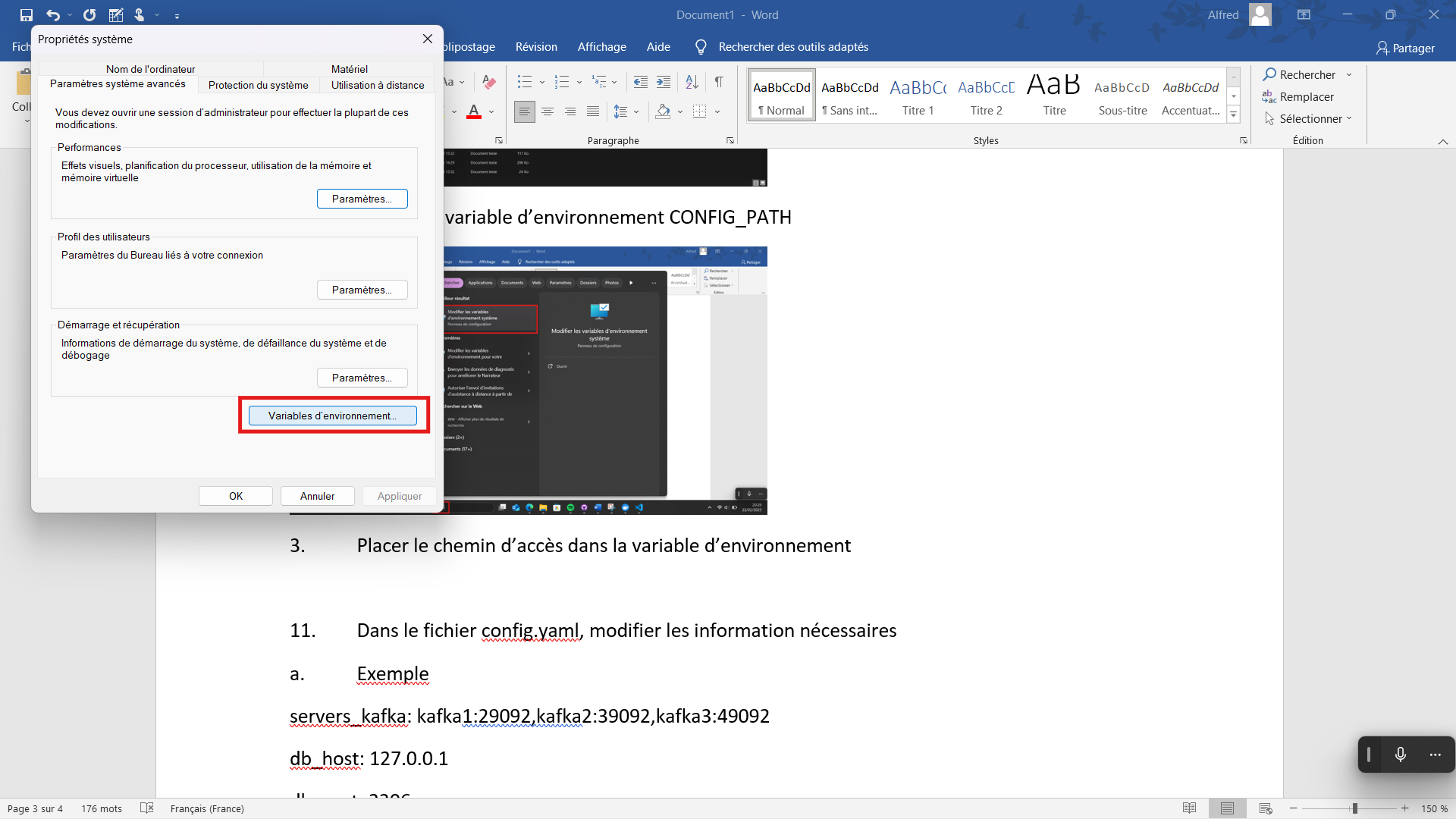
**----------Configuration pour le développement----------**

1. Récupérer le chemin d’accès jusqu’au fichier config.yaml contenu à la racine du projet

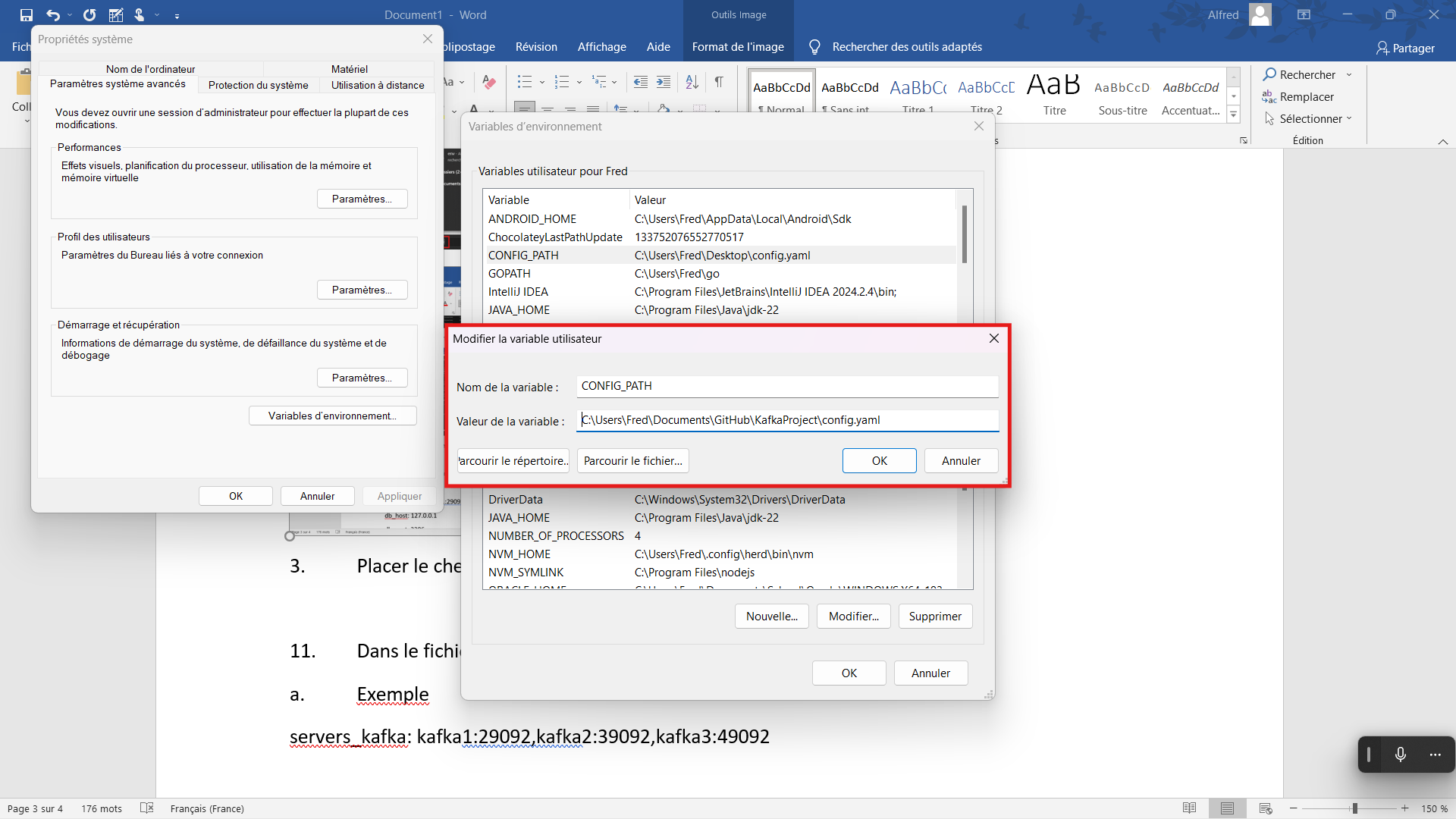


2. Créer une variable d’environnement CONFIG\_PATH

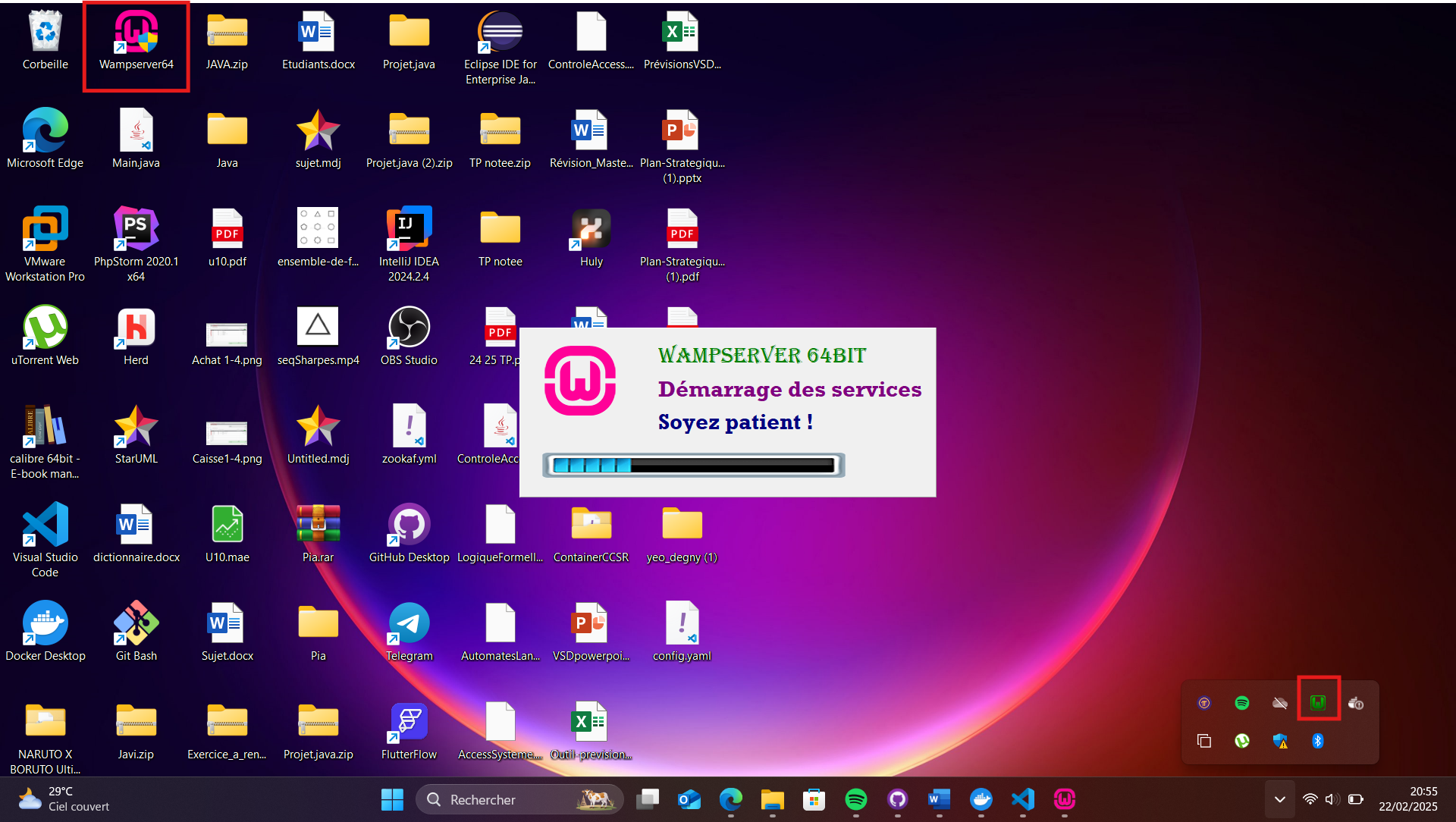




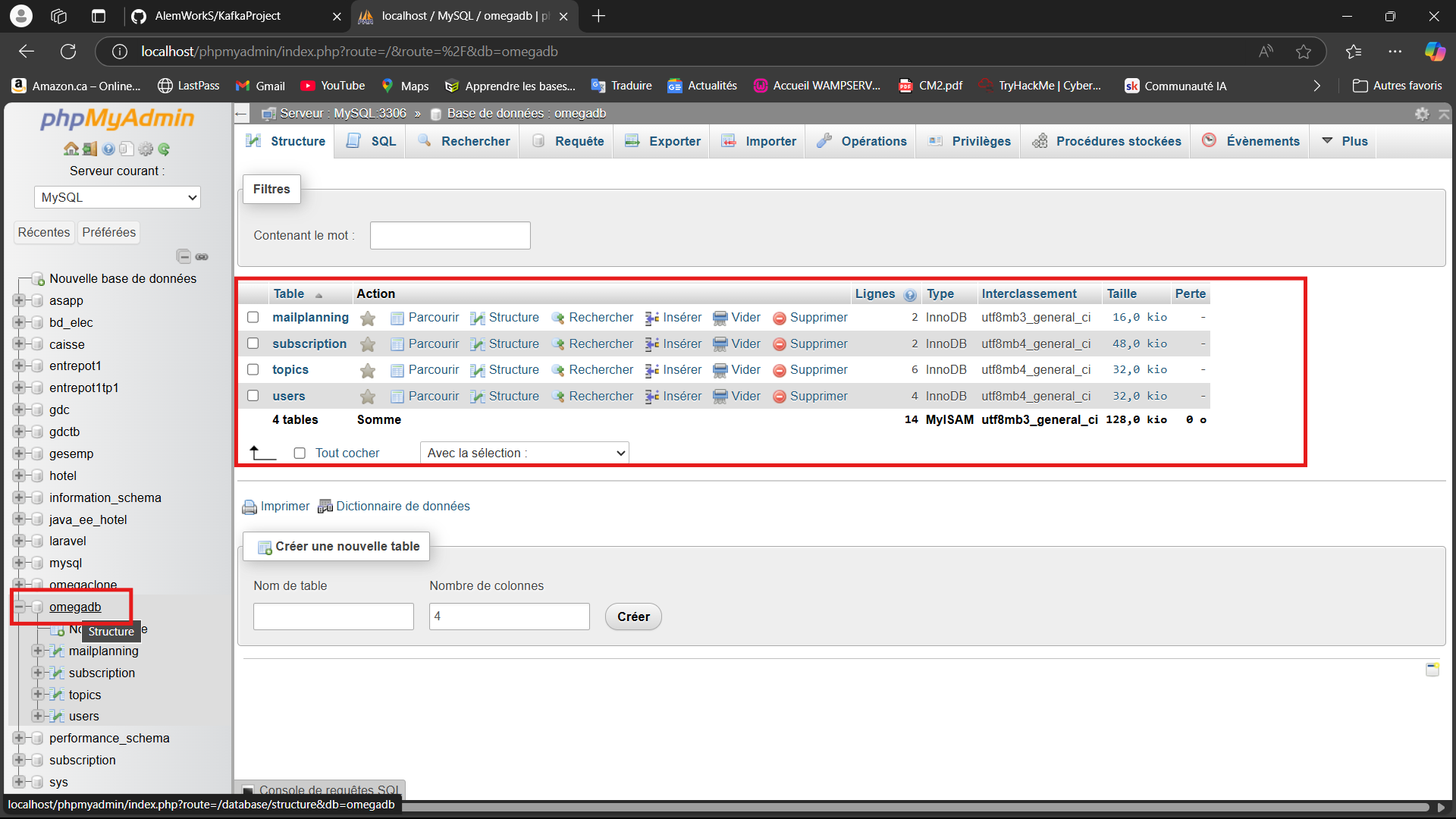
3. Placer le chemin d’accès dans la variable d’environnement



3. Ouvrir votre serveur de base de données



4. Importez sous le nom de omegadb la base de données



5. Dans le fichier config.yaml, modifier les information nécessaires

Exemple :

servers\_kafka: kafka1:29092,kafka2:39092,kafka3:49092

db\_host: 127.0.0.1

db\_port: 3306

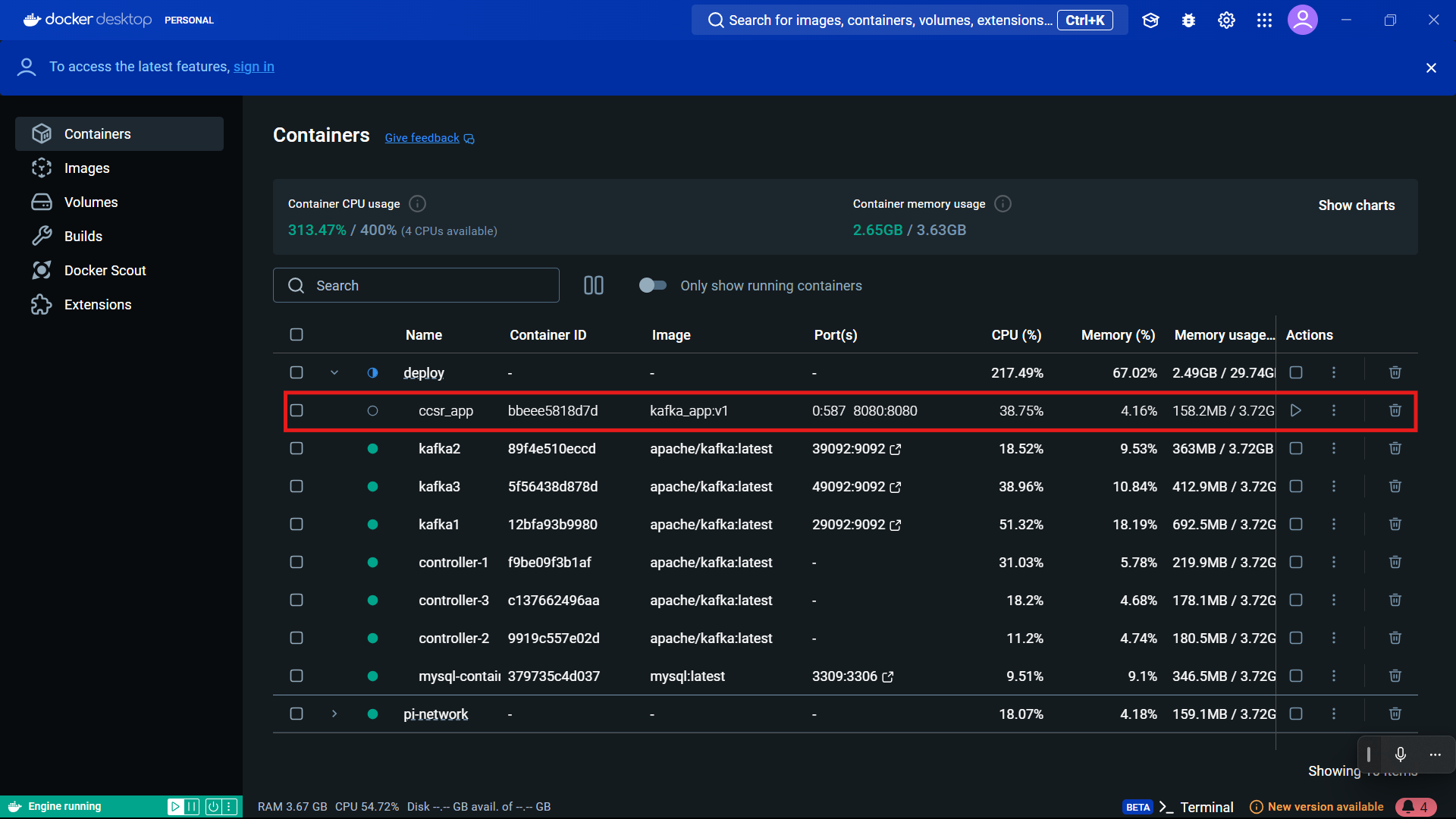
db\_name: omegadb

db\_user: user

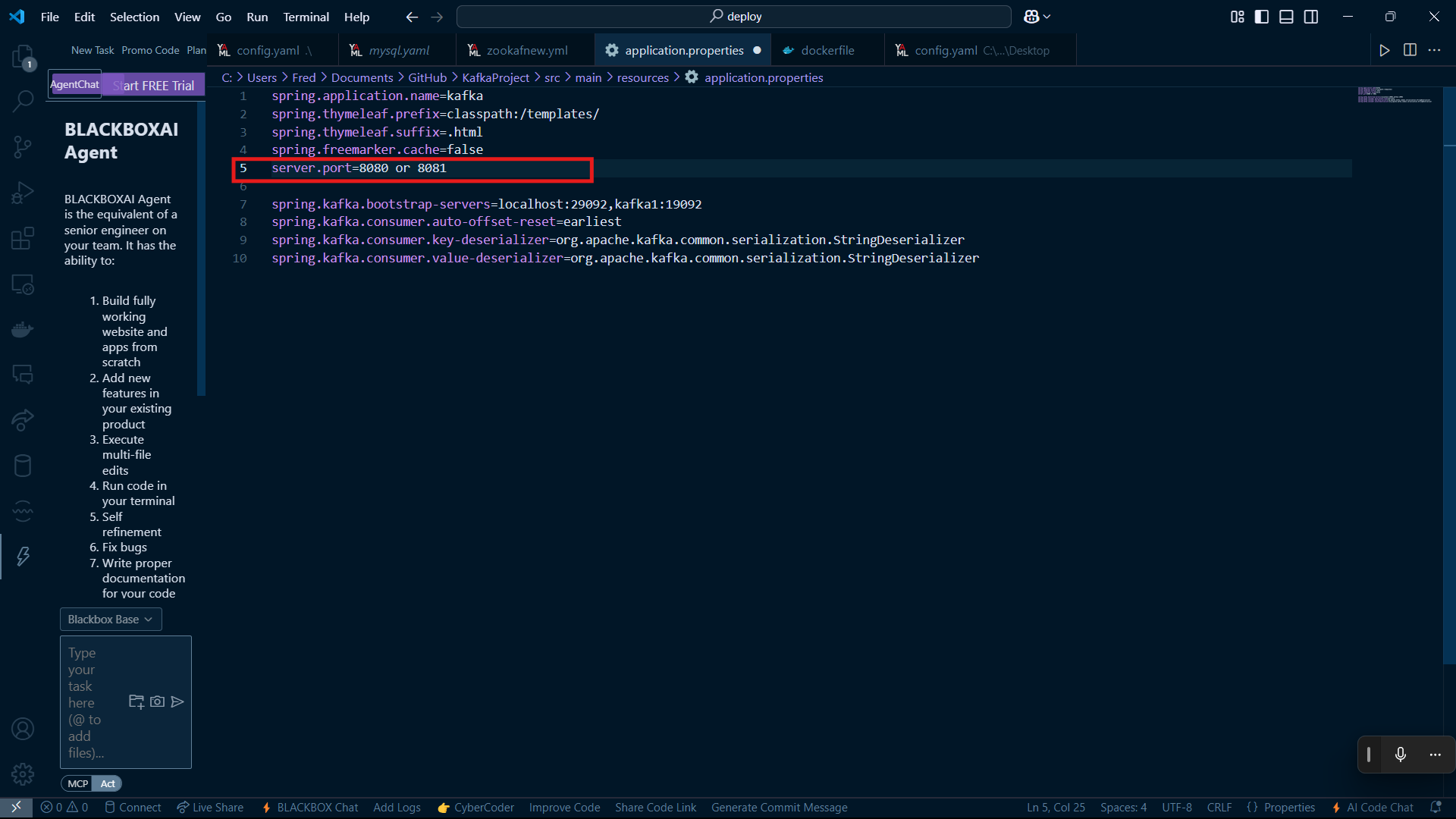
db\_password: password

6. Ouvrez docker et lancez le conteneur *deploy* qui contient les brokers Kafka. ***NB : Pour ne pas avoir un problème de port car l’application sert déjà sur le port 8080 avec docker il y’a 3 solutions :***

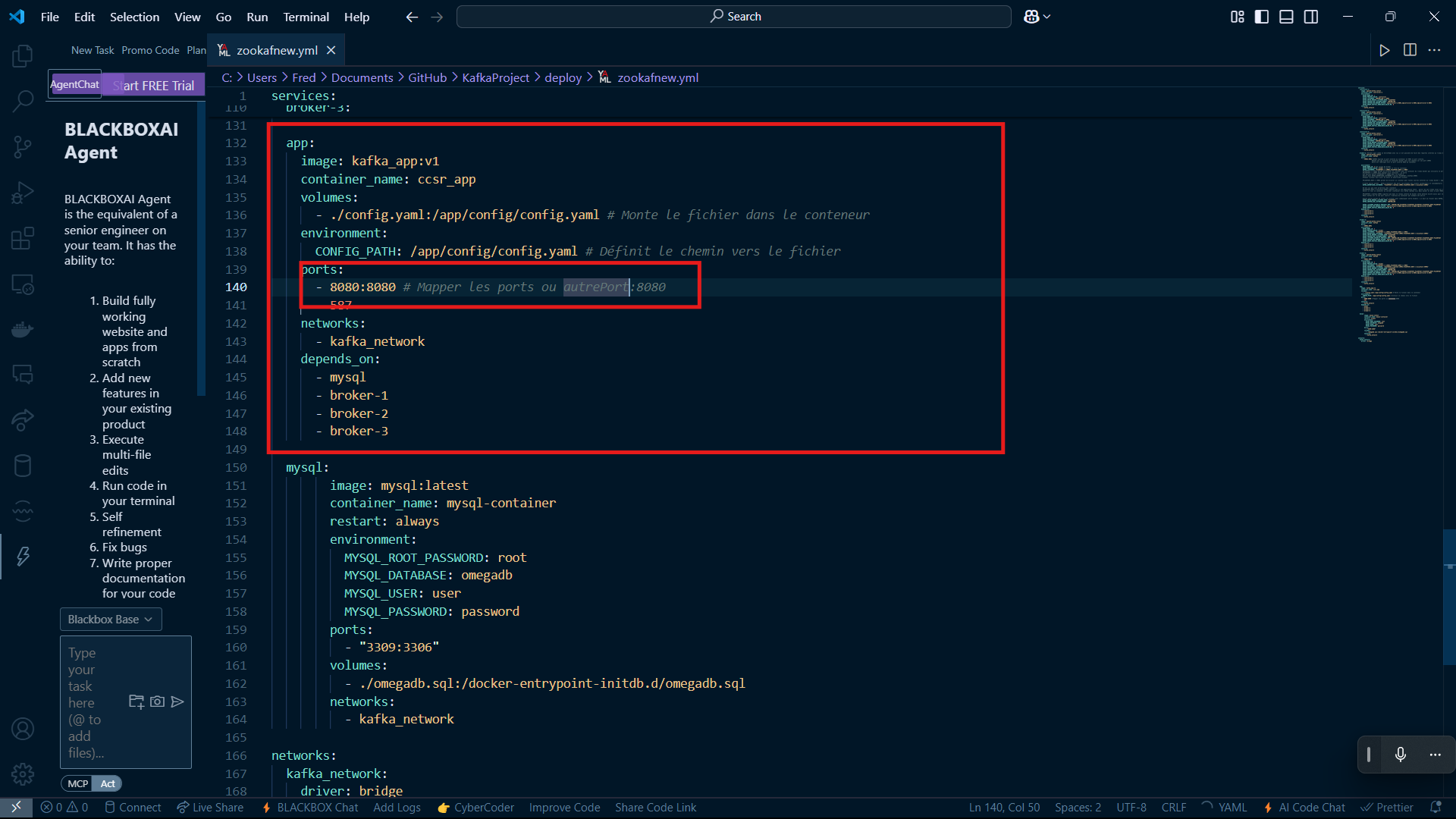
***a. Désactivez le conteneur qui porte le nom ccsr\_app dans le conteneur deploy***

******

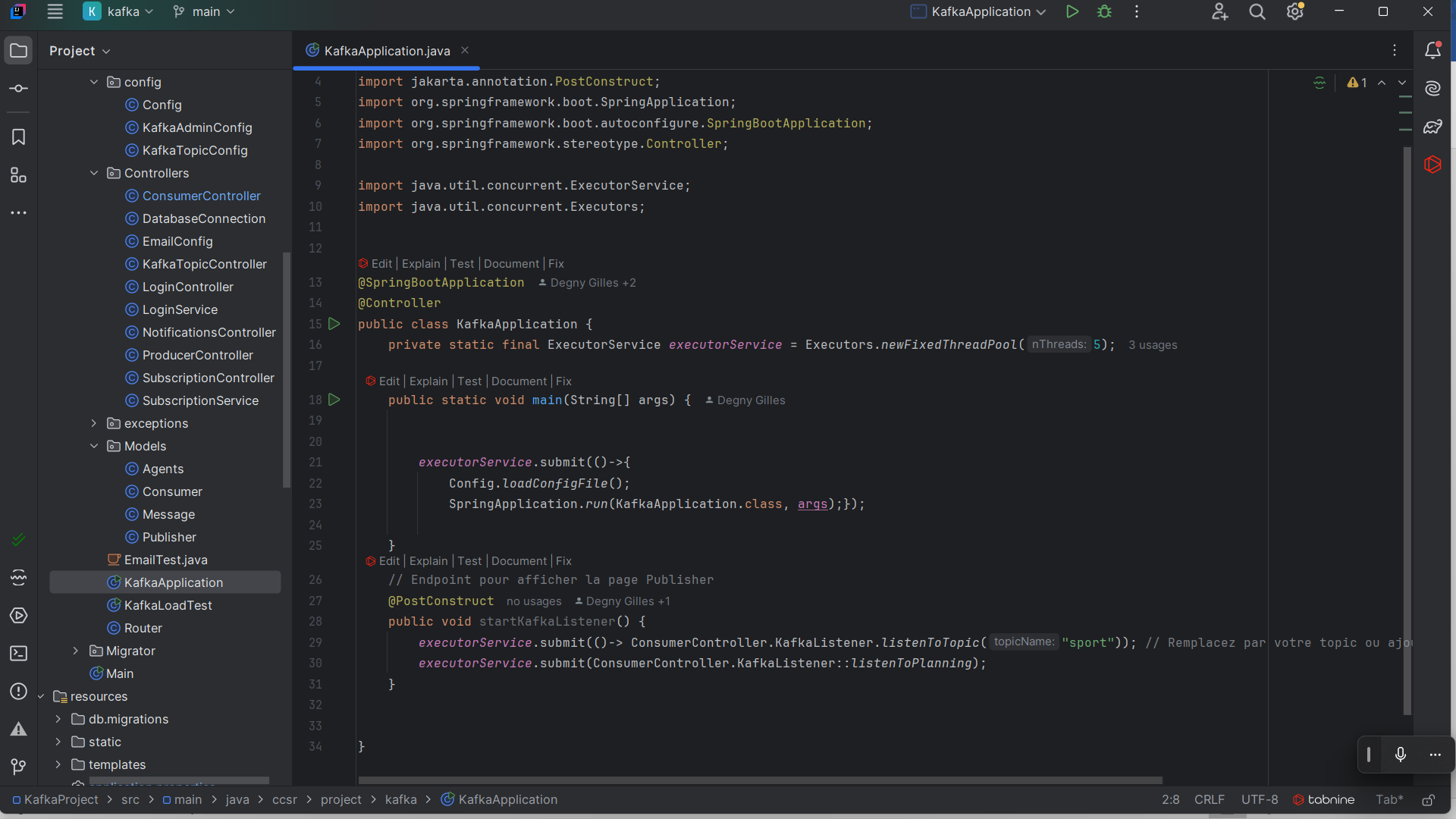
***b. Changez le port dans le fichier "..\src\main\resources\application.properties" (vous pouvez choisir un autre port libre)***

******

***c. Changer le port de sortie sur docker lequel sert à l’application à être disponible sur notre machine locale dans le fichier "..\deploy\zookafnew.yml"***

******

7. Lancer le projet kafka (La class KafkaApplication.java) dans votre IDE



8. Entrez dans un navigateur l’url ‘’localhost:8080’’



Explication de fonctionnement :

L’application utilise des threads qui se lance dès la fin de la compilation. Un daemon qui se charge de lire un **planning** dans la base de données et un autre qui se charge de lire **les nouveaux messages sur les topics**.

Vous pouvez vous inscrire en tant que Consumer ou Producer vous avez la capacité de :

**Consumer** : Lire les messages, S’abonner aux topics, Planifier des notifications, Se connecter, S’inscrire

**Producer** : Créer des messages, Se connecter, S’inscrire

Lorsqu’un message est entré dans un topic au quel un utilisateur/consommateur est abonné et a défini une heure de réception des notifications, le daemon qui se charge de lire les topics chaque 15 seconds va :

1. Rechercher tous les abonnés au topic
2. Lancer un thread pour planifier une notification pour chaque abonné en laissant une marque pour l’autre daemon de planning

Par la suite, Le daemon qui se charge de lire le planning chaque 60 seconds va :

1. Lire la variable marquée à true (envoyé)
2. Confier la tâche à un thread pour envoyer le mail à l’utilisateur qui est obligatoirement abonné et a configuré la réception de notifications.

**Bonne utilisation**