**Rapport Projet : Architecture des composants d’entreprise**

**Réaliser : BOUKAR BRAHIM**

**2. Architecture Micro-services**

* Architecture

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

Description générée automatiquement

* Description des services :

On a deux service department-service et employee-service chaque service auront leur propre objectif

* Mécanismes de communication

**Protocole HTTP/HTTPS :** Cela permet une communication légère et standardisée. Les bibliothèques comme REST Template (pour Java) facilitent l'intégration HTTP entre les services.

**4. Conteneurisation avec Docker**

* **Implémentation**

1/ Création fichier Dockerfile dans la racine de micro-service

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

2/ création le jar de projet :

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

Description générée automatiquement

3/ création image dans docker :

Une image contenant texte, capture d’écran

Description générée automatiquement

4/ vérifier si image était créer :

Une image contenant texte, capture d’écran, Police

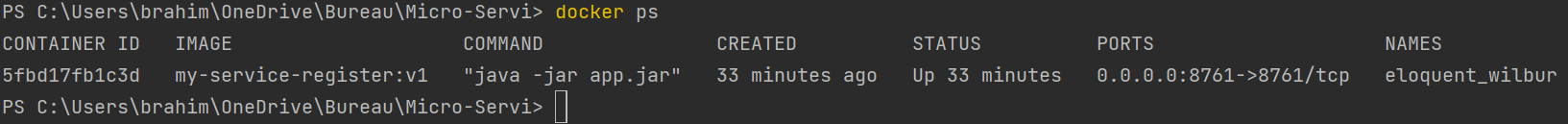
Description générée automatiquement

5/ démarrer docker avec spécification de port intérieur et extérieur :

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

Description générée automatiquement

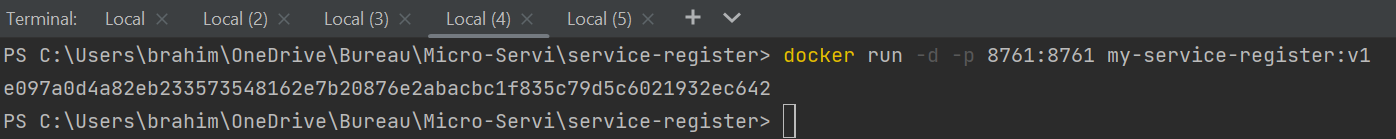
6/ voir si le conteneur est démarré :



Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Il Ya une option pour démarrage le conteneur se forme d’une démo en arrière-plan :



* + **Travailler avec docker-compose**

On va allez vers notre projet Principal on va créer un fichier docker-compose.yml

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

Description générée automatiquement

1/ Chaque service on donne un nom après on va utiliser build pour construire image donc la commande qui avait utiliser (docker build) on va plus faire c’est docker compose qui va le faire

2/ donner le nom de contenaire

3/ utiliser les port 8761:8761

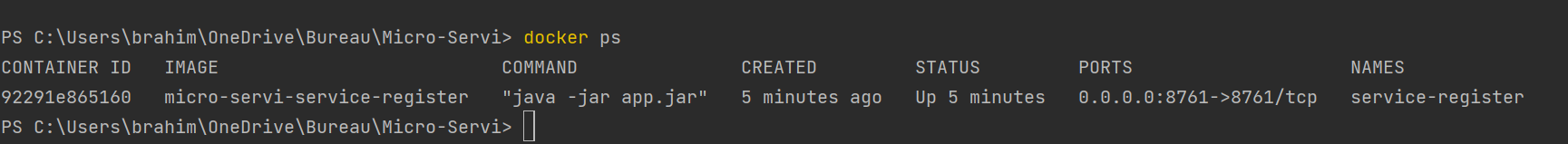
4/ exposer le port 8761

5/ démarrage tous les services qui sont déclarer le fichier docker-compose.yml

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Une fois image est construite il démarre le conteneur, si en taper la commande docker ps on voyer bien que notre docker démarrer



Si on va démarrer en arrière-plan on taper la commande : docker compose up -d

* + **Maintenant on va démarrer le service de configuration**

1/ Première mot on va générer le jar

Avec la commande : ***mvn clean package -DskipTests***

2/ on revient dans le fichier docker-compose.yml en déclare le service avec spécification le port et le nom de contenaire et build

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

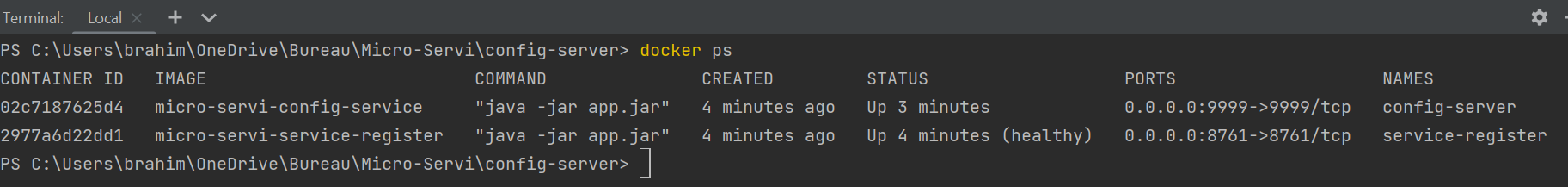
Description générée automatiquement

3/ démarrage les conteneurs avec la commande : ***docker compose up -d –build***

On a ajouté option –build 🡺 parce-que si image-t-il déjà construite on mettant --build il force si image est déjà construit il va construire on neveux

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

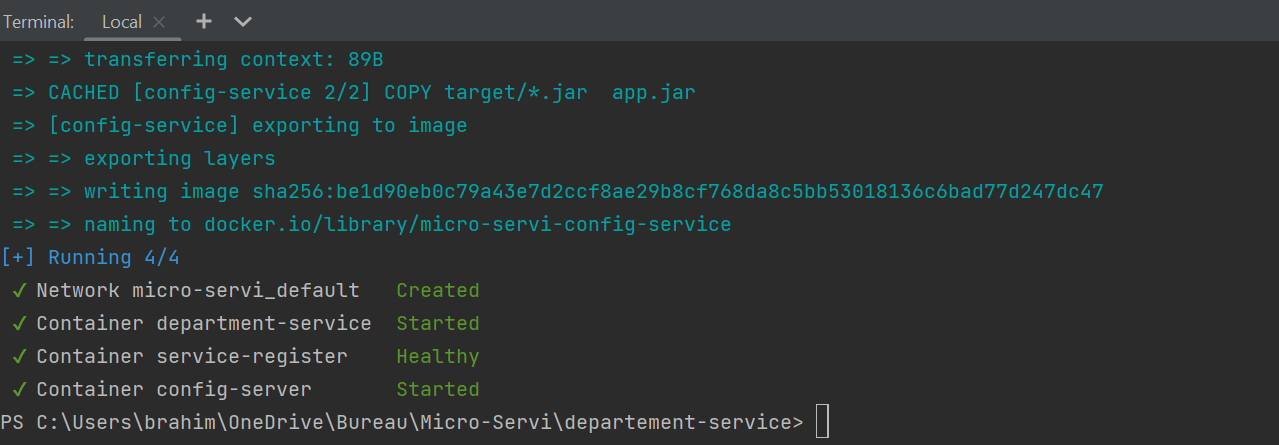
Description générée automatiquement



* + **Maintenant on va démarrer le service de department-services**

Les mêmes instructions que on a déjà fait

1/ démarrage les conteneurs avec la commande : docker compose up -d –build



* + **Maintenant on va démarrer le service de employee-services**

Les mêmes instructions que on a déjà faites

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

* + **Maintenant on va démarrer le service de employee-services**

Une image contenant texte, capture d’écran, Police

Description générée automatiquement