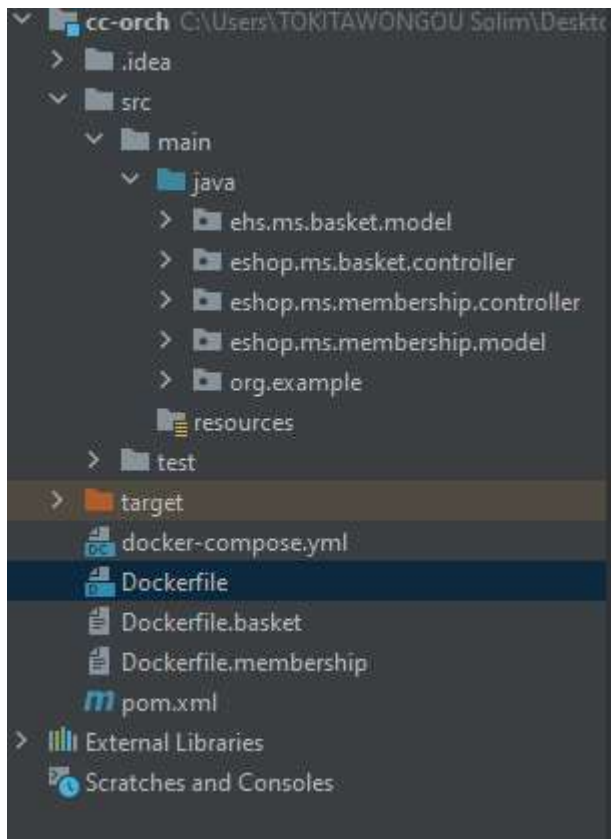


Compte rendu du cc2 : Orchestration des services de données

A- Arborescence du projet



- 1) Création d'un Dockerfile de chaque microservice (Basket et Membership) du TP précédent et test de la construction locale.

- Dockerfile.membership

```
# Étape 1 : Construire l'application avec Maven
FROM maven:3.8.5-openjdk-17 AS builder
WORKDIR /app

# Affiche la version de Maven et Java (facultatif pour débogage)
RUN mvn -v
RUN java -version

# Copier uniquement les fichiers nécessaires pour optimiser le cache Docker
COPY pom.xml .
COPY src ./src

# Compiler l'application
RUN mvn clean package -DskipTests

# Étape 2 : Exécuter l'application avec une image plus légère
FROM openjdk:17-jdk-slim
WORKDIR /app
# Copier le fichier JAR généré
COPY --from=builder /app/target/*.jar app.jar
# Exposer le port de l'API Spring Boot
EXPOSE 8080
# Lancer l'application
ENTRYPOINT ["java", "-jar", "app.jar"]
```

- Dockerfile.membership

```
# Étape 1 : Construire l'application avec Maven
FROM maven:3.8.5-openjdk-17 AS builder
WORKDIR /app

# Copier uniquement les fichiers nécessaires pour optimiser le cache Docker
COPY pom.xml .
COPY src ./src

# Compiler l'application
RUN mvn clean package -DskipTests

# Étape 2 : Exécuter l'application avec une image plus légère
FROM openjdk:17-jdk-slim
WORKDIR /app

# Copier le fichier JAR généré
COPY --from=builder /app/target/*.jar app.jar

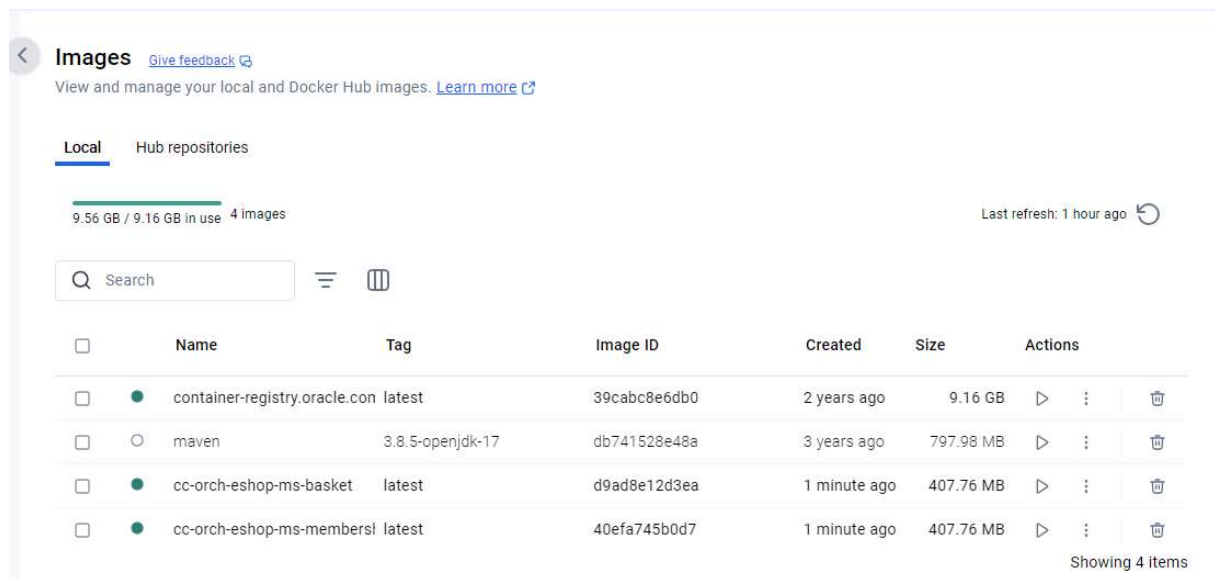
# Exposer le port de l'API Spring Boot
EXPOSE 8080

# Lancer l'application
ENTRYPOINT ["java", "-jar", "app.jar"]
```

- Test en construction locale

```
PS C:\Users\TOKITAWONGOU Solim\Desktop\SIRIUS\smart-farm\cc-orch> docker-compose up
[+] Running 2/2
 ✓ Container cc-orch-eshop-ms-basket-1      Created
 ✓ Container cc-orch-eshop-ms-membership-1  Created
Attaching to eshop-ms-basket-1, eshop-ms-membership-1
eshop-ms-basket-1 | no main manifest attribute, in app.jar
eshop-ms-membership-1 | no main manifest attribute, in app.jar
eshop-ms-membership-1 exited with code 1
eshop-ms-basket-1 exited with code 1
v View in Docker Desktop  o View Config  w Enable Watch
```

2.) Conteneurs et images publiées

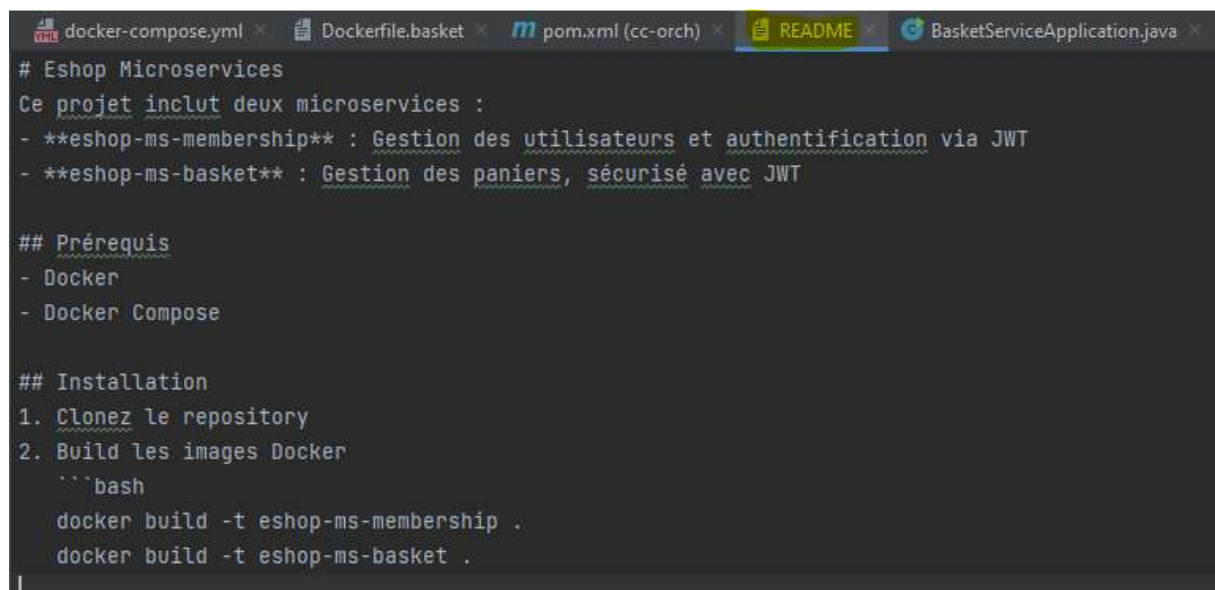


The screenshot shows the Docker Desktop 'Images' tab. It displays a list of local images with columns for Name, Tag, Image ID, Created, Size, and Actions. The images listed are container-registry.oracle.con:latest, maven:3.8.5-openjdk-17, cc-orch-eshop-ms-basket:latest, and cc-orch-eshop-ms-membership:latest. A progress bar at the top indicates 9.56 GB / 9.16 GB in use for 4 images. A search bar and icons for filters and views are also visible.

<input type="checkbox"/>	Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	container-registry.oracle.con	latest	39cab8e6db0	2 years ago	9.16 GB	
<input type="checkbox"/>	maven	3.8.5-openjdk-17	db741528e48a	3 years ago	797.98 MB	
<input type="checkbox"/>	cc-orch-eshop-ms-basket	latest	d9ad8e12d3ea	1 minute ago	407.76 MB	
<input type="checkbox"/>	cc-orch-eshop-ms-membership	latest	40efa745b0d7	1 minute ago	407.76 MB	

Showing 4 items

3.) Création d'un README



The screenshot shows a code editor with a README file. The content is in French and describes the project, prerequisites, and installation steps.

```
# Eshop Microservices
Ce projet inclut deux microservices :
- **eshop-ms-membership** : Gestion des utilisateurs et authentification via JWT
- **eshop-ms-basket** : Gestion des paniers, sécurisé avec JWT

## Prérequis
- Docker
- Docker Compose

## Installation
1. Clonez le repository
2. Build les images Docker
```bash
docker build -t eshop-ms-membership .
docker build -t eshop-ms-basket .
```

#### 4.) Création du fichier docker-compose.yml

```
version: '3'

services:
 eshop-ms-membership:
 build:
 context: . # La racine du projet
 dockerfile: Dockerfile.membership # Utiliser ce Dockerfile pour le service Membership
 ports:
 - "8081:8080" # Mappe le port du service Membership
 environment:
 - SPRING_PROFILES_ACTIVE=prod
 networks:
 - eshop-network

 eshop-ms-basket:
 build:
 context: . # La racine du projet
 dockerfile: Dockerfile.basket # Utiliser ce Dockerfile pour le service Basket
 ports:
 - "8082:8080" # Mappe le port du service Basket
 environment:
 - SPRING_PROFILES_ACTIVE=prod
 networks:
 - eshop-network

networks:
 eshop-network:
 driver: bridge
```

#### 5.) Schéma d'architecture

