**Compte rendu de Contrôle ACE**

**Ingénierie Informatique et Réseaux**



Nom de l’étudiant 1 : ESSAKHI Aimrane

Nom de l’étudiant 2 : MALLOULI Yahya

Nom de l’étudiant 3 : ELAMRI Maryem

# Contexte Général

Notre application de recherche de colocataires a été conçue pour simplifier le processus de recherche de partenaires de colocation compatibles.

Dans un marché immobilier en constante évolution, il est devenu essentiel de fournir une plateforme conviviale pour faciliter la connexion entre des individus partageant des intérêts et des styles de vie similaires.

# Technologie

* Eureka \_ Open Feign
* Docker
* Jenkins
* SonarCloud

# Microservice

Microservice de Gestion des Utilisateurs :

Entité : User

Attributs : ID utilisateur, nom, prénom, adresse e-mail, etc.

Microservice de Gestion des Offres de Colocation :

Entité : Offre de Colocation

Attributs : ID offre, adresse du logement, règles de vie, préférences de colocation, etc.

Relation : Un utilisateur (propriétaire) peut créer plusieurs offres de colocation.

Microservice de Gestion des Demandes de Colocation :

Entité : Demande de Colocation

Attributs : ID demande, ID utilisateur demandeur, ID offre associée, message d'accompagnement, statut (acceptée, rejetée, en attente), etc.

Relations :

Une demande est associée à un utilisateur demandeur.

Une demande est liée à une offre de colocation.

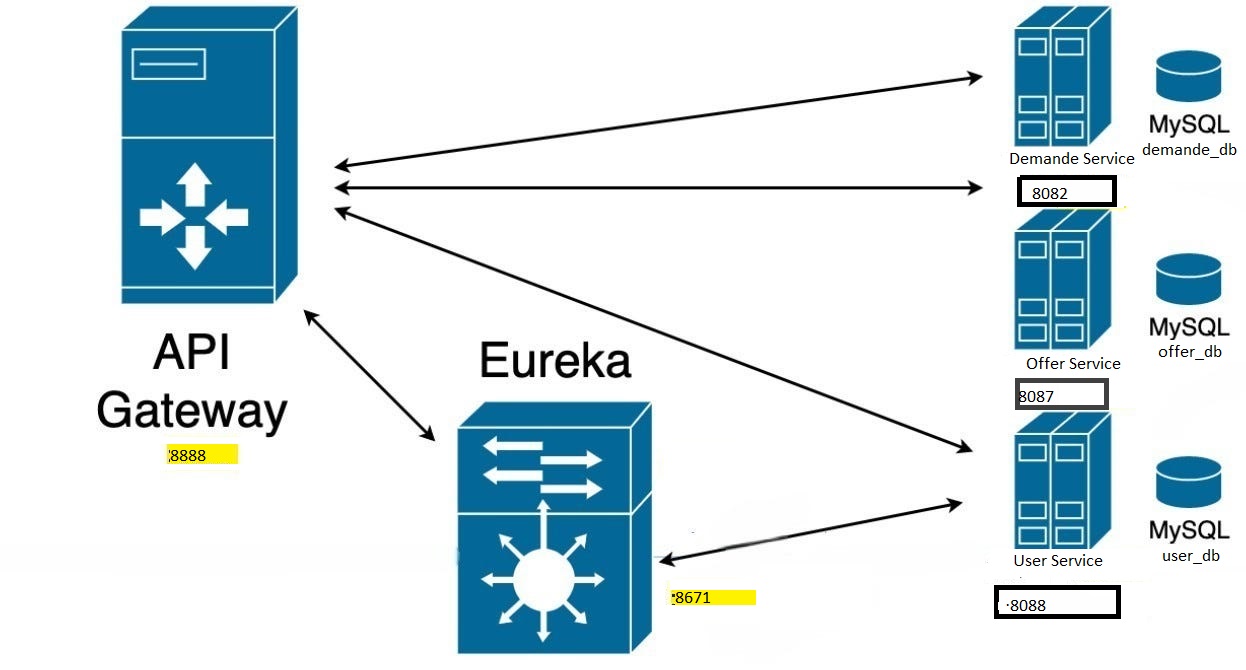
Microservice de Consultation des Offres de Colocation :

Entité : Offre de Colocation (lecture seule)

Attributs : ID offre, adresse du logement, règles de vie, préférences de colocation, etc.

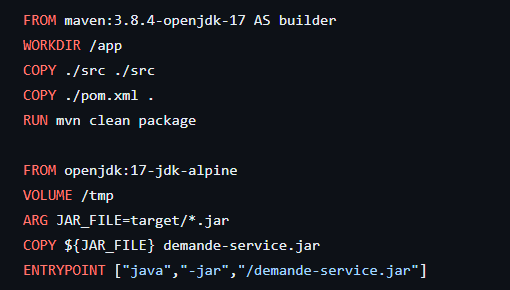
Serveur Eureka :

Rôle : Registre des services, permettant aux microservices de découvrir les autres services disponibles.



# Docker

* DockerFile pour Demande



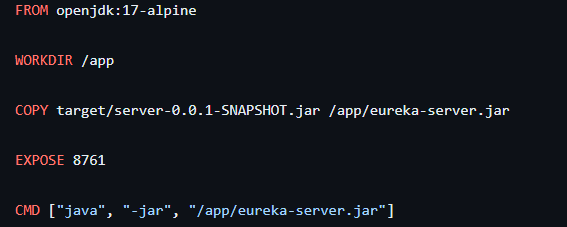
* DockerFile pour User



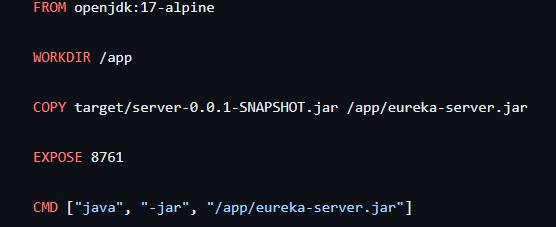
* DockerFile pour User



* DockerFile pour Eureka Server



* DockerFile pour GateWay



* Docker Compose File

version: '3'

services:

eureka-server:

build:

context: ./server

restart: always

ports:

- "8761:8761"

gateway-service:

build:

context: ./gateway\_service

ports:

- "8888:8888"

depends\_on:

- eureka-server

environment:

eureka.client.serviceUrl.defaultZone: http://eureka-server:8761/eureka

mysql:

image: mysql:latest

container\_name: mysql

ports:

- "3306:3306"

restart: unless-stopped

environment:

MYSQL\_ROOT\_PASSWORD: root

phpmyadmin:

image: phpmyadmin/phpmyadmin

environment:

PMA\_HOST: mysql

PMA\_PORT: 3306

MYSQL\_ROOT\_PASSWORD: root

ports:

- "8080:80"

demande-service:

build:

context: ./demande\_service

ports:

- "8082:8082"

depends\_on:

- mysql

- eureka-server

environment:

SPRING\_DATASOURCE\_URL: jdbc:mysql://mysql:3306/demande\_db?createDatabaseIfNotExist=true

SPRING\_DATASOURCE\_USERNAME: root

SPRING\_DATASOURCE\_PASSWORD: root

eureka.client.serviceUrl.defaultZone: http://eureka-server:8761/eureka

healthcheck:

test: "/usr/bin/mysql --user=root --password=root --execute \"SHOW DATABASES;\""

interval: 5s

timeout: 2s

retries: 100

offre-service:

build:

context: ./offre\_service

ports:

- "8087:8087"

depends\_on:

- mysql

- eureka-server

environment:

SPRING\_DATASOURCE\_URL: jdbc:mysql://mysql:3306/offer\_db?createDatabaseIfNotExist=true

SPRING\_DATASOURCE\_USERNAME: root

SPRING\_DATASOURCE\_PASSWORD: root

eureka.client.serviceUrl.defaultZone: http://eureka-server:8761/eureka

healthcheck:

test: "/usr/bin/mysql --user=root --password=root --execute \"SHOW DATABASES;\""

interval: 5s

timeout: 2s

retries: 100

user\_service:

build:

context: ./user\_service

ports:

- "8088:8088"

depends\_on:

- mysql

- eureka-server

environment:

SPRING\_DATASOURCE\_URL: jdbc:mysql://mysql:3306/user\_db?createDatabaseIfNotExist=true

SPRING\_DATASOURCE\_PASSWORD: root

SPRING\_DATASOURCE\_USERNAME: root

eureka.client.serviceUrl.defaultZone: http://eureka-server:8761/eureka

healthcheck:

test: "/usr/bin/mysql --user=root --password=root --execute \"SHOW DATABASES;\""

interval: 5s

timeout: 2s

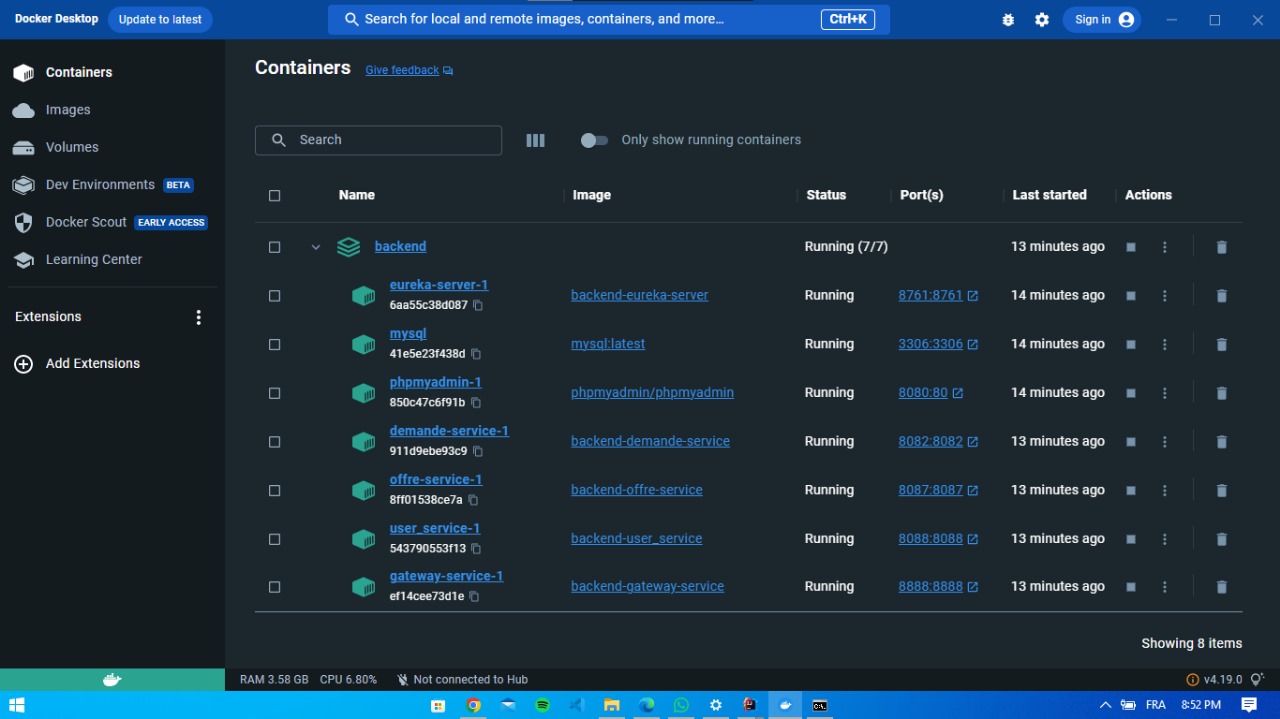
retries: 100

vous devez positioner dans le directory de service et pour chaque service builder une image

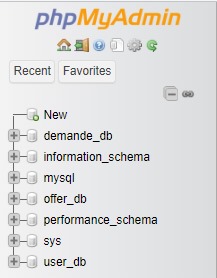
docker buildx build -t service\_name:tag .

dans le directory racine (backend)

docker-compose up -d



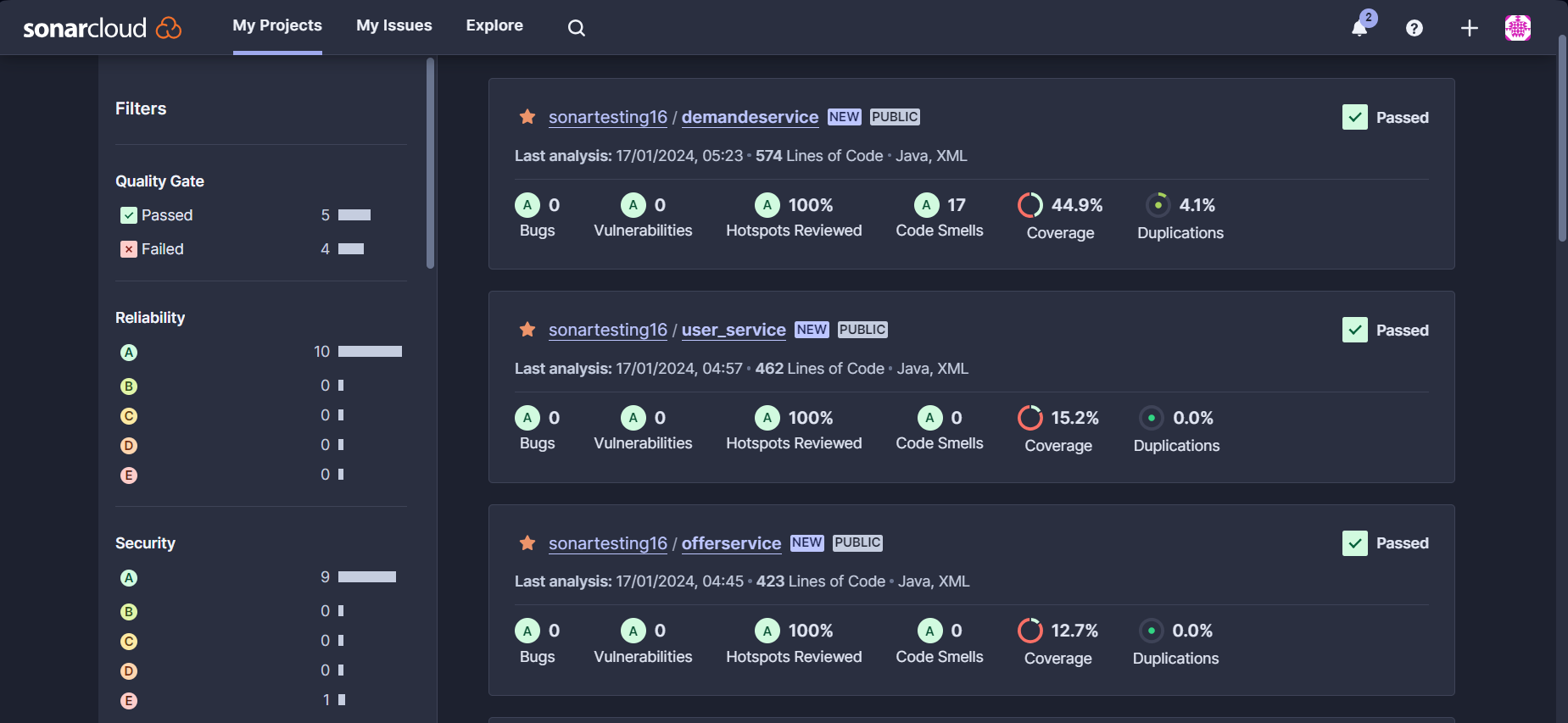
On peut visualizer notre tables on phpmyAdmin :



# Eureka Server



# SonarCloud



# Jankins



