**Département Mathématiques et Informatique**

**Filière :**

**« Génie du Logiciel et des Systèmes Informatiques Distribués »**

**GLSID3**

**Examen Blanc “Systèmes Distribués”**

**Sujet : Système Réactif d’automatisation du processus des contraventions routières liées aux excès vitesses détectés par des radars automatiques**

**Module : Architecture des Systèmes Distribués**

Élaboré par :

ELMAJNI Khaoula

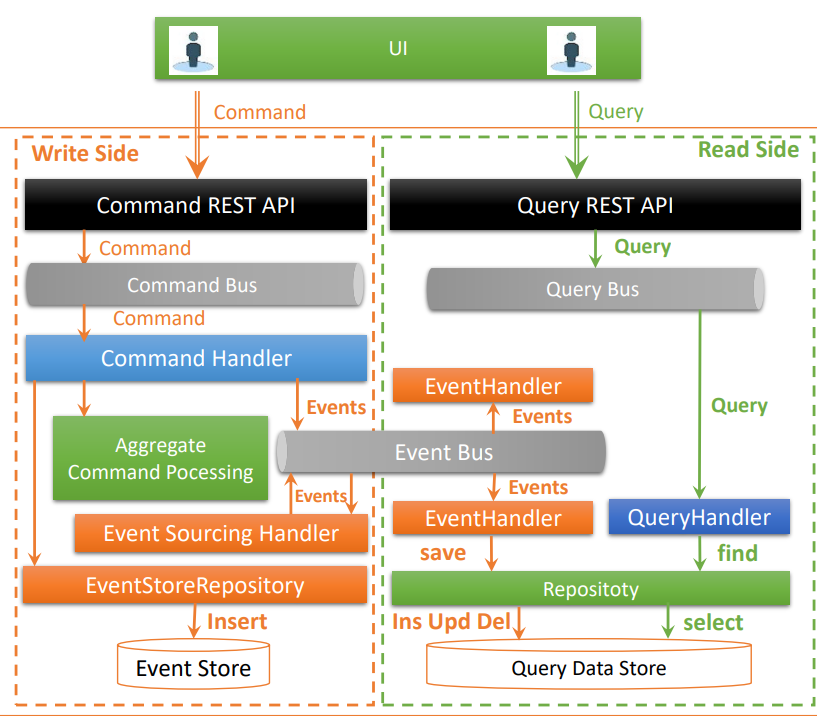
Encadré par :

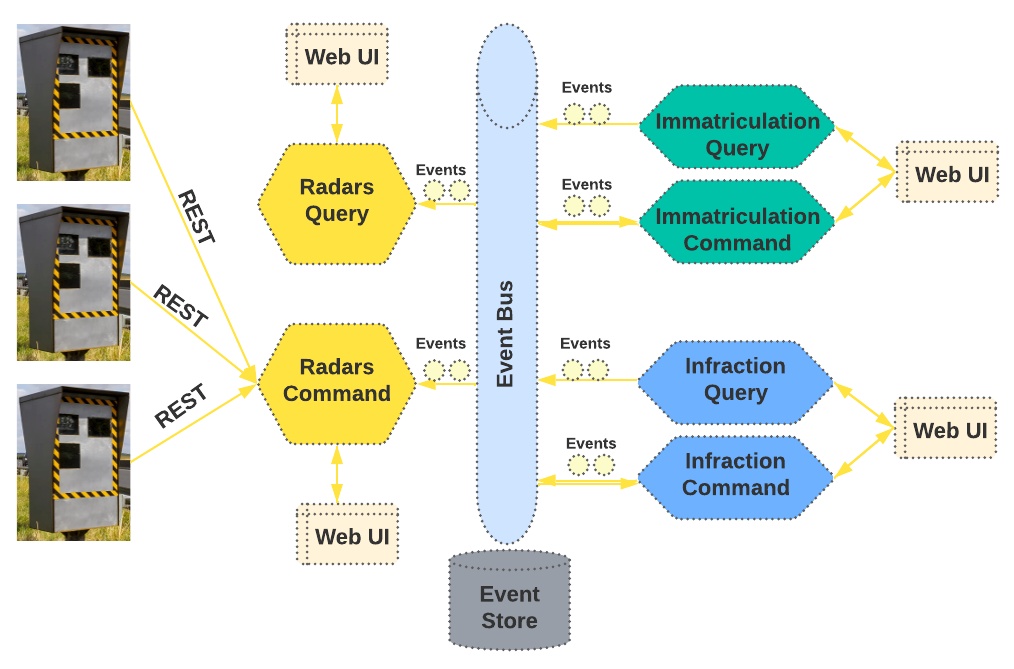
M. EL YOUSSFI Mohammed

**Année Universitaire : 2022-2023**

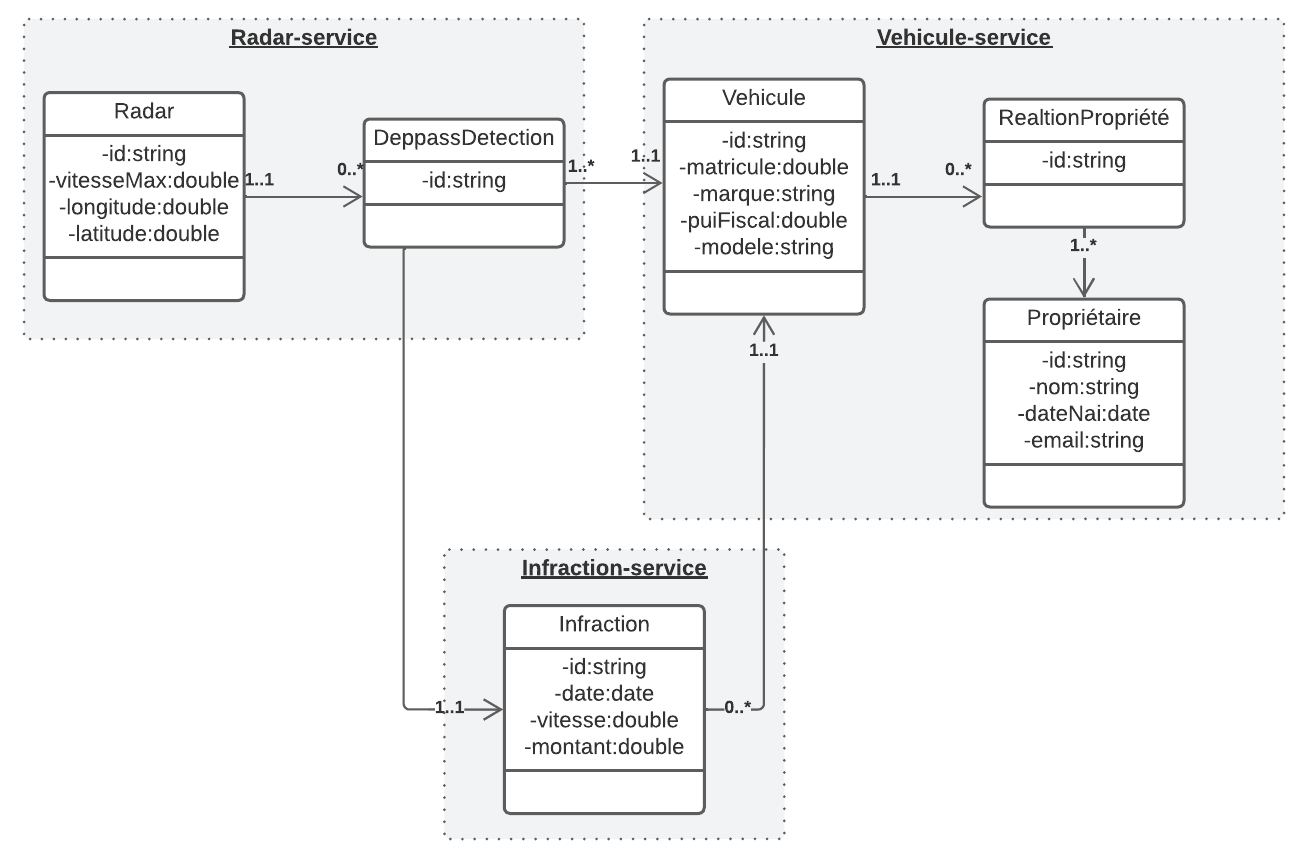
1. Etablir une architecture technique du projet

Architecture CQRS :

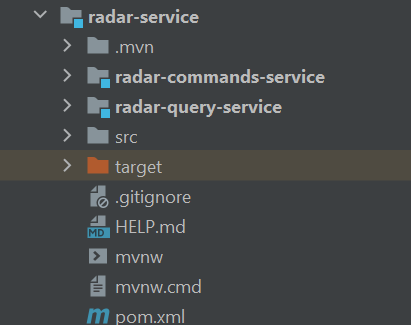


Architecture du système :

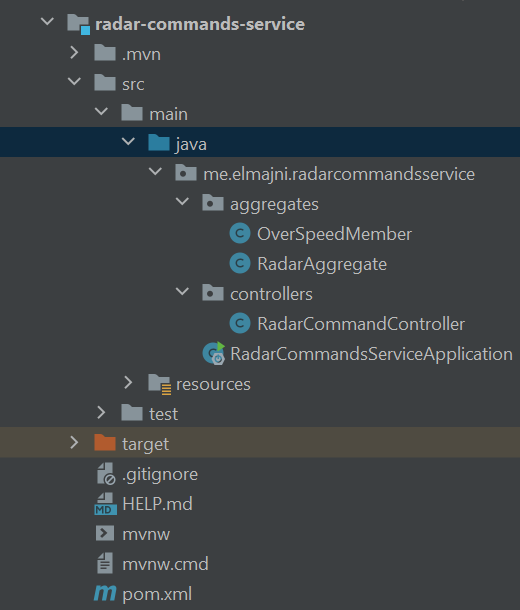
1. Etablir un diagramme de classe global du projet



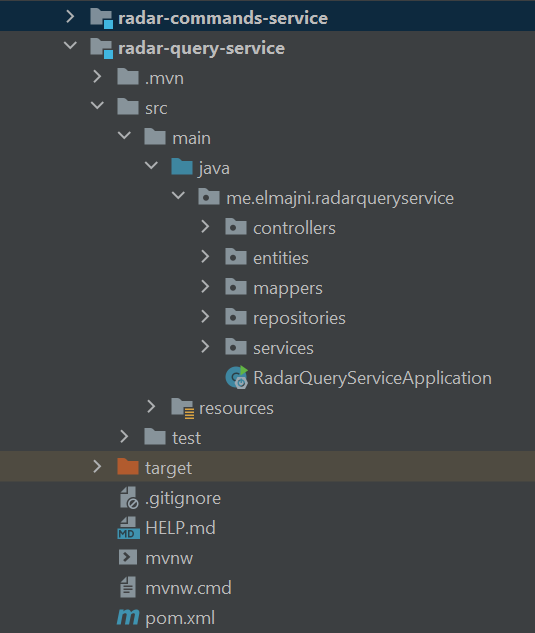
1. Développer le micro-service Radar



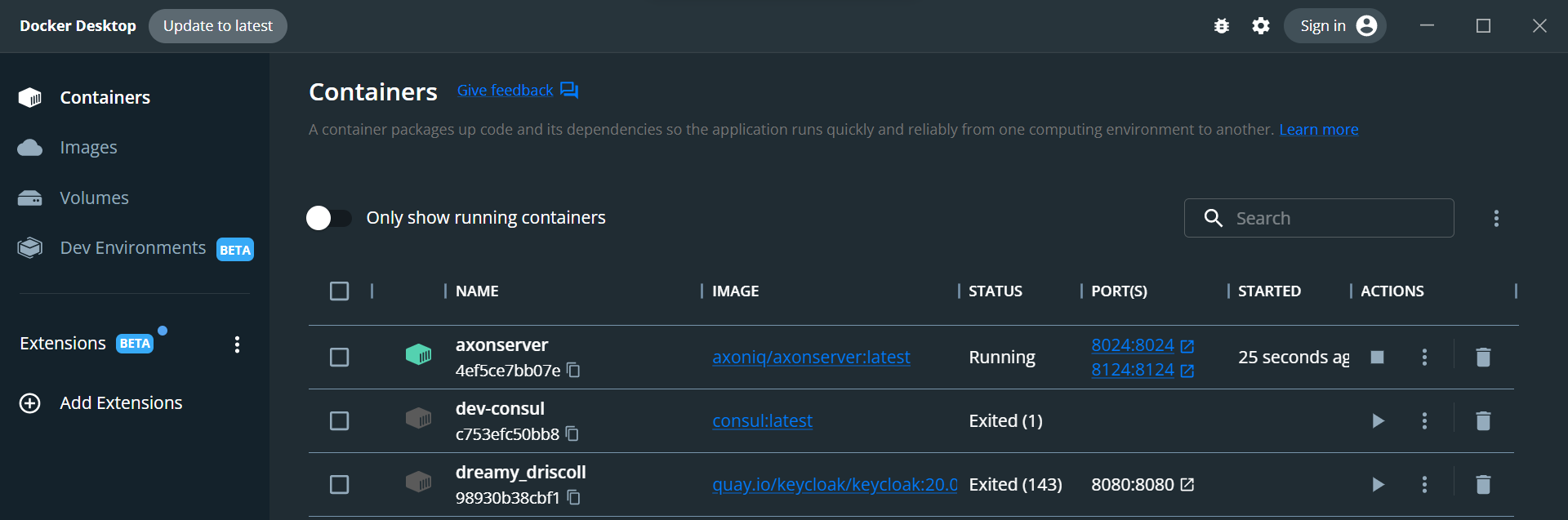
**Le service « radar-commands-service » :**

****

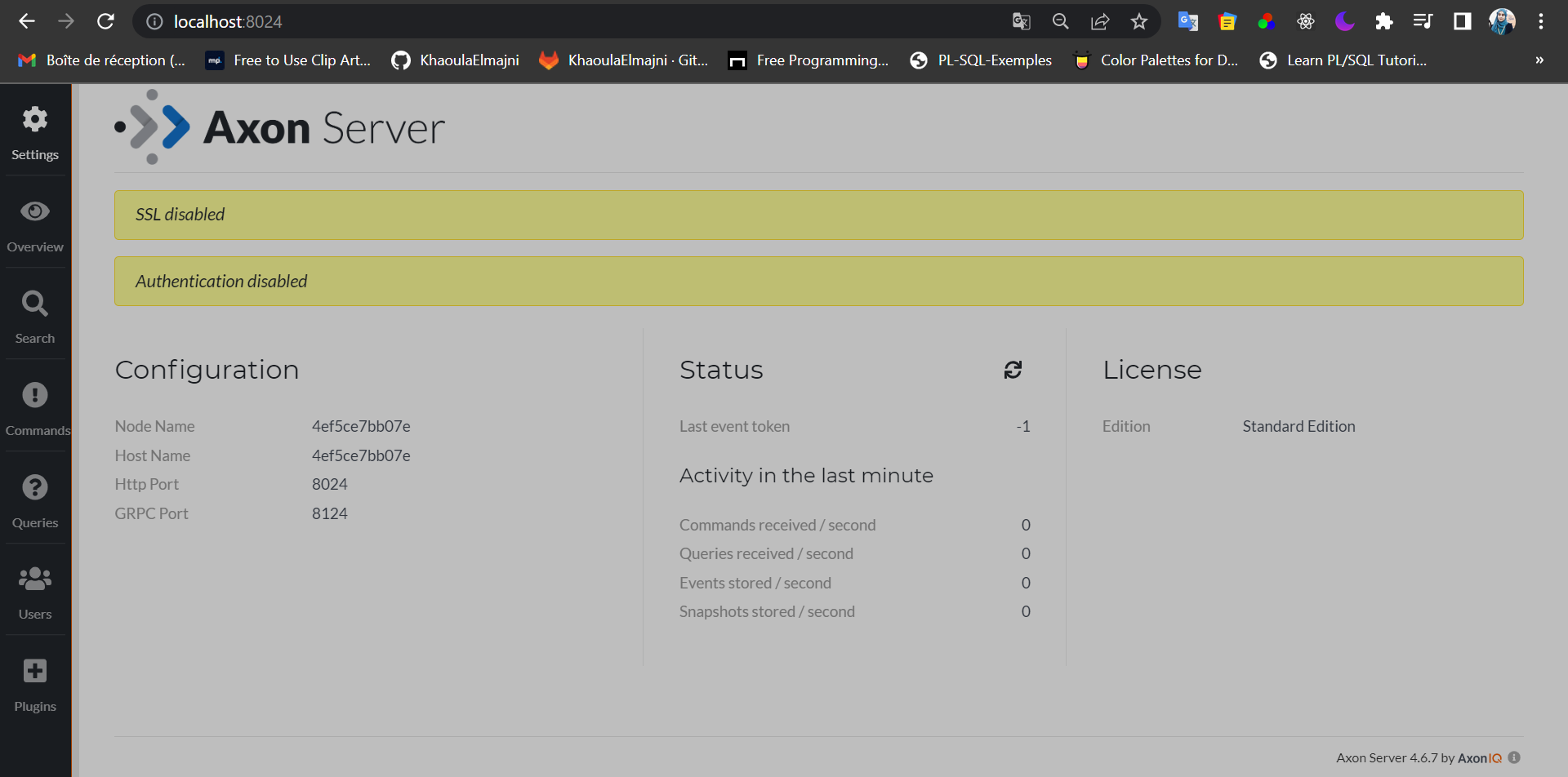
**Le service « radar-query-service » :**

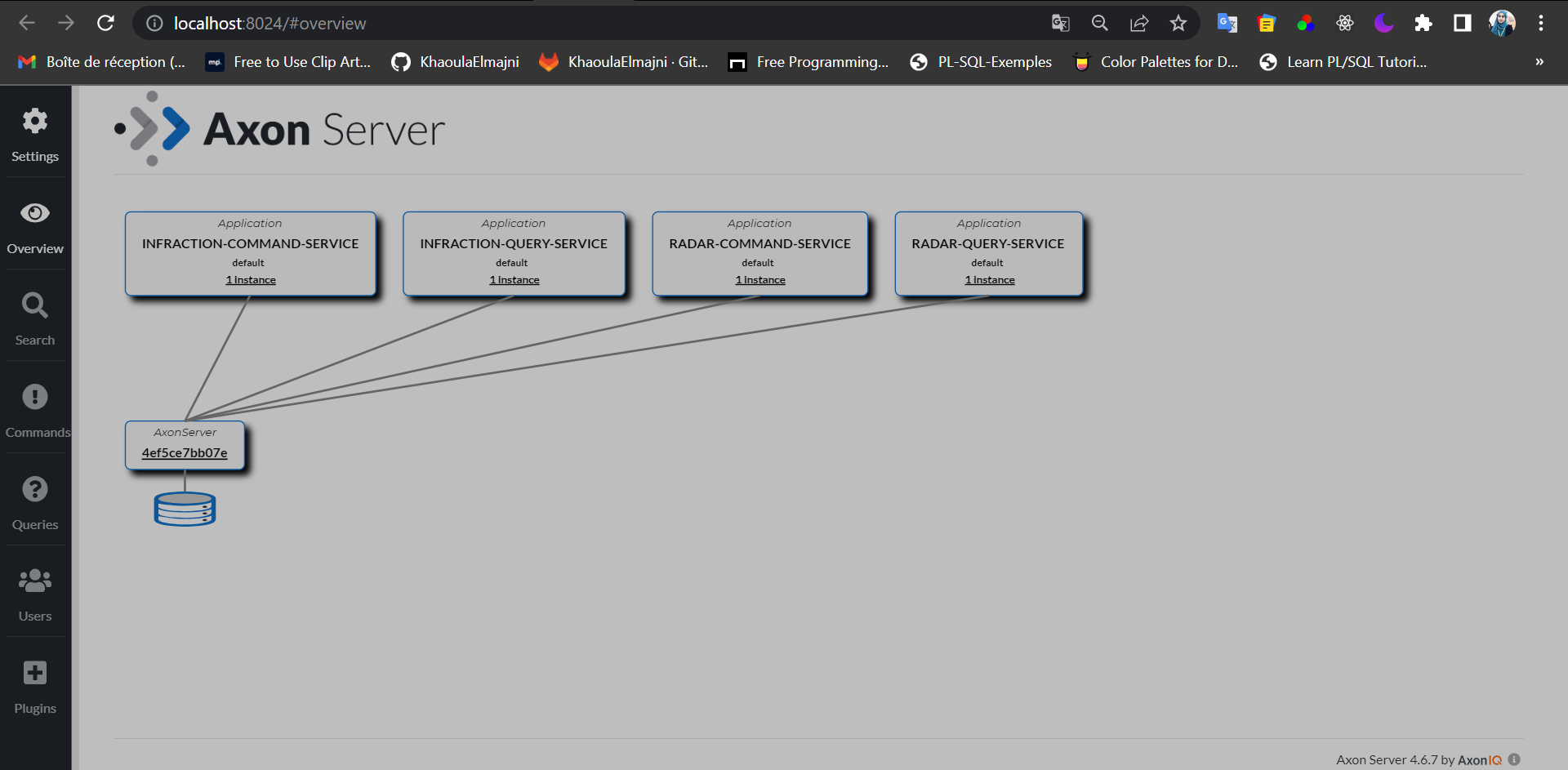


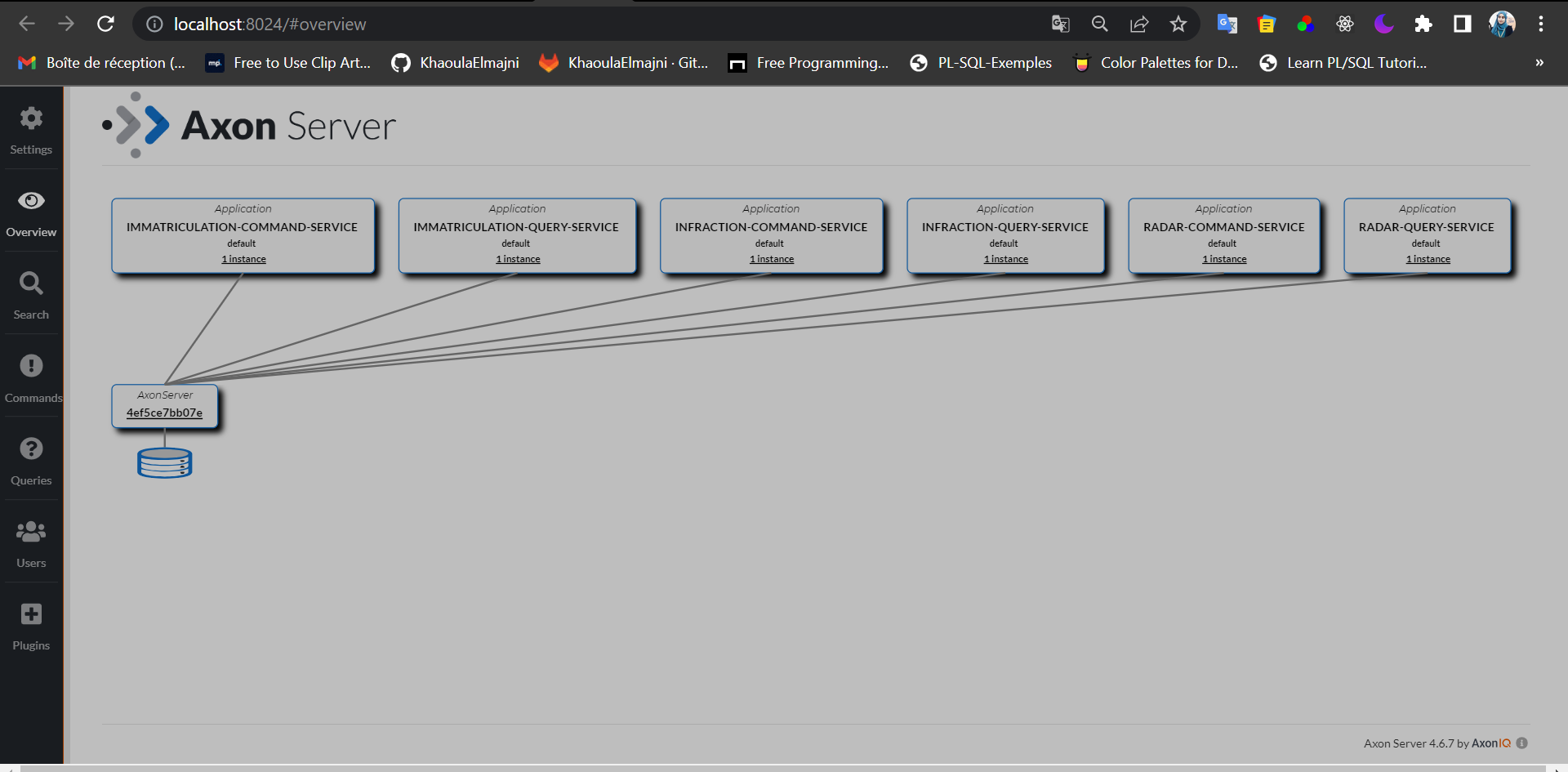
Lancement de Axon dans Docker compose :



Axon dashboard :

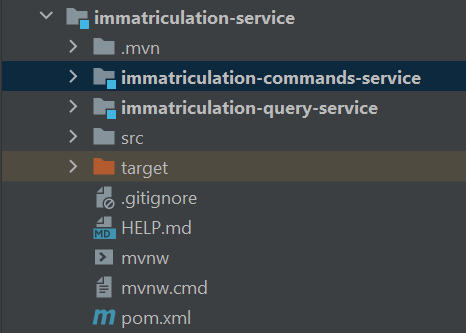




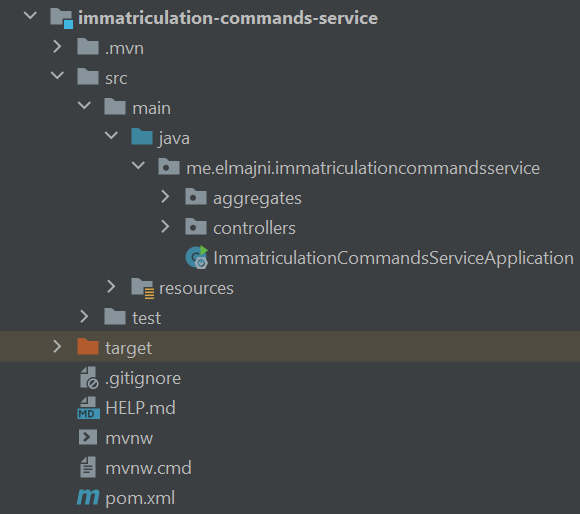




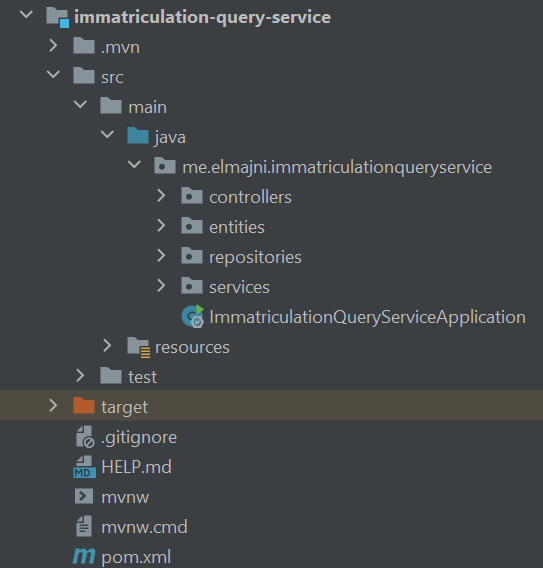
1. Développer le micro-service Immatriculation



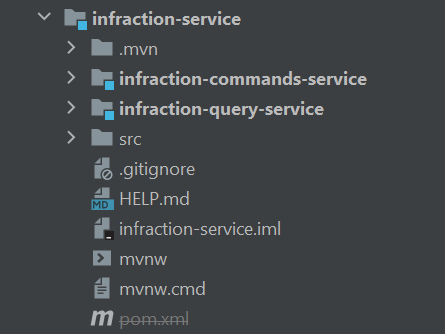
**Le service « Immatriculation-commands-service » :**

****

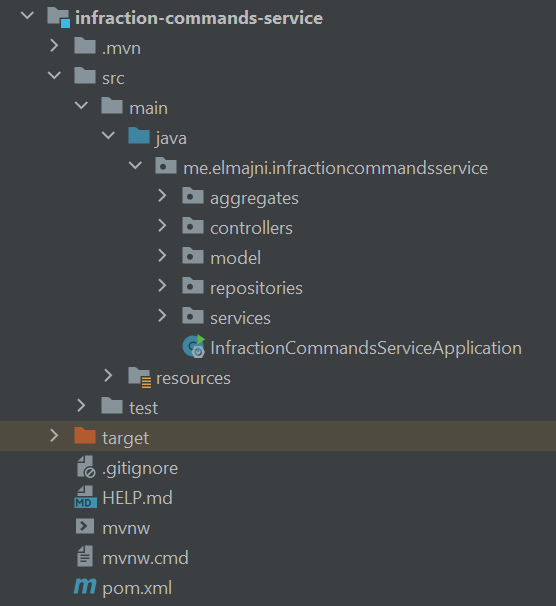
**Le service « Immatriculation-query-service » :**

****

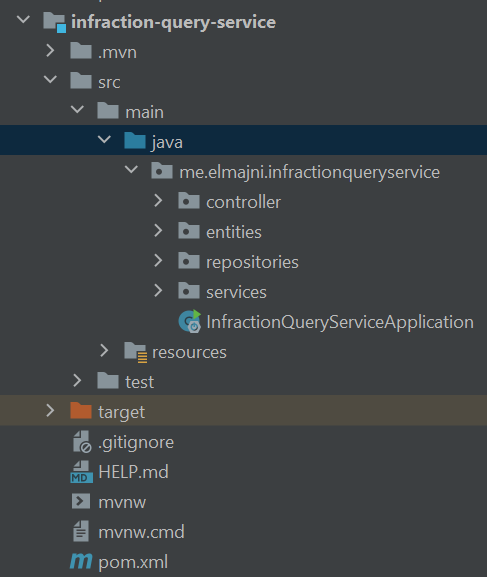
1. Développer le micro-service Infractions

****

**Le service « Infractions-commands-service » :**

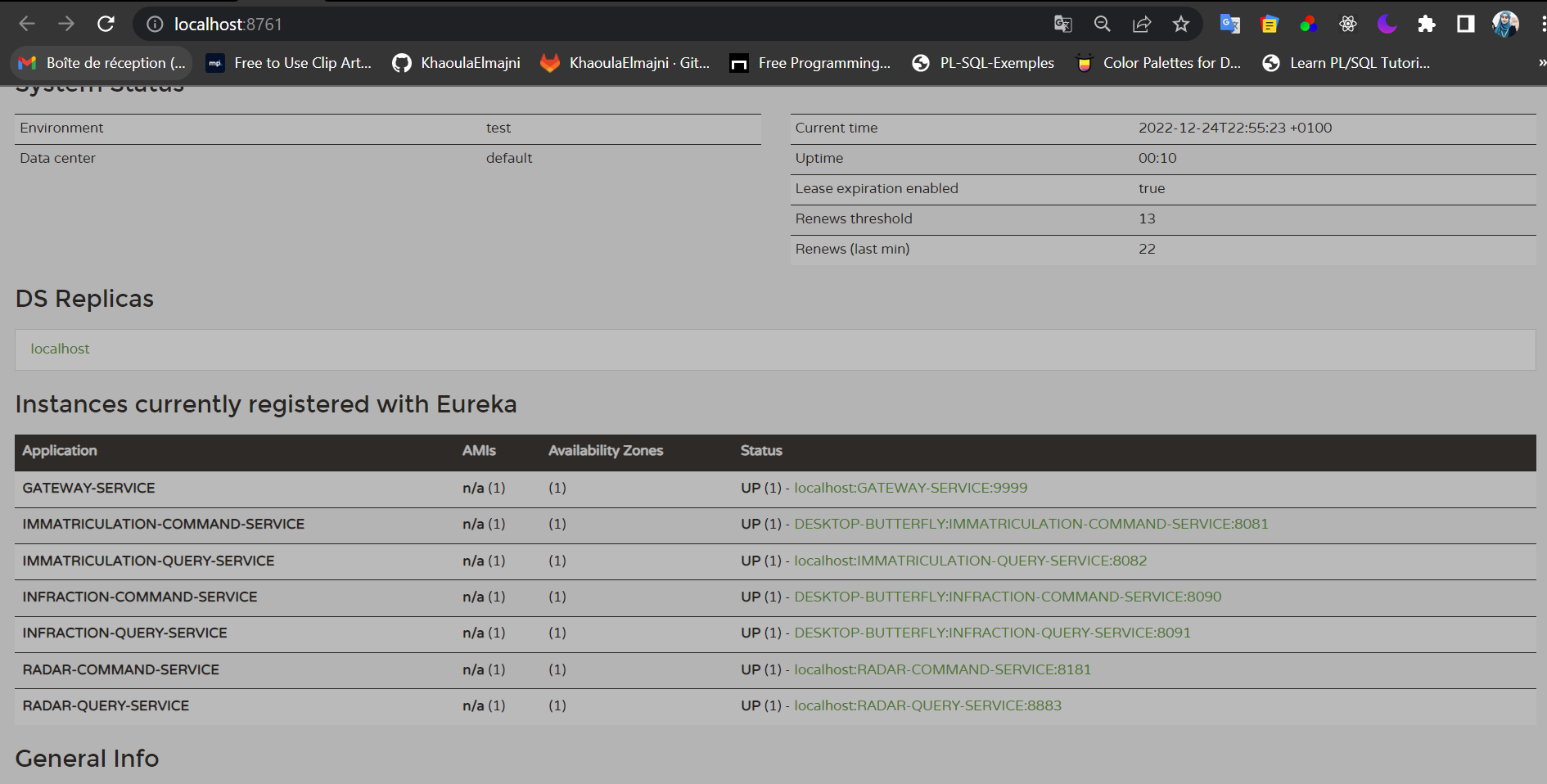
****

**Le service « Infractions** **-query-service » :**

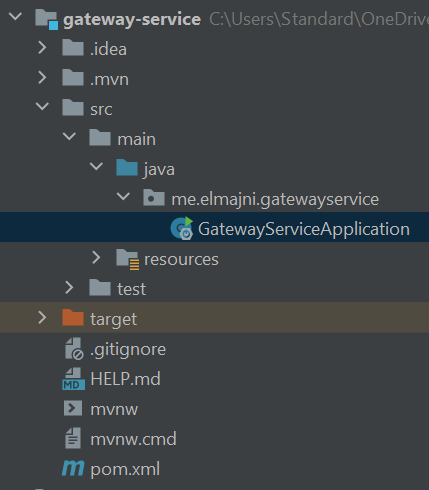


1. Mettre en place les services techniques de l’architecture micro-service (Gateway, Eureka Discovery service)

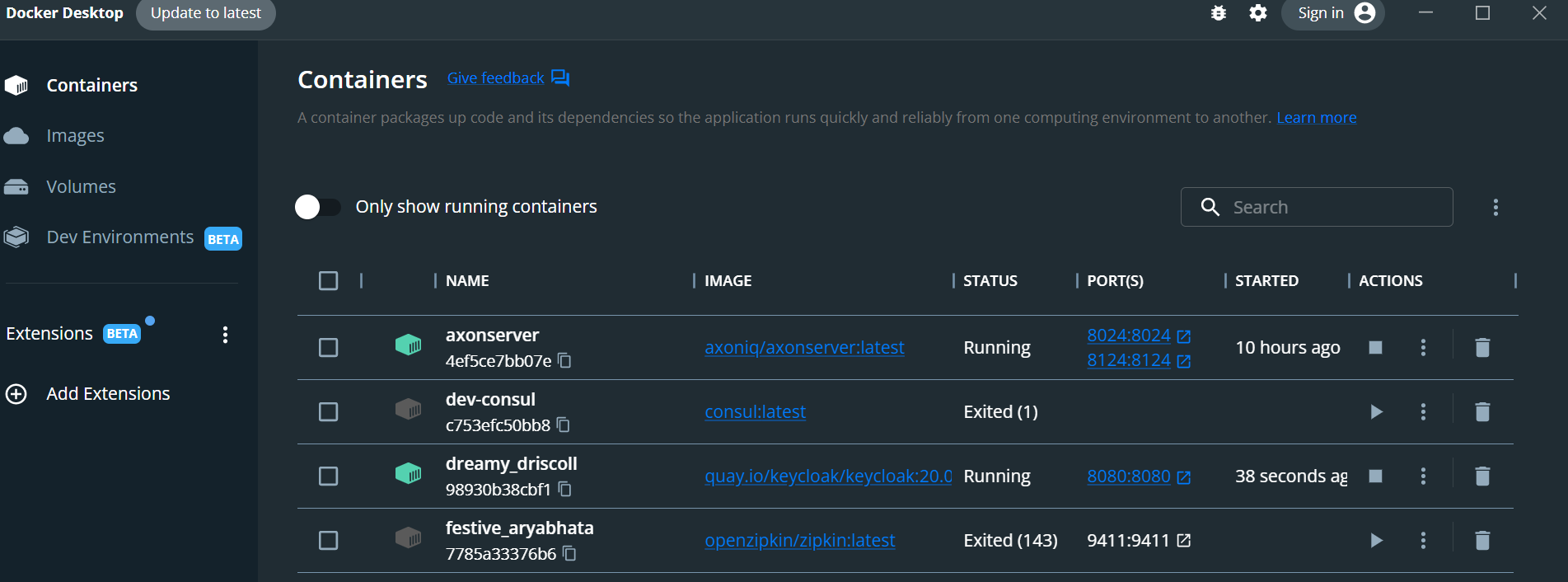
**Le service « Eureka-discovery-service » :**



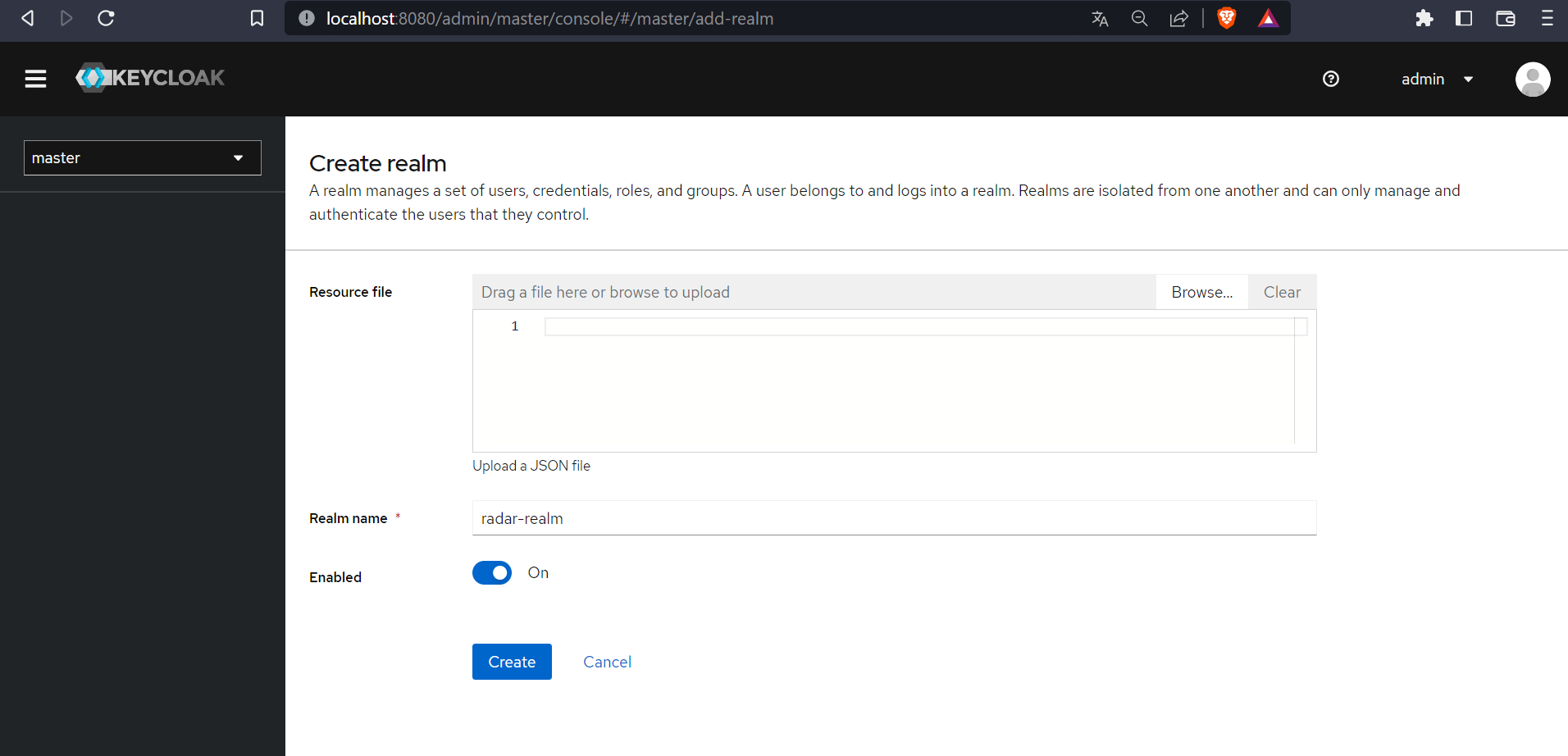
**Le service « Gateway-service » :**



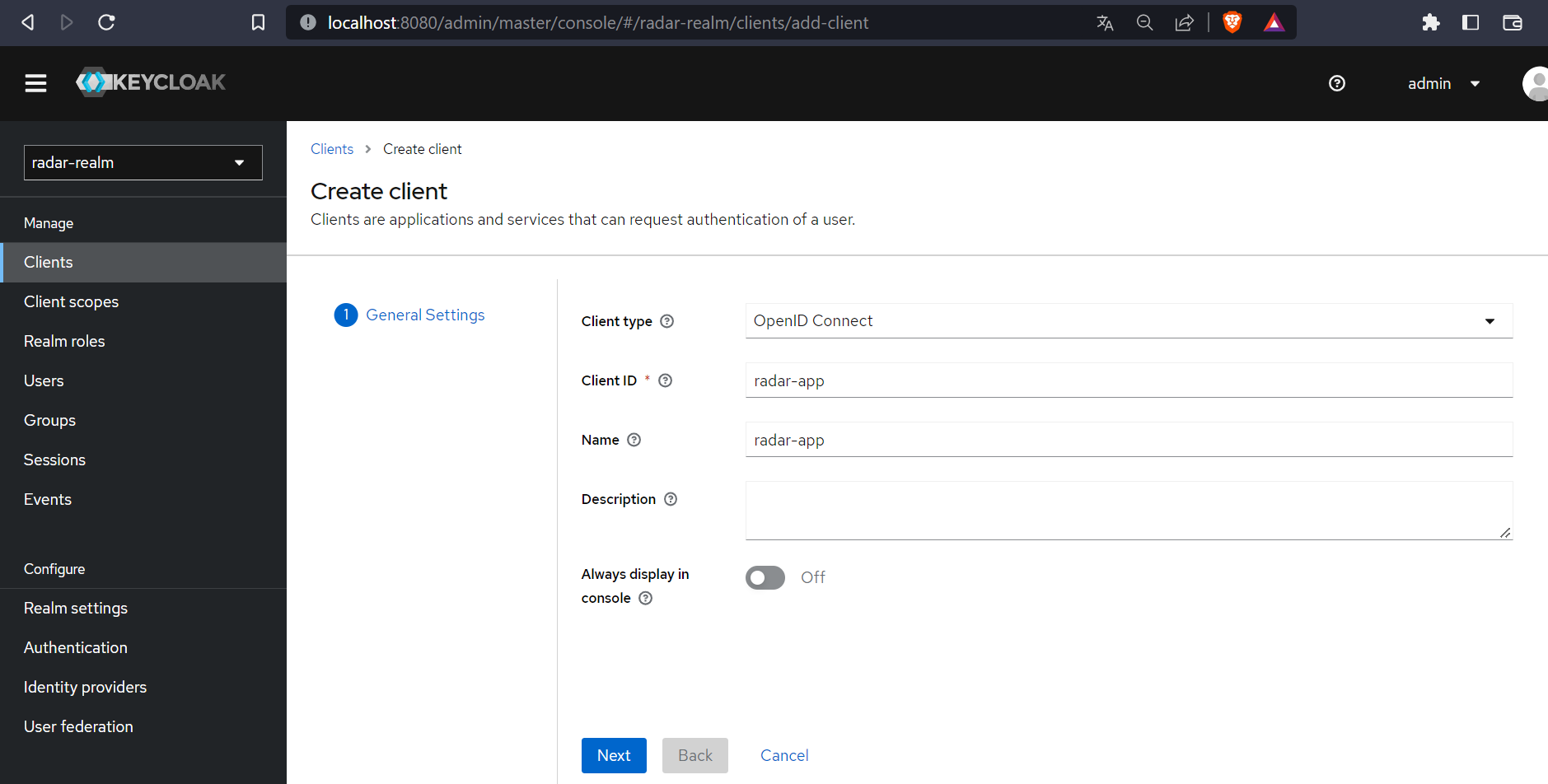
1. Développer votre application Frontend avec Angular ou React
2. Sécuriser votre système avec un système de d’authentification OAuth2 comme Keycloak

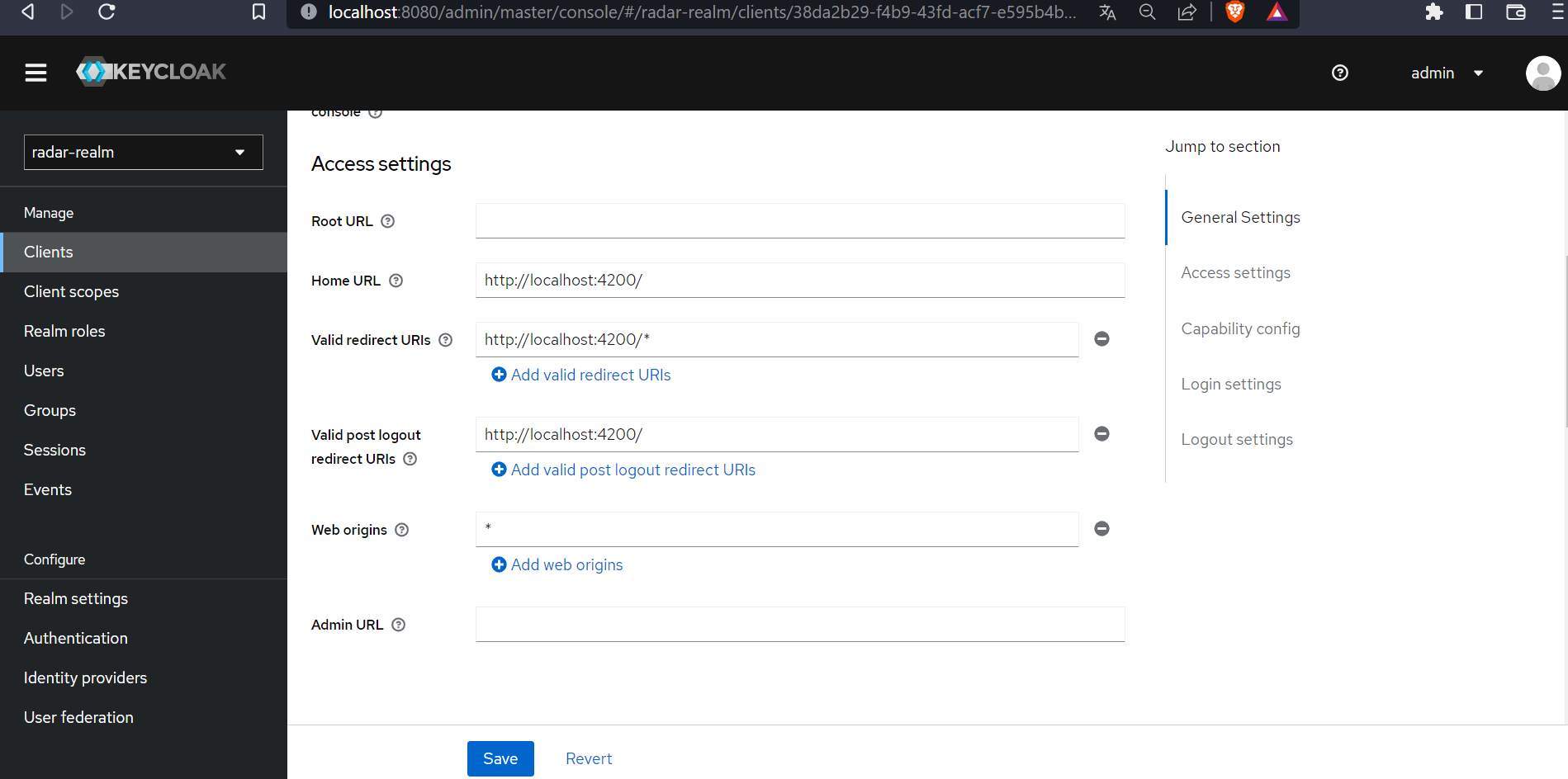


Création d’un realm :

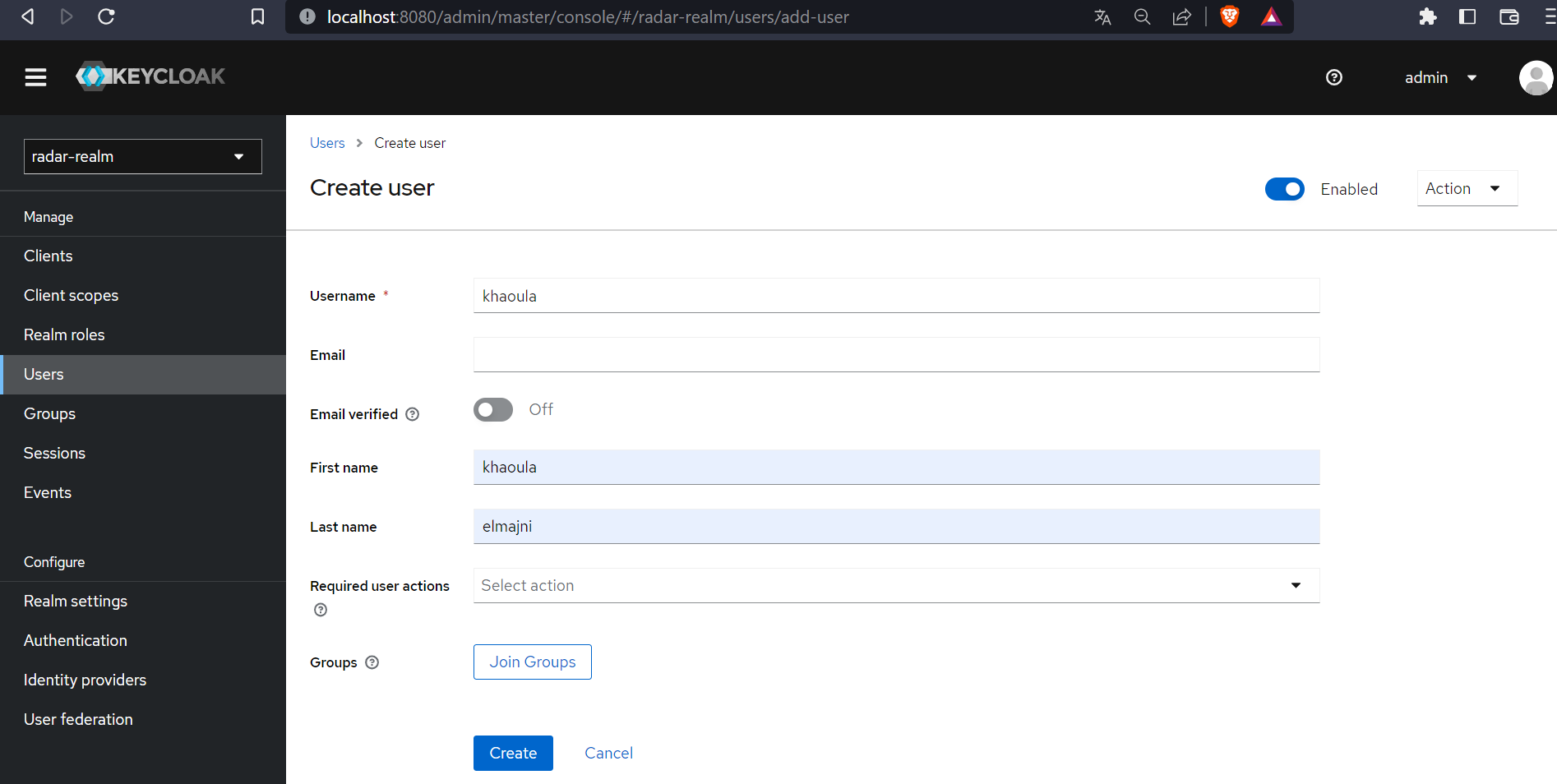


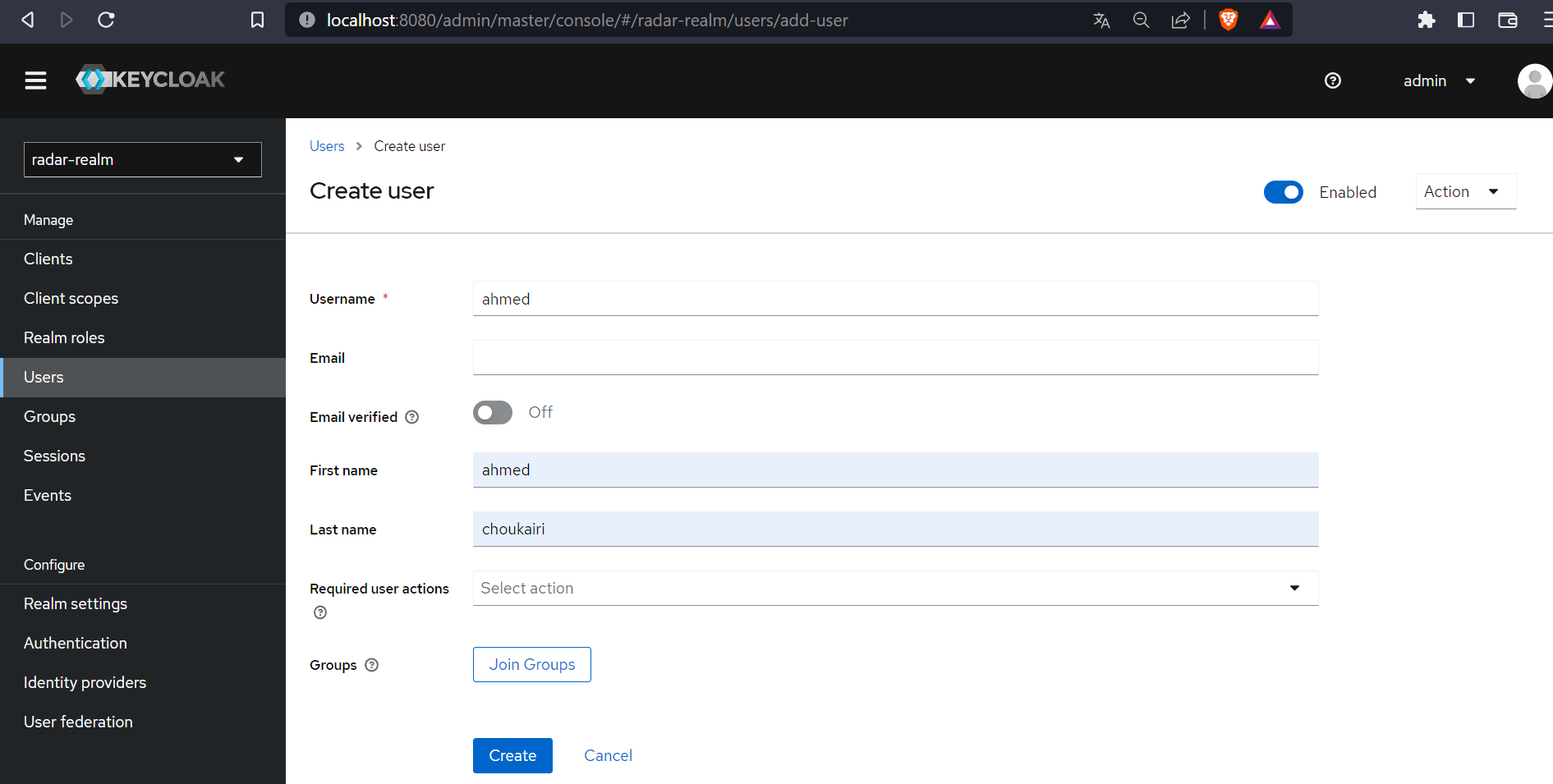
Création d’un client :

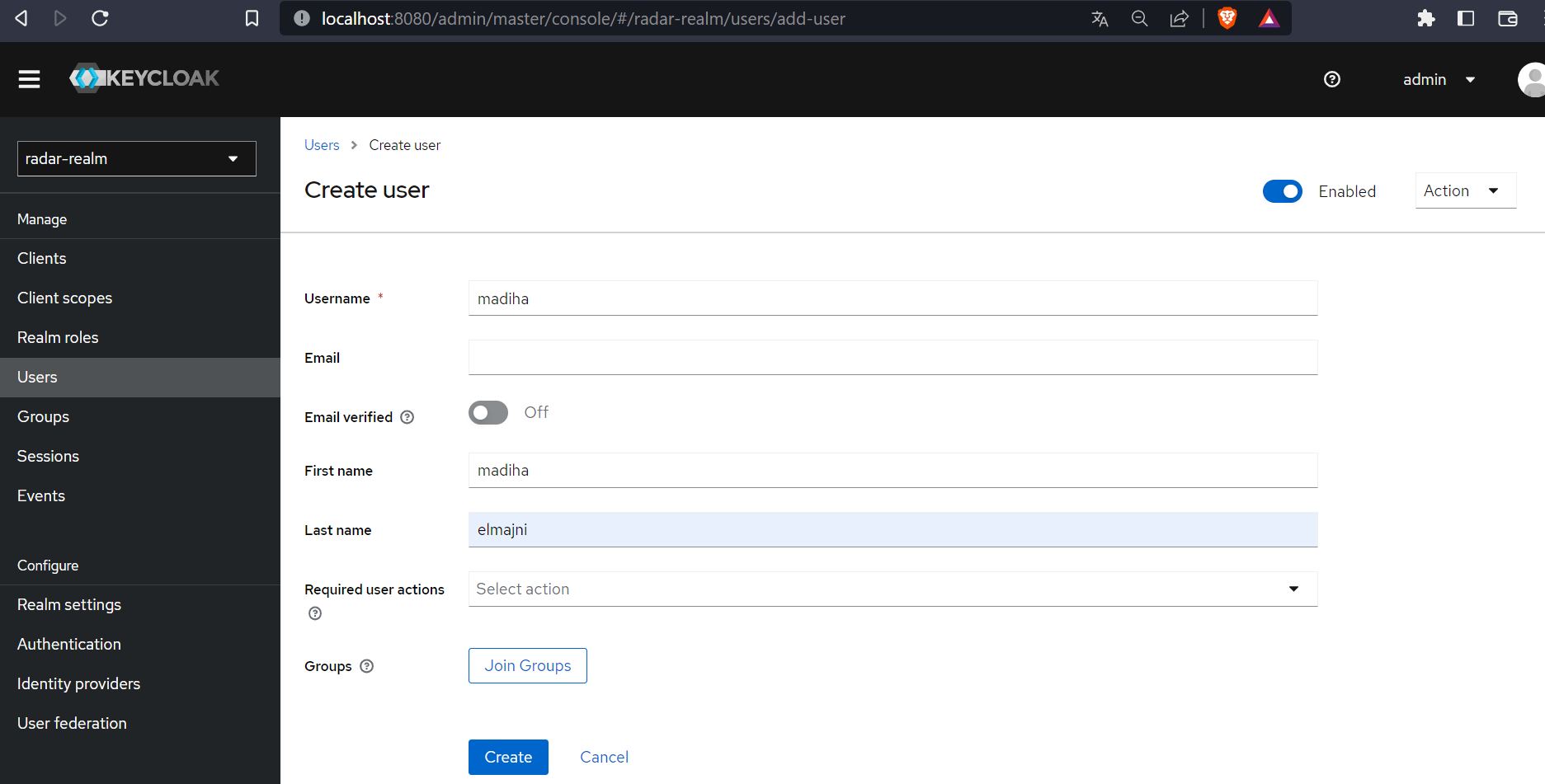




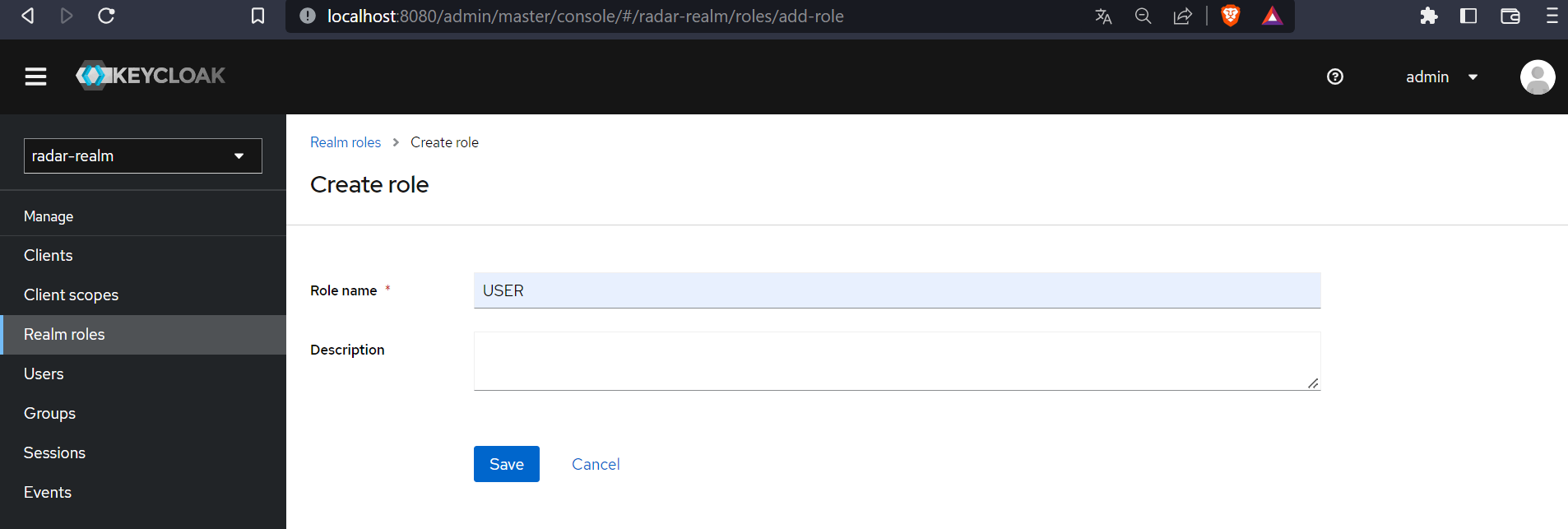
Création des utilisateurs :

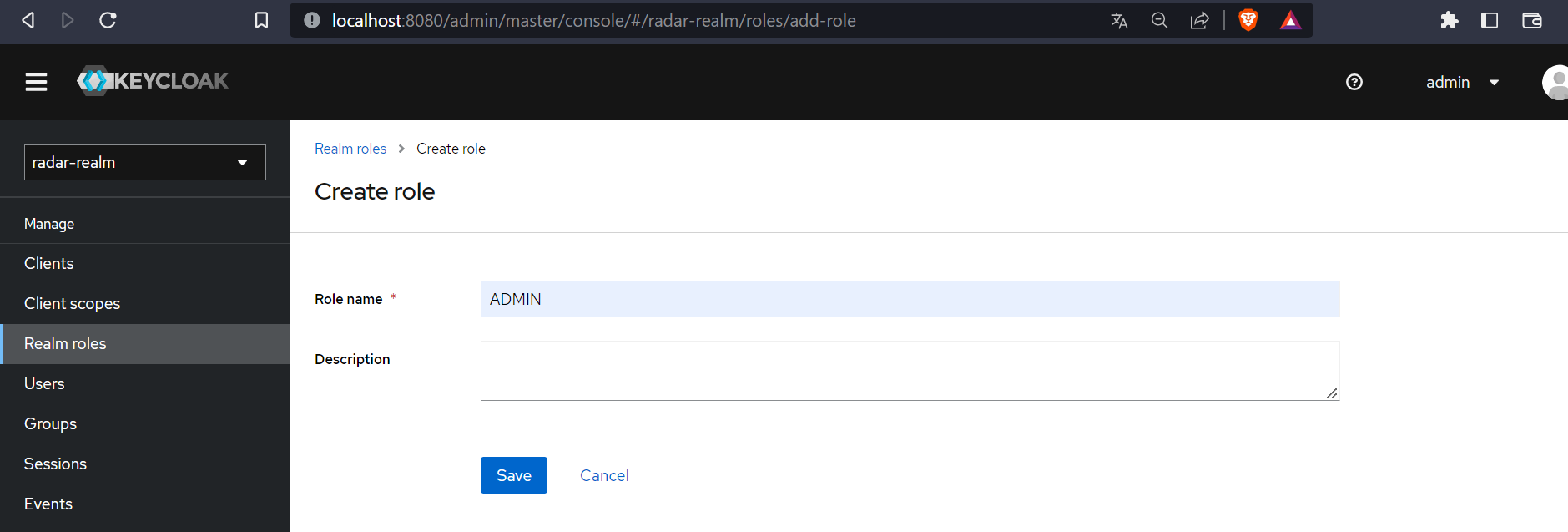


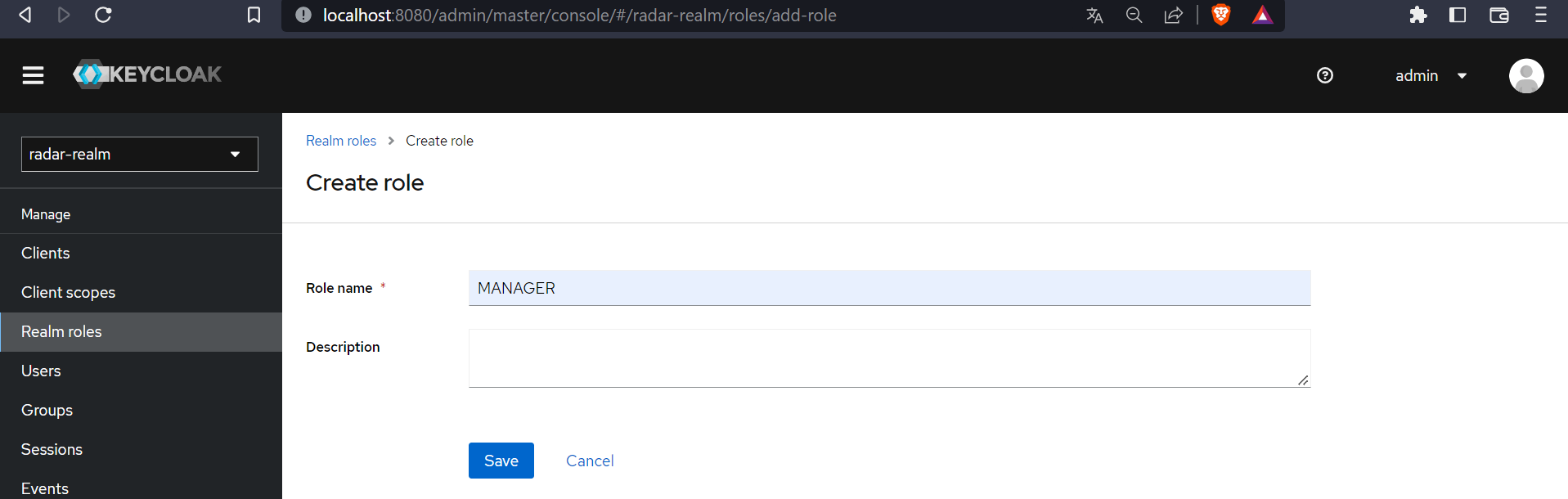




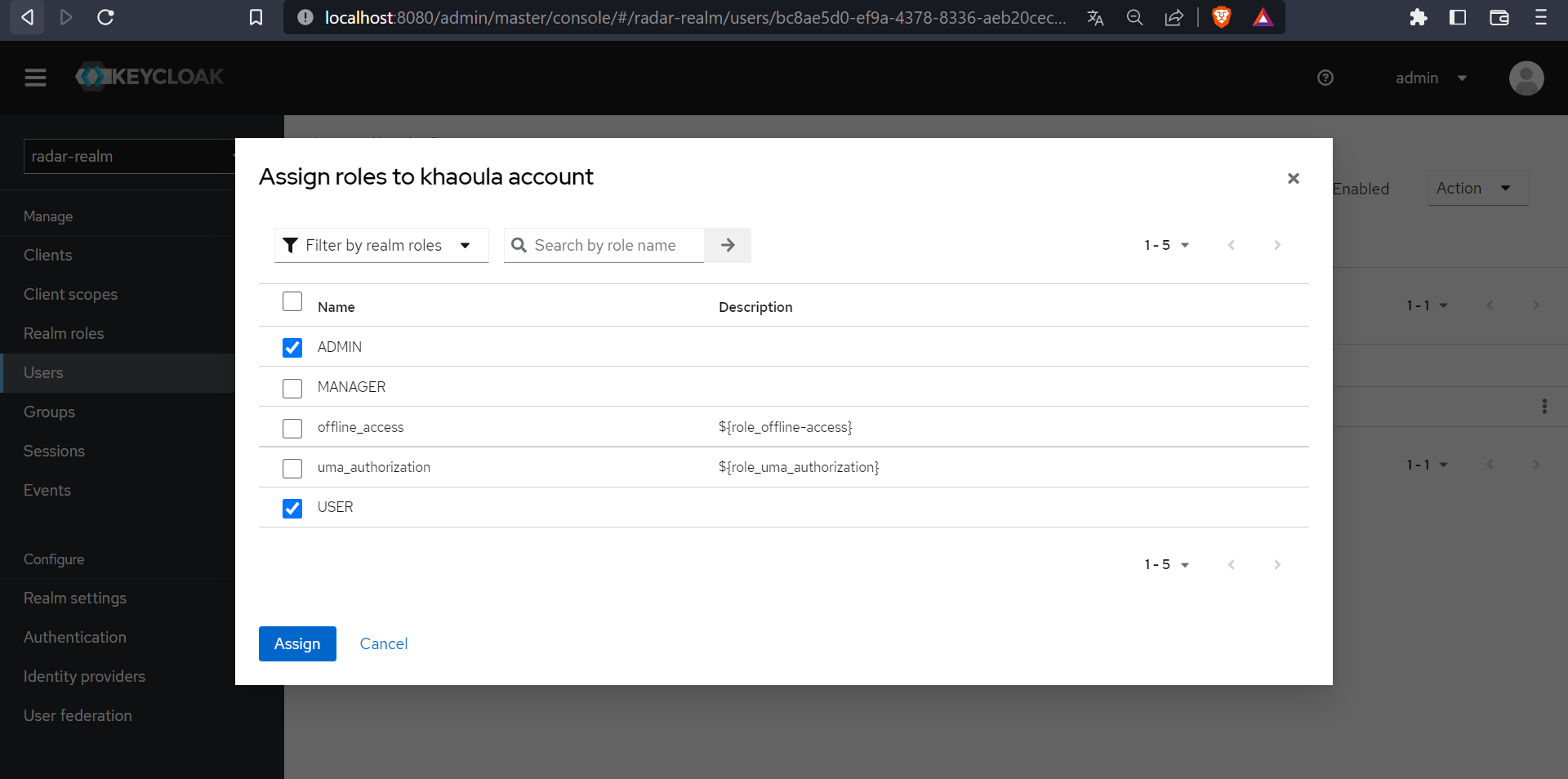
Création des roles :

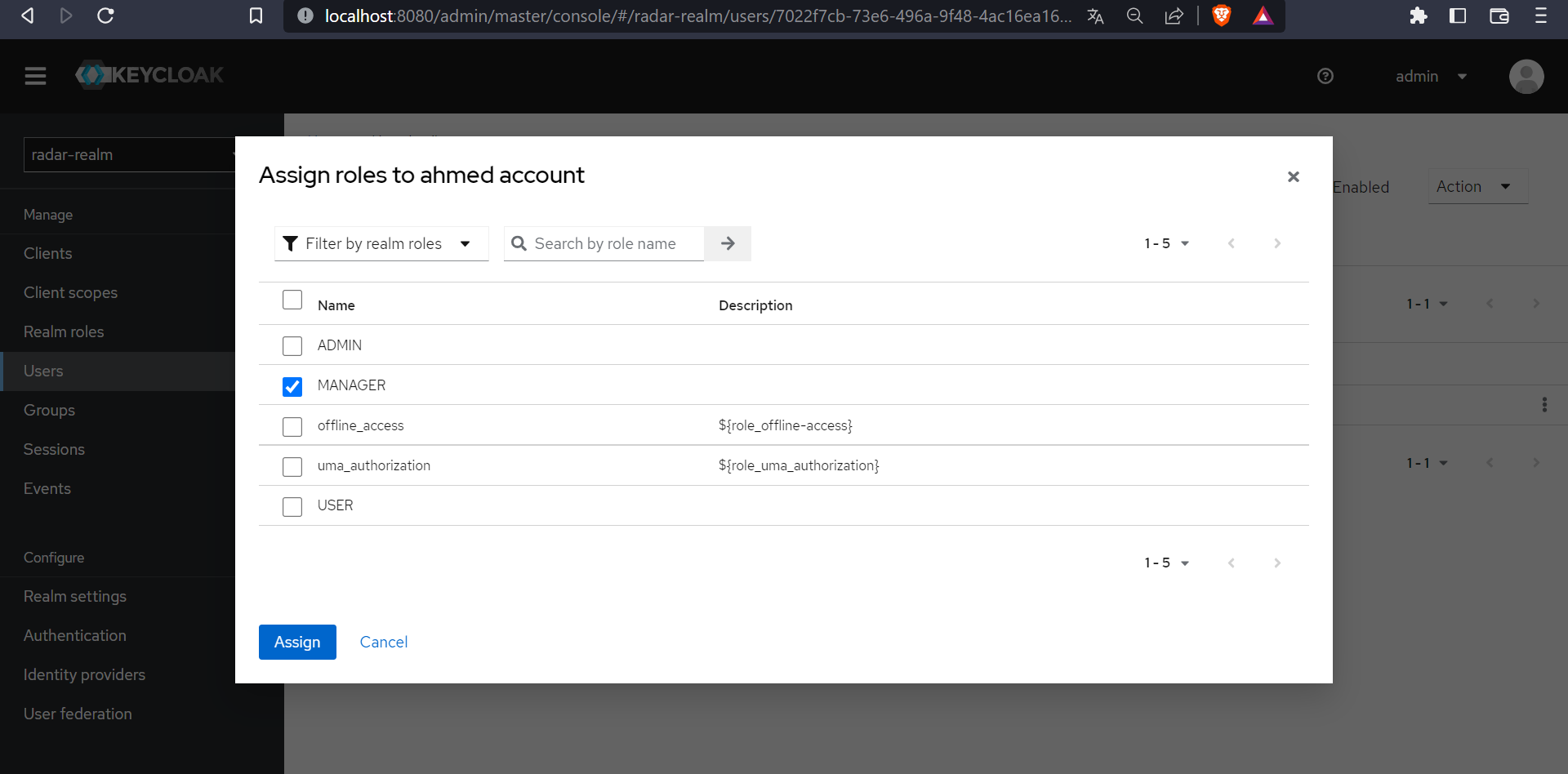


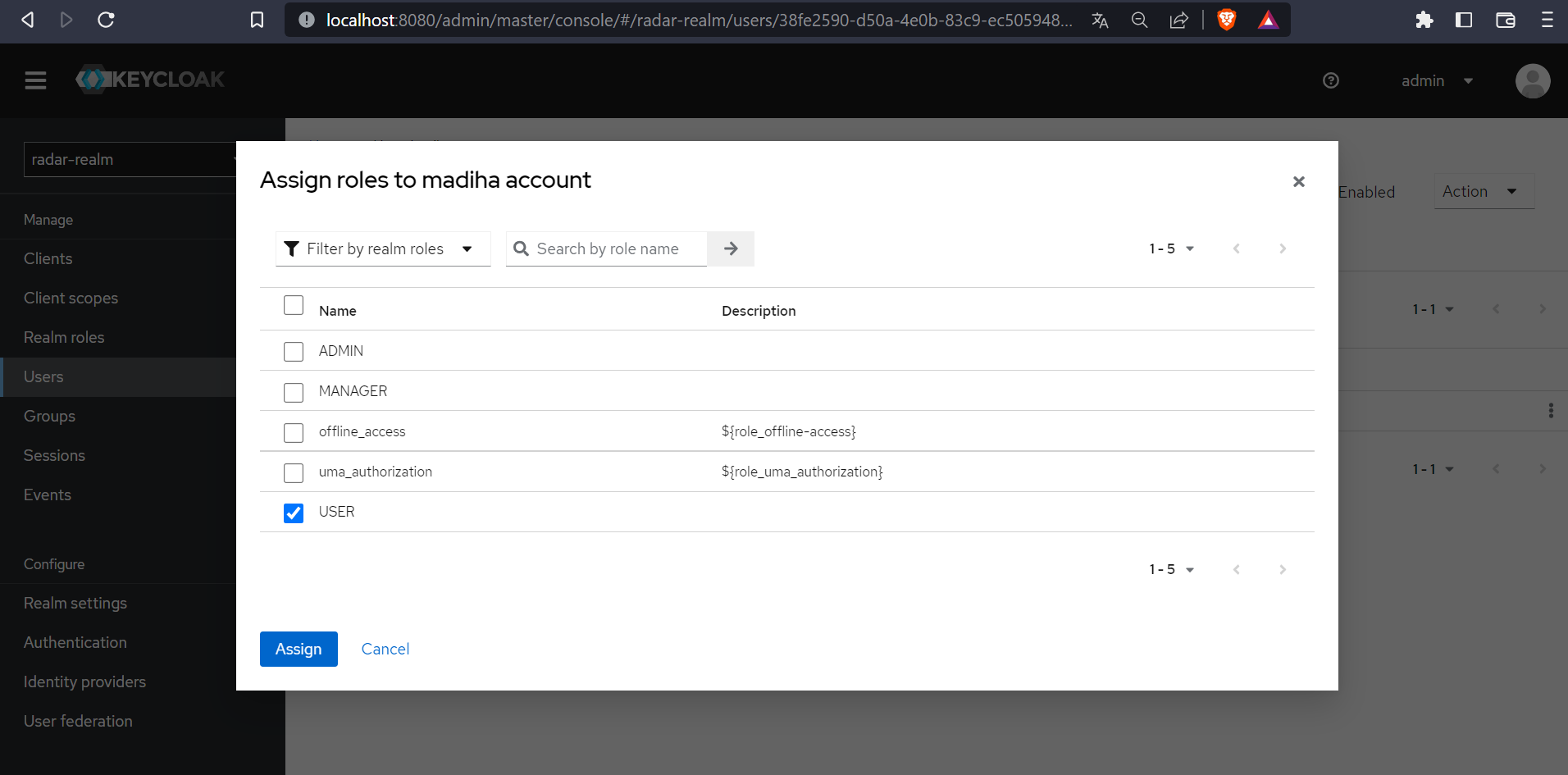




Assignation des roles aux utilisateurs :



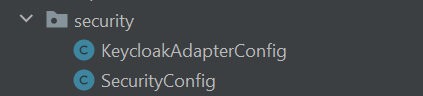




Ajouter les dépendances suivantes dans les 6 micro services:

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-security</artifactId>  
</dependency>  
  
<dependency>  
 <groupId>org.keycloak</groupId>  
 <artifactId>keycloak-spring-boot-starter</artifactId>  
 <version>20.0.1</version>  
</dependency>

Après, la partie de sécurité dans chaque micro-service :



1. Ecrire un script docker-compose.yml pour le déploiement de ce système distribué dans des conteneurs docker.

Le script docker-compose :

version: '3'  
services:  
 eureka-service:  
 build: ./discovery-service/  
 hostname: discovery-service  
 ports:  
 - "8761:8761"  
 networks:  
 - default-network  
  
 immatriculation-commands-service:  
 build: ./immatriculation-commands-service/  
 hostname: immatriculation-commands-service  
 ports:  
 - "8081:8081"  
 depends\_on:  
 - eureka-service  
 environment:  
 - eureka.client.service-url.defaultZone=http://discovery-service:8761  
 networks:  
 - default-network  
  
 immatriculation-query-service:  
 build: ./immatriculation-query-service/  
 hostname: immatriculation-query-service  
 ports:  
 - "8082:8082"  
 restart: on-failure  
 depends\_on:  
 - eureka-service  
 environment:  
 - eureka.client.service-url.defaultZone=http://discovery-service:8761  
 networks:  
 - default-network  
  
 infraction-command-service:  
 build: ./infraction-command-service/  
 hostname: infraction-command-service  
 ports:  
 - "8090:8090"  
 restart: on-failure  
 depends\_on:  
 - eureka-service  
 environment:  
 - eureka.client.service-url.defaultZone=http://discovery-service:8761  
 networks:  
 - default-network  
  
 infraction-query-service:  
 build: ./infraction-query-service/  
 hostname: infraction-query-service  
 ports:  
 - "8091:8091"  
 restart: on-failure  
 depends\_on:  
 - eureka-service  
 environment:  
 - eureka.client.service-url.defaultZone=http://discovery-service:8761  
 networks:  
 - default-network  
  
 radar-command-service:  
 build: ./radar-command-service/  
 hostname: radar-command-service  
 ports:  
 - "8181:8181"  
 restart: on-failure  
 depends\_on:  
 - eureka-service  
 environment:  
 - eureka.client.service-url.defaultZone=http://discovery-service:8761  
 networks:  
 - default-network  
  
 radar-query-service:  
 build: ./radar-query-service/  
 hostname: radar-query-service  
 ports:  
 - "8883:8883"  
 restart: on-failure  
 depends\_on:  
 - eureka-service  
 environment:  
 - eureka.client.service-url.defaultZone=http://discovery-service:8761  
 networks:  
 - default-network  
  
 gateway-service:  
 build: ./gateway-service/  
 hostname: gateway-service  
 ports:  
 - "9999:9999"  
 depends\_on:  
 - eureka-service  
 - immatriculation-query-service  
 - immatriculation-command-service  
 - infraction-command-service  
 - infraction-query-service  
 - radar-command-service  
 - radar-query-service  
 environment:  
 - eureka.client.service-url.defaultZone=http://discovery-service:8761/eureka  
 networks:  
 - default-network  
  
networks:  
 default-network:  
 driver: bridge

