

Arhitectura sistem

Sistemul propus contine o gama diversificata de servicii, ruland in Docker Swarm. Printre acestea se numara:

- Prometheus
- Kafka
- Portainer
- Nginx
- Backend
- Frontend
- Postgres

Sistemul include si un server de Keycloak folosit pentru autorizare si autentificare, dar ruleaza in afara Docker Swarm.

Pentru a asigura conectivitatea, am creat o retea de tip overlay numita `cluster_shared_network`, retea in care am adaugat fiecare serviciu, cu exceptia Keycloak, pe care l-am rulat in retea host si am permis accesul prin port forwarding.

In cadrul sistemului, am expus si mapat o plaja larga de porturi: 5432, 9092, 9443, etc. pentru a putea testa si accesa si din retea host.

Mai jos se poate observa conectivitatea si management-ul porturilor din cadrul sistemului :

Host

Docker overlay network - cluster_shared_network



Nginx a fost folosit drept reverse proxy, si redirectioneaza traficul extern de pe portul 80 (http), la serviciul de backend care asculta pe portul 8888.

Prometheus va monitoriza metricele serviciului Docker, dar si metricele proprii, si le va expune pe portul 9090.

Postgres serveste drept baza de date atat pentru backend cat si pentru Keycloak.

Kafka serveste rolul cozii de mesaje, fiind folosita doar topica email-task, topica pe care vin stringuri json valide, dupa formatul:

```
{ "recipient": "target@xxxxx.com", "message": "xxxxxxxxxxxxxxxx" }
```

Mailsender este un serviciu individual, care asculta non stop mesajele de pe topica email-tasks, le consuma, le parseaza, si trimite email-uri folosind un cont de gmail specializat, datele despre acesta fiind parsate prin env.

Portainer este folosit pentru a gestiona containerele si serviciile docker.

Majoritatea serviciilor primesc informatii despre celelalte servicii prin intermediul variabilelor de mediu.

Pentru partea de CI, am setat github actions prin care se construiesc imaginile de docker, si se incarca apoi pe dockerhub.