Lab 7

# Analiza

## Sieci

Stworzyłam 4 sieci:

- net\_frontend – (WebUI a APIGateway)

- net\_backend\_api – (APIGateway a backend)

- net\_product\_db – (ProductService a baza danych)

- net\_order\_db – (OrderService a baza danych)

## 2. Projekt

Aplikacja składa się z 6 kontenerów:

- WebUI

- APIGateway

- ProductService

- OrderService

- Database

- LoggingAgent

## 3. Analiza

LoggingAgent działa jako sidecar do OrderService i dzieli z nim sieć.

Każdy komponent ma dostęp tylko do sieci i serwisów wymaganych do ich działania.

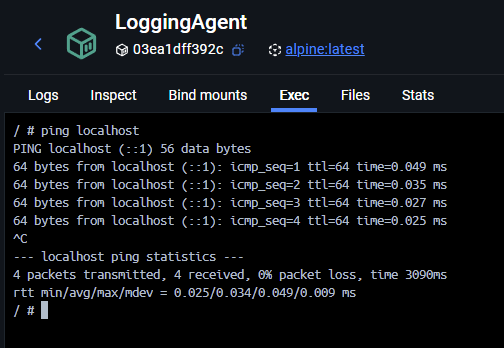
Takie podejście izoluje każdy serwis dzięki czemu nie mogą się komunikować między sobą bezpośrednio.

Prowadzi to również do ochrony bazy danych, dostęp do niej jest tylko z serwisów backendowych.

Prowadzi to również do klarownego podziału roli oraz kontroluje ruch między komponentami.

Na hoście widoczny jest jedynie port dla WebUI.

## 4. Logging Agent



To połączenie działa ponieważ przy tworzeniu została użyta komenda:  
--network container:OrderService  
Przez co oba kontenery współdzielą sieć więc w przypadku LoggingAgent, localhost odnosi się do OrderService.

Próba wywołania LoggingAgent z OrderService nie będzie działała ponieważ LoggingAgent nie ma własnego portu i tak jak wspomniałam localhost odnosi się do OrderService w przypadku tych kontenerów.