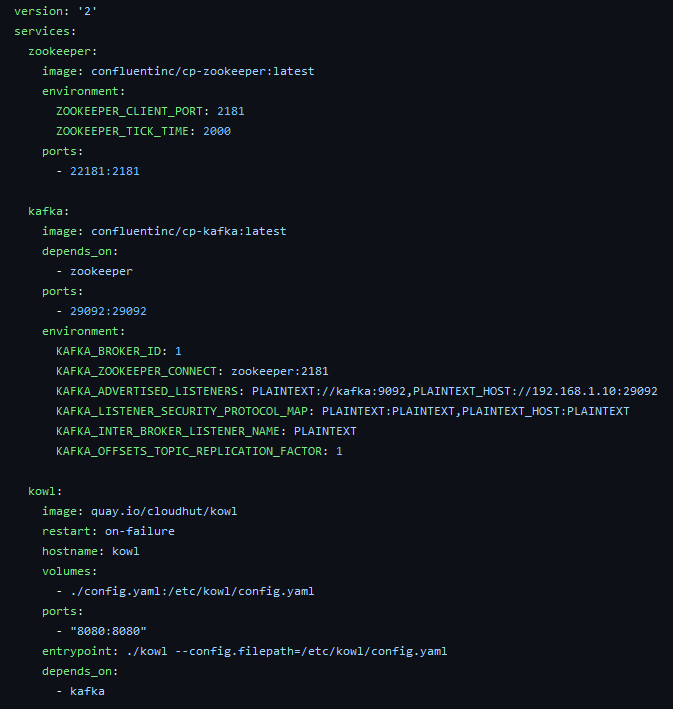
**Implementing Kafka between microservices using Golang and Docker**

Microservices architecture has become increasingly popular for building scalable and resilient applications. However, managing communication between microservices can be challenging. Kafka, a distributed streaming platform, offers an efficient solution for asynchronous communication between microservices. In this tutorial, we'll explore implementing Kafka for communication between two microservices written in Golang, encapsulated in Docker containers.

**Prerequisites:**

* Basic understanding of microservices architecture
* Familiarity with Golang programming language
* Docker installed on your system

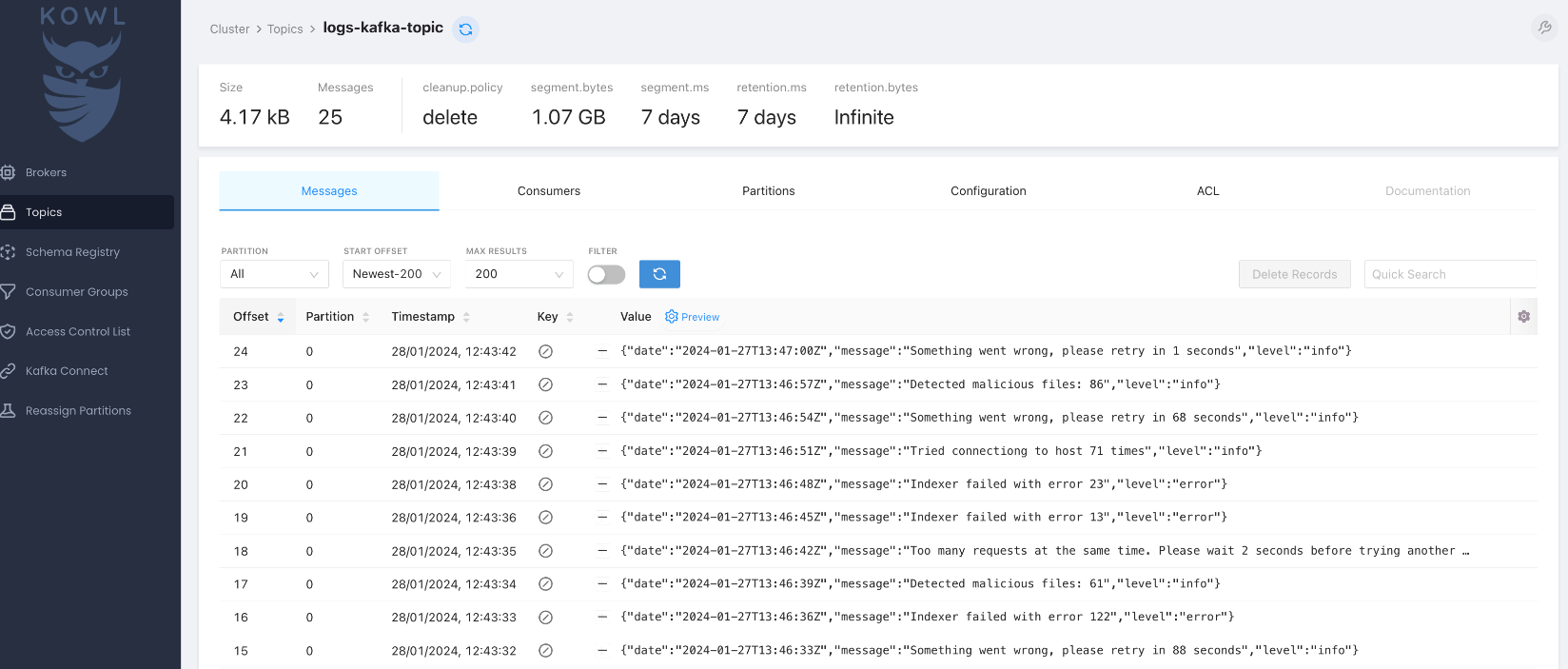
**Setting up Kafka:**



The kowl service is optional, but in provides a UI to visualize the the kafka topics.

*Note: the ip address should b e changed to reflect your local IP address.*

This is the kowl UI:

**

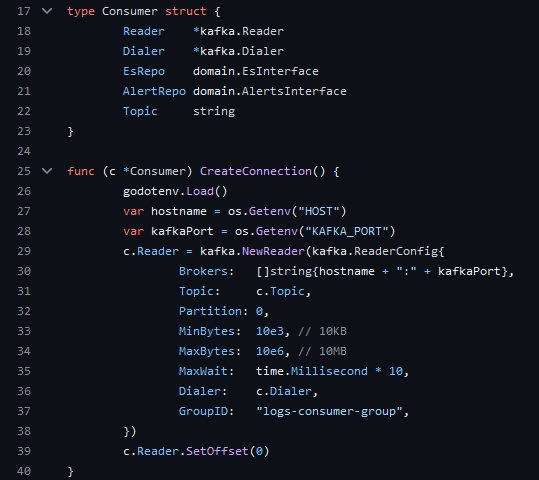
**Producer Microservice**

A computer screen shot of a program code

Description automatically generated

*Note: you should have a* ***.env*** *file where you set the Kafka topic name, Kafka port and the host ip address.*

**Consumer Microservice**

**** **A computer screen shot of text

Description automatically generated**

A screen shot of a computer program

Description automatically generated

*Note: you should have a* ***.env*** *file where you set the Kafka topic name, Kafka port and the host ip address at least.*

The Dockerfiles are the same for both services:

A screenshot of a computer program

Description automatically generated

Both microservices can be started through a docker-compose like this

A computer screen shot of a program

Description automatically generated

*Note: log-collector is the producer, log-indexer is the consumer.*

All the code can be found on my GitHub:

<https://github.com/laviniatunas/logs-microservices/tree/main>