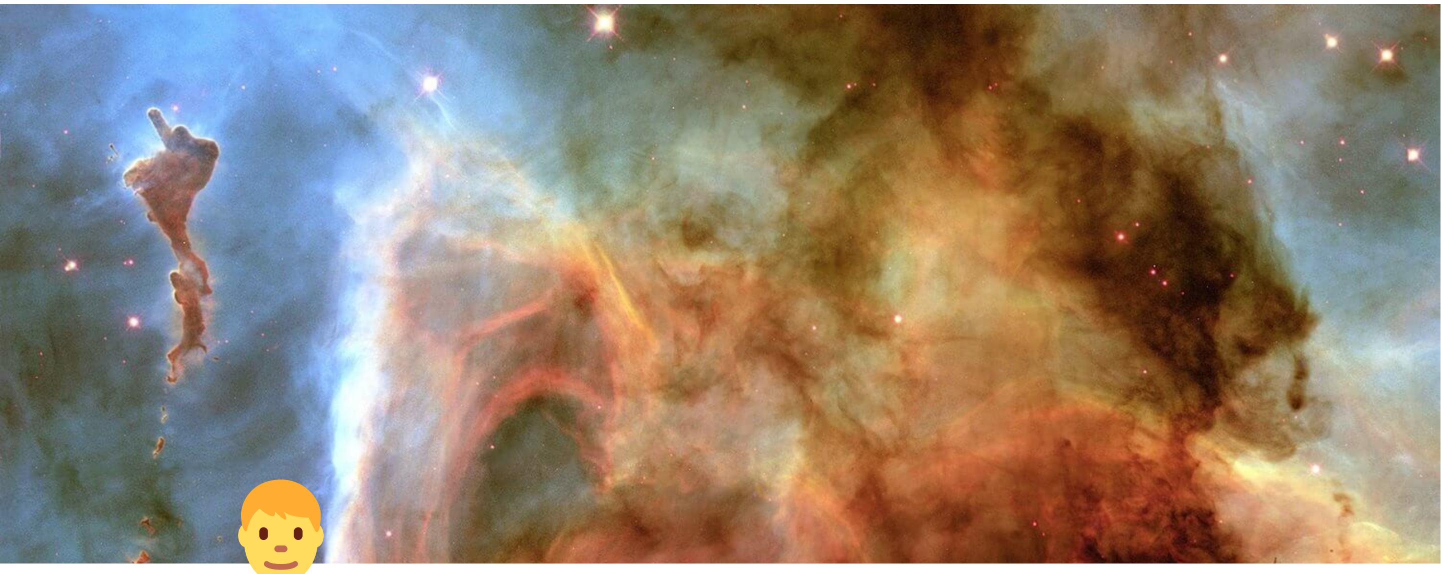

Backend Engineer Challenge Kafka

Last edited by	German Attanasio
Last edited time	July 18, 2024 1:30 PM

Real-Time Analytics Microservice Challenge

Introduction

Hello and welcome! In this challenge, you'll be building a real-time analytics microservice using Golang and Kafka. Your task is to create a service that processes user activity data, performs transformations, and stores aggregated results. This challenge will give you the opportunity to demonstrate your proficiency in Golang, microservices architecture, and Kafka.

All the information you need to successfully complete the challenge is provided here. If you encounter any doubts or questions, we encourage you to make assumptions, use your best judgment, and proceed.

After you've completed the challenge, please email your Github repository to `german@moveo.ai` . Use the subject line "Backend Engineer Challenge — Moveo.AI"

Challenge Details,

1. **Kafka Producer**:
   - Implement a Kafka producer that exposes a REST endpoint and publishes messages to the `incoming.user_activity` topic.
2. **Kafka Consumer**:
   - Implement a Kafka consumer that reads messages from the `incoming.user_activity` topic.
   - Perform data transformations, such as counting the number of users and the number of `page_view` activities.
   - Write aggregated results to a PostgreSQL database. Only one table is needed/recommended.
3. **Docker**:
   - Create a Docker image for both the producer and consumer.

Kafka Topic: `incoming.user_activity`

Example:

```
{ "user_id": "12345", "activity_type": "page_view", "timestamp": "2024-07-01T12:34:56Z", "metadata": { "page_url": "<https://example.com/home>", "referrer": "<https://google.com>" } }
```

Bonus Points

- Ensure the service can be scaled horizontally by adding more consumers. These additional consumers can be run in separate goroutines within the same service.
- Handle task retries and failures with a well-defined strategy.
- Verify that both of your microservices shutdown gracefully when receiving SIGTERM or SIGINT.

Deliverables

- A GitHub repository containing your code and any necessary documentation.
- Detailed instructions for deploying the service, including how to run the Docker containers.

Deadline

Your challenge must be completed and submitted within seven days. We believe this timeframe will provide you with ample opportunity to familiarize yourself with any new concepts, brainstorm, and implement your solution. We look forward to seeing your innovative approach!

> Tips: Remember, more isn't always better—focus on quality over quantity. Feel free to bend or even break the rules if you believe it leads to a better solution, but make sure you can justify your decision and explain your thought process.