Kafka System Monitor

Data Infrastructure Engineer

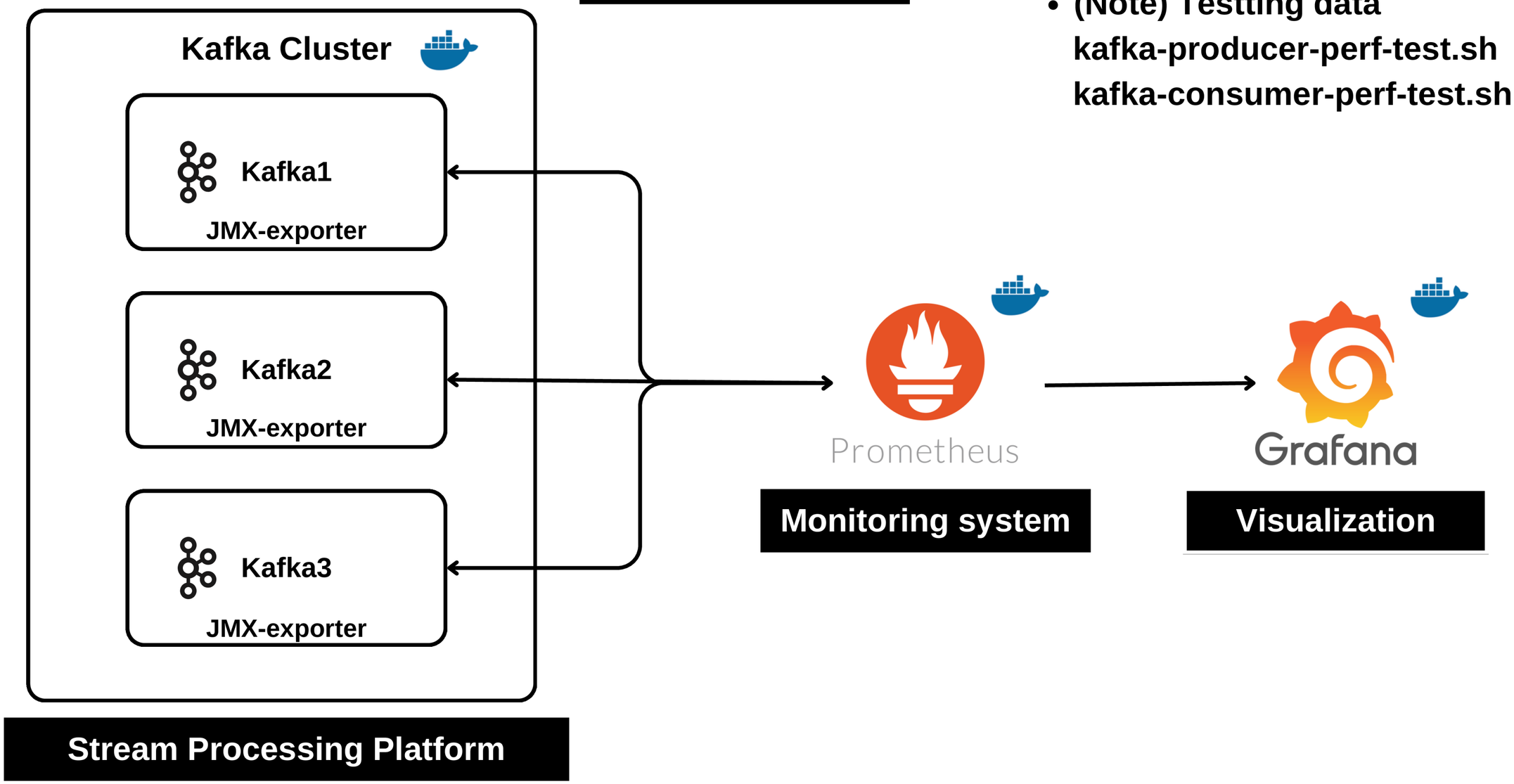
葉宥蓁 (Lina Yeh)

2025.03.03

CONTENTS

Introduction

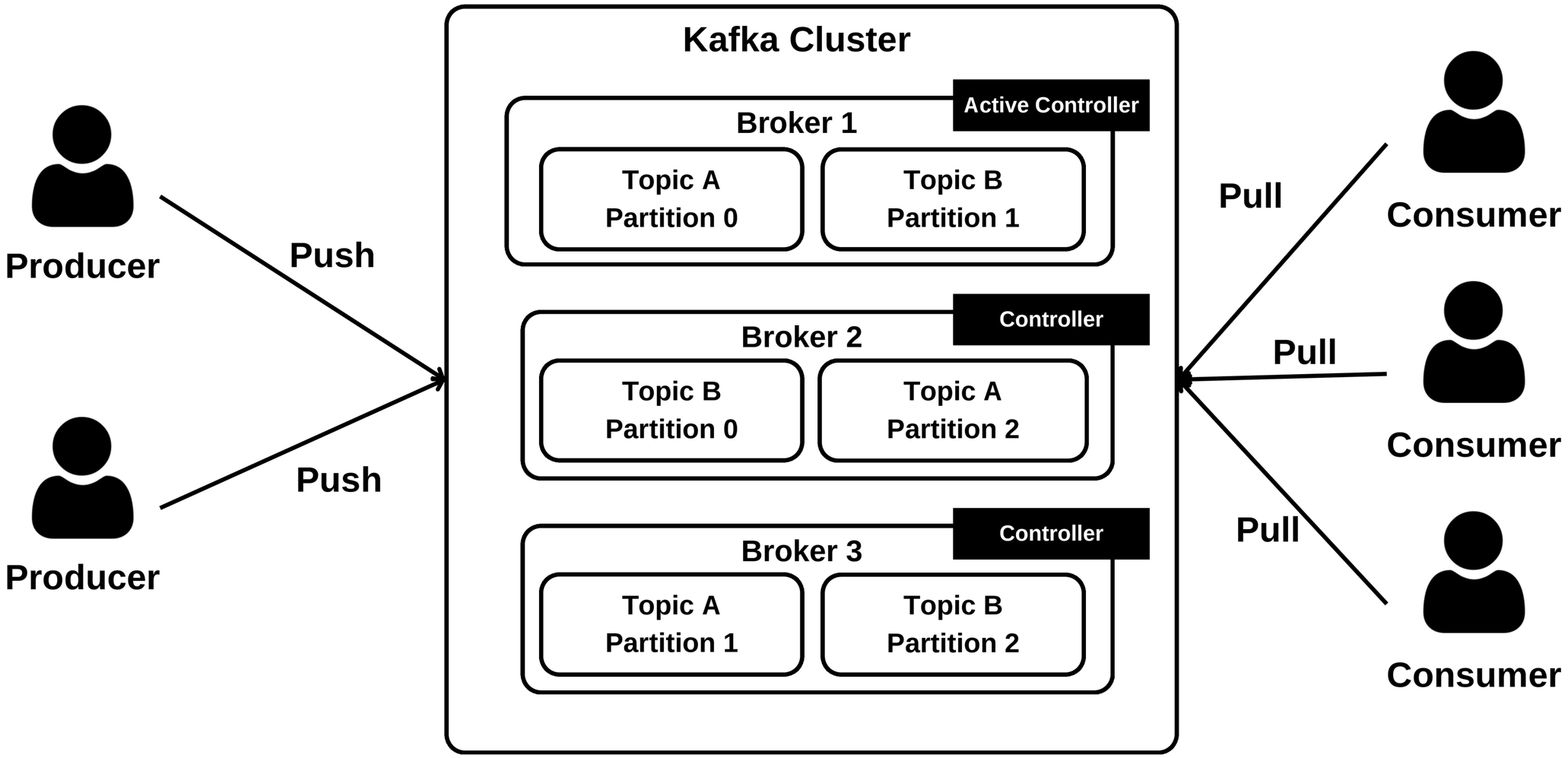
Architecture



- (Note) Testting data
  kafka-producer-perf-test.sh
  kafka-consumer-perf-test.sh

Introduction

Kafka



Introduction

Kafka

Feature

Real-time Data Processing

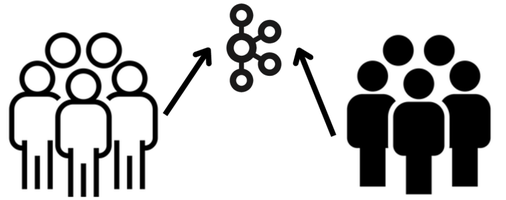
Scalable and Fault-tolerant

High Throughput

Use Case

Activity tracking

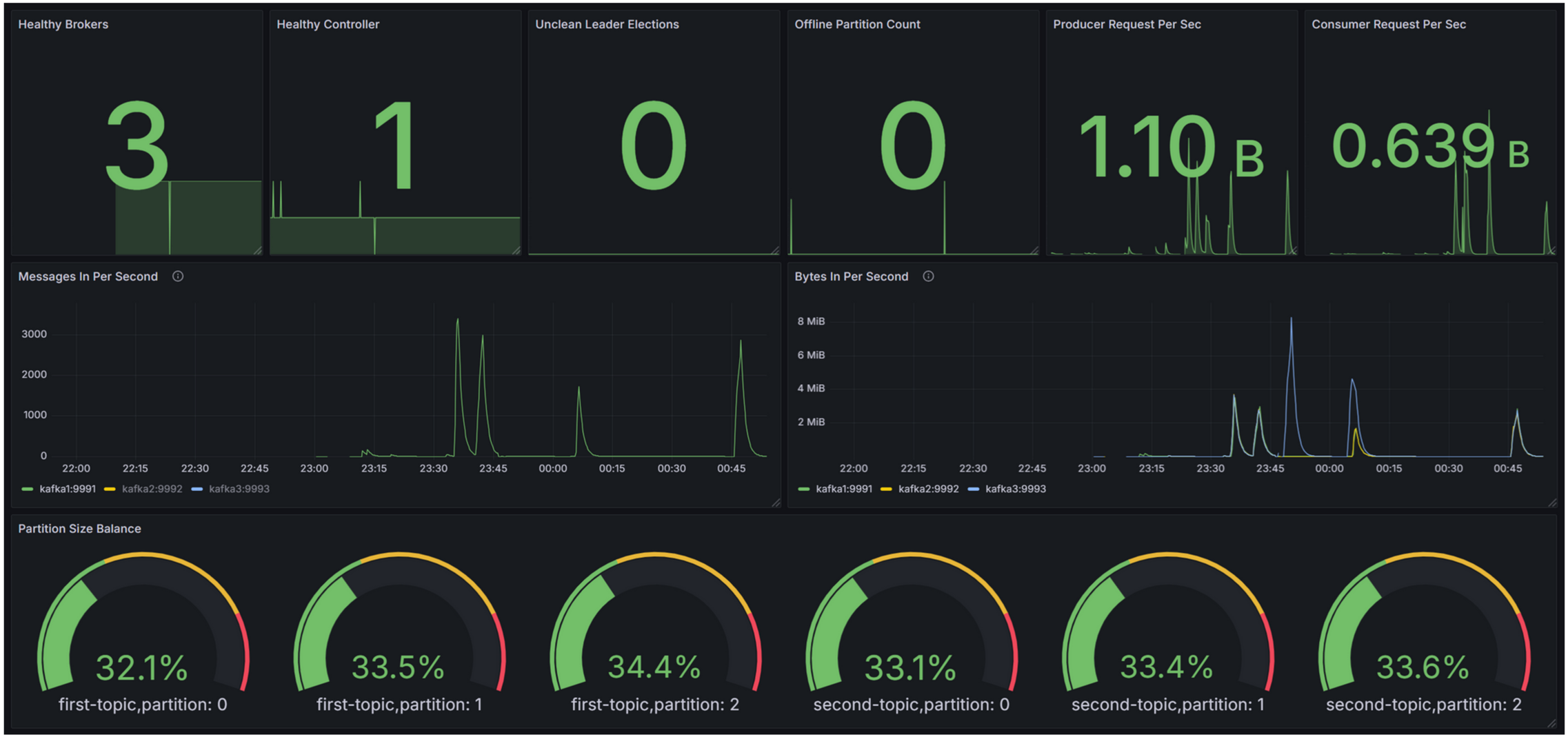
Message Delivery



Change Data Capture (CDC)

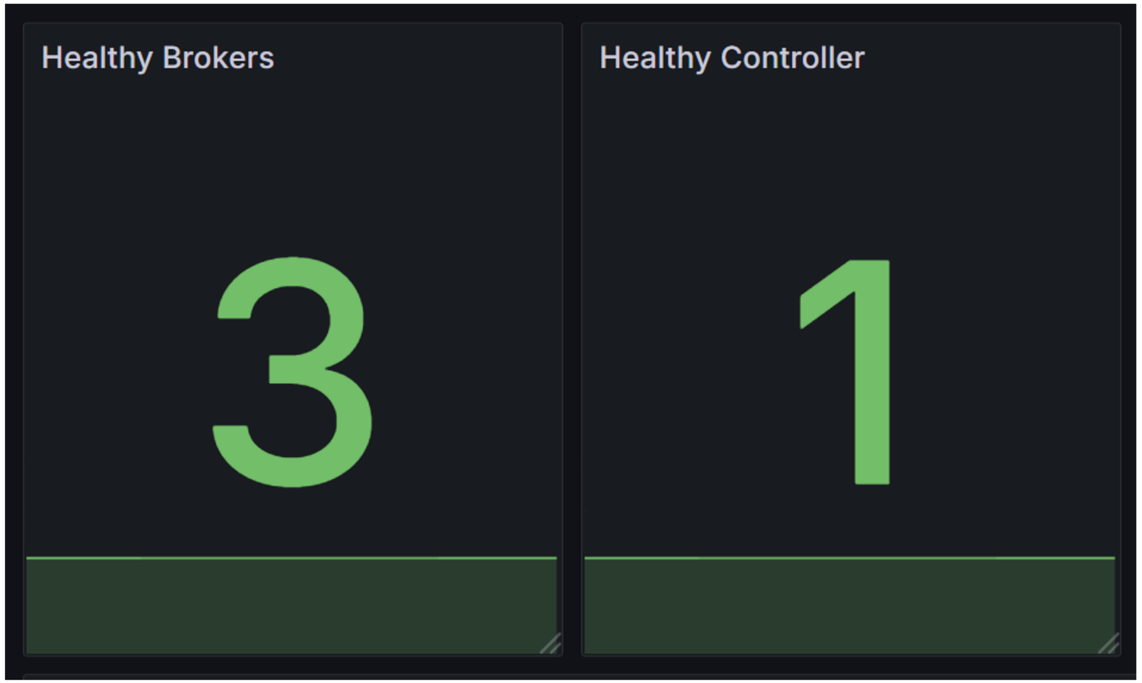
Monitor Metrics

Kafka



Monitor Metrics

Kafka



- **Healthy Brokers**

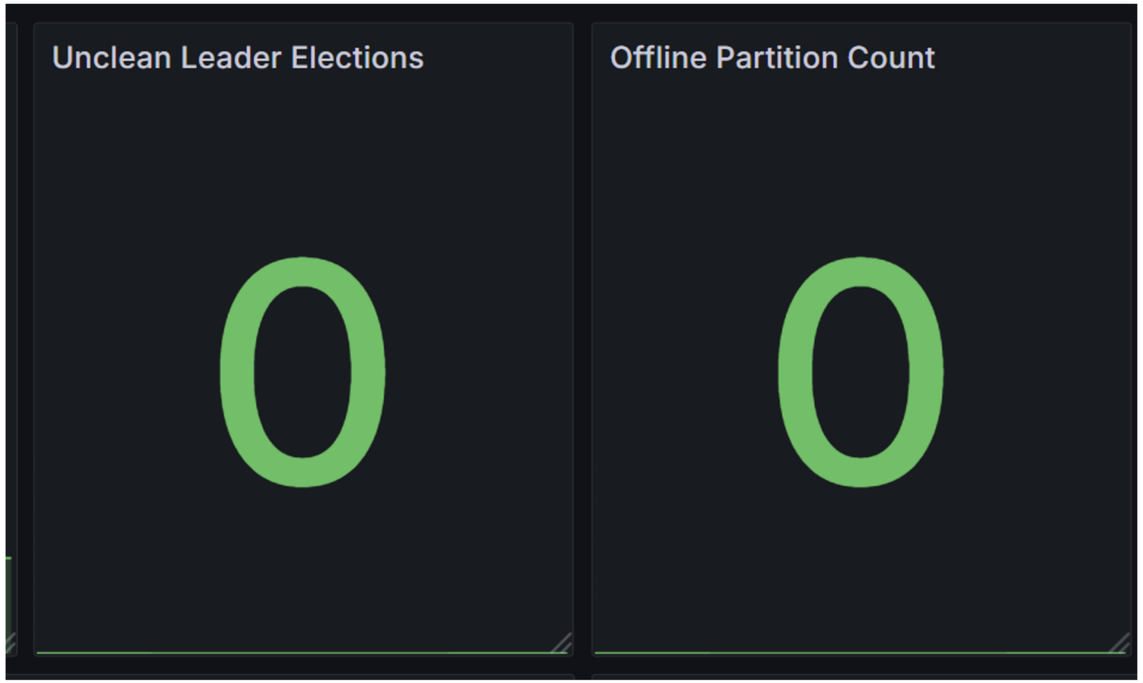
  To monitor health of online brokers and ensure that all brokers are functioning properly.

- **Healthy Controller**

  To monitor the controller that manages partition leadership and metadata updates in the Kafka cluster.

Monitor Metrics

Kafka



- **Unclean Leader Elections**

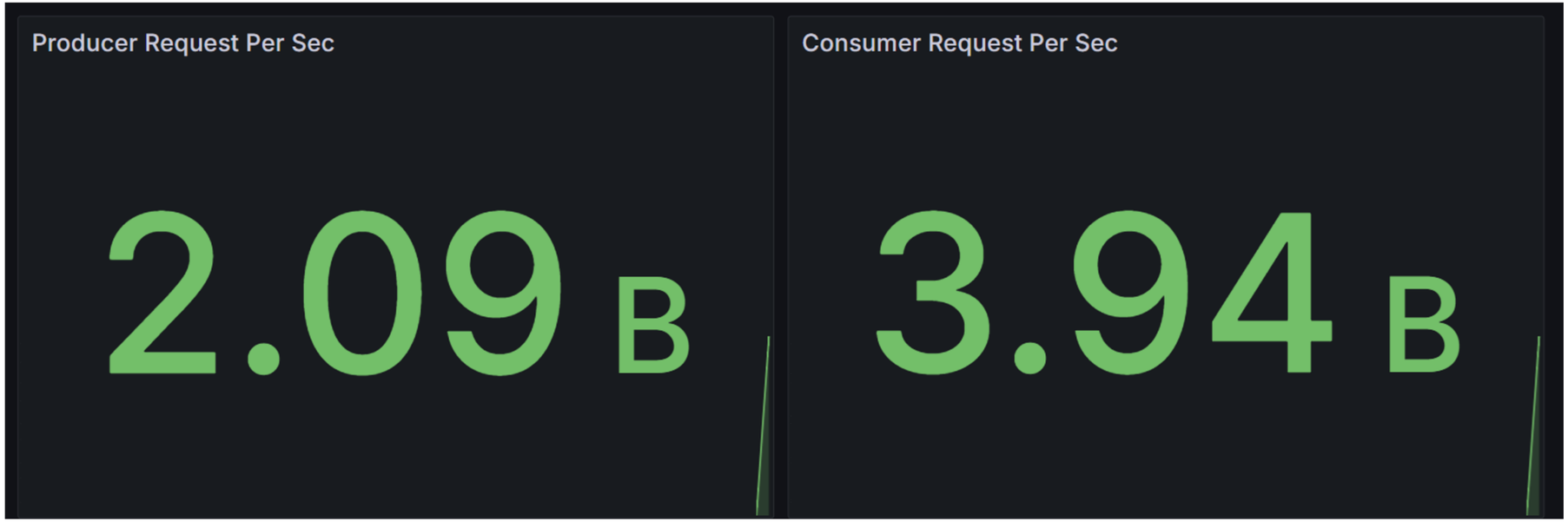
  To indicate partition leader clean. Unsynchronized replicas in the election will result in incomplete data. (Normally, 0 is expected)

- **Offline Partition Count**

  To monitor the health of partition. Offline partitions can lead to data loss. (Normally, 0 is expected)

Monitor Metrics

Kafka

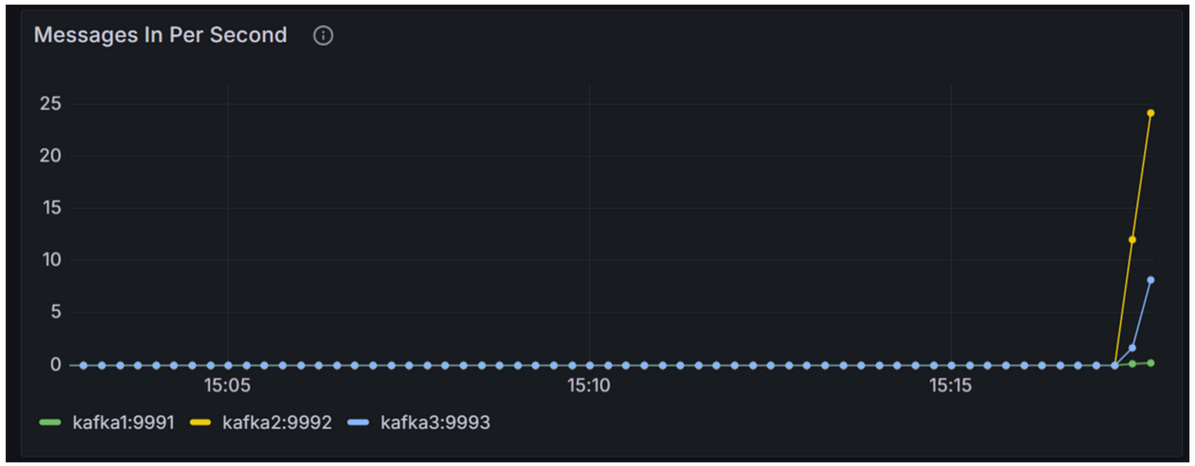


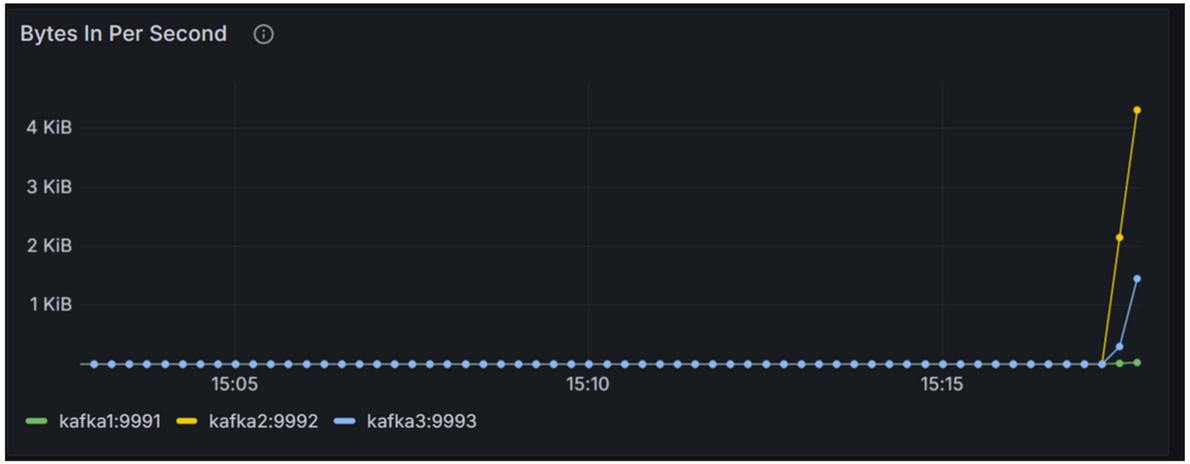
- **Producer Request Per Sec and Consumer Request Per Sec**

  To trace if data production and data consumption are balanced.

Monitor Metrics

Kafka



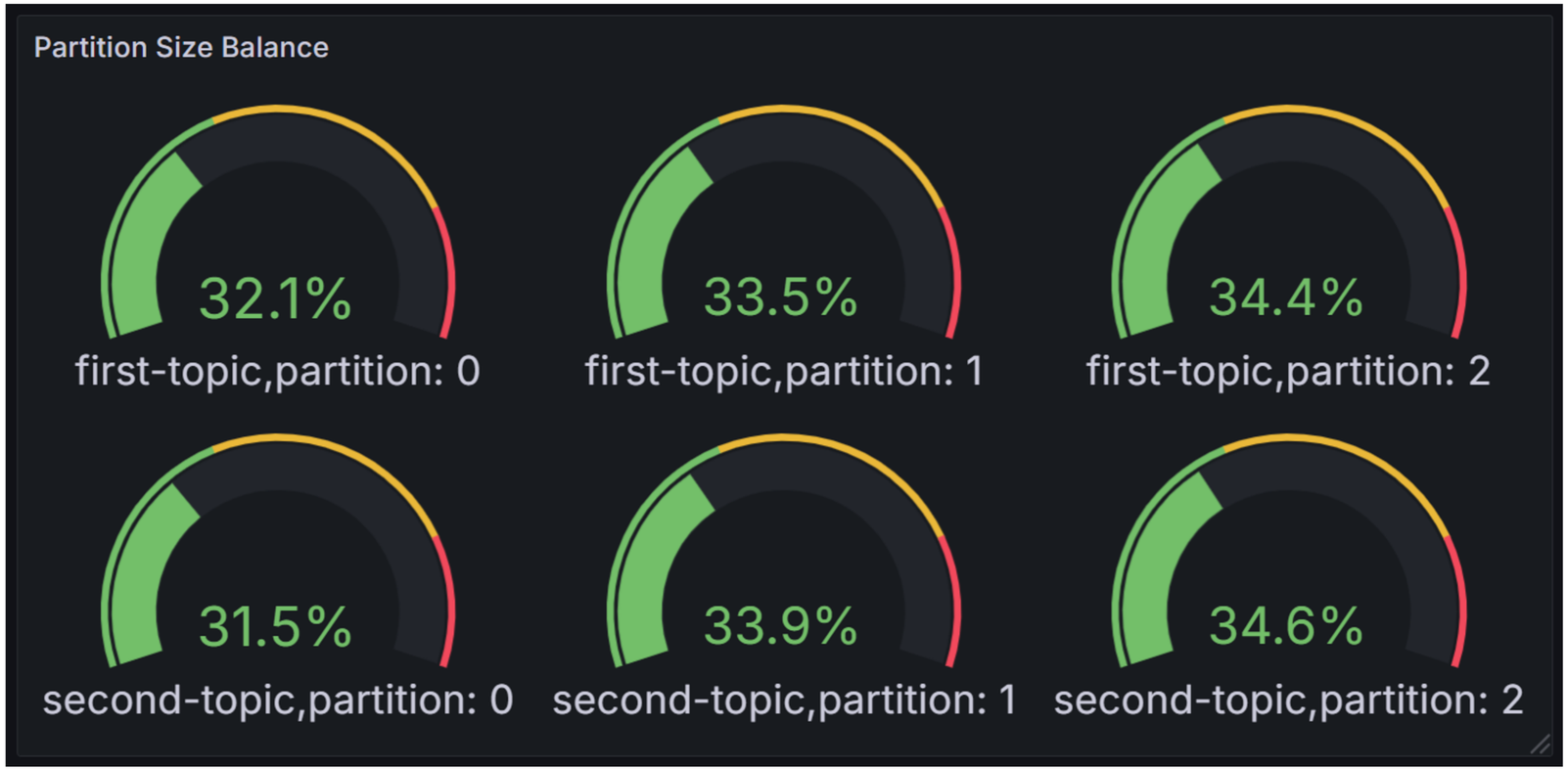


- **Message In Per Second and Bytes In Per Second**

To monitor the loading balance and throughput across brokers.
For different circumstances to analyze loading, not only the amount of messages per second matters, the number of bytes in is also an important factor to be reckoned with.

Monitor Metrics

Kafka

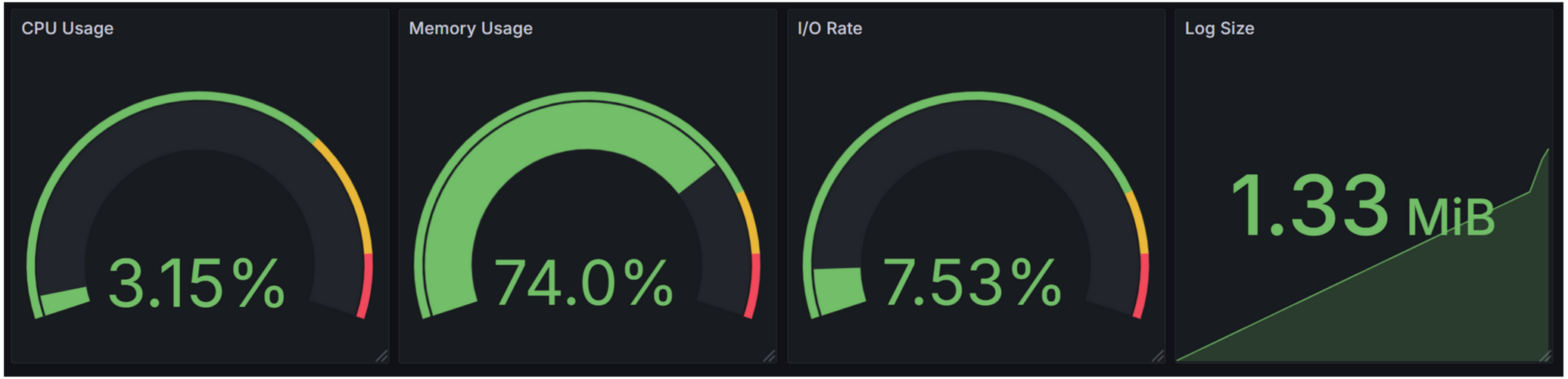


- **Partition Size Balance**

  To check if partitions in various topic are balanced.

Monitor Metrics

System



- **CPU, Memory, I/O and Log size**

To monitor the health of critical hardware status such as CPU usage, memory usage, I/O rate and log size to avoid service interruptions or even data loss from any hardware issues.

Conclusions

- Setup a playground environment with Kafka, Prometheus and Grafana incorporated to monitor to experiment loading balance and monitor data integrity in practical scenario.
- Learned Kafka's distributed concepts, successfully applied them in the system.
- It’s a challenge to find critical metrics (and without redundancy) for observing system performance, loading balance and even the system stability, which are all the key factors to ensure the best performance.

Future work

- Take network performance into account for the monitor, as network delay and bandwidth also play vital role in real world.
- Detect severe consumer lag in real time and notify system maintainers to avoid data loss due to transmission timeout.
- Incorporate automatic backup system when partition or server is down.