# Go x Kafka 101

ODDS | Thaibev

# Day 4

# Objective

## Assumption

- You know some Go fundamentals

## Recommendation

- Write with me

# Coverage

- What Is Apache Kafka?
- What Are Events?
- Getting Started with Apache Kafka and Go

# What Is Apache Kafka?

# What Is Apache Kafka?

Apache Kafka is an event streaming platform used to collect, process, store, and integrate data at scale. It has numerous use cases including distributed logging, stream processing, data integration, and pub/sub messaging.

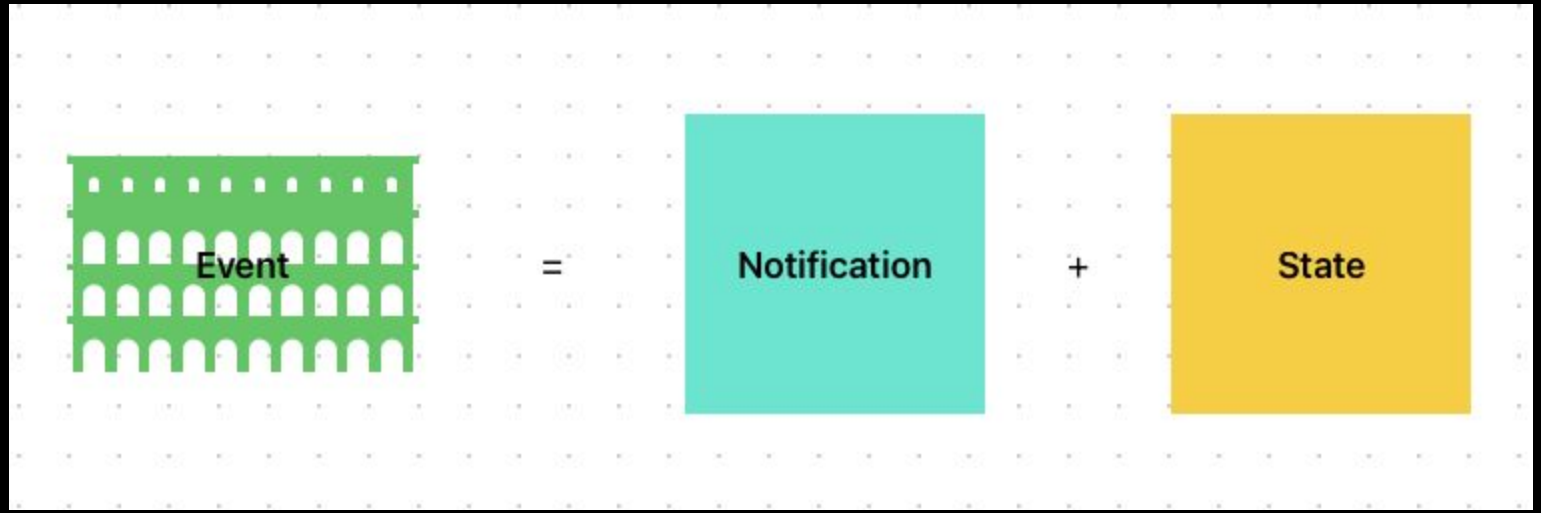
https://developer.confluent.io/learn-kafka/apache-kafka/events/

# What Are Events?

An event is any type of action, incident, or change that's identified or recorded by software or applications. For example, a payment, a website click, or a temperature reading, along with a description of what happened.

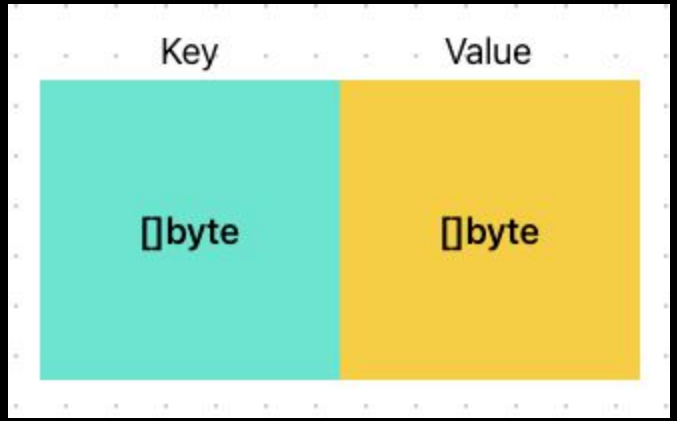
https://developer.confluent.io/learn-kafka/apache-kafka/events/

# What Are Events?

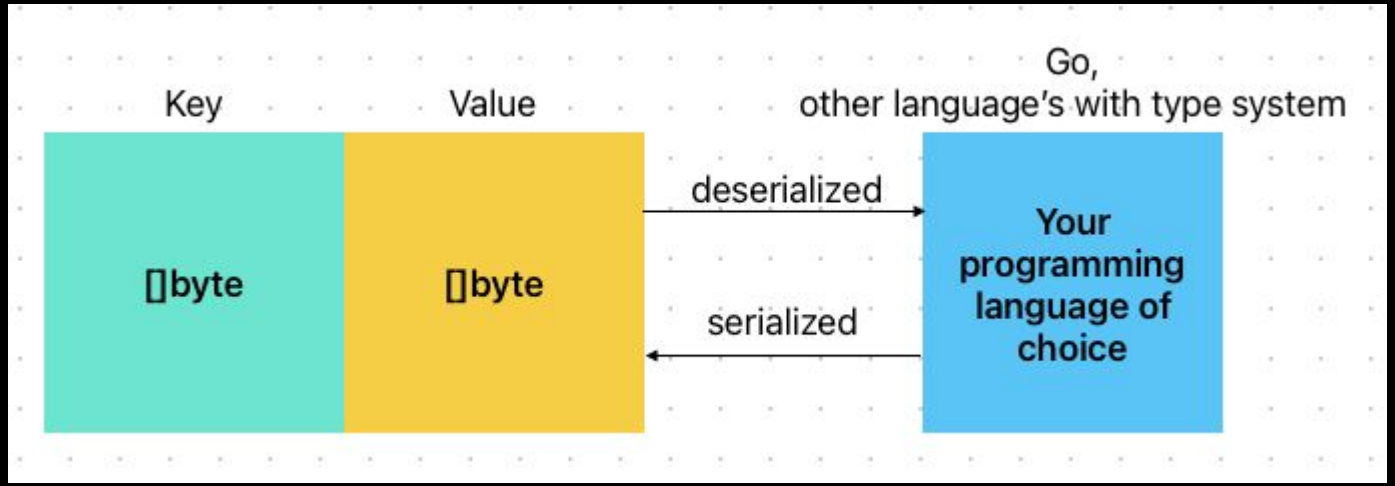



https://developer.confluent.io/learn-kafka/apache-kafka/events/

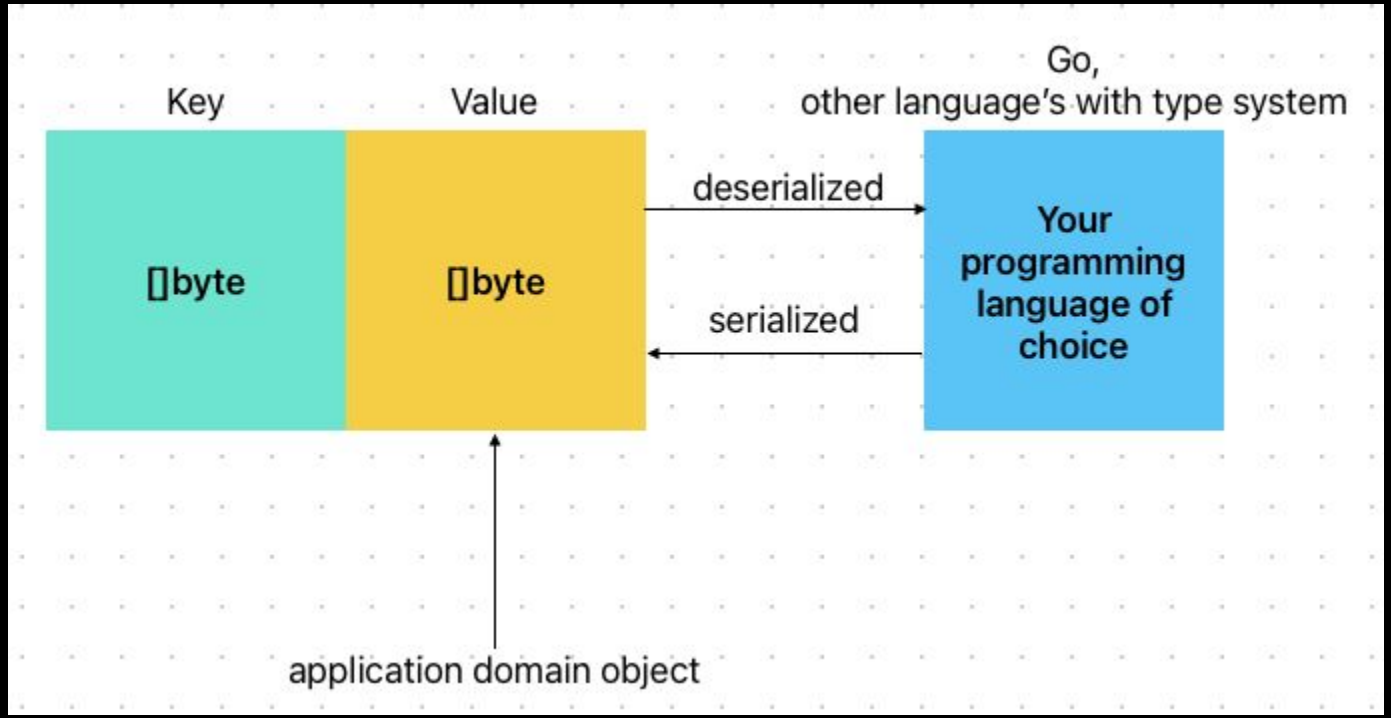
# Key/Value Pairs



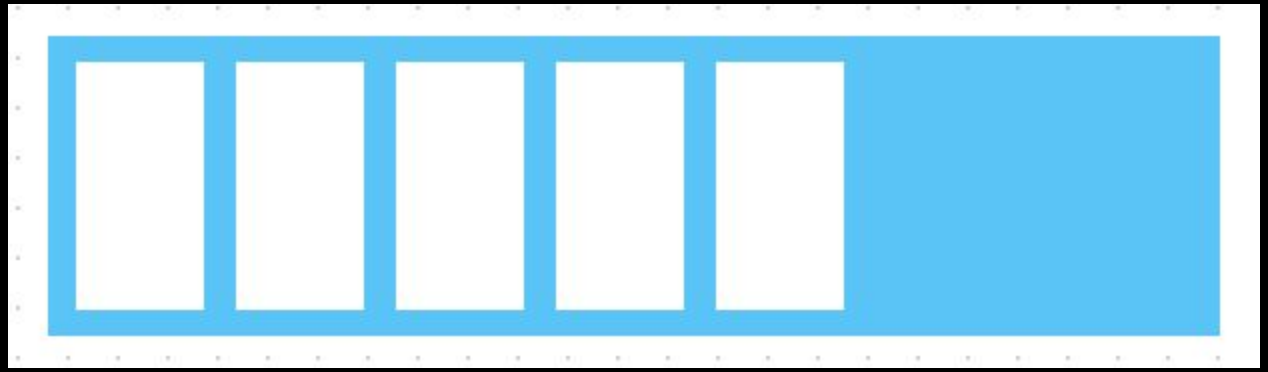
# Key/Value Pairs



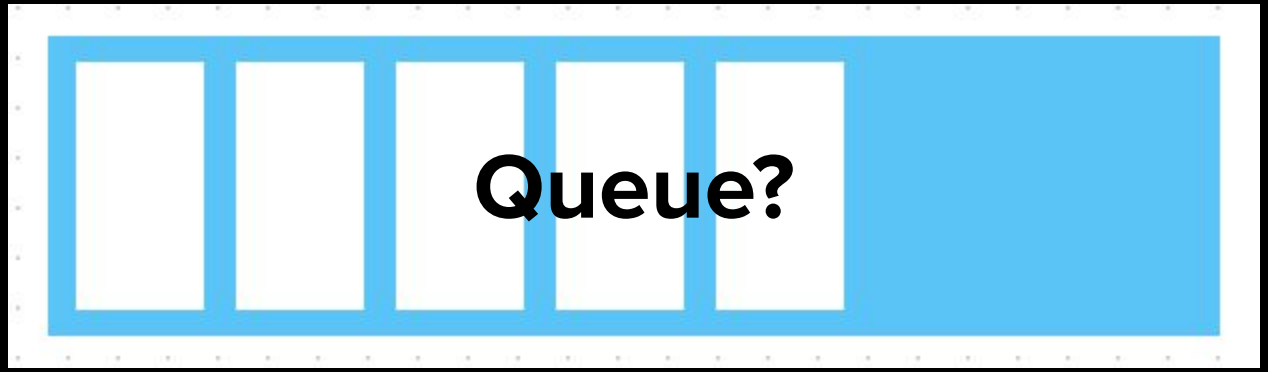
# Key/Value Pairs



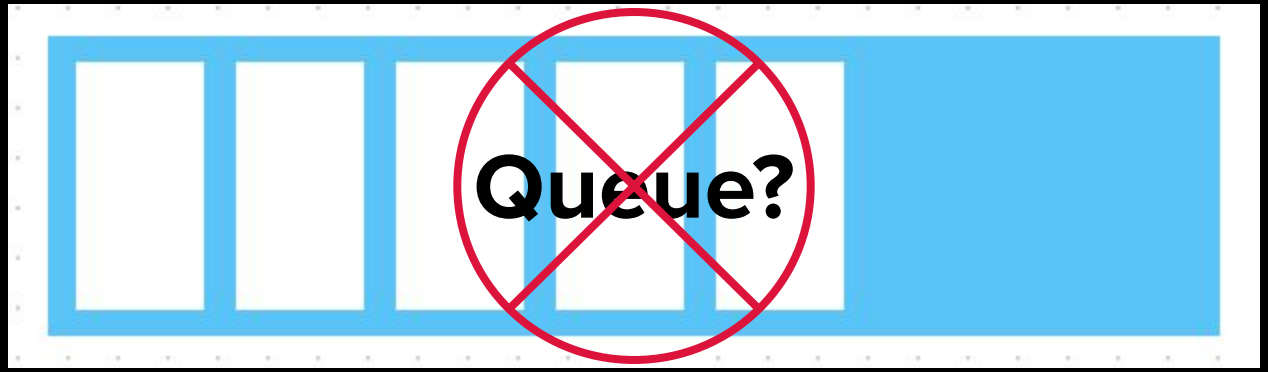
# Kafka Topics



# Kafka Topics



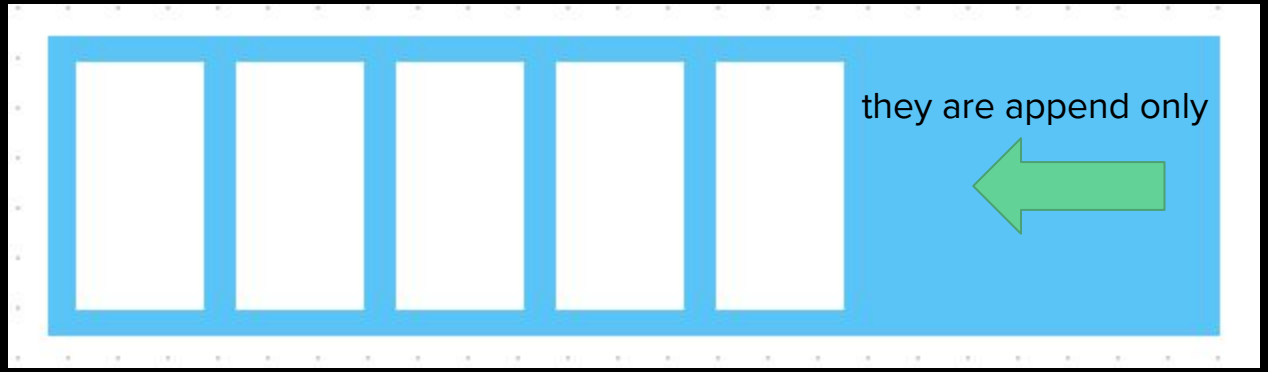
# Kafka Topics



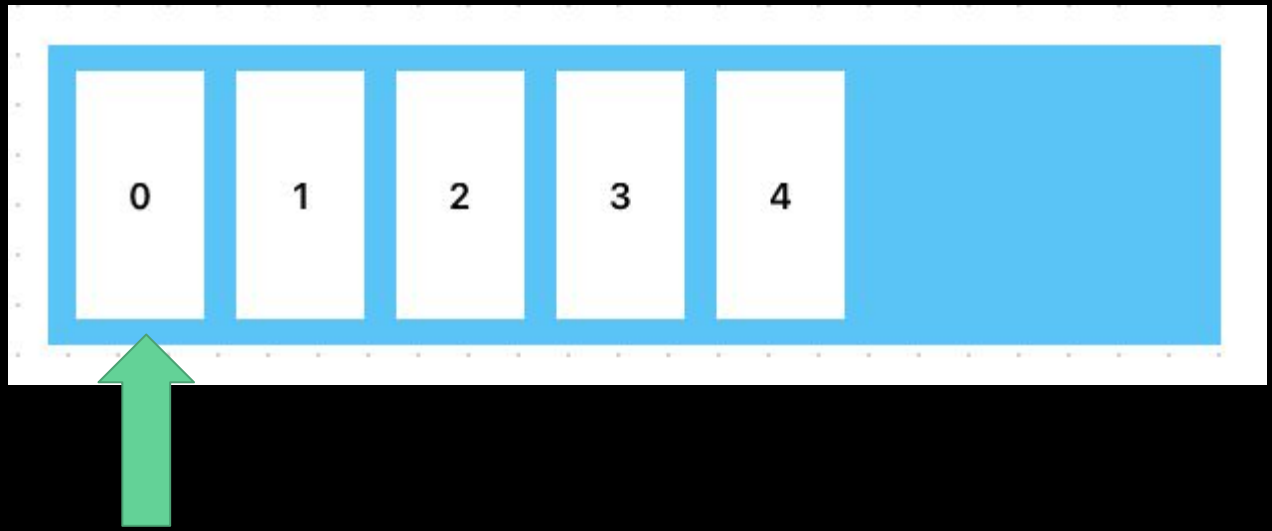
# Kafka Topics

Photo by Sebastian Pociecha on Unsplash

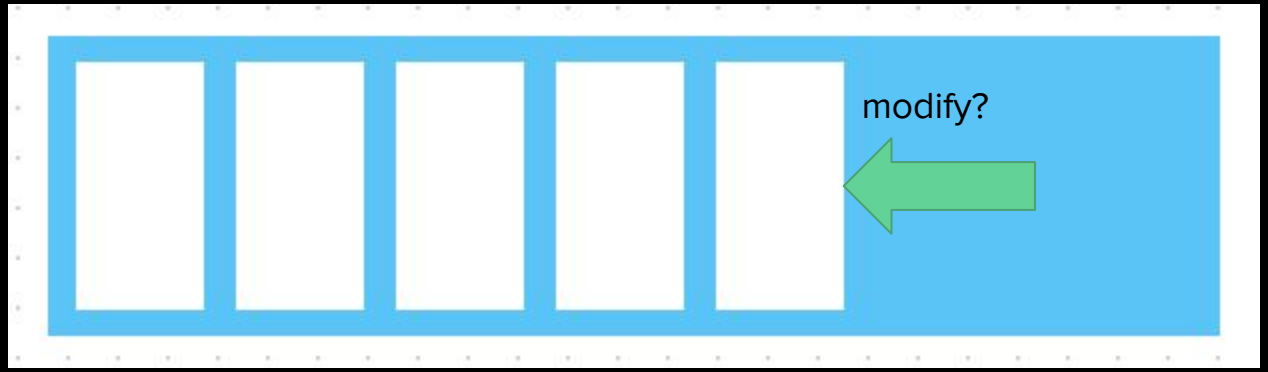
# Kafka Topics



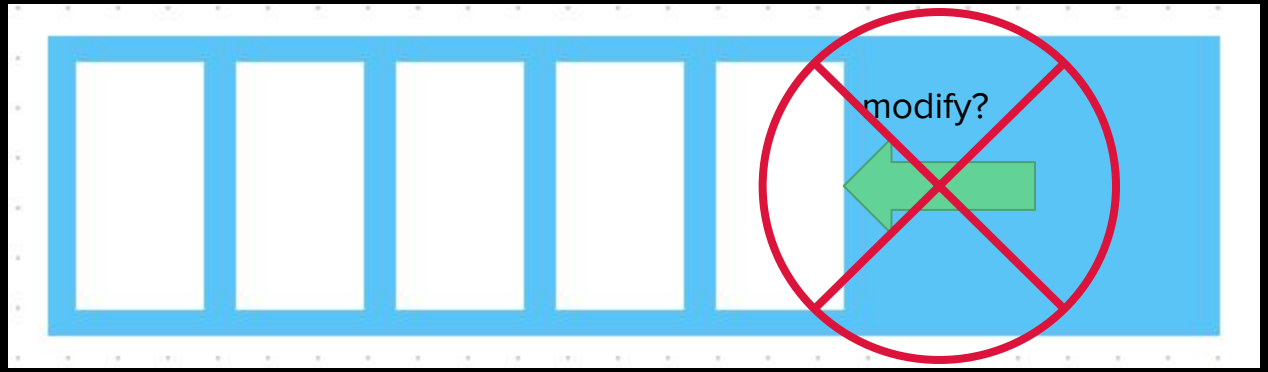
# Kafka Topics



# Kafka Topics



# Kafka Topics



once something has happened, it is exceedingly difficult to make it un-happen.

https://developer.confluent.io/learn-kafka/apache-kafka/events/

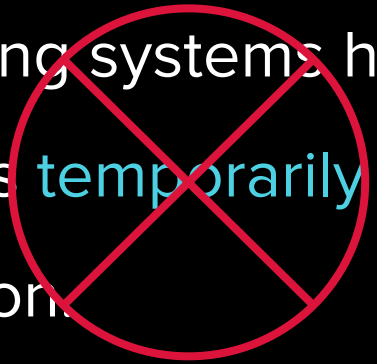
# Kafka Topics

Traditional enterprise messaging systems have topics and queues, which store messages temporarily to buffer them between source and destination.

https://developer.confluent.io/learn-kafka/apache-kafka/events/

# Kafka Topics

Traditional enterprise messaging systems have topics and queues, which store messages temporarily to buffer them between source and destination.



https://developer.confluent.io/learn-kafka/apache-kafka/events/

# Kafka Topics

- Kafka topics are logs
- Every topic can be configured to expire data after it has reached a certain age (or size)
- Kafka topics are files stored on disk

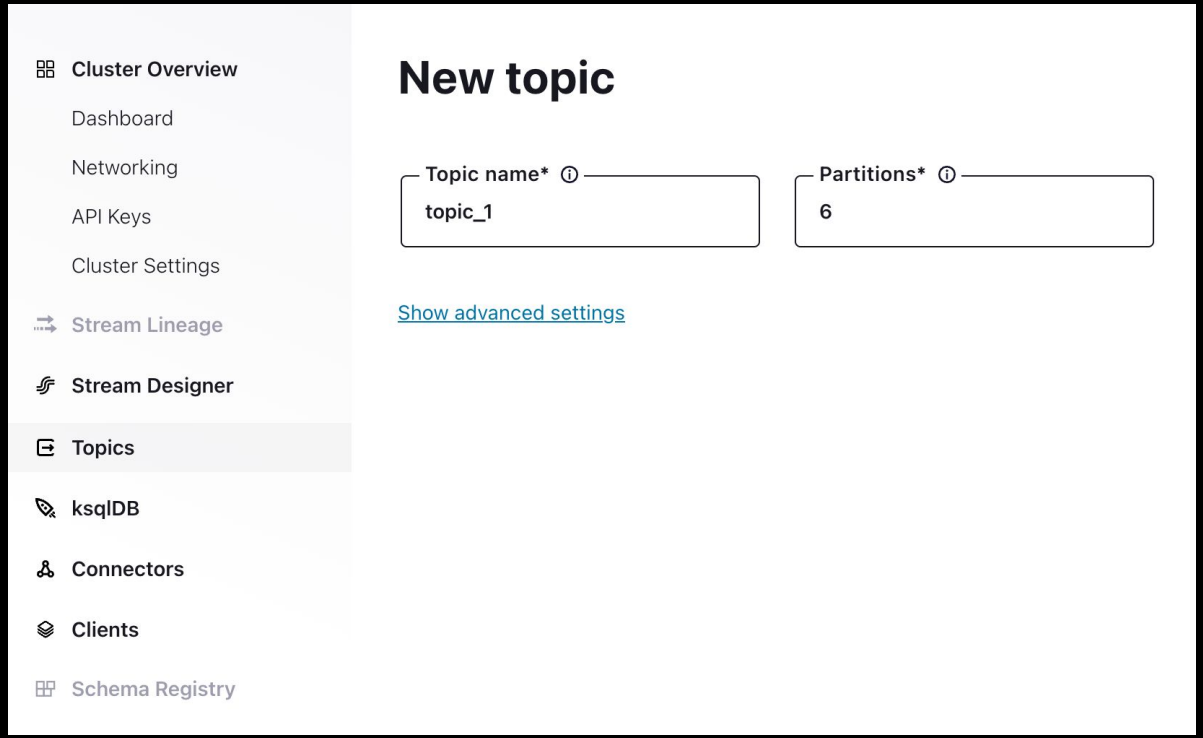
https://developer.confluent.io/learn-kafka/apache-kafka/events/

# Exercise

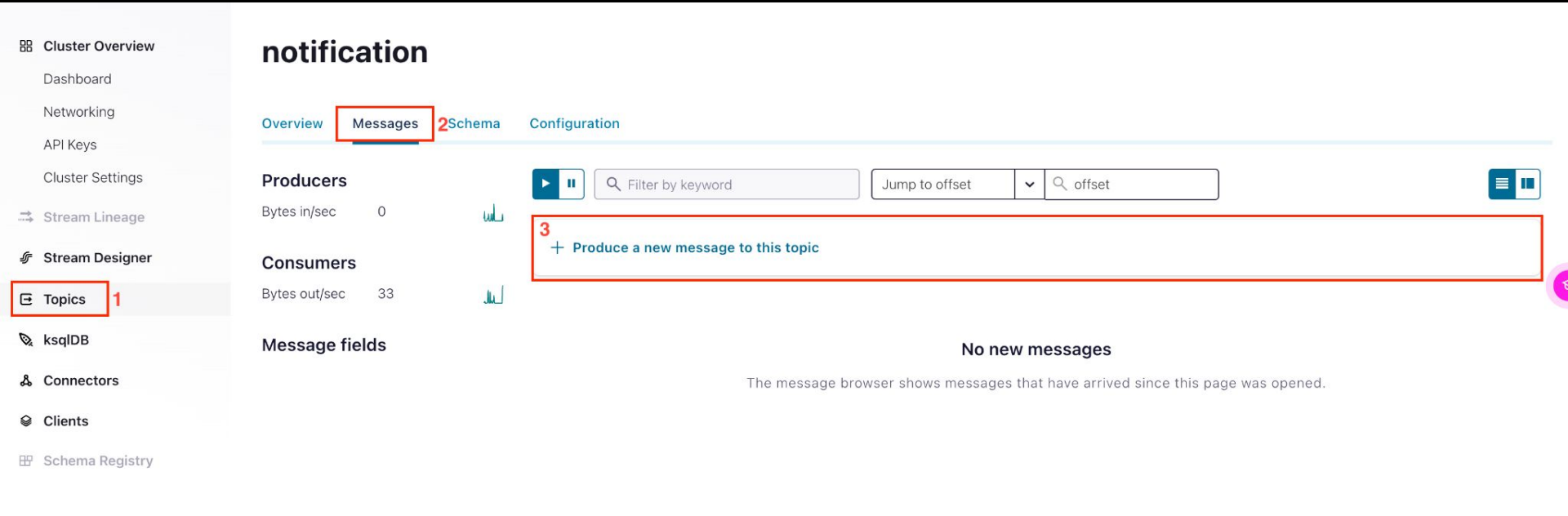
# Exercise

- https://www.confluent.io/confluent-cloud/tryfree/

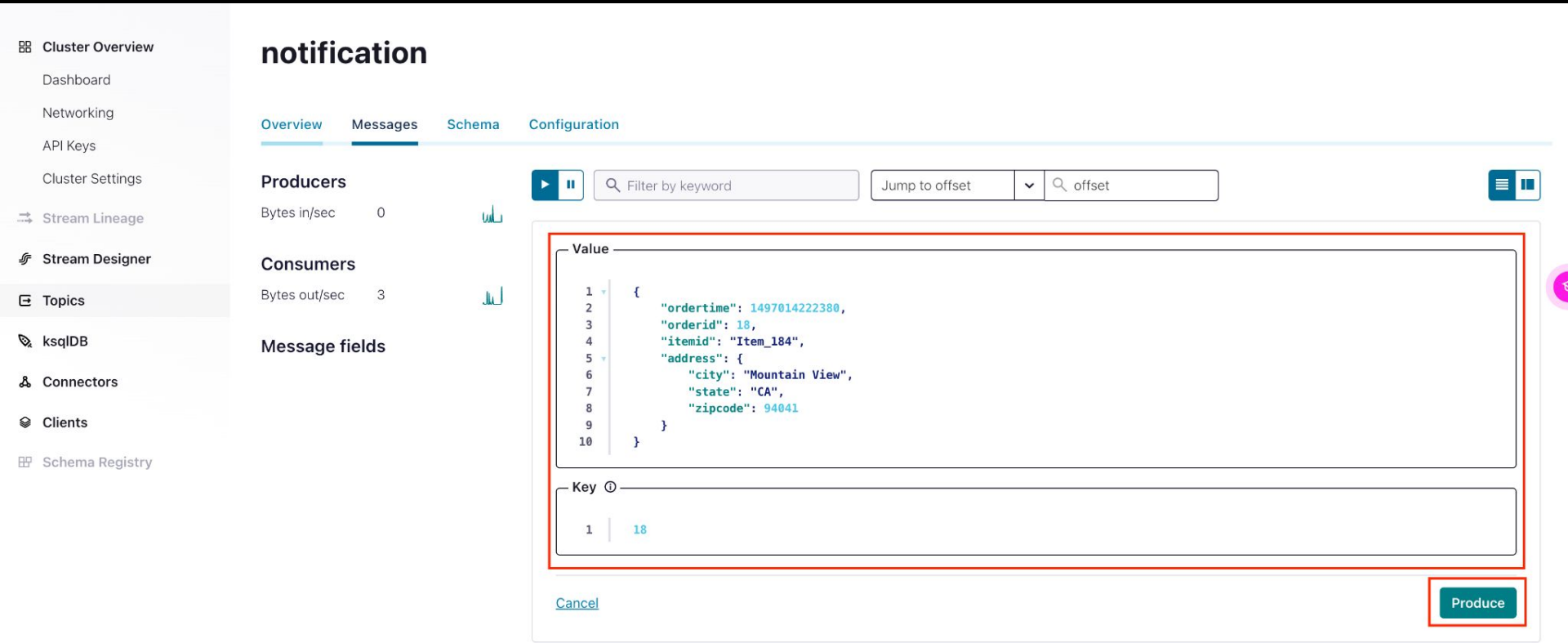
# Exercise


Cluster Overview
Dashboard
Networking
API Keys
Cluster Settings
Stream Lineage
Stream Designer
Topics
ksqlDB
Connectors
Clients
Schema Registry
New topic
Topic name*
topic_1
Partitions*
6
Show advanced settings

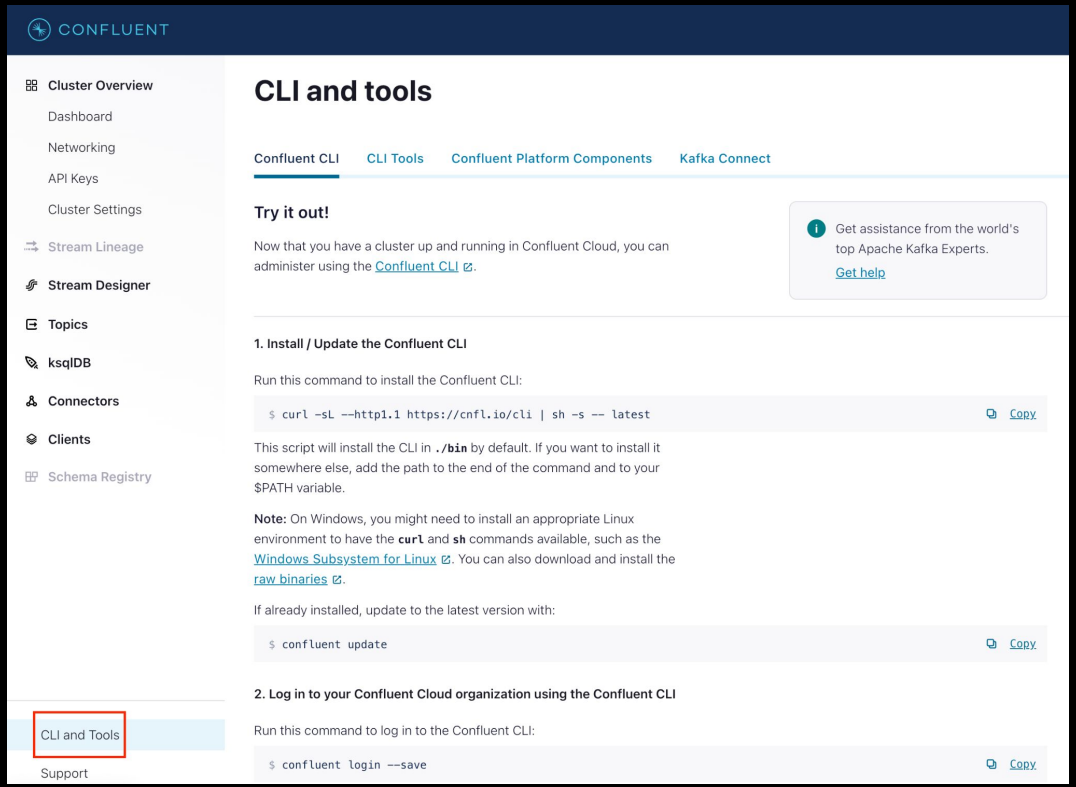
# Exercise


Cluster Overview
Dashboard
Networking
API Keys
Cluster Settings
Stream Lineage
Stream Designer
Topics
1
ksqlDB
Connectors
Clients
Schema Registry
notification
Overview
Messages
2Schema
Configuration
Producers
Bytes in/sec 0
Consumers
Bytes out/sec 33
Message fields
Filter by keyword
Jump to offset
offset
3
Produce a new message to this topic
No new messages
The message browser shows messages that have arrived since this page was opened.

# Exercise


Cluster Overview
Dashboard
Networking
API Keys
Cluster Settings
Stream Lineage
Stream Designer
Topics
ksqlDB
Connectors
Clients
Schema Registry
notification
Overview
Messages
Schema
Configuration
Producers
Bytes in/sec 0
Consumers
Bytes out/sec 3
Message fields
Filter by keyword
Jump to offset
offset
Value
{
"ordertime": 1497014222380,
"orderid": 18,
"itemid": "Item_184",
"address": {
"city": "Mountain View",
"state": "CA",
"zipcode": 94041
}
}
Key
18
Cancel
Produce

# Exercise


CONFLUENT
Cluster Overview
Dashboard
Networking
API Keys
Cluster Settings
Stream Lineage
Stream Designer
Topics
ksqlDB
Connectors
Clients
Schema Registry
CLI and Tools
Support
CLI and tools
Confluent CLI
CLI Tools
Confluent Platform Components
Kafka Connect
Try it out!
Now that you have a cluster up and running in Confluent Cloud, you can administer using the Confluent CLI.
Get assistance from the world's top Apache Kafka Experts.
Get help
1. Install / Update the Confluent CLI
Run this command to install the Confluent CLI:
$ curl -sL --http1.1 https://cnfl.io/cli | sh -s -- latest
Copy
This script will install the CLI in ./bin by default. If you want to install it somewhere else, add the path to the end of the command and to your $PATH variable.
Note: On Windows, you might need to install an appropriate Linux environment to have the curl and sh commands available, such as the Windows Subsystem for Linux. You can also download and install the raw binaries.
If already installed, update to the latest version with:
$ confluent update
Copy
2. Log in to your Confluent Cloud organization using the Confluent CLI
Run this command to log in to the Confluent CLI:
$ confluent login --save
Copy

# Exercise Set Up the Confluent CLI

curl -L --http1.1 https://cnfl.io/cli | sudo sh -s -- -b /usr/local/bin

confluent login --save

confluent environment list

confluent environment use {ID}

confluent kafka cluster list

confluent kafka cluster use {ID}

# Exercise

confluent api-key create --resource {ID}

confluent api-key use {API Key} --resource {ID}

# Exercise Produce and Consume Using the Confluent CLI

confluent kafka topic list

confluent kafka topic consume --from-beginning {topicname}

confluent kafka topic produce {topicname} --parse-key

5:"From the ashes a fire shall awaken"

6:"A light from the shadows shall spring"

7:"Renewed shall be blad that was broken"

8:"The crownless again shall be king"

# Exercise Produce and Consume Using the Confluent CLI

confluent kafka topic produce {topicname}

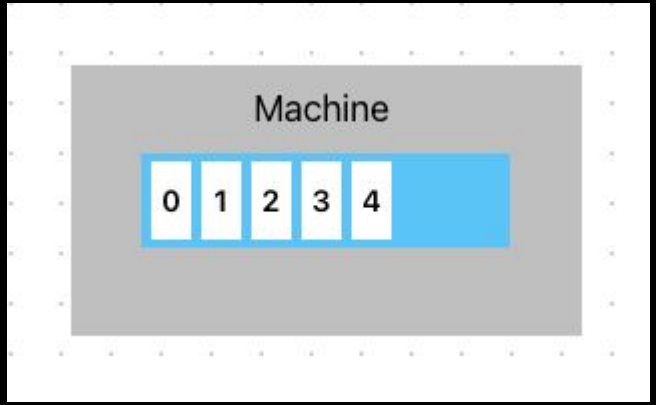
"test"

"test"

"foo"

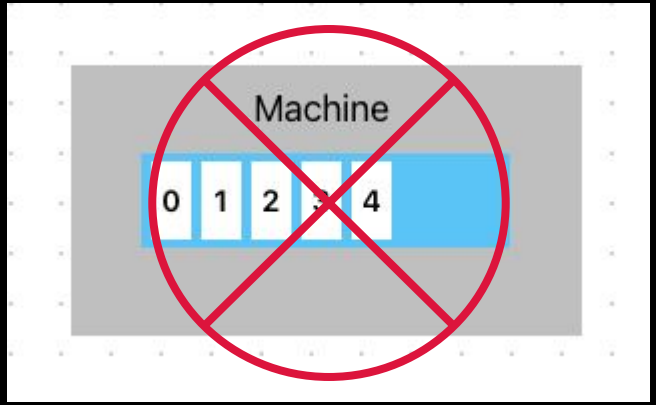
"bar"

# Kafka Partitioning

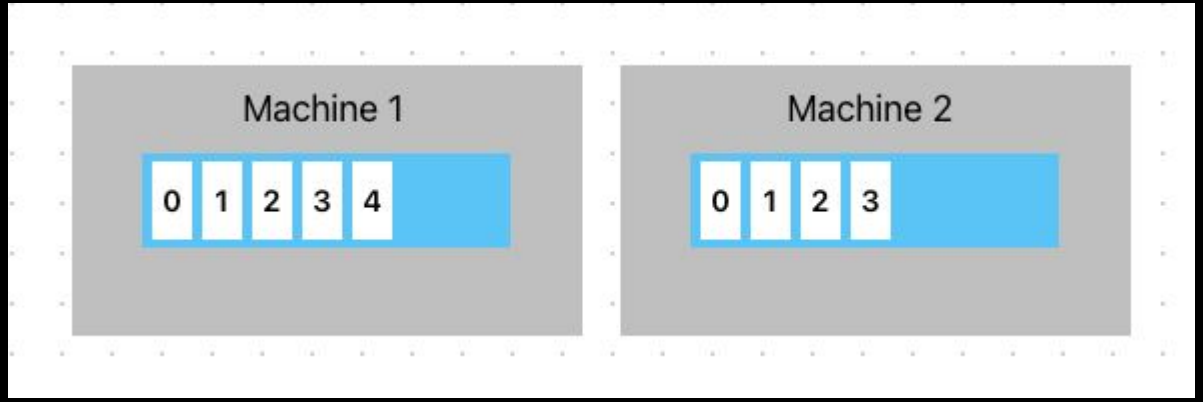



**LIMIT**

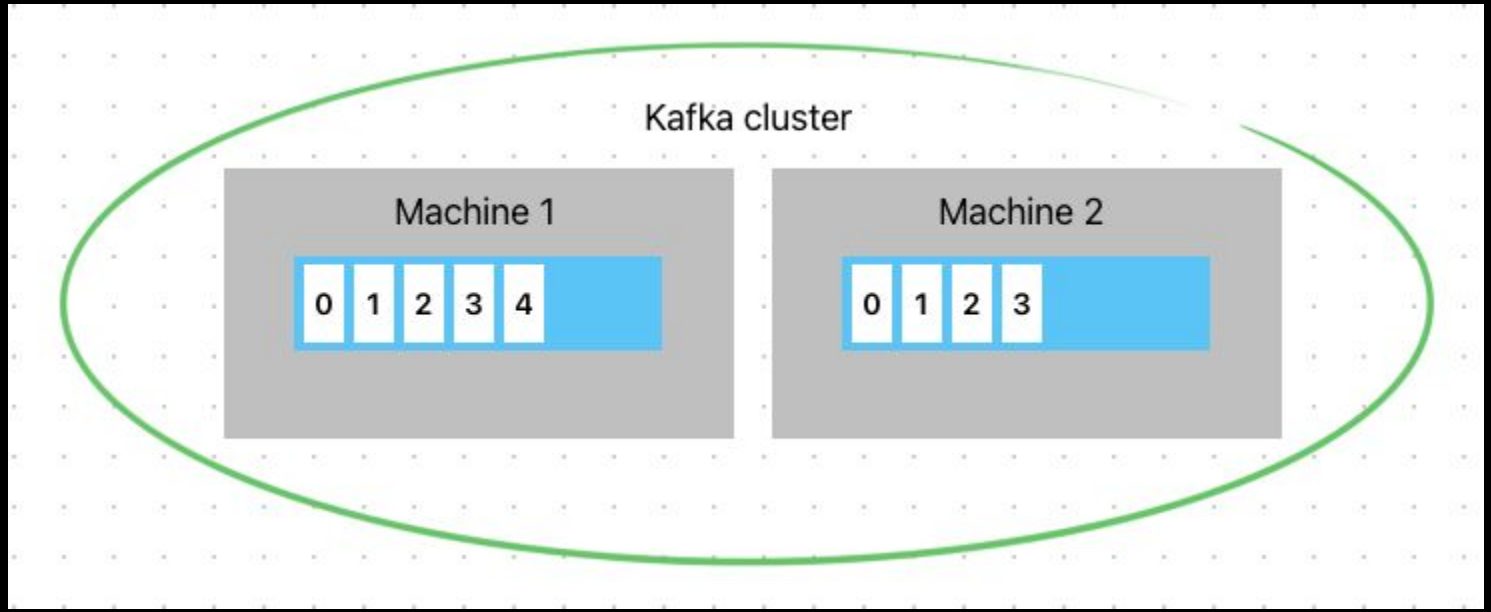
# Kafka Partitioning



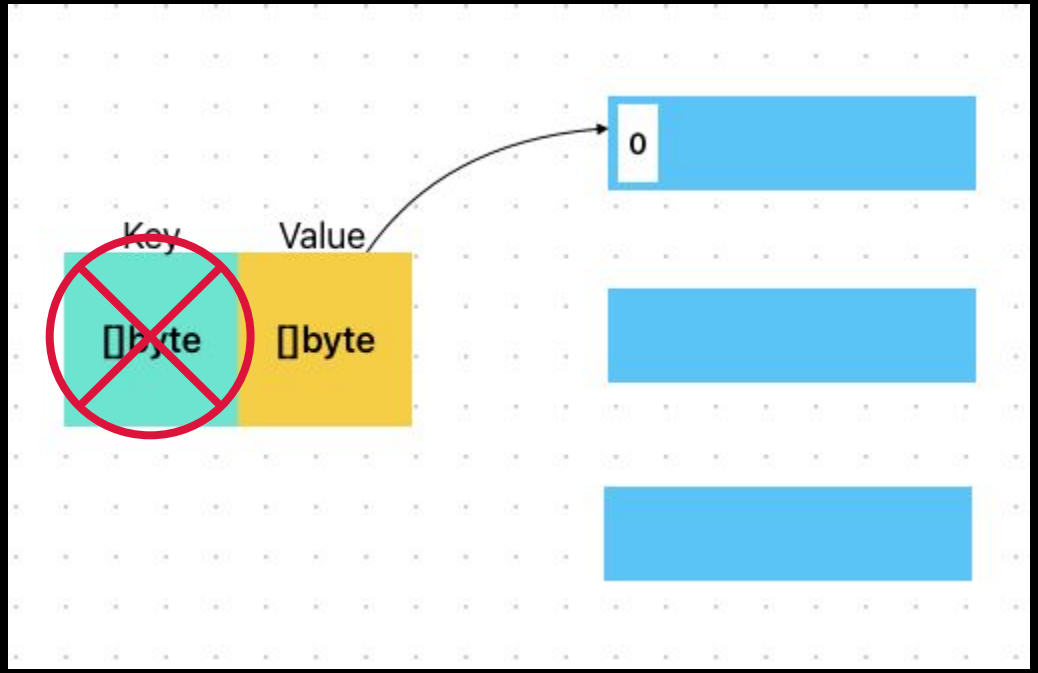
# Kafka Partitioning



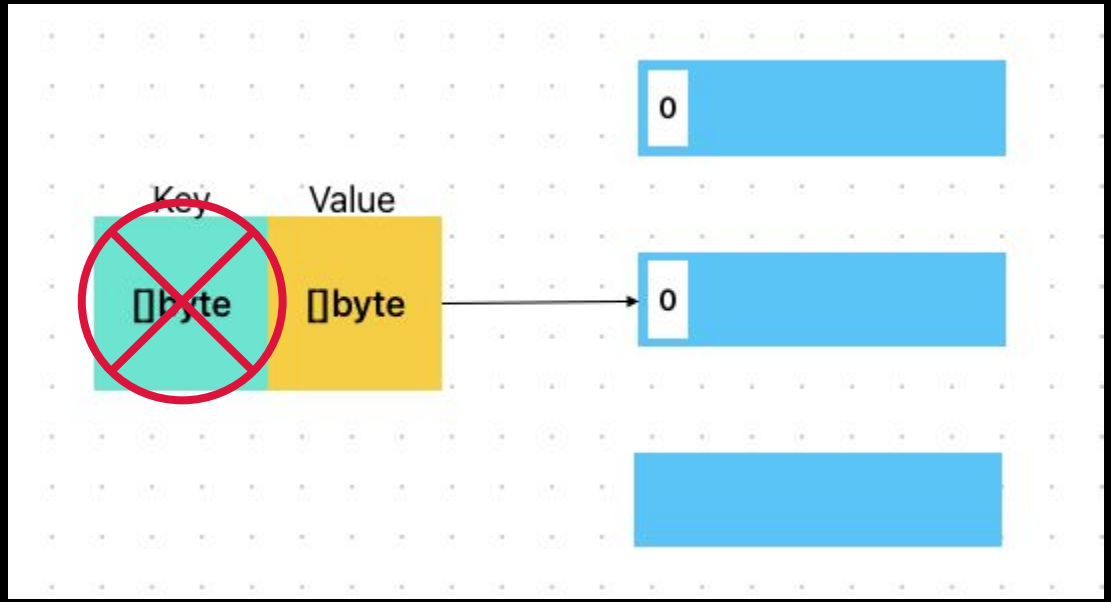
# Kafka Partitioning



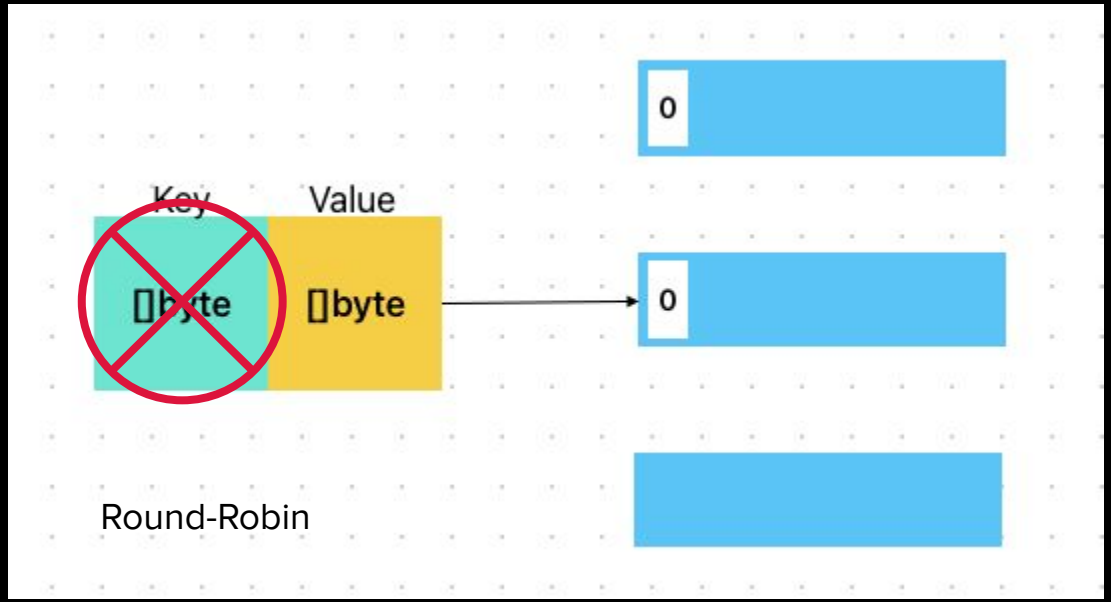
# Kafka Partitioning



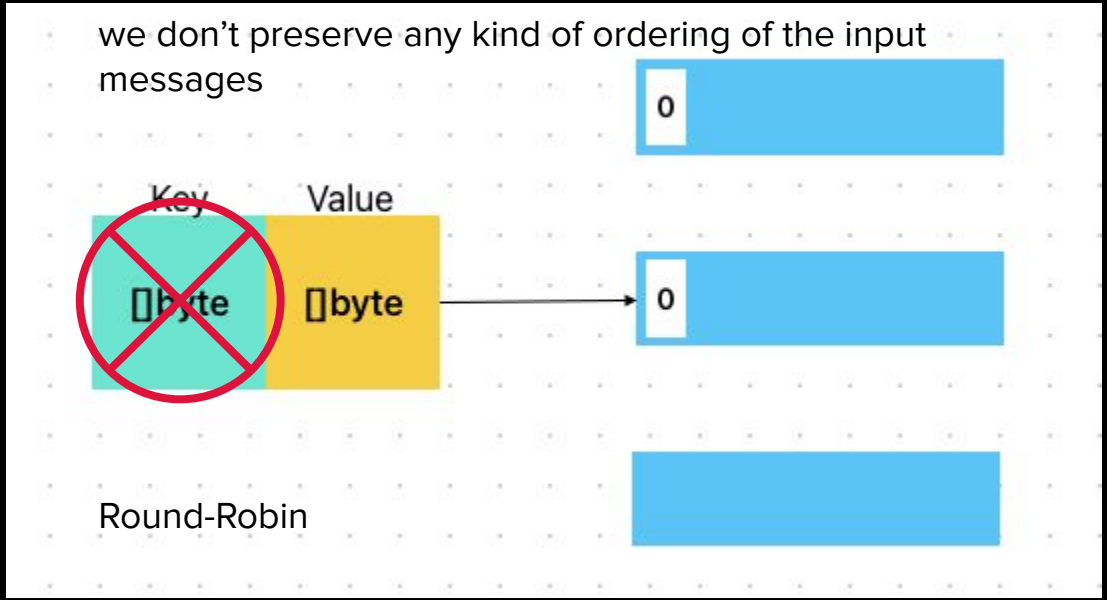
# Kafka Partitioning



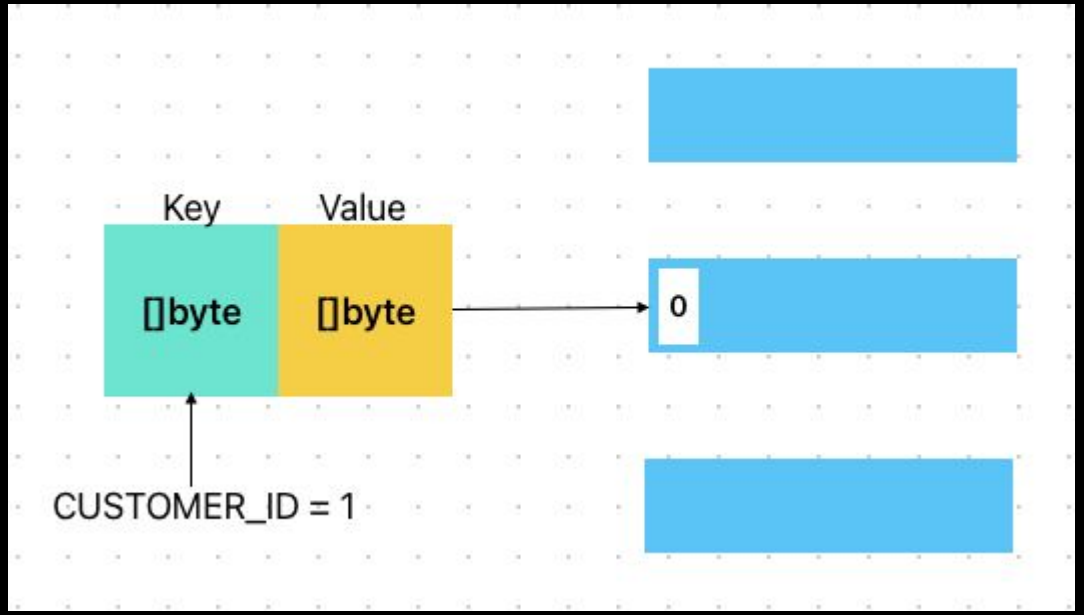
# Kafka Partitioning



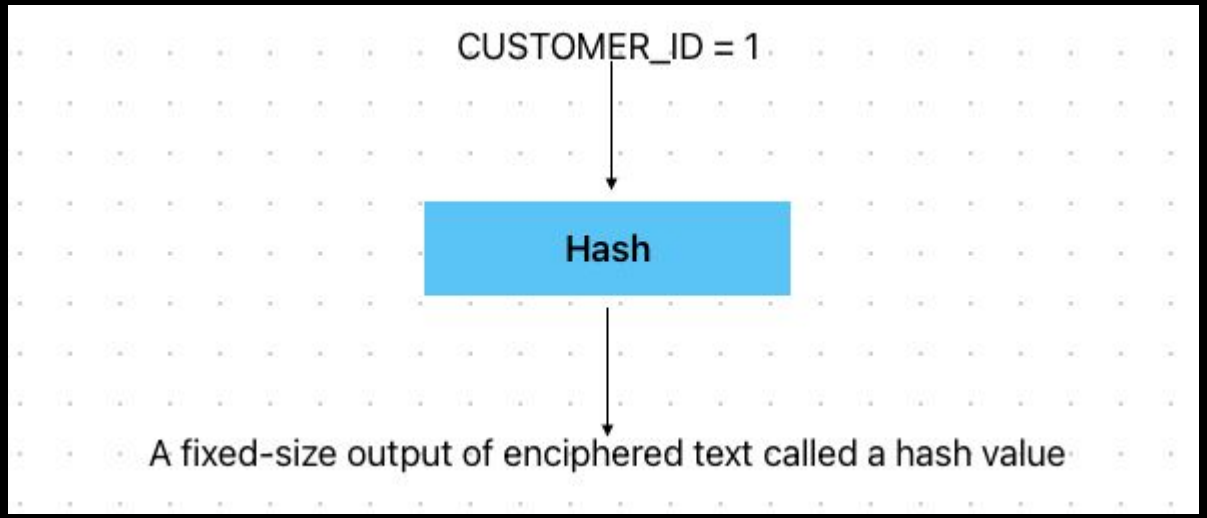
# Kafka Partitioning



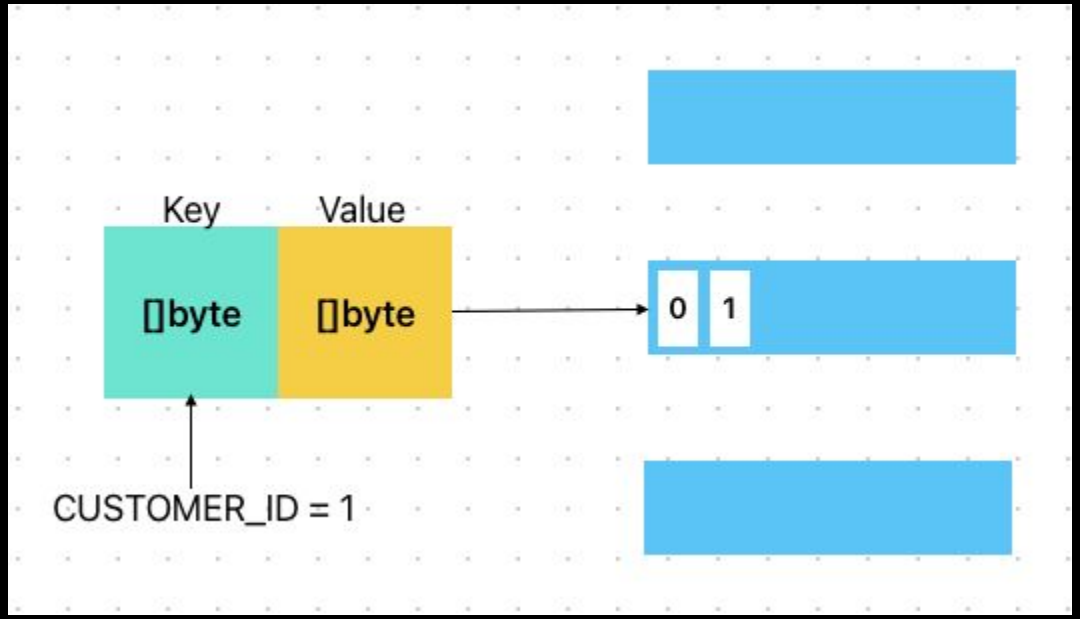
# Kafka Partitioning



# Kafka Partitioning



# Kafka Partitioning



# Kafka Brokers

We have talked about logical. (events, topics, partitions)

From a physical infrastructure standpoint, Apache Kafka is composed of a network of machines called brokers

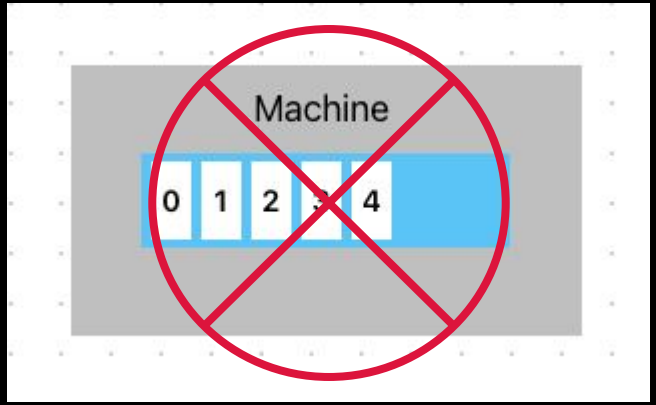
https://developer.confluent.io/learn-kafka/apache-kafka/brokers/

# Kafka Brokers

- Manage partitions
- Handle write and read requests
- Manage replication of partitions

https://developer.confluent.io/learn-kafka/apache-kafka/brokers/

# Replication

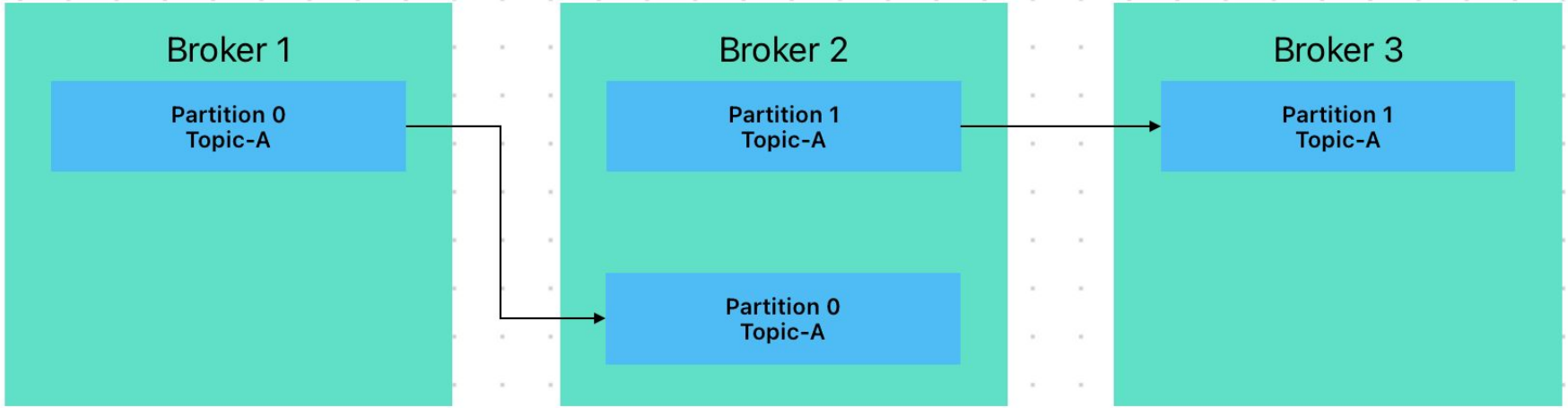


# Replication

- Copies of data
- One lead partition and N-1 followers
- Writes and reads happen to the leader
- An invisible process

https://developer.confluent.io/learn-kafka/apache-kafka/brokers/

# Replication



# Replication

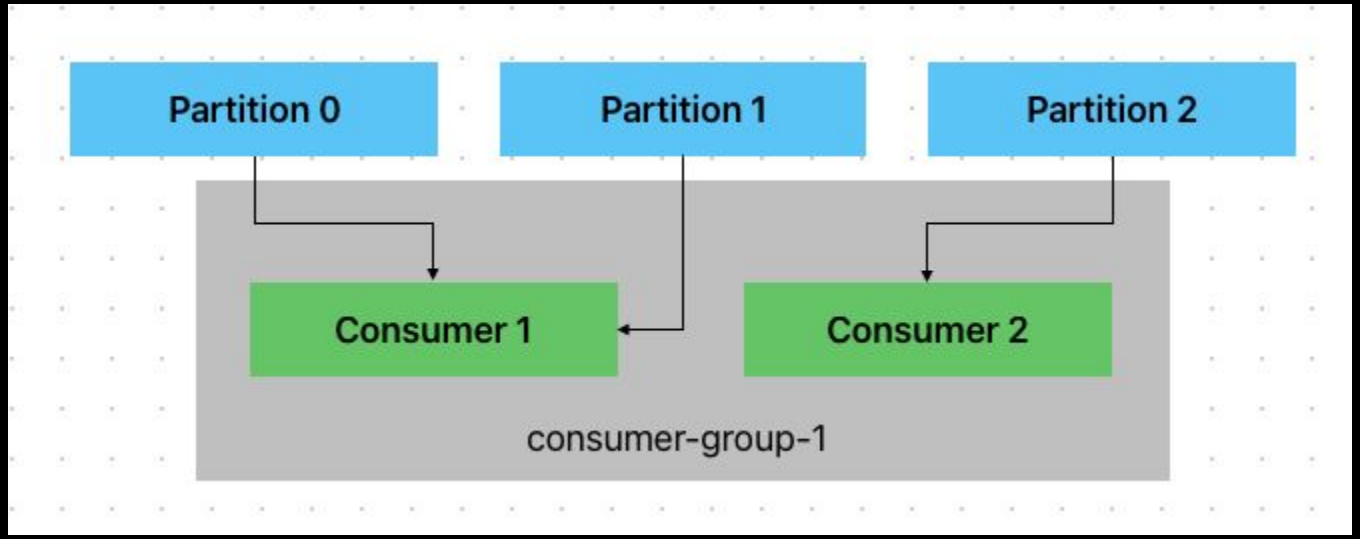
```
acks = all
```

# Kafka Producers

## Create Project

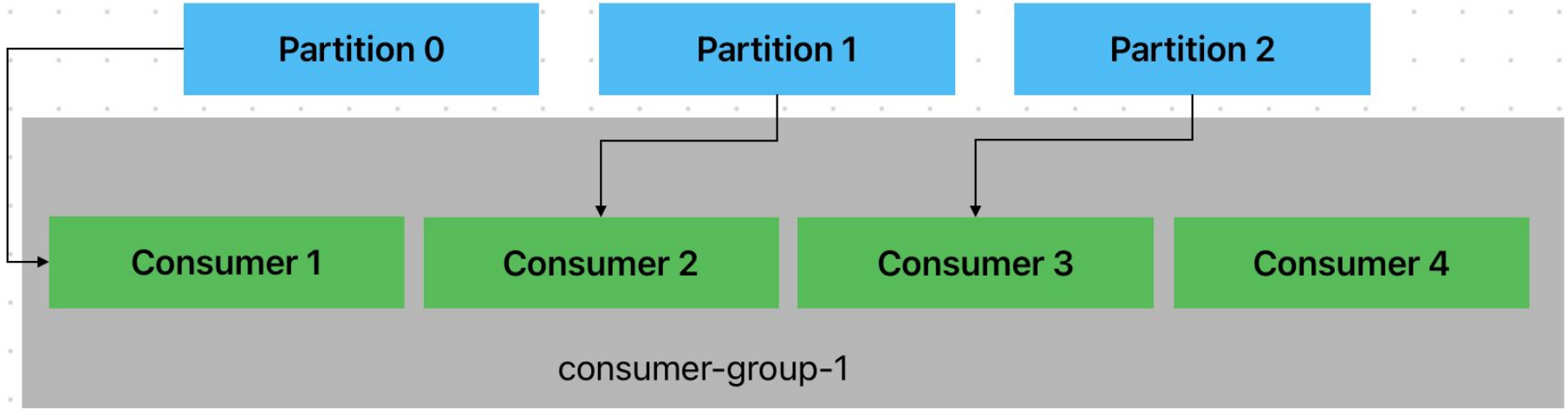
```
$ mkdir kafka-go-getting-started
$ cd kafka-go-getting-started
$ go mod init kafka-go-getting-started
$ go get github.com/segmentio/kafka-go
```

# Kafka Consumers




https://developer.confluent.io/learn-kafka/apache-kafka/brokers/

# Kafka Consumers




https://developer.confluent.io/learn-kafka/apache-kafka/brokers/

# Exercise