IYKRA

Data Fellowship Program

Mastering Kafka for Real-Time Data Stream

by Fariz Wakan

April 15th, 2024

Proprietary of IYKRA - Data Fellowship

Data Fellowship

Intro to Real-Time Data Streams

What real-time data is ..

- Real-time data refers to information that is processed, analyzed, and made **available for use immediately as it is generated**.
- In the context of computing and data processing, real-time implies **minimal delay** between the occurrence of an event and the system's response to that event.
- Real-time data systems are designed to handle and **respond to data instantly** or within a very short time frame, **providing timely insights** and **enabling rapid decision-making**.

Real-Time vs Batch

- Real-time processing is characterized by **immediacy**, **low latency**, and **continuous data streams**, making it suitable for applications requiring instant responses
- Batch processing, on the other hand, involves **processing data in intervals**, often with **higher latency**, making it suitable for non-time-sensitive tasks and large-scale data processing
- The choice between real-time and batch processing depends on the specific requirements and objectives of the application or system being developed

Why real-time data ..

The importance of real-time data streams lies in their ability to provide immediate and actionable insights. Here are key reasons highlighting the importance of real-time data streams,

- Timely decision-making,
- Improved customer experience,
- Fraud detection and security,
- IoT and smart devices,
- Healthcare monitoring,
- Event monitoring and alerting, etc.

Stream Processing Engines

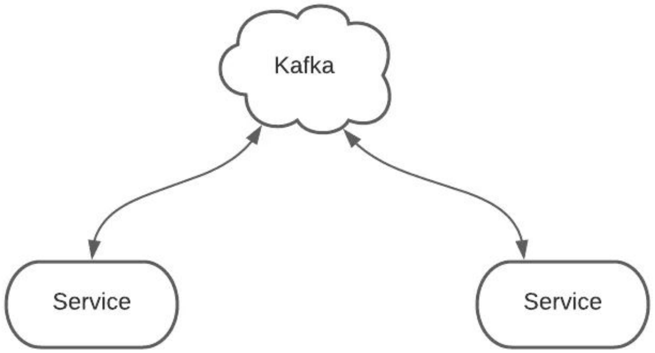
Stream processing engines are software frameworks or platforms designed to handle and analyze data in real-time as it is generated. There are several popular stream processing engines like **Apache Kafka**, Apache Flink, Apache Spark Streaming (Apache Spark), etc.

Data Fellowship

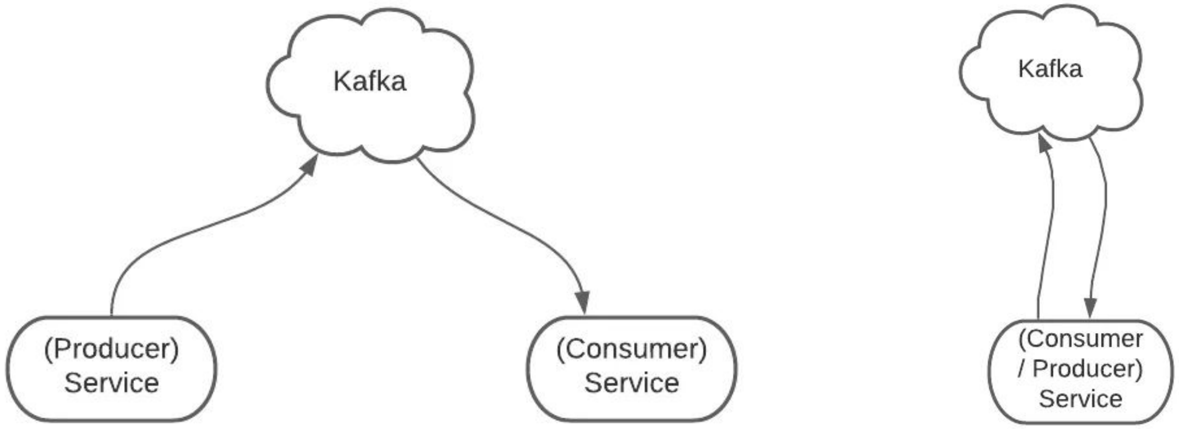
Apache Kafka Basics

Apache Kafka

Apache Kafka is an open-source **distributed streaming platform** that is widely used for building real-time data pipelines and streaming applications. It is designed to handle massive volumes of data and provides fault-tolerant, scalable, and durable event streaming.

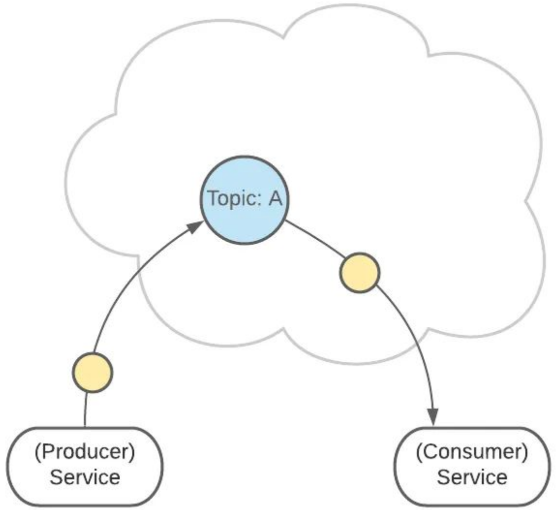


Producers and Consumers

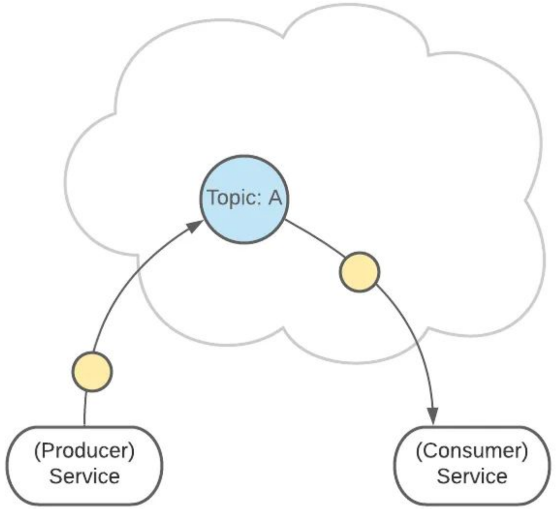


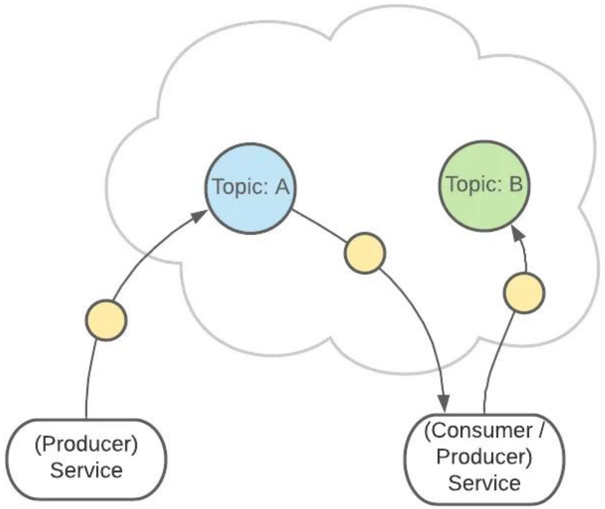
Proprietary of IYKRA - Data Fellowship

Topics

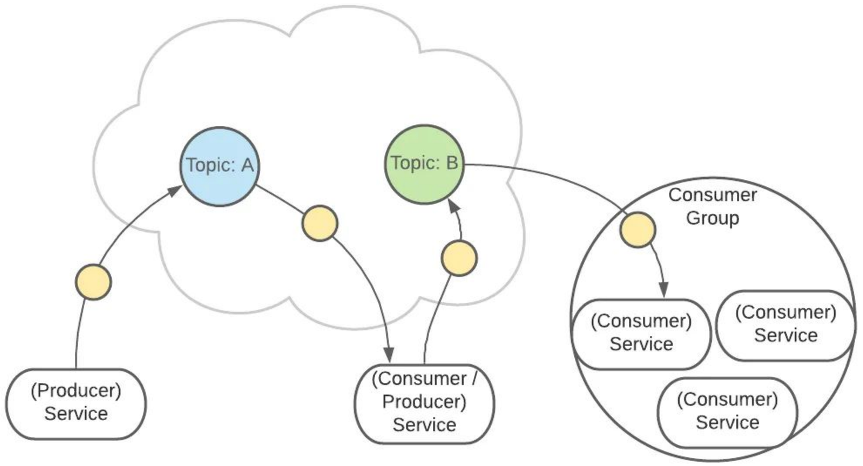


Topics

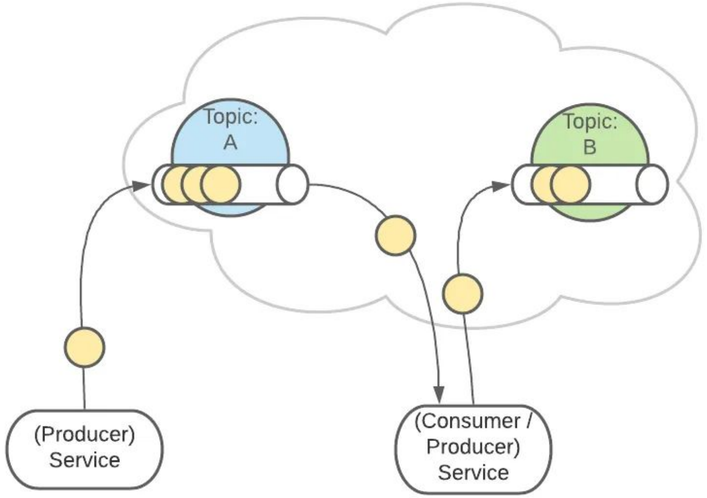

Topic: A
(Producer) Service
(Consumer) Service


Topic: A
Topic: B
(Producer) Service
(Consumer / Producer) Service

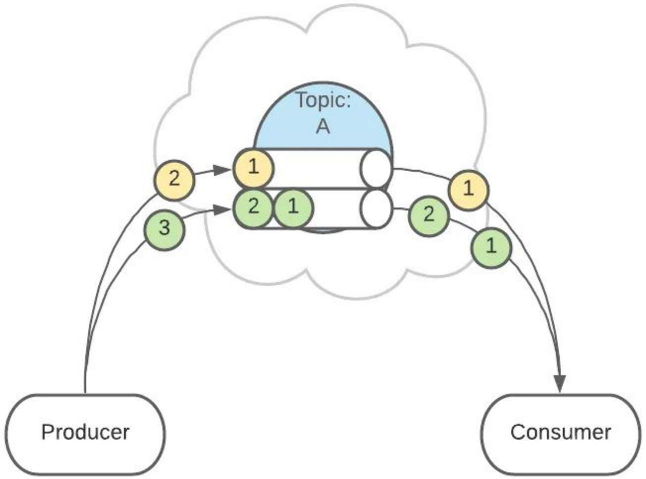
Consumer Group



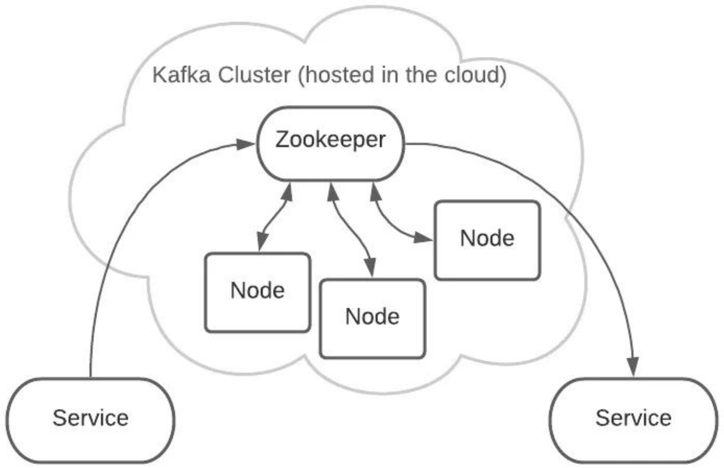
Inside topics, topic as a queue ..



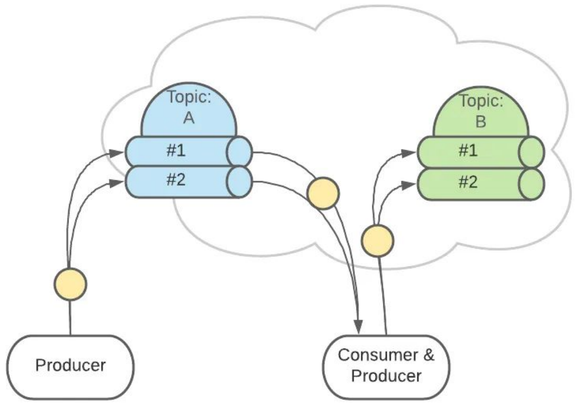
Partitions

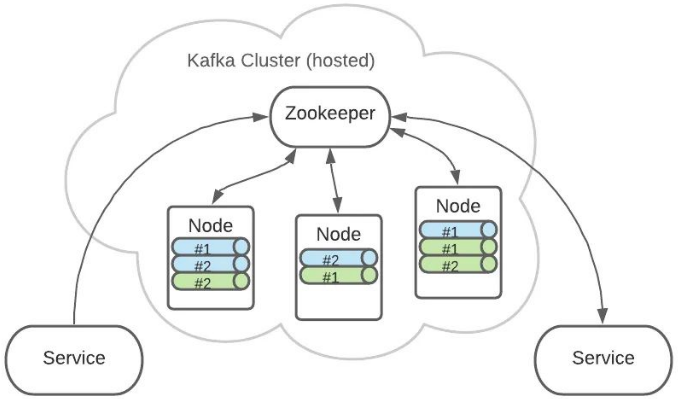


Apache Kafka infrastructure



Apache Kafka infrastructure





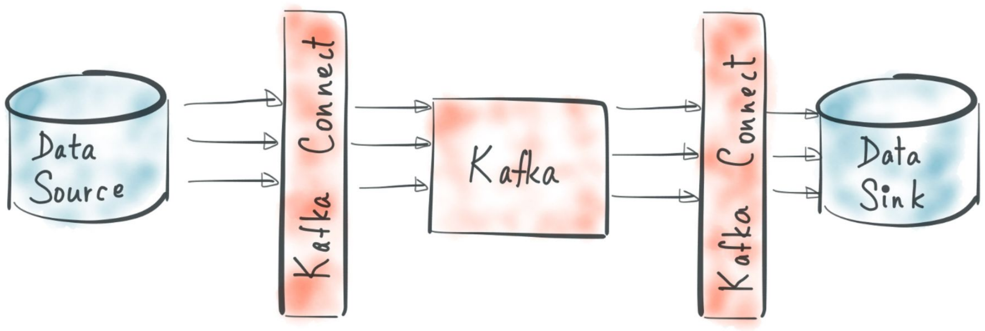
Play with Apache Kafka

Demo Session

Apache Kafka APIs

- Consumer API
- Producer API
- Connect API
- Streams API

Kafka Connect



Data Fellowship

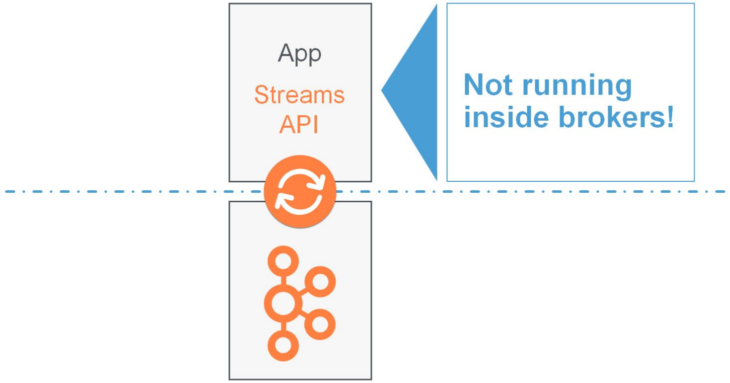
Kafka Streams

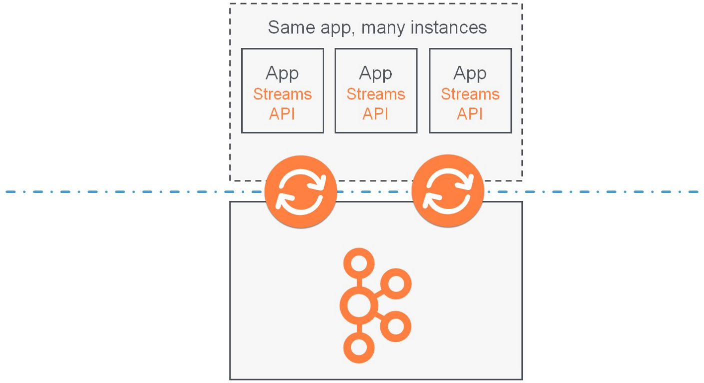
Kafka Streams

- Kafka Streams is a **client library** for building applications and microservices, where the **input and output data are stored in Kafka** clusters
- It combines the simplicity of writing and deploying **standard Java and Scala applications** on the client side with the benefits of Kafka's server-side cluster technology.

Stream Processing App

A stream processing application is any program that makes use of the Kafka Streams library





Streaming Processing Primitives

Kafka Streams offers two ways to define the stream processing topology,

- Kafka Streams DSL
  - Common data transformation operations such as map, filter, join and aggregations.
- Processor API
  - Lower-Level, allows developers define and connect custom processors.

Streaming Processing Primitives

Kafka Streams offers two ways to define the stream processing topology,

- Kafka Streams DSL
  - Common data transformation operations such as map, filter, join and aggregations.
- Processor API
  - Lower-Level, allows developers define and connect custom processors.

Kafka Streams DSL

- Built-in abstractions for **streams and tables** in the form of KStream, KTable, and GlobalKTable.
- Declarative, functional programming style with **stateless transformations** (e.g. map and filter) as well as **stateful transformations** such as aggregations (e.g. count and reduce), joins (e.g. leftJoin), and windowing (e.g. session windows).

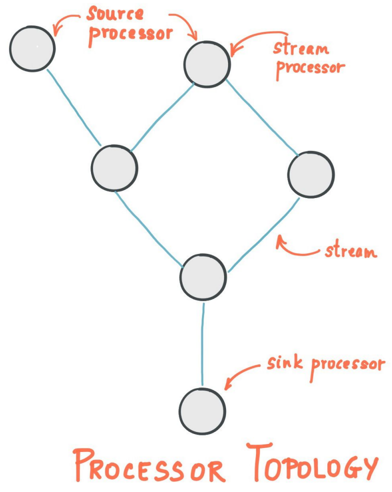
Stream vs Table

Customer	VIP status
Allison	Bronze
Rick	Silver
Rick	Platinum
Allison	Silver
Hugh	Gold

Customer	VIP status
Allison	Silver
Rick	Platinum
Hugh	Gold

Processor Topology

Kafka Streams defines its computational logic through one or more processor topologies, where a processor topology is a graph of stream processors (nodes) that are connected by streams (edges).

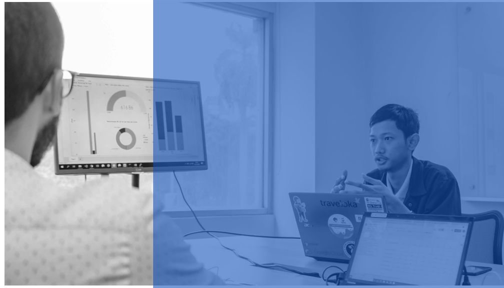


Play with Kafka Streams

Demo Session

References

- Visualizing Kafka by Timothy Stepro
- https://docs.confluent.io/platform/current/streams/concepts.html



Proprietary of IYKRA - Data Fellowship

iykra Data Fellowship

Thank you!