FLOWX.AI

PLATFORM OVERVIEW / Frameworks and standards / Event-driven architecture frameworks / intro-to-kafka-concepts

Contents

PLATFORM OVERVIEW / Frameworks and standards / Event-driven architecture frameworks / Intro to Kafka concepts

What is Kafka?

Apache Kafka is an open-source distributed event streaming platform that can handle a high volume of data and enables you to pass messages from one end-point to another.

Kafka is a unified platform for handling all the real-time data feeds. Kafka supports low latency message delivery and gives a guarantee for fault tolerance in the presence of machine failures. It can handle a large number of diverse consumers. Kafka is very fast, and performs 2 million writes/sec. Kafka persists all data to the disk, which essentially means that all the writes go to the page cache of the OS

(RAM). This makes it very efficient to transfer data from a page cache to a network socket.

Benefits of using Kafka

- **Reliability** − Kafka is distributed, partitioned, replicated, and fault tolerant
- **Scalability** − Kafka messaging system scales easily without downtime
- **Durability** − Kafka uses Distributed commit log which means messages persist on disk as fast as possible
- **Performance** − Kafka has high throughput for both publishing and subscribing messages. It maintains a stable performance even though many TB of messages are stored.

Key Kafka concepts

Events

Kafka encourages you to see the world as sequences of events, which it models as key-value pairs. The key and the value have some kind of structure, usually represented in your language's type system, but fundamentally they can be anything. Events are immutable, as it is (sometimes tragically) impossible to change the past.

Topics

Because the world is filled with so many events, Kafka gives us a means to organize them and keep them in order: topics. A topic is an ordered log of events.

When an external system writes an event to Kafka, it is appended to the end of a topic.

In FLOWX.AI, Kafka handles all communication between the FLOWX Engine and external plugins and integrations. It is also used for notifying running process instances when certain events occur. More information about KAFKA configuration on the section below:

» FLOWX engine setup guide

Producer

A producer is an external application that writes messages to a Kafka cluster, communicating with the cluster using Kafka's network protocol.

Consumer

The consumer is an external application that reads messages from Kafka topics and does some work with them, like filtering, aggregating, or enriching them with other information sources.

» How to create a Kafka producer

» How to create a Kafka consumer

In-depth docs

» Kafka documentation

» How Kafka works

Was this page helpful?