instaclustr

Use Cases

Message Buffer

Providing a reliable, scalable location to temporary store messages to smooth out peaks in message arrival rates or hold messages while processing systems are unavailable.

Use Case Example

Public facing Internet systems where load spikes can occur fairly unpredictably (eg. Slack).

IoT - where volumes tend to be relatively predictable but readings never stop - store up data while processing systems are down.

Log processing - here errors can very quickly increase the volume of logs being produced and overwhelm processing systems. Using Kafka as a buffer can help smooth out these peaks.

Architectural Decoupling

Make your application architecture flexible by hiding knowledge of the applications consuming events from the applications producing them. Commonly associated with the microservices architecture.

Use Case Example

Virtually unlimited. Can be of in any application environment - moderately or extremly complex.

Example financial services space, you might have a core processing system processing transactions and then publishing them to a Kafka queue where they can be used by any number of down-stream applications such as risk analytics or open banking APIs. New downstream applications can be added without any change to the core processing system.

Stream Processing

Implement application logic based on the a streams of events. The simplest example is probably keeping a running count of the number of occurrences of a particular type of event. Slightly more complex examples include calculating an average value over a 5 minute moving window or finding all events in one topic that have a corresponding event in another topic.

Use Case Example

IoT application where the devices are automated thermometers sending temperature readings. Using stream processing, you could keep track of the average temperature over the preceding hour and perform and action if any reading deviated from that by more than say 10%.

In a monitoring system, detecting when a application state is change from "UP" to "DOWN" multiple times in a given period.

Event Sourcing

Event sourcing or the Kappa architecture are architectural patterns where you consider the sequence of changes made (as opposed to the result of those changes) as the truth source for your application. Providing advantages such as traceability and being able to replay changes when requirements for how to represent them change. Kafka is designed for reliably storing a series of events and can provide an ideal data store for this purpose. Kafka's support for compaction can also assist in this case.

Use Case Example

A banking system is a simple example to understand. We consider the series of transactions (deposits and withdrawals) to be the source of truth for the system. The account balance is merely a summary based on this transaction. If this account balance is somehow corrupted or needs to be stored in a different manner it can be recalculated based on the stored transactions.

instaclustr

Build | Deploy | Manage | Scale | Connect

Build your application on a fast, scalable and distributed streaming platform with Instaclustr's Managed Apache Kafka.

Learn more