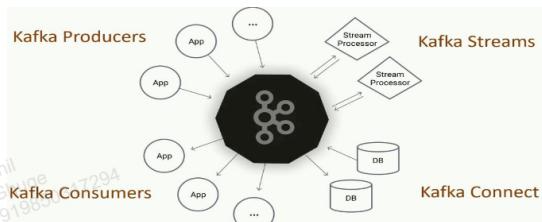


Apache Kafka

What is Apache Kafka?

Apache Kafka is an **open-source distributed event streaming platform** used by thousands of companies for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications.



Apache Kafka is a messaging broker, everything else is a library, framework either interact with the kafka broker or work with the data in the realtime. Kafka broker is a middle man between the producers and the consumers and it plays 3 primary responsibilities:

1. Receive messages from the producers and acknowledge the successful receipt.
2. Store the messages in a log file to safeguard it from potential loss. Storage is also critical to ensure that the consumers can consume it later and they need not read it in real-time.
3. Deliver the messages to the consumers when they request for it.

"Apache Kafka is a horizontally scalable, fault-tolerant, distributed event streaming platform" and is designed for building the real time streaming data architecture.

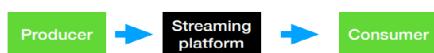
To Understand the Kafka Architecture, we need to understand the

- Kafka Storage Architecture
- Kafka Cluster Architecture
- Work Distribution Architecture

What is an Event Streaming Platform?

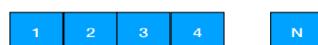
In Event Streaming Platform, there is one or more producers and one or more consumers

- A. subscribe to a stream of records.



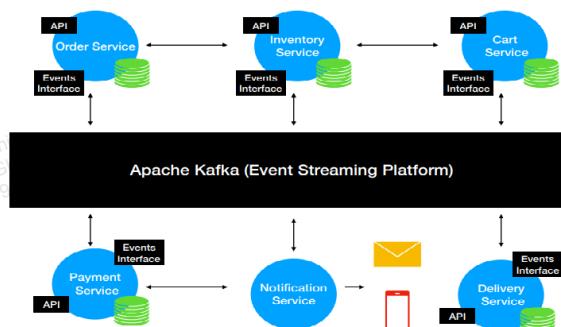
- B. Store stream of Events

Apache Kafka



- C. Analyze and Process Events as they occur

Event Streaming Platform



Differences between the traditional Message System and Kafka Streaming Platform

Traditional Messaging System	Kafka Streaming Platform
Transient Message Persistence. Meaning the message once received by any customer, the message will be deleted from the queue.	Stores events in the file system based on a retention time by default this retention period is 7 days. Events are immutable.

Message will be deleted from the queue.	Days. Events are immutable
Target a specific Consumer	Any Consumer can access a message from the broker
Not a distributed system	It's a distributed streaming system because of its publish and subscribe messaging system

Kafka Use Cases



Anil
Chuge
919850317294

Apache Kafka

Kafka Terminology:

Messaging System

Apache Kafka adopted a messaging system pattern to solve the real-time data integration problem. As the name suggests, a messaging system is all about sending small chunks of data from one system to other systems.

For our retail-chain example, a single invoice could be considered as a message. When we talk about messages in this context, it could be a JSON record or something similar.

There are two approaches to messaging.

1. Point to Point
2. Publish/Subscribe or pub/sub

Point to Point

A point-to-point (PTP) messaging system is built on the concept of message queues, senders, and receivers. Each message is addressed to a specific queue, and receiving clients extract messages from their designated queues. These queues retain all messages sent to them until the messages are consumed or expire.

PTP messaging has the following characteristics:

1. Each message has only one consumer.
2. Sender and receiver are not dependent on each other.
3. The receiver acknowledges the successful receipt of a message.

As the name suggests, PTP is more suitable for the one-to-one scenario. Apache Kafka rejected PTP approach because they wanted to address the many-to-many requirement.



Publish/Subscribe or pub/sub

A publish/subscribe (pub/sub) messaging system is built on the concept of publishers, broker, topic, and subscribers.

Publisher

Any application that sends data should act as a publisher. For our retail-chain example, the POS application works as a publisher, and it sends the invoices for other apps. People use different names such as data publisher, data producer, and sender, however, it means the same thing – a Publisher.

Anil
Chuge
919850317294

Apache Kafka

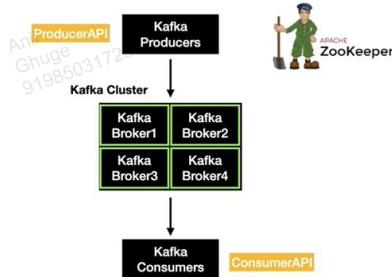
Subscriber

An application that reads data sent by the publisher is a subscriber. For our retail-chain example, services such as shipment and loyalty are subscribers. They read invoices that are posted by the POS.

systems. We might notice people referring to them as data consumers or receivers; however, all that means the same thing – a Subscriber.

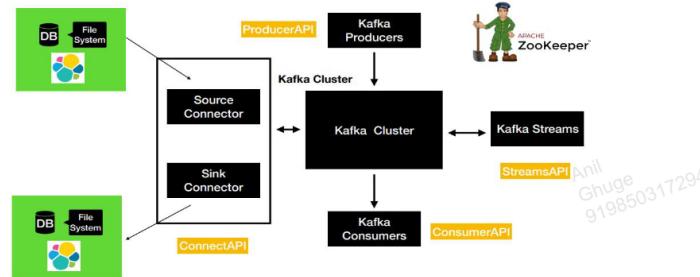
Broker

The Broker is at the heart of the pub/sub messaging system. The Broker is responsible for receiving messages from the publishers, storing them into a log file, and sending the messages to the subscribers.



The broker acts as a middleman between publishers and subscribers. Apache Kafka is a message broker, and it offers a set of APIs to implement Publishers and Subscribers.

- ✓ Producer API
- ✓ Consumer API
- ✓ Connect API and
- ✓ Stream API



Apache Kafka

Any application that wants to send a message should use producer API to send the data to the Kafka Broker. The broker receives the message, sends acknowledgment and persists the data into a log file.

When a consumer application wants to read the message, they use consumer API to read the messages from the broker.

Anil
Ghuge
919850317294