**Amazon Managed Streaming for Apache Kafka**

**What is Kafka?**

It is a distributed data store optimized for ingesting and processing **streaming data** in **real-time**.

**Main Functions/ Need:**

* Publish and subscribe to streams of records
* Effectively store streams of records in the order in which records were generated
* Process streams of records in real time

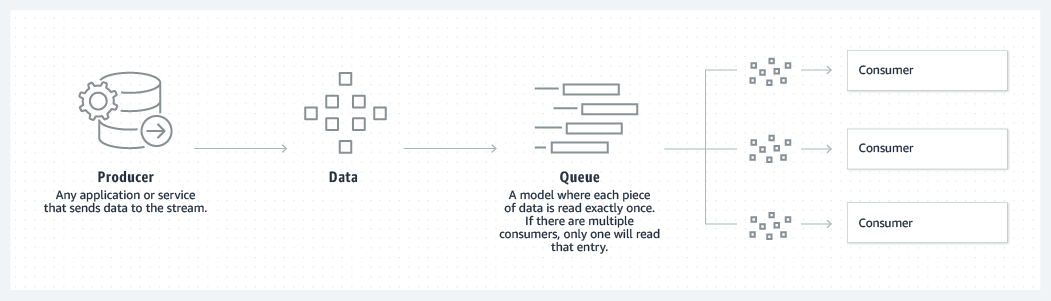
**Popular Use-Cases:**

* If you want to create a data pipeline that takes in user activity data to track how people use your website in real-time, Kafka would be used to ingest and store streaming data while serving reads for the applications powering the data pipeline.
* Kafka is also often used as a message broker solution, which is a platform that processes and mediates communication between two applications.

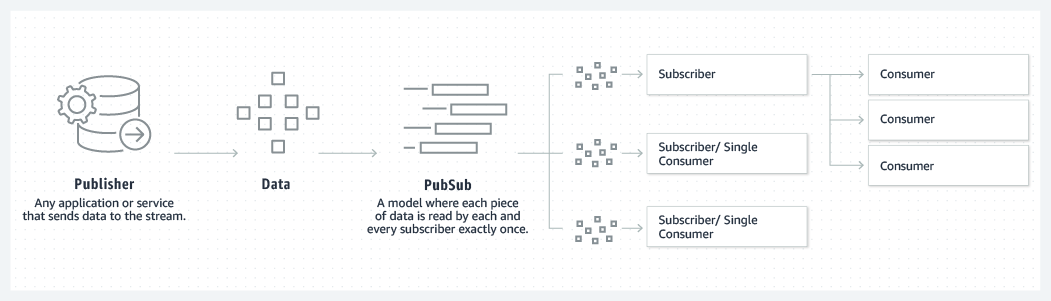
**How does it work?**

Kafka combines two messaging models,

1. **Queuing (not multi-subscribers)**



1. **Publish-subscribe (multi-subscriber)**



Kafka uses a partitioned log model to stitch together these two solutions. A log is an ordered sequence of records, and these logs are broken up into segments, or partitions, that correspond to different subscribers. This means that there can be multiple subscribers to the same topic and each is assigned a partition to allow for higher scalability.

Kafka remedies the two different models by publishing records to different topics. Each topic has a partitioned log, which is a structured commit log that keeps track of all records in order and appends new ones in real time. These partitions are distributed and replicated across multiple servers, allowing for high scalability, fault-tolerance, and parallelism. Each consumer is assigned a partition in the topic, which allows for multi-subscribers while maintaining the order of the data.

**APIs in Kafka:**

* **Producer API:**

used to publish a stream of records to a Kafka topic.

* **Consumer API:**

used to subscribe to topics and process their streams of records.

* **Streams API:**

enables applications to behave as stream processors, which take in an input stream from topic(s) and transform it to an output stream which goes into different output topic(s).

* **Connector API:**

allows users to seamlessly automate the addition of another application or data system to their current Kafka topics.

**Hands-on:**

Step 1: Create an Amazon MSK Cluster

Step 2: Create a Client Machine

Step 3: Create a Topic

Step 4: Produce and Consume Data

Step 5: Use Amazon CloudWatch to View Amazon MSK Metrics

Step 6: Delete the AWS Resources Created for This Tutorial

**Link:** [MSK Labs (workshops.aws)](https://catalog.us-east-1.prod.workshops.aws/workshops/c2b72b6f-666b-4596-b8bc-bafa5dcca741/en-US/overview/overview)