

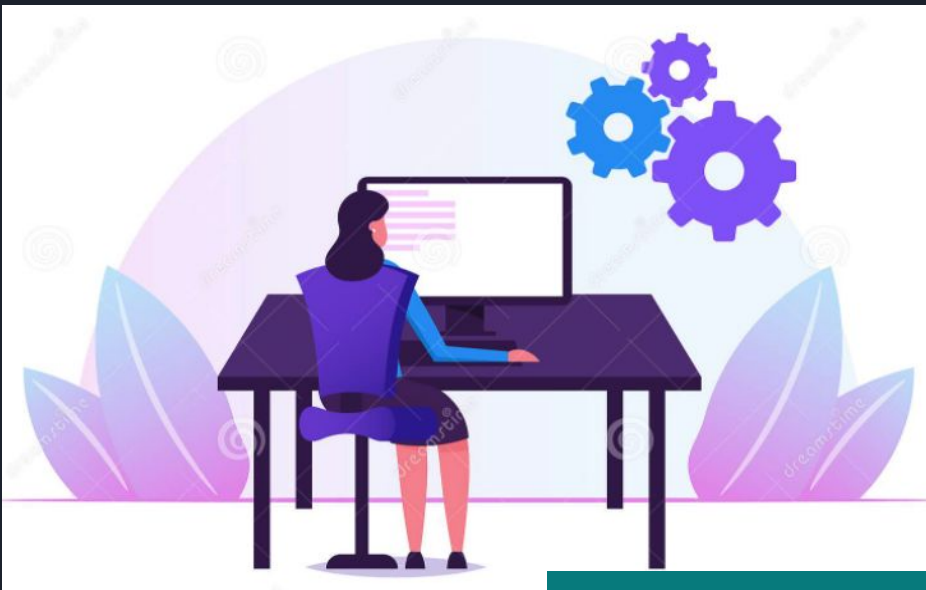


# Welcome!

- We'll start in a moment :)
- We are NOT recording tonight's event. We may plan to take screenshots for social media.
  - ***If you want to remain anonymous***, change your name & keep video off.
- We'll introduce the hosts and break in-between for Q/A.
- We will make some time for Q&A at the end of the presentation as well.
- You can come prepared with questions. And, feel free to take notes.
- Online event best practices:
  - Don't multitask. Distractions reduce your ability to remember concepts.
  - Mute yourself when you aren't talking.
  - We want the session to be interactive.
  - Feel free to unmute and ask questions in the middle of the presentation.
  - Turn on your video if you feel comfortable.
  - Disclaimer: Speaker doesn't know everything!

## Check out:

- [Technical Tracks](#) and [Digital Events](#)
- Get updates – join the [Digital mailing list](#)
- Give us your feedback – take the [Survey](#)



## WWCode Digital + Backend Study Group

Feb 17, 2022



# Backend Study Group

- Welcome from WWCode!
- Our mission: Inspiring women to excel in technology careers.
- Our vision: A world where women are representative as technical executives, founders, VCs, board members and software engineers.



**Rittika Adhikari**  
Software Engineer, Confluent

<https://www.linkedin.com/in/rittika-adhikari/>



**Harini Rajendran**  
Software Engineer, Confluent

<https://www.linkedin.com/in/harajendran/>



**Prachi Shah**  
Director, Women Who Code San Francisco

<https://www.linkedin.com/in/prachishshah/>



# Introduction to Apache Kafka

Presenter: Rittika Adhikari

# Agenda

What is Apache Kafka

Apache Kafka Architecture

Apache Kafka Applications

Pros and Cons of Apache Kafka

Demo

What's Next for Kafka

What do we work on at  
Confluent?



# What is Apache Kafka?



# What is Apache Kafka?

*“Apache Kafka is an open-source distributed event-streaming platform, which is used to build high-performance data pipelines, streaming analytics, data integration, and mission-critical applications.”*



# kafka





# Okay... but what *actually* is Apache Kafka?

- Billions of sources continuously generate *streams* of data / events
  - An event is an (action, time) pair
    - ex: choosing a seat on a flight, requesting an Uber, tire pressure at a certain time
- We want to use these *streams* for *real-time* processing
  - Continuously *consume* and *process* these streams at high speeds, while maintaining the correct *ordering*
    - ex: messaging applications



## Fundamentally, Apache Kafka is...

- A distributed system of **servers** and **clients** that communicate via TCP
- Deployable on bare-metal hardware, VMs, and containers on-premise or in the cloud
- Run as a “cluster” on 1+ servers which can span several datacenters
- Each Kafka Cluster stores streams of *records* (key, value, timestamp) in *topics* (categories)



## Apache Kafka allows us to:

1. Publish (Write) and Subscribe (Read) streams of events continuously.
2. Store streams of events durably & reliably for as long as you want.
3. Process streams of events as they occur, or after they have occurred.



# Apache Kafka offers:

## Permanent Storage

Store data in distributed, durable, fault-tolerant cluster



## High Throughput

Deliver messages at network limited throughput



## High Availability

Connect across geographic regions / availability zones



## Scalable

Thousands of brokers, petabytes of data, etc.



## Stream Processing

Read / Write to streams of data / events



## Connect

Interfaces with several other apps (Postgres, S3, etc.)



## Client Libraries

Stream processing in other languages



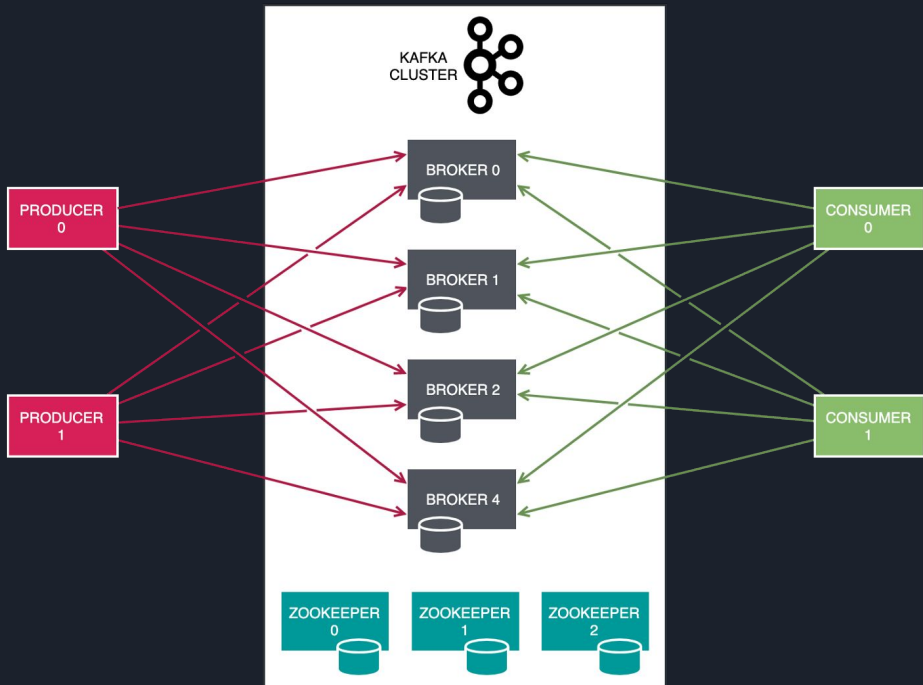
## Open-Source





# Apache Kafka Architecture

# Apache Kafka Architecture



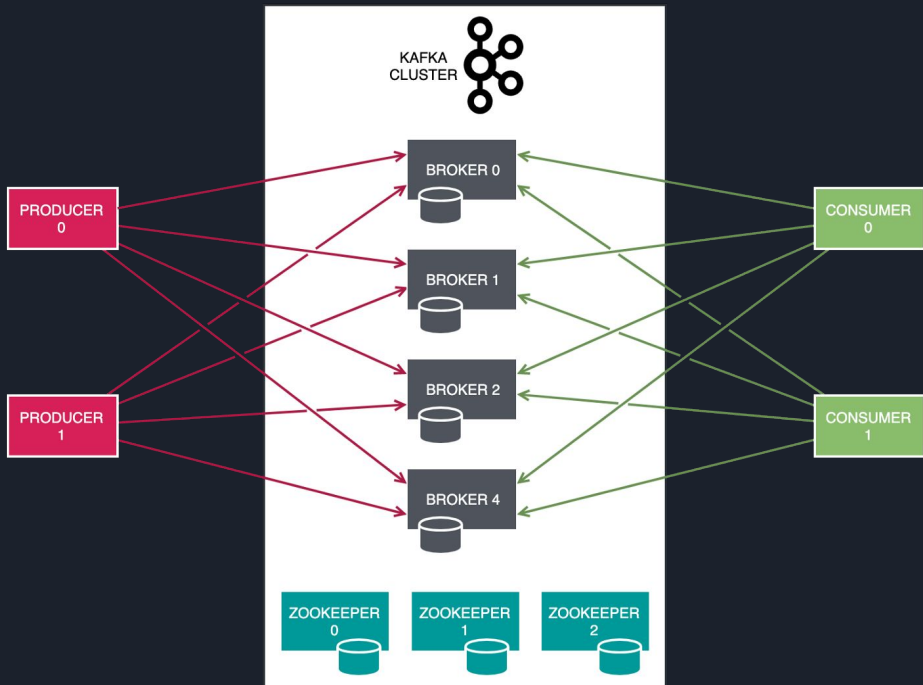
## Broker

Mediator between the Producer & Consumer  
Receives messages from Producer & stores them on-disk, keyed by a unique offset

## Zookeeper

Coordinates which Broker is the leader/follower for a specific replica

# Apache Kafka Architecture



## Producers

Publishes records to the leader of the topic-partition

Leader appends record to commit log & assigns offset

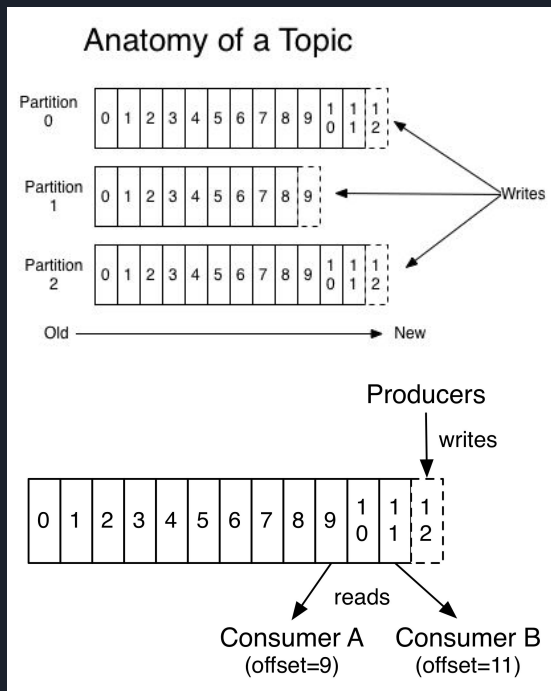
## Consumers

Consumes from a topic-partition starting at a specific offset

## Consumer Group

1+ consumers which consume from a topic

# Topics & Partitions



## Topics

- A category/feed-name to which records are published
- Multi-subscriber (0+ consumers subscribe to the topic)

## Partitions

- Each partition is an ordered, immutable sequence of records that is continually appended to
- Partitions maintain an offset to uniquely identify records
- Partitions are replicated across servers for fault-tolerance





# Apache Kafka APIs

## Producer

Applications *publish* a stream of records to 1+ topics

## Consumer

Applications *subscribe* to topics & process that stream of records

## Streams

Applications act as a *stream processor*, consuming an input stream from 1+ topics & producing an output stream to 1+ topics

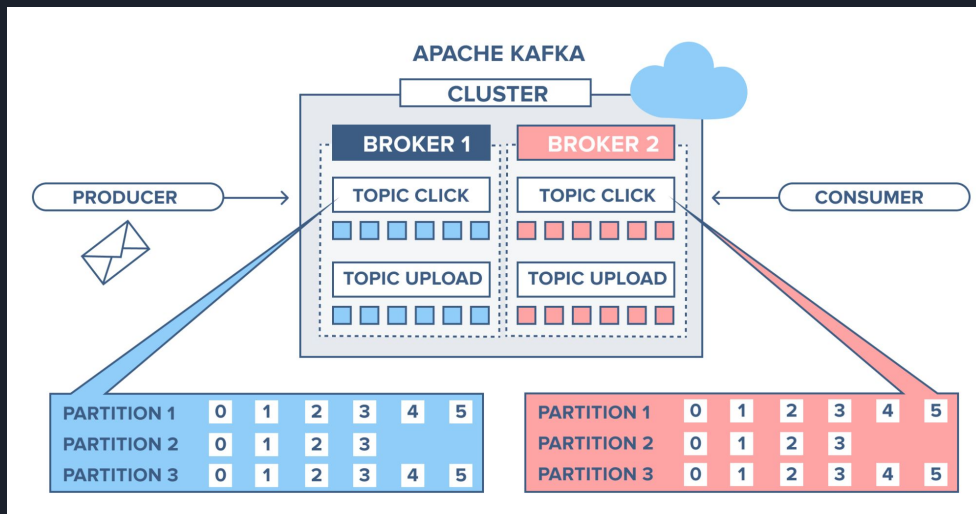
## Connector

Builds/Runs reusable producers & consumers that connect topics to existing applications



# Apache Kafka Applications

# Website Activity Tracking



## Goal

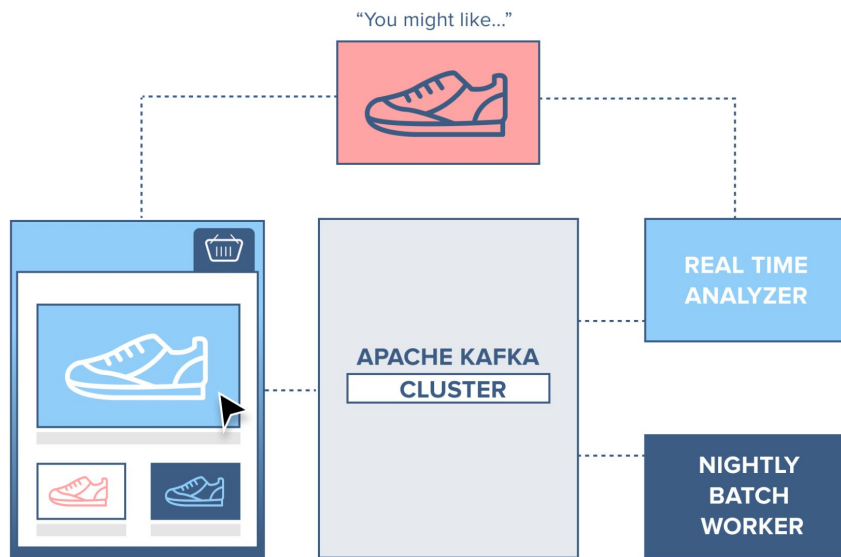
Track user-actions on a website (i.e. clicks, uploads, page views, searches, etc.)

## Application

### Instagram

1. User with id 1 likes a post
2. Web application publishes user id's like to topic-partition click-1
3. Web application consumes records from click-1, updating the number of likes on the post in real-time

# Online Shop



## Goal

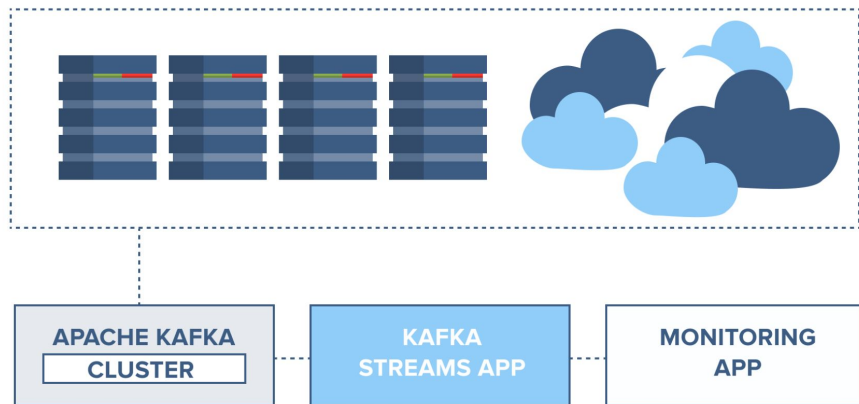
Recommend products based on prior clicks

## Application

Amazon

1. User searches for sneakers, and clicks on a pair of Adidas fashion sneakers. This event is sent to AK cluster.
2. Real Time Analyzer consumes this event and suggests a different pair of fashion sneakers.
3. Nightly Batch Worker consumes this event and sends an email with suggested products.

# Application Health Monitoring



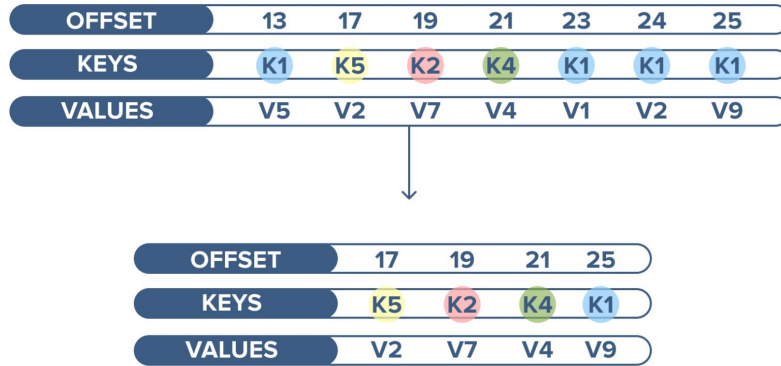
## Goal

Monitor & trigger alerts when there are issues with a server

## Application

1. Server agent produces events to the AK cluster
2. Kafka Streams app consumes these events and sends them to the monitoring app
3. Monitoring app decides whether to trigger an alert based on the current server status

# Database



## Goal

Store data in key, value form

## Application

1. Produce data records to the log.
2. Due to log compaction, when you want to update a (key, value) pair, you just produce a record with the same key but an updated value.



# Pros vs. Cons of Apache Kafka

## Pros

- Low latency
- High throughput
- Fault-tolerant
- Scalability
- Real-time
- Distributed
- High Concurrency
- Persistent
- Durable

## Cons

- No complete set of monitoring tools
- No wildcard topic matching
- Issues with tweaking messages
- Compresses messages, which can lead to reduced performance when decompressing



# Demo





# What's Next for Kafka?

## **KIP-405: Tiered Storage**

Currently, we cannot scale storage without also scaling compute

Separate the storage layer from the compute layer by tiering data to the cloud

## **KIP-500: Replace Zookeeper with Self-Managed Metadata Quorum**

Currently, we are dependent on Zookeeper for leader-election

Remove this dependency through KRaft, which will allow us to manage metadata in a more scalable & robust way

Simplifies deployment and configuration



# What do we work on at Confluent?

- Build Confluent Cloud, a fully managed cloud-native service for Apache Kafka
- Proprietary features for our enterprise version of Apache Kafka (Confluent Platform)
- Contributing back to the community by assisting in & completing KIPs (Kafka Improvement Proposals)

# Backend Study Group



**WWCode Slack Handle: Rittika Adhikari**



<https://www.linkedin.com/in/rittika-adhikari/>

## Resources and References:

- <https://kafka.apache.org/>
- <https://www.ibm.com/cloud/learn/apache-kafka>
- <https://docs.confluent.io/5.5.1/kafka/introduction.html#>
- <https://kafka.apache.org/intro>
- <https://www.cloudkarafka.com/blog/part1-kafka-for-beginners-what-is-apache-kafka.html>
- <https://data-flair.training/blogs/advantages-and-disadvantages-of-kafka/>
- <https://medium.com/swlh/apache-kafka-in-a-nutshell-5782b01d9ffb>

## Backend Study Group:

- Presentations and session recordings found here: WWCode YouTube channel

*You can unmute and talk or use the chat.*

