



Kafka là một trong những nền tảng xử lý luồng dữ liệu phổ biến nhất hiện nay, hàng nghìn tổ chức hàng đầu trên thế giới đang sử dụng Kafka cho các đường ống dẫn dữ liệu hiệu suất cao, phân tích luồng hay tích hợp dữ liệu.

## 1. Kafka là gì?

Apache Kafka là một nền tảng phát triển trực tuyến sự kiện phân tán mã nguồn mở, được phát triển ban đầu bởi LinkedIn vào năm 2011 như một message broker thông lượng cao, để sử dụng trong chính hệ thống của mình. Sau đó, Kafka đã trở thành dự án mã nguồn mở và được chuyển giao cho Apache Software Foundation.



Kafka được thiết kế để xử lý dữ liệu streaming real-time, tức là xử lý các luồng dữ liệu không ngừng trôi qua. Các dữ liệu này có thể bao gồm cả sự kiện hay bản ghi dữ liệu, và chúng có thể được tạo ra từ hàng tỷ nguồn khác nhau như thiết bị cảm biến, ứng dụng web, hệ thống máy chủ và nhiều nguồn dữ liệu khác.

Các ứng dụng của Kafka rất đa dạng, ví dụ như việc lưu trữ dữ liệu, phân tích dữ liệu thời gian thực, tích hợp hệ thống hoặc xây dựng các ứng dụng phản hồi dữ liệu trong thời gian thực. Kafka cho phép các nhà phát triển xây dựng, triển khai các ứng dụng xử lý dữ liệu với tốc độ cao, đáng tin cậy và chính xác.

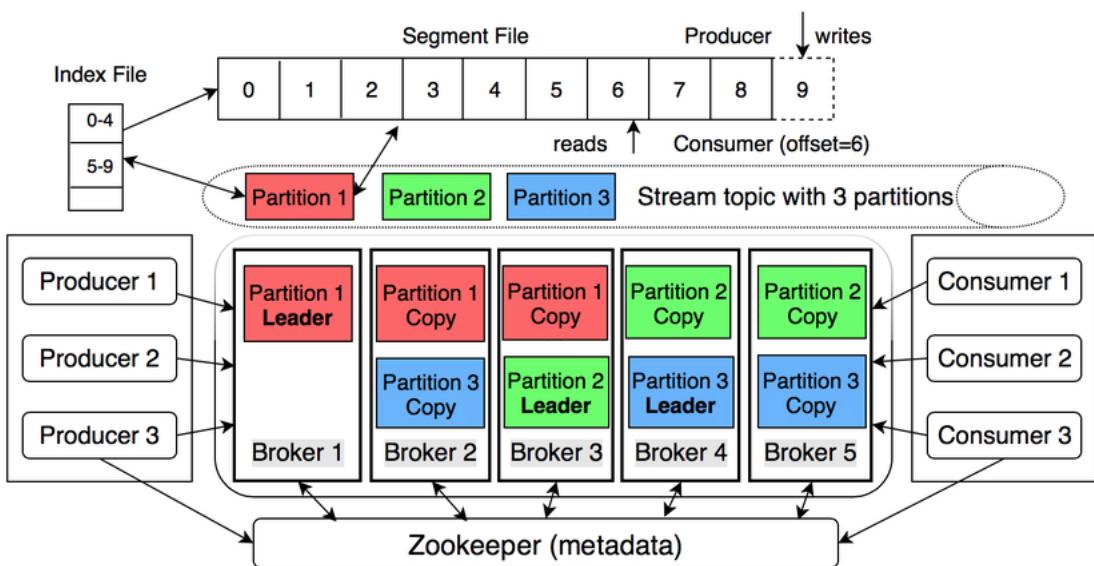
Hiện nay, Kafka là một trong những nền tảng stream dữ liệu phân tán được sử dụng rộng rãi nhất, khi nó có khả năng nhập và xử lý hàng nghìn tỷ bản ghi mỗi ngày mà không gặp vấn đề về hiệu suất quá nhiều. Các tổ chức hàng đầu như Target, Microsoft, AirBnB và Netflix đều sử dụng Kafka để cung cấp trải nghiệm theo thời gian thực và dữ liệu chính xác cho khách hàng của họ.



## 2. Những khái niệm liên quan

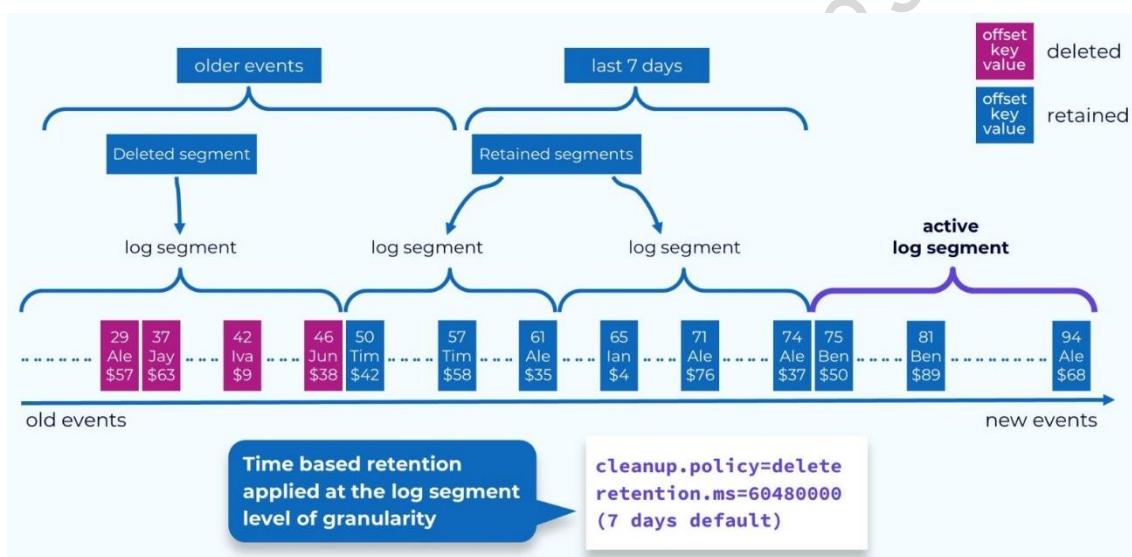
Khi làm việc với Kafka, có một số khái niệm cơ bản sau mà bạn nên để ý:

- Producer: Các ứng dụng tạo ra dữ liệu và gửi các thông điệp định dạng tới máy chủ Kafka. Dữ liệu này thường được gửi dưới dạng mảng byte tới máy chủ.
- Cluster: Một tập hợp các máy chủ, trong đó mỗi tập hợp được gọi là một Broker.
- Broker: Một máy chủ Kafka, đóng vai trò là cầu nối giữa Producer và Consumer để họ có thể trao đổi thông điệp với nhau.
- Partition: Trong trường hợp một topic nhận nhiều hơn số lượng thông điệp quy định trong một khoảng thời gian, chúng ta có thể chia topic này thành các partition. Các partition được chia sẻ giữa các máy chủ trong Cluster để xử lý các thông điệp này.
- Consumer: Các ứng dụng đọc các thông điệp từ một partition bất kỳ trong Kafka. Consumer cho phép người dùng mở rộng số lượng thông điệp được sử dụng tương tự như cách Producer cung cấp các thông điệp.
- Consumer Group: Các nhóm Consumer tổ chức lại với nhau, sử dụng cho một topic cụ thể. Mỗi consumer trong một nhóm chỉ đọc các thông điệp từ một partition duy nhất.
- Topic: Dữ liệu được truyền trong Kafka theo dạng chủ đề (topic). Khi cần truyền dữ liệu cho các ứng dụng riêng biệt, các topic khác nhau sẽ được tạo ra tương ứng.
- Zookeeper: Được sử dụng trong việc quản lý, bố trí và triển khai các Broker, đảm bảo tính liên tục của hệ thống Kafka.



Ngoài các khái niệm cơ bản này, còn có một số khái niệm khác liên quan đến Kafka mà bạn có thể quan tâm khi làm việc với nền tảng:

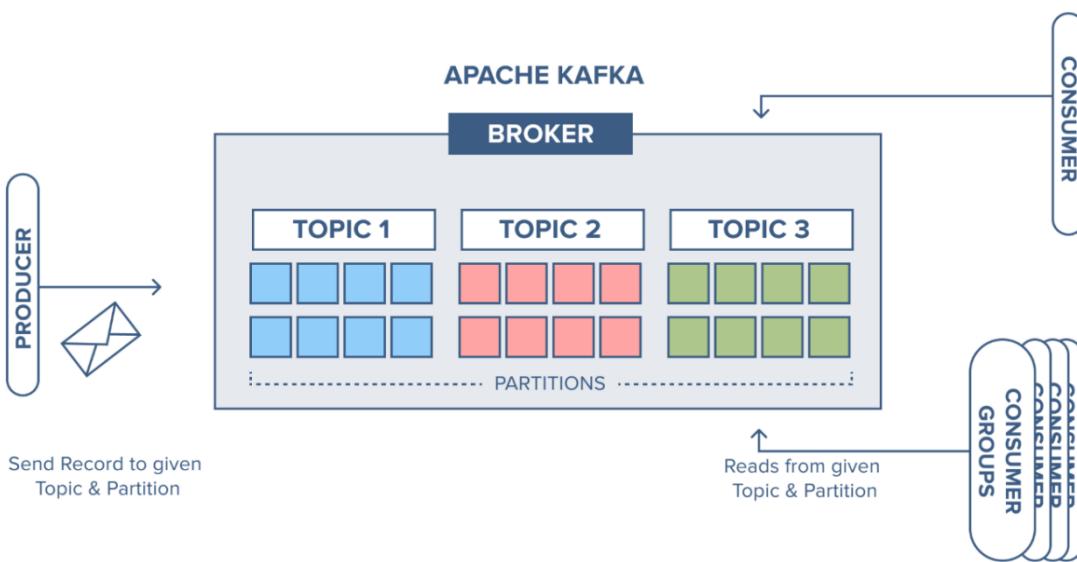
- Offset: Là một con số duy nhất mô tả vị trí của một record trong một partition. Consumer sẽ duy trì offset để theo dõi việc xử lý của mình trong partition.
- Replication: Quá trình sao chép các partition từ một broker sang các broker khác trong một cluster.
- Leader và Follower: Trong mỗi partition, có một broker đóng vai trò là leader, nhận và xử lý các yêu cầu ghi và đọc từ producer và consumer. Các broker khác sẽ là follower, sao chép dữ liệu từ leader để lưu.
- Retention Policy: Quy định cách Kafka lưu trữ dữ liệu trong các topic, bao gồm thời gian giữ lại (time-based retention) và kích thước dung lượng (size-based retention).
- Consumer Lag: Sự chênh lệch giữa offset cuối cùng của một partition và offset mà consumer đang xử lý, thường được sử dụng để đo lường sự trễ xử lý của consumer so với producer.
- Kafka Connect: Một framework để kết nối Kafka với các hệ thống, nguồn dữ liệu bên ngoài, giúp dễ dàng nhập và xuất dữ liệu đi từ và đến Kafka.
- Kafka Streams: Thư viện cho phép xây dựng các ứng dụng xử lý dữ liệu phức tạp trên Kafka, hỗ trợ chuyển đổi dữ liệu, tính toán thống kê và phân tích.
- Security trong Kafka: Bao gồm xác thực (authentication), ủy quyền (authorization) và mã hóa (encryption), đảm bảo tính bảo mật của dữ liệu trong Kafka.



### 3. Những tính năng chính của Kafka

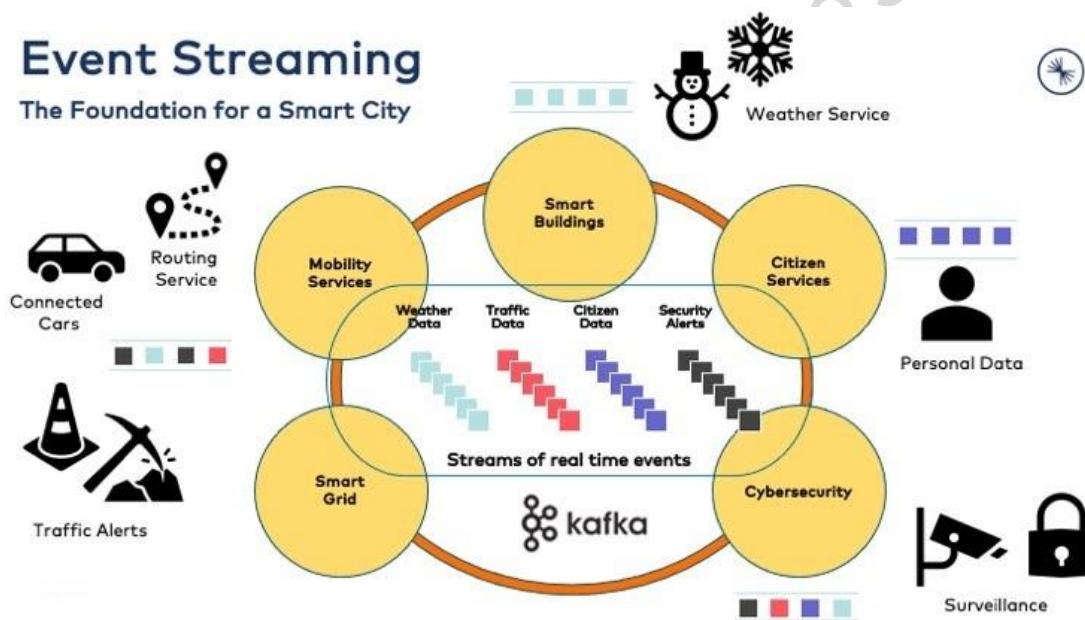
#### 3.1 Sử dụng Kafka để phân phối các message

Phân phối các message được thể hiện qua sơ đồ như sau:



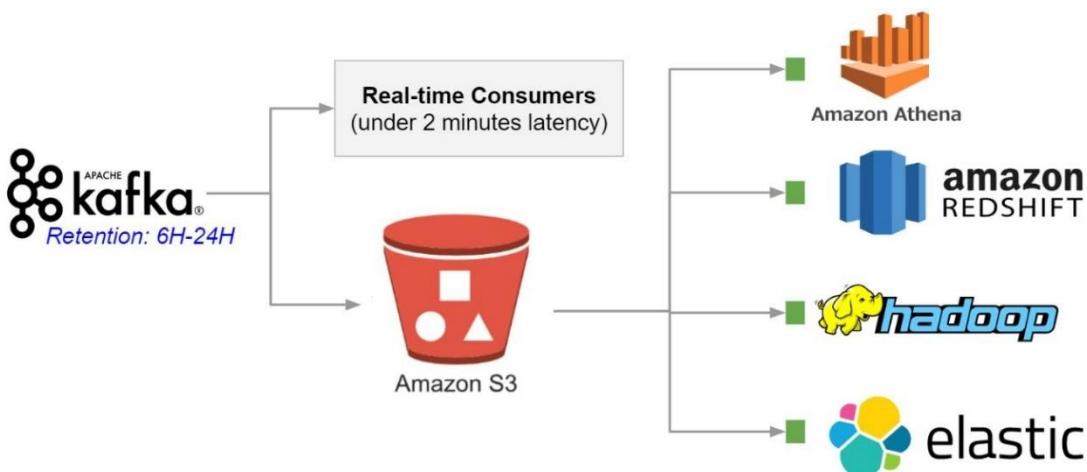
#### 3.2 Event Streaming Kafka

Event Streaming là tính năng chủ yếu của Kafka, là hành động thu thập dữ liệu dưới dạng các luồng sự kiện (event streams) từ các nguồn dữ liệu như cơ sở dữ liệu, cảm biến, thiết bị di động và lưu trữ chúng lâu dài. Nó cho phép Kafka thực hiện các công việc như truy xuất dữ liệu sau này, phân tích, xử lý các luồng sự kiện theo thời gian thực và định tuyến chúng đến các công nghệ đích khác nhau khi cần thiết.



### 3.3 Lưu trữ dữ liệu trên hệ thống Kafka

Kafka còn cung cấp khả năng lưu trữ lượng lớn thông tin dữ liệu, để tạo thành các kho dữ liệu (data lake). Điều này có nghĩa là Kafka có khả năng thu thập, xử lý và lưu trữ luồng dữ liệu thời gian thực cùng với việc lưu trữ dữ liệu theo phong cách chủ động.



Dữ liệu được Kafka lưu trữ có thể được sử dụng để xây dựng các công nghệ tiên tiến như Machine Learning, hay Trí tuệ nhân tạo (AI). Bằng cách sử dụng Kafka làm nền tảng lưu trữ dữ liệu, các nhà phát triển có thể xây dựng các ứng dụng phức tạp và có khả năng dự đoán thông tin quan trọng từ dữ liệu của họ.

### 4. Cơ chế hoạt động của Kafka

Cơ chế hoạt động của Apache Kafka khác biệt so với các message queue truyền thống như RabbitMQ ở một số điểm quan trọng:

	VIETTEL AI RACE	TD181
	TÌM HIỂU VỀ KAFKA	Lần ban hành: 1

#### 4.1 Hệ thống phân tán và mở rộng:

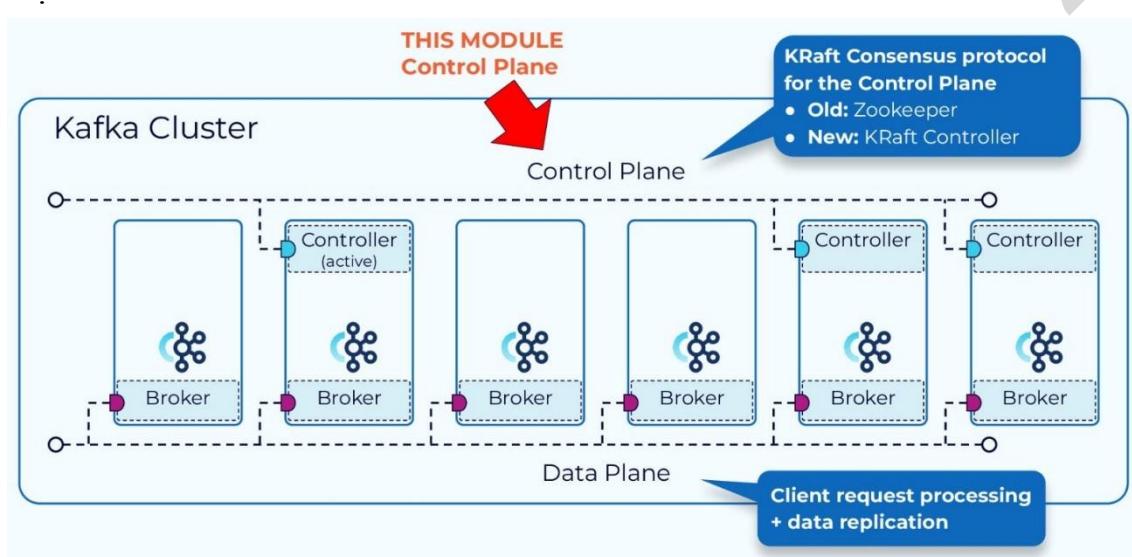
Kafka được thiết kế để hoạt động như một hệ thống phân tán hiện đại, chạy dưới dạng một cụm và có khả năng mở rộng quy mô một cách linh hoạt để xử lý bất kỳ số lượng ứng dụng nào. Nó cho phép Kafka xử lý lượng dữ liệu lớn, đáp ứng được nhu cầu mở rộng của các hệ thống.

#### 4.2 Hệ thống lưu trữ dữ liệu:

Khác với các message queue truyền thống, Kafka không chỉ phục vụ như một hàng đợi message, mà còn được thiết kế để hoạt động như một hệ thống lưu trữ dữ liệu. Kafka cho phép lưu trữ dữ liệu trong thời gian cần thiết và không xóa message ngay sau khi consumer xác nhận đã nhận, cho phép các ứng dụng có thể truy xuất lại dữ liệu theo nhu cầu.

#### 4.3 Xử lý stream processing:

Một trong những điểm mạnh của Kafka là khả năng xử lý stream processing, tức là Kafka có thể tính toán các luồng dẫn xuất và các dataset một cách linh hoạt, cho phép các ứng dụng thực hiện những tính toán phức tạp trên dữ liệu trong thời gian thực.



### 5. Ưu và nhược điểm của Kafka

#### 5.1 Ưu điểm

Apache Kafka có nhiều ưu điểm nổi bật như sau:

	VIETTEL AI RACE	TD181
	TÌM HIỂU VỀ KAFKA	Lần ban hành: 1

- 5.1.1 **Mã nguồn mở (Open-source):** Apache Kafka là một dự án mã nguồn mở, có nghĩa là mọi người đều có thể truy cập mã nguồn, sửa đổi và phát triển theo nhu cầu của họ mà không cần phải lo lắng về các hạn chế cấp phép.
- 5.1.2 **Xử lý dữ liệu lớn (High-throughput):** Kafka có khả năng xử lý một lượng lớn thông tin một cách liên tục mà gần như không cần thời gian chờ, giúp các ứng dụng có thể xử lý dữ liệu với tốc độ cao và đáp ứng được yêu cầu về hiệu suất.
- 5.1.3 **Tần suất cao (High-frequency):** Kafka có thể xử lý cùng lúc nhiều message và nhiều thể loại topic mà không gặp phải các vấn đề về tốc độ xử lý, hỗ trợ việc truyền tải dữ liệu diễn ra hiệu quả.
- 5.1.4 **Khả năng mở rộng (Scalability):** Kafka dễ dàng mở rộng khi có nhu cầu, cho phép hệ thống mở rộng từ một cụm nhỏ đến một cụm lớn hơn.
- 5.1.5 **Tự động lưu trữ message:** Kafka tự động lưu trữ message, để người dùng dễ dàng kiểm tra lại dữ liệu đã được truyền tải hay xử lý.
- 5.1.6 **Cộng đồng hỗ trợ đông đảo:** Kafka có một cộng đồng người dùng đông đảo, cung cấp hỗ trợ nhanh chóng cũng như giải đáp các vấn đề khi cần thiết, giúp người dùng tiếp cận được thông tin, giải pháp dễ dàng.



## 5.2 Nhược điểm

Mặc dù Apache Kafka mang lại nhiều lợi ích cho việc xử lý dữ liệu phân tán và thời gian thực, nhưng nền tảng này cũng có những nhược điểm nhất định:

	VIETTEL AI RACE	TD181
	TÌM HIỂU VỀ KAFKA	Lần ban hành: 1

**5.2.1 Thiếu công cụ giám sát hoàn chỉnh:** Kafka không cung cấp một công cụ giám sát tích hợp hoàn chỉnh, những nhà phát triển có thể phải sử dụng nhiều công cụ khác nhau để quản lý và giám sát hệ thống Kafka, gây ra sự phân tán và khó khăn khi quản lý.

**5.2.2 Không hỗ trợ chọn topic dựa trên wildcard:** Kafka không hỗ trợ chọn topic dựa trên wildcard, có nghĩa là người dùng phải chính xác chỉ định tên topic mà họ muốn xử lý message.

**5.2.3 Hiệu suất giảm khi tăng khối lượng dữ liệu:** Khi khối lượng và kích thước của các message tăng lên, Kafka cần phải nén và giải nén các message này, dẫn đến giảm hiệu suất tổng thể của hệ thống do tốn thêm tài nguyên và thời gian xử lý.

**5.2.4 Xử lý không linh hoạt:** Trong một số trường hợp, khi số lượng queues trong một Cluster Kafka tăng lên, hệ thống có thể trở nên chậm lại và ít nhạy bén hơn trong việc xử lý các message, gây ra gián đoạn và giảm hiệu suất chung.



Mặc dù còn tồn đọng những hạn chế nhưng Kafka vẫn là một lựa chọn hợp lý để xử lý dữ liệu phân tán và thời gian thực nhờ vào các ưu điểm về hiệu suất, tính linh hoạt trong mở rộng hay cộng đồng người dùng đông đảo của nó.