

# Week4\_ApacheKafka

---

## What are events

Events describe an entity's observable state updates over time

e.g :

- GPS coordinates of a car
- Temperature of a room
- Blood pressure of a patient
- RAM usage of an application

Types of events:

- Primitive:

Hello

- A key-value pair

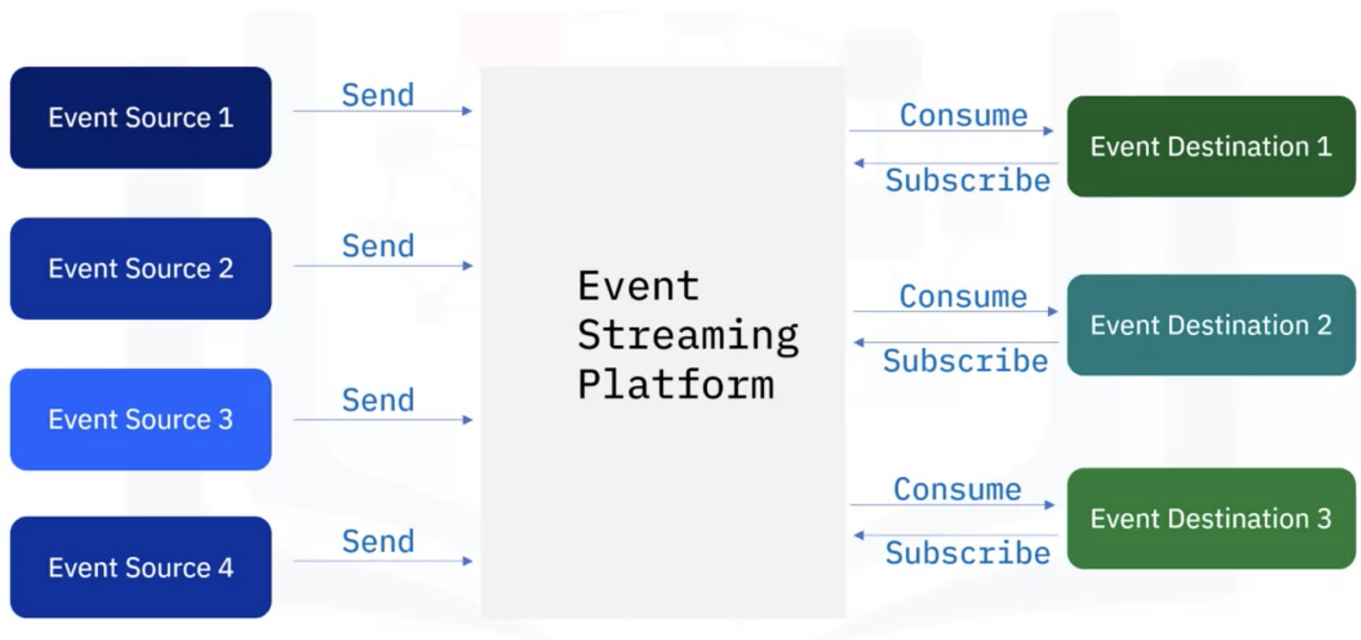
```
key: "car_id_1"  
Value: (43.82, -79.48)
```

- A key-value with a timestamp

```
Key: "patient_id"  
Value: (125,85)  
Timestamp: 2021-07-01 12:00
```

## Event Streaming Platform (ESP)

Acts as a middle layer among various event sources and destination, and provide a unified interface for handling event based ETL



## Common components of an ESP

- Event Broker: designed to receive and consume events. It is the core component of an ESP

`*Event Sources* -> Ingestor -> Processor -> Consumption -> *Event Destination*`

Ingestor efficiently receive events from various event sources

Processor applies data transformation on events such as serializing and deserializing, compressing and decompressing, encryption and decryption

Consumption retrieves events from event store and distributes them to subscribed event destinations

- Event Storage: used for storing events being received from event sources
- Analytic and Query Engine" Used for querying and analyzing the stored events.

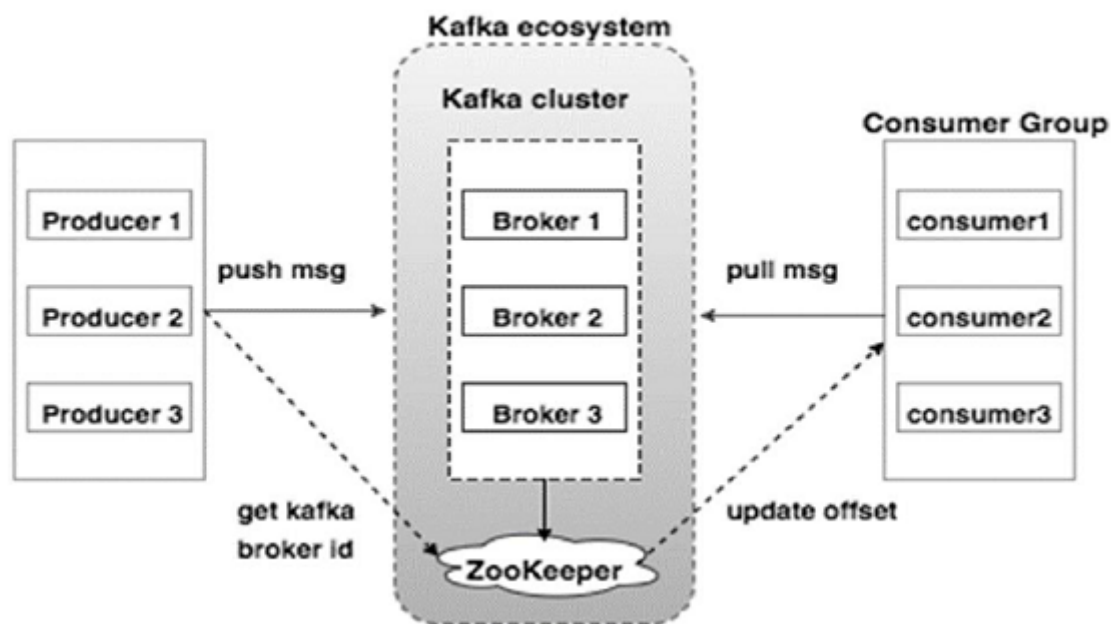
---

## Apache Kafka

Used for monitor and store `User Activities`, collect `Metrics`, store `Logs`, `Financial Transactions`, etc



## Kafka Architecture & features



- Distribution System
- Highly Scalable
- Highly reliable
- Permanent persistency
- Open source

### Broker and topic

Brokers are dedicated servers to receive, store, process and distribute events, and are synced and managed by ZooKeeper.

Each broker contains one or more topics.

Topics are databases that stores specific kind of events, such as logs, transactions and metrics.

Topics can partition and replicate topics among different brokers to increase fault tolerance.

## Kafka topic CLI

- Create a topic:

```
kafka-topics --bootstrap-server localhost:9092 --topic log_topic --create --partitions 2 --replication-factor 2
```

- List topics:

```
kafka-topics --bootstrap-server localhost:9092 --list
```

- Get topic details:

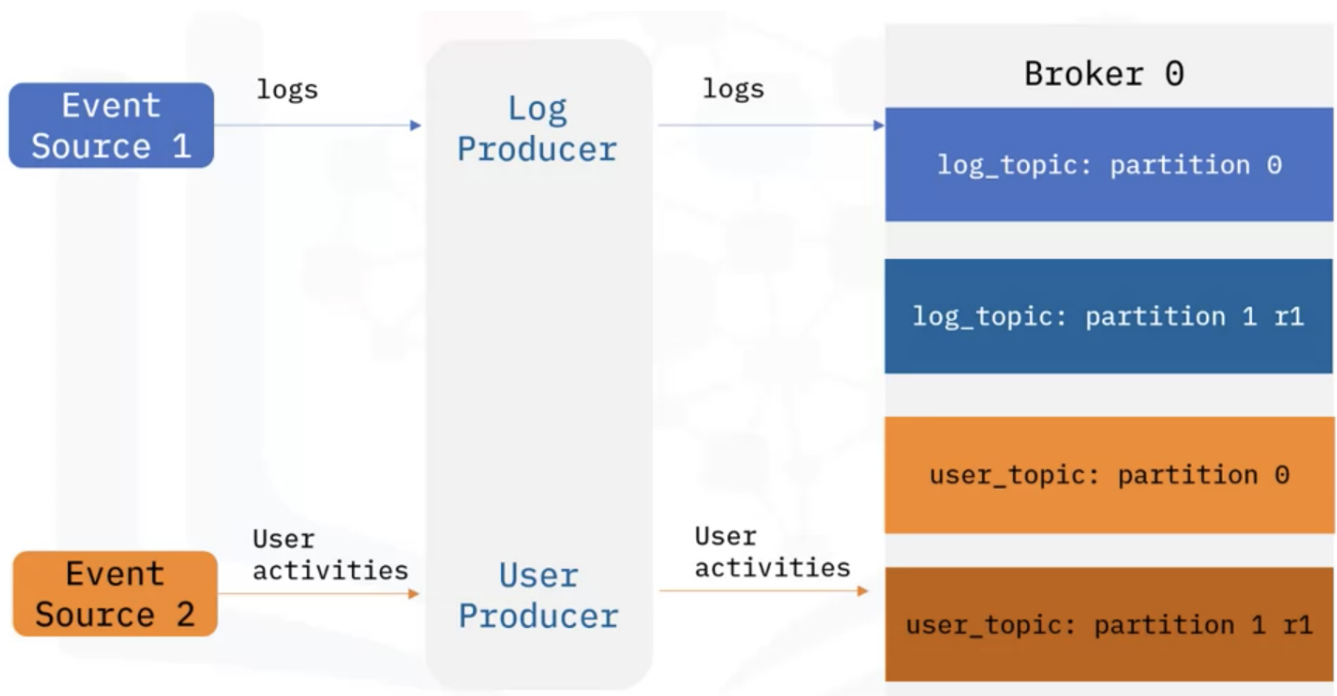
```
kafka-topics --bootstrap-server localhost:9092 --describe log_topic
```

- Delete a topic:

```
kafka-topics --bootstrap-server localhost:9092 --topic log_topic --delete
```

## Kafka Producer

- Client applications that publish events to topic partition
- An event can be optionally associated with a key
- Events associated with the same key will be published to the same topic partition
- Events not associated with any key will be published to topic partitions in rotation



- Start a producer to a topic, without keys:

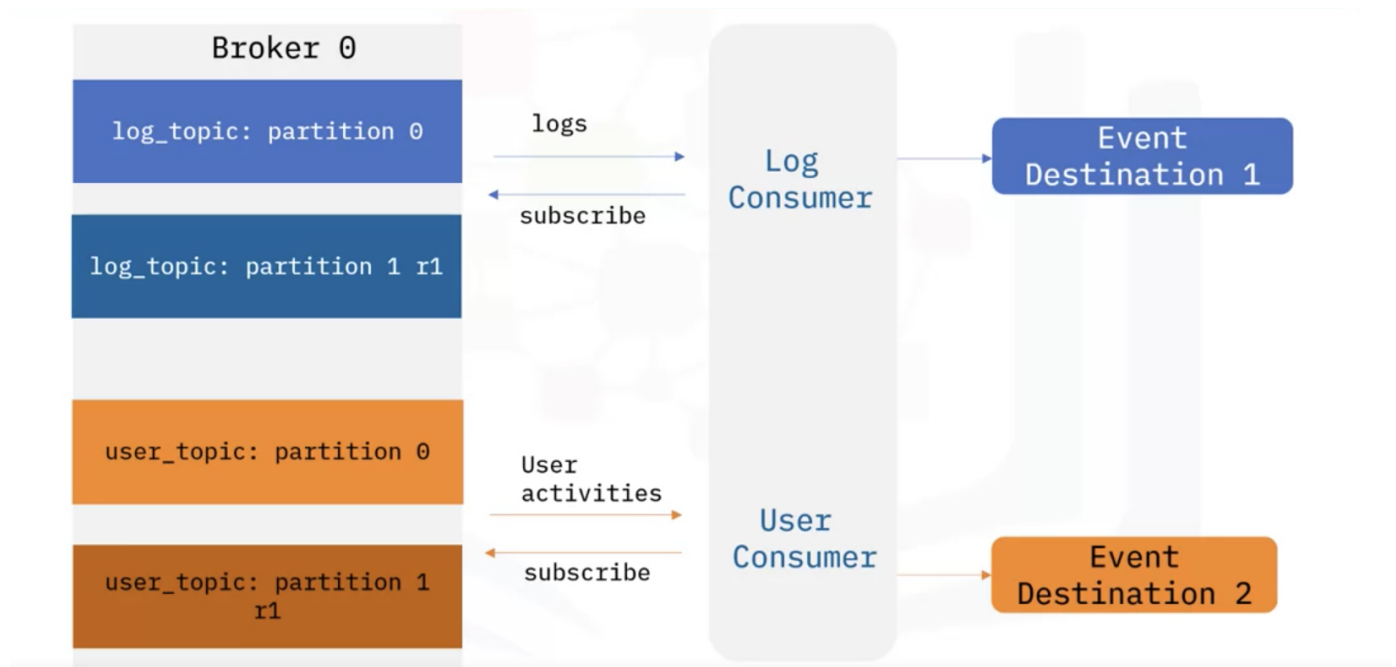
```
kafka-console-producer --broker-list localhost:9092 --topic log_topic
```

- Start a producer to a topic, with keys:

```
kafka-console-producer --broker-list localhost:9092 --topic user_topic --  
property parse.key=true -property key.separator=,
```

## Kafka consumer

- Consumers are clients subscribed to topics
- Consume data in the same order
- Store an offset record for each partition
- Offset can be reset to zero to read all events from the beginning again



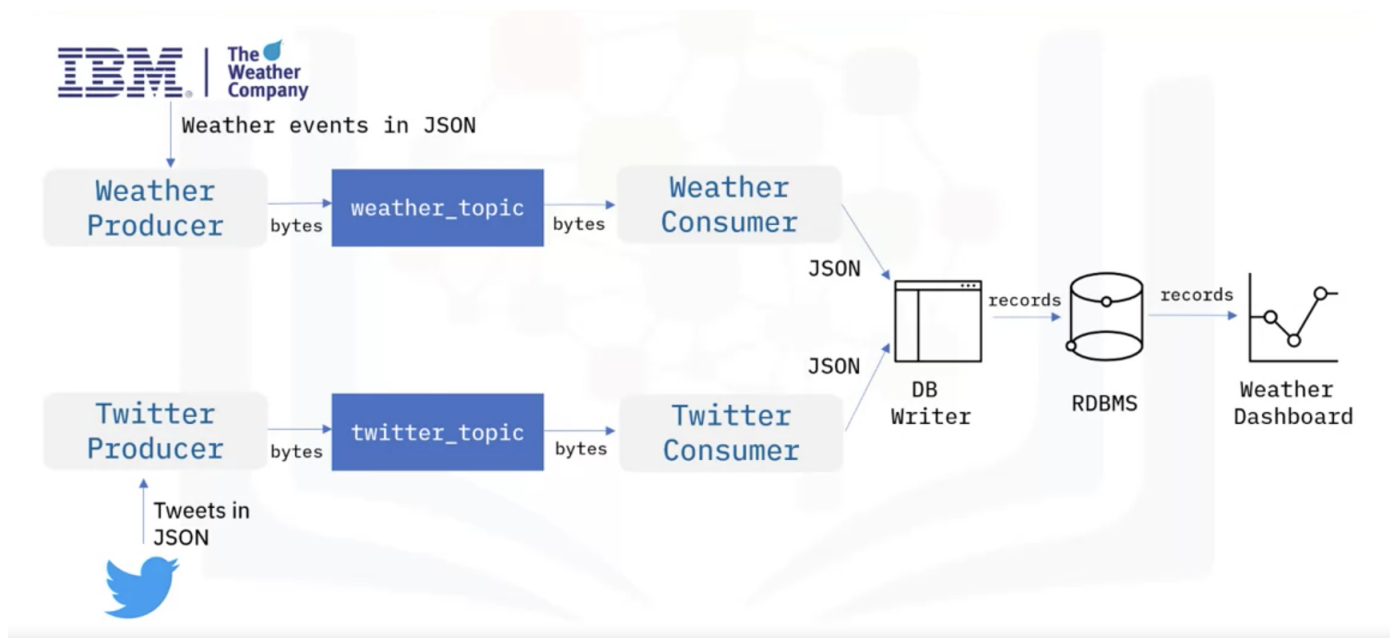
- Read new events from the log\_topic, offset 1:

```
kafka-console-consumer --bootstrap-server localhost:9092 --topic log_topic
```

- Read all events from the log\_topic, offset 0

```
kafka-console-consumer --bootstrap-server localhost:9092 --topic log_topic --from-  
beginning
```

---



## Kafka Streams API

- Simple client library to facilitate data processing in event streaming pipelines
- Processes and analyzes data stored in kafka topics
- Record only processed once
- Processing one record at a time