

## Problem: Show data from multiple microservices in real time.

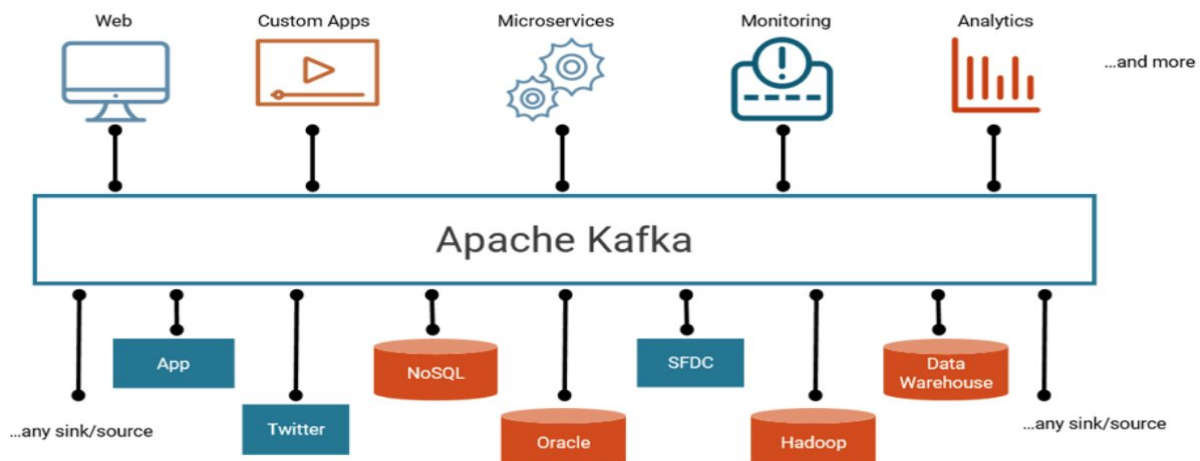
For example :

- Running status.
- Number of trigger event particular microservice.
- Crash count for microservice.
- Message Notification from app.

To handle big event driven ecosystem in real time we need streaming platform.

## What we need?

1. **Crawler:** Script or program to get all information from microservice and push to Kafka server. (Producer)
2. **Apache Kafka :** open-source stream-processing software platform
  - Kafka handle real time stream and its run on cluster.
  - It's easy to scale.
  - It has own database and can store data for 7 days by default.
  - Can be customize for more storage.
  - It has high availability. If one node crash it can create another on cluster.

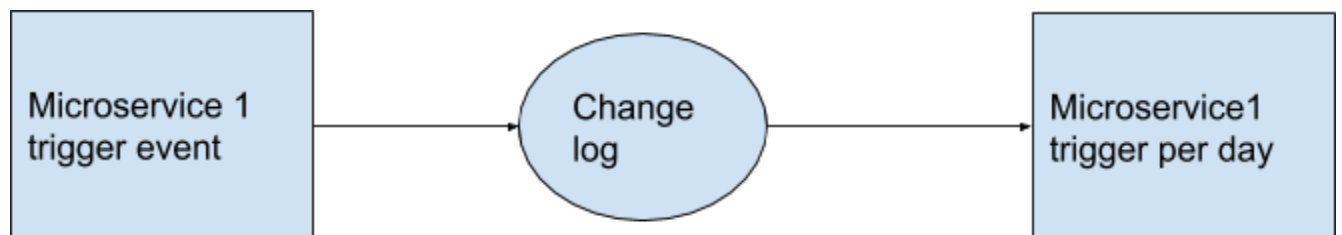


### Steps to stream data:

1. Add crawler to each microservice.
2. Create producer for each microservice to push data to kafka with particular **Topic** name.
3. Create consumer to get data from kafka server by Topic name.
4. Connect consumer with your app, dashboard or any tool to see data.

### Streaming example:

**How many time microservice trigger per day?**



This is contiguous process. Whenever microservice get trigger, it send event to kafka server.

**Use case :** If admin want to know how many time particular microservice trigger in one day ?

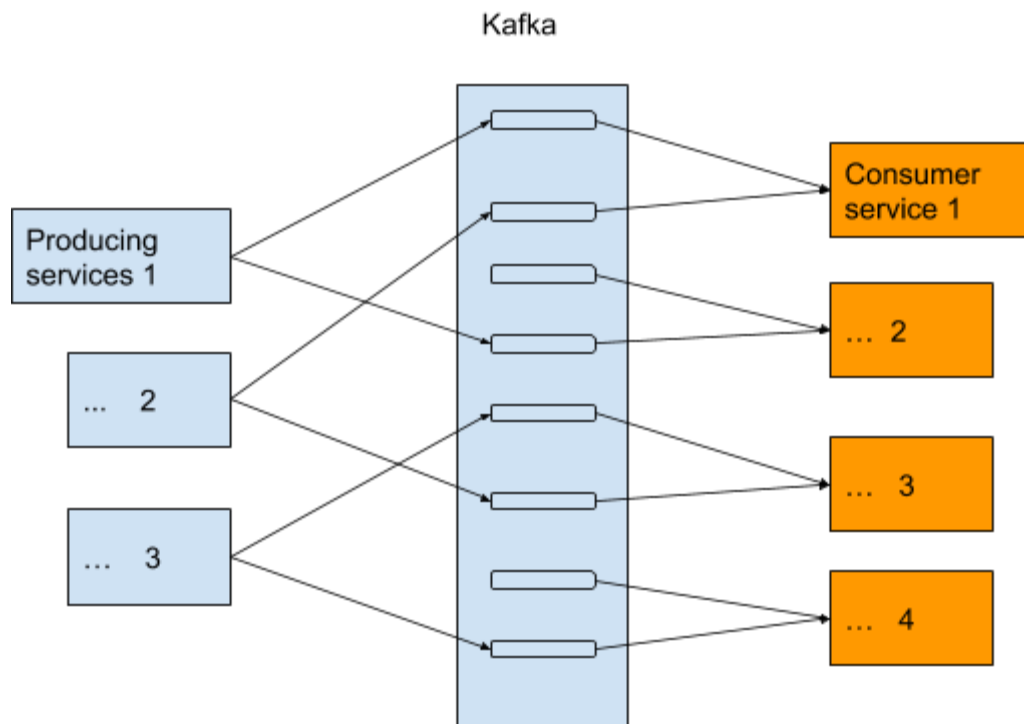
#### Example in KSQL query :

```
CREATE TABLE TRIGGER_PER_DAY AS  
SELECT SERVICE_ID, COUNT(*) FROM SERVICE_TRIGGER  
WINDOW TUMBLING (SIZE 1 DAY)  
GROUP BY SERVICE_ID
```

It will create KTABLE (in memory) and store result to it.

Beside that microservice can send any kind of data such as When it crash. From that we can look how many time particular microservice got crashed in a day or week. We can Query how many apps are unstable.

## Many logs over many machines (scalability):



## Event Driven Architecture Overview:

Build service and make it

1. BroadCast Event out to kafka Topic.
2. Retain them in log, it gives you ability to rewind history. Build new service or reporting based on log.
3. Evolve the event stream with streaming function for any computation you want to do.
4. Recasting the event stream into views when you need to query. For that we can use any suitable database for example SQL, NOSQL or KTable stream (kafka inbuilt database).