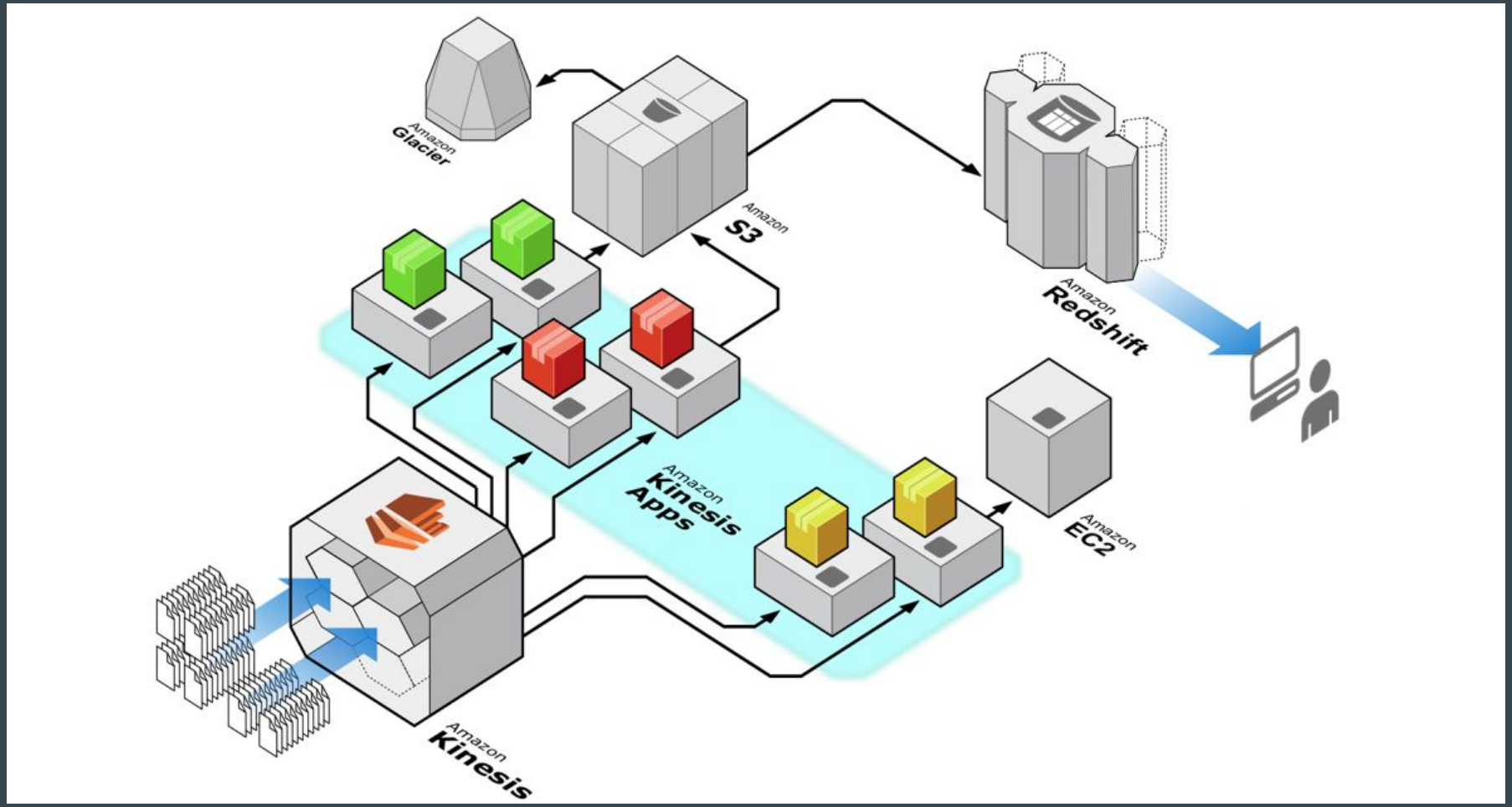
Amazon Kinesis

By Cerize Santos
May 29, 2017


Amazon
Glacier
Amazon
S3
Amazon
Redshift
Amazon
Kinesis
Apps
Amazon
EC2
Amazon
Kinesis

Key concepts

Streams

Ordered sequence of data records.

The data records in a stream are distributed into **shards**.

The total capacity of a stream is the sum of the capacities of its shards.

Producers

- puts data records into Amazon Kinesis streams
- Data sent is kept from 1 to 7 days (1 is the default)

Consumers

- Processes the data records from a stream
- Amazon Kinesis Client Library (KCL) simplify parallel processing of the stream

Key concepts

Shard

base throughput unit

capacity of 1MB/sec data input and 2MB/sec data output. One shard can support up to 1000 PUT records per second.

Specified when Stream is created, or dynamically

Charged on a per-shard basis.

Key concepts

Data Record

Unit of data

Has sequence number, partition key, and data blob

Maximum size of a data blob: 1 megabyte (MB)

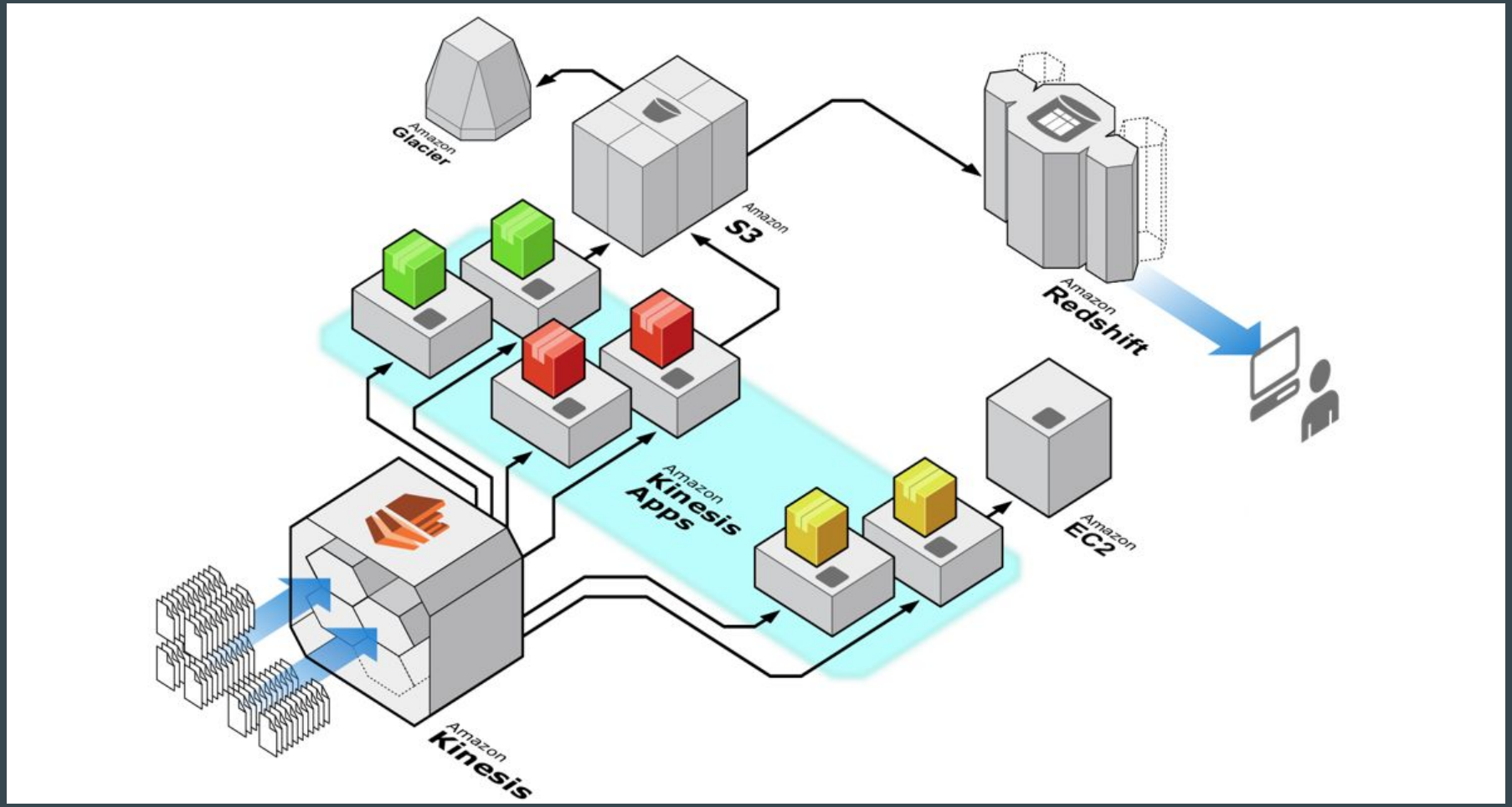
Partition Key

Routes data records to different shards

Sequence Number

Unique identifier for each data record

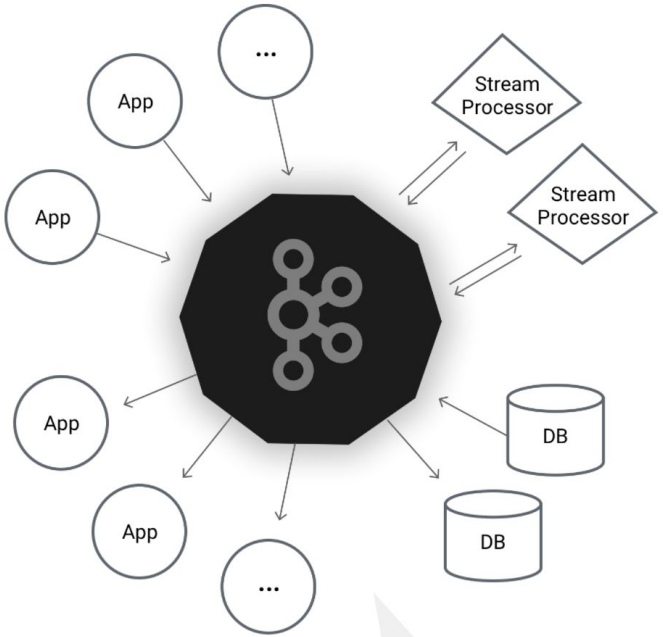
Created when producer calls PutRecord or PutRecords API


Amazon
Glacier
Amazon
S3
Amazon
Redshift
Amazon
Kinesis
Apps
Amazon
EC2
Amazon
Kinesis

Is that something new or special?

APACHE kafka™

A distributed streaming platform



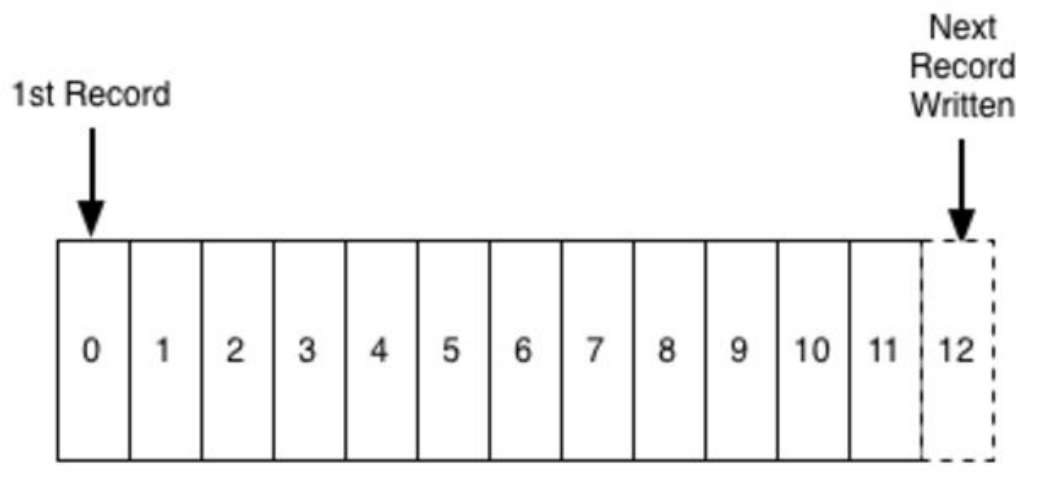
- Open source
- Developed by LinkedIn
- PUBLISH & SUBSCRIBE to streams of data **like a messaging system**
- PROCESS streams of data efficiently and in real time
- STORE streams of data safely in a distributed replicated cluster

“You can't fully understand databases, NoSQL stores, key value stores, replication, paxos, hadoop, version control, or almost any software system without understanding logs”

Jay Kreps


Logs

What is a log?



- Simplest possible storage abstraction
- Append-only
- Ordered by time
- Application logs vs “journal” / “data logs”

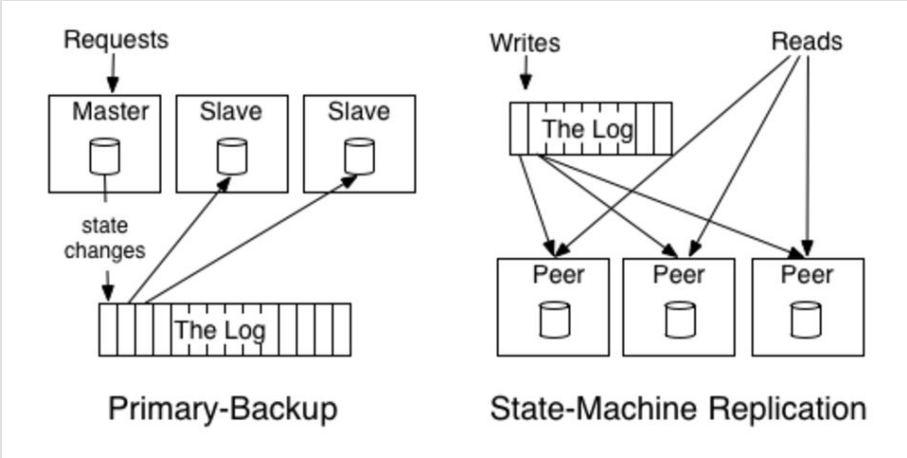
Use 1: Databases

ACID (Atomicity, Consistency, Isolation, Durability)

Log is:

- Immediately persisted;
- Used as the authoritative source in the event of a crash;
- A picture of the database at any moment

Use 2: Distributed system

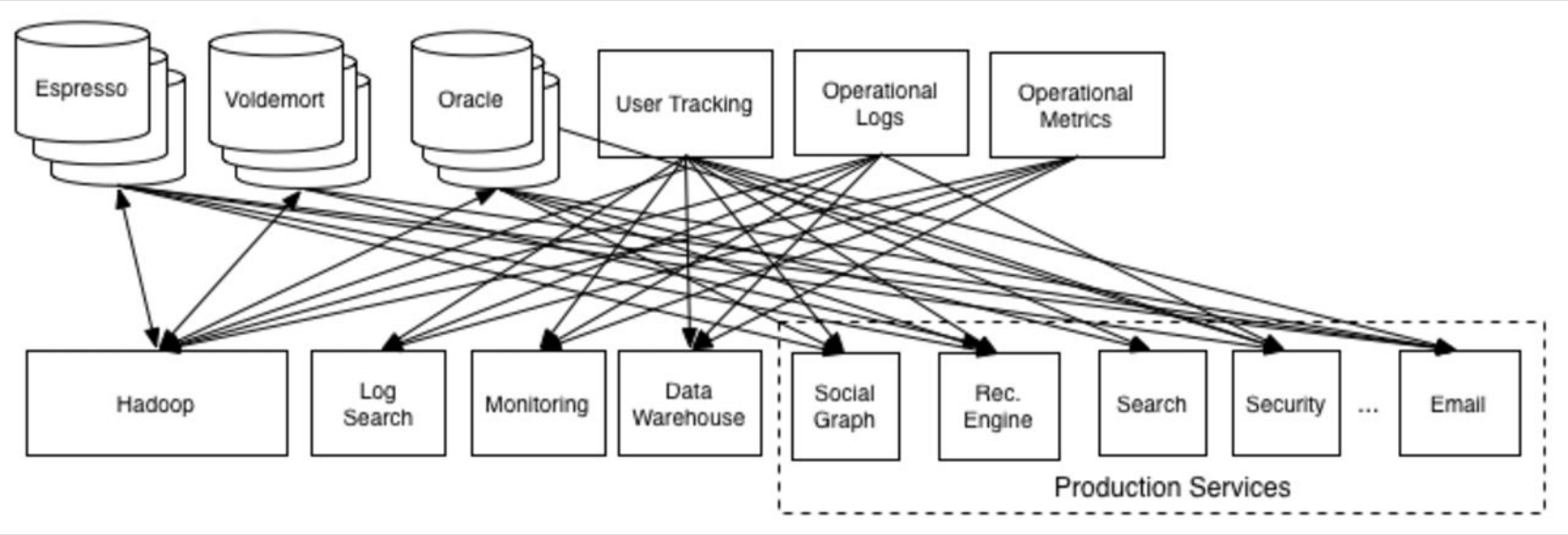


- keeping everything in sync

Use 2: Data Integration

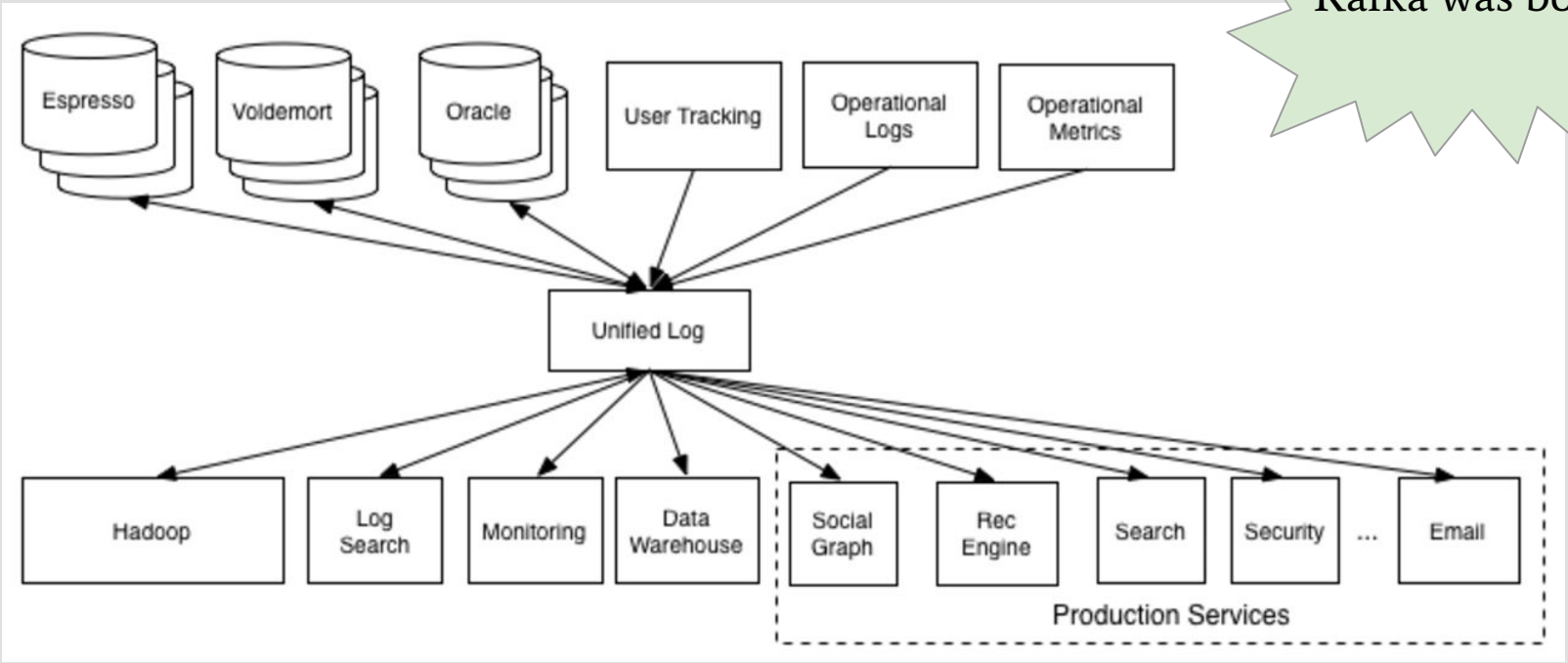
Take all the organization's data and put it into a central log for real-time subscription.

Linkedin before:



Use 2: Data Integration (cont)

Linkedin after:



Use 3: Real-time stream processing

- Stream processing is not limited to real-time processing.
- The real driver for the processing model is the method of data collection
- Includes a notion of time in the data being processed
- Does not require a static snapshot of the data to produce output
- Anything that reads from logs and writes output to logs or other systems

Use 3: Real-time stream processing

How logs help?
Or
Why not other (light-weight) messaging protocol?

- makes each data set to be multisubscriber
- order is maintained
- provide buffering

Resources

- https://kafka.apache.org/
- https://engineering.linkedin.com/distributed-systems/log-what-every-software-engineer-should-know-about-real-time-datas-unifying
- https://blog.mimacom.com/apache-kafka-with-node-js/
- http://docs.aws.amazon.com/streams/latest/dev/kinesis-sample-application.html
- https://blog.mimacom.com/apache-kafka-with-node-js/