Kafka

* Data-pipelines, the real need for messaging systems
* How due to invent of big-data, distributed computing become the need of the hour.
* Distributed messaging

What is kafka <https://kafka.apache.org/intro>

**Apache Kafka® is *a distributed streaming platform*. What exactly does that mean?**

A streaming platform has three key capabilities:

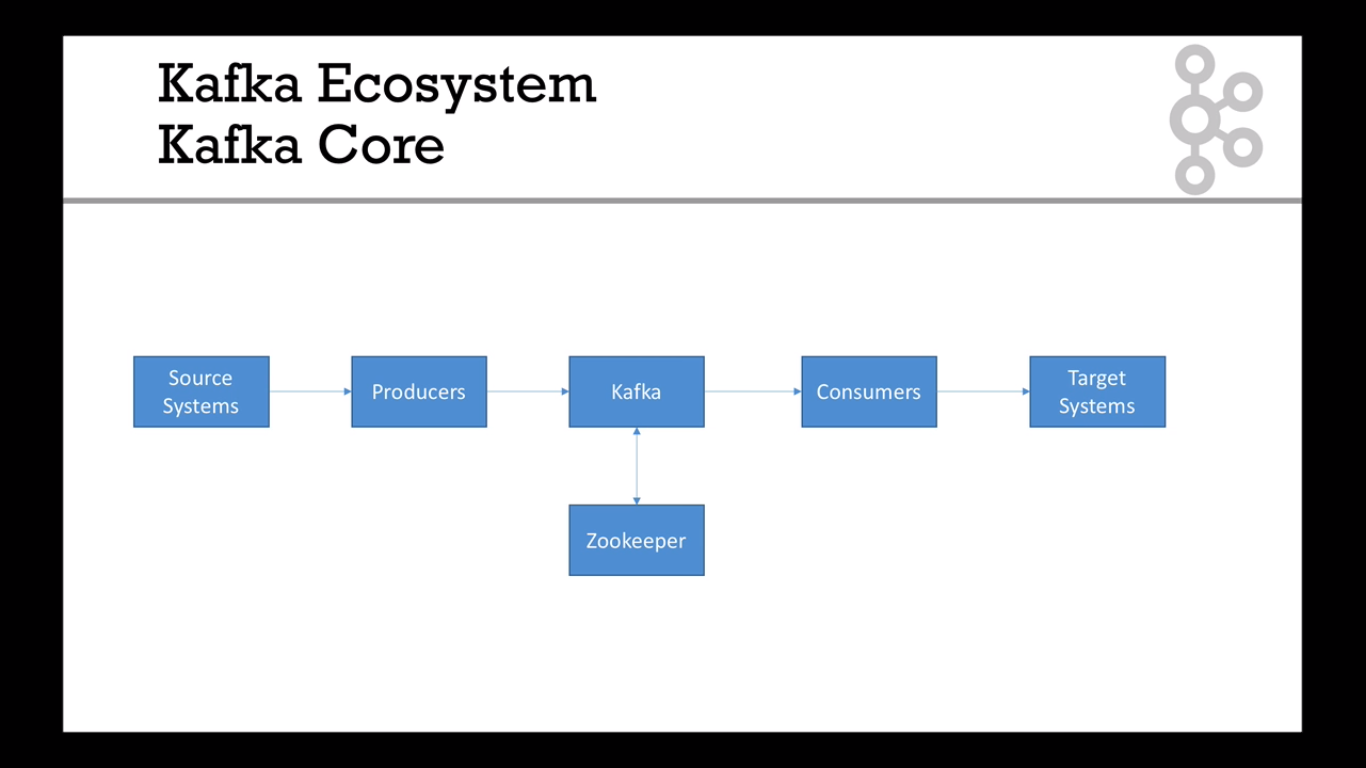
* Publish and subscribe to streams of records, similar to a message queue or enterprise messaging system.
* Store streams of records in a fault-tolerant durable way.
* Process streams of records as they occur.

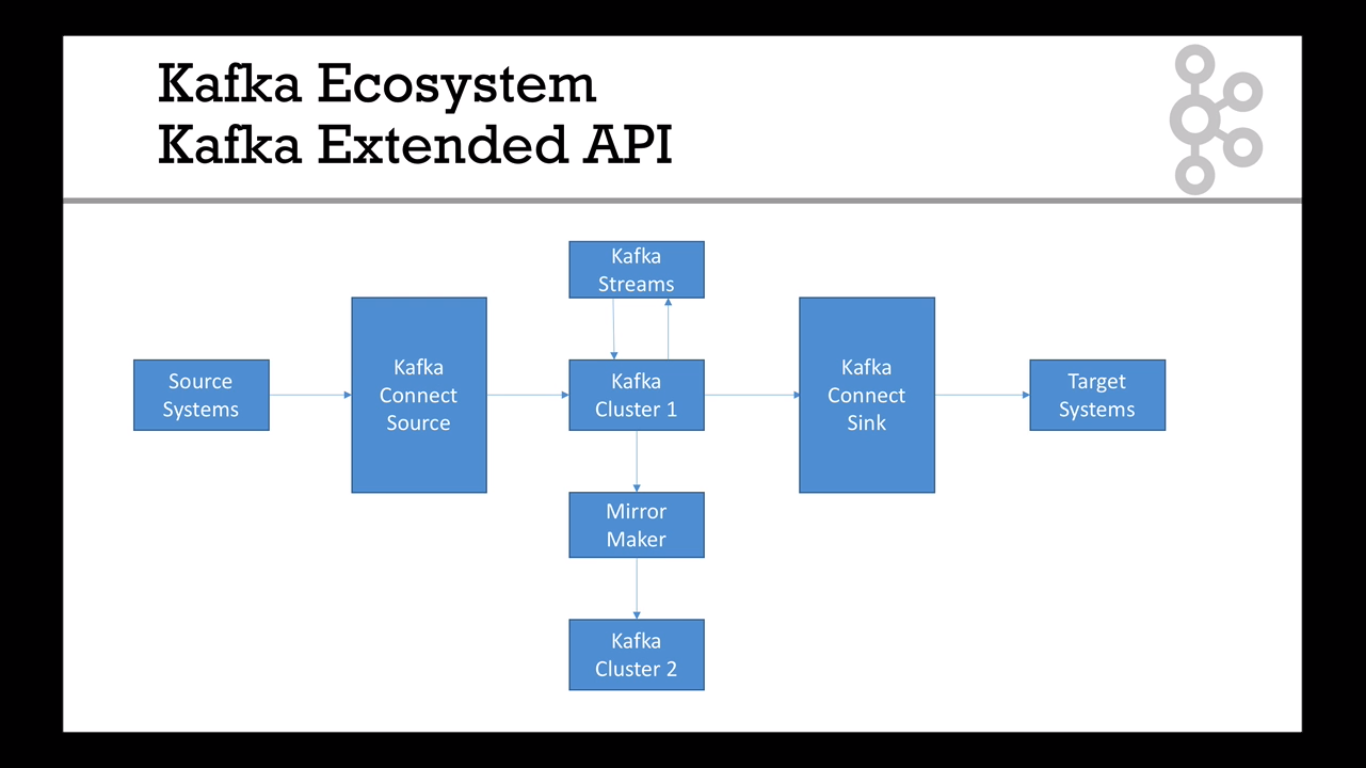
Kafka is generally used for two broad classes of applications:

* Building real-time streaming data pipelines that reliably get data between systems or applications
* Building real-time streaming applications that transform or react to the streams of data

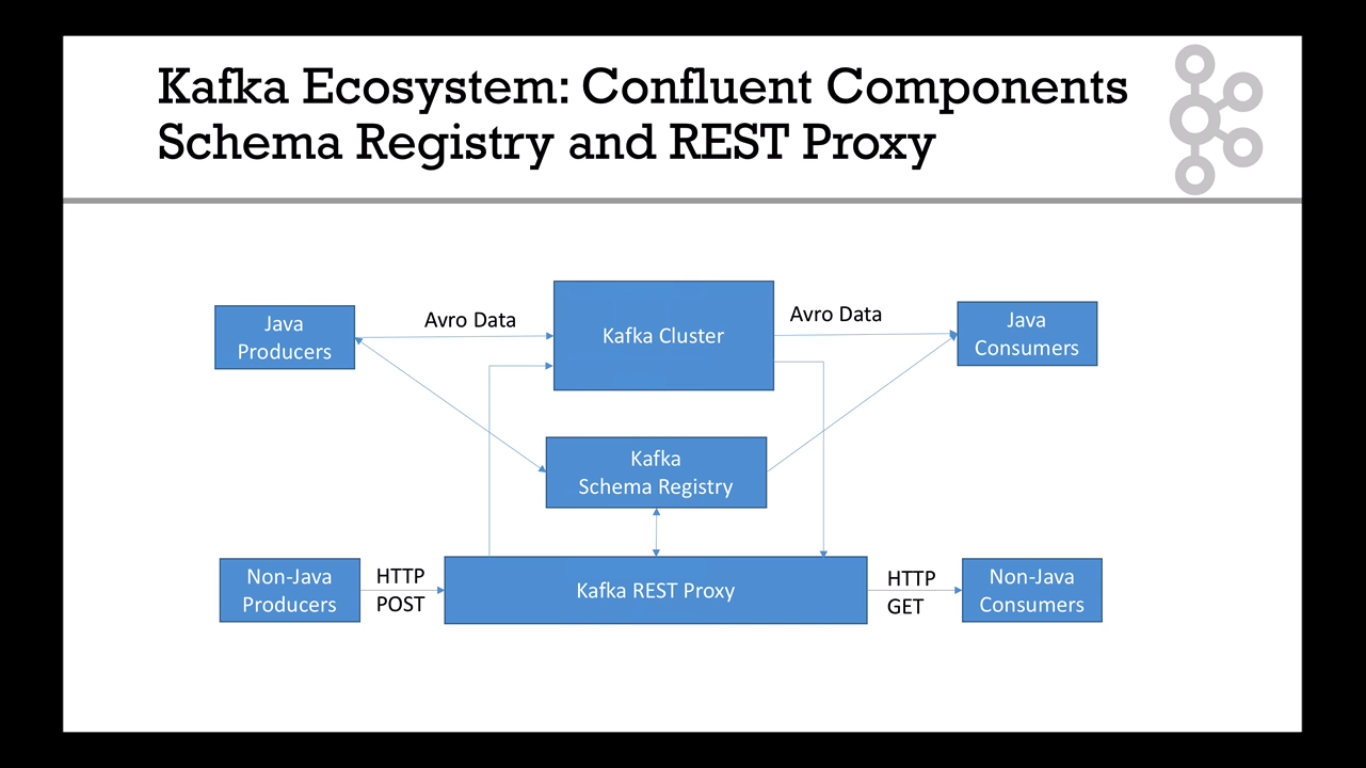
Kafka Ecosystem

* Kafka Core

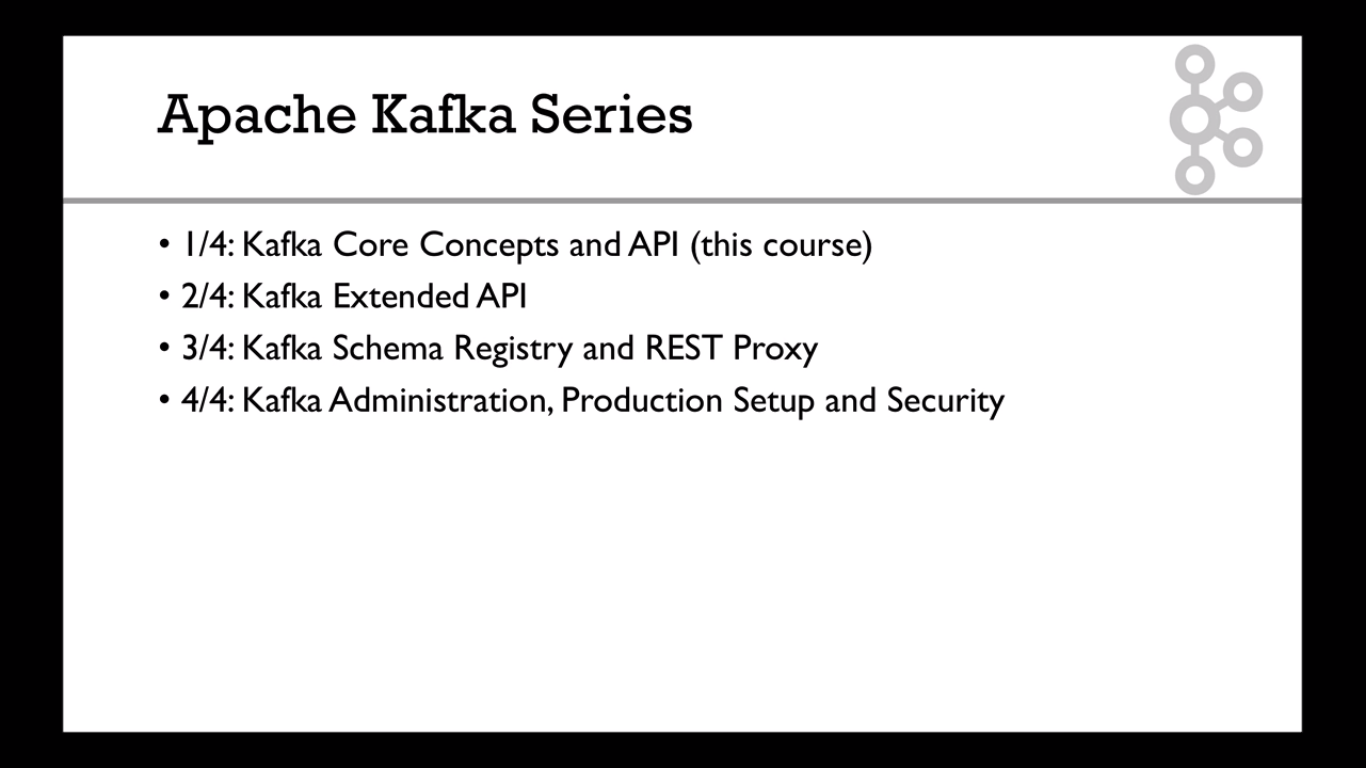


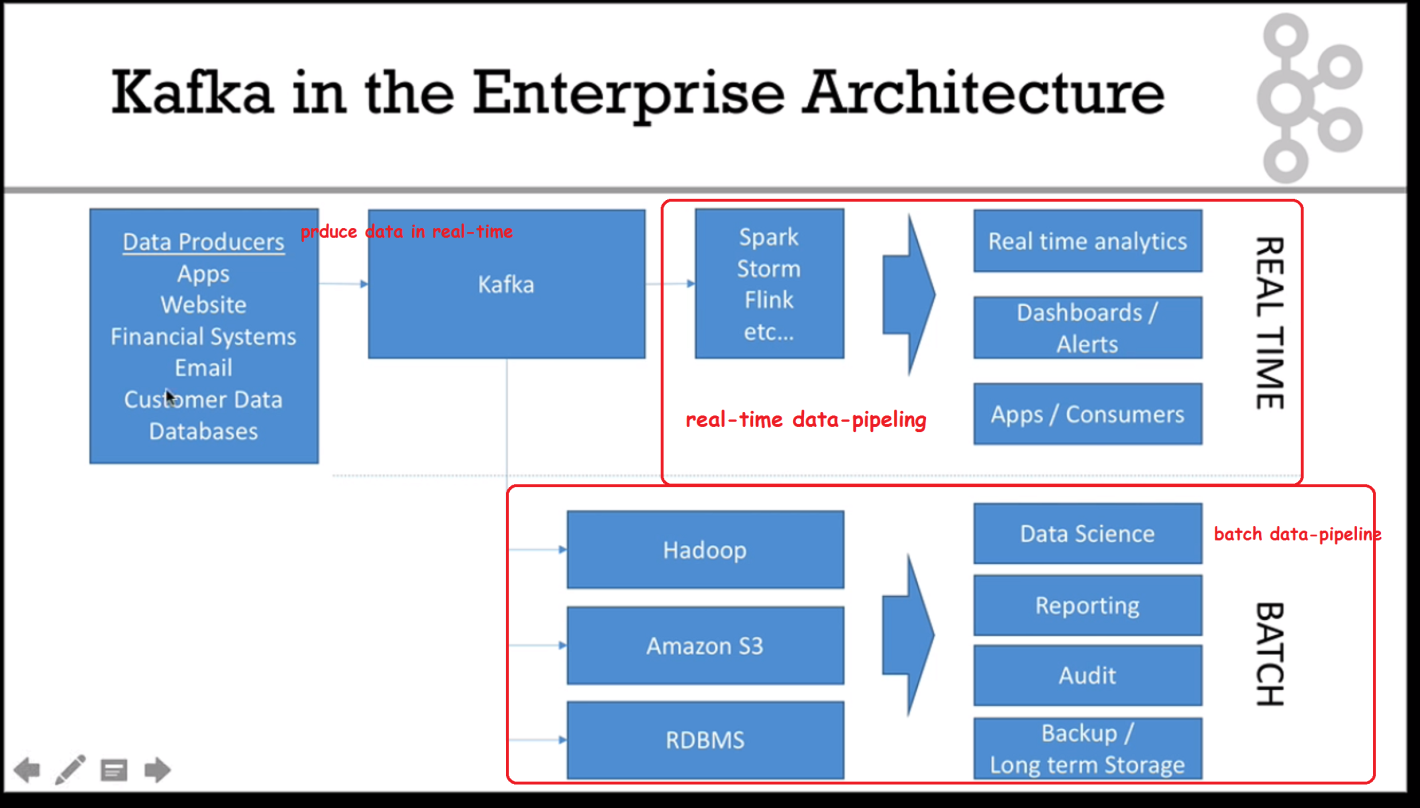
- Kafka Extended API

-Kafka confluent compoenents



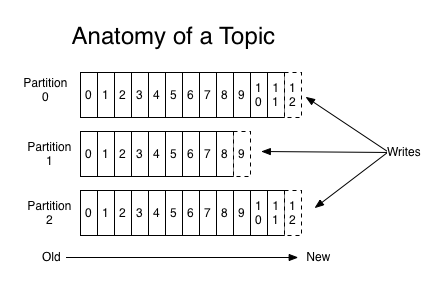
-Suggested Learning Track





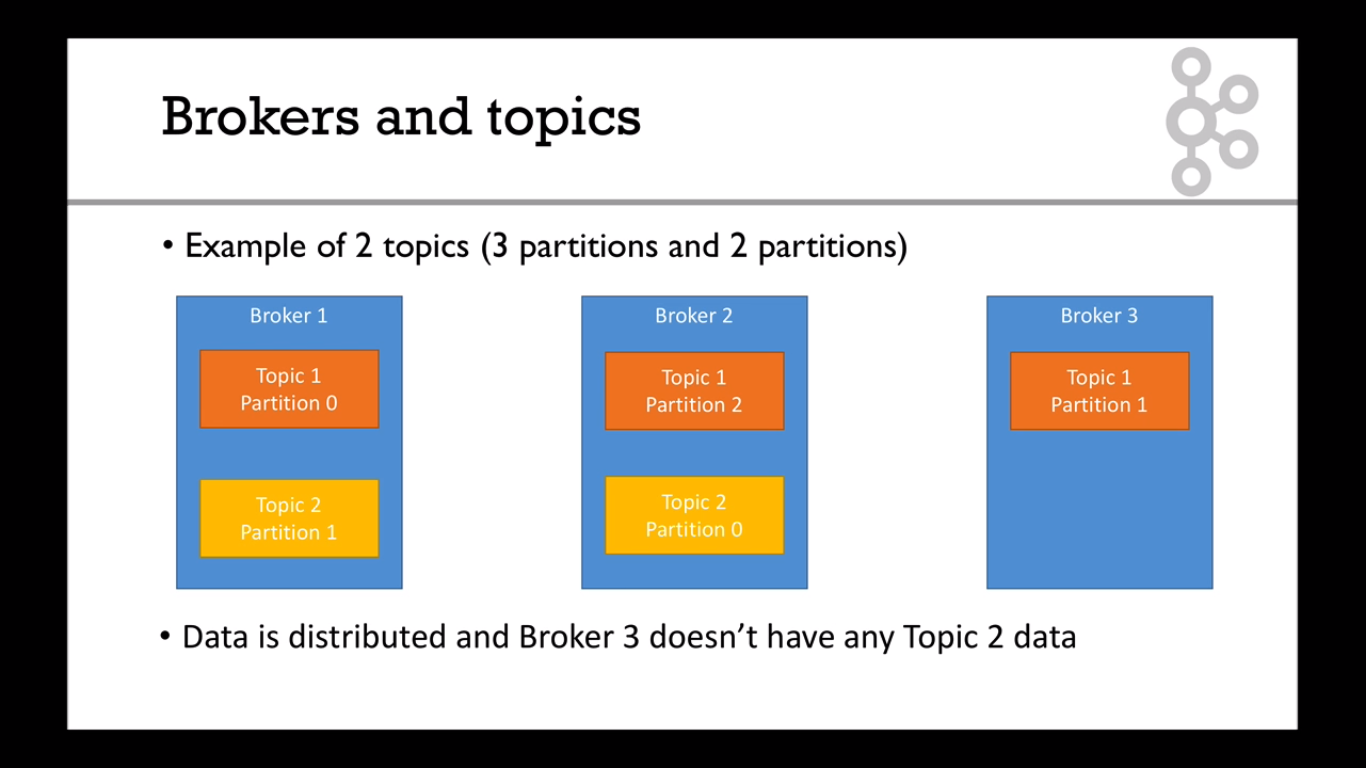
Any data-lake e.g. s3/Hadoop

* Topics & Partitions:
* Topic is basically a stream of data
* Similar to tables in dbase, you can publish data to a topic
* You can have any number of topics, each must have a unique name
* Topics are split into a partition, this is where the actual data is stored
* The data within each partition is ordered
* Inside a partition, each message will gets an incremental id, called the offset
* Topics in Kafka are always multi-subscriber; that is, a topic can have zero, one, or many consumers that subscribe to the data written to it
* You can have any number of partitions per topic, the more partitions, the more parallelism in processing



Brokers:

* Every node in the kafka cluster is a broker
* Each can be identified via a integer
* Each broker contains one partition of a topic
* After connecting to one broker, you are by default connected to all the brokers in the cluster
* The custom property replication factor is used to replicate partitions across brokers
* There is 1 and only 1 leader broker for a given partition in a cluster, only that broker is allowed to serve and receive the data for that partition,
* Other broker nodes are in-sync or replicas of the star broker (for better fault tolerance)
* Replication factor is normally between 2 and 3



* Producers
* Responsible for loading data into the kafka cluster
* They only need to specify the name of the topic,
* If a key is specified along with the data, then the data will be ordered based on the key.
* i.e. all records with the same key-id will be in the same partition
* The producer also has an option for receiving acknowledgement
* Consumer and consumer groups

PRACTICAL

Javacode: <https://haritibcoblog.com/category/apache/kafka/>