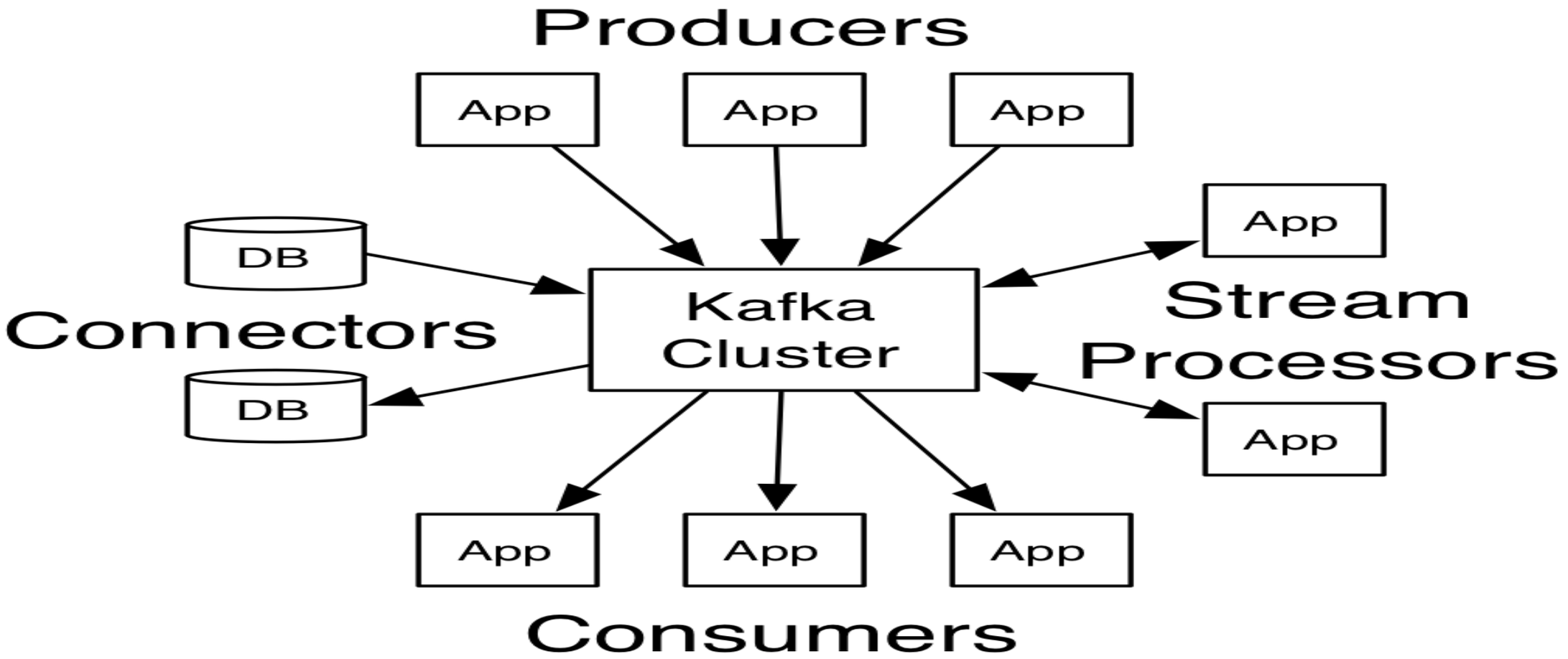
Kafka


Producers
App
App
App
DB
DB
Connectors
Kafka
Cluster
App
Stream
Processors
App
App
App
App
Consumers

Environment

- Kafka is run as a cluster on one or more servers that can span multiple datacenters.
- The Kafka cluster stores streams of *records* in categories called *topics*.
- Each record consists of a key, a value, and a timestamp.
- Kafka has four core APIs:
- The Producer API allows an application to publish a stream of records to one or more Kafka topics.
- The Consumer API allows an application to subscribe to one or more topics and process the stream of records produced to them.
- The Streams API allows an application to act as a *stream processor*, consuming an input stream from one or more topics and producing an output stream to one or more output topics, effectively transforming the input streams to output streams.
- The Connector API allows building and running reusable producers or consumers that connect Kafka topics to existing applications or data systems. For example, a connector to a relational database might capture every change to a table.

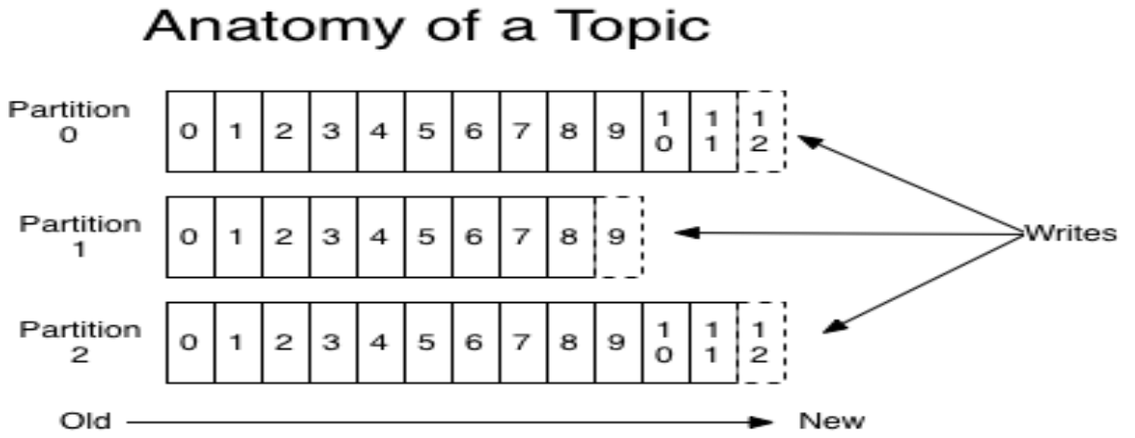
Uses

Uses of kafka

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

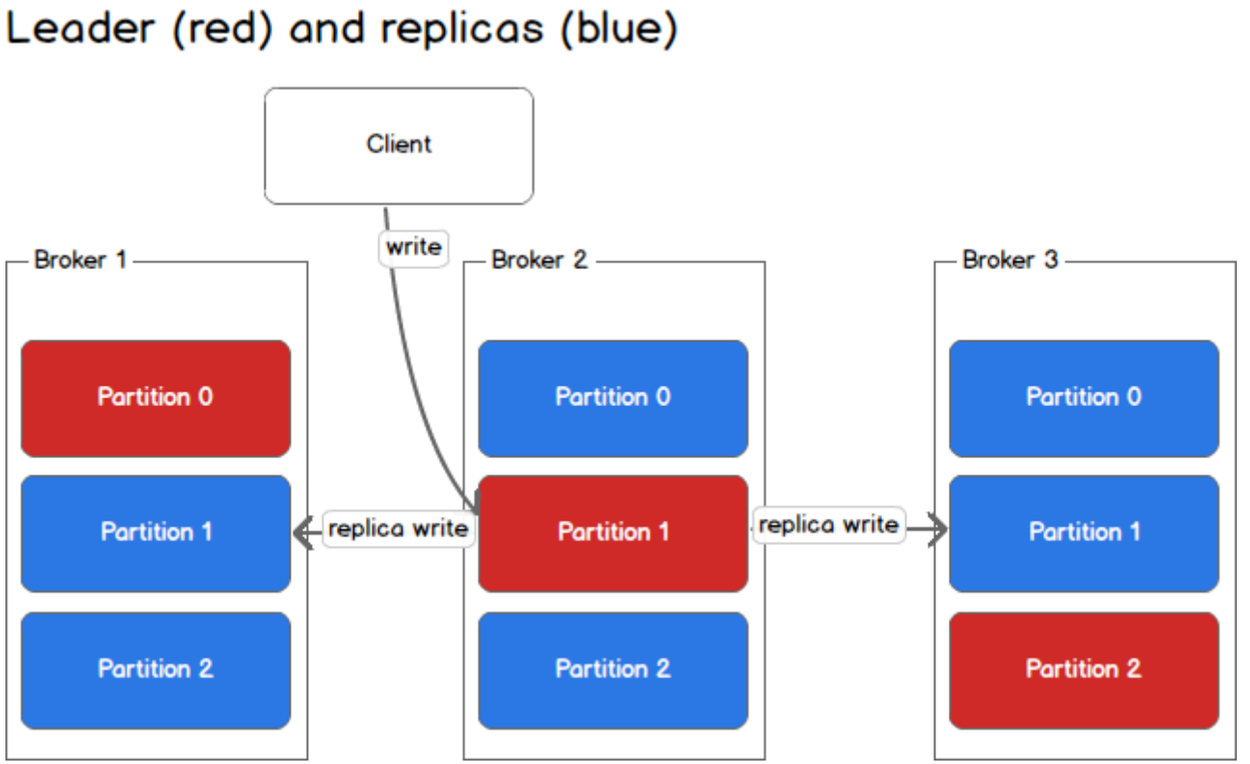
Topic and partition

- For each topic, the Kafka cluster maintains a partitioned log that looks like this:
- Kafka topics are divided into a number of *partitions*. Partitions allow you to parallelize a topic by splitting the data in a particular topic across multiple brokers—each partition can be placed on a separate machine to allow for multipl



Producer Write

- Handle replica write



Partition and Cosumer Group

- One topic—two server-- four partition—two group—six consumer
- Kafka only provides a total order over records *within* a partition, not between different partitions in a topic
- All guarantees are off if you are reading from the same partition using two consumers or writing to the same partition using two producers.