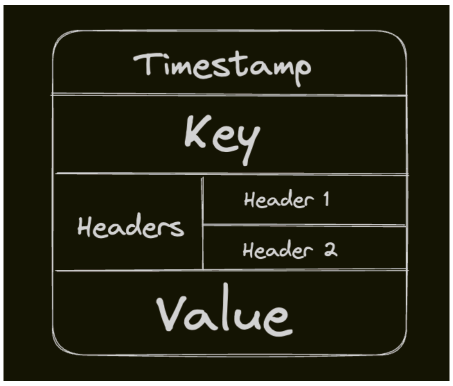
Kafka Common Terms

Before digging deep into Kafka’s architecture, lets first go through some of its common terms.

Brokers : A kafkaesque server is also called broker. Brokers are responsible for reliably storing data provided by the producers and making it available to consumers.

Records : A record is a message or an event that is stored in kafka. It contains a key, a value, a timestamp, and optional metadata headers.



Kafka Message

Topics : Kafka divides its messages into categories called Topics. In simple terms, a topic is like a table in a database and messages are the rows in that table.

- Each message that kafka receives from producer is associated with a topic.
- Consumer can subscribe to a topic to get notified when new messages are added to that topic.
- A topic can have multiple subscribers that read messages from it.
- In a Kafka Cluster, a topic is identified by its name and must be unique.

Messages in a topic can be read as often as needed - unlikely its not deleted as in traditional messaging system. Instead Kafka retains message for a configurable amount of time or untill a storage size is exceeded. Kafka’s performance is effectively constant with respect to data size, so storing data for a long time is perfectly fine.

Producers : Producers are application that publish or write records to kafka.

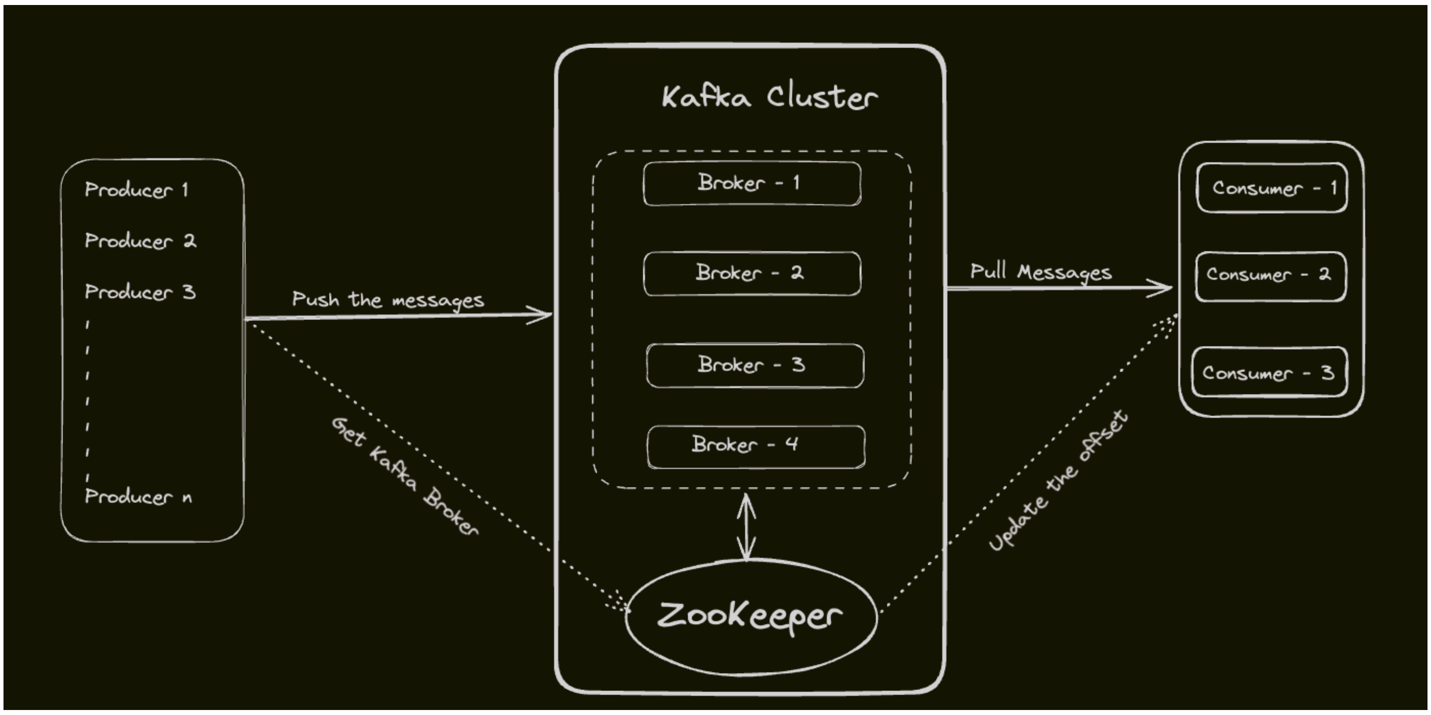
Consumers : Consumers are the applications that subscribe to data from kafkaesque topics. They subscribe to one or more topics and consume published messages by pulling data from the brokers.

In Kafka Producers and Consumers are totally decoupled and agnostic of each other other, which is a key design element to achieve high scalability that kafkaesque is known for.

High-Level Architecture : At High level, applications sends message to a Kafka Broker, and the messages are ready by other applications called Consumers. Message get stored in a topic, and consumers subscribe to the topic to receive new messages.

Kafka Cluster : Kafka is deployed as a cluster of one or more servers, where each server is responsible for running one kafka broker.

Zookeeper : Zookeeper is a distributed system that stores key-value and is used for co-ordination & storing configurations. It is highly optimised for reads. Kafka use zookeeper to co-ordination between Kafka brokers. Zookeeper maintains metadata information about the kafkaesque cluster



High Level Architecture of Kafka