Kafka Workflow

Let's explore different messaging workflows that Kafka offers.

Kafka provides both pub-sub and queue-based messaging systems in a fast, reliable, persisted, fault-tolerance, and zero downtime manner. In both cases, producers simply send the message to a topic, and consumers can choose any one type of messaging system depending on their need. Let us follow the steps in the next section, to understand how the consumer can choose the messaging system of their choice.

Kafka workflow as pub-sub messaging#

Following is the stepwise workflow of the Pub-Sub Messaging:

- Producers publish messages on a topic.
- Kafka broker stores messages in the partitions configured for that particular topic. If the producer did not specify the partition in which the message should be stored, the broker ensures that the messages are equally shared between partitions. If the producer sends two messages and there are two partitions, Kafka will store one message in the first partition and the second message in the second partition.

- Consumer subscribes to a specific topic.
- Once the consumer subscribes to a topic, Kafka will provide the current offset of the topic to the consumer and also saves that offset in the ZooKeeper.
- Consumer will request Kafka at regular intervals for new messages.
- Once Kafka receives the messages from producers, it forwards these messages to the consumer.
- Consumer will receive the message and process it.
- Once the messages are processed, the consumer will send an acknowledgment to the Kafka broker.
- Upon receiving the acknowledgment, Kafka increments the offset and updates it in the ZooKeeper. Since offsets are maintained in the ZooKeeper, the consumer can read the next message correctly, even during broker outages.
- The above flow will repeat until the consumer stops sending the request.
- Consumers can rewind/skip to the desired offset of a topic at any time and read all the subsequent messages.

Kafka workflow for consumer group#

Instead of a single consumer, a group of consumers from one consumer group subscribes to a topic, and the messages are shared among them. Let us check the workflow of this system:

- Producers publish messages on a topic.
- Kafka stores all messages in the partitions configured for that particular topic, similar to the earlier scenario.
- A single consumer subscribes to a specific topic, assume `Topic-01` with Group ID as `Group-1`.

with Group ID as `Group-1`.

- Kafka interacts with the consumer in the same way as pub-sub messaging until a new consumer subscribes to the same topic, `Topic-01`, with the same Group ID as `Group-1`.
- Once the new consumer arrives, Kafka switches its operation to share mode, such that each message is passed to only one of the subscribers of the consumer group `Group-1`. This message transfer is similar to queue-based messaging, as only one consumer of the group consumes a message. Contrary to queue-based messaging, messages are not removed after consumption.
- This message transfer can go on until the number of consumers reaches the number of partitions configured for that particular topic.
- Once the number of consumers exceeds the number of partitions, the new consumer will not receive any message until an existing consumer unsubscribes. This scenario arises because each consumer in Kafka will be assigned a minimum of one partition. Once all the partitions are assigned to the existing consumers, the new consumers will have to wait.

← Back
Consumer Groups

Next →
Role of ZooKeeper

☑ Mark as Completed

Report an Issue