



# Consumer Groups

Let's explore the role of consumer groups in Kafka.

## We'll cover the following



- What is a consumer group?
  - Distributing partitions to consumers within a consumer group

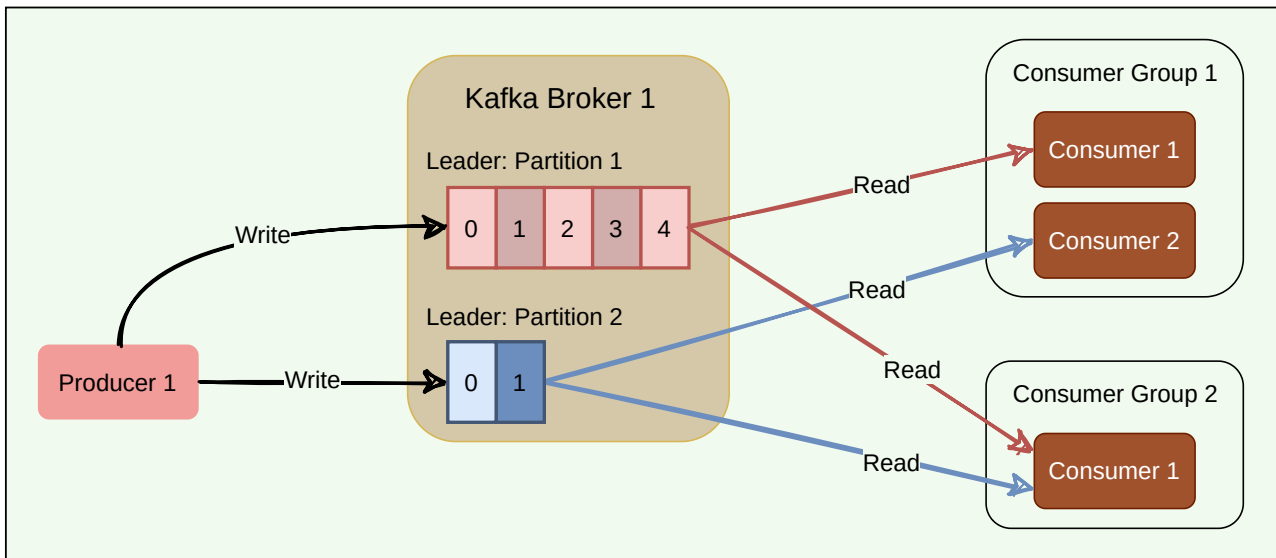
## What is a consumer group?#

A consumer group is basically a set of one or more consumers working together in parallel to consume messages from topic partitions. Messages are equally divided among all the consumers of a group, with no two consumers receiving the same message.

## Distributing partitions to consumers within a consumer group#

Kafka ensures that **only a single consumer reads messages from any partition within a consumer group**. In other words, topic partitions are a unit of parallelism – only one consumer can work on a partition in a consumer group at a time. If a consumer stops, Kafka spreads partitions

across the remaining consumers in the same consumer group. Similarly, every time a consumer is added to or removed from a group, the consumption is rebalanced within the group.



How Kafka distributes partitions to consumers within a consumer group

Consumers pull messages from topic partitions. Different consumers can be responsible for different partitions. Kafka can support a large number of consumers and retain large amounts of data with very little overhead. By using consumer groups, consumers can be parallelized so that multiple consumers can read from multiple partitions on a topic, allowing a very high message processing throughput. The number of partitions impacts consumers' maximum parallelism, as there cannot be more consumers than partitions.

Kafka stores the current offset per consumer group per topic per partition, as it would for a single consumer. This means that unique messages are only sent to a single consumer in a consumer group, and the load is balanced across consumers as equally as possible.

When the number of consumers exceeds the number of partitions in a topic, all new consumers wait in idle mode until an existing consumer unsubscribes from that partition. Similarly, as new consumers join a

consumer group, Kafka initiates a rebalancing if there are more consumers than partitions. Kafka uses any unused consumers as failovers.



Here is a summary of how Kafka manages the distribution of partitions to consumers within a consumer group:

- **Number of consumers in a group = number of partitions:** each consumer consumes one partition.
- **Number of consumers in a group > number of partitions:** some consumers will be idle.
- **Number of consumers in a group < number of partitions:** some consumers will consume more partitions than others.

[← Back](#)[Kafka: Deep Dive](#)[Next →](#)[Kafka Workflow](#)[Mark as Completed](#)[Report an Issue](#)