Controller Broker

This lesson will explain the role of the controller broker in Kafka.

We'll cover the following

- What is the controller broker?
- Split brain
- Generation clock

What is the controller broker?#

Within the Kafka cluster, one broker is elected as the Controller. This Controller broker is responsible for admin operations, such as creating/deleting a topic, adding partitions, assigning leaders to partitions, monitoring broker failures, etc. Furthermore, the Controller periodically checks the health of other brokers in the system. In case it does not receive a response from a particular broker, it performs a failover to another broker. It also communicates the result of the partition leader election to other brokers in the system.

Split brain#

When a controller broker dies, Kafka elects a new controller. One of the problems is that we cannot truly know if the leader has stopped for good and has experienced an intermittent failure like a stop-the-world GC

pause or a temporary network disruption. Nevertheless, the cluster has to move on and pick a new controller. If the original Controller had an intermittent failure, the cluster would end up having a so-called **zombie controller**. A zombie controller can be defined as a controller node that had been previously deemed dead by the cluster and has come back online. Another broker has taken its place, but the zombie controller might not know that yet. This common scenario in distributed systems with two or more active controllers (or central servers) is called split-brain.

We will have two controllers under split-brain, which will be giving out potentially conflicting commands in parallel. If something like this happens in a cluster, it can result in major inconsistencies. How do we handle this situation?

Generation clock#

Split-brain is commonly solved with a **generation clock,** which is simply a monotonically increasing number to indicate a server's generation. In Kafka, the generation clock is implemented through an epoch number. If the old leader had an epoch number of '1', the new one would have '2'. This epoch is included in every request that is sent from the Controller to other brokers. This way, brokers can now easily differentiate the real Controller by simply trusting the Controller with the highest number. The Controller with the highest number is undoubtedly the latest one, since the epoch number is always increasing. This epoch number is stored in ZooKeeper.

← Back

Role of ZooKeeper

Next →

Kafka Delivery Semantics

Mark as Completed


Report an Issue