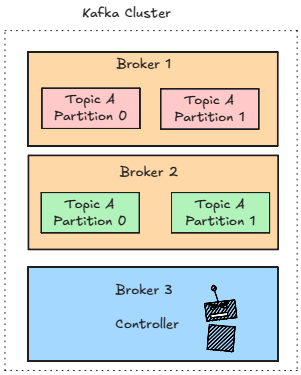
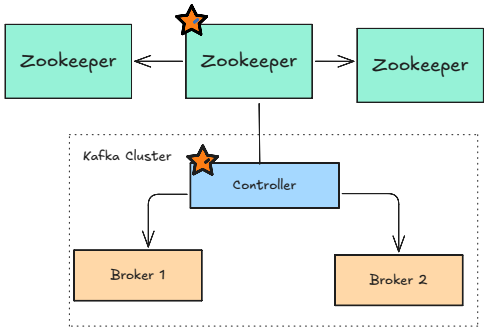
Controller



What is the controller?

- Within the cluster, one broker is elected as a controller.
- Responsible for admin operations (creating deleting topics, adding partitions, assigning leaders to partitions, monitoring failures)
- Propagates metadata changes to other brokers in the clusters

Zookeeper



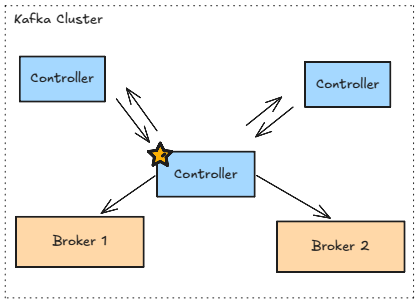
Zookeper and interaction with the controller

- External distributed coordination service
- Served as the centralized metadata store and coordination service for the kafka cluster
- Interact heavily with the Controller Broker to manage the cluster
- Responsible for electing the controller
- Stores metadata (list of active brokers, topic configs, ACLs, etc)
- Sends changes to controller (broker joining or leaving the cluster)

Zookeper drawbacks

- Two systems to operate (Kafka + ZK)
- Slow/complex controller failover (when the controller broker goes down) in large clusters: the controller needs to load all metadata from ZK when it starts, a lot of data in tens of thousads of partitions to read, and needs to notify all brokers
- Metadata spread outside kafka, harder upgrades and tooling

KRaft (Kafka Raft)



Kafka Raft

- Removes zookeper dependency
- Uses the Raft protocol to manage cluster metadata
- Raft is a consensus algorithm that ensures all nodes in a distributed system agree on the same state even if some fail
- It elects a leader that manages log replication to followers
- Guarantees consistency and fault tolerance through majority agreement (quorum)
- It allows broker themselves to agree on metadata and elect a controller internally