

Data Plane: Replication Protocol

Replication



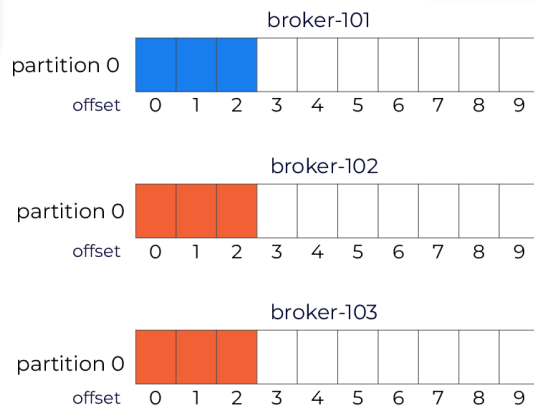
- Copies of data for fault tolerance
- One lead partition and N-1 followers
- In general, writes and reads happen to the leader
- An invisible process to most developers
- Tunable in the producer

@yourtwitterhandle | developer.confluent.io

Kafka Data Replication

Each topic partition is replicated to multiple brokers

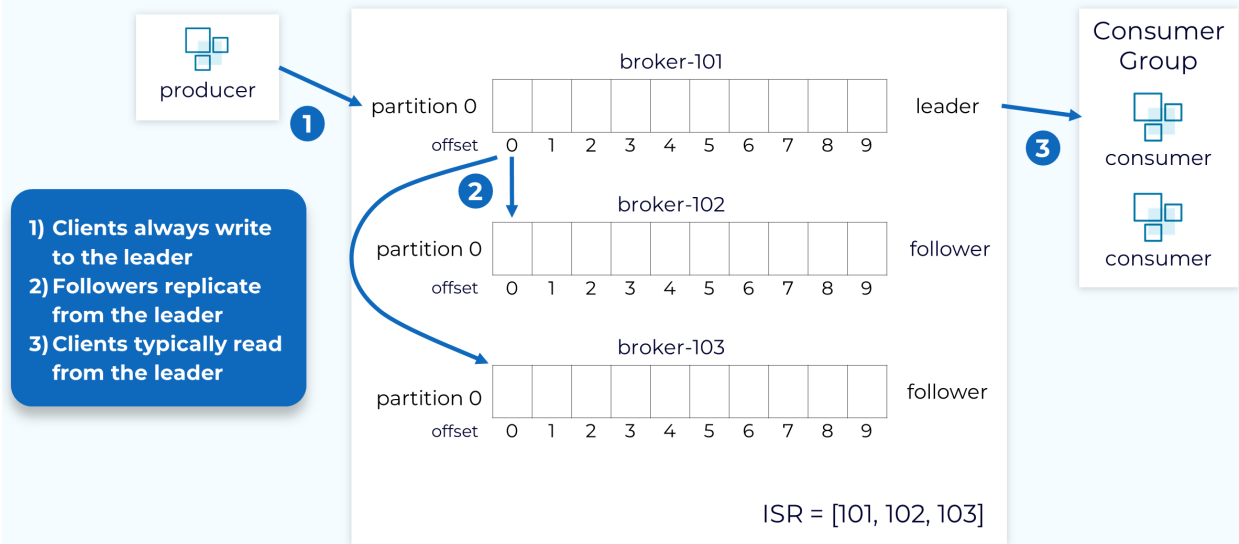
Given n replicas, no data loss with $n-1$ replica failure



Consumer Group



Leader, Follower, and In-Sync Replica (ISR) List

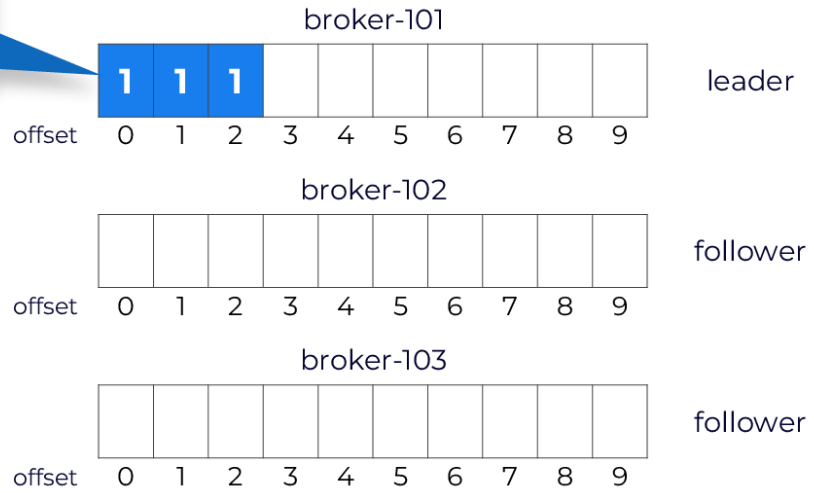


Leader Epoch

In Kafka, a leader epoch refers to **the number of leaders previously assigned by the controller**. Every time a leader fails, the controller selects the new leader, increments the current "leader epoch" by 1, and shares the leader epoch with all replicas

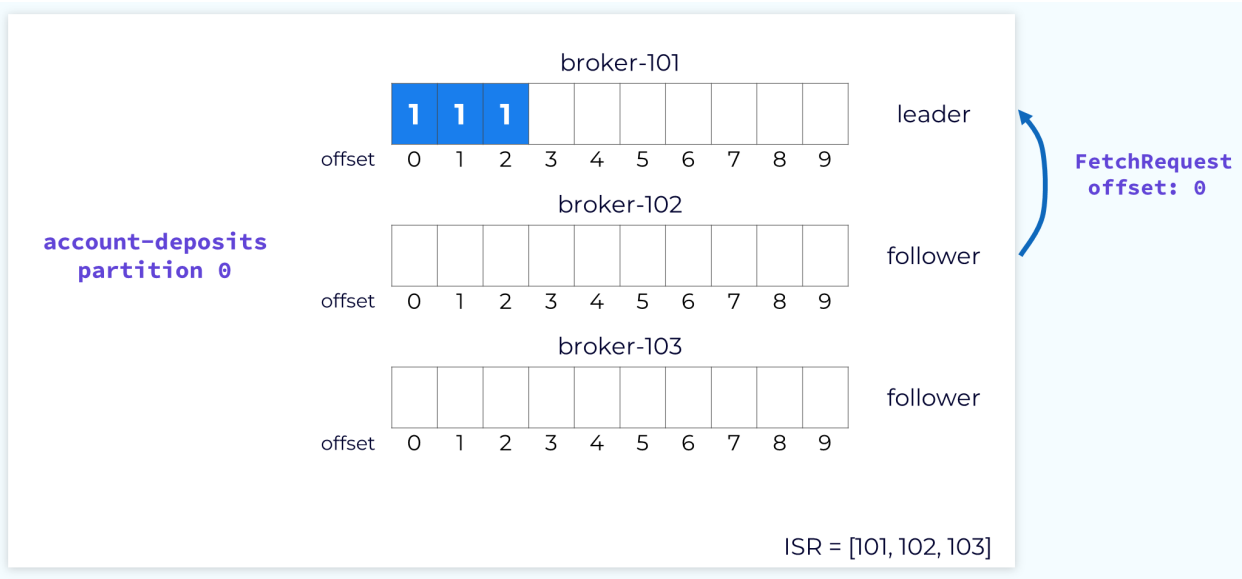
Leader epoch is included for a batch of records

account-deposits
partition 0

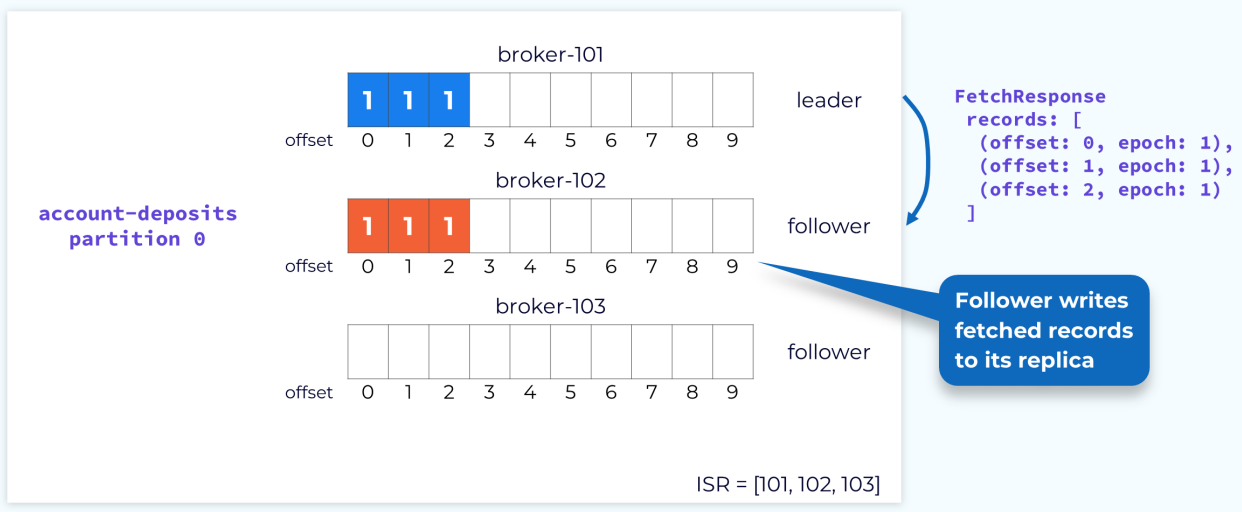


ISR = [101, 102, 103]

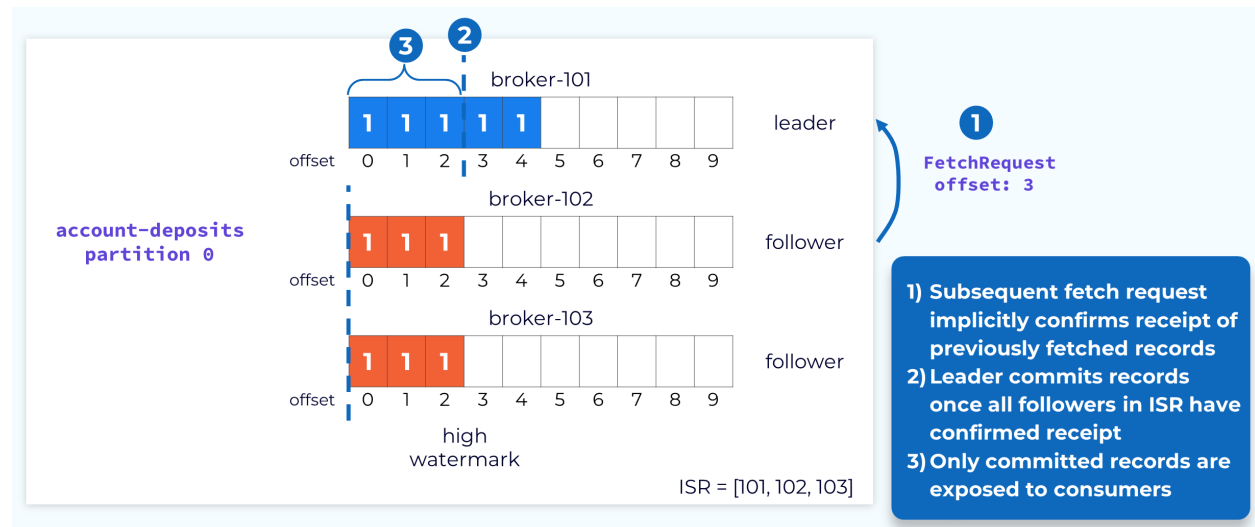
Follower Fetch Request



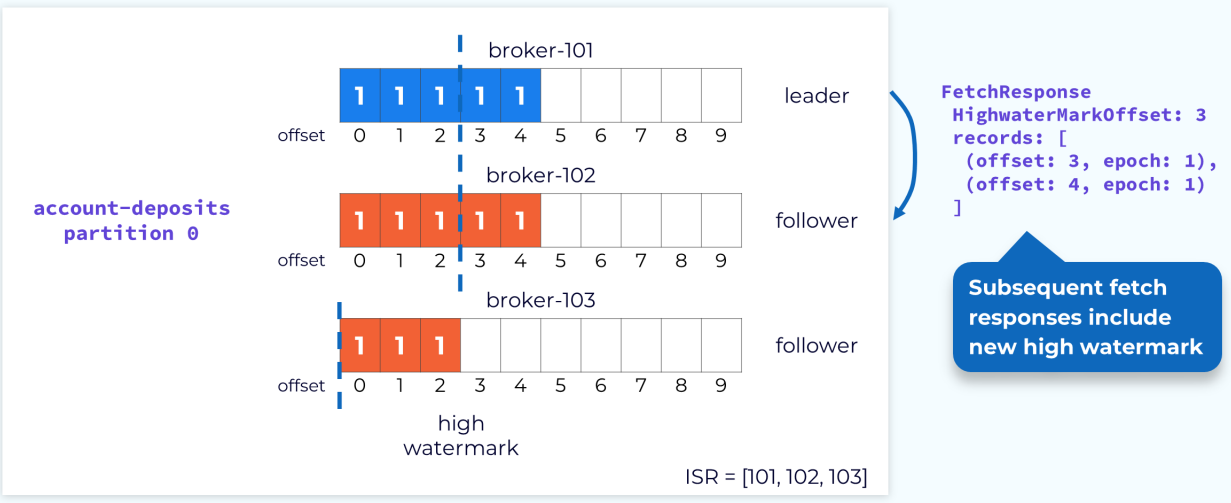
Follower Fetch Response



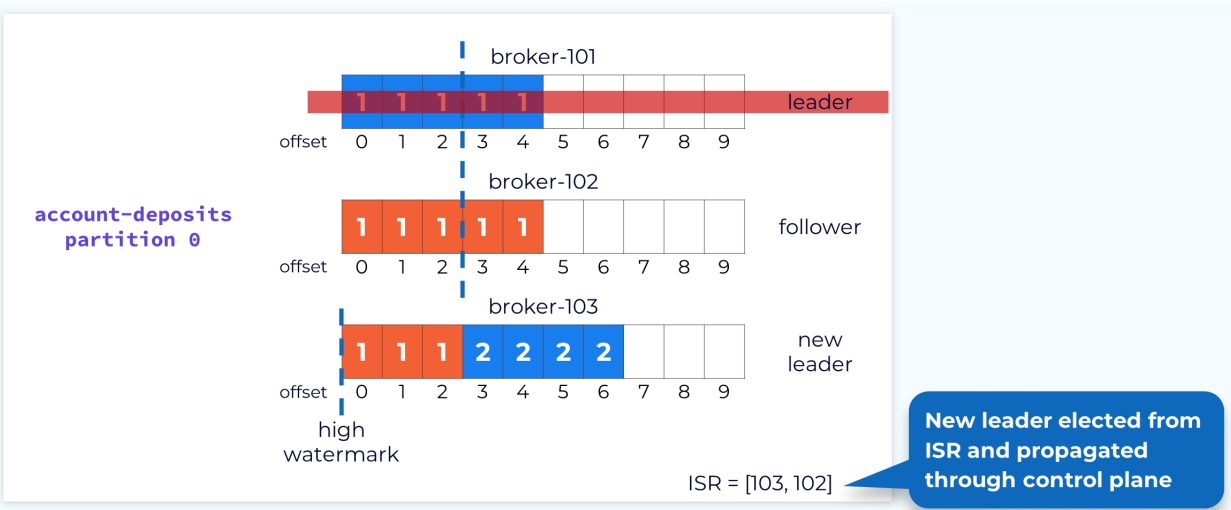
Committing Partition Offsets



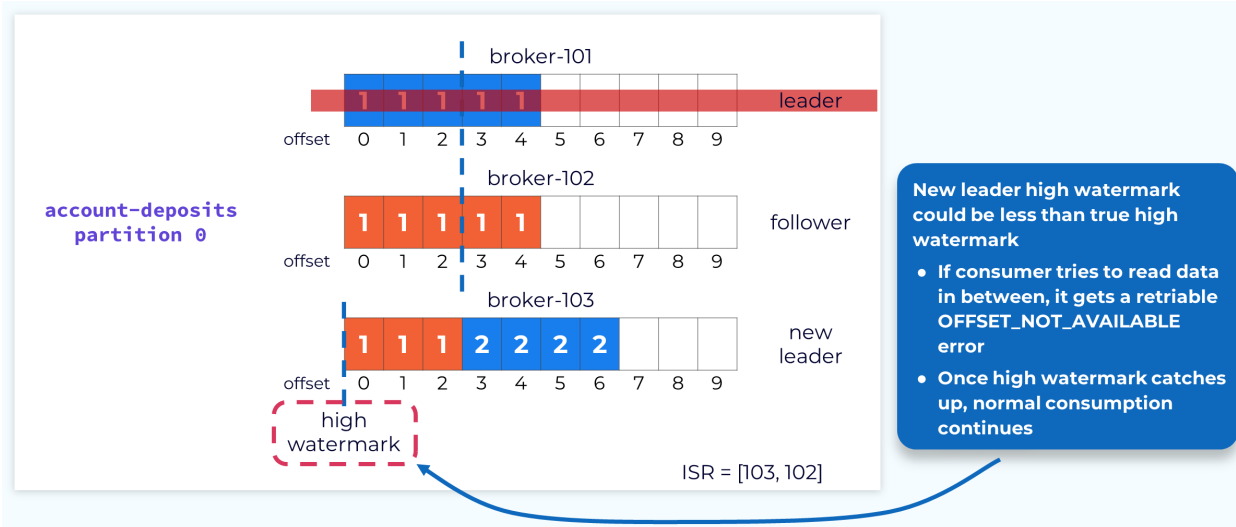
Advancing the Follower High Watermark



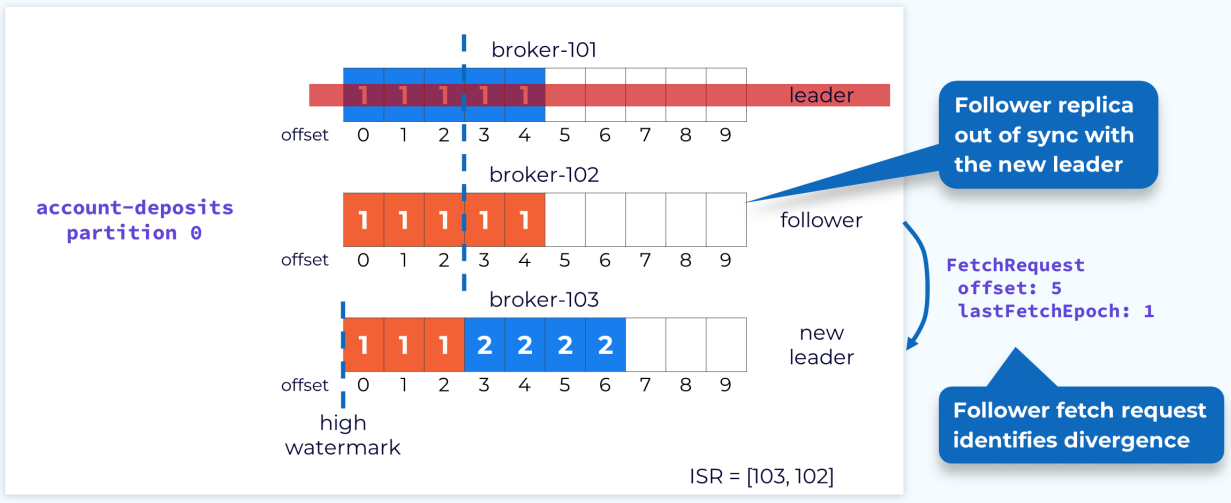
Handling Leader Failure



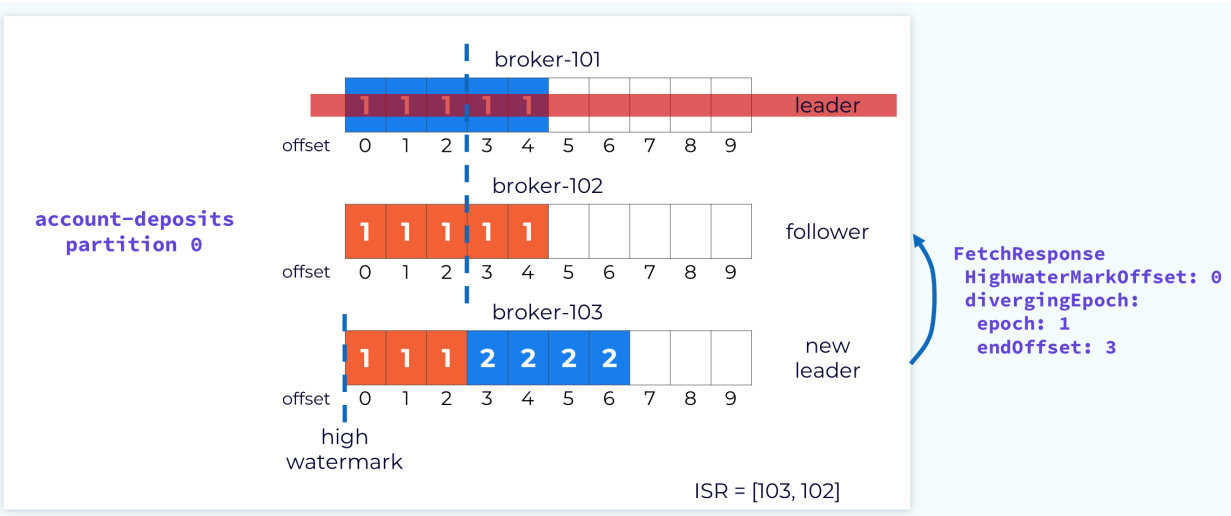
Temporary Decreased High Watermark



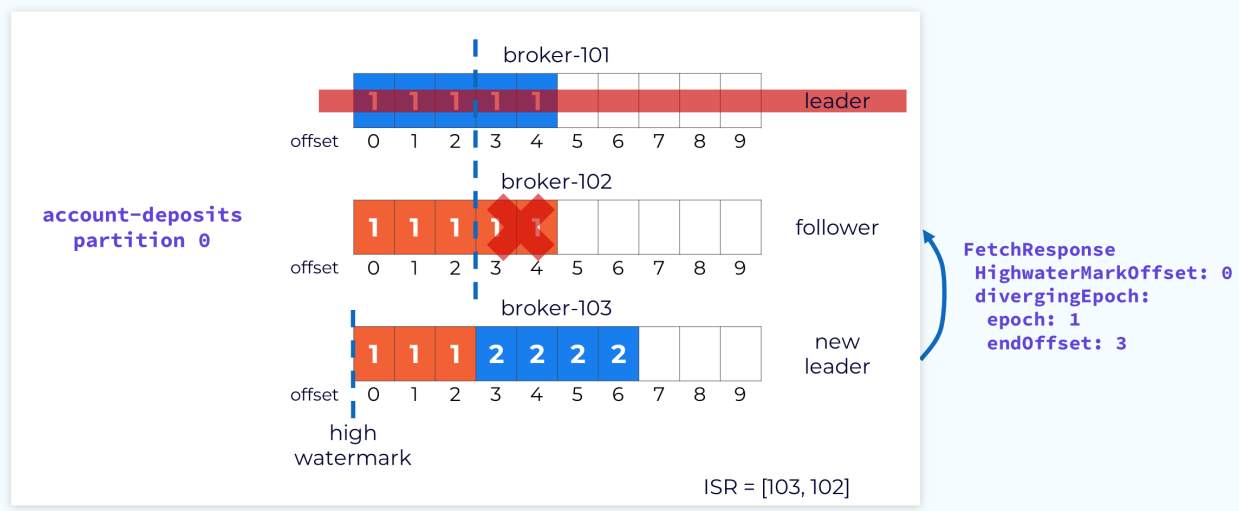
Partition Replica Reconciliation



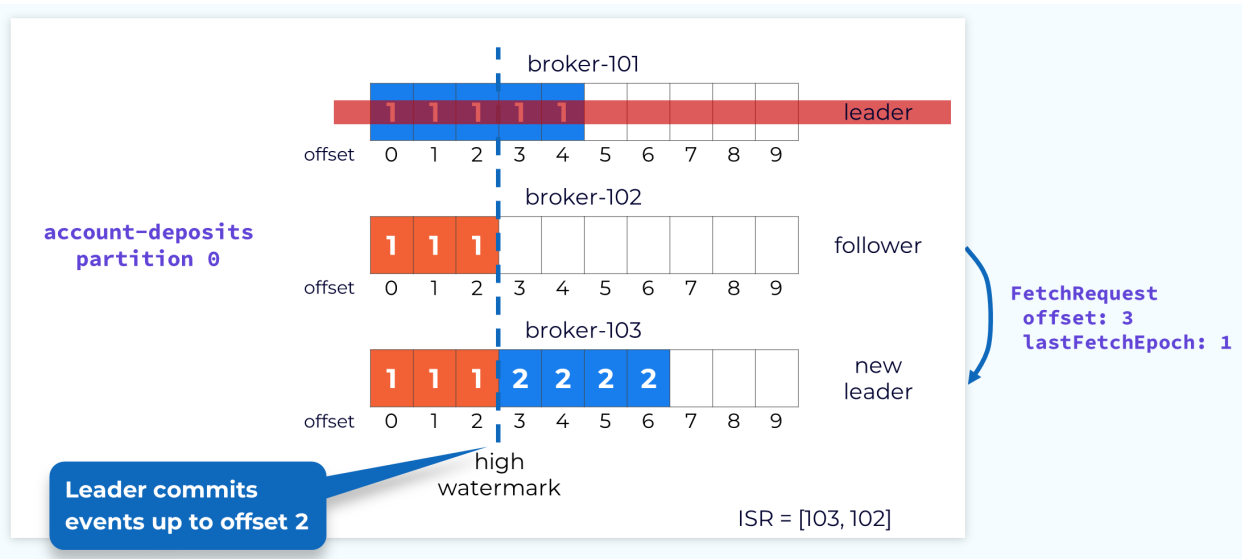
Fetch Response Informs Follower of Divergence



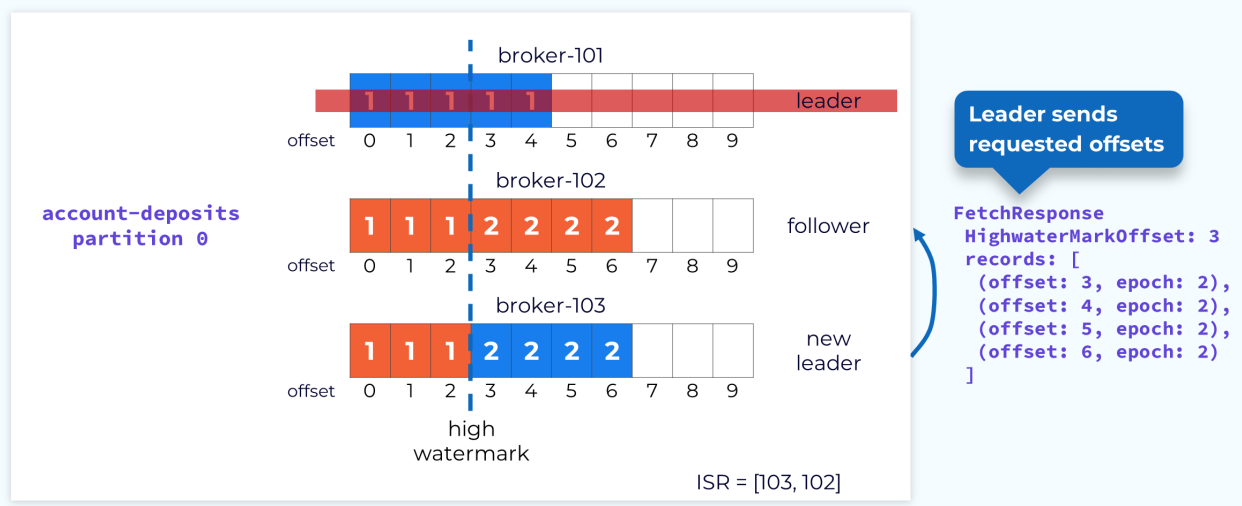
Follower Truncates Log to Match Leader Log



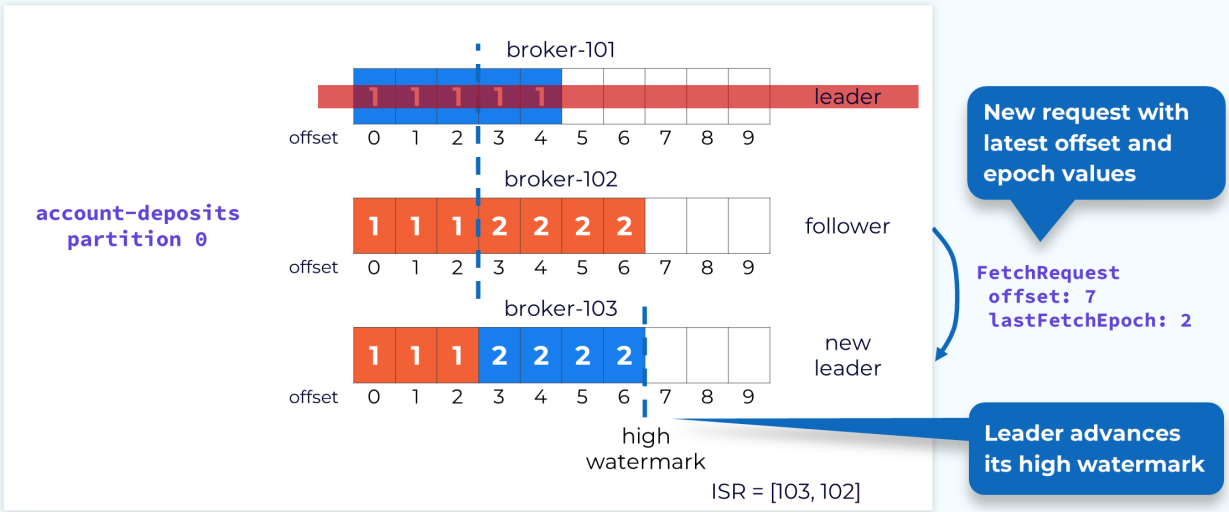
Subsequent Fetch with Updated Offset and Epoch



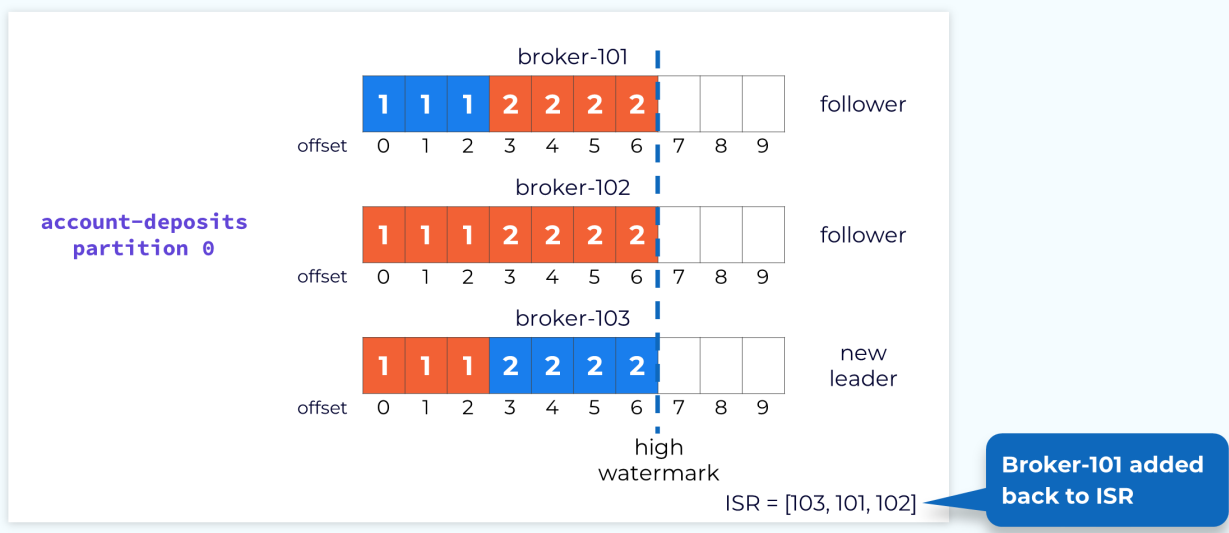
Follower 102 Reconciled



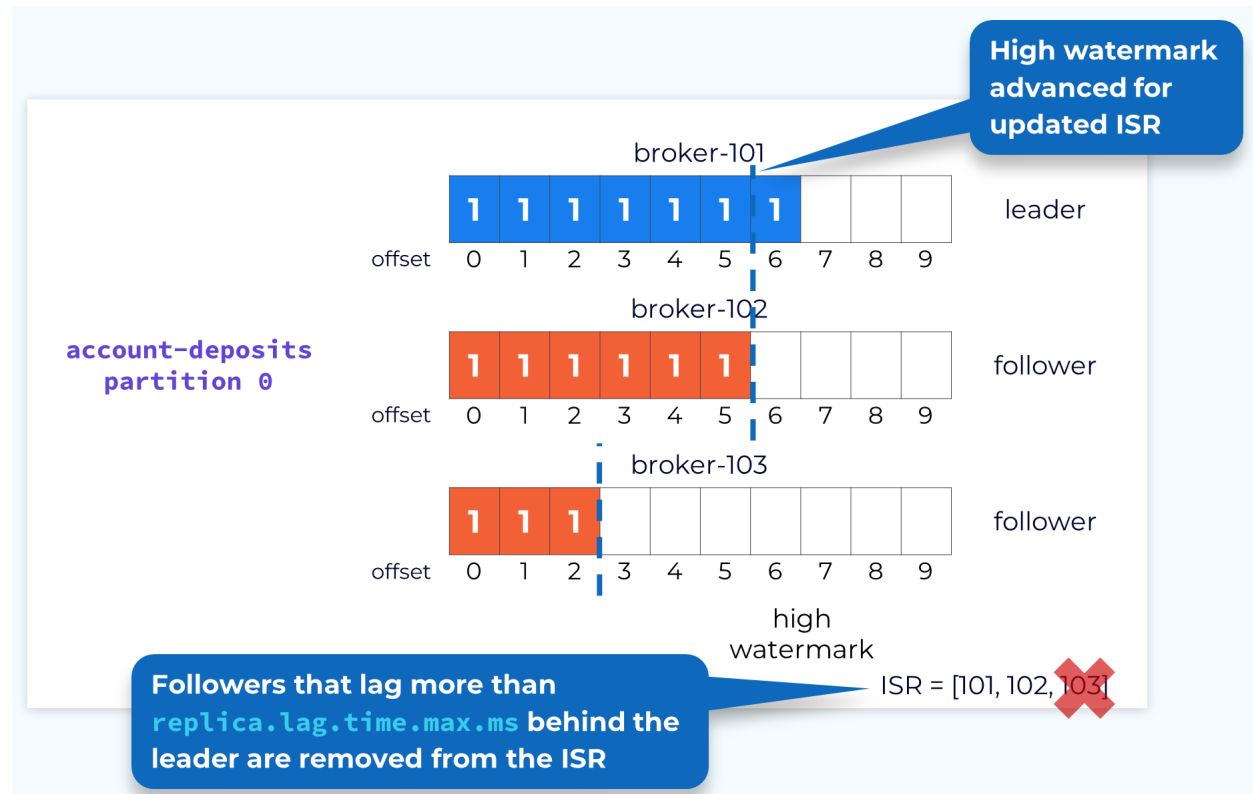
Follower 102 Acknowledges New Records



Follower 101 Rejoins the Cluster

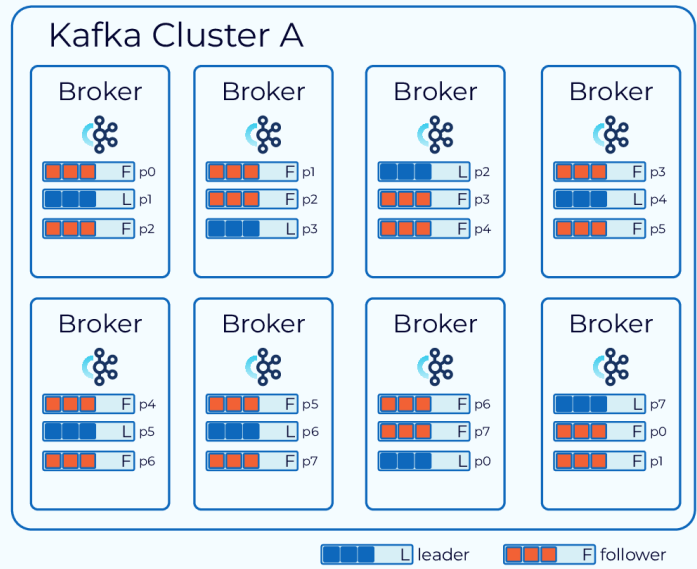


Handling Failed or Slow Followers



Partition Leader Balancing


- First replica considered preferred
- Preferred replica distributed evenly during assignment
- Background thread moves leader to preferred replica when it's in-sync



- As we've seen, the broker containing the leader replica does a bit more work than the follower replicas.
- Because of this it's best not to have a disproportionate number of leader replicas on a single broker.
- To prevent this Kafka has the concept of a preferred replica.
- When a topic is created, the first replica for each partition is designated as the preferred replica.
- Since Kafka is already making an effort to evenly distribute partitions across the available brokers, this will usually result in a good balance of leaders.
- As leader elections occur for various reasons, the leaders might end up on non-preferred replicas and this could lead to an imbalance.
- So, Kafka will periodically check to see if there is an imbalance in leader replicas. It uses a configurable threshold to make this determination.
- If it does find an imbalance it will perform a leader rebalance to get the leaders back on their preferred replicas.

Kafka leader election

In Kafka, leader election is the process of selecting a new leader for a partition when the current leader fails or becomes unavailable...

 <https://levelup.gitconnected.com/kafka-leader-election-4e7dfad2aa18>

