When a Kafka producer encounters a **NotEnoughReplicasException**, it indicates that the producer's request to write a message was not acknowledged by the minimum number of **in-sync replicas (ISRs)** specified in the acks setting. Here’s how the leader broker handles replication and what happens during retries:

1. **Initial Message Publishing**

* **Producer Sends Message:** The Kafka producer sends a message to the leader broker of the partition.
* **Leader Writes to Local Log:** The leader broker writes the message to its local log.
* **Replication to Followers:** The leader broker then replicates this message to the follower brokers (other replicas of the partition).

1. **NotEnoughReplicasException**

* **Insufficient Acknowledgments:** If the leader broker doesn't receive acknowledgments from the required number of followers (based on the acks setting), it will throw a NotEnoughReplicasException.
* **Producer Retry:** The producer catches this exception and will typically retry the request according to its retry policy.

1. **Handling the Retry**

* **Retry Logic:** Upon retry, the producer attempts to resend the message to the leader broker.
* **Leader Broker State:** The leader broker checks the state of the message in its log and the state of the followers.

**Replication Steps During Retry**

1. **Message Check:** The leader broker verifies if the message already exists in its local log (which it should if the initial write was successful, but replication failed).
2. **Replication Retry:** The leader broker attempts to replicate the message to the follower brokers again.

* **Sync with Followers:** The leader broker will continue to send the message to the followers until the required number of acknowledgments are received or the configured retry attempts are exhausted.
* **Follower Catch-up:** If followers were temporarily unavailable or lagging, they might catch up and acknowledge the message during this retry process.

**Key Points**

* **Persistence:** The message is already persisted in the leader's log even if the initial replication attempt failed.
* **Acks Configuration:** The number of acknowledgments required is controlled by the producer's acks setting:

1. **acks=1:** The producer requires only the leader’s acknowledgment.
2. **acks=all or acks=-1:** The producer requires acknowledgment from all in-sync replicas.

* **Retries:** The producer's retry policy (e.g., retries and retry.backoff.ms settings) determines how many times and how quickly the producer will retry sending the message.
* **ISR Changes:** If the cluster state changes (e.g., replicas come back online or go offline), the ISR list might be updated, affecting subsequent replication attempts.

**Example Configuration for Producer**

* acks=all
* retries=5
* retry.backoff.ms=100

This configuration ensures that the producer retries up to 5 times with a 100ms backoff between attempts, waiting for all in-sync replicas to acknowledge the message.

**Conclusion**

When a **NotEnoughReplicasException** occurs, the Kafka producer retries the message, and the leader broker attempts to replicate the message to the follower brokers again. **The replication process ensures that the message is eventually acknowledged by the required number of in-sync replicas, thereby maintaining the** **integrity and reliability of the Kafka cluster**.