KIP-217: Expose a timeout to allow an expired ZK session to be re-created

Status

Current state: ["Cancelled"]

Discussion thread: *here [Change the link from the KIP proposal email archive to your own email thread]*

JIRA: *KAFKA-5473*

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Resolution

After the discussion, we decided not to introduce the reconnect timeout config and just do infinite retry of the creation of the Zookeeper object. This way, the broker can be brought down at the time that the admin determines. There is no need to deprecate this property once ZOOKEEPER-2184 is fixed.

Motivation

When a broker's ZK session expires, the broker will try to reconnect to ZK in a session by instantiating a new Zookeeper object. The reconnection could fail if the ZK host can't be resolved (e.g., DNS issue). Currently, when such a failure occurs, the broker just logs the error. This means that even though the broker is still alive, it will never be re-registered in ZK even after the ZK host can be resolved subsequently. An admin will need to restart the broker manually in order for the broker to be added back to the Kafka cluster.

Public Interfaces

Proposed Changes

We propose a new configuration zookeeper.reconnection.timeout.ms on the broker. When a ZK session expires, a broker will keep retrying the creation of the Zookeeper object until the configured timeout. If creation fails after the timeout, the broker will call exit() and shut itself down. There will be a fixed retry interval of 1 second since a failed re-creation doesn't add new connections to ZK and doesn't introduce much additional overhead. The default value of zookeeper.reconnection.timeout.ms will be max_long. While reconnecting, the ZK SessionState metric will be set to "RECONNECTING".

An alternative is to shut down the broker immediately if the re-creation of the Zookeeper object fails. However, this has the drawback that if all brokers fail the re-creation at the same time, the whole cluster will be down.

There is an existing zookeeper.connection.timeout.ms, which controls the time that a broker connects to ZK during starting up. We choose not to reuse this config for re-connection since we want the broker to fail fast during starting up if the ZK URL is not configured properly.

Compatibility, Deprecation, and Migration Plan

With the change, the broker will still be alive on failing to reconnect to ZK, which is the current behavior. However, the improvement is that the broker may eventually be able to reconnect to ZK automatically.

Rejected Alternatives