KIP-409: Allow creating under-replicated topics and partitions

co-authored-by: Mickael Maison <mickael.maison@gmail.com>

co-authored-by: Edoardo Comar <ecomar@uk.ibm.com>

Status

Current state: Draft

Discussion thread: *here*

JIRA: *here [Change the link from KAFKA-1 to your own ticket]*

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

Motivation

Currently when handling topics or partitions creation requests, Kafka enforces all replicas to be created in order to fulfill the request. While all other functionalities (produce/consume) are fault tolerant and can handle some brokers down, topics and partitions creation stop working as soon as there are no enough replicas available. In small clusters, when one node is unavailable, for example when a broker is being restarted, it's possible that there are not enough online brokers to satisfy topic/partition creation even though there are enough brokers to satisfy the "`min.insync.replicas`" configuration of the topic.

This is mostly impacting clusters with 3 brokers for which topics with replication factor 3 can't be created while a rolling restart is happening. This makes this cluster size unsuitable for environments requiring maximum availability and forces administrators to deploy at least 4 brokers. In case of a stretch cluster spanning 3 availability zones, which is now a relatively common deployment, this forces administrators to have 6 brokers.

This features will allow small clusters to stay fully available over rolling restarts and basic maintenance operations.

Proposed Changes

We would like to allow users to create topics and partitions even when the current available number of brokers is less than the number of requested replicas. It would still require at least enough brokers to satisfy the "`min.insync.replicas`" configuration of the topic. This guarantees producers will be able to use the topic/partition regardless of their `acks` setting, ie the lack of replicas will be invisible to all users. Because it's possible to create a topic with less replicas than "min.insync.replicas", the actual requirement will be min(min.insync.replicas, replicas).

When handling a CreateTopics or CreatePartitions request, in case not enough brokers are available to satisfy the replication factor, placeholder replica ids (-1, -2, -3) will be inserted for the missing replicas. In case a topic or partition is created under-replicated, the error code will still be NONE in the CreateTopics and CreatePartition responses.

When a new broker joins the cluster, the controller will check if any partitions have placeholders replicas ids and if so assign this broker as a replica (only if this broker is not already a replica).
This may lead to some topics/partitions not optimally spread across all racks, but note that this may already happen (without this KIP) when topics /partitions are created while all brokers in a rack are offline (ie: an availability zone is offline). Tracking topics/partitions not optimally spread across all racks can be tackled in a follow up KIP.

Because this feature can make partitions appear under replicated, it is disabled by default and can be enabled by a broker configuration: `enable.under.replicated.topic.creation`.

Public Interfaces

Broker Configuration

NAME	DESCRIPTION	TYPE	DEFAULT	VALID VALUES	IMPORTANCE	DYNAMIC UPDATE

enable.under.replicated.topic.creation	Behavior on Topic or Partition creation without the full set of replicas being active.	BOOLEAN	false	true, false	Medium	cluster-wide

Administrators can disable (false), or enable (true) creations of topics/partitions without the full replication factor. Defaulting to false for compatibility.

Compatibility, Deprecation, and Migration Plan

- For compatibility the default behavior remains unaltered: a number of online brokers greater or equals to the replication factor is required for successful creation.
- No migration is necessary.

Rejected Alternatives

- Store "observed brokers" in Zookeeper and assign replicas to them even if they are offline. Because broker registration is implicit, this alternative required storing extra data in Zookeeper and also required administrators to delete znodes when decommisioning brokers. We cannot tell people to modify ZK directly for this.
- Update `CreateTopicsRequest` and `CreatePartitionsRequest` to allow users to disable creation of under replicated topics. This option adds an extra configuration to think about to end users and it's unclear how many people would want to take advantage of such a feature.
- Update schema of `CreateTopicsResponse` and `CreatePartitionsResponse` to contain the actual replication factor at creation. Like assignments, it's not data the client can act on.
- Create a new error code when a topic/partition is created under replicated. As this is not an error case, it's best to keep returning NONE to avoid breaking existing logic.