KIP-720: Deprecate MirrorMaker v1

- Status
- Motivation
- Public Interfaces
- Proposed Changes
- Compatibility, Deprecation, and Migration Plan
- Rejected Alternatives

Status

Current state: Adopted

Discussion thread: here

Voting thread: here

JIRA: [KAFKA-12436](#) - Getting issue details...

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

Motivation

Currently there are two mirror-maker implementations, the original MirrorMaker and MirrorMaker 2 (MM2), the latter being a complete rewrite leveraging the Connect framework. We believe MirrorMaker 2 is an improvement over the original MirrorMaker when it comes to reliability and functionality for the majority of use cases. We intend to focus on MirrorMaker 2 for future development and hence we propose deprecating the original first version of MirrorMaker with the intention to remove it completely in the future.

Public Interfaces

No API changes are proposed.

Proposed Changes

Legacy MirrorMaker classes will be marked deprecated but kept in-place. The mirror-maker.sh command-line tool will be modified to include a deprecation warning.

Compatibility, Deprecation, and Migration Plan

No functional changes at this point. In the subsequent major release (4.0), we can remove the legacy mirror-maker.sh command-line tool and related code entirely.

N.B. prior to removing MM1 (in v4.0) we will need to land IdentityReplicationPolicy (aka LegacyReplicationPolicy), as described in KIP-382, which provides for use-cases that MM2 cannot otherwise handle out-of-the-box. Various IdentityReplicationPolicy implementations can be found floating around in the meantime.

Rejected Alternatives

- We could provide a shim between the legacy command-line tool and the new connect-based implementation, as originally proposed in KIP-382. However, given the length of time between major releases, it doesn't seem necessary to make the transition between MM1 and MM2 quite so frictionless. Instead, users will have a long time and ample warning to transition away from MM1 at their own pace. If, for some reason, a new major release lands sooner than expected, we can revisit this decision.