KIP-109: Old Consumer Deprecation

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
- [Proposed Changes](#)

This KIP is meant to follow up the discussion and decision around the old Scala consumer deprecation that started by Ismael Juma Juma in this thread.

Status

Current state: Accepted

Discussion thread: *here*

JIRA: *here*

Released: <Kafka Version> (per discussions on the discussion thread and also the voting thread, this KIP will be included in the release following 0.10.2.0)

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

Motivation

Quoting Ismael Juma Juma on the original thread (Oct 25, 2016):

> In 0.10.1.0, we removed the beta label from the new Java consumer documentation and updated the various tools so that they can use the new consumer without having to pass the `--new-consumer` flag (more specifically the new consumer is used if `bootstrap-server` is set). More details of the reasoning can be found in the original discuss thread: http://search-hadoop.com/m/Kafka/uyzND1e4bUP1Rjq721
>
> The old consumers don't have security or `offsetsForTimestamp` (KIP-79) support and the plan is to only add features to the new Java consumer. Even so, the old consumers are a significant maintenance burden as they duplicate protocol request/response classes (the SimpleConsumer exposes them in the public API sadly). I experienced this first hand most recently while working on KIP-74.
>
> Given the above, I propose we deprecate the old consumers in trunk to nudge users in the right direction. Users will have the 0.10.1.0 cycle to start migrating to the new Java consumer with no build warnings. Once they upgrade to the next release (i.e. 0.10.2.0), users who are still using the old consumers will get warnings at build time encouraging them to move to the new consumer, but everything will still work fine.
>
> In a future major release, the old consumers (along with the old producers) will be removed. We will have a separate discuss/vote thread for that to make sure the time is right.

Another enticing feature of the Java consumer, that's missing in the old consumer, is its compatibility with older brokers (0.10.0 and later).

Public Interfaces

No new public interfaces are proposed, or existing interfaces are modified, or removed.

Proposed Changes

As described by Ismael Juma Juma the intention at this point is to just mark the old consumer as deprecated and issue build warnings to users and let them know that the old consumer will be removed in a following major release (potentially 0.11.0.0).

Compatibility, Deprecation, and Migration Plan

There is no impact to existing users other than seeing the deprecation warnings at build time. Since the old consumer will be removed in a next major release a migration plan will be introduced later (some options are already suggested in the original discussion thread), if necessary, to lower the impact (downtime, etc.) to existing users of the old consumer.

Test Plan

None as no behavior changes is proposed at this time.

Rejected Alternatives