KIP-1 - Remove support of request.required.acks

Status

Current state: Accepted

Discussion thread: here

JIRA: KAFKA-1697

Released: 0.8.3

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

Motivation

Until Kafka 0.8.2, if a producer wanted to make sure a message was persisted at a specific number of replicas (N) before an "ack" was returned from the broker, they would specify request.required.acks=N. This was a bit misleading since specifying request.required.acks=2 would not actually protect against data loss in all cases.

In Kafka 0.8.2 we added min.isr feature which is a less misleading implementation of a similar behavior - users can specify request.required.acks=-1 (ack after writing to all ISR) and also specify minimum size of ISR to guarantee a minimum number of guaranteed copies.

In this KIP we propose to remove support of request.required.acks > 1 and return an error message to producers which request that.

New or Changed Public Interfaces

We will stop accepting request.required.acks > 1 from clients implementing the new version of the protocol and may return a new error code.

Proposed Change

In version 0.8.2, when Kafka broker receives request with acks > 1, we will log a warning to server log. The warning will include client_id, client IP and the number of acks. This will allow cluster admins to bring clients into compliance before we remove support of this configuration.

In version 0.8.3 request.required.acks > 1 will lead to error_code 21 sent to the client and an error logged to server log.

Migration Plan, Compatibility and Deprecation

Users upgrading brokers to 0.8.3 will need to change their configuration and avoid request.required.acks >1.

0.8.3 Java clients will get a InvalidRequiredAcksException. Other clients will get error code 21.

Rejected Alternatives

1. Silently replace request.required.acks>1 with request.required.acks=-1. Rejected as possibly very confusing to users who will see higher latency (and possibly lower reliability) with no explanation.
2. Bump protocol version to 1 to indicate new behavior and new error code, and support both behaviors in 0.8.3. Rejected as an overkill.