KIP-296: Connector level configurability for client configs

Expose the full set of configurations within Kafka clients to individual connectors in Kafka Connect

Status

Current state: *Under Discussion*

Discussion thread: *here*

JIRA: KAFKA-6890

Motivation

KAFKA-2798 (0.9.0.0) introduces capabilities for each source connector and sink connector to inherit their client configurations from the worker properties. Within the worker properties, any configuration that has a prefix of "producer." or "consumer." are applied to all source connectors and sink connectors respectively. The intent of the original change was to introduce a namespaced set of configurations to isolate the settings for different components, but it leaves users with a great deal of inflexibility on the connector level, as *all* worker source/sink tasks are forced to exhibit those same set of properties.

This KIP proposes supporting connector-level configurability of producer/consumer client configs that users can optionally provide within connector properties to override worker-level settings.

> ⚠ Unable to render Jira issues macro, execution error.

New or Changed Public Interfaces

- Upon initialization, all Kafka Connect source connectors will iterate through their list of connector properties, looking for those with keys prefixed by "producer."
- Upon initialization, all Kafka Connect sink connectors will iterate through their list of connector properties, looking for those with keys prefixed by "consumer."

Proposed Changes

Connector properties that are prefixed with "producer." and "consumer." are now used to feed into the producer and consumer clients embedded within source and sink connectors respectively. The prefixes will be removed via a String::substring() invocation, and the remainder of the configuration key will be used as the client configuration key. The value is fed directly to the client as the configuration value. If there were client configurations defined at the worker level, they are overriden by the connector-level client configurations.

Example: The default setting for `max.poll.records` is 500 for all Kafka consumers. At present, if we want to modify this consumer client configuration, we could add a configuration to the worker properties that reads "consumer.max.poll.records=1000" to double the maximum number of records returned in a single call to poll(). However, this would impact every single sink connector on that worker. With connector-level configurability of all client configurations, we can target specific connectors and their "max.poll.records" property by adding the "consumer.max.poll.records" configuration to the connector properties instead of the worker properties.

Other useful scenarios for this public API change include being able to set the "consumer.auto.offset.reset" configuration in order to modify the initial offset assignment for a *specific* sink connector, and "producer.enable.idempotence" to provide *specific* source connectors with exactly once delivery semantics to ensure that producer retries do not write duplicates of retried messages.

Compatibility, Deprecation, and Migration Plan

Existing connectors that do not provide any configuration that start with the necessary prefixes remain unaffected. The existing configurations are simply iterated over, as the new code sequence checks them for the necessary prefix.

This is a new feature that is fully backwards-compatible and does not impact the behavior for existing connectors that do not define these properties. Users who wish to utilize these new capabilities on existing connectors should send a PUT request to /connectors/{connector name}/config with a new set of connector properties with at least one client configuration preceded with the necessary prefix.

Rejected Alternatives

N/A