KIP-318: Make Kafka Connect Source idempotent

Status

Current state: *"Inactive" (implemented indirectly)*

Discussion thread: *here [Change the link from the KIP proposal email archive to your own email thread]*

JIRA: *KAFKA-7077*

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

Motivation

Kafka Connect Source currently has a "safe producer" approach to producing data. This means currently:

Worker.java:

```
// These settings are designed to ensure there is no data loss. They *may* be overridden via configs passed to the
// worker, but this may compromise the delivery guarantees of Kafka Connect.
...
producerProps.put(ProducerConfig.ACKS_CONFIG, "all");
producerProps.put(ProducerConfig.MAX_IN_FLIGHT_REQUESTS_PER_CONNECTION, "1");
```

*I*dempotent producers have been introduced in KIP-98 - Exactly Once Delivery and Transactional Messaging and allow produces to be idempotent and therefore:

- Prevent duplicates on network failure (idempotence)
- Increase performance of safe producers by using MAX_IN_FLIGHT_REQUESTS_PER_CONNECTION=5. See:

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

Currently Kafka Connect does not benefit from these two improvements which could be very valuable to existing connectors in order to improve both the performance and safety

Public Interfaces

We will add the following code to **Worker.java**:

```
producerProps.put(ProducerConfig.MAX_IN_FLIGHT_REQUESTS_PER_CONNECTION, "5")
producerProps.put(ProducerConfig.ENABLE_IDEMPOTENCE, true)
```

Proposed Changes

```
Worker.java:
```

```
producerProps.put(ProducerConfig.MAX_IN_FLIGHT_REQUESTS_PER_CONNECTION, "5")
producerProps.put(ProducerConfig.ENABLE_IDEMPOTENCE, true)
```

Compatibility, Deprecation, and Migration Plan

- *What impact (if any) will there be on existing users?*

Users running Kafka Connect against older clusters (< 1.0) may have a compatibility issue due to API changes for idempotent producers.

Rejected Alternatives

-