Record routing

Overview

Record routing in Conduit makes it easier to build one-to-many and many-to-many pipeline topologies. Or, in other words, it's possible to:

- route data from one source to multiple collections in one destination
- route data from multiple sources to multiple destinations.

Through this, a number of use cases are possible:

- Keep two instances synchronized. For example, two database instances with
- multiple tables in each can be kept in sync with a single pipeline with only
- one source and one destination connector.
- Route data from one collection (table, topic, index, etc.) to multiple
- collections on a single server.
- Route data to different instances or different systems altogether.

This can be done in two ways:

- by using a connector that supports writing to multiple collections,
- by using filter processors.

Using a connector that supports writing to multiple collections

TheOpenCDC format recommends that a record's source or destination collection is written to its metadata using theopencdc.collection key .

A destination connector can then utilize that information and write the record to the correct collection.

note It's the connectors that eventually provide the ability to read from multiple source collections or write to multiple destination collections. Consult the connector documentation to check if it supports this feature.

Example: Route data from multiple Kafka topics to multiple PostgreSQL tables on a single database server

Let's assume we have a pipeline that streams data from two Kafka topics,employees_finance andemployees_legal to PostgreSQL (both of the connectors support reads from multiple collections and writes to multiple collections). A pipeline configuration file could look like this:

---

version :

2.2 pipelines : -

id : example - pipeline status : running name : example - pipeline description :

'Kafka to PostgreSQL' connectors : -

id : kafka - source type : source plugin :

"builtin:kafka" name : kafka - source settings : servers :

"localhost:9092" topics :

"employees_finance,employees_legal"

-

id : pg - destination type : destination plugin :

"builtin:postgres" name : pg - destination settings : url :

"postgresql://username:password@localhost/testdb?sslmode=disable" table :

"employees" The Kafka connector will read records like the ones below:

{ "metadata" :

{ "opencdc.collection" :

"employees_finance" , // rest of metadata } , "payload" :

{ "after" :

{ "name" :

"John" } } // other record fields } { "metadata" :

{ "opencdc.collection" :

"employees_legal" , // rest of metadata } , "payload" :

{ "after" :

{ "name" :

"Adam" } } // other record fields } The PostgreSQL destination connector will check the value in theopencdc.collection metadata field. Based on that, it will write John's information to theemployees_finance table and Adam's information to theemployees_legal table.

Using afilter

processor with a condition

In some cases, records from a single source collection need to be written to multiple connectors. Even if a destination connector can write to multiple collections, that feature sometimes cannot be used because the destinations have very different configurations. For example, the records may need to be written to different databases or different systems altogether.

In scenarios like these, we can route records to multiple destination connectors using the built-ifilter processor with acondition .

The advantage of this approach is that there will still be only one source, which means that we will read the data only once.

Example: Route data from a single Kafka topic to different PostgreSQL databases

Let's assume we have a pipeline that reads employee data from a topic. The employees from the finance department (identified bydepartment: finance in the metadata) need to be written to one PostgreSQL database. Employees from the legal department (identified bydepartment: legal in the metadata) need to be written into a different database.

---

version :

2.2 pipelines : -

id : example - pipeline status : running name : example - pipeline description :

'Kafka to PostgreSQL' connectors : -

id : kafka - source type : source plugin :

"builtin:kafka" name : kafka - source settings : servers :

"localhost:9092" topics :

"employees_finance,employees_legal"

-

id : finance type : destination plugin :

"builtin:postgres" name : finance settings : url :

"postgresql://username:password@localhost/finance-db?sslmode=disable" table :

"employees" processors : -

id : employees_finance_only plugin :

"filter"

Filter out records where the metadata field "department" is NOT set to "finance".

condition : { { eq .Metadata.department "finance" | not } }

-

id : legal type : destination plugin :

"builtin:postgres" name : legal settings : url :

"postgresql://username:password@localhost/legal-db?sslmode=disable" table :

"employees" processors : -

id : employees_legal_only plugin :

"filter"

Filter out records where the metadata field "department" is NOT set to "legal".

condition : { { eq .Metadata.department "legal" | not } } Edit this page Previous OpenCDC record Next Pipeline Semantics