Using Kafka Connect Connectors with Conduit

TheConduit Kafka Connect Wrapper connector is a special connector that allows you to useKafka Connect connectors with Conduit. Conduit doesn't come bundled with Kafka Connect connectors, but you can use it to bring any Kafka Connect connector with Conduit.

This connector gives you the ability to:

- Easily migrate from Kafka Connect to Conduit.
- Remove Kafka as a dependency to move data between data infrastructure.
- Leverage a datastore if Conduit doesn't have a native connector.

Since the Conduit Kafka Connect Wrapper itself is written in Java, but most of Conduit's connectors are written in Go, it also serves as a good example of the flexibility of the Conduit Plugin SDK.

Let's begin.

How it works

To use the Kafka Connect wrapper connector, you'll need to:

1. Download theconduit-kafka-connect-wrapper
2. latest release,
3. at the time of writing this, we'll go withv0.4.3
4. and download the fileconduit-kafka-connect-wrapper-v0.4.3.zip
5. .
6. Download Kafka Connect JARs and any dependencies you would like to add.
7. Create a pipeline configuration file.
8. Run Conduit.

Setup

To begin, download theconduit-kafka-connect-wrapper latest release , at the time of writing this, we'll go withv0.4.3 and download the fileconduit-kafka-connect-wrapper-v0.4.3.zip .

Downloading the release is ourpreferred option to get the connector JAR. However, another option to get the JAR file is to build it from source, this could be useful in case you added your own changes to the connector and wanted to test them out, to do that we will clone the connector:

git clone[email protected] :ConduitIO/conduit-kafka-connect-wrapper.git Then, we need to build the connector. The Kafka Connect wrapper connector is written in Java, so it needs to be compiled.

cd conduit-kafka-connect-wrapper

./scripts/dist.sh Runningscripts/dist.sh will create a directory calleddist with following contents:

1. A script (which runs the connector). This script starts a connector instance.
2. Directorylibs
3. . This is where you add the connector JAR itself, and put the Kafka connector JARs and their dependencies (if any).

Now that we have everything setup, we can add the Kafka Connect connectors.

Download a connector and its dependencies

Thelibs directory is where you put the Kafka Connect connector JARs and their dependencies (if any). The wrapper plugin will automatically load connectors and all the other dependencies from alibs directory.

To download a connector from a Maven repository and all of its dependencies, you can usescripts/download-connector.sh . For example:

./scripts/download-connector.sh io.example jdbc-connector 2.1 .3 For usage, run./scripts/download-connector.sh --help .

You can also download them manually if needed, we will use the PostgreSQL Kafka Connect JDBC Connector. To install, add the following:

- Aiven's Kafka Connect JDBC Connectors
- Postgres Connector JAR

This connector allows you to connect to anyJDBC database .

Create Pipeline Configuration File

Now that the Kafka Connect connectors included inlib , we can use it in a pipeline configuration file.

1. Install Conduit
2. .
3. Create a pipeline configuration file: Create a folder calledpipelines
4. at the same level as your Conduit
5. binary. Inside of that folder create a file namedjdbc-to-file.yml
6. , checkSpecifications
7. for more details about Pipeline Configuration Files.

version :

2.0 pipelines : -

id : kafka - connect - pipeline status : running description : This pipeline is for testing connectors : -

id : jdbc - kafka - connect type : source plugin : standalone : conduit - kafka - connect - wrapper settings : wrapper.connector.class :

"io.aiven.connect.jdbc.JdbcSourceConnector" , connection.url :

"jdbc:postgresql://localhost/conduit-test-db" , connection.user :

"username" , connection.password :

"password" , incrementing.column.name :

"id" , mode :

"incrementing" , tables :

"customers" , topic.prefix :

"my_topic_prefix"

id : file - dest type : destination plugin : builtin : file settings : path :

"path/to/the/file.txt" 1. Run Conduit! And see how simple it is to migrate to Conduit.

Note that thewrapper.connector.class should be a class which is present on the classpath, i.e. in one of the JARs in thelibs directory. For more information, check theWrapper Configuration section. Edit this page Previous Connector Lifecycle Next Getting Started