Cloudera DataFlow for Data Hub 7.2.15

Ingesting Data into Apache Kudu in CDP Public Cloud

Date published: 2019-12-16
Date modified: 2022-05-12

CLOUDERA

https://docs.cloudera.com/

Legal Notice

Contents

Ingesting Data into Apache Kudu in CDP Public Cloud....................................4

Understand the use case.....4
Meet the prerequisites.....4
Create the Kudu target table.....5
Build the data flow.....6
Configure the Controller Service.....8
Configure the processor for your data source.....9
Configure the processor for your data target.....11
Start your data flow.....12
Verify that you can write data to Kudu.....12
Next steps.....13

Ingesting Data into Apache Kudu in CDP Public Cloud

You can use an Apache NiFi data flow to ingest data into Apache Kudu in the CDP Public Cloud following these steps.

Understand the use case

You can use Apache NiFi to move data from a range of locations, into Kudu in your Real-time Data Mart cluster in CDP Public Cloud.

Time series use cases analyse data obtained during specified intervals, and enable you to improve performance based on available data. Examples include:

- Optimizing yield or yield quality in a manufacturing plant
- Dynamically optimizing network capacity during peak load of better telecommunications uptime and services

These use cases require that you store events at a high frequency, while providing ad-hoc query and record update abilities. You can use Apache NiFi data flows into Apache Kudu and Apache Impala in a CDP Public Cloud Real-time Data Mart cluster to accomplish these goals.

Time Series use cases also need flexible ingest, at scale. You can use NiFi in a Data Hub Flow Management cluster, in the same environment, to build a data flow that reads your source data, modifies it to match the Kudu schema, and ingests into your Real-time Data Mart cluster for fast analytics and dashboard serving. Once you have successfully built the data flow in NiFi, you can query the data stored in Kudu by running Impala queries in Hue on the Real-time Data Mart cluster.

This use case walks you through the steps associated with creating an ingest-focused data flow from Apache Kafka in a Streaming cluster in CDP Public Cloud, into Apache Kudu in a Real Time Data Mart cluster, in the same CDP Public Cloud environment. This will get you started with creating a data ingest data flow into Kudu. If you are moving data from a location other than Kafka, review the *Getting Started with Apache NiFi* for information about how to build a data flow, and about other data ingest processor options.

Related Information

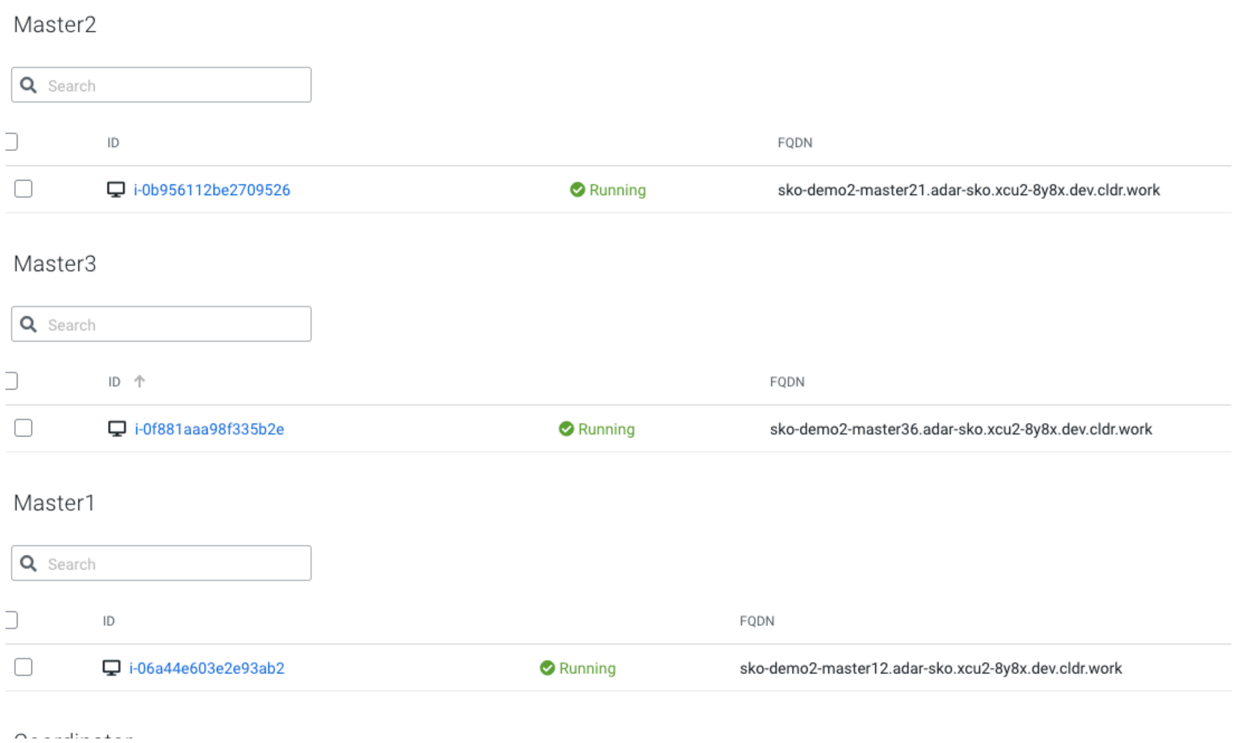
Getting Started with Apache NiFi
Ingesting data into Apache Kafka in CDP Public Cloud
Ingesting Data into Apache Hive in CDP Public Cloud
Ingesting Data into Apache HBase in CDP Public Cloud
Ingesting Data into Amazon S3 Buckets
Ingesting Data into Azure Data Lake Storage

Meet the prerequisites

Use this checklist to make sure that you meet all the requirements before you start building your data flow.

- You have a CDP Public Cloud environment.
- You have a workload username and password set to access Data Hub clusters. The predefined resource role of this user is at least "EnvironmentUser". This resource role provides the ability to view Data Hub clusters and set the FreeIPA password for the environment.
- Your user is synchronized to the CDP Public Cloud environment.
- You have a Flow Management cluster running Apache NiFi.
- You have a Real-Time Data Mart cluster running Kudu, Impala, and Hue. This cluster must be in the same CDP environment as the Flow Management cluster.

- You have Kudu master detail from the “Hardware” tab in your Real Time Data Mart cluster, so that you can configure the `PutKudu` NiFi processor. For example:



Master2

	ID		FQDN
	i-0b956112be2709526	Running	sko-demo2-master21.adar-sko.xcu2-8y8x.dev.cldr.work

Master3

	ID		FQDN
	i-0f881aaa98f335b2e	Running	sko-demo2-master36.adar-sko.xcu2-8y8x.dev.cldr.work

Master1

	ID		FQDN
	i-06a44e603e2e93ab2	Running	sko-demo2-master12.adar-sko.xcu2-8y8x.dev.cldr.work

Related Information

AWS Onboarding Quickstart

Azure Onboarding Quickstart

Understanding roles and resource roles

CDP workload user

Creating your first Flow Management cluster

Create the Kudu target table

Before you can ingest data into Kudu, you need a target Kudu table prepared to receive the data flow.

Procedure

1. Navigate to your Real Time Data Mart cluster and click Hue from the Services pane.
2. Click the Tables icon on the left pane.
3. Select the default database, and click + New to create a new table.
4. In the Type field, select Manually and click Next.
5. Provide the table Name, Format, Primary keys, and any partitions.
6. Click Submit.

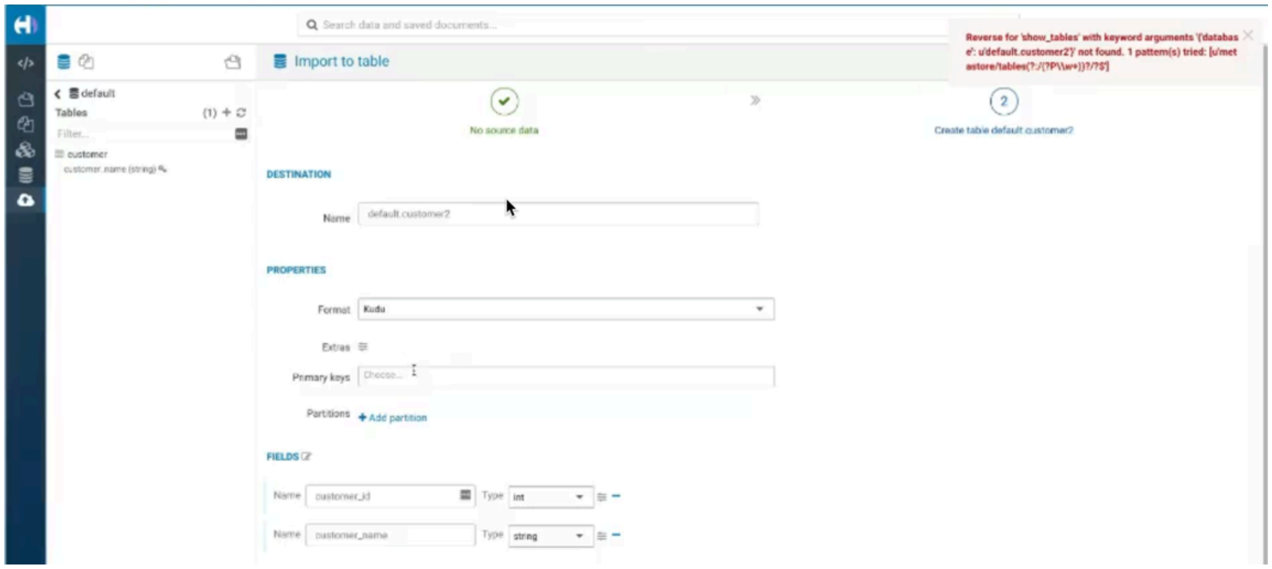
Results

The newly created table displays in the default database Tables pane.

Example

In this example, you create a table with the following values:

Field name	Value
Name	customer
Format	Kudu
Primary keys	customer_id
Partitions	No specified partitions
+ Add Field to include custom field valuecustomer_id	Type equals int
+ Add Field to include a custom field value customer_name	Type equals string


Reverse for 'show_tables' with keyword arguments '{'database': u'default.customer2'}' not found. 1 pattern(s) tried: [u'metastore/tables(?:/(?P\\w+))?/?$']

Import to table

No source data

Create table default.customer2

DESTINATION

Name: default.customer2

PROPERTIES

Format: Kudu

Extras

Primary keys: Choose...

Partitions: + Add partition

FIELDS

Name	Type
customer_id	int
customer_name	string

What to do next

Check the Kudu UI Tables tab, for the name of the table you created. You will need this table name later on, when configuring the NiFi `PutKudu` processor.

Depending on how you created it, it has one of the following formats.

- impala::default.customer
- default.customer

Related Information

Creating Kudu tables

Build the data flow

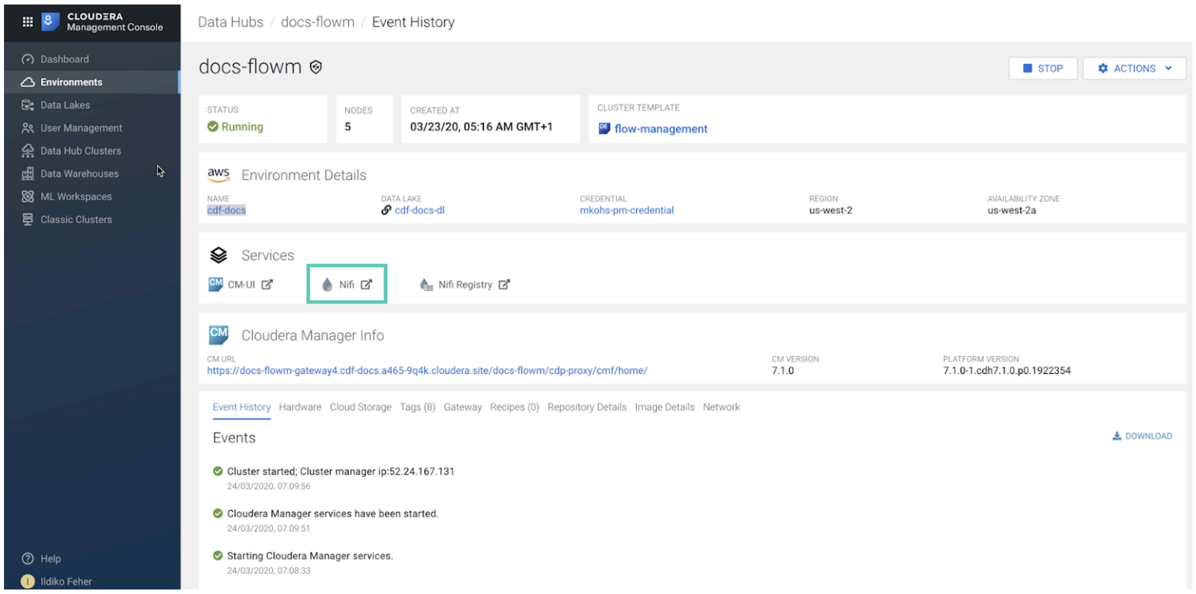
From the Apache NiFi canvas, set up the elements of your data flow. This involves opening NiFi in CDP Public Cloud, adding processors to your NiFi canvas, and connecting the processors.

About this task

You should use the `PutKudu` processor to build your Kudu ingest data flows.

Procedure

1. Open NiFi in CDP Public Cloud.
   a) To access the NiFi service in your Flow Management cluster, navigate to Management Console service > Data Hub Clusters.
   b) Click the tile representing the Flow Management cluster with which you want to work.
   c) Click the NiFi icon in the Services section of the Cluster overview page to access the NiFi UI.



docs-flowm

STATUS: Running
NODES: 5
CREATED AT: 03/23/20, 05:16 AM GMT+1
CLUSTER TEMPLATE: flow-management

Environment Details
NAME: cdf-docs
DATA LAKE: cdf-docs-dl
CREDENTIAL: mkohs-pm-credential
REGION: us-west-2
AVAILABILITY ZONE: us-west-2a

Cloudera Manager Info
CM URL: https://docs-flowm-gateway4.cdf-docs.a465-9q4k.cloudera.site/docs-flowm/cdp-proxy/cmf/home/
CM VERSION: 7.1.0
PLATFORM VERSION: 7.1.0-1.cdh7.1.0.p0.1922354

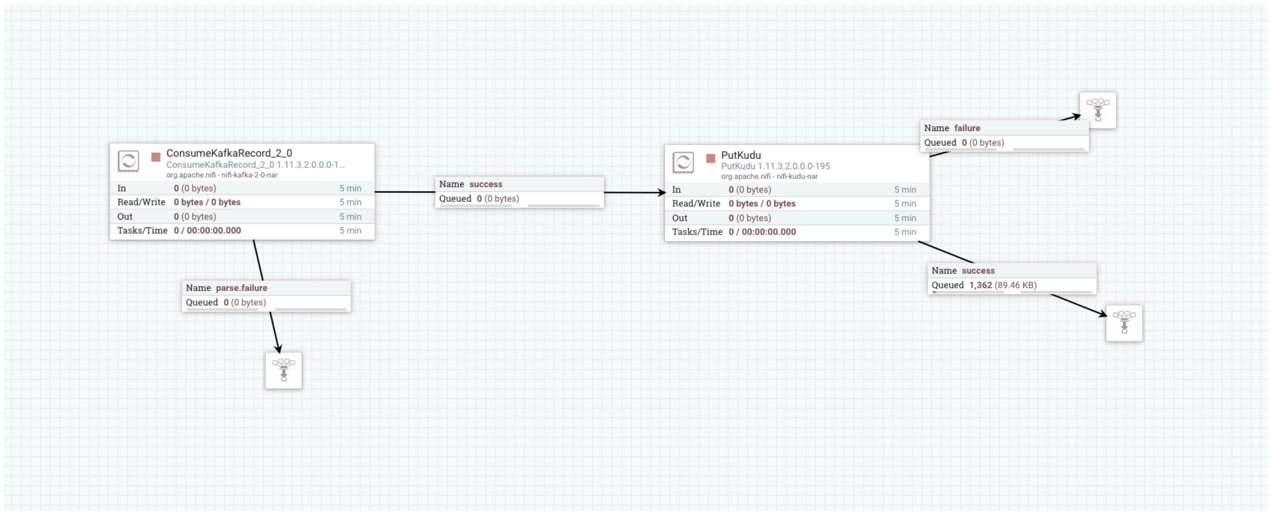
Events
Cluster started; Cluster manager ip:52.24.167.131
24/03/2020, 07:09:56
Cloudera Manager services have been started.
24/03/2020, 07:09:51
Starting Cloudera Manager services.
24/03/2020, 07:08:33

2. Add the NiFi Processors to your canvas.
   a) Select the Processor icon from the Cloudera Flow Management actions pane, and drag a processor to the Canvas.
   b) Use the Add Processor filter box to search for the processor you want to add, and then click Add.
   c) Add each of the processors you want to use for your data flow.
3. Connect the two processors to create a flow.
   a) Click the connection icon in the first processor, and drag it to the second processor.
   b) A Create Connection dialog displays. It has Details and Settings tabs. You can configure the connection's name, FlowFile expiration time period, thresholds for back pressure, load balance strategy, and prioritization.
   c) Click Add to close the dialog box and add the connection to your flow.

      Optionally, you can add success and failure funnels to your data flow, which help you see where flow files are routed when your data flow is running.

Results

Your data flow may look similar to the following:


ConsumeKafkaRecord_2_0
ConsumeKafkaRecord_2_0 1.11.3.2.0.0.0-1...
org.apache.nifi - nifi-kafka-2-0-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min
Name success
Queued 0 (0 bytes)
Name parse.failure
Queued 0 (0 bytes)
PutKudu
PutKudu 1.11.3.2.0.0.0-195
org.apache.nifi - nifi-kudu-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min
Name failure
Queued 0 (0 bytes)
Name success
Queued 1,362 (89.46 KB)

What to do next

Create the Controller Service for your data flow. You will need these services later on as you configure your data flow target processor.

Related Information

Ingesting Data into Apache Kafka in CDP Public Cloud

Consume KafkaRecord_2_0

PutKudu

Building a data flow

Configure the Controller Service

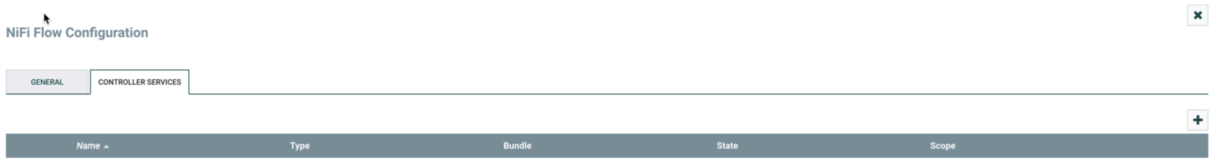
You can add Controller Services to provide shared services to be used by the processors in your data flow. You will use these Controller Services later when you configure your processors.

About this task

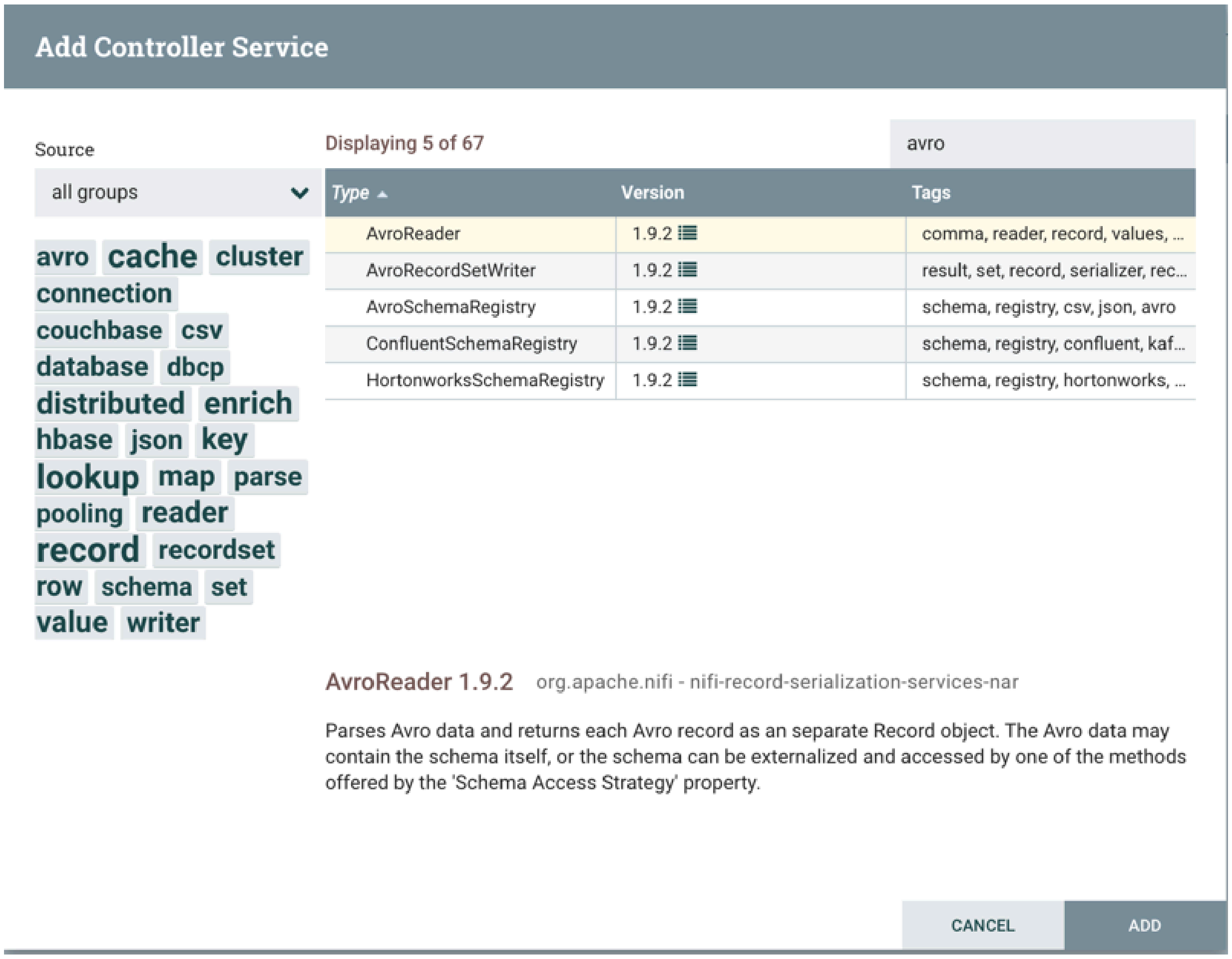
You must define the Controller Services for the processors in your data flow in the onfiguration of the root process group where they will be used.

Procedure

1. To add a Controller Service to your flow, right-click on the Canvas and select Configure from the pop-up menu. This displays the Controller Services Configuration window.
2. Select the Controller Services tab.
3. Click the + button to display the Add Controller Service dialog.



4. Select the required Controller Service and click Add.



Add Controller Service

Source: all groups

Displaying 5 of 67

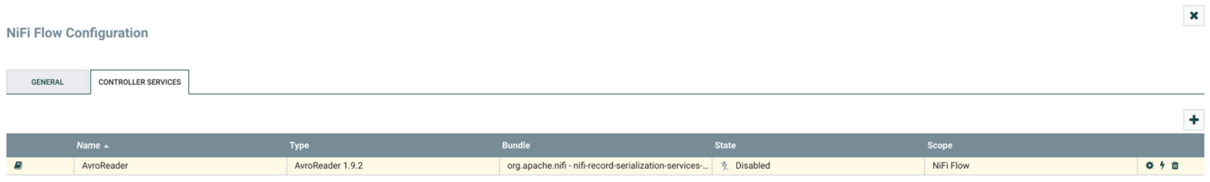
avro

Type	Version	Tags
AvroReader	1.9.2	comma, reader, record, values, ...
AvroRecordSetWriter	1.9.2	result, set, record, serializer, rec...
AvroSchemaRegistry	1.9.2	schema, registry, csv, json, avro
ConfluentSchemaRegistry	1.9.2	schema, registry, confluent, kaf...
HortonworksSchemaRegistry	1.9.2	schema, registry, hortonworks, ...

AvroReader 1.9.2 org.apache.nifi - nifi-record-serialization-services-nar

Parses Avro data and returns each Avro record as an separate Record object. The Avro data may contain the schema itself, or the schema can be externalized and accessed by one of the methods offered by the 'Schema Access Strategy' property.

5. Perform any necessary Controller Service configuration tasks by clicking the Configure icon in the right-hand column.



NiFi Flow Configuration

	Name	Type	Bundle	State	Scope
	AvroReader	AvroReader 1.9.2	org.apache.nifi - nifi-record-serialization-services-...	Disabled	NiFi Flow

6. When you have finished configuring the options you need, save the changes by clicking the Apply button.
7. Enable the Controller Service by clicking the Enable button (flash) in the far-right column of the Controller Services tab.

Configure the processor for your data source

You can set up a data flow to move data from many locations into Apache Kudu. This example assumes that you are configuring `ConsumeKafkaRecord_2_0`. If you are moving data from a location other than Kafka, review *Getting Started with Apache NiFi* for information about how to build a data flow, and about other data consumption processor options.

Before you begin

- Ensure that you have the Ranger policies required to access the Kafka consumer group.
- This use case demonstrates a data flow using data in Kafka, and moving it to Kudu. To review how to ingest data to Kafka, see *Ingesting Data to Apache Kafka*.

Procedure

1. Launch the Configure Processor window, by right-clicking the processor and selecting Configure.
2. Click the Properties tab.
3. Configure `ConsumeKafkaRecord_2_0` with the required values.

   The following table includes a description and example values for the properties required to configure an ingest data flow. For a complete list of `ConsumeKafkaRecord_2_0`, see the *processor documentation*.

Property	Description	Value for ingest to Kudu data flow
Kafka Brokers	Provide a comma-separated list of known Kafka Brokers in the format <host>:<port>.	`Docs-messaging-broker1.cdf-docs.a465-9q4k.cloudera.site:9093, docs-messaging-broker2.cdf-docs.a465-9q4k.cloudera.site:9093, docs-messaging-broker3.cdf-docs.a465-9q4k.cloudera.site:9093`
Topic Name(s)	Enter the name of the Kafka topic from which you want to get data.	*CUSTOMER*
Topic Name Format	Specify whether the topics are a comma separated list of topic names, or a single regular expression.	*NAMES*
Record Reader	Specifies the record reader to use for incoming Flow Files. This can be a range of data formats.	*AVROREADER*
Record Writer	Specifies the format you want to use in order to serialize data before for sending it to the data target. Note: Ensure that the source record writer and the target record reader are using the same Schema.	*AVRORECORDSETWRITER*
Honor Transactions	Specifies whether or not NiFi should honor transactional guarantees when communicating with Kafka.	*FALSE*
Security Protocol	Specifies the protocol used to communicate with brokers. This value corresponds to Kafka's 'security.protocol' property.	*SASL_SSL*
SASL Mechanism	The SASL mechanism to use for authentication. Corresponds to Kafka's 'sasl.mechanism' property.	*PLAIN*
Username	Specify the username when the SASL Mechanism is PLAIN or SCRAM-SHA-256.	*SRV_NIFI-KUDU-INGEST*
Password	Specify the password for the given username when the SASL Mechanism is PLAIN or SCRAM-SHA-256.	*PASSWORD1!*
SSL Context Service	Specifies the SSL Context Service to use for communicating with Kafka.	*DEFAULT*

Property	Description	Value for ingest to Kudu data flow
Group ID	A Group ID is used to identify consumers that are within the same consumer group. Corresponds to Kafka's 'group.id' property.	*NIFI-KUDU-INGEST*

What to do next

Configure the processor for your data target.

Related Information

Getting Started with Apache NiFi

Building a data flow

ConsumeKafkaRecord_2_0

Ingesting data into Apache Kafka in CDP Public Cloud

Configure the processor for your data target

You can set up a data flow to move data into many locations. This example assumes that you are moving data into Apache Kudu using `PutKudu`. If you are moving data into another location, review *Getting Started with Apache NiFi* for information about how to build a data flow, and about other data ingest processor options.

About this task

You can set up a data flow to move data into many locations. This example assumes that you are moving data into Apache Kudu in a Real-time Data Mart cluster, using `PutKudu`. If you are moving data into another location, review *Getting Started with Apache NiFi* for information about how to build a data flow, and about other data ingest processor options.

Before you begin

You have created a machine user in CDP User Management and synchronized this user to your CDP Environment.

Procedure

1. Launch the Configure Processor window, by right-clicking the processor and selecting Configure.
2. Click the Properties tab.
3. Configure `PutKudu` with the required values.

   The following table includes a description and example values for the properties required to configure an ingest data flow into Kudu. For a complete list of `PutKudu`, see the *processor documentation*.

Property	Description	Value for ingest to Kudu data flow
Kudu masters	Comma separated list of the fully qualified domain names (FQDNs) of the Kudu masters to connect to.	These are the values you obtained in *Meet the prerequisites* before you began to build your data flow.
Table name	The name of the Kudu table into which you want to ingest data.	This is the table you created in *Create the Kudu target table*. **Note:** It is important that the specified Kudu table name match the format of the table you created. Depending on how you created it, it has one of the following formats. • impala::default.customer • default.customer

Property	Description	Value for ingest to Kudu data flow
Kerberos Principal	Specify the CDP User name you are using to perform this workflow.	Specify the CDP User Name you created and synced with your CDP Environment in *Meet the prerequisites*.
Kerberos Password	Specify the password for the CDP User you are using to perform this workflow.	For example: Password1!
RecordReader	The service for reading records from incoming flow files.	AvroReader
Kudu Operation Type	Specify operationType for this processor. Valid values are: • INSERT • INSERT_IGNORE • UPSERT • UPDATE • DELETE	UPSERT

Related Information

Getting Started with Apache NiFi

PutKudu

Start your data flow

Start your data flow to verify that you have created a working dataflow and to begin your data ingest process.

Procedure

1. To initiate your data flow, select all the flow components you want to start.
2. Click the Start icon in the Actions toolbar.

   Alternately, right-click a single component and choose Start from the context menu.

Verify that you can write data to Kudu

Once you have configured and started your data flow, you can verify that you are successfully ingesting data into Kudu.

About this task

There are a number of ways to check that data is running through the flow you have built and it actually appears in Hive.

Procedure

1. From your Real-time Data Mart cluster, go to the Impala UI.
2. Search for your data ingest target table.

   For example, in *Create the Kudu target table*, you created a table called default.customer.

   ```
   describe formatted detail.customer;
   select * from default.customer;
   ```

Results

You are able to see data flowing into your default.customer table, with customer_id and customer_name entries.

Next steps

Provides information on what to do once you have moved data into Kudu in CDP Public Cloud.

You have built a simple data flow for an easy way to move data to Kudu. This example data flow enables you to easily design more complex data flows for moving and processing data in Kudu. Here are some ideas for next steps:

- Review *NiFi documentation* to learning more about building and managing data flows.
- Review *NiFi Registry documentation* for more information about versioning data flows.
- Review the *Cloudera Runtime data storage information about Apache Kudu* for information about using Kudu in CDP Public Cloud.

Related Information

NiFi documentation

NiFi Registry documentation

Cloudera Runtime data storage information about Apache Kudu