Cloudera Flow Management 2.1.7

Moving Data to CDW using Iceberg

Date published: 2019-06-26
Date modified: 2024-06-03

Legal Notice

Contents

Moving data into CDW on CDP PvC Data Services using Iceberg table format

You can leverage NiFi data flows to import data into Cloudera Data Warehouse (CDW) on CDP Private Cloud (PvC) Data Services using the Iceberg table format. Follow these steps to build the process.

Understanding the use case

Learn how to use NiFi effectively to move data from various sources into an Iceberg table in a data warehouse on CDP Private Cloud Data Services.

Cloudera Data Warehouse (CDW) can be configured to use the Iceberg table format, optimizing data querying through tools like Impala and Hive. You can use Apache NiFi to move data from a range of locations into a data warehouse cluster running Hive or Impala in CDP Private Cloud Data Services.

This use case guides you through the creation of a data flow that generates FlowFiles containing randomized JSON data and writes this data into an Iceberg table in Hive or Impala. This simple design can get you started with creating an Iceberg ingest data flow. If your specific use case involves a different data source, see the other ingest documents for options and comprehensive guidance on using the appropriate processors.

For more information on Iceberg table format, see *Apache Iceberg features*.

Related Information

Apache Iceberg features

Prerequisites

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

- You have a CDP Private Cloud Data Services environment.
- You have a workload username and password (or keytab) for a service account. This account will be used to move data into the Iceberg table. Make sure that the account has the right permissions in Ranger.
- Your user account is synchronized with the CDP Private Cloud Data Services environment.
- You have a Cloudera Flow Management cluster running Apache NiFi. You will be using it to build your data flow.
- You have a running Data Warehouse cluster in your CDP Private Cloud Data Services environment, and it is properly configured to work with Iceberg tables.

Creating the Iceberg target table in CDW

Before you can ingest data into your Cloudera DataWarehouse (CDW) target in CDP Private Cloud Data Services, make sure that the table where you want to send data with NiFi is already created when you start building your data flow.

The following example shows how to create a table with Hive:

```
CREATE TABLE customer (id int, name string, created_at timestamp)
PARTITIONED BY (country_code string)
STORED BY ICEBERG;
```

Note that Iceberg v2 is also supported. The following example shows how to create a table using Iceberg v2:

```
CREATE TABLE customer (id int, name string, created_at timestamp)
PARTITIONED BY (country_code string)
```

```
STORED BY ICEBERG
TBLPROPERTIES ('format-version' = '2');
```

Building the data flow

Learn how to create a data flow for ingesting data into an Iceberg table in Hive or Impala.

About this task

This process involves leveraging NiFi in your Flow Management cluster. The example data flow you can see below generates FlowFiles containing random JSON data and then writes this data into an Iceberg table.

Procedure

1. Access the NiFi UI using Cloudera Manager.
2. Add the `GenerateFlowFile` processor for data input. This processor creates FlowFiles with random data or custom content.
   a) Drag and drop the processor icon into the canvas. This displays a dialog that allows you to choose the processor you want to add.
   b) Select the `GenerateFlowFile` processor from the list.
   c) Click Add or double-click the required processor type to add it to the canvas.

   You will configure the `GenerateFlowFile` processor to define how to create the sample data in *Configuring the processor for your data source*.
3. Add the `PutIceberg` processor for data output.

   You will configure the `PutIceberg` processor in *Configuring the processor for your data target*.
4. Connect the two processors to create a flow.
   a) Drag the connection icon from the first processor, and drop it on the second processor.

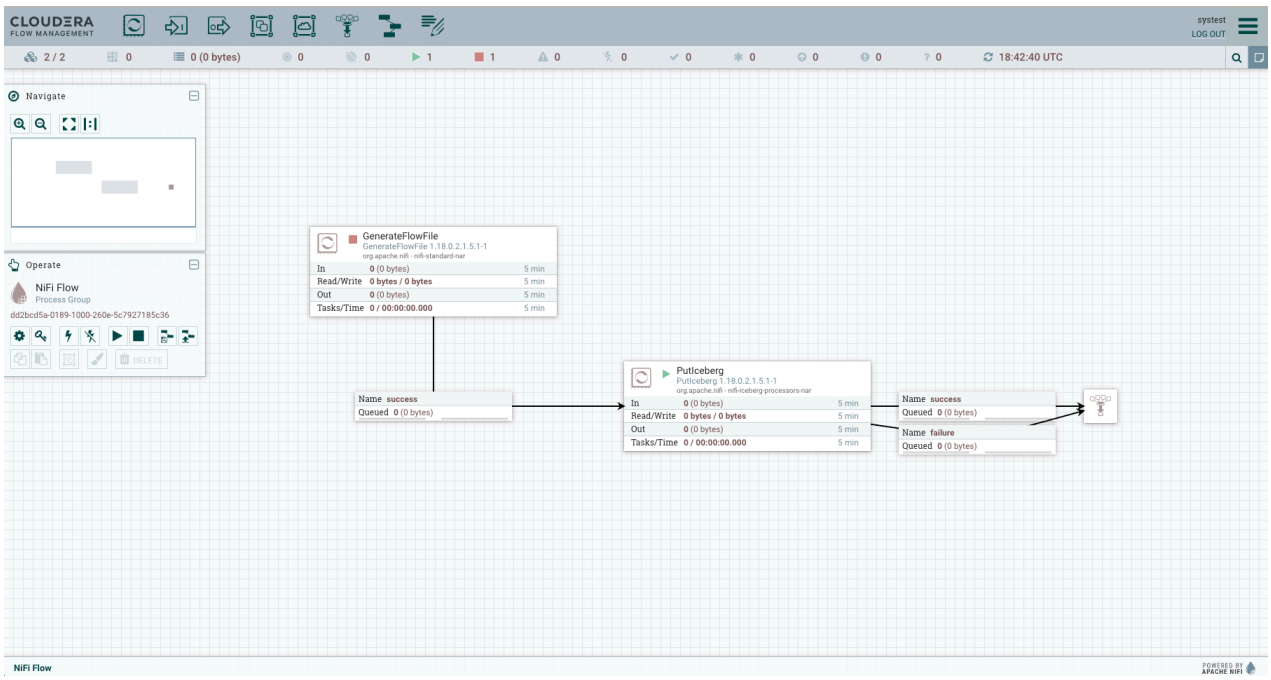
      A Create Connection dialog appears with two tabs: Details and Settings.
   b) Configure the connection.

      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 data flow.
5. You can add success and failure funnels to your data flow, which help you see where flow files are being routed when your flow is running. Connect the `PutIceberg` processor to these funnels with success and failure connections.

   If you want to know more about working with funnels, see the *Apache NiFi User Guide*.

Results

Your data flow will look similar to the following:



What to do next

Create controller services for your data flow. You will need these services when configuring the `PutIceberg` processor.

Related Information

Apache NiFi User Guide

Creating and configuring controller services

Learn how to set up and configure controller services specifically tailored for your CDW Iceberg ingest data flow.

About this task

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.

Procedure

1. To add a controller service to your flow, right-click on the NiFi 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.
5. Click the Configure icon in the right-hand column and configure the options that you need.
6. When you have finished the configuration, click Apply to save the changes.
7. Enable the controller service by clicking the Enable button (flash) in the far-right column of the Controller Services tab.

Example

The following controller services are used in this CDW Iceberg ingest data flow:

- Hive Catalog Controller Service

- Kerberos Password User Service

See the tables below for property details.

Configuring the Hive Catalog Controller Service

In the context of CDW, the Hive Metastore URI is the same as the one for the Hive service of the Base cluster. Rather than specifying the thrift URI directly, a more straightforward approach is to provide the necessary configuration files.

To set up the configuration correctly, you need to supply the following files from your CDP Private Cloud Base environment:

- core-site.xml
- hdfs-site.xml
- hive-site.xml

In Flow Management clusters, these files are made available automatically on every node assuming the Hive gateway role has been added to the NiFi nodes in the Base cluster.

Table 1: Hive Catalog Controller Service properties

Property	Description	Example value for ingest data flow
Hive Metastore URI	Provide the URI of the metastore location.	
Default Warehouse Location	Provide the default warehouse location in the HDFS file system.	
Hadoop Configuration Resources	Add a comma-separated list of Hadoop Configuration files, such as hive-site.xml and core-site.xml for kerberos. Include full paths to the files so that NiFi can refer to those configuration resources from your specified path.	/etc/hive/conf/hive-site.xml,/etc/hadoop/conf/core-site.xml,/etc/hive/conf/hdfs-site.xml

Configuring the Kerberos Password User Service

Use the Kerberos Password User Service so that you do not need to distribute a keytab file across the NiFi nodes of the cluster.

It is best practice to have a dedicated service account created for your specific use case so that you can configure specific policies in Ranger and have better control in case of multi-tenancy with many use cases.

Table 2: Kerberos User Service properties

Property	Description	Example value for ingest data flow
Kerberos Principal	Specify the user name that should be used for authenticating with Kerberos. Use your CDP workload username to set this Authentication property.	srv_nifi_to_iceberg
Kerberos Password	Provide the password that should be used for authenticating with Kerberos. Use your CDP workload password to set this Authentication property.	password (sensitive value)

What to do next

Configure the processor for your data source.

Configuring the processor for your data source

Learn how to configure the GenerateFlowFile data source processor for the CDW Iceberg ingest data flow.

About this task

You can set up a data flow to move data in Iceberg table format into Cloudera Data Warehouse (CDW) from many different locations. This example assumes that you are using sample data generated by the GenerateFlowFile processor.

Procedure

1. Launch the Configure Processor window by right clicking the `GenerateFlowFile` processor and selecting Configure. A configuration dialog with the following tabs is displayed: Settings, Scheduling, Properties, and Comments.
2. Configure the processor according to the behavior you expect in your data flow.

   The `GenerateFlowFile` processor can create many FlowFiles very quickly. Setting the run schedule to a reasonable value is important so that the flow does not overwhelm the system.
3. When you have finished configuring the options you need, save the changes by clicking the Apply button.

   Make sure that you set all required properties, because you cannot start the processor until all mandatory properties have been configured.

Example

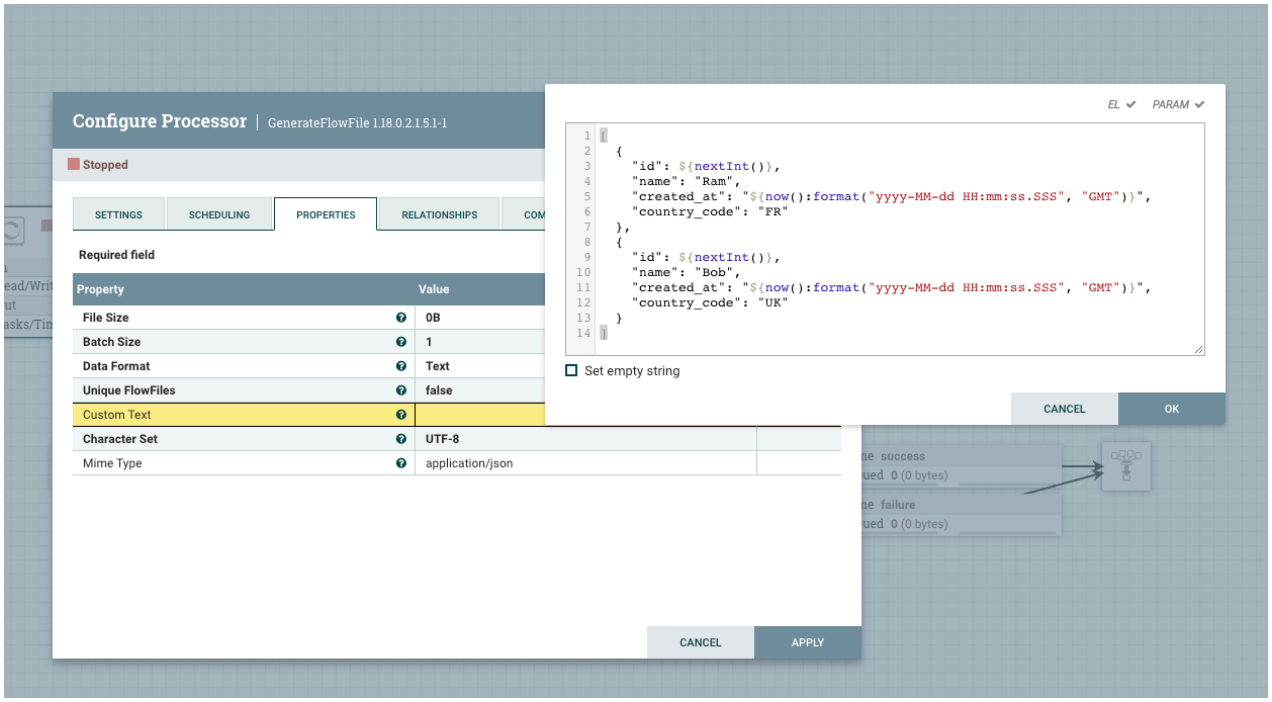
The following settings and properties are used in this example:

Table 3: GenerateFlowFile processor scheduling

Scheduling	Description	Example value for ingest data flow
Run Schedule	Run schedule dictates how often the processor should be scheduled to run. The valid values for this field depend on the selected Scheduling Strategy.	500 ms

Table 4: GenerateFlowFile processor properties

	Description	Example value for ingest data flow
Custom text	If Data Format is text and if Unique FlowFiles is false, you can provide custom to be used as the content of the generated FlowFiles. The expression statement in the example value generates a sequential ID, the current timestamp and some hard coded name and country code.	`[ { "id": ${nextInt()}, "name": "Ram", "created_at": "${now():format("yyyy-MM-dd HH:mm:ss.SSS", "GMT")}", "country_code": "FR" }, { "id": ${nextInt()}, "name": "Bob", "created_at": "${now():format("yyyy-MM-dd HH:mm:ss.SSS", "GMT")}", "country_code": "UK" } ]`



What to do next

Configure the processor for your data target.

Configuring the processor for your data target

Learn how to configure the target processor of the CDW Iceberg ingest data flow.

About this task

This data flow assumes that you are moving data to Cloudera Data Warehouse (CDW). Once your data is collected and available in your flow, you can use the PutIceberg processor to send the data into the CDW table.

Procedure

1. Launch the Configure Processor window, by right-clicking the `PutIceberg` processor and selecting Configure.

   A configuration dialog with the following tabs is displayed: Settings, Scheduling, Properties, and Comments.
2. Configure the processor according to the behavior you expect in your data flow.
3. When you have finished configuring the options you need, click Apply to save the changes.

   Make sure that you set all required properties, because you cannot start the processor until all mandatory properties are configured.

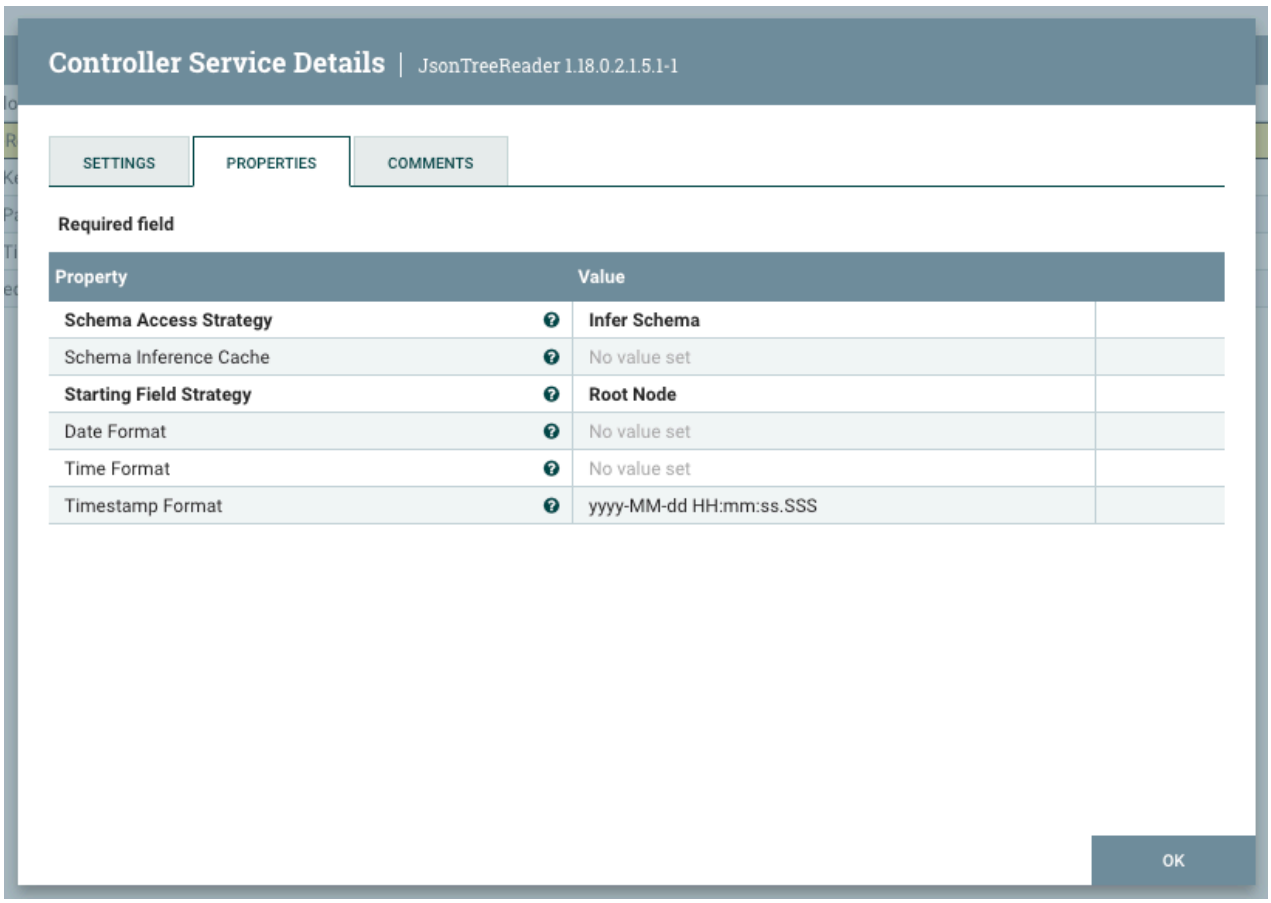
Example

The following settings and properties are used in this example:

Table 5: PutIceberg processor properties

Property	Description	Example value for ingest data flow
Record Reader	Select the Record Reader of your choice based on the format of your incoming data that should be pushed in the table.	JSON Reader
Catalog Service	Add the Hive Catalog Controller Service you have created.	Hive Catalog Service
Catalog Namespace	Set the Catalog Namespace property to the name of the database where you created the destination table.	default
Table Name	Set the Table Name property to the name of the table previously created.	customer
Kerberos User Service	Add the Kerberos User Service of your choice (based on username/password or based on username/keytab) that you have created.	Kerberos Password User Service

Note that the JSON reader provides the expected format for timestamp fields:



What to do next

Your data flow is ready to ingest data into an Iceberg table in CDW. You can now start the flow.

Starting the data flow

Learn how run the flow and verify that data is successfully transferred to the Iceberg table.

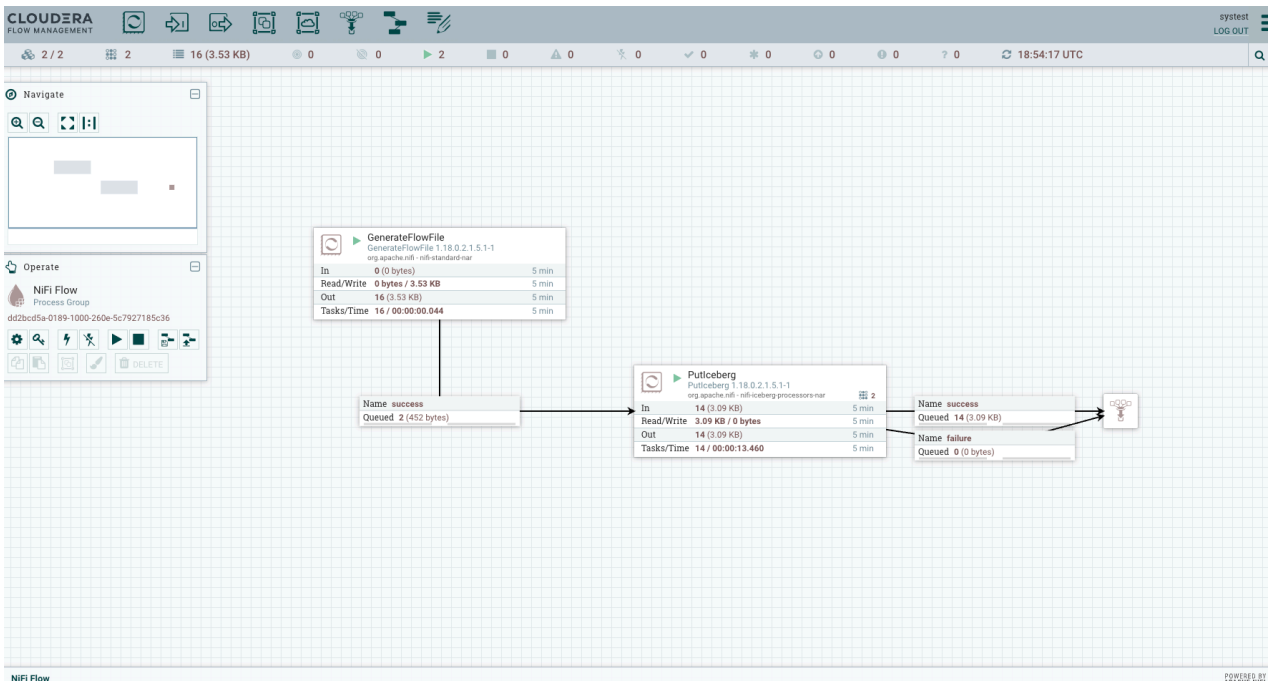
Procedure

1. Select all the data flow components you want to start.
2. Click the Start icon in the Actions toolbar.

   Alternatively, you can right-click an individual component and choose Start from the context menu.

Results

Your data flow should be running without any errors. Data should be generated in the form of FlowFiles, and the files should be written to your Iceberg-formatted table.



To verify the operation of your Iceberg ingest data flow:

- Check that data is running through the flow that you have built and it actually appears in the target table.
- Monitor the NiFi processors to ensure they are not encountering any errors during the data transfer process.
- Use the NiFi user interface to examine the volume of data passing through processors. You can also access status history by right-clicking the processors or connections.
- Query and analyze the ingested data using Hue.