Cloudera Flow Management 2.1.3

Moving data in and out of Snowflake

Date published: 2019-06-26
Date modified: 2022-02-07

https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Pushing data to and moving data from Snowflake using Apache NiFi..... 4

Moving data out of Snowflake..... 4

Before you begin..... 4
Downloading the Snowflake JDBC driver jar file.....4
Adding Snowflake CA certificates to NiFi truststore.....5
Building your dataflow.....6
Creating Controller Services for your dataflow.....7
Configuring your source processor..... 8
Configuring your target processor.....9
Confirming your dataflow success.....10

Pushing data into Snowflake..... 10

Before you begin..... 10
Adding Snowflake CA certificates to NiFi truststore.....11
Building your dataflow.....11
Configuring your Controller Services..... 13
Configure your source processor.....14
Configuring your target processor.....15
Confirming your dataflow success.....15

Next steps.....16

Pushing data to and moving data from Snowflake using Apache NiFi

Cloudera supports pulling data out of Snowflake and pushing data into Snowflake, using Apache NiFi. You can use Apache NiFi to move data back and forth between Snowflake and other systems including Cloudera Data Platform.

Moving data out of Snowflake

You can create a NiFi dataflow to move data out of Snowflake. To do this, you must meet some prerequisites, download the Snowflake JDBC driver jar file, update the NiFi truststore, configure your Controller Services, build the dataflow, and configure the source and target Processors.

Pushing data into Snowflake

You can create a NiFi data to push data into Snowflake. To do this, you must meet some prerequisites, configure your Controller Services, build the dataflow, and configure your source and target Processors.

Moving data out of Snowflake

You can create a NiFi dataflow to move data out of Snowflake. To do this, you must meet some prerequisites, download the Snowflake JDBC driver JAR file, update the NiFi truststore, configure your Controller Services, build your dataflow, and configure the source and target Processors.

Before you begin

Before setting up a NiFi dataflow to pull data from a Snowflake database table, you must meet certain minimum prerequisites.

- You have a CFM cluster running NiFi either in CDP Public Cloud, CDP Private Cloud Base, or as standalone NiFi cluster.

  See the *Related Information* section below for installation instructions appropriate for your deployment objectives.

  - *Creating your First Flow Management Cluster in CDP Public Cloud*
  - *Cloudera Flow Management Deployment in CDP Private Cloud Base*
  - *Cloudera Flow Management Component Installation and Upgrade*
- You have a Snowflake account.
- You have read access to a Snowflake database table.

Related Information

Creating your First Flow Management Cluster in CDP Public Cloud

Cloudera Flow Management Deployment in CDP Private Cloud Base

Cloudera Flow Management Component Installation and Upgrade

Downloading the Snowflake JDBC driver jar file

Before you can create a dataflow that moves data out of a Snowflake database, you must ensure that NiFi can interact with the Snowflake database using a JDBC interface. To do this, you must download the Snowflake JDBC driver JAR file, upload it to each NiFi node in your cluster, and ensure that the proper permissions are set.

About this task

You need the Snowflake driver file, when defining the DBCP Connection Pool Controller Service that is used in the NiFi processors, in the CFM dataflow that moves data out of Snowflake. This driver allows NiFi to interact with the Snowflake database through the JDBC interface.

Before you begin

- You must have reviewed and met the prerequisites.
- You must have logged in as an admin user.
- You must have created a directory on all the NiFi nodes where you will make the driver file available for NiFi

Procedure

1. Download the latest Snowflake JDBC JAR file from the Snowflake website.

   For information on how to get the Snowflake JDBC JAR file, see the Snowflake documentation in the *Related information* section below.
2. Upload the JAR file to each of the NiFi nodes in your cluster.
3. As a root user, on every NiFi node in your cluster, update the permissions for the Snowflake JDBC JAR file, and make sure this can be accessed and read by the `nifi` user:

   ```
   chown nifi: <snowflake-jdbc>.jar
   ```

What to do next

After you have downloaded the Snowflake JDBC driver , proceed to the following steps.

- Add Snowflake CA certificates to the NiFi truststore
- Build the dataflow.
- Create the Controller Services for your dataflow.
- Configure your source Processor.
- Configure your target Processor.
- Confirm your dataflow is successful

Related Information

Downloading / Integrating the JDBC Driver

Adding Snowflake CA certificates to NiFi truststore

You must ensure that NiFi can communicate securely with Snowflake. To do this, configure NiFi to trust the Snowflake Certificate Authority (CA) by merging the default Snowflake JDK truststore content into the NiFi truststore.

About this task

The Snowflake endpoints have certificates signed by a Certificate Authority (CA). You must configure NiFi so that the Snowflake CAs are trusted by NiFi. The recommended approach is to follow the steps documented in *How to Add Root and Intermediate CAs to Truststore for TLS/SSL*. See the link in the *Related information* below.

Another approach is to merge the content of the default truststore of the JDK into the NiFi truststore. This approach is describe here.

Note:

If you have added the Snowflake CA into the NiFi truststore, you may skip this section. For example, this is the case if you are using Flow Management clusters in CDP Public Cloud.

Before you begin

You must have downloaded the Snowflake JDBC driver jar file.

Procedure

1. Merge the content of the JDK truststore (represented by the file cacerts), using the below command:

```
keytool
-importkeystore
-srckeystore [***path to cacerts***]
-destkeystore [***path to NiFi truststore***]
```

2. When prompted, enter the password of the JDK truststore. The default is changeit.
3. Enter the NiFi truststore password.

What to do next

If you have changed the truststore, you must restart NiFi before the changes are taken into account.

After you have finished adding Snowflake certificates to the NiFi truststores, you may move on to the following steps.

- Build the dataflow.
- Create the Controller Services for your dataflow.
- Configure your source Processor.
- Configure your target Processor.
- Confirm your dataflow is successful.

Related Information

How to Add Root and Intermediate CAs to Truststore for TLS/SSL

Building your dataflow

Set up the elements of your NiFi dataflow that enables you to move data out of Snowflake using Apache NiFi. This involves opening NiFi in CDP Public Cloud, adding processors to your NiFi canvas, and connecting the processors.

About this task

When you are building a data flow to move data out of Snowflake using Apache NiFi, you can consider using the following processors to build your dataflow:

- ListDatabaseTables
- ExecuteSQLRecord

Before you begin

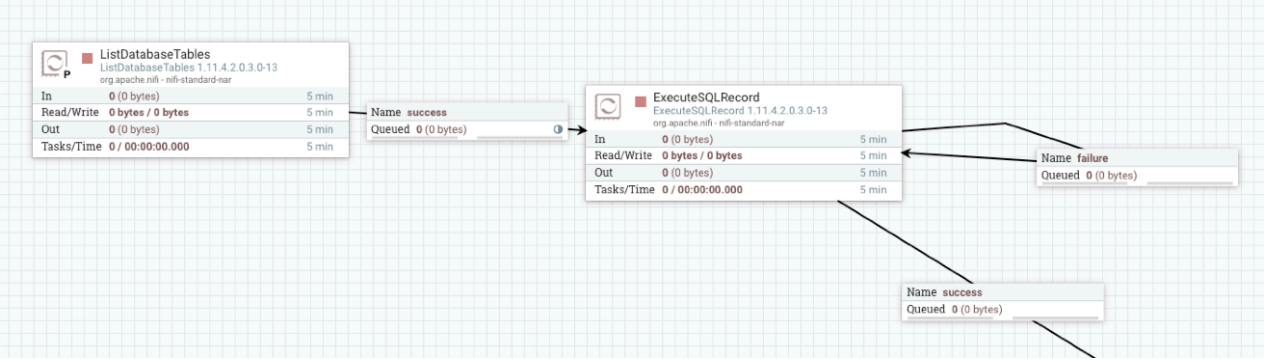
You must have added the Snowflake CA certificates to the NiFi truststore.

Procedure

1. Launch NiFi from your CDP Public Cloud or CDP Private Cloud Base cluster.
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 dataflow.
   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. 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



What to do next

Once you have finished building the dataflow, move on to the following steps:

- Create Controller Services for your dataflow.
- Configure your source Processor.
- Configure your target Processor.

Creating Controller Services for your dataflow

You can add Controller Services that can provide shared services to be used by the processors in your dataflow. Create them after you build the NiFi dataflow and before you configure the Processors, so that they are available when you configure your NiFi Processors.

About this task

For this dataflow, create two Controller Services.

- `AvroWriter`
- `DBCPConnectionPool`

See the *Related Information* section below, for links to complete details about these Apache NiFi Controller Services.

Before you begin

You must have built your dataflow.

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 + Add 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 necessary options.
6. Click Apply to save the changes.
7. Click the Enable button (flash) in the far-right column of the Controller Services tab to enable the Controller Service.

Example

You require the following Controller Services for this example of a NiFi dataflow moving data out of Snowflake.

AvroRecordSetWriter

You use this Controller Service as the writer, when writing the data as Flow Files in NiFi. Avro is a good option in most cases but you may want to choose another writer depending on your requirements.

In general, you may accept the default configurations.

DBCPConnectionPool

This Controller Service allows NiFi to interact with the Snowflake database through the JDBC interface.

Property	Description	Example value for a dataflow pulling data from Snowflake
Name		SnowflakeJDBCConnectionPool
Database Connection URL		`jdbc:snowflake://[***account_name***].snowflakecomputing.com/?[***connection_params***]` Where: • account_name is similar to cb56215.europe-west2.gcp • connection_parameters is similar to db=DEMO_DB
Database Driver Class Name		net.snowflake.client.jdbc.SnowflakeDriver

What to do next

After you have finished configuring your Controller Services:

- Configure your source Processor
- Configure your target Processor
- Confirm your dataflow is running

Related Information

AvroRecordSetWriter
DBCPConnectionPool

Configuring your source processor

You can use the `ListDatabaseTable` processor to get data from your Snowflake table. To do this, launch the Configure Processor window, specify the necessary configurations, and start the process to verify that you can view the Snowflake table.

About this task

Use the ListDatabaseTables processor to get data from Snowflake database tables.

Set the proper Scheduling Strategy on the processor in accordance with your use case. You use this processor to list the tables available in the remote database. You can update the value of the Refresh Interval property in combination with the Scheduling Strategy to determine at which frequency tables will be listed.

Configure this processor to run only on the primary node of your NiFi cluster.

Before you begin

- You must have built the dataflow.
- You must have created your Controller Services.

Procedure

1. Launch the Configure Process window, by right-clicking the Processor and selecting Configure.
2. Configure ListDatabaseTables according to your use case and operational objectives.
3. Configure Database Connection Pooling Service to call the database connection Controller Service you created in the preceding step.
4. You can use the Scheduling Strategy and the Refresh Interval configuration properties to determine how frequently the tables will be listed in the Snowflake database.
5. Use the Table Name Pattern value to specify whether you want to pull all the tables or just some.

   If you want to retreive a few of the tables, define a regular expression. If the property value is not set, all tables are retrieved.

Results

You can confirm that you have configured ListDatabaseTables correctly by starting the processor and confirming that you can view the Snowflake table.

What to do next

After you have finished configuring your source Processor, proceed to:

- Configure your target Processor
- Confirm your dataflow is running

Related Information

ListDatabaseTables

Configuring your target processor

In a dataflow that is pulling data from a Snowflake database, configure the `ExecuteSQLRecord` processor to handle data pooling from remote tables. To do this, launch the processor configuration window and provide the configurations appropriate for your use case.

About this task

Configure ExecuteSQLRecord.

The ExecuteSQLRecord processor handles data pooling from the remote tables. To load balance the execution of the SQL queries across the NiFi nodes, you should update the configuration of the relationship between the ListDatabase Tables processor and the ExecuteSQLRecord processor and define a round robin load balancing strategy. This way the flow files generated on the primary node by the ListDatabaseTables processors will be shuffled across the nodes of the NiFi cluster to distribute the workload.

Procedure

1. Right-click `ExecuteSQLRecord` and click Configure Processor.
2. For Database Connection Pooling Service, select your DBCPConnectionPool controller service you defined earlier.
3. For SQL select query, if you want to extract all the data from the remote tables, you can use the SQL query:

```
SELECT * FROM ${db.table.fullname}
```

4. For Record Writer, select the Avro writer controller service you defined earlier.
5. For Normalize Table/Column Names, you may want to set this property to true in case some of the column names are not compatible with Avro specifications.

What to do next

After you have configured your target processor, proceed to confirming your dataflow success.

Related Information

ExecuteSQLRecord

Confirming your dataflow success

Confirm that you have successfully built a dataflow to move data out of Snowflake database tables by starting your dataflow and verifying that data is moving through it.

About this task

In a real life use case, you may want to connect your `ExecuteSQLRecord` processor with a processor that can send your data to an external system. For details about ingesting data into CDP Public Cloud clusters or cloud provider storage solutions, see the *Related Information* section below.

Procedure

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

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

Results

Your flow should run without any errors. Data should be pulled from the Snowflake database tables using `ListDatabaseTables` and pooled using `ExecuteSQLRecord`.

Related Information

Ingesting data in CDP Public Cloud clusters

Pushing data into Snowflake

You can create a NiFi data flow to push data into a Snowflake database table. To do this, you must meet certain prerequisites, configure your Controller Services, build the dataflow, and configure your source and target processors.

Before you begin

Before setting up a NiFi dataflow to push data to a Snowflake database table, you must meet certain minimum prerequisites.

- You have a CFM cluster running NiFi.
- You have a Snowflake account.
- Your Snowflake user account has write access to a Snowflake database table.
- Set up your destination Snowflake table with the right columns and columns type.
- Table columns for this example are R_REGIONKEY, R_NAME, and R_COMMENT

Adding Snowflake CA certificates to NiFi truststore

You must ensure that NiFi can communicate securely with Snowflake. To do this, configure NiFi to trust the Snowflake Certificate Authority (CA) by merging the default Snowflake JDK truststore content into the NiFi truststore.

About this task

The Snowflake endpoints have certificates signed by a Certificate Authority (CA). You must configure NiFi so that the Snowflake CAs are trusted by NiFi. The recommended approach is to follow the steps documented in *How to Add Root and Intermediate CAs to Truststore for TLS/SSL*. See the link in the *Related information* below.

Another approach is to merge the content of the default truststore of the JDK into the NiFi truststore. This approach is described here.

Note:

If you have added the Snowflake CA into the NiFi truststore, you may skip this section. For example, this is the case if you are using Flow Management clusters in CDP Public Cloud.

Before you begin

You have reviewed and met the prerequisites

Procedure

1. Merge the content of the JDK truststore (represented by the file cacerts), you can use the below command:

```
keytool
-importkeystore
-srckeystore [***path to cacerts***]
-destkeystore [***path to NiFi truststore***]
```

2. When prompted, enter the password of the JDK truststore. The default is changeit.
3. Enter the NiFi truststore password.

What to do next

If you have changed the truststore, you must restart NiFi before the changes are taken into account.

After you have finished adding Snowflake certificates to the NiFi truststores, you may move on to the following steps.

- Build the dataflow.
- Create the Controller Services for your dataflow.
- Configure your source Processor.
- Configure your target Processor.
- Confirm your dataflow is successful.

Related Information

How to Add Root and Intermediate CAs to Truststore for TLS/SSL

Building your dataflow

Set up the elements of your NiFi dataflow that enables you to move data into Snowflake using Apache NiFi. This involves opening NiFi in CDP Public Cloud, adding processors to your NiFi canvas, and connecting the processors.

About this task

When you are building a data flow to move data into Snowflake using Apache NiFi, you can consider using the following processors to build your dataflow:

- GenerateFlowFile
- PutDatabaseRecord

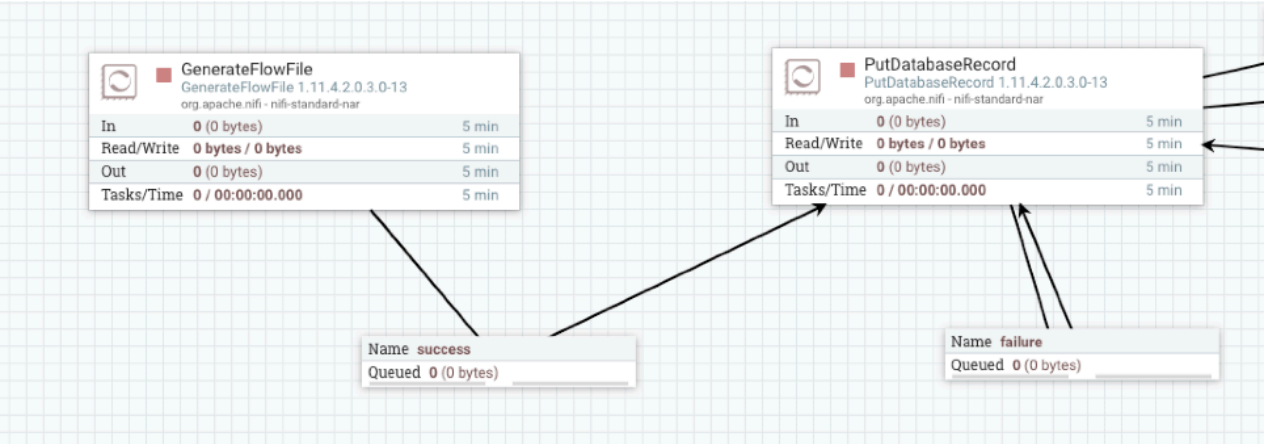
Before you begin

You must have reviewed and met the prerequisites.

Procedure

1. Launch the NiFi from your CDP Public Cloud or CDP Private Cloud Base cluster.
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 dataflow.
   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



What to do next

Once you have finished building the dataflow, move on to the following steps:

- Create Controller Services for your dataflow.
- Configure your source Processor.
- Configure your target Processor.
- Confirm your dataflow success

Configuring your Controller Services

You can add Controller Services that can provide shared services to be used by the Processors in your dataflow. Create them after you build the NiFi dataflow and before you configure the Processors, so that they are available when you configure your NiFi Processors.

Before you begin

You have built the dataflow on the NiFi canvas.

About this task

For this dataflow, create two Controller Services.

- `DBCPConnectionPool`
- `CSVReader`

See the *Related Information* section below, for links to full details about these Apache NiFi Controller Services.

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.
5. Click the Configure icon in the right-hand column and configure the necessary options.
6. When you have finished configuring the options you need, click Apply to save the changes.
7. Click the Enable button (flash) in the far-right column of the Controller Services tab to enable the Controller Service.

Example

DBCPConnectionPool

Reuse the same `DBCPConnectionPool` Controller Service that you created before to interact with the Snowflake database.

Property	Description	Example value for Snowflake ingest dataflow
Name		Rename to SnowflakeJDBCConnectionPool
Database Connection URL		`jdbc:snowflake://<account_name>.snowflakecomputing.com/?<connection_params>` Where: • <account_name> is similar to cb56215.europe-west2.gcp • <connection_parameters> is similar to db=DEMO_DB
Database Driver Class Name		net.snowflake.client.jdbc.SnowflakeDriver

CSVReader

In this example, you are generating CSV data to ingest into the Snowflake remote database table. As a result, you must also configure a `CSVReader` Controller Service to parse the data. You require the CSV data to correctly construct the queries executed against the table to ingest the data.

You can accept the default configurations.

What to do next

Once you have configured your Controller Services, proceed to the following tasks:

- Configure your source Processor.
- Configure your target Processor.
- Confirm your dataflow success

Related Information

DBCPConnectionPool

CSVReader

Configure your source processor

You can set up a dataflow to push data into Snowflake database tables from many different locations. To do this, start by configuring the processor for your data source by launching the Configure Processor window and specifying the necessary configurations.

About this task

Configure the `GenerateFlowFile` processor to create random data for this example dataflow. `GenerateFlowFile` is useful when you are testing or creating proof of concept dataflows. When you have confirmed that this dataflow meets your business use case, you can replace it with a processor getting data from your actual data source.

See *Related Information* for full details on this Apache NiFi Processor.

Before you begin

- You must have built the dataflow.
- You must have configured your Controller Services.

Procedure

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

   See the *Example* section below for recommended configuration to satisfy this example use case.
3. Save the changes by clicking Apply .

Example

The following settings and properties are used in this example:

Table 1: 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.	60 s

Table 2: GenerateFlowFile processor properties

<Title>?	Description	Example value for ingest data flow

Custom text	If the value of Data Format is text and if Unique FlowFiles is set to false, you can provide custom to be used as the content of the generated FlowFiles.
	The expression statement in the example value generates a random ID between 1 and 10 000, with random last names assigned.
		`R_REGIONKEY, R_NAME, R_COMMENT`
		`100,foo1, blablabla`
		`101, foo2, blabla`
		`102, foo3, bla`

What to do next

Once you have configured your Controller Services, proceed to the following tasks:

- Configure your target Processor.
- Confirm your dataflow success.

Related Information

GenerateFlowFile

Configuring your target processor

In a dataflow pushing data to a Snowflake database table, configure the `PutDatabaseRecord` processor to move data into the Snowflake database table. To do this, launch the processor configuration window and provide the configurations appropriate for your use case.

About this task

Configure `PutDatabaseRecord` Processor. See *Related Information* for complete details on this Apache NiFi processor.

Procedure

1. Right-click `PutDatabaseRecord` and click Configure Processor.
2. In the Record Reader field, specify the CSV Reader Controller Service you created in the earlier step.
3. In the Statement Type field, select INSERT to insert data into the remote database table.
4. In the Database Connection Pooling Service field, select the DBCP Connection Pool Controller Service you created previously.
5. In the Table Name field, specify the name of your remote Snowflake database table. In this example, it is NIFI TEST
6. (Optional) Depending on your requirements, you may want to auto-terminate the success relationship of the processor, and have a self-connecting loop for the failure relationship.

What to do next

After you have configured your target processor, proceed to confirming your dataflow success.

Related Information

PutDatabaseRecord

Confirming your dataflow success

Confirm that you have successfully built a dataflow to push data into Snowflake database tables by starting your dataflow and verifying that data is moving through it.

Procedure

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

   Alternatively, right-click a single component and choose Start from the context menu.
3. Go to the Snowflake web interface and confirm that data is correctly ingested into your target table.

Next steps

You now have all the required information to create your NiFi flows to answer your own use cases interacting with the Snowflake data warehousing solution. You can get NiFi up and running for a few minutes using CDP Public Cloud and have flows taking care of moving data back and forth between Snowflake and other systems easily and efficiently.