Cloudera DataFlow for Data Hub 7.3.1

Ingesting data into Apache Hive in Cloudera on cloud

Date published: 2019-12-16
Date modified: 2025-06-27

CLOUDERA

Legal Notice

Contents

Ingesting data into Hive

Understand the use case

You can use Apache NiFi to move data from a range of locations into a Data Engineering cluster running Apache Hive in Cloudera on cloud.

This use case walks you through the process of creating a data flow focused on ingesting data from Apache Kafka into Apache Hive. The PutHive3Streaming processor is used to ingest data into a Hive table. This processor interacts with the Hive Metastore to determine the storage location of the Hive table and writes data files directly to that location, eliminating the need for a running HiveServer2. After the delta files are written, the NiFi processor updates the table metadata in the Hive Metastore, ensuring that future queries include the newly ingested data.

> **Note:** Although this process does not require a running HiveServer2 or a cluster for executing Hive queries, Cloudera strongly recommends having a Data Engineering cluster available in the same environment so that it can run Hive's compaction background processes which reconcile the delta files created by NiFi into base files. Without compaction, an excessive number of delta files can slow down Hive queries, as the engine has to reconcile all the delta files during query execution.

If you are moving data from a source other than Kafka, review the *Getting Started with Apache NiFi* for information on building data flows and exploring other data ingestion options. You can also find instructions on building a flow for ingesting data into other Cloudera components.

Related Information

Getting Started with Apache NiFi

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 Cloudera on cloud environment.
- You have a workload username and password set to access Cloudera Data Hub clusters. The predefined resource role of this user is at least "EnvironmentUser". This resource role provides the ability to view Cloudera Data Hub clusters and set the FreeIPA password for the environment.
- Your user is synchronized to the Cloudera on cloud environment.
- You have a Flow Management cluster running Apache NiFi.
- You have a running Data Engineering cluster in the same Cloudera environment.

  > **Note:** Data can be ingested into Hive without an active Data Engineering cluster. However, for production workloads, it is strongly recommended to provision a Data Engineering cluster, which ensures that Hive's automated compaction jobs can run concurrently with NiFi data ingestion, optimizing query performance. For more information, see *Understand the use case*.

- You have created a new database and transactional table for Hive data ingest.
- You have a policy in Ranger that allows a user to write data to your Hive table.

Related Information

Understand the use case

AWS Onboarding Quickstart

Azure Onboarding Quickstart

Understanding roles and resource roles

Cloudera workload user

Setting up your Flow Management cluster

Configure the service account

Configure the Service Account you will use to ingest data into Hive.

Procedure

1. Navigate to the Cloudera Management Console UI.
2. From the User Management tab, create a new machine user.

3. Once the user is created, set a workload password and synchronize the changes to the required environments.

4. From the Access tab, add the machine user as an Environment User to the environment.

Update Resource Roles for nifi-hive-ingest   ✕

Resource Roles

■	Role ⬍	Description
☐	DWAdmin ⓘ	Grants permission to create, delete, and update Cloudera Data Warehouse clusters for a given CDP environment.
☐	DWUser ⓘ	Grants permission to view Cloudera Data Warehouse cluster for a given CDP environment.
☐	EnvironmentAdmin ⓘ	Grants all the rights to an environment.
☑	EnvironmentUser ⓘ	Grants permission to set the workload password for the environment.
☐	MLAdmin ⓘ	Grants permission to create and delete Cloudera Machine Learning workspaces for a given CDP environment. MLAdmins will also have Site Administrator level access to all the workspaces provisioned using this environment. That is, they can run workloads, monitor, and manage all user activity on these workspaces.
☐	MLBusinessUser ⓘ	Grants permission to list Cloudera Machine Learning workspaces for a given CDP environment. MLBusinessUsers will also be able to view shared machine learning applications
☐	MLUser ⓘ	Grants permission to list Cloudera Machine Learning workspaces for a given CDP environment. MLUsers will also be able to run workloads on all the workspaces provisioned using this environment.

Cancel   **Update Roles**

5. Go back to the environments view and click Actions - Synchronize Users to Free IPA.

Results

Once the synchronization is complete, the required user is present in the environment and has the right permissions to access the object store.

Create IDBroker mapping

To enable your Cloudera user to utilize the central authentication features Cloudera provides and to exchange credentials for AWS or Azure access tokens, you have to map this Cloudera user to the correct IAM role or Azure Managed Service Identity (MSI). The option to add/modify these mappings is available from the Cloudera Management Console in your Cloudera environment.

Procedure

1. Go to the environment in which your Flow Management and Data Engineering clusters are running.

2. To access IDBroker Mappings, click Actions | Manage Access and select the IDBroker Mapping tab in the next screen, where you can provide mappings for users or groups.

3. Click Edit.

4. Add a new mapping for your service user, mapping the user to an existing IAM role or Azure Managed Identity Resource ID that has access to the underlying storage which is used by the target Hive table.

For example:

5. Add your Cloudera user and the corresponding AWS or Azure role that provides write access to your folder in your S3 bucket or ADLS folder to the Current Mappings section.

> **Note:**
>
> You can get the AWS IAM role Amazon Resource Name (ARN) from the Roles Summary page in AWS and can copy into the IDBroker role field.
>
> You can get the Azure Managed Identity Resource ID by navigating to Managed Identities->Your Managed Identity->Properties->Resource ID.

6. Click Save and Sync.

7. Ensure that your IDBroker mapping change is synchronized to the environment successfully.

What to do next

Create a Hive table and add Ranger policies that allow your machine user write access to your Hive table.

Related Information

IDBroker

Onboarding Cloudera users and groups for AWS cloud storage

Onboarding Cloudera users and groups for Azure cloud storage

Create the Hive target table

Before you can ingest data into Apache Hive in Cloudera on cloud, ensure that you have a Hive target table. These steps walk you through creating a simple table. Modify these instructions based on your data ingest target table needs.

About this task

You require a transactional Hive table. This is the type of table created by default.

Procedure

1. In the Cloudera Management Console, go to your Data Engineering cluster.
2. From Services, click Data Analytics Studio, and select Compose from the left-navigation pane.
3. Create a new target database.

```
create DATABASE retail
```

4. Create a new target table.

```
create table retail.customer(customer_id int, customer_name string)
```

What to do next

Add policies to Ranger so that the Cloudera user you are using to create this data flow can ingest data into the Hive target table.

Add Ranger policies

Add Ranger policies to ensure that you have write access to your Hive tables.

Procedure

1. From the Ranger cm_hive service in your Data Lake, create a new policy called Database access, and allow select permissions on the database in which your Hive target table is stored.

2. Create another policy to allow select, update, and Write permissions to ingest into the Hive target table.

For example:

Obtain Hive connection details

To enable an Apache NiFi data flow to communicate with Hive, you must obtain the Hive connection details by downloading several client configuration files. The NiFi processors require these files to parse the configuration values and use those values to communicate with Hive.

Procedure

1. From the Services pane in the Data Engineering cluster that contains the Hive instance to which you want to ingest data, click CM-UI.

2. Click Hive Metastore, located on the Cloudera Manager Clusters pane.

3. From the Hive Metastore view, click Actions Download Client Configuration to download the client configuration files as a zip file.
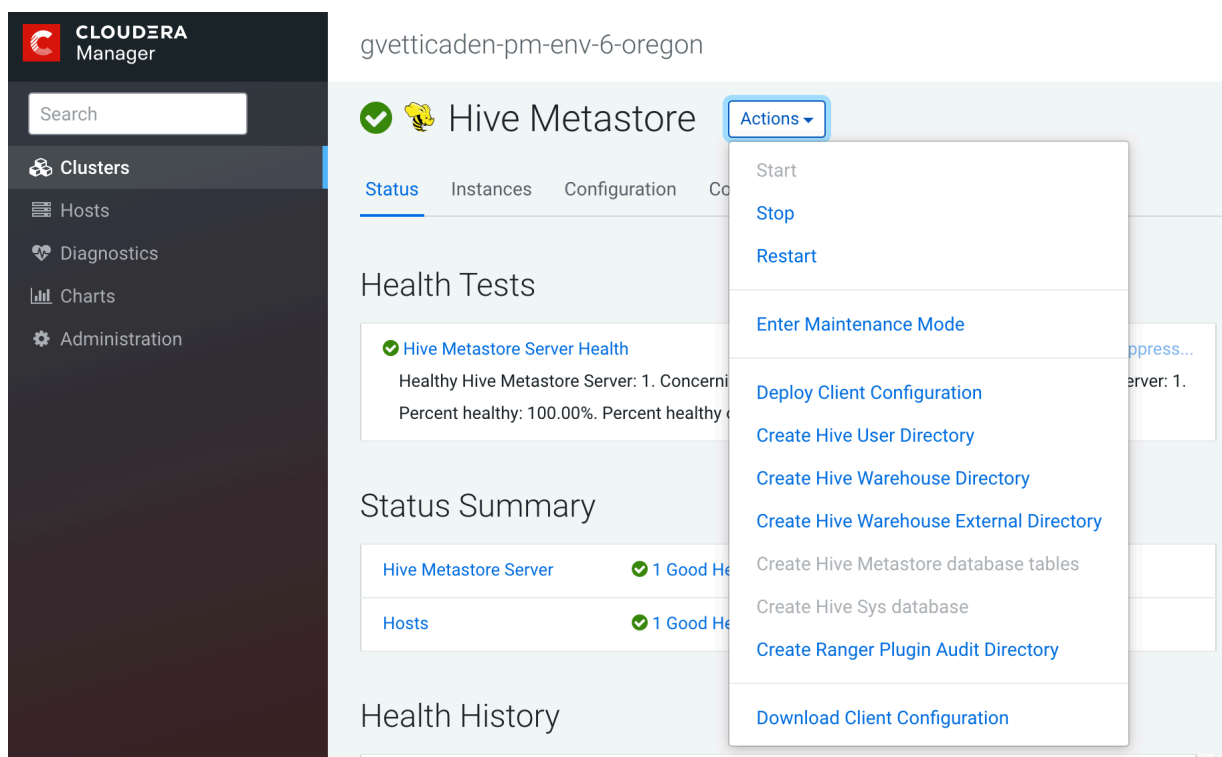
4. Expand the zip file and upload the following files to each of the NiFi nodes in your Flow Management cluster:

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

For example:

```
scp hdfs-site.xml cdp_username@publiclp:/tmp
```

You may also copy these files to a more permanent location later.

5. Adjust the file permissions to ensure that the NiFi processors can read them.

For example:

```
chmod 755 /tmp/*-site.xml
```

Build the data flow

From the Apache NiFi canvas, set up the elements of your data flow. This involves opening NiFi in Cloudera on cloud, adding processors to your NiFi canvas, and connecting the processors.

About this task

When you are building a data flow to ingest data into Apache Hive following are the preferred Hive processors

- `Select_Hive_QL_3` – To get data out of Hive tables.
- `PutHive3Streaming` – To write data to managed and external Hive tables.

Procedure

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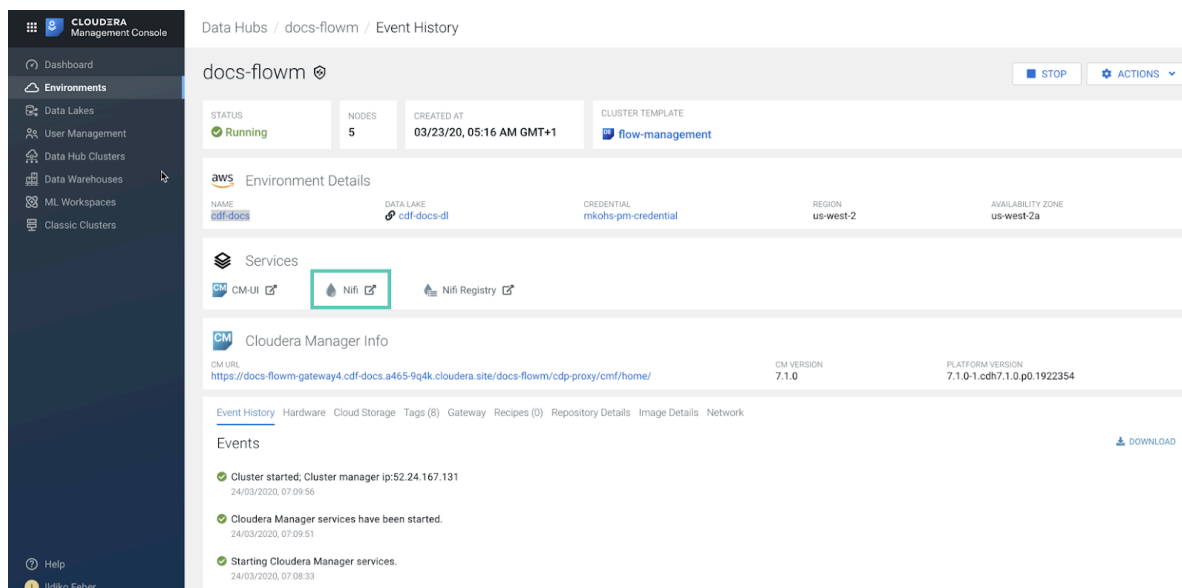   a) To aces the NiFi service in your Flow Management cluster, navigate to the Cloudera Management Console and click Data Hub Clusters.
   b) Click the tile representing the Flow Management cluster with which you want to work.
   c) Click the NiFi icon in the Services section of the Cluster overview page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

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:
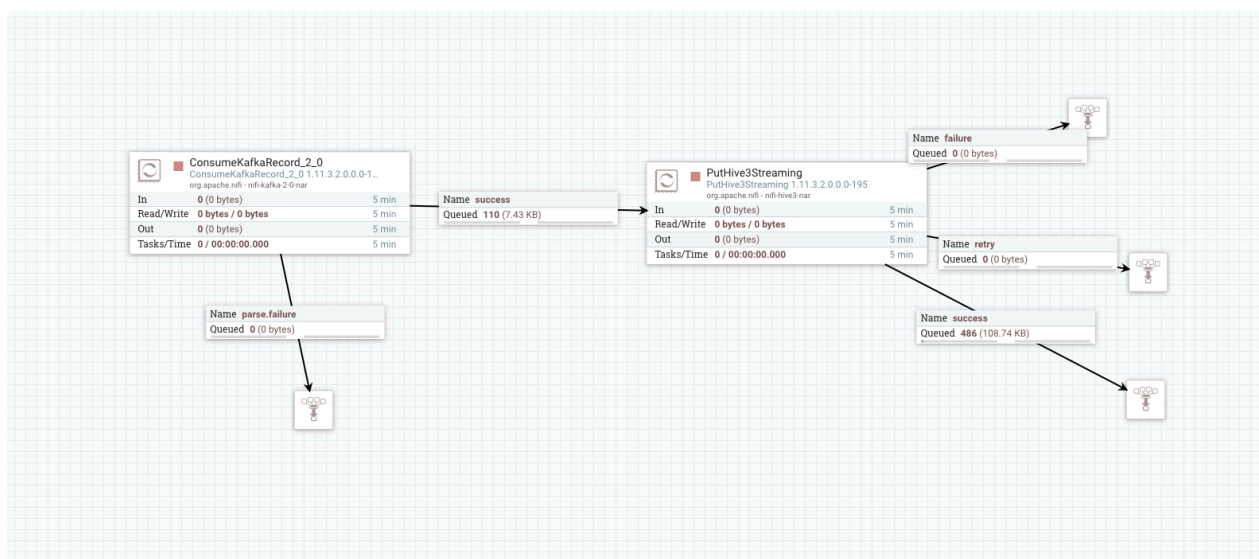
What to do next

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 Cloudera on cloud

Consume KafkaRecord_2_0

Building a data flow

Configure the controller services

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 configuration 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.

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

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.

Example

In this example, configure the following Controller Services:

- AvroReader
- AvroWriter
- HortonworksSchemaRegistry

AvroReader Controller Service

Table 1: AvroReader Controller Service properties

Property	Description	Example value for ingest data flow
Schema Access Strategy	Specify how to obtain the schema to be used for interpreting the data.	HWX Content-Encoded Schema Reference
Schema Registry	Specify the Controller Service to use for the Schema Registry.	CDPSchemaRegistry
Schema Name	Specify the name of the schema to look up in the Schema Registry property.	customer

WriteAvro Controller Servie

Table 2: WriteAvro Controller Service properties

Property	Description	Example value for ingest data flow
Schema Write Strategy	Specifies how the schema for a Record should be added to the data.	HWX Content Encoded Schema Reference
Schema Access Strategy	Specifies how to obtain the schema that is to be used for interpreting the data.	Use Schema Name Property
Schema Registry	Specifies the Controller Service to use for the Schema Registry.	HortonworksSchemaRegistry
Schema name	Specifies the name of the schema to lookup in the Schema Registry property.	*CUSTOMER*

HortonworksSchema Registry

Table 3: HortonworksSchemaRegistry Controller Service properties

Property	Description	Example value for ingest data flow
Schema Registry URL	Provide the URL of the schema registry that this Controller Service should connect to, including version. In the format: https://host :7790/api/v1	https://docs-messaging-master0.cdf-docs.a465 -9q4k.cloudera.site:7790/api/v1
SSL Context Service	Specify the SSL Context Service to use for communicating with Schema Registry. Use the pre-configured SSLContextProvider.	Default NiFi SSL Context Service
Kerberos Principal	Specify the user name that should be used for authenticating with Kerberos. Use your Cloudera workload username to set this Authentication property.	srv_nifi-hive-ingest
Kerberos Password	Provide the password that should be used for authenticating with Kerberos. Use your Cloudera workload password to set this Authentication property.	password

Configure the processor for your data source

You can set up a data flow to move data from many locations into Apache Hive. 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 Hive. 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 Hive data flow

Kafka Brokers	Provide a comma-separated list of known Kafka Brokers in the format <host>:<port>.	`Docs-messaging-broke`
		`r1.cdf-docs.a465-9q4`
		`k.cloudera.site:9093,`
		`docs-messaging-broker2`
		`.cdf-docs.a465-9q4k.`
		`cloudera.site:9093,`
		`docs-messaging-broke`
		`r3.cdf-docs.a465-9q4`
		`k.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.	*CSVRECORDSETWRITER*
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-HIVE-INGEST*

Property	Description	Value for ingest to Hive data flow
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*
Offset Reset	Allows you to manage the condition when there is no initial offset in Kafka or if the current offset does not exist any more on the server (e.g. because that data has been deleted). Corresponds to Kafka's `auto.offset.reset` property.	*LATEST*
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-HIVE-INGEST*

What to do next

Configure the processor for your data target.

Related Information

ConsumeKafka_2_0

Building a data flow

Ingesting data into Apache Kafka in Cloudera on 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 Hive using `PutHive3Streaming`. 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.

Procedure

1. Launch the Configure Processor window, by right-clicking the processor and selecting Configure.
2. Click the Properties tab.
3. Configure `PutHive3Streaming` with the required values.

   The following table includes a description and example values for the properties required to configure an ingest data flow into Hive. For a complete list of `PutHive3Streaming`, see the *processor documentation*.

Property	Description	Value for ingest to Hive data flow
Database name	The name of the database in which to put the data.	retail
Table name	The name of the database table in which to put the data.	customer
Record Reader		ReadAvro
Hive Metastore URI	The URI location for the Hive Metastore. Note that this is not the location of the Hive Server. The default port for the Hive metastore is 9083.	*THIFT://<DATALAKE-MASTER-HOST-NAME:9083*

Property	Description	Value for ingest to Hive data flow
Hive Configuration Resources	A file or comma separated list of files which contains the Hive configuration.	Replace with the file paths for the three files you downloaded in *Obtain the Hive connection details*: • hdfs-site.xml • hive-site.xml • core-site.xml
Kerberos Principal	Specify your Cloudera machine user name.	*SRV_NIFI-HIVE-INGEST*
Kerberos Password	Specify your Cloudera machine user password.	*PASSWORD1!*

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 your data flow

Learn how you can verify the operation of your Hive ingest data flow.

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. Review the Hive ingest data flow on the NiFi canvas and confirm that the NiFi processors are not producing errors.
2. You can look at the processors in the UI, where you can see the amount of data that has gone through them. You can also right click on the processors, or on connections to view status history.
3. From the Data Analytics Studio UI, in the Compose tab, confirm that your Hive table is consuming content:

```
select * from retail.customer
```

Results
You are able to see data flowing into your retail.customer table.

Next steps

Provides information on what to do once you have moved data into Hive in Cloudera on cloud.

You have built a simple data flow for an easy way to move data to Hive. This example data flow enables you to easily design more complex data flows for moving and processing data in Hive. 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 access documentation* for information about working with Hive, Impala, and Hue.

Related Information

NiFi documentation

NiFi Registry documentation

Cloudera Runtime data access documentation