

Kafka to Cloudera Operational Database

Date published: 2021-04-06

Date modified: 2024-06-03



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

ReadyFlow: Kafka to Cloudera Operational Database.....	4
Prerequisites.....	4
List of required configuration parameters for the Kafka to Cloudera Operational Database ReadyFlow.....	9

ReadyFlow: Kafka to Cloudera Operational Database

You can use an Apache NiFi data flow to ingest data into Cloudera Operational Database through Cloudera DataFlow. Learn how to use NiFi to move data from a range of locations to Cloudera Operational Database in Cloudera Public Cloud.

This ReadyFlow consumes JSON, CSV or Avro data from a source Kafka topic, parses the schema by looking up the schema name in the Cloudera Schema Registry and ingests it into an HBase table in Cloudera Operational Database. Failed HBase write operations are retried automatically to handle transient issues. Define a KPI on the `failure_WriteToCOD` connection to monitor failed write operations.

ReadyFlow details	
Source	Kafka topic
Source Format	JSON, CSV, Avro
Destination	Cloudera Operational Database
Destination Format	HBase Table

Today's scalable web applications for use cases like hotel or flight bookings as well as mobile banking applications are relying on an equally scalable database which can serve data at a very low latency. Cloudera Operational Database in Cloudera DataFlow is powered by Apache HBase and provides application developers with everything they need to build scalable applications on top of it.

You can use Apache NiFi data flows into Apache HBase in a Cloudera Operational Database cluster to make sure that the applications you build on top always have access to the latest data.

Prerequisites

Learn how to collect the information you need to deploy the Kafka to Cloudera Operational Database (COD) ReadyFlow, and meet other prerequisites.

Use the following checklist to ensure that you meet all the requirements before you start building your data flow:

For your data ingest source

- You have created a Streams Messaging cluster in Cloudera Public Cloud to host your Schema Registry.
For information on how to create a Streams Messaging cluster, see [Setting up your Streams Messaging Cluster](#).

- You have created at least one Kafka topic.
 1. Navigate to Management Console > Environments and select your environment.
 2. Select your Streams Messaging cluster.
 3. Click on the Streams Messaging Manager icon.
 4. Navigate to the Topics page.
 5. Click Add New and provide the following information:
 - Topic name
 - Number of partitions
 - Level of availability
 - Cleanup policy

**Tip:**

SMM has automatically set Kafka topic configuration parameters. To manually adjust them, click Advanced.

6. Click Save.
- You have created a schema for your data and have uploaded it to the Schema Registry in the Streams Messaging cluster.

For information on how to create a new schema, see [Creating a new schema](#). For example:

```
{
  "type": "record",
  "name": "SensorReading",
  "namespace": "com.cloudera.example",
  "doc": "This is a sample sensor reading",
  "fields": [
    {
      "name": "sensor_id",
      "doc": "Sensor identification number.",
      "type": "int"
    },
    {
      "name": "sensor_ts",
      "doc": "Timestamp of the collected readings.",
      "type": "long"
    },
    {
      "name": "sensor_0",
      "doc": "Reading #0.",
      "type": "int"
    },
    {
      "name": "sensor_1",
      "doc": "Reading #1.",
      "type": "int"
    },
    {
      "name": "sensor_2",
      "doc": "Reading #2.",
      "type": "int"
    },
    {
      "name": "sensor_3",
      "doc": "Reading #3.",
      "type": "int"
    }
  ]
}
```

```
}
```

- You have the Schema Registry Host Name.
 - From the Management Console, go to Data Hub Clusters and select the Streams Messaging cluster you are using.
 - Navigate to the **Hardware** tab to locate the Master Node FQDN. Schema Registry is always running on the Master node, so copy the Master node FQDN.
- You have the Kafka broker end points.
 - From the Management Console, click Data Hub Clusters.
 - Select the Streams Messaging cluster from which you want to ingest data.
 - Click the Hardware tab.
 - Note the Kafka Broker FQDNs for each node in your cluster.
 - Construct your Kafka Broker Endpoints by using the FQDN and Port number 9093 separated by a colon. Separate endpoints by a comma. For example:

```
broker1.fqdn:9093,broker2.fqdn:9093,broker3.fqdn:9093
```

Kafka broker FQDNs are listed under the **Core_broker** section.

- You have the Kafka Consumer Group ID.
This ID is defined by the user. Pick an ID and then create a Ranger policy for it. Use the ID when deploying the flow in Cloudera DataFlow.
- You have assigned the Cloudera Workload User policies to access the consumer group ID and topic.
 - Navigate to Management Console > Environments, and select the environment where you have created your cluster.
 - Select Ranger. You are redirected to the Ranger **Service Manager** page.
 - Select your Streams Messaging cluster under the **Kafka** folder.
 - Create a policy to enable your Workload User to access the Kafka source topic.
 - On the **Create Policy** page, give the policy a name, select topic from the drop-down list, add the user, and assign the Consume permission.
 - Create another policy to give your Workload User access to the consumer group ID.
 - On the **Create Policy** page, give the policy a name, select consumergroup from the drop-down list, add the user, and assign the Consume permission.
- You have assigned the Cloudera Workload User read-access to the schema.
 - Navigate to Management Console > Environments, and select the environment where you have created your cluster.
 - Select Ranger. You are redirected to the Ranger **Service Manager** page.
 - Select your Streams Messaging cluster under the **Schema Registry** folder.
 - Click Add New Policy.
 - On the **Create Policy** page, give the policy a name, specify the schema details, add the user, and assign the Read permission.

For Cloudera DataFlow

- You have enabled Cloudera DataFlow for an environment.
For information on how to enable Cloudera DataFlow for an environment, see [Enabling Cloudera DataFlow for an Environment](#).
- You have created a Machine User to use as the Cloudera Workload User.

- You have given the Cloudera Workload User the EnvironmentUser role.
 - From the Management Console, go to the environment for which Cloudera DataFlow is enabled.
 - From the Actions drop down, click Manage Access.
 - Identify the user you want to use as a Workload User.

**Note:**

The Cloudera Workload User can be a machine user or your own user name. It is best practice to create a dedicated Machine user for this.


- Give that user EnvironmentUser role.
- You have synchronized your user to the Cloudera Public Cloud environment that you enabled for Cloudera DataFlow.

For information on how to synchronize your user to FreeIPA, see [Performing User Sync](#).

- You have granted your Cloudera user the DFCatalogAdmin and DFFlowAdmin roles to enable your user to add the ReadyFlow to the Catalog and deploy the flow definition.
 - Give a user permission to add the ReadyFlow to the Catalog.
 - From the Management Console, click User Management.
 - Enter the name of the user or group you wish to authorize in the Search field.
 - Select the user or group from the list that displays.
 - Click Roles Update Roles .
 - From Update Roles, select DFCatalogAdmin and click Update.



Note: If the ReadyFlow is already in the Catalog, then you can give your user just the DFCatalogViewer role.

- Give your user or group permission to deploy flow definitions.
 - From the Management Console, click Environments to display the Environment List page.
 - Select the environment to which you want your user or group to deploy flow definitions.
 - Click Actions Manage Access to display the Environment Access page.
 - Enter the name of your user or group you wish to authorize in the Search field.
 - Select your user or group and click Update Roles.
 - Select DFFlowAdmin from the list of roles.
 - Click Update Roles.
- Give your user or group access to the Project where the ReadyFlow will be deployed.
 - Go to DataFlow Projects .
 - Select the project where you want to manage access rights and click  More Manage Access .
- Start typing the name of the user or group you want to add and select them from the list.
- Select the Resource Roles you want to grant.
- Click Update Roles.
- Click Synchronize Users.

For your data ingest target

- Ensure that the HBase table you are ingesting data to exists. If not, create one.
 1. From Cloudera Shared Data Experience UI, click Operational Database from the left navigation pane.
 2. Click Create Database.
 3. Select the environment for which Cloudera DataFlow is enabled.
 4. Enter a name for your database, and click Create Database.
 5. Go to the newly created database from the Databases page.
 6. Go to Hue UI by clicking Hue SQL Editor.
 7. Click the HBase icon to go to HBase home.
 8. Click New Table.

The Create New Table dialog appears.

9. Enter table name and column family name, and click Submit.

A blank table is created.

10. Go to the newly created table and click New Row.

The Insert New Row dialog appears.

11. Click Add Field, and then specify row key, column name, and cell value.



Note: The column name should follow the format: family: column_name, where, family is the column family name.

12. Click Submit.

- Obtain the table name, column family name, and row identifier of the HBase table in Cloudera Operational Database.

1. From Cloudera Shared Data Experience UI, click Operational Database from the left navigation pane.
2. Select the database where your HBase table exists.
3. Go to Hue UI by clicking Hue SQL Editor.
4. Click the HBase icon to go to HBase home.
5. Click the HBase table in Cloudera Operational Database.
6. After the table appears, obtain the table name, column family name, and row identifier.

- You have set Ranger policies for HBase table.

1. From the Cloudera Management Console, click Environments.
2. Use the search field to find and select the Cloudera environment for which Cloudera DataFlow is enabled.
3. Go to the Ranger UI by clicking Ranger.
4. Select your database from the HBase folder.
5. Click Add New Policy.
6. Enter policy name, HBase table name, HBase column-family name, and HBase column value.
7. In Allow Conditions section, enter the name of the Machine User, you created in Cloudera, prefixed with srv_.
8. Click Add Permissions, and assign Read and Write permissions to the user.
9. Click Add.

- Obtain the hbase-site.xml file.

To get the hbase-site.xml file from Cloudera Data Hub:

1. From the Cloudera Management Console, click Environments.
2. Use the search field to find and select the Cloudera Public Cloud environment for which Cloudera DataFlow is enabled.
3. Go to Data Hubs.
4. Select the Cloudera Operational Database cluster.
5. Go to Cloudera Manager by clicking CM-UI.
6. Click Clusters from the left-navigation pane, and click hbase.
7. Click Actions Download Client Configuration to download the client configuration zip file.
8. Unzip the zip file to obtain the hbase-site.xml file.

To get the hbase-site.xml file from Cloudera Shared Data Experience:

1. From Cloudera Shared Data Experience UI, click Operational Database from the left navigation pane.
2. Select the database where your HBase table exists.
3. Go to HBase Client Tarball tab.
4. Copy the HBase Client Configuration URL.
5. Use the URL as a command to download the client configuration zip file.
6. Unzip the zip file to obtain the hbase-site.xml file.

Related Concepts

[List of required configuration parameters for the Kafka to Cloudera Operational Database ReadyFlow](#)

List of required configuration parameters for the Kafka to Cloudera Operational Database ReadyFlow

When deploying the Kafka to Cloudera Operational Database ReadyFlow, you have to provide the following parameters. Use the information you collected in *Prerequisites*.

Table 1: Kafka to Cloudera Operational Database ReadyFlow configuration parameters

Parameter Name	Description
CDP Workload User	Specify the Cloudera machine user or workload user name that you want to use to authenticate to Kafka. Ensure this user has the appropriate access rights in Ranger for the source and target Kafka topics.
CDP Workload User Password	Specify the Cloudera machine user or workload user name that you want to use to authenticate to Kafka.
CDPEnvironment	Use this parameter to upload the hbase-site.xml file of your target Hbase cluster. Cloudera DataFlow will also use this parameter to auto-populate the Flow Deployment with additional Hadoop configuration files required to interact with HBase.
COD Column Family Name	Specify the column family to use when inserting data into Cloudera Operational Database.
COD Row Identifier Field Name	Specify the name of a record field whose name should be used as the row ID for the given record.
COD Table Name	Specify the target table name in Cloudera Operational Database.
CSV Delimiter	If your source data is CSV, specify the delimiter here,
Data Input Format	Specify the desired format for your output data. You can use "CSV", "JSON" or "AVRO" with this ReadyFlow.

Parameter Name	Description
Kafka Broker Endpoint	Specify the Kafka bootstrap servers string as a comma separated list.
Kafka Consumer Group ID	Specify the id for the consumer group used for the source topic you are consuming from.
Kafka Source Topic	Specify a topic name that you want to read from.
Schema Name	Specify the schema name to be looked up in the Schema Registry.
Schema Registry Hostname	Specify the host name of the Schema Registry you want to connect to. This must be the direct hostname of the Schema Registry itself, not the Knox endpoint.

Related Concepts[Prerequisites](#)**Related Information**[Deploying a ReadyFlow](#)