

## Kafka to Kudu

Date published: 2021-04-06

Date modified: 2025-09-30



# Legal Notice

© Cloudera Inc. 2025. 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

<b>ReadyFlow: Kafka to Kudu.....</b>	<b>4</b>
<b>Prerequisites.....</b>	<b>4</b>
<b>Required parameters.....</b>	<b>8</b>

## ReadyFlow: Kafka to Kudu

This use case shows you how you can move your data from a Kafka topic into Apache Kudu in your Cloudera on cloud Real-time Data Mart cluster. You can learn how to create such a data flow easily using the Kafka to Kudu ReadyFlow.

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 a Kudu table. You can pick the Kudu operation (INSERT, INSERT\_IGNORE, UPSERT, UPDATE, DELETE, UPDATE\_IGNORE, DELETE\_IGNORE) that fits best for your use case. Failed Kudu write operations are retried automatically to handle transient issues. Define a KPI on the `failure_WriteToKudu` connection to monitor failed write operations.

Kafka to Kudu ReadyFlow details	
Source	Kafka topic
Source Format	JSON, CSV, Avro
Destination	Kudu
Destination Format	Kudu Table

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.

## Prerequisites

Learn how to collect the information you need to deploy the Kafka to Kudu ReadyFlow, and meet other prerequisites.

### For your data ingest source

- You have created a Streams Messaging cluster in Cloudera on 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.
  1. From the Management Console, go to Data Hub Clusters and select the Streams Messaging cluster you are using.
  2. 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 Data Flow.
- 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 Data Flow

- You have enabled Cloudera Data Flow for an environment.  
For information on how to enable Cloudera Data Flow for an environment, see [Enabling Cloudera Data Flow 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 Data Flow 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 on cloud environment that you enabled for Cloudera Data Flow.

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.

1. Give a user permission to add the ReadyFlow to the Catalog.

- a. From the Management Console, click User Management.
- b. Enter the name of the user or group you wish to authorize in the Search field.
- c. Select the user or group from the list that displays.
- d. Click Roles Update Roles .
- e. 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.

2. Give your user or group permission to deploy flow definitions.

- a. From the Management Console, click Environments to display the Environment List page.
- b. Select the environment to which you want your user or group to deploy flow definitions.
- c. Click Actions Manage Access to display the Environment Access page.
- d. Enter the name of your user or group you wish to authorize in the Search field.
- e. Select your user or group and click Update Roles.
- f. Select DFFlowAdmin from the list of roles.
- g. Click Update Roles.

3. Give your user or group access to the Project where the ReadyFlow will be deployed.

- a. Go to Data Flow Projects .
- b. Select the project where you want to manage access rights and click  More Manage Access .

4. Start typing the name of the user or group you want to add and select them from the list.

5. Select the Resource Roles you want to grant.

6. Click Update Roles.

7. Click Synchronize Users.

### For your data ingest target

- You have a Real-Time Data Mart cluster running Kudu, Impala, and Hue in the same environment for which Cloudera Data Flow has been enabled.
- You have the Kudu Master hostnames.

1. From Management Console, click Data Hub Clusters.

2. Select the Real-Time Data Mart cluster to which you want to ingest data into.

3. Click the Hardware tab.

4. Copy the FQDN for each Kudu Master.

- You have created the Kudu table that you want to ingest data into.

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

7. Check the Kudu UI **Tables** tab for the name of the table you created. You will need this table name when you use the Cloudera Data Flow Deployment wizard to deploy the ReadyFlow.

- You have assigned permissions via IDBroker or in Ranger to enable the Cloudera Workload User to access the Kudu table that you want to ingest data into.
  - From the base cluster on Cloudera on cloud, select Ranger.
  - Select your Real Time Data Mart cluster from the **Kudu** folder.
  - Click Add New Policy policy.
  - On the **Create Policy** page, enter the Kudu table name in the topic field.
  - Add the Cloudera Workload User in the Select User field.
  - Add the Insert and Select permissions in the Permissions field.
  - Click Save.

### Related Concepts

[Required parameters](#)

## Required parameters

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

**Table 1: Kafka to Kudu 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 and Kudu. Ensure this user has the appropriate access rights to the Kafka topics and Kudu table.
CDP Workload User Password	Specify the password of the Cloudera machine user or workload user you are using to authenticate against Kafka and Kudu.
CSV Delimiter	If your source data is CSV, specify the delimiter here.
Data Input Format	Specify the format of your input data. You can use "CSV", "JSON" or "AVRO" with this ReadyFlow.
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.
Kudu Master Hosts	Specify the Kudu Master hostnames in a comma separated list.
Kudu Operation Type	Specify the operation that you want to use when writing data to Kudu. Valid values are: <ul style="list-style-type: none"> <li>INSERT</li> <li>INSERT_IGNORE</li> <li>UPSERT</li> <li>UPDATE</li> <li>DELETE</li> </ul>
Kudu Table Name	Specify the Kudu table name you want to write to.
Schema Name	Specify the schema name to be looked up in the Schema Registry.
Schema Registry Hostname	Specify the hostname 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](#)