

Confluent Cloud to S3/ADLS

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: Confluent Cloud to S3/ADLS..... | 4 |
| Prerequisites..... | 4 |
| List of required configuration parameters for the Confluent Cloud to S3/ ADLS ReadyFlow..... | 7 |

ReadyFlow: Confluent Cloud to S3/ADLS

You can use the Confluent Cloud to S3/ADLS ReadyFlow to ingest JSON, CSV or Avro data from a source Kafka topic in Confluent Cloud to a destination S3 or ADLS location, while filtering events using a SQL query.

This ReadyFlow consumes JSON, CSV or Avro data from a source Kafka topic in Confluent Cloud and parses the schema by looking up the schema name in the Confluent Schema Registry. You can filter events by specifying a SQL query in the 'Filter Rule' parameter. The filtered records are then converted to the specified output data format. The flow then writes out a file every time its size has either reached 100MB or five minutes have passed. Files can reach a maximum size of 1GB. The filtered events are then written to the destination Amazon S3 or Azure Data Lake Service (ADLS) location. Failed S3 or ADLS write operations are retried automatically to handle transient issues. Define a KPI on the failure_WriteToS3/ADLS connection to monitor failed write operations.



Note: This ReadyFlow leverages Cloudera Public Cloud's centralized access control for cloud storage access. Make sure to either set up Ranger policies or an IDBroker mapping allowing your workload user access to the target S3 or ADLS location.

| ReadyFlow details | |
|--------------------|-----------------------------|
| Source | Confluent Cloud Kafka Topic |
| Source Format | JSON, CSV, Avro |
| Destination | Amazon S3 or ADLS |
| Destination Format | JSON, CSV, Avro |

Prerequisites

Learn how to collect the information you need to deploy the Confluent Cloud to S3/ADLS ReadyFlow, and meet other prerequisites.

For your data ingest source

- You have created a Confluent Cloud Kafka cluster.
- You have created at least one Kafka topic.
- You have created a schema for your data in the Confluent Cloud Schema Registry.
- You have the Confluent Cloud Kafka broker endpoint.
- You have the Confluent Cloud Kafka client API Key and Secret.
- You have the Kafka Consumer Group ID.
- You have the Confluent Cloud Schema Registry endpoint.
- You have the Confluent Cloud Schema Registry API Key and Secret.

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 ADLS data ingest target

- You have your ADLS container and path into which you want to ingest data.

- You have performed one of the following to configure access to your ADLS folder:
 - You have configured access to the ADLS folders with a RAZ enabled environment.

It is a best practice to enable RAZ to control access to your object store folders. This allows you to use your Cloudera Public Cloud credentials to access ADLS folders, increases auditability, and makes object store data ingest workflows portable across cloud providers.

1. Ensure that Fine-grained access control is enabled for your Cloudera DataFlow environment.
2. From the Ranger UI, navigate to the ADLS repository.
3. Create a policy to govern access to the ADLS container and path used in your ingest workflow. For example: adls-to-adls-avro-ingest



Tip: The Path field must begin with a forward slash (/).

4. Add the machine user that you have created for your ingest workflow to ingest the policy you just created.

For more information, see *Ranger policies for RAZ-enabled Azure environment*.

- You have configured access to ADLS folders using ID Broker mapping.

If your environment is not RAZ-enabled, you can configure access to ADLS folders using ID Broker mapping.

1. Access IDBroker mappings.
 - a. To access IDBroker mappings in your environment, click **Actions Manage Access**.
 - b. Choose the IDBroker Mappings tab where you can provide mappings for users or groups and click **Edit**.
2. Add your Cloudera Workload User and the corresponding Azure role that provides write access to your folder in ADLS to the Current Mappings section by clicking the blue + sign.



Note: You can get the Azure Managed Identity Resource ID from the Azure Portal by navigating to **Managed Identities Your Managed Identity Properties Resource ID**. The selected Azure MSI role must have a trust policy allowing IDBroker to assume this role.

3. Click **Save and Sync**.

For your S3 data ingest target

- You have your source S3 path and bucket.

- Perform one of the following to configure access to S3 buckets:

- You have configured access to S3 buckets with a RAZ enabled environment.

It is a best practice to enable RAZ to control access to your object store buckets. This allows you to use your Cloudera credentials to access S3 buckets, increases auditability, and makes object store data ingest workflows portable across cloud providers.

1. Ensure that Fine-grained access control is enabled for your Cloudera DataFlow environment.
2. From the Ranger UI, navigate to the S3 repository.
3. Create a policy to govern access to the S3 bucket and path used in your ingest workflow.



Tip:

The Path field must begin with a forward slash (/).

4. Add the machine user that you have created for your ingest workflow to the policy you just created.

For more information, see *Creating Ranger policy to use in RAZ-enabled AWS environment*.

- You have configured access to S3 buckets using ID Broker mapping.

If your environment is not RAZ-enabled, you can configure access to S3 buckets using ID Broker mapping.

1. Access IDBroker mappings.
 - a. To access IDBroker mappings in your environment, click **Actions Manage Access**.
 - b. Choose the IDBroker Mappings tab where you can provide mappings for users or groups and click **Edit**.
2. Add your Cloudera Workload User and the corresponding AWS role that provides write access to your folder in your S3 bucket to the **Current Mappings** section by clicking the blue + sign.



Note: You can get the AWS IAM role ARN from the Roles Summary page in AWS and can copy it into the IDBroker role field. The selected AWS IAM role must have a trust policy allowing IDBroker to assume this role.

3. Click **Save and Sync**.

Related Concepts

[List of required configuration parameters for the Confluent Cloud to S3/ADLS ReadyFlow](#)

List of required configuration parameters for the Confluent Cloud to S3/ADLS ReadyFlow

When deploying the Confluent Cloud to S3/ADLS ReadyFlow, you have to provide the following parameters. Use the information you collected in *Prerequisites*.

Table 1: Confluent Cloud to S3/ADLS ReadyFlow configuration parameters

| Parameter Name | Description |
|----------------------------|---|
| CDP Workload User | Specify the Cloudera machine user or workload username that you want to use to authenticate to the object stores. Ensure this user has the appropriate access rights to the object store locations in Ranger or IDBroker. |
| CDP Workload User Password | Specify the password of the Cloudera machine user or workload user you are using to authenticate against the object stores (via IDBroker). |
| CSV Delimiter | If your source data is CSV, specify the delimiter here. |

| Parameter Name | Description |
|---|---|
| Data Input Format | Specify the format of your input data. You can use "CSV", "JSON" or "AVRO" with this ReadyFlow. <ul style="list-style-type: none"> • CSV • JSON • AVRO |
| Data Output Format | Specify the desired format for your output data. You can use "CSV", "JSON" or "AVRO" with this ReadyFlow. |
| Destination S3 or ADLS Path | Specify the name of the destination S3 or ADLS path you want to write to. Make sure that the path starts with "/". |
| Destination S3 or ADLS Storage Location | Specify the name of the destination S3 bucket or ADLS Container you want to write to. For S3, enter a value in the form: s3a://[Destination S3 Bucket] For ADLS, enter a value in the form: abfs://[Destination ADLS File System]@[Destination ADLS Storage Account].dfs.core.windows.net |
| Filter Rule | Specify the filter rule expressed in SQL to filter streaming events for the destination database. Records matching the filter will be written to the destination database. The default value forwards all records. |
| Kafka Broker Endpoint | Specify the Kafka bootstrap server. |
| Kafka Client API Key | Specify the API Key to connect to the Kafka cluster. |
| Kafka Client API Secret | Specify the API Secret to connect to the Kafka cluster. |
| Kafka Consumer Group Id | The name of 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. |
| Kafka Schema Name | Specify the schema name to be looked up in the Confluent Schema Registry for the Source Kafka Topic. |
| Kafka Source Topic | Specify the topic name that you want to read from. |
| Schema Registry Client Key | Specify the API Key to connect to the Confluent Schema Registry. |
| Schema Registry Client Secret | Specify the API Secret to connect to the Confluent Schema Registry. |
| Schema Registry Endpoint | Specify the Schema Registry API endpoint. |

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