Cloudera Runtime 7.2.15

Using Schema Registry

Date published: 2019-11-08
Date modified: 2022-05-12

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

Adding a new schema

To add a new schema to Schema Registry, provide information about the schema, select a compatibility policy and upload the schema text from a file. A schema is a collection of information that describes your data. The schema's metadata includes group, version and description.

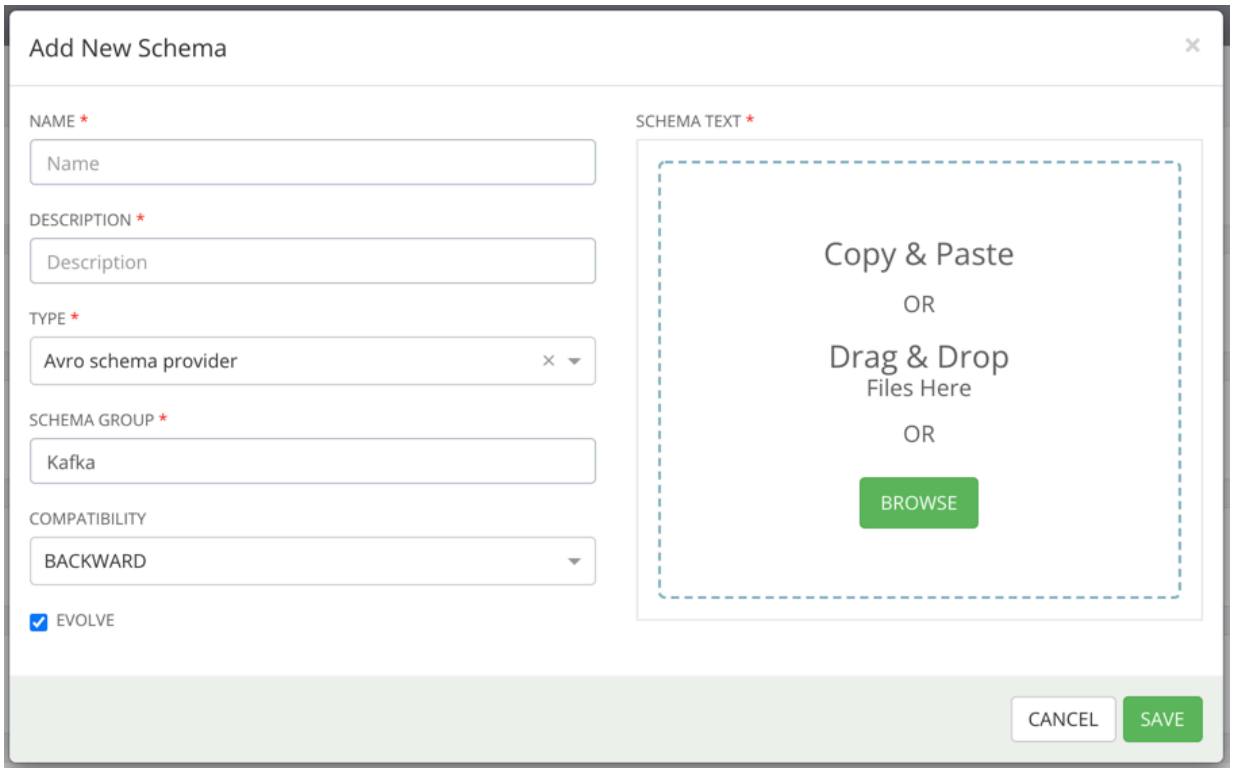
Before you begin

Ensure that you understand compatibility policies. Once selected, you cannot change the compatibility policy for a schema.

Procedure

1. From the Schema Registry UI, click the + icon.

   The Add New Schema dialog appears.



2. Add the schema metadata as follows:
   - Name

     A unique name for each schema. Used as a key to look up schemas.
   - Description

     A short description of the schema.
   - Schema Type

     The schema format. Select one of the following formats:
     - Avro schema provider
     - Json schema provider
   - Schema Group

     Allows you to group schemas in any logical order.
   - Compatibility

     This option appears only when the Avro schema provider option is selected. Sets the compatibility policy for the schema. Once set, this cannot be changed. Select one of the following options:
     - Backward
     - Forward
     - Both
     - None

     For more information about compatibility, see *Compatibility Policies*.
   - Schema Text
3. To allow schema to evolve over time by creating multiple versions, select the Evolve checkbox.

   Note:

   Unselecting Evolve means that you can only have one version of a schema.
4. Click Choose File to upload a new schema.

Related Information

Compatibility Policies

Querying a schema

Learn how to search for a particular schema in the Schema Registry UI.

You can use the Search box at the top of the Schema Registry UI to search for schemas by name, description, or schema text. To search, you can enter a schema name, key words, or any text string to return schemas with that value.

To return to your full list of schemas, clear the search bar and press Enter on your keyboard.

Evolving a schema

You evolve a schema when you create a new version. Schema Registry tracks the changes made to your schema, and stores each set of changes in a separate version of the schema. When multiple versions exist, you can select which version you want to use. Ensure that you understand compatibility policies, as they determine how you can evolve your schemas.

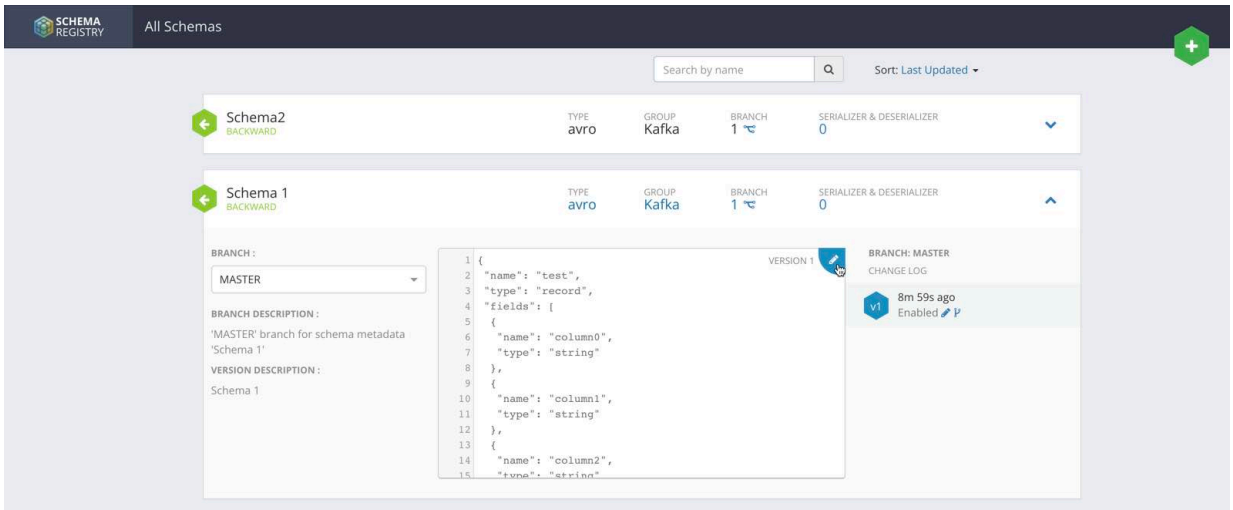
Before you begin

- You have selected the Evolve checkbox when initially adding the schema.

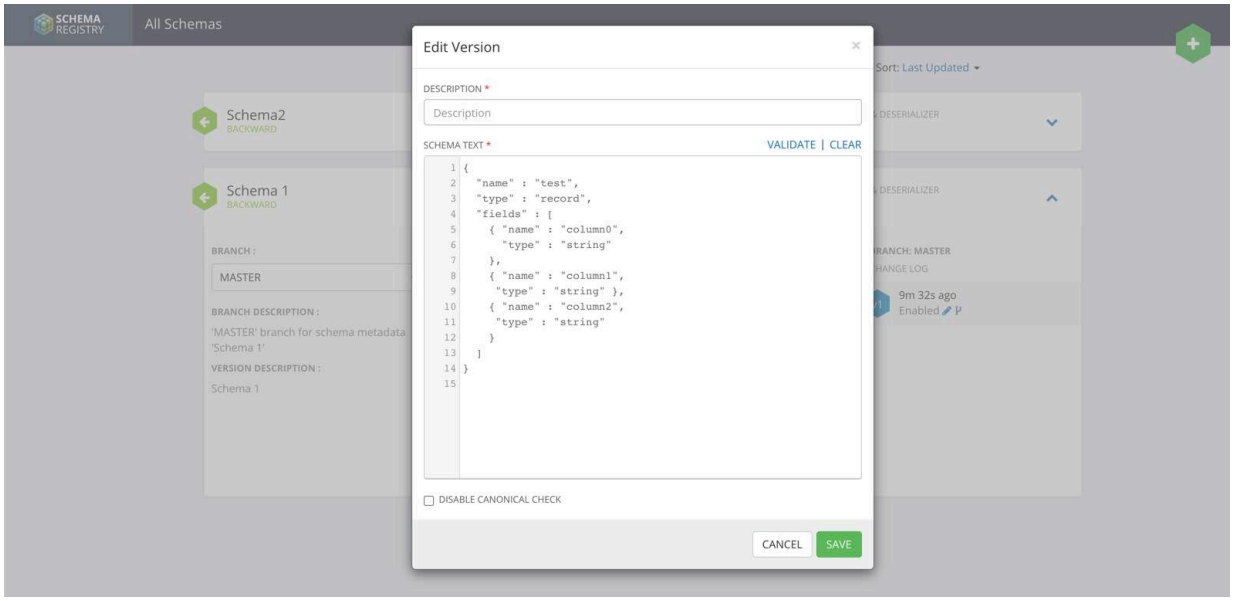
- You have saved the schema you want to evolve in a file.

Procedure

1. From the Schema Registry UI, identify the schema that you want to version.
2. Click the pencil icon to open the Edit Version dialog.



The Edit Version dialog appears.



3. Add a description of what has changed in this new version of the schema.

   You can view the description in the Schema Registry UI to easily understand what has changed in each version of the schema. Cloudera recommends that you add as much detail as you can.
4. Optional. Select the Disable Canonical Check checkbox if the schema version should be added despite being canonically similar to an existing schema version.

   The canonical check checkbox only applies within versions of schemas and not a new schema.
5. Click Choose File to upload the schema you want to evolve.
6. Click OK.

Deleting a schema

You can use the Schema Registry REST API to delete schemas.

About this task

You can access the Schema Registry API Swagger documentation directly from the UI. To do this, append your URL with: /swagger/

For example: http://localhost:9090/swagger/

Procedure

1. Find the DELETE /api/v1/schemaregistry/schemas/{schema_name} API under Schema.
2. Replace {schema_name} with the schema you wish to delete.
3. Invoke the API.

Importing Confluent Schema Registry schemas into Cloudera Schema Registry

If you are migrating away from Confluent's Schema Registry, you can use the schema-import tool to import your schemas to the Cloudera Schema Registry service.

About this task

Confluent Schema Registry stores all schemas in a Kafka topic usually named _schemas. To import these schemas you must copy the content to a text file and upload to Cloudera's Schema Registry service.

Before you begin

You must have the export-import access policy with the Create permission assigned to your user account.

For more information on Schema Registry access policies, see *Schema Registry Authorization through Ranger Access Policies*

Procedure

1. Copy the content of the Kafka topic (usually named _schemas), to a text file by using the following command:

```
kafka-console-consumer --from-beginning --topic _schemas --formatter kaf
ka.tools.DefaultMessageFormatter --property print.key=true --property pr
int.value=true --bootstrap-server `hostname`:9092
```

   Here is an example of a text file with the content of the Kafka topic:

```
{"keytype":"SCHEMA","subject":"Kafka-key","version":1,"magic":1}
{"subject":"Kafka-key","version":1,"id":1,"schema":"\"string\"","deleted
":false}
{"keytype":"SCHEMA","subject":"Car","version":1,"magic":1}      {"subje
ct":"Car","version":1,"id":2,"schema":"{\"type\":\"record\",\"name\":\"C
ar\",\"namespace\":\"com.cloudera\",\"fields\":[{\"name\":\"model\",\"type
\":\"string\"},{\"name\":\"color\",\"type\":\"string\",\"default\":\"blue
\"},{\"name\":\"price\",\"type\":\"string\",\"default\":\"0\"},{\"name\":
\"year\",\"type\":[\"null\",\"string\"],\"default\":null}]}","deleted":f
alse}
```

```
{"keytype":"SCHEMA","subject":"Car","version":2,"magic":1}      {"subject"
:"Car","version":2,"id":3,"schema":"{\"type\":\"record\",\"name\":\"Car\",
\"namespace\":\"com.cloudera\",\"fields\":[{\"name\":\"model\",\"type\":\"
string\"},{\"name\":\"color\",\"type\":\"string\",\"default\":\"blue\"},
{\"name\":\"price\",\"type\":\"string\",\"default\":\"0\"}]}","deleted":
false}
```

2. Upload the text file to Cloudera's Schema Registry service by using the following endpoint: /api/v1/schemaregistry/import

   Use the following command:

```
curl -u : --negotiate --form file='<filename>' http://`hostname`:7788/api/
v1/schemaregistry/import?format=1&failOnError=true
```

   Where,

   - format: Can be 0 or 1. However, for schema content from Confluent, the file must be 1.
   - failOnError=true: This setting is useful if the process fails to import all the schemas when one or more schemas already exist in the target Schema Registry.
   - file='<filename>': Enter the name of the file.

   Here is an example of the command:

```
curl -u : --negotiate --form file='@schemadump.json' http://`hostname`:7
788/api/v1/schemaregistry/import?format=1&failOnError=true
```

   Here is an example of the response:

```
{"successCount":45,"failedCount":0,"failedIds":[]}
```

Related Information

Schema Registry Authorization through Ranger Access Policies

Importing and exporting schemas

Schemas stored in Schema Registry can be exported to a JSON file. The exported JSON file can then be imported into another Schema Registry database. Learn how to export and import schemas between different Schema Regitry instances allowing services to exchange data without the challenge of managing and sharing schemas between them.

During an import, SchemaMetadata, SchemaBranch, and SchemaVersion objects are put into the database. These objects retain their ID as well as a number of other properties that are available in the JSON file used for import. This way, serializing and deserializing protocols can continue to function without any change and Schema Registry clients can seamlessly switch between different Schema Registry instances. Both import and export operations are done using the Schema Registry API.

Exporting schemas

Schemas registered in Schema Registry can be exported into a JSON file using the Schema Registry API. The file that you export can be used to import the exported schemas into a different Schema Registry.

About this task

You have two options. You can export schemas using the command line with curl or the Schema Registry Swagger UI. The following steps walk you through how you can export schemas by using the Schema Registry Swagger UI.

Before you begin

- Ensure that the cluster, its hosts, and all its services are healthy.
- Ensure that Schema Registry is commissioned and running.
- Ensure that you have access to all credentials that are required to access and use Schema Registry.
- Ensure that you are logged in as a user with access to the CDP Environment containing the cluster with Schema Registry.

Procedure

1. In Management Console, go to Data Hub clusters.
2. Find and select the Data Hub cluster you want to export schemas from.
3. Under Services click Schema Registry.

   The Schema Registry web UI opens in a new tab.
4. In the tab that has the Schema Registry web UI open, replace /ui/# at the end of the URL with /swagger.
5. Select GET /api/v1/schemaregistry/schemas/aggregated.
6. Click Try it out.
7. Click Execute.
8. Click Download found at the bottom right of the Response body pane to download exported schemas.

Results

All schemas stored in the Schema Registry server are exported to a JSON file.

Importing schemas

Schemas exported from one Schema Registry instance can be imported into a different Schema Registry instance using the Schema Registry API.

About this task

You have two options. You can import schemas using the command line with curl or the Schema Registry Swagger UI. The following steps walk you through how you import schemas by using the Schema Registry Swagger UI.

Important: Schema import is only recommended to an empty Schema Registry server or a Schema Registry server that has previously imported schemas. Importing schemas to a Schema Registry that contains manually added schemas is only possible if the Schema Registry instance has correctly configured ID ranges. Otherwise, importing schemas might result in overlapping ID ranges and a failed import. For more information on how to set up ID ranges, see *ID ranges in Schema Registry*.

On successful import, a number of SchemaMetadata, SchemaBranch, and SchemaVersionInfo objects are added to the Schema Registry database.

Objects added to the database retain the following property values from the import file:

- Schema Metadata objects retain the ID, type, schemaGroup, name, description, compatibility, validationLevel, and evolve properties.
- SchemaBranch objects retain the ID, name, schemaMetadataName, and description properties.
- SchemaVersionInfo objects retain the ID, schemaMetadataId, name, description, version, and schemaText properties.

A successful import also results in a JSON file that contains the following:

```
{
  "successCount": 0,
  "failedCount": 0,
  "failedIds": [
```

```
    0
  ]
}
```

Where,

- successCount is the number of successfully imported SchemaVersionInfo objects.
- failedCount is the number of SchemaVersionInfo objects that failed to import.
- failedIds contains the IDs of failed SchemaVersionInfo objects.

Schemas (metadata/branch/version) that are already present in the database are skipped during an import. Skipped schemas are not counted in successCount or failedCount.

Before you begin

- Ensure that the cluster, its hosts, and all its services are healthy.
- Ensure that Schema Registry is commissioned and running.
- Ensure that you have access to all credentials that are required to access and use Schema Registry.
- Ensure that you have access to a JSON file that contains the exported schemas that you want to import. For more information on how to export schemas, see *Exporting schemas*.
- Ensure that you are logged in as a user with access to the CDP Environment containing the cluster with Schema Registry.

Procedure

1. In Management Console, go to Data Hub clusters.
2. Find and select the Data Hub cluster you want to import schemas to.
3. Under Services click Schema Registry.

   The Schema Registry web UI opens in a new tab.
4. In the tab that has the Schema Registry web UI open, replace /ui/# at the end of the URL with /swagger.
5. Click 3. Export/Import.
6. Click POST /api/v1/schemaregistry/import.
7. Click Try it out.
8. Set format to 0.
9. Set failOnError to true.
10. Click Choose file and select a JSON file containing the schema data that you want to import.
11. Click Execute.

Results

Schemas are successfully imported into the target Schema Registry instance.

ID ranges in Schema Registry

In case your deployment has many Schema Registry instances in different clusters, you might want to ensure that each Schema Registry operates in a preset ID range so that there are no overlapping IDs. To achieve this, you need to set an ID range.

When an ID range is set, all Schema Registry entities (schema metadata, branches, versions, and so on) are created with an ID that falls within the configured range. Configuring a range can simplify schema export and import between Schema Registry instances where conflicting IDs can cause the import process to fail.

Consequently, having non-overlapping ID ranges allocated to each Schema Registry makes data replication easier when schema IDs are embedded in the data. This is important because replicated data can only be deserialized with compatible schemas on the replication target site.

When IDs reach the range maximum value, adding more schema fails until a new ID range is configured. A new range must be larger than the current one. The new range must also not overlap with any other ranges.

Important: Cloudera does not recommend that you import schemas into a Schema Registry's ID range. IDs assigned sequentially from the range start value might reach the imported IDs.

Setting a Schema Registry ID range

Learn how to set an ID range for schemas in Schema Registry using Cloudera Manager.

About this task

An ID range that is unique to a Schema Registry instance can be set in Cloudera Manager on the Schema Registry service's configuration page. Configuring an ID range does not alter already existing schemas.

Before you begin

- Ensure that the cluster, its hosts, and all its services are healthy.
- Ensure that Schema Registry is commissioned and running.
- Ensure that you have access to all credentials that are required to access and use Schema Registry.
- Plan ahead and designate ID ranges for each Schema Registry instance. Ensure that the planned ranges do not overlap.
- Ensure that the ID range you plan to configure has enough IDs. Base the size of your range on the approximate number of schemas you expect to have.

Procedure

1. In Cloudera Manager, select the Schema Registry service.
2. Go to Configuration.
3. Find and configure the following properties:
   - ID Offset Range Minimum
   - ID Offset Range Maximum

   Configure the properties according to your planned ID ranges. For example, if you want to have an ID range of 1-100, set ID Offset Range Minimum to 1 and ID Offset Range Maximum to 100.
4. Click Save Changes.
5. Restart the Schema Registry.

Results

When new schemas are added, either with the UI or the API, the schema IDs are allocated from the configured range.