Confluent Cloud Services Agreement Exit Assistance

Transferring Data Out of Confluent Cloud

This document serves to provide general guidance in transferring your data from a Confluent Cloud Managed cluster to an on-premise Open Source version of Apache Kafka once you terminate your Confluent Cloud Services Agreement with Confluent.

The objective of this document is to ensure that you are able to effect a smooth transition of your Content from Confluent Cloud to an alternative solution of your choice.

N.B There is not just one way to exit from your Confluent Cloud Managed Cluster, the option you choose to take may depend on your business application and requirements as stated this is overall guidance.

Defined terms used in this document have the same meanings given to them in your Agreement with Confluent.

Determining What Data is Important

You are fully responsible for ensuring that the environment that you wish to move your Content to from Confluent Cloud is adequately sized to meet your needs. Confluent is unable to advise you in this regard.

Confluent will work together with you prior to the termination of your Agreement with Confluent in order to manage a smooth transfer of the Content from the Cloud Service when the contract is terminated or expires.

Transfer Using Confluent Replicator

Confluent Cloud does not expose the ZooKeeper configuration to users. Confluent Replicator does not require a direct connection to Apache ZooKeeper™.

In this example, the source is the current CCloud cluster and the destination is the On-Premise open-source Apache Kafka cluster . A separate node called a worker node is also configured to run Confluent Replicator.

1. Configure replicator to bootstrap to the destination kafka cluster, and source from CCloud, see

A sample configuration would be like:

```
{
  "name": "replicate-topic",
  "config": {
    "connector.class": "io.confluent.connect.replicator.ReplicatorSourceConnector",
    "topic.whitelist": "srctopic",
    "topic.rename.format": "srctopic_replica",
    "key.converter": "io.confluent.connect.replicator.util.ByteArrayConverter",
    "value.converter": "io.confluent.connect.replicator.util.ByteArrayConverter",
    "dest.kafka.bootstrap.servers": "<destination Apache Kafka cluster>",
    "dest.kafka.security.protocol": "SASL_SSL", #optional if security is enabled on destination
    "dest.kafka.sasl.jaas.config": "org.apache.kafka.common.security.plain.PlainLoginModule required
username=\"dst-key\" password=\"dst-secret\";",#optional if security is enabled on destination
    "dest.kafka.sasl.mechanism": "PLAIN",#optional if security is enabled on destination
    "src.kafka.bootstrap.servers": "<your ccloud bootstrap endpoint>",
    "src.kafka.security.protocol": "SASL_SSL",
    "src.kafka.sasl.jaas.config": "org.apache.kafka.common.security.plain.PlainLoginModule required
username=\"ccloud-key\" password=\"ccloud-secret\";",
    "src.kafka.sasl.mechanism": "PLAIN",
    "tasks.max": "1", #adjust to match the data volume you need to backup
    "confluent.topic.replication.factor": "1",
    "confluent.license": "license here"  #Optional You can use the 30 days trial license
  }
}
```

2. Run Confluent Replicator until all past data has been copied. Confluent Replicator reports lag in the JMX metrics that are exposed by the replicator worker JVMs. Each client emits its offset via an MBean with the following name:

```
kafka.consumer:type=consumer-fetch-manager-metrics,client-id={client.id}
```

Within the MBean, the following metrics measure lag:

```
{topic}-{partition number}.records-lag
```

The lag metrics should gradually approach zero and stabilize as Confluent Replicator runs.

At this point, the destination cluster is in sync with the origin CCloud cluster, and Confluent Replicator is copying data that is currently being produced to your origin cluster in near real-time.

Retrieving Your Schemas from CCloud Schema Registry

If you need to backup existing schemas from CCloud Schema Registry, you can follow those steps:

- List every subjects using the REST API
  https://docs.confluent.io/current/schema-registry/develop/api.html#get--subjects
  `Example : curl --user <SR_API_KEY>:<SR_API_PASSWORD>`
  `https://<CCLOUD_SR_ENDPOINT>/subjects`
  That will return a list of subject :
  - subject1
  - subject2 and so on

- For every subject, list the existing subject versions using the REST API
  https://docs.confluent.io/current/schema-registry/develop/api.html#get--subjects-(string-%20subject)-versions
  `Example : curl --user <SR_API_KEY>:<SR_API_PASSWORD>`
  `https://<CCLOUD_SR_ENDPOINT>/subjects/<subjectName>/versions`
  That will return a list of versions :
  - 1
  - 2 and so on

- For every subject/version, download the corresponding schema using the REST API
  https://docs.confluent.io/current/schema-registry/develop/api.html#get--subjects-(string-%20subject)-versions-(version-id-%20version)
  `Example : curl --user <SR_API_KEY>:<SR_API_PASSWORD>`
  `https://<CCLOUD_SR_ENDPOINT>/subjects/<subjectName>/versions/<version>`
  That will return a JSON object like the following :

```
{
  "subject": "mySubject",
  "version": 1,
  "id": 1,
  "schema": "Your Schema"
}
```

- You can then extract the schema content from the object, using for instance JQ command line tool :
  https://stedolan.github.io/jq/
  `Example : curl --user <SR_API_KEY>:<SR_API_PASSWORD>`
  `https://<CCLOUD_SR_ENDPOINT>/subjects/<subjectName>/versions/<version> | jq '.schema'`
  This will only output the schema field value, being "Your Schema" in the previous example

At that point you will be have downloaded every schemas from CCloud Schema Registry and you will be able to use them independently from Confluent Cloud

Validation of Successful Transition

Please ensure you validate successful transition back to your On-Premise open-source Apache Kafka cluster.

Customer's Responsibility for Content Export

Please note, you are solely responsible for exporting your Content from the Cloud Service prior to expiration or termination of your Agreement with Confluent.

You further acknowledge that following termination of your Agreement with Confluent you will have no further access to any Content on the Cloud Service.

Suggested Reading

For more about Replicator, see Multi-DC Deployment Architectures.

Confluent's Data Processing Addendum for Customers, at https://www.confluent.io/cloud-customer-dpa

Confluent's Cloud Security Addendum, located at https://www.confluent.io/cloud-enterprise-security-addendum/

Last Updated: September 2020