Cloudera Data Flow for Data Hub 7.2.2

Analyzing data with Apache Kafka in CDP Public Cloud

Date published: 2020-06-23
Date modified: 2020-06-23

CLOUDERA

https://docs.cloudera.com/

Legal Notice

Contents

Understand the use case

As a basic analytical use case, you can use Streaming Analytics clusters connected to the Streams Messaging cluster to build an end-to-end streaming analytics application with Kafka as source and sink.

When choosing Kafka as a connector for a Flink application, you can create a scalable communication channel. Kafka, as a source and sink, is responsible for delivering input records to Flink, and receiving the transformed data from Flink.

This use case details the steps to connect a Streaming Analytics cluster with a Streams Messaging cluster, and to submit your Flink jobs and monitor the correct behaviour of your application in the Data Hub environment. The Stateful Flink Application Tutorial is used as an example to guide you through the basic steps to analyze your data with Kafka in CPD Public Cloud.

The Stateful Flink Application Tutorial details an inventory use case for an e-commerce site. The business logic in the application includes handling transactions and managing queries that are later summarized to monitor the state of the inventory.

Note: In the Analyzing data with Kafka use case, the steps that are used from the Stateful Flink Application Tutorial are highlighted in grey. These steps are not mandatory to use Streaming Analytics in Data Hub, but serves as an example for practice.

For more information about the tutorial, see the Stateful Flink Application Tutorial.

Related Information

Stateful Flink Application Tutorial

Prepare your environment

Before you start analyzing your data using Flink and Kafka in CDP Public Cloud, you need to register and prepare your environment so that a chosen user or group can use the clusters and services in the environment, and submit Flink jobs securely.

As a first step, an admin needs to register an AWS or Azure environment. In CDP Public Cloud, this environment is based on the virtual private network in your cloud provider account. This means that CDP can access and identify the resources in your cloud provider account which is used for CDP services, and shared between clusters in the environment. After registering your AWS or Azure environment, you can provision clusters, and set users or groups to access your environment and services.

For more information about registering an AWS or Azure environment, see the Management Console documentation.

In this use case, an administrator with AdminEnvironment resource role registers an AWS environment where flink_users group is added as EnvironmentUser. The members of flink_users have the same resource role that is set for the group.

Assign resource roles

As an administrator, you need to give permissions to users or groups to be able to access and perform tasks in your Data Hub environment.

Procedure

1. Navigate to Management Console > Environments and select your environment.
2. Click Actions > Manage Access .
3. Search for a user or group that needs access to the environment.

4. Select EnvironmentUser role from the list of Resource Roles.
5. Click Update Roles.
   The Resource Role for the selected user or group will be updated.
6. Navigate to Management Console > Environments , and select the environment where you want to create a cluster.
7. Click Actions > Synchronize Users to FreeIPA .
8. Click Synchronize Users.

   Note: There might be cases where the status of the environment is synchornized with warnings and has failed status. This does not indicate that the synchronization has failed.

   The following short video also details the procedure how to assign the resource roles.

Create IDBroker mapping

As an administrator, you must create IDBroker mapping for a user or group to access cloud storage. As a part of Knox, the IDBroker allows a user to exchange cluster authentication for temporary cloud credentials.

Procedure

1. Navigate to Management Console > Environments and select your environment.
2. Click Actions > Manage Access .
3. Click on the IDBroker Mappings tab.
4. Click Edit to add a new user or group and assign roles to have writing access for the cloud storage.
5. Search for the user or group you need to map.
6. Go to the IAM Summary page where you ca nfind information about your cloud storage account.
7. Copy the Role ARN.
8. Go back to the IDBroker Mapping interface on the Cloudera Management Console page.
9. Paste the Role ARN to your selected user or group.
10. Click Save and Sync.

   The following short video also details the procedure how to create the IDBroker mapping.

Set workload password

As a user, you need to set a workload password for your EnvironmentUser account to be able to access the Flink nodes through SSH connection.

Procedure

1. Navigate to Management Console > Environments and select your environment.
2. Click Actions > Manage Access .
3. Click Workload Password.
4. Give a chosen workload password for your user.
5. Confirm the given password by typing it again.
6. Click Set Workload Password.

   The following short video also details the procedure how to set the workload password.

Create your streaming clusters

As a user, you need to create the Streaming Analytics and Streams Messaging clusters. In this use case, the streaming clusters are created in the same Data Hub environment.

Procedure

1. Navigate to Management Console > Environments , and select the environment where you want to create a cluster.
2. Click Create Data Hub.
3. Select Streaming Analytics Light Duty cluster from Cluster Definition.
4. Provide a chosen cluster name.
5. Click Provision Cluster.
6. Navigate back to Management Console > Environments , and select the environment where you have created the Streaming Analytics cluster.
7. Click Create Data Hub.
8. Select Streams Messaging Light Duty cluster from Cluster Definition.
9. Provide a chosen cluster name.
10. Click Provision Cluster.

    The following short video also details the procedure how to create the streaming clusters.

Set Ranger policies

As an administrator, you need to set resource-based Ranger policies to give users or groups access to write and read Kafka topics.

Procedure

1. Navigate to Management Console > Environments , and select the environment where you have created your cluster.
2. Click on the Data Lake tab.
3. Select Ranger from the services.
   You are redirected to the Ranger user interface.
4. Select your Streams Messaging cluster under the Kafka folder on the Service Manager page.
5. Click Add new policy.
6. Provide the details for the topic policy.
   a) Give a chosen policy name.
   b) Select topic as Kafka resource type.
   c) Provide a prefix that is included in the name of the flink topics you need to access when writing messages.
7. Select the user or groups under Allow Conditions.
8. Click Add Permissions.
9. Select individual or all permissions for the policy.
10. Click Add.
    You are redirected to the List of Policies page.
11. Click Add new policy.

12. Provide the details for the consumer group policy.
    a) Give a chosen policy name.
    b) Select consumergroup as Kafka resource type.
    c) Provide a prefix that is included in the name of the flink topics you need to access when reading messages.
13. Select the user or groups under Allow Conditions.
14. Click Add Permissions.
15. Select individual or all permissions for the policy.
16. Click Add.

    The following short video also details the procedure how to set the Ranger policies.

Retrieve keytab file

As a user, you need to retrieve the keytab file of your profile and upload it to the Streaming Analytics cluster to securely submit your Flink jobs.

Procedure

1. Navigate to Management Console > Environments , and select the environment where you have created your cluster.
2. Click on your profile name.
3. Click Profile.
4. Click Actions > Get Keytab.
5. Choose the environment where your Data Hub cluster is running.
6. Click Download.
7. Save the keytab file in a chosen location.

   The following short video also details the procedure how to retrieve the keytab file.
8. Upload your keytab file to the Streaming Analytics cluster.

```
scp <location>/<your_keytab_file> <workload_username>@<manager_node_FQDN
>:.
                        Password:<your_workload_password>
```

Create Atlas entity type definitions

As an administrator, you must create the Atlas entity type definitions before submitting a Flink job, and enable Atlas in Cloudera Manager to use Atlas for metadata management.

Procedure

1. Copy and paste the following command:

```
curl -k -u <atlas_admin>:<atlas_admin_pwd> --location --request
```

2. Provide your workload username and password as atlas admin.
3. Navigate to Management Console > Environments , and select the environment where you have created your cluster.
4. Click Data Lake.
5. Click Endpoints.
6. Click on the copy icon beside the Atlas endpoint.

7. Replace the atlas server URL with the copied Atlas endpoint.

```
POST '<atlas_endpoint_url>/v2/types/typedefs'
```

8. Copy and paste the entity type definitions.

```
--header 'Content-Type: application/json' \
--data-raw '{
    "enumDefs": [],
    "structDefs": [],
    "classificationDefs": [],
    "entityDefs": [
        {
            "name": "flink_application",
            "superTypes": [
                "Process"
            ],
            "serviceType": "flink",
            "typeVersion": "1.0",
            "attributeDefs": [
                {
                    "name": "id",
                    "typeName": "string",
                    "cardinality": "SINGLE",
                    "isIndexable": true,
                    "isOptional": false,
                    "isUnique": true
                },
                {
                    "name": "startTime",
                    "typeName": "date",
                    "cardinality": "SINGLE",
                    "isIndexable": false,
                    "isOptional": true,
                    "isUnique": false
                },
                {
                    "name": "endTime",
                    "typeName": "date",
                    "cardinality": "SINGLE",
                    "isIndexable": false,
                    "isOptional": true,
                    "isUnique": false
                },
                {
                    "name": "conf",
                    "typeName": "map<string,string>",
                    "cardinality": "SINGLE",
                    "isIndexable": false,
                    "isOptional": true,
                    "isUnique": false
                },
                {
                    "name": "inputs",
                    "typeName": "array<string>",
                    "cardinality": "LIST",
                    "isIndexable": false,
                    "isOptional": false,
                    "isUnique": false
                },
                {
                    "name": "outputs",
                    "typeName": "array<string>",
                    "cardinality": "LIST",
                    "isIndexable": false,
                    "isOptional": false,
                    "isUnique": false
                }
            ]
        }
    ],
    "relationshipDefs": []
}'
```

The following short video also details the procedure how to create the Atlas entity type definitons.

9. Navigate to Management Console > Environments , and select the environment where you have created your cluster.
10. Click on the Streaming Analytics cluster.
11. Select Cloudera Manager UI from the Services.
12. Select Flink from the list of clusters.
13. Click Configuration.
14. Search for enable atlas in the search bar.
15. Make sure that Atlas is enabled in the configurations.

The following short video also details the procedure how to enable Atlas for Flink in Cloudera Manager.

Analyzing your data with Kafka

After preparing your environment, you need to connect Kafka to Flink, and create Kafka topics where the input and output data is messaged. When Kafka is ready, you need to generate data to your Kafka topic and let Flink apply the computations you have added in your application design.

Procedure

1. Create your streaming Flink application with Kafka as source and sink.

   The Stateful Tutorial has a detailed guide how to create your application using Kafka as source and sink. For more information, see the Setting up Kafka inputs and outputs section.

2. Build your Flink project with maven after creating the application logic.

   ```
   mvn clean package
   ```

3. Upload your Flink project to the Streaming Analytics cluster.

   ```
   scp <location>/flink-stateful-tutorial-1.2-SNAPSHOT.jar <your_workload_u
   sername>@<manager_node_FQDN>:.
   Password:<your_workload_password>
   ```

4. Upload your keytab file to the Streaming Analytics cluster, if you have not uploaded it yet.

   ```
   scp <location>/<your_keytab_file> <your_workload_username>@<manager_node
   _FQDN>:.
   Password:<your_workload_password>
   ```

   Important: You will not be able to submit Flink jobs on a Data Hub cluster if you do not upload your keytab file to the Streaming Analytics cluster.

5. Configure the Flink job properties to securely connect to the Streams Messaging cluster.

Using the job.properties file from the Stateful Tutorial, replace your kafka brokers and trustore location. The input and output topics have "flink" as a prefix in the following example to comply with the set Ranger policies.

```
kafka.bootstrap.servers=<your_kafka_broker>:9093,<your_kafka_broker1>:90
93,<your_kafka_broker2>:9093
kafka.group.id=flink
kafka.flink.partition-discovery.interval-millis=60000
kafka.retries=3
transaction.input.topic=flink.transaction.log.1
generate.queries=false
query.input.topic=flink.query.input.log.1
query.output.topic=flink.query.output.log.1
num.items=1000000
sleep=100

kafka.security.protocol=SASL_SSL
kafka.sasl.kerberos.service.name=kafka
kafka.ssl.truststore.location=/var/lib/cloudera-scm-agent/agent-cert/cm-
auto-global_truststore.jks
```

Note: You can use the following command to get the user related truststore location and password:

```
cat /etc/flink/conf/flink-conf.yaml | grep truststore
```

After running the command, truststore location and password is displayed in the command line.

To retrieve the Kafka broker hostnames:

a. Navigate to Management Console > Environments , and select the environment where you have created your cluster.
b. Click on the Streams Messaging cluster.
c. Click Hardware.
d. Search for the Kafka brokers.
e. Click the copy icon next to the Kafka brokers to obtain the hostnames.

Note: The job.properties file shown in the video contains JAAS configuration for secure Kafka connection. You do not need to add the JAAS configuration property as it is automatically generated, but you can overwrite the default setting by adding the configuration to the job.properties file shown in the video.

6. Upload the Flink job properties file to the Streaming Analytics cluster.

```
scp <location>/config/job.properties <your_workload_username>@<manager_n
ode_FQDN>:.
Password:<your_workload_password>
```

7. Use ls command to list and check if you have every necessary file on your Streaming Analytics cluster.

After listing the files on your cluster, you should be able to see your keytab file, the Stateful Tutorial jar file and the job.properties file.

8. Create your topics in Streams Messaging Manager (SMM).
   a) Go to Management Console > Data Hub Clusters.
   b) Search for your Streams Messaging cluster.
   c) Open SMM from the list of Services.
   d) Go to Topics > Add new.
   e) Provide information for Topic Name, Partitions, Availability, and Limits.

   Note: Make sure to give the topic its name based on the prefix previously provided for the Ranger policy.

   Create three topics with 16 partitions for each. You need to name them flink.tutorial.transaction.log, flink.tutorial.query.input.log, and flink.tutorial.query.input.log to comply with the Stateful Flink Application Tutorial.
9. Start generating data to your target Kafka topic.

   Important: You must pass the Kerberos keytab file and principal name to the Flink run command in the following format:

   ```
   -yD security.kerberos.login.keytab=<your_keytab_filename> \
                               -yD security.kerberos.login.principal=<c
   sso_name> \
   ```

   Submit the Data Generator job from the Stateful Flink Application Tutorial.

   ```
   flink run -yD security.kerberos.login.keytab=<your_keytab_filename> \
   -yD security.kerberos.login.principal=<csso_name> \
   -m yarn-cluster -d -p 2 -ys 2 -ynm Datagenerator \
   -c com.cloudera.streaming.examples.flink.KafkaDataGeneratorJob \
   flink-stateful-tutorial-1.2-SNAPSHOT.jar job.properties
   ```
10. Open SMM and check if the data is generated.

    After waiting a few minutes, you can notice the Data in count increases as the generated data arrives to the Kafka topic. You can check the incoming data in the Data explorer tab.
11. Deploy your Flink streaming application.

    Submit the Transaction Job from the Stateful Flink Application Tutorial.

    ```
    flink run -yD security.kerberos.login.keytab=<your_keytab_filename> \
    -yD security.kerberos.login.principal=<csso_name> \
    -m yarn-cluster -d -p 2 -ys 2 -ynm TransactionProcessor \
    flink-stateful-tutorial-1.2-SNAPSHOT.jar job.properties
    ```
12. Open SMM and check the log of your application.

    You can notice how the query data results are shown in SMM.

What to do next

You have the following options to monitor and manage your Flink applications:

Job monitoring with Flink Dashboard

After submitting a Flink job, you can always use the Flink Dashboard to review if the job submission was successful. Later you can use the Flink Dashboard to monitor the history of all your submitted and completed jobs.

You can access the Flink Dashboard directly from your Data Hub cluster.

1. Go to Management Console > Data Hub Clusters.

2. Search for your Streaming Analytics cluster.
3. Select Flink Dashboard from the list of Services.

You are redirected to the Flink Dashboard user interface where you can select from the submitted jobs.

For Data Generator Job

For Transaction Processor Job

Note: You cannot save the Completed jobs into cloud storage.

Metadata governance with Atlas

You can use Atlas to find, organize and manage different assets of data about your Flink applications and how they relate to each other. This enables a range of data stewardship and regulatory compliance use cases.

Procedure

1. Go to Management Console > Data Lakes .
2. Search for your environment from the list of available environments.
3. Select Atlas from the list of services.
4. Search and select flink_application from the Search By Type bar.
5. Click the Name of your application.
6. Select Properties, Lineage, Relationships, Classifications or Audits tabs for more information about your application.

   For Data Generator Job

   For Transaction Processor Job

Related Information

Flink metadata collection using Atlas

Data querying with SQL Client

Learn more about querying data with Flink SQL Client using Kafka catalogs.

Before you begin

You need to configure the SQL Client using the following methods:

- SQL Client Default settings - default SQL configurations that are specific for Catalogs and connectors in the /etc/flink/conf/sql-client-defaults.yaml file.
- Custom environment settings - using the -e sql-env.yaml paramter when starting the SQL client can override the regular Flink and Default SQL settings.

As every Flink SQL query is an independent Flink job, you can decide if you want to run them as standalone (per-job) YARN applications, or you can run them on a Flink session cluster.

For more information about the needed configurations, see the SQL Client documentation.

Important: You need to retrieve the user keytab file before starting the SQL Client. For more information, see the Retrieve keytab file section.

Procedure

1. Start the SQL Client.

   flink-sql-client embedded (-e sql-env.yaml)
2. Create a source table based on your streaming application parameters.

   The following table can be created when using the Stateful Flink Application tutorial:

```
CREATE TABLE ItemTransactions (
 transactionId    BIGINT,
 ts    BIGINT,
 itemId    STRING,
 quantity INT,
 event_time AS CAST(from_unixtime(floor(ts/1000)) AS TIMESTAMP(3)),
 WATERMARK FOR event_time AS event_time - INTERVAL '5' SECOND
) WITH (
 'connector.type'      = 'kafka',
 'connector.version'   = 'universal',
 'connector.topic'     = 'transaction.log.1',
 'connector.startup-mode' = 'earliest-offset',
 'connector.properties.bootstrap.servers' = '<broker_address>',
 'connector.properties.security.protocol'            = 'SASL_SSL',
       'connector.properties.sasl.kerberos.service.name'   = 'kafka',
       'connector.properties.ssl.truststore.location'      = '<absolute_
path_to_jks>',
       'format.type' = 'json'
      );
```

3. Check that the table creation was successful with the SHOW TABLES command.
4. Use the SELECT query to check how the input data is generated.

   The following SELECT query can be used when using the Stateful Flink Application tutorial:

```
> SELECT * FROM ItemTransactions;
```

5. Press Q to exit the SELECT query and to stop the Flink job.
6. Use the different SQL statements to create more queries about your streaming application.

Related Information

Flink SQL Tutorial

SQL Client

Retrieve keytab file