

Kafka Connect GCS Sink Service

Kafka connect GCS sink service is used to consume records from Kafka topic and persist the record in google cloud storage. In this documentation we can find the details of the topics, consumer groups and storage details. Also we can know more about the configurations we have in this sink application.

Prerequisites - [Kafka connect overview link](#)

Repo - <https://gecgithub01.walmart.com/dsi-dataventures-luminate/audit-api-logs-gcs-sink>

ADT - [API Logs for Product Metrics](#) - ADT

Topic - Publisher service publishes Kafka record to the topic. From here the data will be read by the Kafka connect.

Environment	Topic	Link
stage	api_logs_audit_stg	https://lenses.kafka-v2-luminate-core-stg.eus.prod.us.walmart.net:8080/data/topics/api_logs_audit_stg/data , https://lenses.kafka-v2-luminate-core-stg.scus.prod.us.walmart.net:8080/data/topics/api_logs_audit_stg/data
prod	api_logs_audit_prod	https://lenses.kafka-v2-luminate-core-prod.eus.prod.us.walmart.net:8080/data/topics/api_logs_audit_prod/data , https://lenses.kafka-v2-luminate-core-prod.scus.prod.us.walmart.net:8080/data/topics/api_logs_audit_prod/data

Consumer group - Kafka connect service application acts as the consumer group. We need to subscribe to a specific topic. Each region (eus2, scus) has one instance of consumer group running.

Consumers	stage	prod
connect-audit-log-gcs-sink-connector	https://lenses.kafka-v2-luminate-core-stg.eus.prod.us.walmart.net:8080/data/consumers/connect-audit-log-gcs-sink-connector , https://lenses.kafka-v2-luminate-core-stg.scus.prod.us.walmart.net:8080/data/consumers/connect-audit-log-gcs-sink-connector	https://lenses.kafka-v2-luminate-core-prod.eus.prod.us.walmart.net:8080/data/consumers/connect-audit-log-gcs-sink-connector , https://lenses.kafka-v2-luminate-core-prod.scus.prod.us.walmart.net:8080/data/consumers/connect-audit-log-gcs-sink-connector

Flush configuration - We have flush configuration defined at GCS service. The Kafka records will be cached at the sink connector until the threshold is reached. Once any of the flush configuration threshold is reached the accumulated Kafka records will be flushed to the GCS bucket and offset will be committed. **Note** - We'll see the lag pile up in the consumer when any of the threshold is yet to be met. This needs to be considered when setting up the lag related alerts.

Flush property	Threshold
size	50 MB
interval	10 Mins
count	5000 records

Storage - The sink service is storing the data into the buckets in google cloud storage (GCS) after consuming the Kafka record from specific topic. Currently we are consuming records from US **wm-site-id** - **1704989259133687000**. Canada, Mexico records will not be consumed in the first release.

Environment	Bucket - US
stage	audit-api-logs-us-test
prod	audit-api-logs-us-prod

Folder structure - Below is the folder structure we see in the GCS.

```

<schema_name>
<bucket_name>
.index-eus2
.index-scus
api_logs
  <service_name=<Service_Name>>
    <date=<yyyy-mm-dd>>
      <endpoint_name=<endpoint_name>>
        <topic_name>(<partition>_<first_offset_after_flush>)<_region>.parquet

```

Folder / File	Description
schema_name	Schema name defined for hive table creation Data Discovery - Hive Tables & Workflow
bucket_name	Bucket name of GCS
.index-eus2	Folder which holds the offset count of eus2 cluster. This is a connector property which can be updated from GCS application. Used in appending same for the file generated.
.index-scus	Folder which holds the offset count of scus cluster. This is a connector property which can be updated from GCS application. Used in appending same for the file generated.
api_logs	Root folder where the Kafka records are stored.
service_name=<Service_Name>	This is one of the partition key. Read from Kafka record. Ex service_name=NRT
date=<yyyy-mm-dd>	This is one of the partition key. Inserted by the lenses connector. Ex date=2025-03-12
endpoint_name=<endpoint_name>	This is one of the partition key. Read from Kafka record. Ex endpoint_name=transactionHistory
<topic_name> (<partition>_<first_offset_after_flush>) <_region>.parquet	File name which holds the actual Kafka records in parquet format. Ex api_logs_audit_stg (2_000003185231)_eus2.parquet

Sink service load test - Some load tests were run on the up-stream system. Message produced v/s message consumed is recorded here for the sink service - [Product metrics - Audit API Logs Sink Service Load Testing](#). Further this will be updated with more testing links