

Cloud Tools Guidance

Author: Michael Kohs George Vetticaden Timothy Spann

Date: 04/19/2023

Last Updated: 5/01/2023

Notice

This document assumes that you have registered for an account, activated it and logged into the CDP Sandbox. This is for authorized users only who have attended the webinar and have read the training materials.

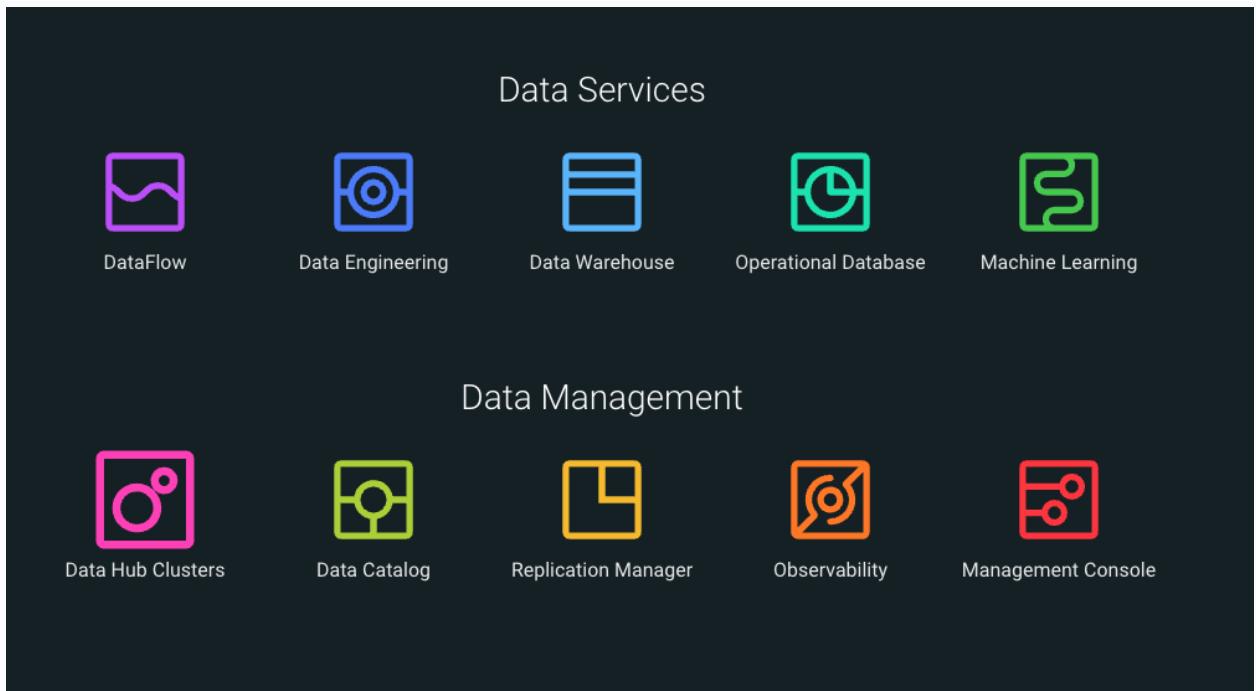
A short guide and references are listed [here](#).

THIS IS NOT FOR USE WITH THE FIRST TWO TUTORIALS. THIS IS FOR BUILDING ASSETS FOR YOUR OWN NEW FLOWS.

1. How To Build Data Assets

1.1 Create a Kafka Topic

1. Navigate to Data Hub Clusters



2. Navigate to the **oss-kafka-demo** cluster

The screenshot shows the Cloudera Management Console interface. On the left is a sidebar with navigation links: Dashboard, Environments, Data Lakes, User Management, **Data Hub Clusters** (highlighted in red), Data Warehouses, My Workspaces, Classic Clusters, Audit, Shared Resources, and Global Settings. The main content area is titled 'Data Hubs' and shows a table of running clusters:

Status	Name	Cloud Provider	Environment	Data Hub Type	Version	Node Count	Created
Running	oss-flink-demo	AWS	oss-demo-aws	7.2.16 - Streaming Analytics Light Duty with Apache Flink	CDH 7.2.16	6	03/21/23, 03:20 PM EDT
Running	oss-kafka-demo	AWS	oss-demo-aws	7.2.16 - Streams Messaging Light Duty: Apache Kafka, Schema Registry, Streams Messaging Manager, Streams Replication Manager, Cruise Control	CDH 7.2.16	4	03/21/23, 03:20 PM EDT
Running	oss-kudu-demo	AWS	oss-demo-aws	7.2.16 - Real-time Data Mart: Apache Impala, Hue, Apache Kudu, Apache Spark	CDH 7.2.16	7	03/21/23, 03:20 PM EDT
Running	oss-nifi-demo	AWS	oss-demo-aws	7.2.16 - Flow Management Light Duty with Apache Nifi, Apache NiFi Registry	CDH 7.2.16	4	03/21/23, 03:20 PM EDT

3. Navigate to Streams Messaging Manager



Data Hubs / oss-kafka-demo / Event History

oss-kafka-demo

cm-cdp-datalake.us-west-1-52519921-84c5-45c4-af51-c008a6eb1c9-cluster-b2a68dc9-693e-40df-b032-664123c1c2b6

STATUS Running **NODES** 4 **CREATED AT** 03/21/23, 03:20 PM EDT **CLUSTER TEMPLATE** 7.2.16 - Streams Messaging Light Duty, Apache Kafka, Schema Registry, Streams Messaging Manager, Streams Replication Manager, Cruise Control

STATUS REASON Synced instance states with the cloud provider.

aws Environment Details

NAME oss-demo-aws **DATA LAKE** oss-ow-dl **CREDENTIAL** oss-credential-aws **REGION** us-east-2 **AVAILABILITY ZONE** N/A

Services CM-U Schema Registry Streams Messaging Manager Token Integration

CM-U Cloudera Manager Info

CM URL: https://oss-kafka-demo-gateway.oss-demo.qsm5-opic.cloudera.svc.oss-kafka-demo/cdp-proxy/cm/home/ CM VERSION: 7.8.0 RUNTIME VERSION: 7.2.16-1-cdb7.2.16.p2.38683602 LOGS: Command logs, Service logs

Event History Autocreate Endpoints (5) Tags (8) Nodes Network Load Balancers Telemetry Repository Details Image Details Recipes (0) Cloud Storage Database Upgrade

Events Show All Autocreate Cluster

- Cloudera Manager reported that node(s) oss-kafka-demo-master0 oss-demo.qsm5-opic.cloudera.svc status became HEALTHY. 3/27/2023, 12:48:31 AM
- COP services have been installed 3/25/2023, 1:42:27 PM
- Installing COP services 3/25/2023, 1:38:05 PM
- Pre-flight STS endpoint accessibility check result: OK 3/25/2023, 1:37:25 PM
- Pre-flight S3 endpoint accessibility check result: OK 3/25/2023, 1:37:25 PM
- Pre-flight Service cloudera.com accessibility check result: OK 3/25/2023, 1:37:25 PM
- Pre-flight Datadis S3 API (https://cloudera-dbus-prod.s3.amazonaws.com) accessibility check result: OK 3/25/2023, 1:37:25 PM

DOWNLOAD

Info: Streams Messaging Manager (SMM) is a tool for working with Apache Kafka.

4. Now that you are in SMM.

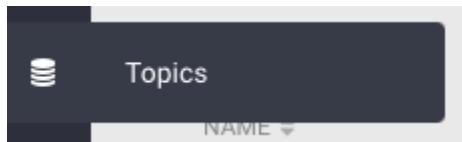
Overview

Cluster: kafka-10fd

Producers		Brokers		Topics		Consumer Groups	
TOPICS (35)	BROKERS (3)	NAME	DATA IN	DATA OUT	MESSAGES IN	CONSUMER GROUPS	CURRENT LOG SIZE
Producers (14)	ACTIVE (11)	__consumer_offsets	49 KB	49 KB	481	0	7 MB
	PASSIVE (0)	__CruiseControlMetrics	778 KB	778 KB	69k	0	31 MB
	ALL	__KafkaCruiseControlModelTrainingSamples	30 KB	0B	90	0	207 KB
		KafkaCruiseControlSampleCoreProducer	8 KB				
		CloudDataConnectMetricsReporter	240				
		CruiseControlMetricsReporter	69K				
		connector-producer-MirrorHeartbeatConnector-0	1.8K				
		Generator-producer	4.3K				
		george_veticaiden_coff	847				
		__smm_alert_notifications	0B	0B	0	0	0B
		__smm_consumer_metrics	0B	0B	0	1	0B
		__smm_producer_metrics	16 KB	16 KB	240	1	4 MB
		__smm-app-consumer-metrics-keys-index-chan	0B	0B	0	0	0B
		elog	0B	0B	0	0	157 B
		__smm-app-smm-consumer-table-15m-changelog	0B	0B	0	0	0B
		on	0B	0B	0	1	0B
		__smm-app-smm-consumer-table-15m-repartition	0B	0B	0	0	0B
		og	0B	0B	0	0	0B
		__smm-app-smm-consumer-table-30s-changelog	0B	0B	0	0	0B

Consumer Groups (6)

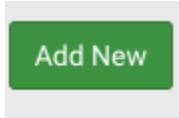
5. Navigate to the round icon third from the top, click this **Topic** button.



6. You are now in the Topic browser.

Topics								
	Bytes In 2 MB	Bytes Out 1 MB	Produced Per Sec 3	Fetched Per Sec 1,825	In Sync Replicas 816	Out Of Sync 0	Under Replicated 0	Offline Partitions 0
Topics (34)								
 _smm-app-smm-producer-table-30s-repartition	11 KB	11 KB	237	1	445 KB	  		
 _smm-app-smm-producer-table-15m-changelog	10 KB	0B	236	0	512 KB	  		
 heartbeats	172 KB	0B	1.8k	0	1 MB	  		
 srmm-service-status-connector-metrics-minutes-store-changelog	18 KB	0B	90	0	6 MB	  		
 tim_syslog_critical	0B	0B	0	0	N/A	  		
 _smm-app-smm-producer-table-15m-repartition	11 KB	11 KB	237	1	445 KB	  		
 _smm-service-cluster-metrics-minutes-store-changelog	0B	0B	0	0	0B	  		
 _smm-app-smm-consumer-table-15m-repartition	0B	0B	0	1	0B	  		
 _smm-app-smm-consumer-table-30s-repartition	0B	0B	0	1	0B	  		
 _smm_consumer_metrics	0B	0B	0	1	0B	  		
 _KafkaCruiseControlPartitionMetricSamples	171 KB	0B	8.9k	0	2 MB	  		
 srmm-metrics.secondary.internal	0B	0B	0	1	0B	  		
 _smm-app-smm-consumer-table-15m-changelog	0B	0B	0	0	0B	  		

7. Click **Add New** to build a new topic.



8. Enter the name of your topic prefixed with your “**Workload User Name**“ <yourusername>_yournewtopic, ex: tim_younewtopic.

Add Topic

TOPIC NAME

PARTITIONS

Availability

 MAXIMUM	 HIGH	 MODERATE	 LOW	 CUSTOM
---	--	--	--	--

REPLICATION FACTOR 3	REPLICATION FACTOR 3	REPLICATION FACTOR 2	REPLICATION FACTOR 1
MIN INSYNC REPLICA 2	MIN INSYNC REPLICA 1	MIN INSYNC REPLICA 1	MIN INSYNC REPLICA 1

Limits

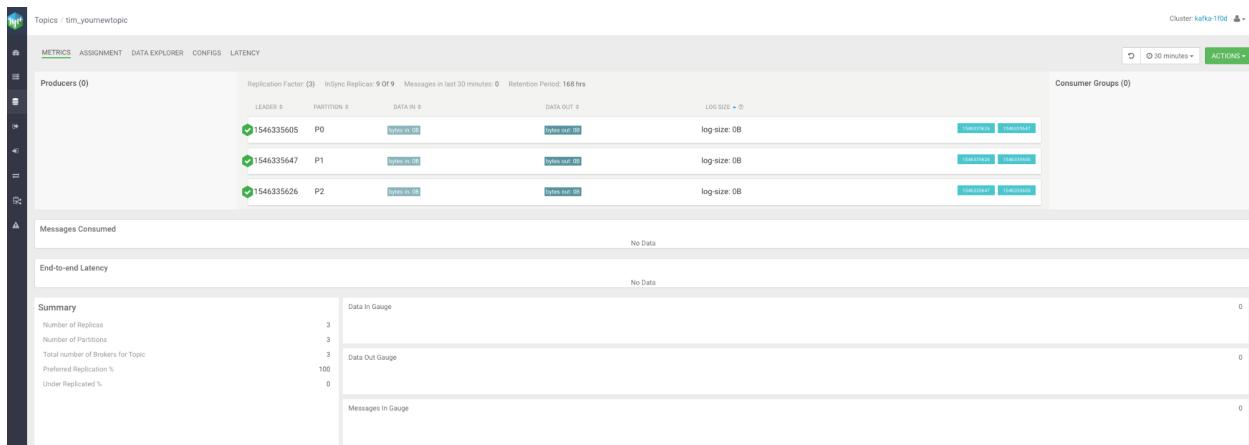
CLEANUP.POLICY

[Advanced](#) [Cancel](#) [Save](#)

- Enter the name of your topic prefixed with your Workload User Name, ex: **tim_yournewtopic**. For settings you should create it with **(3 partitions, cleanup.policy: delete, availability maximum)** as shown above.

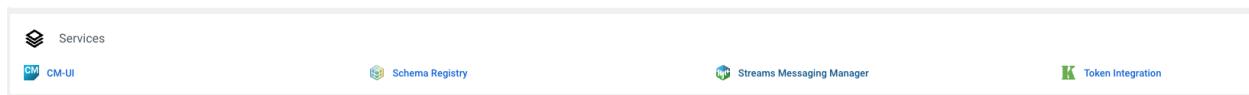
Congratulations! You have built a new topic.

tim_yournewtopic		0B	0B	0	0	N/A	
<small>Replication Factor: (3) InSync Replicas: 9 Of 9 Retention Period: 168 hrs</small>							
Producers (0)		LEADER: 0	PARTITION: 0	DATA IN: 0B	DATA OUT: 0B	LOG SIZE: 0	Consumer Groups (0)
		1546335605	P0	0B/in 0B	0B/out 0B	0	
		1546335647	P1	0B/in 0B	0B/out 0B	0	
		1546335626	P2	0B/in 0B	0B/out 0B	0	



1.2 Create a Schema If You Need One. Not Required For Using Kafka Topics or Tutorials.

1. Navigate to **Schema Registry** from the Kafka Data Hub.



Schema Registry

2. You will see existing schemas.

All Schemas					
		Type	Group	Branch	Serializer & Deserializer
	syslog_avro <small>BACKGROUND</small>	avro	Kafka	1 ↗	0
	syslog_transformed <small>BACKGROUND</small>	avro	Kafka	1 ↗	0
	syslog <small>BACKGROUND</small>	avro	Kafka	1 ↗	0

3. Click the white plus sign in the gray hexagon to create a new schema.



4. You can now add a new schema by entering a unique name starting with your **Workload User Name** (ex: **tim**), followed by a short description and then the schema text as shown. If you need examples, see the github list at the end of this guide.

Add New Schema

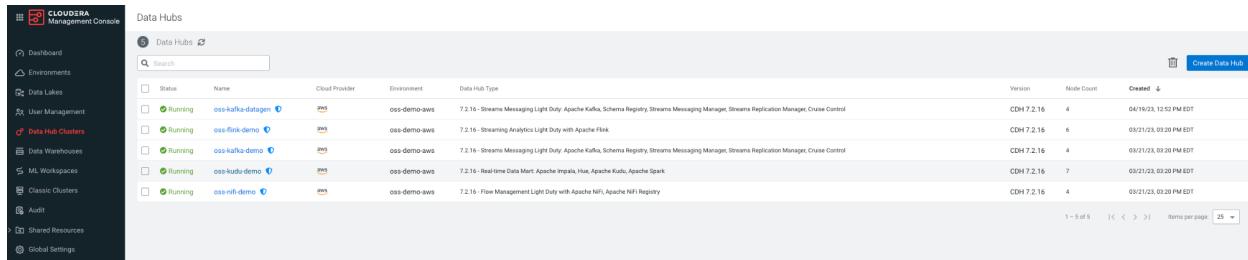
NAME *	SCHEMA TEXT *
tim_yourschema	<pre>65 "name": "pressure_in", 66 "type": ["double"] 67 }, 68 { 69 "name": "dewpoint_string", 70 "type": ["null", "string"] 71 }, 72 { 73 "name": "dewpoint_f", 74 "type": ["double"] 75 }, 76 { 77 "name": "dewpoint_c", 78 "type": ["double"] 79 } 80 } 81 } 82 }</pre>
DESCRIPTION *	<input type="text" value="tim your schema"/>
TYPE *	Avro schema provider
SCHEMA GROUP *	Kafka
COMPATIBILITY	BACKWARD
<input checked="" type="checkbox"/> EVOLVE	
<input type="button" value="CANCEL"/> <input type="button" value="SAVE"/>	

5. Click Save and you have a new schema. If there were errors they will be shown and you can fix them. For more help see, [Schema Registry Documentation](#) and [Schema Registry Public Cloud](#).

Congratulations! You have built a new schema. Start using it in your DataFlow application.

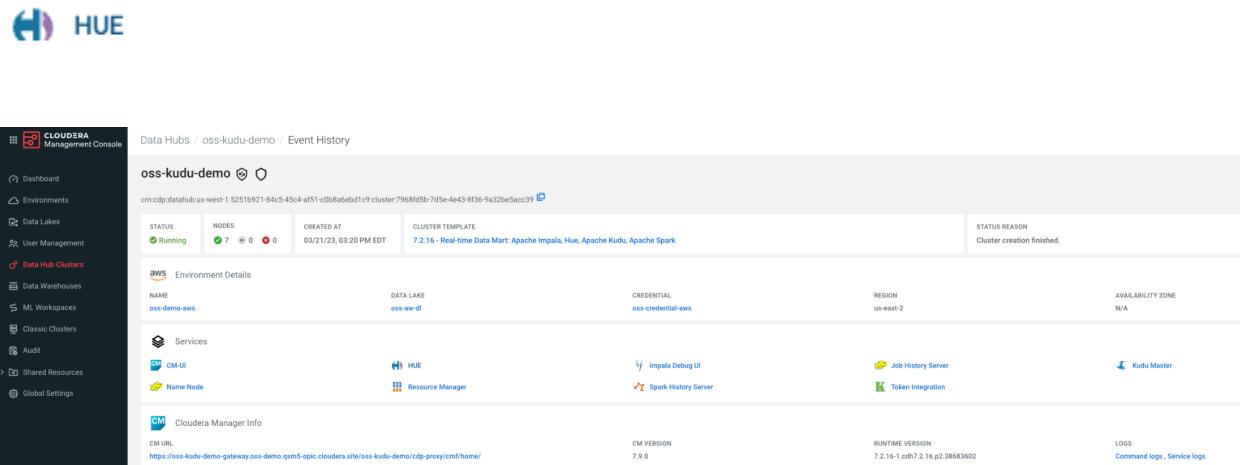
1.3 Create an Apache Iceberg Table

1. Navigate to **oss-kudu-demo** from the Data Hubs list



Status	Name	Cloud Provider	Environment	Data Hub Type	Version	Node Count	Created
Running	oss-kafka-dataset	aws	oss-demo-aws	7.2.16 - Streams Messaging Light Duty, Apache Kafka, Schema Registry, Streams Messaging Manager, Streams Replication Manager, Cruise Control	CDH 7.2.16	4	03/19/23, 12:52 PM EDT
Running	oss-flink-demo	aws	oss-demo-aws	7.2.16 - Streaming Analytics Light Duty with Apache Flink	CDH 7.2.16	6	03/21/23, 03:29 PM EDT
Running	oss-kudu-demo	aws	oss-demo-aws	7.2.16 - Streams Messaging Light Duty, Apache Kafka, Schema Registry, Streams Messaging Manager, Streams Replication Manager, Cruise Control	CDH 7.2.16	4	03/21/23, 03:29 PM EDT
Running	oss-kudu-demo	aws	oss-demo-aws	7.2.16 - Real-time Data Mart: Apache Impala, Hue, Apache Kudu, Apache Spark	CDH 7.2.16	7	03/21/23, 03:29 PM EDT
Running	oss-nifi-demo	aws	oss-demo-aws	7.2.16 - Flow Management Light Duty with Apache Nifi, Apache Nifi Registry	CDH 7.2.16	4	03/21/23, 03:29 PM EDT

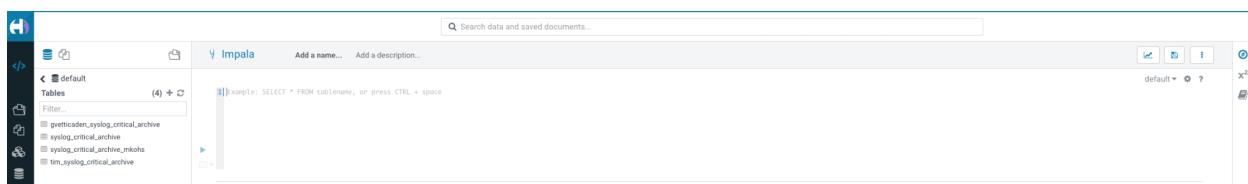
2. Navigate to **Hue** from the Kudu Data Hub.



NAME	DATA LAKE	CREDENTIAL	REGION	AVAILABILITY ZONE
oss-demo-aws	oss-aw-dl	oss-credential-aws	us-east-2	N/A

CM URL	CM VERSION	RUNTIME VERSION	LOGS
https://oss-kudu-demo-gateway.oss-demo.qm5-qpc.cloudera.site/oss-kudu-demo/cdp-proxy/cm/home/	7.9.0	7.2.16-cdh7.2.16-p2.38683602	Command logs, Service logs

3. Inside of **Hue** you can now create your table.

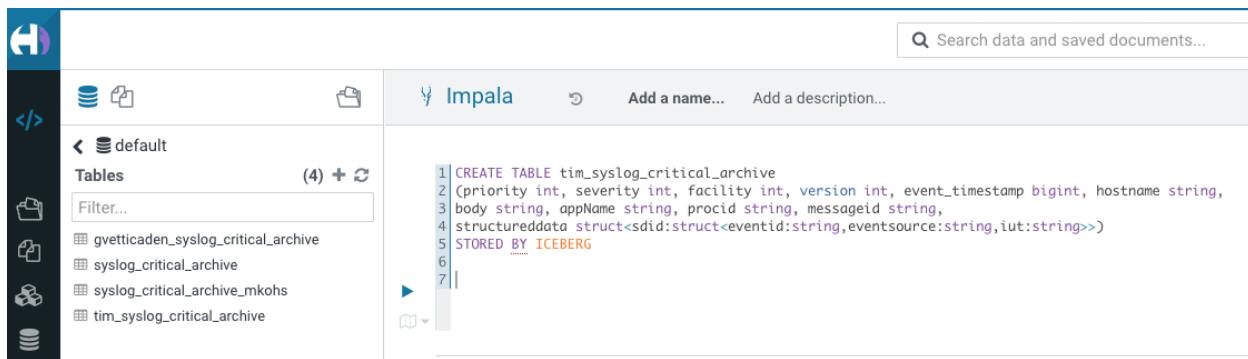


4. Navigate to your database, this was created for you.

Info: The database name pattern is your email address and then all special characters are replaced with **underscore** and then **_db** is appended to that to make the db name and the ranger policy is created to limit access to just the user and those that are in the admin group. For example:

5. Create your Apache Iceberg table, it must be prefixed with your **Work Load User Name (userid)**.

```
CREATE TABLE <>userid>>_syslog_critical_archive
(priority int, severity int, facility int, version int, event_timestamp bigint, hostname string,
body string, appName string, procid string, messageid string,
structureddata struct<sdid:struct<eventid:string,eventsoure:string,iut:string>>)
STORED BY ICEBERG
```



The screenshot shows the Impala UI interface. On the left, there's a sidebar with icons for HDFS, tables, databases, and other data sources. The main area shows a table named 'default' with four rows. Below the table, there's a search bar with placeholder text 'Search data and saved documents...' and buttons for 'Add a name...' and 'Add a description...'. The central part of the screen displays the SQL code for creating an Apache Iceberg table, which is identical to the one shown in the previous text block.

```
1 CREATE TABLE tim_syslog_critical_archive
2 (priority int, severity int, facility int, version int, event_timestamp bigint, hostname string,
3 body string, appName string, procid string, messageid string,
4 structureddata struct<sdid:struct<eventid:string,eventsoure:string,iut:string>>)
5 STORED BY ICEBERG
6
7
```

6. Your table is created in **s3a://oss-uat2/iceberg/**

Impala Add a name... Add a description...

```
1 CREATE TABLE youruserid_syslog_critical_archive
2 (priority int, severity int, facility int, version int, event_timestamp bigint, hostname string,
3 body string, appName string, procid string, messageid string,
4 structureddata struct<sdid:struct<eventid:string,eventsoure:string,iut:string>>)
5 STORED BY ICEBERG
6
7
```

No logs available at this moment.

Query History Saved Queries Results (1)

summary

1 Table has been created.

7. Once you have sent data to your table, you can query it.

The screenshot shows the Impala Query Editor interface. At the top, there are tabs for 'Impala' and 'Add a name...', and buttons for 'Add a description...', 'Run', 'Save', and 'Cancel'. Below the tabs, the query text is displayed:

```
1| select * from tim_syslog_critical_archive
```

Execution details at the bottom of the editor pane show two queries running, both completed at 100%:

Query 66488d2fdbb7aa0e:9b76784200000000 100% Complete (11 out of 11)
 Query 66488d2fdbb7aa0e:9b76784200000000 100% Complete (11 out of 11)

The status bar indicates '0.86s default'.

Below the editor is a 'Results (100+)' section with a table. The table has the following columns:

	priority	severity	facility	version	event_timestamp	hostname	body	appname	procid	messageid
1	40	0	5	1	1681866785339	host3.example.com	application9 has exited cleanly	application9	9290	ID36
2	185	1	23	1	1681866785382	host5.example.com	application3 has completed gracefully	application3	8641	ID43
3	104	0	13	1	1681866785382	host9.example.com	application5 has stopped unexpectedly	application5	3544	ID42
4	153	1	19	1	1681866785382	host3.example.com	application10 has stopped unexpectedly	application10	6211	ID4
5	74	2	9	1	1681866785476	host1.example.com	application9 has exited cleanly	application9	8852	ID28
6	89	1	11	1	1681866785476	host5.example.com	application6 has stopped unexpectedly	application6	1559	ID13
7	178	2	22	1	1681866785437	host3.example.com	application7 has exited cleanly	application7	834	ID1
8	114	2	14	1	1681866785438	host2.example.com	application11 has stopped unexpectedly	application10	2579	ID21
9	129	1	16	1	1681866785381	host4.example.com	application5 has started successfully	application5	8691	ID39
10	113	1	14	1	1681866785475	host5.example.com	application6 has exited cleanly	application6	5560	ID18
11	0	0	0	1	1681866785382	host3.example.com	application4 has started successfully	application4	8664	ID7
12	9	1	1	1	1681866785438	host1.example.com	application8 has exited cleanly	application8	8115	ID2
13	74	2	9	1	1681866785475	host1.example.com	application9 has completed gracefully	application9	4930	ID34
14	178	2	22	1	1681866785339	host6.example.com	application3 has exited cleanly	application3	1760	ID18
15	66	2	8	1	1681866785338	host2.example.com	application5 has started successfully	application5	5519	ID26
16	176	0	22	1	1681866605324	host9.example.com	application5 has started successfully	application5	2050	ID29
17	56	0	7	1	1681866605324	host8.example.com	application4 has exited cleanly	application4	2878	ID34
18	49	1	6	1	1681866605279	host8.example.com	application10 has stopped unexpectedly	application10	1227	ID21
19	137	1	17	1	1681866605325	host8.example.com	application3 has exited cleanly	application3	2588	ID48
20	24	0	3	1	1681866605380	host6.example.com	application7 has completed gracefully	application7	7808	ID43
21	25	1	3	1	1681866605380	host9.example.com	application1 has exited cleanly	application1	6989	ID11
22	64	0	8	1	1681866605402	host2.example.com	application6 has started successfully	application6	3418	ID11
23	106	2	13	1	1681866605402	host3.example.com	application5 has completed gracefully	application5	3687	ID43
24	105	1	13	1	1681866605278	host4.example.com	application3 has started successfully	application3	6063	ID2

Additional Documentation

- [Create a Table](#)
- [Query a Table](#)
- [Apache Iceberg Table Properties](#)

2. Streaming Data Sets Available for Apps

The following Kafka topics are being populated with streaming data for you.

These come from the read-only Kafka cluster.

Navigate to the **Data Hub Clusters**.

Status	Name	Cloud Provider	Environment	Data Hub Type	Version	Node Count	Created
Running	oss-kafka-datagen	aws	oss-demo-aws	7.2.16 - Streams Messaging Light Duty: Apache Kafka, Schema Registry, Streams Messaging Manager, Streams Replication Manager, Cruise Control	CDH 7.2.16	4	04/19/23, 12:52 PM EDT
Running	oss-flink-demo	aws	oss-demo-aws	7.2.16 - Streaming Analytics Light Duty with Apache Flink	CDH 7.2.16	6	03/21/23, 03:20 PM EDT
Running	oss-kafka-demo	aws	oss-demo-aws	7.2.16 - Streams Messaging Light Duty: Apache Kafka, Schema Registry, Streams Messaging Manager, Streams Replication Manager, Cruise Control	CDH 7.2.16	4	03/21/23, 03:20 PM EDT
Running	oss-kudu-demo	aws	oss-demo-aws	7.2.16 - Real-time Data Mart: Apache Impala, Hue, Apache Kudu, Apache Spark	CDH 7.2.16	7	03/21/23, 03:20 PM EDT
Running	oss-nifi-demo	aws	oss-demo-aws	7.2.16 - Flow Management Light Duty with Apache NiFi, Apache NiFi Registry	CDH 7.2.16	4	03/21/23, 03:20 PM EDT

Click on **oss-kafka-datagen**.

Data Hubs / oss-kafka-datagen / Event History

oss-kafka-datagen

crn:cdp:datahub:us-west-1:5251b921-84c5-45c4-af51-c0b8a6ebd1c9:cluster:546bb32-da6b-45e8-b0bf-00ed29b4a561

STATUS	NODES	CREATED AT	CLUSTER TEMPLATE
Running	4 0 0	04/19/23, 12:52 PM EDT	7.2.16 - Streams Messaging Light Duty: Apache Kafka, Schema Registry, Streams Messaging Manager, Streams Replication Manager, Cruise Control

STATUS REASON
Cluster creation finished.

aws Environment Details

NAME	DATA LAKE	CREDENTIAL
oss-demo-aws	oss-aw-dl	oss-credential-aws

Services

CM UI Schema Registry Streams Messaging Manager

Click Schema Registry.

The screenshot shows the Click Schema Registry interface with the title "All Schemas". At the top, there is a search bar labeled "Search by name" and a sorting option "Sort: Last Updated". Below the header, a table lists seven schemas:

Name	Type	Group	Branch	Serializer & Deserializer	Action
orders BOTH	avro	Kafka	1 ↗	0	▼
plant BOTH	avro	Kafka	1 ↗	0	▼
sensor BOTH	avro	Kafka	1 ↗	SERIALIZER & DESERIALIZER	▼
sensor_data BOTH	avro	Kafka	1 ↗	SERIALIZER & DESERIALIZER	▼
weather BOTH	avro	Kafka	1 ↗	0	▼
transactions BOTH	avro	Kafka	1 ↗	SERIALIZER & DESERIALIZER	▼
keycloakMsg BACKWARD	avro	Kafka	1 ↗	0	▼

Click Streams Messaging Manager.

The screenshot shows the Click Streams Messaging Manager interface with the title "Overview". It displays four main sections: Producers, Brokers, Topics, and Consumer Groups. The Producers section shows 8 of 17 active producers, including "customer", "mm2-offsets.secondary.inter", "sensor_data", "transactions", "ip_address", "weather", "plant", "sensor", and "mm2-configs.secondary.inter". The Topics section shows 12 of 40 topics. The Consumer Groups section shows 2 of 4 consumer groups. On the right side, there is a "Consumer Groups (2)" panel. The bottom right corner indicates a cluster named "kafka-d47b" and a log entry for "tim_cdf" at offset 68.

Use these brokers to connect to them:

Brokers

oss-kafka-datagen-corebroker1.oss-demo.qsm5-opic.cloudera.site:9093,oss-kafka-datagen-corebroker0.oss-demo.qsm5-opic.cloudera.site:9093,oss-kafka-datagen-corebroker2.oss-demo.qsm5-opic.cloudera.site:9093

Use this link for Schema Registry

<https://#{Schema2}:7790/api/v1>

Schema Registry Parameter Hostname: Schema2

oss-kafka-datagen-master0.oss-demo.qsm5-opic.cloudera.site

To View Schemas in the Schema Registry click the icon from the datahub

<https://oss-kafka-datagen-gateway.oss-demo.qsm5-opic.cloudera.site/oss-kafka-datagen/cdp-proxy/schema-registry/ui/#/>

Schemas

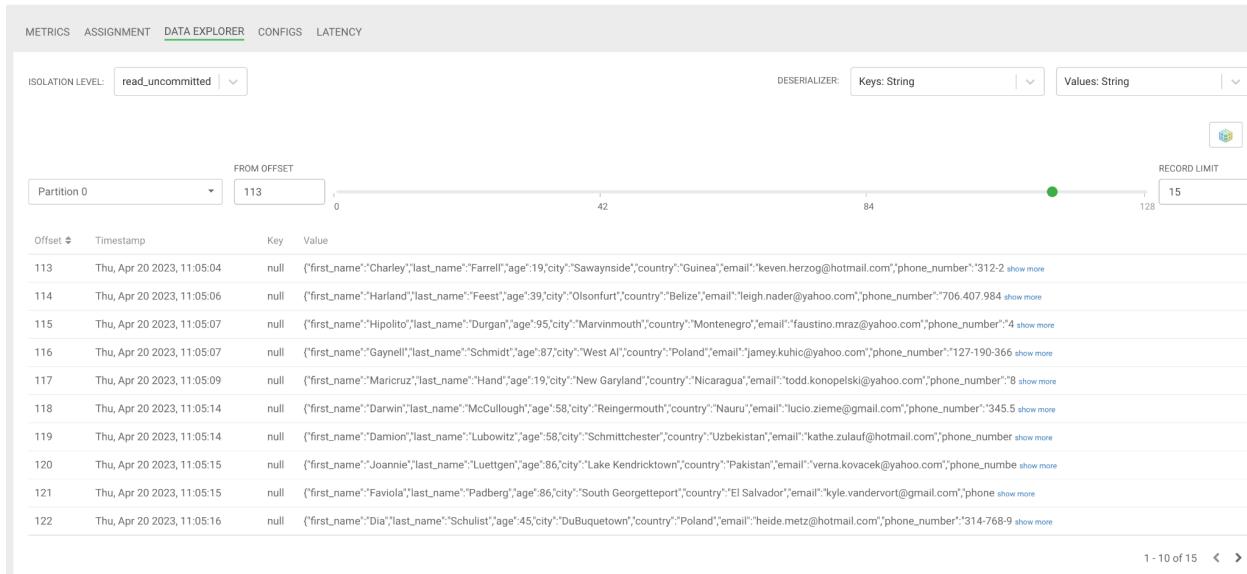
<https://github.com/tspannhw/FLaNK-DataFlows/tree/main/schemas>

Group ID: yourid_cdf

Customers ([customer](#))

Example Row

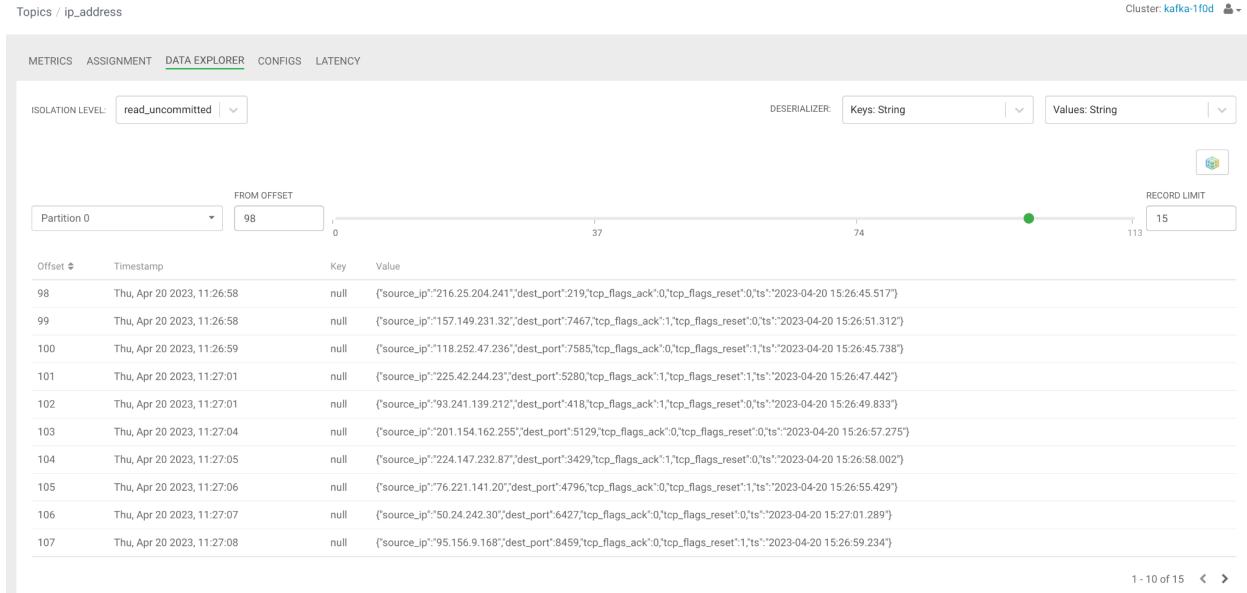
{"first_name": "Charley", "last_name": "Farrell", "age": 19, "city": "Sawaynside", "country": "Guinea", "email": "keven.hertzog@hotmail.com", "phone_number": "312-269-6619"}



IP Tables ([ip_address](#))

Example Row

```
{"source_ip": "216.25.204.241", "dest_port": 219, "tcp_flags_ack": 0, "tcp_flags_reset": 0, "ts": "2023-04-20 15:26:45.517"}
```



Orders ([orders](#))

Example Row

{"order_id":84170282,"city":"Wintheiserton","street_address":"80206 Caroyln Lakes","amount":29,"order_time":"2023-04-20 13:25:06.097","order_status":"DELIVERED"}

Topics: transactions Cluster: kafka-10d

METRICS ASSIGNMENT DATA EXPLORER CONFIGS LATENCY ISOLATION LEVEL: read_uncommitted DESERIALIZER: Keys: String Values: String RECORD LIMIT: 15

Partition 0 FROM OFFSET: 0 0 4 8 14 15

Offset #	Timestamp	Key	Value
0	Thu, Apr 20 2023, 12:15:46	null	{"sender_id":40816,"receiver_id":96056,"amount":557,"execution_date":"2023-04-20 16:15:30.744","currency":"UAH"}
1	Thu, Apr 20 2023, 12:15:48	null	{"sender_id":14917,"receiver_id":83314,"amount":1353,"execution_date":"2023-04-20 16:15:39.235","currency":"MOP"}
2	Thu, Apr 20 2023, 12:15:48	null	{"sender_id":20997,"receiver_id":81505,"amount":3223,"execution_date":"2023-04-20 16:15:34.373","currency":"UAH"}
3	Thu, Apr 20 2023, 12:15:49	null	{"sender_id":56753,"receiver_id":19382,"amount":3099,"execution_date":"2023-04-20 16:15:38.982","currency":"COP"}
4	Thu, Apr 20 2023, 12:15:50	null	{"sender_id":8110,"receiver_id":87971,"amount":2163,"execution_date":"2023-04-20 16:15:47.105","currency":"UGX"}
5	Thu, Apr 20 2023, 12:15:51	null	{"sender_id":84617,"receiver_id":23815,"amount":1983,"execution_date":"2023-04-20 16:15:44.879","currency":"MAD"}
6	Thu, Apr 20 2023, 12:15:52	null	{"sender_id":71642,"receiver_id":38571,"amount":661,"execution_date":"2023-04-20 16:15:49.484","currency":"TOP"}
7	Thu, Apr 20 2023, 12:15:54	null	{"sender_id":47771,"receiver_id":71971,"amount":958,"execution_date":"2023-04-20 16:15:44.894","currency":"MDN"}
8	Thu, Apr 20 2023, 12:15:54	null	{"sender_id":15283,"receiver_id":1492,"amount":4120,"execution_date":"2023-04-20 16:15:48.694","currency":"LKR"}
9	Thu, Apr 20 2023, 12:15:55	null	{"sender_id":59570,"receiver_id":68750,"amount":1050,"execution_date":"2023-04-20 16:15:52.357","currency":"MVR"}

1 - 10 of 14

Plants (plant)

Example Row

{"plant_id":829,"city":"Lake Gerald","lat":-39.568679,"lon":-151.64497,"country":"Eritrea"}

Topics: plant Cluster: kafka-10d

METRICS ASSIGNMENT DATA EXPLORER CONFIGS LATENCY ISOLATION LEVEL: readUncommitted DESERIALIZER: Keys: String Values: String RECORD LIMIT: 15

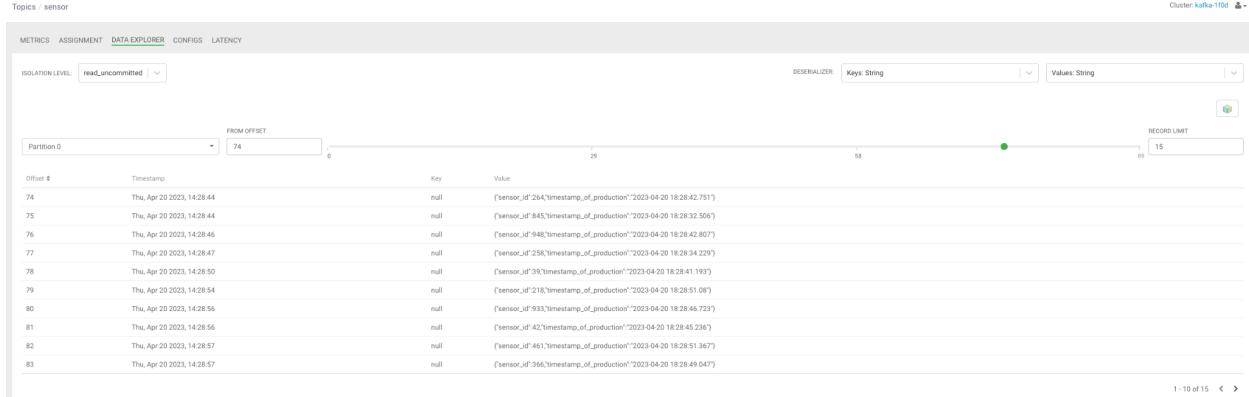
Partition 0 FROM OFFSET: 25 8 13 18 25 26 27 28 29 30 31 32 33 34

Offset #	Timestamp	Key	Value
25	Thu, Apr 20 2023, 14:17:36	null	{"plant_id":829,"city":"Lake Gerald","lat":-39.568679,"lon":-151.64497,"country":"Eritrea"}
26	Thu, Apr 20 2023, 14:17:38	null	{"plant_id":966,"city":"Kelleystead","lat":-39.742722,"lon":-25.432469,"country":"Democratic People's Republic of Korea"}
27	Thu, Apr 20 2023, 14:17:38	null	{"plant_id":244,"city":"Ullensborgh","lat":-28.280251,"lon":-117.6778,"country":"Italy"}
28	Thu, Apr 20 2023, 14:17:39	null	{"plant_id":403,"city":"Greenbury","lat":-61.663793,"lon":-120.44550,"country":"South Africa"}
29	Thu, Apr 20 2023, 14:17:39	null	{"plant_id":654,"city":"Frankfurt","lat":-43.054197,"lon":-186.44948,"country":"Belize"}
30	Thu, Apr 20 2023, 14:17:39	null	{"plant_id":561,"city":"East Tyrone","lat":-53.03913,"lon":-2.02273,"country":"Portugal"}
31	Thu, Apr 20 2023, 14:17:39	null	{"plant_id":421,"city":"Princessmouf","lat":-55.313097,"lon":-16.417304,"country":"Luxembourg"}
32	Thu, Apr 20 2023, 14:17:39	null	{"plant_id":873,"city":"Hilland","lat":-67.799363,"lon":-153.93980,"country":"Timor Leste"}
33	Thu, Apr 20 2023, 14:17:40	null	{"plant_id":270,"city":"McCluretown","lat":-50.730992,"lon":-119.46428,"country":"Sweden"}
34	Thu, Apr 20 2023, 14:17:40	null	{"plant_id":90,"city":"Claudinetor","lat":-48.426563,"lon":-48.981618,"country":"Marshall Islands"}

Sensors (sensor)

Example Row

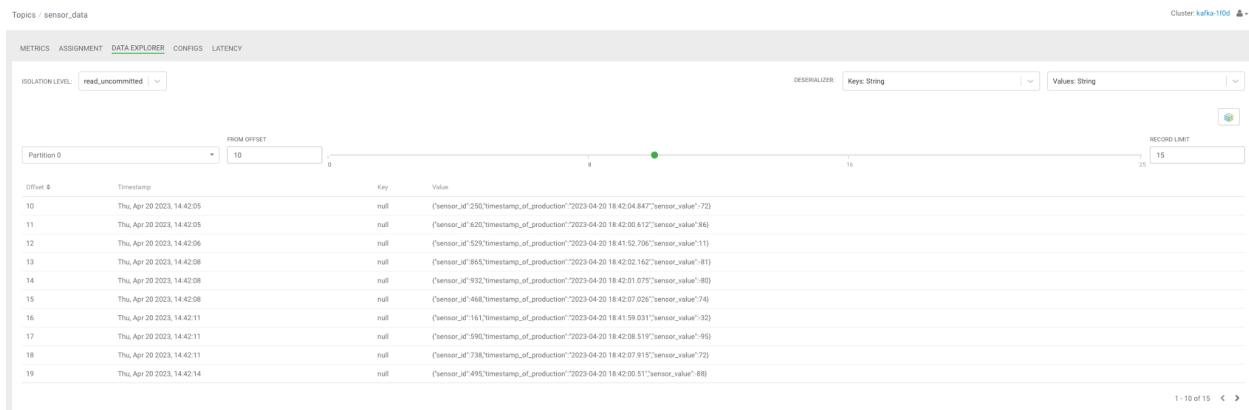
{"sensor_id":264,"timestamp_of_production":"2023-04-20 18:28:42.751"}



Sensor Data ([sensor_data](#))

Example Row

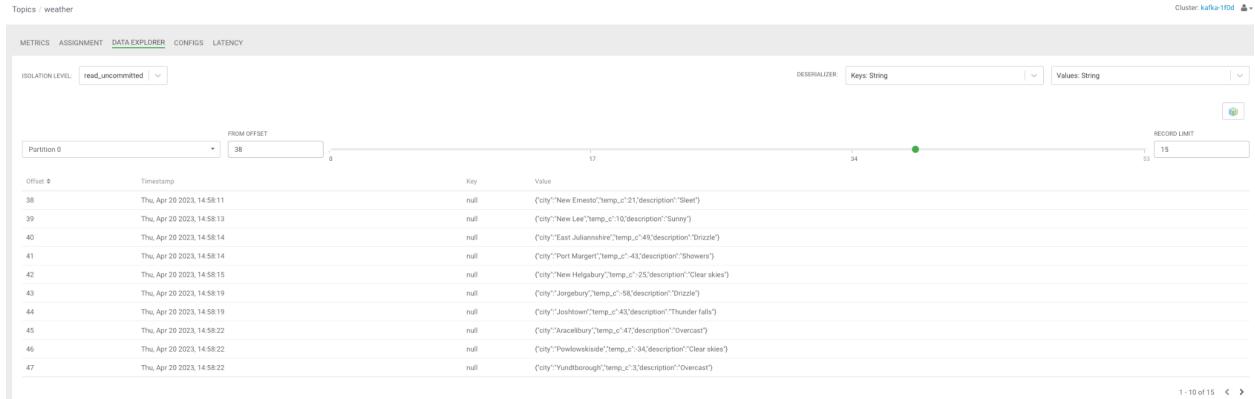
```
{"sensor_id":250,"timestamp_of_production":"2023-04-20 18:42:04.847","sensor_value":-72}
```



Weather ([weather](#))

Example Row

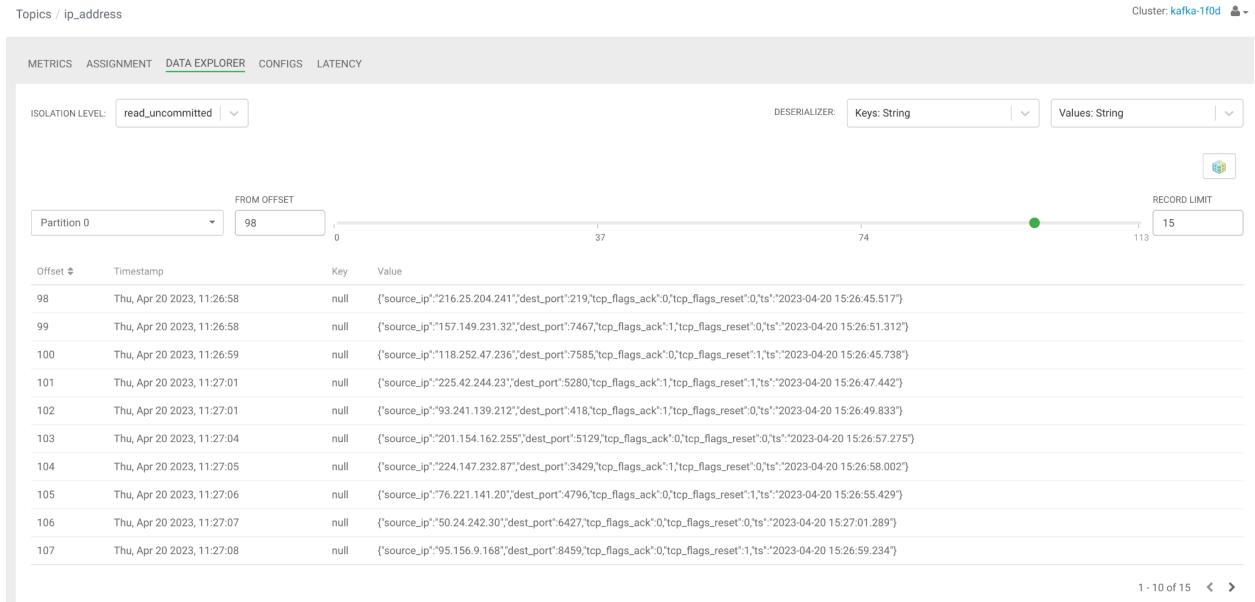
```
{"city":"New Ernesto","temp_c":21,"description":"Sleet"}
```



Transactions ([transactions](#))

Example Row

```
{"sender_id":40816,"receiver_id":96057,"amount":557,"execution_date":"2023-04-20 16:15:30.744","currency":"UYU"}
```



These are realistic generated data sources that you can use, they are available from read-only Kafka topics. These can be consumed by any developers in the sandbox.

Make sure you name your Kafka Consumer your Workload Username _ Some Name.

Ex: tim_customerdata_reader

3. Bring Your Own Data (Public Only)

- Data is **visible** and **downloadable** to all, make sure it is safe, free, **open**, public data.
- Public REST Feeds are good
 - Wikipedia
<https://docs.cloudera.com/dataflow/cloud/flow-designer-beginners-guide-readyflow/topics/cdf-flow-designer-getting-started-readyflow.html>
 - https://gbfs.citibikenyc.com/gbfs/en/station_status.json
 - https://travel.state.gov/_res/rss/TAsTWs.xml
 - https://www.njtransit.com/rss/BusAdvisories_feed.xml
 - https://www.njtransit.com/rss/RailAdvisories_feed.xml
 - https://www.njtransit.com/rss/LightRailAdvisories_feed.xml
 - https://www.njtransit.com/rss/CustomerNotices_feed.xml
 - https://w1.weather.gov/xml/current_obs/all_xml.zip
 - <https://dailymed.nlm.nih.gov/dailymed/services/v2/spls.json?page=1&pagesize=100>
 - <https://dailymed.nlm.nih.gov/dailymed/services/v2/drugnames.json?pagesize=100>
 - <https://dailymed.nlm.nih.gov/dailymed/rss.cfm>
- Generic data files
 - <https://aws.amazon.com/data-exchange>
- Simulators
 - Use external data simulators via REST
 - Use GeneralFlowFile see:
<https://www.datainmotion.dev/2019/04/integration-testing-for-apache-nifi.html>
- Schemas, Data Sources and Examples
 - <https://github.com/tspannhw/FLaNK-AllTheStreams/>
 - <https://github.com/tspannhw/FLaNK-DataFlows>
 - <https://github.com/tspannhw/FLaNK-TravelAdvisory/>
 - <https://github.com/tspannhw/FLiP-Current22-LetsMonitorAllTheThings>
 - <https://github.com/tspannhw/create-nifi-kafka-flink-apps>
 - <https://www.datainmotion.dev/2021/01/flank-real-time-transit-information-for.html>

