



LAB

01 – Cloudera Data Flow

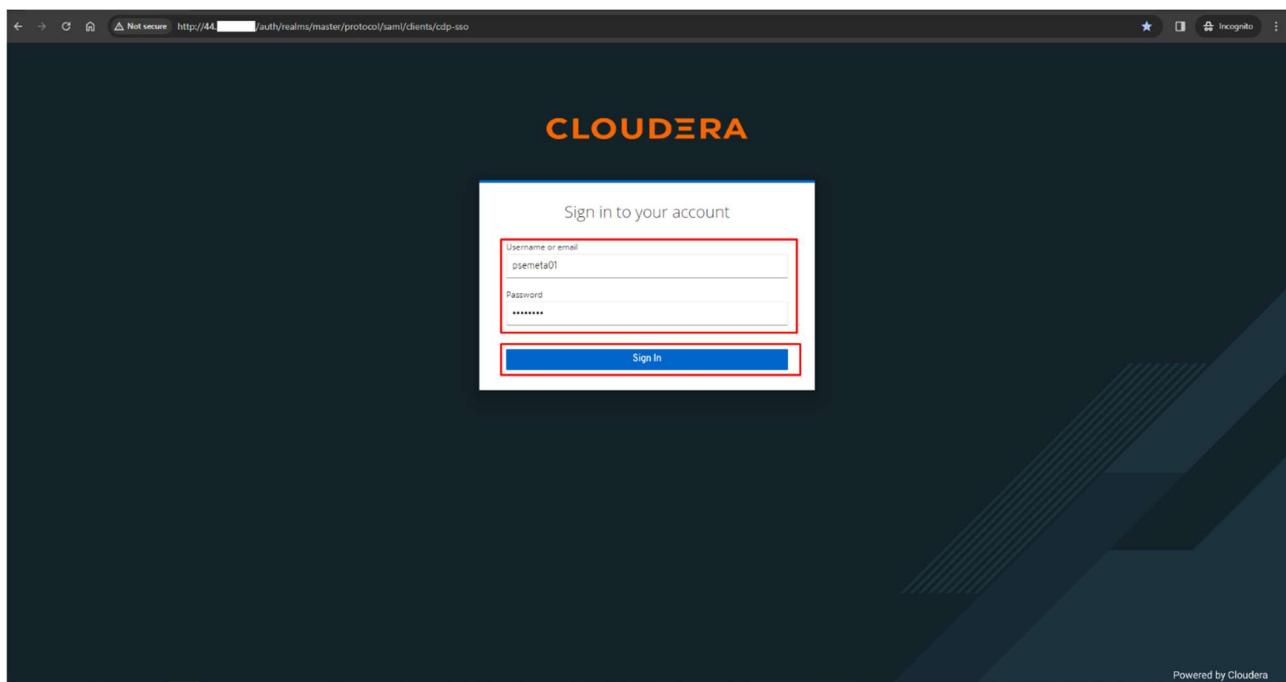
Data Lifecycle CDP Public Cloud

Data Flow Lab

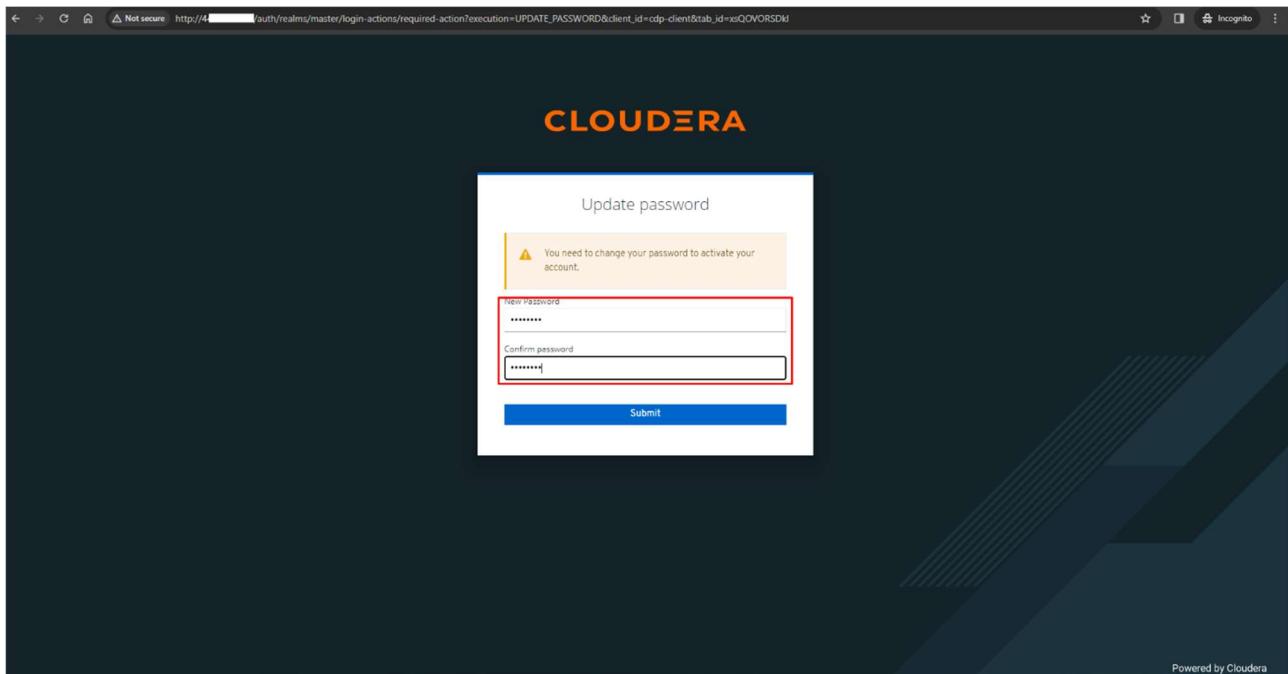
Goals:

- Consume data from a Kafka topic.
- Convert the data to Parquet format.
- Store the data in a table in the Lakehouse.

1. Login to the environment using the URL provided by the instructor.
The below page is a Keycloak instance which is used as an IdP here.

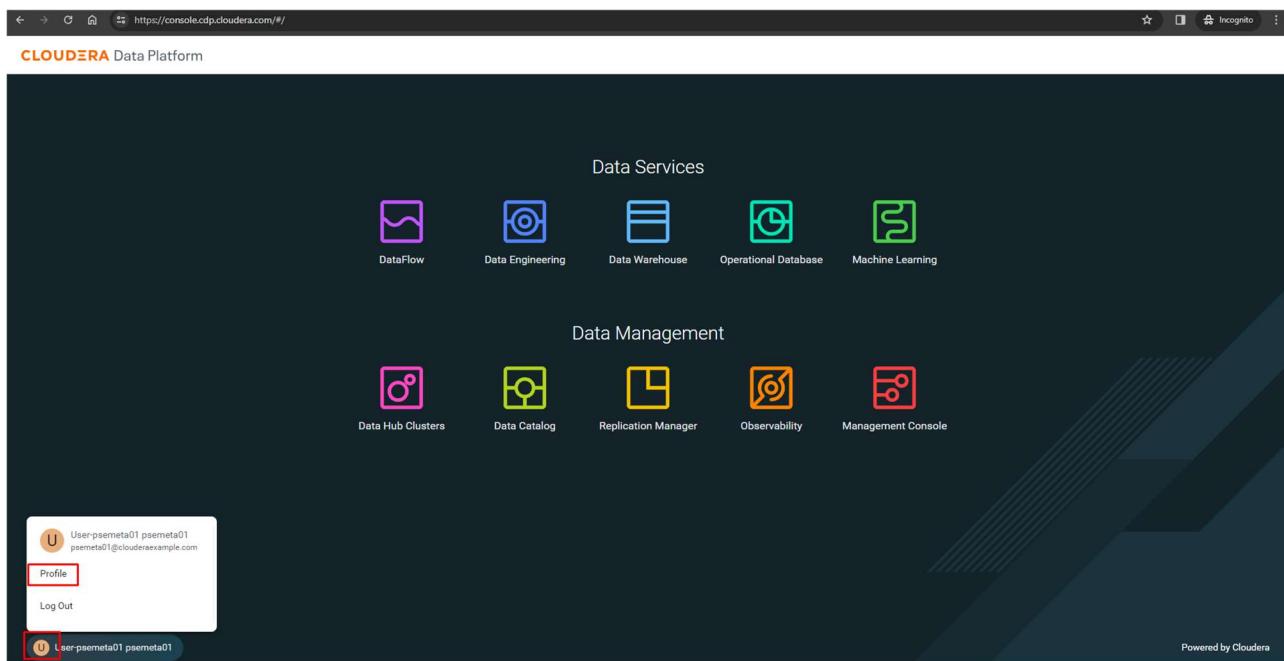


It might ask to change the password. Keep the password same as earlier which is – **changeme**



This is the CDP Console homepage.

Now you will set a new workload password. Click on your login name at the lower bottom corner and then click on **Profile**.



Click on '**Set Workload Password**'.

The screenshot shows the Cloudera Management Console interface. On the left, there's a sidebar with various navigation options like Dashboard, Environments, Data Lakes, User Management, Data Hub Clusters, Data Warehouses, ML Workspaces, Classic Clusters, Audit, Consumption, Shared Resources, and Global Settings. The User Management section is currently selected. The main content area displays a user profile for 'User-psemeta01 psemeta01'. The profile includes fields for Name (User-psemeta01 psemeta01), Email (psemeta01@clouderaexample.com), Workload User Name (psemeta01), CRN (cm:altus:iam:us-west-1:d1a4553c-a799-432d-8e54-372cc2ab95f2:user:651...), Tenant ID (d1a4553c-a799-432d-8e54-372cc2ab95f2), Identity Provider (psemeta), Last Interactive Login (03/22/2024 10:24 AM +04), Profile Management (View profile), and Workload Password (Set Workload Password). A red box highlights the 'Set Workload Password' button. Below this, there are tabs for Access Keys, Roles, Resources, Groups, and SSH Keys. A message indicates that access rights are missing: 'You don't have the access rights. Please contact your admin.'

Set the password as – **Changeme123! (Note C caps)** in both the fields and click on **Set Workload Password**.

The screenshot shows the 'Workload Password' configuration page. It features two input fields: 'Password' and 'Confirm Password', both containing '*****'. A red box highlights the 'Set Workload Password' button. A note below the fields states: 'If you use keytabs, you need to regenerate them after changing your workload password. You can do this from your user profile > Actions > Get Keytab.' A red box also highlights the 'Set Workload Password' button.

Password is set successfully.

The screenshot shows the Cloudera Management Console interface. On the left, there's a sidebar with various navigation options like Dashboard, Environments, Data Lakes, User Management (which is highlighted in red), Data Hub Clusters, Data Warehouses, ML Workspaces, Classic Clusters, Audit, Consumption, Shared Resources, and Global Settings. The main content area shows a user profile for 'User-psemeta01 psemeta01'. The profile includes fields for Name, Email, Workload User Name, CRN, Tenant ID, Identity Provider, Last Interactive Login, Profile Management, and Workload Password. A success message box at the top right says 'Success: Workload password is updated and it is being synced to environments.' Below the profile, there are tabs for Access Keys, Roles, Resources, Groups, and SSH Keys. At the bottom, there's a note: 'You don't have the access rights. Please contact your admin.'

Now go back to the main page by clicking on the **Cloudera Management Console** on top left corner.

Click on '**DataFlow**' once you reach the landing page.

The screenshot shows the Cloudera Data Platform landing page. It features a dark background with several service icons and names. In the 'Data Services' section, there are icons for DataFlow (highlighted with a red box), Data Engineering, Data Warehouse, Operational Database, and Machine Learning. In the 'Data Management' section, there are icons for Data Hub Clusters, Data Catalog, Replication Manager, Observability, and Management Console. At the bottom left, there's a user indicator 'User-psemeta01 psemeta01'. At the bottom right, it says 'Powered by Cloudera'.

Once in DataFlow, click on the option **Catalog** from the left menu. The data ingestion application templates are listed here. For this workshop, we have created and published a template that allows you to read Kafka topic data and ingest/store it in the Lakehouse provided by CDP Public Cloud.

On the search bar type **lab_**.

Click on the Flow called **lab_kafka_to_lakehouse** to start deploying it.

https://console.us-west-1.cdp.cloudera.com/dfx/ui/#/flows?searchTerm=lab_

REFRESHED: 6 seconds ago

Import Flow Definition

Items per page: 10 | 1 - 2 of 2 | < >

Name	Type	Versions	Last Updated
lab_kafka_to_lakehouse	Custom Flow Definition	1	8 hours ago
lab_s3_to_kafka	Custom Flow Definition	1	8 hours ago

CLOUDERA DataFlow

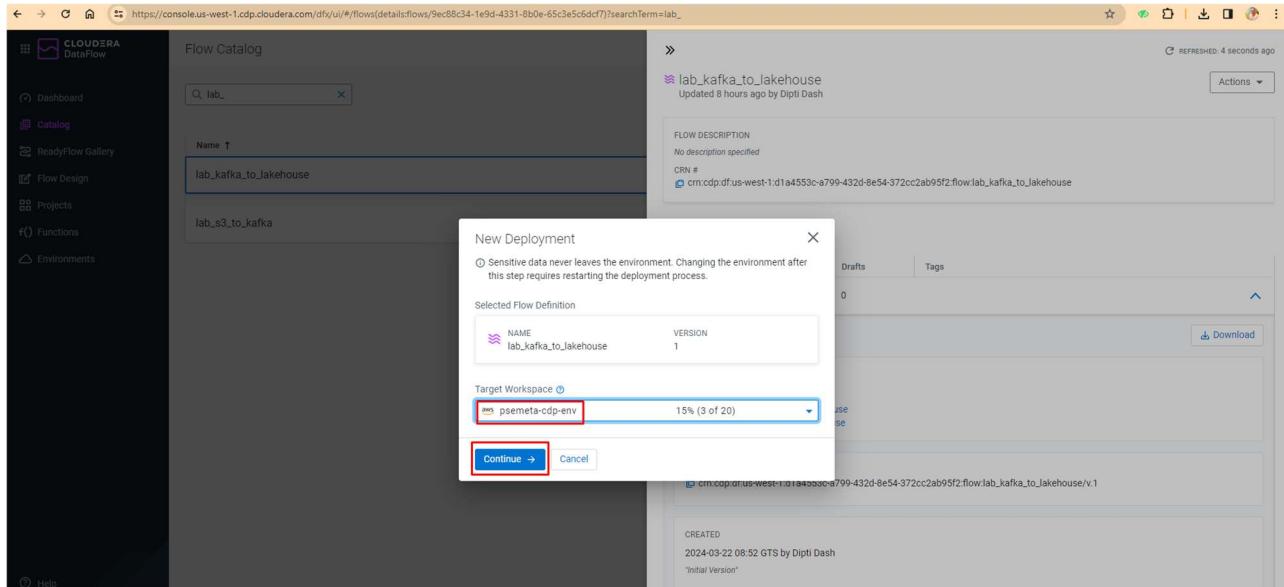
Dashboard Catalog ReadyFlow Gallery Flow Design Projects Functions Environments Help User:psemeta01 psemeta01

2. When clicked, the following panel appears with the Flow information. It shows the available versions, creation date, creator user, and a button **Deploy** to start the deployment. Click **Deploy**.

The screenshot shows the Cloudera DataFlow Catalog interface. On the left, there's a sidebar with options like Dashboard, Catalog, ReadyFlow Gallery, Flow Design, Projects, Functions, and Environments. The main area is titled 'Flow Catalog' and has a search bar with 'lab_'. Below it, a list shows 'Name ↑' with 'lab_kafka_to_lakehouse' selected. To the right, a detailed view of the flow 'lab_kafka_to_lakehouse' is displayed. This view includes:

- Flow Description:** No description specified.
- CRN #:** crn:cdp:df:us-west-1:d1a4553c-a799-432d-8e54-372cc2ab95f2:flow:lab_kafka_to_lakehouse
- Deployment Status:** Only show deployed versions. There are three tabs: Version (selected), Deployments, and Drafts. Under Version, it shows Version 1, Version 2, and Version 0. A blue 'Deploy →' button is highlighted with a red box.
- Deployments (2):** aws_psemeta-cdp-env
 - psemeta01_kakatolakehouse
 - psemeta31_lakatolakehouse
- CRN #:** crn:cdp:df:us-west-1:d1a4553c-a799-432d-8e54-372cc2ab95f2:flow:lab_kafka_to_lakehouse/v.1
- CREATED:** 2024-03-22 08:52 GTS by Dipti Dash
"Initial Version"
- Custom Tag:** Add a custom tag (input field) and a color picker.

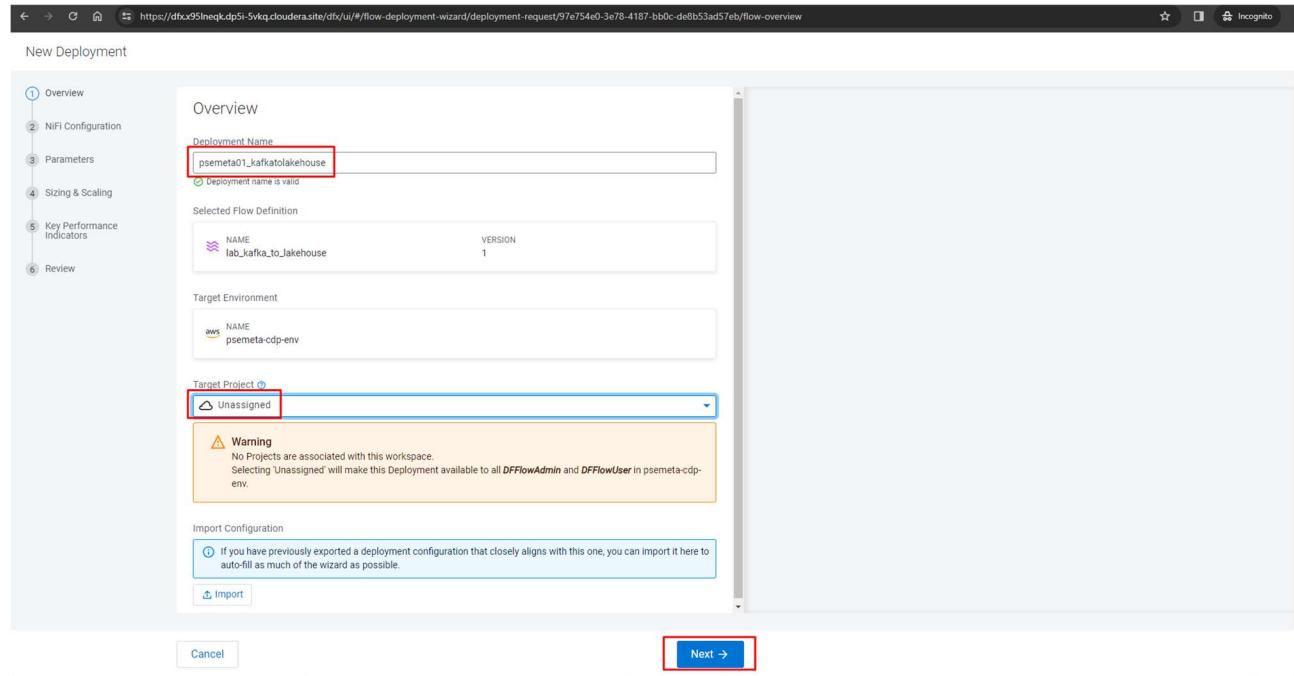
3. The following popup window allows you to select the DataFlow cluster in which you want to deploy the Flow. In this case, the cluster to be selected is **psemeta-cdp-env**. The workshop instructor will tell you which environment to select. Once selected, click **Continue**.



4. From this point, you will need to enter the Flow configuration. Start by assigning a Deployment Name, Target Project, and click Next.

Deployment Name: <assigned_user>_kafkatolakehouse (Ex: psemeta01_kafkatolakehouse)
Target project: Unassigned

Click on **Next**.



The screenshot shows the 'Overview' step of the flow deployment wizard. On the left, a vertical navigation bar lists steps 1 through 6: Overview, NIFI Configuration, Parameters, Sizing & Scaling, Key Performance Indicators, and Review. Step 1 is currently selected. The main area is titled 'Overview' and contains the following fields:

- Deployment Name:** psemeta01_kafkatolakehouse (highlighted with a red box)
- Selected Flow Definition:** lab_kafka_to_lakehouse (version 1)
- Target Environment:** aws (NAME: psemeta-cdp-env)
- Target Project:** Unassigned (highlighted with a red box)

A warning message below the target project dropdown states: "Warning: No Projects are associated with this workspace. Selecting 'Unassigned' will make this Deployment available to all DFFFlowAdmin and DFFFlowUser in psemeta-cdp-env."

At the bottom, there are 'Cancel' and 'Next →' buttons. The 'Next →' button is highlighted with a blue box.

5. Make sure the option **Automatically start flow upon successful deployment** is checked and click **Next**.

New Deployment

NiFi Configuration

NiFi Runtime Version

nifi 1.x

Latest Version (1.25.0.2.3.13.1-1)

[Change Version](#)

1.x is the stable version of NiFi, compatible with the existing flows

nifi 2.x

Latest Version (2.0.0.4.3.0.0-52)

[Tech Preview](#)

[Change Version](#)

2.x is the brand new NiFi version, having extra features like Python processors

This version is for [Tech Preview](#)

Autostart Behavior

Automatically start flow upon successful deployment

Inbound Connections

Allow NiFi to receive data [?](#)

Custom NAR Configuration

This flow deployment uses custom NARs [?](#)

Custom Python Configuration

This flow deployment uses custom python processors [?](#)

Overview

DEPLOYMENT NAME: psemeta99_kafkatalakehouse

FLOW DEFINITION: lab_kafka_to_lakehouse v.1

ENVIRONMENT DEPLOYING TO: psemeta-cdp-env

PROJECT ASSIGNING TO: Unassigned

← Previous **Next →** Cancel

6. In this part of Parameters, you must enter the following values:

CDP Workload User Password: Enter the Workload Password that you had set at the beginning of this workshop. It was something like – **Changeme123!** Or something that you had set on your own.

CDP Workload User: enter the assigned user number, **psemeta01**, for example.

NOTE: for the purposes of the workshop, your user (e.g. **psemeta01**) is also the name of the **database** where you will store the data (which has already been created for you), and the name of the **Kafka Consumer Group ID** (keep it has your user **psemeta01**) for reading messages.

Database: **psemeta01**

Kafka Consumer Group ID: **psemeta01**

Kafka Topic: **telco_data**

New Deployment

- Overview
- NiFi Configuration**
- Parameters
- Sizing & Scaling
- Key Performance Indicators
- Review

Parameters
Data entered here never leaves the environment in your cloud account. Provide parameter values directly in the text input or upload a file for parameters that expect a file.

The selected flow references an external Default NiFi SSL Context Service. Hence, DataFlow will automatically create a matching SSL Context Service with a keystore and truststore generated from the target environment's FreeIPA certificate.

SHOW: Sensitive No value

parameters (7)	
CDP Workload User Password	12/100K
Changeme123	38
CDP Workload Username	9/100K
psemeta01	
CDPEnvironment	
core-site.xml	0
ssl-client.xml	0
hive-site.xml	0
Select File	Drop file or browse

Cancel
← Previous
Next →

Kafka Brokers:

This value should be provided by the instructor.

mtn-streams-corebroker1.psemeta.dp5i-5vkq.cloudera.site:9093,mtn-streams-corebroker0.psemeta.dp5i-5vkq.cloudera.site:9093,mtn-streams-corebroker2.psemeta.dp5i-5vkq.cloudera.site:9093

New Deployment

- Overview
- NiFi Configuration**
- Parameters
- Sizing & Scaling
- Key Performance Indicators
- Review

Parameters
DataFlow automatically adds all required configuration files to interact with Data Lake services. Unnecessary files that are added won't impact the deployment process.

Database
psemeta01

Kafka Brokers
mtn-streams-corebroker1.psemeta.dp5i-5vkq.cloudera.site:9093,mtn-streams-corebroker0.psemeta.dp5i-5vkq.cloudera.site:9093,mtn-streams-corebroker2.psemeta.dp5i-5vkq.cloudera.site:9093

Kafka Consumer Group Id
psemeta01

Kafka Topic
telco_data

Cancel
← Previous
Next →

Review that the parameters were entered correctly. Then click **Next**.

7. There is no need to configure auto-scaling parameters. Click **Next**.

8. We are also not going to configure KPIs now. Click **Next** to continue the configuration.

9. Review all the information entered for your Flow, then click on **Deploy** to start the deployment process.

10. The blue box indicates that the Flow deployment process has been started. By clicking on the button **Load More** you will be able to see the different stages of the deployment. After about 60 to 90 seconds approximately, the last event should be **Deployment Successful**.

https://console.us-west-1.cdp.cloudera.com/dfv/ui#/deployments/details/environments/f07ba706-23a4-47f5-bb27-b480400a9191/deployments/b647a221-e63e-461c-8883-72e9eb7ce14c/alerts

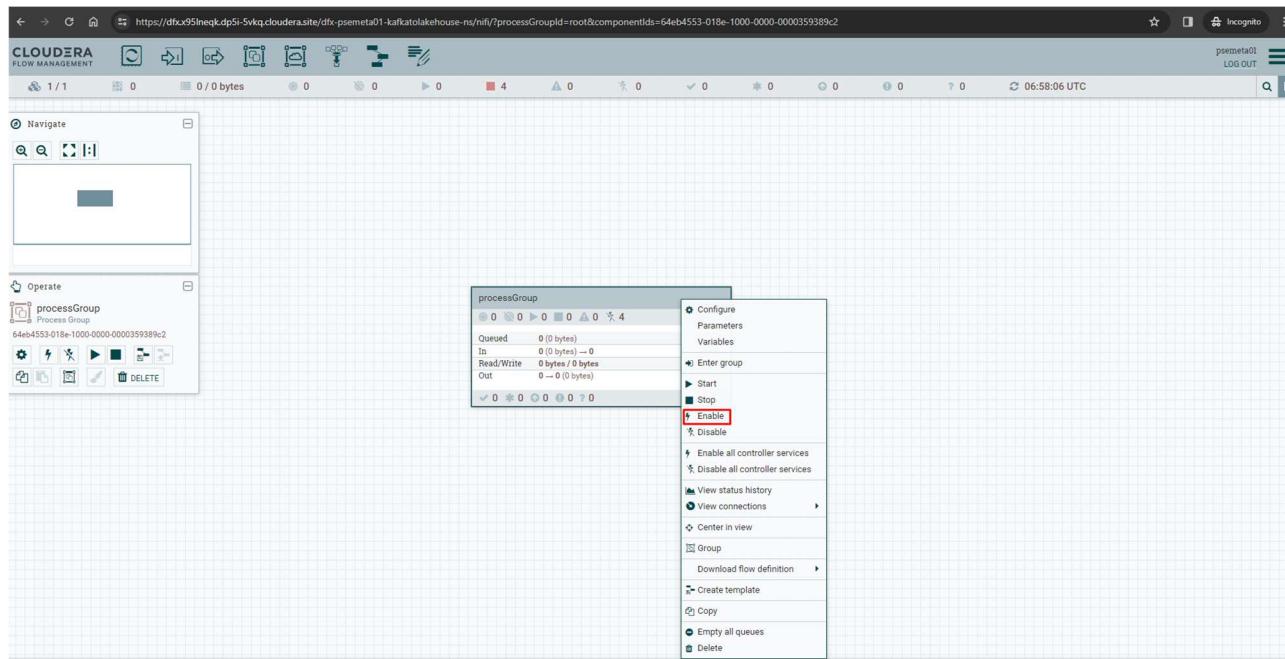
The screenshot shows the Cloudera DataFlow dashboard with the following details:

- Left Sidebar:** Includes links for Dashboard, Catalog, ReadyFlow Gallery, Flow Design, Projects, Functions, and Environments.
- Top Header:** Shows the URL and includes Incognito mode and other browser controls.
- Dashboard Overview:** Displays three flows:
 - diptidash_kafka_to_lakehouse:** Status: Flow Stopped.
 - diptidash_s3_to_kafka_flow:** Status: Flow Stopped.
 - psemeta01_kafkatalakehouse:** Status: Deploying.
- Right Panel - Deployment Details:**
 - Alerts:** Shows three alerts:
 - Deployment Successful:** Successfully deployed [psemeta01_kafkatalakehouse].
 - NiFi Flow Started:** Started flow for deployment [psemeta01_kafkatalakehouse].
 - Starting NiFi Flow:** Starting flow for deployment [psemeta01_kafkatalakehouse].
 - Event History:** Shows two events:
 - Provisioning NiFi Cluster (2024-03-22 10:44 GTS)
 - Deployment Initiated (2024-03-22 10:44 GTS)
- Bottom Buttons:** Includes "Load More" and other navigation controls.

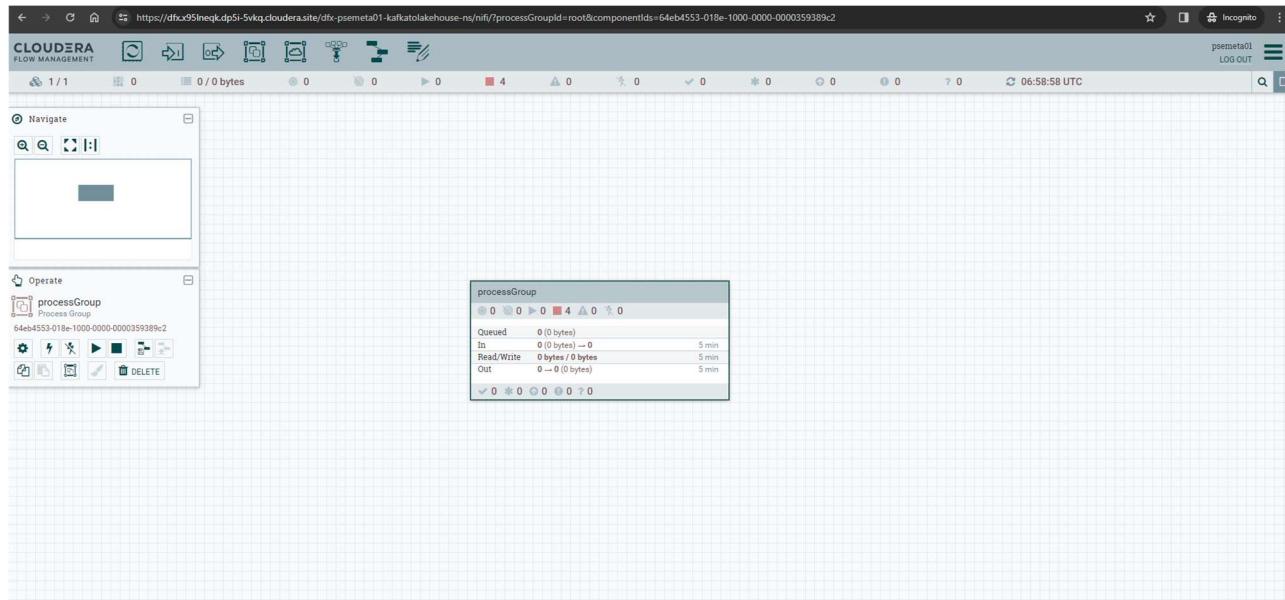
11. Once the deployment is finished, click on **Actions – View in NiFi** to see the details of the recently deployed Flow.

The screenshot shows the Cloudera DataFlow dashboard. On the left sidebar, there are links for Dashboard, Catalog, ReadyFlow Gallery, Flow Design, Projects, Functions, and Environments. The main area displays a list of flows under the 'Status' tab. One flow, 'dptidash_kafka_to_lakehouse', is listed as 'Stopped'. Another flow, 'dptidash_s3_to_kafka_flow', is also listed as 'Stopped'. A third flow, 'psemeta01_kafkatolakehouse', is shown as 'Deploying'. The right side of the screen shows deployment details for 'psemeta01_kafkatolakehouse'. It includes a 'KPIs' section, a 'System Metrics' section, and an 'Alerts' section (which is currently selected). The 'Alerts' section shows 'No alerts to display'. Below this is an 'Event History' section with a table of events. The table has columns for 'Event Type', 'Timestamp', and a dropdown arrow. The events listed are: Deployment Successful (2024-03-22 10:48 GTS), NiFi Flow Started (2024-03-22 10:48 GTS), KPI Alert Rules Activated (2024-03-22 10:48 GTS), Activating KPI Alert Rules (2024-03-22 10:48 GTS), Starting NiFi Flow (2024-03-22 10:48 GTS), Default Alert Rules Activated (2024-03-22 10:48 GTS), Activating Default Alert Rules (2024-03-22 10:48 GTS), NiFi Flow Imported (2024-03-22 10:48 GTS), Importing NiFi Flow (2024-03-22 10:48 GTS), and Preparing for NiFi Flow Import (2024-03-22 10:48 GTS). A red box highlights the 'View in NiFi' button in the 'Actions' dropdown menu. The URL in the browser bar is <https://console.us-west-1.cdp.cloudera.com/dflx/u/#/deployments/detail/environments/f07ba706-23a4-47f5-bb27-b480400a9191/deployments/b647a21-e63e-461c-8883-72e9eb7ce14c/alerts>.

12. The process group needs to be **enabled** first. Hence, right click the processGroup and click on **Enable**.



Double click the processor **processGroup**.

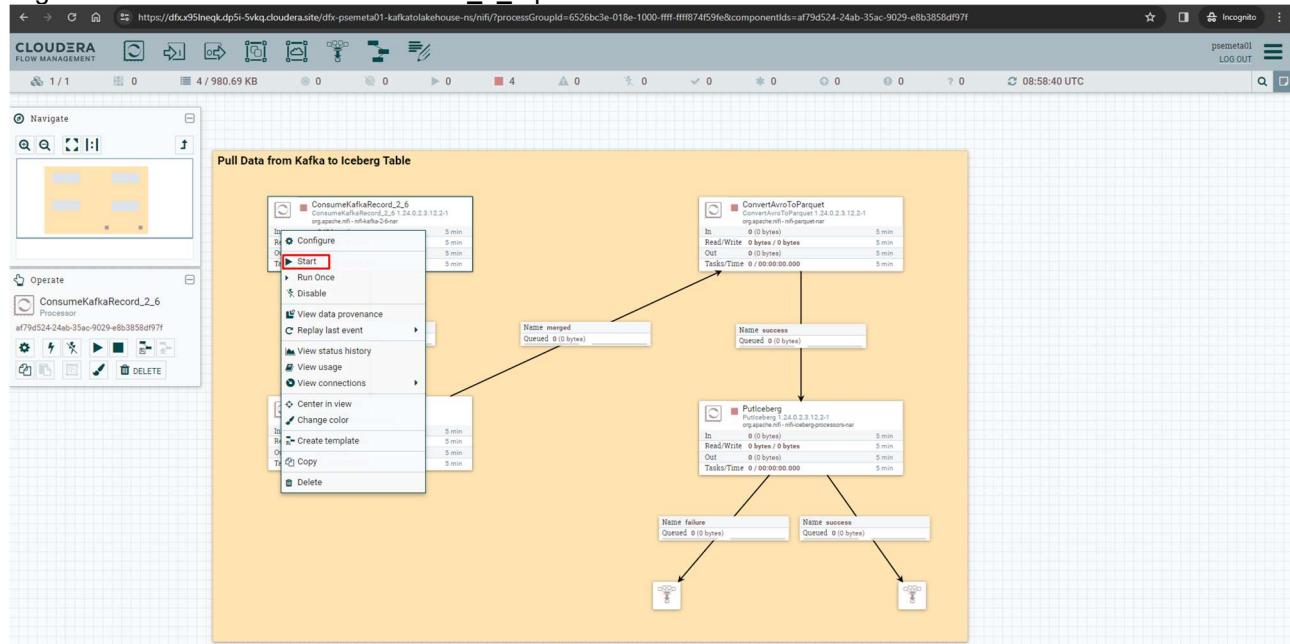


16. When opening the Process Group, you should be able to see the Processors that compose the Flow application. To summarize, there are four Processors:

- ConsumeKafkaRecord**, consumes data from the Kafka topic, reading the data in JSON and outputting in AVRO.
- MergeRecords**, to group the flow files and streamline the data flow.
- ConvertAvroToParquet**, conversion needed to store the data in PARQUET format.
- PutIceberg**, to insert the data into the table in the Lakehouse. The destination table is called `telco_kafka_iceberg`, and each user has an assigned database (user_id is the name of the database).

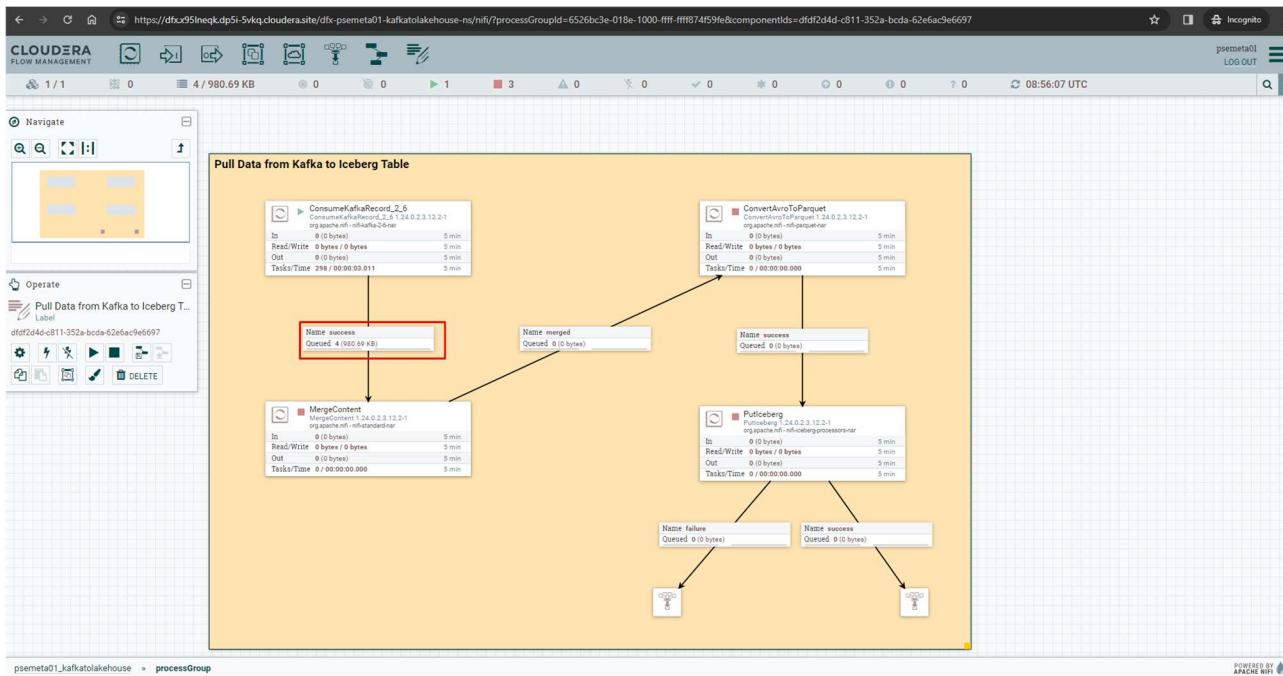
As you can see, the Processors are not started, they are paused.

Right Click on **ConsumeKafkaRecord_2_6** processor and click on **Start**.

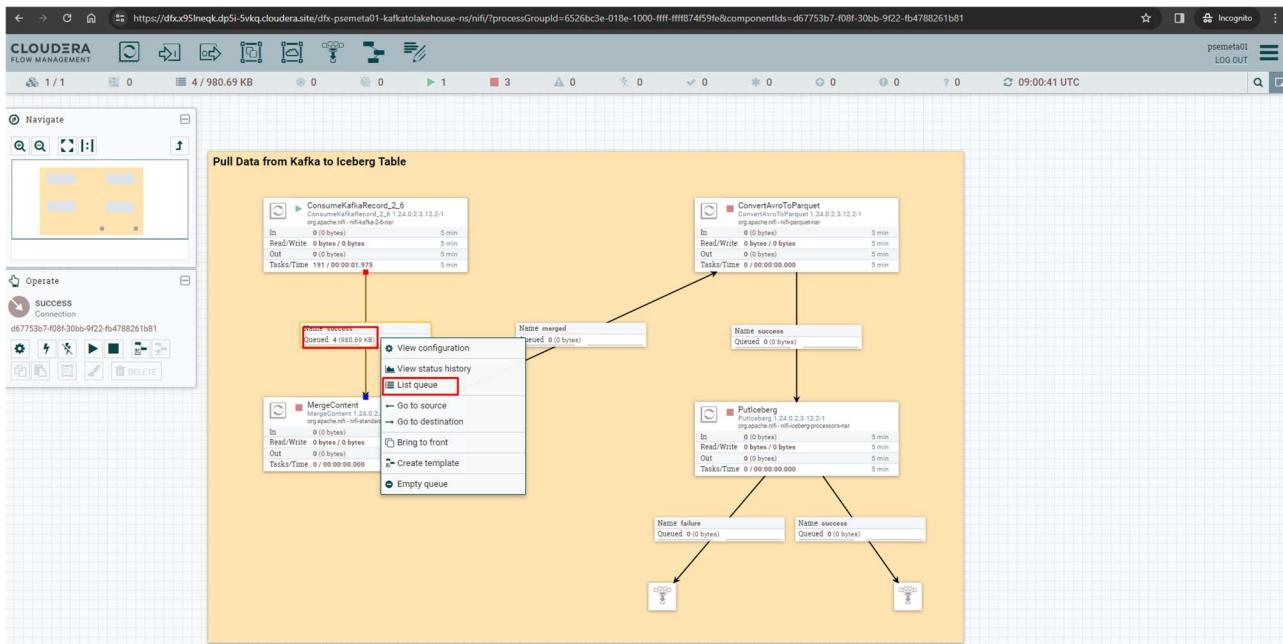


17. Flow Management allows us to see and access data in motion during the execution of the data flow. Between Processors **ConsumeKafkaRecord** (just started) and **MergeRecords**, there is a connection. This connection is what joins the Processors and transmits data from one to the other, and you can check how much data is queued at every step of the process.

You will see data start to queue up in the connector shortly after you start the first processor.



Right click on the queue and then click on the **List queue**.



18. You will see the data that is listed here. Click on the eye icon on the extreme right.

Position	UUID	Filename	File Size	Queued Duration	Lineage Duration	Penalized	Node
1	72166a04-1c1e-4c43-941-d42fb9e29300	72166a04-1c1e-4c43-941-d42fb9e29300	275.54 KB	01:03:33.912	01:03:34.301	No	dfn-nf-0-dfx-nf-dfx-psmeta01-kafkatakehouse-nf-nif
2	708fce1-3550-4558-880e-6edc5cc559d4	708fce1-3550-4558-880e-6edc5cc559d4	275.98 KB	01:03:33.715	01:03:33.874	No	dfn-nf-0-dfx-nf-dfx-psmeta01-kafkatakehouse-nf-nif
3	52f3b033-4274-1cd9-6d7f-54264ae31504	52f3b033-4274-1cd9-6d7f-54264ae31504	275.91 KB	01:03:33.574	01:03:33.695	No	dfn-nf-0-dfx-nf-dfx-psmeta01-kafkatakehouse-nf-nif
4	20daa25-c84e-45d6-a616-6a398c1329ec	20daa25-c84e-45d6-a616-6a398c1329ec	155.25 KB	01:03:33.504	01:03:33.564	No	dfn-nf-0-dfx-nf-dfx-psmeta01-kafkatakehouse-nf-nif

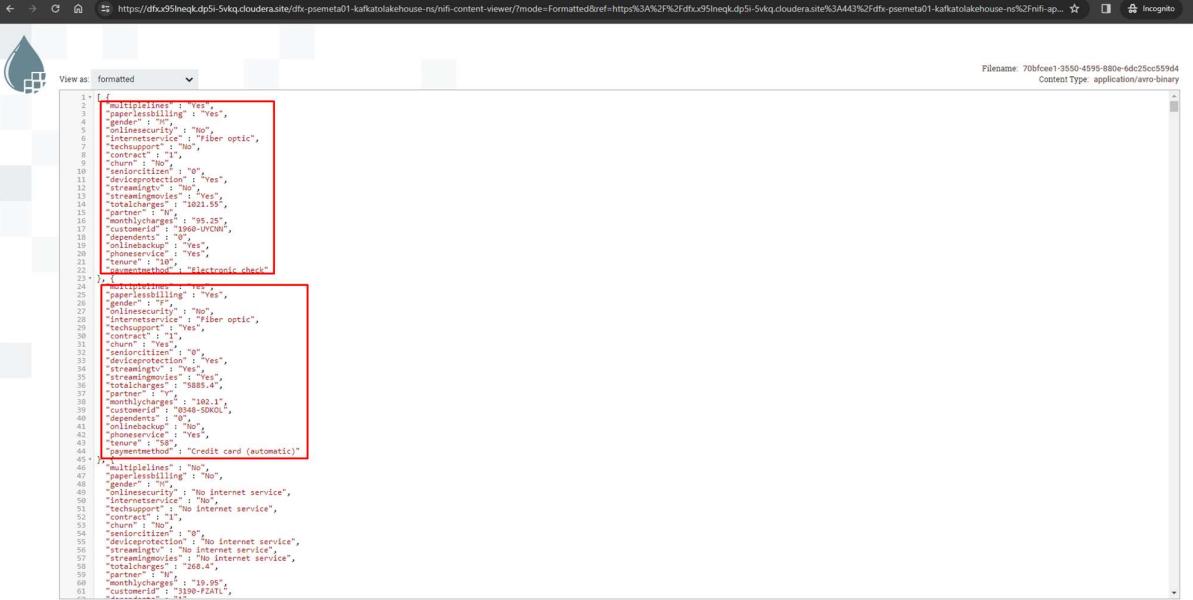
The new window that opens shows the data of the FlowFile content. Being in AVRO format, it is not fully readable. A deserializer must be selected to correctly display the data. For this, in the upper left, select the option **formatted** from the menu **View as**.

```

{
  "record": {
    "name": "nifiRecord",
    "namespace": "org.apache.nifi",
    "fields": [
      {
        "name": "multipleLines",
        "type": "string",
        "null": true
      },
      {
        "name": "paperlessBilling",
        "type": "string",
        "null": true
      }
    ],
    "name": "gender",
    "type": "string",
    "null": true
  },
  "values": [
    {
      "id": 1,
      "multipleLines": null,
      "paperlessBilling": null,
      "gender": "Male"
    },
    {
      "id": 2,
      "multipleLines": null,
      "paperlessBilling": null,
      "gender": "Female"
    },
    {
      "id": 3,
      "multipleLines": null,
      "paperlessBilling": null,
      "gender": "Other"
    },
    {
      "id": 4,
      "multipleLines": null,
      "paperlessBilling": null,
      "gender": "Non-Binary"
    }
  ]
}

```

19. Now you can display the data correctly. Notice that the fields or attributes indicated at the beginning of the workshop appear. You can close that FlowFile window and the popups, returning to the canvas with the four Processors.

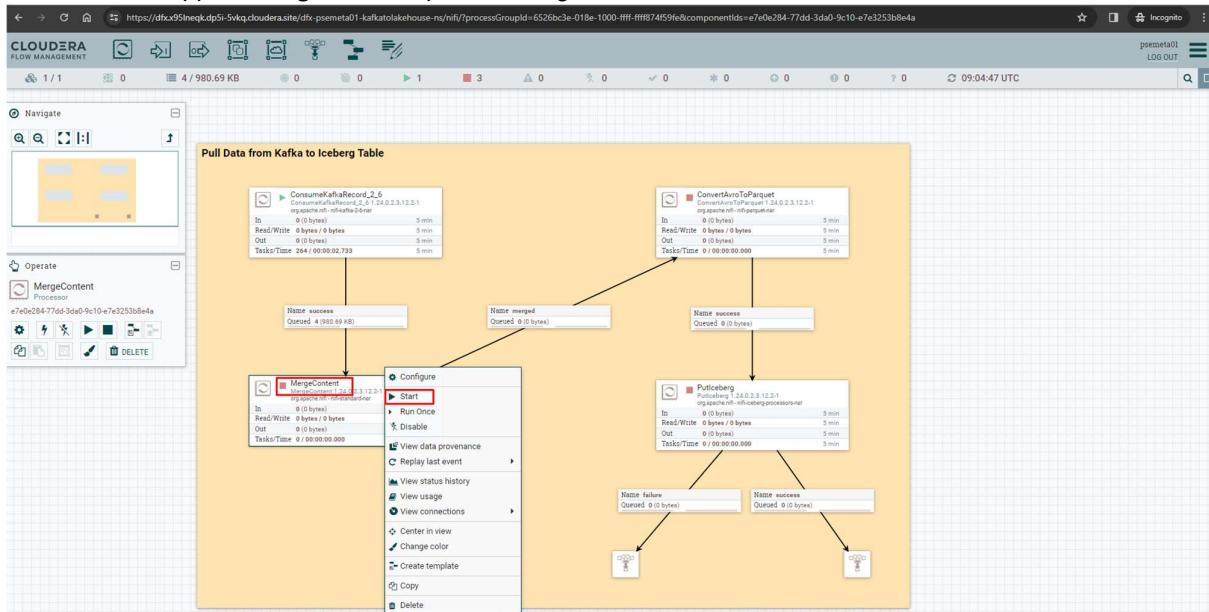


```

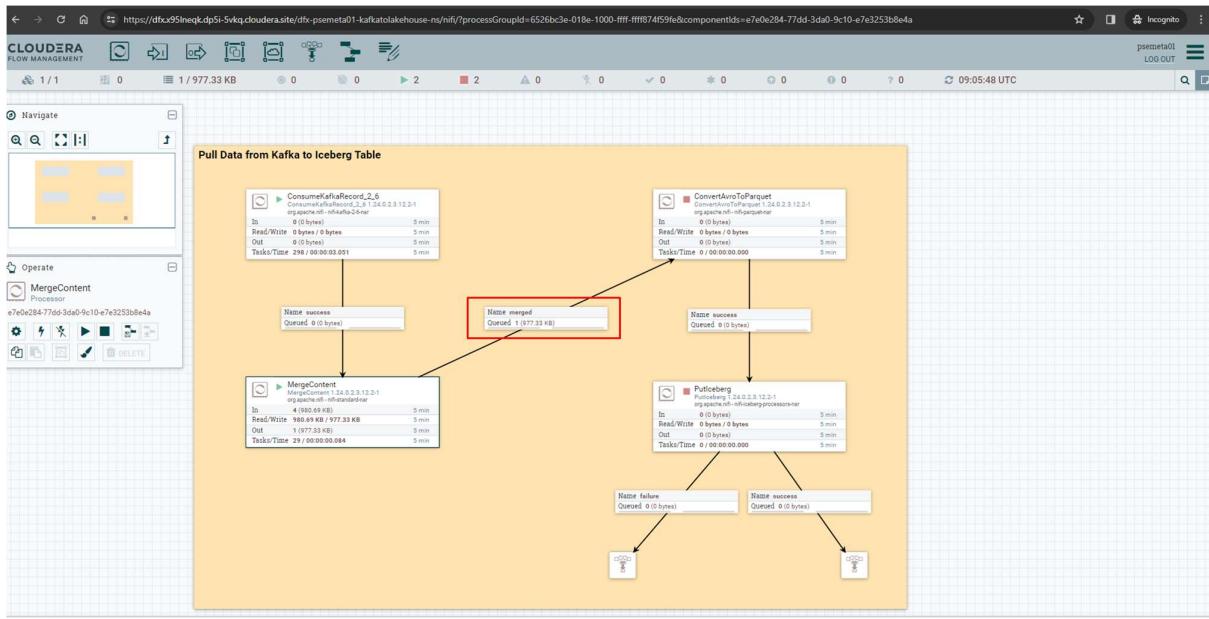
1: {
2:   "multiplelines": "Yes",
3:   "phoneline": "Yes",
4:   "gender": "F",
5:   "internetservice": "No",
6:   "techsupport": "Fiber optic",
7:   "onlinesecurity": "No",
8:   "churn": "No",
9:   "seniorcitizen": "Yes",
10:  "streamingtv": "No",
11:  "tenure": "1 year",
12:  "totalcharges": "1021.55",
13:  "monthlycharges": "95.35",
14:  "customerid": "1968-UNCW",
15:  "onlinesecurity": "Yes",
16:  "streamingtv": "Yes",
17:  "tenure": "1 year",
18:  "totalcharges": "1021.55",
19:  "monthlycharges": "95.35",
20:  "customerid": "1968-UNCW",
21:  "onlinesecurity": "Yes",
22:  "streamingtv": "Yes",
23:  "tenure": "1 year",
24:  "paymethod": "Electronic check",
25: },
26: {
27:   "multiplelines": "Yes",
28:   "phoneline": "Yes",
29:   "gender": "F",
30:   "internetservice": "Fiber optic",
31:   "techsupport": "Yes",
32:   "onlinesecurity": "Yes",
33:   "churn": "Yes",
34:   "seniorcitizen": "Yes",
35:   "streamingtv": "Yes",
36:   "tenure": "1 year",
37:   "totalcharges": "5885.4",
38:   "monthlycharges": "102.1",
39:   "customerid": "9348-SOKOL",
40:   "onlinesecurity": "Yes",
41:   "streamingtv": "No",
42:   "tenure": "58",
43:   "paymethod": "Credit card (automatic)",
44: },
45: {
46:   "multiplelines": "No",
47:   "phoneline": "No",
48:   "gender": "M",
49:   "internetservice": "No internet service",
50:   "techsupport": "No",
51:   "onlinesecurity": "No internet service",
52:   "churn": "No",
53:   "seniorcitizen": "Yes",
54:   "streamingtv": "No",
55:   "deviceprotection": "No internet service",
56:   "tenure": "1 year",
57:   "totalcharges": "268.4",
58:   "monthlycharges": "268.4",
59:   "customerid": "3190-FZATL",
60: }

```

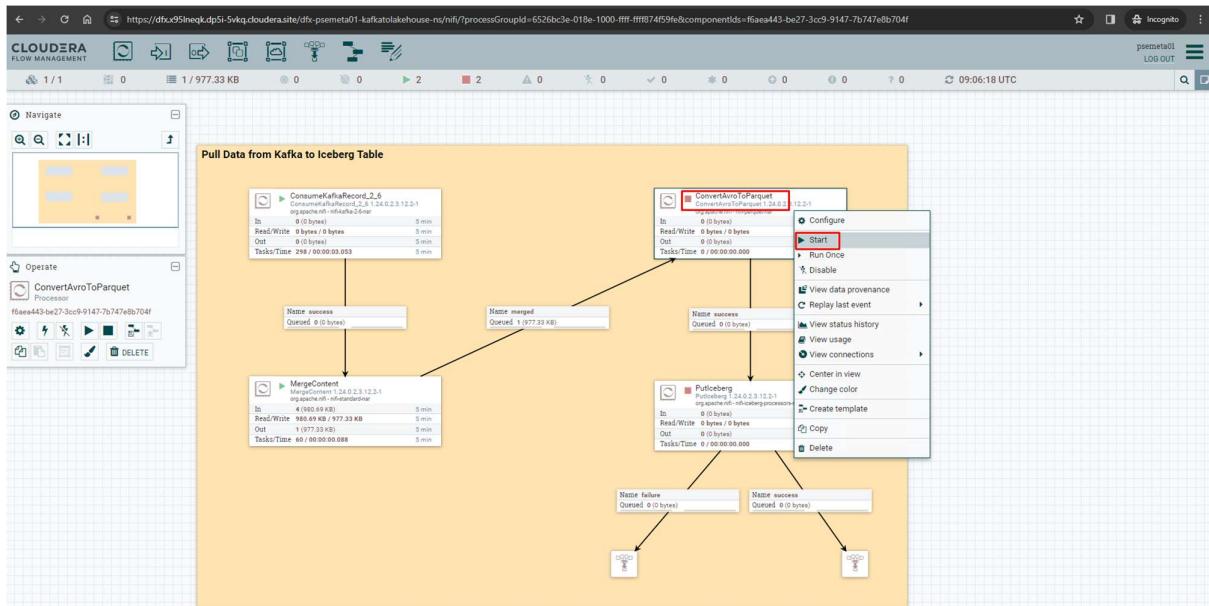
20. Start the stopped: **MergeContent** processor again to resume the flow.

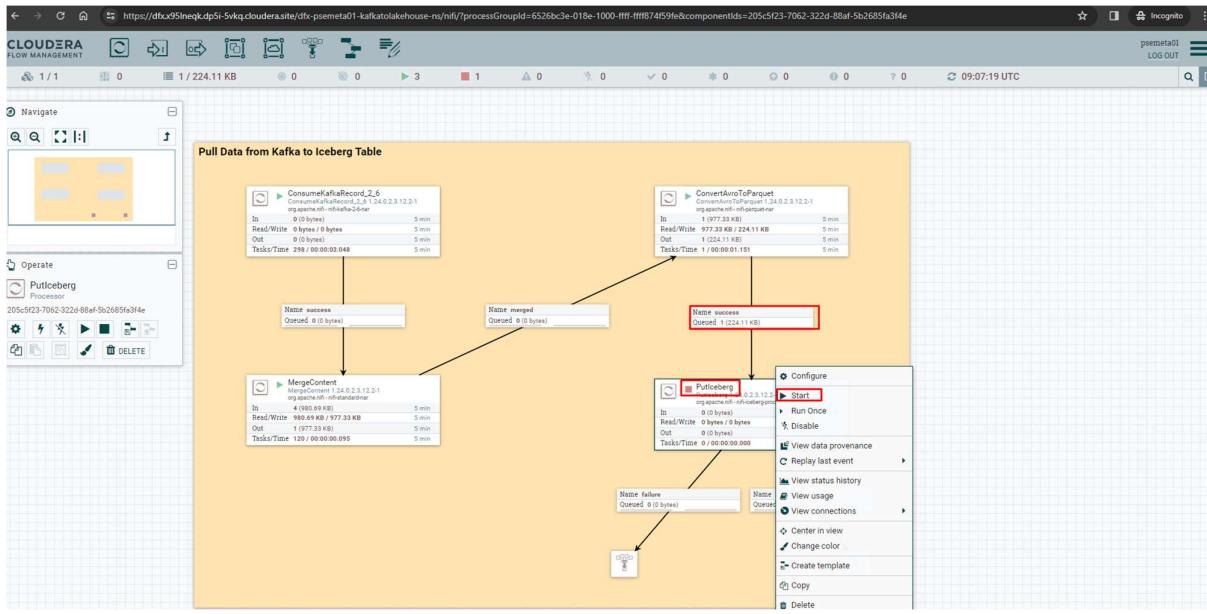


You can now see the records getting merged and passed through to the next processor.

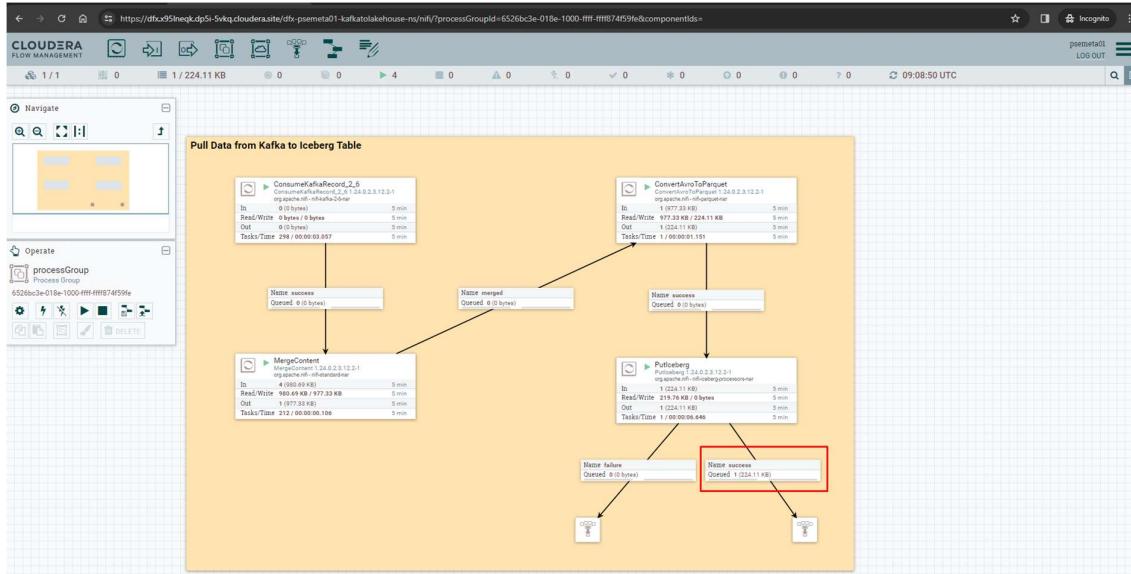


Start the next 2 processors as well **ConvertAvroToParquet** & **PutIceberg**.





If the previous steps were executed correctly, the connection of the Processor **PutIceberg** to a funnel should be of type **success**.



Once you have reached this step ask the user to check if the data got loaded. Or you can do the same by logging into the virtual warehouse.