

Data Lifecycle CDP Public Cloud

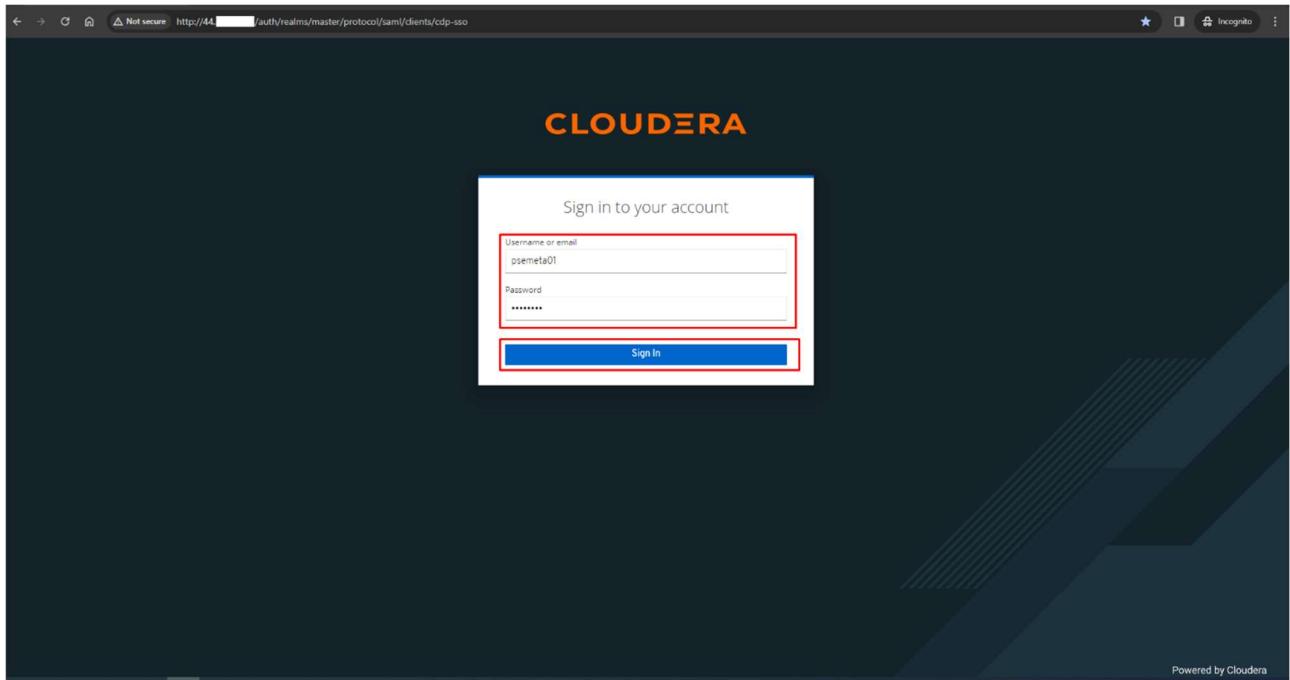
Data Flow Lab

Goals:

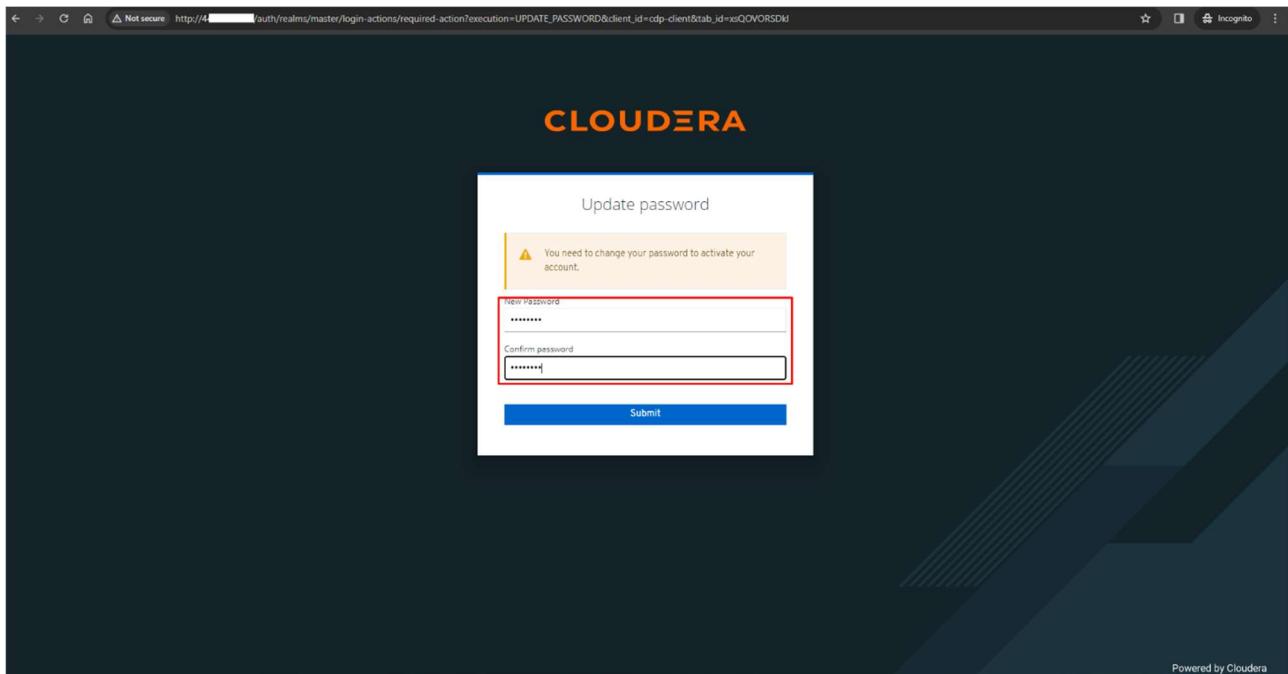
- Consume data from a Kafka topic.
- Convert the data to Parquet format.
- Store the data in a table in the Lakehouse.

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.

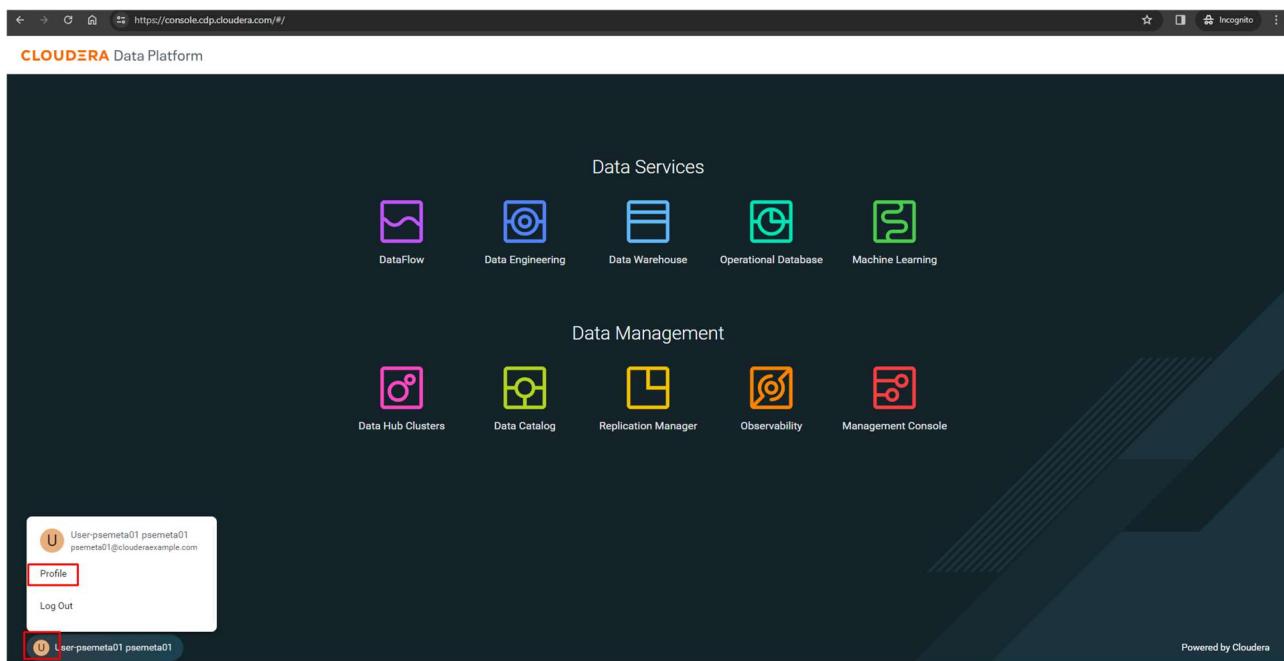


It might ask to change the password. Keep the password same as earlier which is – **changeme**



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: one for 'Password' and one for 'Confirm Password', both containing the value '*****'. 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.'

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 is titled "Users / User-psemeta01 psemeta01". It displays user details: Name (User-psemeta01 psemeta01), Email (psemeta01@clouderalexample.com), Workload User Name (psemeta01), CRN (cn:altus:iam:us-west-1:d1a4553c-a799-432d-8e54-372cc2ab95f2), 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 success message in a red-bordered box at the top right says "Success: Workload password is updated and it is being synced to environments." Below the user details, 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." and a "Help" link.

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. At the top, there's a header with the URL https://console.altus.cloudera.com/. The main content area is divided into two sections: "Data Services" and "Data Management". Under "Data Services", there are icons for DataFlow (highlighted with a red border), Data Engineering, Data Warehouse, Operational Database, and Machine Learning. Under "Data Management", there are icons for Data Hub Clusters, Data Catalog, Replication Manager, Observability, and Management Console. At the bottom left, there's a user notification: "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

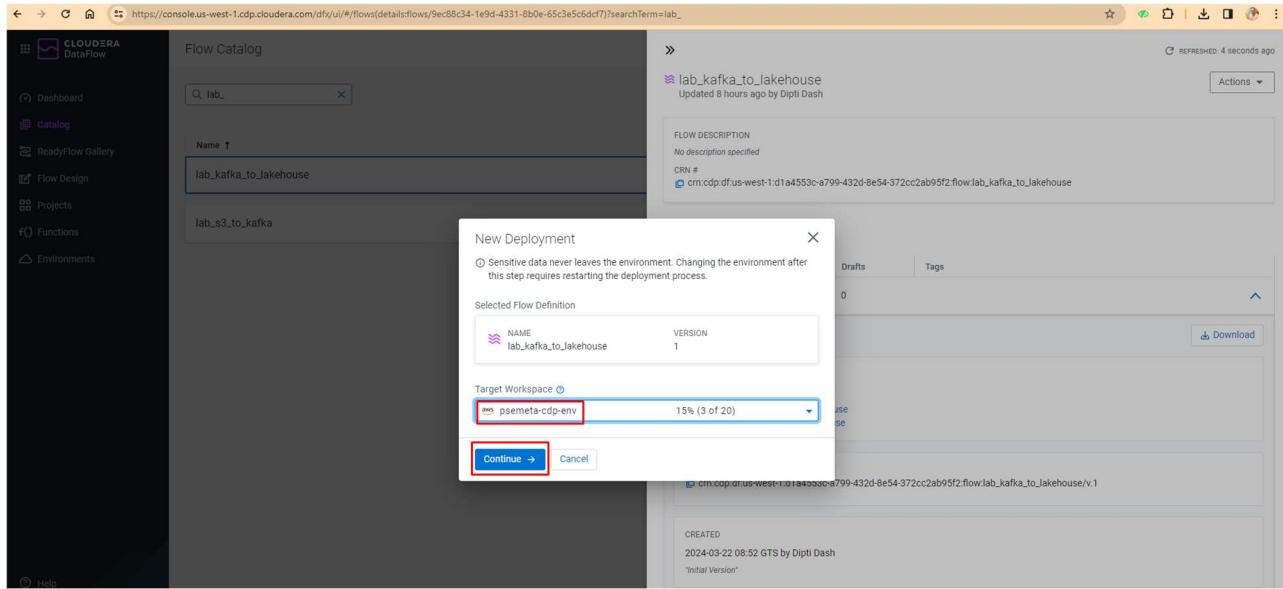
Flow Catalog

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 interface. On the left, a sidebar menu includes Dashboard, Catalog (selected), ReadyFlow Gallery, Flow Design, Projects, Functions, and Environments. The main area is titled 'Flow Catalog' and contains a search bar with the placeholder 'lab_'. Below the search bar, a table lists flows with columns 'Name' and 'Actions'. Two flows are listed: 'lab_kafka_to_lakehouse' (selected) and 'lab_s3_to_kafka'. The right side of the screen displays detailed information for the selected flow:

- Flow Details:** Name: lab_kafka_to_lakehouse, Updated: 8 hours ago by Dipti Dash.
- 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. A table shows three versions: Version 1 (Deployed 2 times), Version 2 (Deployed 0 times), and Drafts (0). The 'Deploy' button is highlighted with a red box.
- Deployments:** (2)
 - aws_psemeta-cdp-env
 - psemeta01_kafkatolakehouse
 - psemeta31_lafkatolakehouse
- 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 (0/64), Color (None).

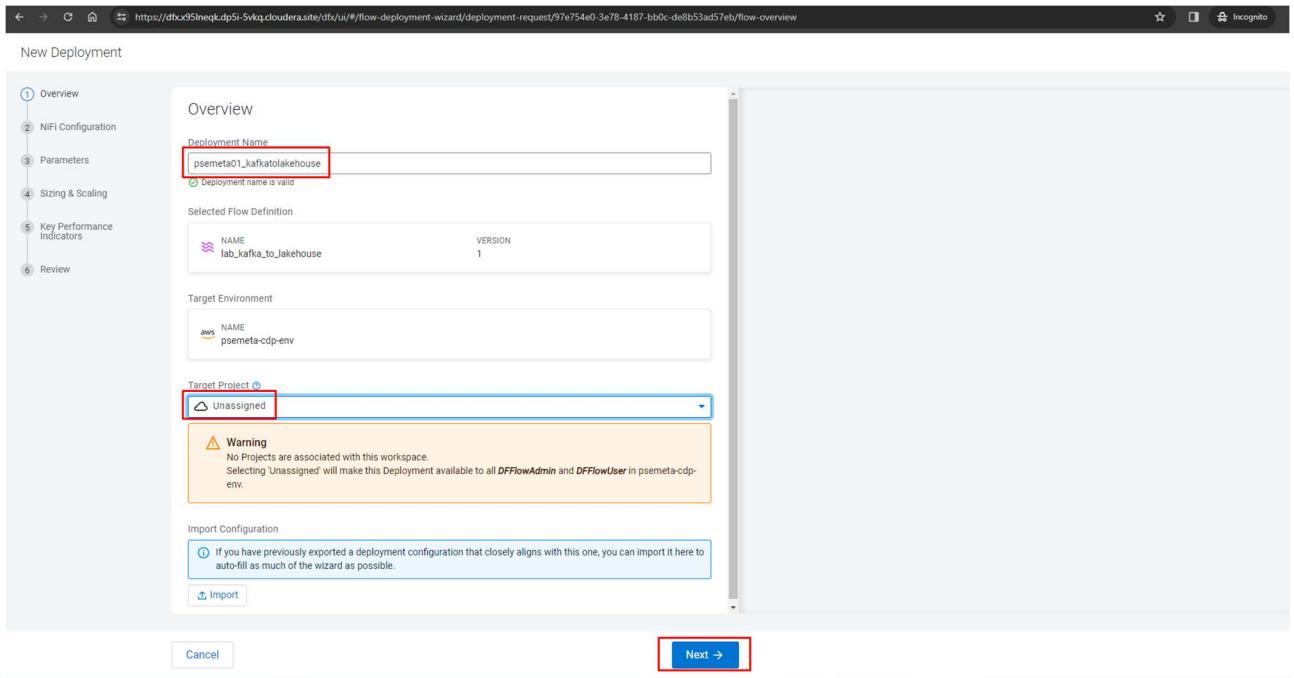
3. The following popup window allows you to select the DataFlow cluster in which you want to deploy the Flow. In this case, the cluster to be selected is **psemeta-cdp-env**. The workshop instructor will tell you which environment to select. Once selected, click **Continue**.



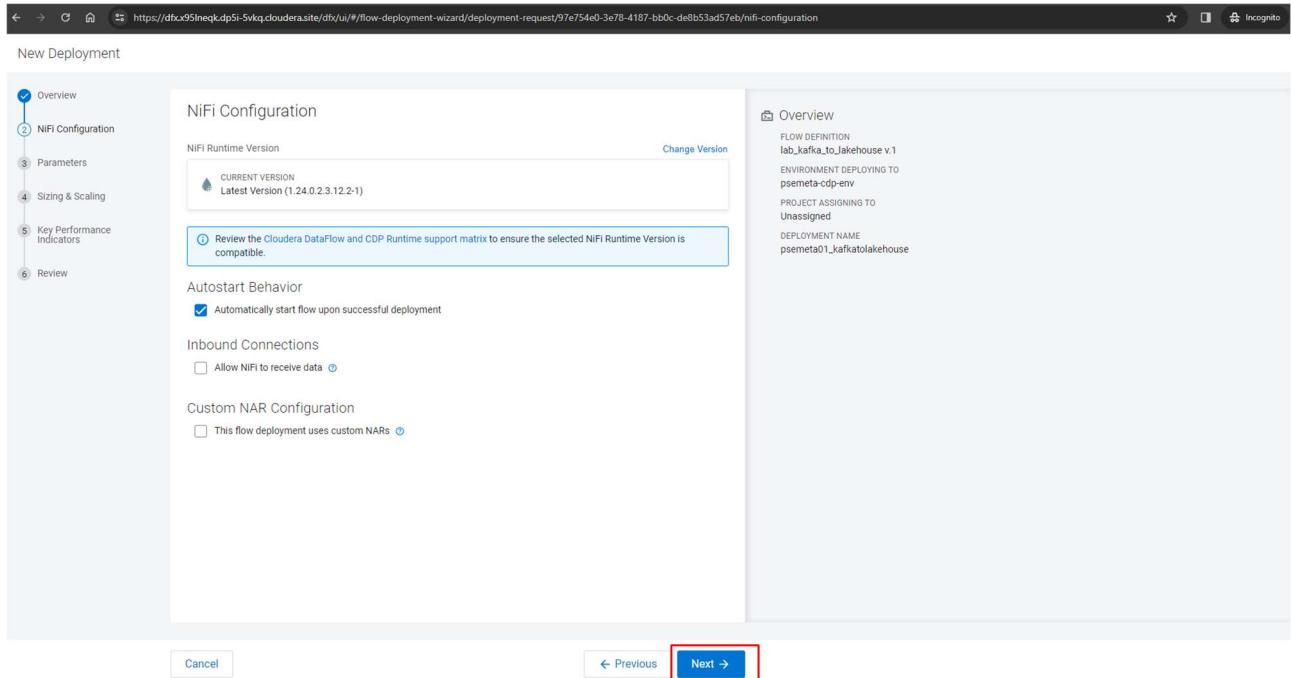
4. From this point, you will need to enter the Flow configuration. Start by assigning a Deployment Name, Target Project, and click Next.

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

Click on **Next**.



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



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!**

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**

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

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

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

New Deployment

Sizing & Scaling
Select the NiFi node size and the number of nodes provisioned for your flow.

NiFi Node Sizing

<input checked="" type="radio"/> Extra Small	<input type="radio"/> Small	<input type="radio"/> Medium	<input type="radio"/> Large
2 vCores Per Node 4 GB Per Node	3 vCores Per Node 6 GB Per Node	6 vCores Per Node 12 GB Per Node	12 vCores Per Node 24 GB Per Node

Number of NiFi Nodes
Auto Scaling
 Disabled
Nodes:

Storage Selection

<input checked="" type="radio"/> Standard	<input type="radio"/> Performance
512 GB Content Repo Size 512 GB Provenance Repo Size 256 GB Flow File Repo Size 3000 IOPS 150 MB/s Max Throughput	512 GB Content Repo Size 512 GB Provenance Repo Size 256 GB Flow File Repo Size 6000 IOPS 300 MB/s Max Throughput

Overview
FLOW DEFINITION: lab_kafka_to_lakehouse v.1
ENVIRONMENT: psemeta-cdp-env
PROJECT: Unassigned
DEPLOYMENT NAME: psemeta01_kafkatolakehouse

NiFi Configuration
NIFI RUNTIME VERSION: Latest Version (1.24.0.2.3.12.2-1)
AUTO-START FLOW: Yes
INBOUND CONNECTIONS: No
CUSTOM NAR CONFIGURATION: No

Parameters
parameters
COP WORKLOAD USER PASSWORD: [Sensitive Value Provided]
COP WORKLOAD USERNAME: psemeta01
COPENVIRONMENT: core-site.xml
ssl-client.xml
hive-site.xml

Cancel **← Previous** **Next >**

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

New Deployment

Key Performance Indicators
Set up KPIs to track specific performance metrics of a deployed flow. Click and drag to reorder how they are displayed.
[Learn more](#)

Overview
FLOW DEFINITION: lab_kafka_to_lakehouse v.1
ENVIRONMENT: psemeta-cdp-env
PROJECT: Unassigned
DEPLOYMENT NAME: psemeta01_kafkatolakehouse

NiFi Configuration
NIFI RUNTIME VERSION: Latest Version (1.24.0.2.3.12.2-1)
AUTO-START FLOW: Yes
INBOUND CONNECTIONS: No
CUSTOM NAR CONFIGURATION: No

Parameters
parameters
COP WORKLOAD USER PASSWORD: [Sensitive Value Provided]
COP WORKLOAD USERNAME: psemeta01
COPENVIRONMENT: core-site.xml
ssl-client.xml
hive-site.xml

Cancel **← Previous** **Next >**

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

New Deployment

Review

View CLI Command

Overview

FLOW DEFINITION: lab_kafka_to_lakehouse v.1
ENVIRONMENT: deploying to psemeta-cdp-env
PROJECT ASSIGNING TO: Unassigned
DEPLOYMENT NAME: psemeta01_kafkatolakehouse

NiFi Configuration

NIFI RUNTIME VERSION: Latest Version (1.24.0.2.3.12.2-1)
AUTO-START FLOW: Yes
INBOUND CONNECTIONS: No
CUSTOM NAR CONFIGURATION: No

Parameters

parameters
CDP WORKLOAD USER PASSWORD: [Sensitive Value Provided]
CDP WORKLOAD USERNAME: psemeta01
CDP ENVIRONMENT: core-site.xml

Cancel **Previous** **Deploy**

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**.

CLOUDERA DataFlow

Dashboard

Status Environments Deployments Projects Reset

Status	Name
Flow Stopped	dipitdash_kafka_to_lakehouse psemeta-cdp-env Unassigned
Flow Stopped	dipitdash_s3_to_kafka_flow psemeta-cdp-env Unassigned
Deploying	psemeta01_kafkatolakehouse psemeta-cdp-env Unassigned

Alerts

Deployment Initiated
Initiated deployment of [psemeta01_kafkatolakehouse]

KPIs System Metrics Alerts

Active Alerts: No alerts to display.

Event History:

- Provisioning NiFi Cluster 2024-03-22 10:44 GTS
- Deployment Initiated 2024-03-22 10:44 GTS

Show Only: Info Warning Error

Load More

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".

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, there's a sidebar with links like Dashboard, Catalog, ReadyFlow Gallery, Flow Design, Projects, Functions, and Environments. The main area has tabs for Status, Environments, Deployments, and Projects. Under Deployments, there's a table with two rows: 'dptidash_kafka_to_lakehouse' (Status: Stopped) and 'dptidash_s3_to_kafka_flow' (Status: Stopped). A third row, 'psemeta01_kafkatolakehouse', is highlighted with a blue border and shows a 'Deploying' status. To the right of the table, there's a 'KPIs' section, an 'Alerts' section (which is currently active), and an 'Event History' section. The 'Event History' table lists various deployment events with their timestamps. At the top right, there's a 'REFRESHED: 6 seconds ago' message and a 'Actions' dropdown menu. This menu contains options like 'Manage Deployment', 'View in NiFi' (which is highlighted with a red box), 'View Workspace', and 'Manage Deployments'. Below the table, there's a 'Load More' button.

Status	Name
Stopped	dptidash_kafka_to_lakehouse
Stopped	dptidash_s3_to_kafka_flow
Deploying	psemeta01_kafkatolakehouse

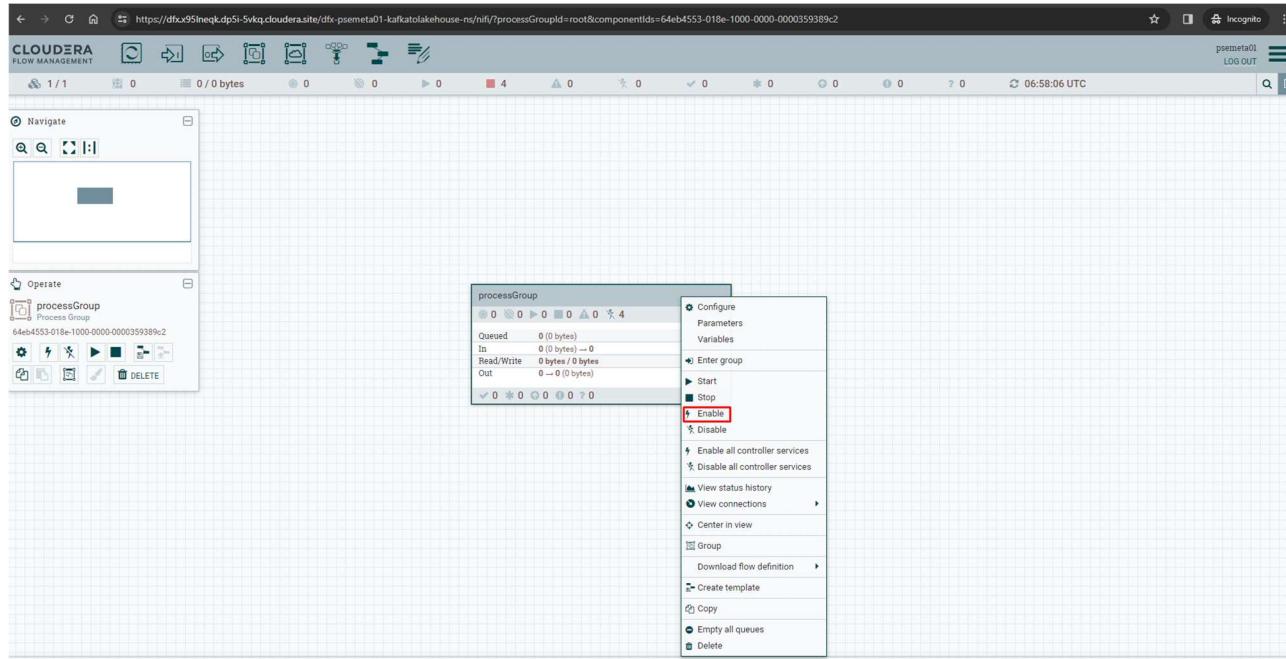
Actions

- Manage Deployment
- View in NiFi**
- View Workspace
- Manage Deployments

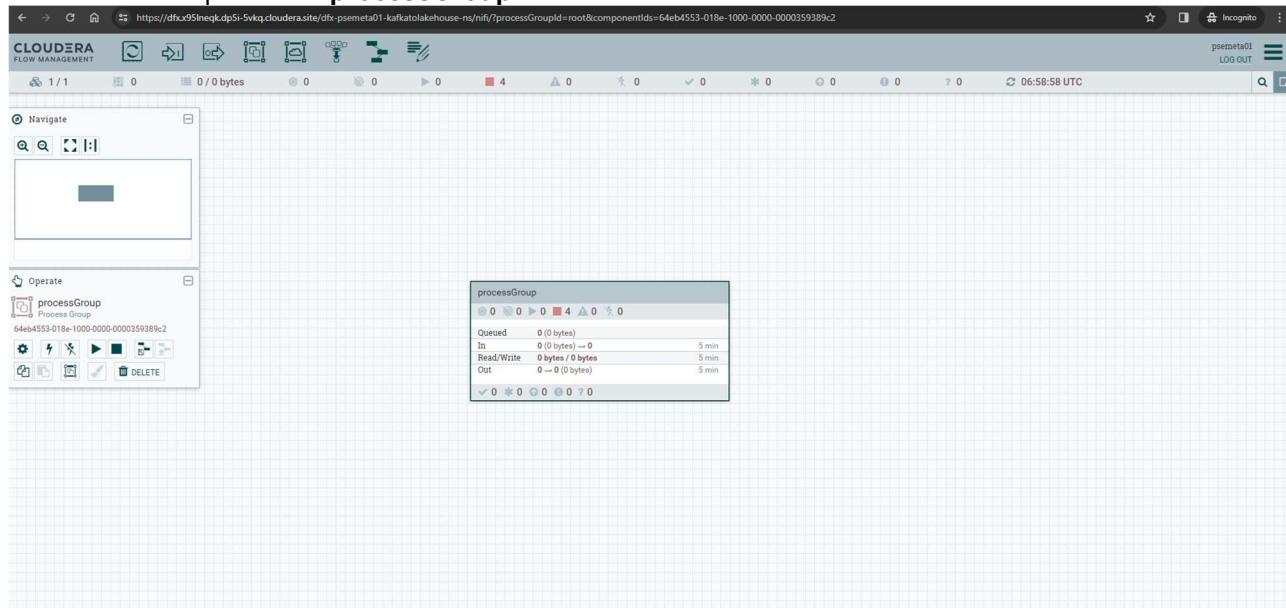
Event History	
Show Only:	
<input checked="" type="checkbox"/>	Deployment Successful
<input checked="" type="checkbox"/>	NiFi Flow Started
<input checked="" type="checkbox"/>	KPI Alert Rules Activated
<input checked="" type="checkbox"/>	Activating KPI Alert Rules
<input checked="" type="checkbox"/>	Starting NiFi Flow
<input checked="" type="checkbox"/>	Default Alert Rules Activated
<input checked="" type="checkbox"/>	Activating Default Alert Rules
<input checked="" type="checkbox"/>	NiFi Flow Imported
<input checked="" type="checkbox"/>	Importing NiFi Flow
<input checked="" type="checkbox"/>	Preparing For NiFi Flow Import

Load More

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



Double click the processor **processGroup**.

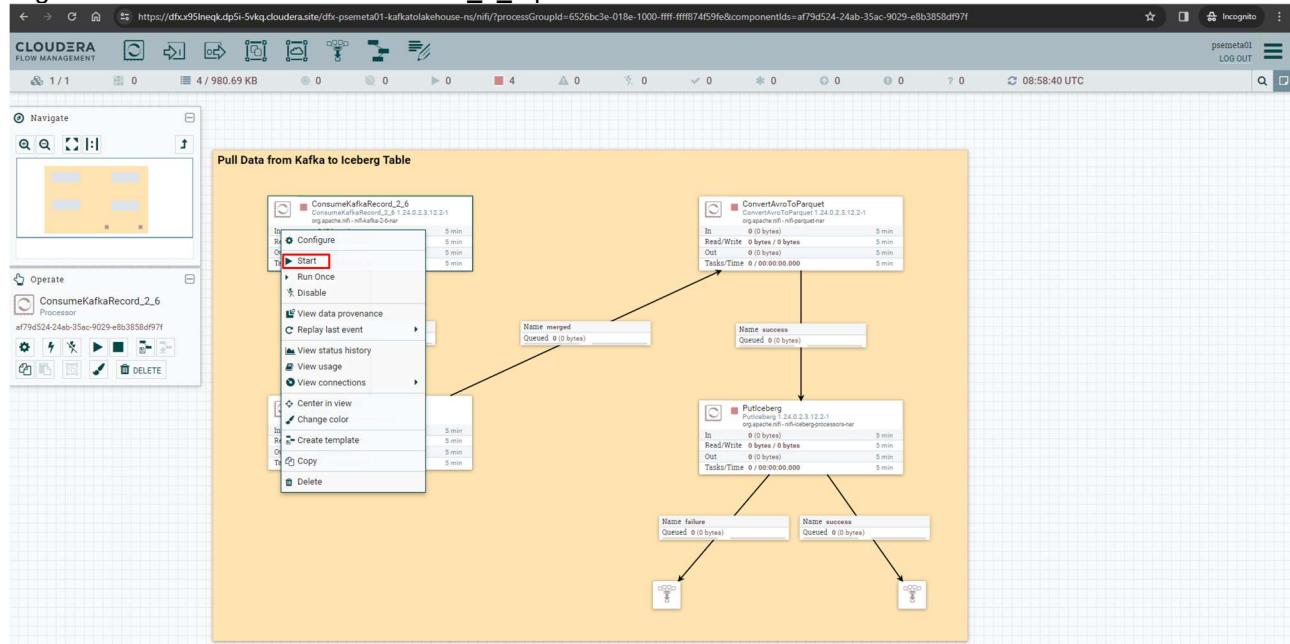


16. When opening the Process Group, you should be able to see the Processors that compose the Flow application. To summarize, there are four Processors:

- a. **ConsumeKafkaRecord**, consumes data from the Kafka topic, reading the data in JSON and outputting in AVRO.
- b. **MergeRecords**, to group the flow files and streamline the data flow.
- c. **ConvertAvroToParquet**, conversion needed to store the data in PARQUET format.
- d. **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).

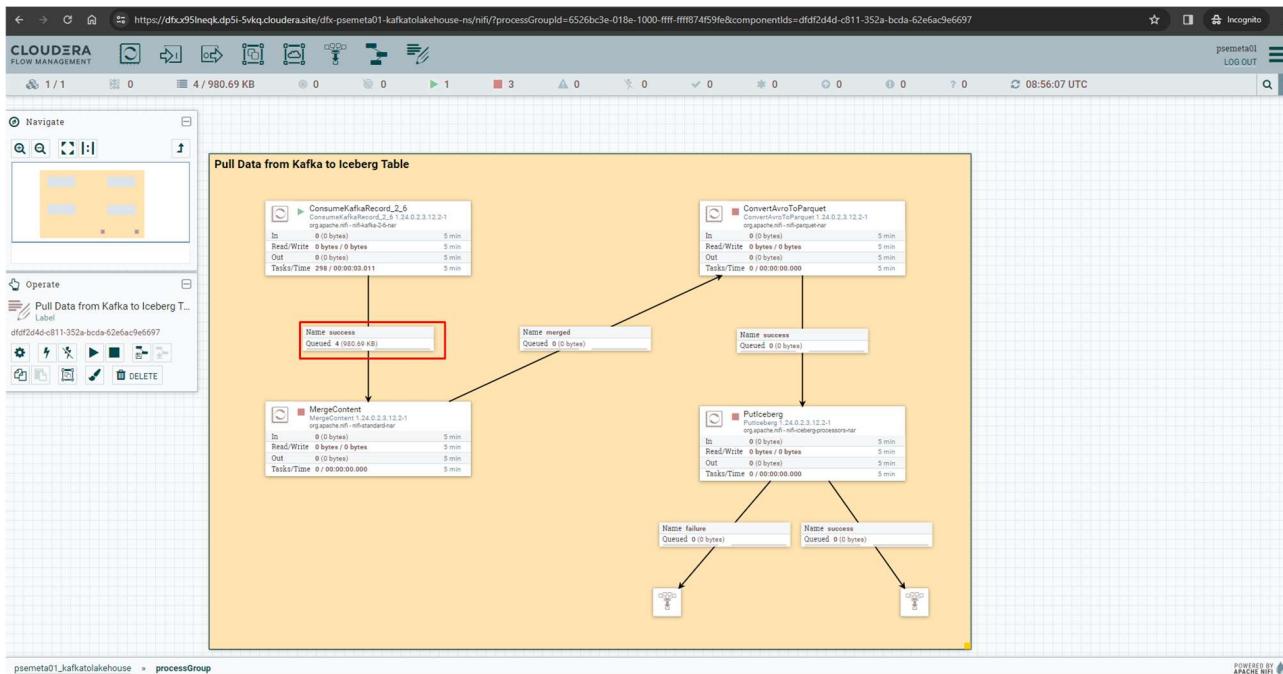
As you can see, the Processors are not started, they are paused.

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

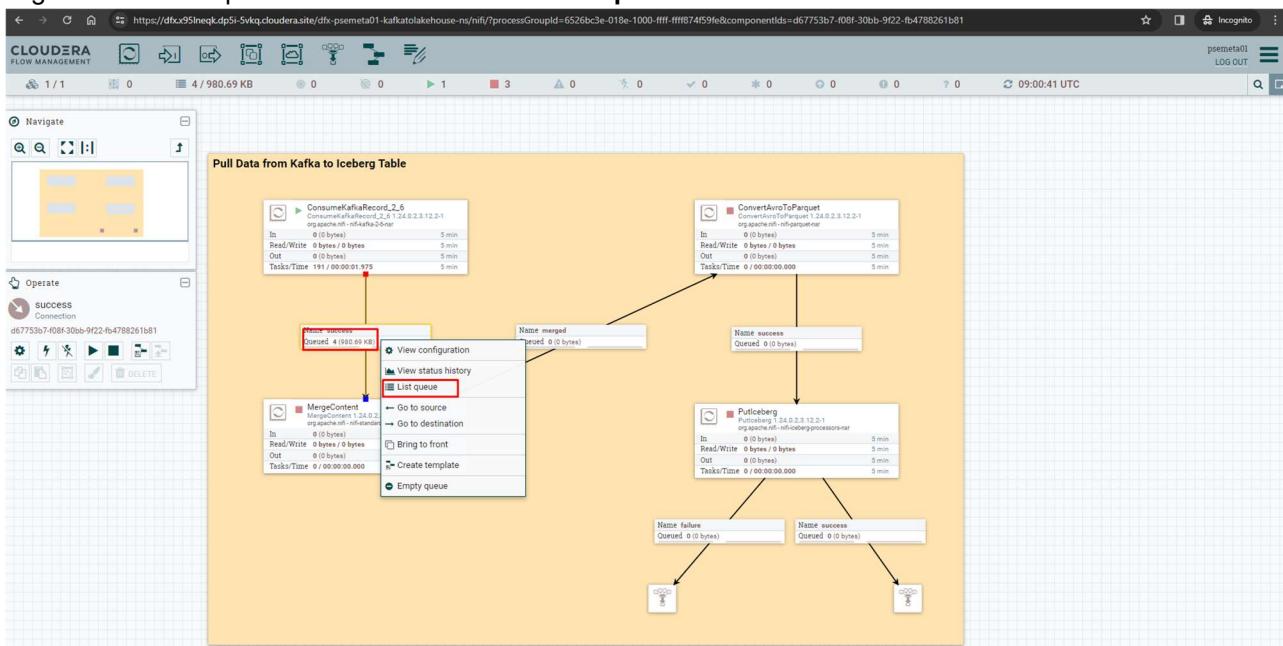


17. Flow Management allows us to see and access data in motion during the execution of the data flow. Between Processors **ConsumeKafkaRecord** (just started) and **MergeRecords**, there is a connection. This connection is what joins the Processors and transmits data from one to the other, and you can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

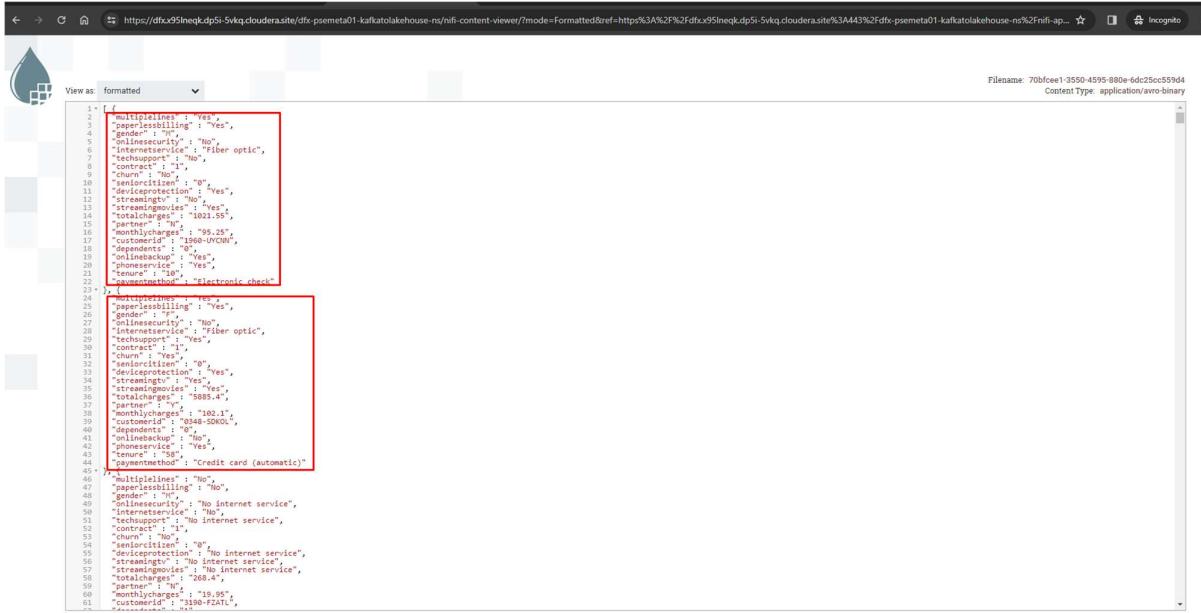


18. You will see the data that is listed here. Click on the eye icon on the extreme right.

CloudBees DataFlow							
SUCCESS							
Displaying 4 of 4 (980.69 KB)							
Position	UUID	Filename	File Size	Queued Duration	Lineage Duration	Penalized	Node
1	7216ea041c1e4c43b941-d42f69e29300	7216ea04-1c1e-4c43-b941-d42f69e29300	275.54 KB	01:03:33.912	01:03:34.301	No	dfx-nf-nf-dtv-nfl-dfx-psmeta01:kafka:topic1...   
2	70bfee13550-4595-880e-6ed2cc55944	70bfee13550-4595-880e-6ed2cc55944	273.98 KB	01:03:33.715	01:03:33.874	No	dfx-nf-nf-dtv-nfl-dfx-psmeta01:kafka:topic1...   
3	52f3b033-ff27-41cd-9671-54264ae31504	52f3b033-ff27-41cd-9671-54264ae31504	275.91 KB	01:03:33.574	01:03:33.695	No	dfx-nf-nf-dtv-nfl-dfx-psmeta01:kafka:topic1...   
4	20daaf25-c84e-45d6-a616-6a398c1329ec	20daaf25-c84e-45d6-a616-6a398c1329ec	155.25 KB	01:03:33.504	01:03:33.564	No	dfx-nf-nf-dtv-nfl-dfx-psmeta01:kafka:topic1...   

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**.

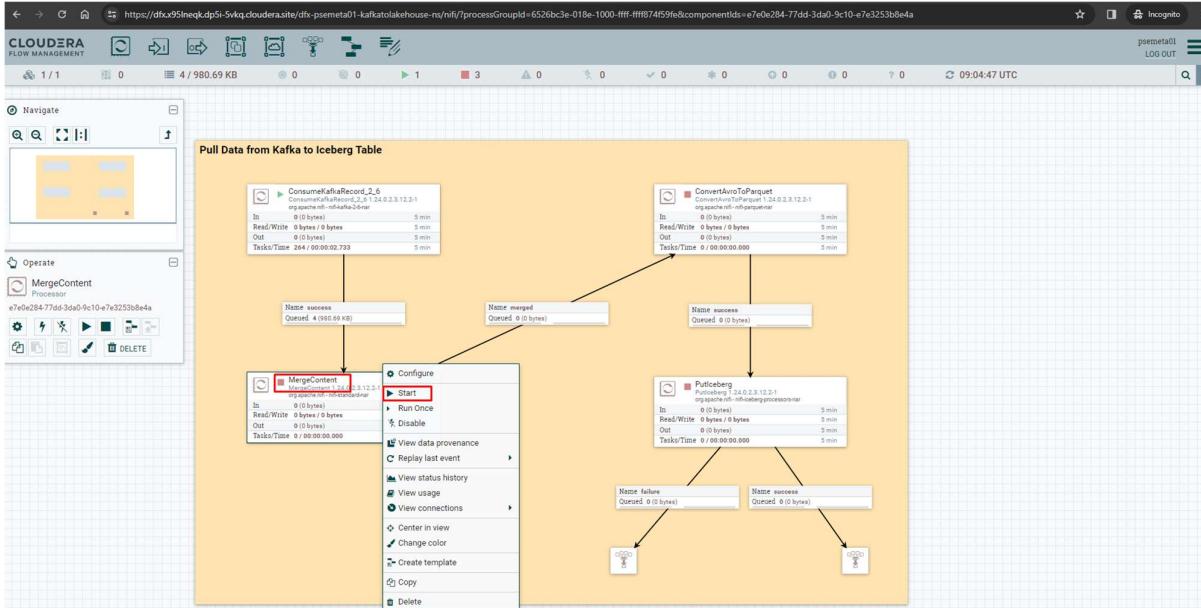
19. Now you can display the data correctly. Notice that the fields or attributes indicated at the beginning of the workshop appear. You can close that FlowFile window and the popups, returning to the canvas with the four Processors.



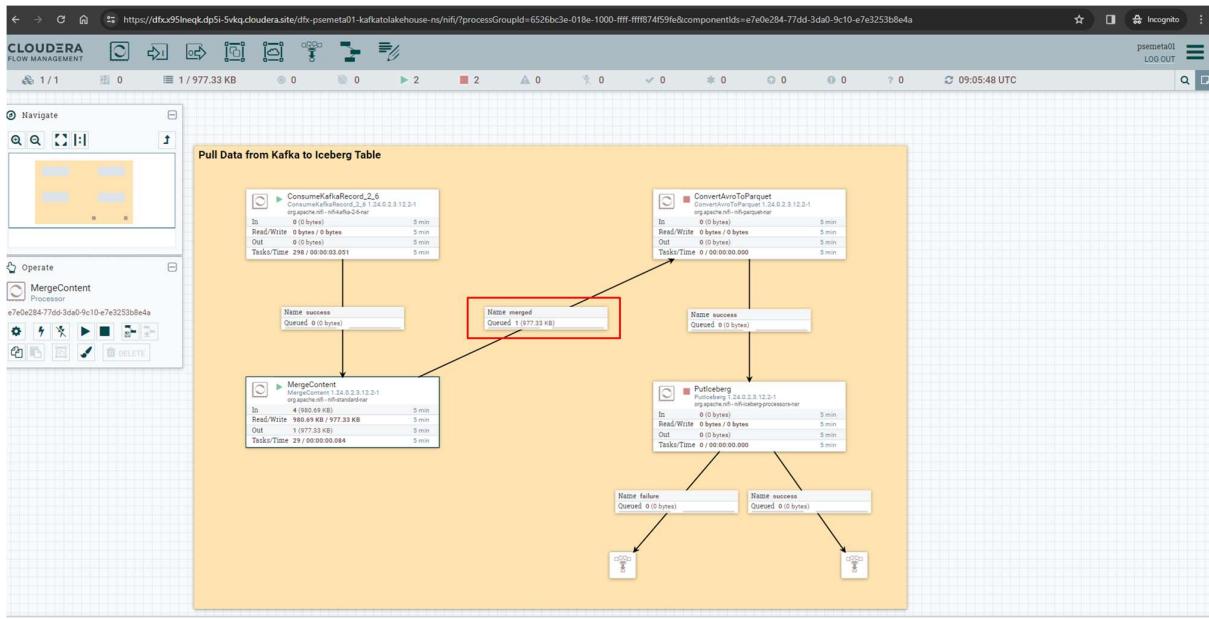
```

1: {
2:   "multiplelines": "Yes",
3:   "paperlessbilling": "Yes",
4:   "gender": "Male",
5:   "partner": "No",
6:   "internetservice": "Fiber optic",
7:   "onlinesecurity": "No",
8:   "contract": "1 year",
9:   "churn": "No",
10:  "seniorcitizen": "No",
11:  "deviceprotection": "Yes",
12:  "streamingmovies": "Yes",
13:  "totalcharges": "1021.55",
14:  "monthlycharges": "99.25",
15:  "customerid": "70bfce1-3550-4595-880e-6dc25cc559d4",
16:  "dependents": "0",
17:  "phoneservice": "Yes",
18:  "tenure": "10",
19:  "internet": "Electronic check"
20: },
21: {
22:   "multiplelines": "Yes",
23:   "paperlessbilling": "Yes",
24:   "gender": "Female",
25:   "partner": "Yes",
26:   "internetservice": "Fiber optic",
27:   "onlinesecurity": "Yes",
28:   "contract": "1 year",
29:   "churn": "Yes",
30:   "seniorcitizen": "No",
31:   "deviceprotection": "Yes",
32:   "streamingmovies": "Yes",
33:   "totalcharges": "1021.55",
34:   "monthlycharges": "99.25",
35:   "customerid": "70bfce1-3550-4595-880e-6dc25cc559d4",
36:   "dependents": "0",
37:   "phoneservice": "Yes",
38:   "tenure": "8",
39:   "internet": "Credit card (automatic)"
40: },
41: {
42:   "multiplelines": "No",
43:   "paperlessbilling": "No",
44:   "gender": "Male",
45:   "partner": "Yes",
46:   "internetservice": "No Internet service",
47:   "onlinesecurity": "No",
48:   "contract": "1 year",
49:   "churn": "Yes",
50:   "seniorcitizen": "Yes",
51:   "deviceprotection": "No Internet service",
52:   "streamingmovies": "No Internet service",
53:   "totalcharges": "1021.55",
54:   "monthlycharges": "99.25",
55:   "customerid": "70bfce1-3550-4595-880e-6dc25cc559d4",
56:   "dependents": "0",
57:   "phoneservice": "Yes",
58:   "tenure": "8",
59:   "internet": "Fiber optic"
60: }
61: ]
62: 
```

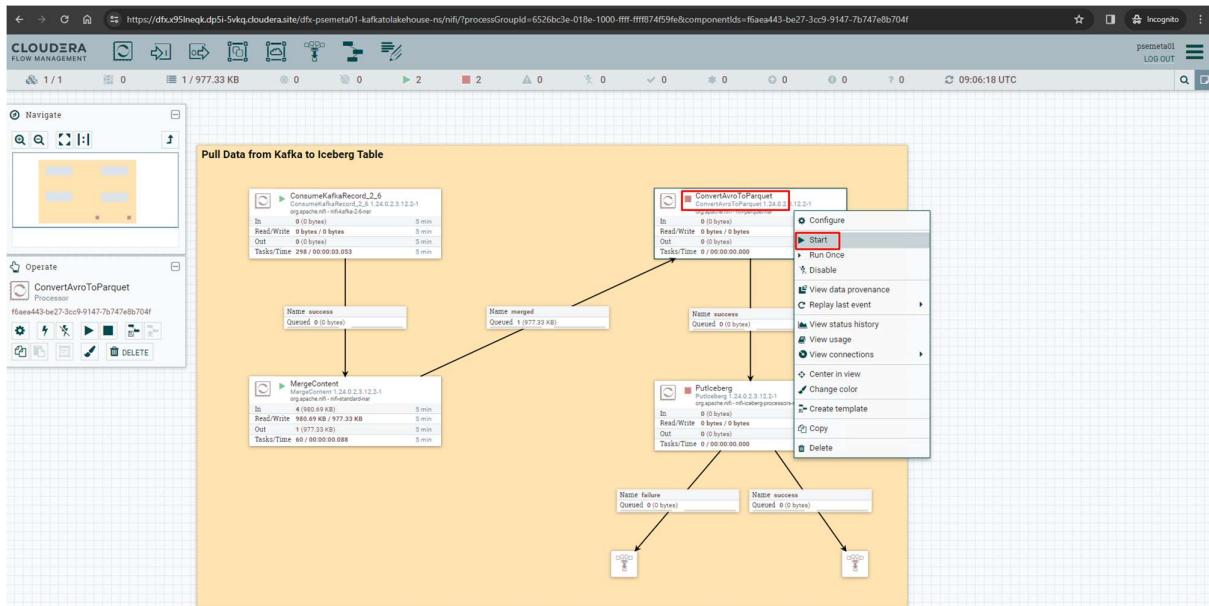
20. Start the stopped: **MergeContent** processor again to resume the flow.

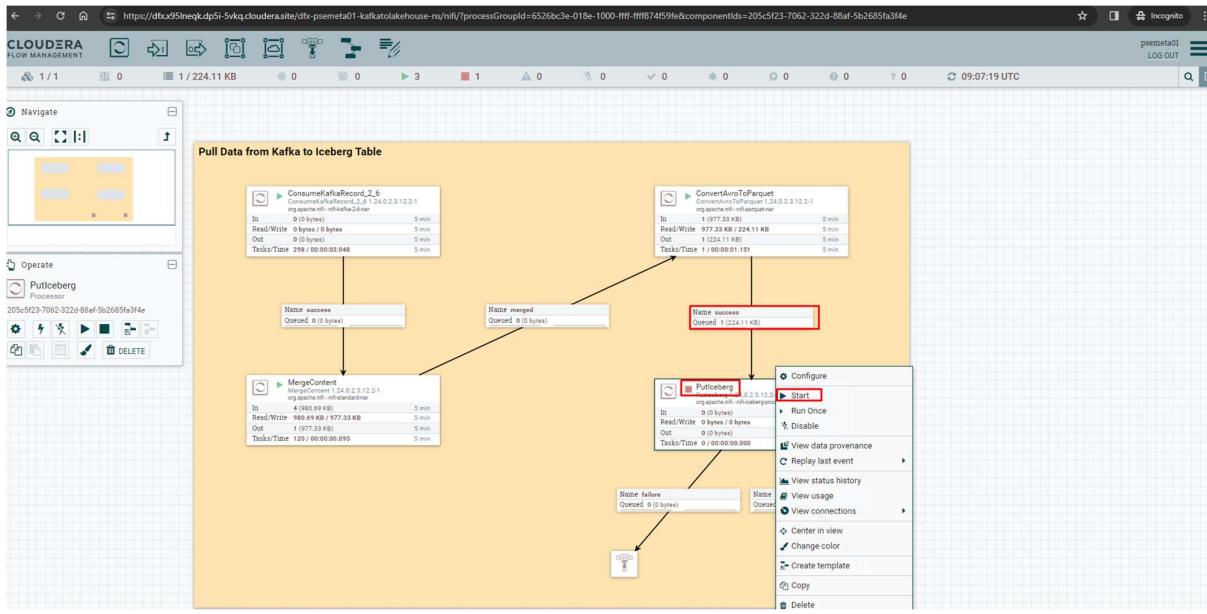


You can now see the records getting merged and passed through to the next processor.

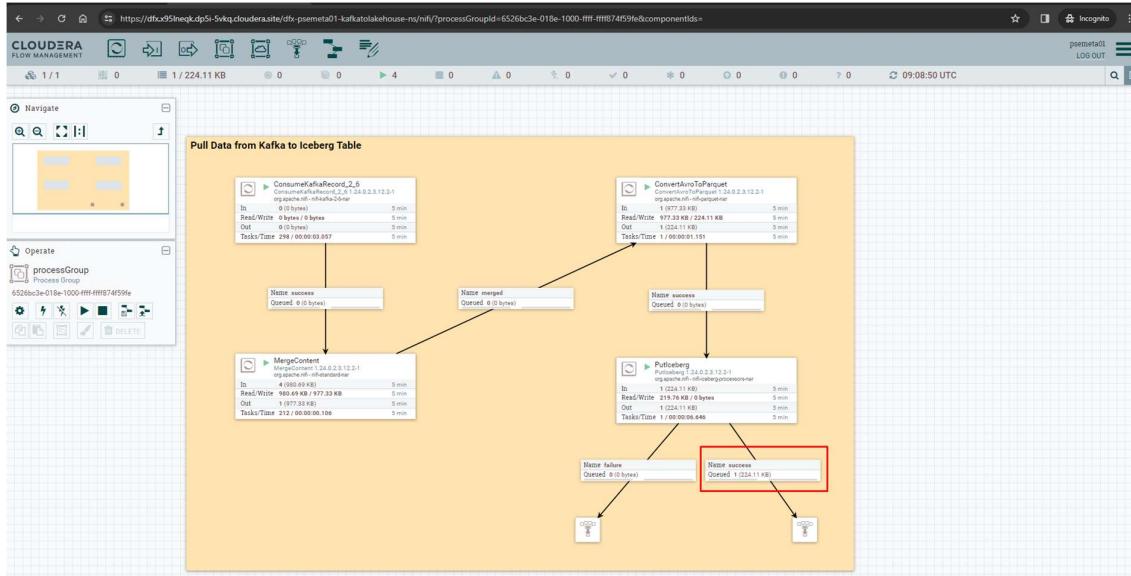


Start the next 2 processors as well **ConvertAvroToParquet** & **PutIceberg**.





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.