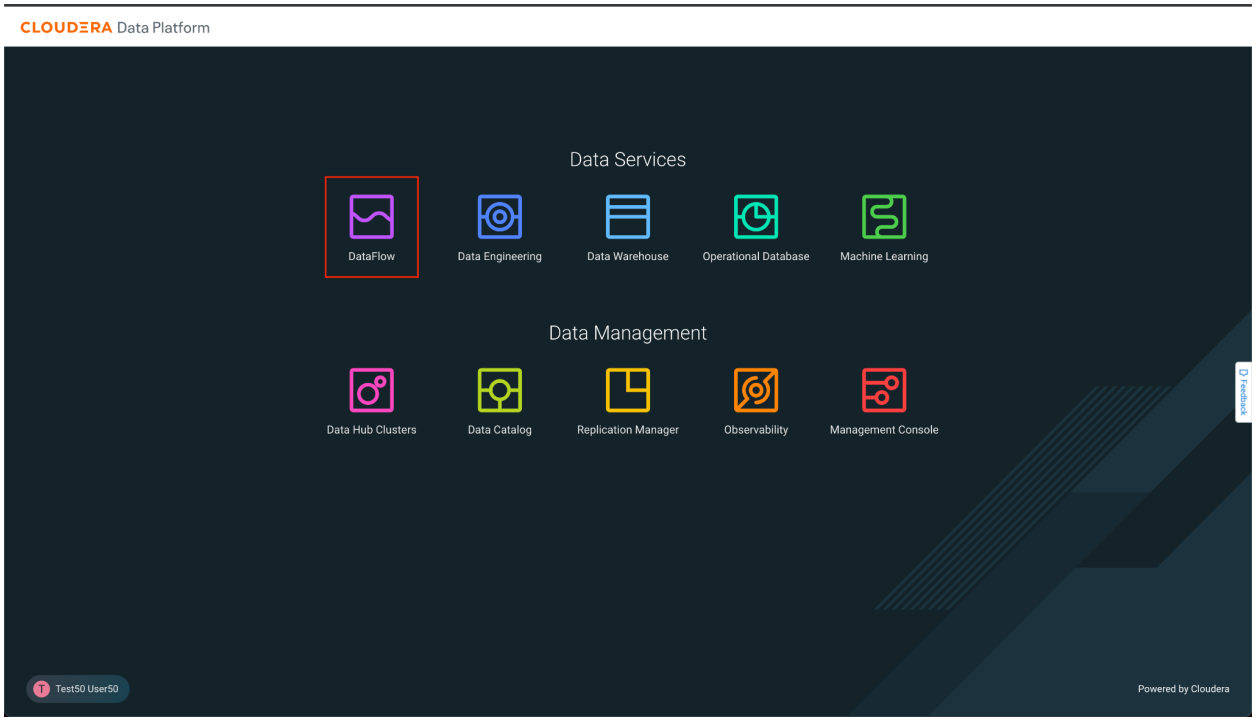
Data Lifecycle CDP Public Cloud

Data Flow Lab

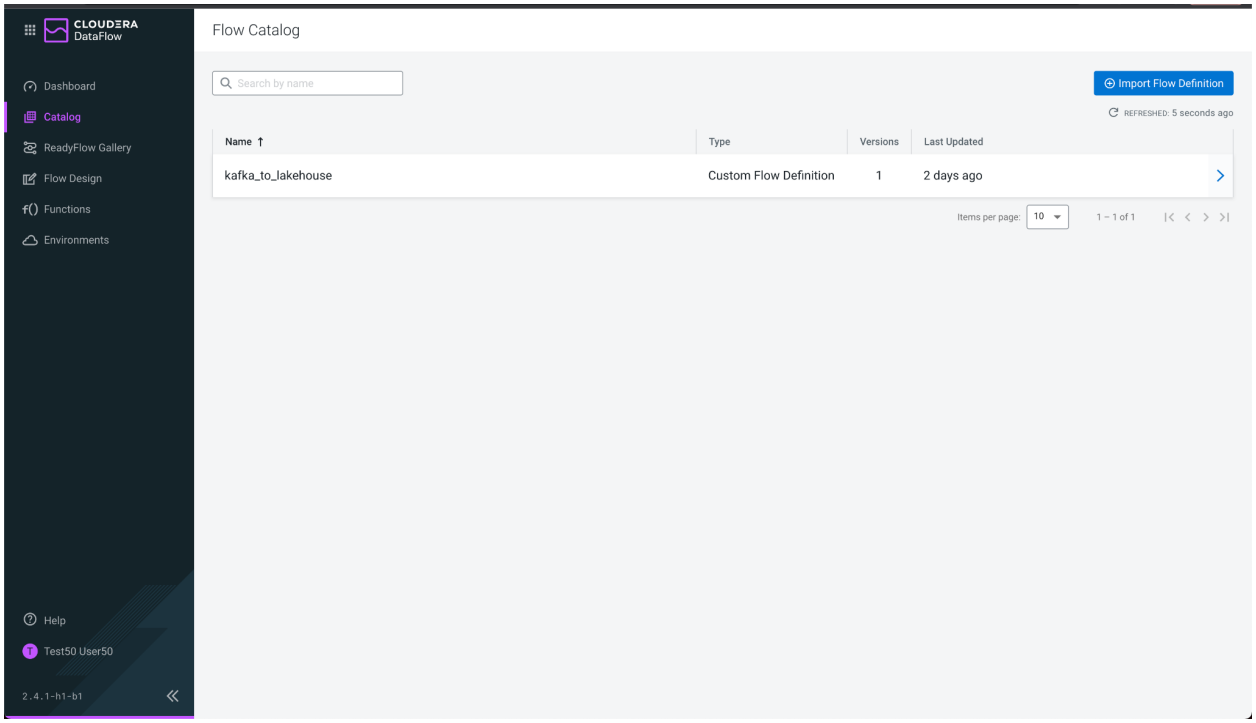
Goals:

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

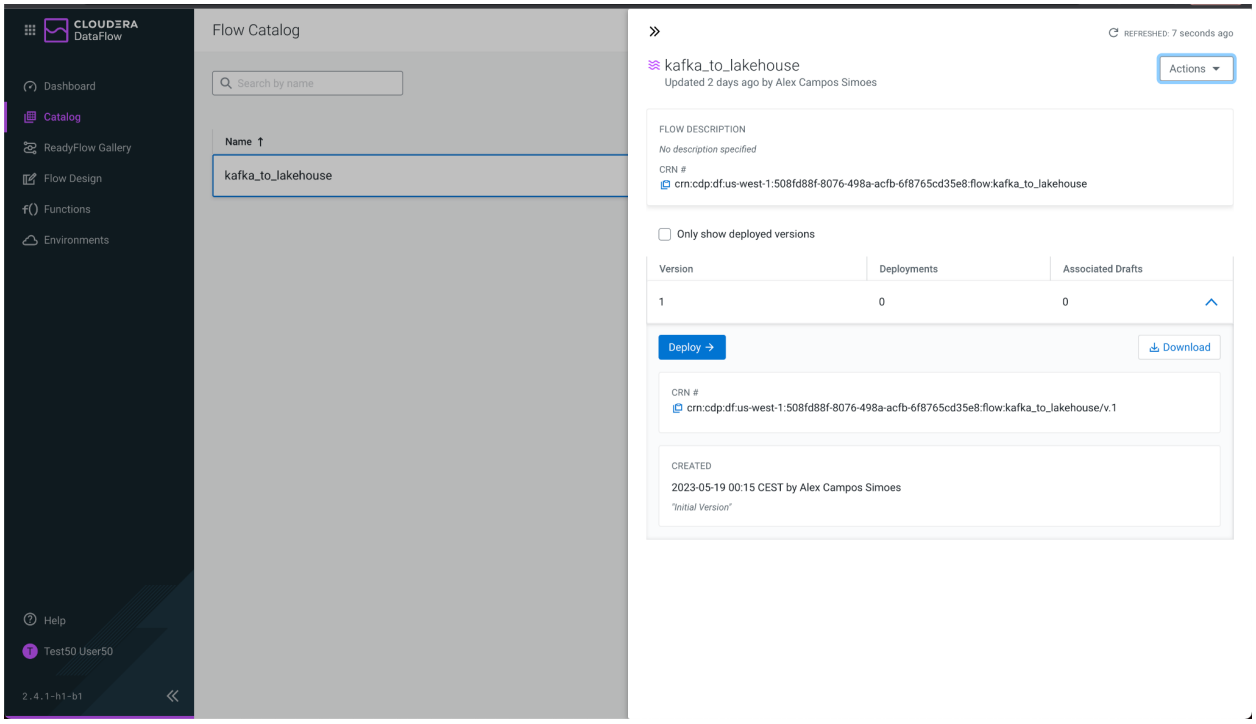
1. Click on DataFlow from CDP PC Home:



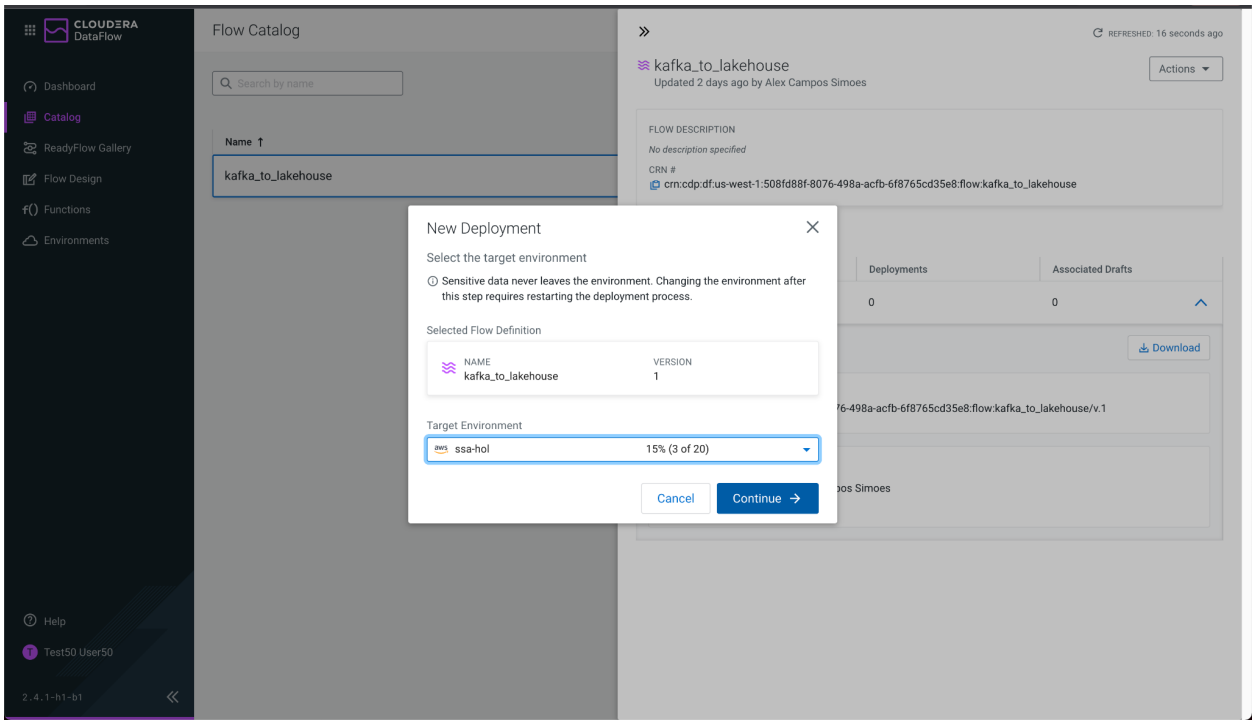
2. Once in DataFlow, click on the option **Catalog** from the left menu. The data ingestion application templates are listed here. For the purpose of this workshop, we have created and published a template that allows you to read Kafka topic data and ingest/store it in the Lakehouse provided by CDP Public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



3. 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 on that button.

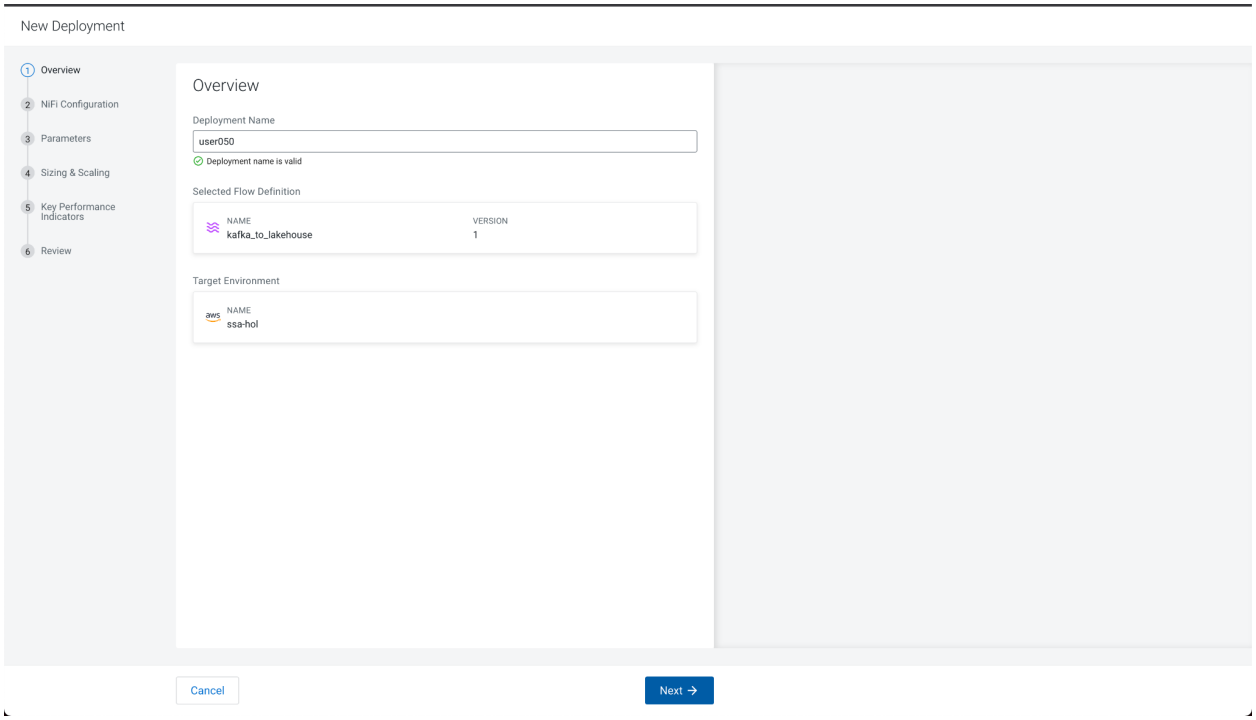


4. The following popup window allows you to select the DataFlow cluster in which you want to deploy the Flow. In this case, the cluster to be selected is **ssa-hol**. The workshop instructor will tell you which environment to select. Once selected, click **Continue**.

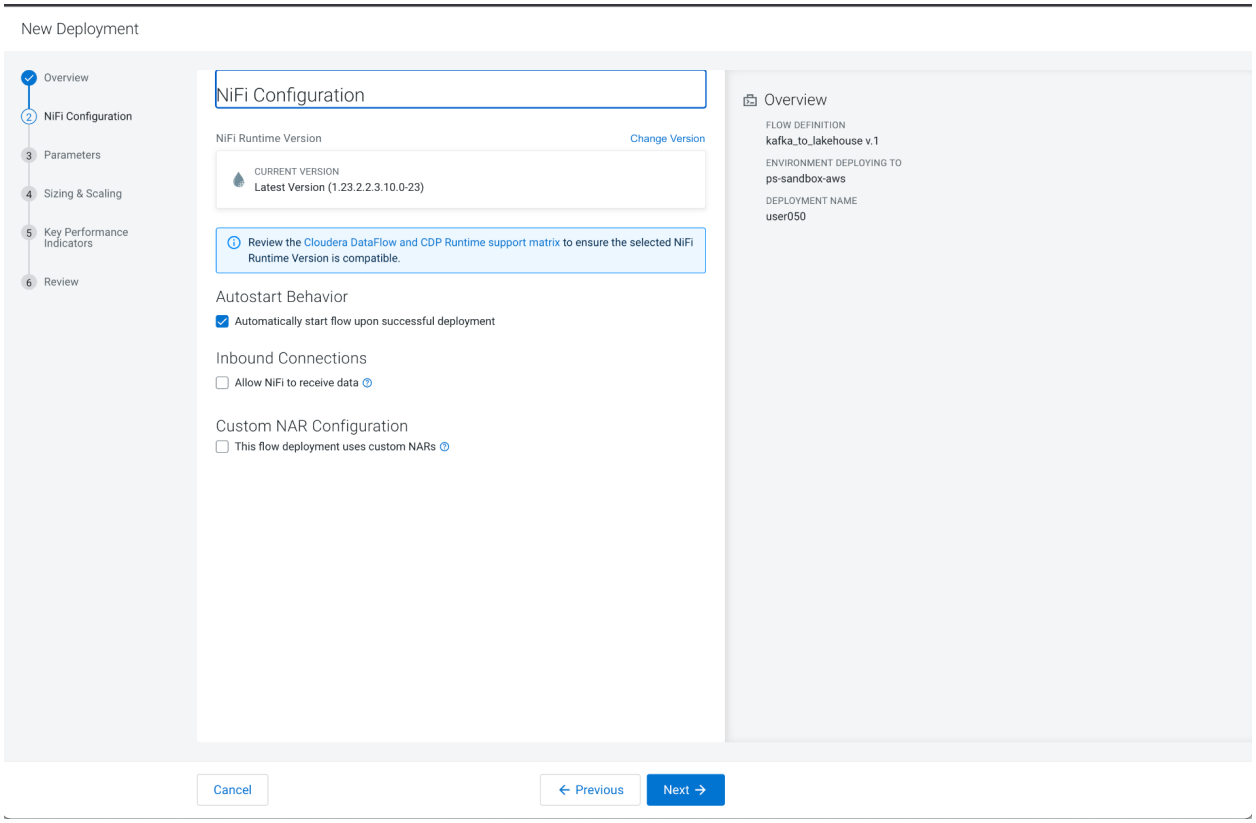


5. From this point, you will need to enter the Flow configuration. Start by assigning a name (**Deployment Name**) and click **Next**.

*For the purposes of this workshop, please name the Flow with the assigned username -**user050**, for example.*



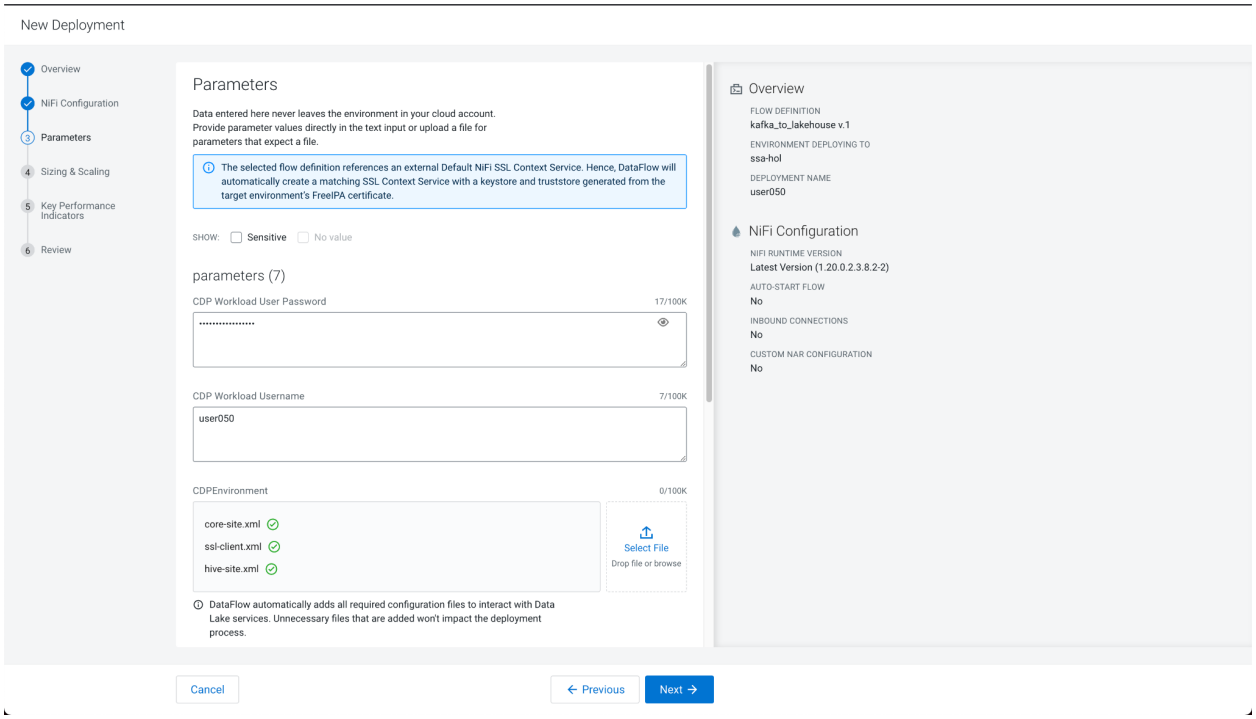
6. Make sure the option **Automatically start flow upon successful deployment** is checked and click **Next.**

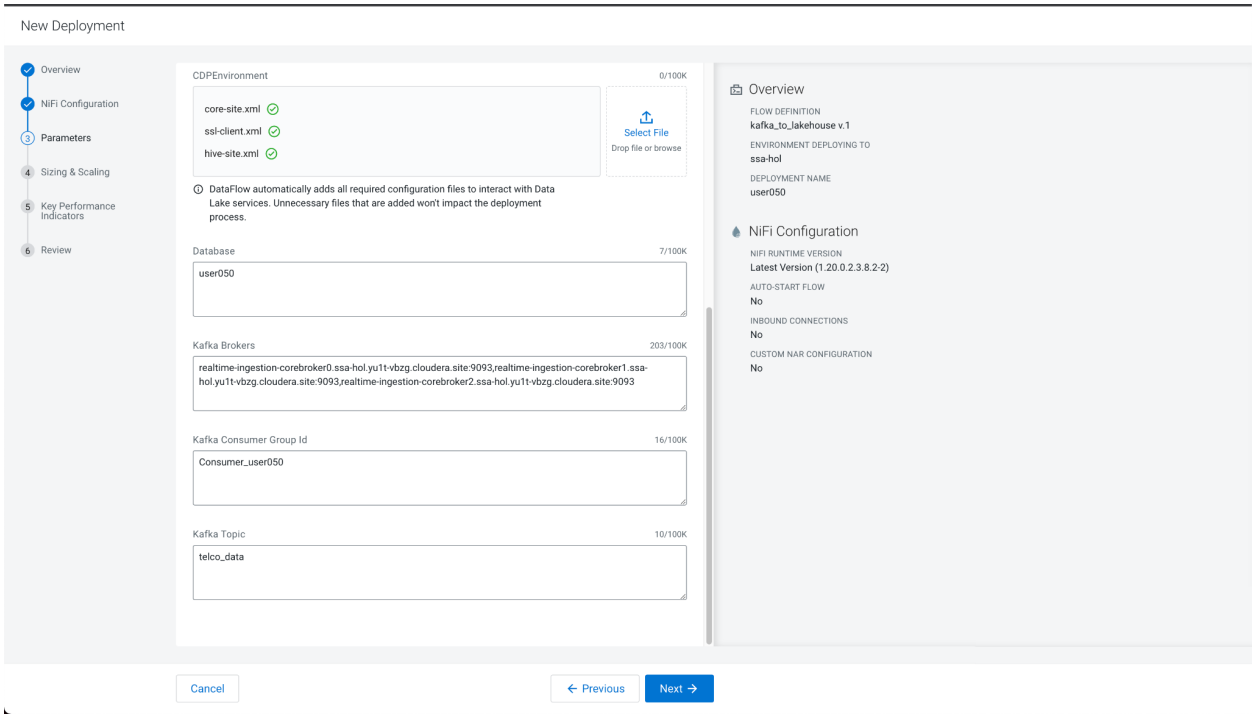


7. In this part of Parameters, you must enter the following values:

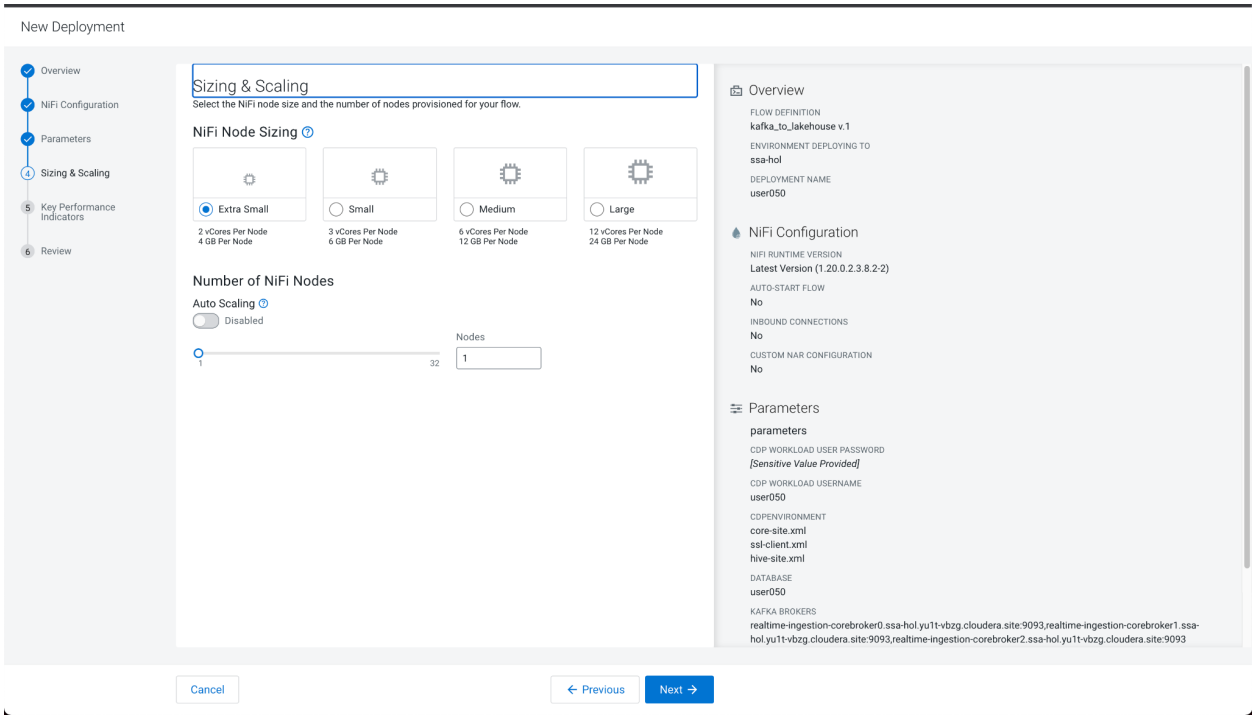
CDP Workload User Password: Enter the Workload Password shared at the beginning of the workshop.
CDP Workload Username: enter the assigned user number, *user050*, for example.
Database: enter the assigned user number, *user050*, for example. This database and the tables are already pre-created for you. We'll review it later.
Kafka Consumer Group Id: Enter a unique value using the assigned user. You can combine with the user id assigned for you.

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

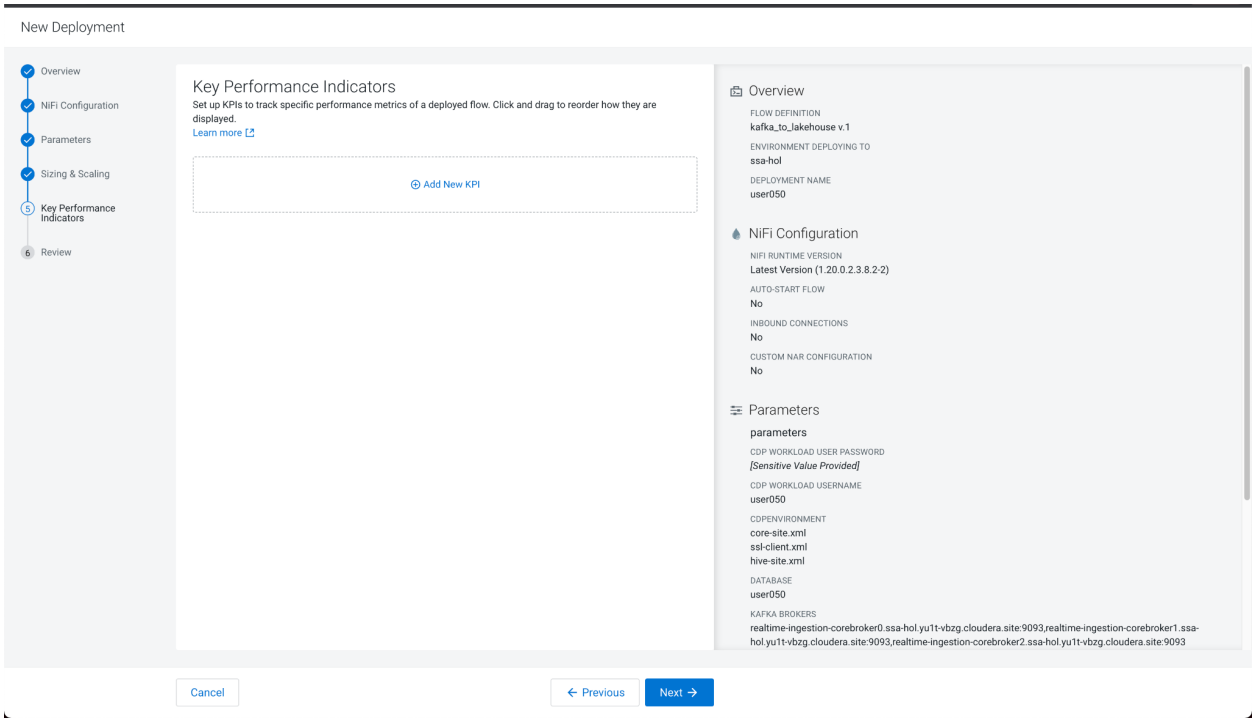




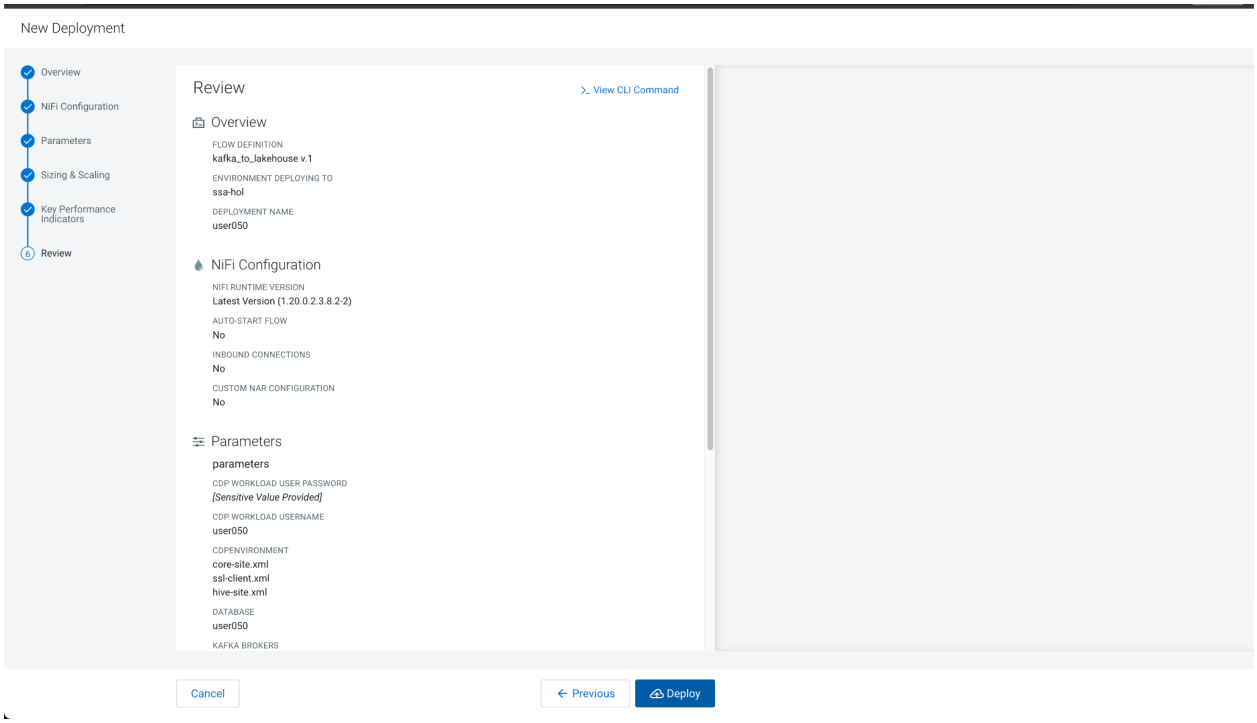
8. There is no need to configure auto scaling parameters, then click on **Next.**



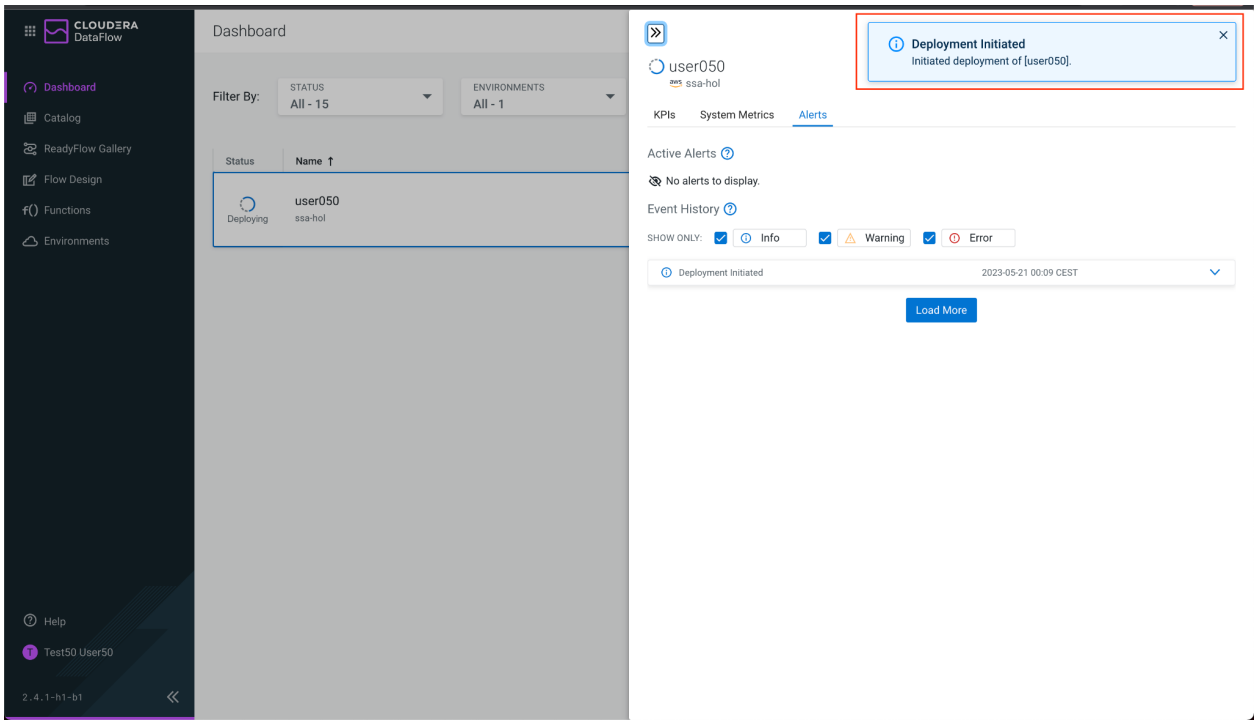
9. We are also not going to configure KPIs by now, then click on **Next** to continue the configuration.



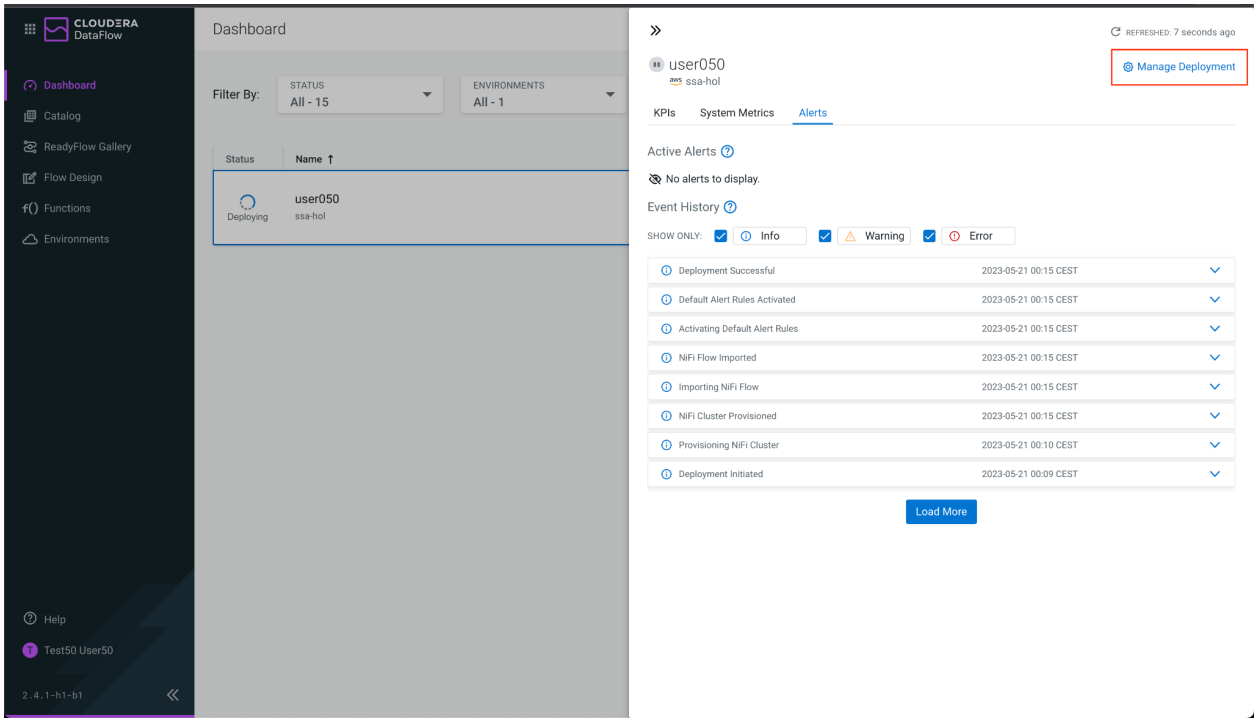
10. Review all the information entered for your Flow, then click on **Deploy** to start the deployment process.



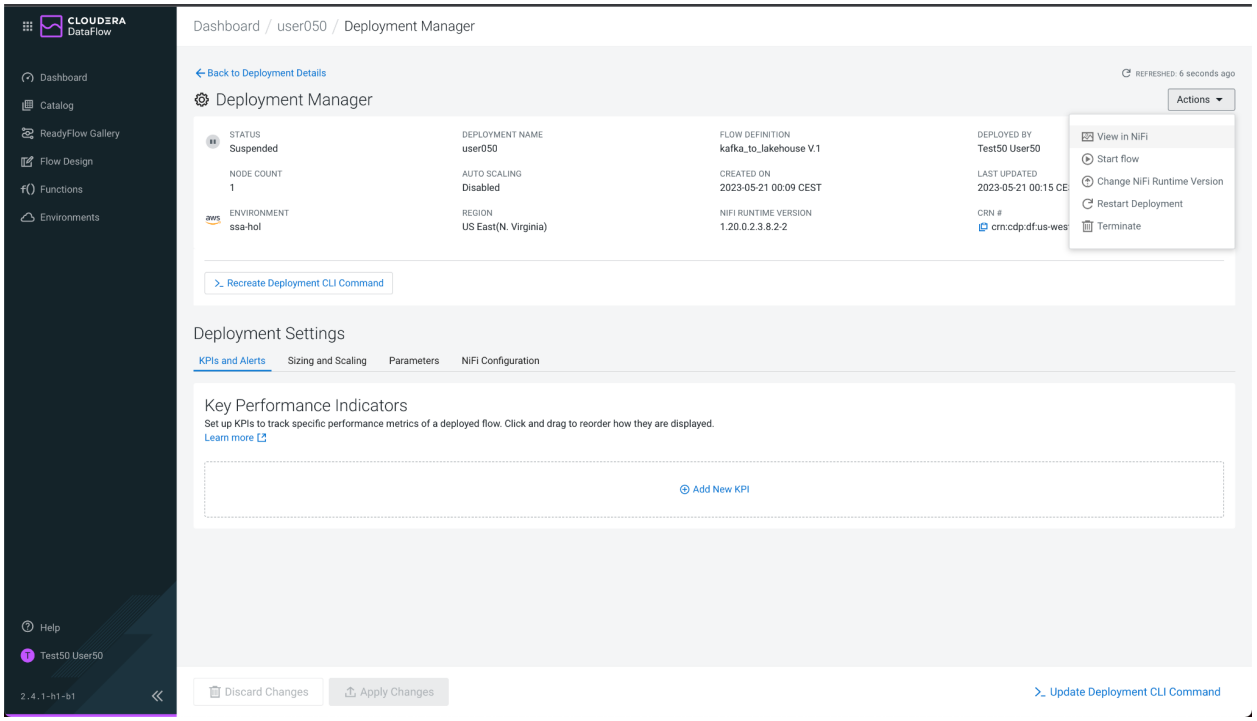
11. The blue box indicates that the Flow deployment process has been started. By clicking on the button **Load More** you will be able to see the different stages of the deployment. After about 60 to 90 seconds approximately, the last event should be *Deployment Successful*.



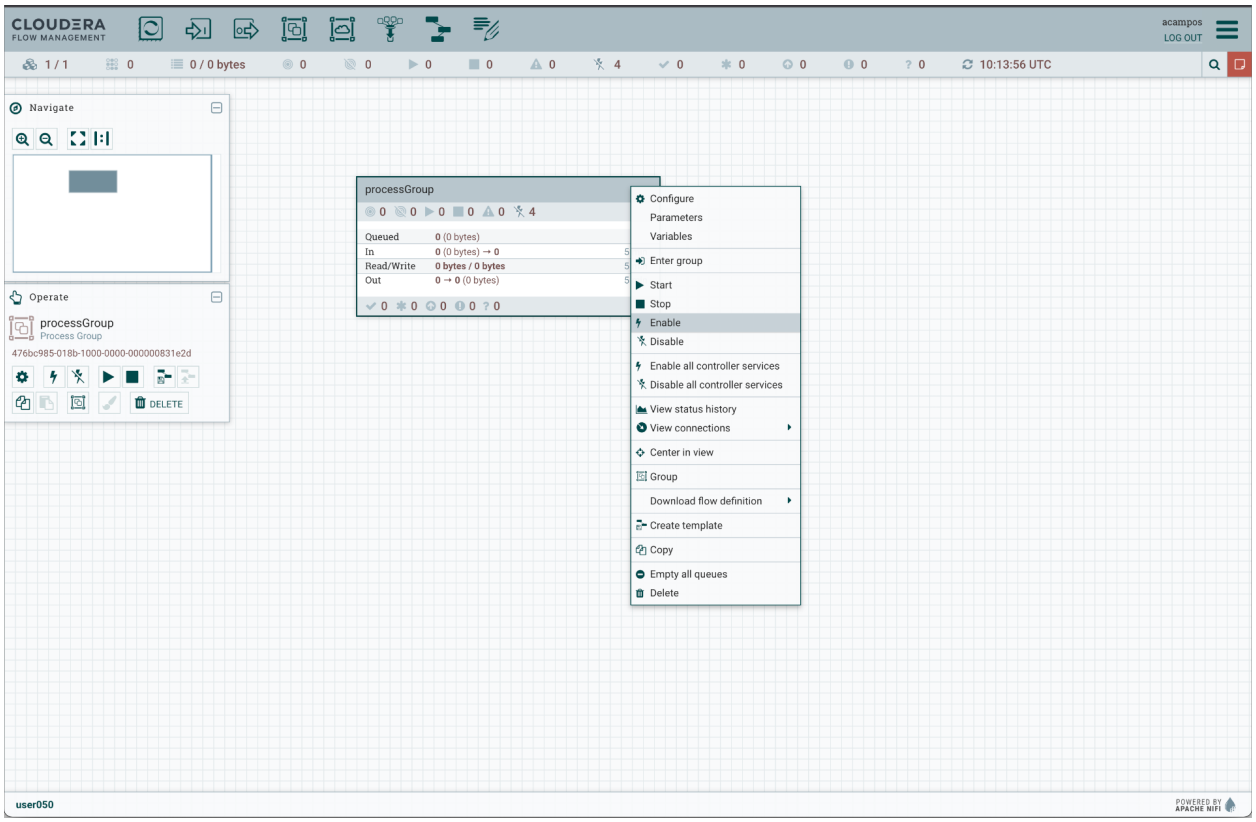
12. Once the deployment is finished, click on **Manage Deployment** to see the details of the recently deployed Flow.



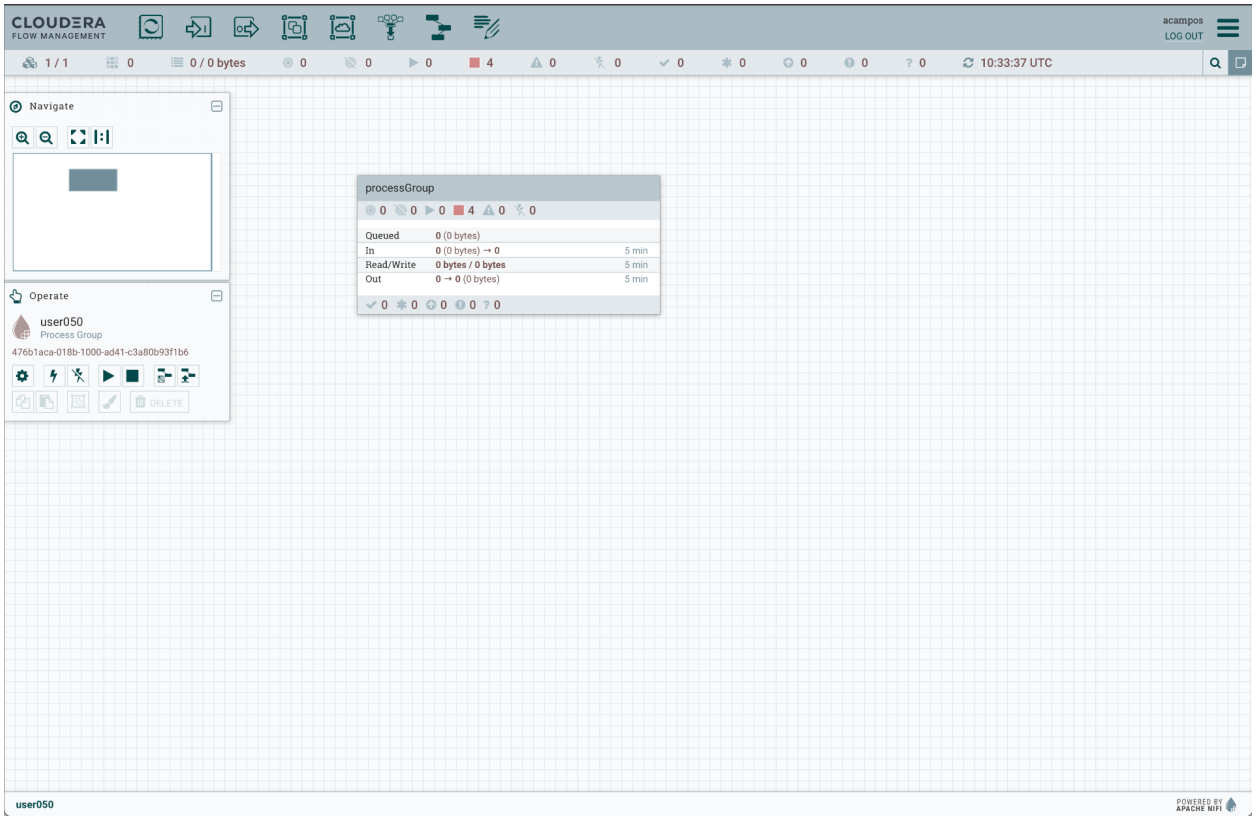
13. In this window you will see the Flow information displayed. It is time to execute the application processes from the graphical Flow Management interface. Click on **Actions** -> **View in NiFi**, to open Cloudera Flow Management canvas in a new window/tab.



14. In the new window you should be able to see the Flow Management canvas with one process group (a box) titled **processGroup**. You first need to enable the process group. Right-click on the Process Group and then select the option **Enable**.



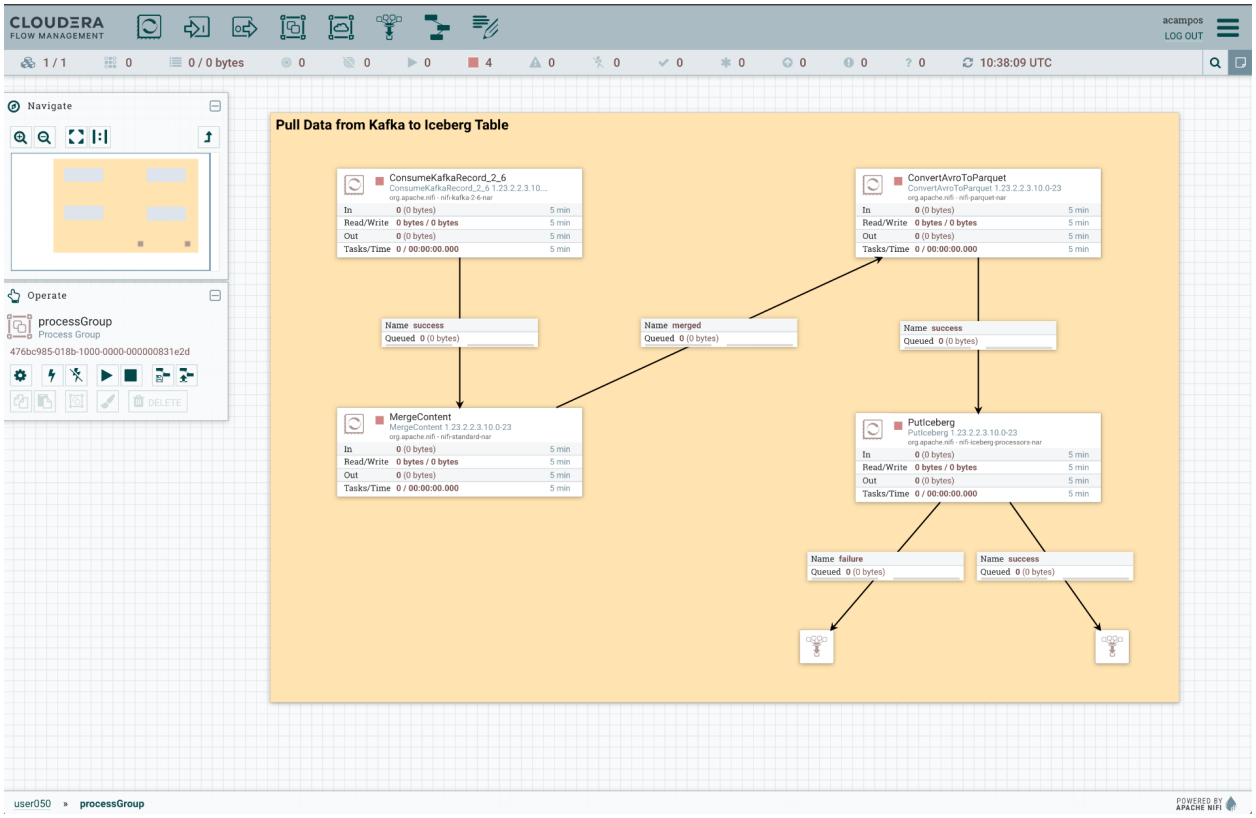
15. Then double click on the Process Group to open it.



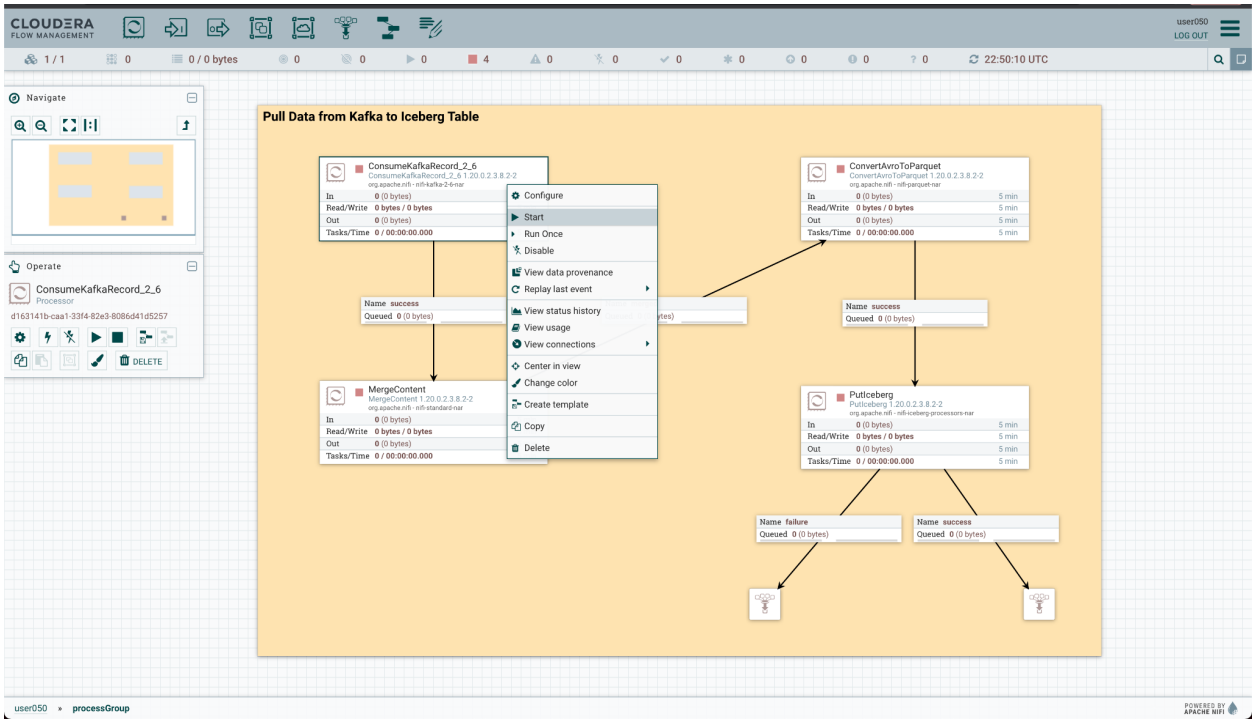
16. When opening the Process Group, you should be able to see the Processors that compose the Flow application. To summarize, there are four Processors:

ConsumeKakfaRecord, processor to consume data from the Kafka topic, reading the data in JSON format and outputting in AVRO format.
MergeContent, 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.

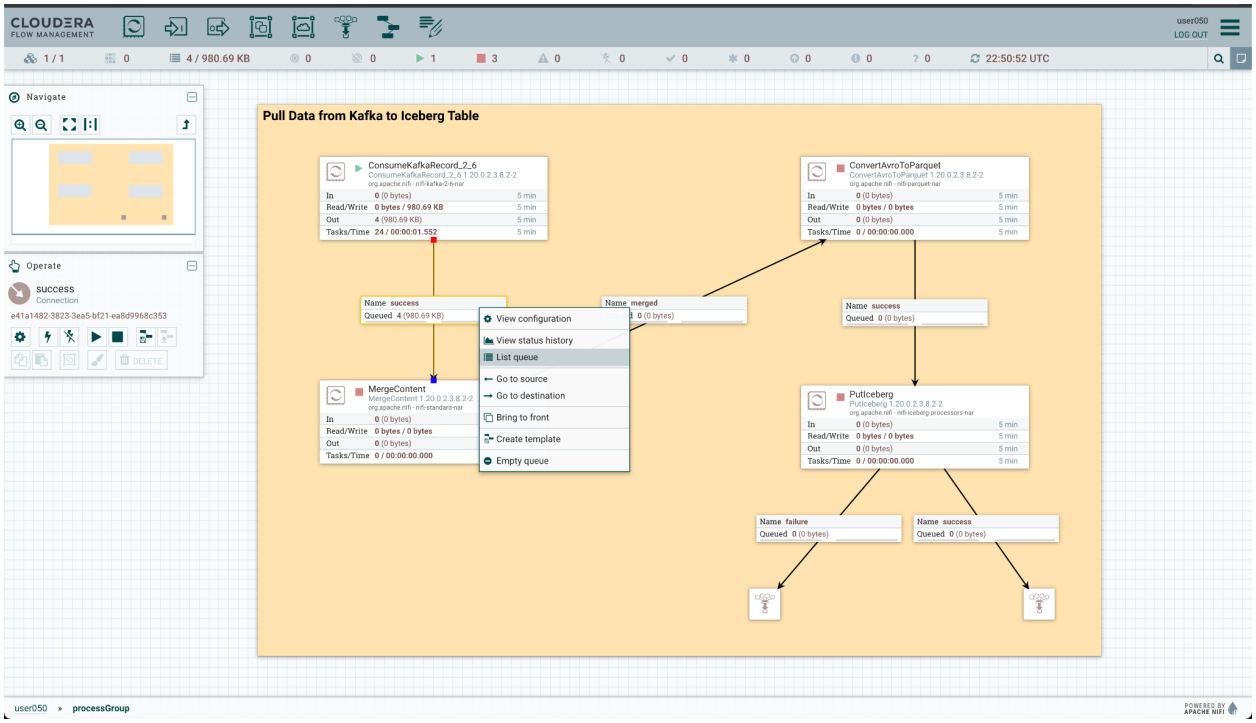


17. Now you are ready to initiate the pipeline. Start with **ConsumeKafkaRecord**, by right-clicking on it, and then clicking on **Start**. This will start consuming the Kafka topic data.

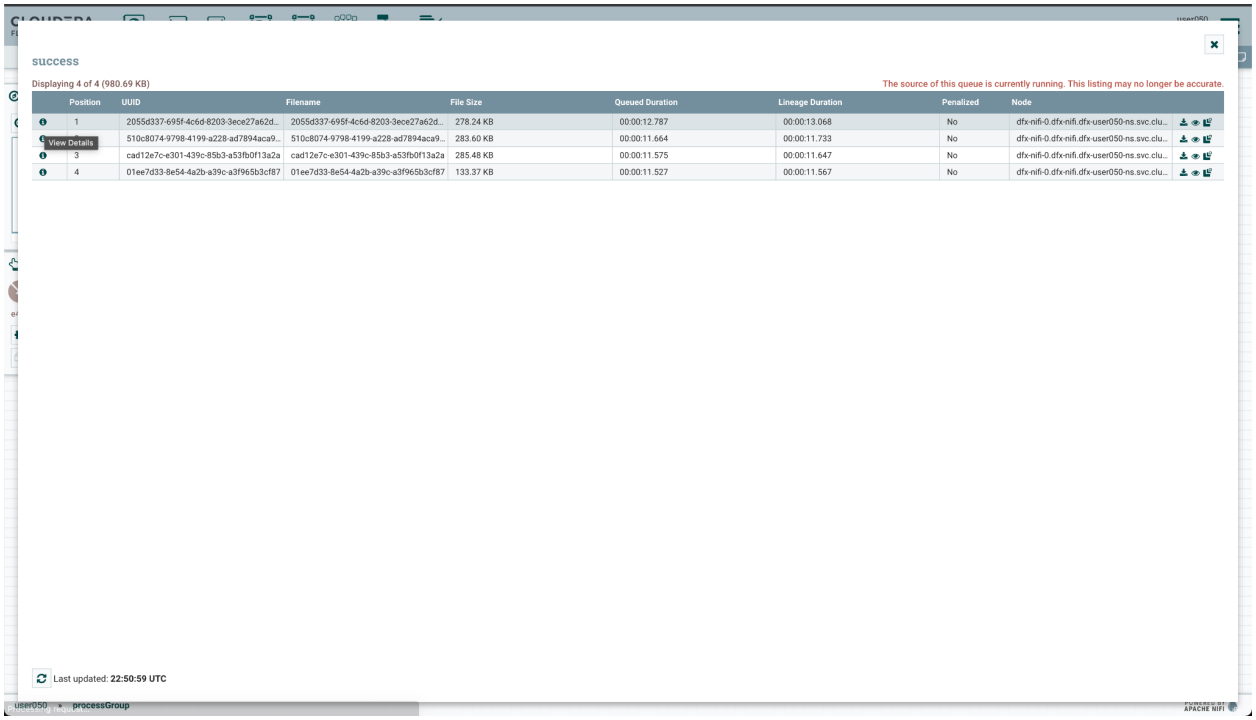


18. Flow Management allows us to see and access data in motion during the execution of the data flow. Between Processors **ConsumeKafkaRecord** (just started) and **MergeContent**, there is a connection. This connection is what joins the Processors and transmits data from one to the other.

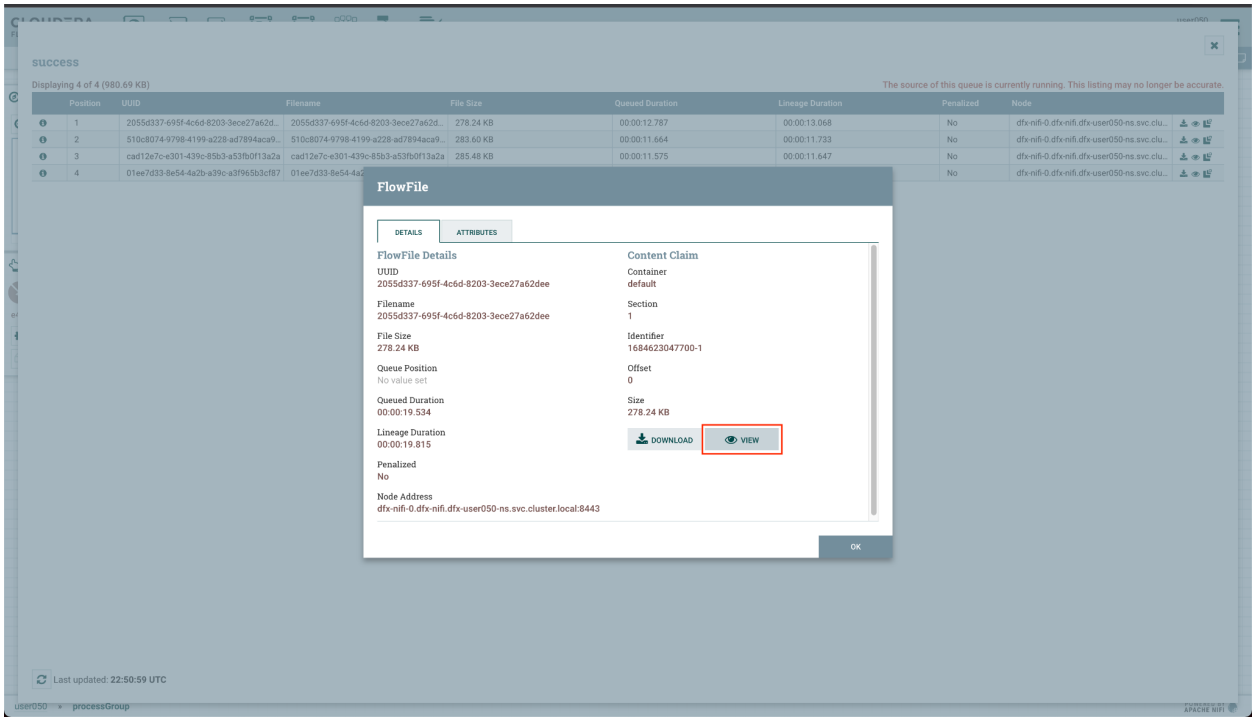
To check how much data is queued on this connection, refresh the counter by pressing the Ctrl+R (Windows) or Command+R (Mac) combination on the keyboard. This will allow the current metrics of the entire data stream to be updated. At some point there should be a number next to the legend **Queued** in the connection between **ConsumeKafkaRecord** and **MergeContent.** To see the queued data, right click on the connection and click on the option **List Queue**, opening a popup window.



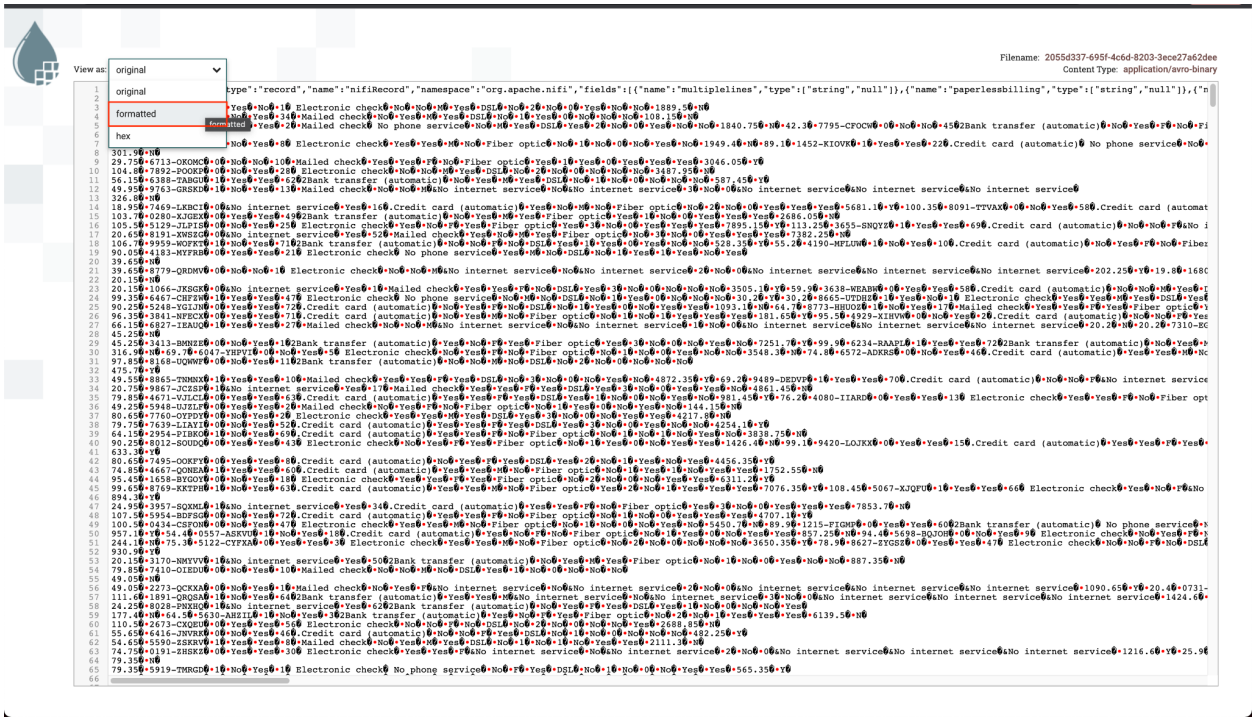
19. The next popup window lists the queued data. Click on the information icon (i) that appears on the left side to view the events.



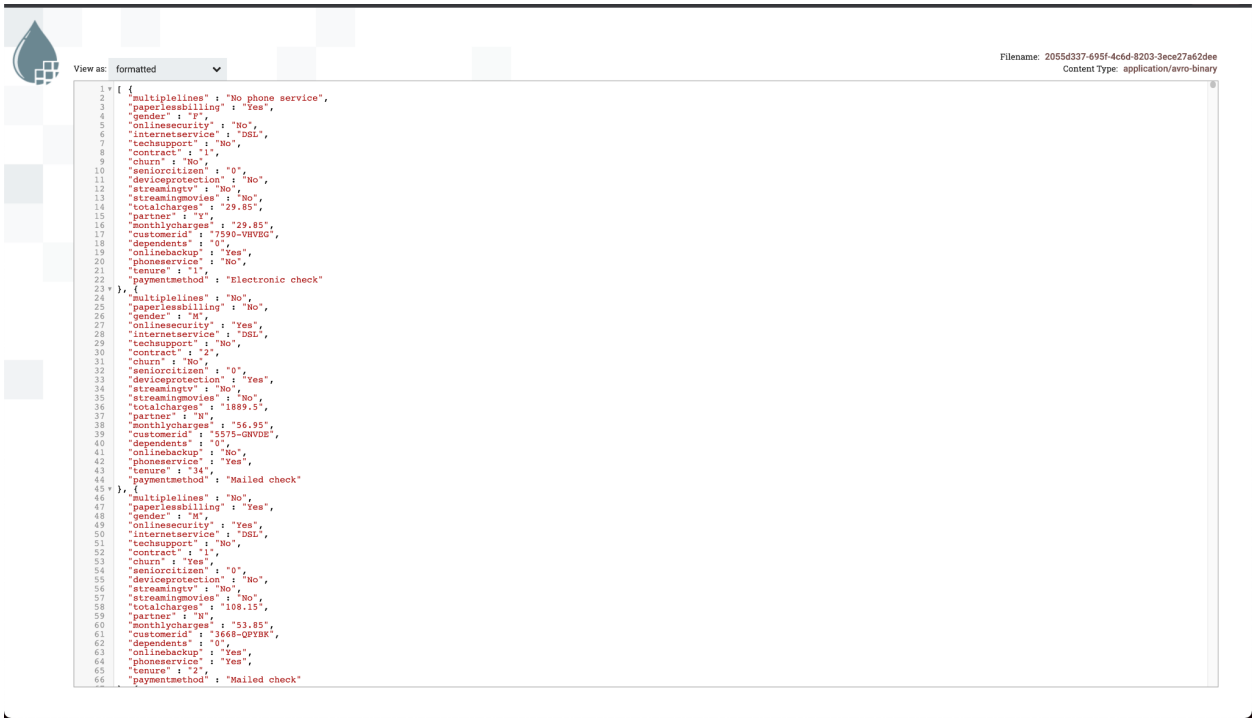
20. Once the FlowFile detail window appears, click on the button **VIEW** to open the content of consumed events.



21. 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**.

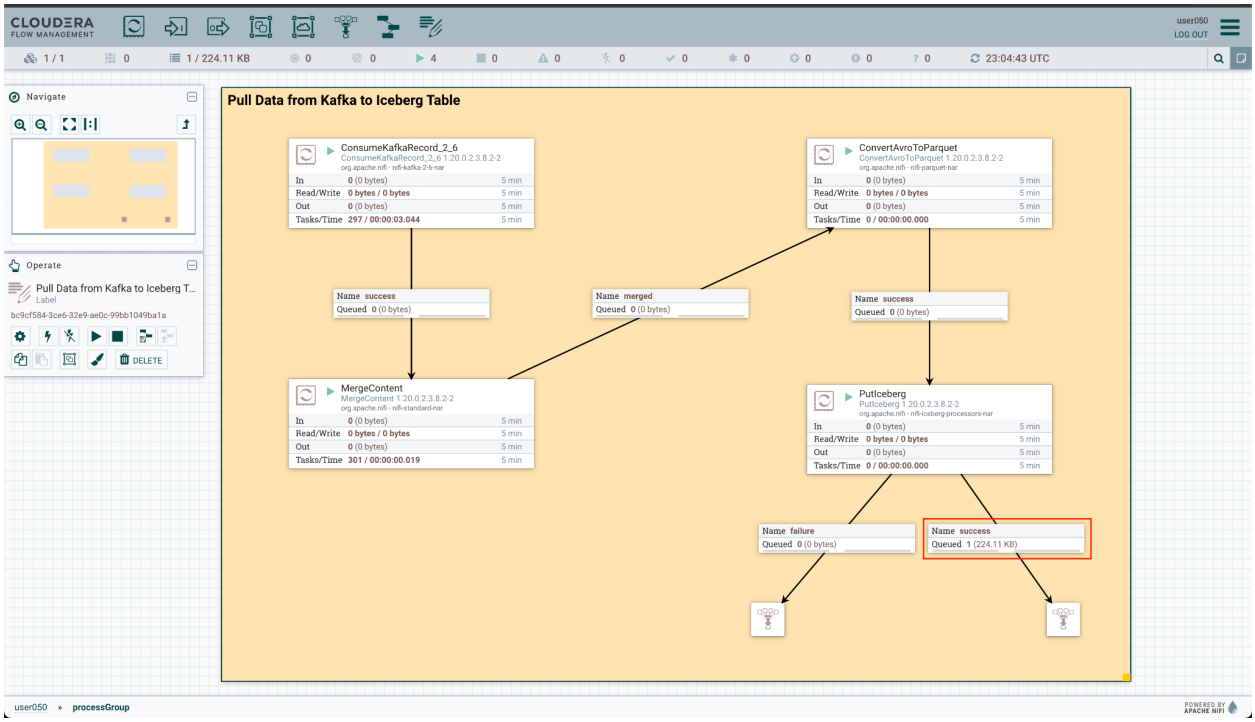


22. Now you can display the data correctly. Notice that the fields or attributes indicated at the beginning of the workshop appear. You can close that FlowFile window and the popups, returning to the canvas with the four Processors.



23. Continue running each of the Processors in order: **MergeContent**, after **ConvertAvroToParquet** and finally **PutIceberg**. Remember that you can refresh the flow counters with the combination Control+R or Command+R.

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


CLOUDERA
FLOW MANAGEMENT
user050
LOG OUT
1 / 1
0
1 / 224.11 KB
0
0
4
0
0
0
0
0
0
0
0
23:04:43 UTC
Navigate
Operate
Pull Data from Kafka to Iceberg T...
Label
bc9cf584-3ce6-32e9-ae0c-99bb1049ba1a
DELETE
Pull Data from Kafka to Iceberg Table
ConsumeKafkaRecord_2_6
ConsumeKafkaRecord_2_6 1.20.0.2.3.8.2-2
org.apache.nifi - nifi-kafka-2-6-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 297 / 00:00:03.044 5 min
ConvertAvroToParquet
ConvertAvroToParquet 1.20.0.2.3.8.2-2
org.apache.nifi - nifi-parquet-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min
Name success
Queued 0 (0 bytes)
Name merged
Queued 0 (0 bytes)
Name success
Queued 0 (0 bytes)
MergeContent
MergeContent 1.20.0.2.3.8.2-2
org.apache.nifi - nifi-standard-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 301 / 00:00:00.019 5 min
PutIceberg
PutIceberg 1.20.0.2.3.8.2-2
org.apache.nifi - nifi-iceberg-processors-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min
Name failure
Queued 0 (0 bytes)
Name success
Queued 1 (224.11 KB)
user050 » processGroup
POWERED BY
APACHE NIFI