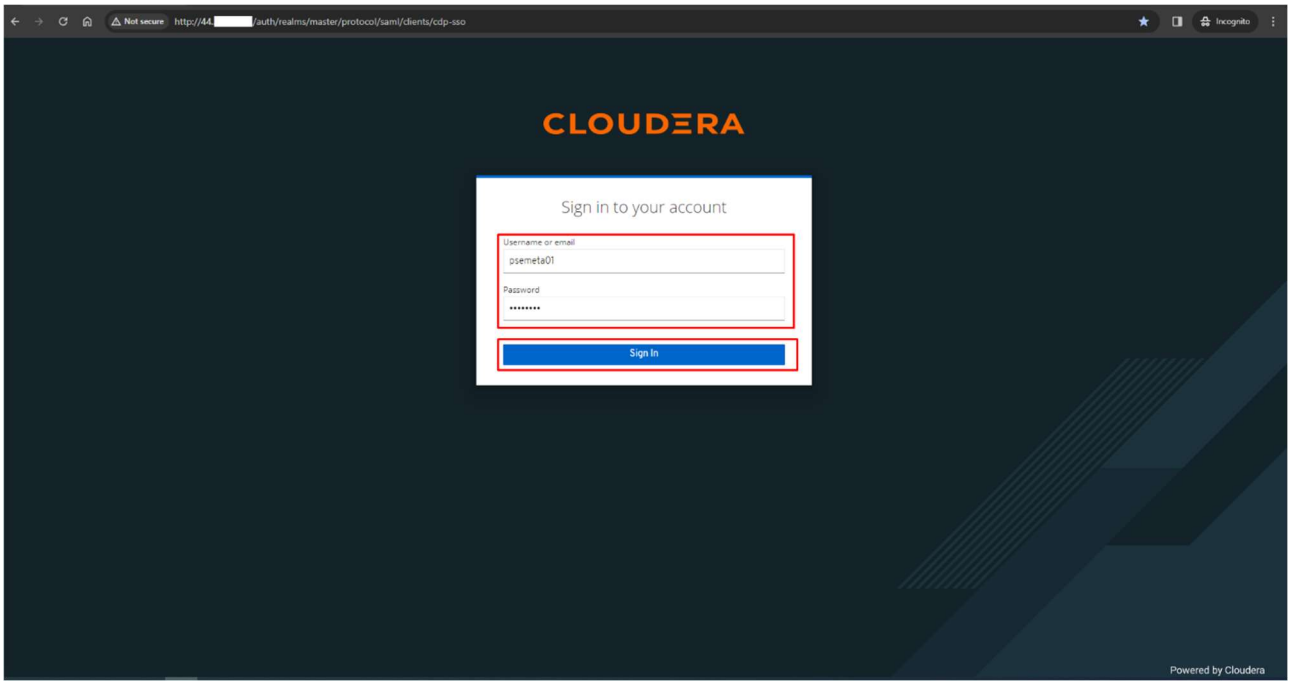
Data Lifecycle CDP Public Cloud

Data Flow Lab

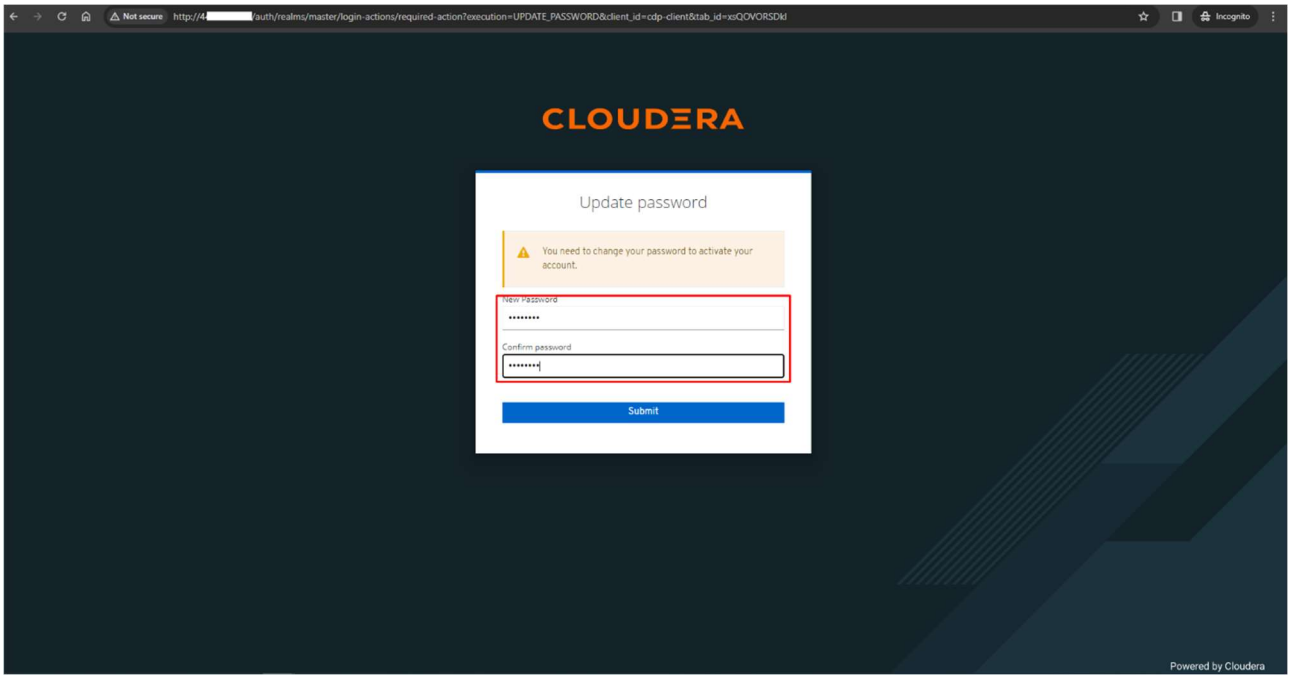
Goals:

- 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.

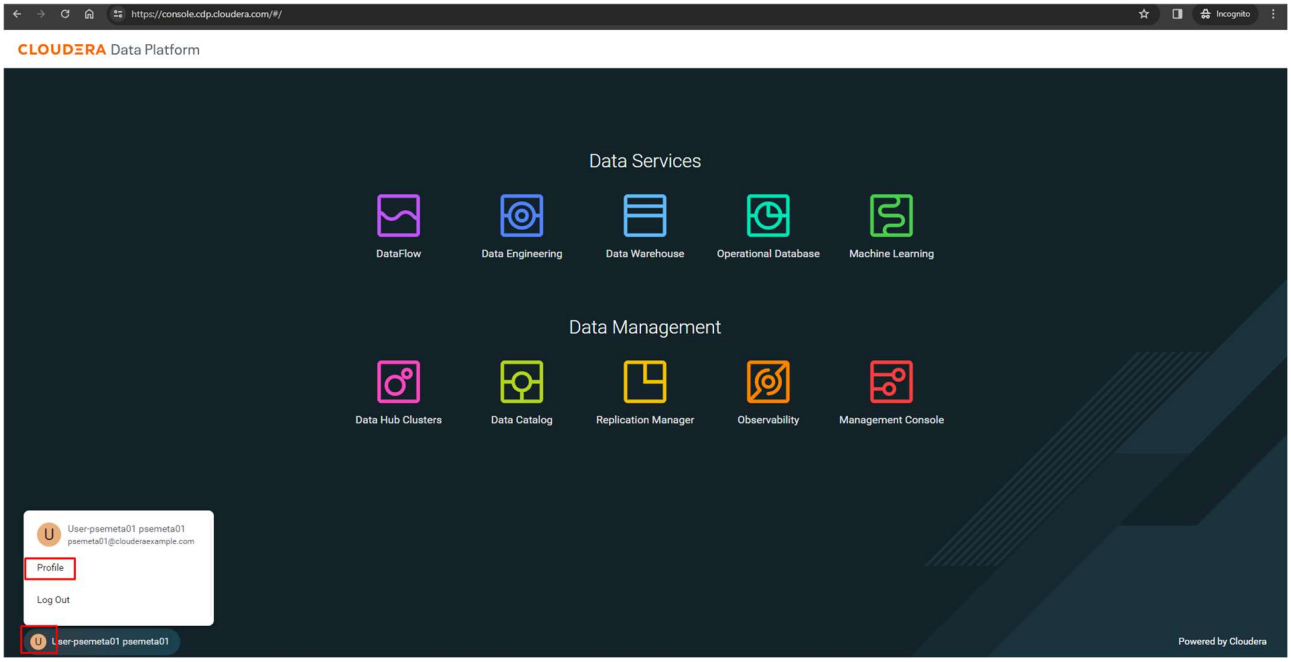


It might ask to change the password. Keep the password same as earlier which is – **changeme**

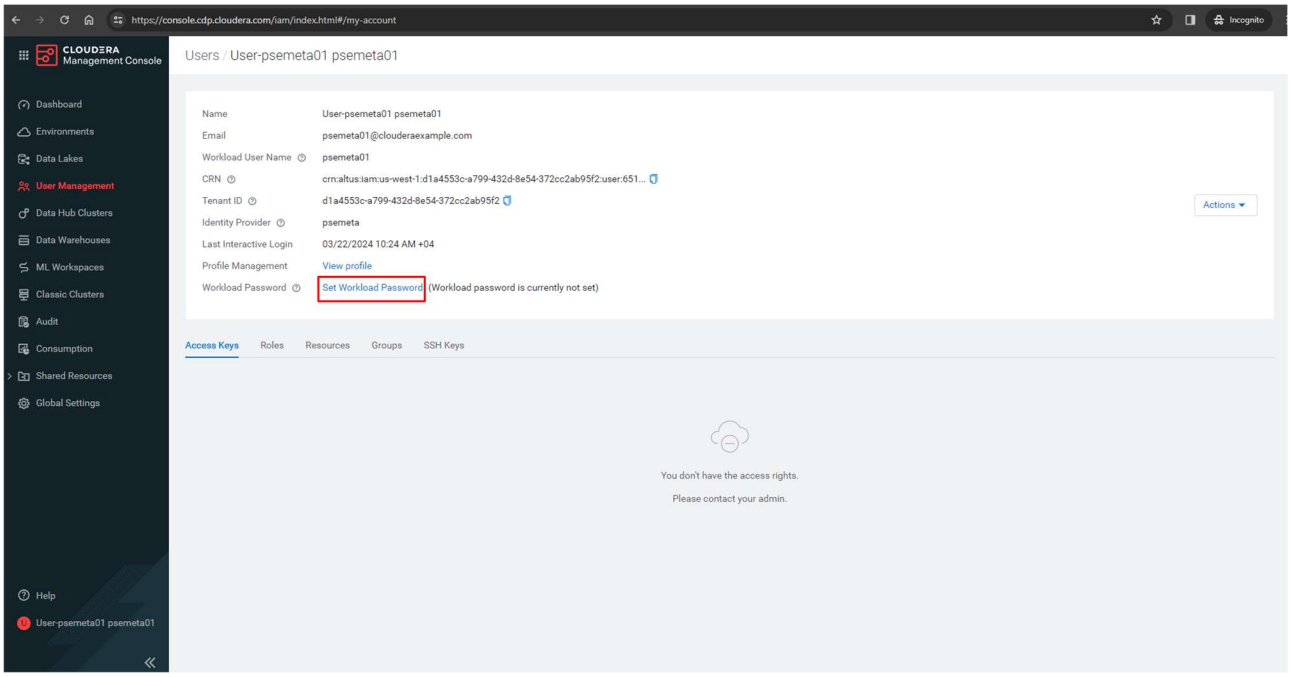


This is the CDP Console homepage.

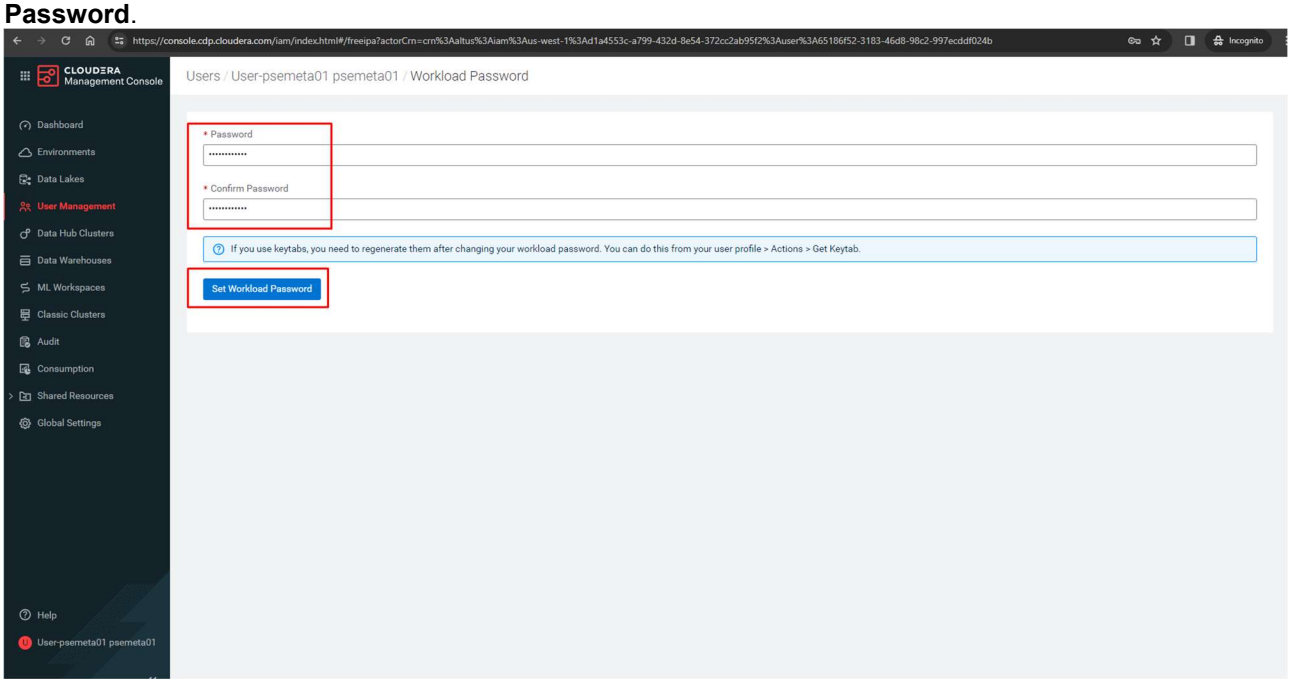
Now you will set a new workload password. Click on your login name at the lower bottom corner and then click on **Profile**.



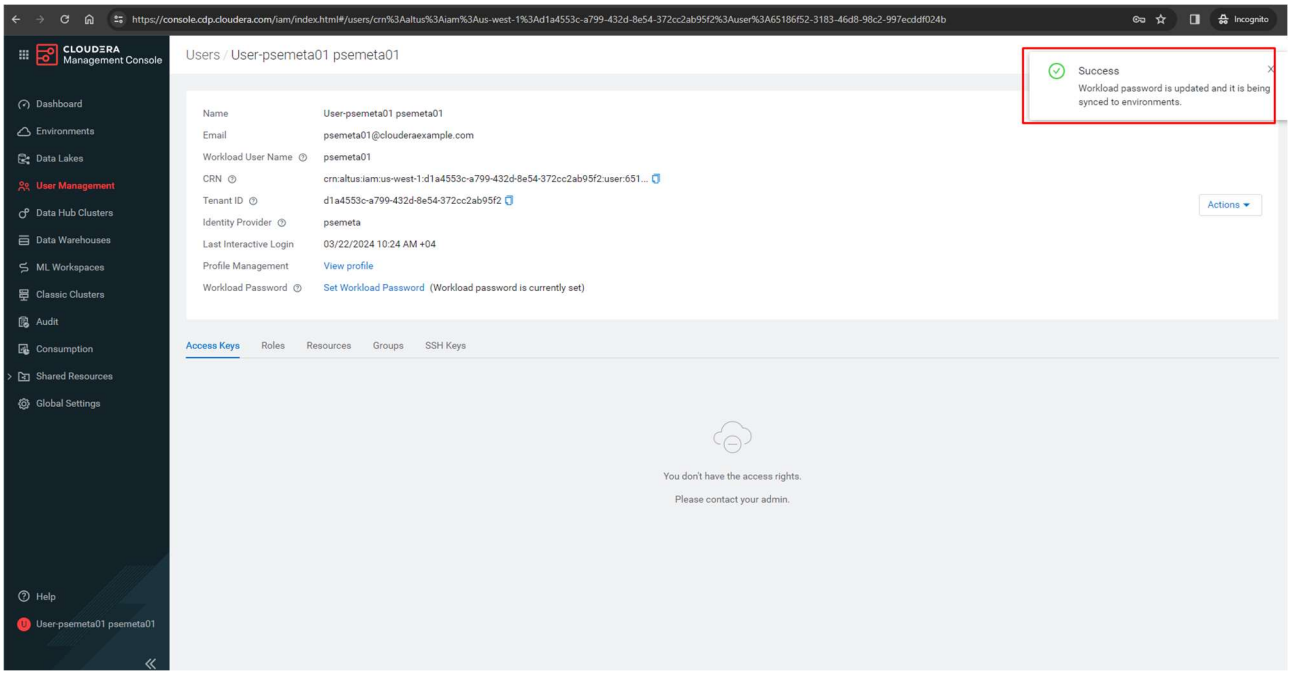
Click on **‘Set Workload Password’**.



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

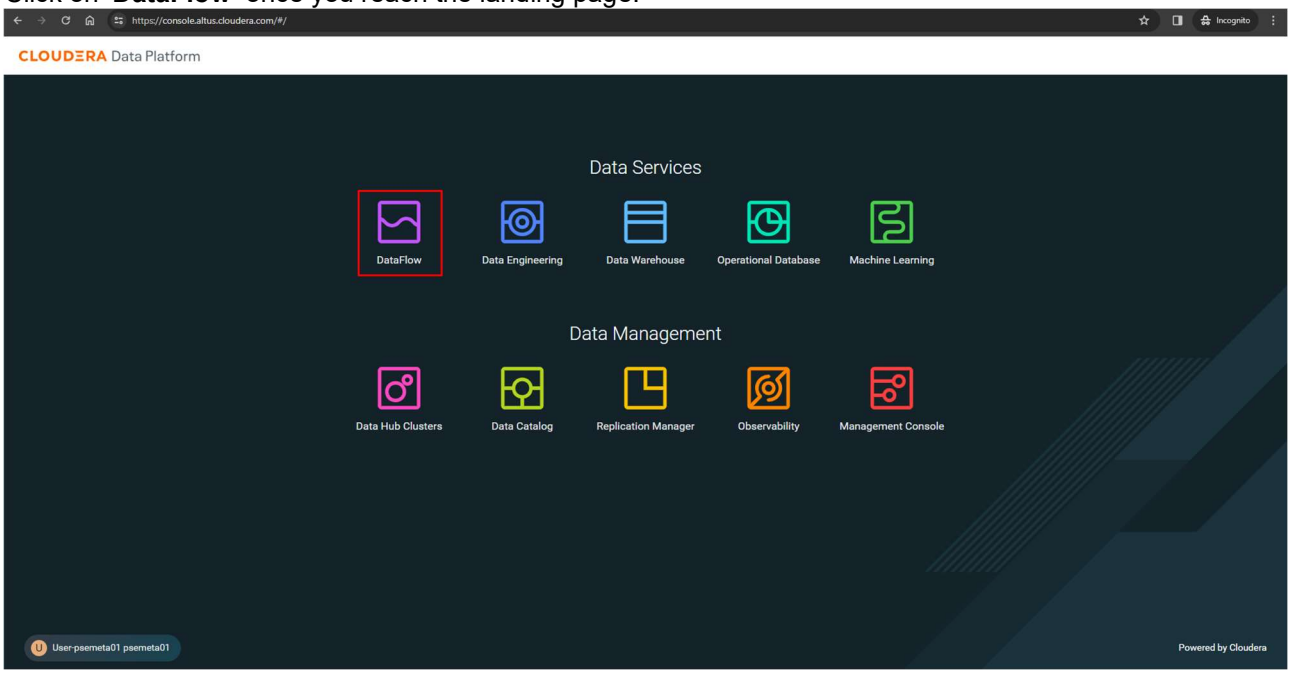


Password is set successfully.

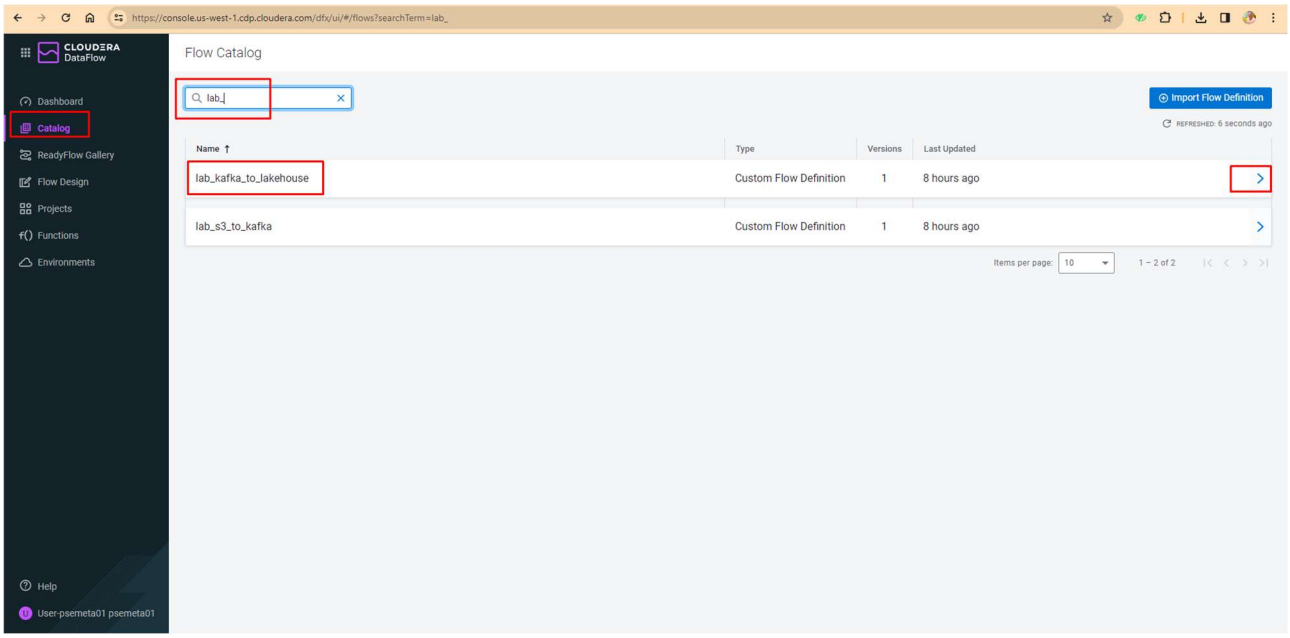


Now go back to the main page by clicking on the **Cloudera Management Console** on top left corner.

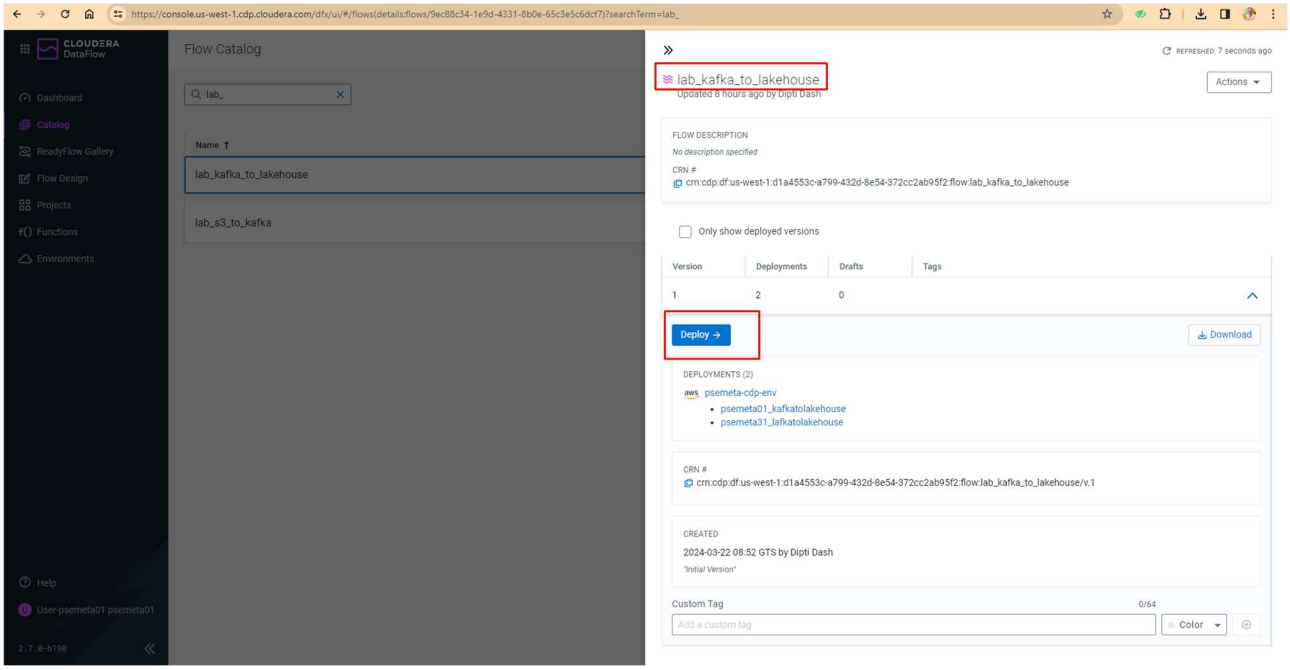
Click on '**DataFlow'** once you reach the landing page.



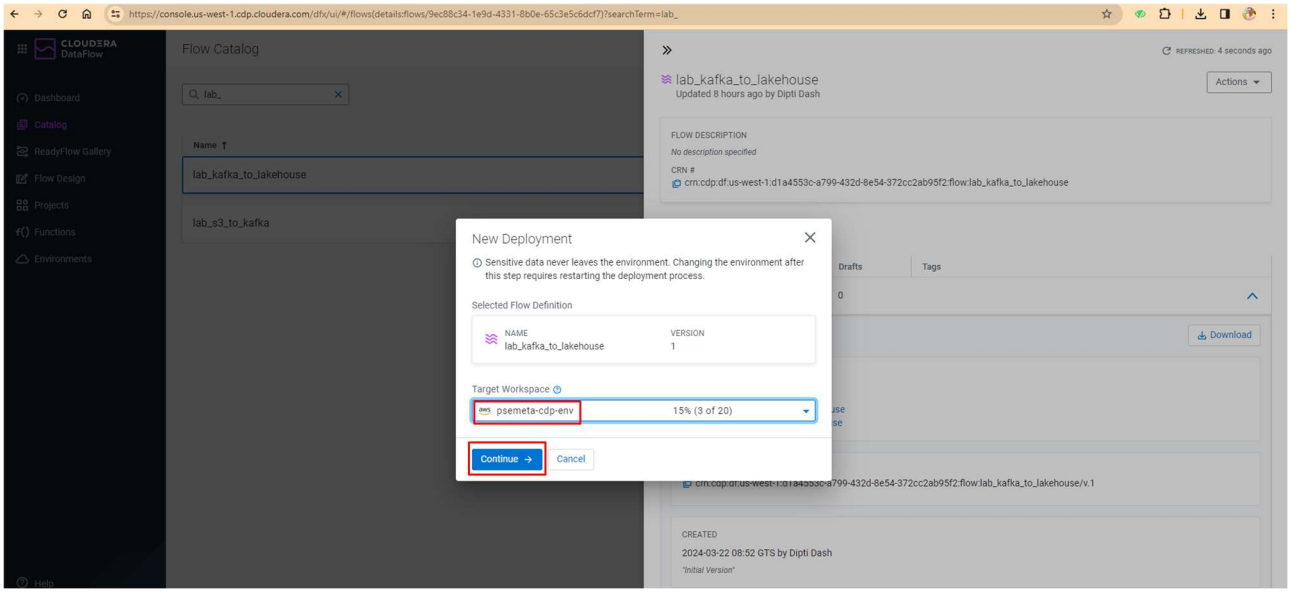
Once in DataFlow, click on the option **Catalog** from the left menu. The data ingestion application templates are listed here. For this workshop, we have created and published a template that allows you to read Kafka topic data and ingest/store it in the Lakehouse provided by CDP Public Cloud.
On the 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_
CLOUDERA
DataFlow
Dashboard
Catalog
ReadyFlow Gallery
Flow Design
Projects
Functions
Environments
Help
User-psemeta01 psemeta01
Flow Catalog
lab_
Import Flow Definition
REFRESHED: 6 seconds ago
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
Items per page:
10
1 – 2 of 2

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**.



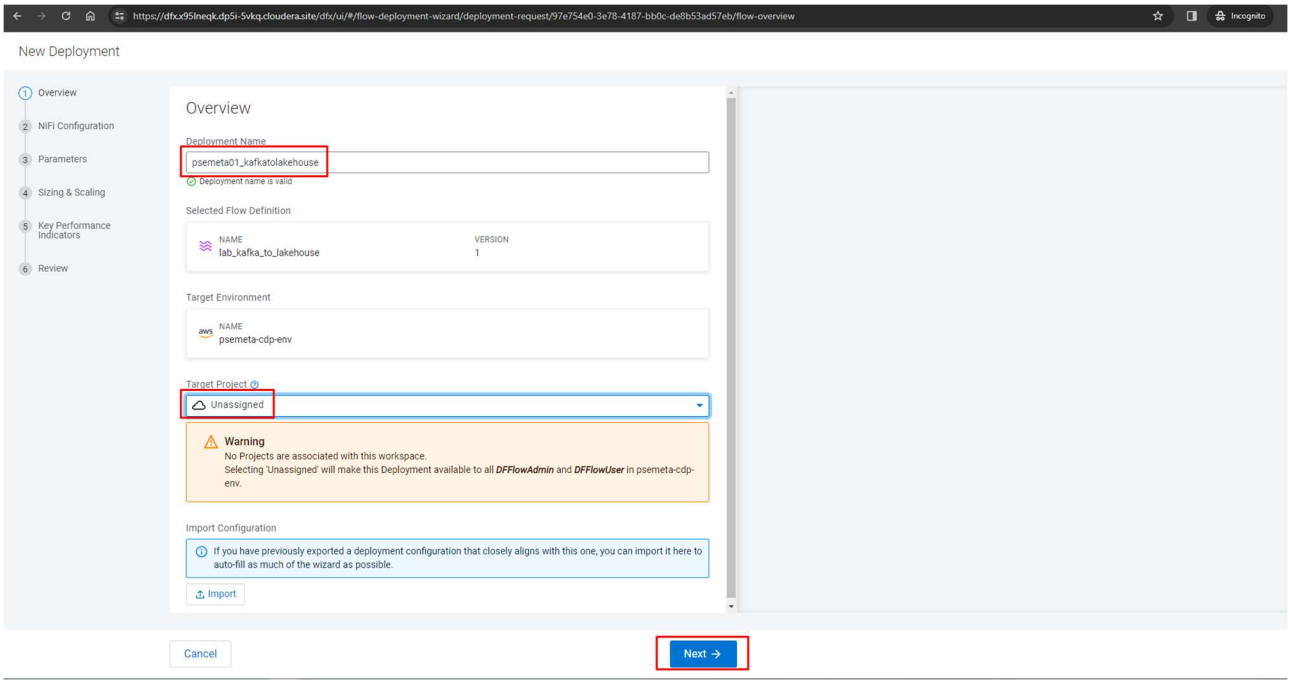
3. The following popup window allows you to select the DataFlow cluster in which you want to deploy the Flow. In this case, the cluster to be selected is **psemeta-cdp-env**. The workshop instructor will tell you which environment to select. Once selected, click **Continue**.



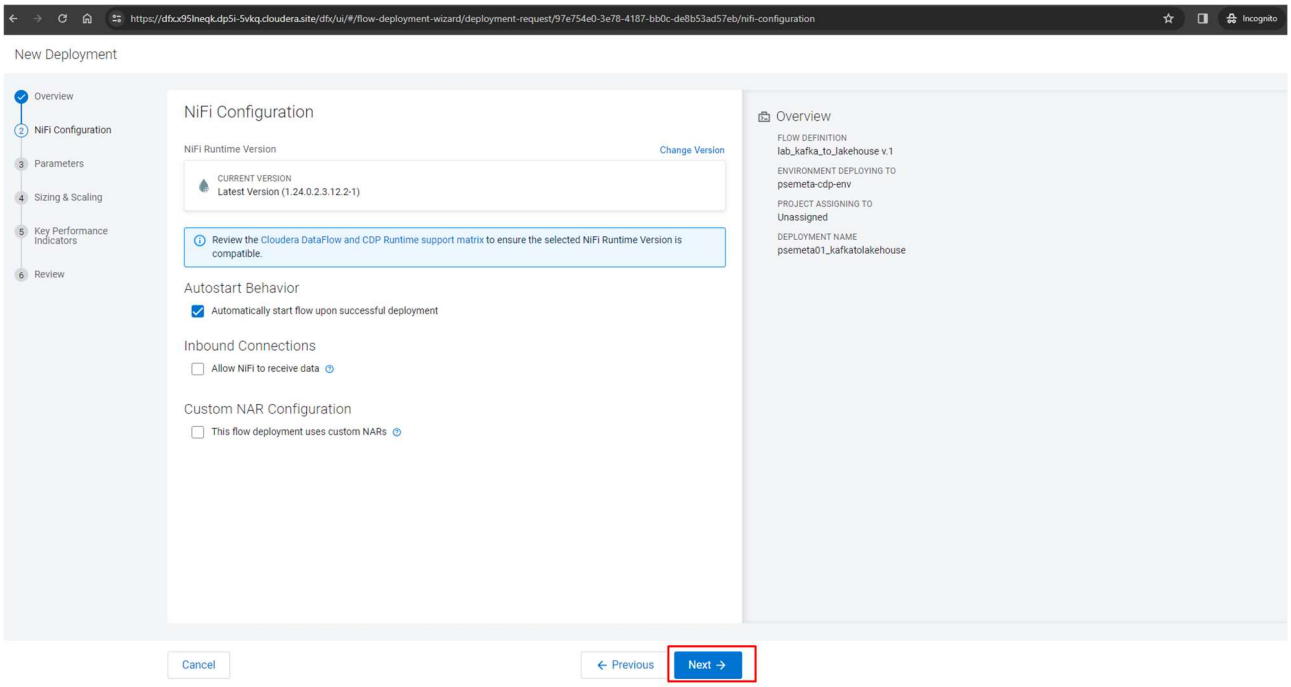
4. From this point, you will need to enter the Flow configuration. Start by assigning a Deployment Name, Target Project, and click Next.

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.**

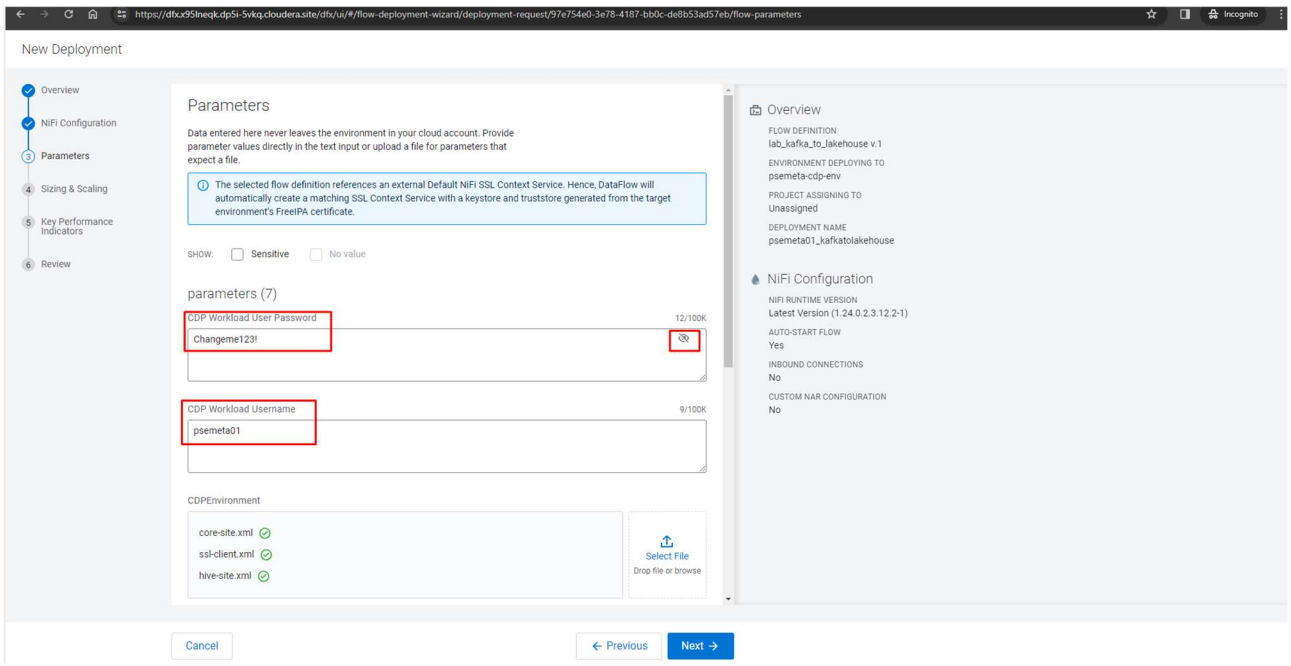


6. In this part of Parameters, you must enter the following values:

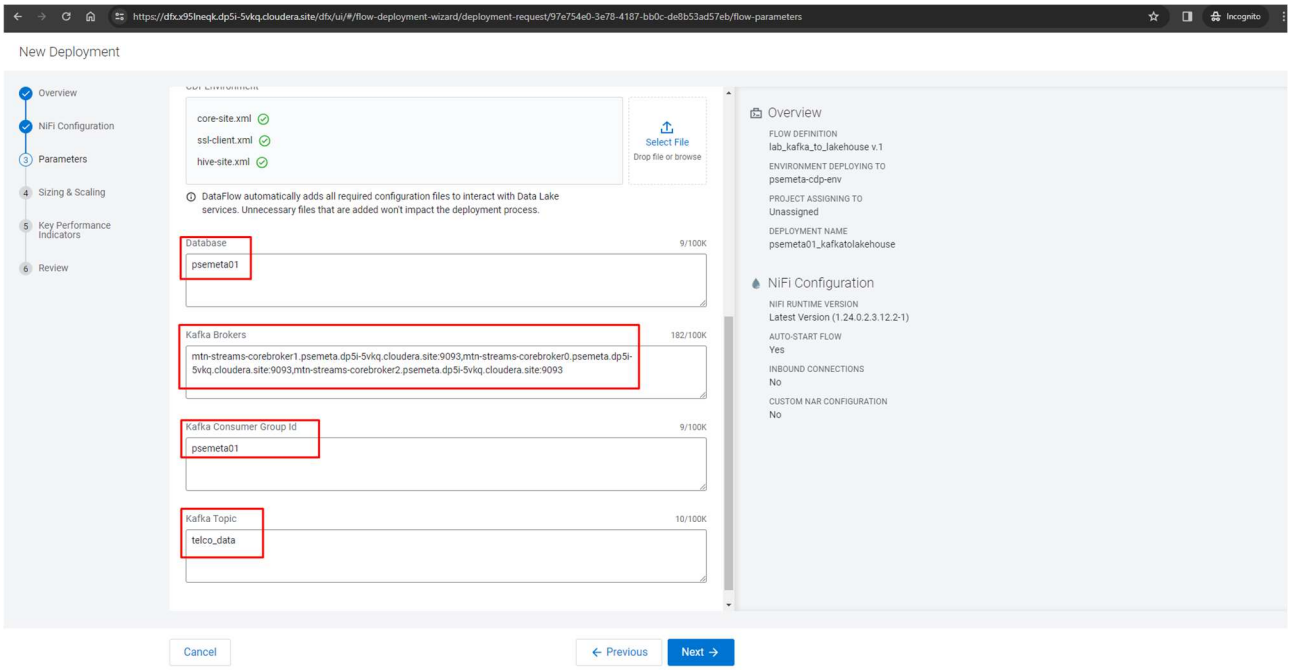
CDP Workload User Password: Enter the Workload Password that you had set at the beginning of this workshop. It was something like – **Changeme123!**
CDP Workload User: enter the assigned user number, ***psemeta01***, for example.

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

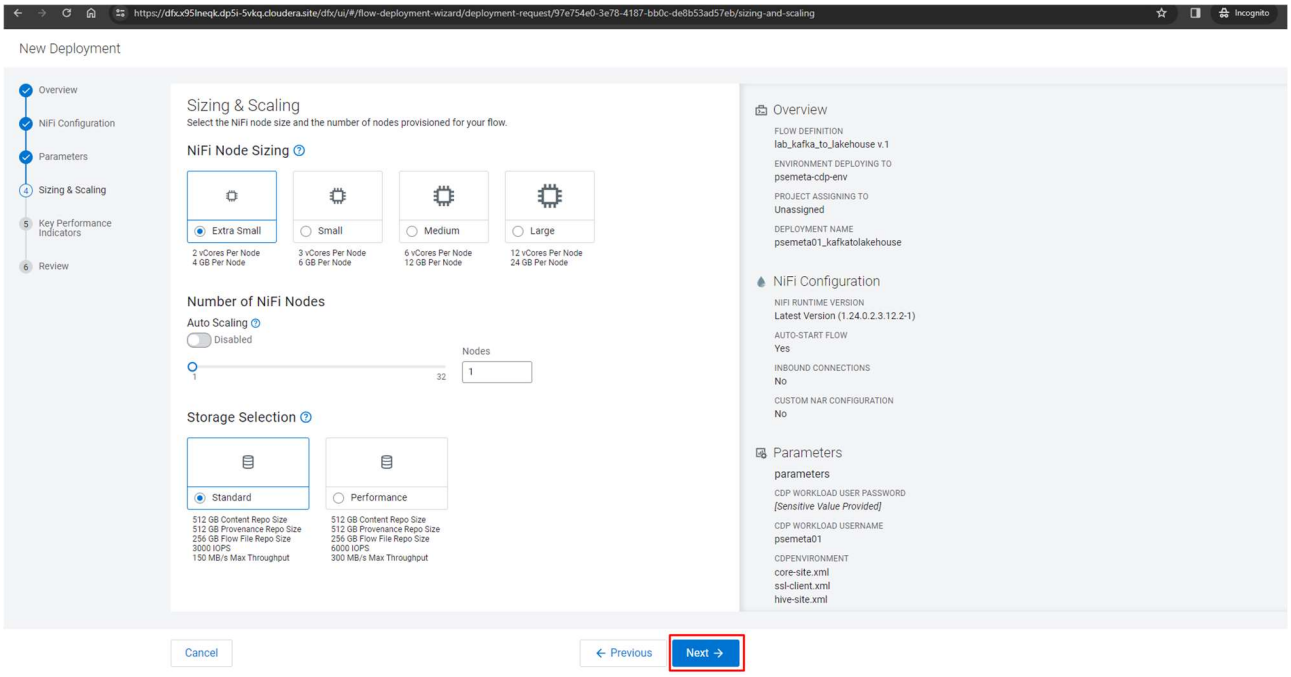


Kafka Brokers:
This value should be provided by the instructor.
mtn-streams-corebroker1.psemeta.dp5i-5vkq.cloudera.site:9093,mtn-streams-corebroker0.psemeta.dp5i-5vkq.cloudera.site:9093,mtn-streams-corebroker2.psemeta.dp5i-5vkq.cloudera.site:9093

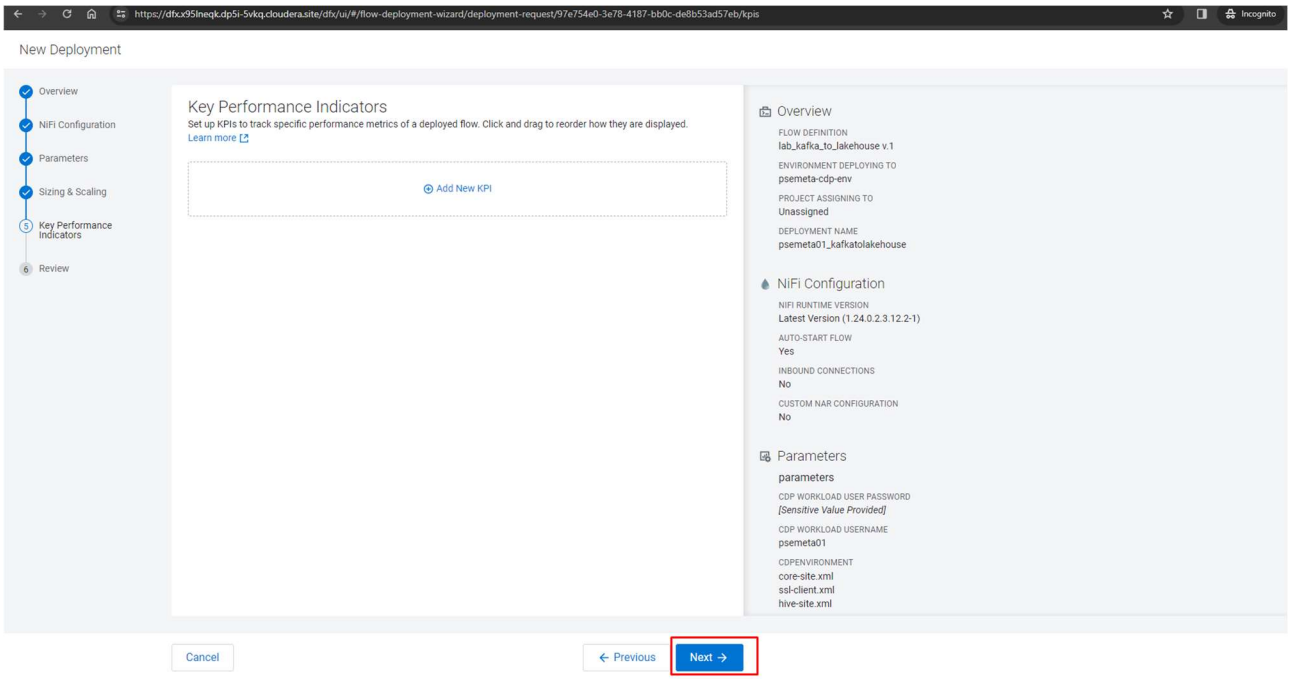


Review that the parameters were entered correctly. Then click **Next.**

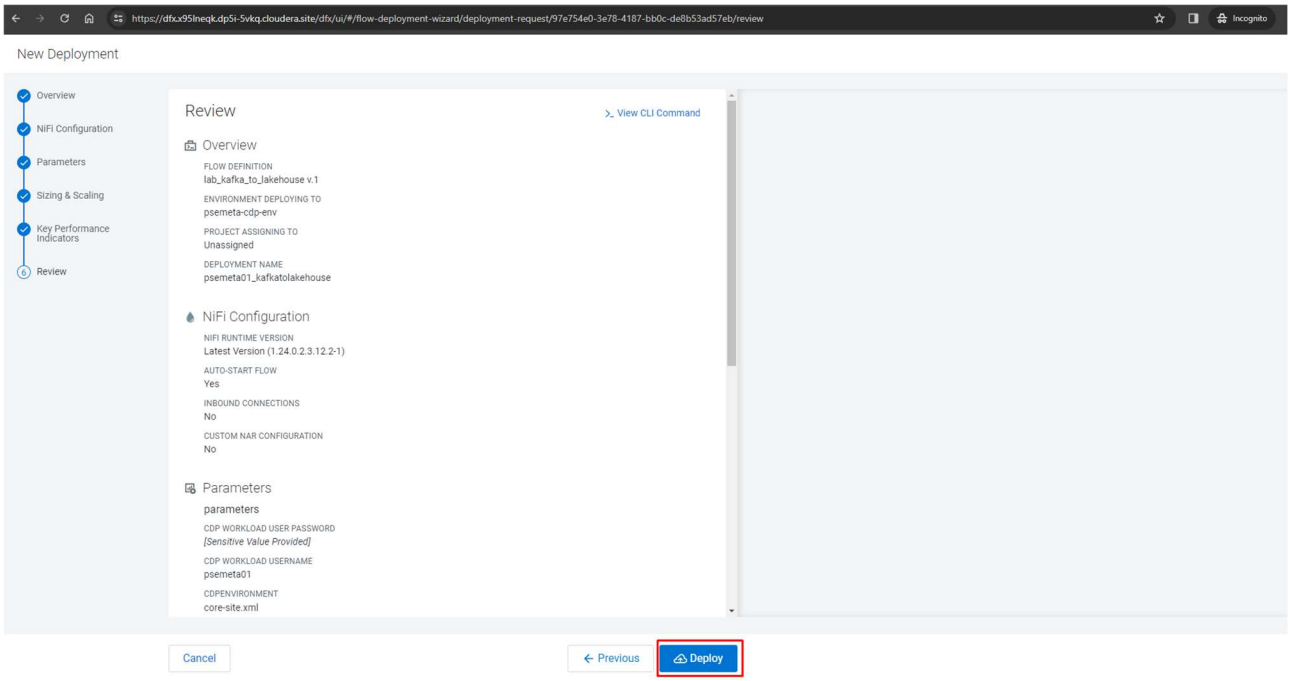
7. There is no need to configure auto-scaling parameters. Click **Next.**



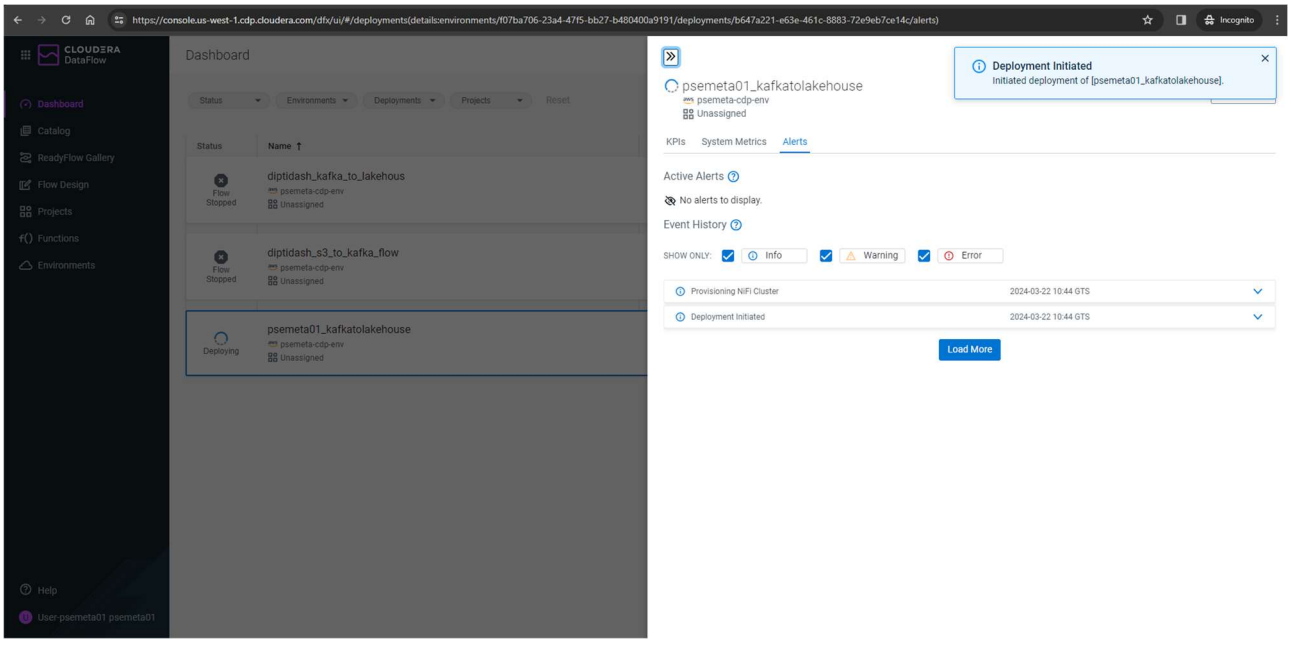
8. We are also not going to configure KPIs now. Click **Next** to continue the configuration.

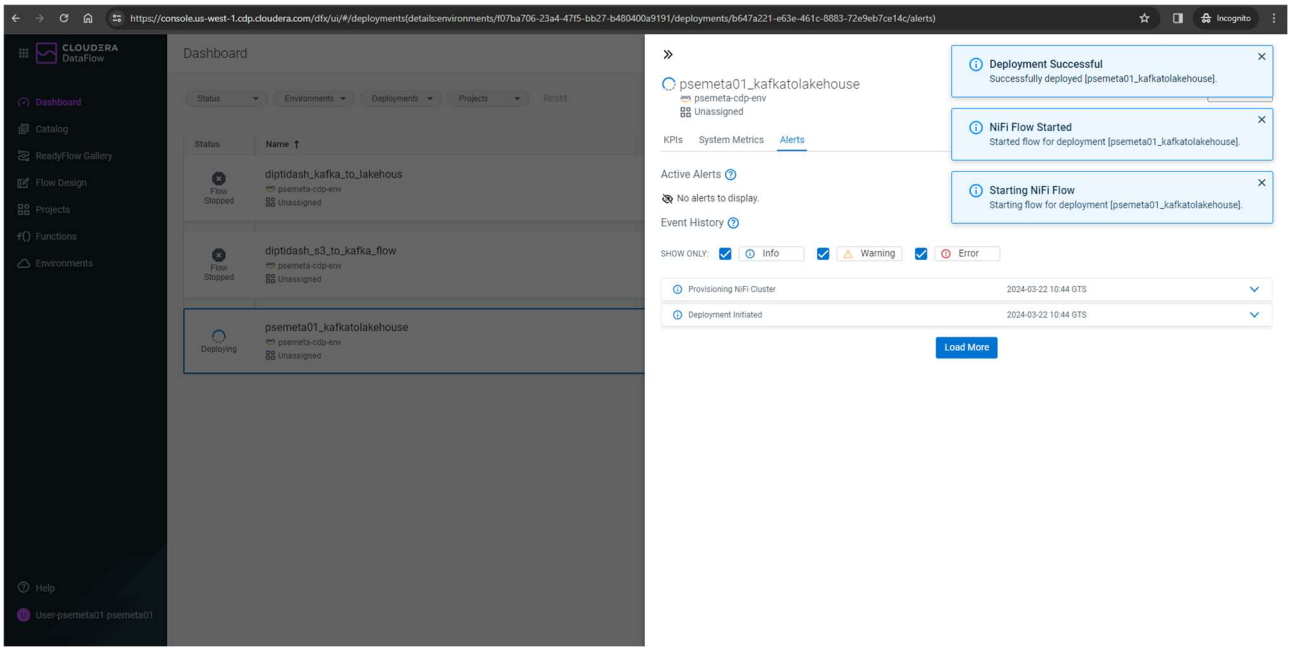


9. Review all the information entered for your Flow, then click on **Deploy** to start the deployment process.

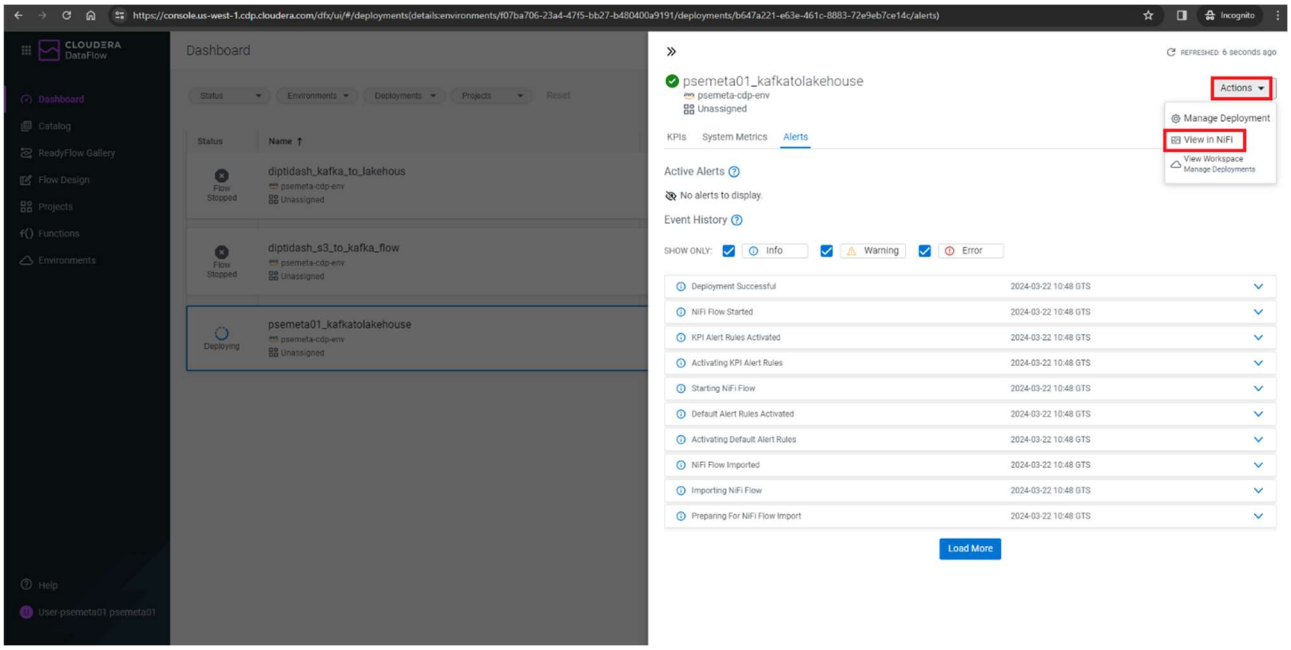


10. The blue box indicates that the Flow deployment process has been started. By clicking on the button **Load More** you will be able to see the different stages of the deployment. After about 60 to 90 seconds approximately, the last event should be ***Deployment Successful***.

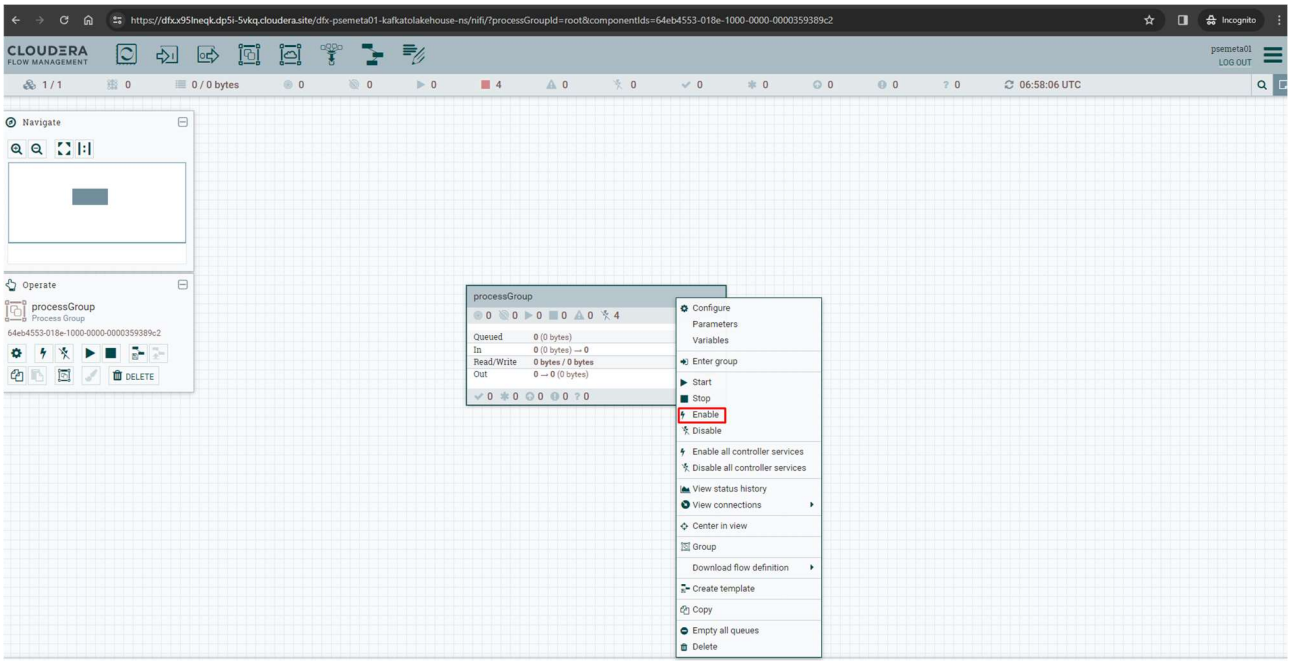



https://console.us-west-1.cdp.cloudera.com/dfx/ui/#/deployments(detailsEnvironments/f07ba706-23a4-47f5-bb27-b480400a9191/deployments/b647a221-e63e-461c-8883-72e9eb7ce14c/alerts)
Incognito
CLOUDERA DataFlow
Dashboard
Catalog
ReadyFlow Gallery
Flow Design
Projects
Functions
Environments
Help
User-psemeta01 psemeta01
Dashboard
Status
Environments
Deployments
Projects
Reset
Status
Name
Flow Stopped
diptidash_kafka_to_lakehous
psemeta-cdp-env
Unassigned
Flow Stopped
diptidash_s3_to_kafka_flow
psemeta-cdp-env
Unassigned
Deploying
psemeta01_kafkatolakehouse
psemeta-cdp-env
Unassigned
psemeta01_kafkatolakehouse
psemeta-cdp-env
Unassigned
KPIs
System Metrics
Alerts
Active Alerts
No alerts to display.
Event History
SHOW ONLY:
Info
Warning
Error
Provisioning NiFi Cluster
2024-03-22 10:44 GTS
Deployment Initiated
2024-03-22 10:44 GTS
Load More
Deployment Successful
Successfully deployed [psemeta01_kafkatolakehouse].
NiFi Flow Started
Started flow for deployment [psemeta01_kafkatolakehouse].
Starting NiFi Flow
Starting flow for deployment [psemeta01_kafkatolakehouse].

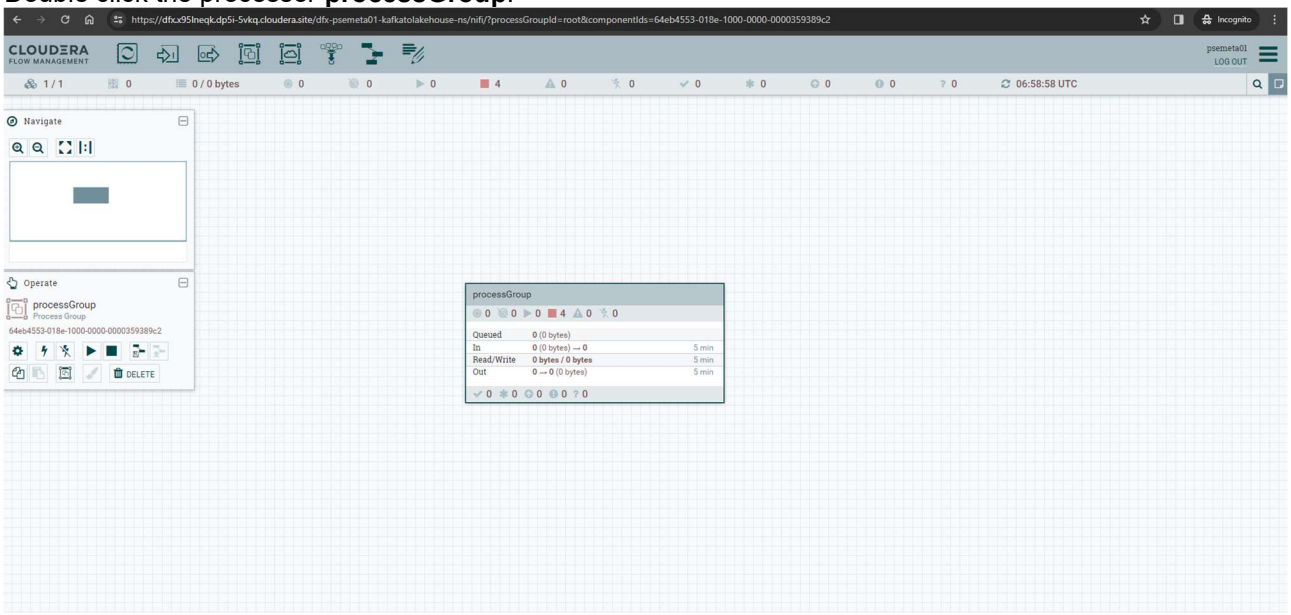
11. Once the deployment is finished, click on **Actions – View in NiFi** to see the details of the recently deployed Flow.



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



Double click the processor **processGroup**.

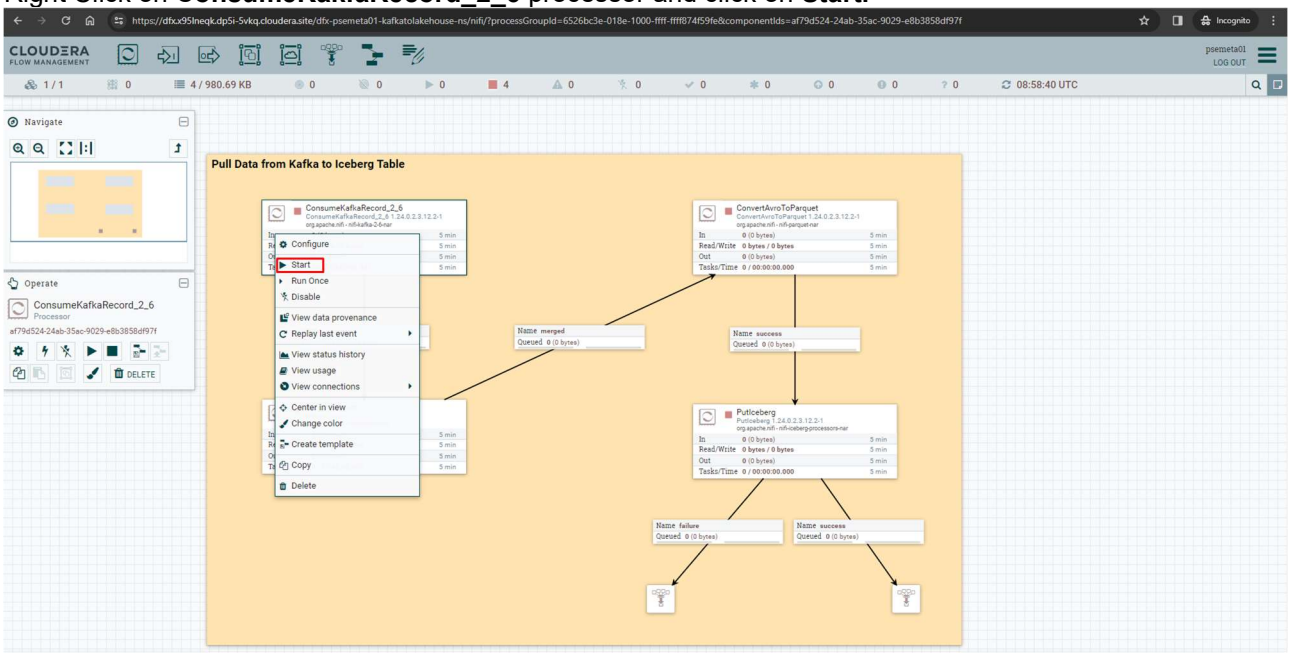


16. When opening the Process Group, you should be able to see the Processors that compose the Flow application. To summarize, there are four Processors:

a. ConsumeKakfaRecord, 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).

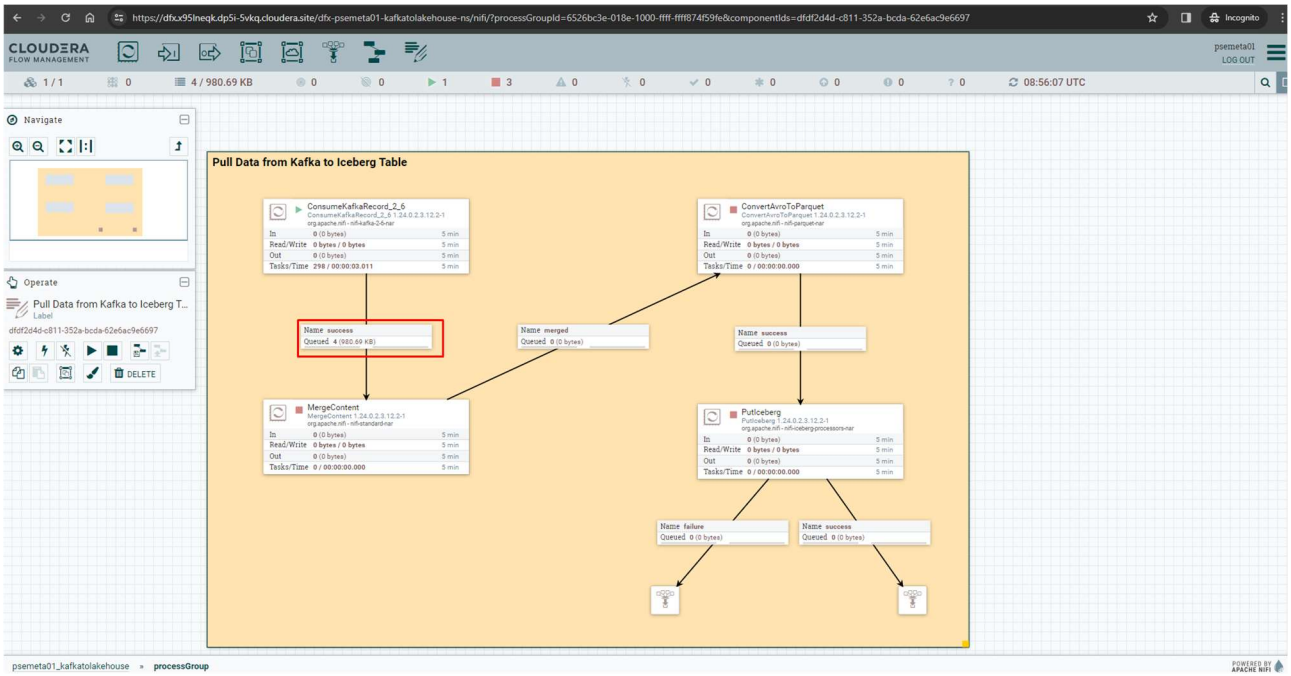
As you can see, the Processors are not started, they are paused.

Right Click on **ConsumeKakfaRecord_2_6** processor and click on **Start.**

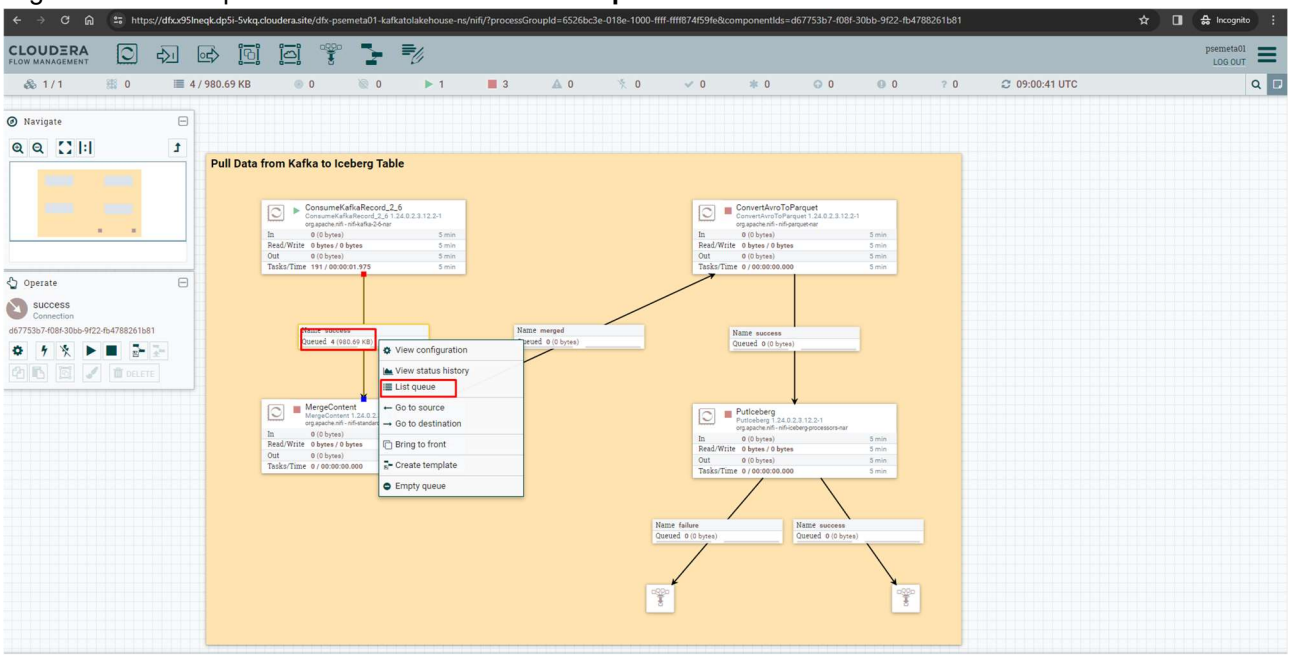


17. Flow Management allows us to see and access data in motion during the execution of the data flow. Between Processors **ConsumeKafkaRecord** (just started) and **MergeRecords**, there is a connection. This connection is what joins the Processors and transmits data from one to theother, and you can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

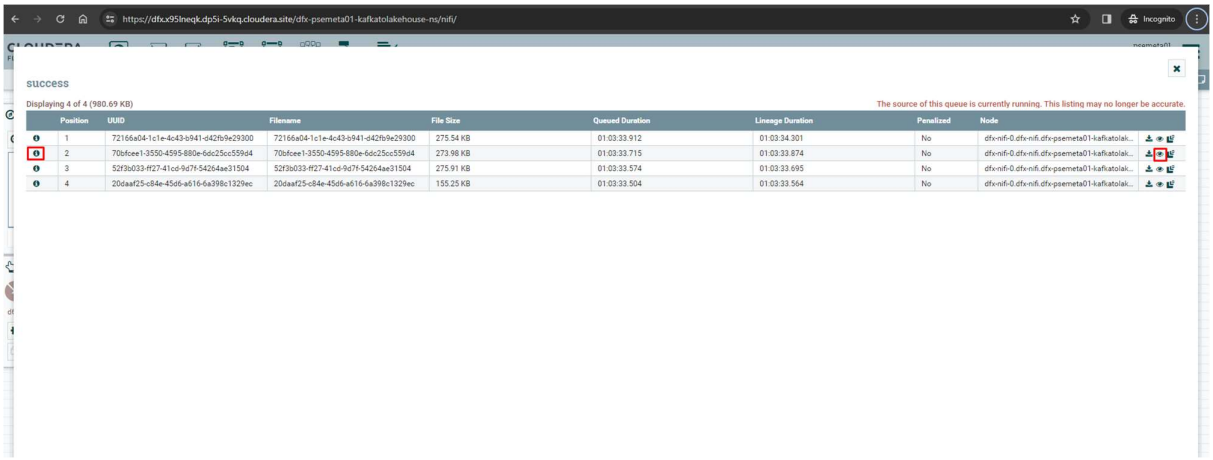
You will see data start to queue up in the connector shortly after you start the first processor.



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

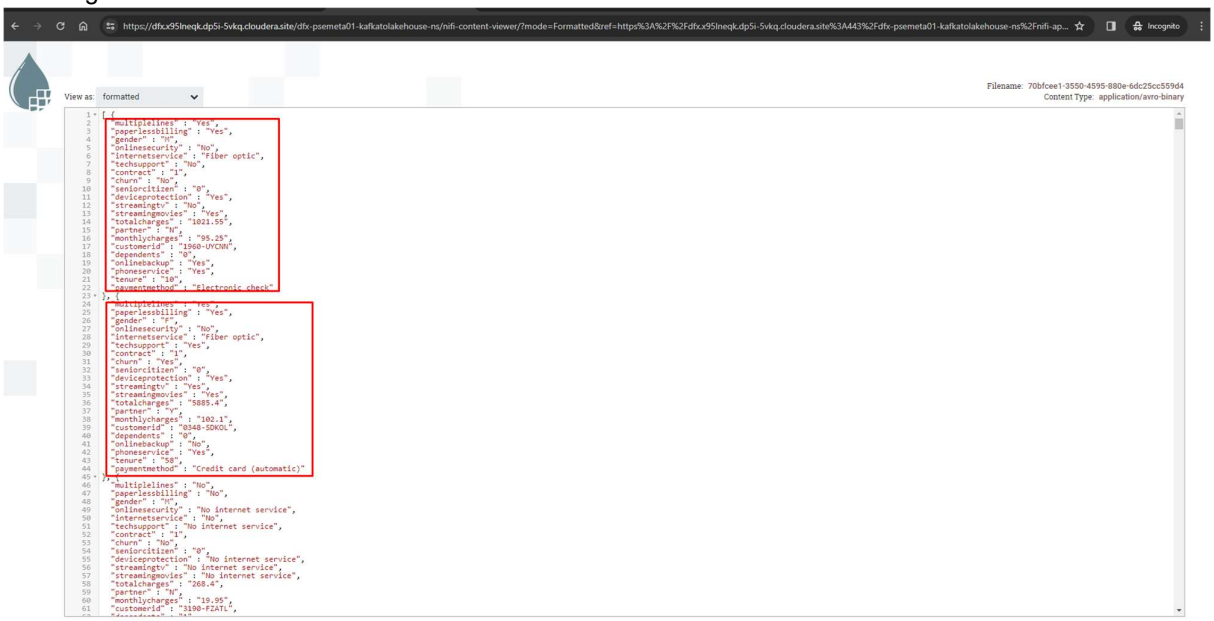


18. You will see the data that is listed here. Click on the eye icon on the extreme right.

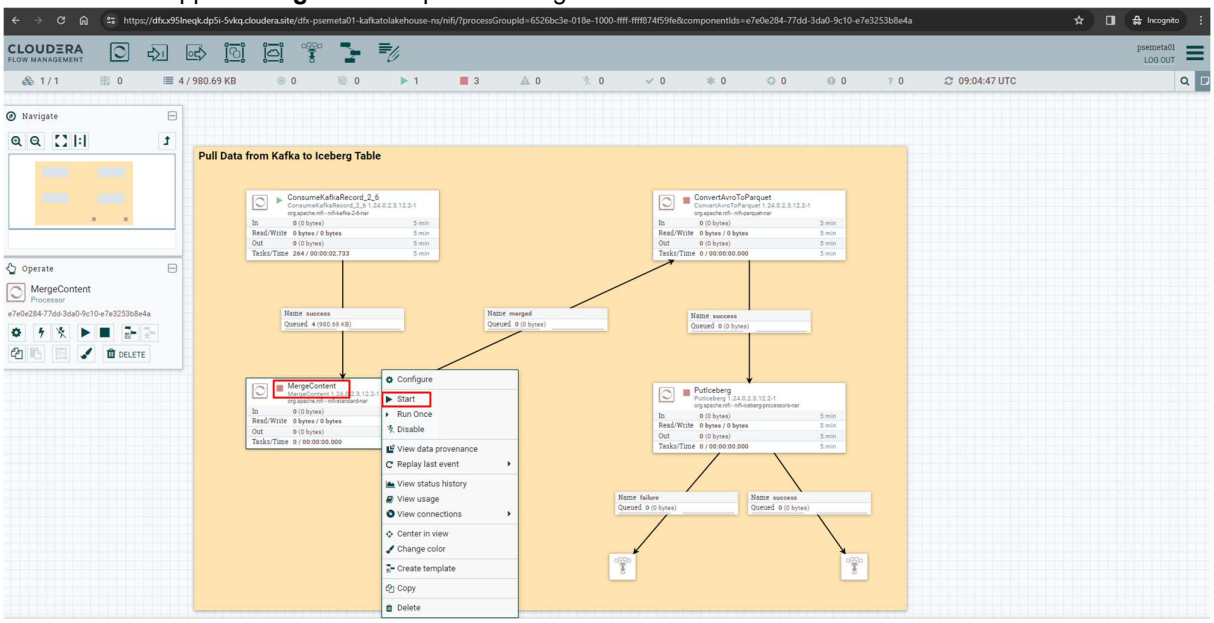


The new window that opens shows the data of the FlowFile content. Being in AVRO format, it is not fully readable. A deserializer must be selected to correctly display the data. For this, in the upper left, select the option **formatted** from the menu **View as**.

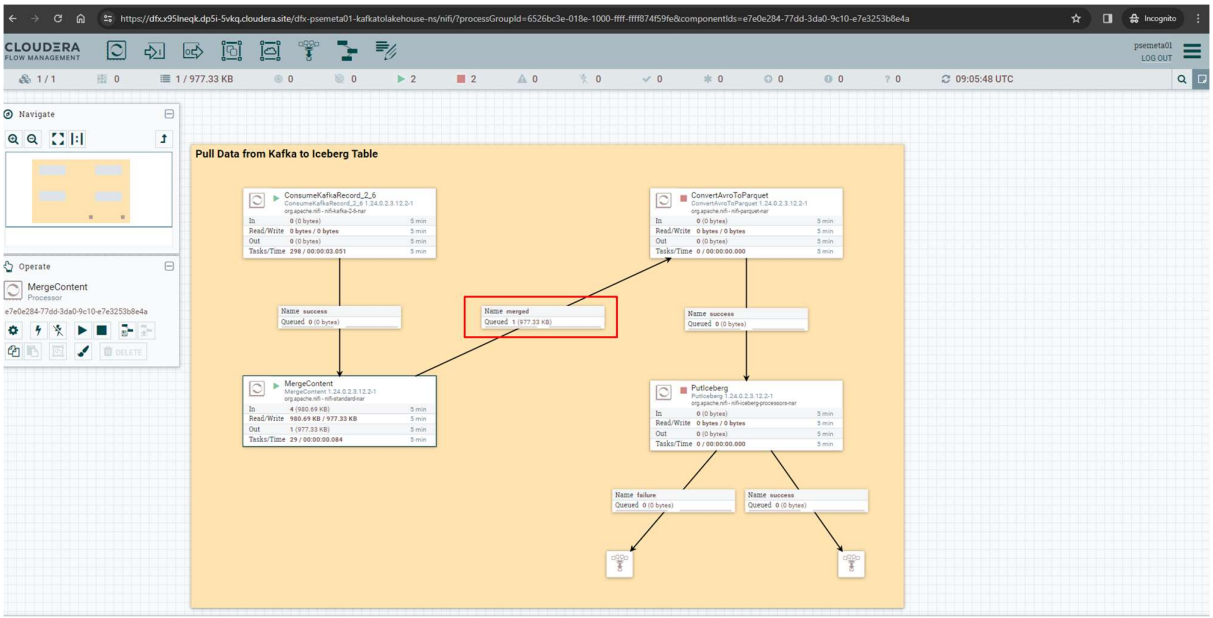
19. Now you can display the data correctly. Notice that the fields or attributes indicated at the beginning of the workshop appear. You can close that FlowFile window and the popups, returning to the canvas with the four Processors.



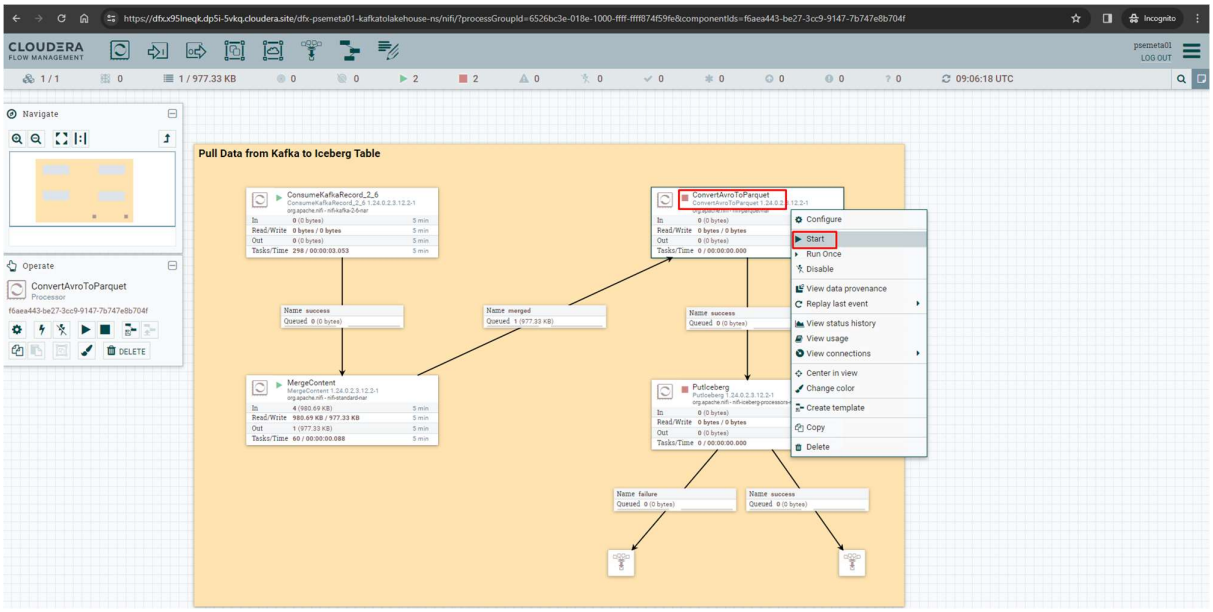
20. Start the stopped: **MergeContent** processor again to resume the flow.

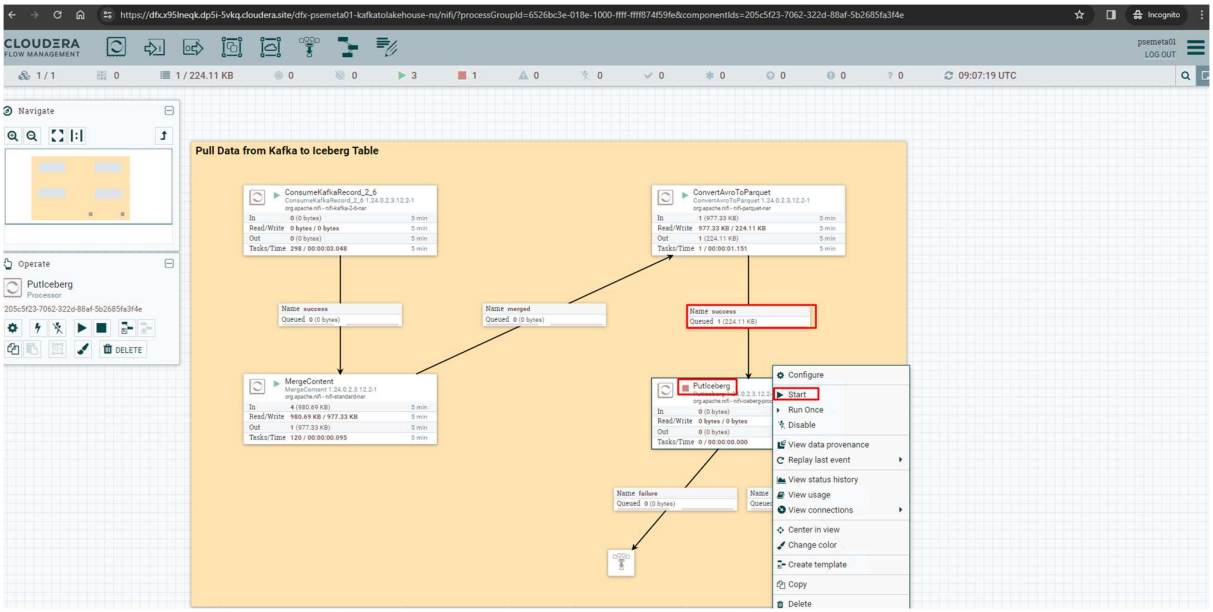


You can now see the records getting merged and passed through to the next processor.

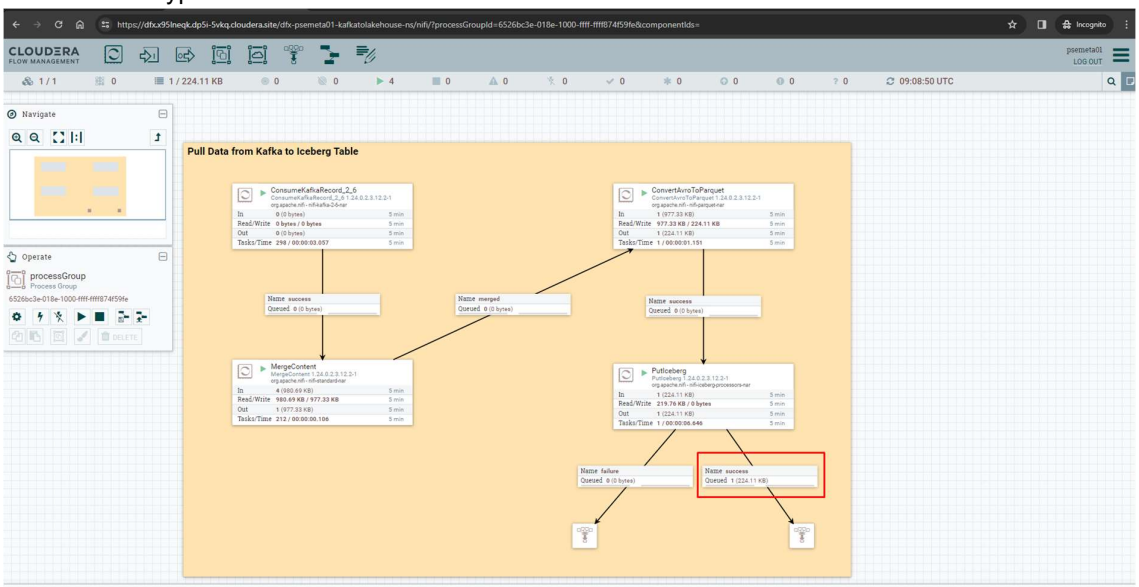


Start the next 2 processors as well **ConvertAvroToParquet** & **PutIceberg.**





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.