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:

A screen shot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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’.

A screenshot of a computer

Description automatically generated

Click on **‘Set Workload Password’**.

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

Password is set successfully.

A screenshot of a computer

Description automatically generated

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.

A screenshot of a computer

Description automatically generated

1. 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. Click on the Flow called **lab\_kafka\_to\_lakehouse** to start deploying it.

A screenshot of a computer

Description automatically generated

1. 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.

A screenshot of a computer

Description automatically generated

1. 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**.

A screenshot of a computer

Description automatically generated

1. From this point, you will need to enter the Flow configuration. Start by assigning a

**Deployment Name**, **Target Project**, and click **Next**.

For the purposes of this workshop, please name the Flow starting with your assigned username. For example, **psemeta01\_kafkatolakehouse**

Select **Unassigned** for the project, which is a way to organize your flows. Click on **Next**.

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

1. 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**

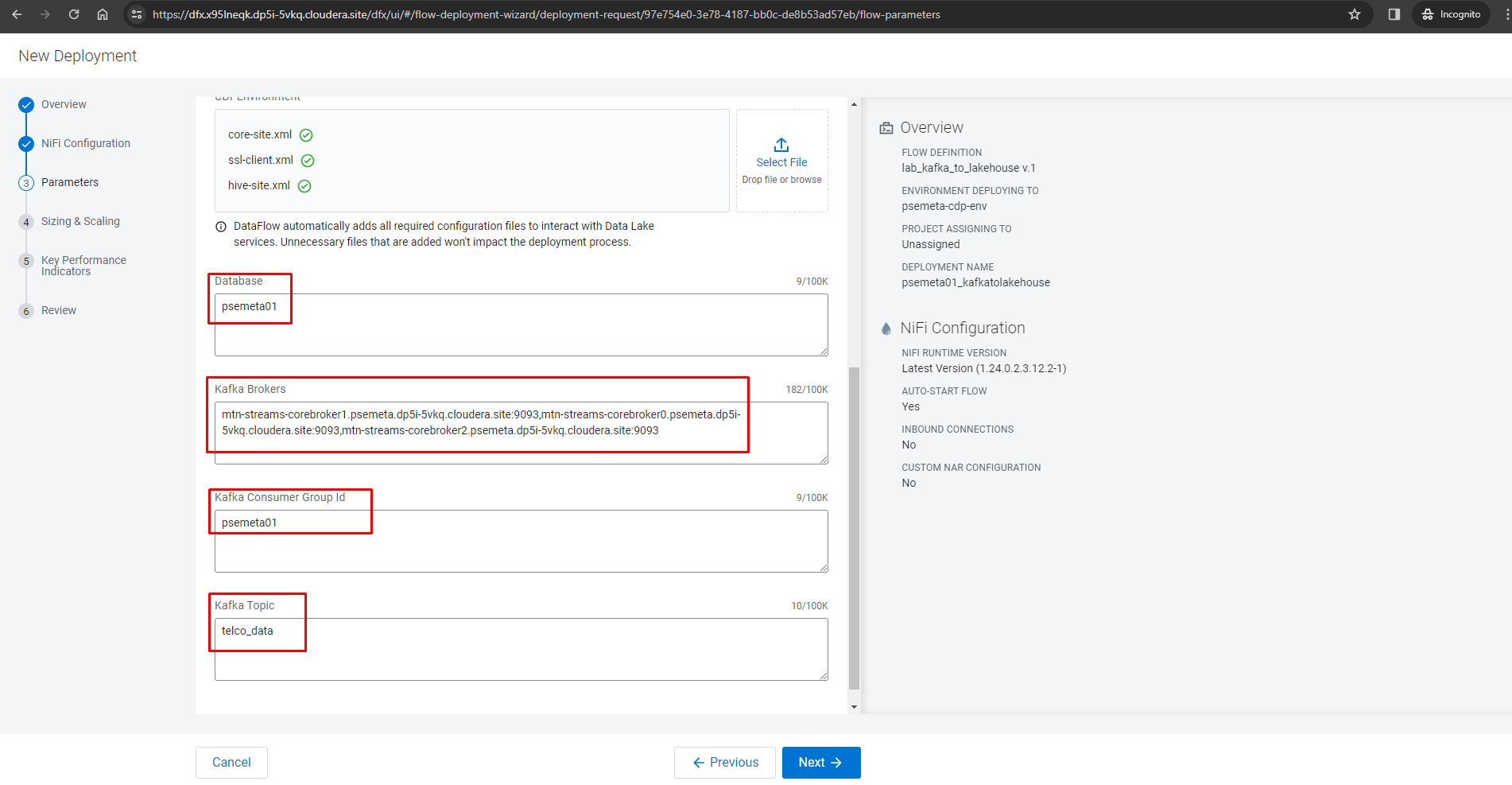
A screenshot of a computer

Description automatically generated

**Kafka Brokers:**

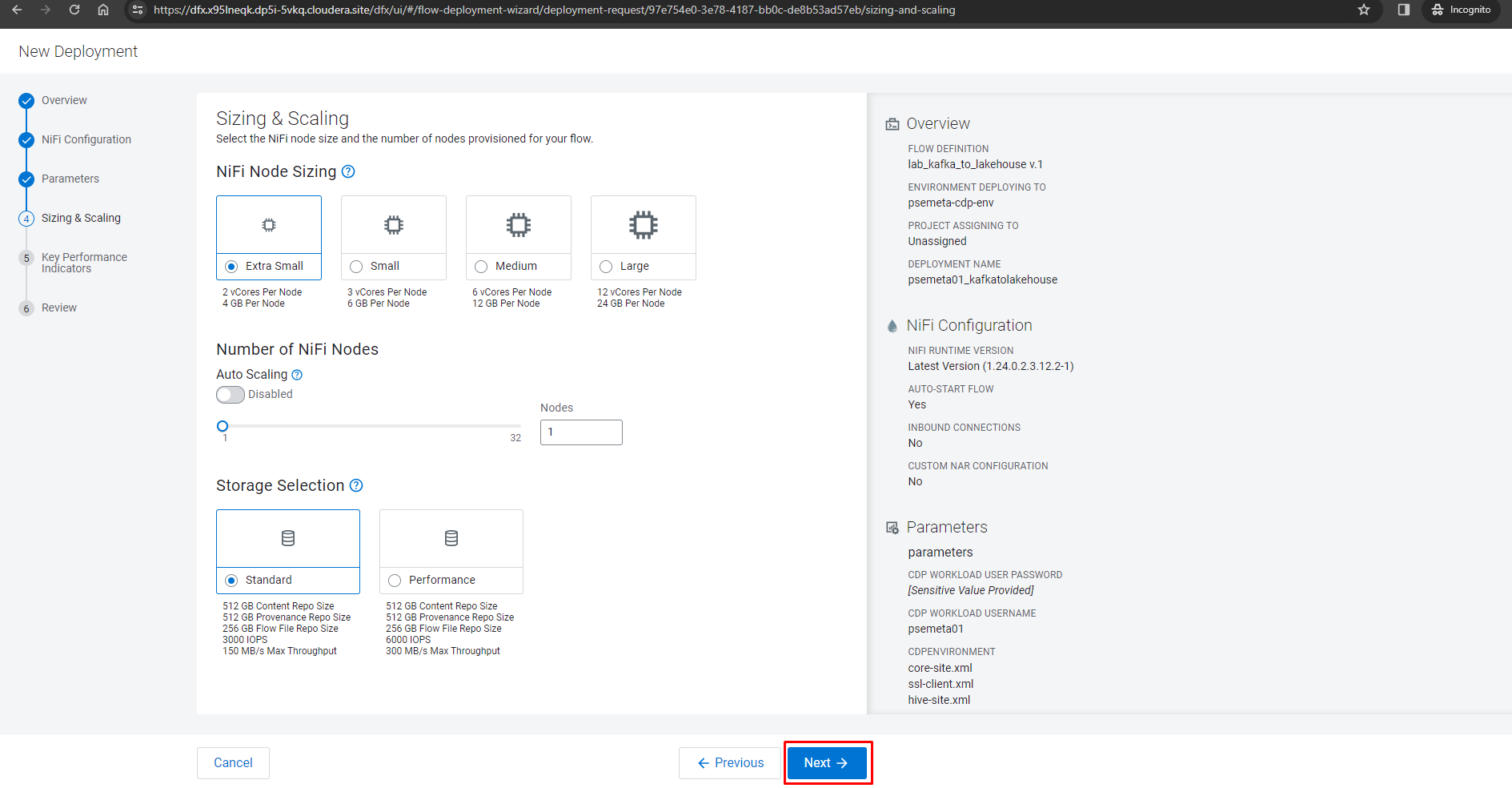
This value should be provided by the instructor.

**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.**

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



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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

1. 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*.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

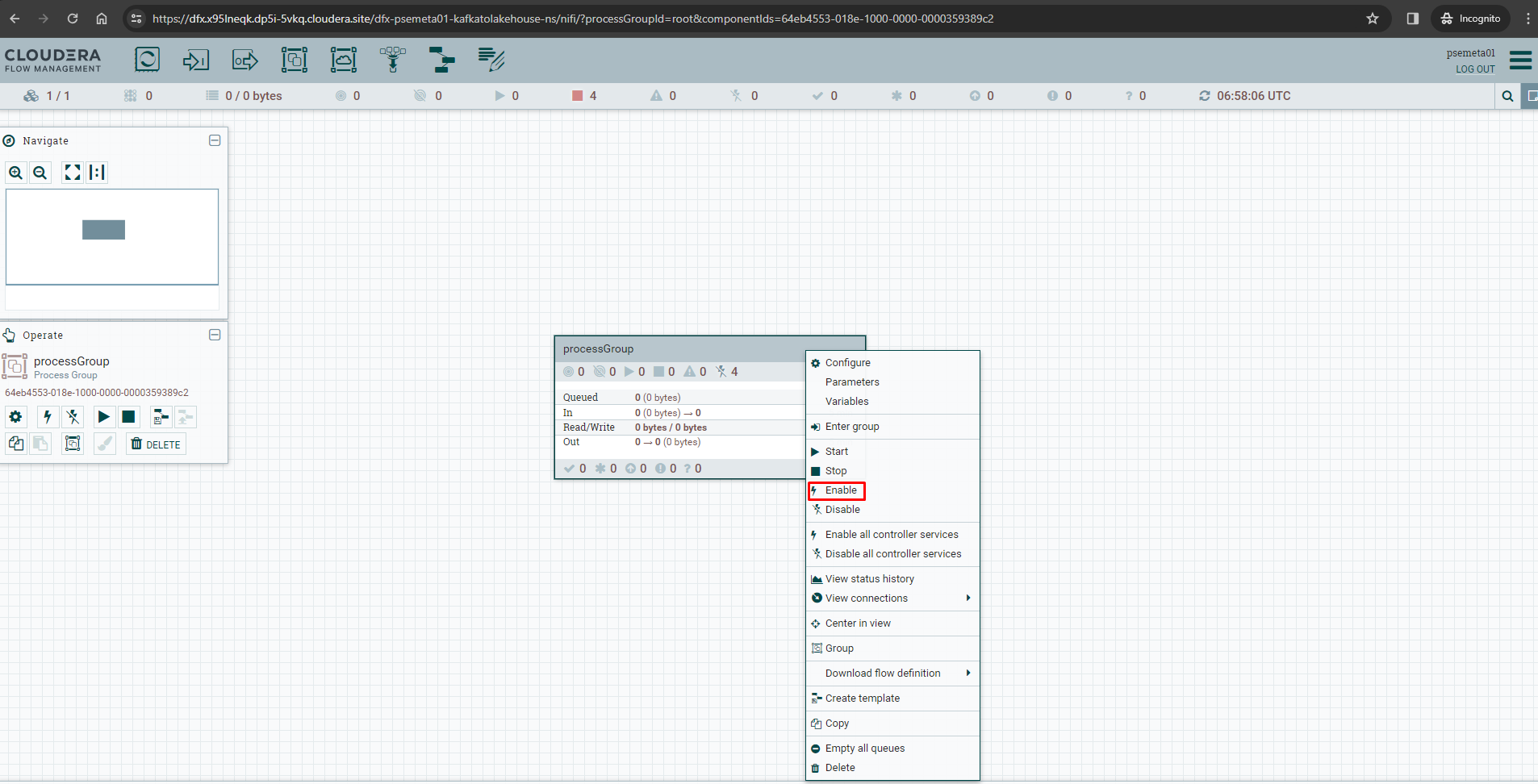
Description automatically generated

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

A screenshot of a computer

Description automatically generated

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



Double click the **processGroup**.

A screenshot of a computer

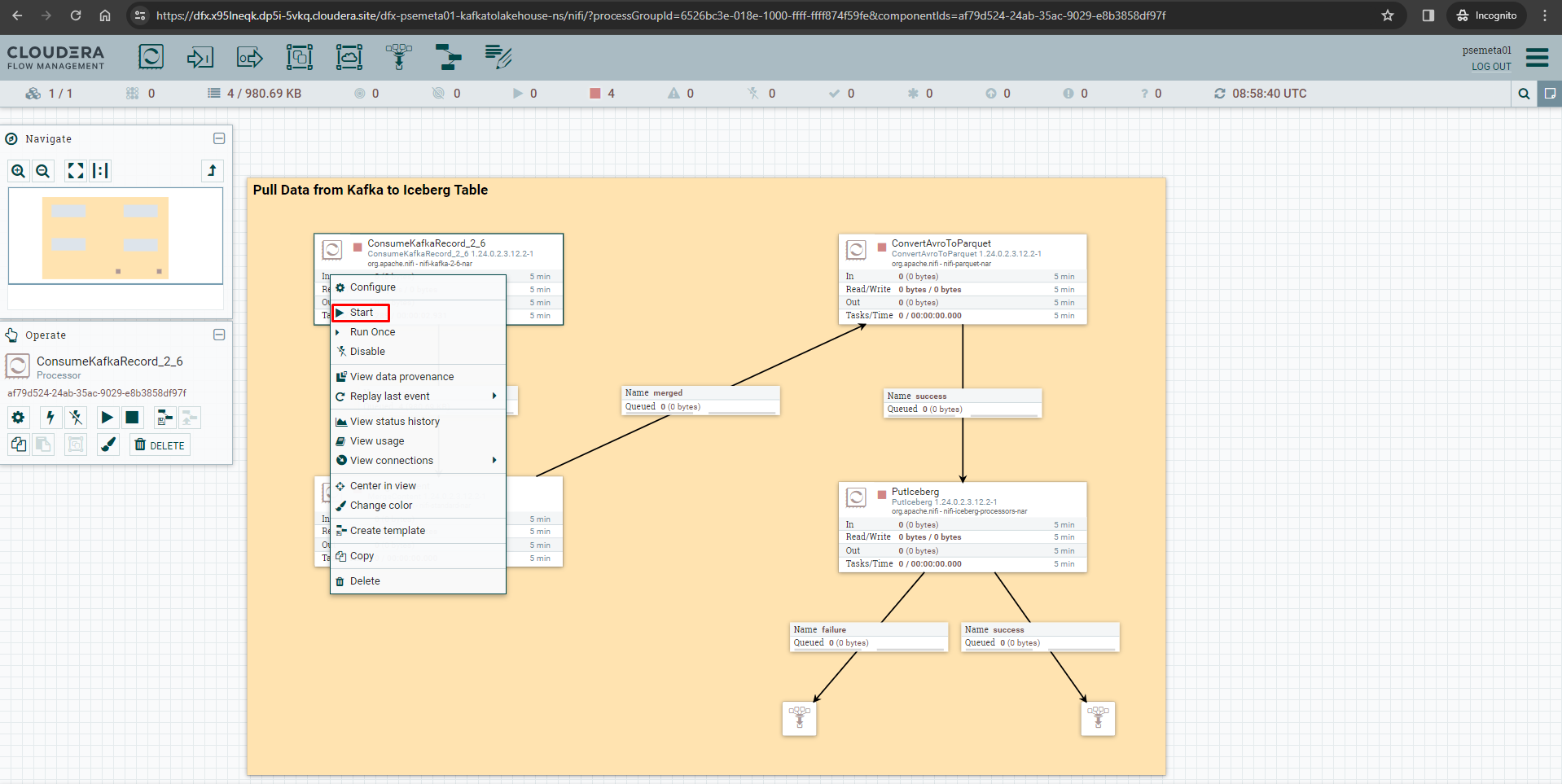
Description automatically generated

1. 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**, 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 **ConsumeKakfaRecord\_2\_6** processor and click on **Start.**



1. 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.

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

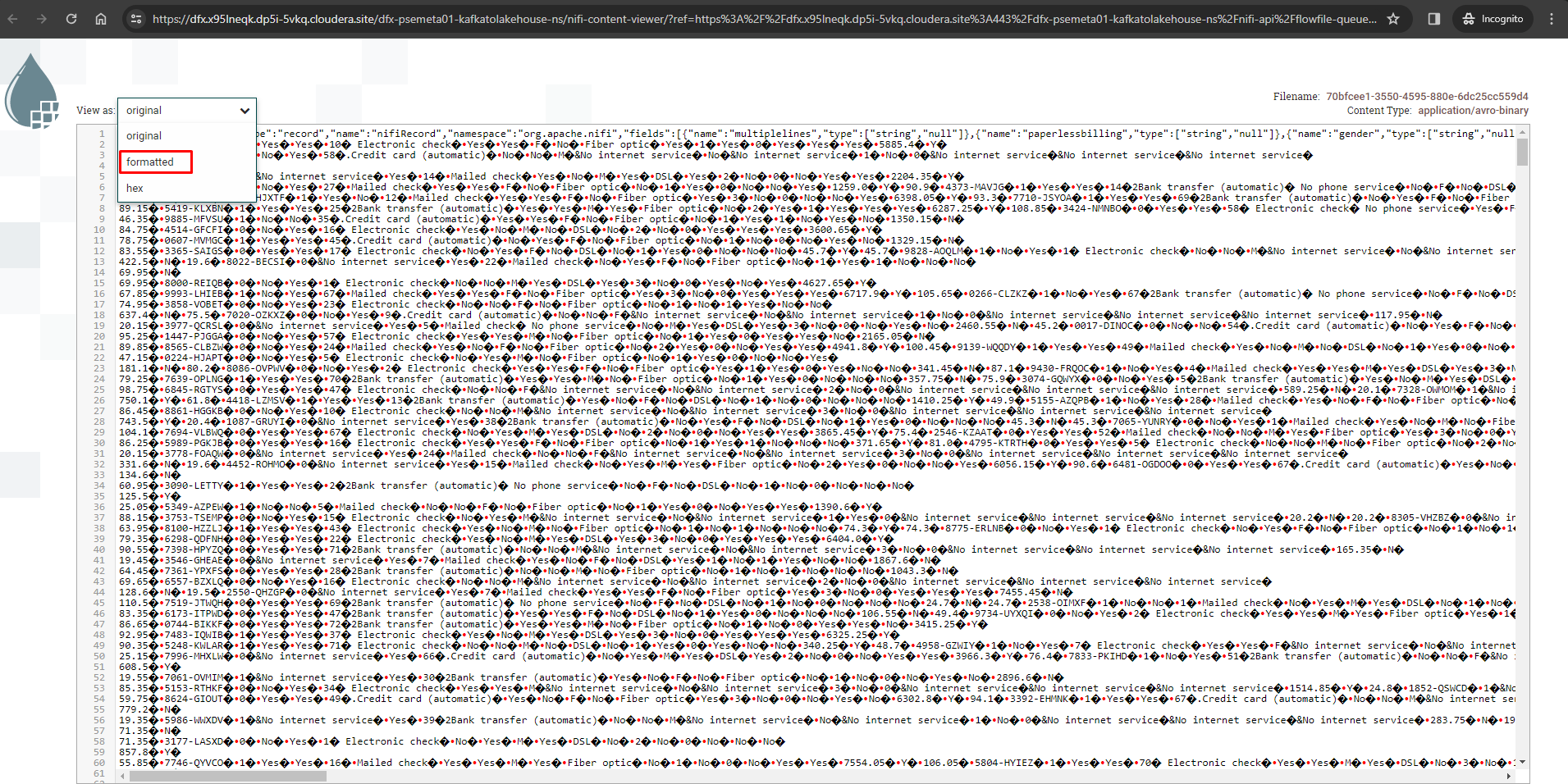
Description automatically generated

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

A screenshot of a computer

Description automatically generated

1. 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**.



1. 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.

A screen shot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

1. **BONUS**: NiFi is a powerful ingestion tool that gives you granular visibility into everything that’s done to the data - for example, right-click on any processor and then click on **View data provenance** to see this in action.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

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.

A screenshot of a computer

Description automatically generated