CLOUDERA

Paris Public Cloud Hands on Lab

Workshop Guide

18 Oct 2023

Jacques Marchand
Charles Aad
Cristina Sanchez

TABLE OF CONTENTS

Foreword 2
1. Introduction 3
2. Data Flow Lab: 4
    2.1. Goals 4
    2.2. Lab 1 - Ingest Kafka streams to Iceberg table 4
    2.3. Lab 2 - Stream Messaging Manager - Optional 19
3. Data Engineering 22
    3.1. Goals 22
    3.2. Lab 1 - Enrich the Ingested Iceberg table 22
4. Data Warehouse 32
    4.1. Goals 32
    4.2. Dashboard Development 32
5. Machine Learning 42
    5.1. Goals 42
    5.2. Create a Machine Learning Model for Churn Prediction 42
6. Optional Labs 60
    6.1. Goals 60
    6.2. ML -Deploy Applied Machine Learning Model - Optional 60
    6.3. Add a third visual element - Optional 63
    6.4. One more visual element - Optional 63
    6.5. Data Discovery and SQL Analysis Using HUE Dashboard - Optional 64
    6.6. Part 2: Add a New Field - Optional 68
7. Take-aways 80

Foreword

Document Status

This document does not form a contract or offer to contract.

Response Limitation

All products or company names are used for identification purposes only, and maybe trademarks of their respective owners.

Confidentiality

The material contained in this document represents proprietary information pertaining to Cloudera products and methods.

Validity

Cloudera does not take responsibility for the changes and product updates post publication of this document and does not commit to keeping it updated.

Change log

Date	Name	Change
Sept 2023	Alex Campos Simoes	Workshop creation
12 Oct 2023	Cristina Sánchez	Update of Workshop elements
13 Oct 2023	Jacques Marchand	Add optional exercices
16 Oct 202	Charles Aad	Document merge, sanity check

1. Introduction

At Cloudera we believe that data can make what is impossible today, possible tomorrow. We empower people to transform complex data into clear and actionable insights. Cloudera delivers an enterprise data platform for any data, anywhere, from the Edge to AI and it's powered by the relentless innovation of the open source community, Cloudera advances digital transformation for the world's largest enterprises.

Today, Cloudera offers a mature and operational data lake stack. Since Cloudera is uniquely positioned in the on premises space, private cloud and public cloud, it can deliver a highly differentiated hybrid and multi-cloud vision.

The scope of this workshop is to experiment with the latest stack on public cloud through an overly simplified telco customer churn use case. All this is built for the purposes of experimentation.

This workshop guide is a step by step document to follow in order to deliver the workshop and is completed by

- A preparation guide (internal)
- A presentation designed to facilitate the experimentation. (to be delivered to you)

At Cloudera, we thank you for your confidence and for experimenting with our products.

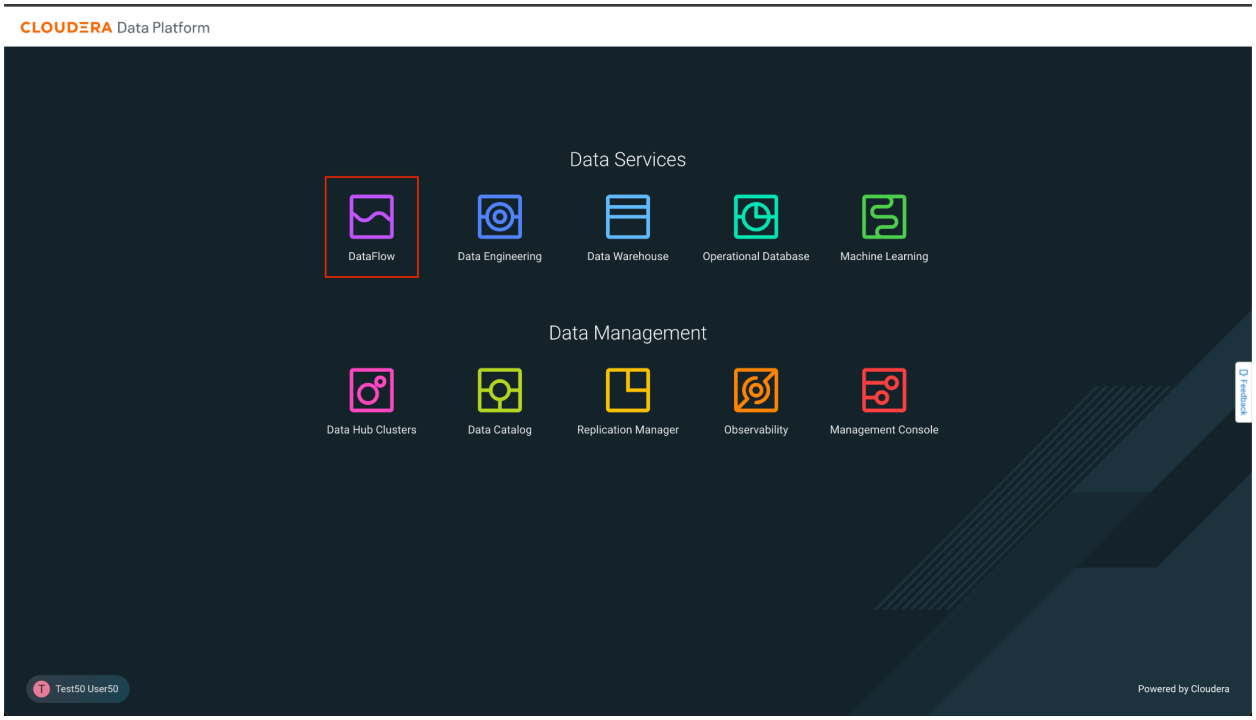
2. Data Flow Lab:

2.1. Goals

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

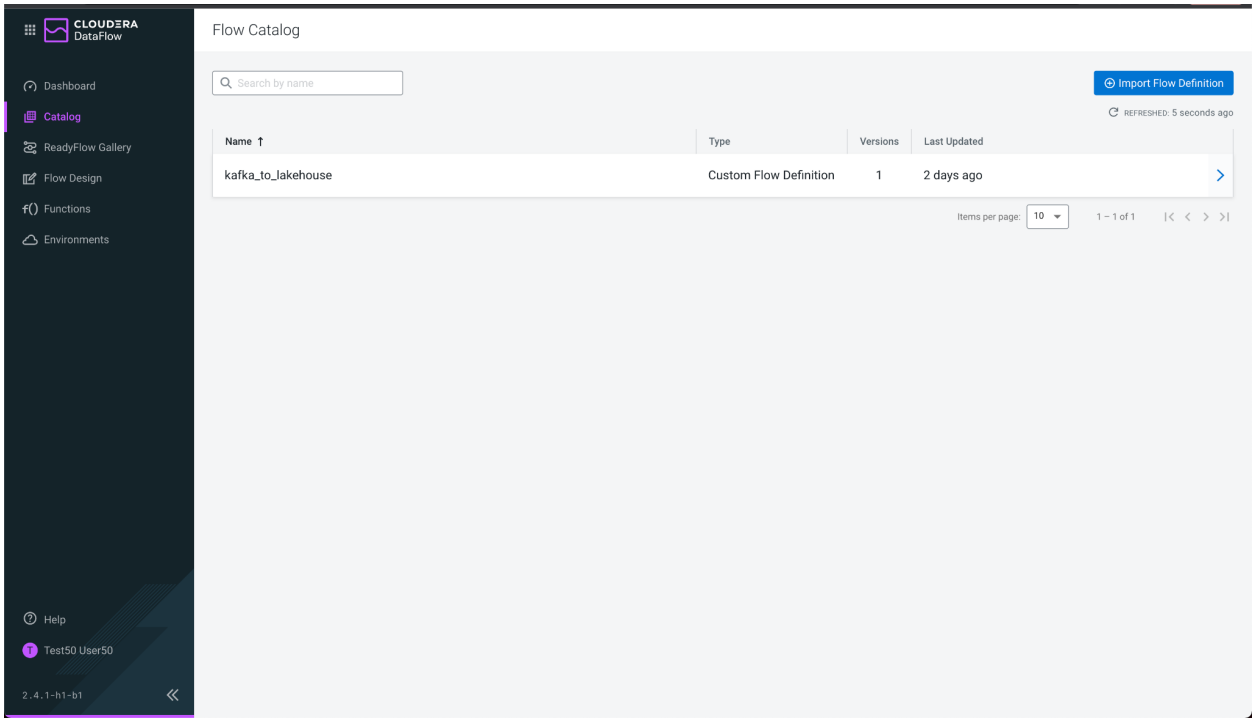
2.2. Lab 1 - Ingest Kafka streams to Iceberg table

1. Click on DataFlow from CDP PC Home:

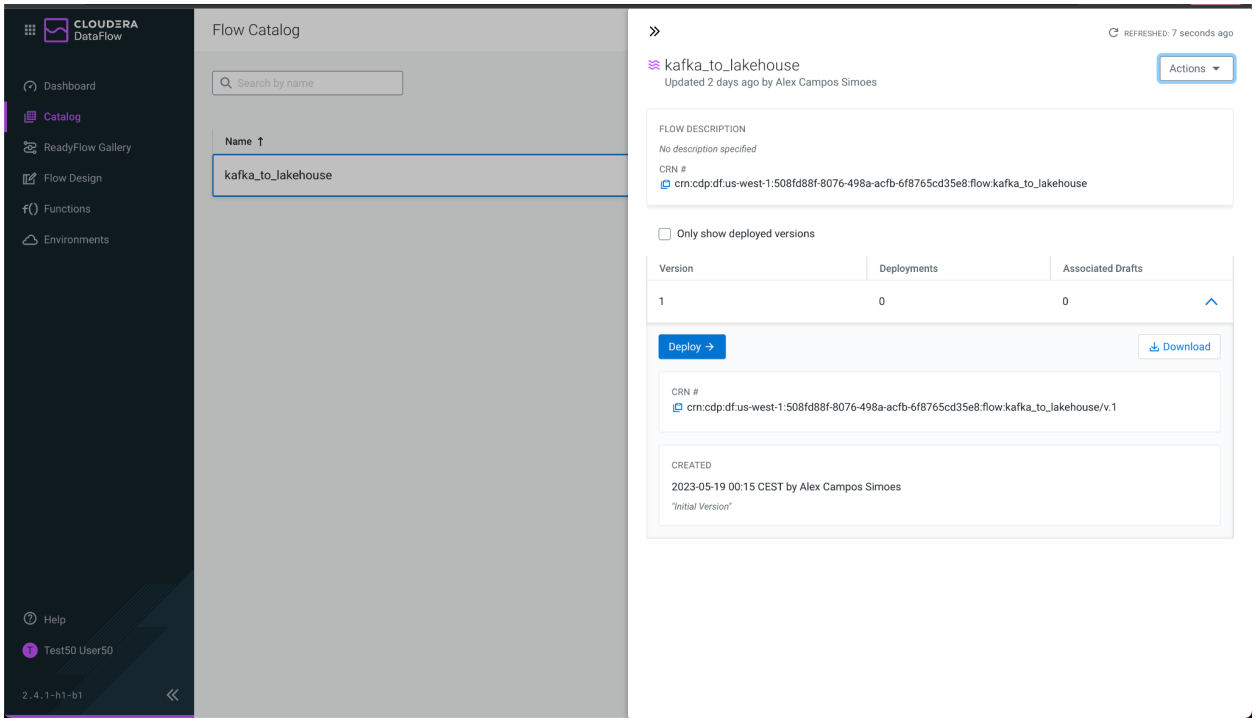


2. Once in DataFlow, click on the option **Catalog** from the left menu. The data ingestion application templates are listed here. For the purpose of this workshop, we have created and published a template that allows you to read Kafka topic data and ingest/store it in the Lakehouse provided by CDP Public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

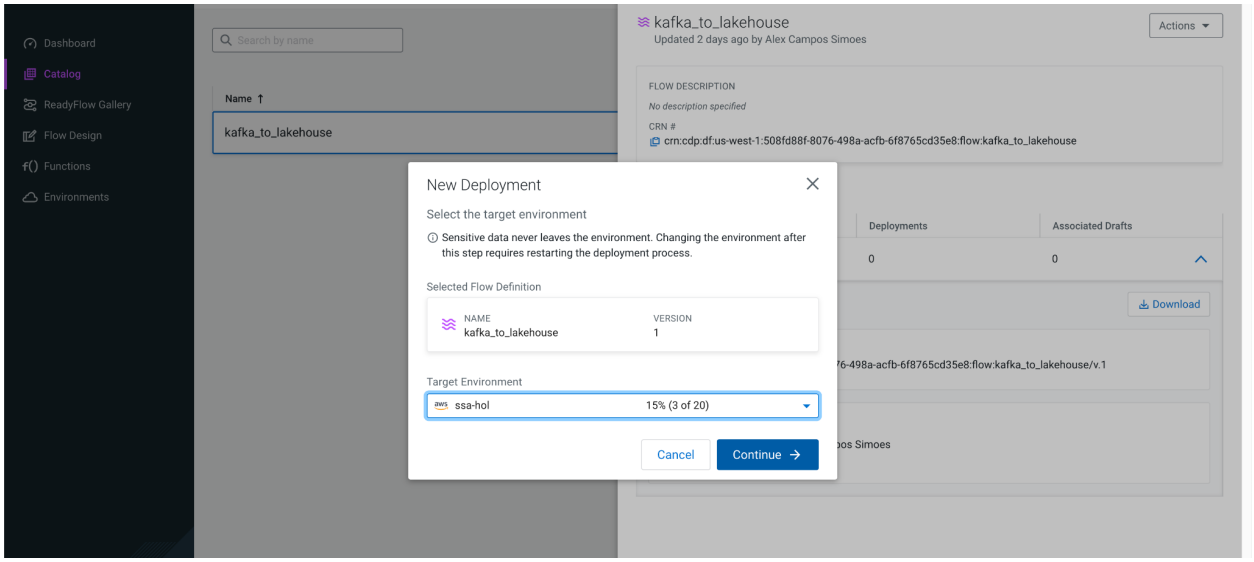
start deploying it.



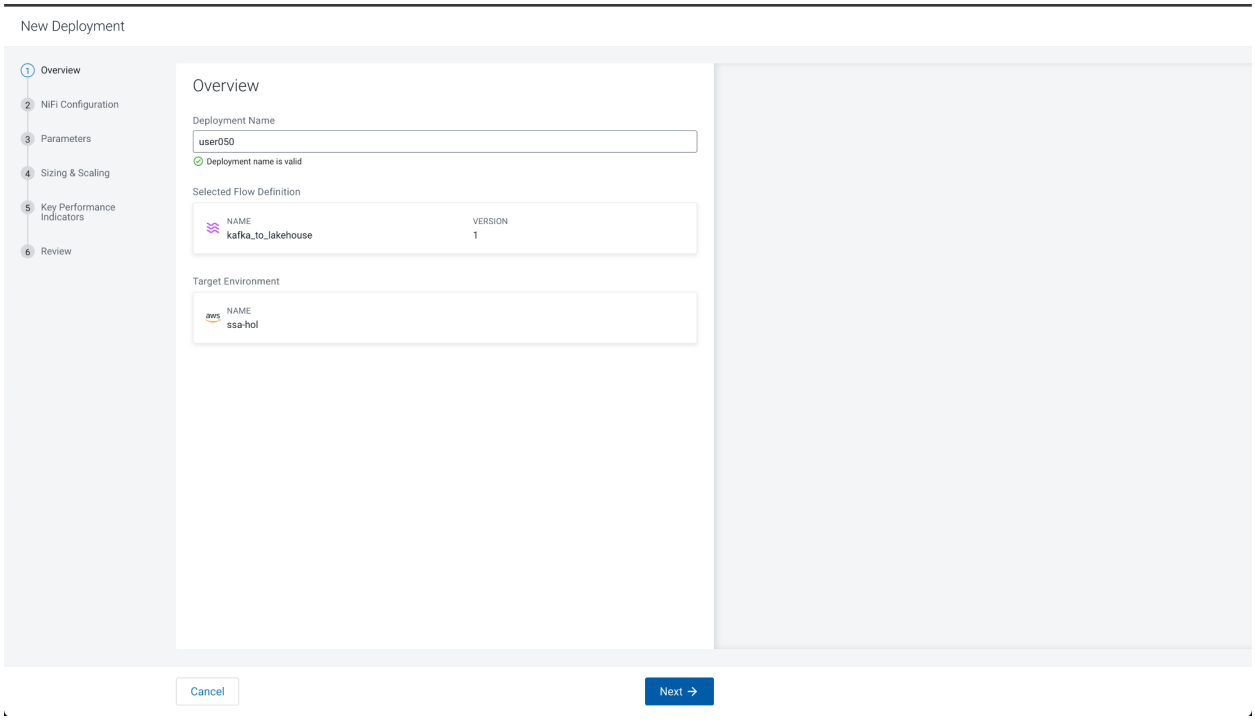
3. When clicked, the following panel appears with the Flow information. It shows the available versions, creation date, creator user, and a button **Deploy** to start the deployment. Click on that button.



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**.

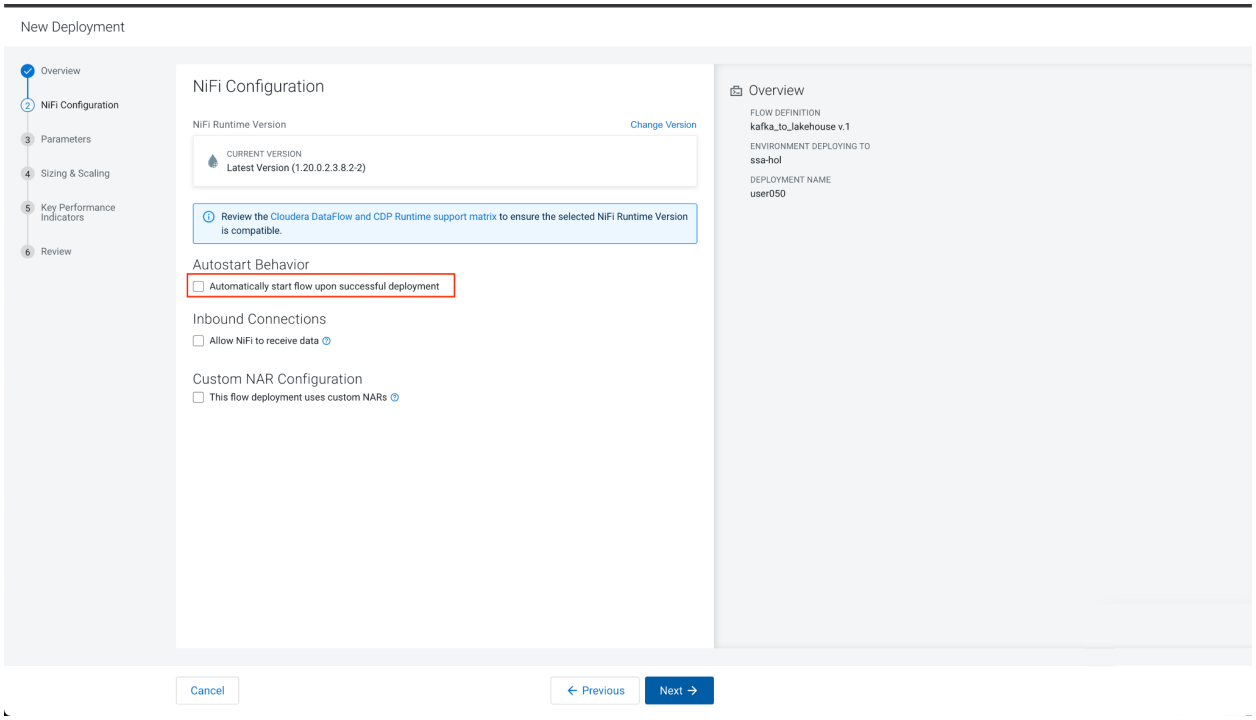


5. From this point, you will need to enter the Flow configuration. Start by assigning a name (**Deployment Name**) and click **Next**.
*For the purposes of this workshop, please name the Flow with the assigned username **-user050**, for example.*



6. Uncheck the option **Automatically start flow upon successful deployment** and click **Next.**

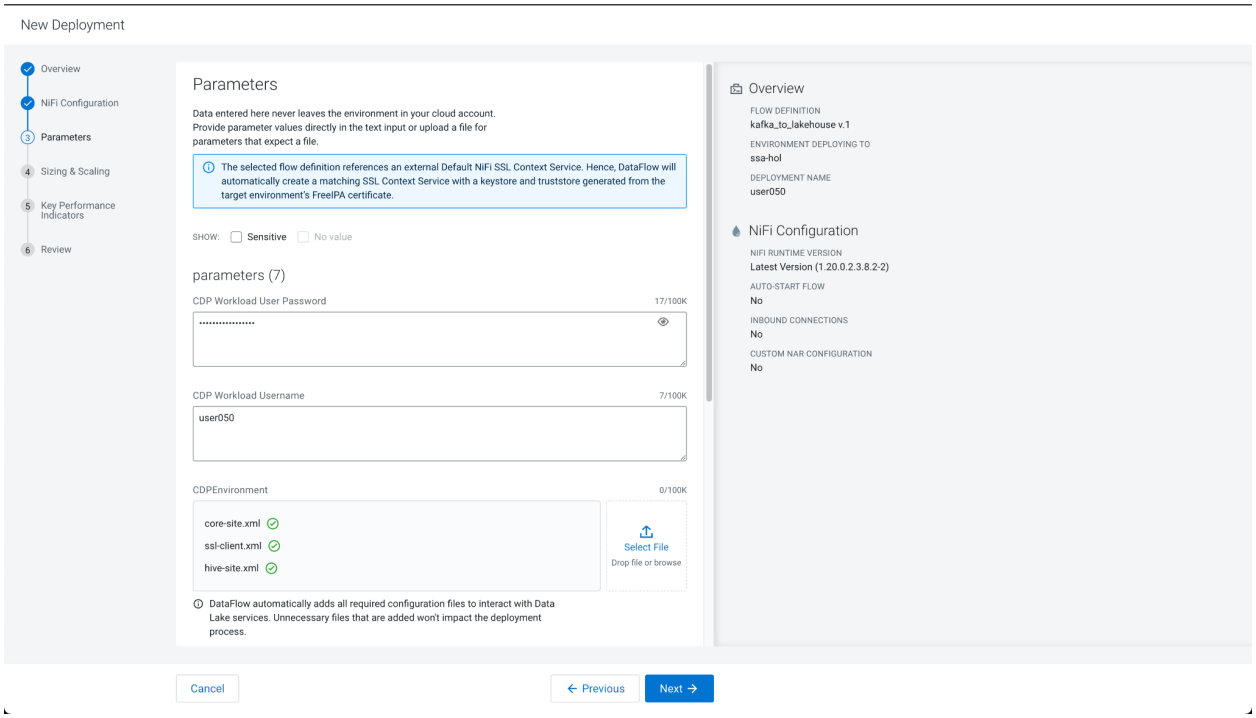
We are going to run Flow step by step, so we don't want it to start automatically.

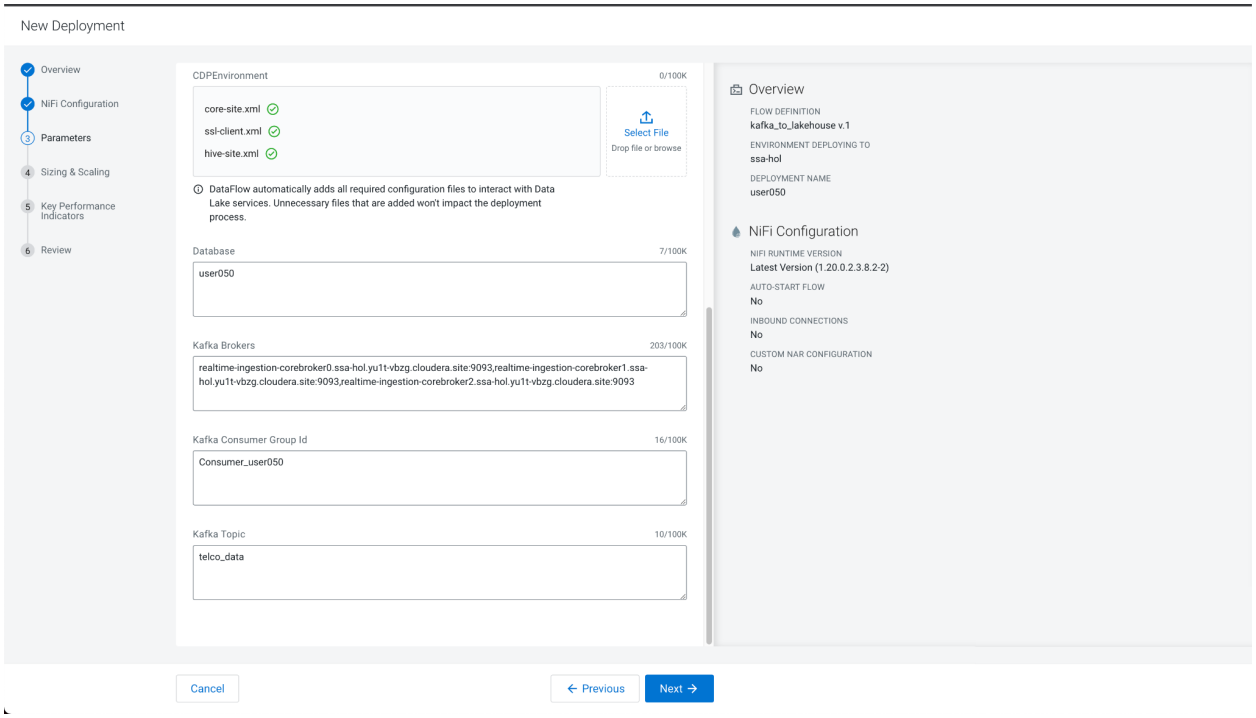


7. In this part of Parameters, you must enter the following values:

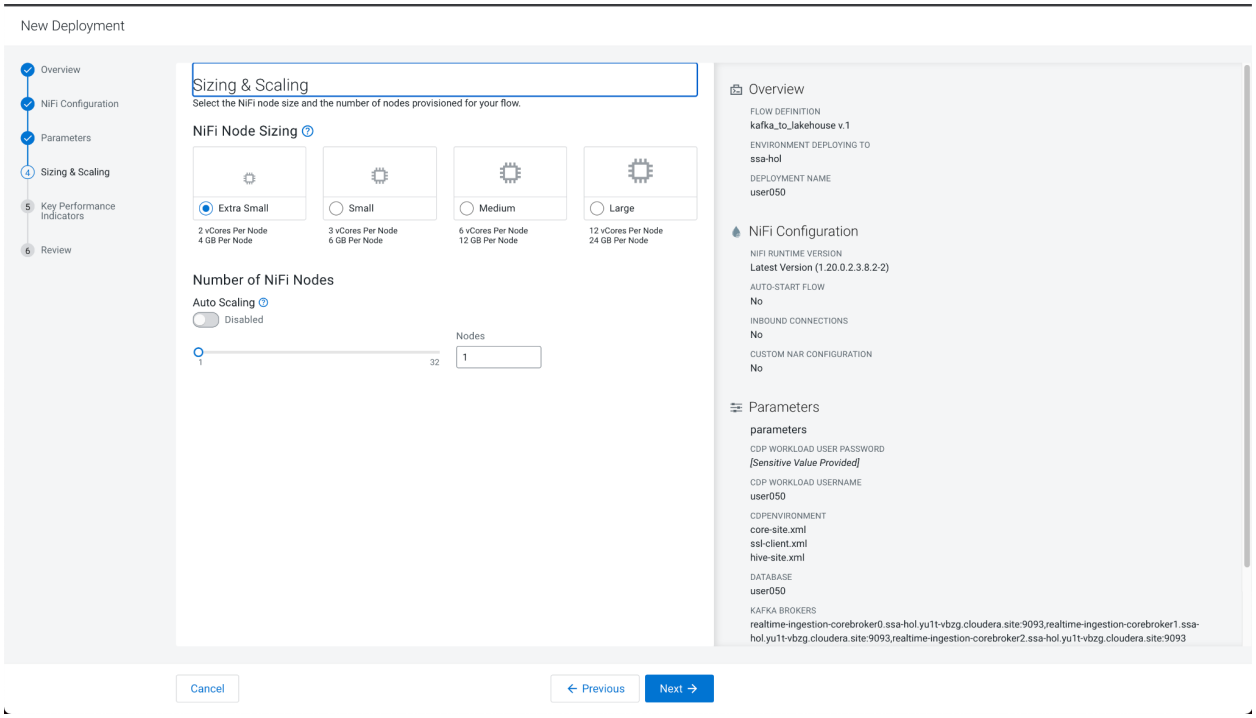
CDP Workload User Password: Enter the Workload Password shared at the beginning of the workshop.
CDP Workload Username: enter the assigned user number, *user050*, for example.
Database: enter the assigned user number, *user050*, for example. This database and the tables are already pre-created for you. 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.**

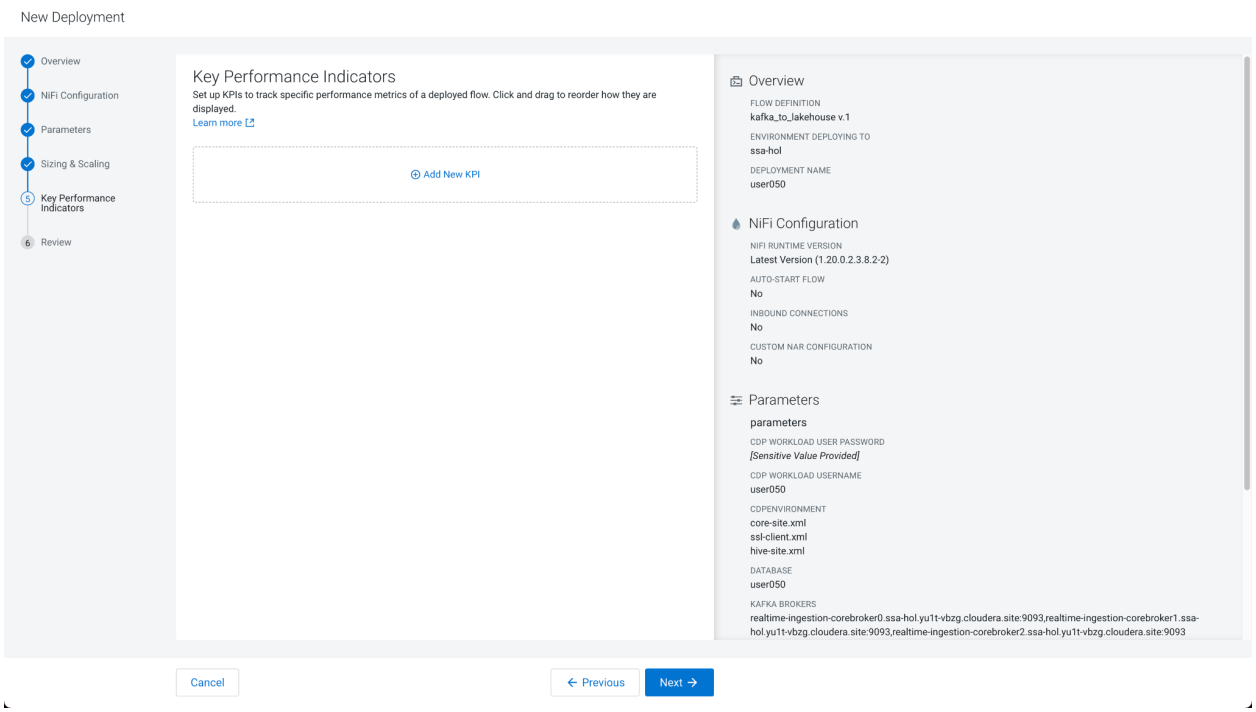




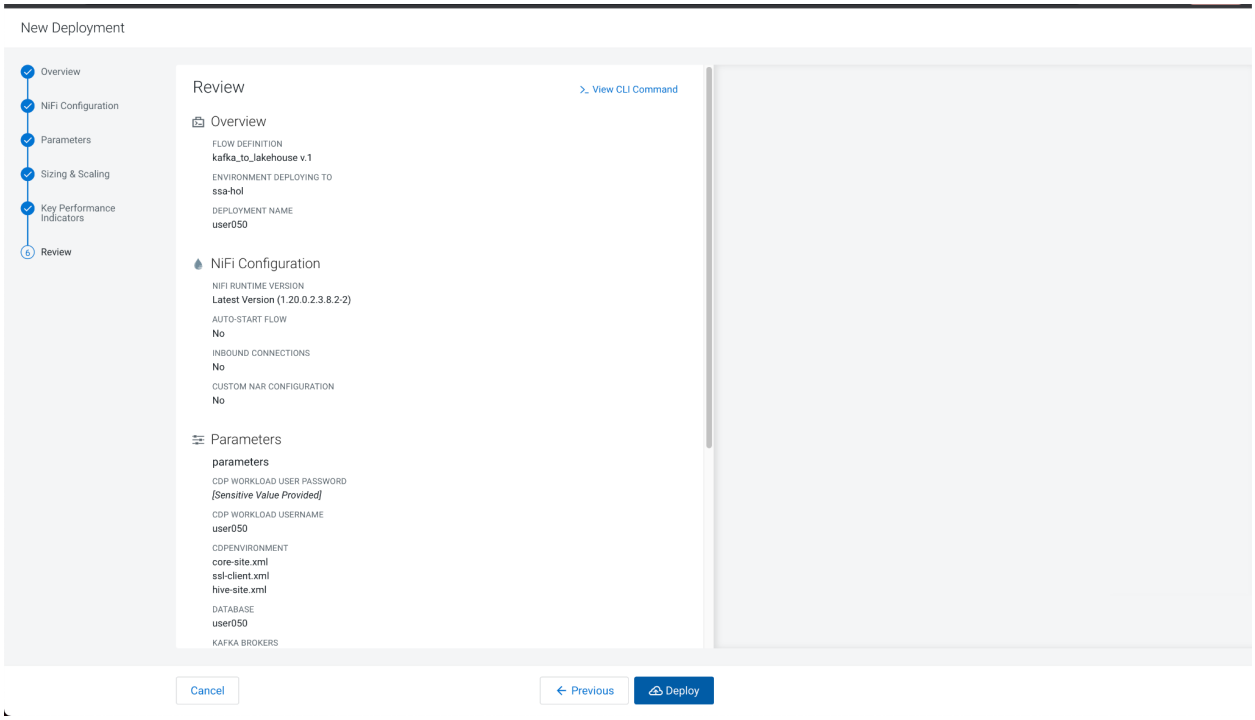
8. There is no need to configure auto scaling parameters, then click on **Next**



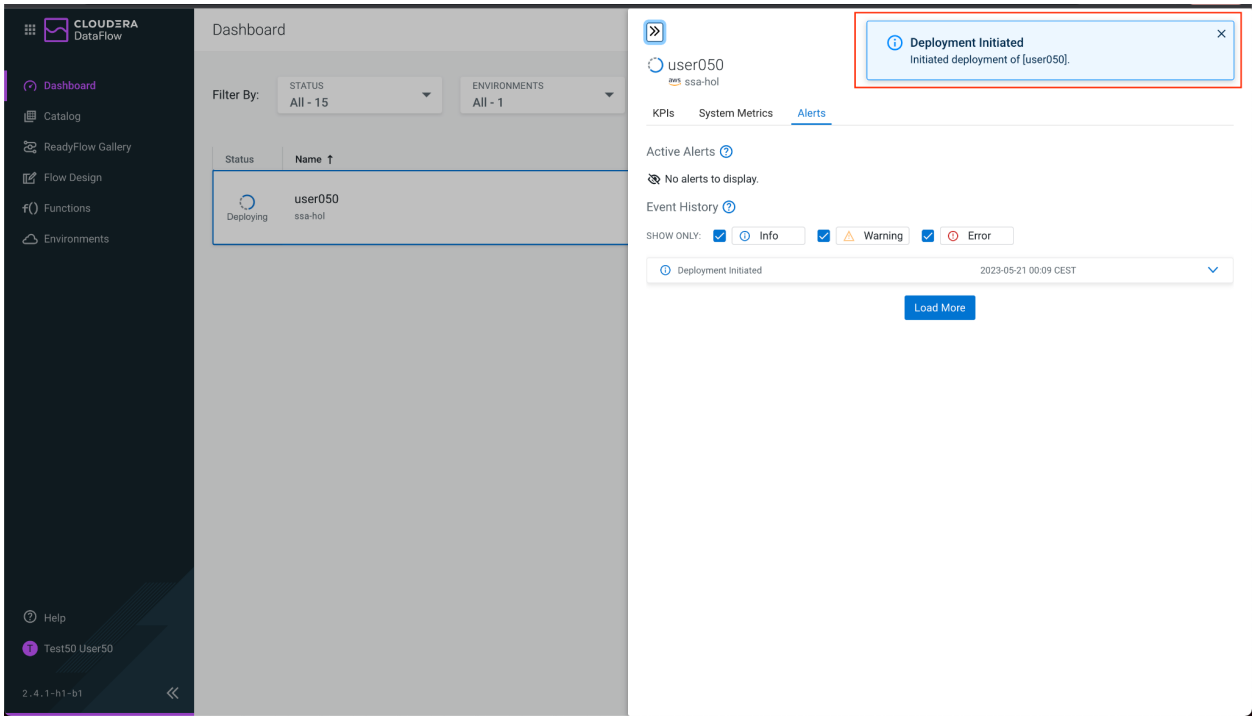
9. We are also not going to configure KPIs by now, then click on **Next** to continue the configuration.



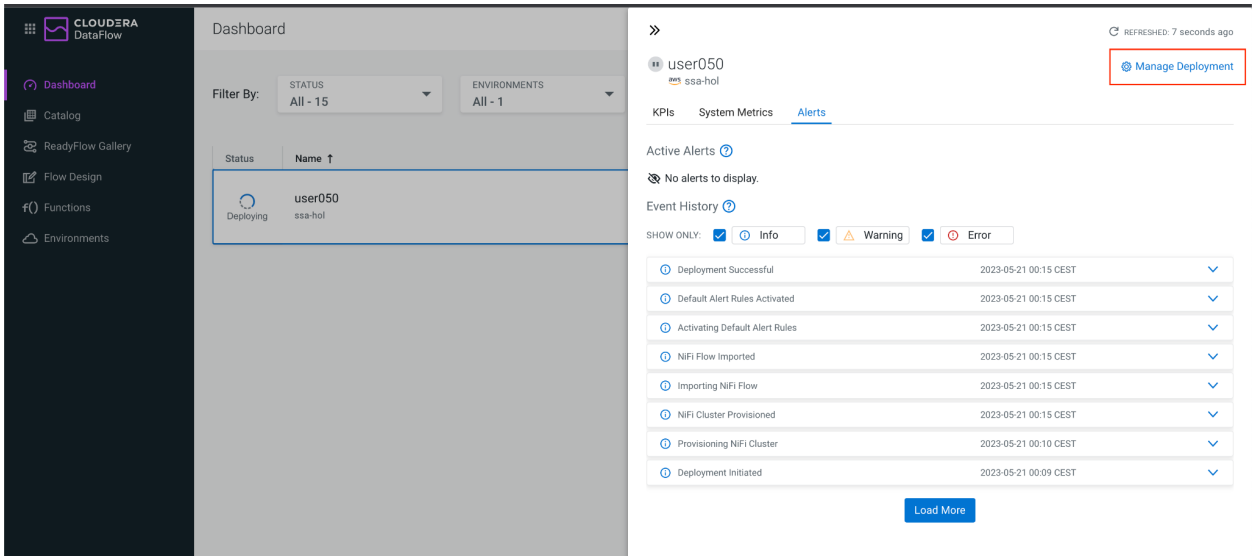
10. Review all the information entered for your Flow, then click on **Deploy** to start the deployment process.



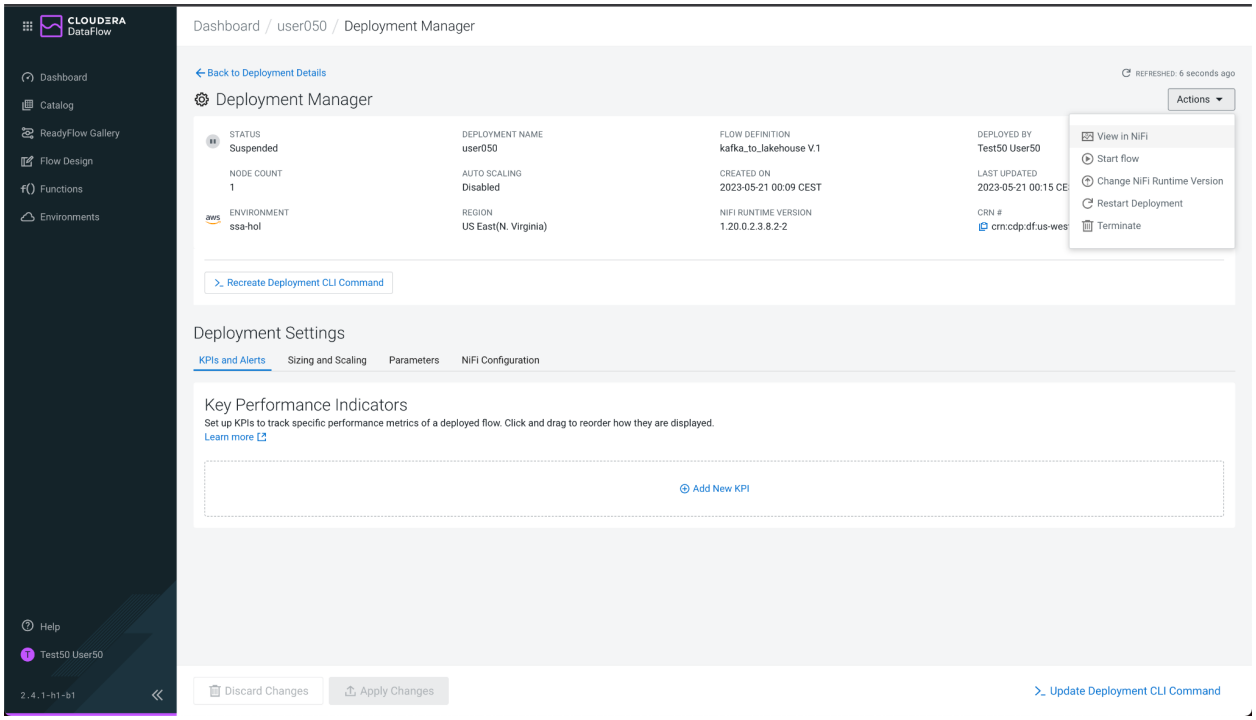
11. The blue box indicates that the Flow deployment process has been started. By clicking on the button **Load More** you will be able to see the different stages of the deployment. After about 60 to 90 seconds approximately, the last event should be *Deployment Successful*.



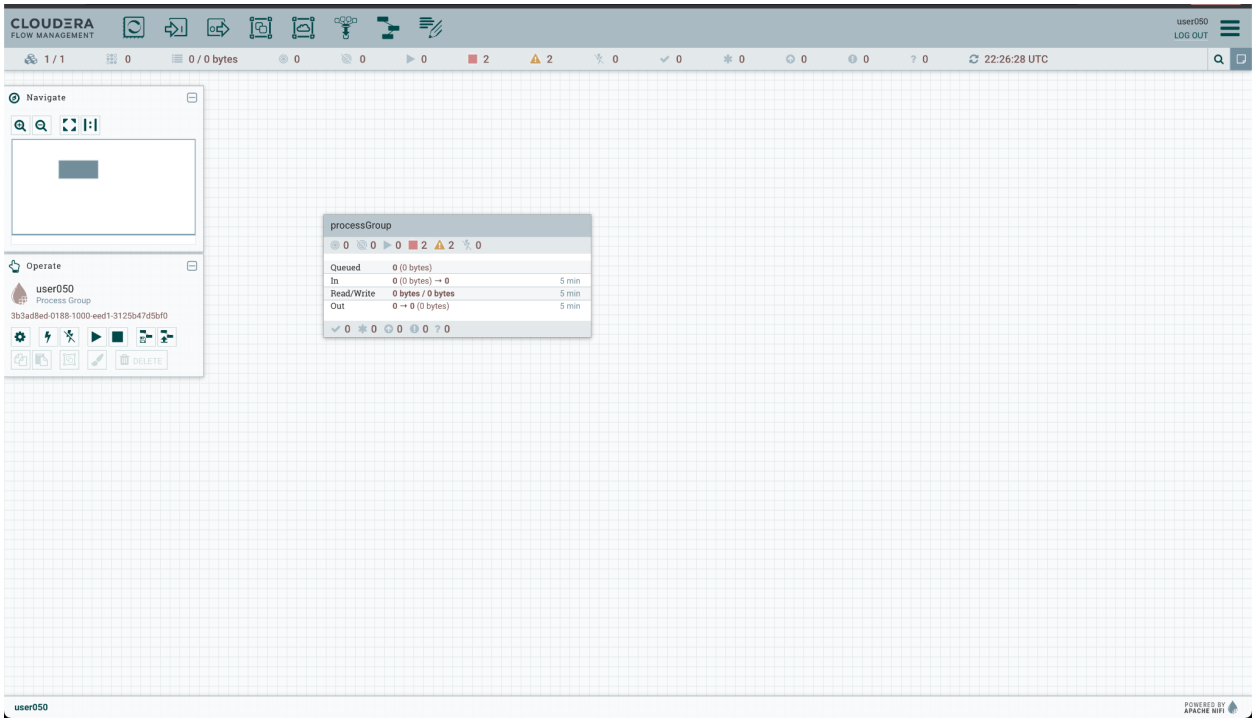
12. Once the deployment is finished, click on **Manage Deployment** to see the details of the recently deployed Flow.



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.



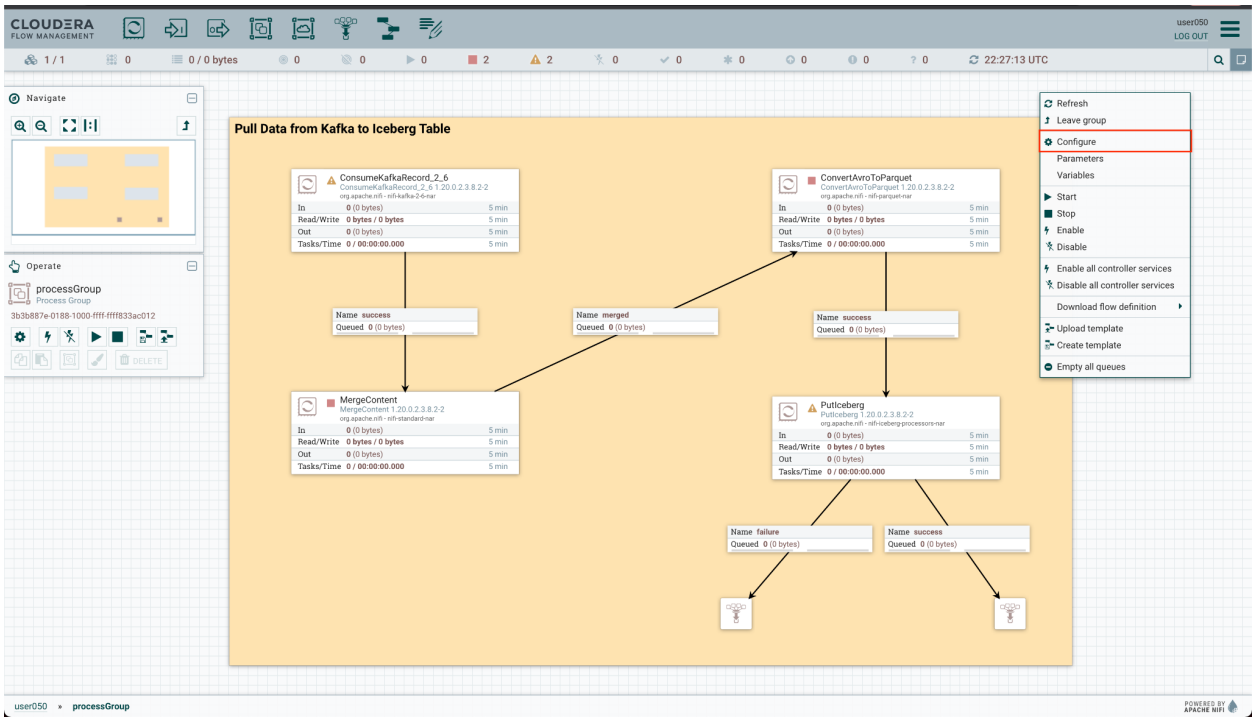
14. In the new window you should be able to see the Flow Management canvas with one process group (a box). The canvas is where the Flow Management applications are built. Double click on the box; the only visible box, which is a Process Group and should be titled **processGroup**.



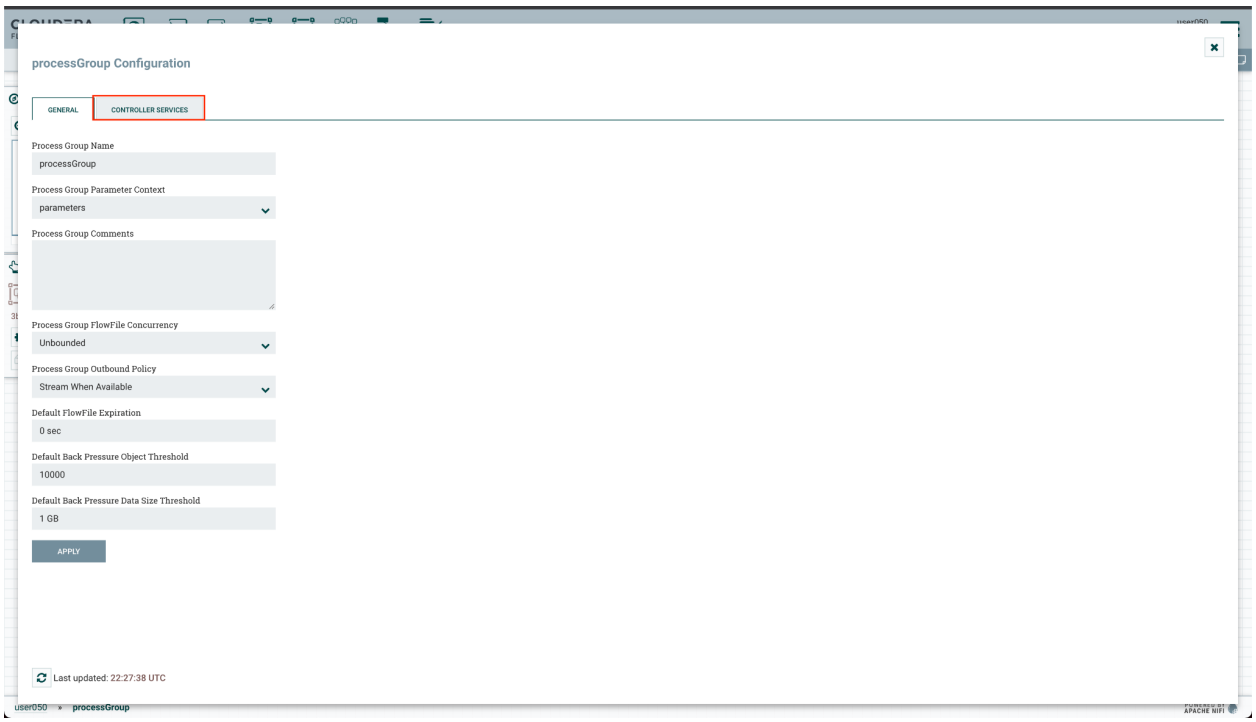
15. When opening the Process Group, you should be able to see the Processors that compose the Flow application. To summarize, there are four Processors:

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).

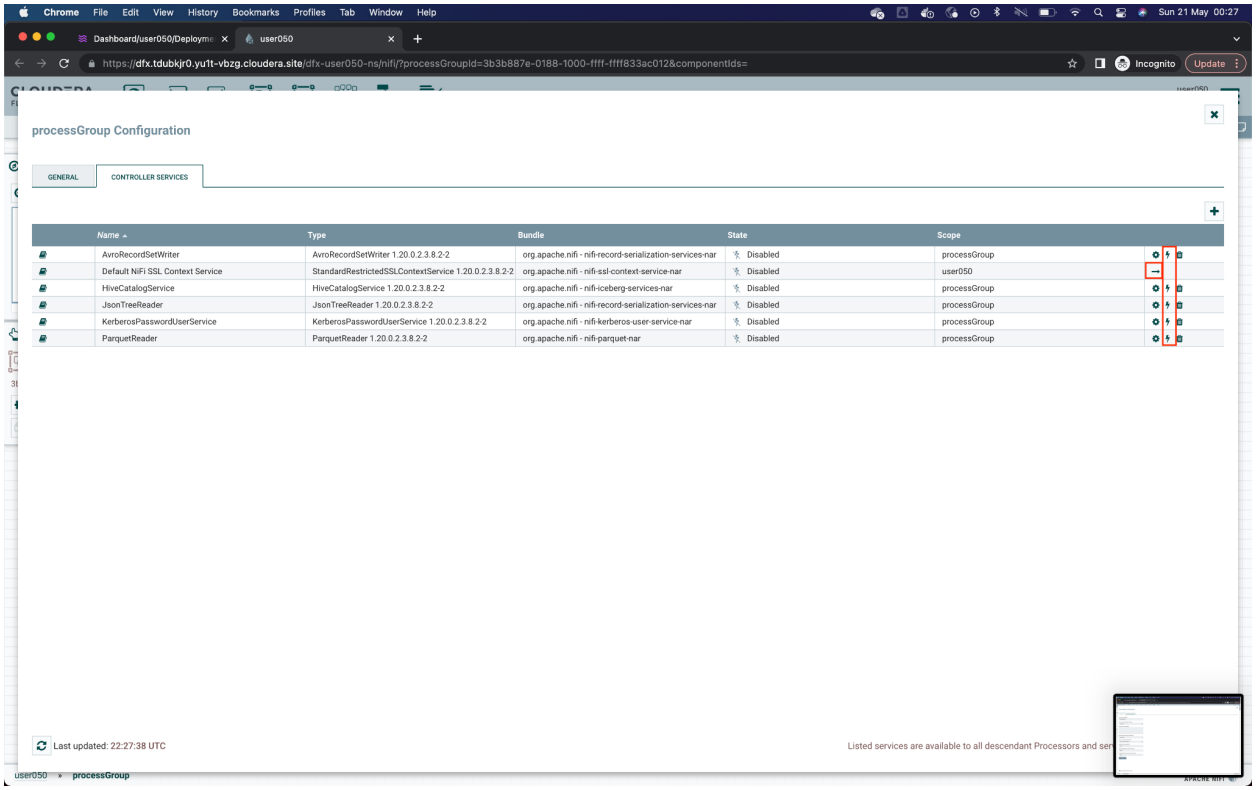
As you can see, the Processors are not started, and some have an error message/alert icon. The latter is because there are components of the data flow that must be activated before. To activate them - the *Controller Services* - right click on the canvas and click on the option **Configure** from the floating menu that appears.



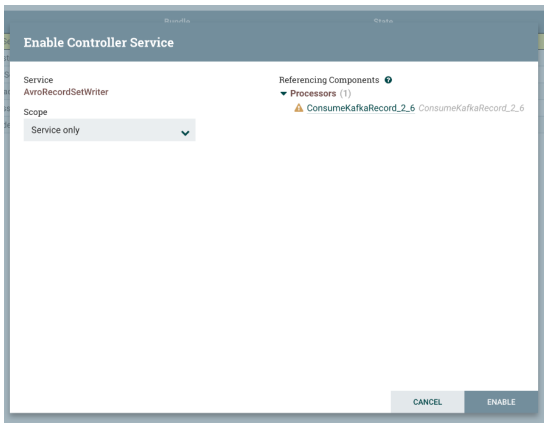
16. In the pop-up window that opens, select the tab**Controller Services**.



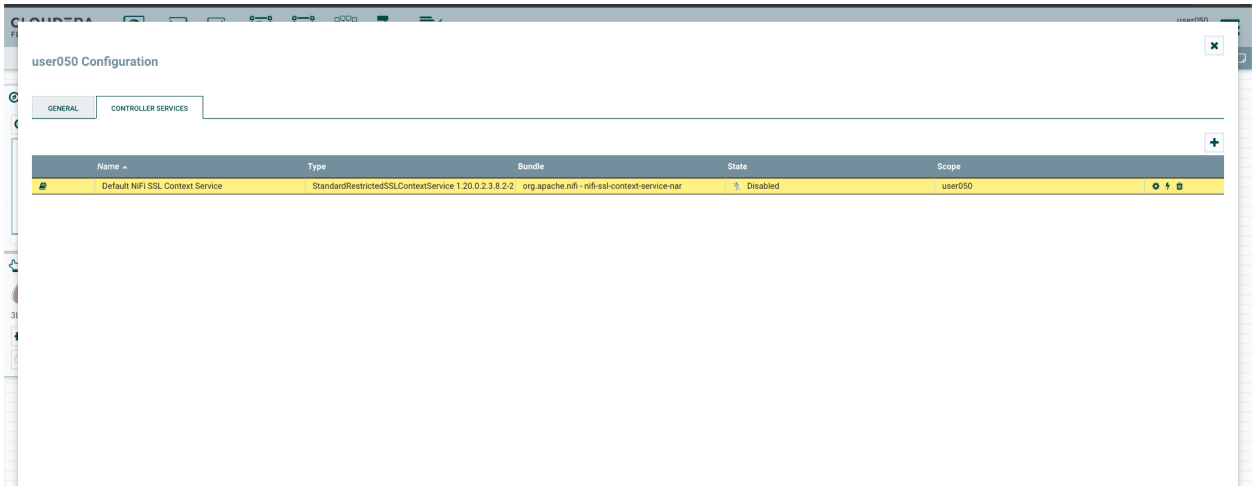
17. The**Controller Services** of the data flow. Each of them must be activated. The following Controllers must be activated first:**AvroReaderSetWriter**, **HiveCatalogService**, **JsonTreeReader**, **KerberosPasswordUserService** and**ParquetReader** clicking on the icon lightning which appears on the right (marked in red).



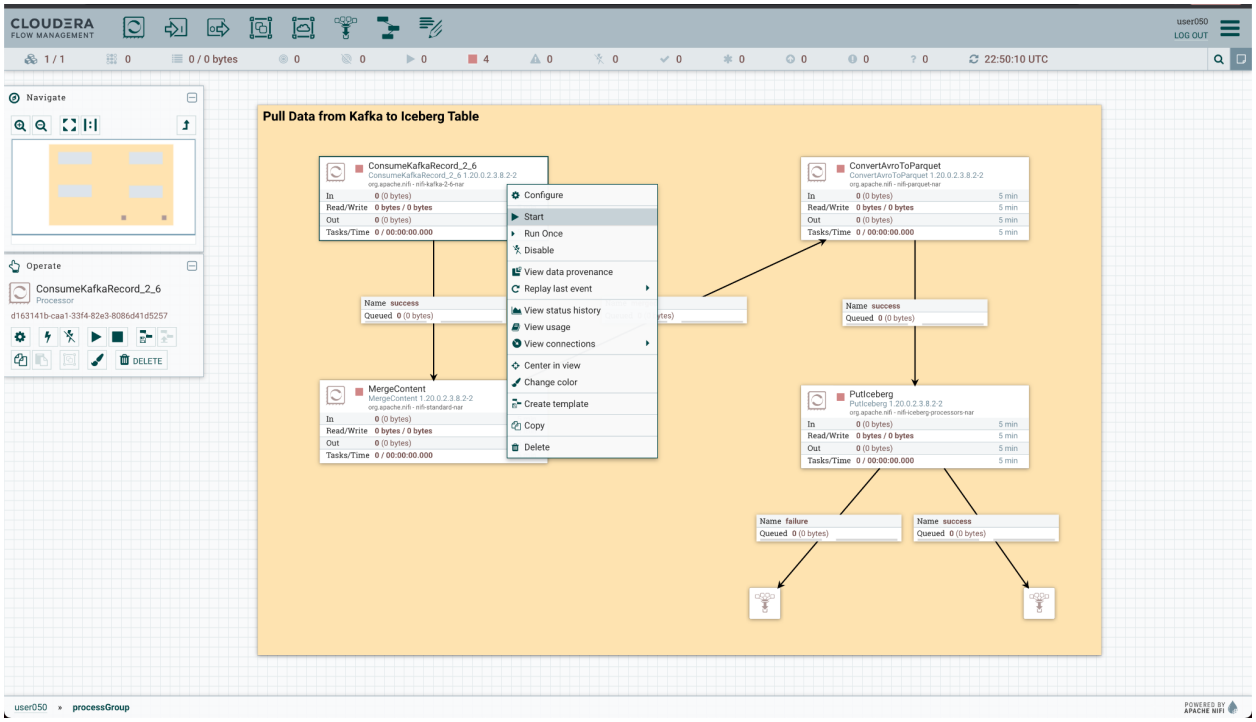
Click the button**Enable** in the enable confirmation window of each Controller Service. Then close that window to enable the next Controller Service.



To activate the Default NiFi SSL Context Service, you must click on the arrow →. Finally clicking on the lightning bolt icon ϟ controller service is activated**Default NiFi SSL Context Service,** which will also present a window to enable it.

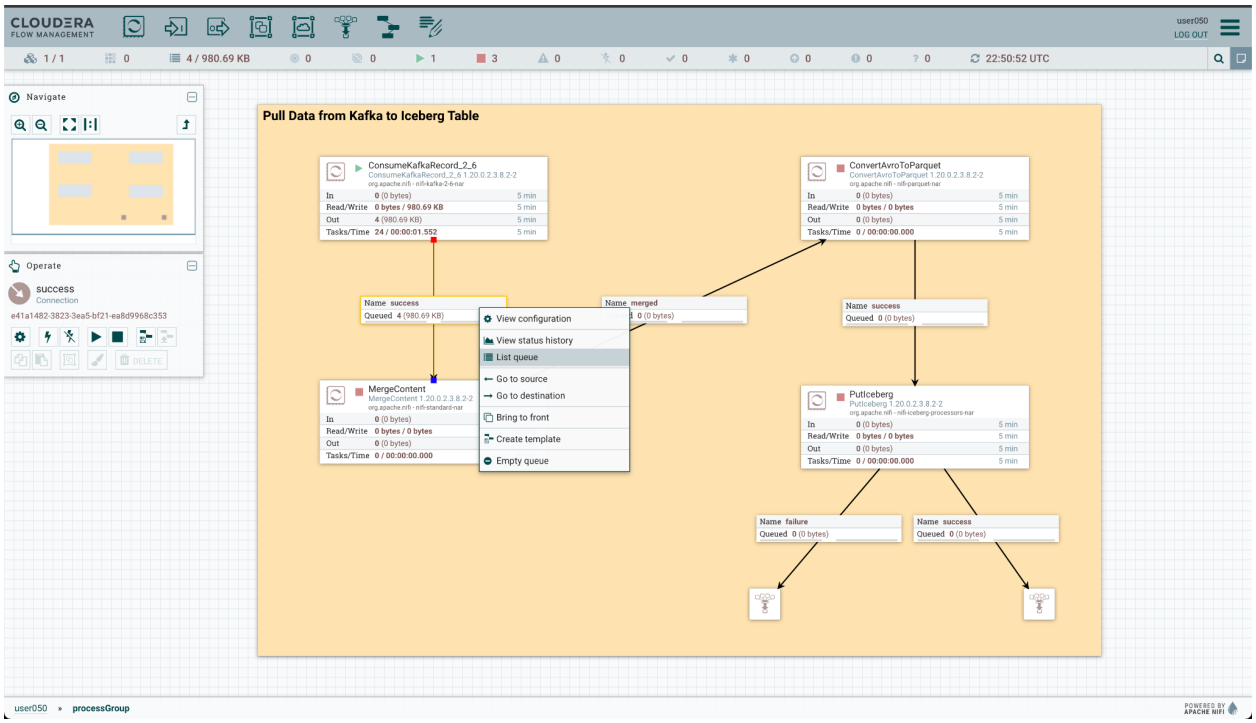


18. Close the Controller Services window, making sure all are enabled. Return to the Process Group by double-clicking on it. It's time to execute **Processors**. Start with ConsumeKafkaRecord, by right-clicking on it, and then clicking on **Start**. This will start consuming the Kafka topic data.

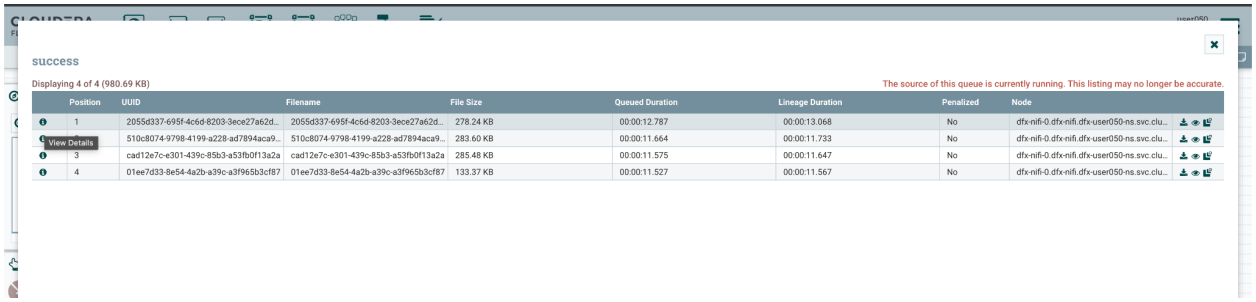


19. Flow Management allows us to see and access data in motion during the execution of the data flow. Between Processors **ConsumeKafkaRecord** (just started) and **MergeContent**, there is a connection. This connection is what joins the Processors and transmits data from one to the other.

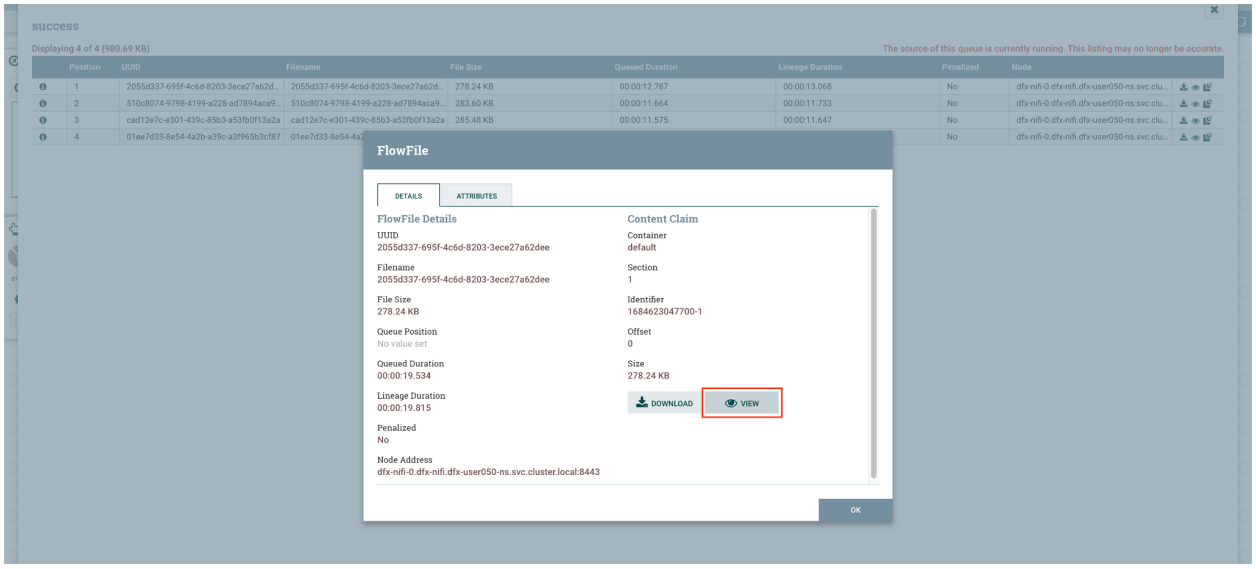
To check how much data is queued on this connection, refresh the counter by pressing the Ctrl+R (Windows) or Command+R (Mac) combination on the keyboard. This will allow the current metrics of the entire data stream to be updated. At some point there should be a number next to the legend **Queued** in 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.



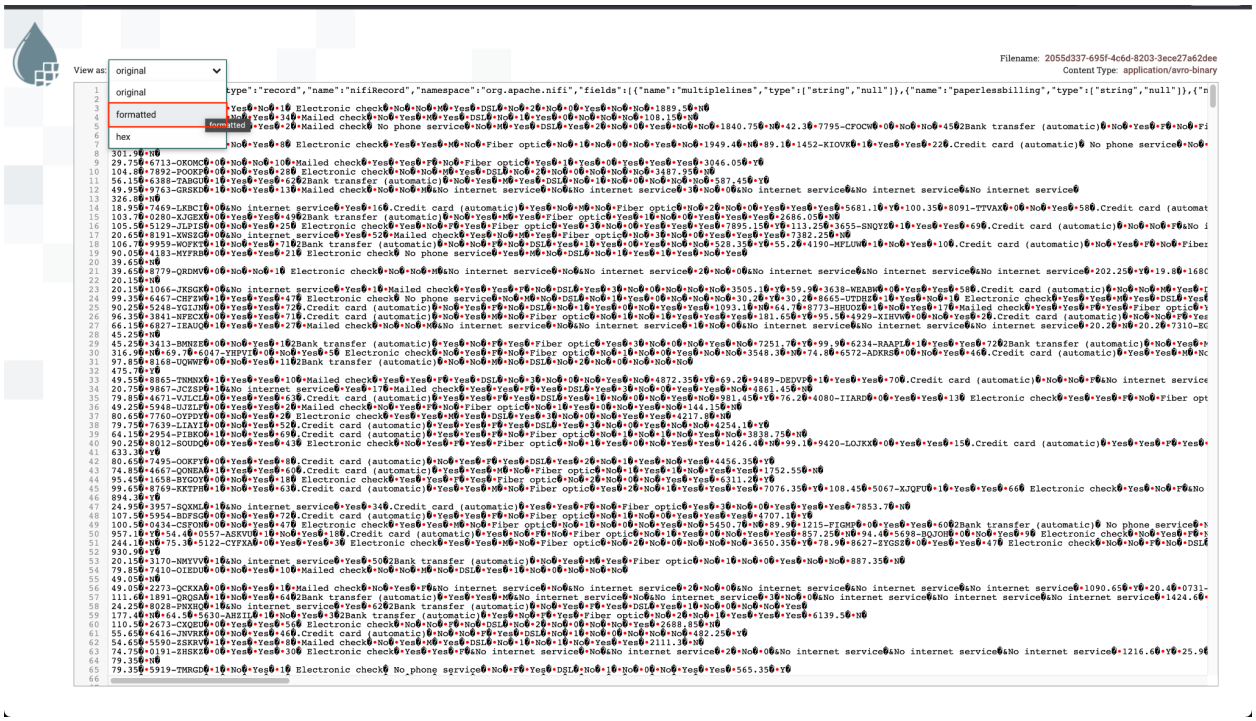
20. The next popup window lists the queued data. Click on the information icon (i) that appears on the left side to view the events.



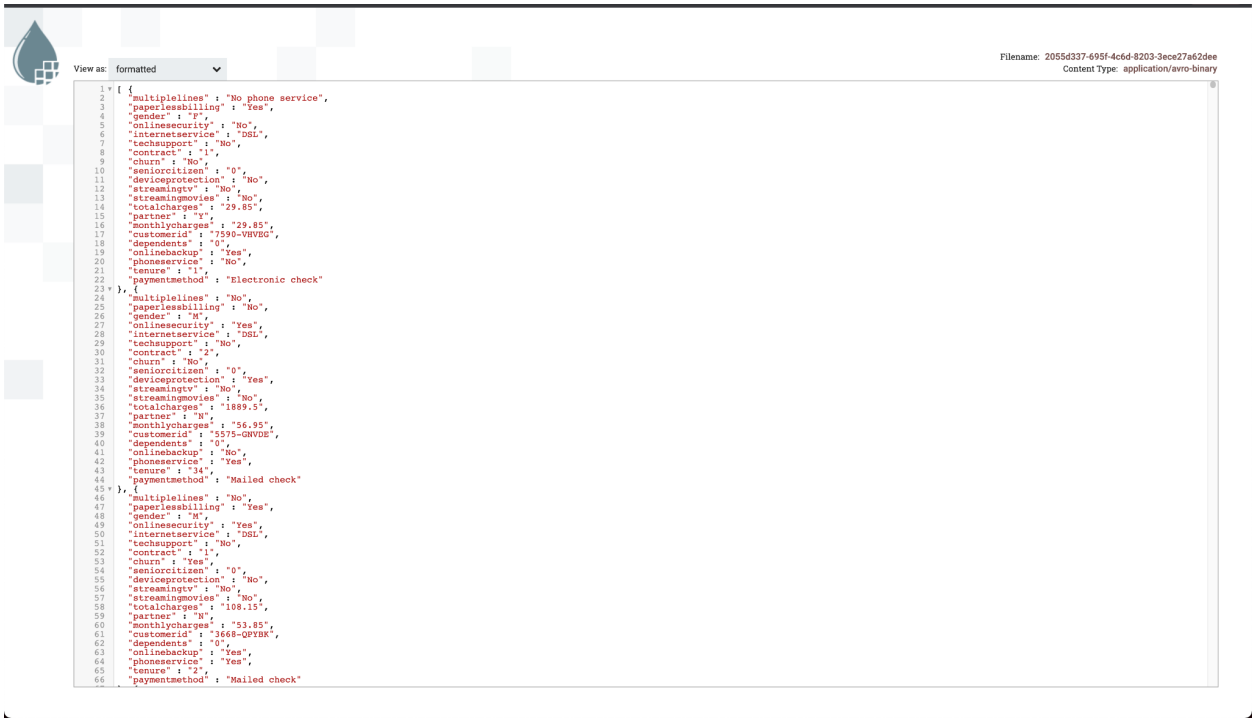
21. Once the FlowFile detail window appears, click on the button **VIEW** to open the content of consumed events.



22. 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**.

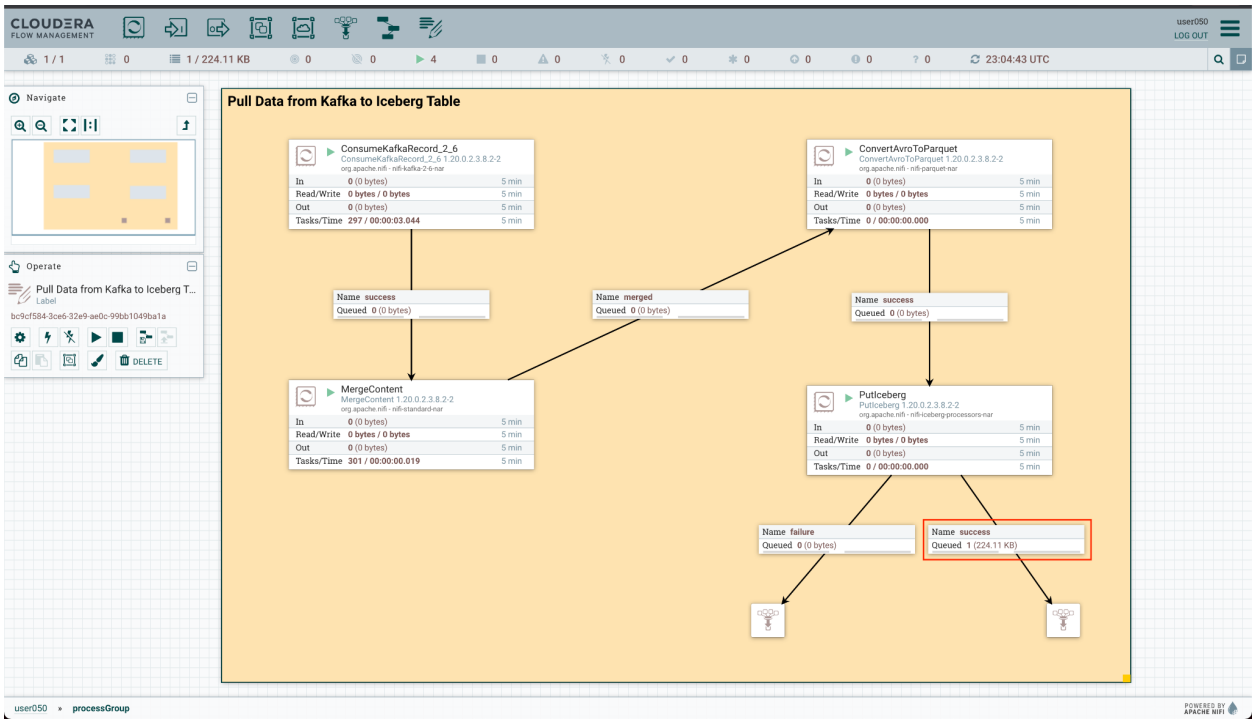


23. Now you can display the data correctly. Notice that the fields or attributes indicated at the beginning of the workshop appear. You can close that FlowFile window and the popups, returning to the canvas with the four Processors.



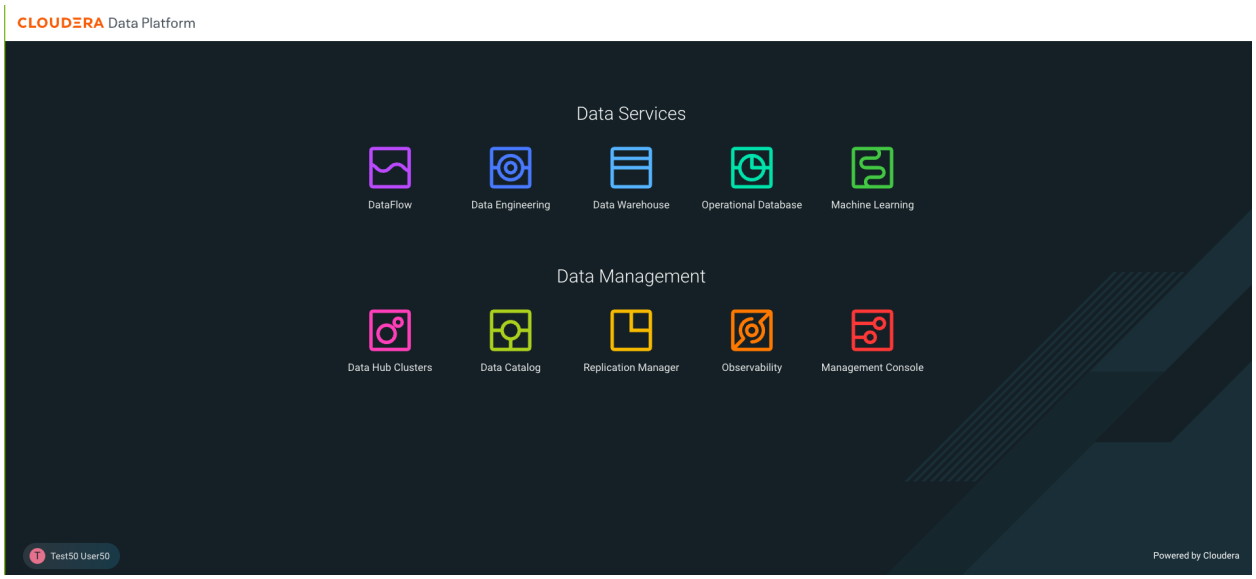
24. Continue running each of the Processors in order:**MergeContent**, after **ConvertAvroToParquet** and finally **PutIceberg**. Remember that you can refresh the flow counters with the combination Control+R or Command+R.

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

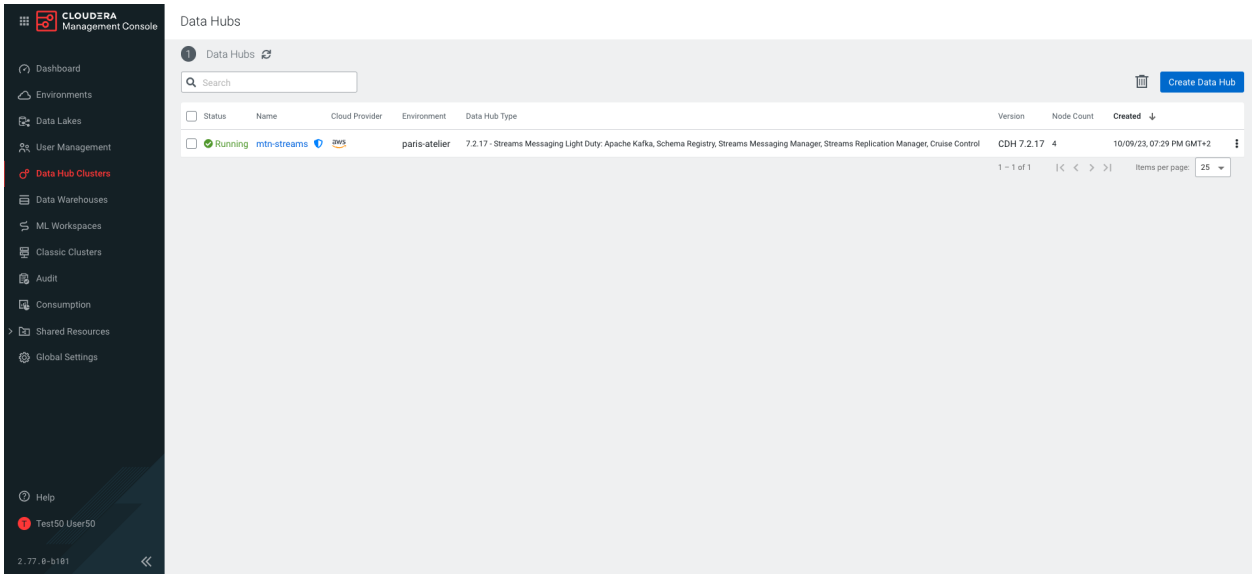


2.3. Lab 2 - Stream Messaging Manager - Optional

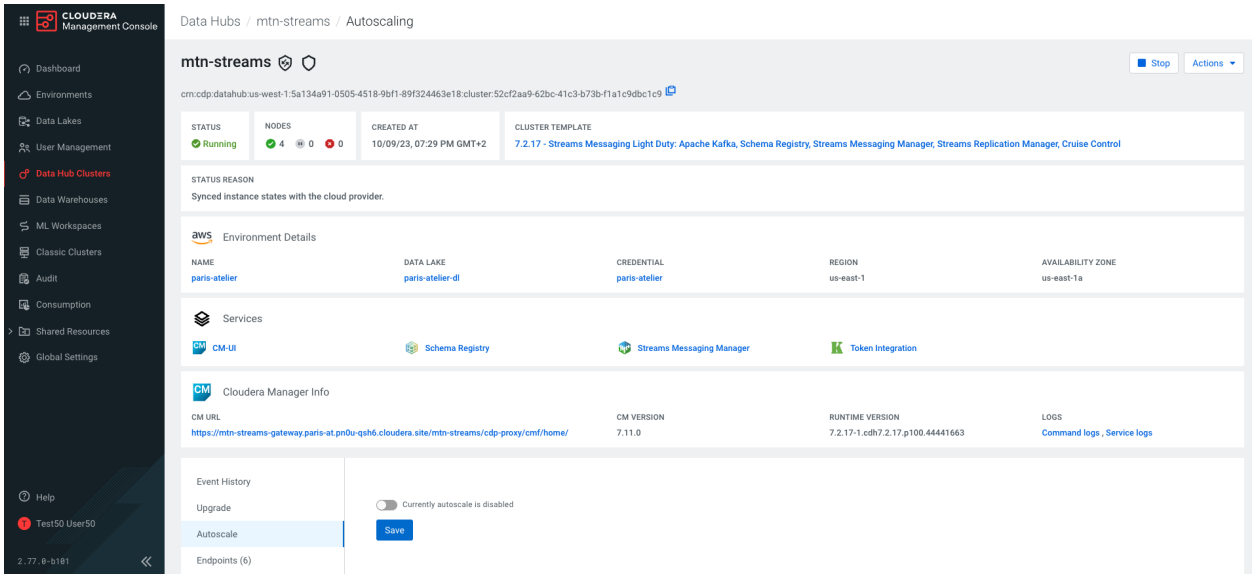
1. On your Cloudera Data Platform landing page,
    - Click on DataHub Clusters



2. The list of all your cluster loads
    - Click on mtn-streams



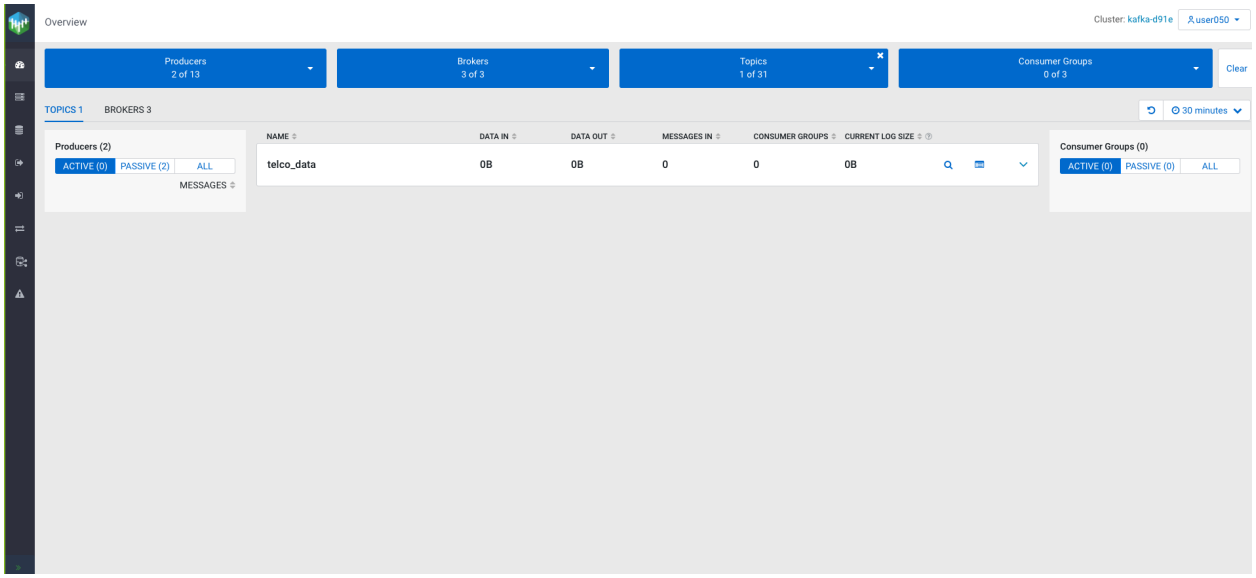
3. Your management page for the datalake loads. You can manage and scale your data lake, review history, endpoints, tags, telemetry.
    - Click on Streams Messaging Manager



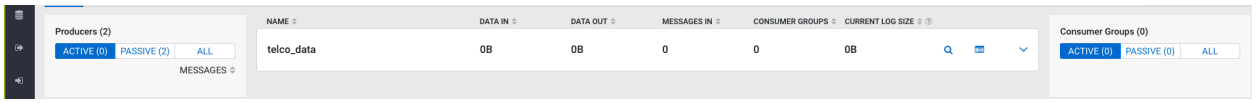
4. Your streams messaging page opens. It has a list of producers, brokers, topics, and consumer groups.
    - Click on topics

- Click on the topic: telco data

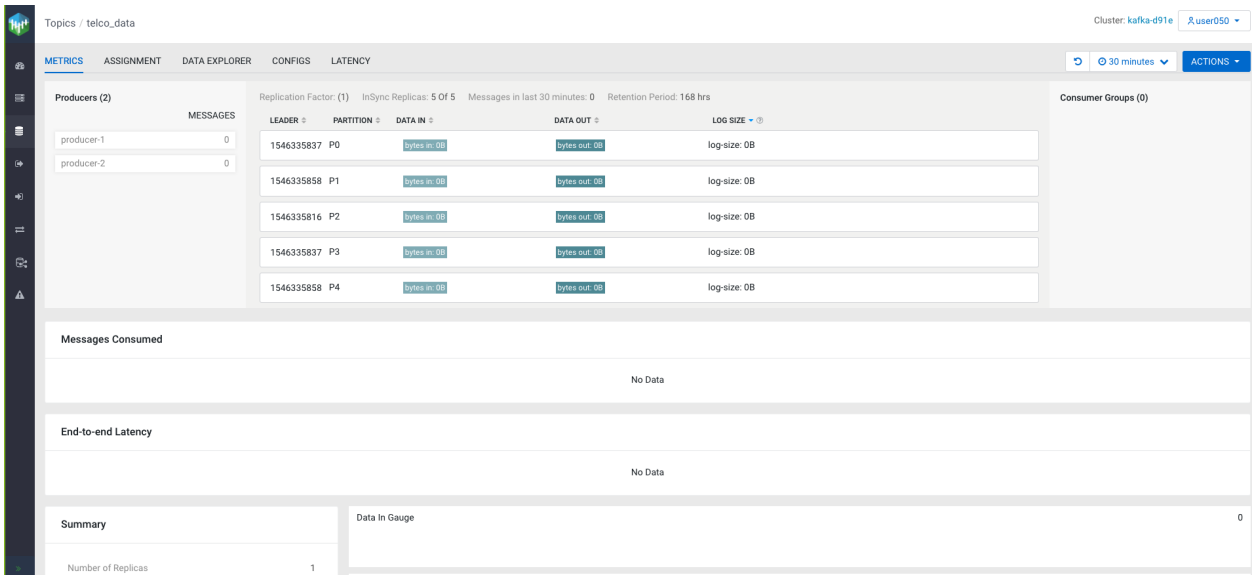
When loaded, this page will include all the data coming for your telco data CDRs.



- Then Click on explore (small icon to the right of the magnifying glass)



5. Your topics page loads, Click on Data Explorer



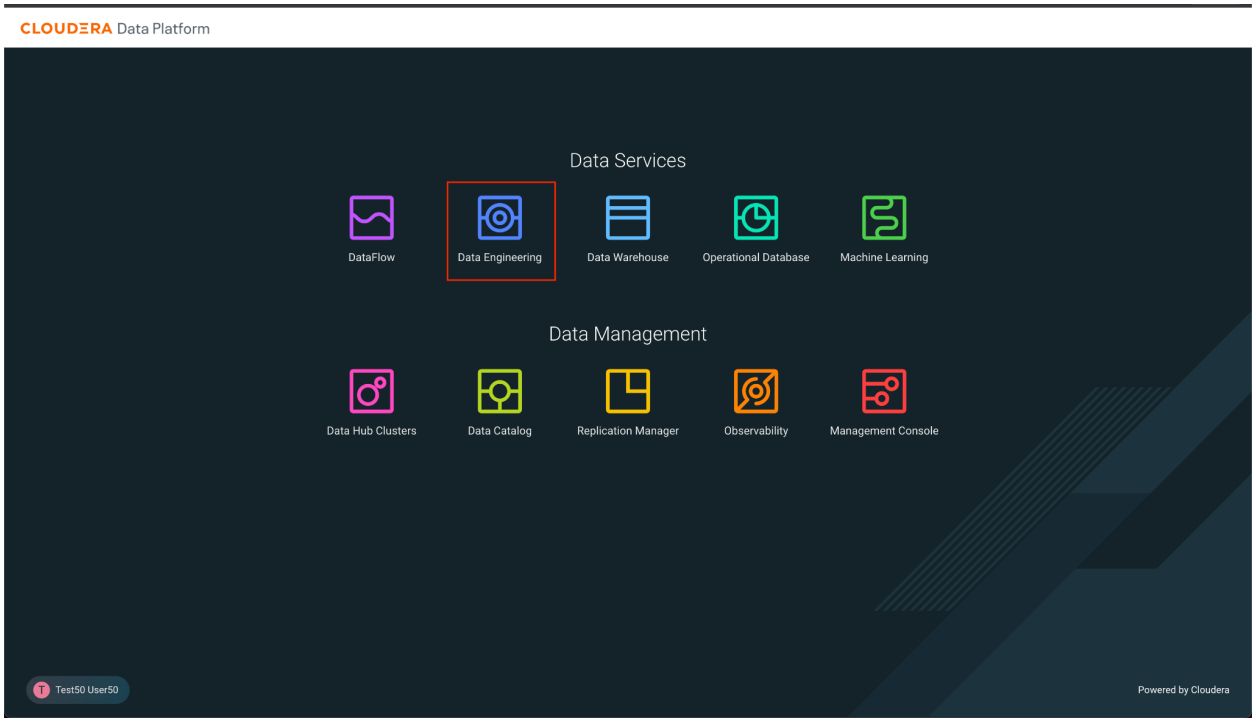
3. Data Engineering

3.1. Goals

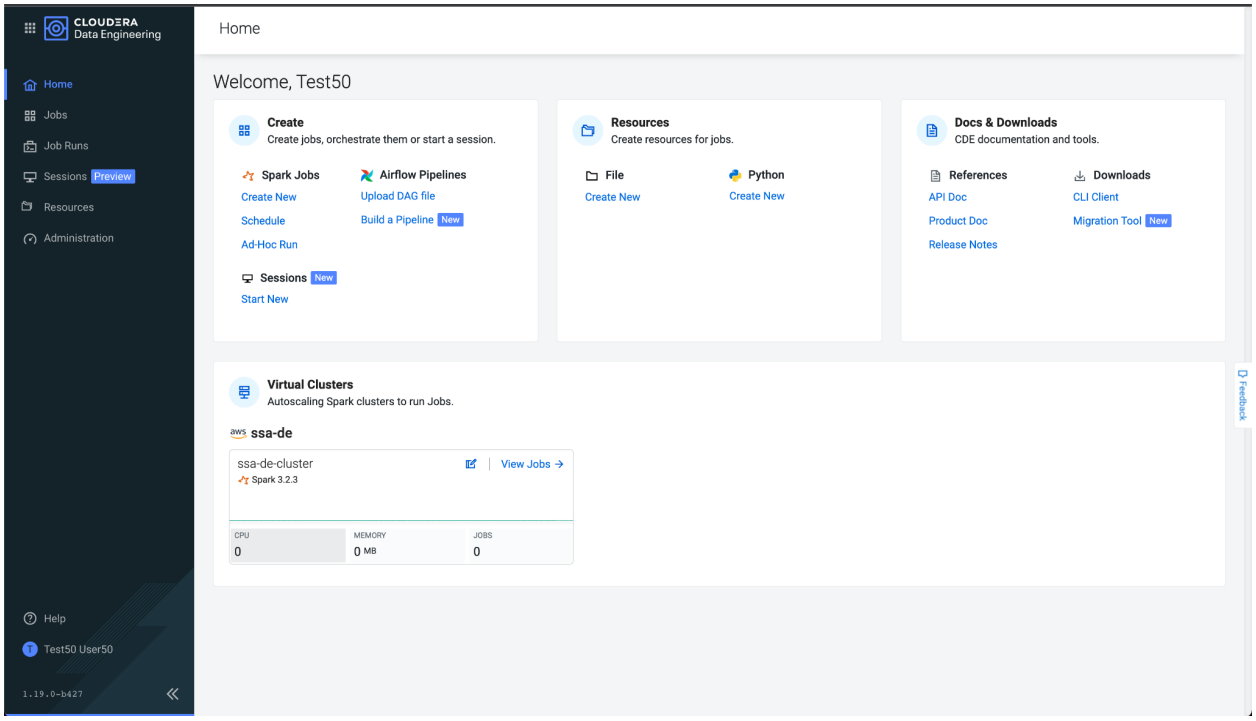
- Run a data enrichment process
- Run a process to simulate changes to the data
- Configure the execution of a pipeline using low-code/no-code tools

3.2. Lab 1 - Enrich the Ingested Iceberg table

1. Click on DataFlow from CDP PC Home:



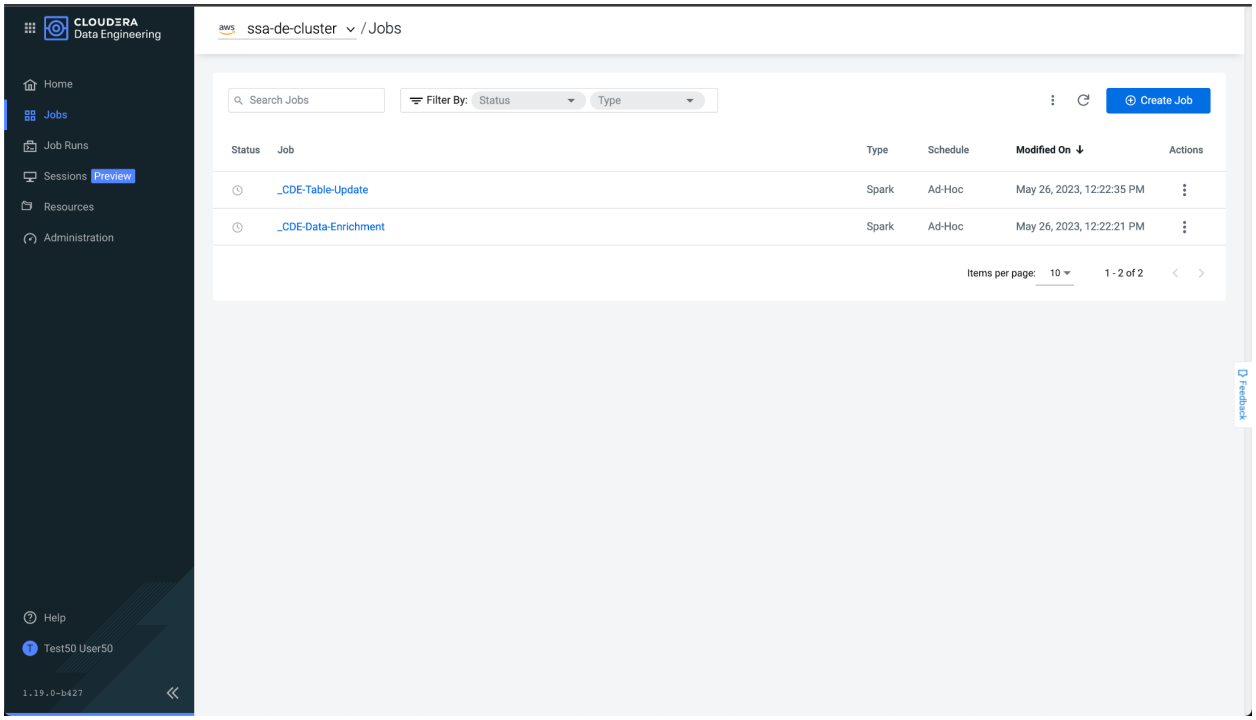
2. The Data Engineering Home shows all the actions that can be done, such as Jobs in Spark and pipelines in Airflow, Resources and useful information/documentation. Click on the option **Jobs** from the left menu to create a dataflow in Airflow.



3. Here the available tasks are listed. For the purposes of this workshop, two Jobs have been configured:

- **CDE-Table-Update**, generate random changes and enrich table to visualize Lakehouse Time Travel functionality.
- **CDE-Data-Enrichment**, process in Spark (Python) to enrich the data ingested from Kafka and save to a new table.

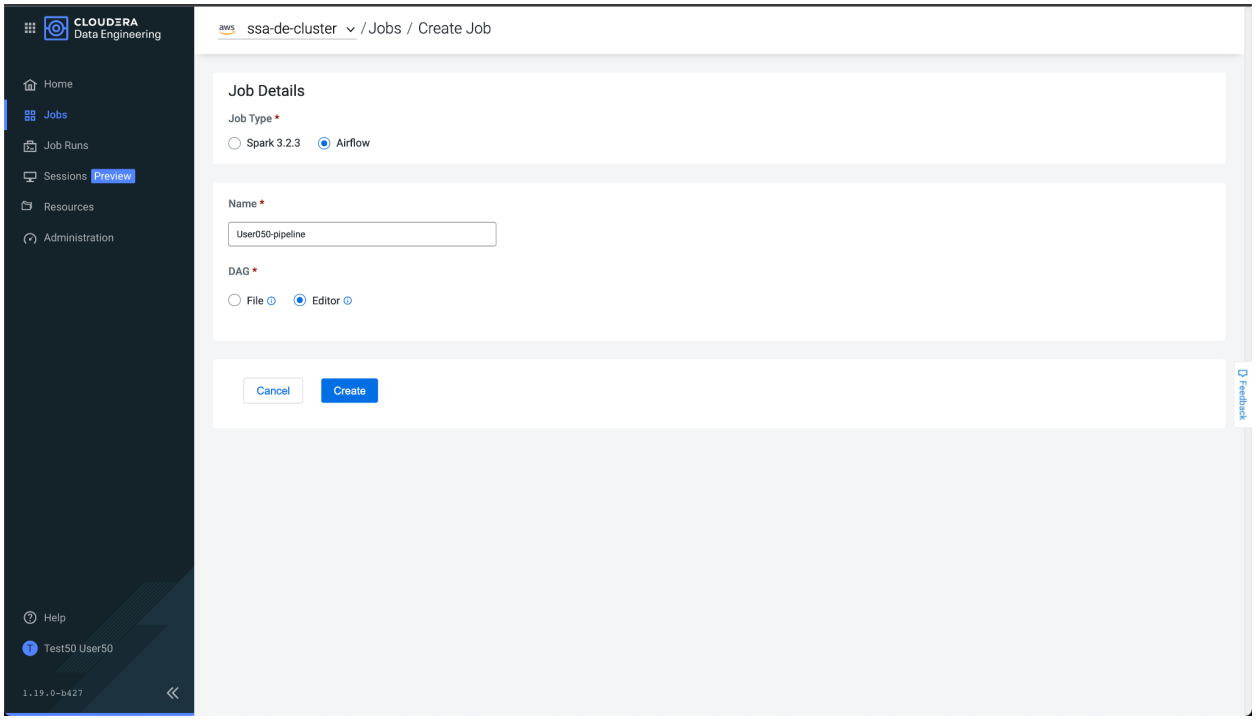
It is time to create our Job in Airflow. Click on **Create Job.**



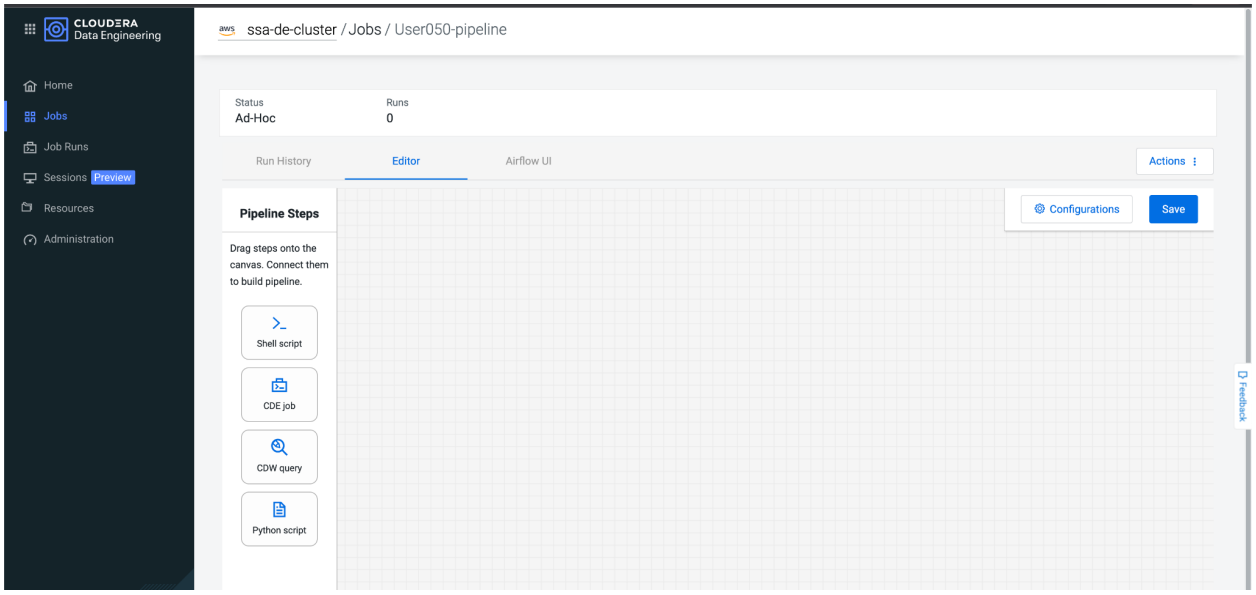
4. In the Job creation form, you must enter the following information:

- Job Type: Airflow
- Name: Use the naming <assigned user>-pipeline. Replace <assigned user> with the user assigned to you. For example, user050
- DAG: Editor, to graphically configure the task.

Once entering the values correctly, click on**Create**.

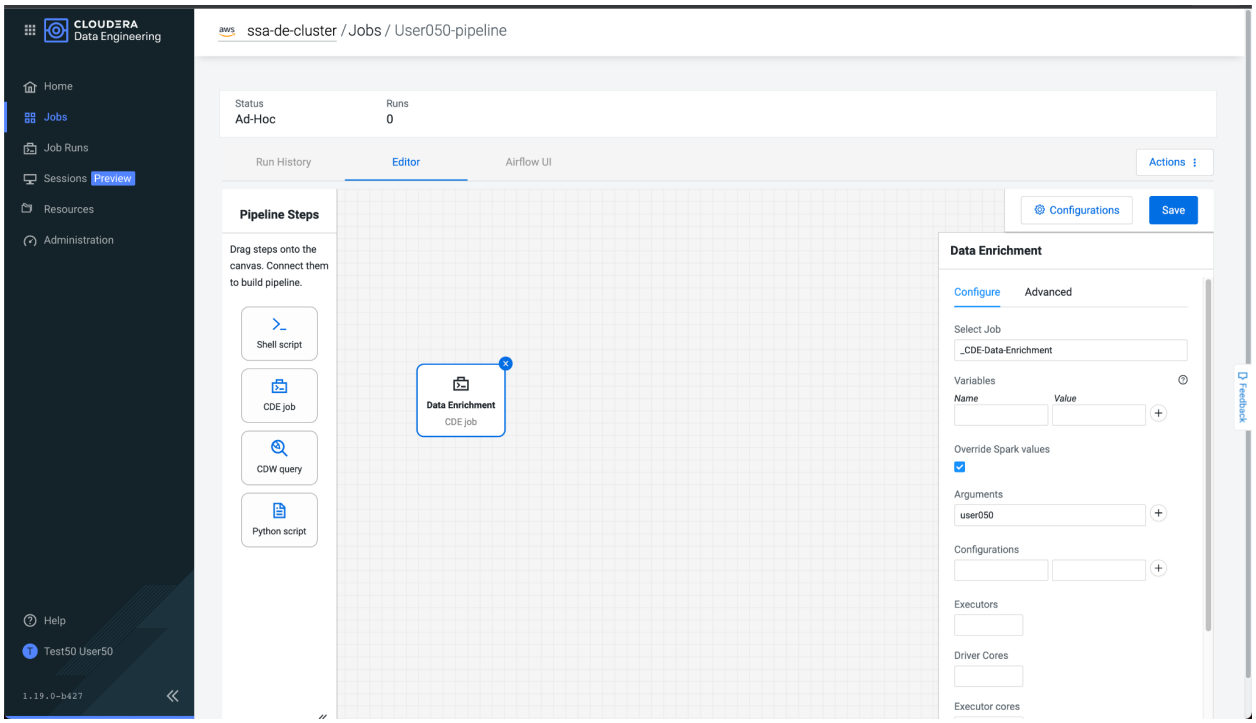


5. On the Job editing screen, select the Editor tab, and you will see the following canvas to drag the steps of the pipeline that we are going to create. In our case, we are going to create two CDE Jobs and relate them.



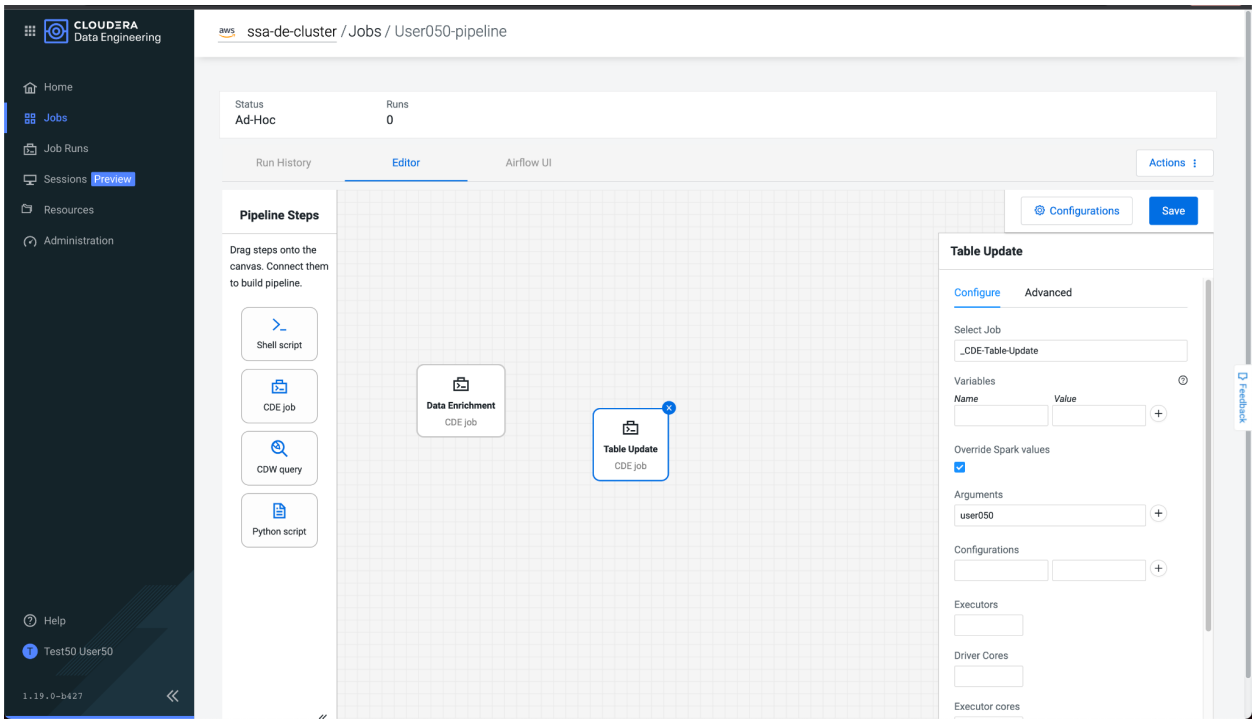
6. Let's start with the first Job. Click on the CDE Job button and drag onto the canvas, entering the following settings:

- **title/name**: Data Enrichment
- **Select Job**: select the Job_*CDE-Data-Enrichment*
- Check the checkbox **Override Spark values**. Additional options will appear below.
- **Arguments**: <assigned user>. Use the username assigned to you. For example, user050

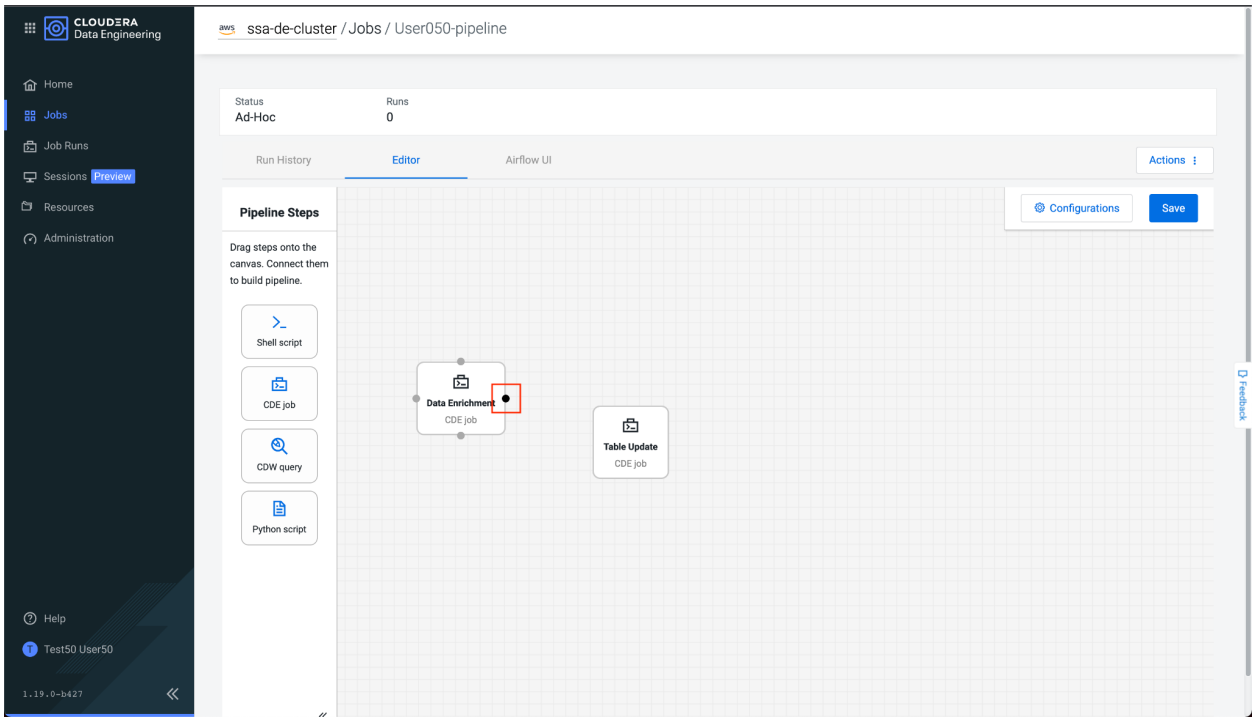


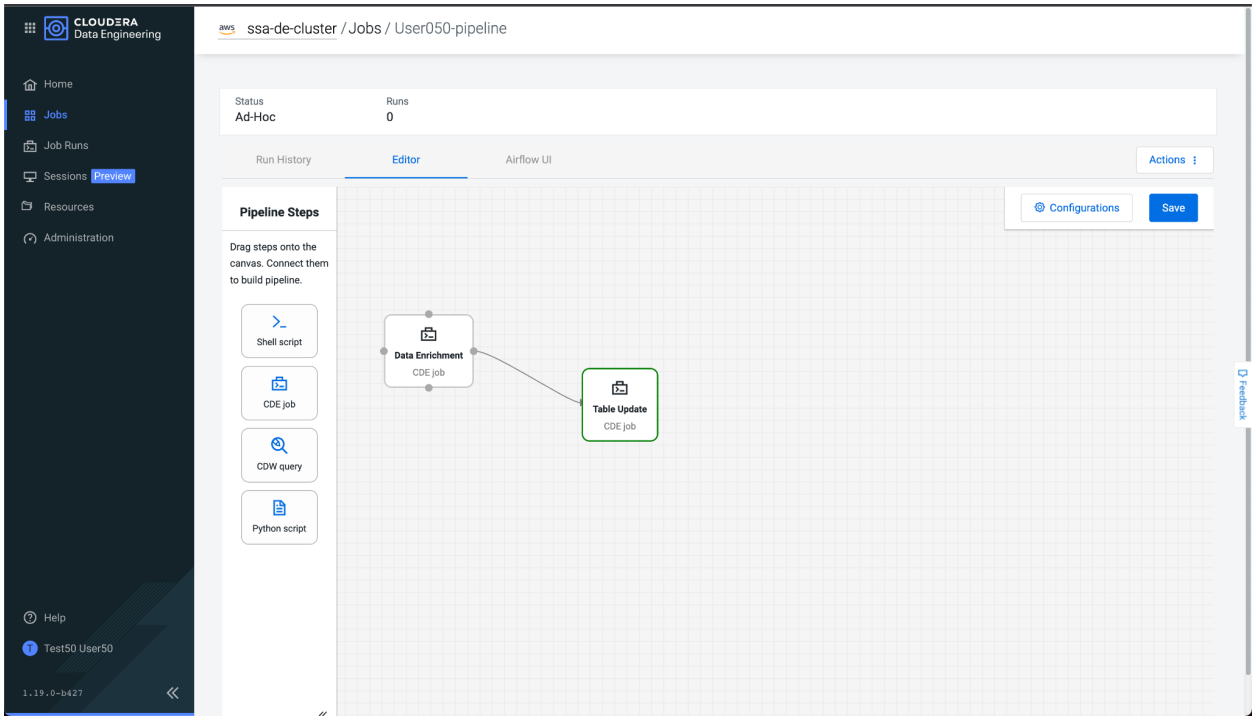
7. Configure the second Job. Click on the CDE Job button and drag onto the canvas, entering the following settings:

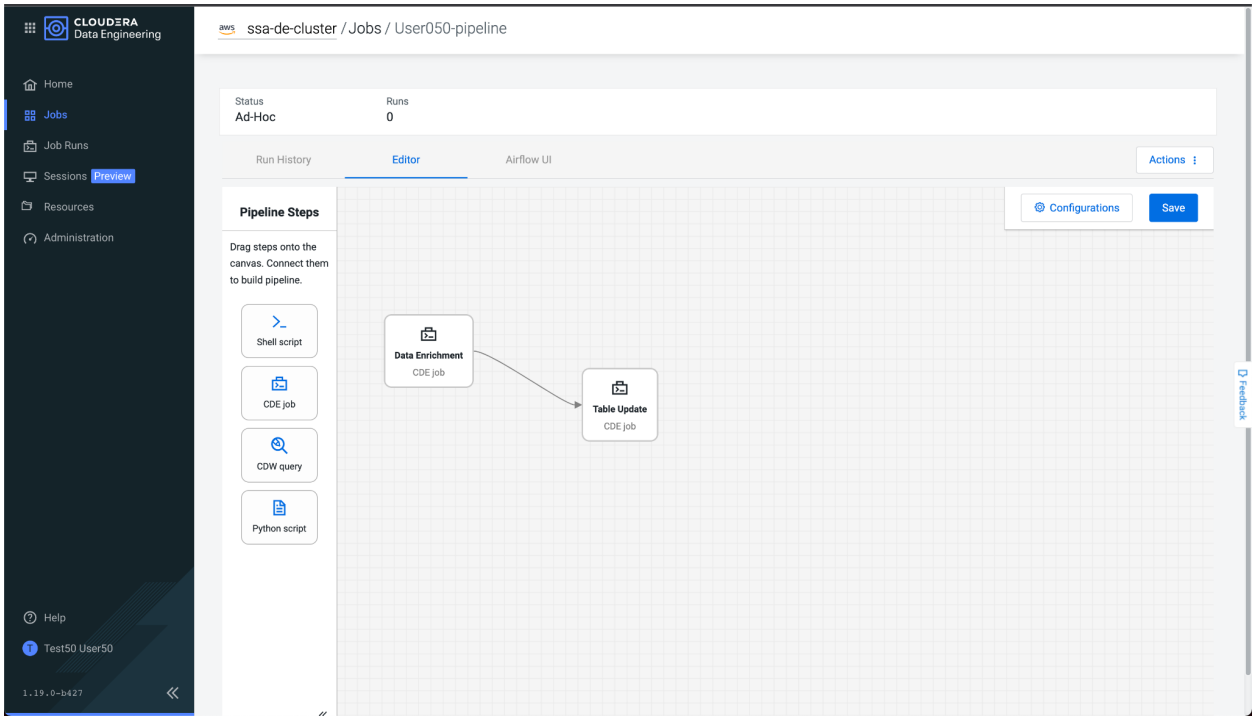
- **title/name**: Table Update
- **Select Job**: select the Job_*CDE-Table-Update*
- Check the checkbox **Override Spark values**. Additional options will appear below.
- **Arguments**: <assigned user>. Use the username assigned to you. For example, user050



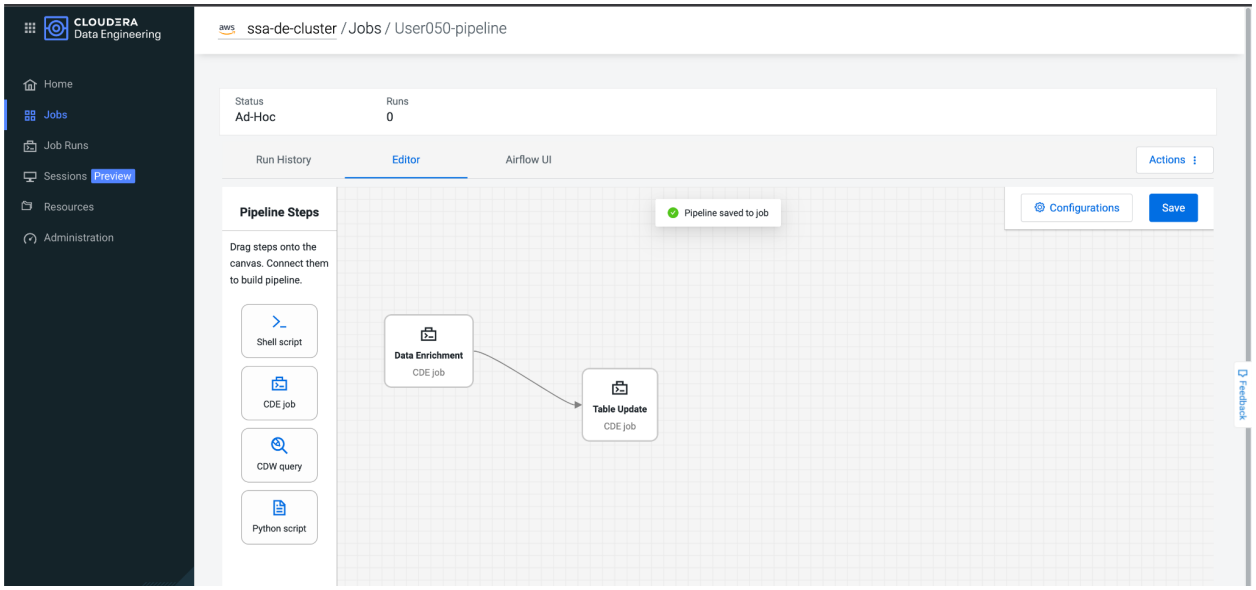
8. To set up the execution sequence, bind **Data Enrichment** with **Table Update**. For that, click on the right connector of the job of **Data Enrichment** and drag to the left connector of **Table Update**.



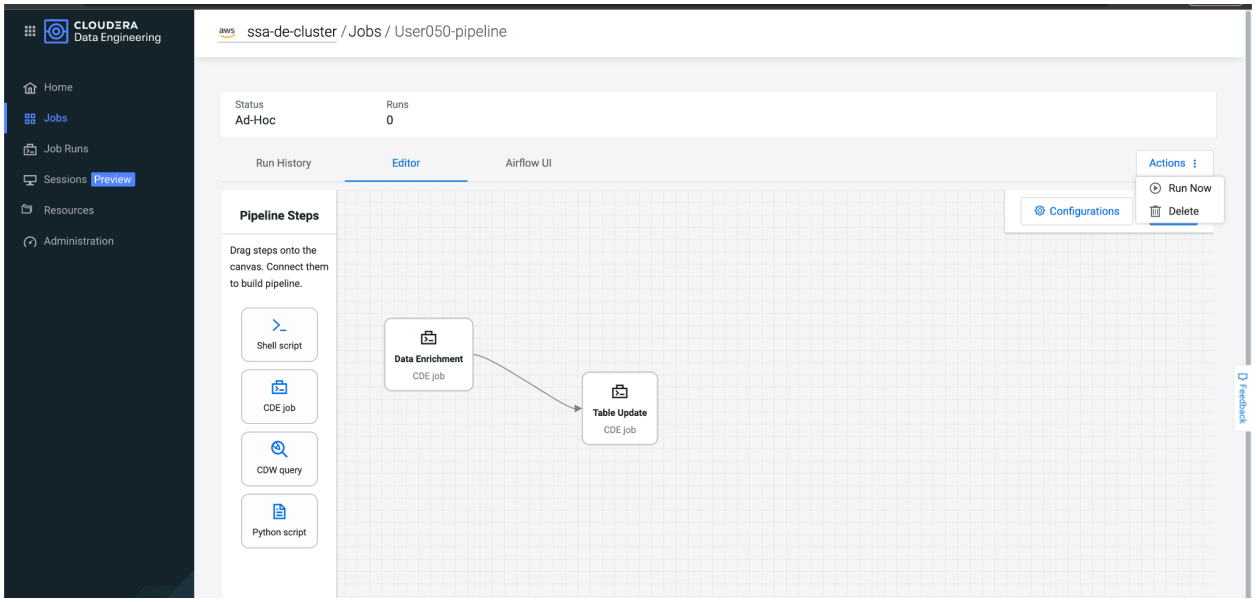




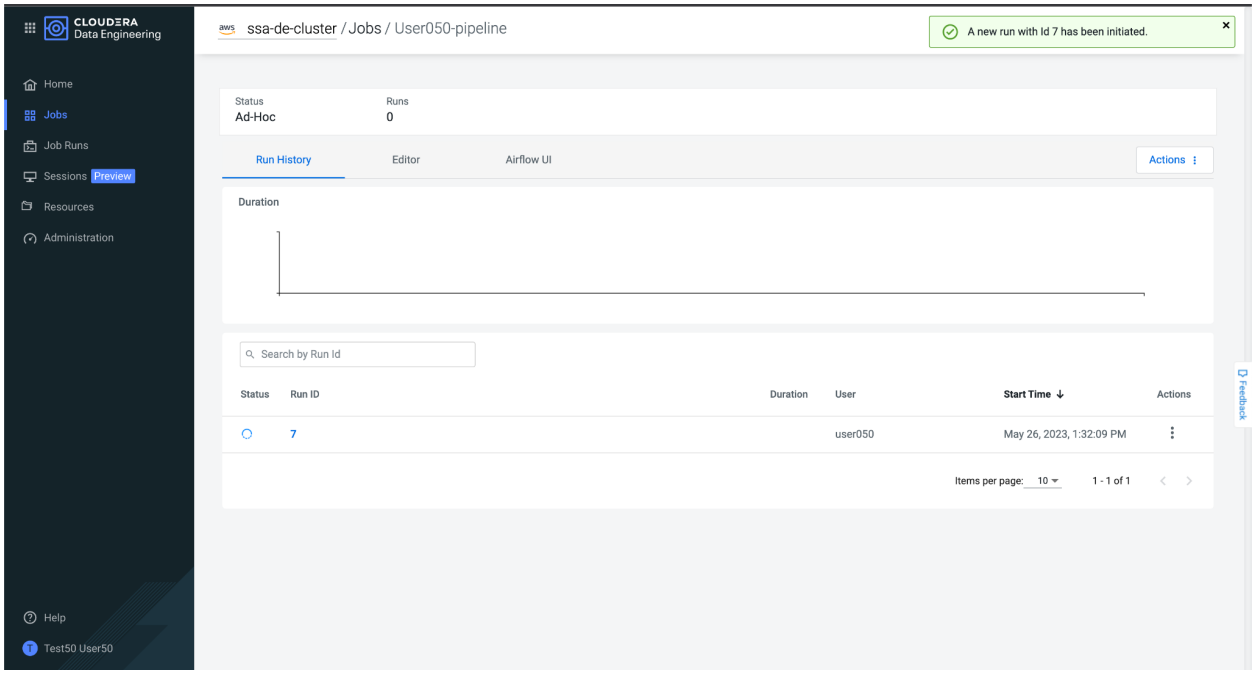
9. Once the Jobs have been joined, click on **Save** to save the settings made. You should see a message indicating **Pipeline saved to job**.



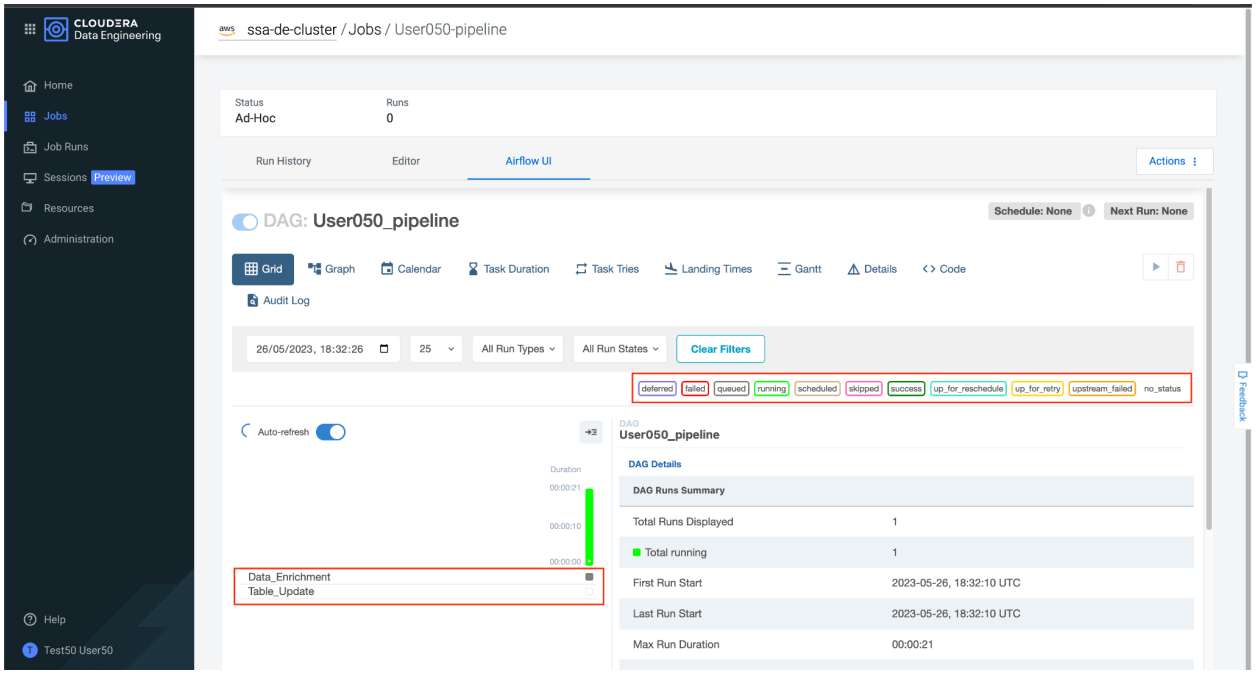
10. The time has come to run the pipeline. On the upper right side of the canvas, click **Actions -> Run Now**.



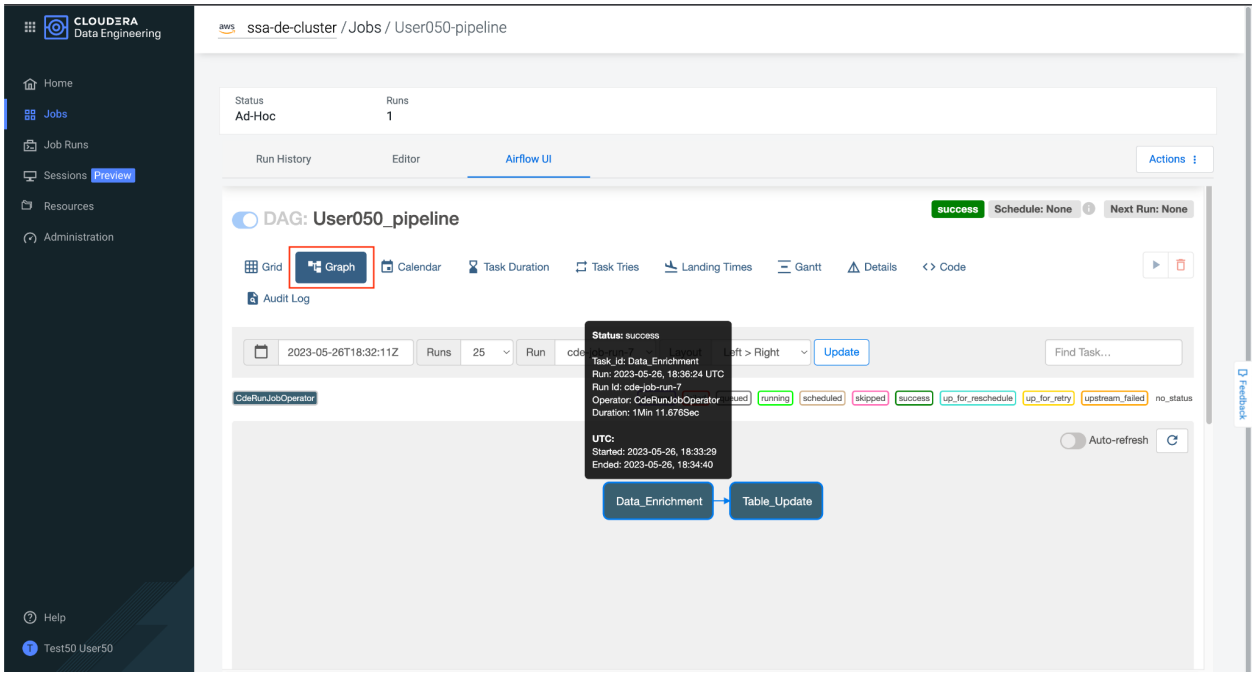
11. You should see the pipeline execution screen, indicating that the execution has been initialized.



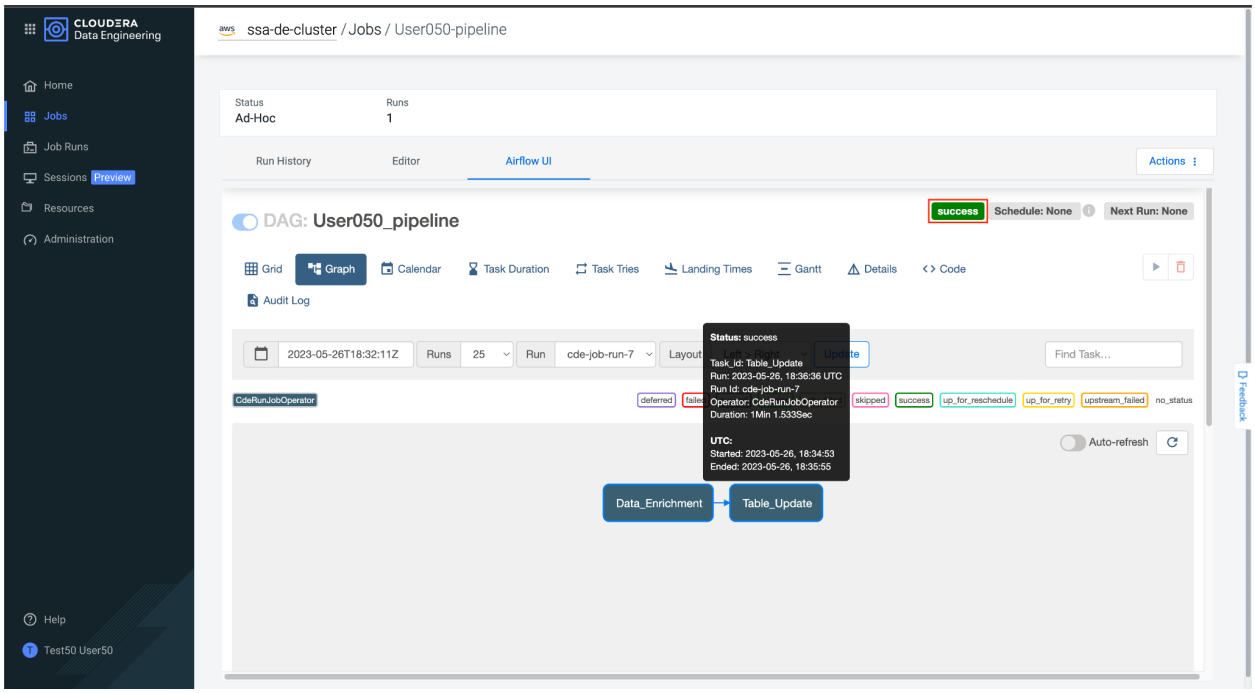
12. Click on the Airflow UI tab to see the execution detail of each step in the pipeline. The configured Data Enrichment and Table Update jobs are listed at the bottom left. The colors indicate the status of each job. Make sure the radio button **Auto-refresh** is enabled to automatically display the status of jobs.



13. You can see more information about the execution by clicking on the view **Graph**. Hovering the mouse over the Job name displays specific information for each step in the pipeline. Make sure the pipeline status is Success, which indicates that the entire pipeline was able to run without issue.



*The execution status appears next to the name of the pipeline (marked in red). If it is green and indicates **Success**, it means that the execution was successful.*



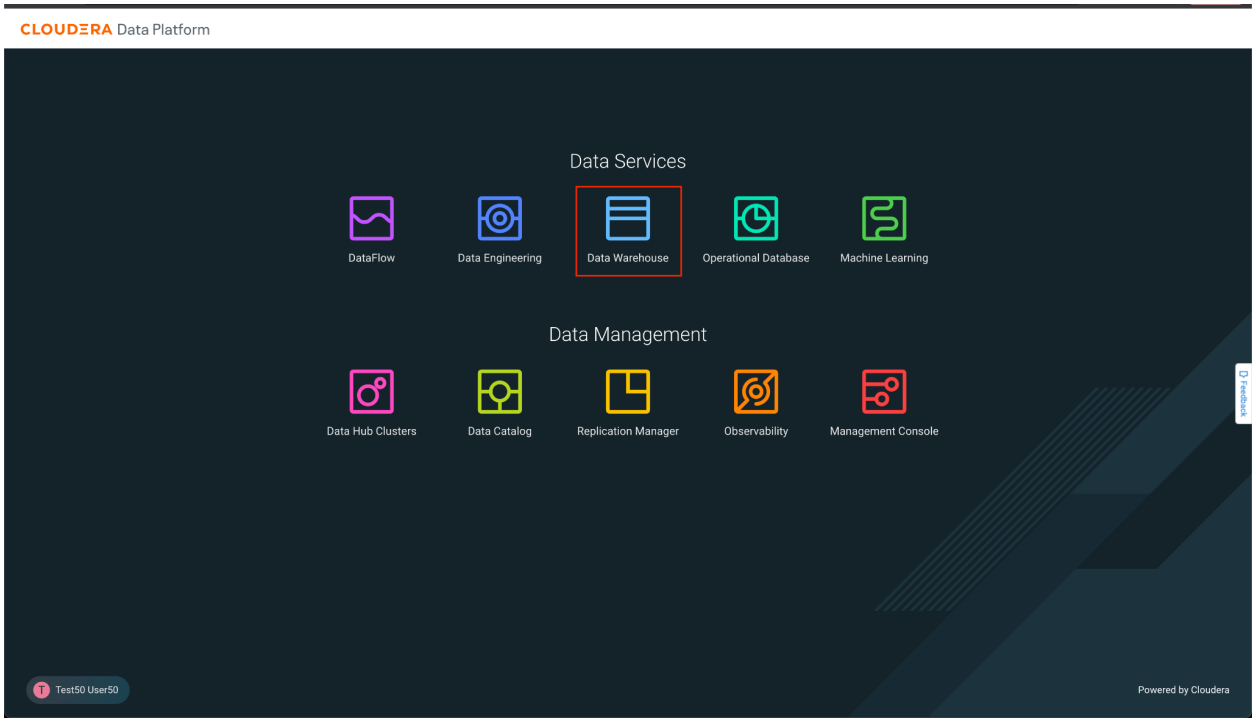
4. Data Warehouse

4.1. Goals

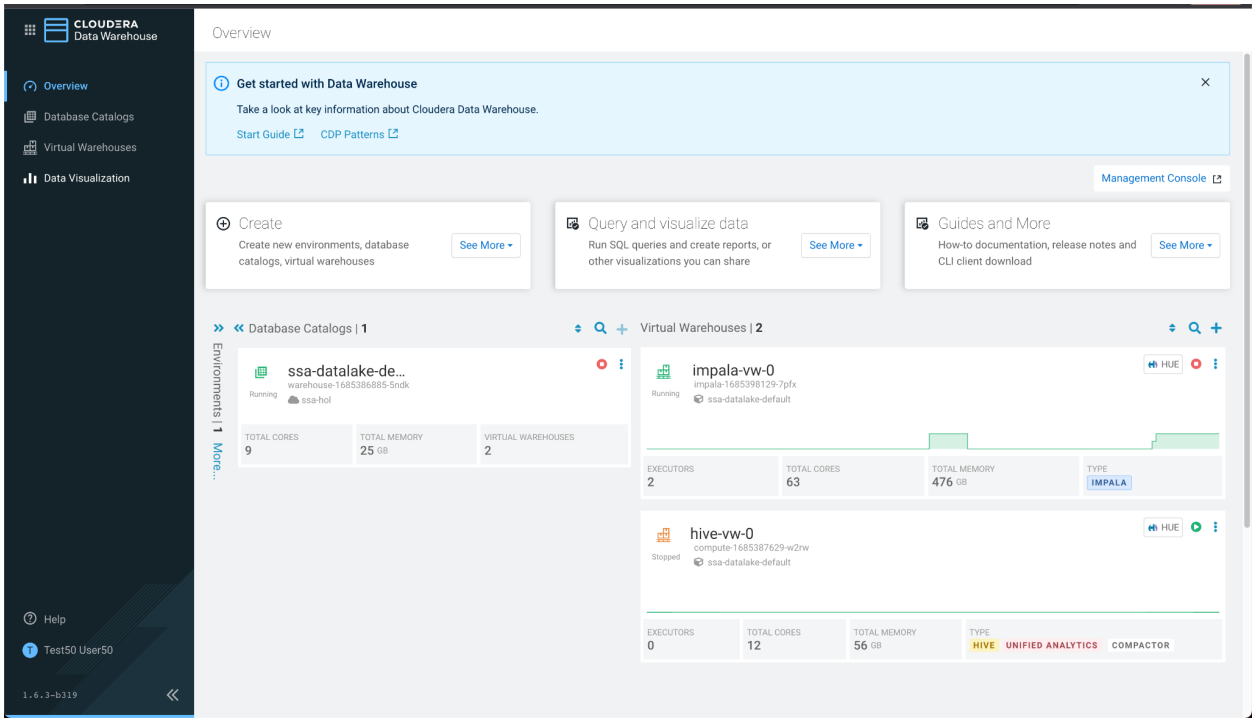
- Create a dataset pointing to the table
- Create a dashboard with metrics and dimensions

4.2. Dashboard Development

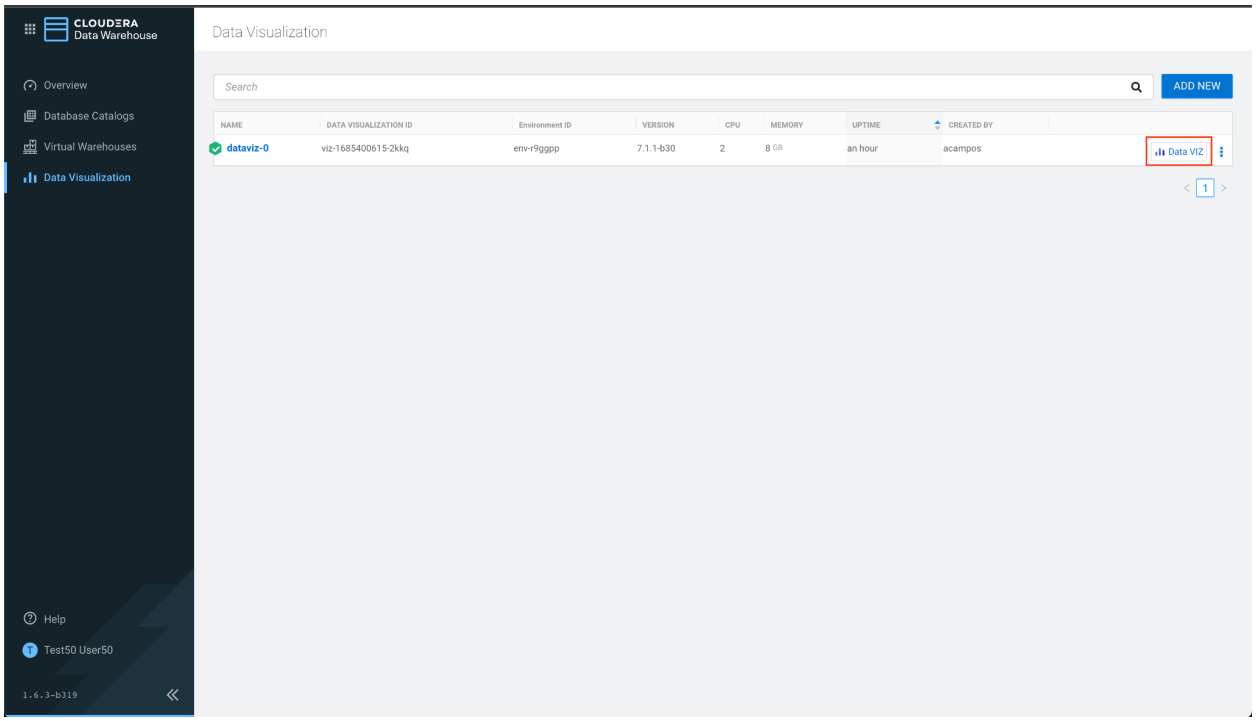
1. Click on Data Warehouse from CDP PC Home:



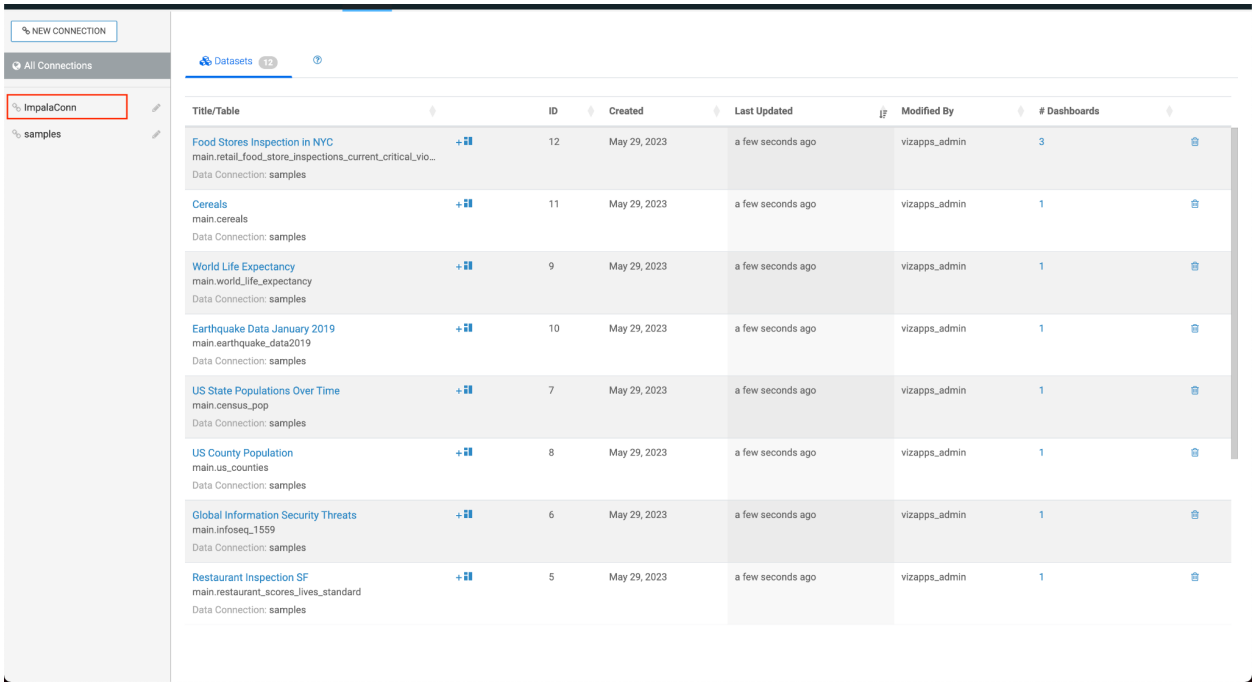
2. Data Warehouse welcome screen. Click on Data Visualization in the left menu.



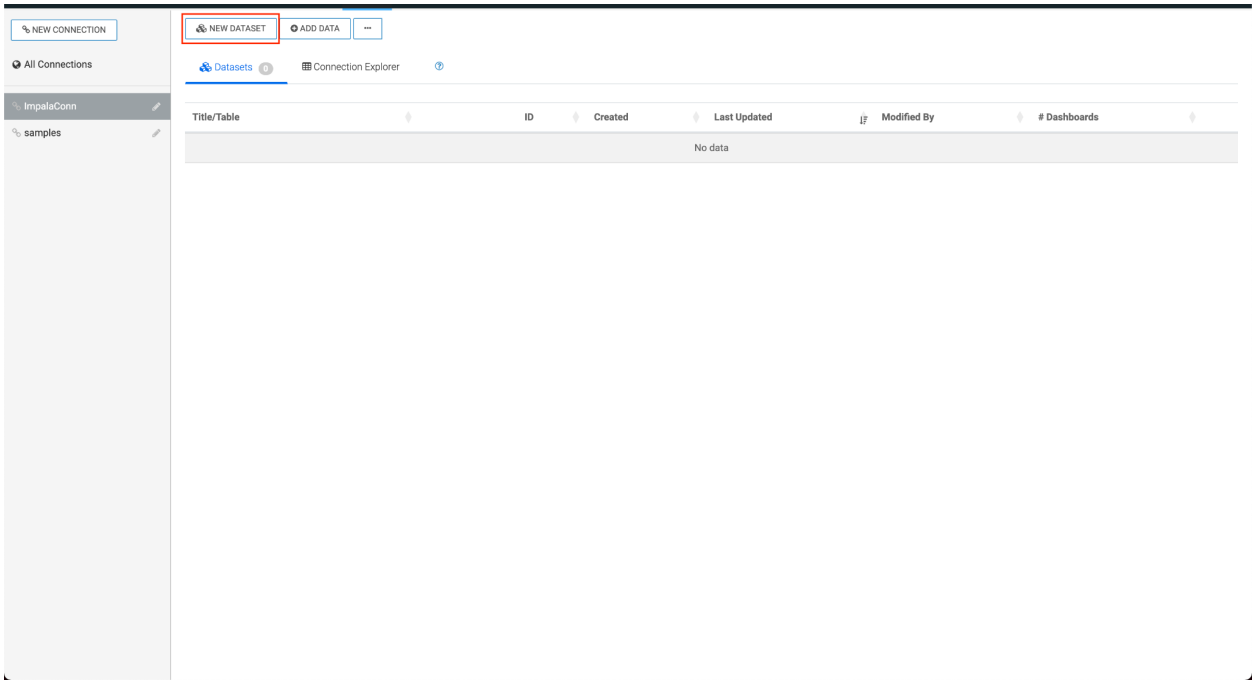
3. In Data Visualization, click on the button **Data Viz** from which they were assigned.



4. Once in Data Visualization, go to the Data option from the top menu, and then to the Connector **ImpalaConn** from the left menu.



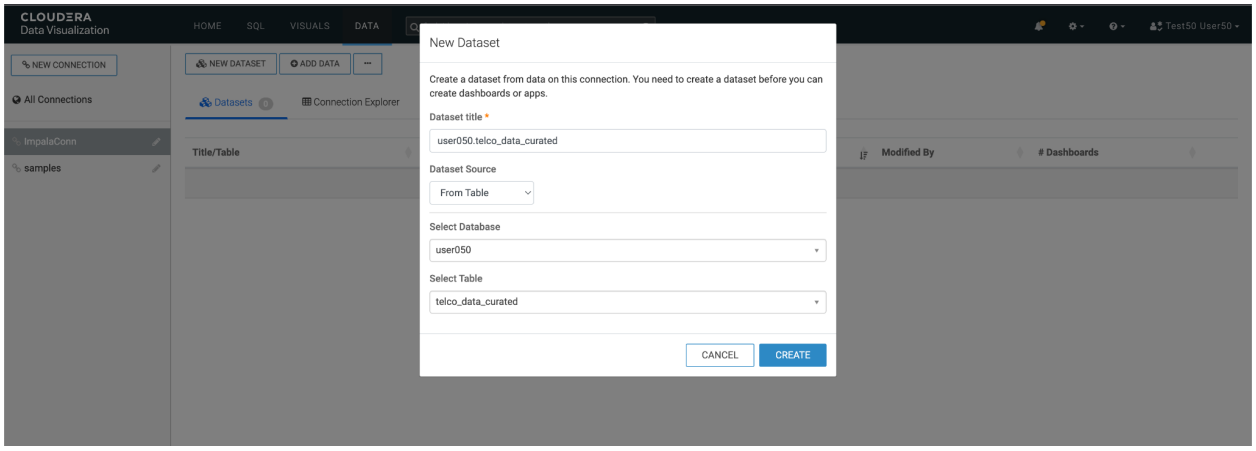
5. We have to create a new data source, for that, click on New Dataset and a window will appear to enter the information of the new data source.



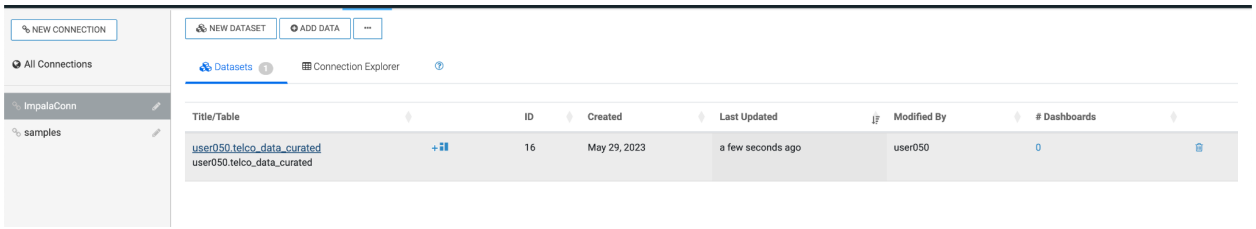
6. Enter the information for the new data source:

Dataset title: <assigned_user>.telco_curated_data
Dataset Source: From table
Select Database: <assigned_user>
Select Table: telco_data_curated

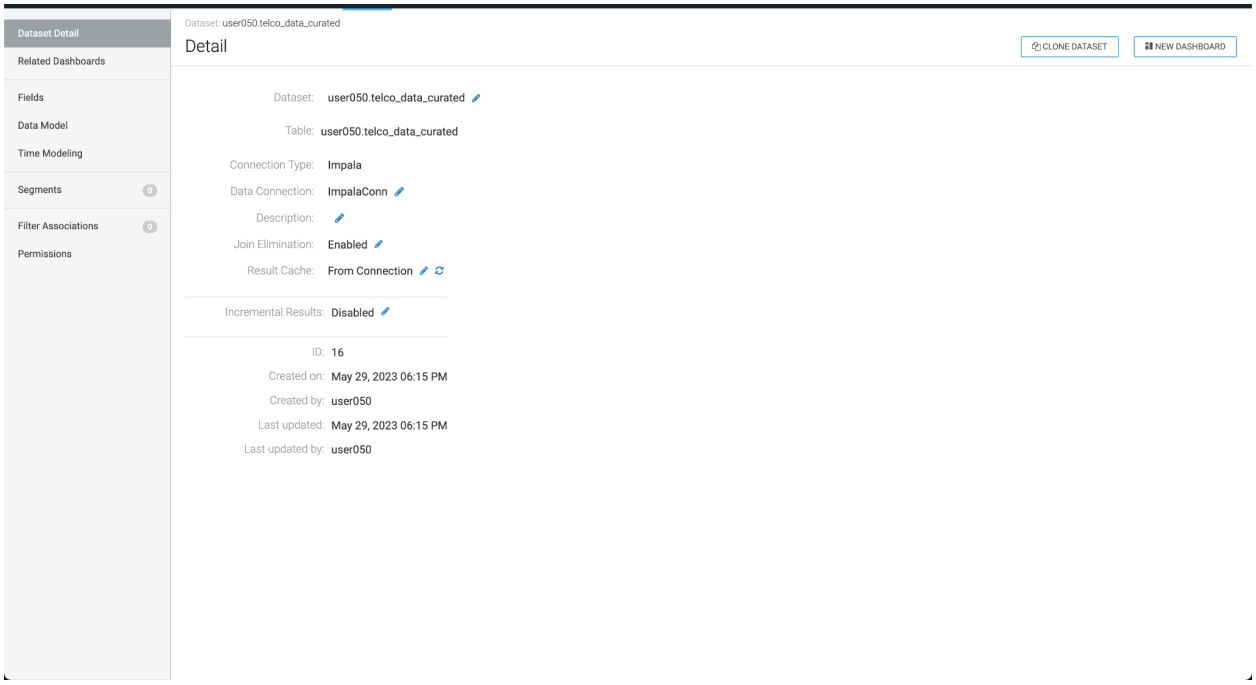
Click on Create to create the new Dataset.



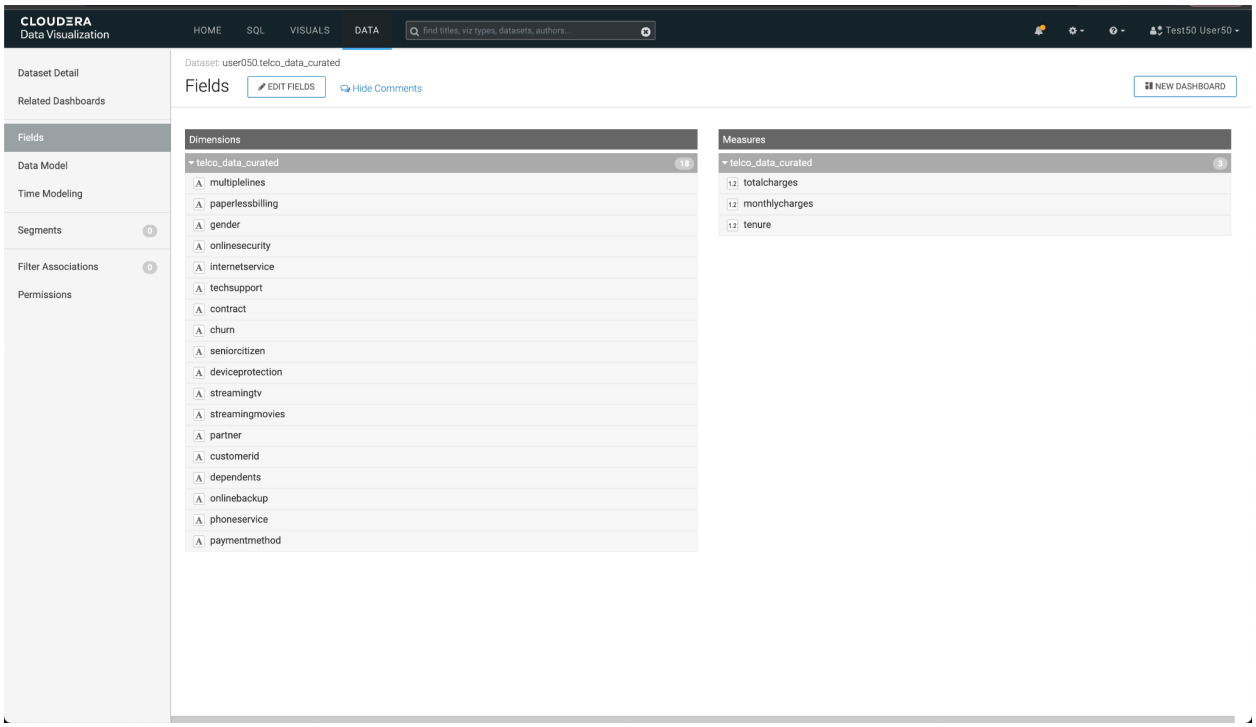
7. The new Dataset should appear in the list. Click on the dataset that you just created.



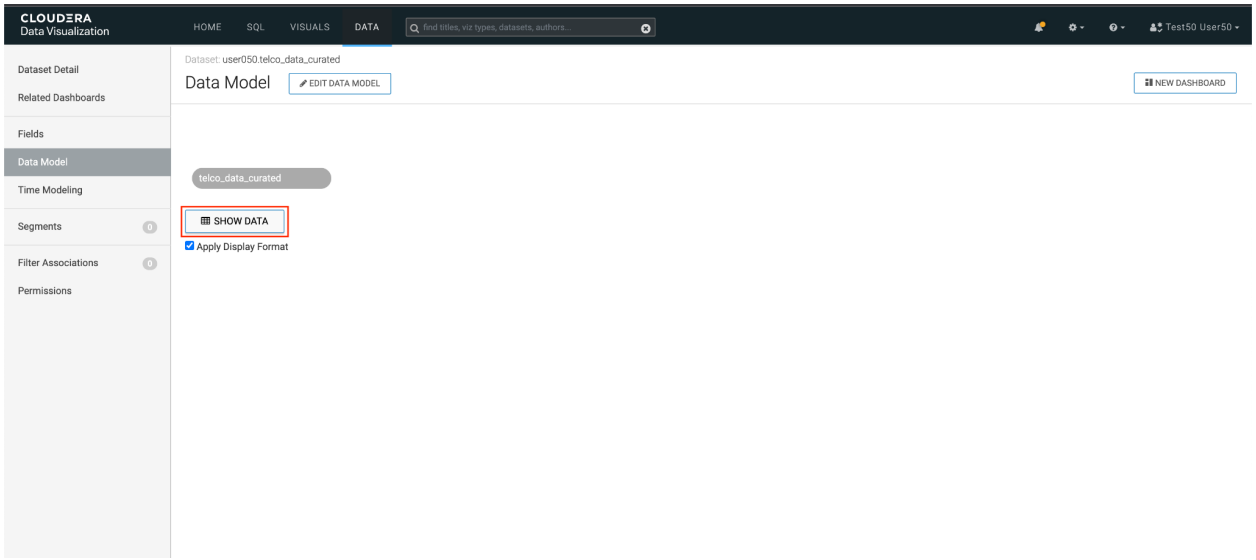
8. Here you will see the details of the dataset.



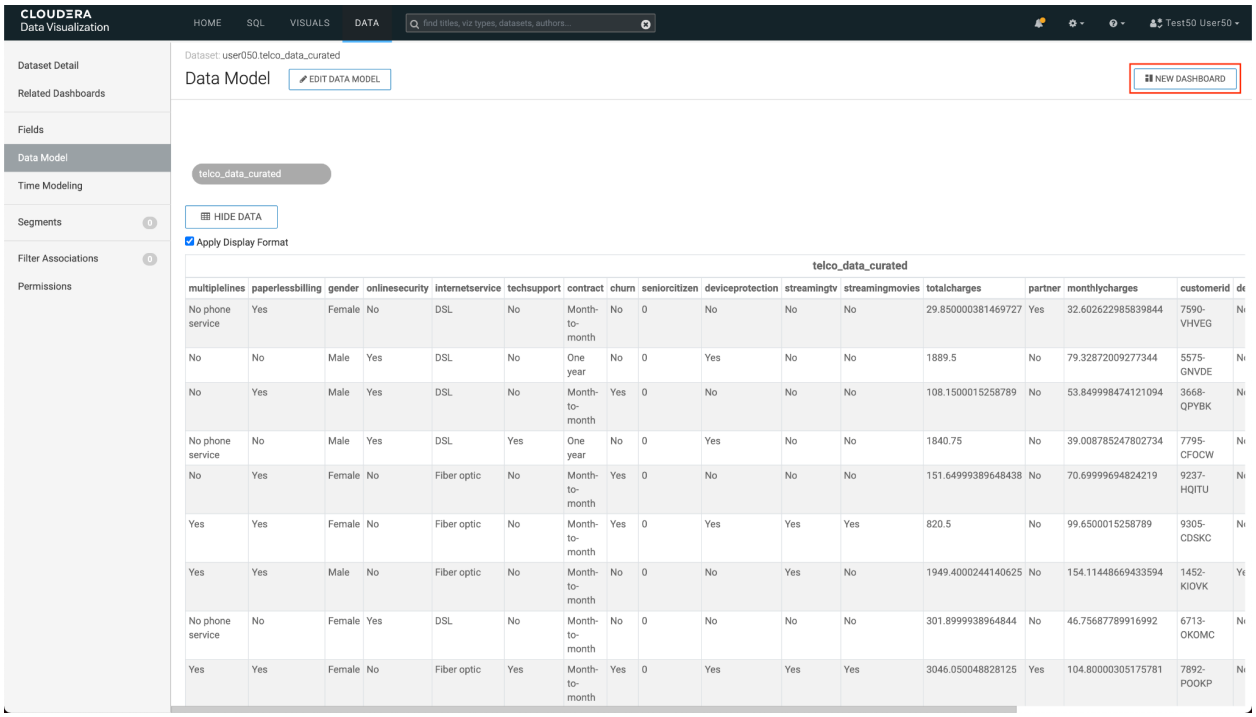
9. Click on **Fields** (left menu) to see the fields automatically captured during the dataset creation process.



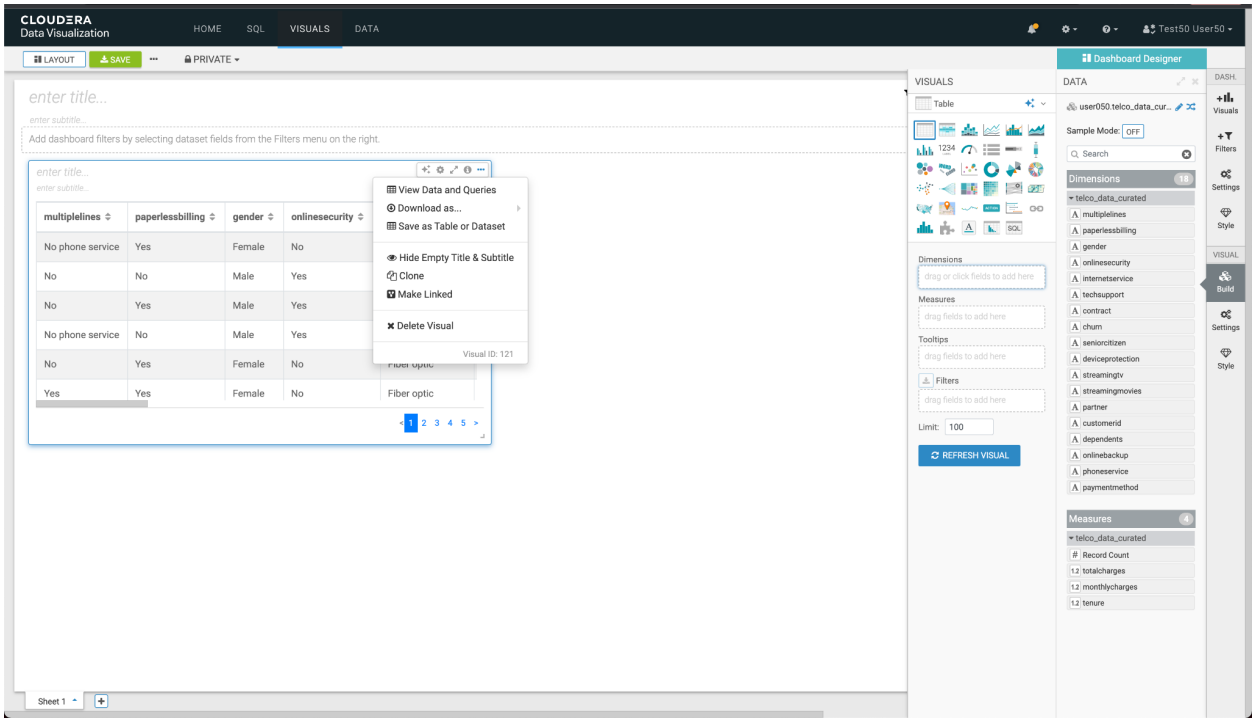
10. You can also preview the data from this screen. Click on **Data Model** (left menu) and then on the button **Show Data** that appears in the center.



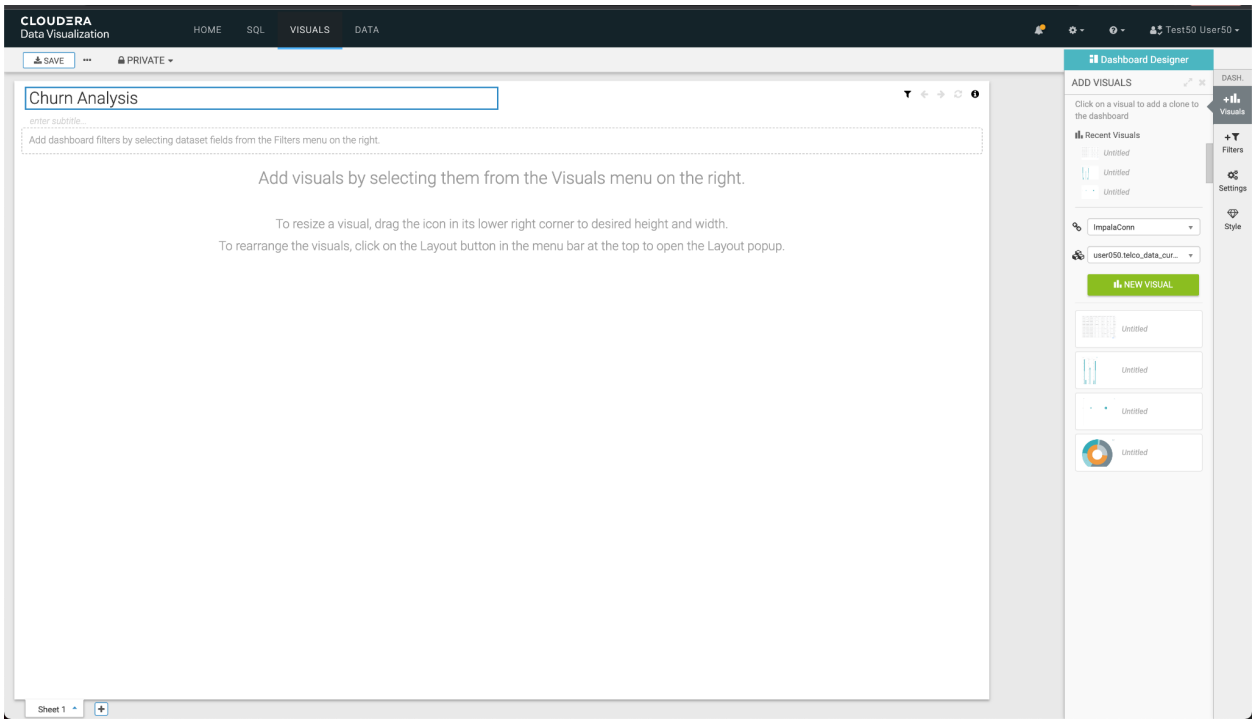
11. At this moment, a query to the Virtual Warehouse is executed to retrieve the data from the data set. Notice the columns and values. Click New Dashboard to create a new dashboard.



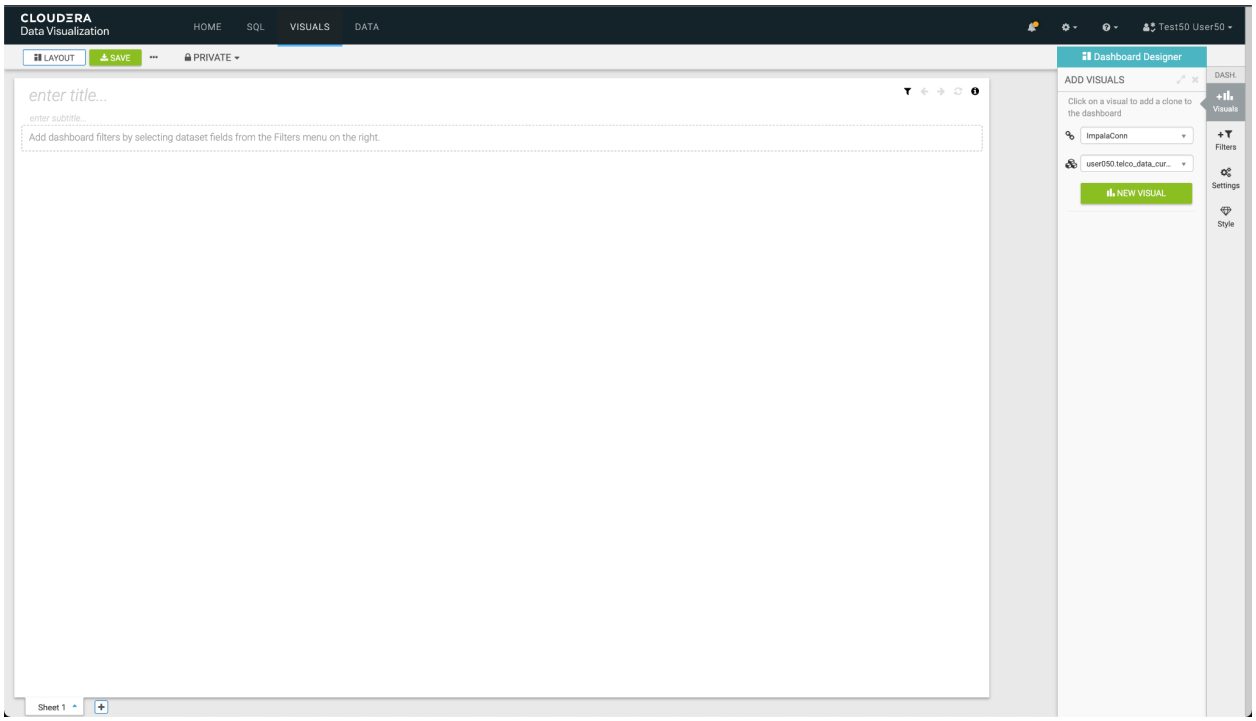
12. When opening the design canvas of a new panel, remove the element that is added by default, by clicking on the three dots (...) button at the top right of the element, and then clicking on the option **Delete Visual**



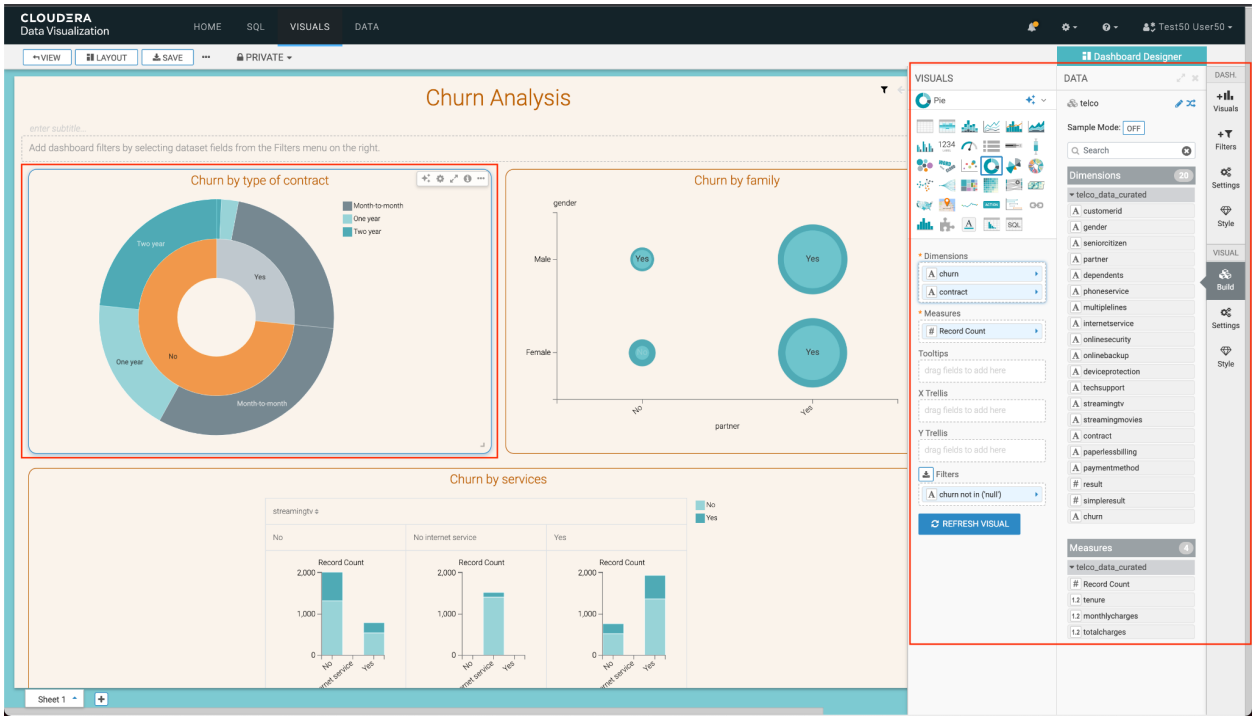
At the top of the canvas, in the enter title field, enter the name *Churn Analysis* to identify the dashboard.



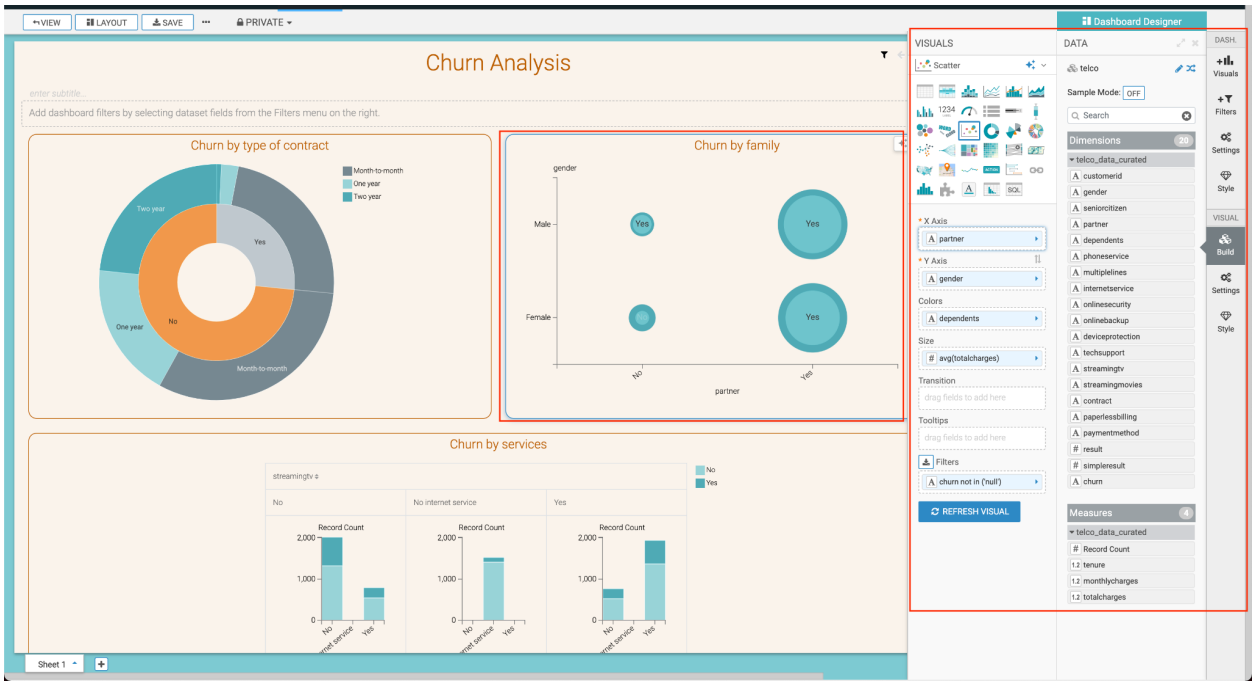
13. To add a new visual element, click on the button **Visuals** from the right menu, select the dataset that corresponds to them, and click on the button **New Visual.**



14. Add the first visual element, which is a pie chart with the dimensions **churn** and **contract**, with the metric of **Record count**. Once finished, click the button **Refresh Visual**.



15. Add the second visual element, which is a scatter chart with the dimension **partner** like X Axis, **gender** how Y Axis, **dependents** as Colors and **avg (total charges)** as Size. Once finished, click the button **Refresh Visual**.



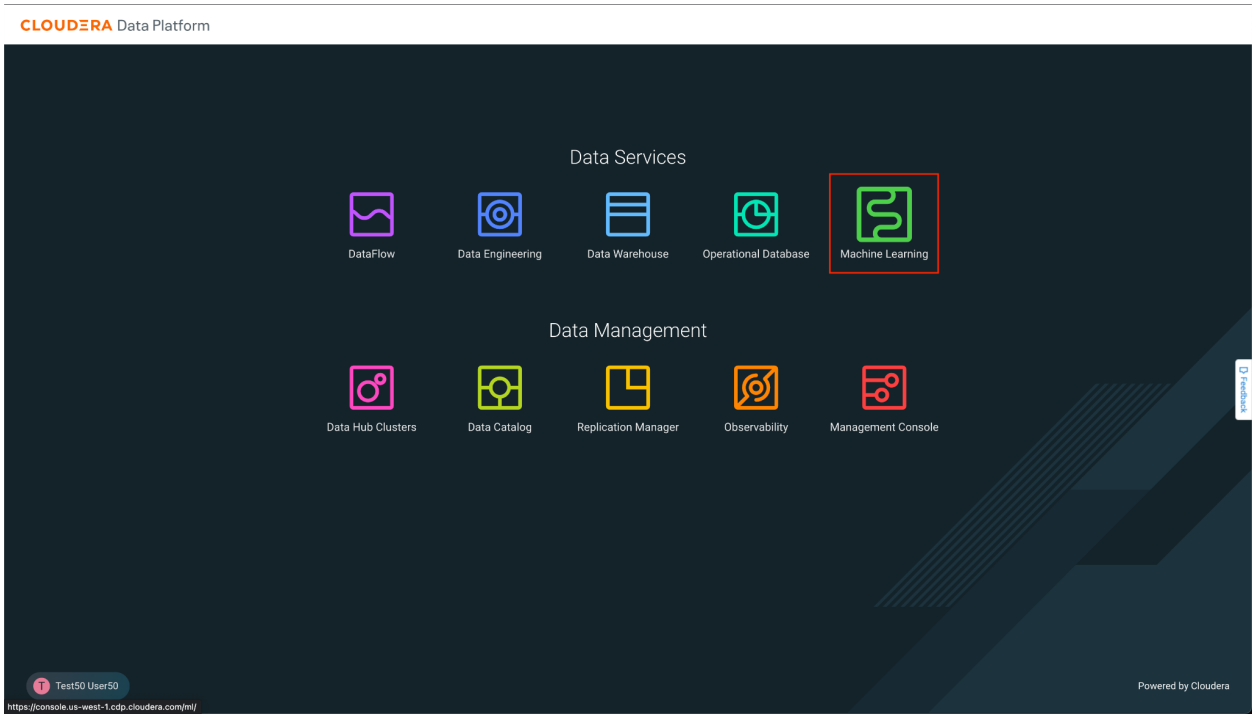
5. Machine Learning

5.1. Goals

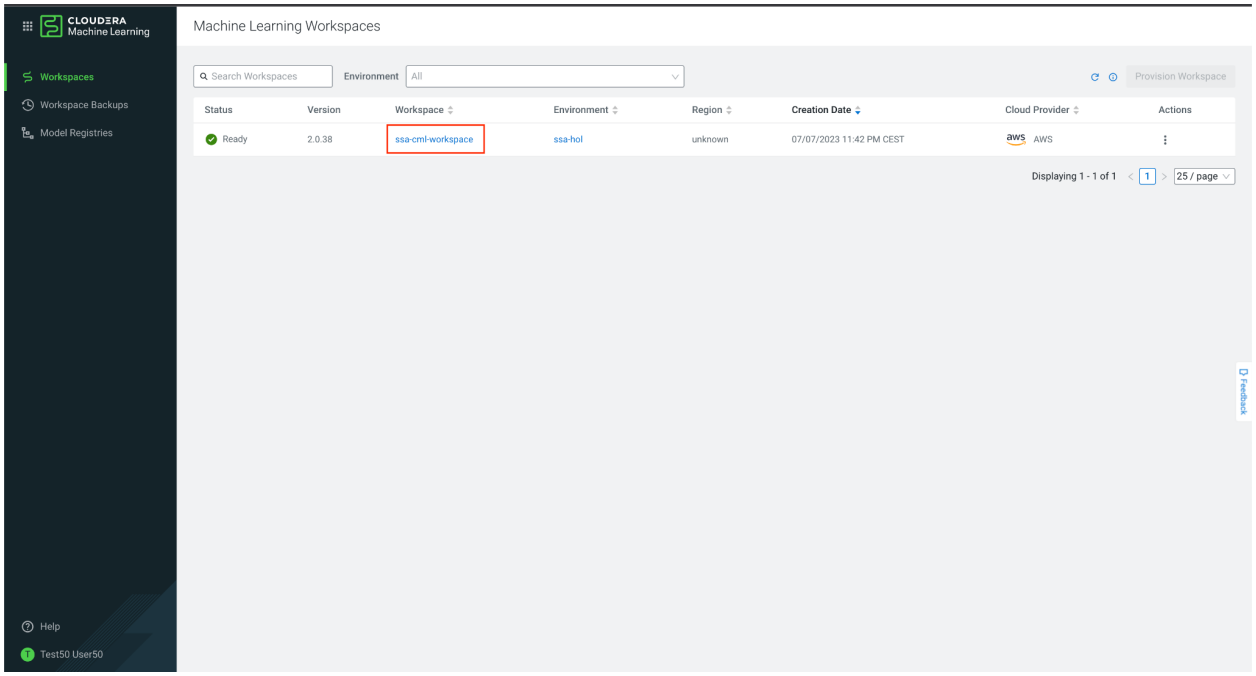
- Train a model to predict if a customer will churn
- Deploy/expose model as REST API

5.2. Create a Machine Learning Model for Churn Prediction

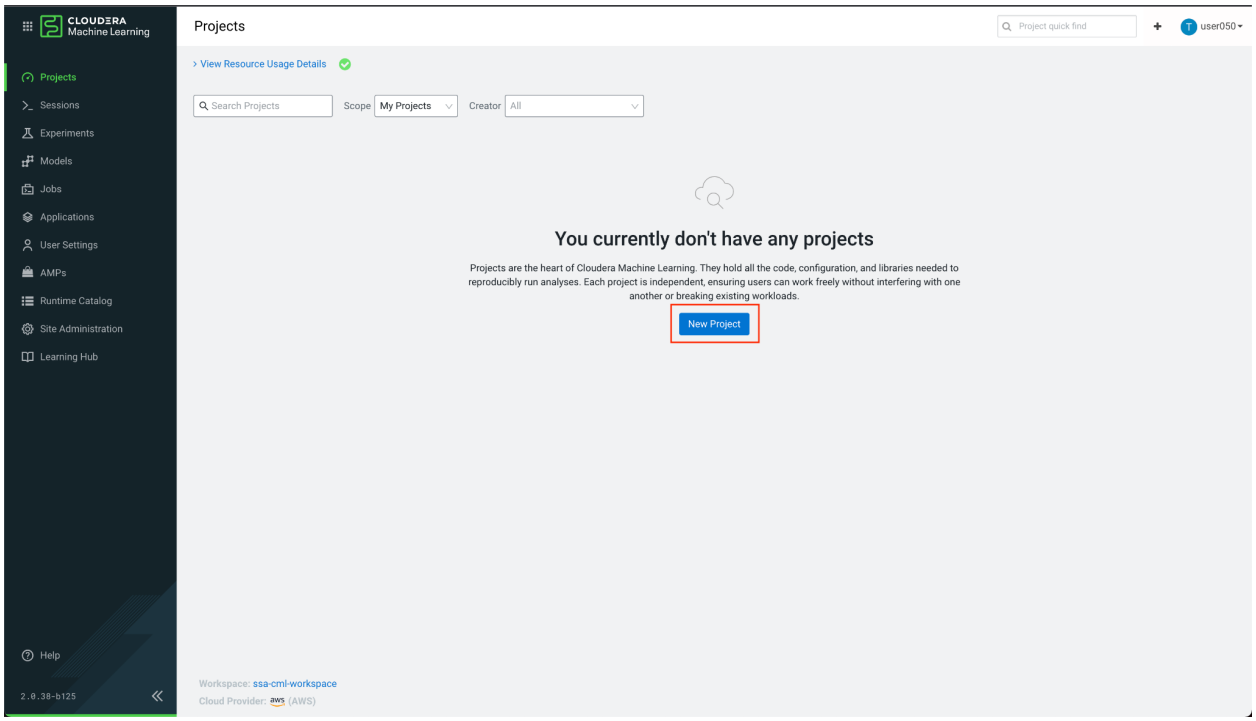
1. Click on Machine Learning from CDP PC Home:



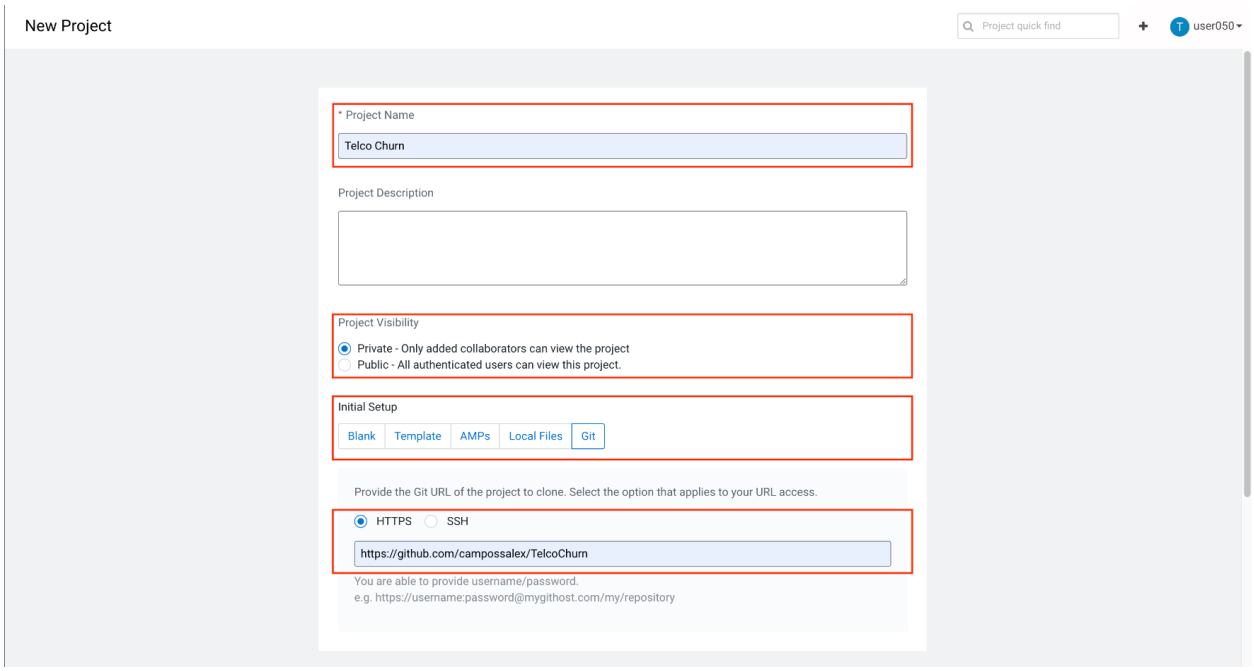
2. This is a screen to select a Workspace, which is compute resource allocation for Data Science related jobs. Click on the only Workspace that appears.



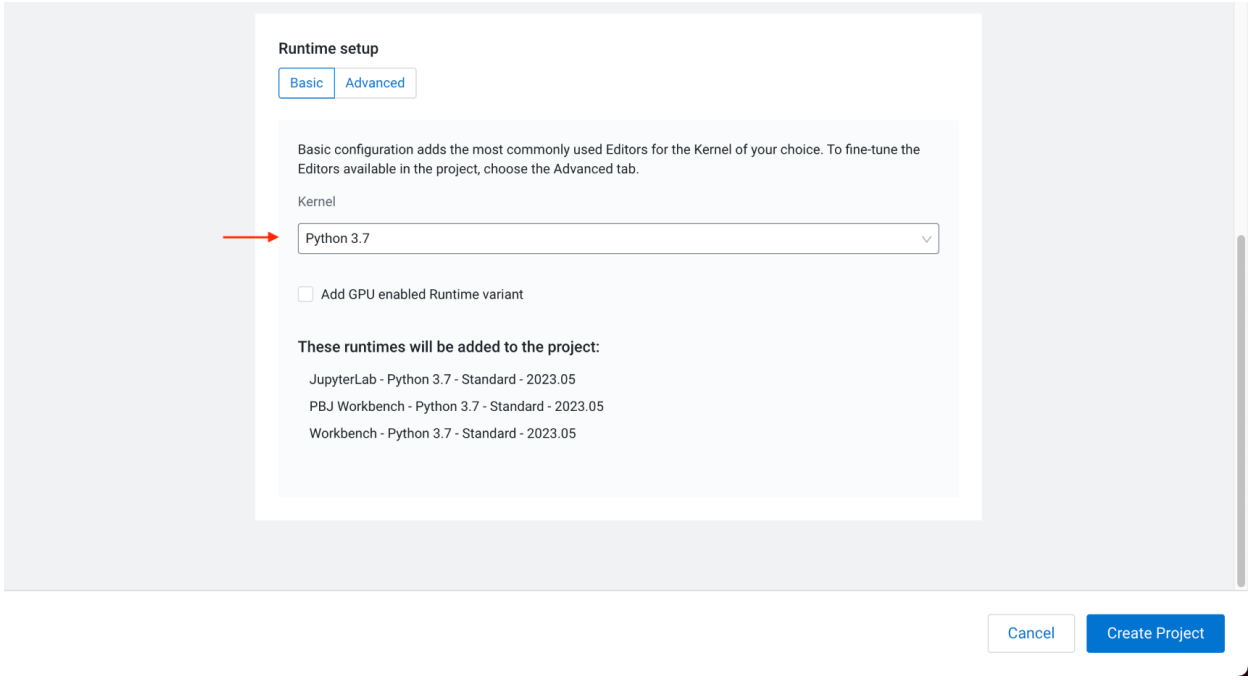
3. Once in the Workspace, you should see the following interface. Here are the projects you have created. It is time to create a new project. Click on the blue button **New Project**.



4. Enter the following information to create a new project:
Project Name: User0xx Telco Churn
Project Visibility: Private
Initial Setup, select Git
In the text field below HTTPS, enter the url of the git repo:
https://github.com/campossalex/TelcoChurn



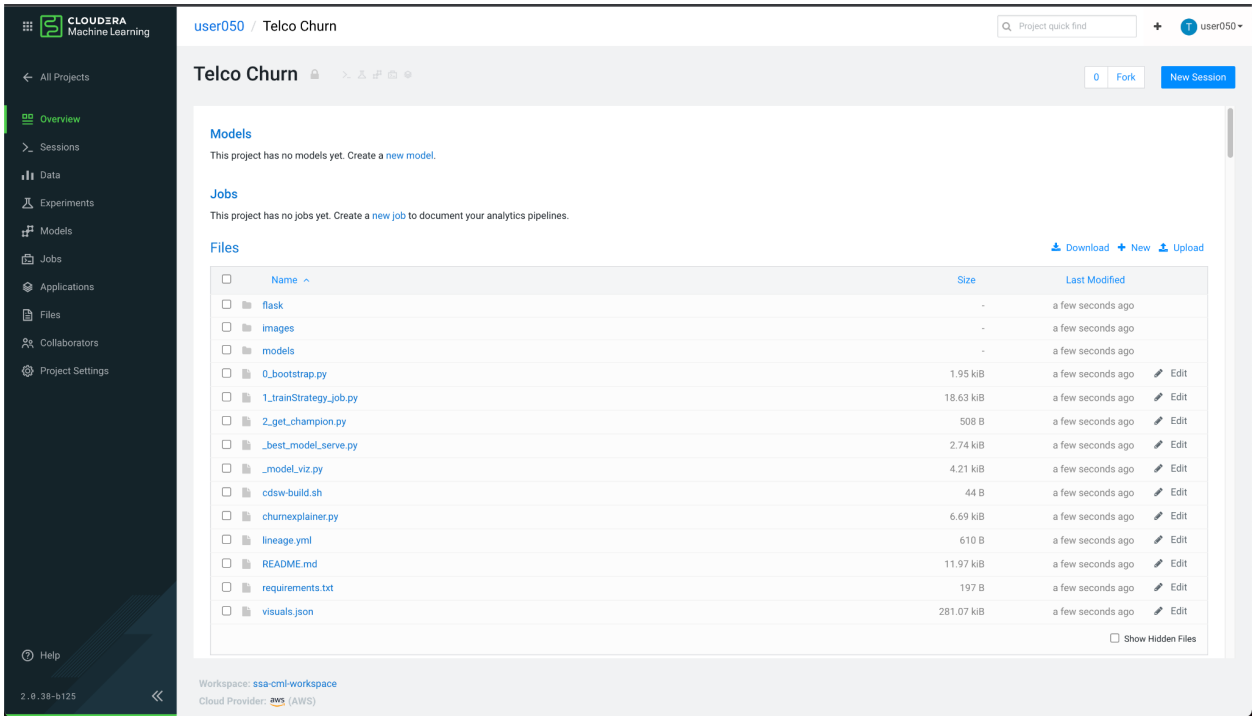
Make sure to select **Python 3.7** in the Kernel selector. Click the button **Create Project**



5. Once the project is created, you should see the following screen:

Models, deploy and manage models as REST APIs to serve predictions.
Jobs, automate and orchestrate the execution of batch analytics workloads
Files, assets that are part of the project, such as files, scripts and code

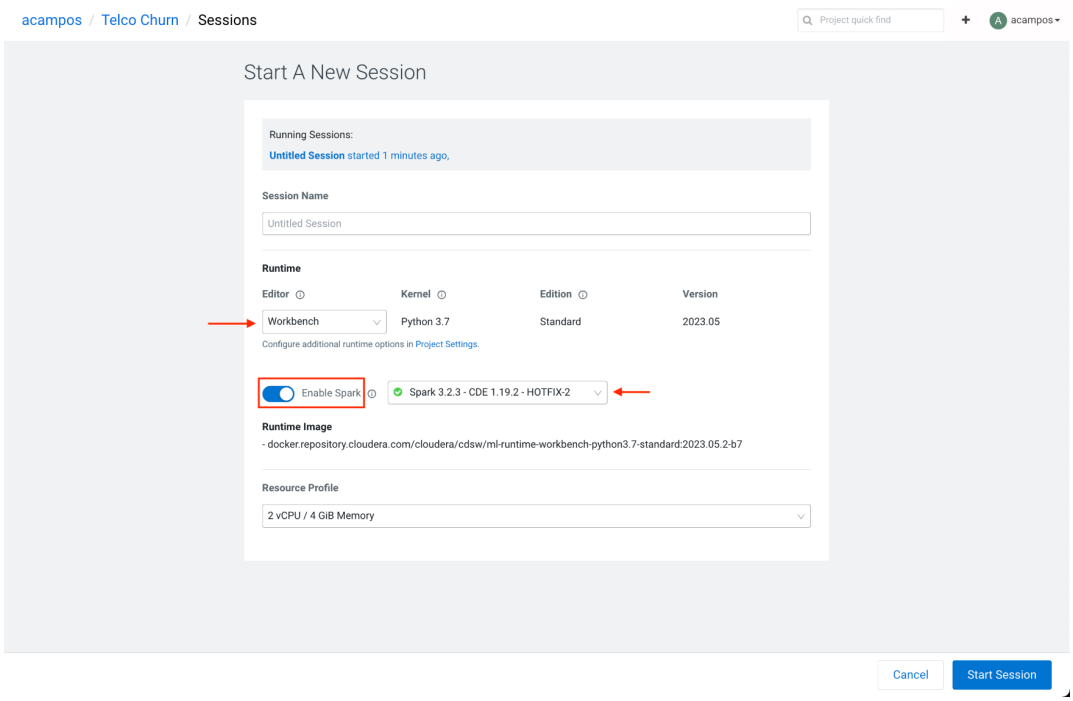
This Telco Churn project consists of running three scripts. The way of execution is through a session, which is the allocation of isolated compute resources for each user. For this, you must click on the blue button **New Session**, located in the upper right.



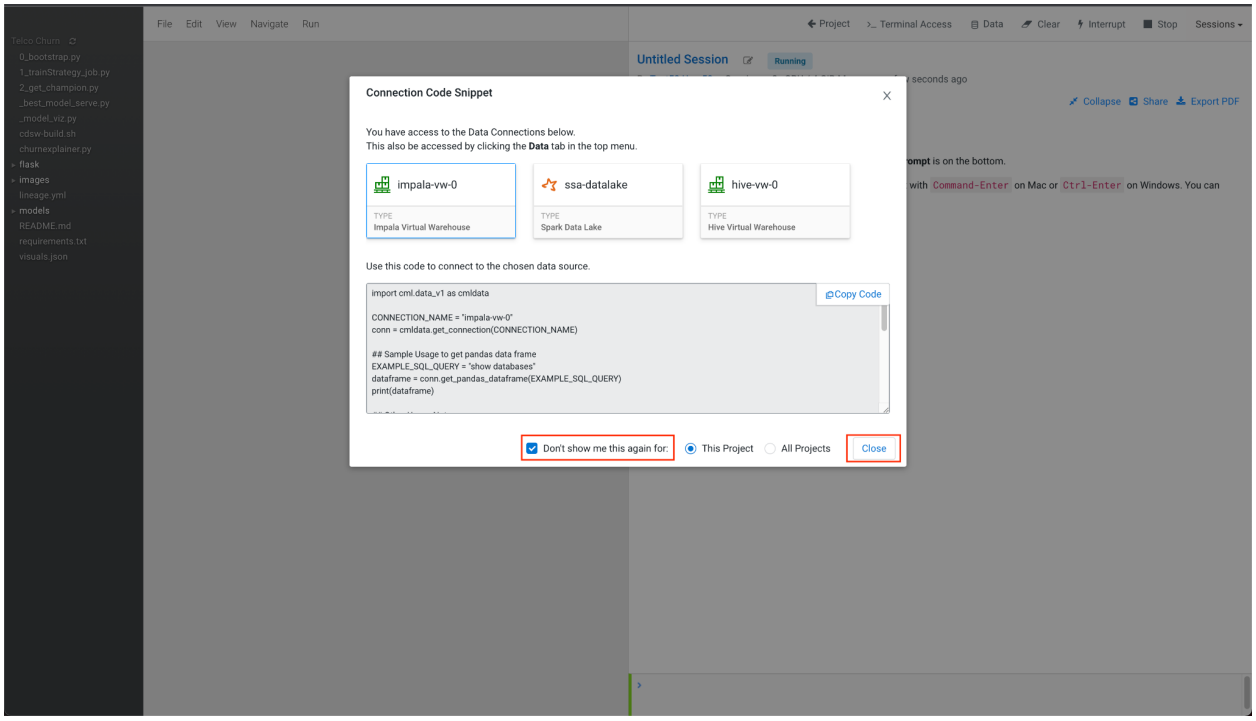
6. When starting a new session, make sure:

- Name the session: User0XX_Workbench_Session
- Select **Workbench** in the Editor selector.
- Enable **Spark**, marking the corresponding check.
- Select **Spark 3.2.x**, in the Spark version selector.
- Ressource Profile: 2vCPU / 4 GiB

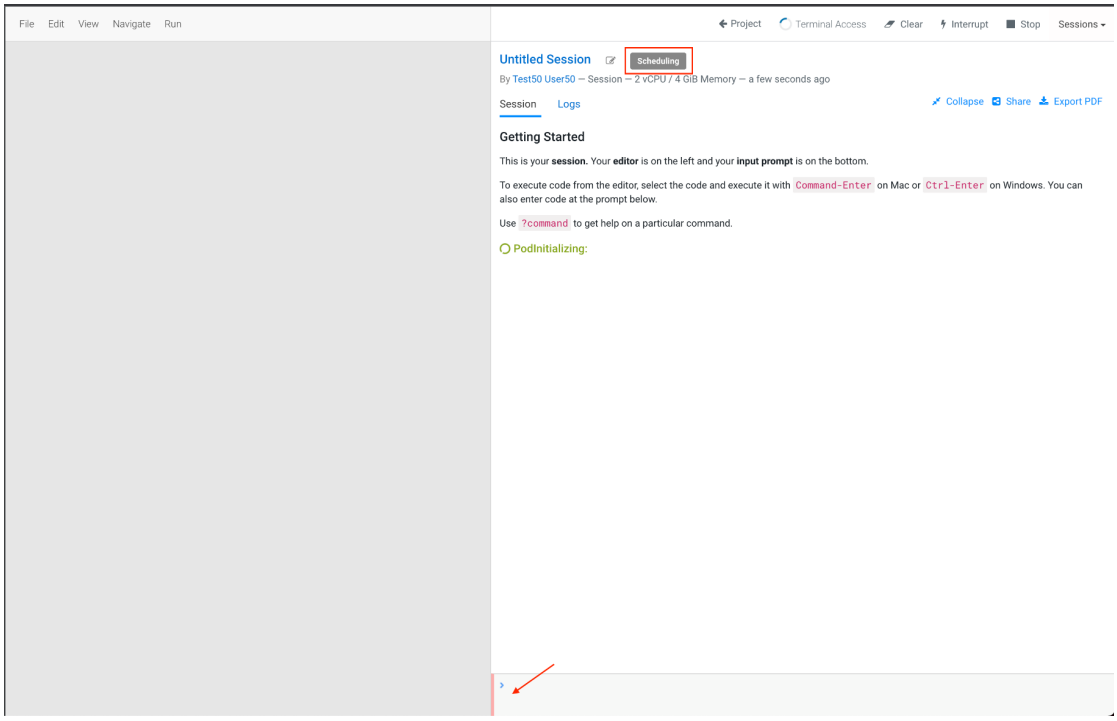
Click on the button **Start Session**



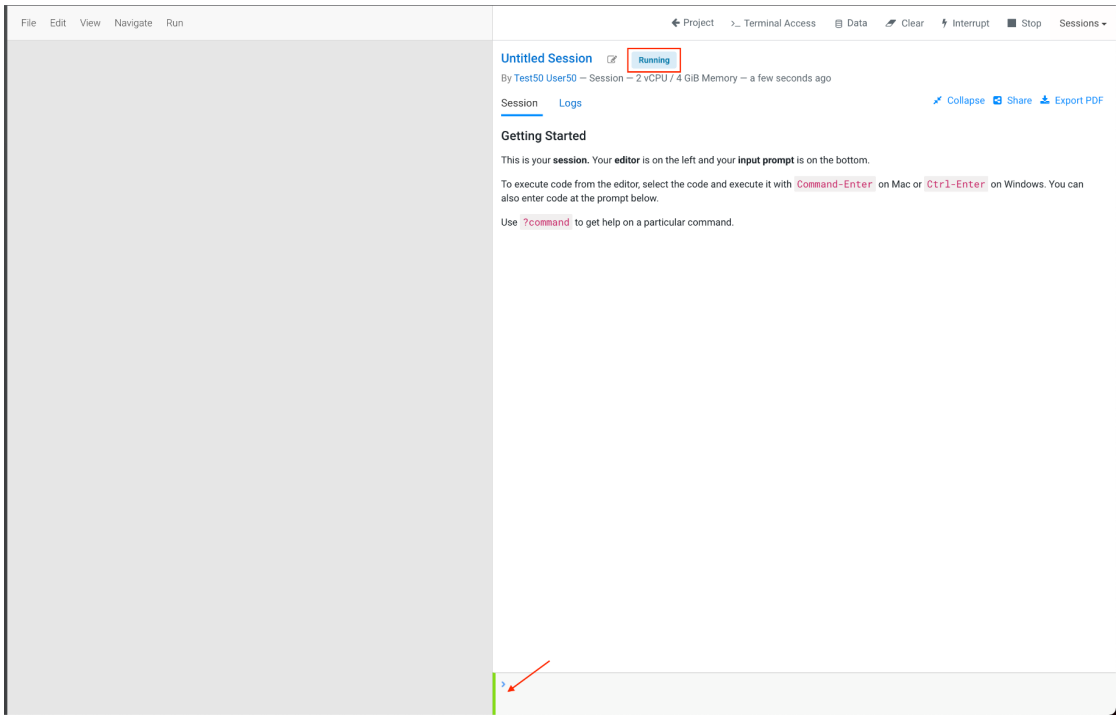
7. When you start a session for the first time, it will ask if you want to use a data connection. This project does not need this type of connection. mark the check of **Don’t show me this again**, and then click the button **Close**, so this window will not appear anymore.



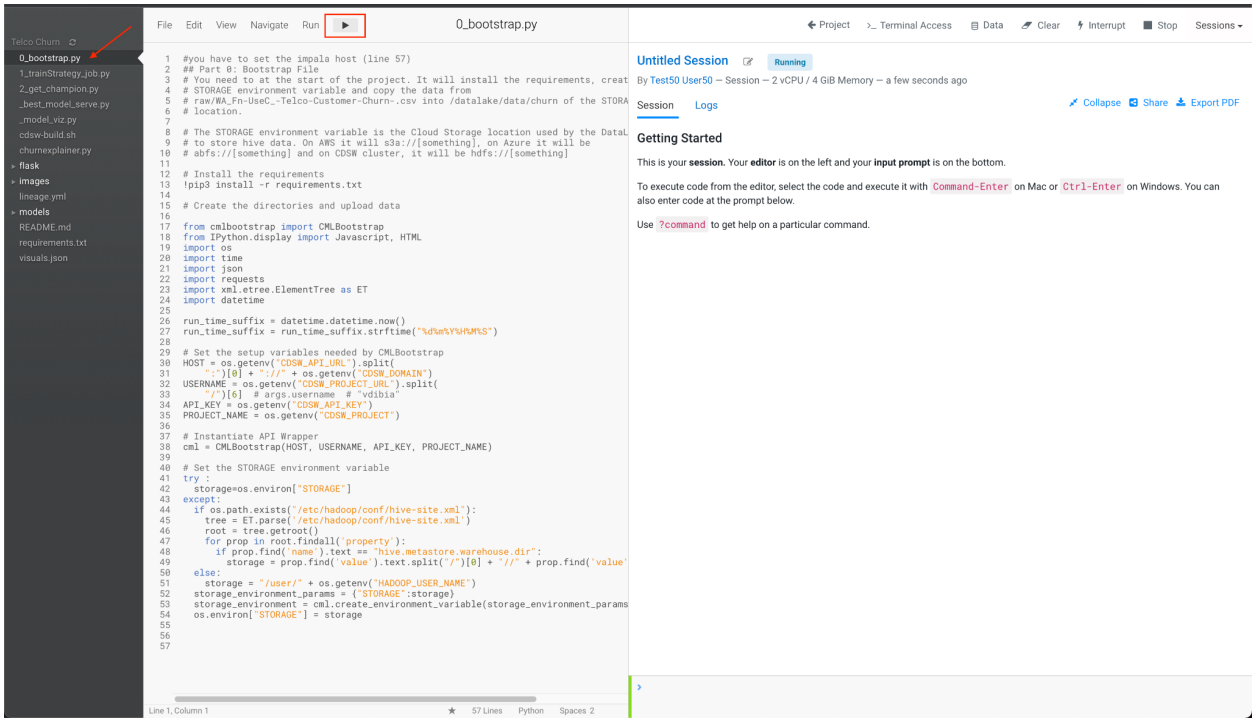
8. The editor/notebook located on the right side of the window will be in **Scheduling** status, and the bottom command bar flashing red. This means that CML is allocating computation for your session.



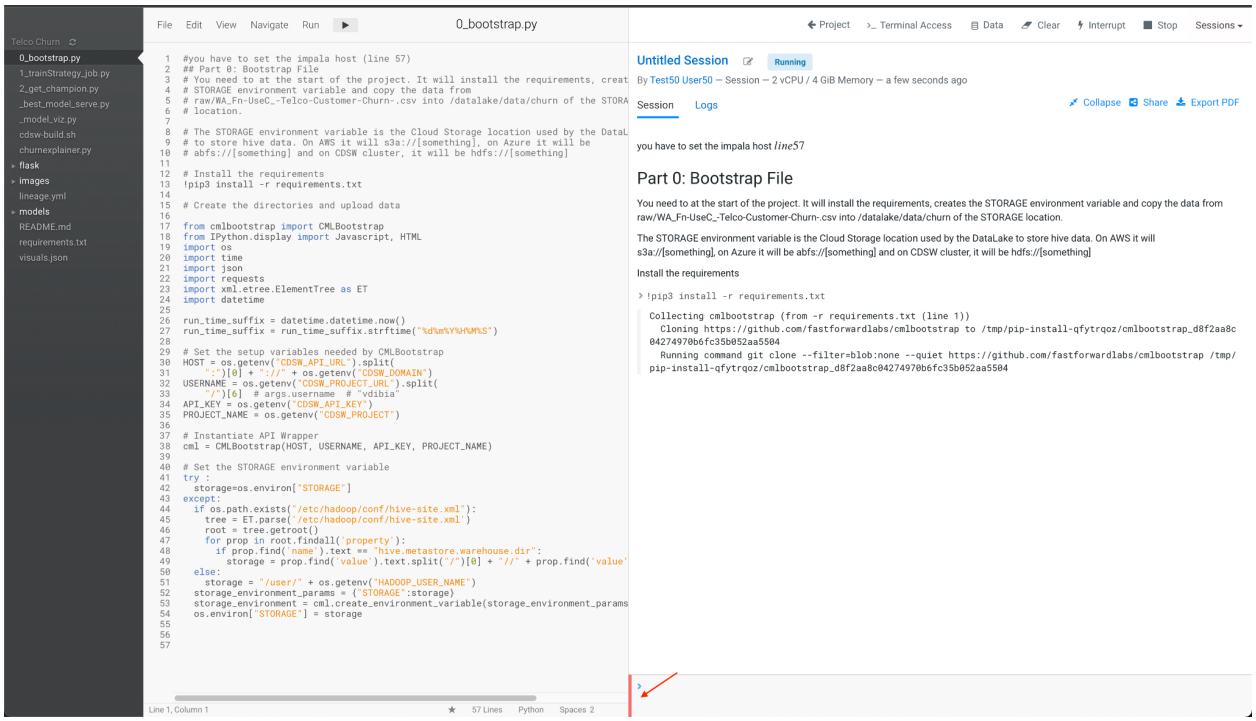
After a few seconds, the status changes to **Running**, and the command bar to green. This means that the session is ready to run code.



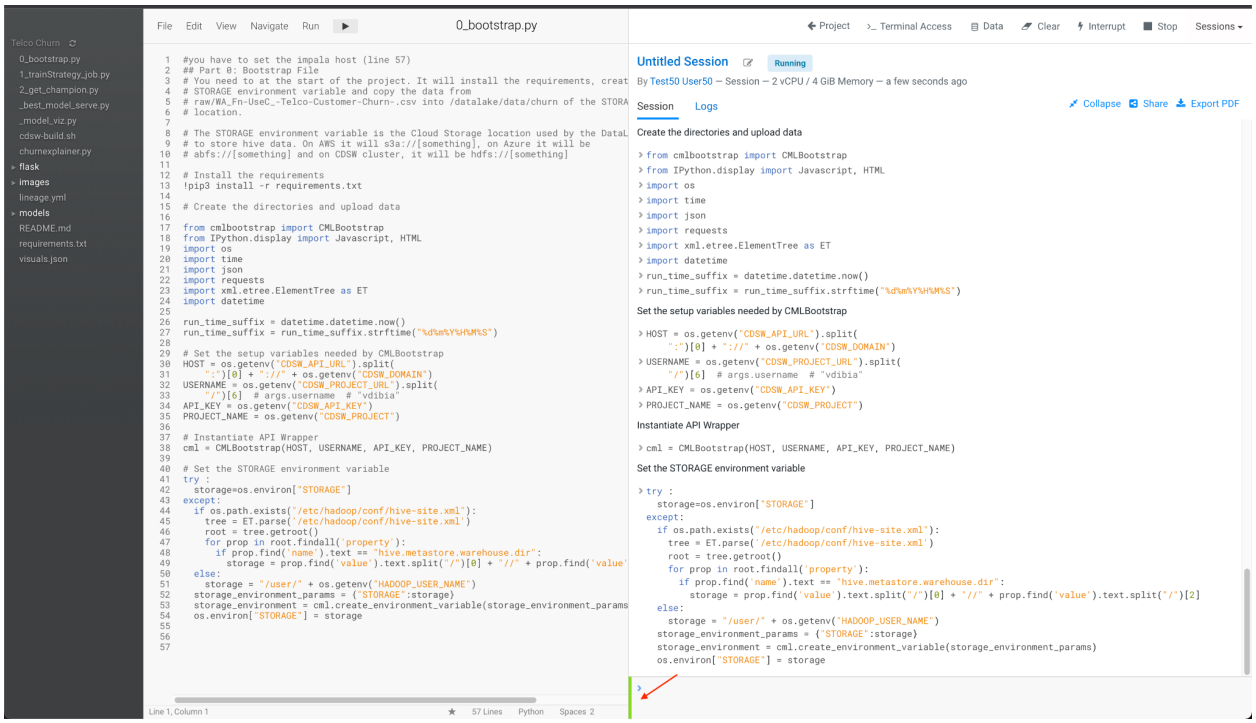
9. The first script/code to run is **0_bootstrap.py**. This Python code configures the libraries required for the project and integration with Lakehouse tables you populated before. Select (just one click) the file in the bar located on the left side of the interface, this will make the code appear in the editor. Once the file is selected, click on the button ▶ to run the code.



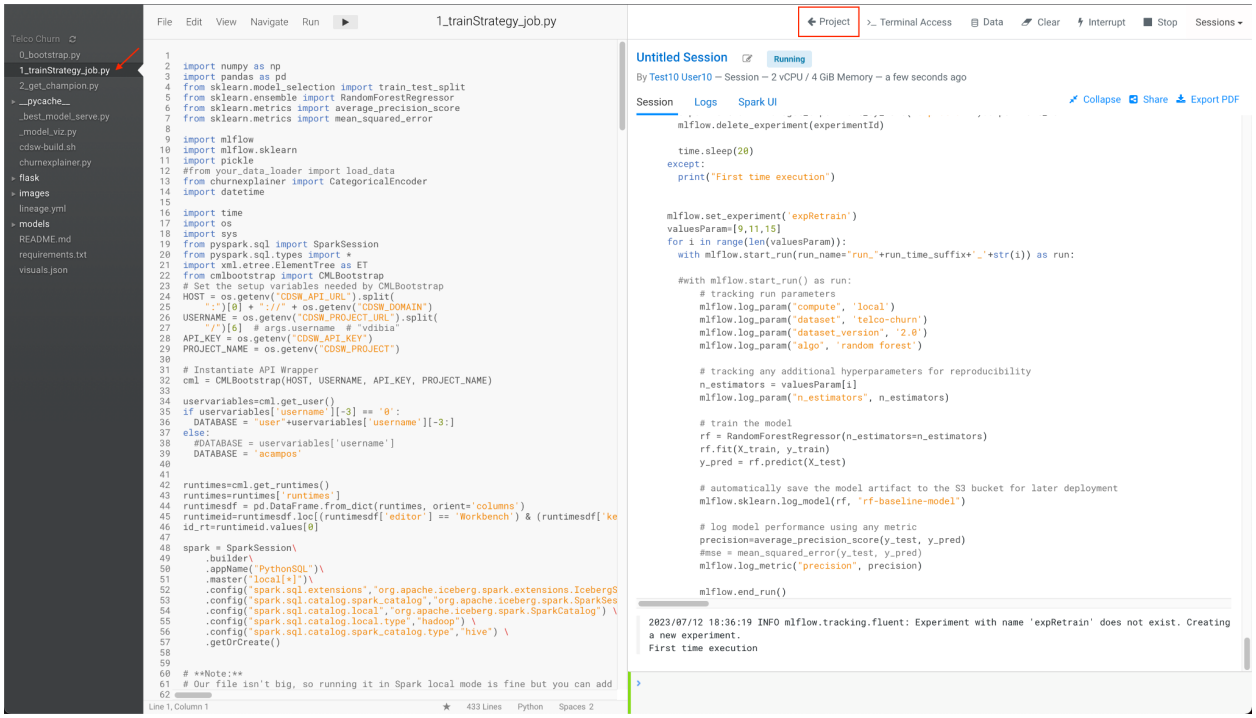
When you start execution, you will see code output on the right side of the interface, and the bottom command bar flashing red, indicating that it is busy.



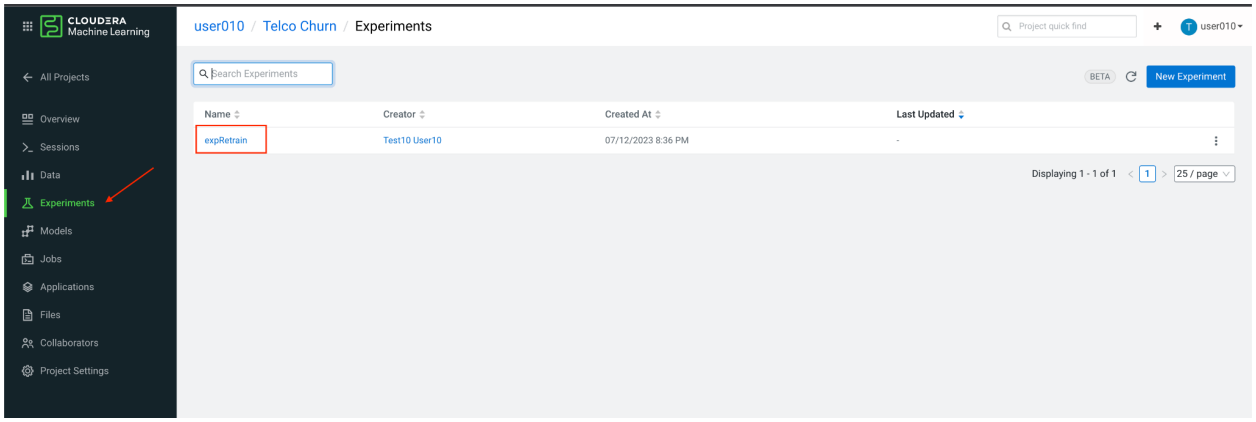
The green command bar indicates that the execution of the code has been finished. This bootstrap code takes 3-4 minutes to run.



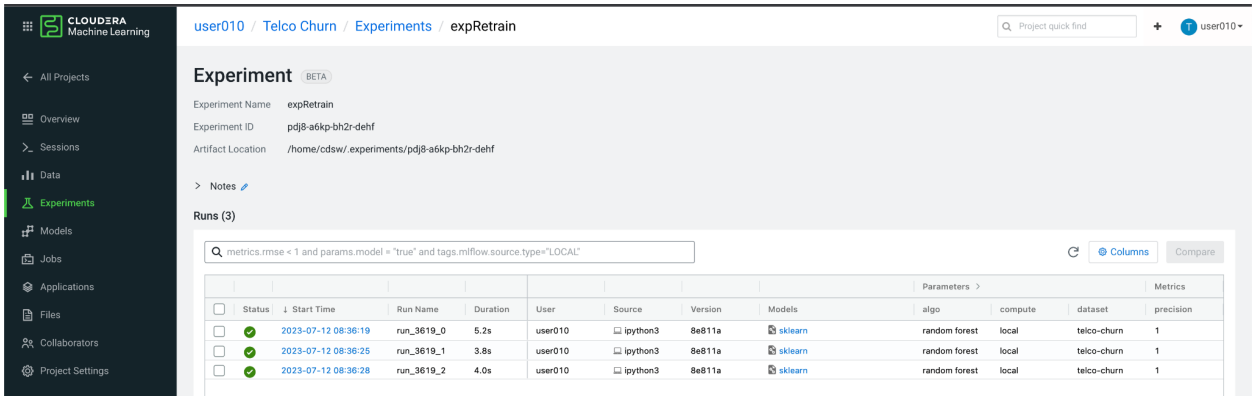
10. The second script/code to run is **1_trainStrategy_job.py**. This Python code will create the Experiment to run the model with three different hyper parameters and records the precision. Select (just one click) the file in the bar located on the left side of the interface, this will make the code appear in the editor. Once the file is selected, click on the button ▶ to run the code. Once the execution is finished (approximately 1 minute), click on the button **Project**, located in the upper right bar of the session to go back to the project home.



11. Once back in project gome, click on the **Experiments** option, from the left menu, and then on **expRetrain** in the list of Experiments that appears.

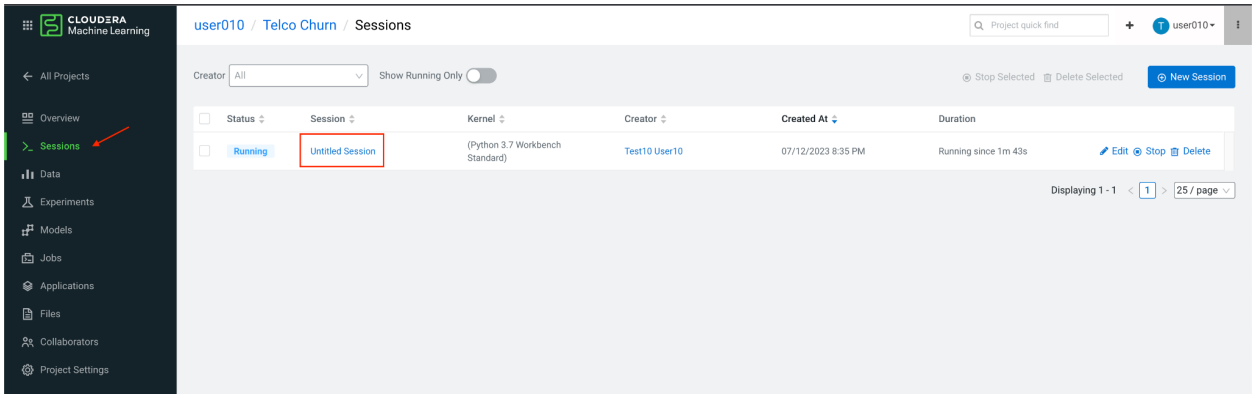


12. On this screen you will see the three runs of this experiment. Look at the last column, where **precision** attribute displays. This is the precision that each hyper parameter is delivering.

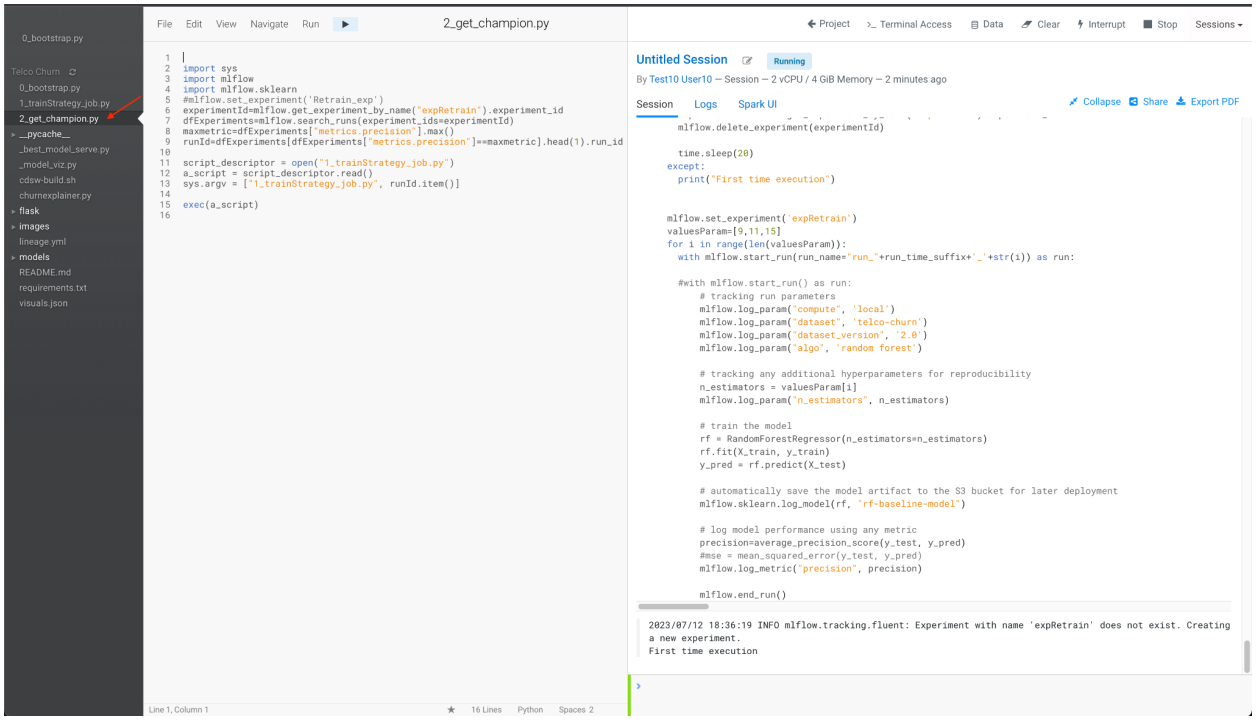


13. Let's go back to the session to run the last code. Since sessions run in Kubernetes containers, it's very easy to get back to where we were. Click on the option **Sessions** from the left menu, and later in the only session that will appear in the list.
Go to

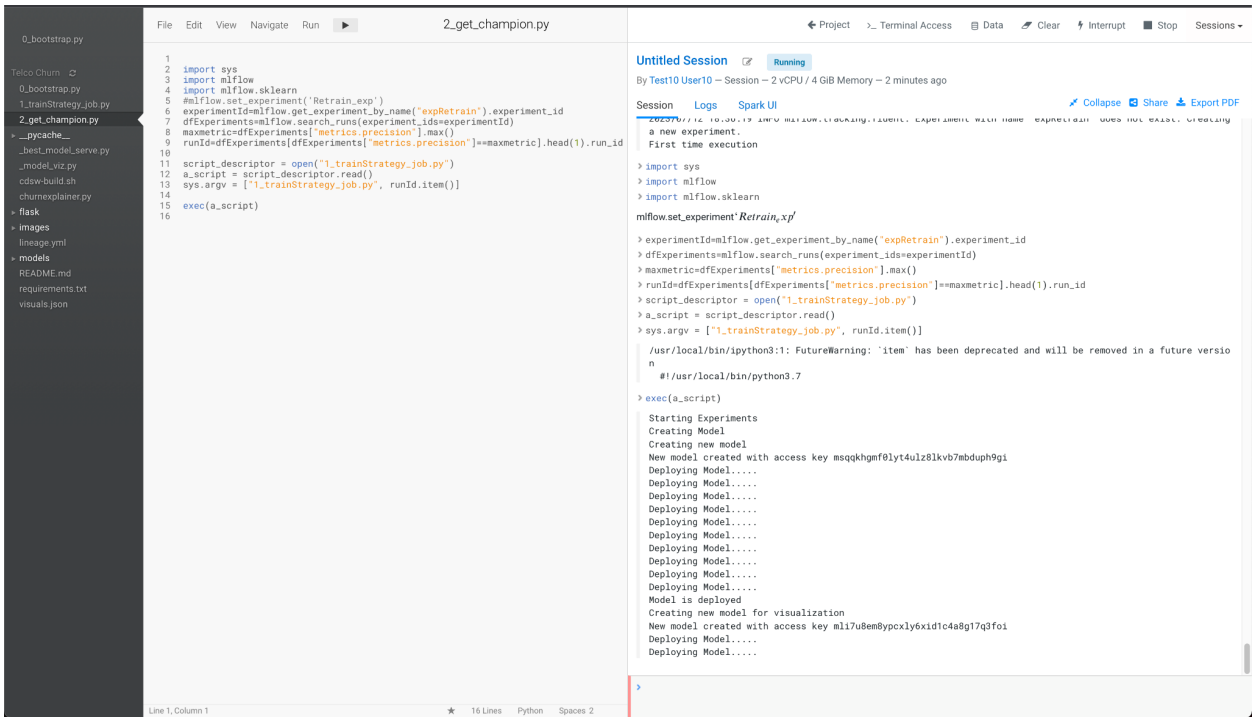
- Sessions on the left menu
- Click on session name: User0XX_Workbench_Session
- If you didn't name your session when you started it (step 6), it should be called *Untitled Session*.



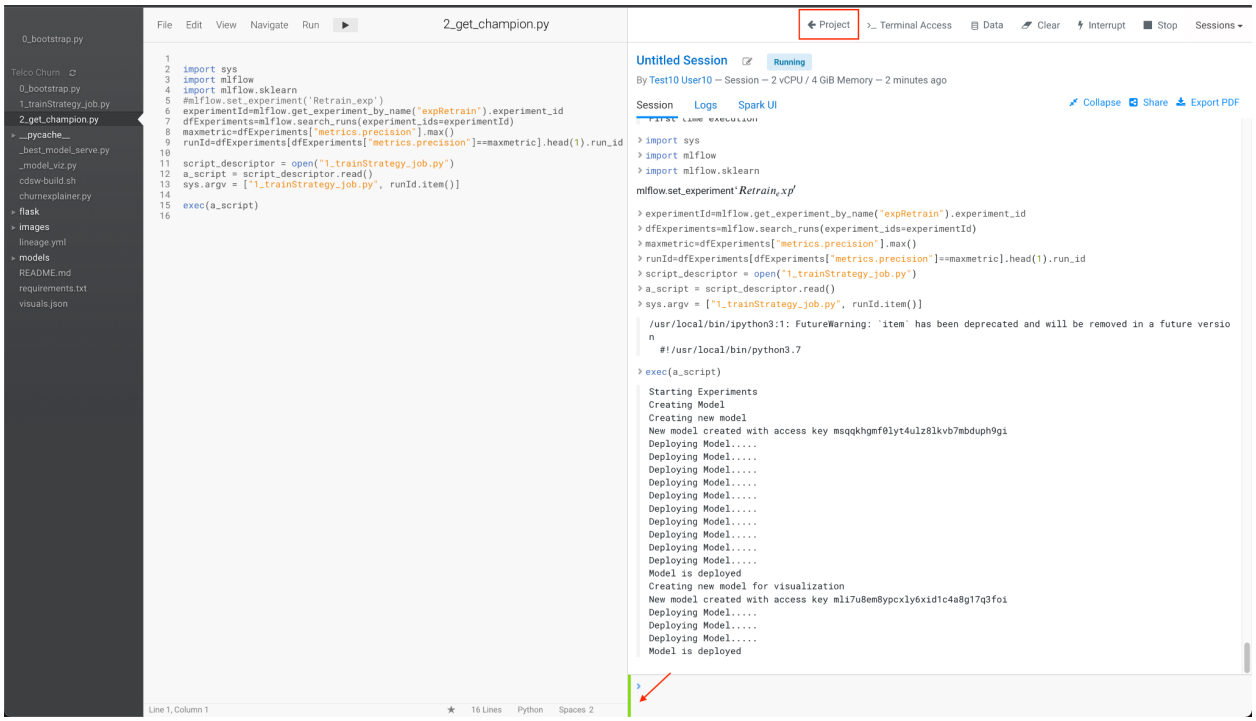
14. The third and last script/code to run is **2_get_champion.py**. This Python code takes the hyper parameter of the execution of the Experiment with better precision and deploys a Model as REST API, to be integrated in Data Visualization. Select (just one click) the file in the bar located on the left side of the interface, this will make the code appear in the editor. Once the file is selected, click on the button ▶ to run the code.



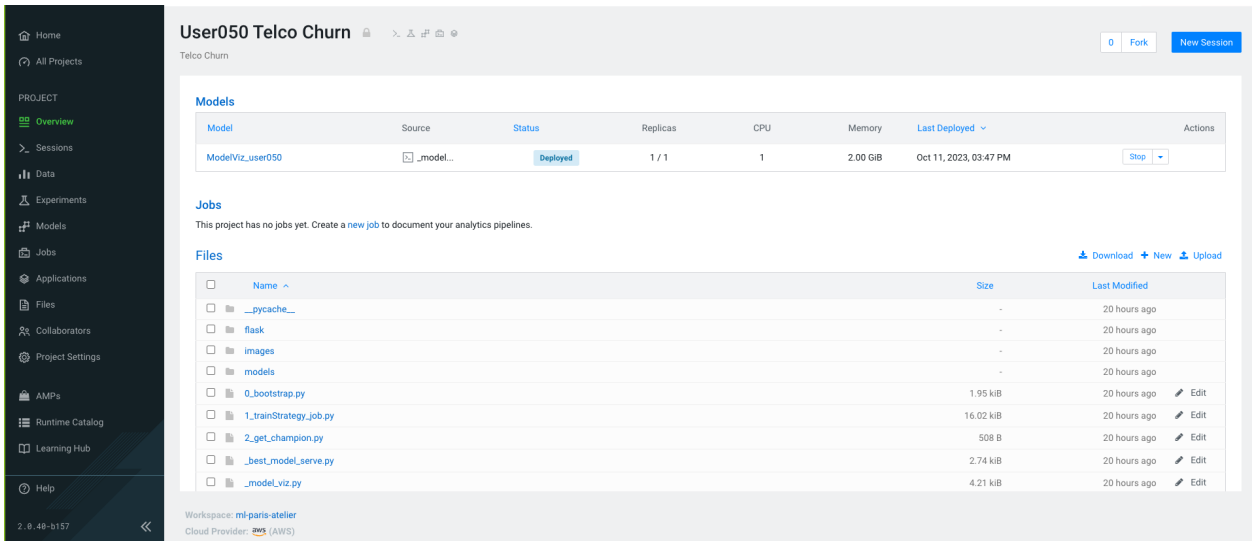
After a few seconds, you will see the following message “Deploying Model…” repeated several times, and the bottom command bar will be red.



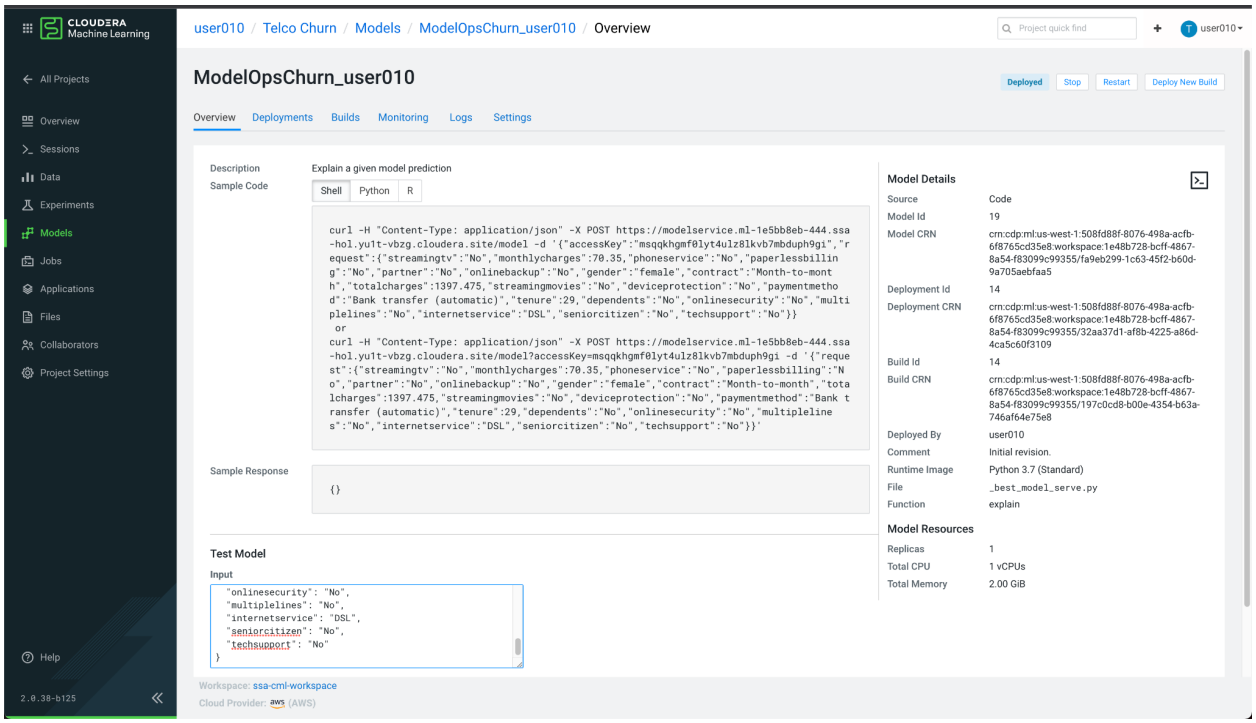
After about 2 minutes, the last message should be "Model is deployed", and the bar will be green. It means that the Deployment of the Model is complete.
Click on the button **Project**, located in the upper right bar of the session to return to the home page of the project.



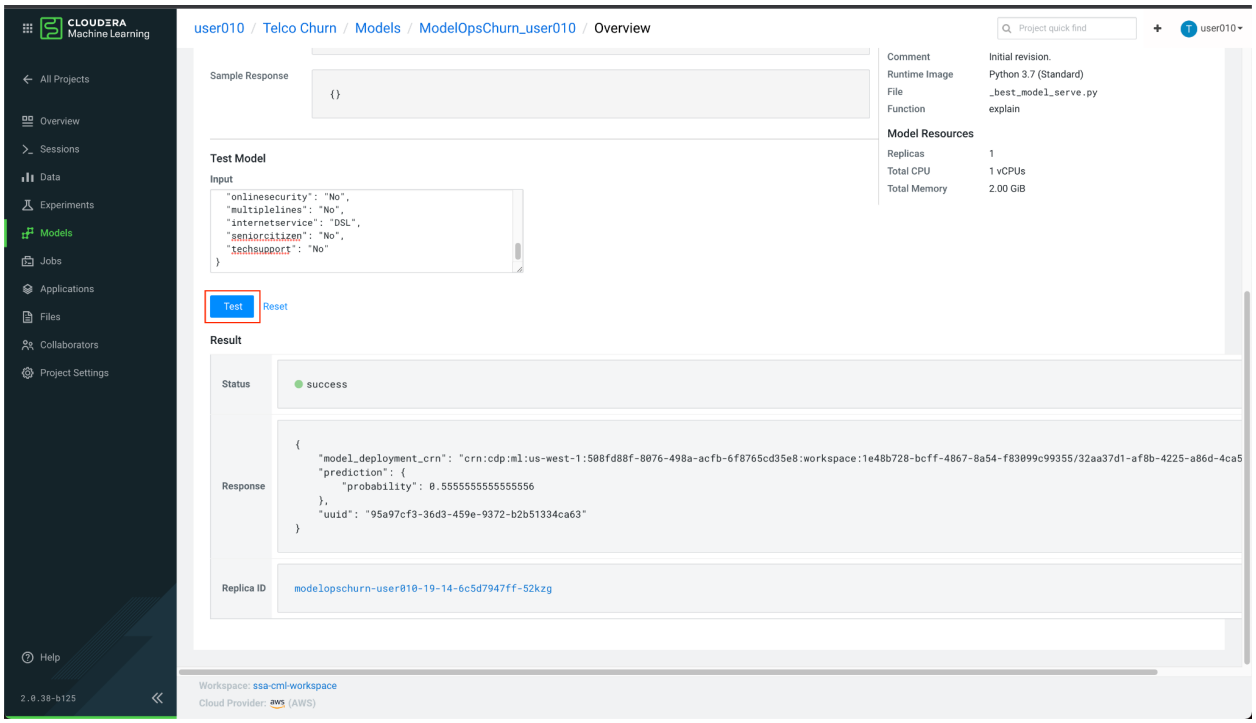
15. Once on the home page of the project, you will see the Model displayed. Click on the one that starts with **ModelViz_user0XX.**



16. Here you will see Model information and settings in the Overview tab.



To test it and make a request to the model, scroll down, and click on the button **Test**, which will take the value in JSON format that is in the field **Input** and will make the request call to the model. What you see in the field **Result** is the response from the model in JSON format. If you wish, you can change some of the parameters of the **Input** field (for example, change some values from *Not* to *Yes*), and call the model again, and observe the value of the attribute *probability* of the response to see if there were any changes.



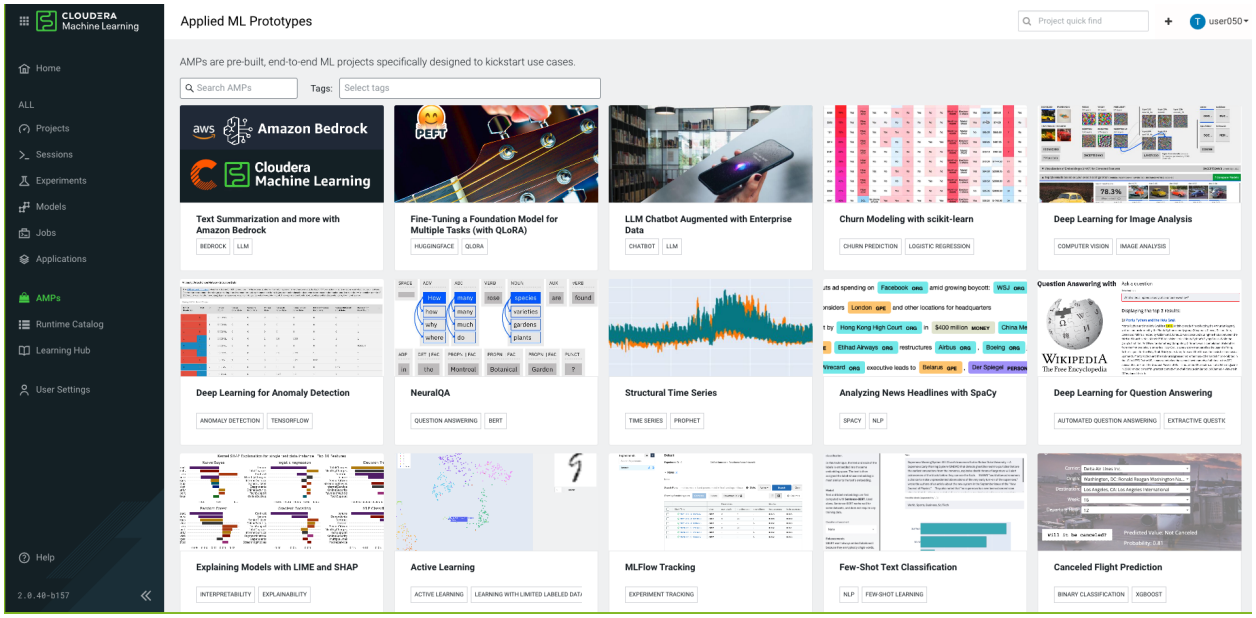
6. Optional Labs

6.1. Goals

- Deploy Applied Machine Learning Models
- Discover and query data using HUE
- Add a new field that makes calls to the ML model
- Add the new field to the dashboard

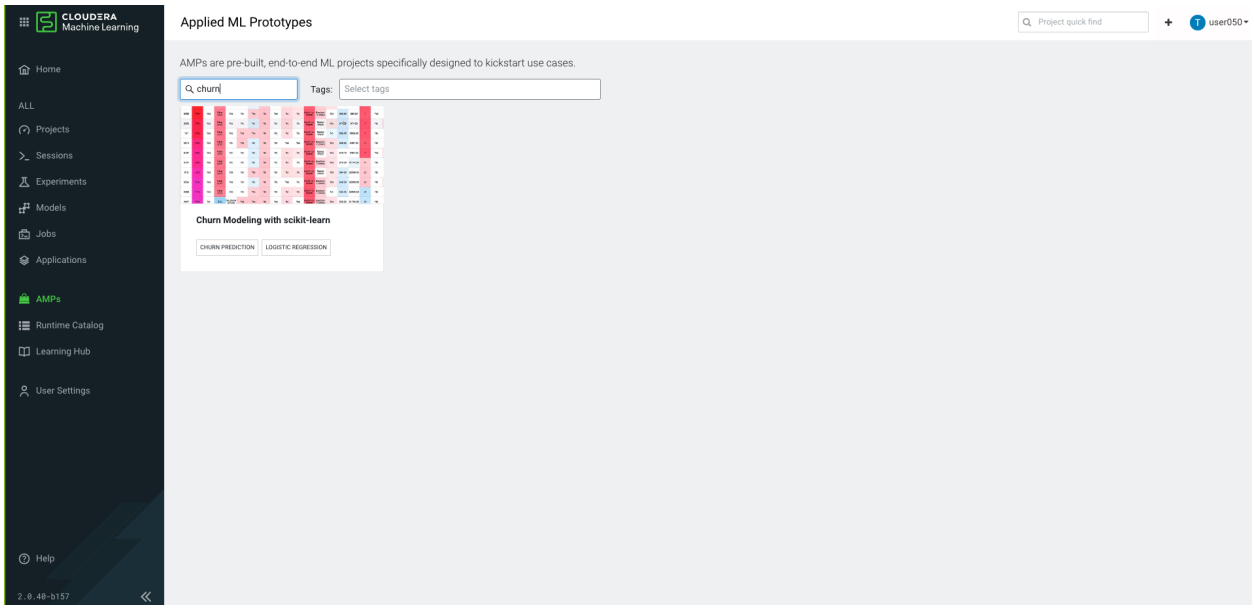
6.2. ML -Deploy Applied Machine Learning Model - Optional

1. In you Cloudera Machine learning home screen, click on AMPs on the left menu

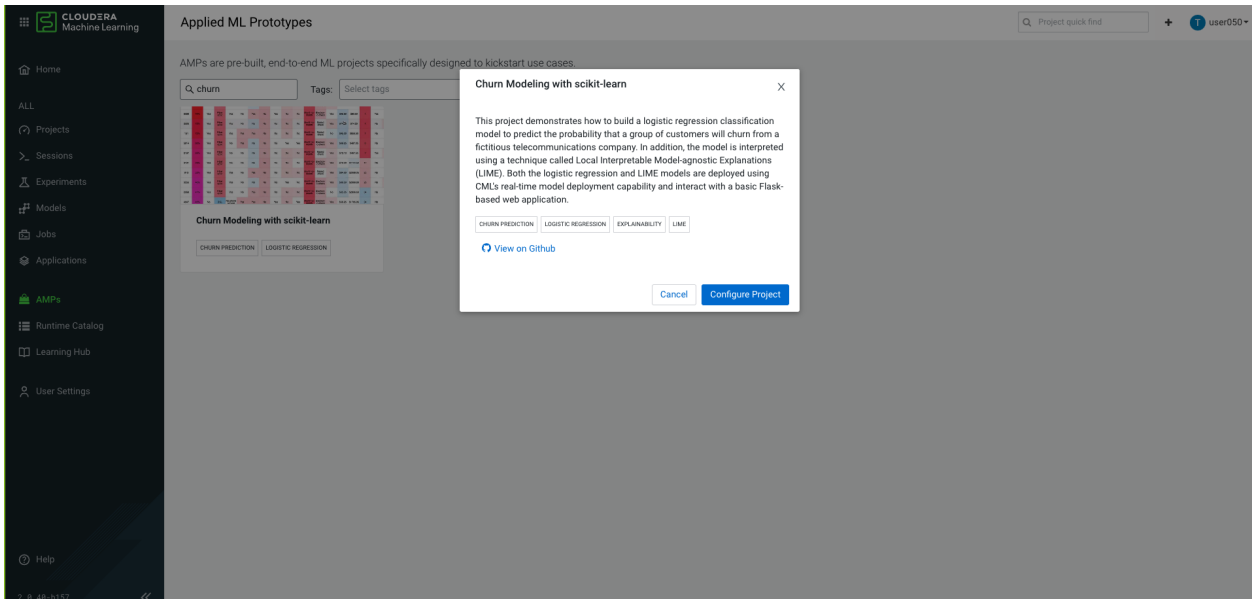


2. In the search bar on top, type: Churn

   A tile will filter: “Churn Modeling with scikit-learn

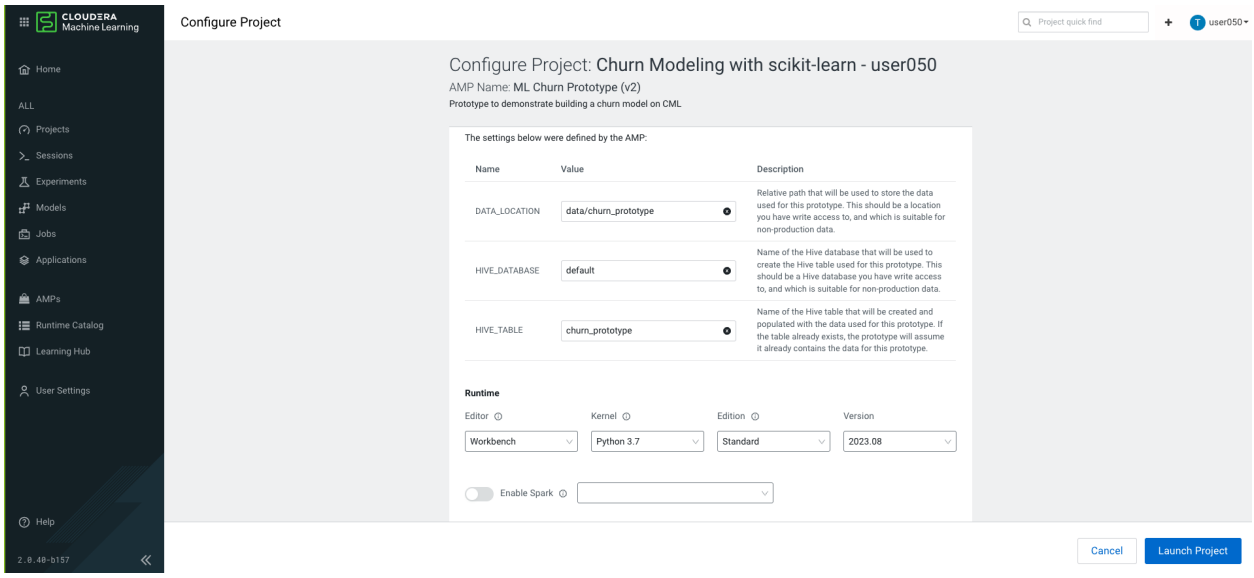


3. Click on the tile → Then click on Configure Project

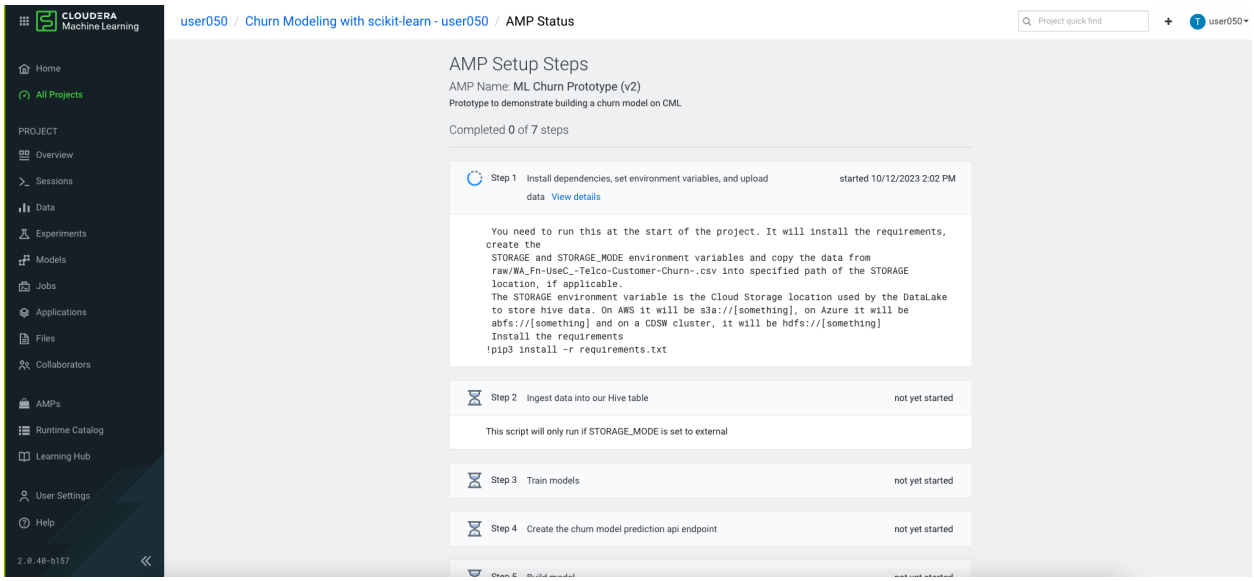


4. Project configuration screen launches. You can setup all the necessary steps for successful project configuration. Leave everything as-is.

   Do Not Click Launch Project - this will be done by the facilitator.

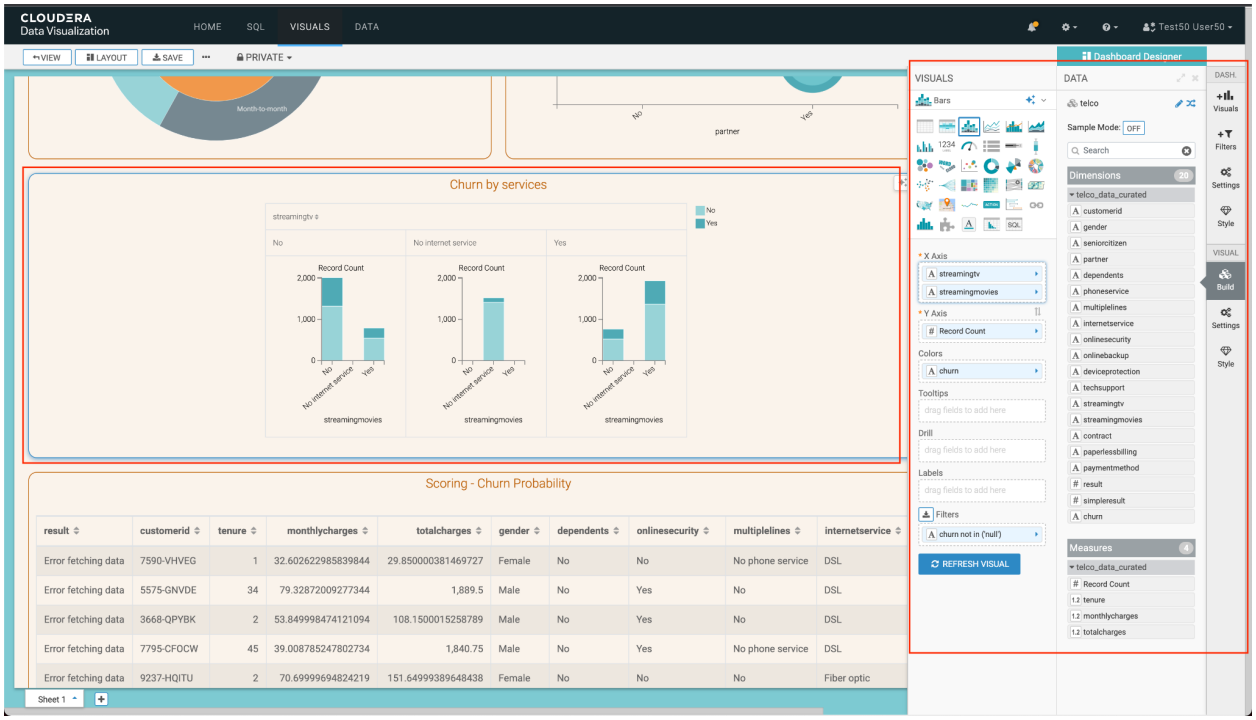


5. Deployment launches



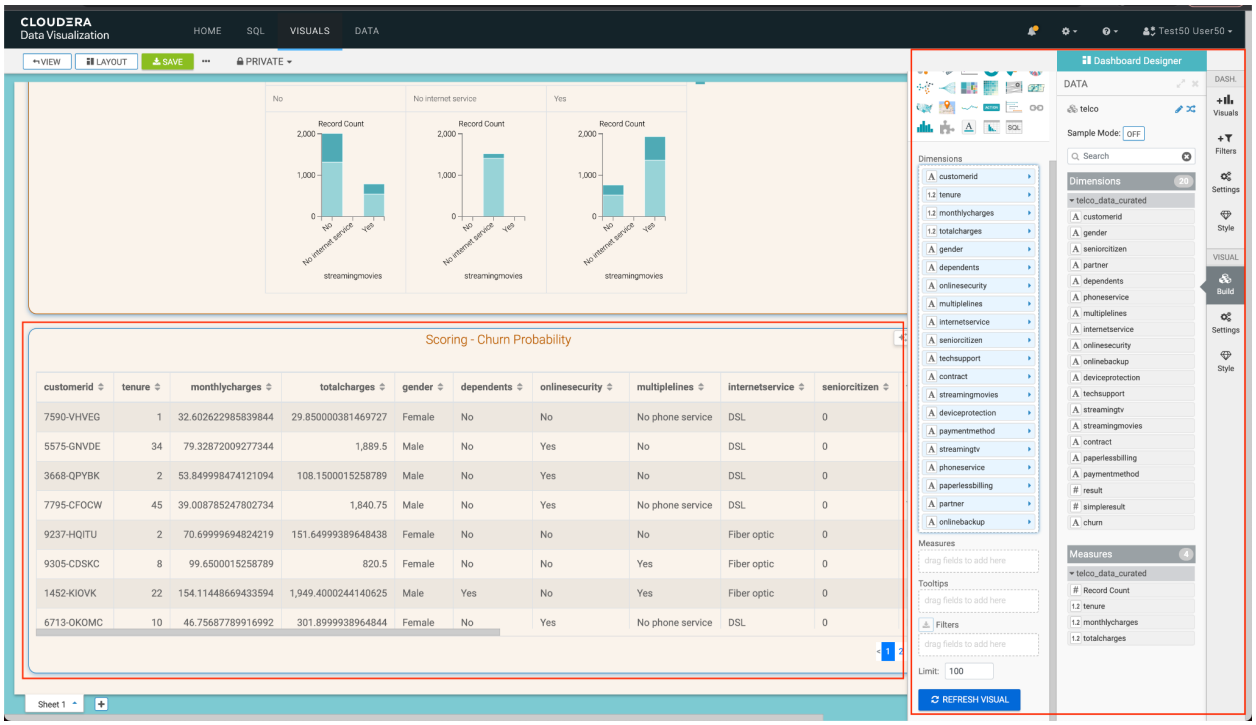
6.3. Add a third visual element - Optional

1. Add the third visual element, which is a bar chart with the dimensions **streamingtv** and **streamingmovies** like X Axis,
2. **Record Count** how Y Axis and **churn** how Colors. Once finished, click the button **Refresh Visual**.



6.4. One more visual element - Optional

1. If you do not want to call a machine learning model, please follow the below steps to add an extra element.
2. Add the fourth and last visual element, which is a table with the dimensions and metrics of the dataset. Be sure to add all 17 dimensions and 3 metrics to the table. Once finished, click the button **Refresh Visual**.

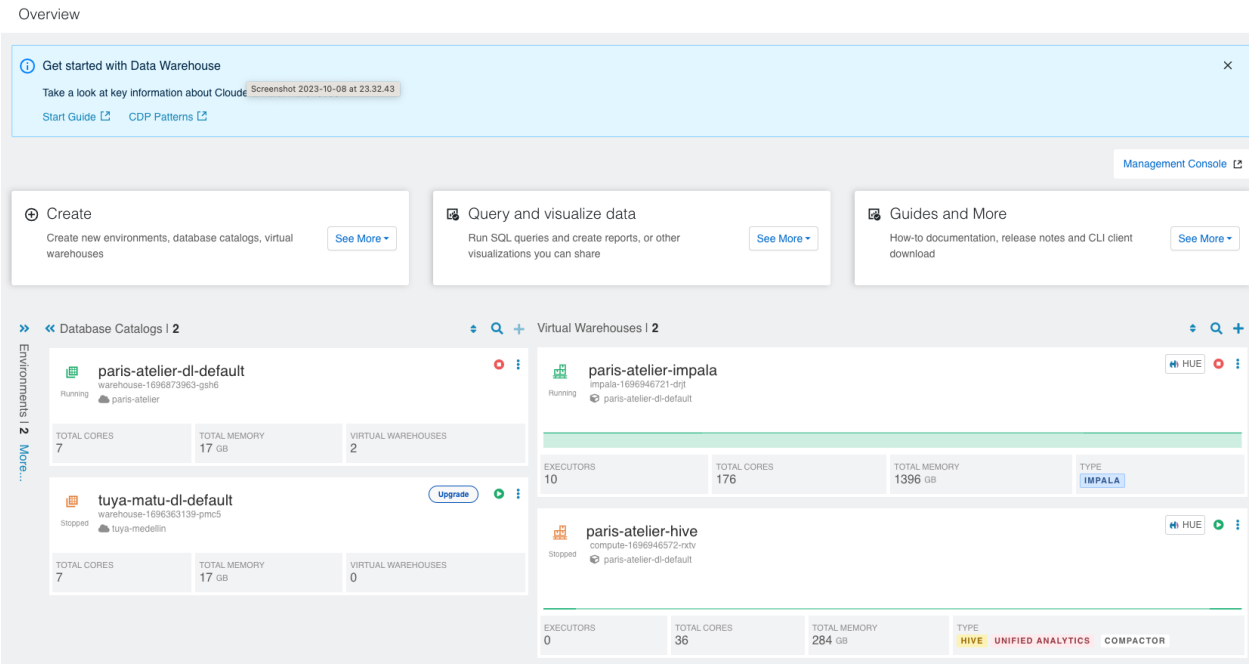


3. Save the dashboard by clicking the button **Save** from the top menu.

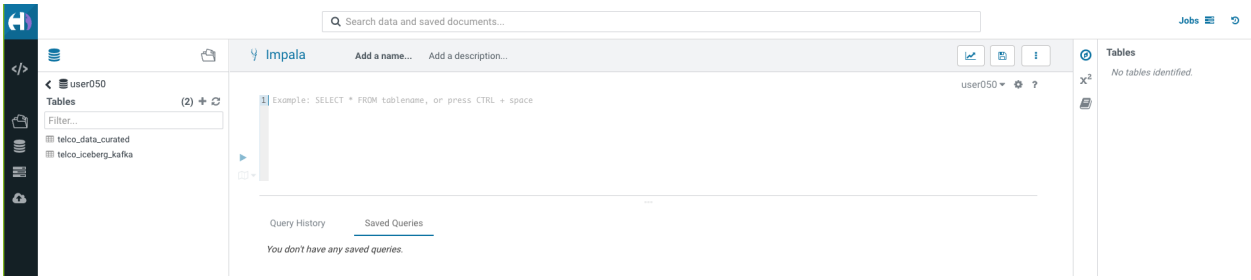
6.5. Data Discovery and SQL Analysis Using HUE Dashboard - Optional

On your Cloudera landing page, go do Data Warehouse.

1. Click on the **HUE** button, in your Data Warehouse dashboard.



2. This is your HUE tool.
   Hue is a web-based interface, simplifying data exploration, job scheduling, and management.
   It offers a user-friendly environment for running SQL queries, managing files, and visualizing data. Hue enhances productivity by providing a centralized platform for big data ecosystem components, making tasks easier and more accessible.



3. Inside your Hue window, run the following SQL statement:
   a. Run the following statement:

```
describe formatted user0XX.telco_data_curated
```

   You will get a visual description of your table as shown below:

	name	type	comment
1	# col_name	data_type	comment
2		NULL	NULL
3	multiplelines	string	NULL
4	paperlessbilling	string	NULL
5	gender	string	NULL
6	onlinesecurity	string	NULL
7	internetservice	string	NULL
8	techsupport	string	NULL
9	contract	string	NULL
10	churn	string	NULL
11	seniorcitizen	string	NULL
12	deviceprotection	string	NULL
13	streamingtv	string	NULL
14	streamingmovies	string	NULL
15	totalcharges	float	NULL
16	partner	string	NULL
17	monthlycharges	float	NULL
18	customerid	string	NULL
19	dependents	string	NULL

b. We are interested in a few properties, scroll to line 32. Notice the location

32	Location:	s3a://paris-atelier/my-data/warehouse/tablespace/external/hive/user050.db/telco_data_curated	NULL

c. Scroll to line 52. Notice the table_Type

52		table_type	ICEBERG

4. Inside your Hue window, run the following SQL statement:

```
describe history user0XX.telco_data_curated
```

	creation_time	snapshot_id	parent_id	is_current_ancestor
1	2023-10-11 09:22:59.302000000	77495325292598376	NULL	TRUE
2	2023-10-11 09:23:37.192000000	821975665367142062	77495325292598376	TRUE

Notice the snapshot history for your tables. Next we will go and query them.

5. Inside your Hue window, run the following SQL statement:

```
SELECT count (*)
FROM  user0XX.telco_data_curated
for SYSTEM_VERSION as of  <first_snapshot_id>
```

	count(*)
1	0

6. Inside your Hue window, run the following SQL statement:

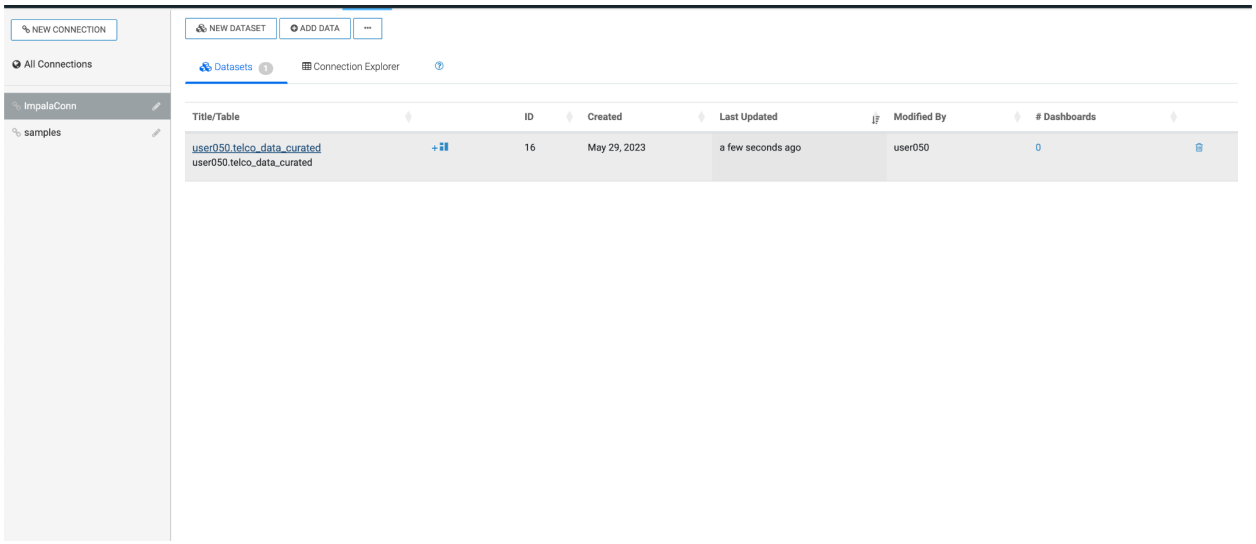
```
SELECT count (*)
FROM  user050.telco_data_curated
for SYSTEM_VERSION as of <last_snapshot_id>
```

	count(*)
1	7043

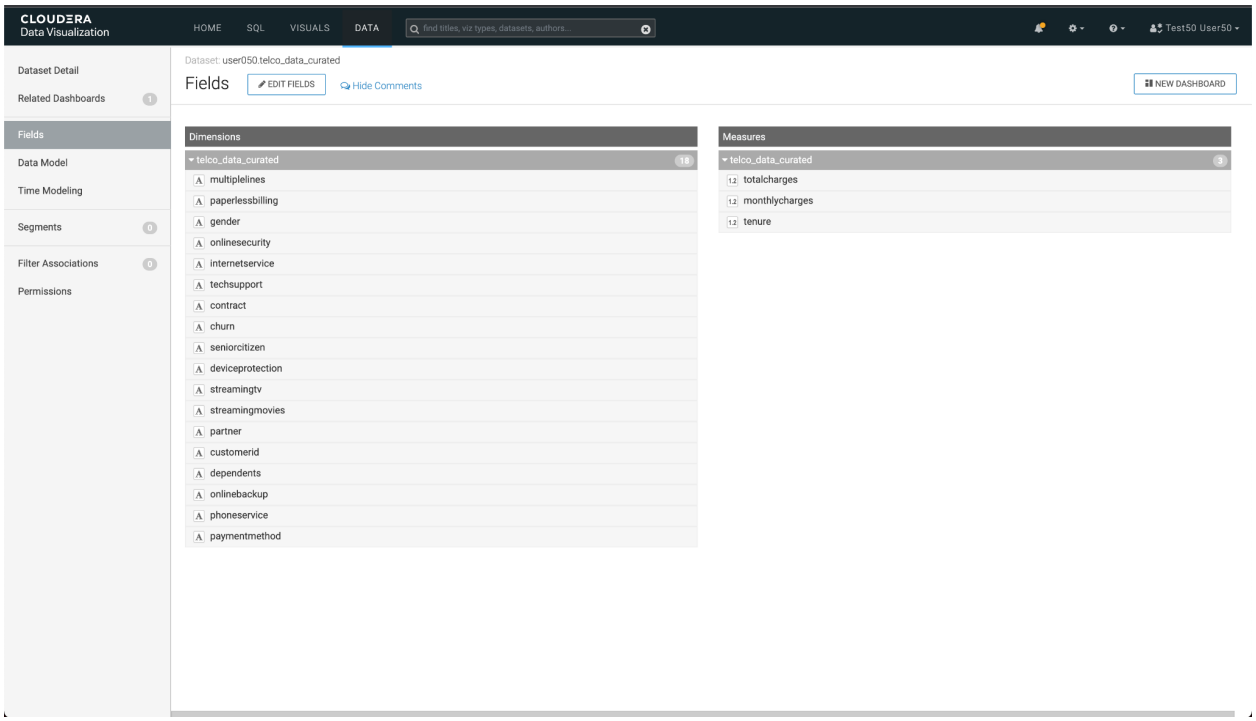
7. Explore with free SQL

6.6. Part 2: Add a New Field - Optional

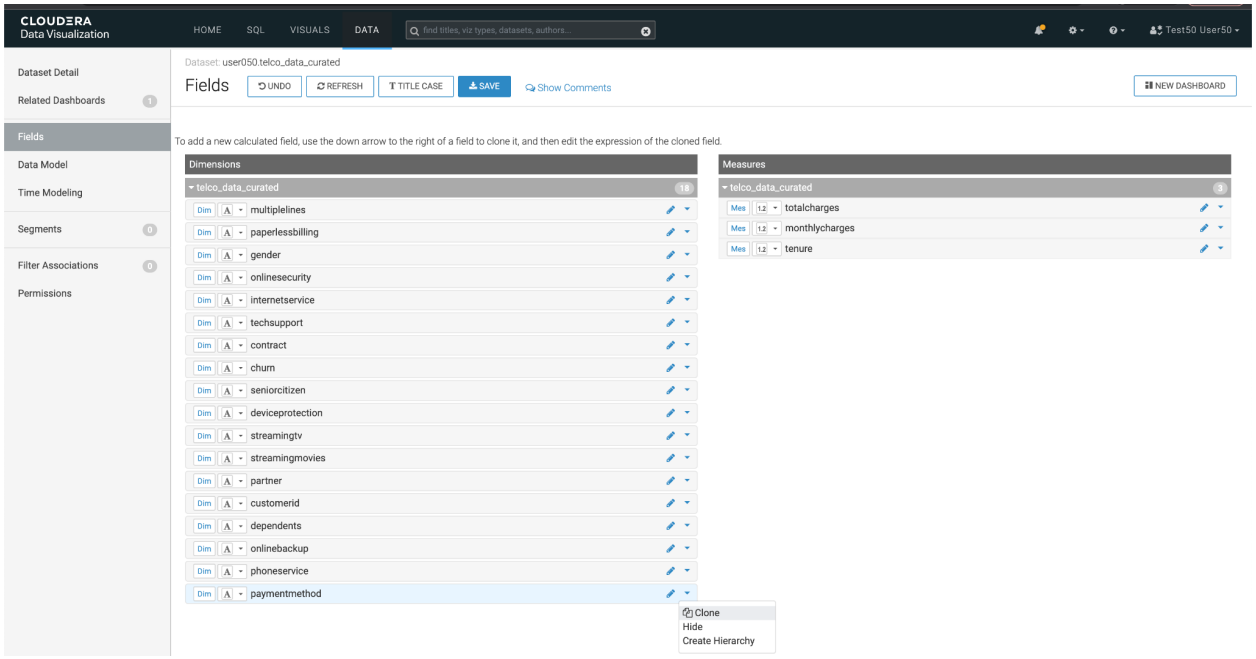
1. Edit the previously created Dataset, in Data -> <user_assigned>.telco_data_curated.



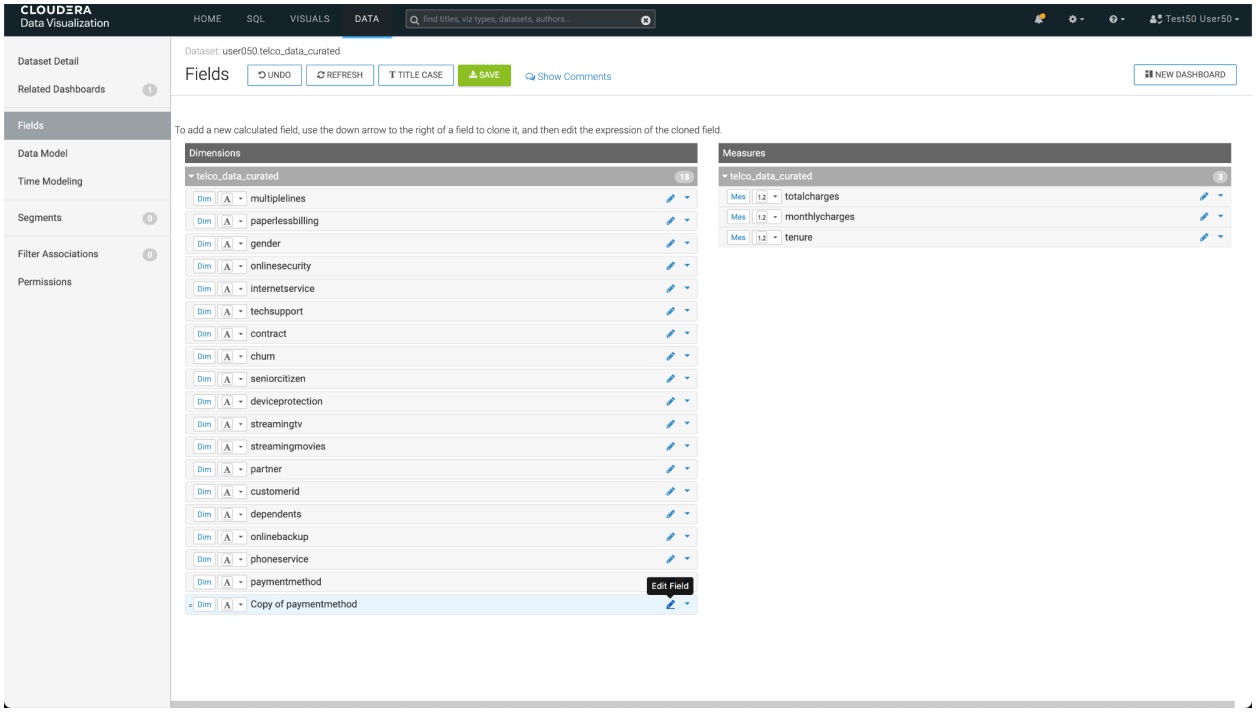
2. Once in the Dataset, go to **Fields** in the left menu and then click on **Edit Field** to edit the fields of your dataset.



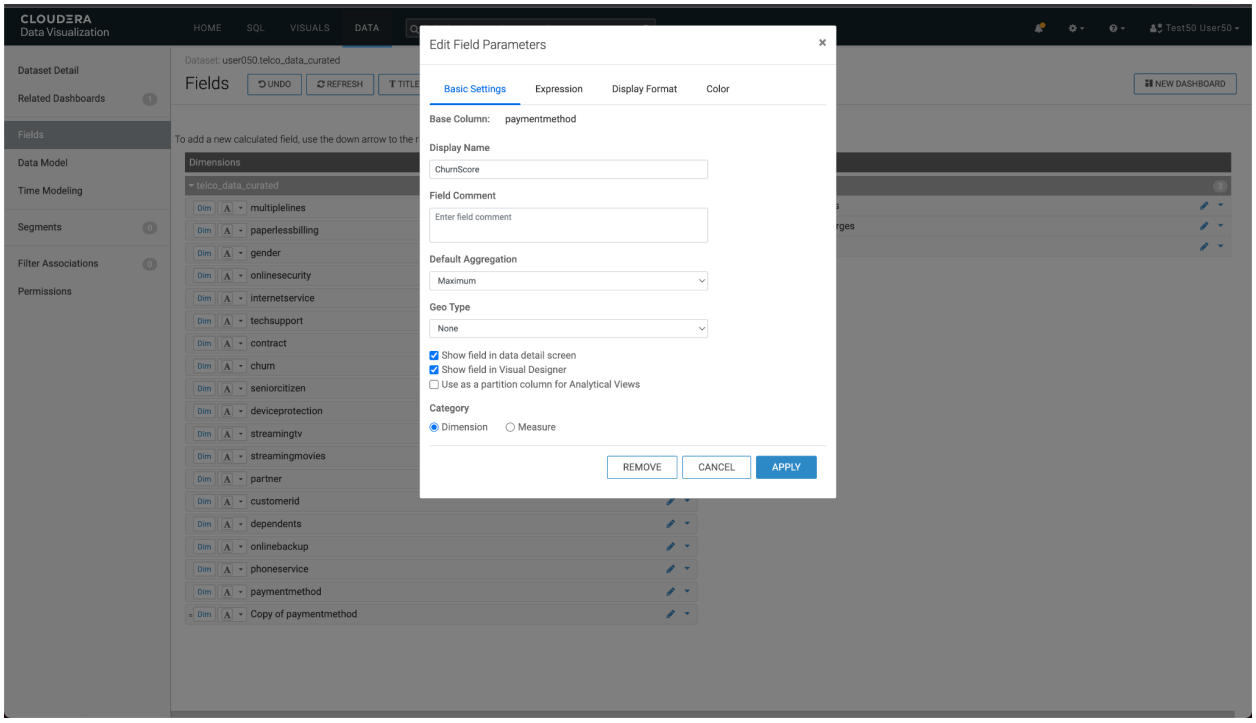
3. In the list of **Dimensions**, click the down arrow of the last field in the list, and select the option **Clone**.



4. Once the field is cloned, click on the pencil next to the field to edit it.

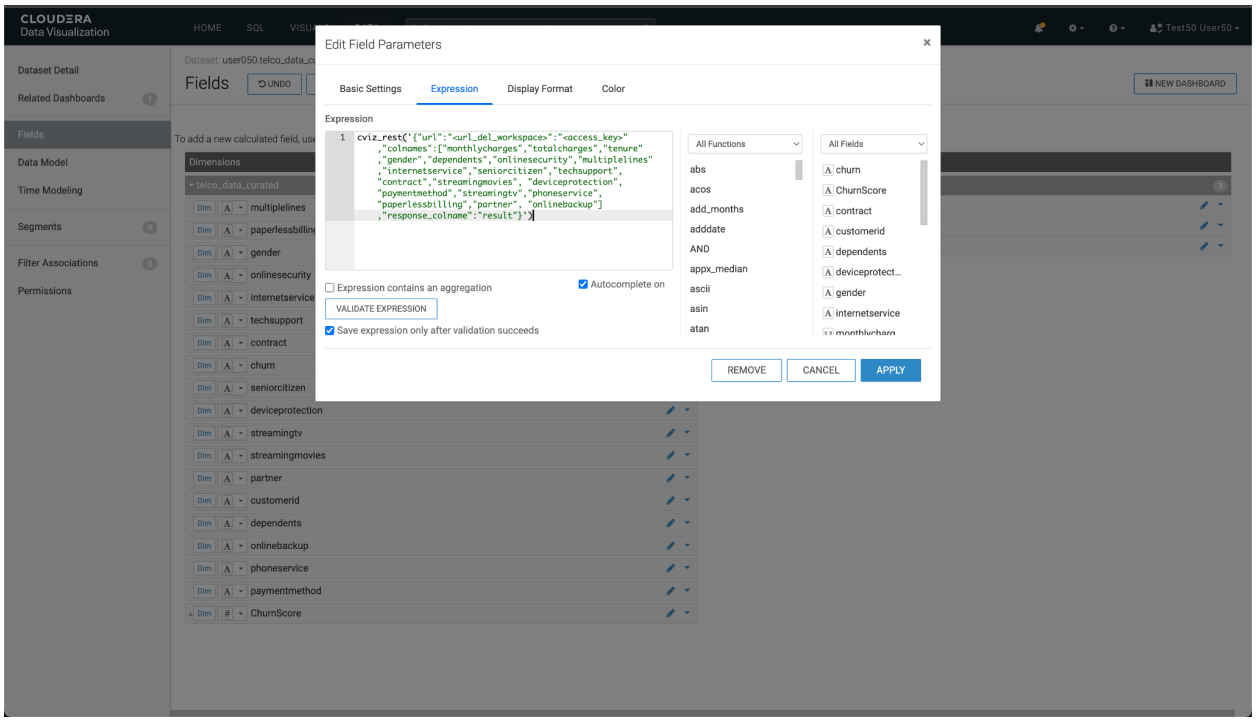


5. In the popup window that appears, enter the name of the new field in **Display Name**. We suggest that you enter *ChurnScore.*

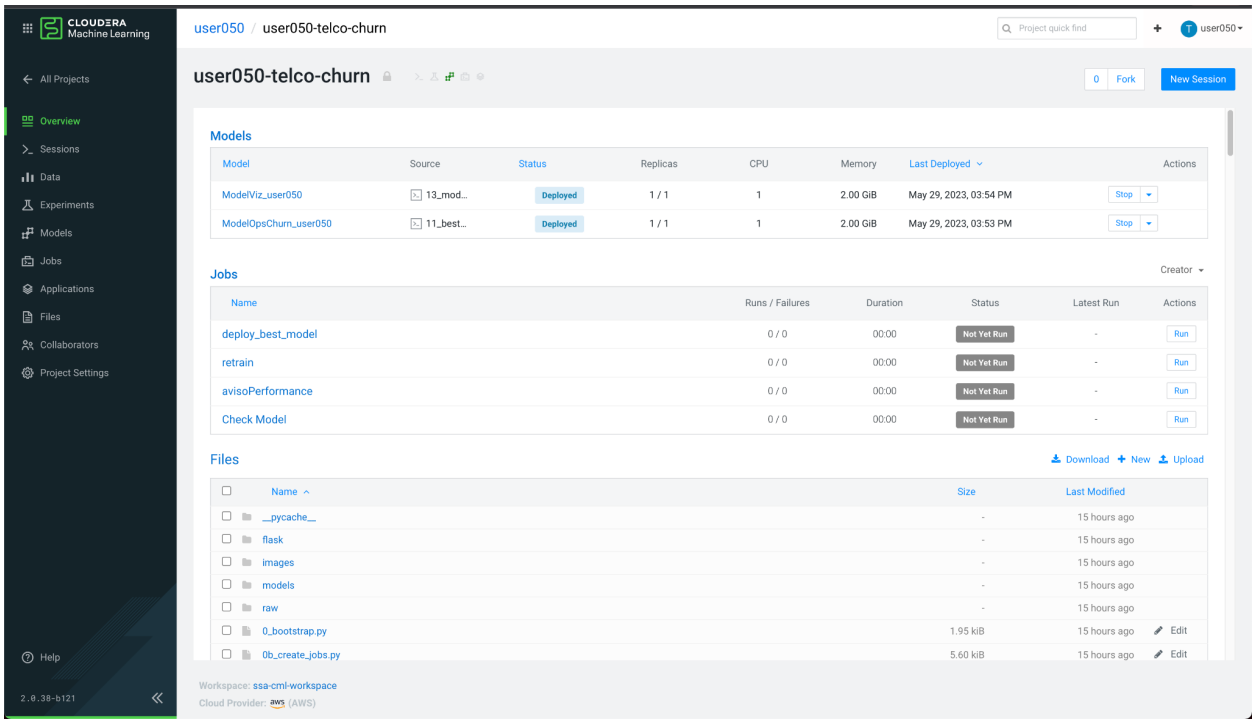


6. Go to the Expressions tab and enter the following value in the Expression field. This will allow you to call the REST API of the Model you have previously deployed.

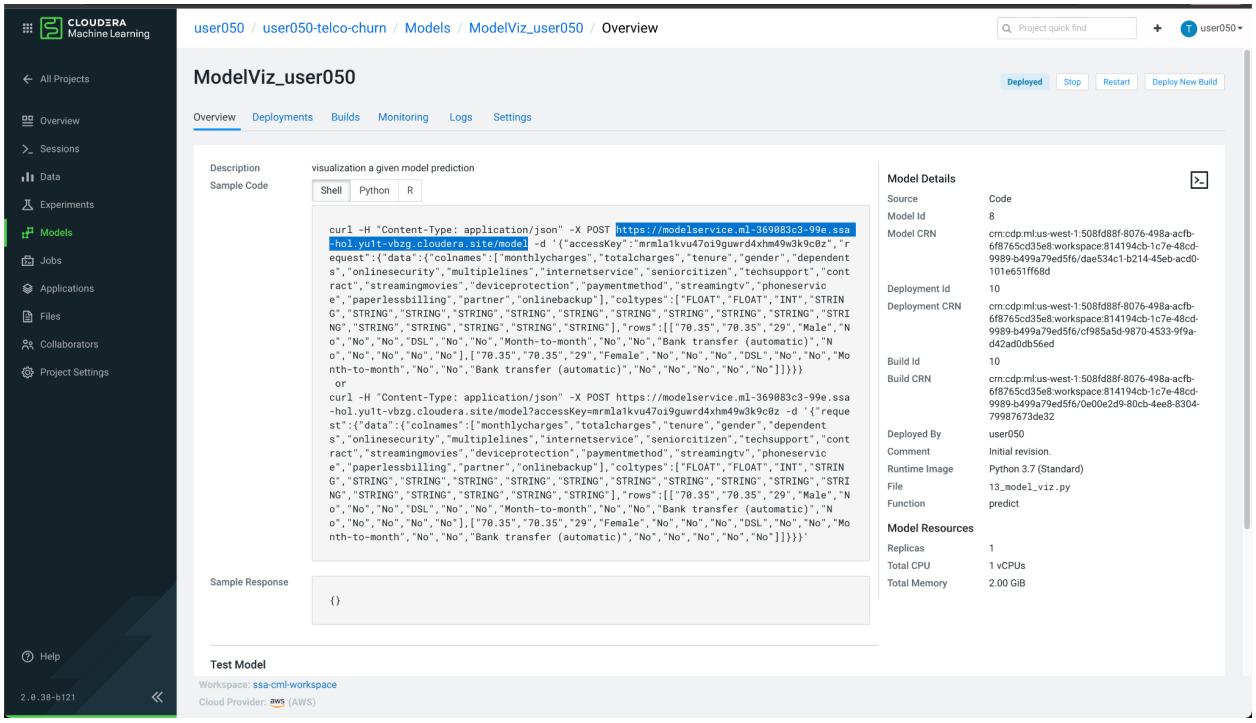
cviz_rest('{"url":"<url_del_workspace>","accessKey":"<access_key>","colnames":["monthlycharges","totalcharges","tenure","gender","dependents","onlinesecurity","multiplelines","internetservice","seniorcitizen","techsupport", "contract","streamingmovies", "deviceprotection", "paymentmethod","streamingtv","phoneservice", "paperlessbilling","partner", "onlinebackup"],"response_colname":"result"}')



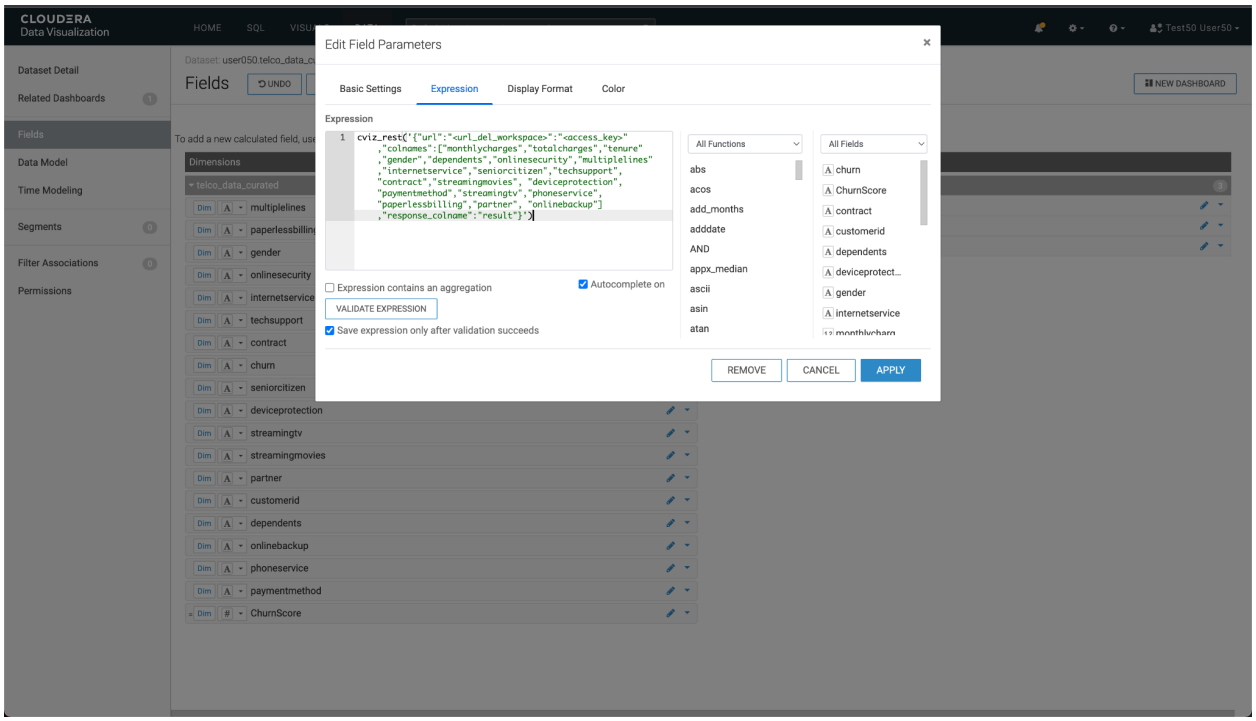
7. Being in CML in another tab of the web browser, go to the section of **Models** of your project, and click on the Model that begins with the name *ModelViz*, followed by your assigned username.



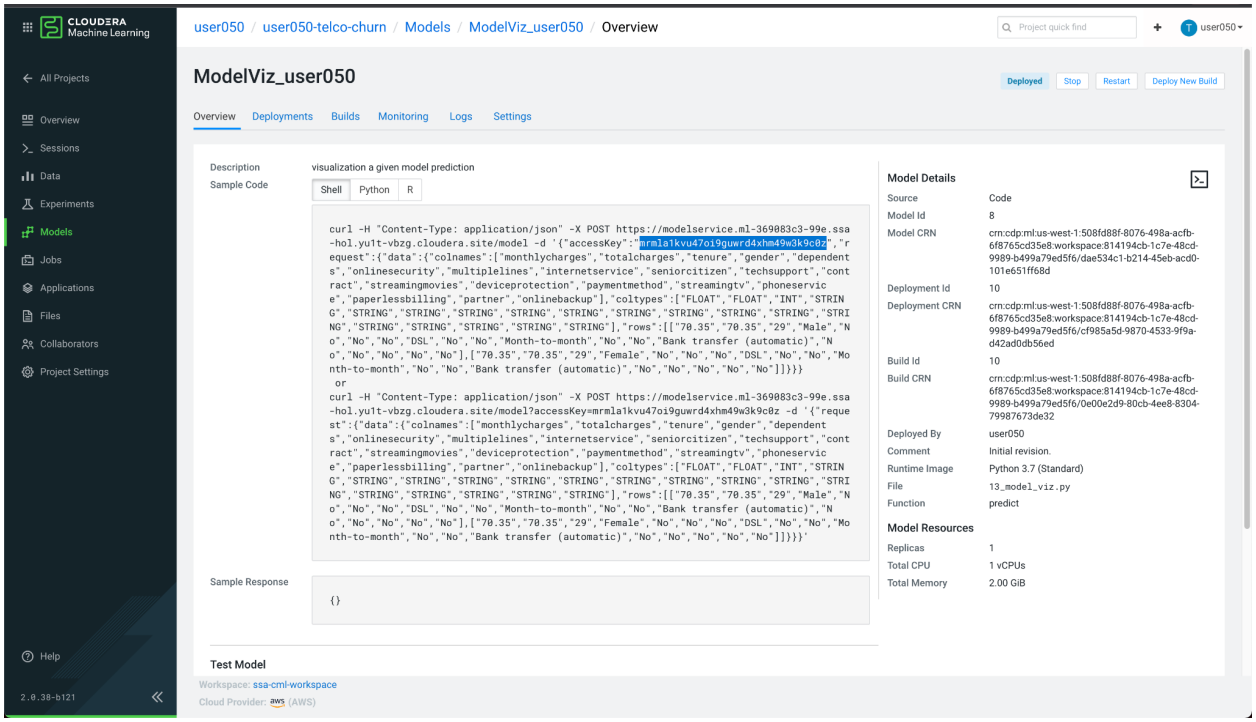
8. In the Overview tab, copy the URL that allows you to interact and call the workspace API.



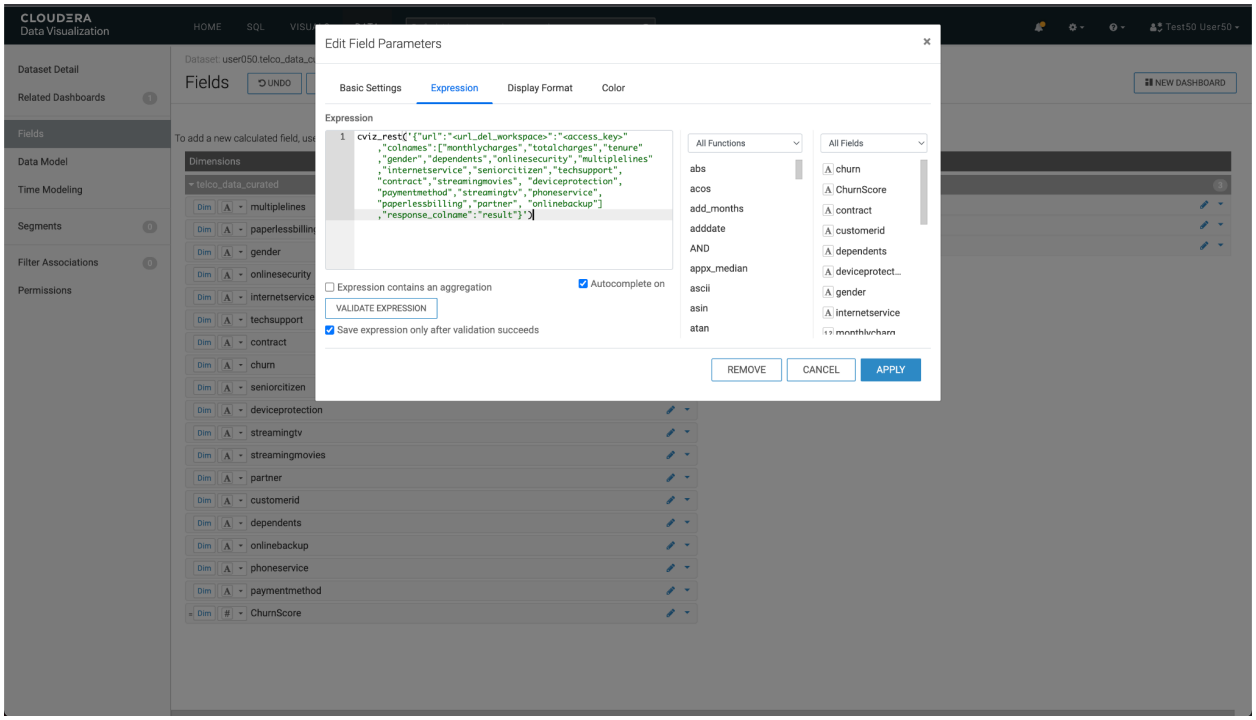
Replace the copied value in the attribute **<url_del_workspace>** of the Expression field.



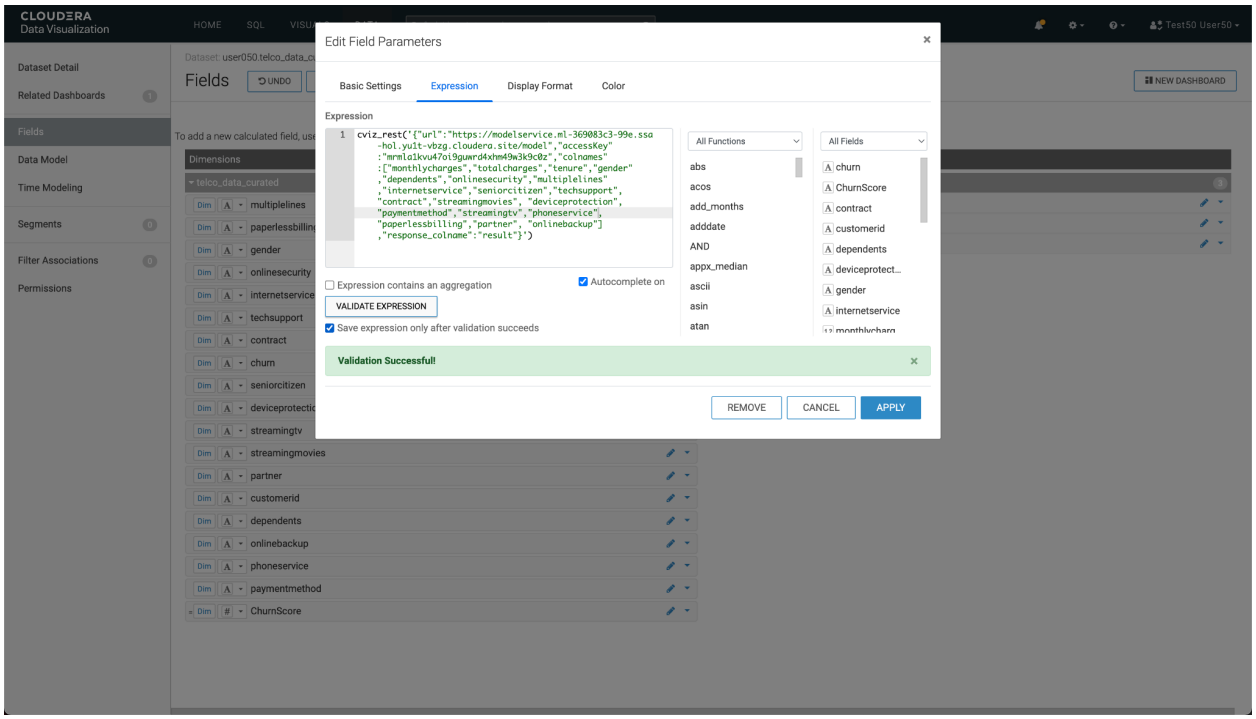
9. Returning to the CML, copy the accessKey of the model.



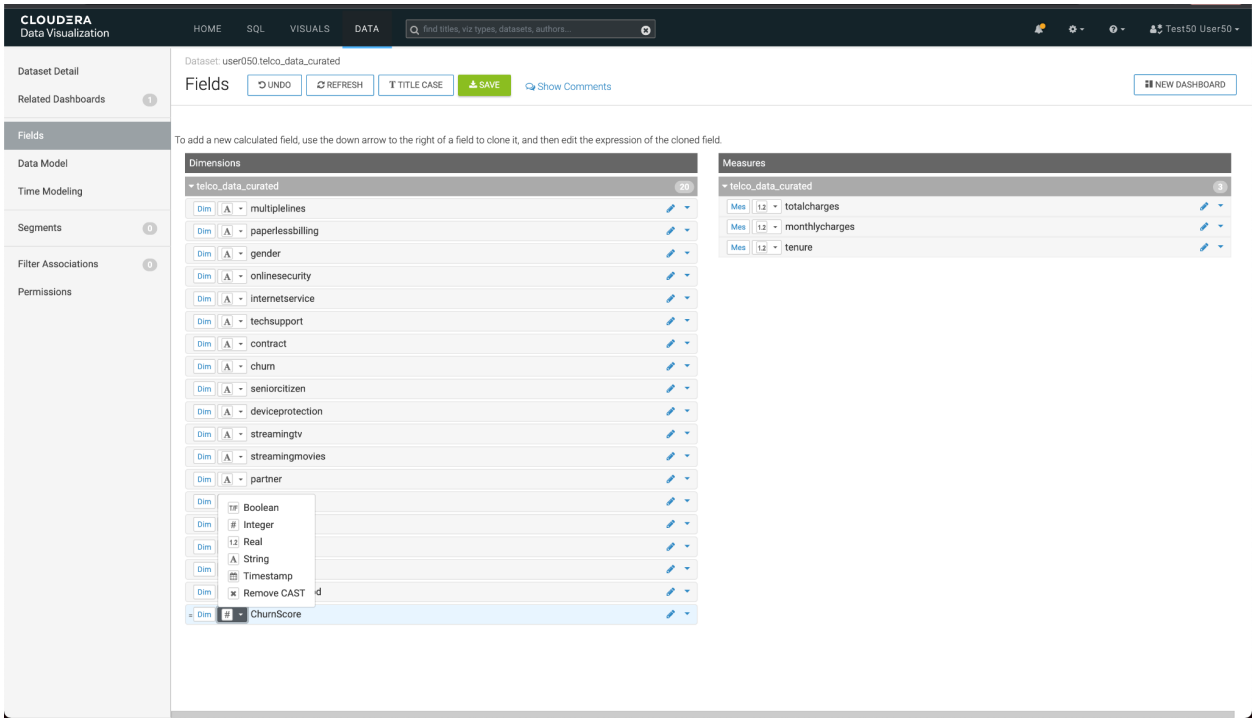
Replace the copied value in the attribute **<access_key>** of the Expression field.The format should be as follows, e.g.
cviz_rest('{"url":"https://modelservice.ml-b200bd6f-fb9.za-mtn-l.yu1t-vbzg.cloudera.site/model","accessKey":"mjy1fowabqiwpfjb19s9ht6xmuvy0f2j","colnames":["monthlycharges","totalcharges", "tenure","gender","dependents","onlinesecurity","multiplelines","internetservice","seniorcitizen","techsupport", "contract","streamingmovies", "deviceprotection", "paymentmethod","streamingtv","phoneservice", "paperlessbilling","partner", "onlinebackup"],"response_colname":"result"}')



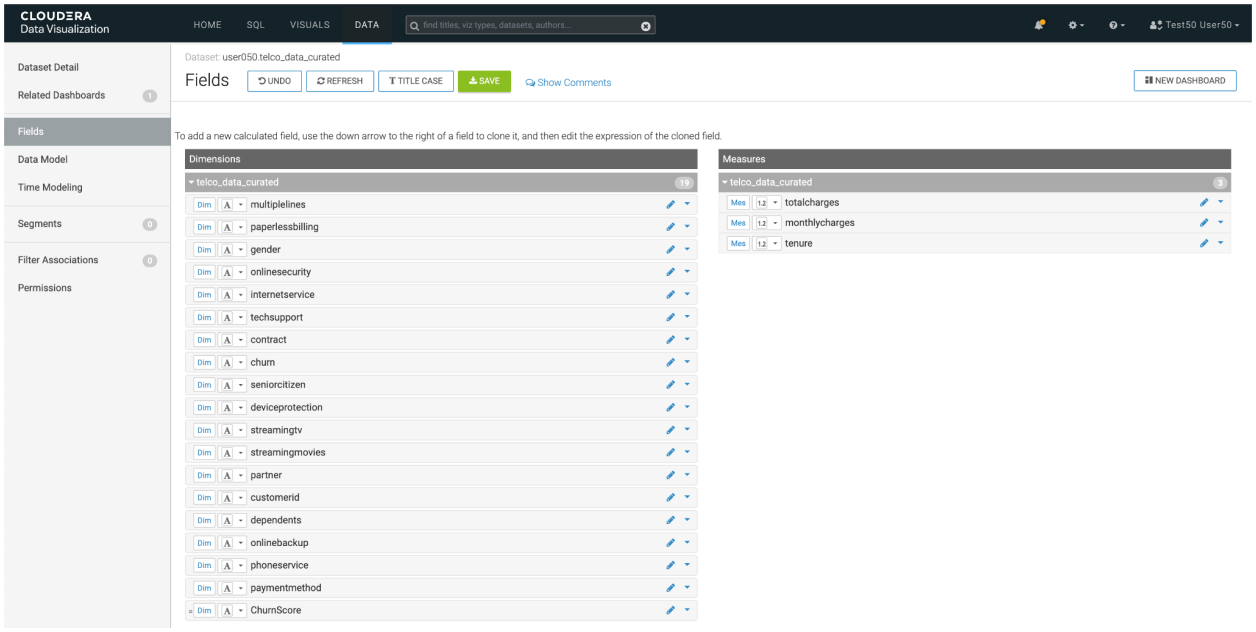
10. Finish the process of copying the *url del workspace* and the *accessKey*, click the Validate Expression button at the top of the window. If the message appears in green *Validation Successful*, Click on **Apply** to save the settings made.



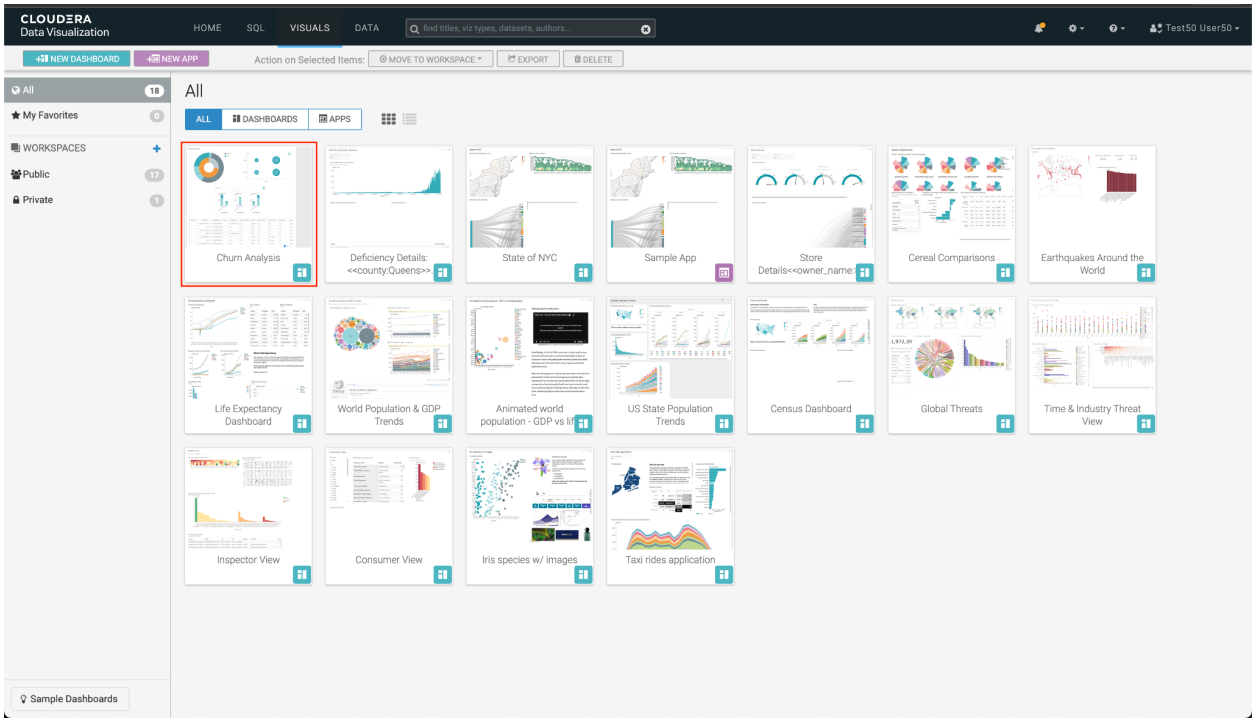
11. The new field should appear in the list of fields. Change the data type, selecting the type *Integer*, which is represented by the symbol **#**



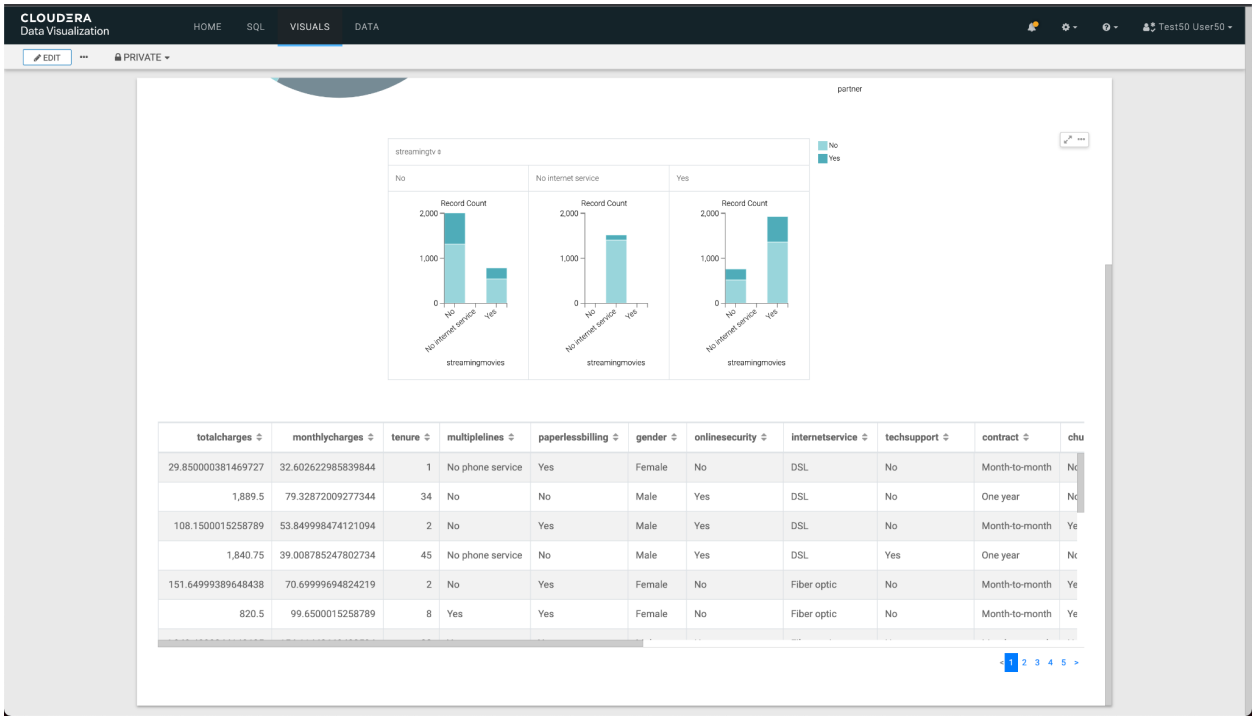
12. Finish the process by clicking on the green button with the legend **SAVE** in the top menu.



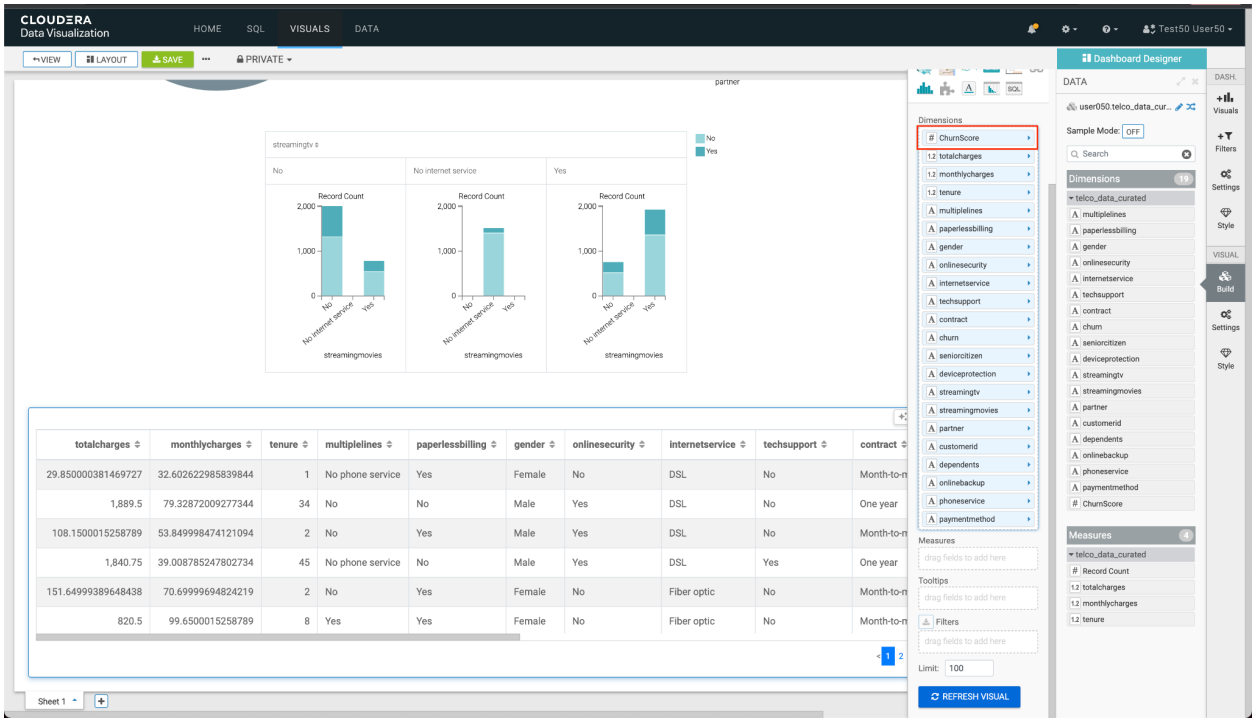
13. Return to the dashboard, selecting the option **VISUALS** from the top menu, and clicking on the name of the dashboard that was previously created.



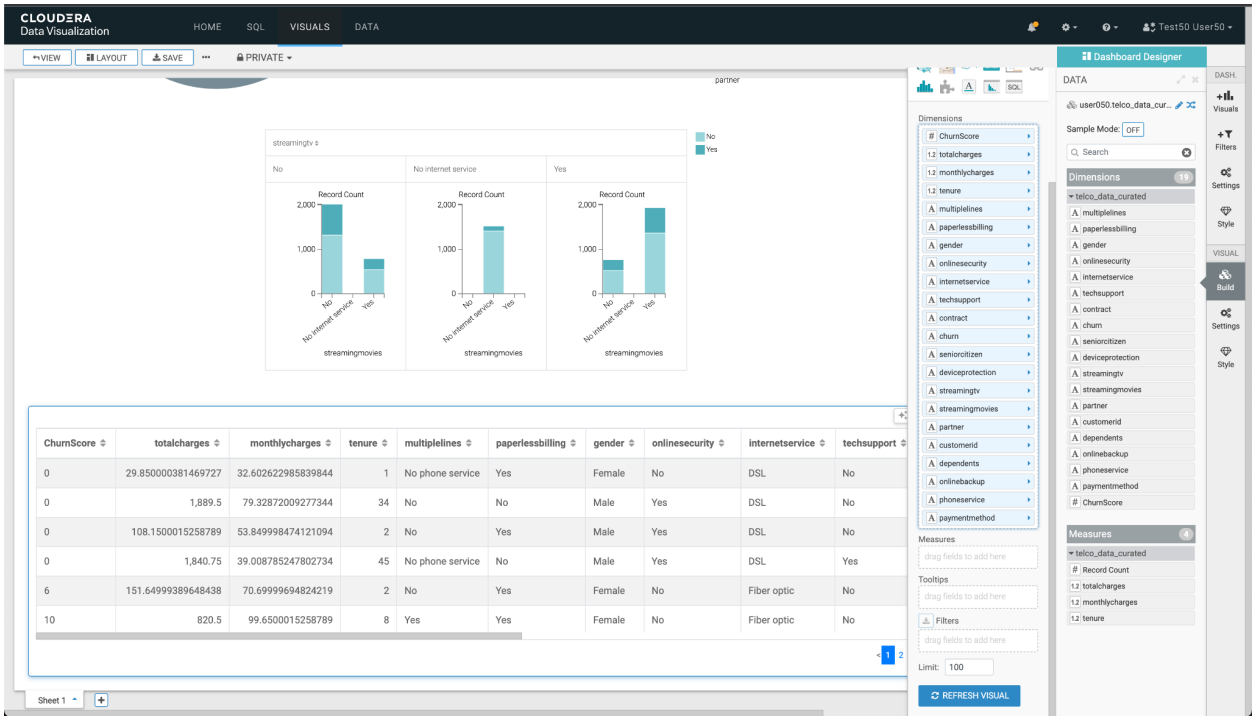
14. Once in the dashboard, click on the button **Edit** which is in the upper left.



15. Edit the lower table by clicking on it and then on the option **Build** from the right vertical menu. Add the new field,**ChurnScore**, at the beginning of the table, by clicking and dragging from the option **Dimensions** available.



16. Click on the Refresh Visual button to update the data. The new column should appear *ChurnScore* then at the beginning of the table, with a value of numeric type. Finish the process by clicking the button **SAVE** from the top left menu.



7. Take-aways

Cloudera Data Platform (CDP) is a hybrid data platform designed for unmatched freedom to choose—any cloud, any analytics, any data.
CDP delivers faster and easier data management and data analytics for data anywhere, with optimal performance, scalability, and security.
With CDP you get all the advantages of CDP Private Cloud and CDP Public Cloud for faster time to value and increased IT control.

