



## Les Ateliers Cloudera

# Paris Public Cloud Hands on Lab Workshop Guide

18 Jan 2024

Jacques Marchand

Charles Aad

Olivier Meignan

Patrick Cousin

## TABLE OF CONTENTS

Foreword	2
1. Introduction	3
○ Preliminary steps	3
2. Data Flow Lab:	6
○ Goals	6
○ Lab 1 - Ingest Kafka streams to Iceberg table	6
○ Lab 2 - Stream Messaging Manager - Optional	20
3. Data Engineering	23
○ Goals	23
○ Lab 1 - Enrich the Ingested Iceberg table	23
4. Data Warehouse	34
○ Goals	34
○ Dashboard Development	34
5. Machine Learning	43
○ Goals	43
○ Create a Machine Learning Model for Churn Prediction	43
6. Optional Labs	58
○ Goals	58
○ ML -Deploy Applied Machine Learning Model (Instructor Only)	58
○ Add a third visual element - Optional	60
○ One more visual element - Optional	61
○ Data Discovery and SQL Analysis Using HUE Dashboard - Optional	62
○ Part 2: Add a New Field - Optional	65
7. Take-aways	76

---

# 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 2023	Charles Aad	Document merge, sanity check
15 Jan 2024	Charles Aad	Update
16 Jan 2024	Jacques Marchand	Update

---

# 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.

## ○ Preliminary steps

A workshop has been designed and set up for you. Please connect using the below link:

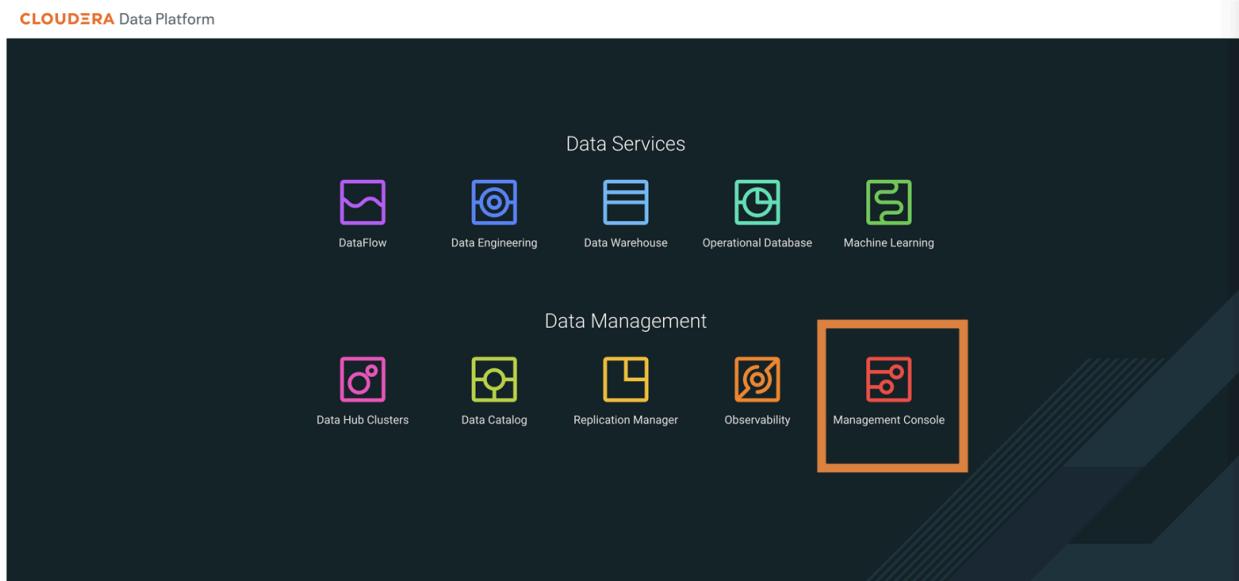
<https://login.cdpworkshops.cloudera.com/auth/realms/field-marketing-emea/protocol/saml/clients/cdp-sso>

username (use the one assigned to you): user0XX

password: G0yvxvdms5srhyKF

**Important:** Please keep this user assigned to you (user0XX) , you will need it in the future

Now you have to set the password for your workload environment.



- Go to User Management
- Search for the user the team has assigned to you: (example user050)

A screenshot of the Cloudera Management Console's User Management page. The left sidebar shows navigation links like Dashboard, Environments, Data Lakes, User Management (which is selected and highlighted in red), Data Hub Clusters, Data Warehouses, ML Workspaces, Classic Clusters, Audit, Consumption, Shared Resources, Global Settings, Help, and a Test50 User50 entry. The main content area is titled "User Management" and shows a table of users. The table has columns: Type, Name, Email, Identity Provider, Workload User Name, and Password Expiring. One row is visible, showing "Test50 User50", "user050@localhost.com", "marketing-events1-keycloak-idp", and "user050". At the bottom of the table, it says "Displaying 1 - 1 of 1 &lt; 1 &gt; [25 / page ▾]".

- Click on your user name
- Then click on Set Workload Password

Users / Test50 User50

Name	Test50 User50
Email	user050@localhost.com
Workload User Name	user050
CRN	crn:altus:iam:us-west-1:5a134a91-0505-4518-9bf1-89f324463e18:user:bf7b...
Tenant ID	5a134a91-0505-4518-9bf1-89f324463e18
Identity Provider	marketing-events1-keycloak-idp
Last Interactive Login	10/17/2023 11:26 AM CEST
Profile Management	<a href="#">View profile</a>
Workload Password	<a href="#">Set Workload Password</a> (Workload password is currently set)

Access Keys Roles Resources Groups SSH Keys

No access keys found.

Generate Access Key

- Set the password (**Paris2024**),
- click on Set Workload Password.

Users / Test50 User50 / Workload Password

\* Password  
\*\*\*\*\*

\* Confirm Password  
\*\*\*\*\*

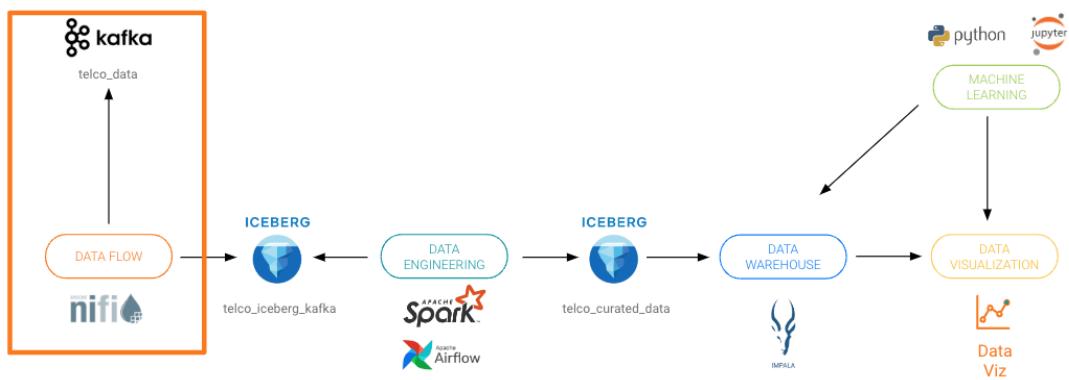
If you use keytabs, you need to regenerate them after changing your workload password. You can do this from your user profile > Actions > Get Keytab.

Set Workload Password

## 2. 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

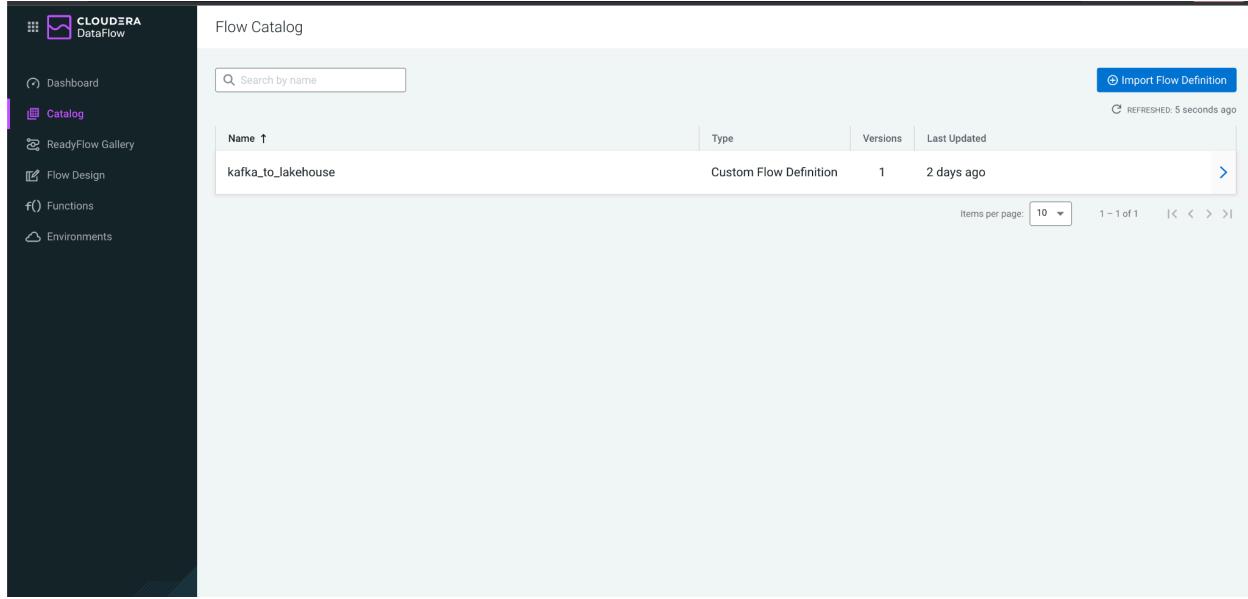


### ○ Lab 1 - Ingest Kafka streams to Iceberg table

#### 1. Click on DataFlow from CDP PC Home:

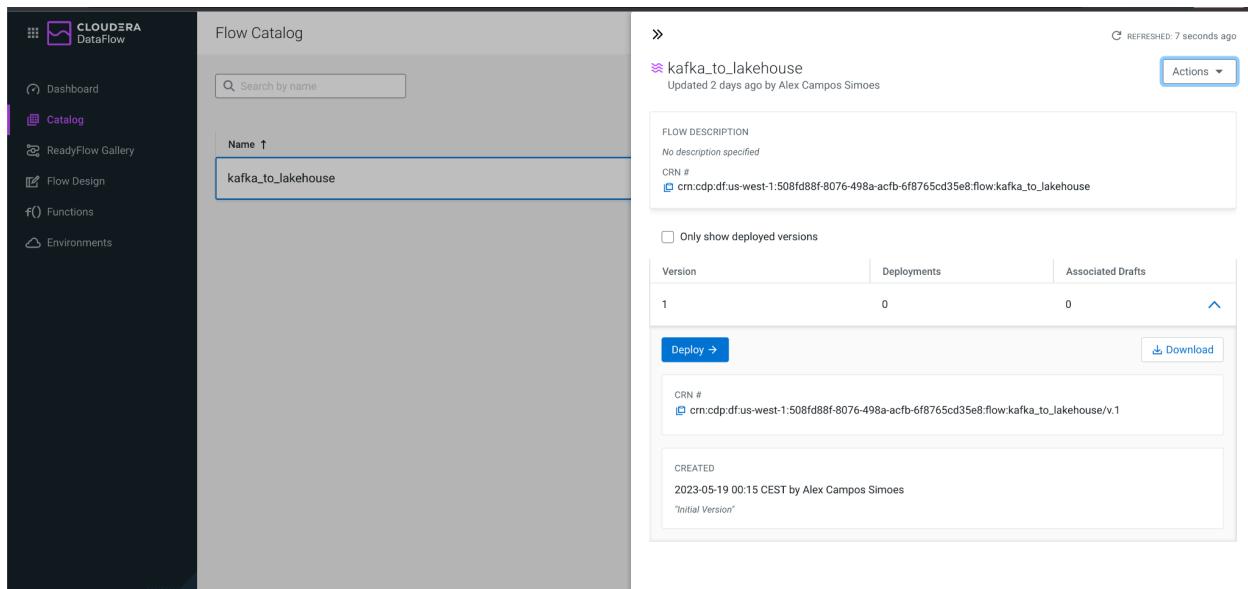
The screenshot shows the Cloudera Data Platform (CDP) PC Home interface. At the top left, the Cloudera logo is visible. Below it, the main navigation bar includes 'Data Services', 'Data Management', and 'Feedback'. The 'Data Services' section features a 'DataFlow' icon highlighted with a red border. Other icons in this section include 'Data Engineering', 'Data Warehouse', 'Operational Database', and 'Machine Learning'. The 'Data Management' section includes icons for 'Data Hub Clusters', 'Data Catalog', 'Replication Manager', 'Observability', and 'Management Console'.

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.



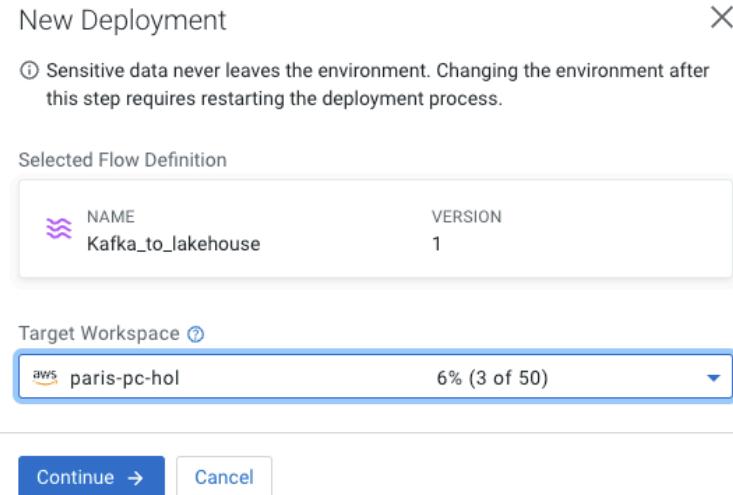
The screenshot shows the Cloudera DataFlow interface. On the left, a dark sidebar menu includes options like Dashboard, Catalog (which is selected and highlighted in purple), ReadyFlow Gallery, Flow Design, Functions, and Environments. The main area is titled 'Flow Catalog' and contains a search bar labeled 'Search by name'. A table lists a single flow entry: 'kafka\_to\_lakehouse'. The table columns are Name (sorted ascending), Type, Versions, and Last Updated. The flow details are: Type 'Custom Flow Definition', Versions '1', Last Updated '2 days ago'. Below the table are pagination controls for 'Items per page' (set to 10) and navigation arrows. A small note at the top right says 'REFRESHED: 5 seconds ago'.

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.



This screenshot shows the detailed view for the 'kafka\_to\_lakehouse' flow. The left sidebar remains the same. The main panel has a header with the flow name 'kafka\_to\_lakehouse' and a 'REFRESHED: 7 seconds ago' message. To the right of the flow name is a 'Actions' button. Below the header, there's a 'FLOW DESCRIPTION' section with 'No description specified' and a 'CRN #' field containing 'cm:cdp:df:us-west-1:508fd88f-8076-498a-acfb-6f8765cd35e8:flow:kafka\_to\_lakehouse'. There's also a checkbox for 'Only show deployed versions'. A table below shows one version: Version 1, Deployments 0, Associated Drafts 0. A 'Deploy →' button is located next to the table. Further down, a 'CREATED' section shows the date '2023-05-19 00:15 CEST' and the creator 'Alex Campos Simoes', with a note 'Initial Version'. At the bottom right of the main panel is a 'Download' 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 **paris-pc-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**.

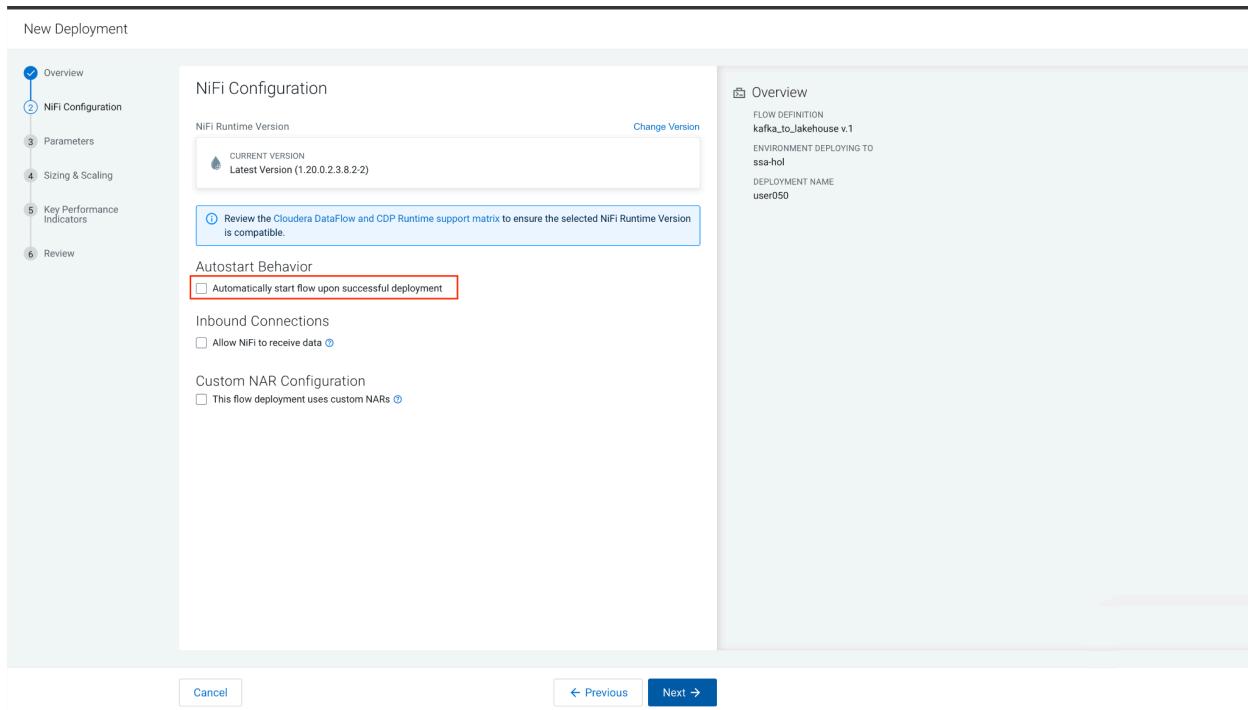
*For the purposes of this workshop, please name the Flow with your assigned username -user050, for example.*

You can also assign the flow to a project. For the sake of simplicity, you can keep it blank.

The screenshot shows the 'New Deployment' wizard at the 'Overview' step. The left sidebar lists steps 1 through 6: Overview (selected), NiFi Configuration, Parameters, Sizing & Scaling, Key Performance Indicators, and Review. The main area shows the 'Overview' section with a 'Deployment Name' field containing 'user050' (with a validation message 'Deployment name is valid'). Below it is the 'Selected Flow Definition' section, which is identical to the one shown in the previous screenshot. The 'Target Environment' section shows 'paris-pc-hol'. The 'Target Project' section shows 'Unassigned'. A warning message in an orange box states: 'No Projects are associated with this workspace. Selecting "Unassigned" will make this Deployment available to all DFFlowAdmin and DFFlowUser in paris-pc-hol.' At the bottom, there is an 'Import Configuration' section with a note about importing a previously exported configuration.

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. ("Paris2024")

**CDP Workload Username:** enter the assigned user number, *user050*, for example.

**Important:** choose the correct username, example : *user050*

**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**.

New Deployment

Parameters

Data entered here never leaves the environment in your cloud account. Provide parameter values directly in the text input or upload a file for parameters that expect a file.

The selected flow definition references an external Default NiFi SSL Context Service. Hence, DataFlow will automatically create a matching SSL Context Service with a keystore and truststore generated from the target environment's FreeIPA certificate.

SHOW:  Sensitive  No value

parameters (7)

CDP Workload User Password 9/100K  
Paris2024

CDP Workload Username 7/100K  
user050

CDPEnvironment

core-site.xml

ssl-client.xml

hive-site.xml

Select File

DataFlow automatically adds all required configuration files to interact with Data Lake services. Unnecessary files that are added won't impact the deployment process.

Database 7/100K  
user050

Kafka Brokers 202/100K  
paris-pc-hol-smm-corebroker2.paris-pc.djki-j7ns.cloudera.site:9093, paris-pc-hol-smm-corebroker1.paris-pc.djki-j7ns.cloudera.site:9093, paris-pc-hol-smm-corebroker0.paris-pc.djki-j7ns.cloudera.site:9093

Kafka Consumer Group Id 16/100K  
Consumer\_user050

Kafka Topic 10/100K  
telco\_data



Overview  
NiFi Configuration  
**Parameters**  
Sizing & Scaling  
Key Performance Indicators  
Review

**CLOUDERA**

Paris Hands on Lab

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

New Deployment

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

**NiFi Node Sizing**

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

**Number of NiFi Nodes**

Auto Scaling

Disabled

Nodes

**Storage Selection**

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

**Cancel** **← Previous** **Next →**

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

New Deployment

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

**Add New KPI**

**Overview**  
FLOW DEFINITION: kafka\_to\_jakeshouse v.1  
ENVIRONMENT: DEPLOYING TO: ssa-hol  
DEPLOYMENT NAME: user050

**NiFi Configuration**  
NIFI RUNTIME VERSION: Latest Version (1.20.0.2.3.8.2.2)  
AUTO START FLOW: No  
INBOUND CONNECTIONS: No  
CUSTOM NAR CONFIGURATION: No

**Parameters**  
parameters  
COP WORKLOAD USER PASSWORD: *[Sensitive Value Provided]*  
COP WORKLOAD USERNAME: user050  
COP ENVIRONMENT: core-site.xml  
core-site.xml  
hive-site.xml  
DATABASE: user050  
KAFKA BROKERS: realtime-ingestion-corebroker0.ssa-hol.yu1-vbzg.cloudera.site:9093,realtime-ingestion-corebroker1.ssa-hol.yu1-vbzg.cloudera.site:9093,realtime-ingestion-corebroker2.ssa-hol.yu1-vbzg.cloudera.site:9093

**Cancel** **← Previous** **Next →**

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

New Deployment

Review

[View CLI Command](#)

**Overview**

FLOW DEFINITION  
kafka\_to\_lakehouse v.1

ENVIRONMENT DEPLOYING TO  
ssa-hol

DEPLOYMENT NAME  
user050

**NiFi Configuration**

NIFI RUNTIME VERSION  
Latest Version (1.20.0.2.3.8.2-2)

AUTO-START FLOW  
No

INBOUND CONNECTIONS  
No

CUSTOM NAR CONFIGURATION  
No

**Parameters**

parameters

CDP WORKLOAD USER PASSWORD  
[Sensitive Value Provided]

CDP WORKLOAD USERNAME  
user050

CDPENVIRONMENT  
core-site.xml  
ssl-client.xml  
hive-site.xml

DATABASE  
user050

KAFKA BROKERS

[Cancel](#) [Previous](#) [Deploy](#)

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*.

CLOUDERA DataFlow

Dashboard

Filter By: STATUS All - 15 ENVIRONMENTS All - 1

Status	Name
Deploying	user050

**user050**  
ssa-hol

**Deployment Initiated**  
Initiated deployment of [user050].

KPIs System Metrics Alerts

Active Alerts [?](#)  
No alerts to display.

Event History [?](#)

SHOW ONLY:  Info  Warning  Error

Deployment Initiated 2023-05-21 00:09 CEST

[Load More](#)

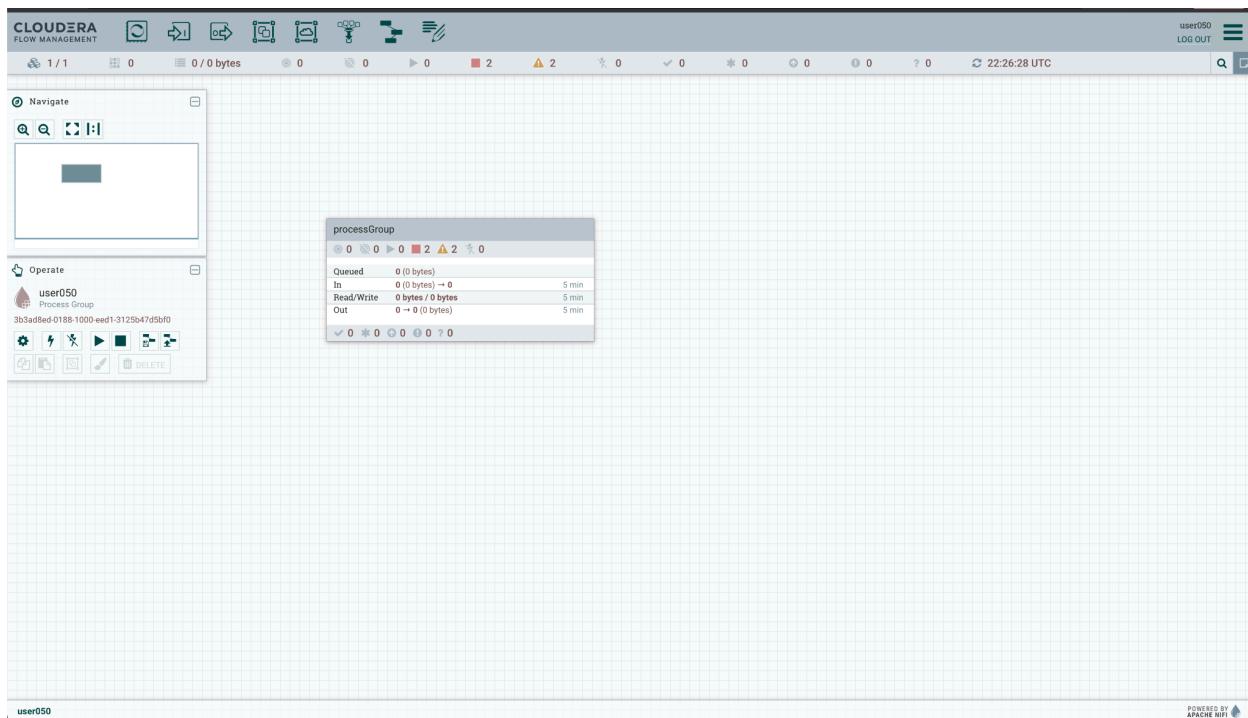
12. Once the deployment is finished, click on **Manage Deployment** to see the details of the recently deployed Flow.

The screenshot shows the Cloudera DataFlow interface. On the left sidebar, there are several navigation options: Dashboard, Catalog, ReadyFlow Gallery, Flow Design, Functions, and Environments. The main area is titled 'Dashboard' and shows a table with one row for 'user050'. The table columns are 'Status' (Deploying) and 'Name' (ssa-hol). To the right of the table, there is a detailed view for 'user050' under the 'aws ssa-hol' environment. The 'Alerts' tab is selected, showing a message 'No alerts to display.' Below this is the 'Event History' section, which lists various deployment events with their timestamps. A 'Load More' button is at the bottom of this list. In the top right corner, there is a status indicator 'REFRESHED: 7 seconds ago' and a 'Manage Deployment' button, which is highlighted with a red box.

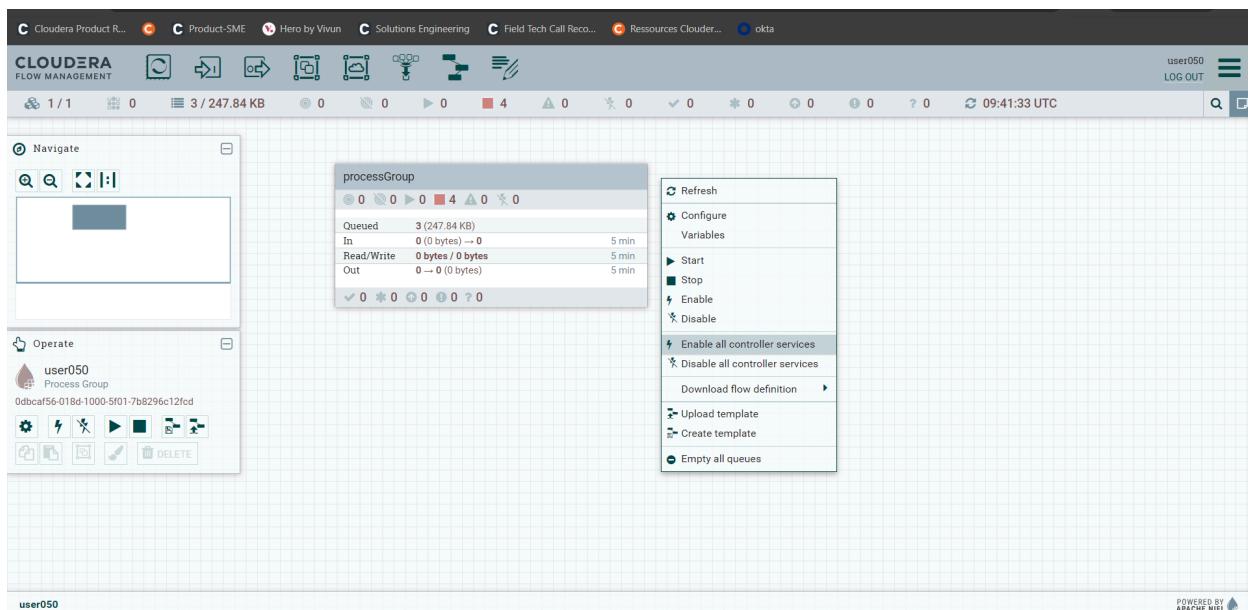
13. In this window you will see the Flow information displayed. It is time to execute the application processes from the graphical Flow Management interface. Click on **Actions -> View in NiFi**, to open Cloudera Flow Management canvas in a new window/tab.

The screenshot shows the 'Deployment Manager' page for 'user050'. The left sidebar has the same navigation options as the previous screen. The main content area includes sections for 'Deployment Settings' (with tabs for KPIs and Alerts, Sizing and Scaling, Parameters, and NIFI Configuration), 'Key Performance Indicators' (with a note to set up specific performance metrics and a 'Learn more' link), and a large central area for managing the deployment. At the bottom, there are buttons for 'Discard Changes', 'Apply Changes', and 'Update Deployment CLI Command'. On the right side, there is a 'Actions' dropdown menu with several options: 'View in NiFi', 'Start flow', 'Change NiFi Runtime Version', 'Restart Deployment', and 'Terminate'. The 'View in NiFi' option is highlighted with a red box.

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.



Right click on the canvas (not in the ProcessGroup) and click on the option “**Enable all Controller Services**” from the floating menu that appears.



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

**ConsumeKafkaRecord**, 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.

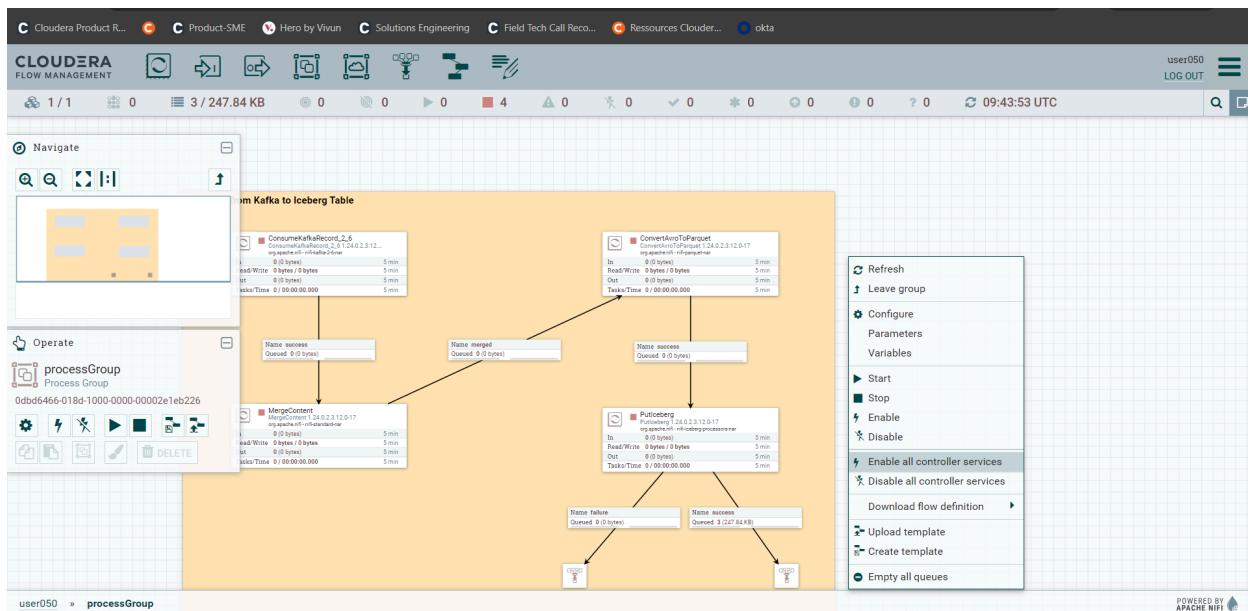
**ConvertAvroToParquet**, conversion needed to store the data in PARQUET format.

**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).

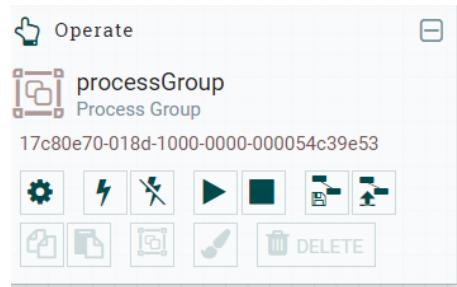
As you can see, the Processors are not started, and some have an error message/alert icon. The latter is because there are components of the data flow that must be activated before.

16. Double click on the process group.

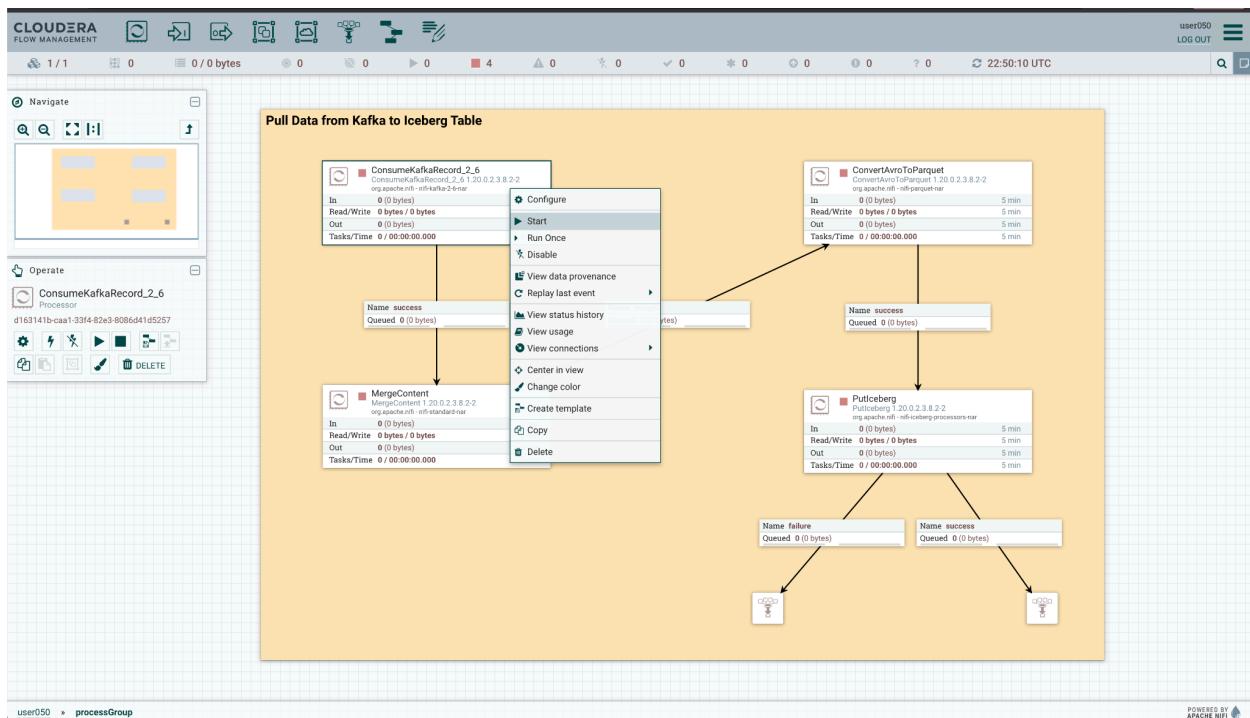
Right click outside the yellow rectangle and then select the option “**Enable all Controller Services**” from the floating menu that appears.



If necessary, in the operate window, click on “Enable”



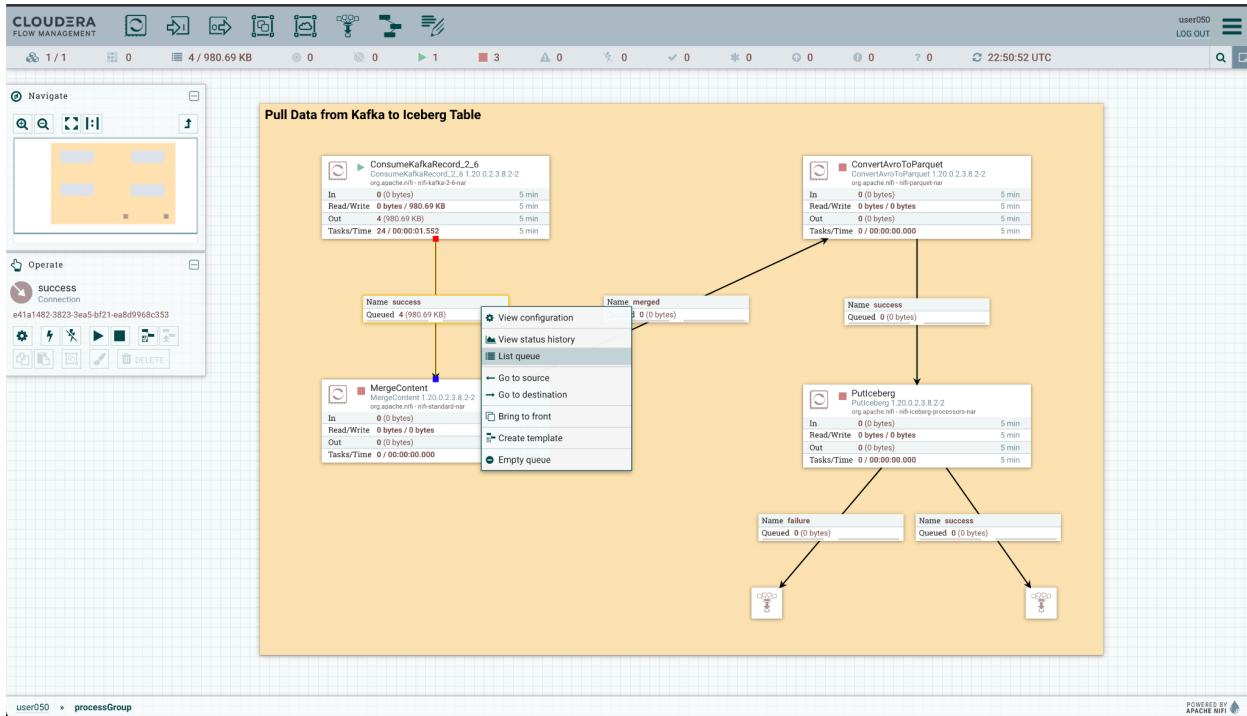
17. It's time to execute **Processors**. Start with **ConsumeKafkaRecord**, by right-clicking on it, and then clicking on **Start**. This will start consuming the Kafka topic data.



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

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

next to the legend **Queued** in the connection between **ConsumeKafkaRecord** and **MergeContent**. To see the queued data, right click on the connection and click on the option **List Queue**, opening a popup window.

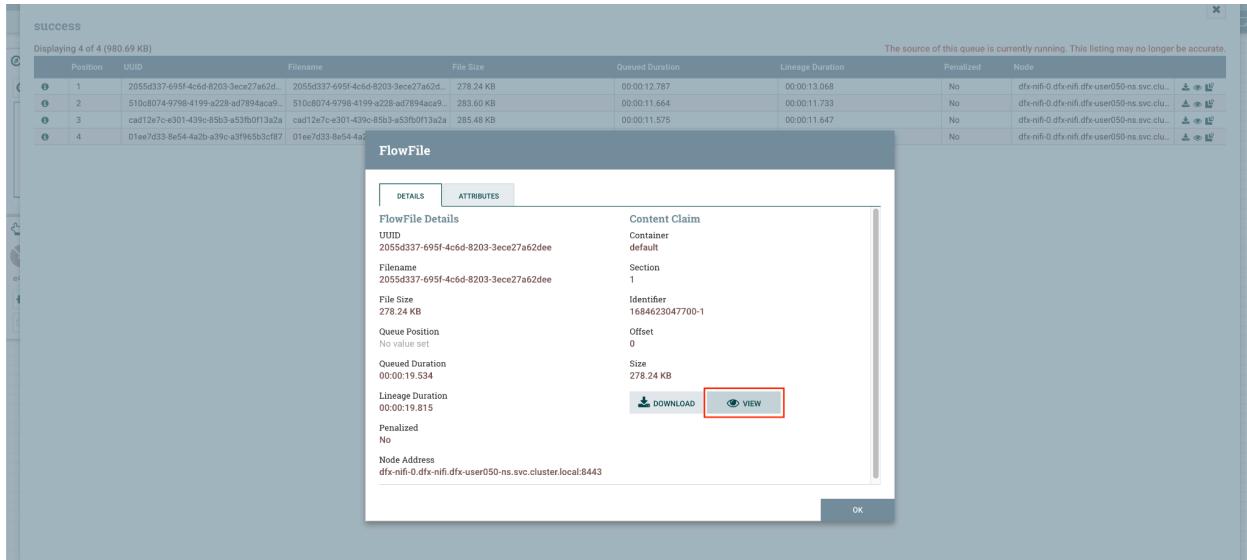


19. The next popup window lists the queued data. Click on the information icon (i) that appears on the left side to view the events.

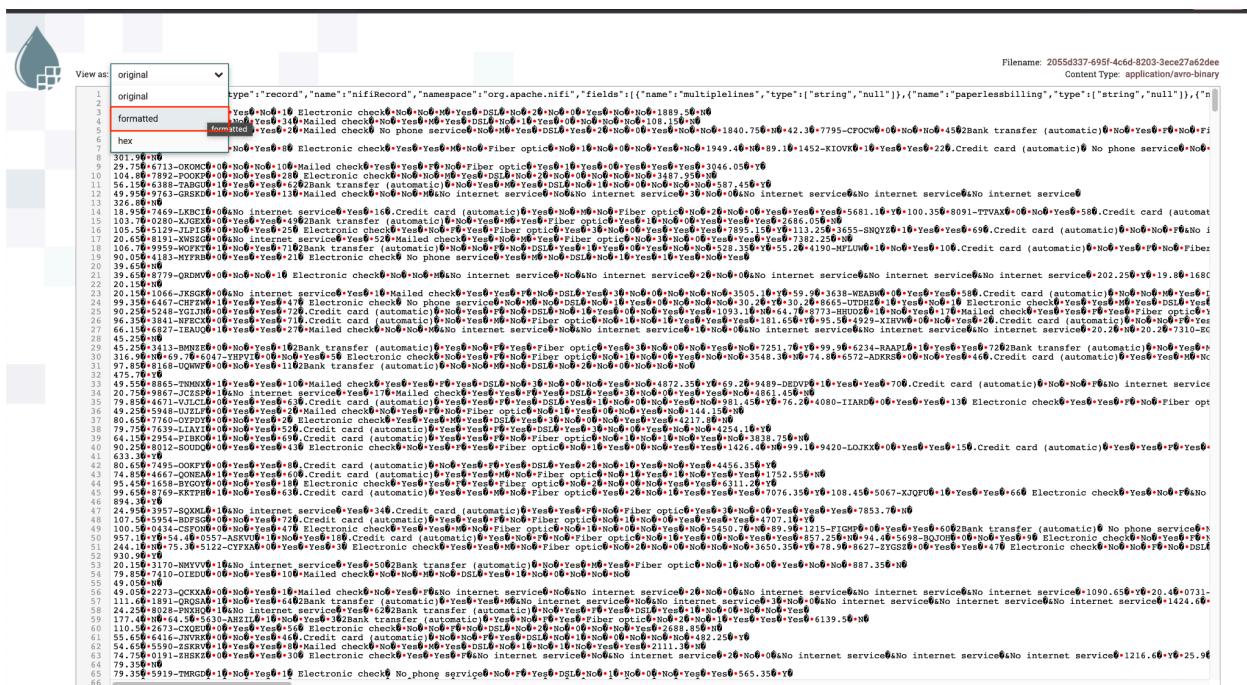
Displaying 4 of 4 (980.69 KB)							
Position	UUID	Filename	File Size	Queued Duration	Lineage Duration	Penalized	Node
1	2055d337-695f-4c6d-8203-3ece27a62d...	2055d337-695f-4c6d-8203-3ece27a62d...	278.24 KB	00:00:12.787	00:00:13.068	No	dfx-nifi-0-dfx-nifi-dfx-user050-ns.svc.c...
i	510c8074-9798-4199-a228-ad794acab...	510c8074-9798-4199-a228-ad794acab...	283.60 KB	00:00:11.664	00:00:11.733	No	dfx-nifi-0-dfx-nifi-dfx-user050-ns.svc.c...
3	cad12e7c-e301-439c-85b3-a53fb0f13a2a	cad12e7c-e301-439c-85b3-a53fb0f13a2a	285.48 KB	00:00:11.575	00:00:11.647	No	dfx-nifi-0-dfx-nifi-dfx-user050-ns.svc.c...
4	01ee7d33-8e54-4a2b-a39c-a3f965b3cf87	01ee7d33-8e54-4a2b-a39c-a3f965b3cf87	133.37 KB	00:00:11.527	00:00:11.567	No	dfx-nifi-0-dfx-nifi-dfx-user050-ns.svc.c...

The source of this queue is currently running. This listing may no longer be accurate.

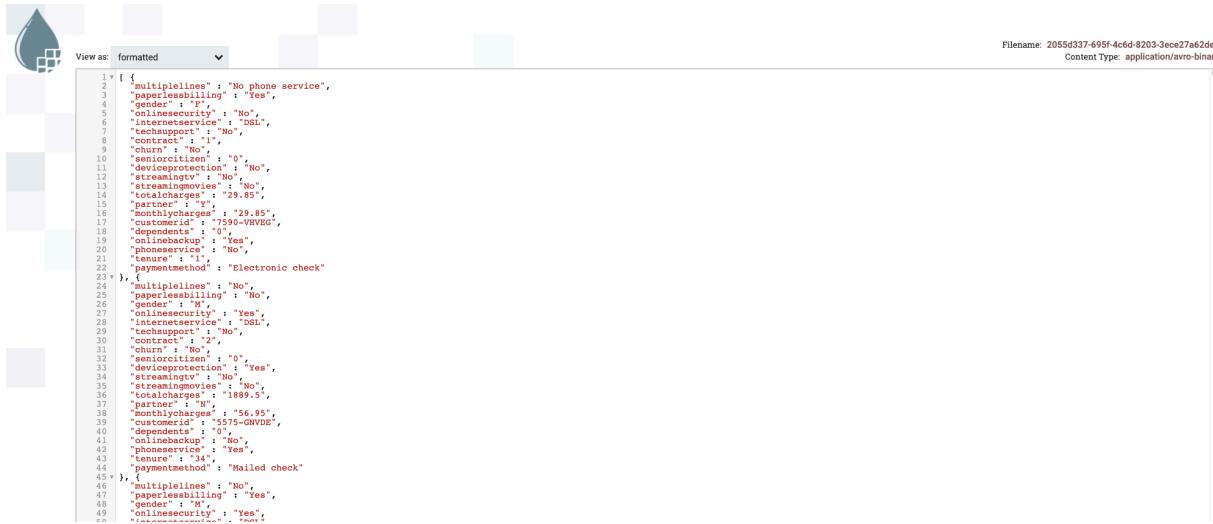
20. Once the FlowFile detail window appears, click on the button **VIEW** to open the content of consumed events.



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



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



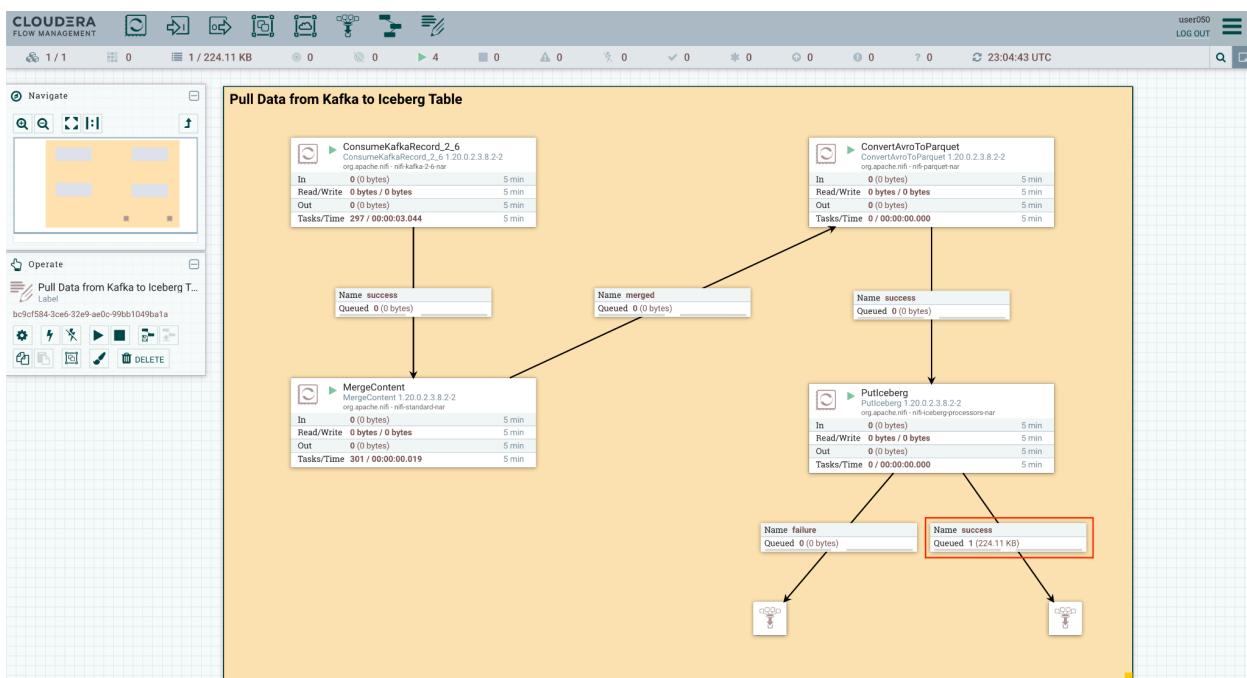
```

View as: formatted
Filename: 2055d37-695f-4cdd-8203-3ce27a62dee
Content Type: application/avro-binary

1: {
2:   "multiplelines": "No phone service",
3:   "paperlessbilling": "Yes",
4:   "gender": "F",
5:   "onlinesecurity": "No",
6:   "internetservice": "DSL",
7:   "techsupport": "No",
8:   "contract": "One year",
9:   "churn": "No",
10:  "seniorcitizen": "0",
11:  "partner": "No",
12:  "streamingtv": "No",
13:  "streamingmovies": "No",
14:  "monthlycharges": "29.85",
15:  "partner": "Y",
16:  "monthlycharges": "29.85",
17:  "customerserviceid": "7590-WVW05",
18:  "dependents": "0",
19:  "gender": "M",
20:  "onlinesecurity": "Yes",
21:  "internetservice": "DSL",
22:  "techsupport": "Electronic check"
23: },
24: {
25:   "multiplelines": "No",
26:   "paperlessbilling": "No",
27:   "gender": "M",
28:   "onlinesecurity": "Yes",
29:   "internetservice": "DSL",
30:   "techsupport": "No",
31:   "contract": "2 year",
32:   "churn": "No",
33:   "seniorcitizen": "0",
34:   "partner": "Yes",
35:   "streamingtv": "No",
36:   "streamingmovies": "1889.5",
37:   "partner": "N",
38:   "monthlycharges": "56.95",
39:   "customerserviceid": "7595-QNVE0",
40:   "dependents": "0",
41:   "onlinebackup": "No",
42:   "tenure": "34",
43:   "paymentmethod": "Mailed check"
44: },
45: {
46:   "multiplelines": "No",
47:   "paperlessbilling": "Yes",
48:   "gender": "M",
49:   "onlinesecurity": "Yes"
50: }

```

23. Continue running each of the Processors in order:**MergeContent**, after **ConvertAvroToParquet** and finally **PutIceberg**. Remember that you can refresh the flow counters with the combination Control+R or Command+R.  
If the previous steps were executed correctly, the connection of the Processor **PutIceberg** to a funnel should be of type **success**.



## o Lab 2 - Stream Messaging Manager - Optional

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

The screenshot shows the Cloudera Data Platform landing page. At the top left is the 'CLOUDERA Data Platform' logo. Below it are two sections: 'Data Services' and 'Data Management'. The 'Data Services' section contains icons for DataFlow, Data Engineering, Data Warehouse, Operational Database, and Machine Learning. The 'Data Management' section contains icons for Data Hub Clusters, Data Catalog, Replication Manager, Observability, and Management Console. At the bottom left is a user profile for 'Test50 User50'. At the bottom right is the text 'Powered by Cloudera'.

2. The list of all your cluster loads
  - o Click on [paris-pc-hol-smm](#)

The screenshot shows the Cloudera Management Console. The left sidebar has a 'Data Hub Clusters' section selected. The main area is titled 'Data Hubs' and shows a table with one entry:

Status	Name	Cloud Provider	Environment	Data Hub Type	Version	Node Count	Created
Running	paris-pc-hol-smm	aws	paris-pc-hol	7.2.17 - Streams Messaging Light Duty: Apache Kafka, Schema Registry, Streams Messaging Manager, Streams Replication Manager, Cruise Control	CDH 7.2.17	4	01/11/24, 03:47 PM GMT+1

At the bottom of the table are navigation links for page 1 of 1 and item count settings.

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

Data Hubs / paris-pc-hol-smm / Event History

**paris-pc-hol-smm**

cm:cdp:datadub:us-west-1:fe6d9b0f-d3f6-48a1-bd4d-ec67f77c08c2:cluster:64385b83-914a-4646-bdf6-b47f388d436c

**STATUS** Running

**NODES** 4 0 0

**CREATED AT** 01/11/24, 03:47 PM GMT+1

**CLUSTER TEMPLATE** 7.2.17 - Streams Messaging Light Duty: Apache Kafka, Schema Registry, Streams Messaging Manager, Streams Replication Manager, Cruise Control

**ENVIRONMENT DETAILS**

NAME: paris-pc-hol	DATA LAKE: paris-pc-hol-dl	CREDENTIAL: paris-pc-hol	REGION: us-east-1	AVAILABILITY ZONE: us-east-1b
--------------------	----------------------------	--------------------------	-------------------	-------------------------------

**SERVICES**

- CM-UI
- Schema Registry
- Streams Messaging Manager
- Token Integration

**CM CLOUDERA MANAGER INFO**

CM URL: https://paris-pc-hol-smm-gateway.paris-pc.djklj7ns.cloudera.site/paris-pc-hol-smm/cdp-proxy/cm7/home/	CM VERSION: 7.11.0	RUNTIME VERSION: 7.2.17-1.cdh7.2.17.p200.46967063	LOGS: Command logs, Service logs
---	--------------------	---	----------------------------------

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.

Overview

Cluster: kafka-d91e User:050

**TOPICS 1 BROKERS 3**

NAME	DATA IN	DATA OUT	MESSAGES IN	CONSUMER GROUPS	CURRENT LOG SIZE
telco_data	0B	0B	0	0	0B

**Consumer Groups (0)**

ACTIVE	PASSIVE	ALL
--------	---------	-----

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

The screenshot shows the Apache Kafka UI interface. On the left, there's a sidebar with icons for topics, brokers, and configurations. The main area is titled 'Topics / telco\_data'. It has tabs for METRICS, ASSIGNMENT, DATA EXPLORER, CONFIGS, and LATENCY. The METRICS tab is selected. At the top, it shows 'Producers (2)' with 'ACTIVE (0)', 'PASSIVE (2)', and 'ALL' buttons, and a 'MESSAGES' dropdown. Below that, it lists 'telco\_data' with metrics: DATA IN 0B, DATA OUT 0B, MESSAGES IN 0, CONSUMER GROUPS 0, CURRENT LOG SIZE 0B. To the right, there's a section for 'Consumer Groups (0)' with similar buttons. The bottom part of the screen shows detailed producer and consumer group information.

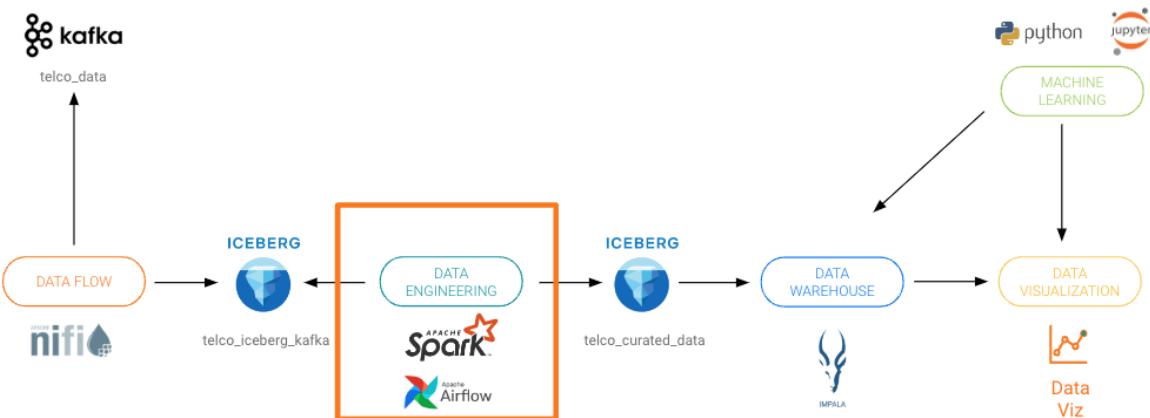
## 5. Your topics page loads, Click on Data Explorer

This screenshot shows the Apache Kafka UI for the 'telco\_data' topic. The left sidebar has icons for topics, brokers, and configurations. The main area is titled 'Topics / telco\_data'. It has tabs for METRICS, ASSIGNMENT, DATA EXPLORER, CONFIGS, and LATENCY. The METRICS tab is selected. At the top, it shows 'Producers (2)' with 'ACTIVE (0)', 'PASSIVE (2)', and 'ALL' buttons, and a 'MESSAGES' dropdown. Below that, it lists 'telco\_data' with metrics: DATA IN 0B, DATA OUT 0B, MESSAGES IN 0, CONSUMER GROUPS 0, CURRENT LOG SIZE 0B. To the right, there's a section for 'Consumer Groups (0)' with similar buttons. The bottom part of the screen shows detailed producer and consumer group information.

### 3. Data Engineering

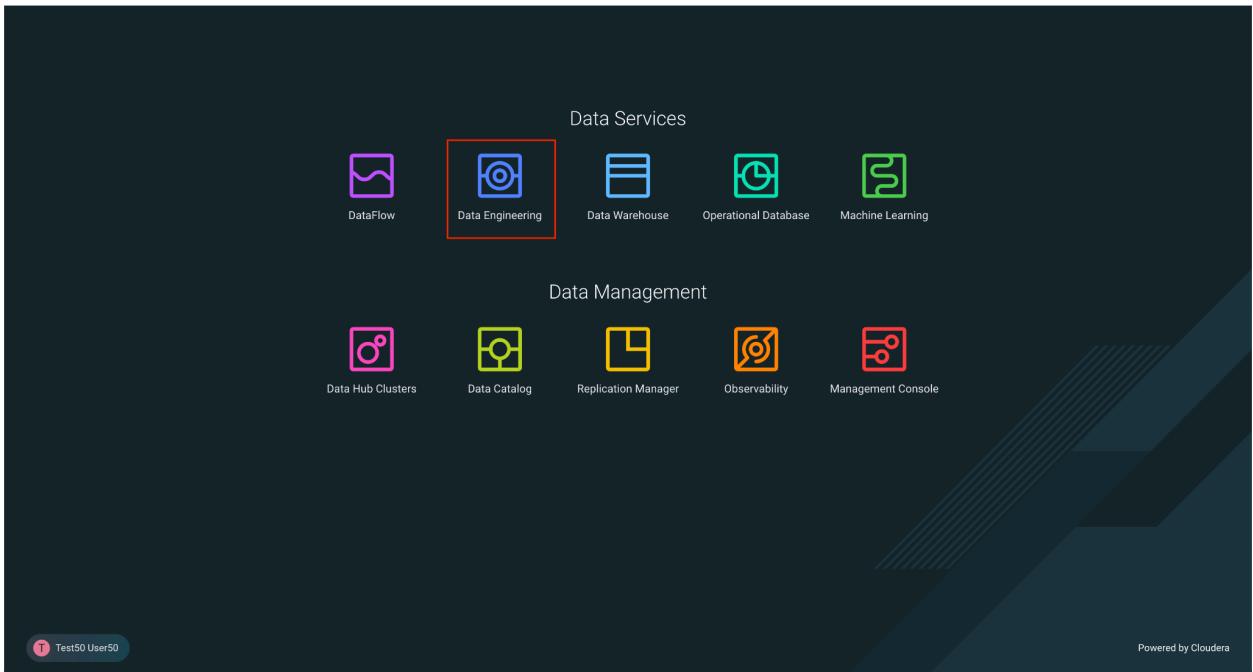
- Goals

- 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



- Lab 1 - Enrich the Ingested Iceberg table

1. Click on Data Engineering 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**.

The screenshot shows the Cloudera Data Engineering Home page. On the left, there is a sidebar with the following navigation options:

- Home
- Jobs
- Job Runs
- Sessions (Preview)
- Resources
- Administration
- Help
- Test50 User50

The main content area is titled "Welcome, Test50". It features several sections:

- Create**: Create jobs, orchestrate them or start a session.
  - Spark Jobs: Create New, Schedule, Ad-Hoc Run
  - Airflow Pipelines: Upload DAG file, Build a Pipeline (New)
- Resources**: Create resources for jobs.
  - File: Create New
  - Python: Create New
- Docs & Downloads**: CDE documentation and tools.
  - References: API Doc, Product Doc, Release Notes
  - Downloads: CLI Client, Migration Tool (New)
- Virtual Clusters**: Autoscaling Spark clusters to run Jobs.
  - aws ssa-de
    - ssa-de-cluster (View Jobs →)
    - Spark 3.2.3
  - CPU: 0, MEMORY: 0 MB, JOBS: 0

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**.

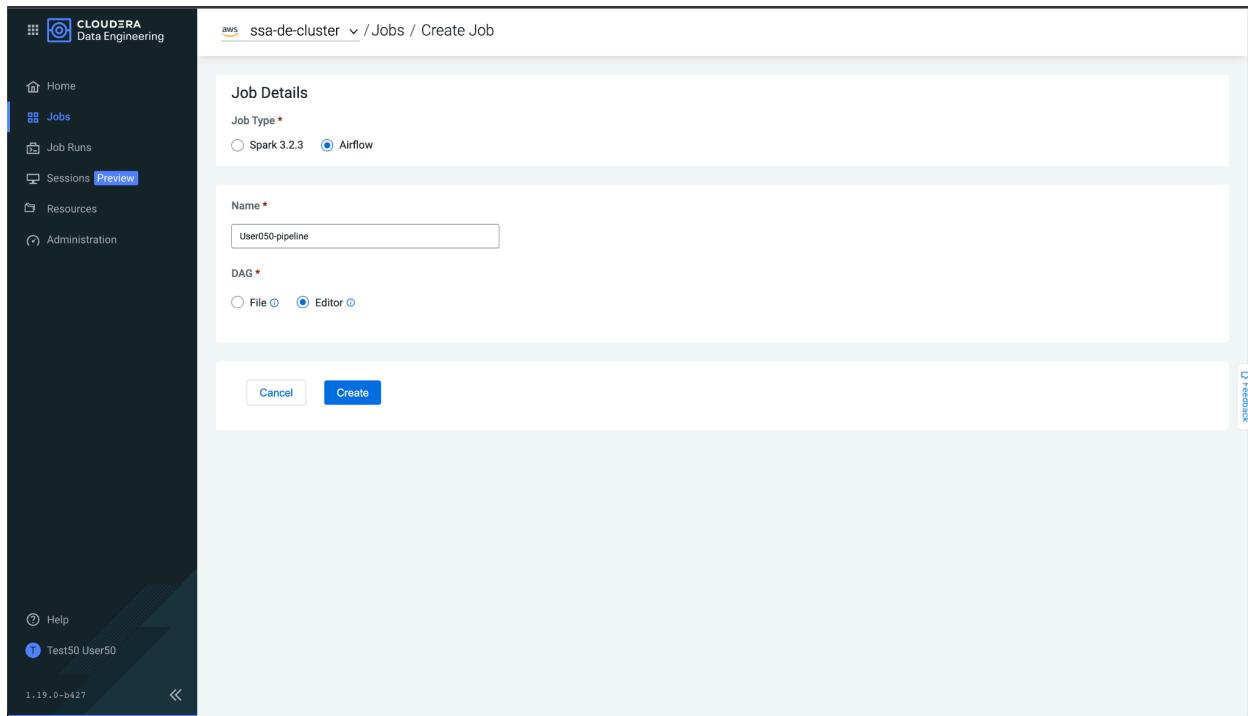
Status	Job	Type	Schedule	Modified On	Actions
<span>Running</span>	_CDE-Table-Update	Spark	Ad-Hoc	May 26, 2023, 12:22:35 PM	<span>⋮</span>
<span>Running</span>	_CDE-Data-Enrichment	Spark	Ad-Hoc	May 26, 2023, 12:22:21 PM	<span>⋮</span>

Items per page: 10 ▾ 1 - 2 of 2 < >

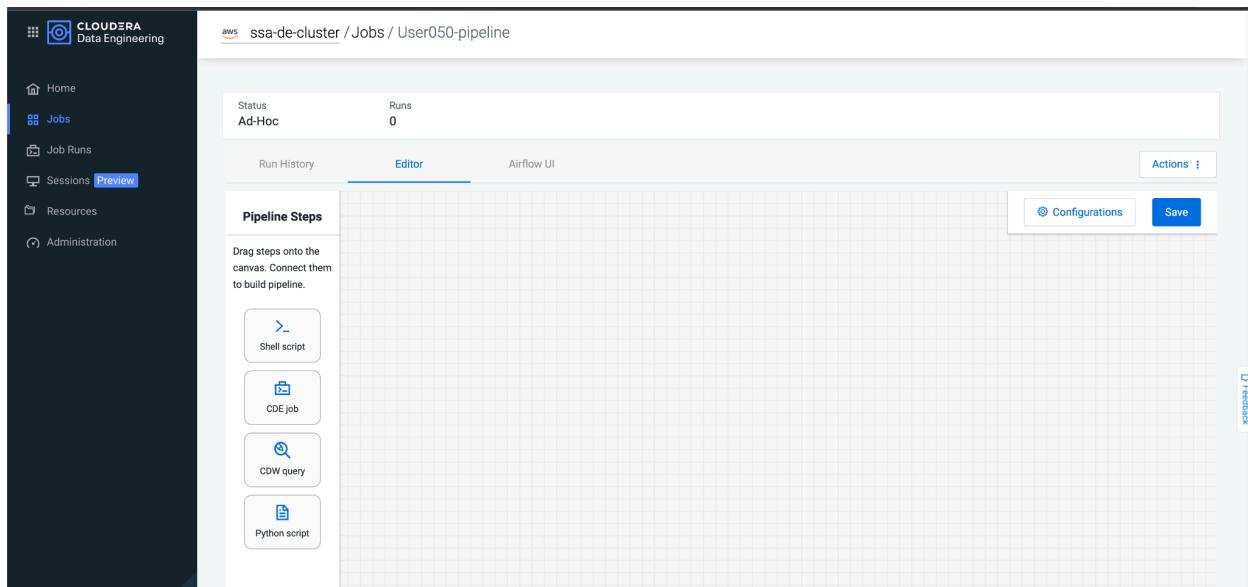
4. In the Job creation form, you must enter the following information:

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

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

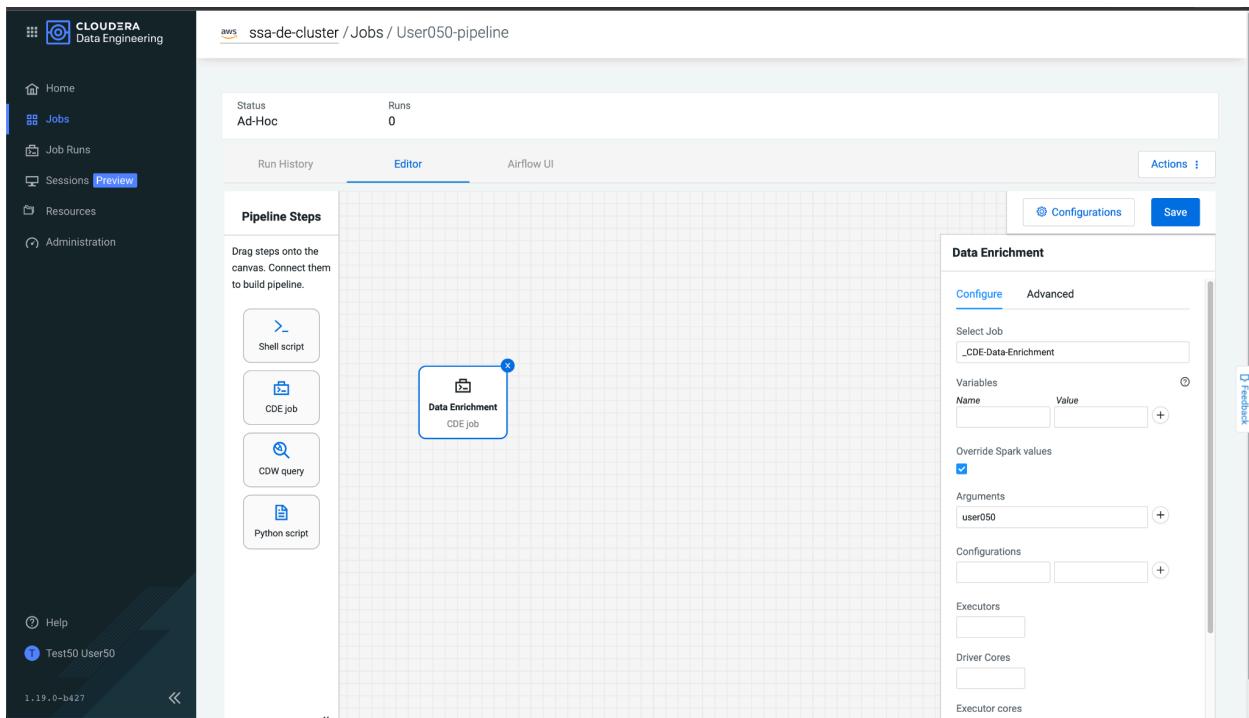


5. On the Job editing screen, select the Editor tab, and you will see the following canvas to drag the steps of the pipeline that we are going to create. In our case, we are going to create two CDE Jobs and relate them.



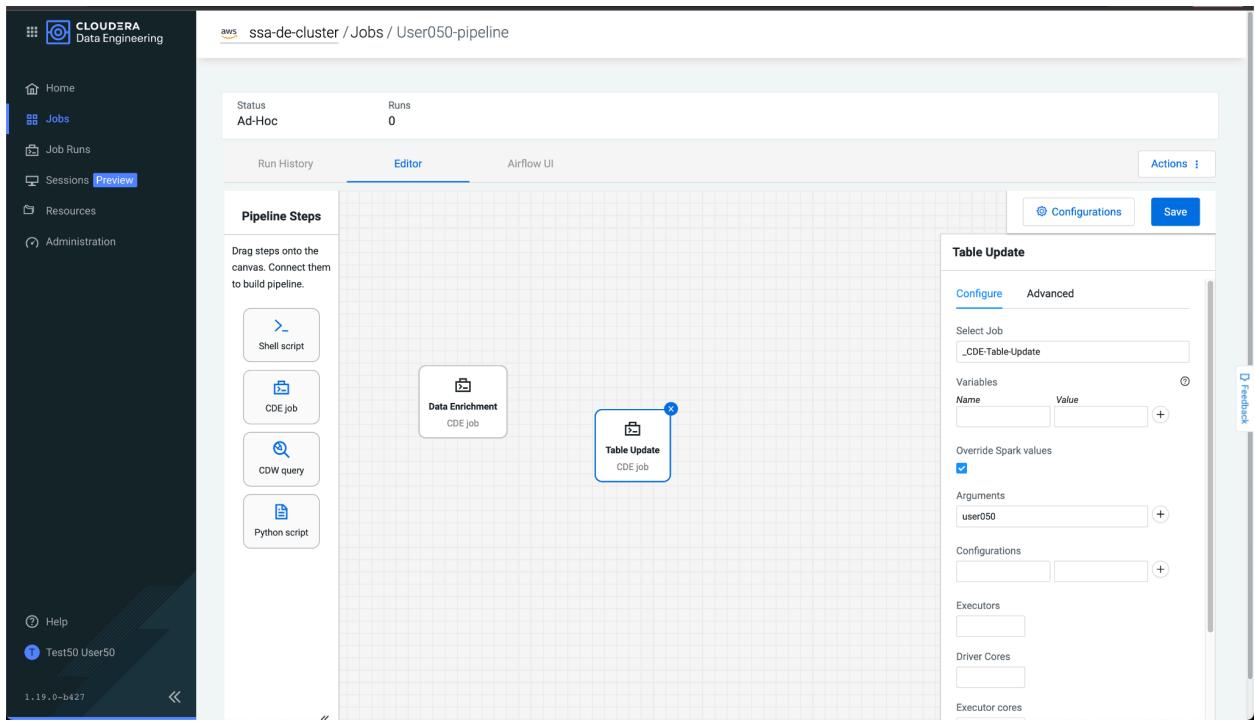
6. Let's start with the first Job. Click on the CDE Job button and drag onto the canvas, entering the following settings:

- **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
- Click on "Save" button

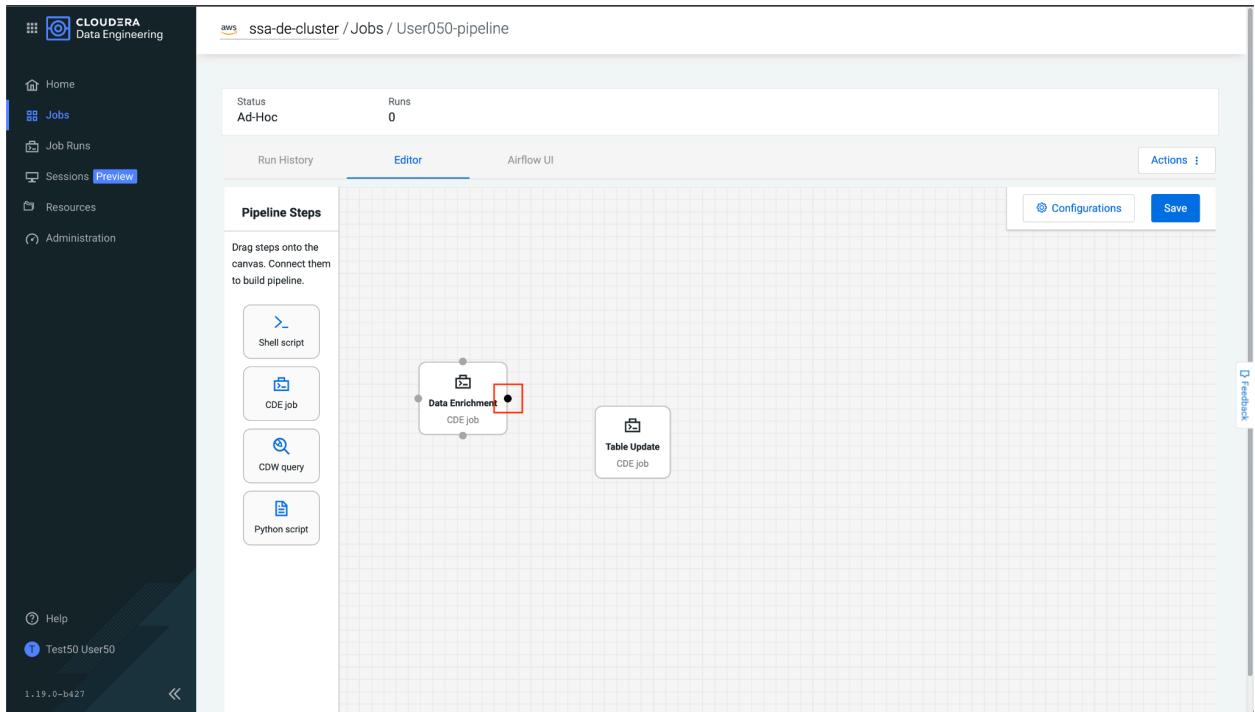


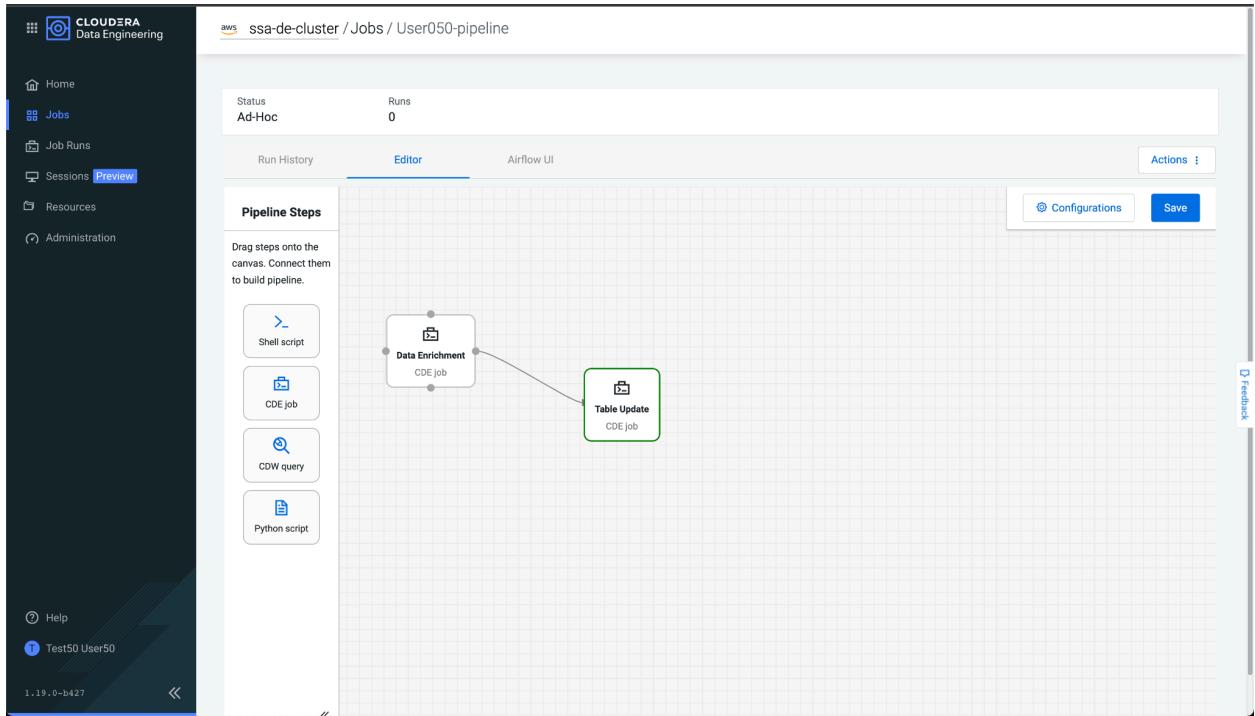
7. Configure the second Job. Click on the CDE Job button and drag onto the canvas, entering the following settings:

- **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

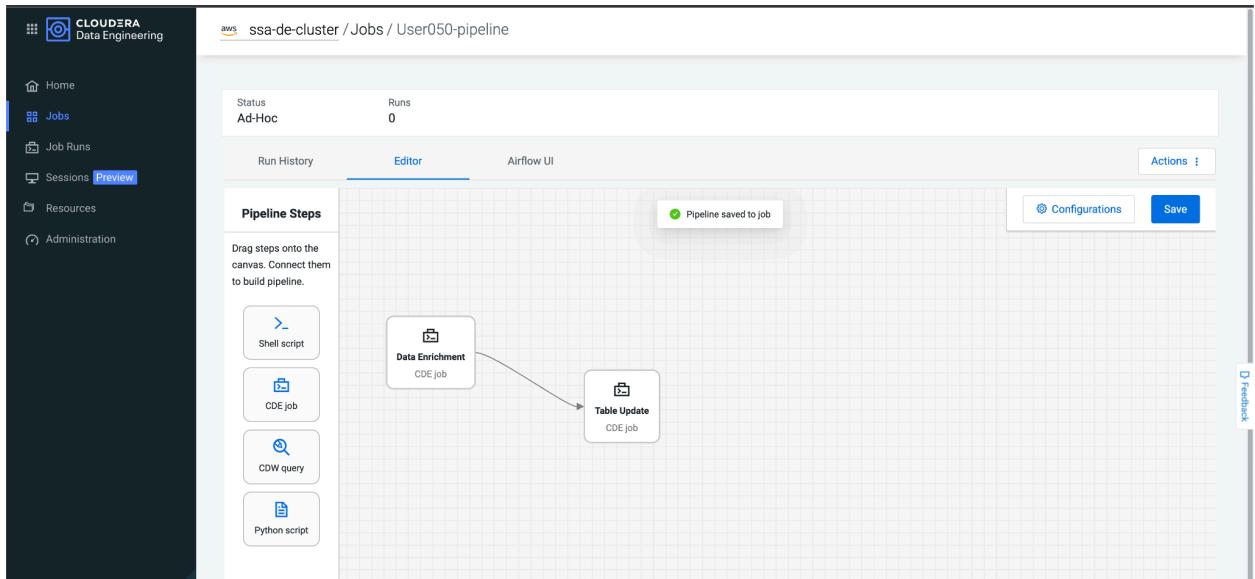


8. To set up the execution sequence, bind **Data Enrichment** with **Table Update**. For that, click on the right connector of the job of **Data Enrichment** and drag to the left connector of **Table Update**.



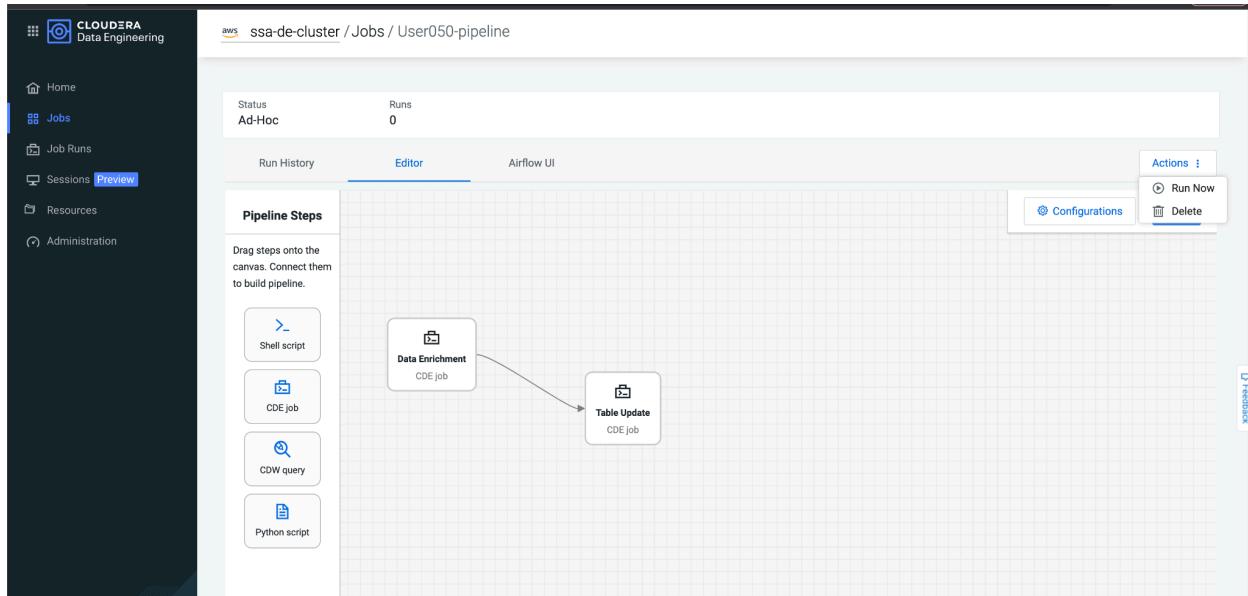


9. Once the Jobs have been joined, click on **Save** to save the settings made. You should see a message indicating **Pipeline saved to job**.

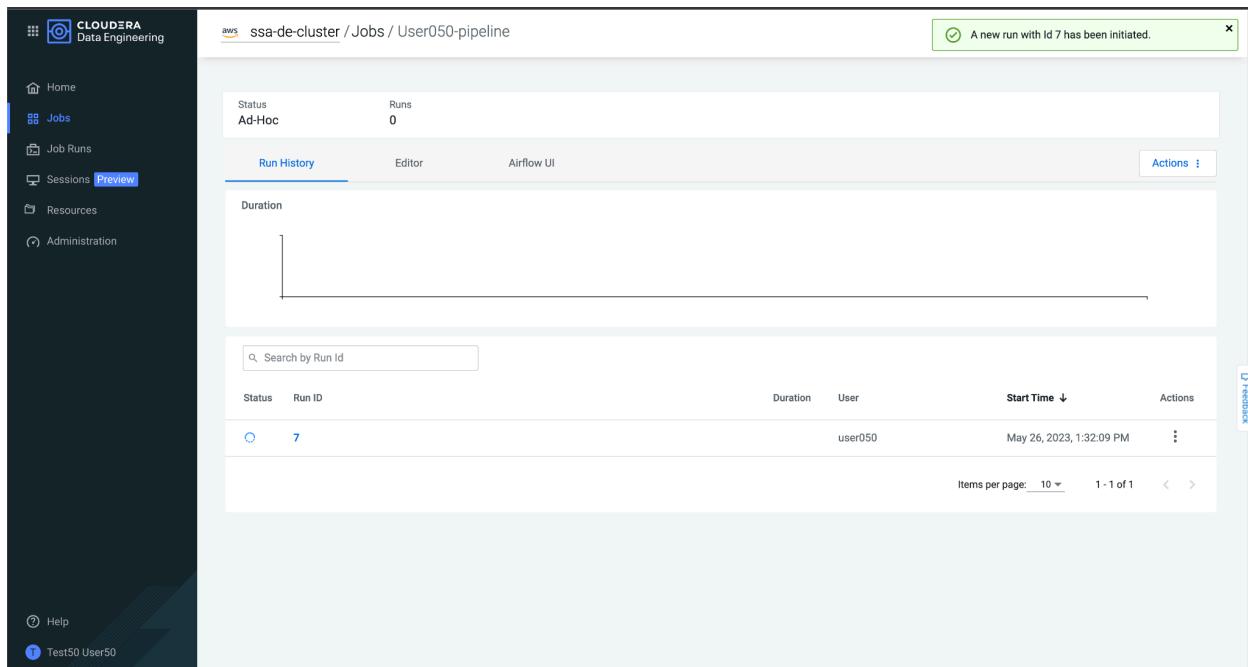


10. The time has come to run the pipeline. On the upper right side of the canvas, click:

- **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.

The screenshot shows the Cloudera Data Engineering interface. On the left, there's a sidebar with options like Home, Jobs (which is selected), Job Runs, Sessions (Preview), Resources, Administration, Help, and Test50 User50. The main area is titled "aws ssa-de-cluster / Jobs / User050-pipeline". It shows a summary card with Status Ad-Hoc and Runs 0. Below it, tabs for Run History, Editor, and Airflow UI are visible, with Airflow UI being the active tab. A sub-header says "DAG: User050\_pipeline". There are buttons for Grid, Graph, Calendar, Task Duration, Task Tries, Landing Times, Gantt, Details, and Code. An Audit Log link is also present. A date range selector shows 26/05/2023, 18:32:26 to 25, with dropdowns for All Run Types and All Run States. A "Clear Filters" button is nearby. A "Auto-refresh" toggle switch is turned on. The main content area displays a timeline for the DAG, showing tasks from 00:00:00 to 00:00:21. Two tasks are highlighted: "Data\_Enrichment" and "Table\_Update", which are both shown in green, indicating they are running. To the right, a "DAG Details" section provides summary statistics: Total Runs Displayed (1), Total running (1), First Run Start (2023-05-26, 18:32:10 UTC), Last Run Start (2023-05-26, 18:32:10 UTC), and Max Run Duration (00:00:21). A legend below the timeline lists run states: deferred, failed, queued, running, scheduled, skipped, success, up\_for\_reschedule, up\_for\_retry, upstream\_failed, and no\_status. A "Feedback" button is located on the far right.

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.

Status: success

Task\_id: Data\_Enrichment

Run: 2023-05-26, 18:32:24 UTC

Operator: CdeRunJobOperator

Duration: 1Min 11.675Sec

UTC:

Started: 2023-05-26, 18:33:29

Ended: 2023-05-26, 18:34:40

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.

Status: success

Task\_id: Table\_Update

Run: 2023-05-26, 18:36:36 UTC

Operator: CdeRunJobOperator

Duration: 1Min 1.530Sec

UTC:

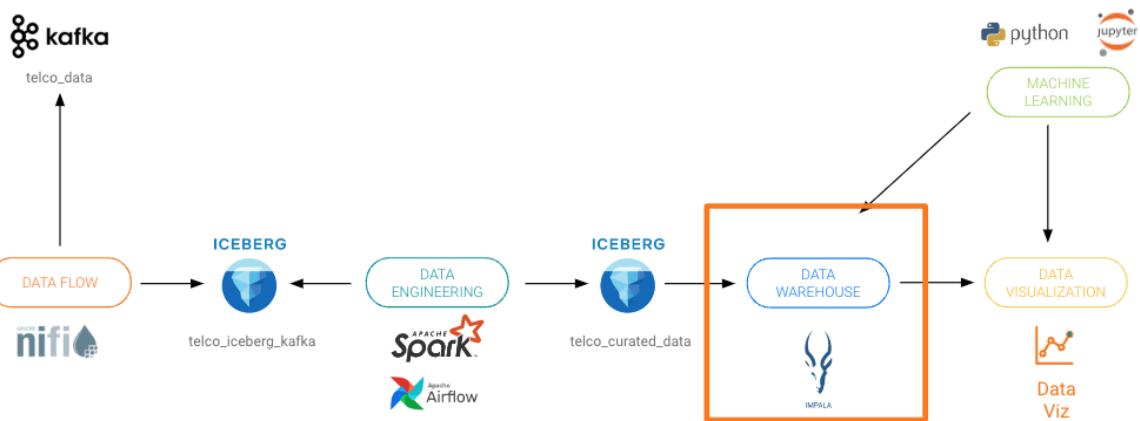
Started: 2023-05-26, 18:34:53

Ended: 2023-05-26, 18:35:55

## 4. Data Warehouse

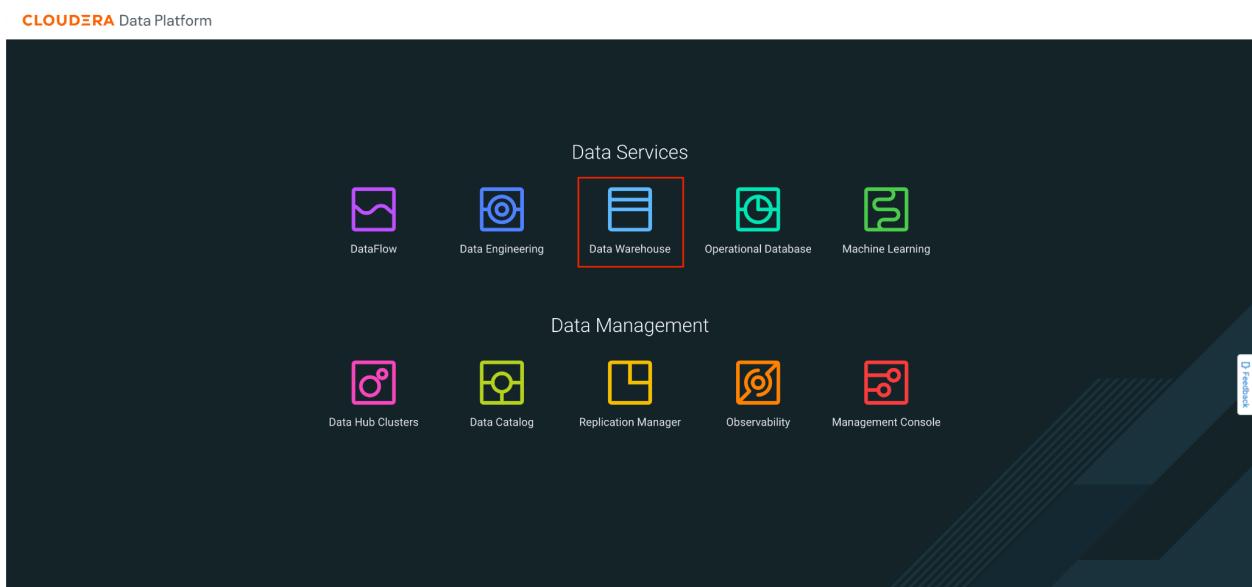
### ○ Goals

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



### ○ Dashboard Development

1. Click on Data Warehouse from CDP PC Home:



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

Welcome to Cloudera Data Warehouse Service

Cloudera Data Warehouse (CDW) is a cloud-native self-service analytic experience that enables BI analysts to go from zero to query in minutes.

Create  
Create new environments, database catalogs, virtual warehouses [See More ▾](#)

Query and Visualize Data  
Run SQL queries and create reports, or other visualizations you can share [See More ▾](#)

Resources and Downloads  
Documentation, release notes, JDBC/ODBC drivers, CLI client downloads, and more [See More ▾](#)

Environments (1) Database Catalogs (1) **Virtual Warehouses (2)**

Status	Name	Type	Version	CPU	Nodes	Apps	Uptime	Actions
Good Health	paris-pc-hol-vw-hive compute-1705325211-ggv5 paris-pc-hol-dl-default paris-pc-hol	Hive Compactor Unified Analytics	2023.0.16.3-2	33		HUE	a day	<a href="#">Suspend</a> <a href="#">More</a>
Good Health	paris-pc-hol-vw impala-1704985628-7rlh paris-pc-hol-dl-default paris-pc-hol	Impala	2023.0.16.3-2	178		HUE	5 days	<a href="#">Suspend</a> <a href="#">More</a>

3. Click on the button

- **Data Visualization** from which they were assigned.
- Click on “DataViz”

CLOUDERA Data Warehouse

Overview

Database Catalogs

Virtual Warehouses

**Data Visualization**

Data Visualization

NAME	DATA VISUALIZATION ID	Environment ID	VERSION	CPU	MEMORY	UPTIME	CREATED BY
dataviz-0	viz-1705409218...	env-bh6d4	7.1.0.2-3	4	16 GB	16 minutes	charles.aad

4. Once in Data Visualization,

- Close eventually the “What’s New” window
- Go to the “Data” option from the top menu, and then to the Connector **ImpalaConn** from the left menu.

Title/Table	ID	Created	Last Updated	Modified By	# Dashboards
Food Stores Inspection in NYC main.retail_food_store_inspections_current_critical_vio...	12	May 29, 2023	a few seconds ago	vizapps_admin	3
Cereals main.cereals	11	May 29, 2023	a few seconds ago	vizapps_admin	1
World Life Expectancy main.world_life_expectancy	9	May 29, 2023	a few seconds ago	vizapps_admin	1
Earthquake Data January 2019 main.earthquake_data2019	10	May 29, 2023	a few seconds ago	vizapps_admin	1
US State Populations Over Time main.census_pop	7	May 29, 2023	a few seconds ago	vizapps_admin	1
US County Population main.us_counties	8	May 29, 2023	a few seconds ago	vizapps_admin	1
Global Information Security Threats main.infoseq_1559	6	May 29, 2023	a few seconds ago	vizapps_admin	1
Restaurant Inspection SF main.restaurant_scores_lives_standard	5	May 29, 2023	a few seconds ago	vizapps_admin	1

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.

Title/Table	ID	Created	Last Updated	Modified By	# Dashboards
No data					

6. Enter the information for the new data source:

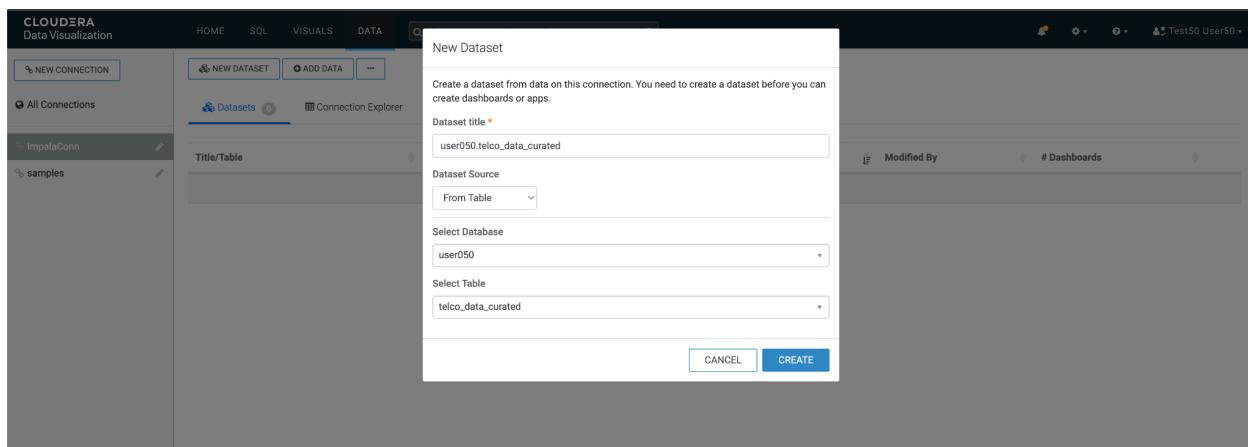
**Dataset title:** <assigned\_user>.telco\_curated\_data

**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.

The screenshot shows the 'Dataset Detail' page for a dataset named 'user050.telco\_data\_curated'. The left sidebar includes options like 'Dataset Detail', 'Related Dashboards', 'Fields', 'Data Model', 'Time Modeling', 'Segments', 'Filter Associations', and 'Permissions'. The main content area displays dataset details: Connection Type (Impala), Data Connection (ImpalaConn), Description (empty), Join Elimination (Enabled), Result Cache (From Connection), and Incremental Results (Disabled). It also shows the dataset ID (16), creation date (May 29, 2023 06:15 PM), creator (user050), last update date (May 29, 2023 06:15 PM), and last updater (user050).

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

The screenshot shows the 'Fields' page for the same dataset. The left sidebar is identical to the previous screen. The main content area is divided into 'Dimensions' and 'Measures'. The 'Dimensions' section lists 18 fields: multiplelines, paperslessbilling, gender, onlinesecurity, internetservice, techsupport, contract, churn, seniorcitizen, deviceprotection, streamingtv, streamingmovies, partner, customerid, dependents, onlinebackup, phoneservice, and paymentmethod. The 'Measures' section lists 3 fields: totalcharges, monthlycharges, and tenure.

## 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.

The screenshot shows the 'Data Model' page for the dataset. The left sidebar includes 'Dataset Detail', 'Related Dashboards', 'Fields', and 'Data Model'. The main content area shows a single dataset entry: 'telco\_data\_curated'. In the center, there is a large button labeled 'SHOW DATA' with a red border, which is the button mentioned in the instructions. Below it is a checkbox for 'Apply Display Format'.

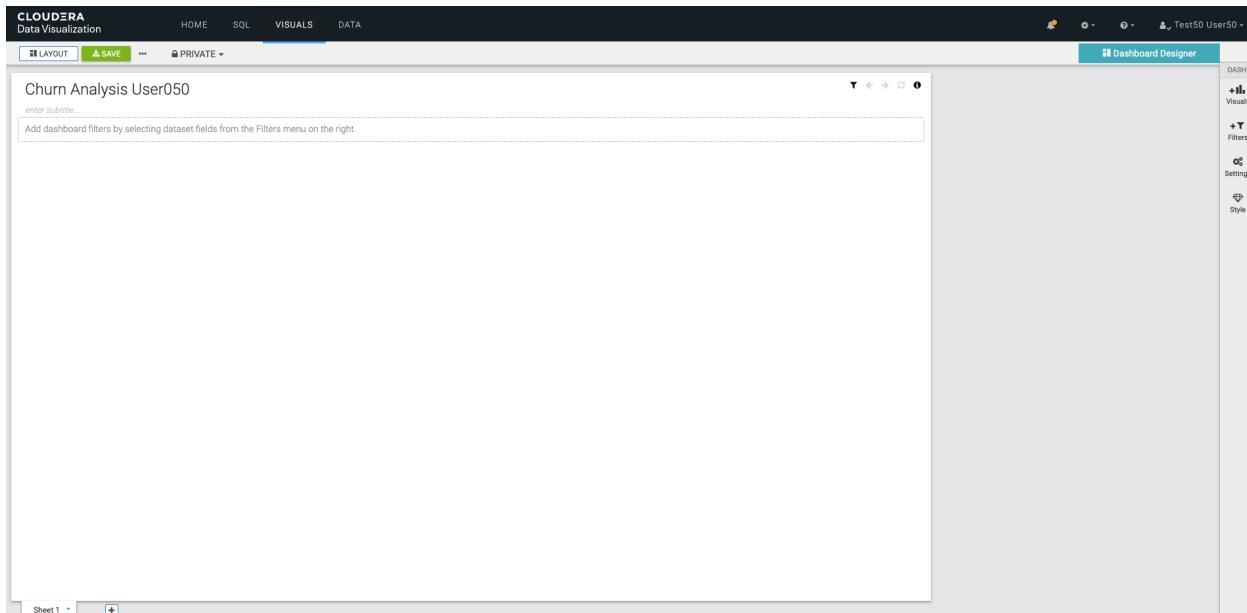
11. At this moment, a query to the Virtual Warehouse is executed to retrieve the data from the data set. Notice the columns and values. Click New Dashboard to create a new dashboard.

The screenshot shows the Cloudera Data Visualization interface. On the left, there's a sidebar with sections like Dataset Detail, Related Dashboards, Fields, Data Model (which is selected), Time Modeling, Segments, Filter Associations, and Permissions. The main area displays the 'telco\_data\_curated' dataset with various columns: multiplexes, paperlessbilling, gender, onlinesecurity, internetservice, techsupport, contract, churn, seniorcitizen, deviceprotection, streamingtv, streamingmovies, totalcharges, partner, monthlycharges, customerid, and d. A red box highlights the 'NEW DASHBOARD' button in the top right corner of the main content area.

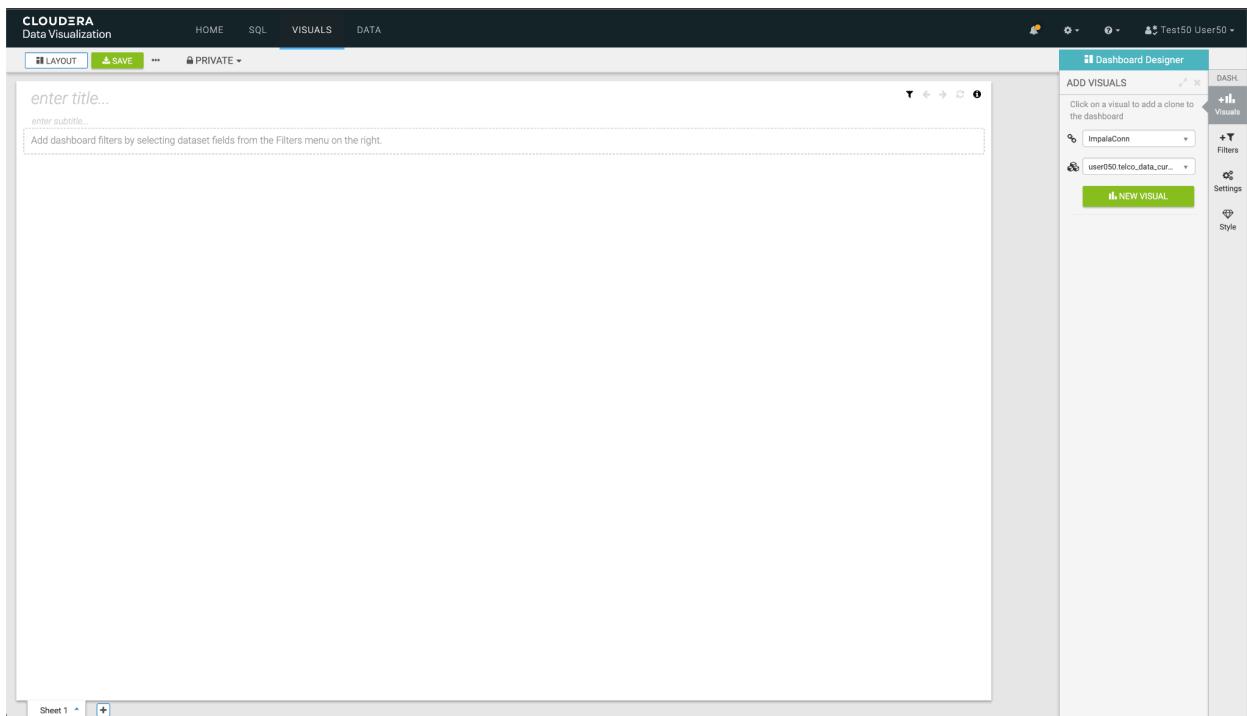
12. When opening the design canvas of a new panel, remove the element that is added by default, by clicking on the three dots (...) button at the top right of the element, and then clicking on the option **Delete Visual**

The screenshot shows the Cloudera Data Visualization interface with the 'LAYOUT' tab selected. On the left, there's a panel with 'enter title...' and 'enter subtitle...'. In the center, there's a table visual with columns: multiplexes, paperlessbilling, gender, and onlinesecurity. A context menu is open over the table, with the 'Delete Visual' option highlighted. To the right, there's a 'Dashboard Designer' sidebar containing sections for DATA, VISUALS, Dimensions, Measures, and Filters. A red box highlights the 'Delete Visual' option in the context menu.

At the top of the canvas, in the enter title field, enter the name *Churn Analysis\_<userid>* to identify the dashboard.

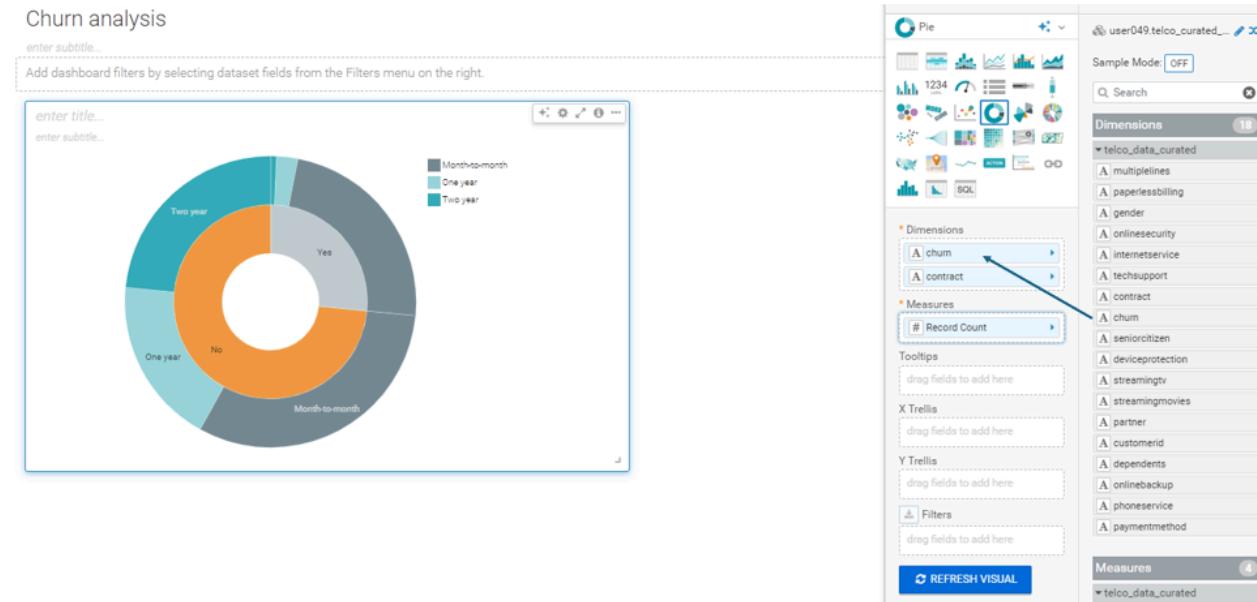


13. To add a new visual element, click on the button **Visuals** from the right menu, select the dataset that corresponds to them, and click on the button **New Visual**.

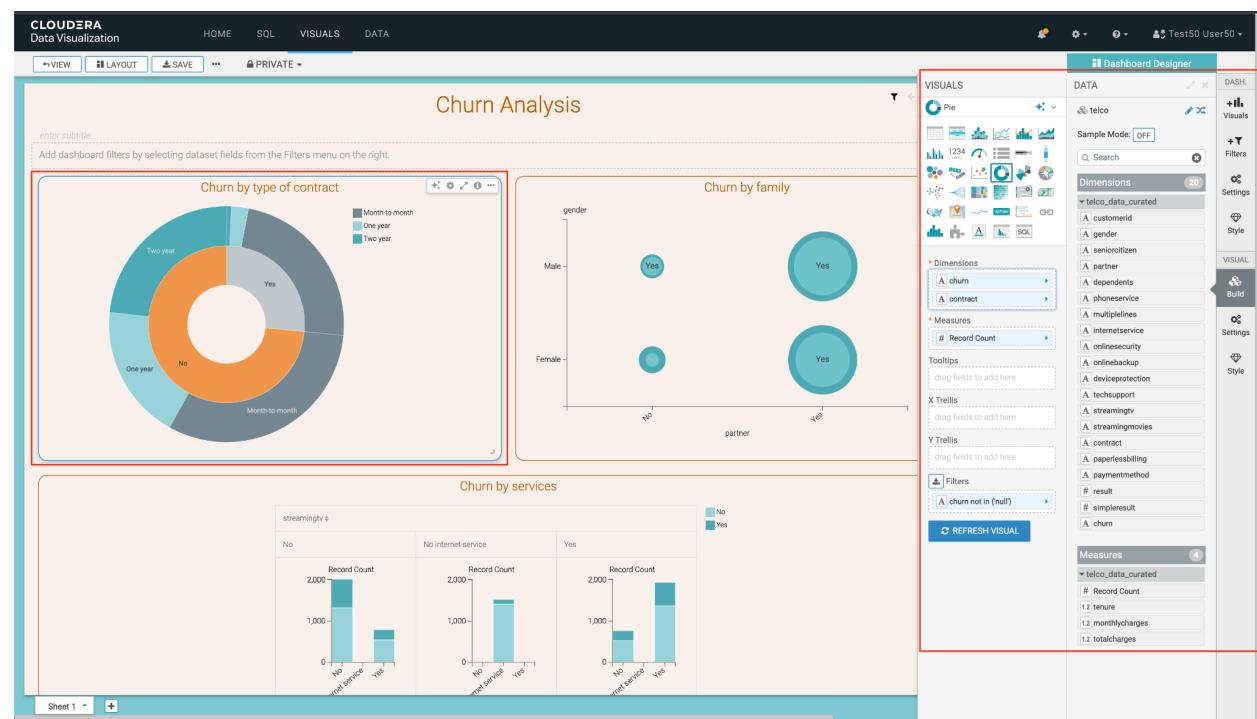


#### 14. Add the first visual element,

- Type: **pie chart**
- Dimensions: **churn** and **contract** (drag fields)
- Measures: **Record count**



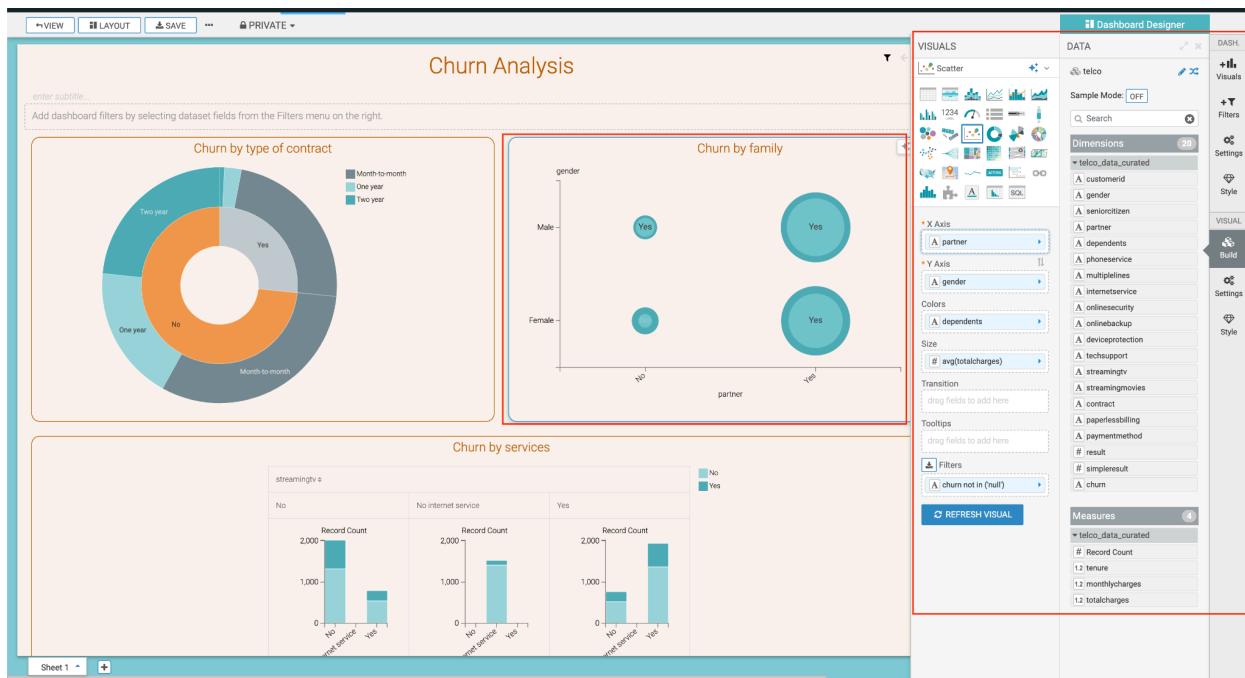
Once finished, click the button Refresh Visual.



## 15. Add the second visual element:

- Type: scatter chart
- X Axis: partner
- Y Axis: gender
- Colors: dependents
- Size: total charges
  - Click on the small arrow to the right
  - Click on the arrow next to aggregates
  - Select Avg
  - In the end you should have avg (total charges)

Once finished, click the button Refresh Visual.



## 16. Save Dashboard:

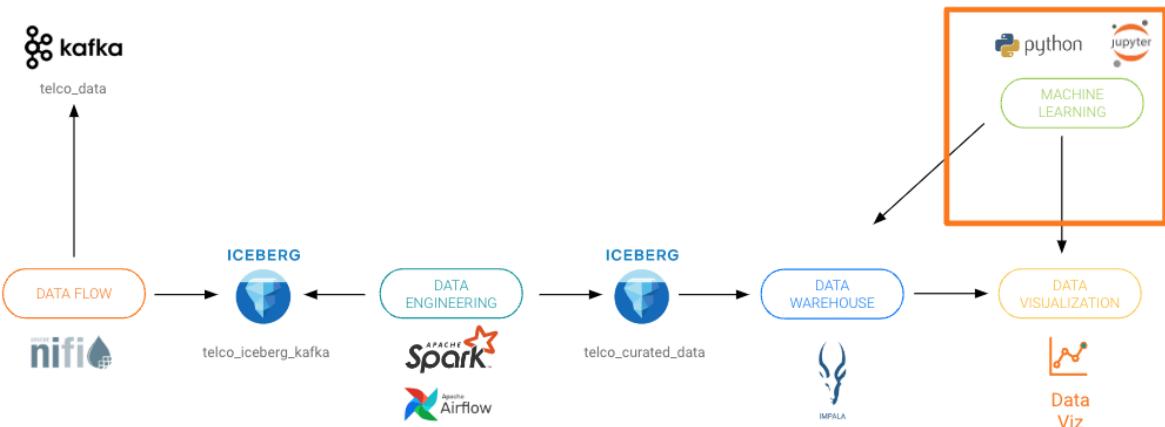
- Click on Save
- Click on View

Note : The third Visual you can see in the screen copy above is considered as an option. If enough time is left, depending on your priorities, you will be able to create it.

## 5. Machine Learning

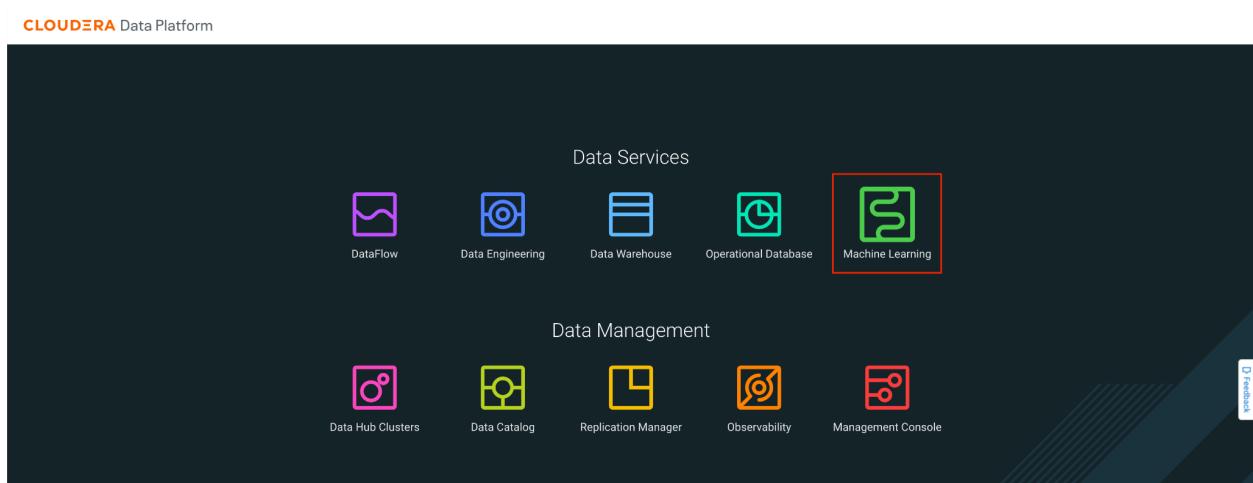
### ○ Goals

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



### ○ 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.

Status	Version	Workspace	Environment	Region	Creation Date	Cloud Provider	Actions
Ready	2.0.38	ssa-cml-workspace	ssa-hol	unknown	07/07/2023 11:42 PM CEST	aws AWS	

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 **Create a 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/camposalex/TelcoChurn>

New Project

Project Name  
Telco Churn

Project Description

Project Visibility  
 Private - Only added collaborators can view the project  
 Public - All authenticated users can view this project.

Initial Setup  
[Blank](#) [Template](#) [AMPS](#) [Local Files](#) [Git](#)

Provide the Git URL of the project to clone. Select the option that applies to your URL access.  
 HTTPS  SSH  
 https://github.com/campossalex/TelcoChurn

You are able to provide username/password.  
 e.g. https://username:password@mygithost.com/my/repository

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

Runtime setup

[Basic](#) [Advanced](#)

Basic configuration adds the most commonly used Editors for the Kernel of your choice. To fine-tune the Editors available in the project, choose the Advanced tab.

Kernel  
 Python 3.7 →

Add GPU enabled Runtime variant

These runtimes will be added to the project:

- JupyterLab - Python 3.7 - Standard - 2023.05
- PBJ Workbench - Python 3.7 - Standard - 2023.05
- Workbench - Python 3.7 - Standard - 2023.05

[Cancel](#) [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

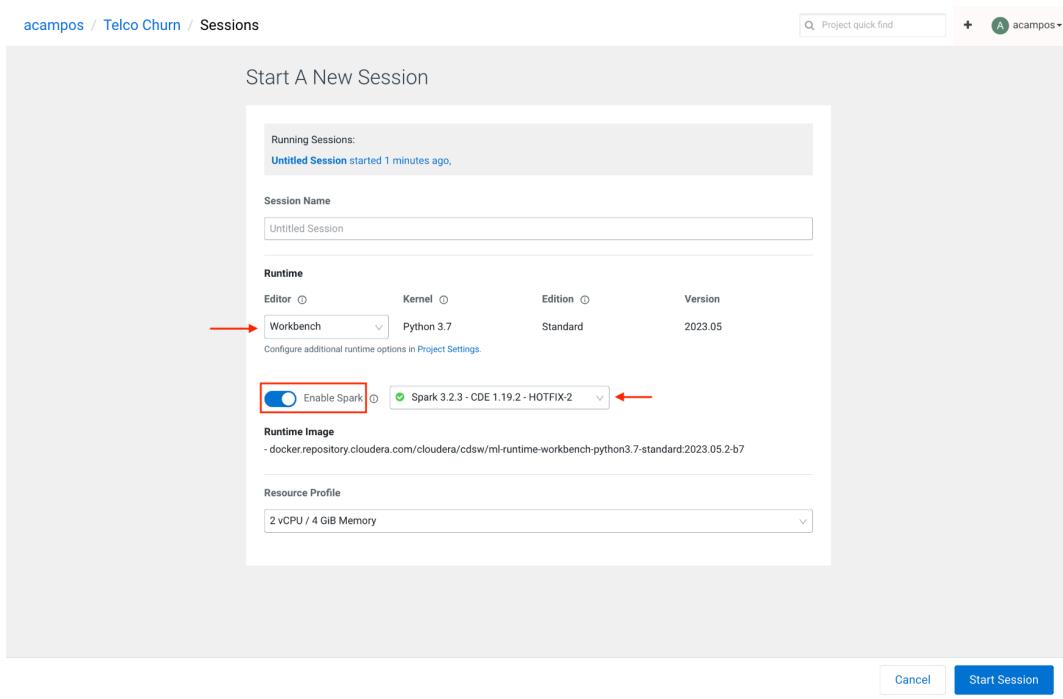
This Telco Churn project consists of running three scripts. The way of execution is through a session, which is the allocation of isolated compute resources for each user. For this, you must click on the blue button **New Session**, located in the upper right.

Name	Size	Last Modified
flask	-	a few seconds ago
images	-	a few seconds ago
models	-	a few seconds ago
0_bootstrap.py	1.95 kB	a few seconds ago
1_trainStrategy_job.py	18.63 kB	a few seconds ago
2_get_champion.py	508 B	a few seconds ago
_best_model_serve.py	2.74 kB	a few seconds ago
_model_viz.py	4.21 kB	a few seconds ago
cdsw-build.sh	44 B	a few seconds ago
chumexplainer.py	6.69 kB	a few seconds ago
lineage.yml	610 B	a few seconds ago
README.md	11.97 kB	a few seconds ago
requirements.txt	197 B	a few seconds ago
visuals.json	281.07 kB	a few seconds ago

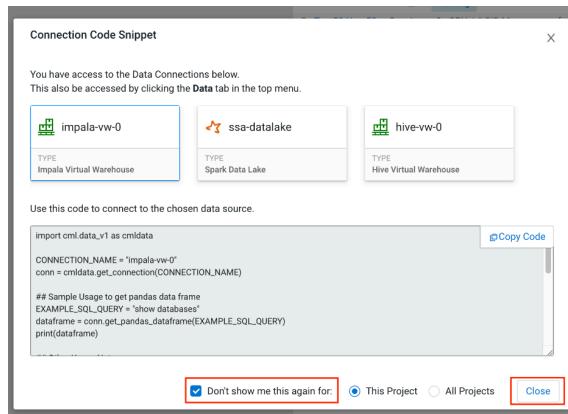
## 6. When starting a new session, make sure:

- 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

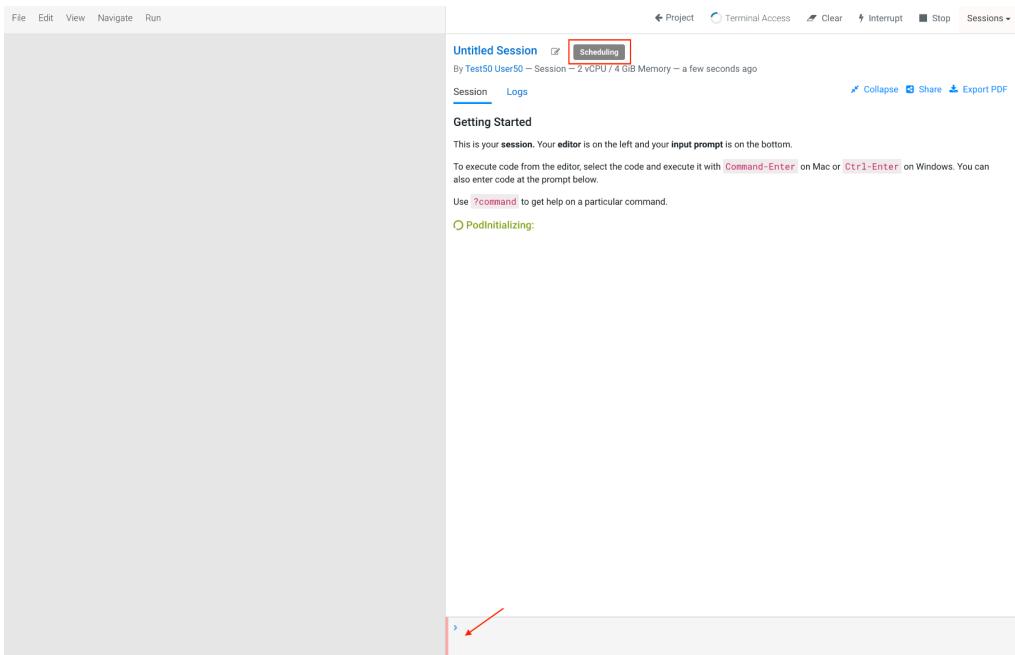
Click on the button **Start Session**



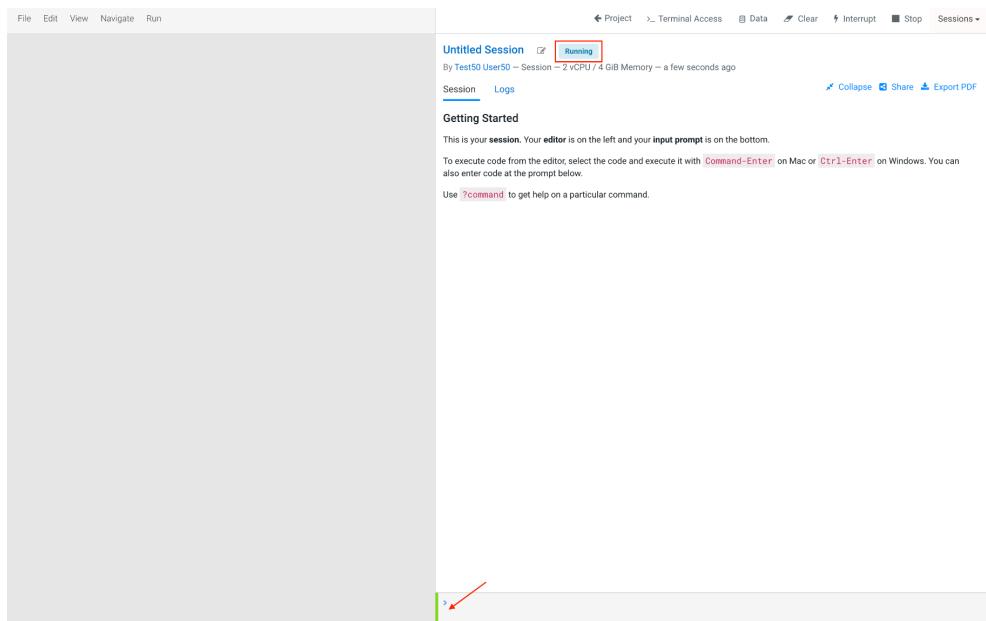
7. When you start a session for the first time, it will ask if you want to use a data connection. This project does not need this type of connection. mark the check of **Don't show me this again**, and then click the button **Close**, so this window will not appear anymore.



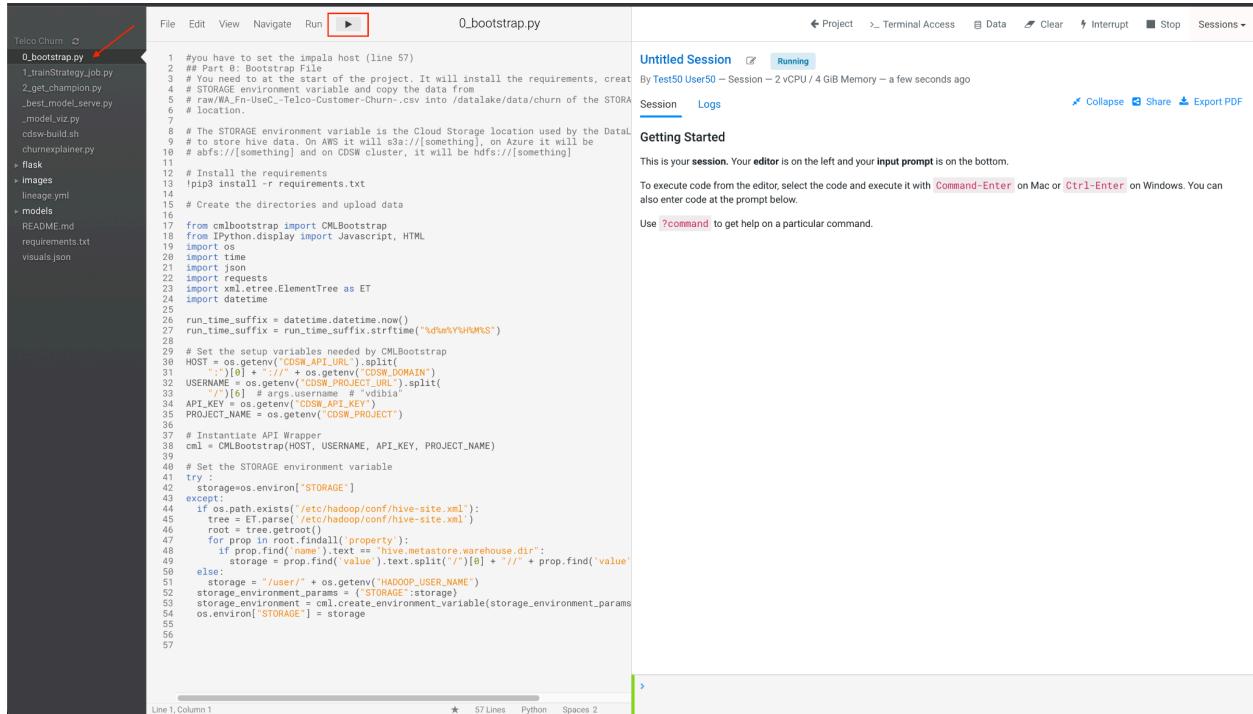
8. The editor/notebook located on the right side of the window will be in **Scheduling** status, and the bottom command bar flashing red. This means that CML is allocating computation for your session.



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.



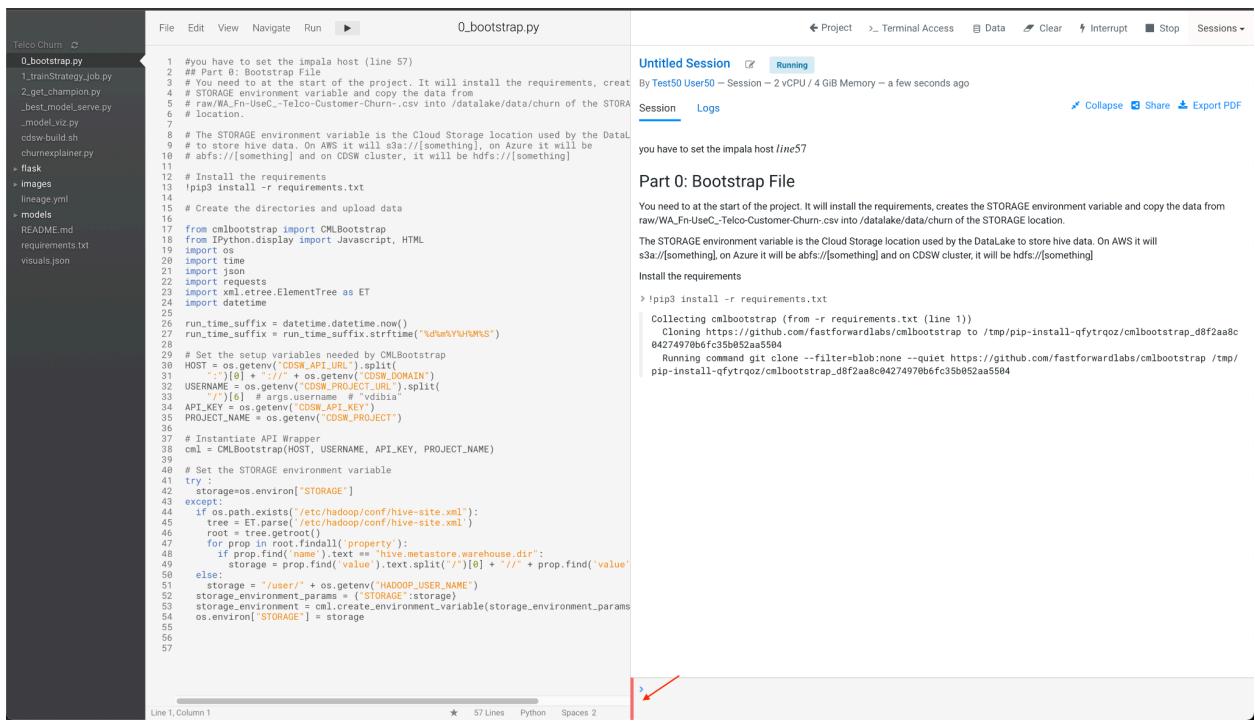
The screenshot shows the Cloudera Data Platform interface. On the left, there is a sidebar with a tree view of files and folders for a project named "Telco Churn". A red arrow points to the file "0\_bootstrap.py" in the sidebar. The main area is a code editor titled "0\_bootstrap.py" containing Python code. At the top of the editor, there is a toolbar with a "Run" button, which has a red box around it. The code itself is a script that sets up storage environment variables and creates a CMLBootstrap object. It includes imports for os, json, requests, ET, and xml.etree.ElementTree. It uses os.getenv to get values for CDSW\_API\_URL, CDSW\_DOMAIN, CDSW\_PROJECT\_URL, API\_KEY, and PROJECT\_NAME. It then sets the STORAGE environment variable and creates a CMLBootstrap object with these parameters. The code ends with a try-except block that attempts to parse the /etc/hadoop/conf/hive-site.xml file to find the hive.metastore.warehouse.dir value. The bottom of the interface shows a command bar with buttons for Project, Terminal Access, Data, Clear, Interrupt, Stop, and Sessions. The "Running" status is indicated next to the Session button. The "Logs" tab is selected in the session bar. The status bar at the bottom shows "Line 1, Column 1", "57 Lines", "Python", and "Spaces 2".

```

1 #you have to set the impala host (line 57)
2 ## Part 0: Bootstrap file
3 # You need to at the start of the project. It will install the requirements, create
4 # STORAGE environment variable and copy the data from
5 # raw//MA_Fn-UseC_-Telco-Customer-Churn-.csv into //datalake/data/churn of the STORA
6 # location.
7
8 # The STORAGE environment variable is the Cloud Storage location used by the Data
9 # store. On AWS it will s3://[something], on Azure it will be
10 # abfs://[something] and on CDSW cluster, it will be hdfs://[something]
11
12 # Install the requirements
13 !pip3 install -r requirements.txt
14
15 # Create the directories and upload data
16
17 from cmlbootstrap import CMLBootstrap
18 from IPython.display import Javascript, HTML
19 import os
20 import time
21 import json
22 import requests
23 import xml.etree.ElementTree as ET
24 import datetime
25
26 run_time_suffix = datetime.datetime.now()
27 run_time_suffix = run_time_suffix.strftime("%d%b%Y%H%M%S")
28
29 # Set the auth variables needed by CMLBootstrap
30 HOST = os.getenv("CDSW_API_URL").split(
31     ":" )[0] + ":" + os.getenv("CDSW_DOMAIN")
32 USERNAME = os.getenv("CDSW_PROJECT_URL").split(
33     "/" )[1] + os.getenv("CDSW_PROJECT_NAME") + "bibis"
34 API_KEY = os.getenv("CDSW_API_KEY")
35 PROJECT_NAME = os.getenv("CDSW_PROJECT")
36
37 # Instantiate API Wrapper
38 cml = CMLBootstrap(HOST, USERNAME, API_KEY, PROJECT_NAME)
39
40 # Set the STORAGE environment variable
41 try:
42     storage=os.environ['STORAGE']
43 except:
44     if os.path.exists('/etc/hadoop/conf/hive-site.xml'):
45         tree=ET.parse('/etc/hadoop/conf/hive-site.xml')
46         root = tree.getroot()
47         for prop in root.findall('property'):
48             if prop.find('name').text == "hive.metastore.warehouse.dir":
49                 storage = prop.find('value').text.split('/')[-1] + '/' + prop.find('value').text
50             else:
51                 storage = "/user/" + os.getenv('HADOOP_USER_NAME')
52     storage_environment_params = {'STORAGE':storage}
53     storage_environment = cml.create_environment_variable(storage_environment_params)
54     os.environ[ 'STORAGE' ] = storage
55
56
57

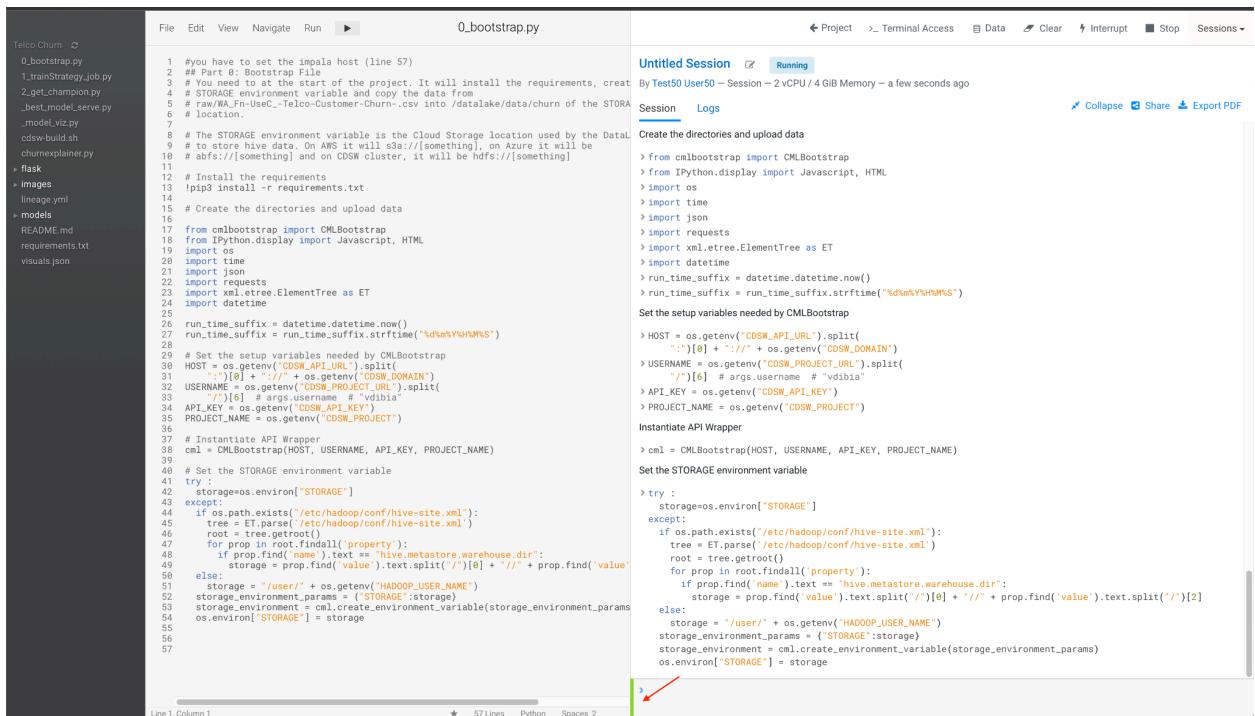
```

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.



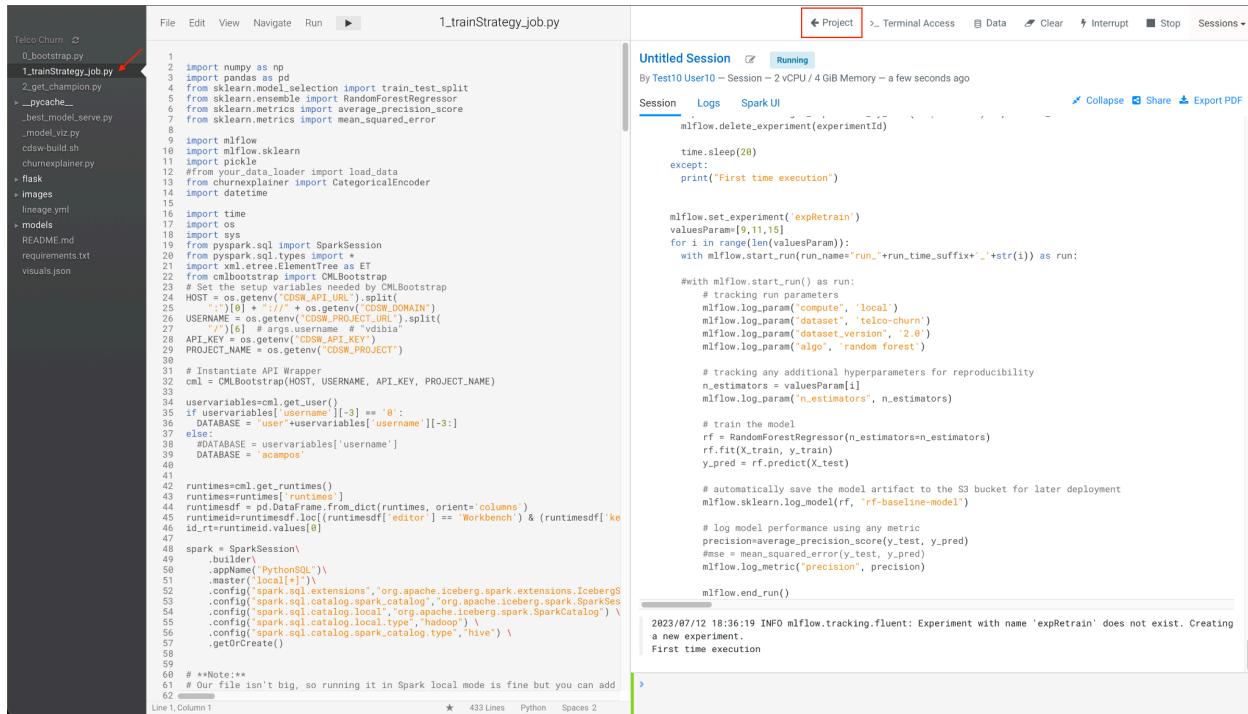
The green command bar indicates that the execution of the code has been finished. This bootstrap code takes 3-4 minutes to run.

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



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.



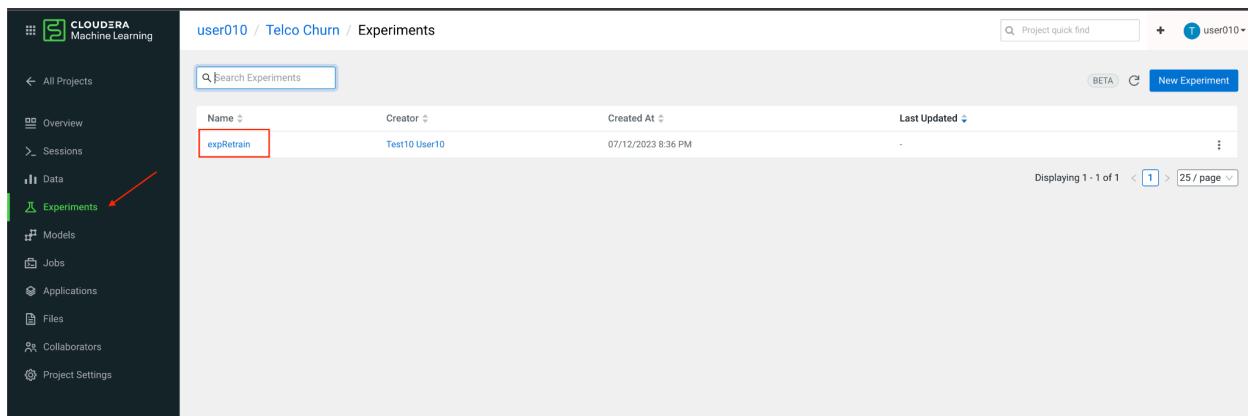
The screenshot shows the Jupyter Notebook interface. On the left, a sidebar lists files: 0\_bootstrap.py, 1\_trainStrategy\_job.py (highlighted with a red arrow), 2\_get\_champion.py, \_pycache\_, \_best\_model.serve.py, \_model\_viz.py, cdkw-build.sh, churnexplainer.py, flask, images, lineage.yml, models, README.md, requirements.txt, and visualis.json. The main area displays the code for `1_trainStrategy_job.py`. The code imports various libraries like numpy, pandas, sklearn, and mlflow. It sets up a SparkSession, loads data from a CSV file, and uses a RandomForestRegressor. It also handles command-line arguments for host, user, and project name. The code then creates a CMLBootstrap object, sets up a runtime environment, and starts a Random Forest classifier. Finally, it logs metrics and ends the experiment. The right side shows the "Untitled Session" running, with the status bar indicating "Running". The terminal output shows the experiment starting and running successfully.

```

File Edit View Navigate Run > Project < Terminal Access Data Clear Interrupt Stop Sessions
File Edit View Navigate Run > Project < Terminal Access Data Clear Interrupt Stop Sessions
1_trainStrategy_job.py
1
2 import numpy as np
3 import pandas as pd
4 from sklearn.model_selection import train_test_split
5 from sklearn.ensemble import RandomForestRegressor
6 from sklearn.metrics import average_precision_score
7 from sklearn.metrics import mean_squared_error
8
9 import mlflow
10 import mlflow.sklearn
11 import pickle
12 #from mlflow.data import load_data
13 #from churnexplainer import CategoricalEncoder
14 import datetime
15
16 import time
17 import os
18 import sys
19 from pyspark.sql import SparkSession
20 from pyspark.sql.types import *
21 import xml.etree.ElementTree as ET
22 from cmlbootstrap import CMLBootstrap
23 # Set environment variables by CMLBootstrap
24 HOST = os.getenv('CDSW_API_URL').split(
25     ':')[0] + ":" + os.getenv('CDSW_DOMAIN')
26 USERNAME = os.getenv('CDSW_PROJECT_URL').split(
27     '/')[1]
28 API_KEY = os.getenv('CDSW_API_KEY')
29 PROJECT_NAME = os.getenv('CDSW_PROJECT')
30
31 # Instantiate API Wrapper
32 cml = CMLBootstrap(HOST, USERNAME, API_KEY, PROJECT_NAME)
33
34 uservariables=cml.get_user()
35 if uservariables['username'][:-3] == '0':
36     DATABASE = uservariables['username'][:-3]
37 else:
38     #DATABASE = uservariables['username']
39     DATABASE = 'acmpos'
40
41
42 runtimes=cml.get_runtimes()
43 runtimes=runtimes[runtimes['runtimeid']]
44 runtimesdf = pd.DataFrame.from_dict(runtimes, orient='columns')
45 runtimeid=runtimesdf.loc[(runtimesdf['editor'] == 'Workbench') & (runtimesdf['ke
46 id']==runtimeid.values[0])
47
48 spark = SparkSession
49 .builder
50 .appName("PythonSQL")
51 .master("local[1]")
52 .config("spark.sql.extensions", "org.apache.iceberg.spark.extensions.IcebergS
53 .config("spark.sql.catalog.spark_catalog", "org.apache.iceberg.spark.SparkCatalog")
54 .config("spark.sql.catalog.spark_catalog", "org.apache.iceberg.spark.SparkCatalog")
55 .config("spark.sql.catalog.local_type", "Hadoop")
56 .config("spark.sql.catalog.spark_catalog_type", "hive")
57 .getOrCreate()
58
59
60 # **Note:**#
61 # Our file isn't big, so running it in Spark local mode is fine but you can add
62

```

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



The screenshot shows the Cloudera Machine Learning project home. The left sidebar has a green highlighted "Experiments" section. The main area shows a table of experiments. One experiment, "expRetrain", is highlighted with a red arrow and is listed in the table. The table columns are Name, Creator, Created At, and Last Updated. The status bar indicates "Displaying 1 - 1 of 1".

Name	Creator	Created At	Last Updated
expRetrain	Test10 User10	07/12/2023 8:36 PM	-

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.

Status	Start Time	Run Name	Duration	User	Source	Version	Models	algo	compute	dataset	precision
<input type="checkbox"/>	2023-07-12 08:36:19	run_3619_0	5.2s	user010	ipython3	8e811a	sklearn	random forest	local	telco-churn	1
<input type="checkbox"/>	2023-07-12 08:36:25	run_3619_1	3.8s	user010	ipython3	8e811a	sklearn	random forest	local	telco-churn	1
<input type="checkbox"/>	2023-07-12 08:36:28	run_3619_2	4.0s	user010	ipython3	8e811a	sklearn	random forest	local	telco-churn	1

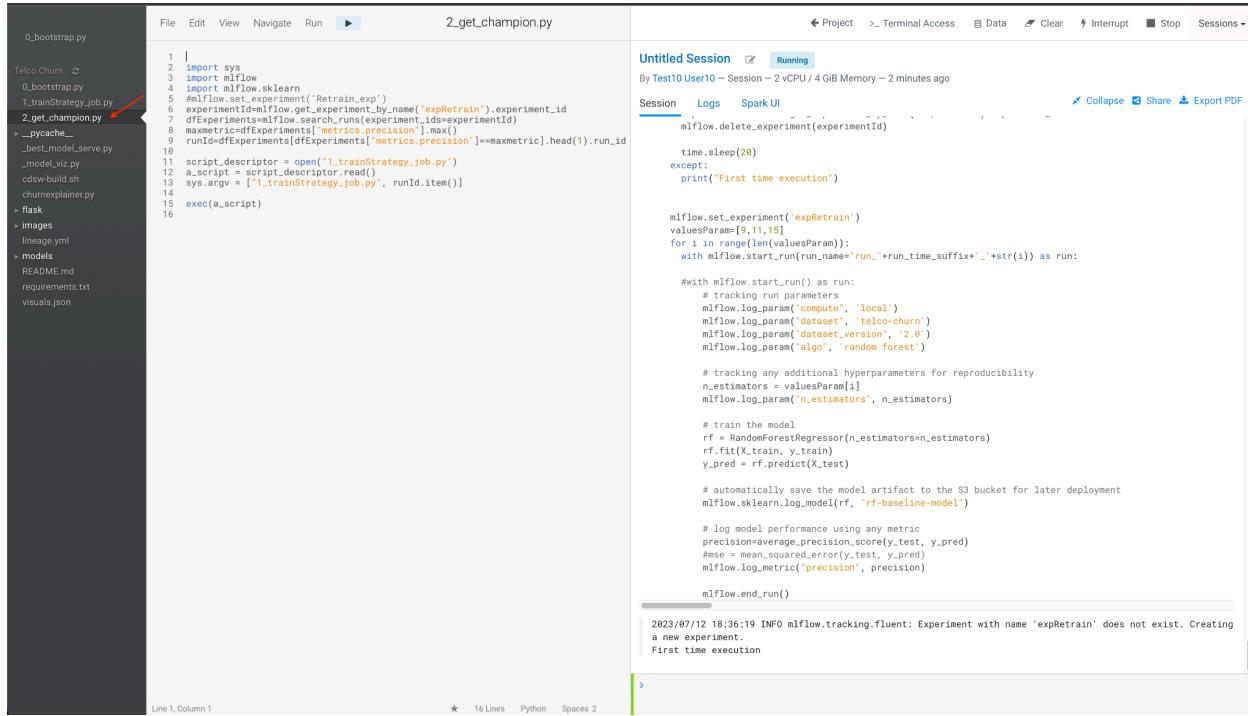
13. Let's go back to the session to run the last code. Since sessions run in Kubernetes containers, it's very easy to get back to where we were. Click on the option **Sessions** from the left menu, and later in the only session that will appear in the list.

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*.

Status	Session	Kernel	Creator	Created At	Duration
<input type="checkbox"/> Running	Untitled Session	(Python 3.7 Workbench Standard)	Test10 User10	07/12/2023 8:35 PM	Running since 1m 43s

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.



```

File Edit View Navigate Run ▶ Project Terminal Access Data Clear Interrupt Stop Sessions
2_get_champion.py
Telco Churn
0_bootstrap.py
1_trainStrategy_job.py
2_get_champion.py
> _pycache_
> _best_model_serve.py
> _model_viz.py
> cdsw-build.sh
> chimeExplainer.py
> flask
> images
> lineage.yml
> models
> README.md
requirements.txt
visuals.json

1 | import sys
2 | import mlflow
3 | import mlflow.sklearn
4 | experimentId=mlflow.get_experiment_by_name("expRetrain").experiment_id
5 | dfExperiments=mlflow.search_runs(experiment_ids=experimentId)
6 | dfExperiments=dfExperiments[["metrics_precision"]].max()
7 | runId=dfExperiments[dfExperiments["metrics_precision"]==maxMetric].head(1).run_id
8 |
9 | runId=dfExperiments[dfExperiments["metrics_precision"]==maxMetric].head(1).run_id
10 |
11 | script_descriptor = open("1_trainStrategy_job.py")
12 | a.script = script_descriptor.read()
13 | sys.argv = ["1_trainStrategy_job.py", runId.item()]
14 |
15 | exec(a.script)
16

```

Untitled Session Running  
By Test10 User10 – Session – 2 vCPU / 4 GiB Memory – 2 minutes ago

Session Logs Spark UI Collapse Share Export PDF

```

mlflow.delete_experiment(experimentId)

time.sleep(20)
except:
    print("First time execution")

mlflow.set_experiment('expRetrain')
valuesParam=[9,11,15]
for i in range(len(valuesParam)):
    with mlflow.start_run(run_name="run_" + run_time_suffix + "_" + str(i)) as run:
        # tracking any additional hyperparameters for reproducibility
        n_estimators = valuesParam[i]
        mlflow.log_param("n_estimators", n_estimators)

        # train the model
        rf = RandomForestRegressor(n_estimators=n_estimators)
        rf.fit(X_train, y_train)
        y_pred = rf.predict(X_test)

        # automatically save the model artifact to the S3 bucket for later deployment
        mlflow.sklearn.log_model(rf, "rf-baseline-model")

        # log model performance using any metric
        precision=average_precision_score(y_test, y_pred)
        mse = mean_squared_error(y_test, y_pred)
        mlflow.log_metric("precision", precision)

mlflow.end_run()

2023/07/12 18:36:19 INFO mlflow.tracking.fluent: Experiment with name 'expRetrain' does not exist. Creating a new experiment.
First time execution

```

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.

```

File Edit View Navigate Run > Project Terminal Access Data Clear Interrupt Stop Sessions
2_get_champion.py
1 import sys
2 import mlflow
3 import mlflow.sklearn
4 #mlflow.set_experiment('Retrain_exp')
5 experimentId=mlflow.get_experiment_by_name("expRetrain").experiment_id
6 dfExperiments=dfExperiments[dfExperiments['run_id'].max()==experimentId]
7 dfExperiments=dfExperiments[['metrics.precision']].max()
8 maxmetric=dfExperiments[['metrics.precision']]==maxmetric
9 runId=dfExperiments[dfExperiments[['metrics.precision']]==maxmetric].head(1).run_id
10
11 script_descriptor = open("1_trainStrategy_job.py")
12 a_script = script_descriptor.read()
13 sys.argv = ["1_trainStrategy_job.py", runId.item()]
14
15 exec(a_script)
16

```

Untitled Session Running  
By Test10 User10 – Session – 2 vCPU / 4 GiB Memory – 2 minutes ago  
Session Logs Spark UI Collapse Share Export PDF

```

> import sys
> import mlflow
> import mlflow.sklearn
> mlflow.set_experiment'Retrain_exp'
> experimentId=mlflow.get_experiment_by_name("expRetrain").experiment_id
> dfExperiments=dfExperiments[dfExperiments['run_id'].max()==experimentId]
> maxmetric=dfExperiments[['metrics.precision']].max()
> runId=dfExperiments[dfExperiments[['metrics.precision']]==maxmetric].head(1).run_id
> script_descriptor = open("1_trainStrategy_job.py")
> a_script = script_descriptor.read()
> sys.argv = ["1_trainStrategy_job.py", runId.item()]
> /usr/local/bin/python3.11: FutureWarning: 'item' has been deprecated and will be removed in a future version
n
> #!/usr/local/bin/python3.7
> exec(a_script)

Starting Experiments
Creating Model
Creating new model
New model created with access key msqqkhgmnf0lyt4ulz81kvb7mbduph9gi
Deploying Model...
Deploying Model....
Deploying Model.....
Model is deployed
Creating new model for visualization
New model created with access key ml17u0em8ypcxly6xid1c4a8g17q3foi
Deploying Model...
Deploying Model....
Deploying Model.....
Deploying Model.....
Model is deployed

```

Line 1, Column 1 ★ 16 Lines Python Spaces 2

15. Once on the home page of the project, you will see the Model displayed. Click on the one that starts with **ModelViz\_user0XX**.

Model	Source	Status	Replicas	CPU	Memory	Last Deployed	Actions
ModelViz_user050	.._model...	Deployed	1 / 1	1	2.00 GiB	Oct 11, 2023, 03:47 PM	<button>Stop</button>

Jobs  
This project has no jobs yet. Create a [new job](#) to document your analytics pipelines.

Files

Name	Size	Last Modified
.._pycache_	-	20 hours ago
.._flask	-	20 hours ago
.._images	-	20 hours ago
.._models	-	20 hours ago
.._0_bootstrap.py	1.95 kB	20 hours ago
.._1_trainStrategy.job.py	16.02 kB	20 hours ago
.._2_get_champion.py	508 B	20 hours ago
.._best_model.serve.py	2.74 kB	20 hours ago
.._model_viz.py	4.21 kB	20 hours ago

Workspace: mi-paris-atelier  
Cloud Provider: AWS (AWS)

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

The screenshot shows the Cloudera Machine Learning interface for the ModelOpsChurn\_user010 model. The left sidebar contains links for All Projects, Overview, Sessions, Data, Experiments, Models (selected), Jobs, Applications, Files, Collaborators, and Project Settings. The right panel displays the model's overview, deployment status (Deployed), and various configuration details. A central box shows code snippets for explaining a prediction and a sample response. At the bottom, there is a 'Test Model' section with an 'Input' field containing JSON data and a 'Result' field which is currently empty.

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.

The screenshot shows the Cloudera Machine Learning interface for a project named "user010".

- Sample Response:** A JSON object: `{}`

```
{
  "onlinesecurity": "No",
  "multiplelines": "No",
  "internetservice": "DSL",
  "seniorcitizen": "No",
  "techsupport": "No"
}
```
- Test Model:**
  - Input:** A JSON object with the same fields as the sample response.
  - Buttons:** "Test" (highlighted with a red border) and "Reset"
- Model Resources:**

Comment	Initial revision.
Runtime Image	Python 3.7 (Standard)
File	_best_model.serve.py
Function	explain
- Result:**
  - Status:** success
  - Response:** A JSON object containing deployment details and prediction probability.
  - Replica ID:** modelopschurn-user010-19-14-6c5d7947ff-52kzg

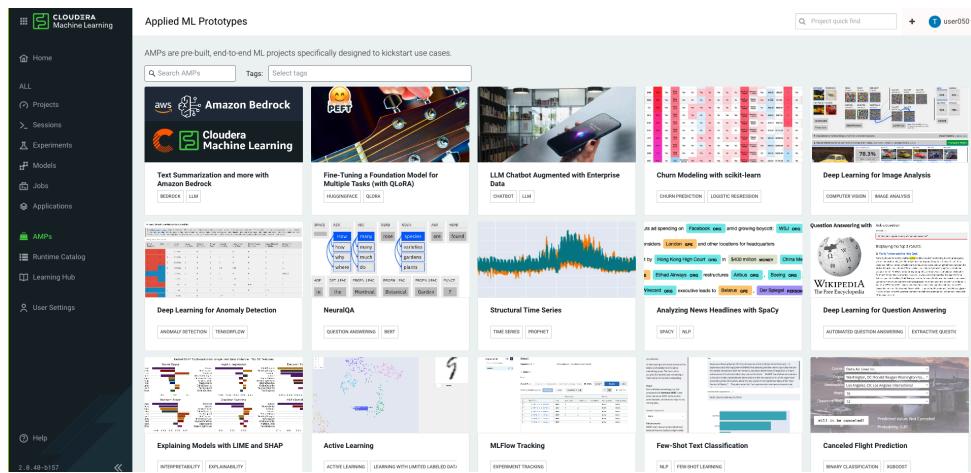
Bottom navigation: Workspace: ssa-cml-workspace, Cloud Provider: AWS (AWS)

## 6. Optional Labs

### ○ Goals

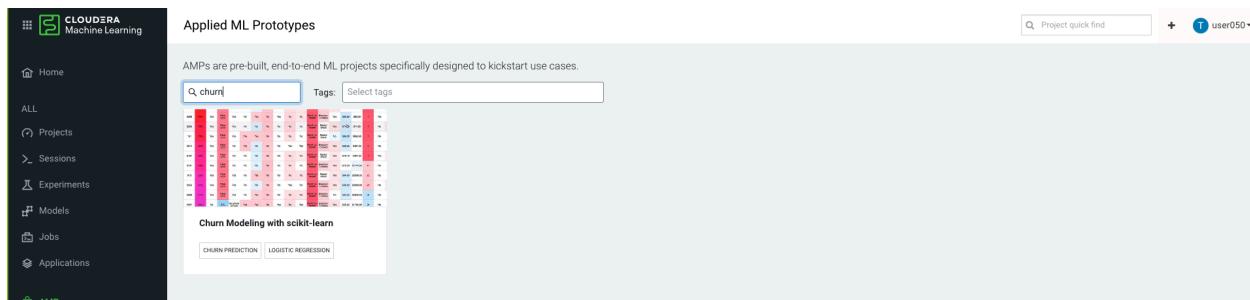
- Deploy Applied Machine Learning Models
  - Discover and query data using HUE
  - Add a new field that makes calls to the ML model
  - Add the new field to the dashboard
- **ML -Deploy Applied Machine Learning Model (Instructor Only)**

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

The screenshot shows the Cloudera Machine Learning interface. On the left, there's a sidebar with various navigation options like Home, Projects, Sessions, Experiments, Models, Jobs, Applications, and AMPs. The AMPs section is currently selected. In the main area, there's a search bar and a list of pre-built ML projects. One project, 'churn', is highlighted. A modal window titled 'Churn Modeling with scikit-learn' is overlaid on the page. The modal contains a brief description of the project, which is a logistic regression classification model for churn prediction using scikit-learn. It also lists four categories: CHURN PREDICTION, LOGISTIC REGRESSION, EXPLAINABILITY, and LIME. Below these categories are two buttons: 'View on Github' and 'Configure Project'. The 'Configure Project' button is highlighted.

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

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

The screenshot shows the 'Configure Project' screen for the 'Churn Modeling with scikit-learn' prototype. The title is 'Configure Project: Churn Modeling with scikit-learn - user050'. It shows the AMP Name as 'ML Churn Prototype (v2)' and a note that it's a prototype to demonstrate building a churn model on CML. The screen is divided into sections:

- The settings below were defined by the AMP:**

Name	Value	Description
DATA_LOCATION	data/churn_prototype	Relative path that will be used to store the data used for this prototype. This should be a location you have write access to, and which is suitable for non-production data.
HIVE_DATABASE	default	Name of the Hive database that will be used to create the Hive table used for this prototype. This should be a Hive database you have write access to, and which is suitable for non-production data.
HIVE_TABLE	churn_prototype	Name of the Hive table that will be created and populated with the data used for this prototype. If the table already exists, the prototype will assume it already contains the data for this prototype.
- Runtime**

Editor	Kernel	Edition	Version
Workbench	Python 3.7	Standard	2023.08

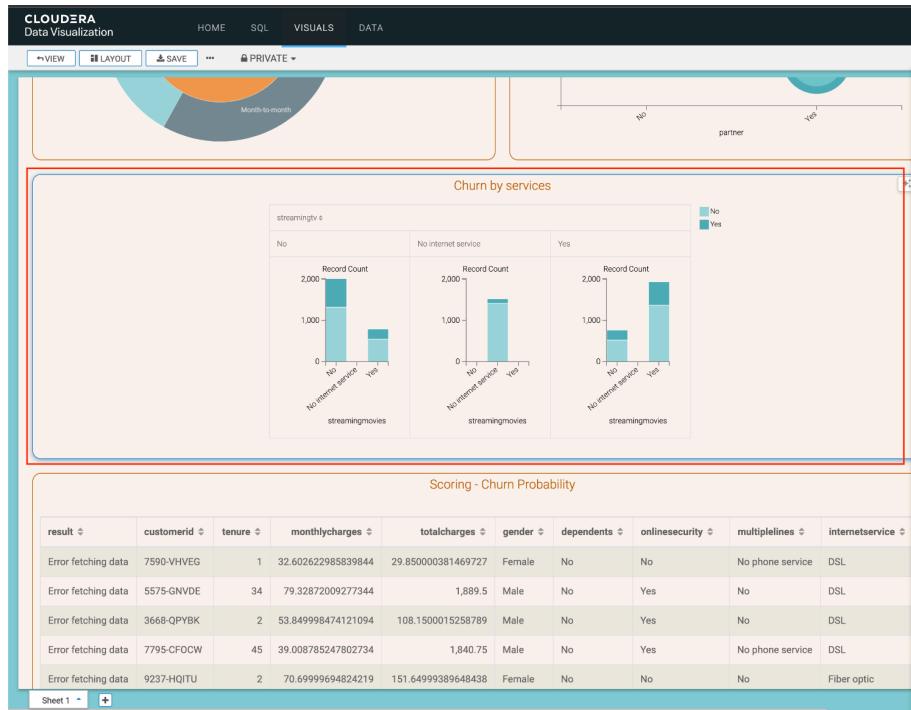
Enable Spark:  [disabled]
- Buttons:** Cancel, Launch Project

## 5. Deployment launches

The screenshot shows the Cloudera Machine Learning interface. On the left, there's a sidebar with navigation links like Home, All Projects, Project Overview, Sessions, Data, Experiments, Models, Jobs, Applications, Files, Collaborators, AMPs, Runtime Catalog, Learning Hub, User Settings, Help, and a footer note about version 2.0.48-b157. The main content area is titled 'AMP Status' for 'ML Churn Prototype (v2)'. It says 'Completed 0 of 7 steps'. Step 1 is shown as 'started 10/12/2023 2:02 PM' with a link to 'View details'. The step description mentions installing requirements, creating storage, and copying data from raw/WA\_Fn-UseC\_-Telco-Customer-Churn-.csv to specified path of the Storage location. Steps 2, 3, and 4 are listed as 'not yet started'.

### ○ Add a third visual element - Optional

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



## ○ One more visual element - Optional

1. If you 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**.

The screenshot shows the Cloudera Dashboard Designer interface. On the left, there are three panels: 'Dimensions' (containing 17 items), 'Measures' (containing 3 items), and 'Tooltips' (with a placeholder 'drag fields to add here'). On the right, the main workspace displays a 'DATA' section with a connection icon and a 'Dimensions' panel containing 18 items. Below it is a 'Measures' panel containing 4 items. A search bar and a sample mode switch are also visible.

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

## ○ Data Discovery and SQL Analysis Using HUE Dashboard - Optional

1. On your Cloudera landing page, go do Data Warehouse.
  - Click on the **HUE** button, in your Data Warehouse dashboard.
  - Make sure to select the one with **Impala** enabled.

Overview

The screenshot shows the Cloudera Data Warehouse HUE dashboard. At the top, there's a header with a 'Get started with Data Warehouse' section and a 'Management Console' link. Below the header, there are three main sections: 'Create' (with a 'See More' link), 'Query and visualize data' (with a 'See More' link), and 'Guides and More' (with a 'See More' link). The main content area displays 'Database Catalogs' and 'Virtual Warehouses'. Under 'Database Catalogs', there are two entries: 'paris-atelier-dl-default' (Running) and 'tuya-matu-dl-default' (Stopped). Each entry shows 'TOTAL CORES', 'TOTAL MEMORY', and 'VIRTUAL WAREHOUSES'. Under 'Virtual Warehouses', there are three entries: 'paris-atelier-impala' (Running), 'paris-atelier-hive' (Stopped), and another 'paris-atelier-dl-default' entry (Running). Each virtual warehouse entry shows 'EXECUTORS', 'TOTAL CORES', 'TOTAL MEMORY', and 'TYPE' (IMPALA or HIVE).

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.

The screenshot shows the Hue web interface. At the top, there's a search bar with placeholder text 'Search data and saved documents...'. Below the search bar, there's a navigation bar with icons for Tables, Jobs, and other tools. The main area is divided into several panels: a left sidebar with a tree view of 'Tables' under 'user050' (including 'telco\_data\_curated' and 'telco\_iceberg\_kafka'), a central panel for 'Impala' queries with a text input field containing 'Example: SELECT \* FROM tablename, or press CTRL + space', and a right panel titled 'Tables' which says 'No tables identified'. There are also tabs for 'Jobs' and 'File Browser'.

3. Inside your Hue window, run the following SQL 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

- 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

- 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

## ○ Part 2: Add a New Field - Optional

1. Edit the previously created Dataset, in Data -> <user\_assigned>.telco\_data\_curated.

The screenshot shows the Cloudera Data Visualization interface. On the left, there's a sidebar with connection management (New Connection, All Connections, ImpalaConn, samples). The main area is titled 'Datasets' and shows a table of datasets. One dataset is selected: 'user050.telco\_data\_curated'. The table columns include Title/Table, ID, Created, Last Updated, Modified By, and # Dashboards. The dataset details show it was created on May 29, 2023, by user050, and has 0 dashboards.

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.

The screenshot shows the 'Fields' page for the dataset 'user050.telco\_data\_curated'. The left sidebar lists 'Dataset Detail', 'Related Dashboards', 'Fields' (selected), 'Data Model', 'Time Modeling', 'Segments', 'Filter Associations', and 'Permissions'. The main area is divided into 'Dimensions' and 'Measures'. The 'Dimensions' section contains 18 items under 'telco\_data\_curated': multiplelines, paperlessbilling, gender, onlinesecurity, internetservice, techsupport, contract, churn, seniorcitizen, deviceprotection, streamingtv, streamingmovies, partner, customerid, dependents, onlinebackup, phoneservice, and paymentmethod. The 'Measures' section contains 3 items under 'telco\_data.curated': totalcharges, monthlycharges, and tenure.

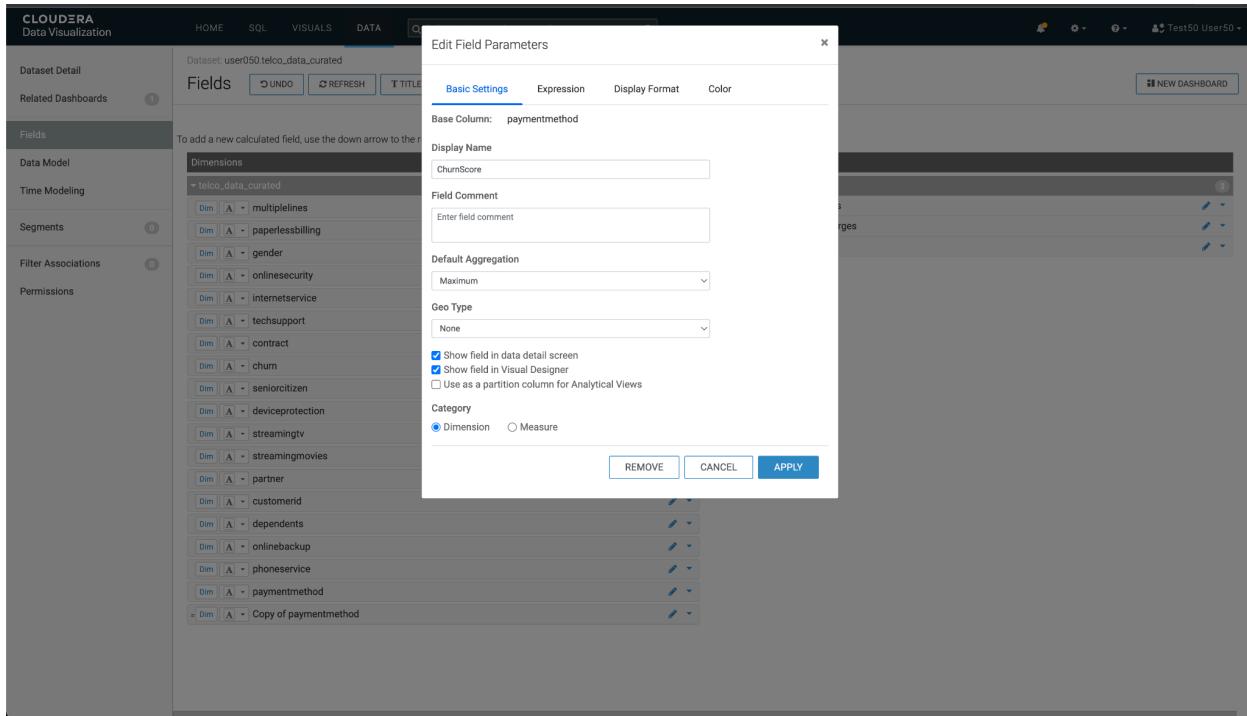
3. In the list of **Dimensions**, click the down arrow of the last field in the list (paymentmethod), and select the option **Clone**.

The screenshot shows the Cloudera Data Visualization interface. The left sidebar has sections for Dataset Detail, Related Dashboards, Fields, Data Model, Time Modeling, Segments, Filter Associations, and Permissions. The main area is titled 'Dataset: user050.telco\_data\_curated'. It has tabs for Fields, Measures, and Data. The Fields tab is active, showing the 'Dimensions' and 'Measures' sections. In the Dimensions section, the 'telco\_data\_curated' group contains 18 items, including 'multiplelines', 'paperlessbilling', 'gender', 'onlinesecurity', 'internetservice', 'techsupport', 'contract', 'churn', 'seniorcitizen', 'deviceprotection', 'streamingtv', 'streamingmovies', 'partner', 'customerid', 'dependents', 'onlinebackup', 'phoneservice', and 'paymentmethod'. The 'paymentmethod' item has a small downward arrow icon to its right. A tooltip box labeled 'Clone' is overlaid on this arrow. Below the dimensions, there are options to 'Hide' or 'Create Hierarchy'. The Measures section shows three items: 'totalcharges', 'monthlycharges', and 'tenure'.

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

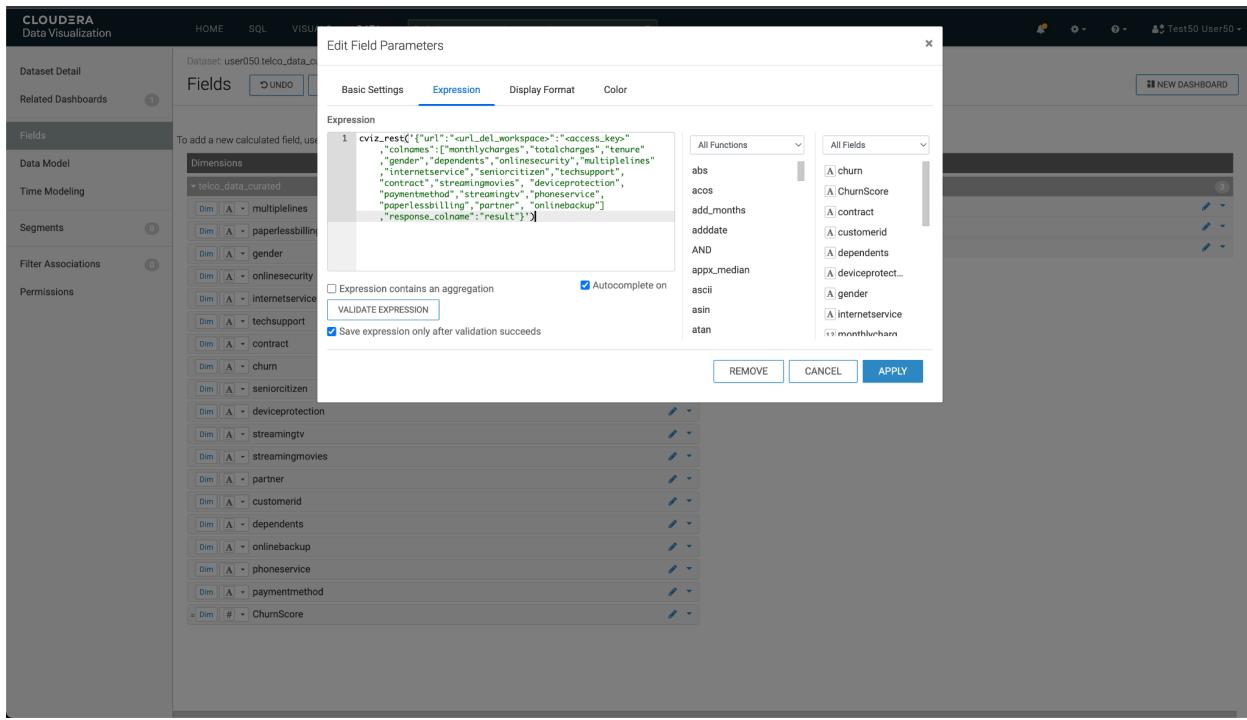
This screenshot shows the same interface after cloning the 'paymentmethod' dimension. The 'Dimensions' section now includes an additional item: '+ Dim | A | - Copy of paymentmethod'. This new item also has a small downward arrow icon to its right. A tooltip box labeled 'Edit Field' is overlaid on this arrow. The rest of the interface remains the same, with the 'Measures' section still showing 'totalcharges', 'monthlycharges', and 'tenure'.

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","streamingtvtv","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.

Model	Source	Status	Replicas	CPU	Memory	Last Deployed	Actions
ModelViz_user050	13_mod...	Deployed	1 / 1	1	2.00 GiB	May 29, 2023, 03:54 PM	<button>Stop</button>
ModelOpsChurn_user050	11_best...	Deployed	1 / 1	1	2.00 GiB	May 29, 2023, 03:53 PM	<button>Stop</button>

Name	Runs / Failures	Duration	Status	Latest Run	Actions
deploy_best_model	0 / 0	00:00	Not Yet Run	-	<button>Run</button>
retrain	0 / 0	00:00	Not Yet Run	-	<button>Run</button>
avisoPerformance	0 / 0	00:00	Not Yet Run	-	<button>Run</button>
Check Model	0 / 0	00:00	Not Yet Run	-	<button>Run</button>

Name	Size	Last Modified
__pycache__	-	15 hours ago
flask	-	15 hours ago
images	-	15 hours ago
models	-	15 hours ago
raw	-	15 hours ago
0_bootstrap.py	1.95 kB	15 hours ago
0b_create_jobs.py	5.60 kB	15 hours ago

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.

The screenshot shows the Cloudera Data Visualization interface. A modal window titled "Edit Field Parameters" is open, specifically the "Expression" tab. The expression input field contains the following JSON query:

```

curl -H "Content-Type: application/json" -X POST https://modelservice.ml-369883c3-99e.ssa-hol.yuit-vbzg.cloudera.site/model -d '{"accessKey": "urllalikvu7g19uwerd4xh4m9w3k0c0", "request": {"data": {"colnames": ["monthlycharges", "totalcharges", "tenure", "gender", "dependents", "onlinesecurity", "multiplelines", "internetservice", "seniorcitizen", "techsupport", "contract", "streamingtv", "paperlessbilling", "partner", "phoneservice", "onlinebackup"], "response.colname": "result"}}, "columns": [{"name": "ChurnScore", "type": "float"}]}

```

Below the expression input, there are several checkboxes: "Expression contains an aggregation" (unchecked), "Autocomplete on" (checked), "VALIDATE EXPRESSION" (button), "Save expression only after validation succeeds" (checkbox checked), and "REMOVE", "CANCEL", and "APPLY" buttons.

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

The screenshot shows the Cloudera Machine Learning (CML) interface. On the left, a sidebar navigation includes "All Projects", "Overview", "Sessions", "Data", "Experiments", "Models" (selected), "Jobs", "Applications", "Files", "Collaborators", and "Project Settings".

The main content area is titled "ModelViz\_user050" and shows the "Overview" tab. It displays the following information:

- Description:** visualization a given model prediction
- Sample Code:**
  - Shell
  - Python
  - R

```

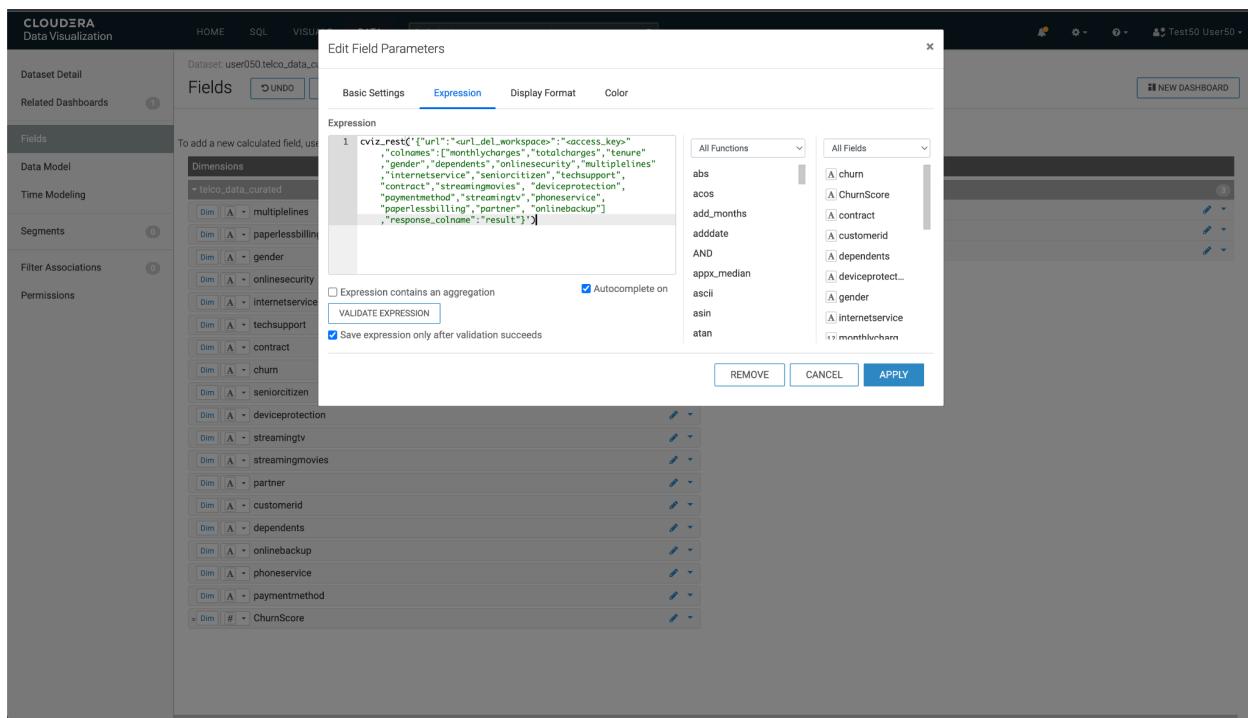
curl -H "Content-Type: application/json" -X POST https://modelservice.ml-369883c3-99e.ssa-hol.yuit-vbzg.cloudera.site/model -d '{"accessKey": "urllalikvu7g19uwerd4xh4m9w3k0c0", "request": {"data": {"colnames": ["monthlycharges", "totalcharges", "tenure", "gender", "dependents", "onlinesecurity", "multiplelines", "internetservice", "seniorcitizen", "techsupport", "contract", "streamingmovies", "deviceprotection", "paymentmethod", "streamingtv", "paperlessbilling", "partner", "phoneservice", "onlinebackup"], "response.colname": "result"}}, "columns": [{"name": "ChurnScore", "type": "float"}]}'
```
- Model Details:**

Source	Code
Model Id	8
Model CRN	cm.cdp.ml.us-west-1:508fd88f-8076-498a-acfb-6f8765cd5e8workspace814194cb-1c7e-48cd-9989-b499a79ed5f6/daes534c1-b214-45eb-acd0-101e651ff68d
Deployment Id	10
Deployment CRN	cm.cdp.ml.us-west-1:508fd88f-8076-498a-acfb-6f8765cd5e8workspace814194cb-1c7e-48cd-9989-b499a79ed5f6/cf985a5d-9870-4533-9f9a-d42addb56ed
Build Id	10
Build CRN	cm.cdp.ml.us-west-1:508fd88f-8076-498a-acfb-6f8765cd5e8workspace814194cb-1c7e-48cd-9989-b499a79ed5f6/0e00e2d9-80cb-4ee8-8304-79987673de32
- Deployed By:** user050
- Comment:** Initial revision.
- Runtime Image:** Python 3.7 (Standard)
- File:** 13\_model\_viz.py
- Function:** predict
- Model Resources:**

Replicas	1
Total CPU	1 vCPUs
Total Memory	2.00 GiB

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":"mjy1fowabqiwfpfb19s9ht6xmuvy0f2j","colnames":["monthlycharges","totalcharges","tenure","gender","dependents","onlinesecurity","multiplelines","internetservice","seniorcitizen","techsupport","contract","streamingmovies","deviceprotection","paymentmethod","streamingtvtv","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.

The screenshot shows the Cloudera Data Visualization interface. A modal window titled "Edit Field Parameters" is open, specifically on the "Expression" tab. The expression input field contains a JSON-like code snippet:

```

1 cviz.restC(["url":"https://modelservice.ml-369083c3-99e.sso-hol.yuit-vbzg.cloudro.site/model","accessKey":"mmmlolikv47oi5gwerd4kh493k9c2z","colnames":["monthlycharges","totalcharges","tenure","gender","dependents","partner","seniorcitizen","internetservice","deviceprotection","contract","streamingmovies","deviceprotection","paymentmethod","streamingtv","phoneservice","paperlessbilling","partner","onlinebackup"],"response_colname":"result")

```

Below the expression input, there are several checkboxes: "Expression contains an aggregation" (unchecked), "Autocomplete on" (checked), and "Save expression only after validation succeeds" (checked). At the bottom of the dialog, a green bar displays "Validation Successful!". There are "REMOVE", "CANCEL", and "APPLY" buttons at the bottom right.

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 **#**

The screenshot shows the Cloudera Data Visualization interface with the "Fields" panel open. A new field, "ChurnScore", has been added to the "Dimensions" section. The data type for this field is currently set to "Boolean". Below the data type dropdown, there are other options: "Integer" (represented by "#"), "Real" (represented by ".2"), "String", "Timestamp", and "Remove CAST". The "Measures" section is also visible on the right side of the Fields panel.

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

The screenshot shows the Cloudera Data Visualization interface. In the top navigation bar, the dataset 'user050:telco\_data\_curated' is selected. The main area is titled 'Fields' and contains two sections: 'Dimensions' and 'Measures'. The 'Dimensions' section lists 19 items under 'telco\_data\_curated', including 'multiplelines', 'paperlessbilling', 'gender', 'onlinesecurity', 'internetservice', 'techsupport', 'contract', 'churn', 'seniorcitizen', 'deviceprotection', 'streamingtv', 'streamingmovies', 'partner', 'customerid', 'dependents', 'onlinebackup', 'phoneservice', 'paymentmethod', and 'ChurnScore'. The 'Measures' section lists 3 items under 'telco\_data\_curated', including 'totalcharges', 'monthlycharges', and 'tenure'. At the bottom right of the Fields section, there is a green 'SAVE' button with a checkmark icon.

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.

The screenshot shows the Cloudera Data Visualization dashboard page. The top navigation bar includes 'HOME', 'SQL', 'VISUALS' (which is highlighted in blue), 'DATA', and a search bar. Below the navigation, there are buttons for 'NEW DASHBOARD' and 'NEW APP'. On the left, a sidebar shows 'All' (18), 'My Favorites' (0), 'WORKSPACES' (17), 'Public' (1), and 'Private' (1). The main area is titled 'All' and displays a grid of sample dashboards. One dashboard, 'Churn Analysis', is highlighted with a red box. Other visible dashboards include 'Deficiency Details: <>county:Queens>>', 'State of NYC', 'Sample App', 'Store Details:<owner\_name>', 'Cereal Comparisons', 'Earthquakes Around the World', 'Life Expectancy Dashboard', 'World Population & GDP Trends', 'Animated world population - GDP vs life expectancy', 'US State Population Trends', 'Census Dashboard', 'Global Threats', 'Time & Industry Threat View', 'Inspector View', 'Consumer View', 'Iris species w/ images', and 'Taxi rides application'. Each dashboard has a preview image and a 'View' button.

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

The screenshot shows a dashboard titled "partner". It contains three stacked bar charts under the heading "streamingtv". Each chart has "Record Count" on the y-axis (0 to 2,000) and "streamingmovies" on the x-axis. The legend indicates "No" (light blue) and "Yes" (dark blue). The first chart shows "No internet service" with approximately 1,800 for No and 200 for Yes. The second chart shows "No" with approximately 1,500 and "Yes" with approximately 500. The third chart shows "No" with approximately 1,000 and "Yes" with approximately 1,000. Below the charts is a table with columns: totalcharge, monthlycharges, tenure, multiplelines, paperlessbilling, gender, onlinesecurity, internetservice, techsupport, contract, and chn. The table contains six rows of data.

totalcharge	monthlycharges	tenure	multiplelines	paperlessbilling	gender	onlinesecurity	internetservice	techsupport	contract	chn
29.850000381469727	32.602622985839844	1	No phone service	Yes	Female	No	DSL	No	Month-to-month	N
1,889.5	79.32872009277344	34	No	No	Male	Yes	DSL	No	One year	N
108.1500015258789	53.849998474121094	2	No	Yes	Male	Yes	DSL	No	Month-to-month	Y
1,840.75	39.008785247802734	45	No phone service	No	Male	Yes	DSL	Yes	One year	N
151.64999389648438	70.69999694824219	2	No	Yes	Female	No	Fiber optic	No	Month-to-month	Y
820.5	99.6500015258789	8	Yes	Yes	Female	No	Fiber optic	No	Month-to-month	Y

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.

Important : Put all the objects in the **Dimensions** section of the Dashboard Designer. The Measures section should then be empty. (see screen copy below)

The screenshot shows the same dashboard as above, but with the "Build" menu open on the right side of the interface. The "Dimensions" section of the build menu is highlighted with a red box and contains the field "# ChurnScore". The "Measures" section is also visible, showing fields like "# record count", "totalcharges", "monthlycharges", and "tenure". The table below the charts now includes an additional column for "ChurnScore" at the beginning of the row headers.

ChurnScore	totalcharge	monthlycharges	tenure	multiplelines	paperlessbilling	gender	onlinesecurity	internetservice	techsupport	contract	chn
29.850000381469727	32.602622985839844	1	No phone service	Yes	Female	No	DSL	No	Month-to-month	N	
1,889.5	79.32872009277344	34	No	No	Male	Yes	DSL	No	One year	N	
108.1500015258789	53.849998474121094	2	No	Yes	Male	Yes	DSL	No	Month-to-month	Y	
1,840.75	39.008785247802734	45	No phone service	No	Male	Yes	DSL	Yes	One year	N	
151.64999389648438	70.69999694824219	2	No	Yes	Female	No	Fiber optic	No	Month-to-month	Y	
820.5	99.6500015258789	8	Yes	Yes	Female	No	Fiber optic	No	Month-to-month	Y	

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.

The screenshot shows the Cloudera Data Visualization interface. At the top, there are navigation tabs: HOME, SQL, VISUALS, and DATA. Below the tabs are buttons for VIEW, LAYOUT, and SAVE, along with a PRIVATE dropdown. The main area displays three stacked bar charts under the category 'streamingtv'. Each chart has 'Record Count' on the y-axis (0 to 2,000) and two categories on the x-axis: 'No internet service' and 'Yes'. The first chart is for 'streamingtv', the second for 'streamingmovies', and the third for 'partner'. The legend indicates that teal represents 'No' and light blue represents 'Yes'. To the right of the charts is a table with 10 rows of data. The columns are: ChurnScore, totalcharges, monthlycharges, tenure, multiplelines, paperlessbilling, gender, onlinesecurity, internetservice, and techsupport. The 'ChurnScore' column is highlighted in yellow. The table data is as follows:

ChurnScore	totalcharges	monthlycharges	tenure	multiplelines	paperlessbilling	gender	onlinesecurity	internetservice	techsupport
0	29.850000381469727	32.602622985839844	1	No phone service	Yes	Female	No	DSL	No
0	1,889.5	79.32872009277344	34	No	No	Male	Yes	DSL	No
0	108.1500015258789	53.849998474121094	2	No	Yes	Male	Yes	DSL	No
0	1,840.75	39.008785247802734	45	No phone service	No	Male	Yes	DSL	Yes
6	151.64999389648438	70.69999694824219	2	No	Yes	Female	No	Fiber optic	No
10	820.5	99.6500015258789	8	Yes	Yes	Female	No	Fiber optic	No

On the right side of the interface is the 'Dashboard Designer' panel, which includes sections for Dimensions, Measures, and Filters. The Dimensions section lists various fields like ChurnScore, totalcharges, tenure, etc. The Measures section lists Record Count, totalcharges, monthlycharges, and tenure. The Filters section is currently empty. At the bottom right of the Designer panel is a 'REFRESH VISUAL' button.

---

## 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.