CLOUDERA

DATA FLOW HANDS - ON

STUDENT GUIDE

Pre-requisites

For the ease of carrying out the workshop and considering the time at hand, we have already taken care of some of the steps that need to be considered before we can start with the actual Lab steps. The prerequisites that need to be in place are:

1. Streams Messaging Data Hub Cluster should be created and running.
2. Stream analytics Data Hub cluster should be created and running.
3. Data provider should be configured in SQL Stream Builder.
4. Have access to the file syslog-to-kafka.json.
5. Environment should be enabled as part of the CDF Data Service.

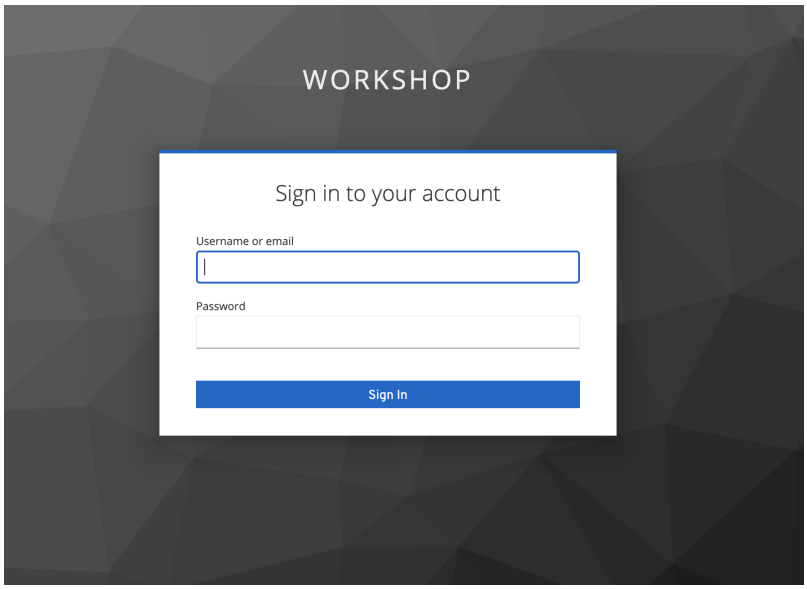
Lab 0 basically talks about verifying different aspects wrt to access and connections before we could begin with the actual steps.

Lab 0 - Introduction and setup

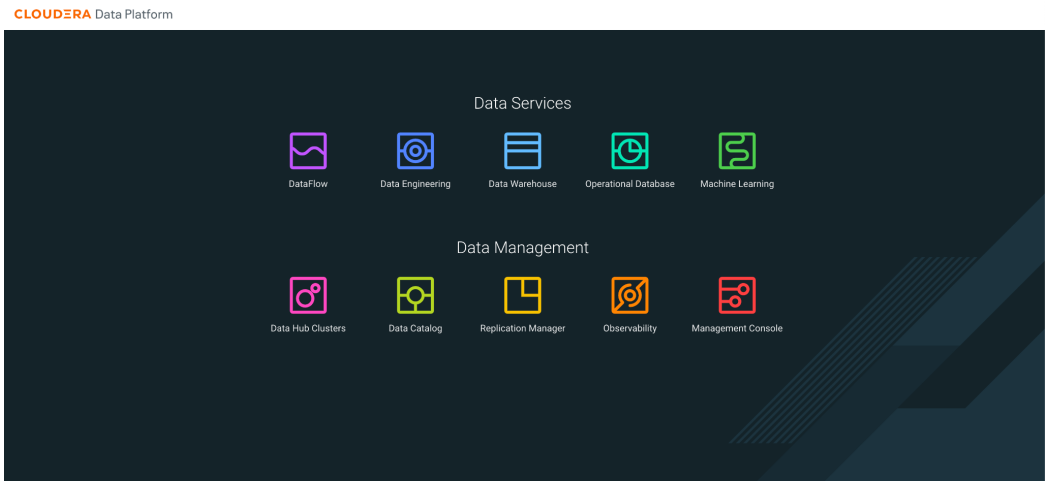
1. Verify access to the workshop environment

- The **INSTRUCTOR** will share the Workshop link and the credentials before the start of the workshop
- Open the shared link and login with the credentials assigned to you.

<Will be shared by the instructor at the start>



- You should land on the CDP Console as shown below.



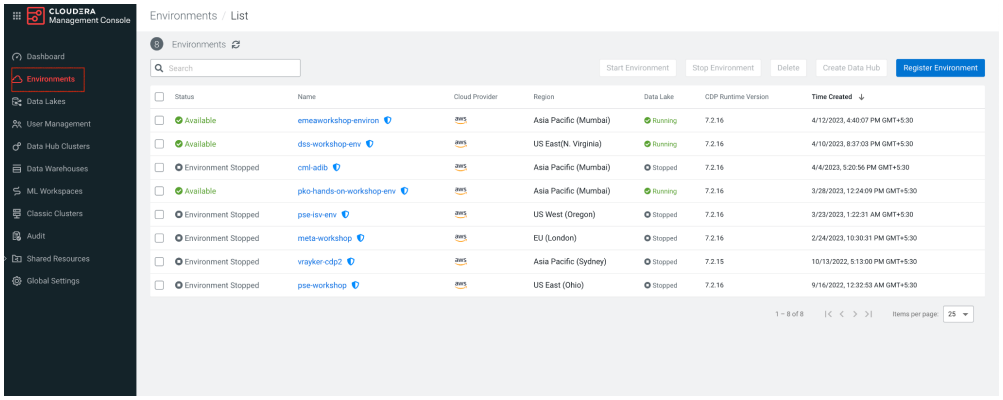
2. Verify permissions in Apache Ranger

NOTE: THESE STEPS HAVE ALREADY BEEN DONE FOR YOU, THIS SECTION WILL WALK YOU THROUGH HOW PERMISSIONS/POLICIES ARE MANAGED IN RANGER. PLEASE DO NOT EXECUTE THE STEPS IN THIS SECTION OR CHANGE ANYTHING.

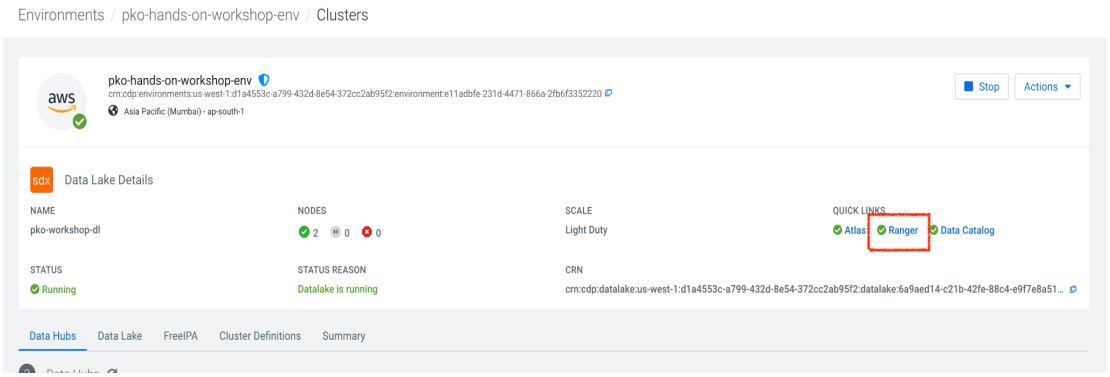
2.1 Accessing Apache Ranger

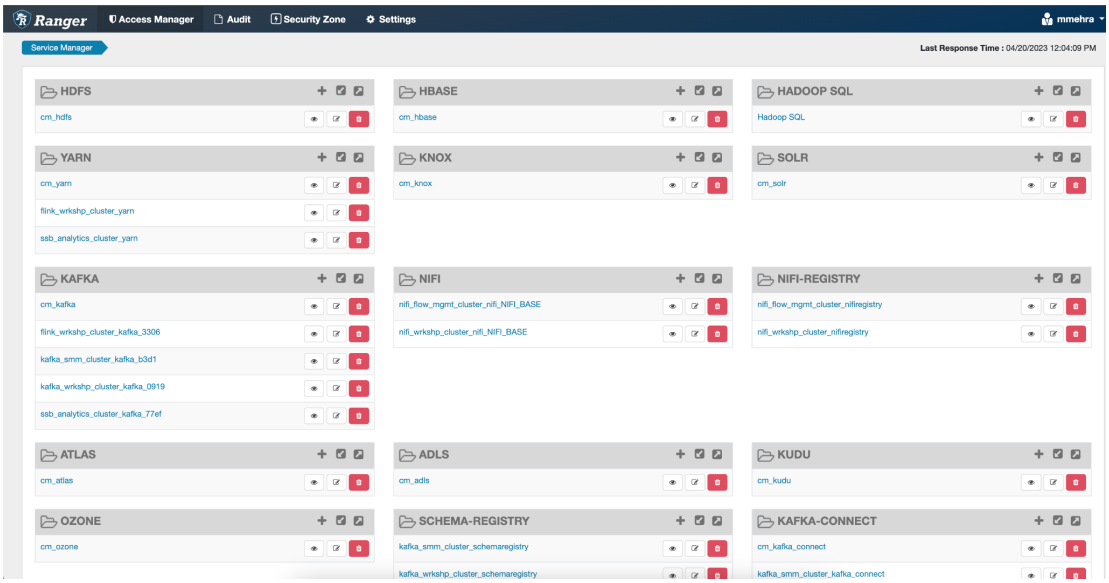
Step 1 : Click on Management Console

Step 2 : Click on Environments on the left tab



Step 3 : Select the environment that is shared by the instructor and click on the **Ranger** quick link to access the Ranger UI

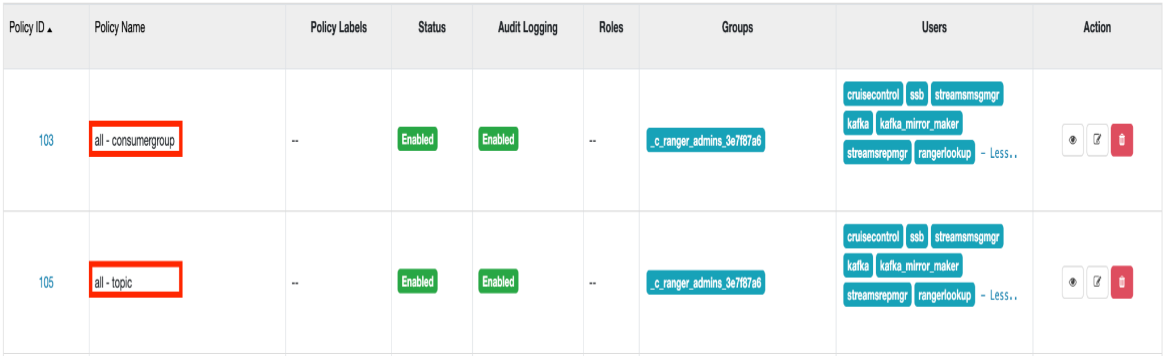




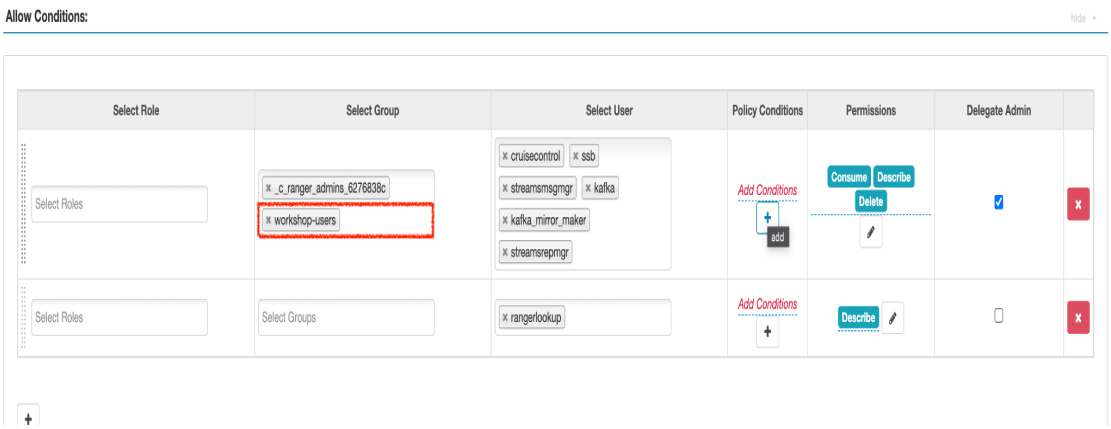
2.2 Kafka Permissions

1. In Ranger, select the Kafka repository that's associated with the stream messaging datahub.

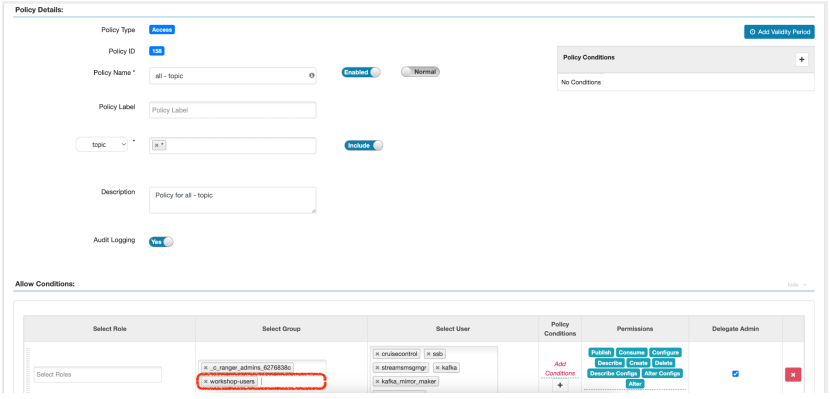
2. Verify if the user group(**workshop-users**) who will be performing the workshop is present in both **all-consumergroup** and **all-topic.**



- All-consumergroup



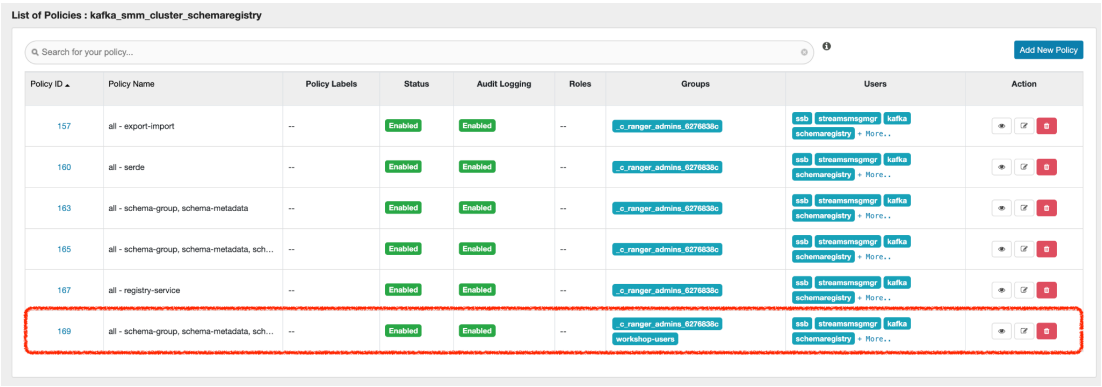
- all-topic

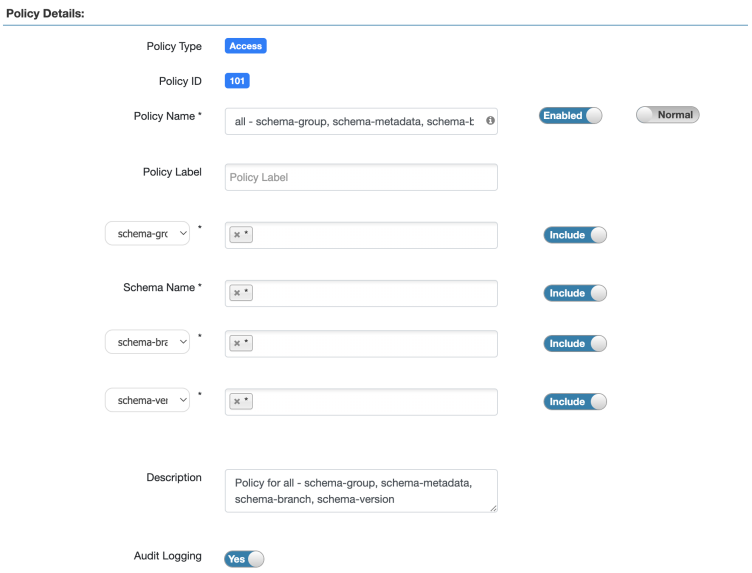


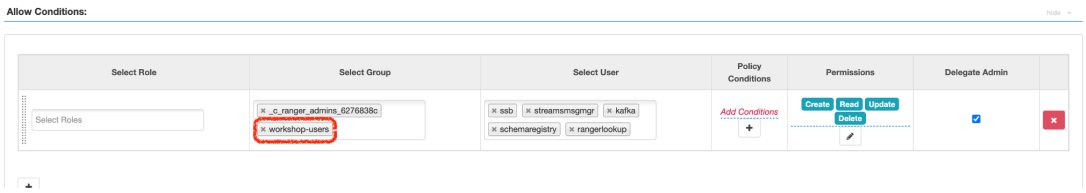
2.3 Schema Registry Permissions

1. In Ranger, select the Schema Registry repository that's associated with the stream messaging datahub.

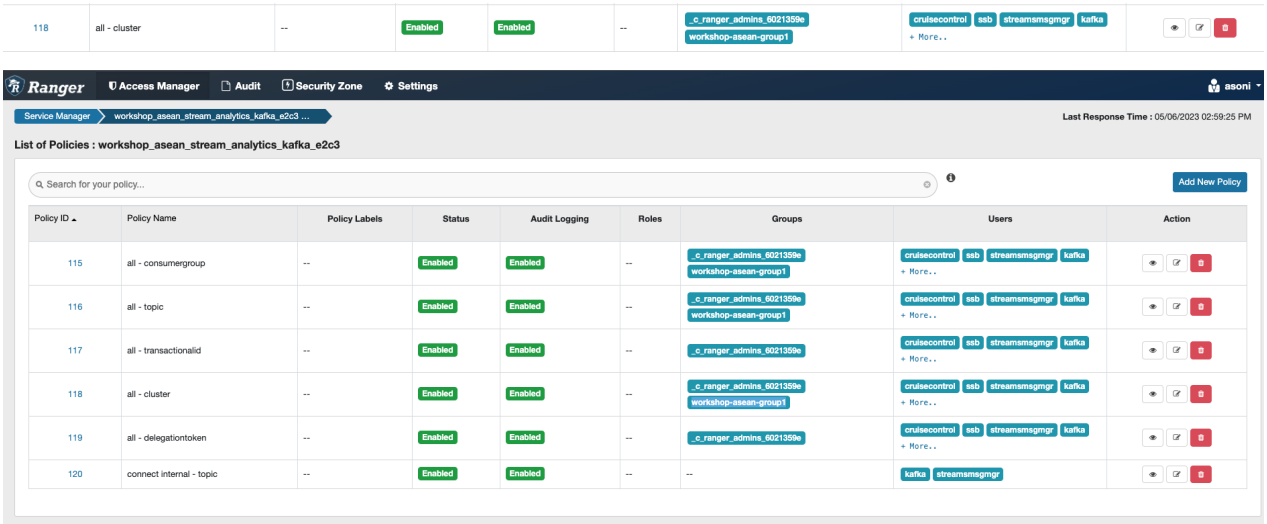
2. Verify if the user group(**workshop-users**) who will be performing the workshop is present in the Policy : **all - schema-group, schema-metadata, schema-branch, schema-version.**







Lastly, add the user group to the Streaming Analytics Data Hub Kafka Cluster Ranger permission as below:

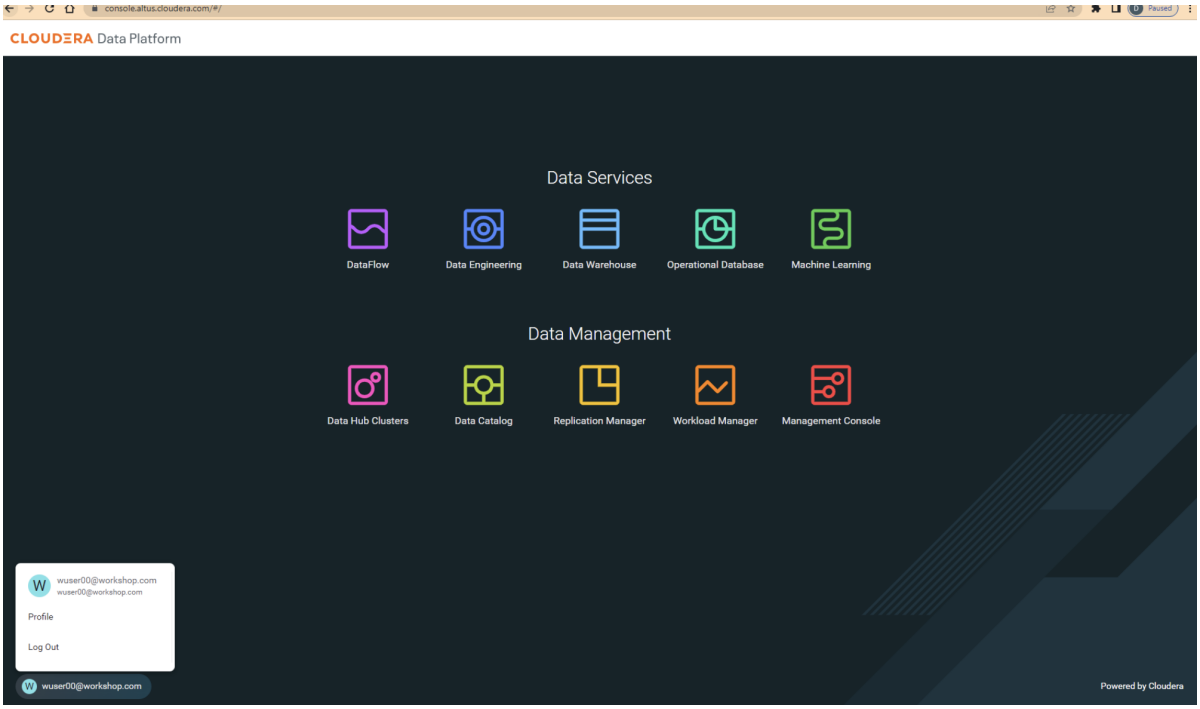


3. Update workload password

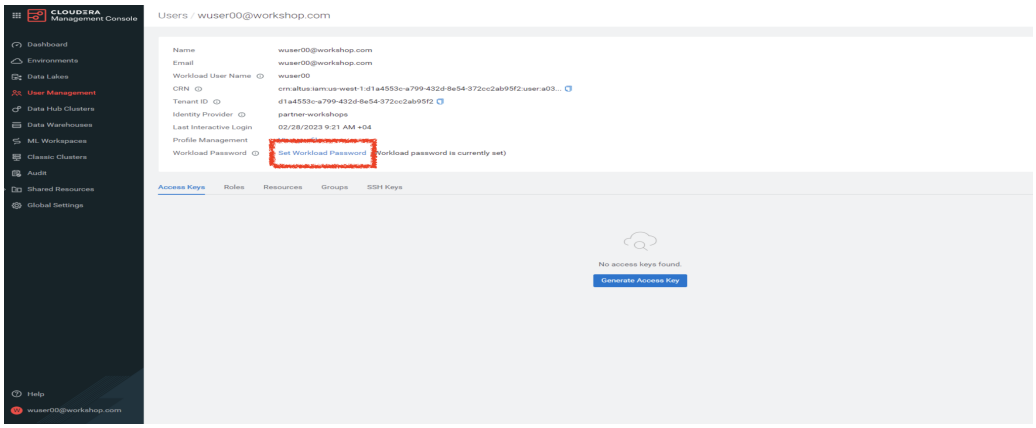
NOTE: THESE STEPS NEED TO BE PERFORMED BEFORE MOVING FORWARD

You will need to define your CDP Workload Password that will be used to access non-SSO interfaces. You may read more about it here. Please keep it with you. If you have forgotten it, you will be able to repeat this process and define another one.

- Click on your user name (Ex: apac00@workshop.com) at the lower left corner. Click on Profile.

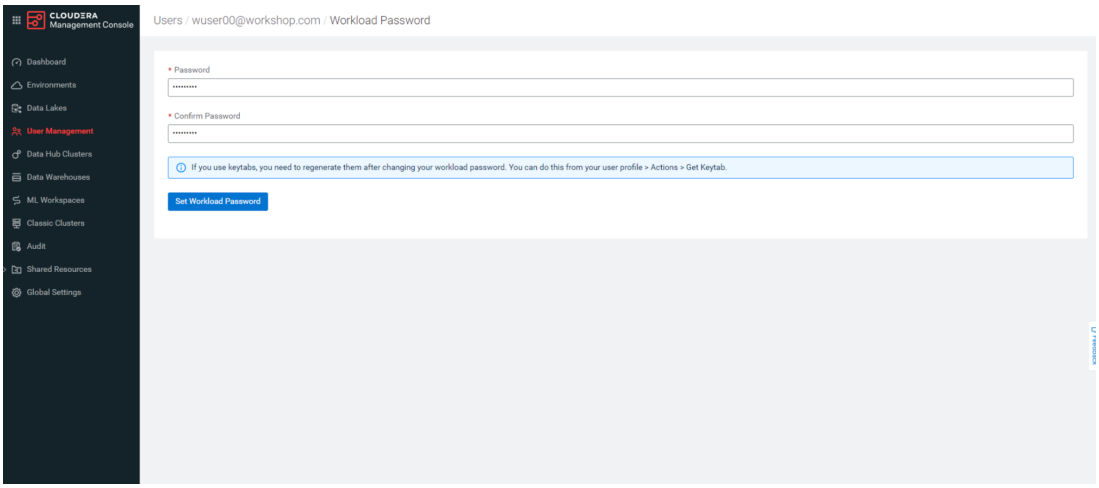


- Click option **Set Workload Password**.



- Enter the shared password.

NOTE: PLEASE ENTER THE SAME PASSWORD THAT WAS SHARED BY THE INSTRUCTOR. FAILING TO DO SO WILL LEAD TO ERRORS IN OUR LAB STEPS LATER ON



- Click the button Set Workload Password.

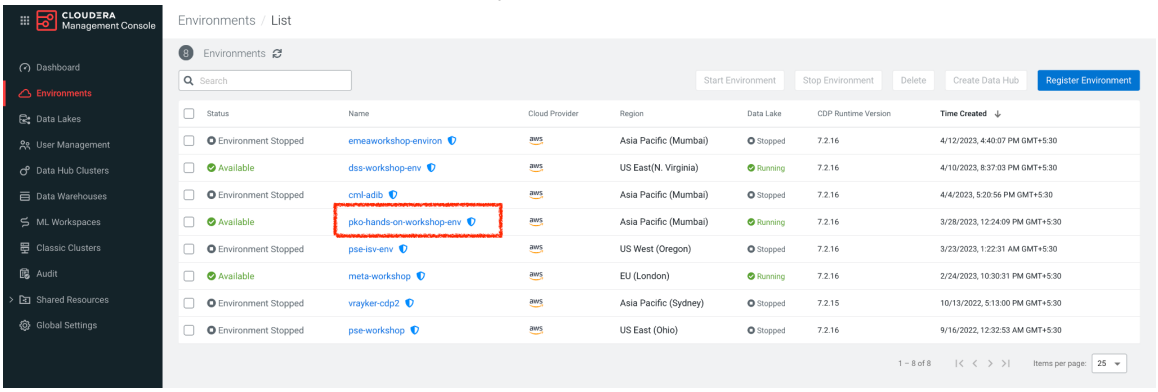
4. Obtain the Kafka Broker List

We will require the broker list to configure our processors to connect to our Kafka brokers which allow consumers to connect and fetch messages by partition, topic or offset.

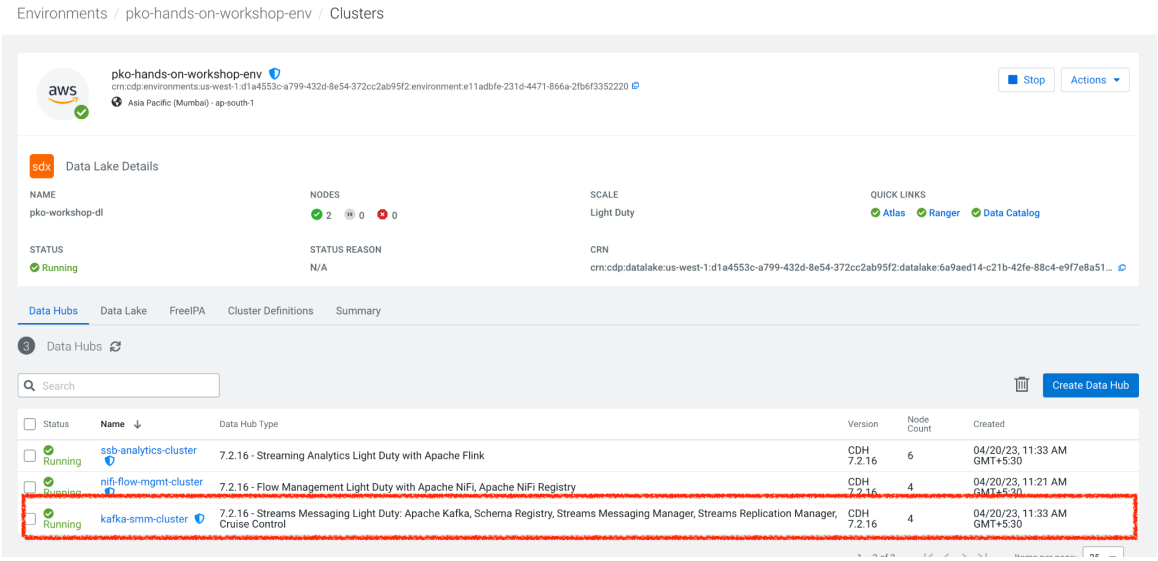
This information can be found in the Data Hub cluster associated to the Streams Messaging Manager

Step 1 : Access the Data Hub

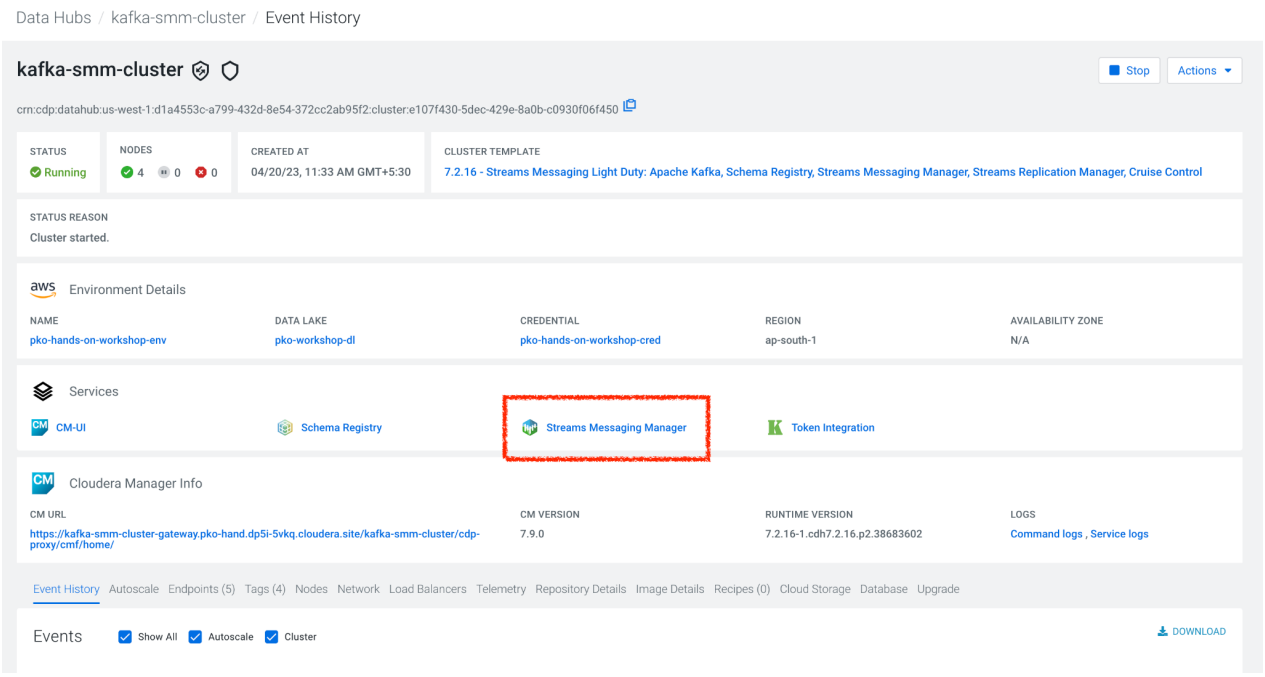
- Go to the environment that is shared by the INSTRUCTOR



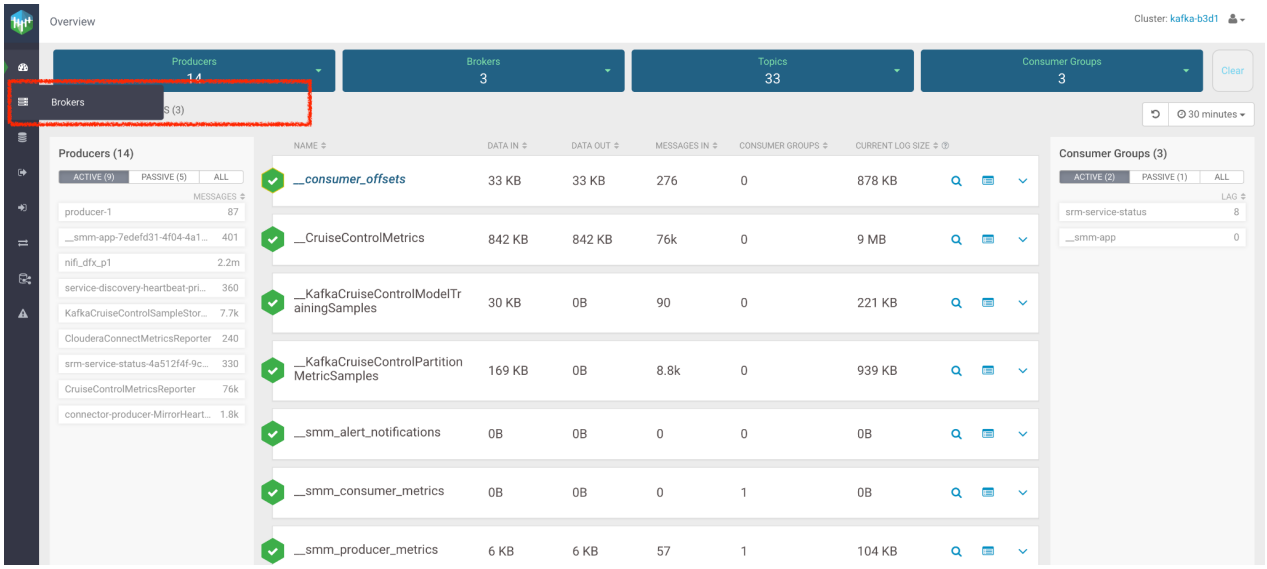
- Click on the Data Hub associated with Streams Messaging Manager (kafka-smm-cluster)



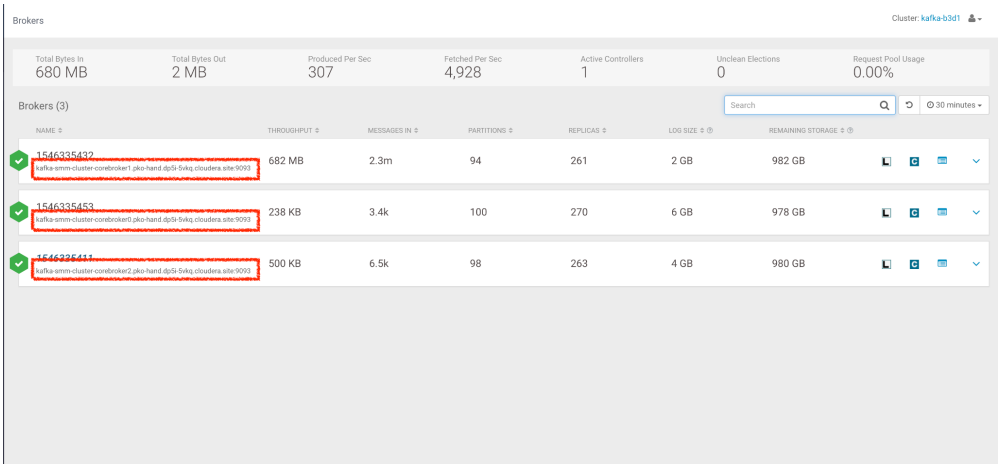
Step 2 : Go to the Streams Messaging Interface



Step 3 : Select Brokers from the left tab



Step 4 : Save the broker list

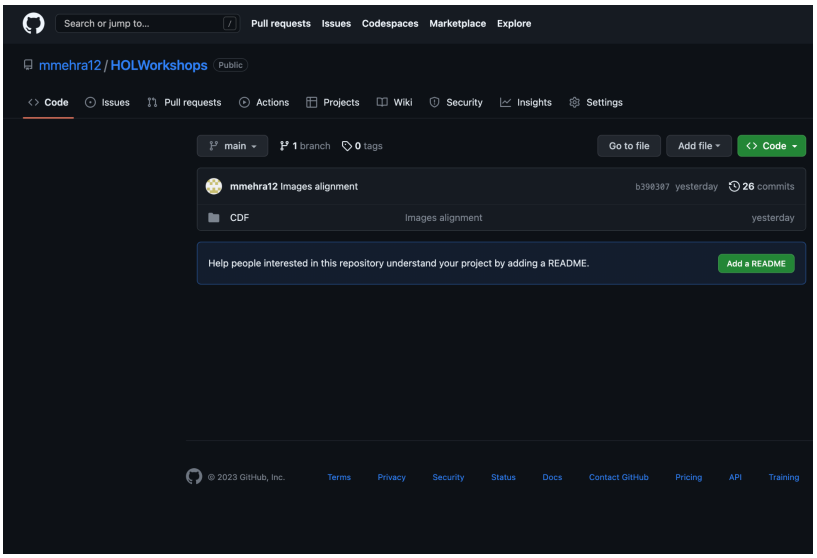


Example :

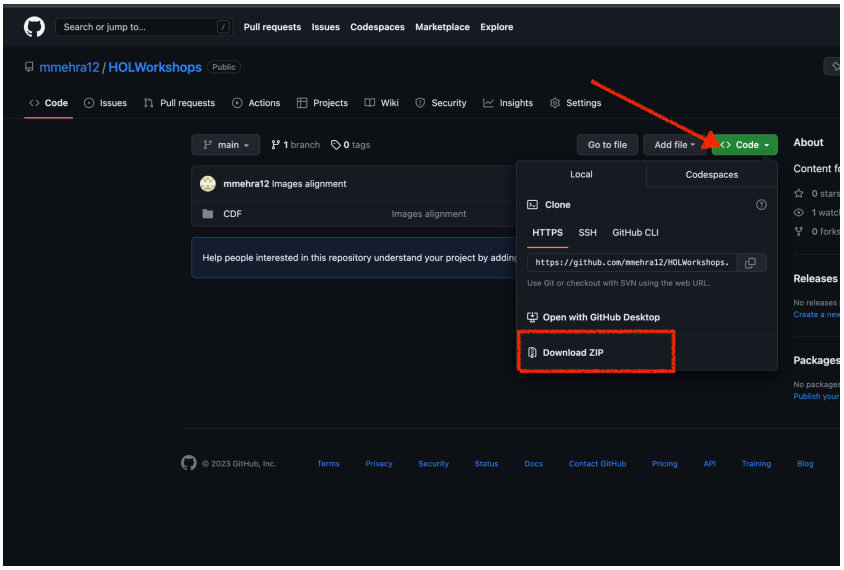
kafka-smm-cluster-corebroker1.pko-hand.dp5i-5vkq.cloudera.site:9093
kafka-smm-cluster-corebroker0.pko-hand.dp5i-5vkq.cloudera.site:9093
kafka-smm-cluster-corebroker2.pko-hand.dp5i-5vkq.cloudera.site:9093

5. Download Resources from GitHub

Step 1 : Access the URL shared by the instructor for GitHub

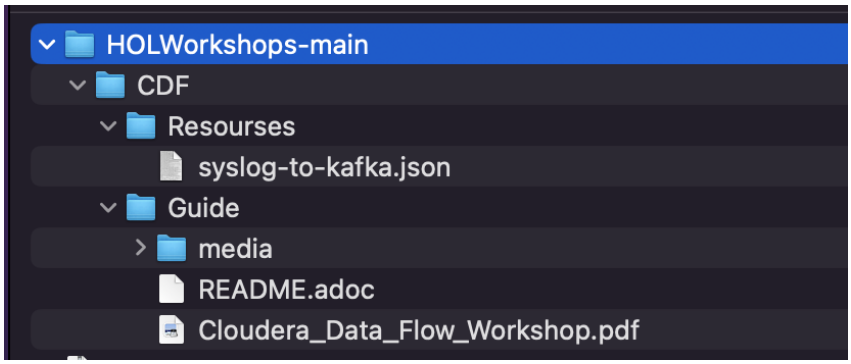


Step 2 : Download the repo as a zip file



Step 3 : Uncompress the Files

Uncompress the Files and you should have the following files and folders within it



We will use this at a later point in our Labs

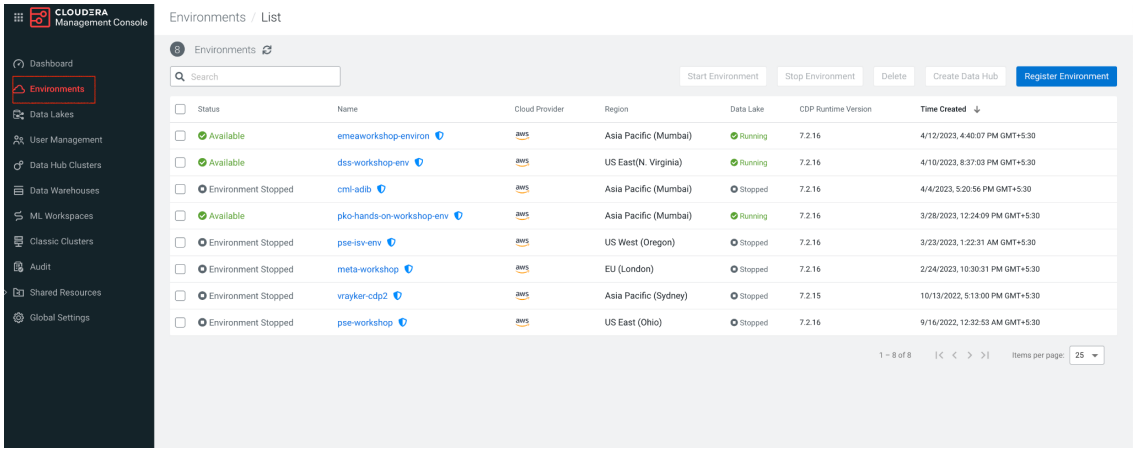
6. Unlock your KeyTab

To run queries on the SQL Stream Builder you need to have your KeyTab unlocked. This is mainly for authentication purposes. As the credential you are using is sometimes reused as part of other people doing the same lab it is possible that your Keytab is already unlocked. We have shared the steps for both the scenarios:

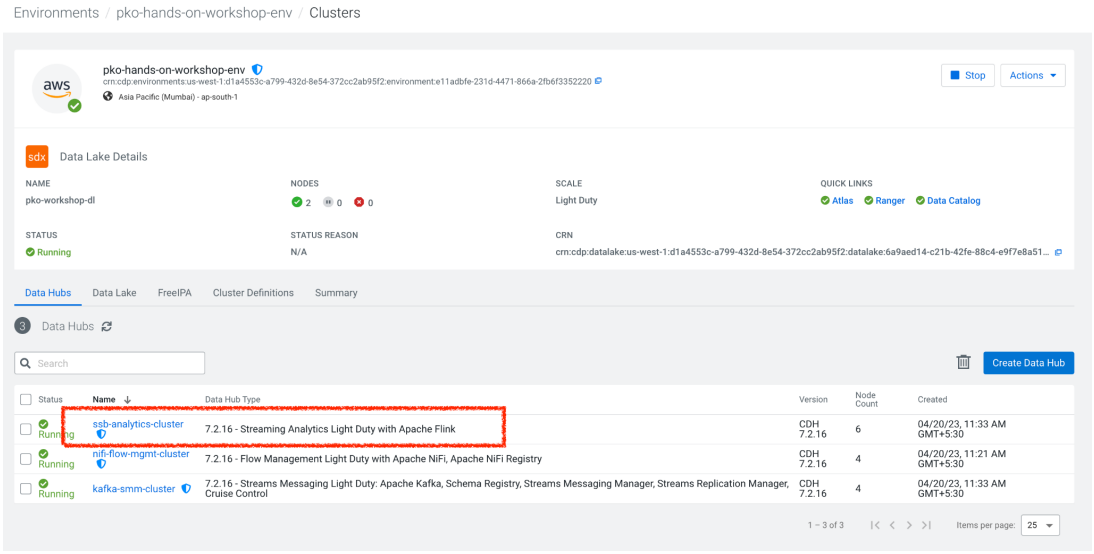
1. Unlock your Keytab if it is not unlocked already

Step 1 : Go to the SSB Data Hub

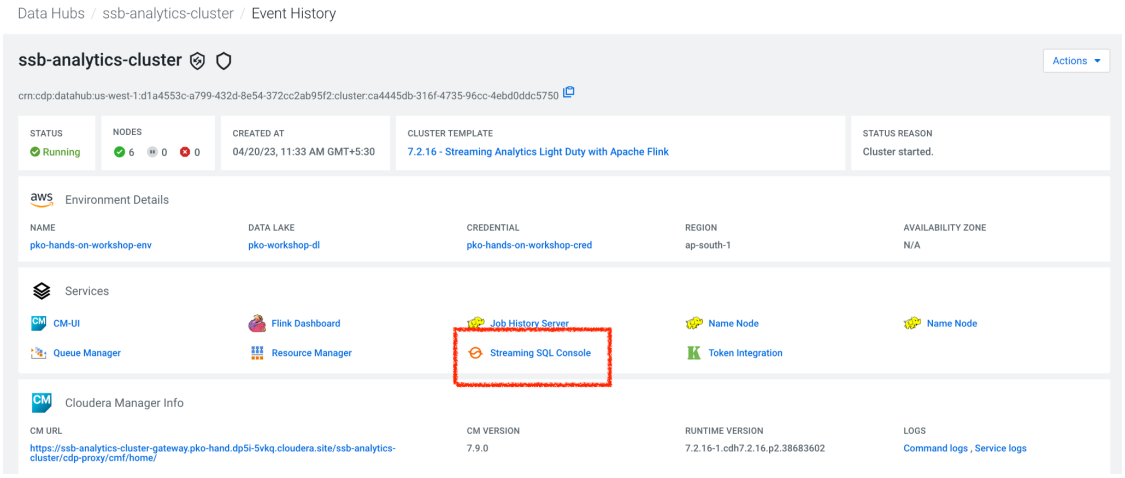
Click on Environments on the left tab and select the environment that is shared by the INSTRUCTOR



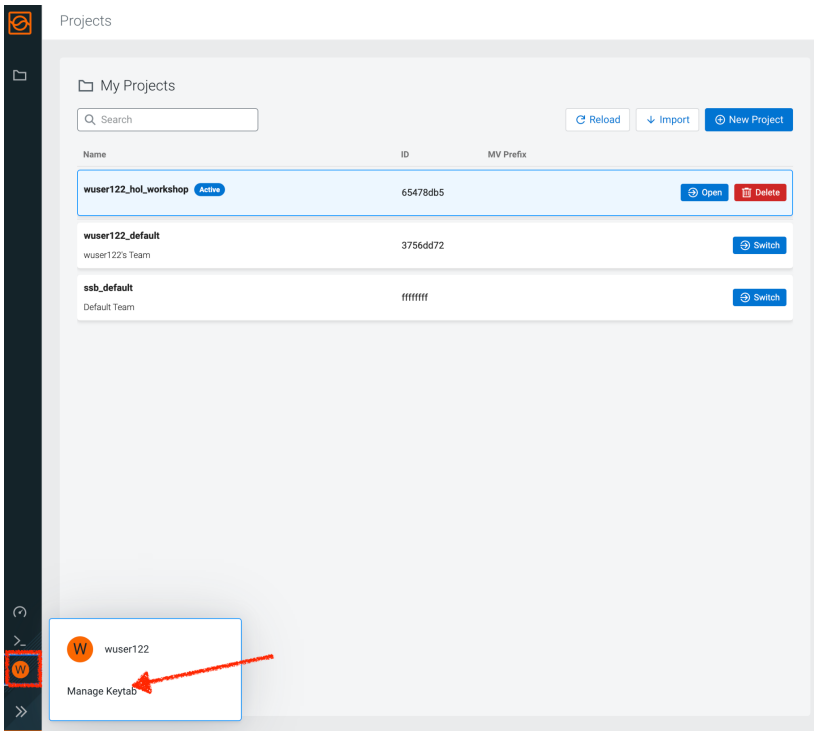
Click on the DataHub associated with SQL Stream Builder (ssb-analytics-cluster)



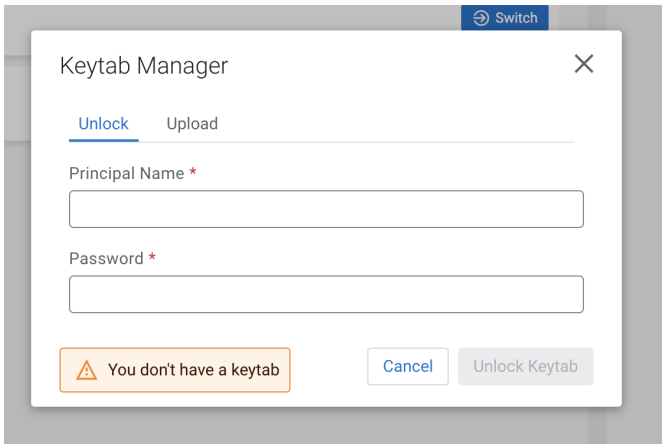
Step 2 : Open the SSB UI by clicking on **Streaming SQL Console**



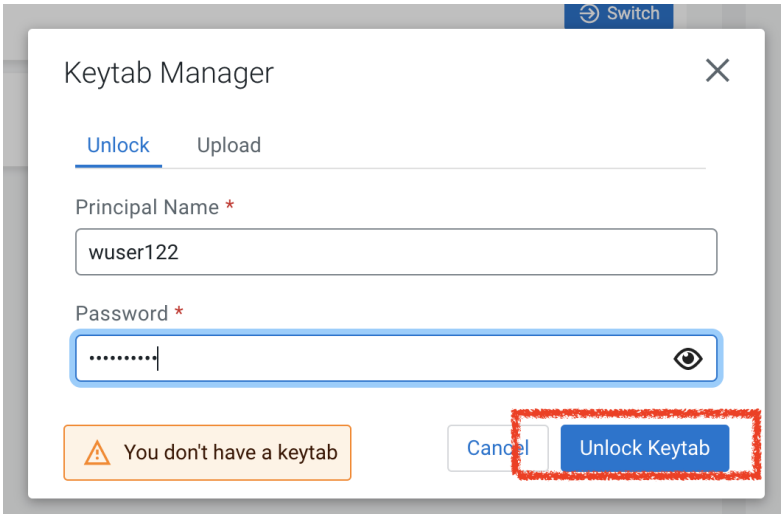
Step 3 : Click on the User name at the bottom left of the screen and select Manage Keytab

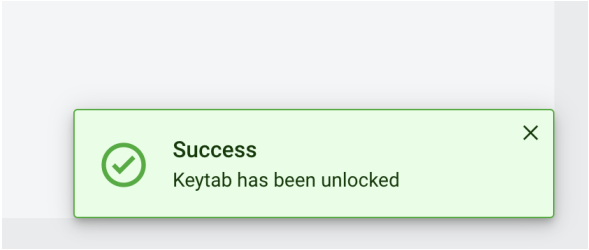


Step 4 : Enter your Workload Username (apacXY) and Password.



Step 5 : Click on unlock KeyTab

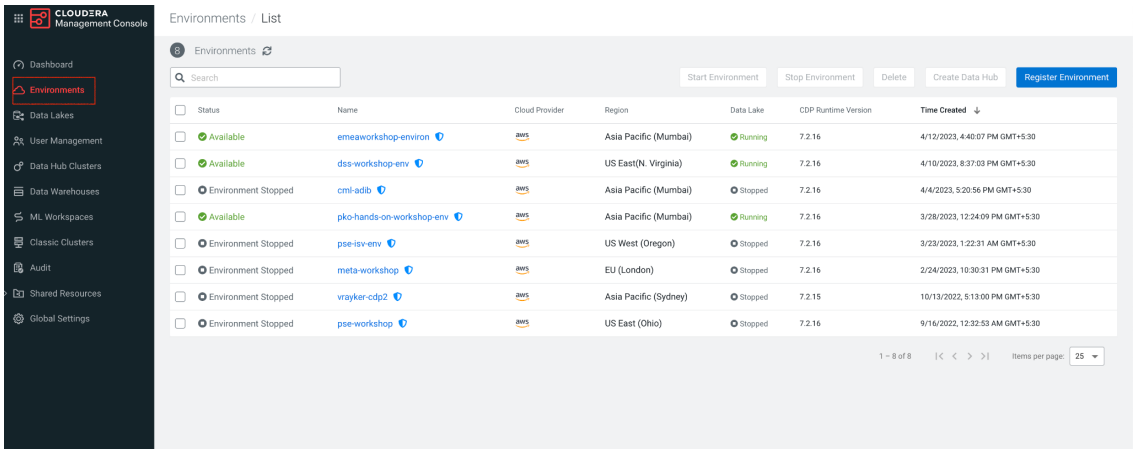




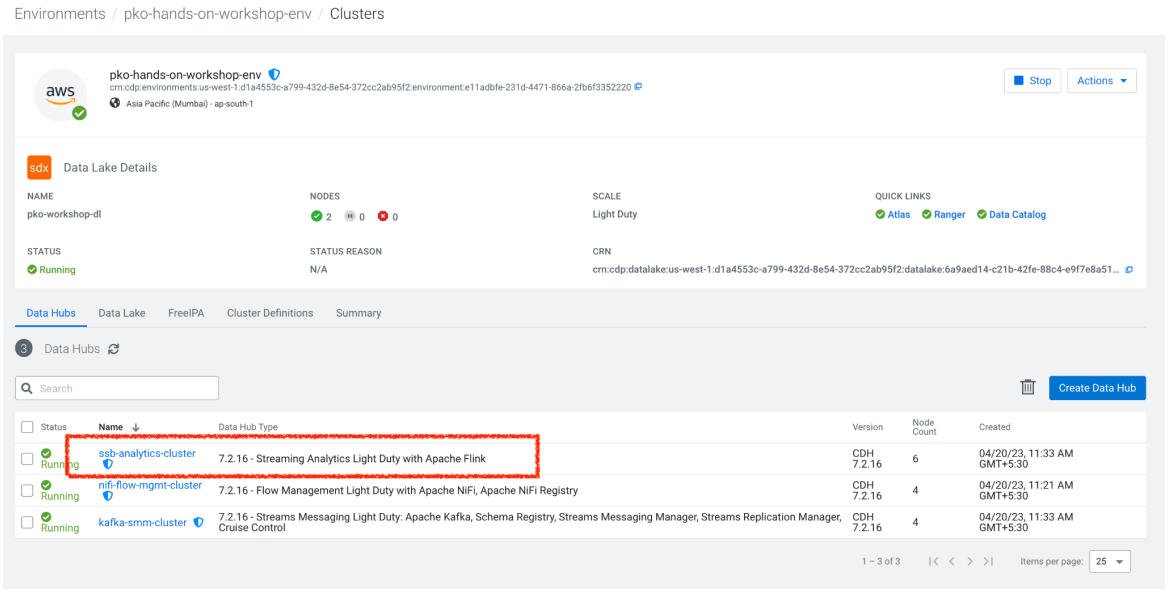
2. Reset your KeyTab if it is already unlocked

Step 1 : Go to the SSB Data Hub

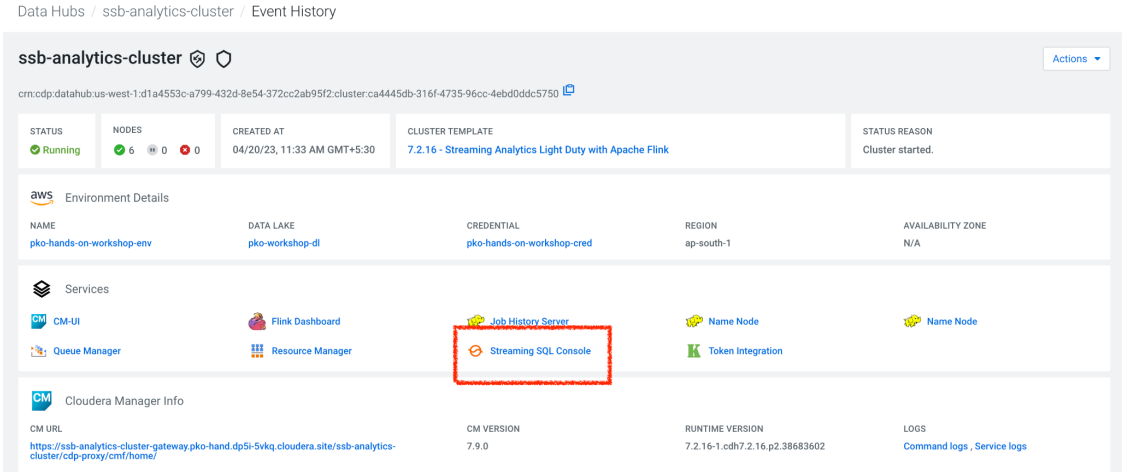
Click on Environments on the left tab and select the environment that is shared by the INSTRUCTOR



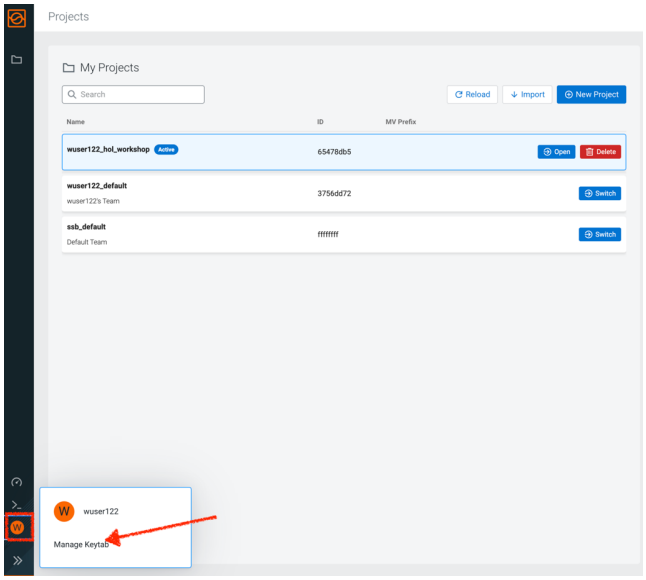
Click on the DataHub associated with SQL Stream Builder (ssb-analytics-cluster)



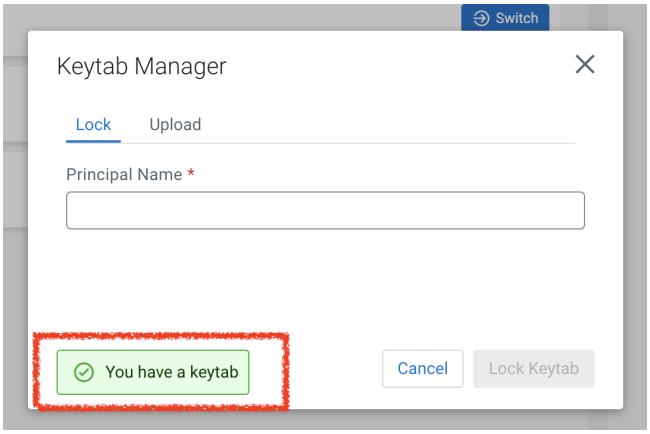
Step 2 : Open the SSB UI by clicking on **Streaming SQL Console**



Step 3 : Click on the User name at the bottom left of the screen and select Manage Keytab



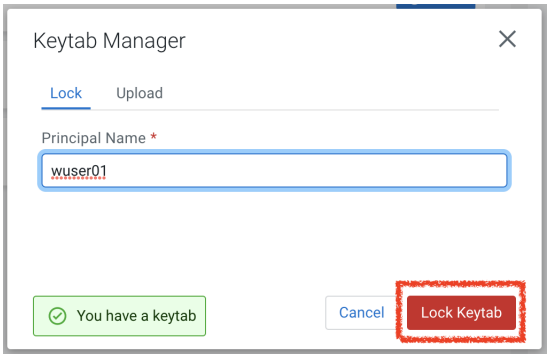
If you get the following dialog box it means that your Keytab is already unlocked. **But it would be necessary to reset here by locking it and unlocking it again using your newly set workload password**



Step 3 : Enter your Principal Name which is the same as your workload username

Example : apacXY

Click on Lock KeyTab



You can now continue from the STEP 3 in the "***Unlock your KeyTab if not unlocked already***" section above

Lab 1 : Create a Flow using the Flow Designer

---

1. Overview

Creating a data flow for CDF-PC is the same process as creating any data flow within Nifi with 3 very important steps:

- The data flow that would be used for CDF-PC must be self contained within a process group
- Data flows for CDF-PC must use parameters for any property on a processor that is modifiable, e.g. user names, Kafka topics, etc.
- All queues need to have meaningful names (instead of Success, Fail, and Retry). These names will be used to define Key Performance Indicators in CDF-PC.

The following is a step by step guide in building a data flow for use within CDF-PC.

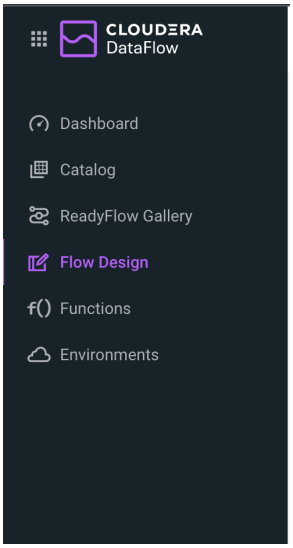
2. Building the Data Flow

2.1. Create the canvas to design your flow

Step 1: Access the DataFlow Data Service

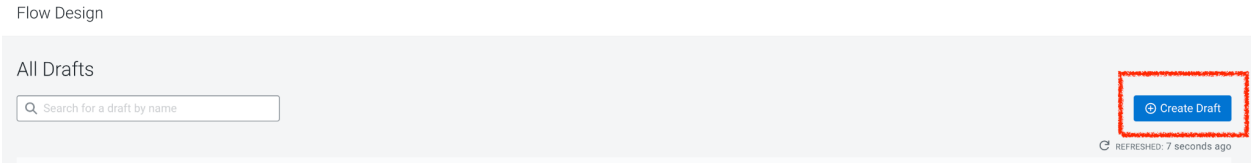
Access the DataFlow dataservice from the Management Console

Step 2: Go to the Flow Design



Step 3: Create a new Draft

(This will be the main process group of your flow)

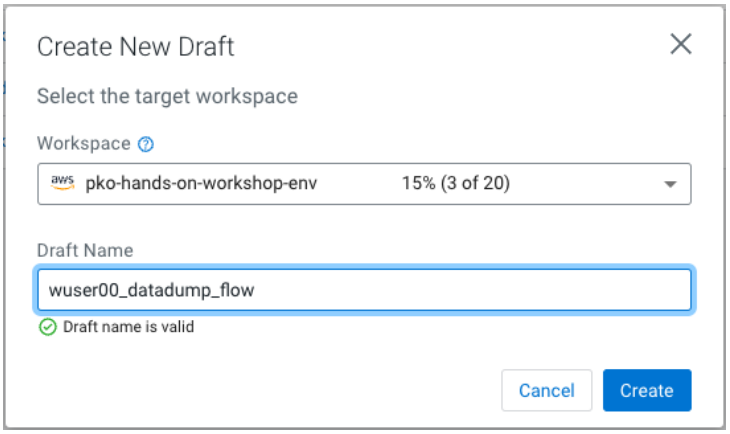


Step 4: Select the appropriate environment

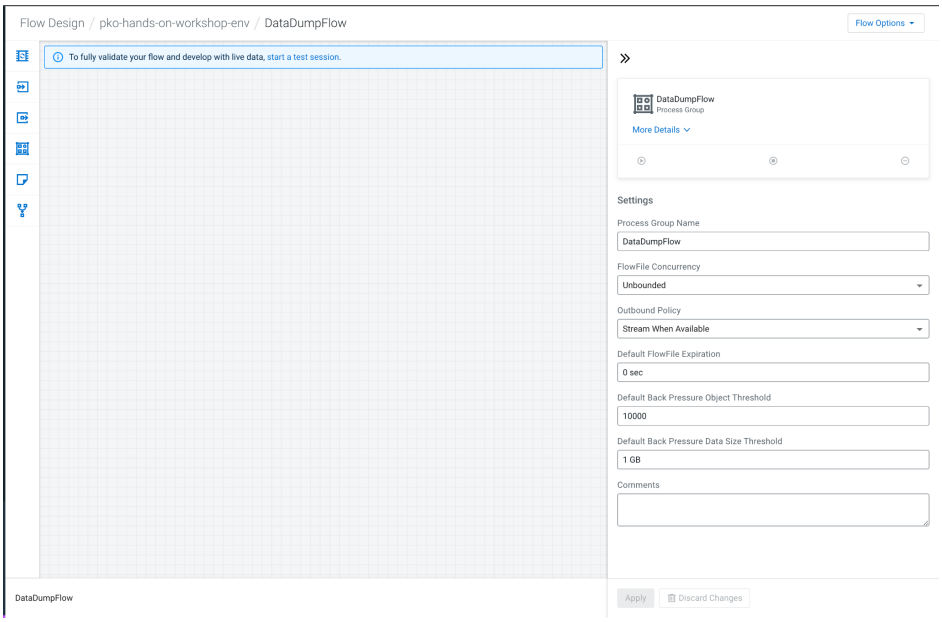
Select the appropriate environment as part of the workspace and give your flow a name and click on **CREATE**

Workspace Name : ***The name of the environment will be shared by the INSTRUCTOR***

Draft Name : {user_id}_datadump_flow
Example : apacXY_datadump_flow

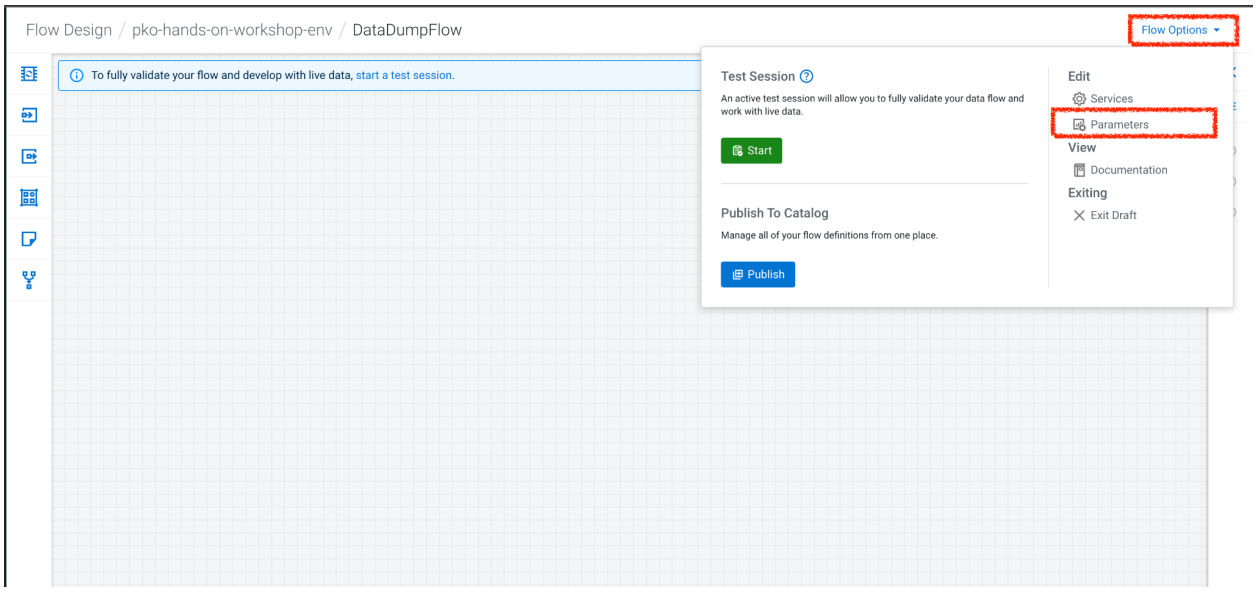


On successful creation of the Draft, you should now be redirected to the canvas on which you can design your flow



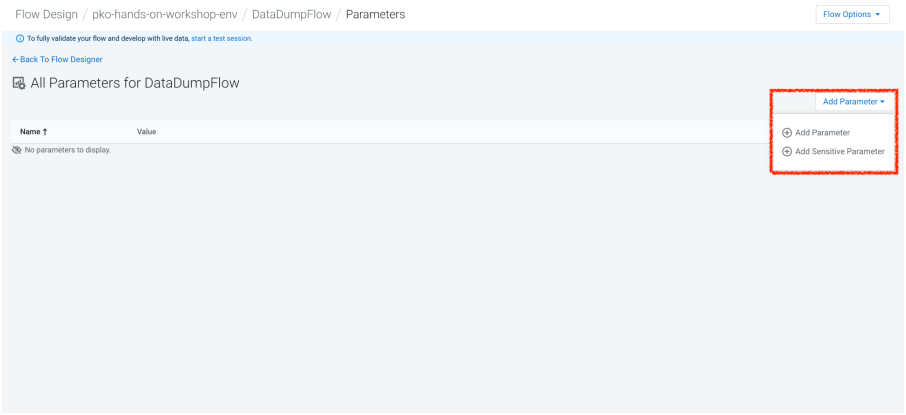
2.2. Adding new parameters

Step 1: Click on the **FLOW OPTIONS** on the top right corner of your canvas and then select **PARAMETERS**



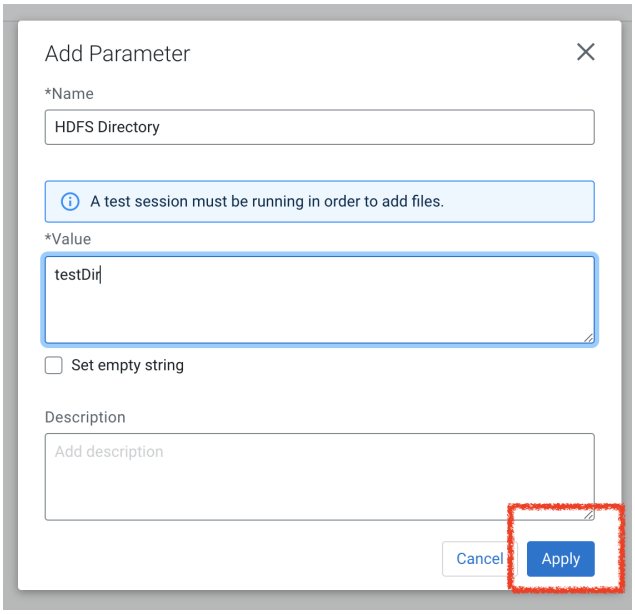
Step 2: Configure Parameters

The next step is to configure what is called a parameter. These parameters are reused within the flow multiple times and will also be configurable at the time of deployment. Click on **ADD PARAMETER** to add non sensitive values, for any sensitive parameter please select **ADD SENSITIVE PARAMETER.**

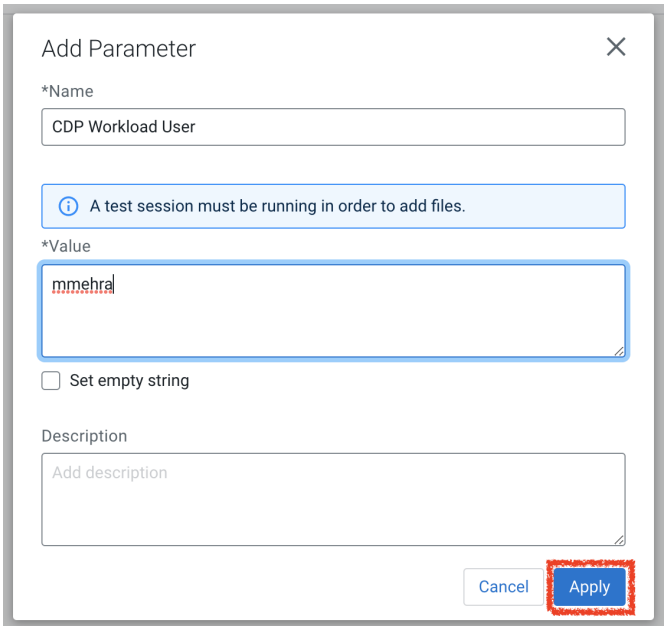


We need to add the following parameters.

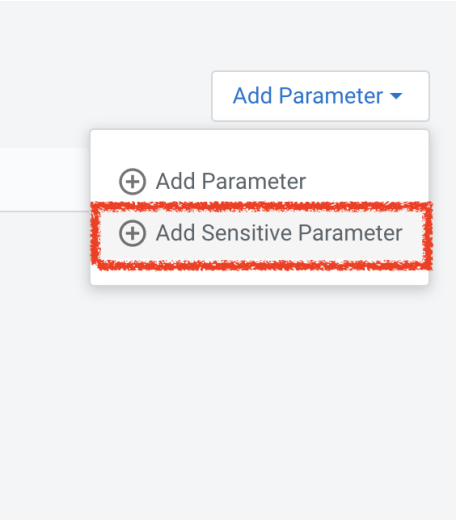
- S3 Directory
    - Selection under Add Parameter : ***Add Parameter***
    - Name : S3 Directory
    - Value : LabData or TestDir

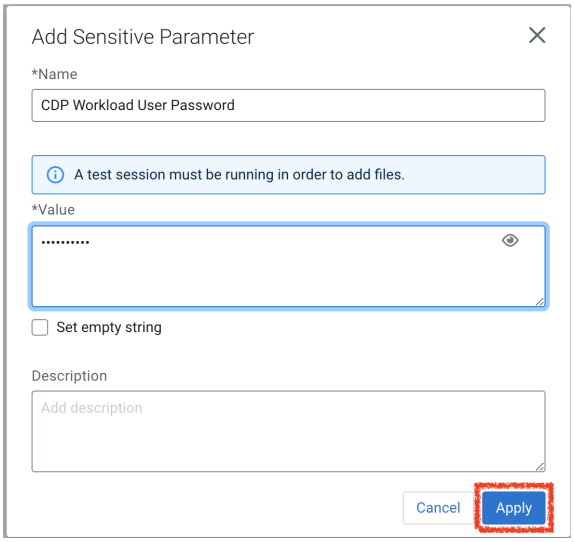


- CDP Workload User
    - Selection under Add Parameter : ***Add Parameter***
    - Name : CDP Workload User
    - Value : <The username assigned to you>
        - EXAMPLE : apac01
        - IMPORTANT: do not add the domain '@workload.com'

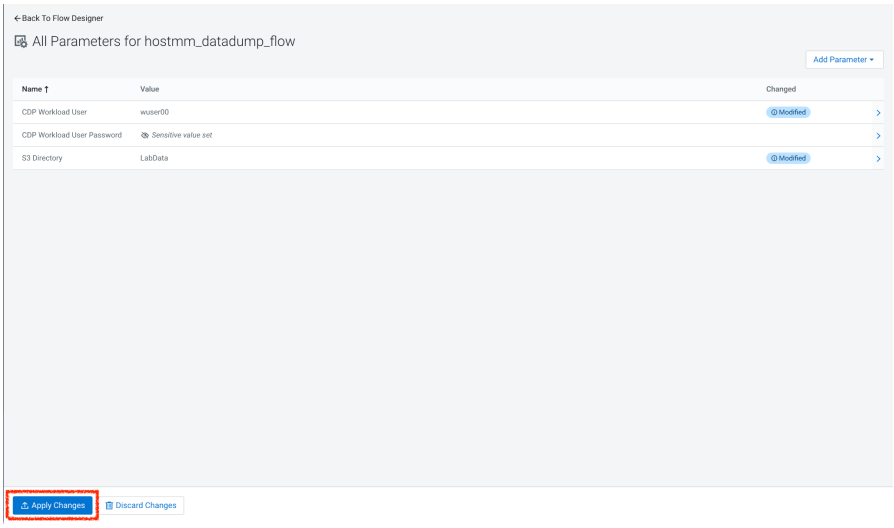


- CDP Workload User Password - [Sensitive Field]
    - Selection under Add Parameter : ***Add Sensitive Parameter***
    - Name : CDP Workload User Password
    - Value : <Workload Password set by yourself in Lab 0>
        - EXAMPLE : ApacLabs@23

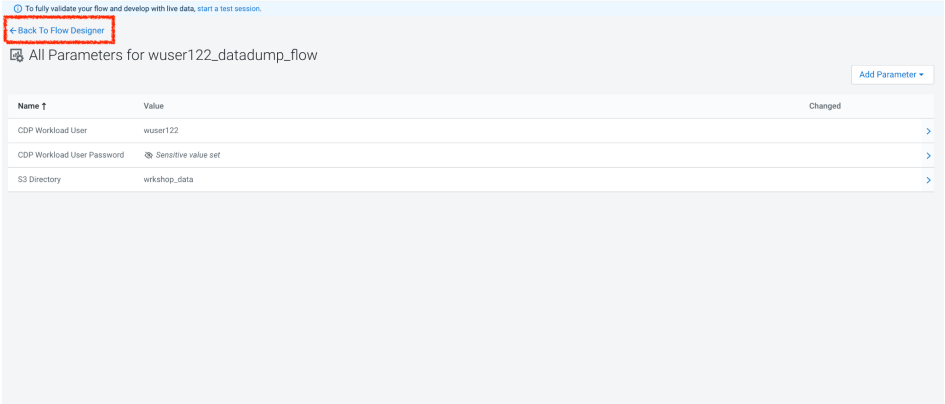




Click **APPLY CHANGES**



Now go back to the Flow Designer. Click *'Back to Flow Designer'*



Now that we have created these parameters, we can easily search and reuse them within our dataflow. This is especially useful for **CDP Workload User** and **CDP Workload User Password**.

NOTE ONLY:

To search for existing parameters:

1. Open a processor's configuration and proceed to the properties tab.
2. Enter: **#{**
3. Hit 'control+spacebar'

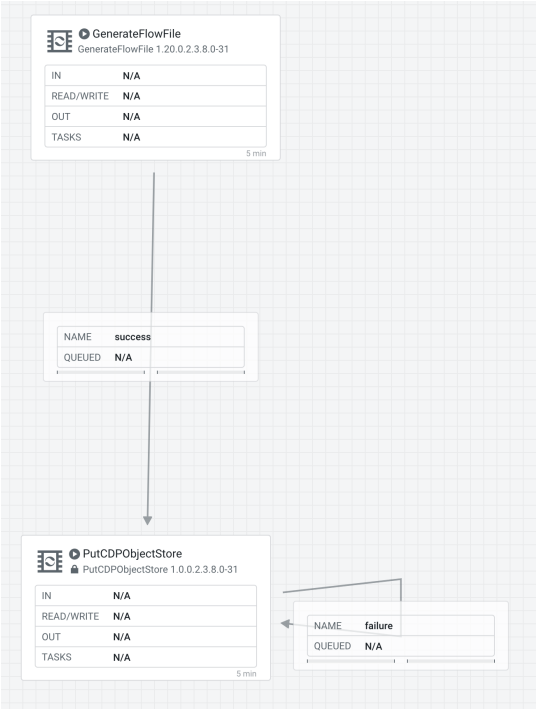
This will bring up a list of existing parameters that are not tagged as sensitive.

2.3. Create the Flow

Let's go back to the canvas to start designing our flow. This flow will contain 2 Processors:

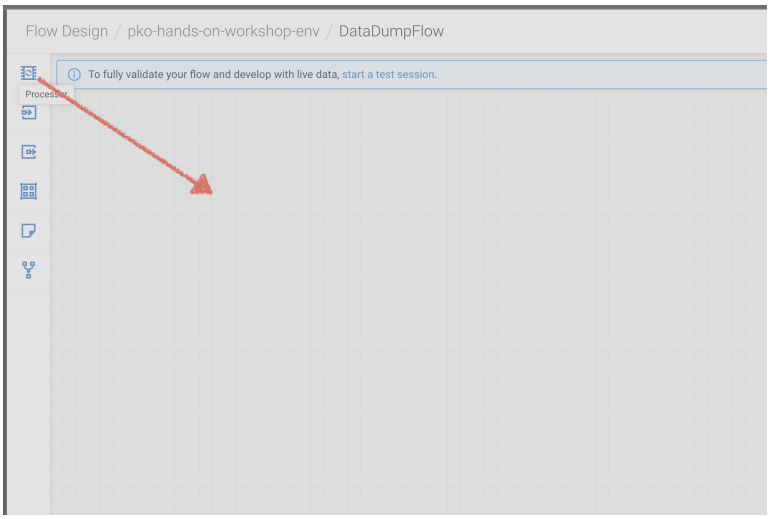
- **GenerateFlowFile** - Generates random data
- **PutCDPObjectStore** - Loads data into HDFS(S3)

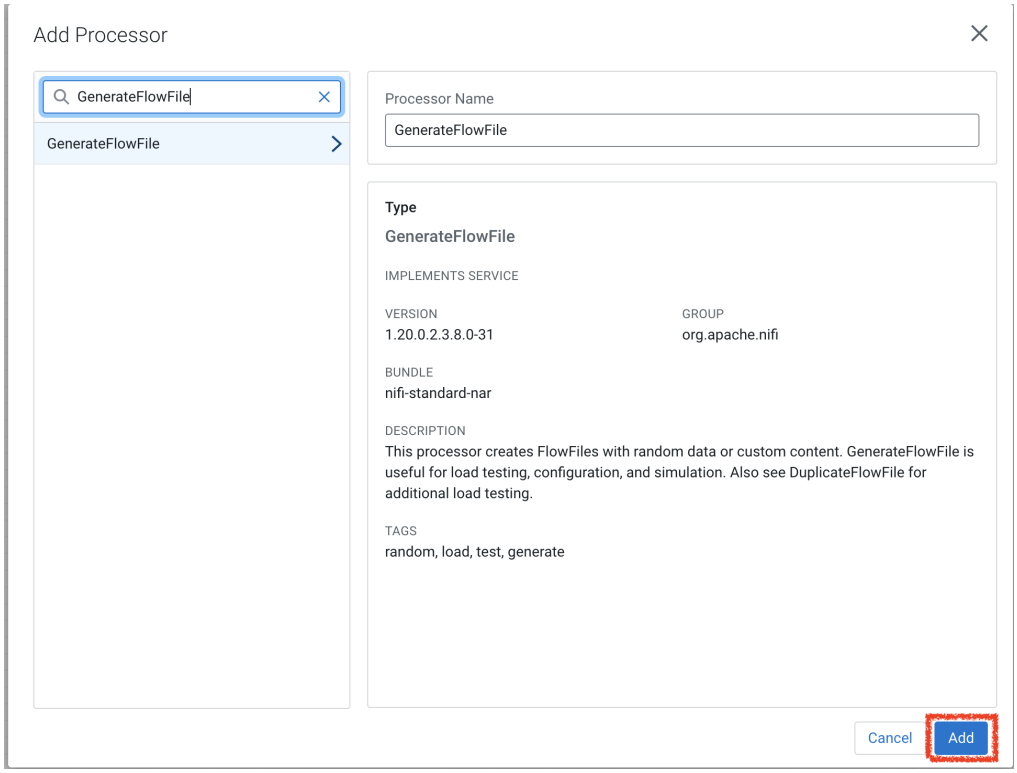
Our final flow will look like this:



Step 1: Add **GenerateFlowFile** processor

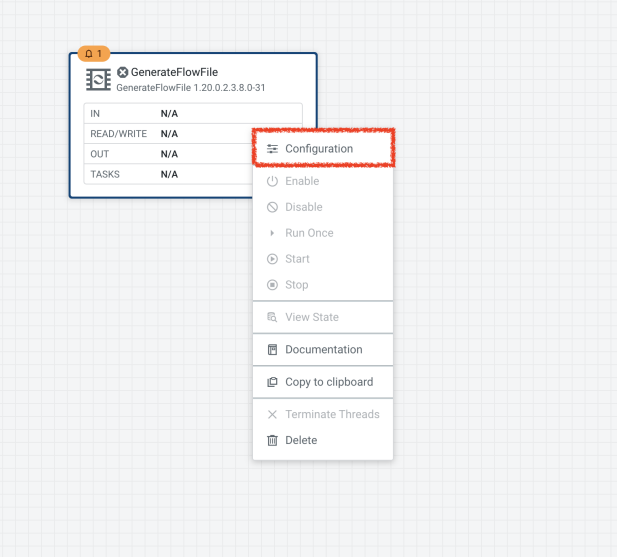
Pull the Processor onto the canvas and select **GenerateFlowFile** Processor and click on **ADD.**





Step 2: Configure **GenerateFlowFile** processor

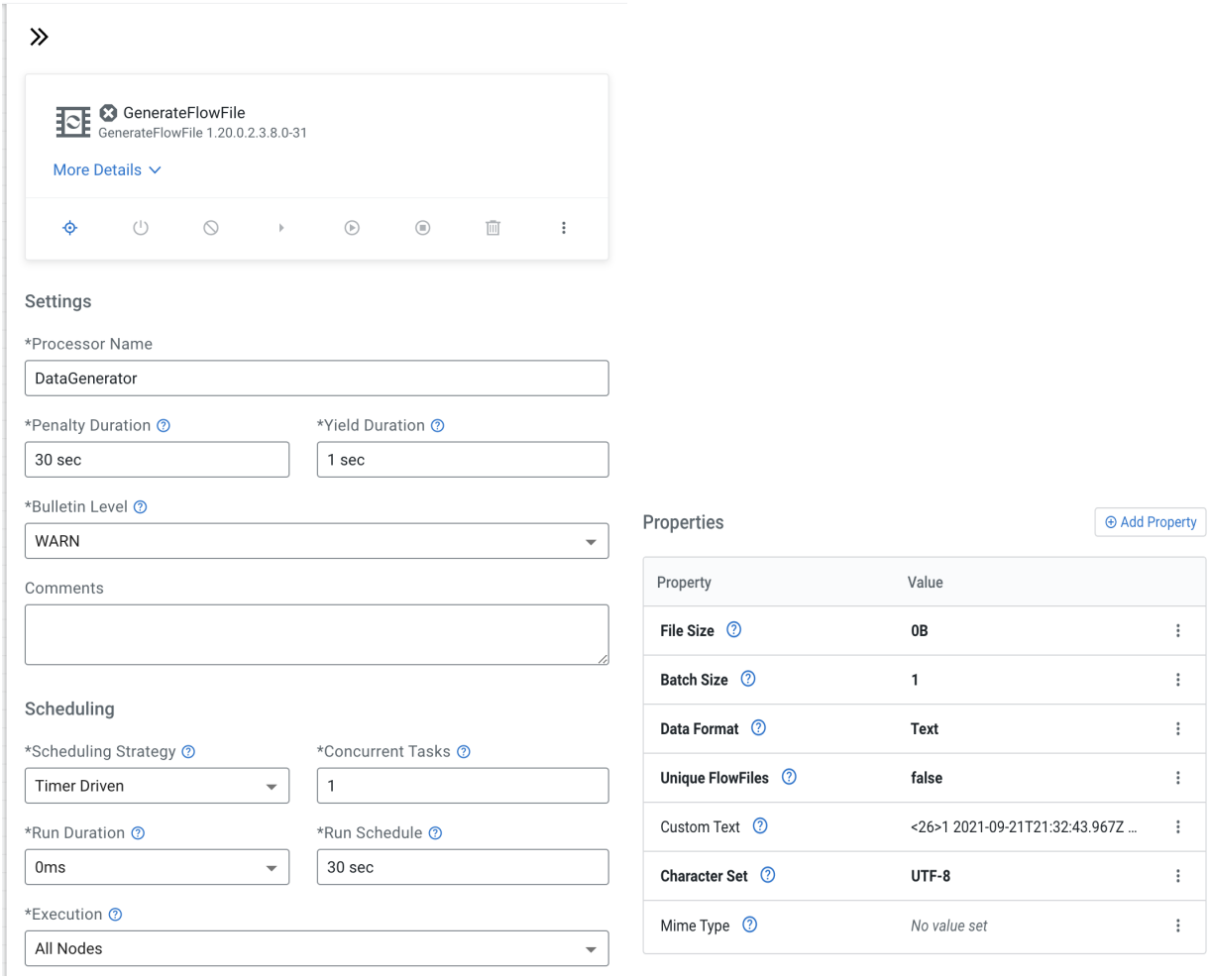
The GenerateFlowFile Processor will now be on your canvas and you can configure it in the following way by right clicking and selecting **Configuration**.



Configure the processor in the following way

Property	Value
Processor Name	DataGenerator
Scheduling Strategy(default)	Timer Driven
Run Duration(default)	0 ms
Run Schedule	30 sec
Execution(default)	All Nodes
Custom Text	<26>1 2021-09-21T21:32:43.967Z host1.example.com application4 3064 ID42 [exampleSDID@873 iut="4" eventSource="application" eventId="58"] application4 has stopped unexpectedly

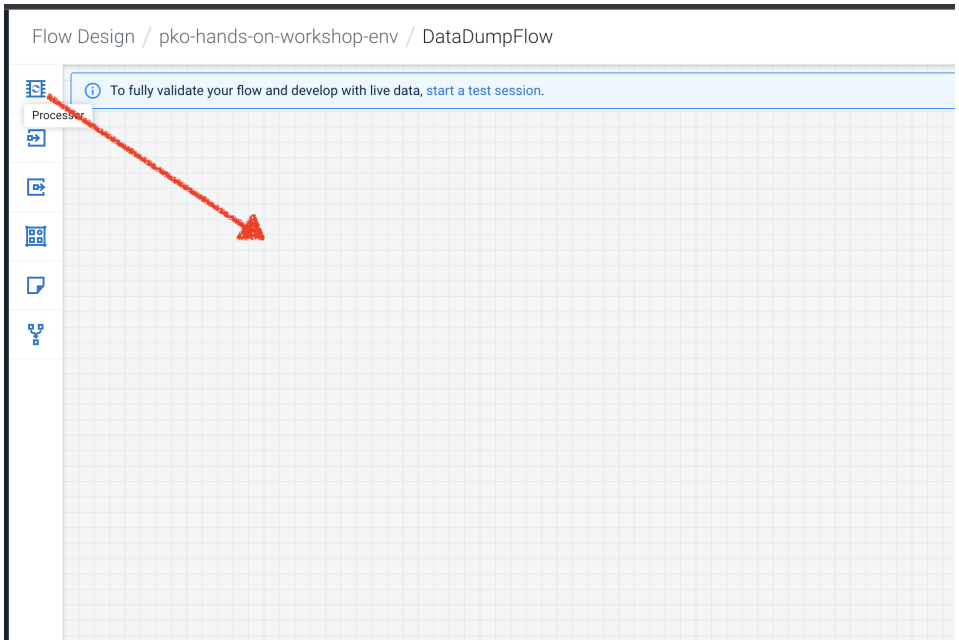
This represents a syslog out in RFC5424 format. Subsequent portions of this workshop will leverage this same syslog format.

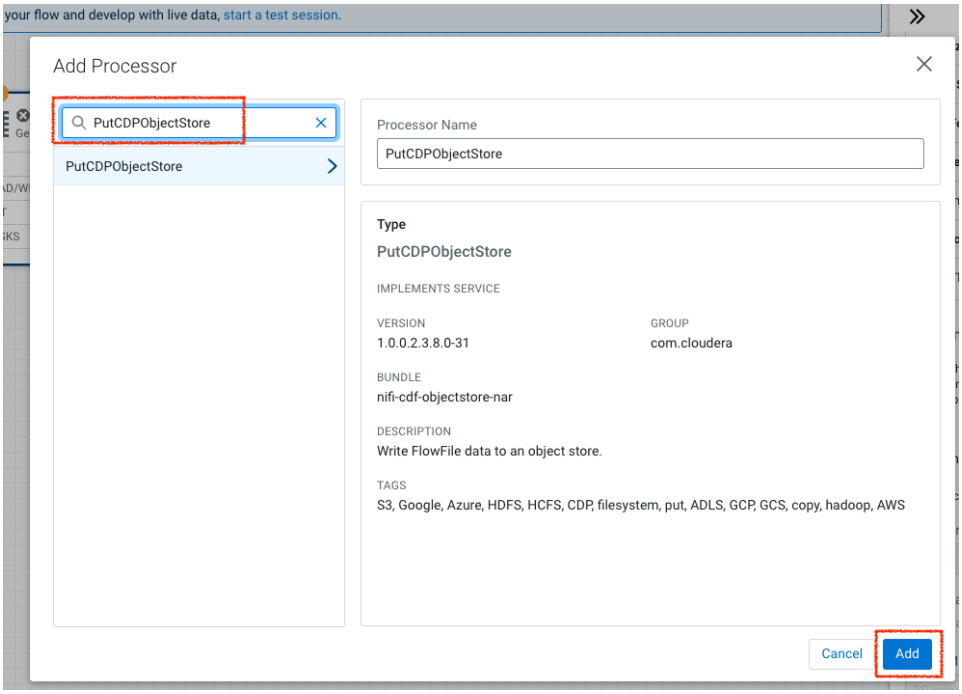


Click on **APPLY.**

Step 3: Add **PutCDPObjectStore** processor

Pull a new Processor onto the canvas and select **PutCDPObjectStore** Processor and click on **ADD.**





Step 4: Configure **PutCDPObjectStore** processor

The PutCDPObjectStore Processor needs to be configured as follows:

Property	Value
Processor Name	Move2S3
Scheduling Strategy(default)	Timer Driven
Run Duration(default)	0 ms
Run Schedule(default)	0 sec
Execution(default)	All Nodes
Directory	#{S3 Directory}
CDP Username	#{CDP Workload User}
CDP Password	#{CDP Workload User Password}
Auto Terminate Relationships:	Check the “Terminate” box under “success”

PutCDPObjectStore
PutCDPObjectStore 1.0.0.2.3.8.0-31

More Details

Settings

*Processor Name: Move2S3

*Penalty Duration: 30 sec

*Yield Duration: 1 sec

*Bulletin Level: WARN

Comments

Scheduling

*Scheduling Strategy: Timer Driven

*Concurrent Tasks: 1

*Run Duration: 0ms

*Run Schedule: 0 sec

*Execution: All Nodes

Properties

Add Property

Property	Value
Storage Location	*No value set*
Directory	**#{HDFS Directory}**
Conflict Resolution Strategy	**fail**
Kerberos Credentials Service	*No value set*
CDP Username	#{CDP Workload User}
CDP Password	#{CDP Workload User Password}
Writing Strategy	**Simple write**
cdp.configuration.resources	#{CDPEnvironment}

Relationships

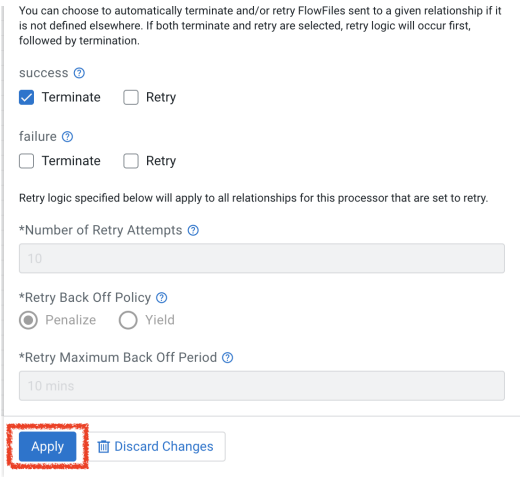
You can choose to automatically terminate and/or retry FlowFiles sent to a given relationship if it is not defined elsewhere. If both terminate and retry are selected, retry logic will occur first, followed by termination.

success: Terminate, Retry

failure: Terminate, Retry

Retry logic specified below will apply to all relationships for this processor that are set to retry.

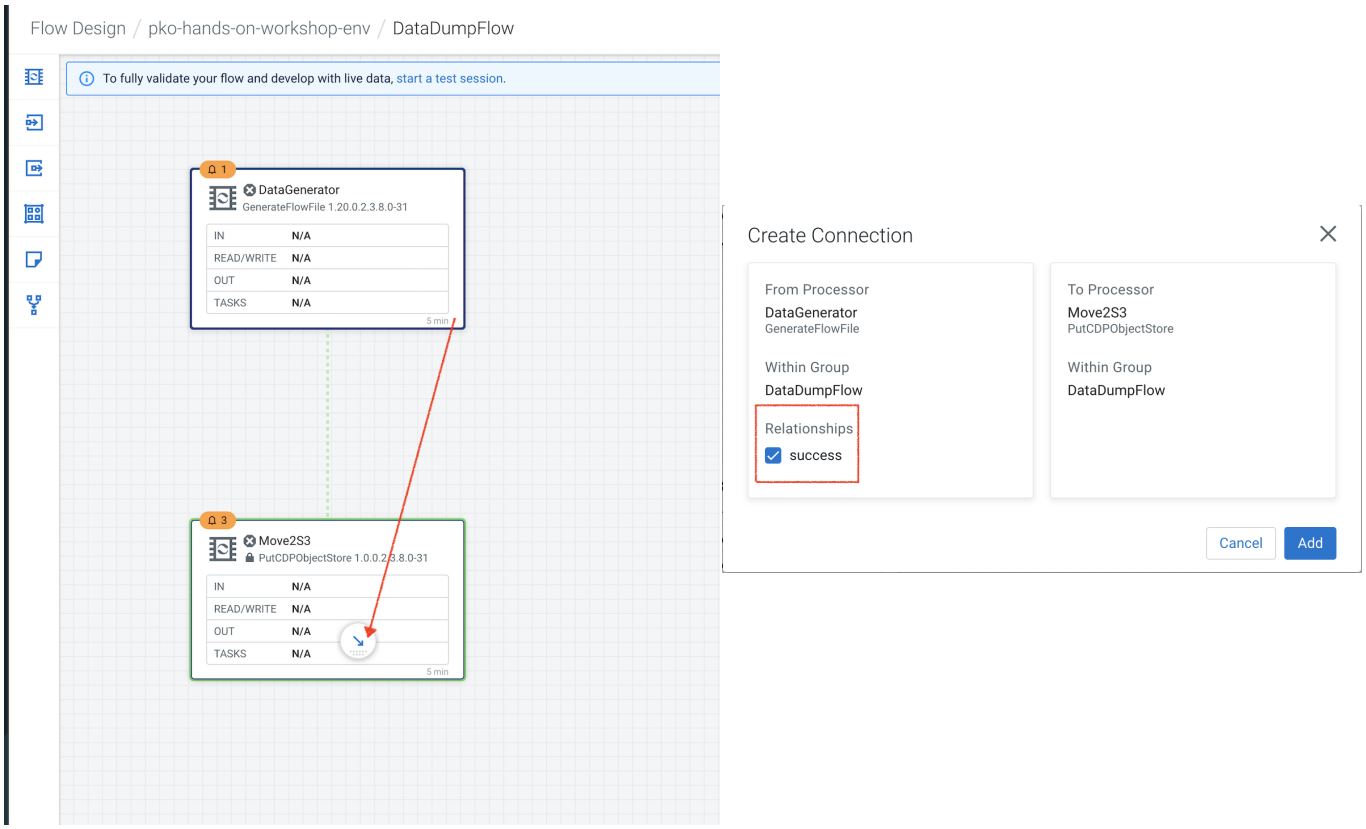
*Number of Retry Attempts: 10



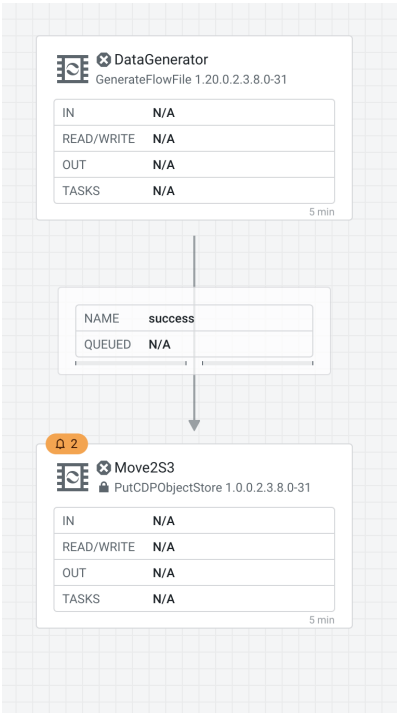
Click APPLY

Step 5: Create connection between processors

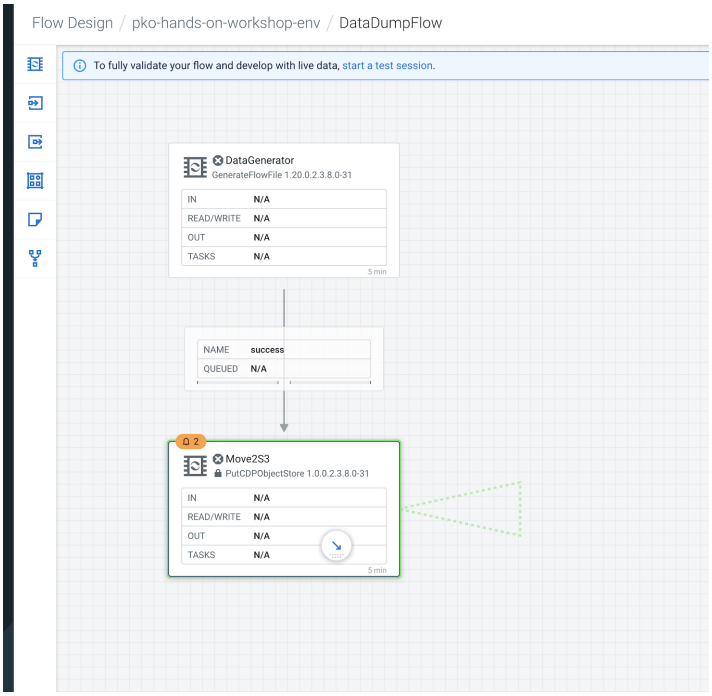
Connect the two processors by dragging the arrow from **DataGenerator** processor to the **Move2S3** processor and select on **SUCCESS** relation and click **ADD**

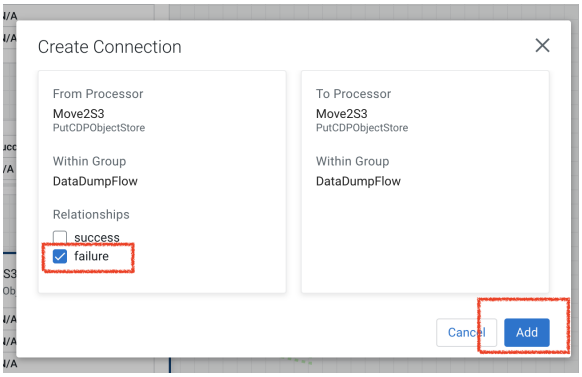


Your flow will now look something like this:



The Move2S3 processor does not know what to do in case of a failure, let's add a retry queue to it. This can be done by dragging the arrow on the processor outwards then back to itself, as below:





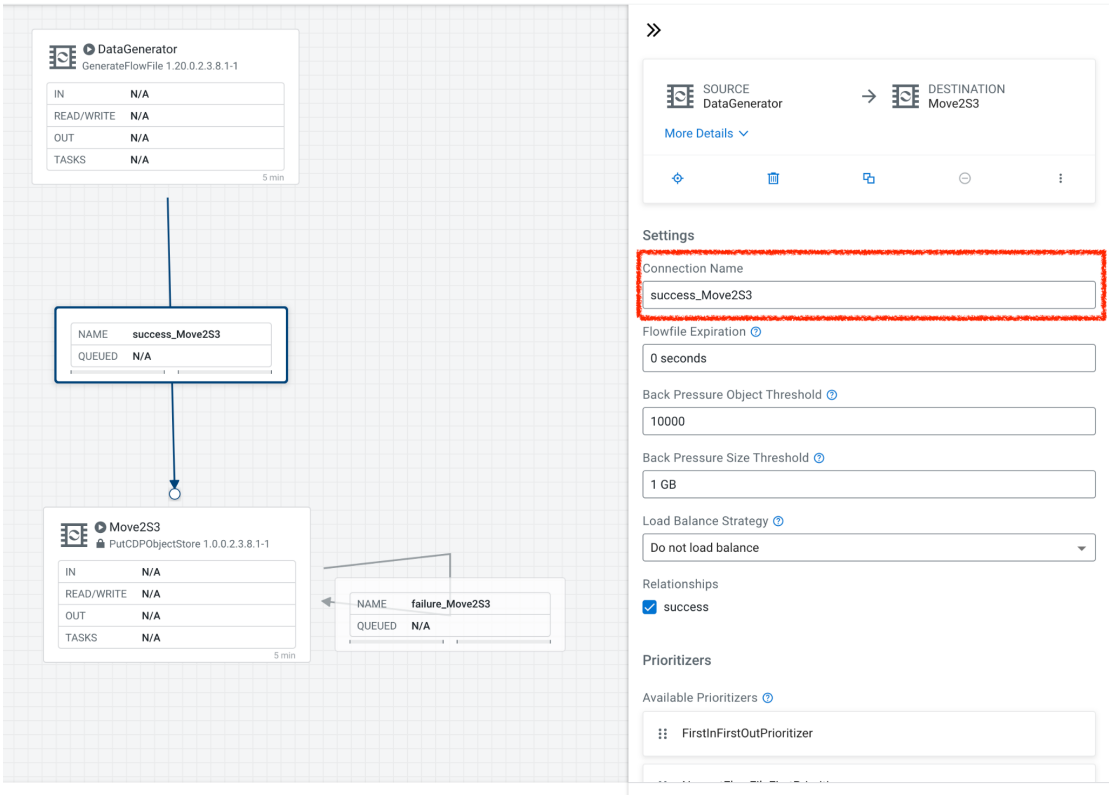

DataGenerator
GenerateFlowFile 1.20.0.2.3.8.0-31
IN N/A
READ/WRITE N/A
OUT N/A
TASKS N/A
5 min
NAME success
QUEUED N/A
1
Move2S3
PutCDPObjectStore 1.0.0.2.3.8.0-31
IN N/A
READ/WRITE N/A
OUT N/A
TASKS N/A
5 min
NAME failure
QUEUED N/A

2.4. Naming the queues

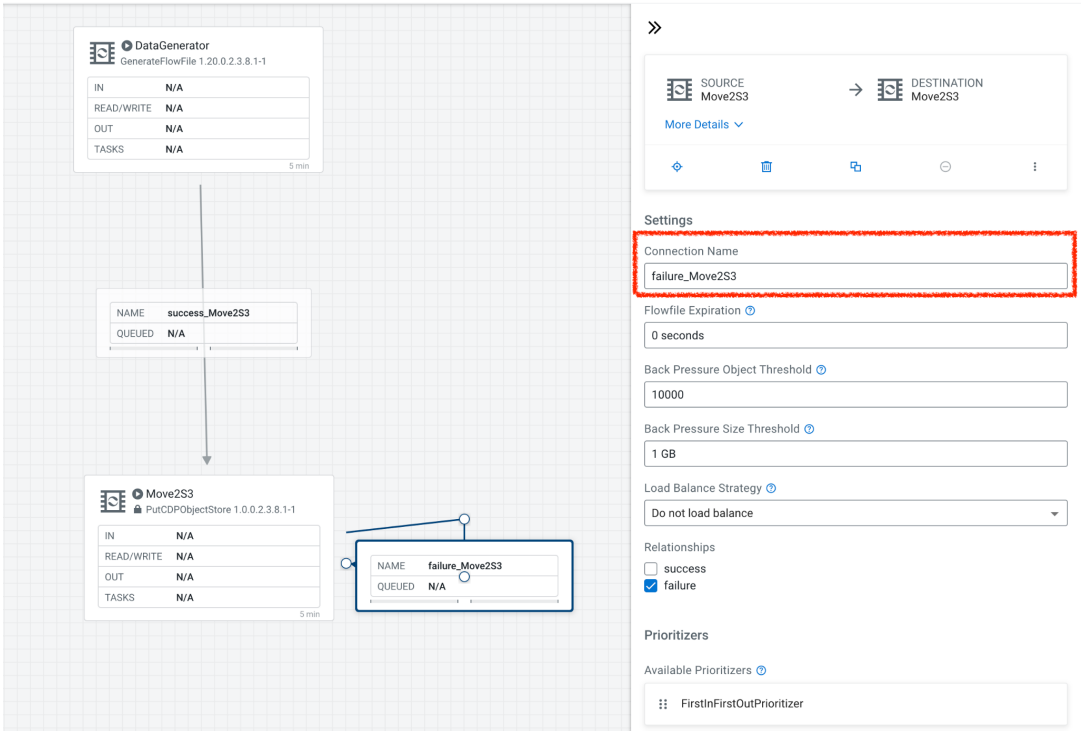
Providing unique names to all queues is very important as they are used to define Key Performance Indicators upon which CDF-PC will auto-scale.

To name a queue, double-click the queue and give it a unique name. A best practice here is to start the existing queue name (i.e. success, failure, retry, etc…) and add the source and destination processor information.

For example, the success queue between **DataGenerator** and **Move2S3** is named **success_Move2S3.**



The failure queue for **Move2S3** is named **failure_Move2S3**.

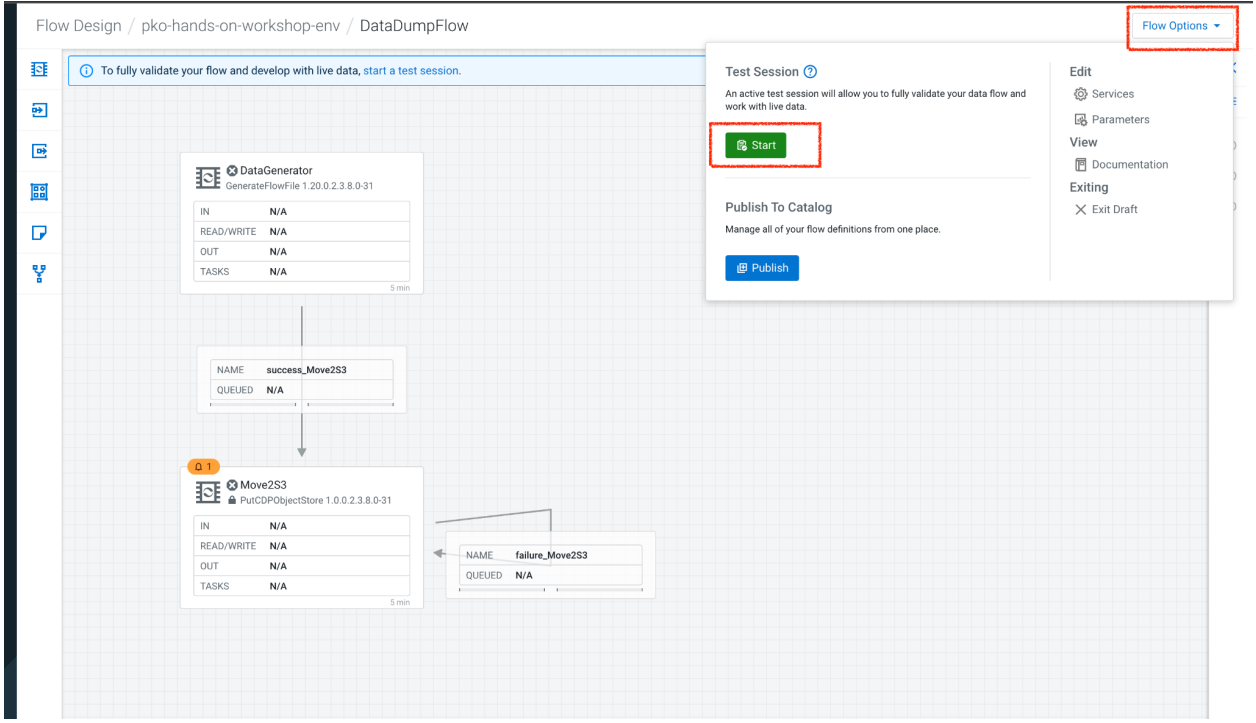


3. Testing the Data Flow

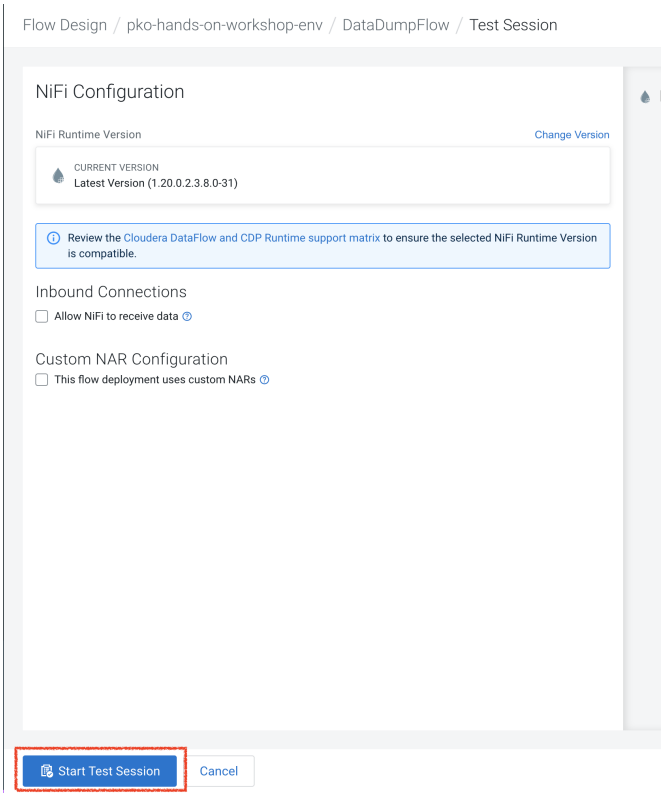
Step 1: Start test session

To test your flow we need to first start the test session

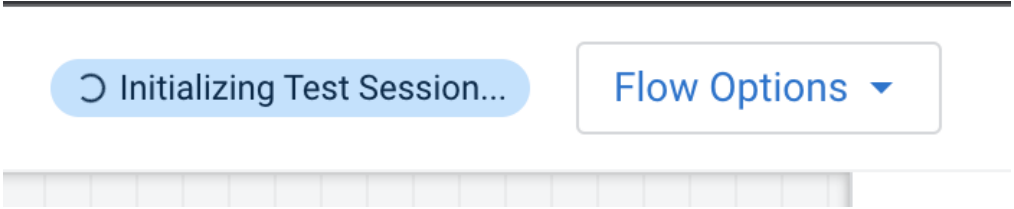
Click on **FLOW OPTIONS** and then select **START** on TEST SESSION



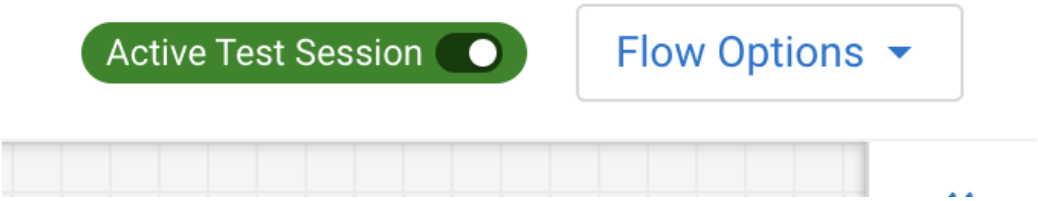
In the next window, click **START SESSION**



The activation should take about a couple of minutes. While this happens you will see this at the top right corner of your screen

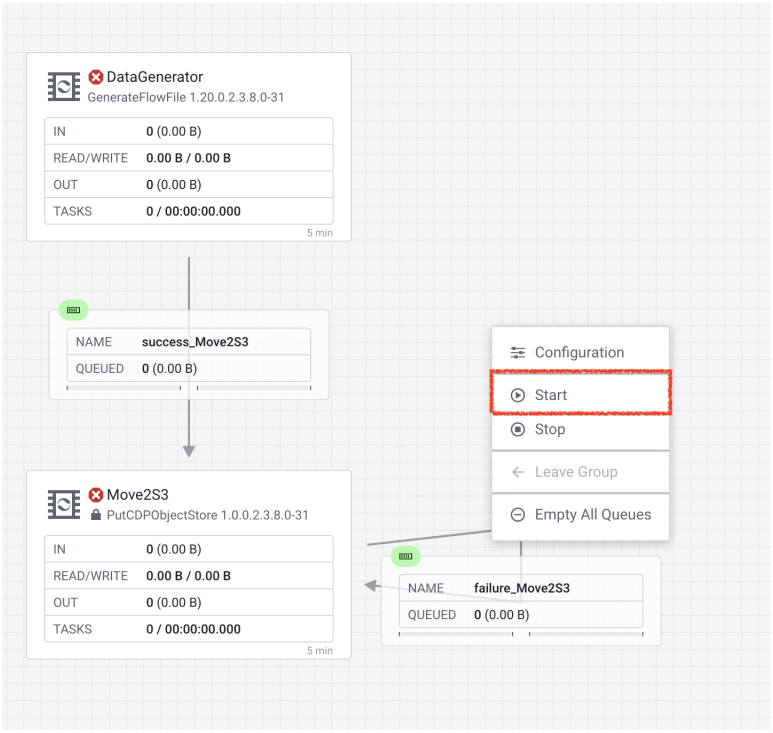


Once the Test Session is ready you will see the following message on the top right corner of your screen.

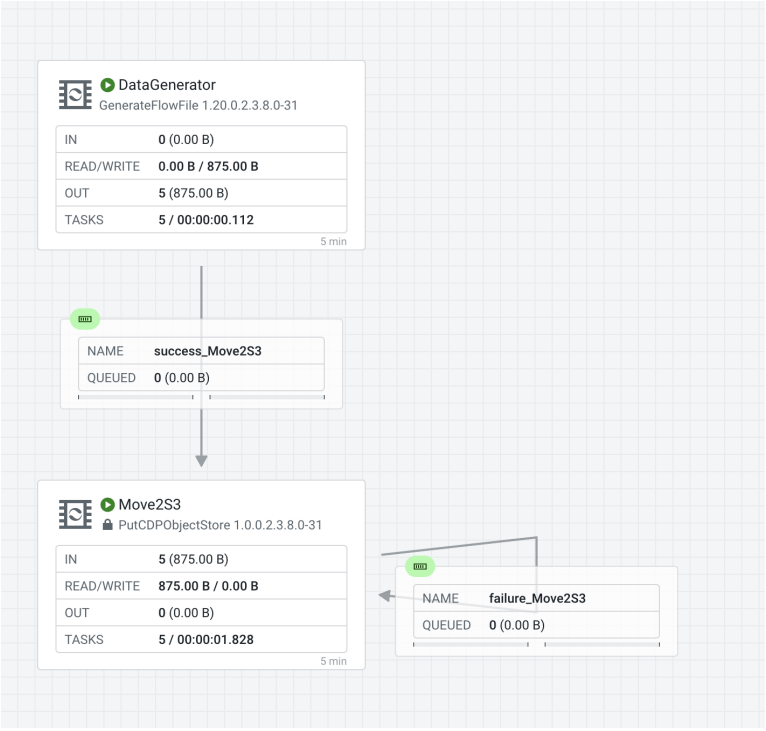


Step 2: Run the flow

Right click on the empty part of the canvas and select START.

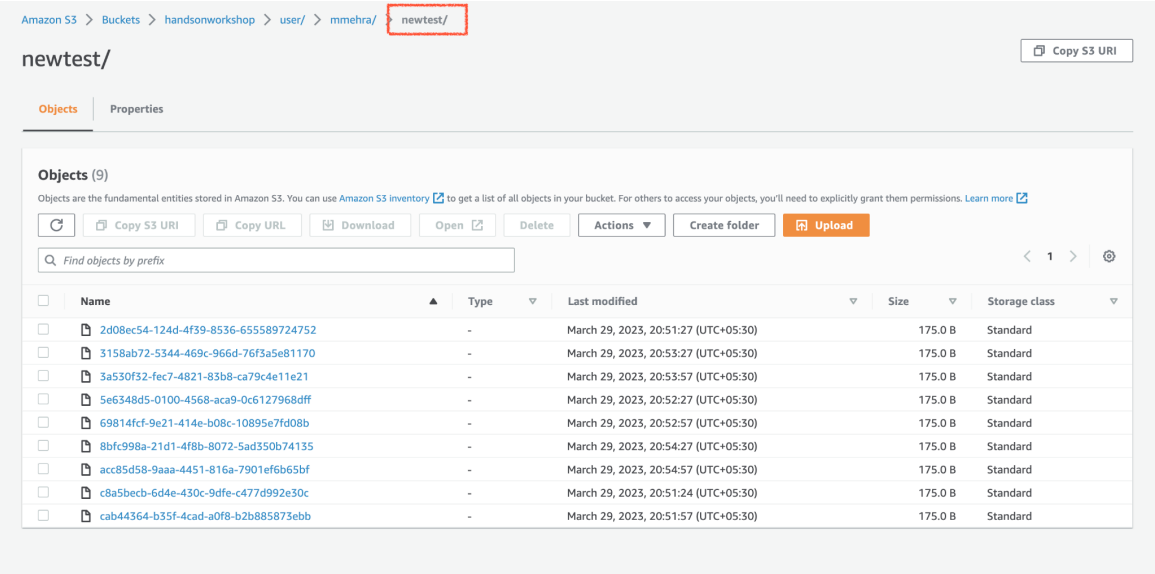


Both the processors should now be in the START state.



You will now see files coming into the folder which was specified as the Directory on the S3 bucket which is the Base data store for this environment.

Name ↑	Value
CDP Workload User	mmehra
CDP Workload User Password	*Sensitive value set*
CDPEnvironment	hive-site.xml, core-site.xml, ssl-client.xml
Default SSL Context Keystore	/home/nifi/additional/secret/ssl_keystore/ssl-keystore.jks
Default SSL Context Keystore Password	*Sensitive value set*
Default SSL Context Keystore Type	JKS
Default SSL Context Truststore	/home/nifi/additional/secret/ssl_truststore/ssl-truststore.jks
Default SSL Context Truststore Password	*Sensitive value set*
Default SSL Context Truststore Type	JKS
HDFS Directory	newtest



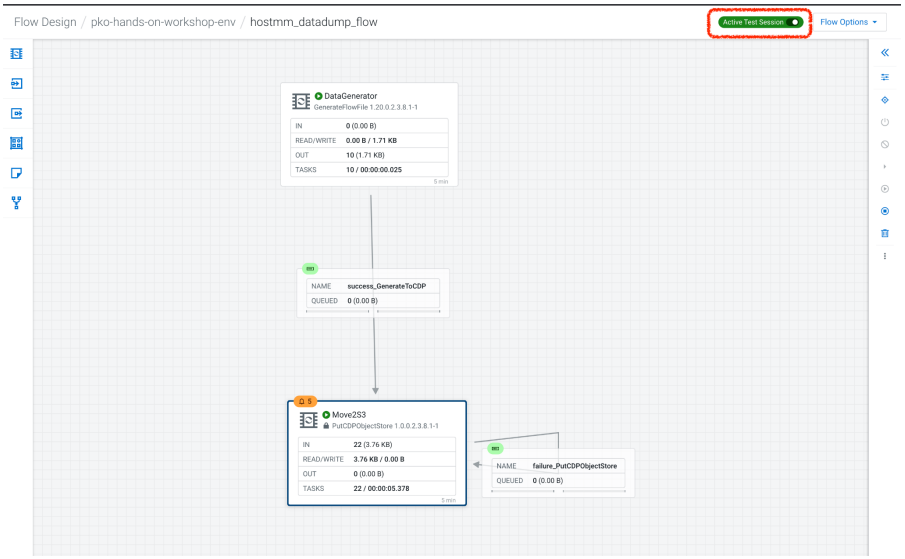
[You will not be able to access this S3 bucket by yourself but the instructor will show you where everyone's data is moving to]

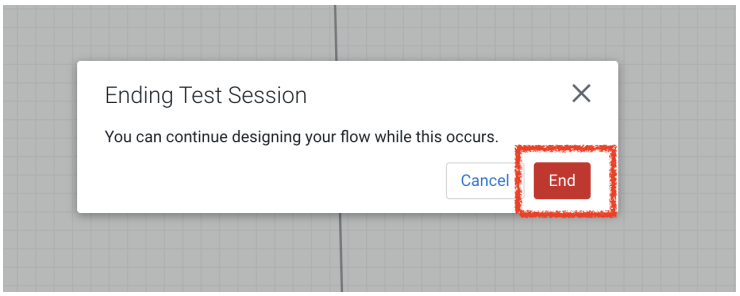
4. Move the Flow to the Flow Catalog

After the flow has been created and tested we can now PUBLISH the flow to the Flow Catalog

Step 1: STOP the current test session

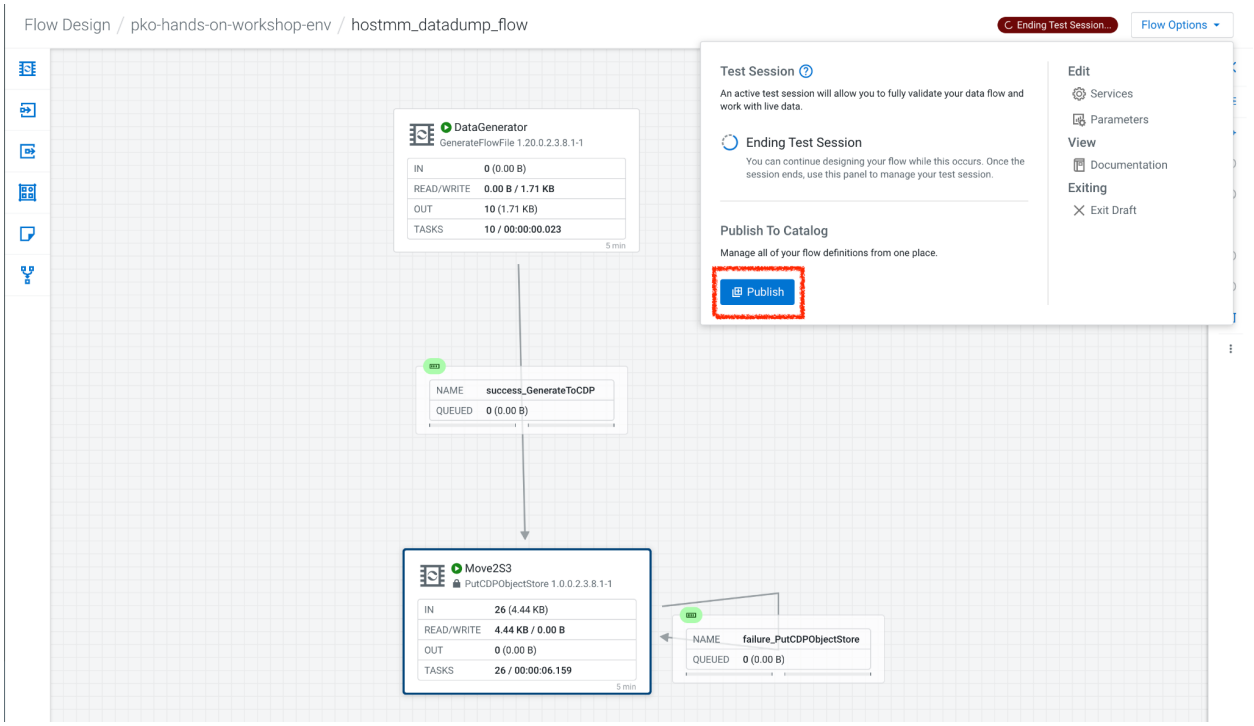
STOP the current test session by clicking on the green tab on top right and click **END**





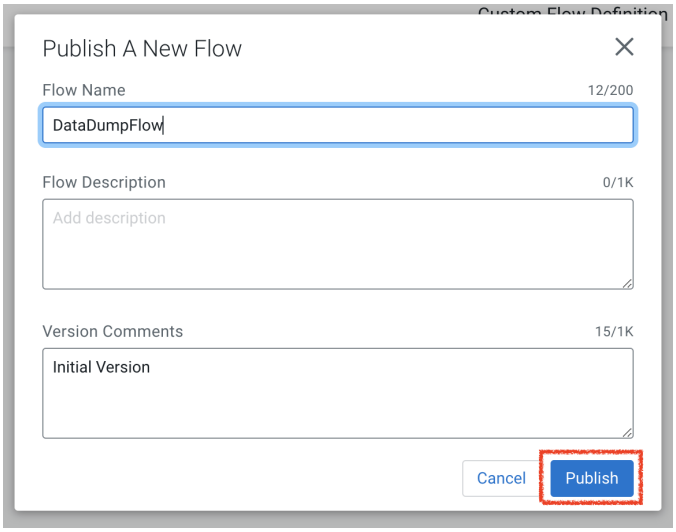
Step 2: PUBLISH the flow

Once the session stops, click on **FLOW OPTION** on the top right corner of your screen and click on **PUBLISH**

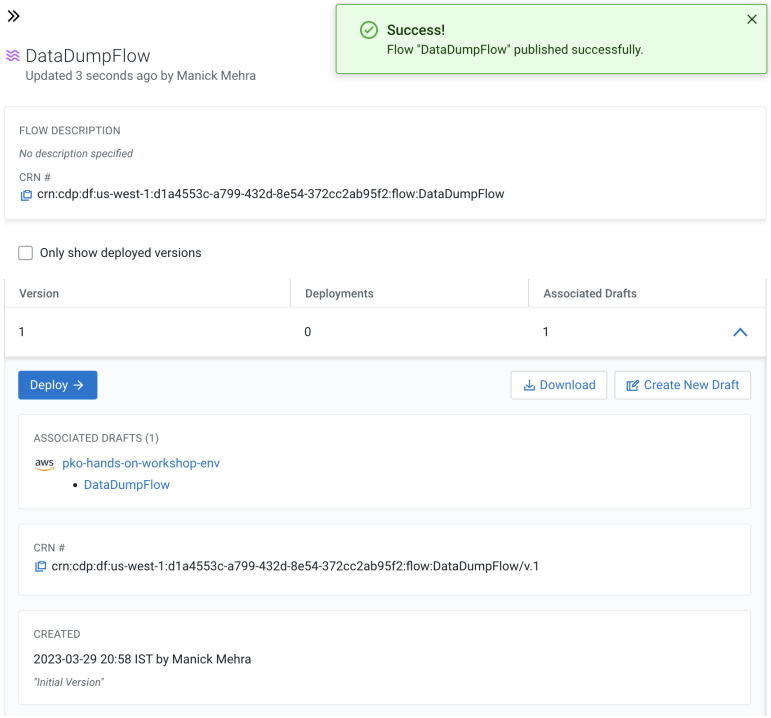


Step 3: Give your flow a name and click on **PUBLISH**

```
Flow Name : {user_id}_datadump_flow
```

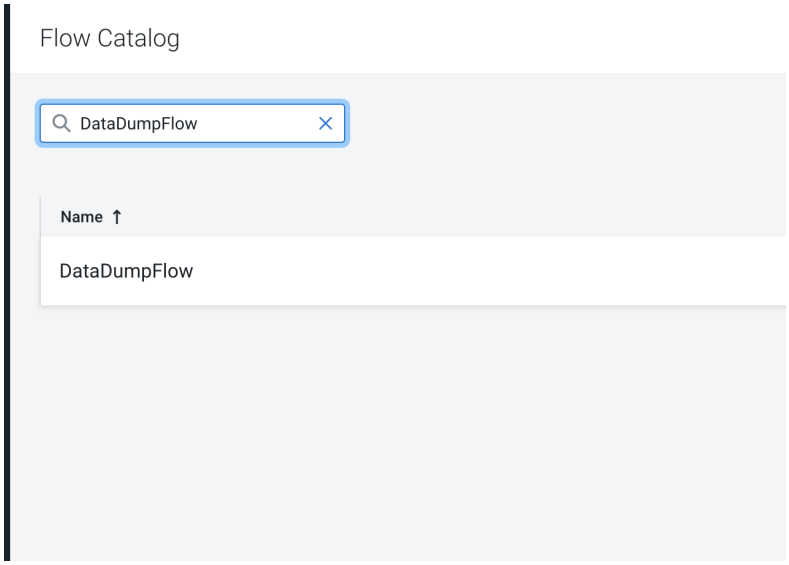


The flow will now be visible on the **FLOW CATALOG** and is ready to be deployed

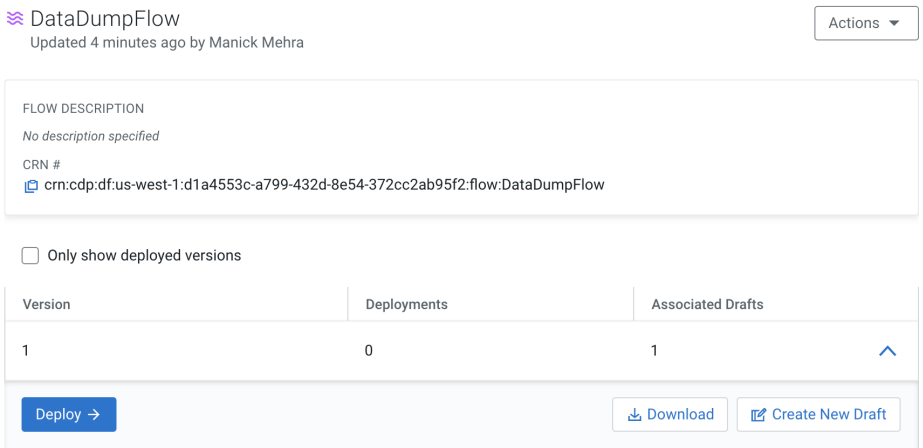


5. Deploying the Flow

Step 1: Search for the flow in the Flow Catalog

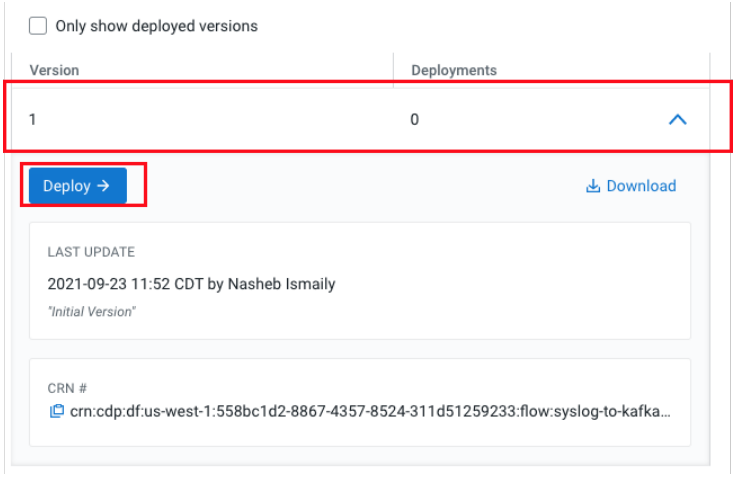


Click on the Flow, you should see the following:



Step 2: Deploy the flow

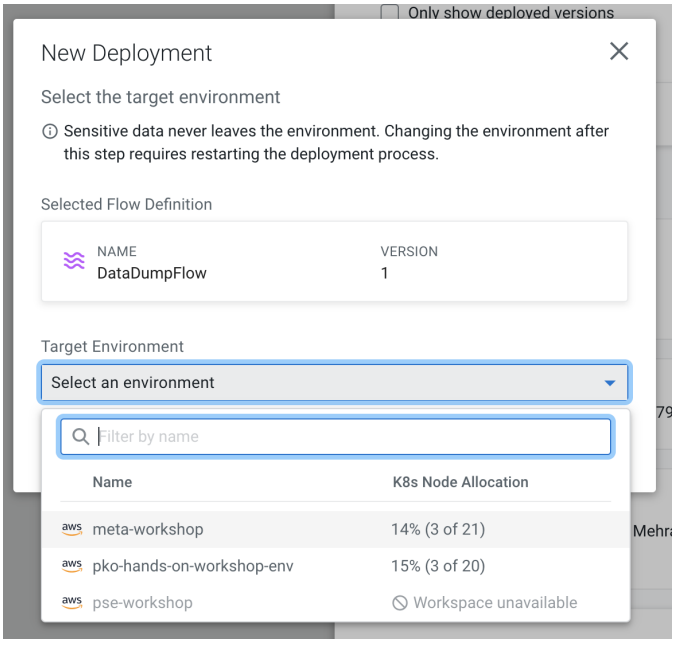
Click on **Version 1**, you should see a **Deploy** Option appear shortly. Then click on **Deploy**.



Step 3: Select the CDP environment

Select the CDP environment where this flow will be deployed and click on **CONTINUE**

NOTE: THE NAME OF THE ENVIRONMENT WILL BE SHARED BY THE INSTRUCTOR

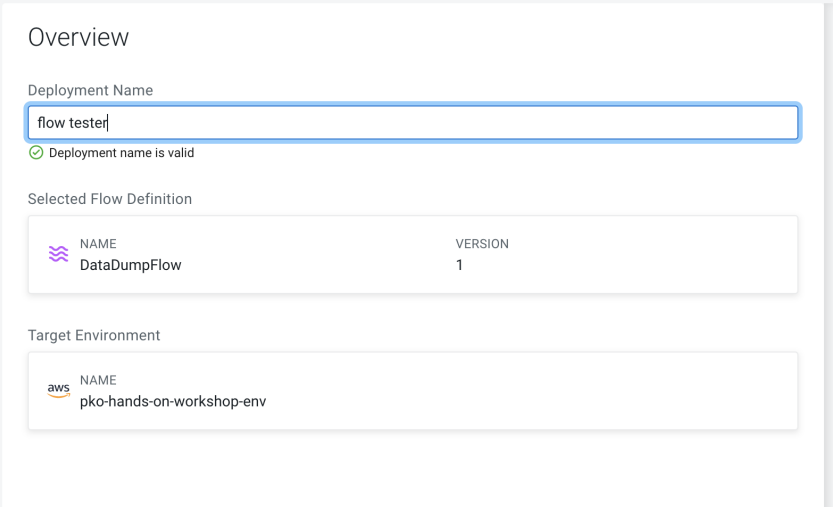


Step 4: Deployment Name

Give the deployment a unique name({user_id}_flow_prod), then click Next.

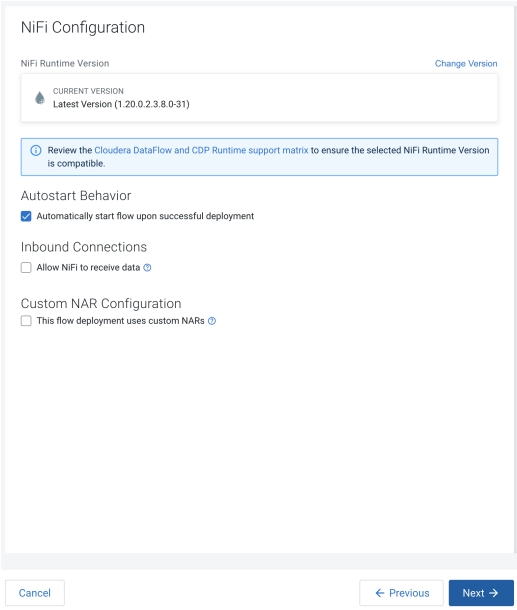
Example : apac01_flow_prod

Click NEXT



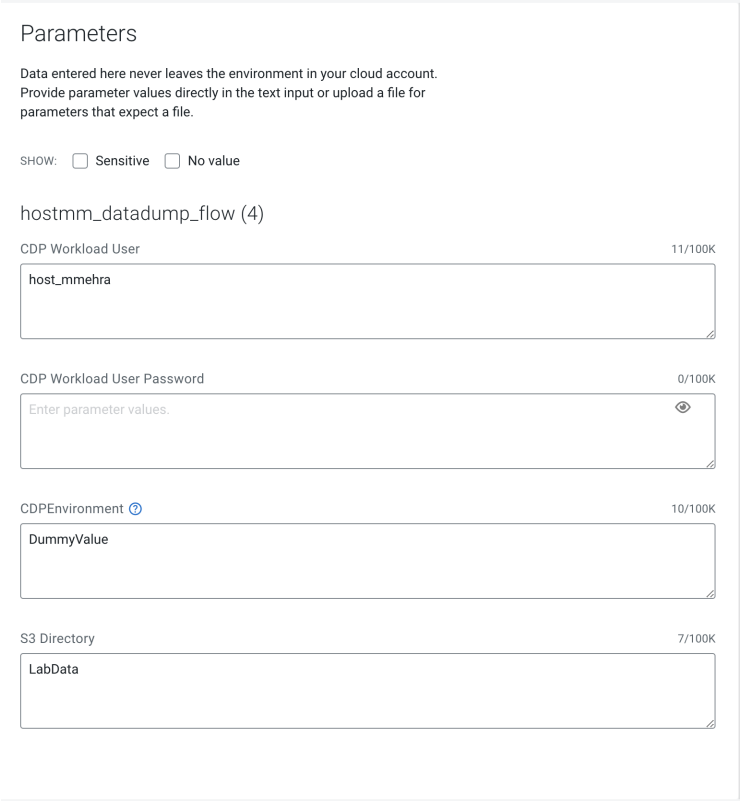
Step 5: Set the NiFi Configuration

We can let everything be the default here and click NEXT



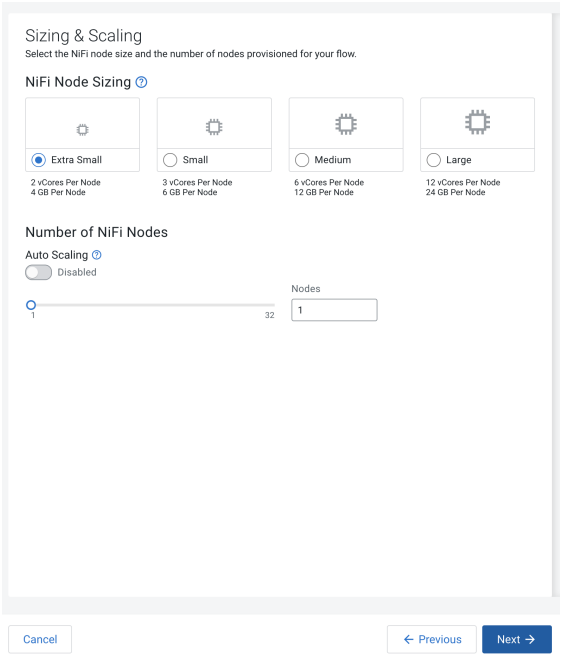
Step 6: Set the Parameters

Set the Username, Password and the Directory name and click NEXT
CDP Workload User: apacXY
CDP Workload User Password: ApacLabs@23
S3 Directory: dirFlowCatalogDataDump
CDP Environment : DummyParameter
[The CDP Environment parameter that shows here is used at the time we perform a test run on our test session. It holds the CDP Environment configuration resources files such as ssl-client.xml, hive-site.xml and core-site.xml. You do not have to specify these to deploy your flow from the flow catalog as it automatically picks up those files,hence we give a dummy value to this. To avoid giving a dummy value, this parameter can be deleted before we publish the flow]



Step 7: Set the cluster size

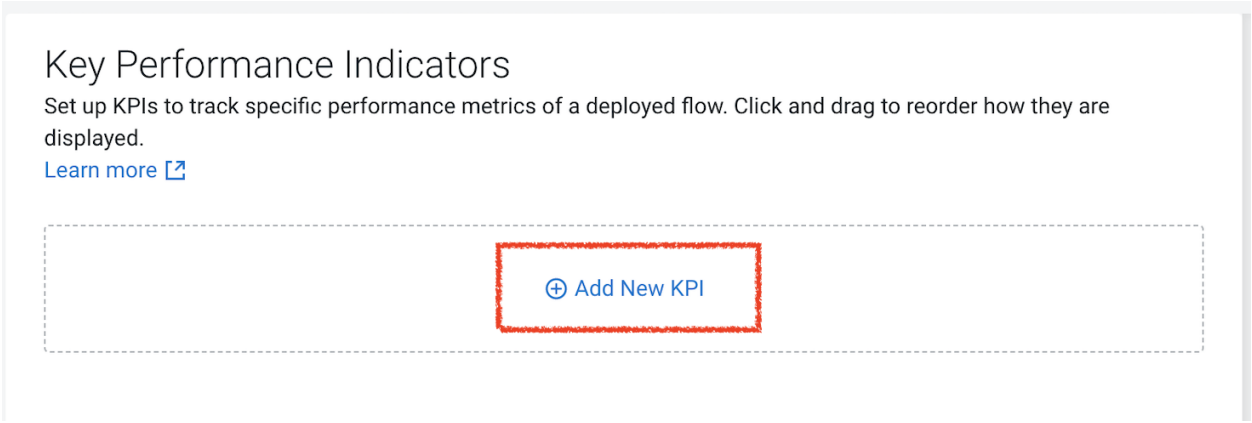
Select the Extra Small size and click NEXT. In this step you can configure how your flow will autoscale, but keep it disabled for this lab.



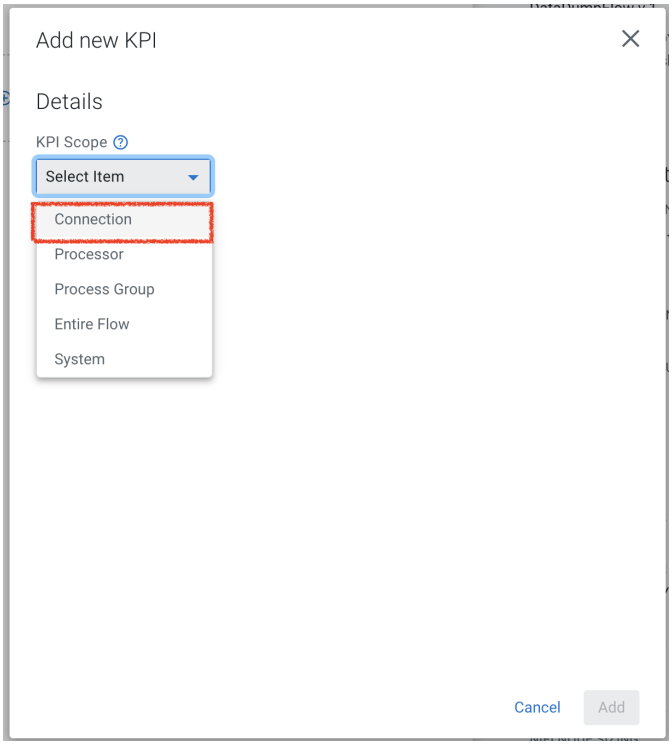
Step 8: Add Key Performance indicators

Set up KPIs to track specific performance metrics of a deployed flow.

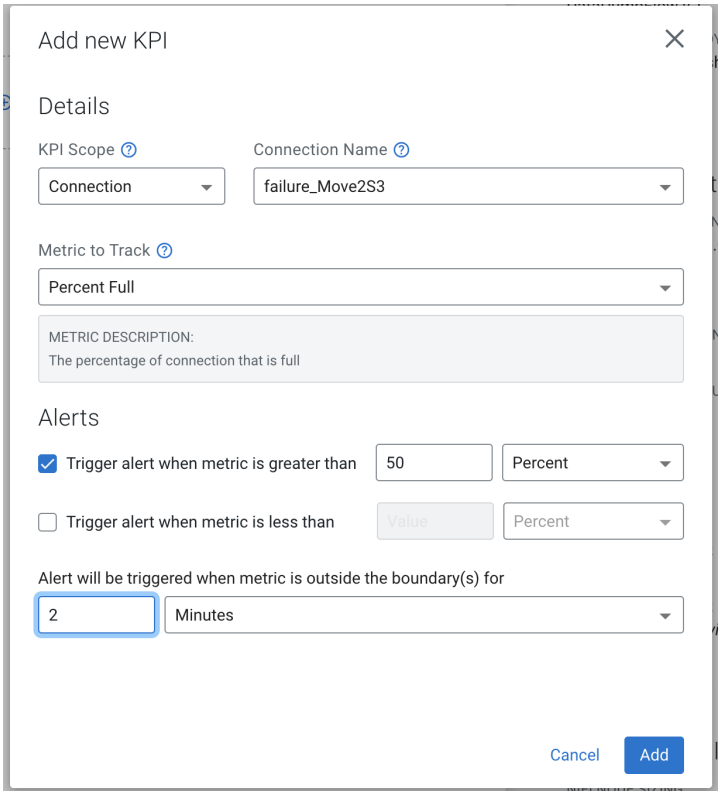
Click on “Add New KPI”



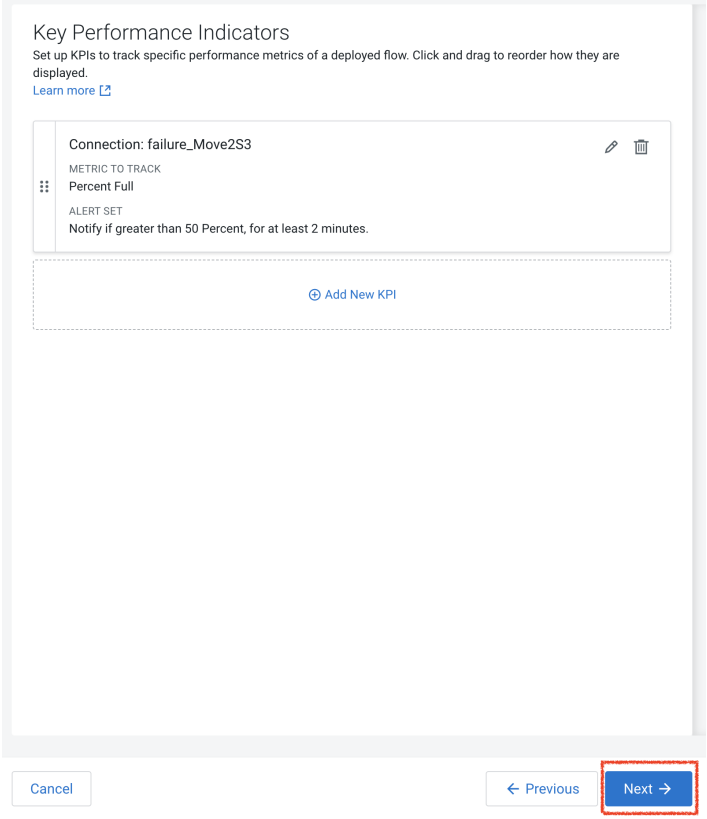
In the KPI Scope drop-down list, choose “Connection”



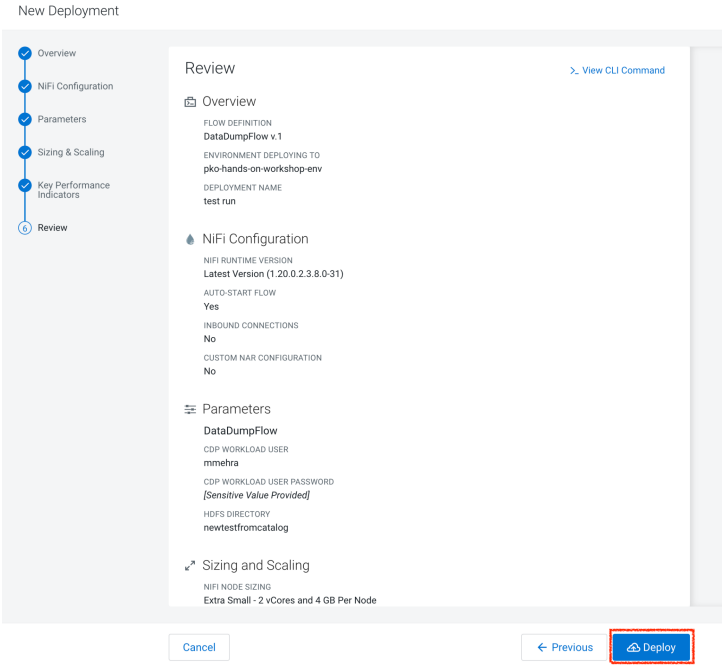
In the “Add New KPI” window, add an alert as below



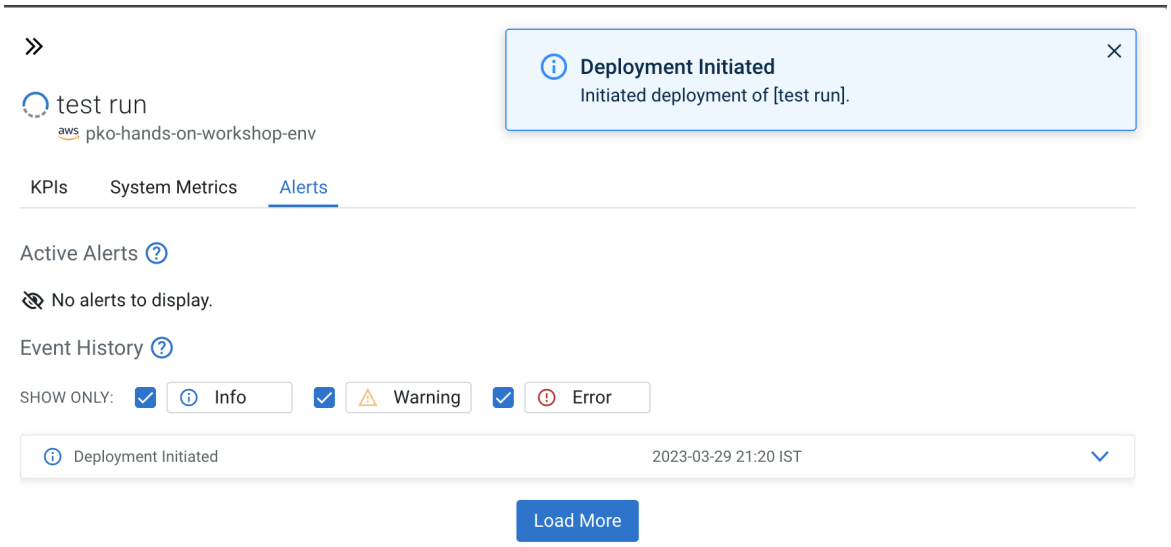
Click Add and then Click Next



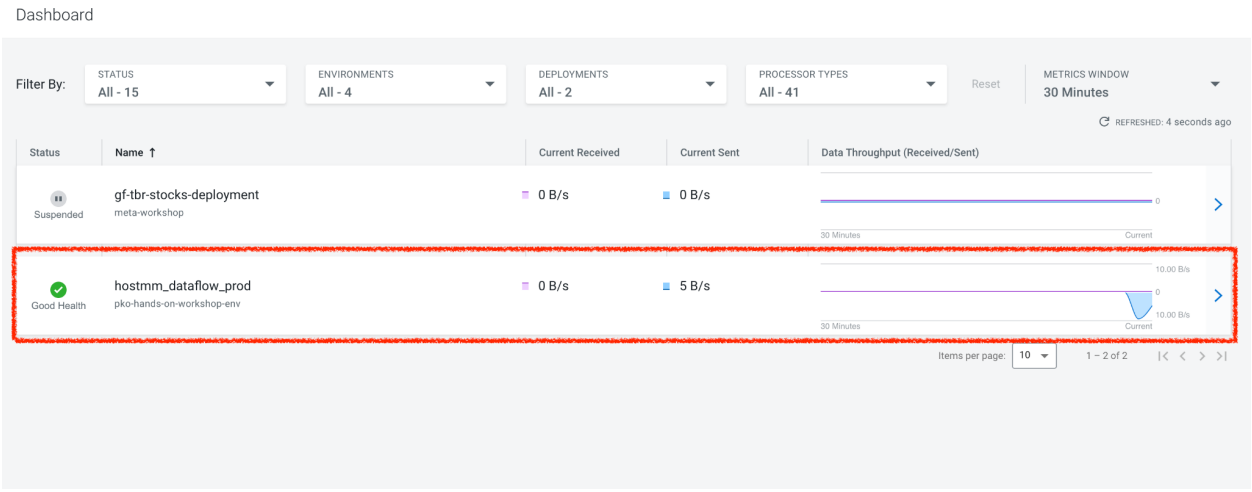
Step 9: Click **Deploy**



The “Deployment Initiated” message will be displayed. Wait until the flow deployment is completed, which might take a few minutes.

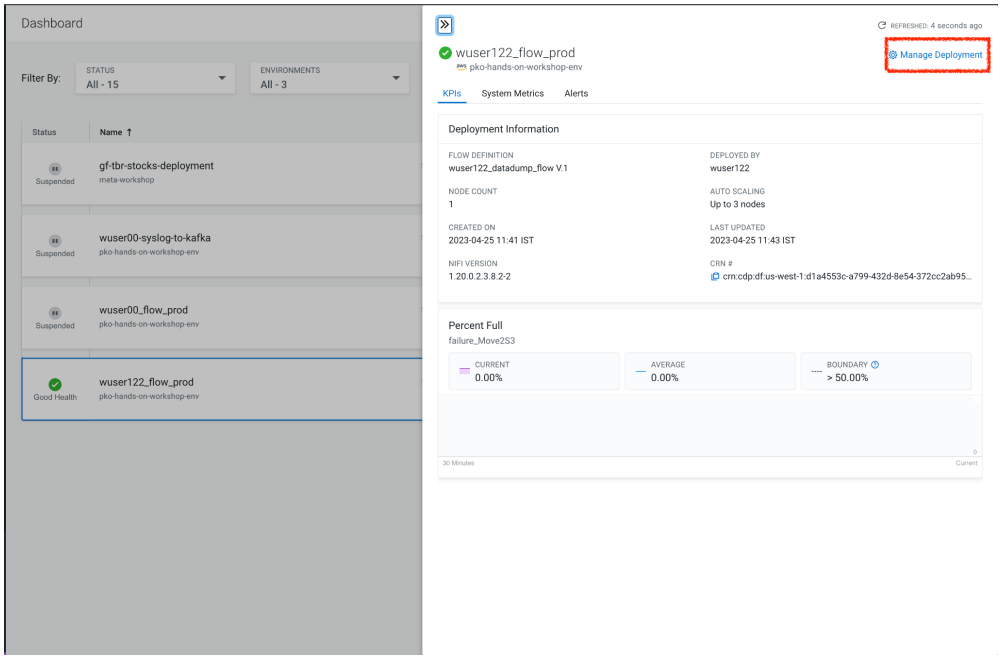


When deployed, the flow will show up on the Data flow dashboard, as below:



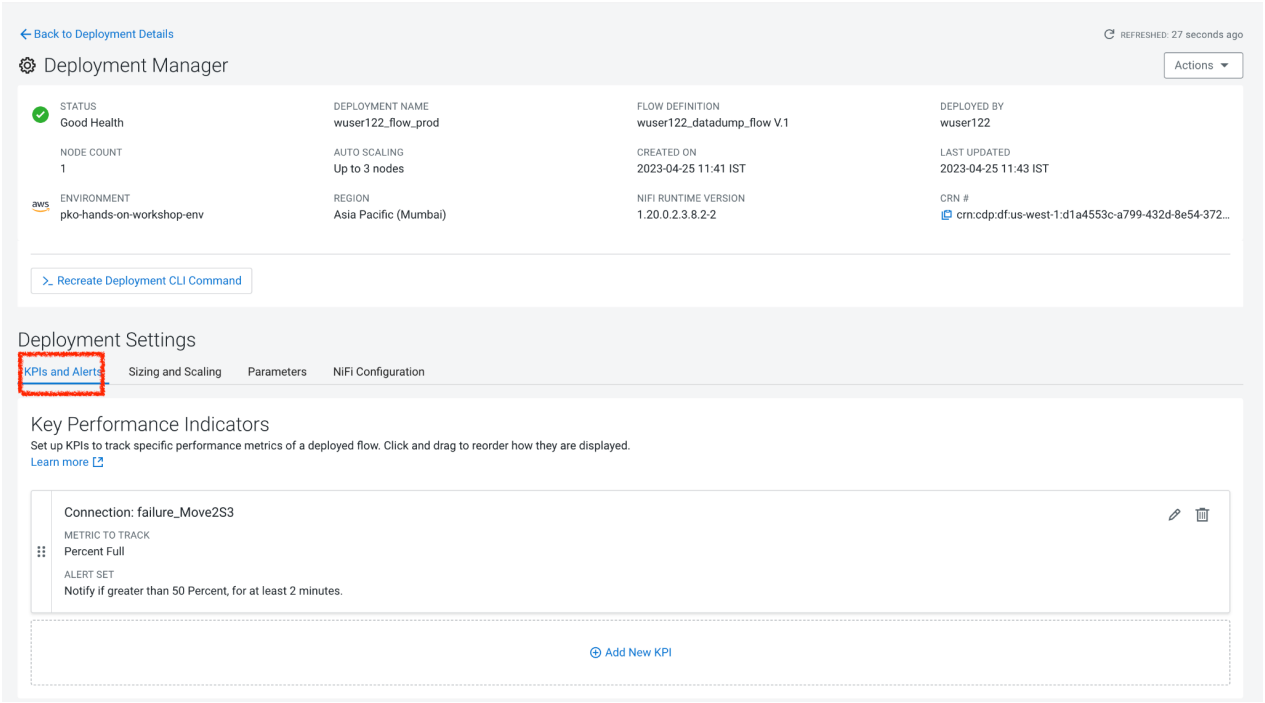
6. Viewing details of the deployed flow

Click on the flow in the Dashboard and select Manage Deployment



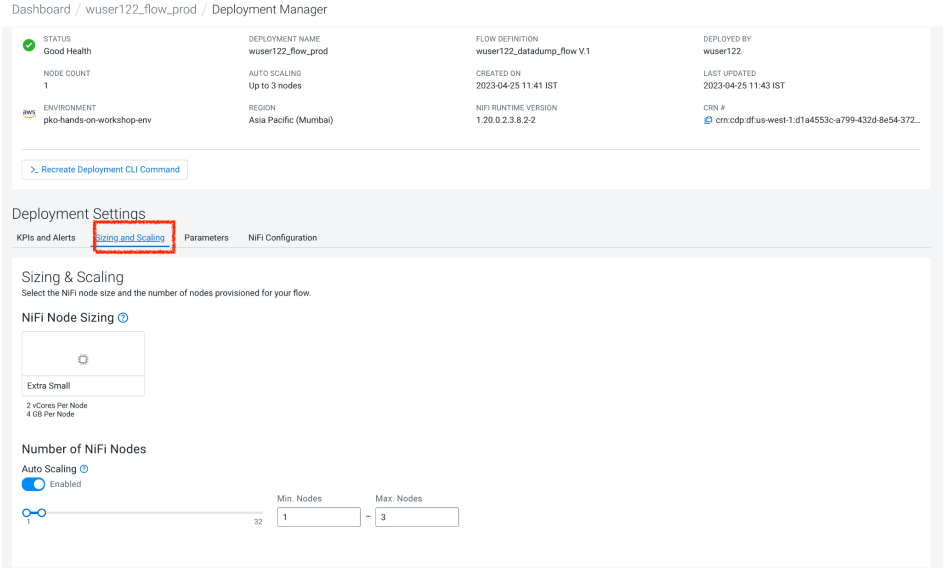
Step 1 : Manage KPI and Alerts

Click on the KPI tab to get the list of KPIs that have been set. You also have an option to modify or add more KPIs to your flow here.



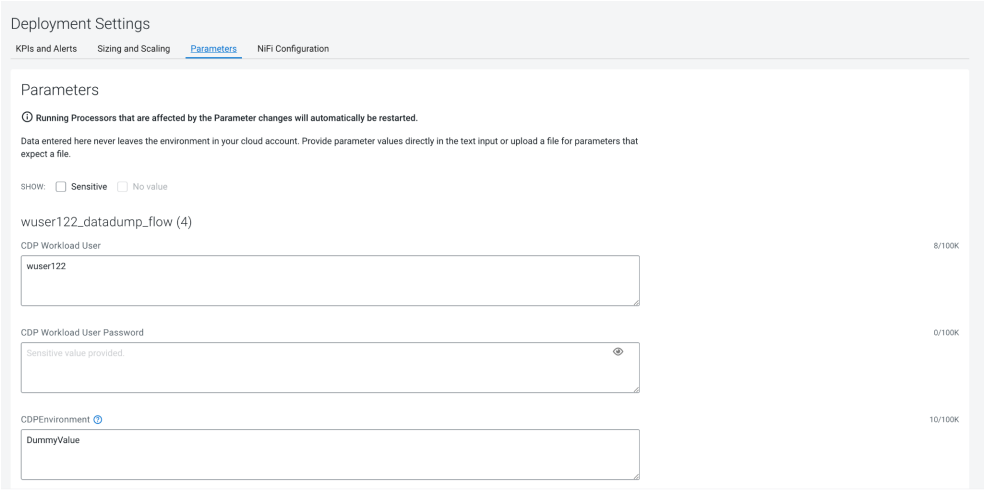
Step 2 : Manage Sizing and Scaling

Click on the Sizing and Scaling tab to get detailed information



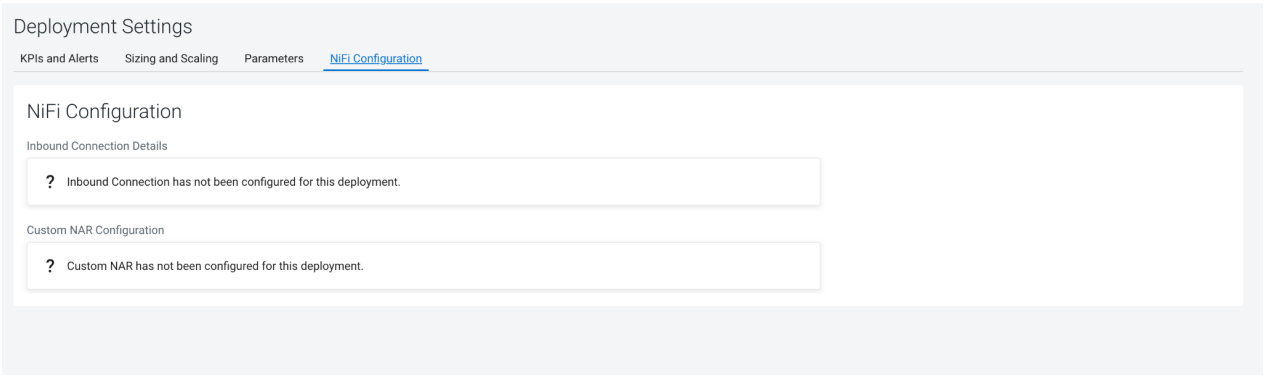
Step 3 : Manage Parameters

The parameters that we earlier created can be managed from the Parameters tab. Click on Parameters.



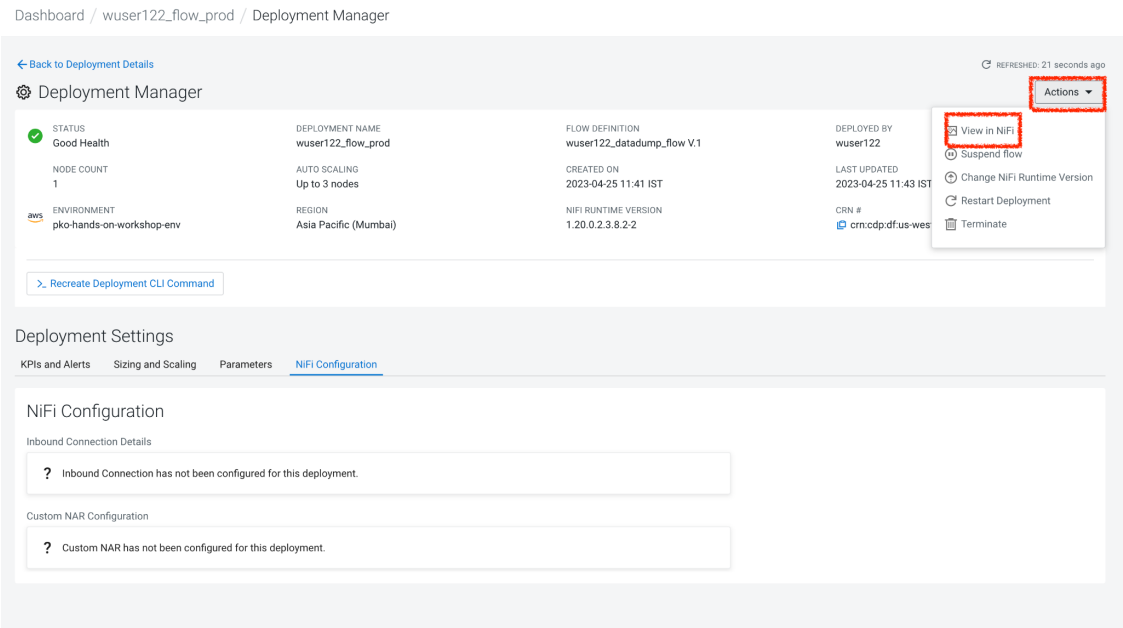
Step 4 : NiFi Configurations

If you have set any configuration wrt to Nifi they will show up on the ‘NiFi Configuration’ tab

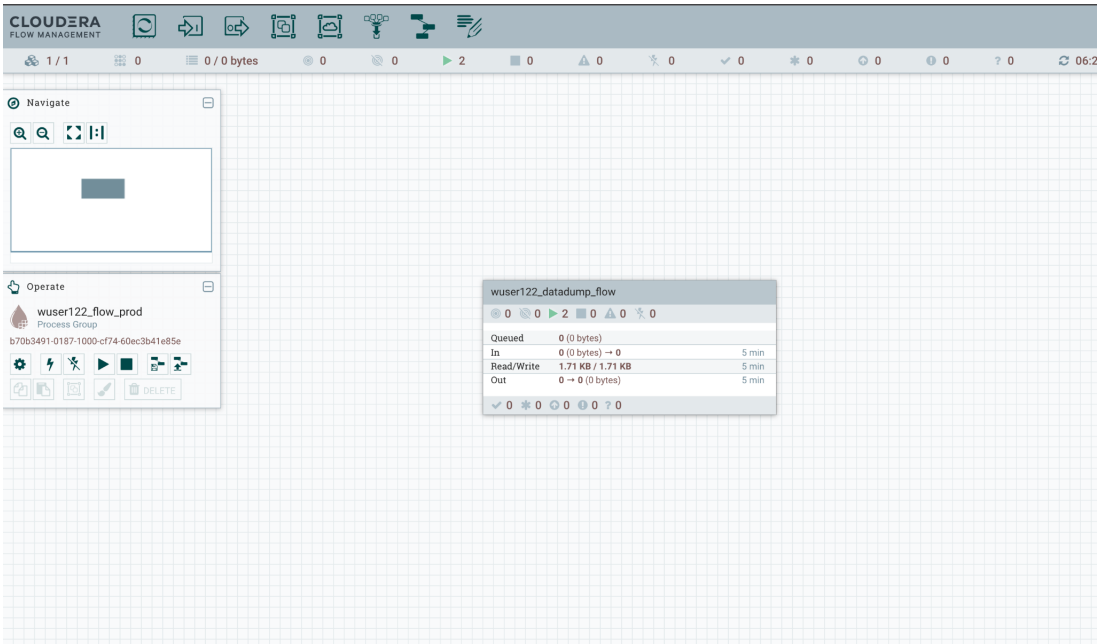


Step 5 : View the deployed flow in NiFi

- Select ACTIONS on the Deployment Manager page and then click on ‘View in NiFi’



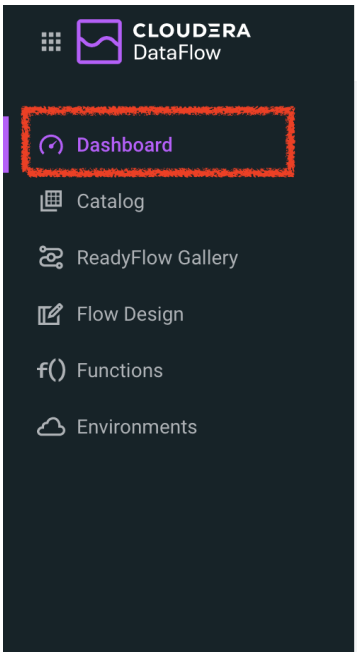
This will open the flow in the NiFi UI.



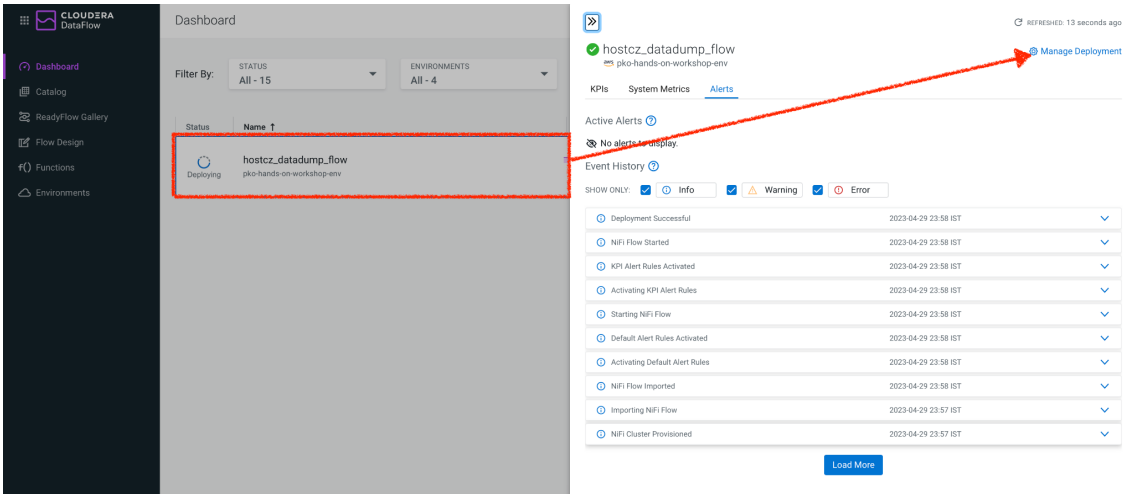
Step 6 : Terminate the flow

As we have completed the Lab, it is best to terminate this flow. Follow the below given procedure to terminate your flow.

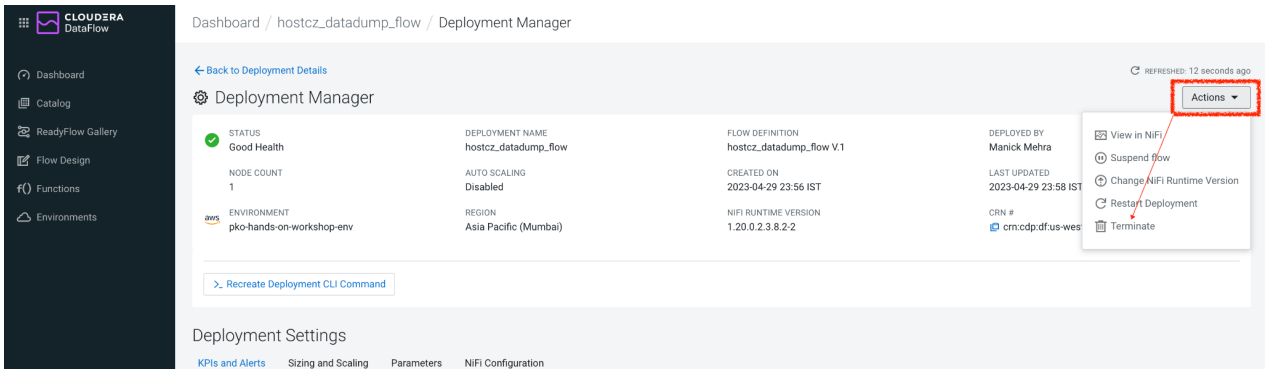
Select Dashboard from the Cloudera Data Flow UI



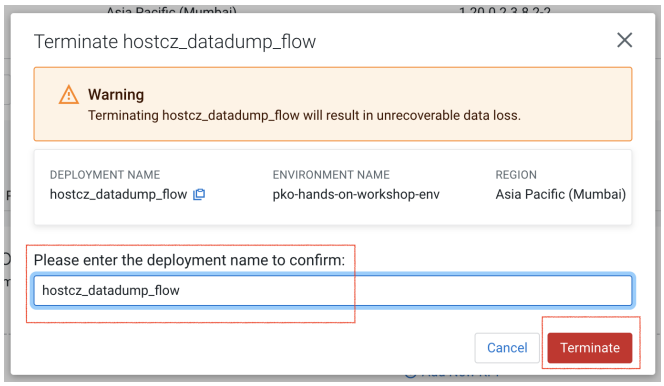
Select your flow and go to Manage Deployment



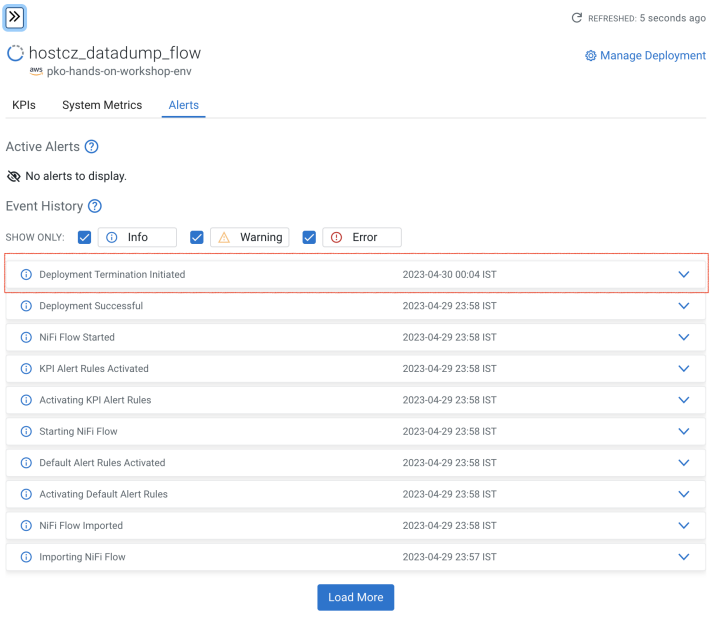
On the Deployment Manager Page, Select **Actions** and click on **Terminate**



In the next dialog box, enter the name of the flow we are trying to terminate and click on **Terminate**



You will now see that the termination process has started.



Lab 2 : Migrating Existing Data Flows to CDF-PC

1. Overview

The purpose of this workshop is to demonstrate how existing NiFi flows can be migrated to the Data Flow Experience. This workshop will leverage an existing NiFi flow template that has been designed with the best practices for CDF-PC flow deployment.

The existing NiFi Flow will perform the following actions:

1. Generate random syslogs in 5424 Format
2. convert the incoming data to a JSON using record writers
3. Apply a SQL filter to the JSON records
4. Send the transformed syslog messages to Kafka

Note that a parameter context has already been defined in the flow and the queues have been uniquely named.

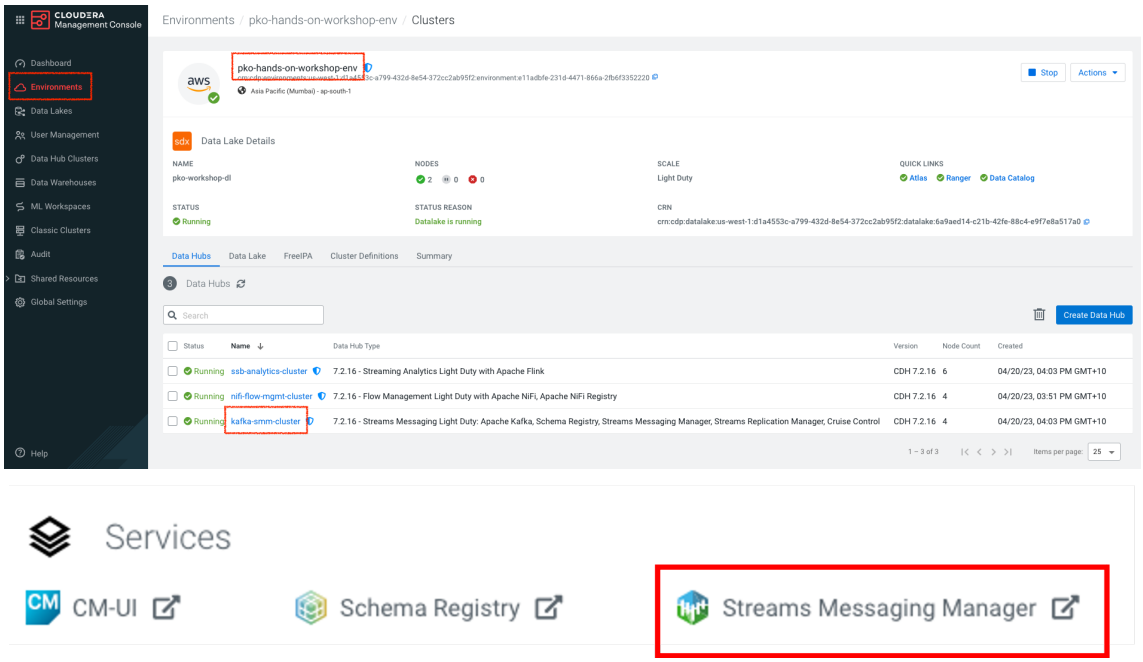
For this we will be leveraging the DataHubs which have already been created, namely:

- `workshop-asean-streams-mess`
- `workshop-asean-stream-analytics`

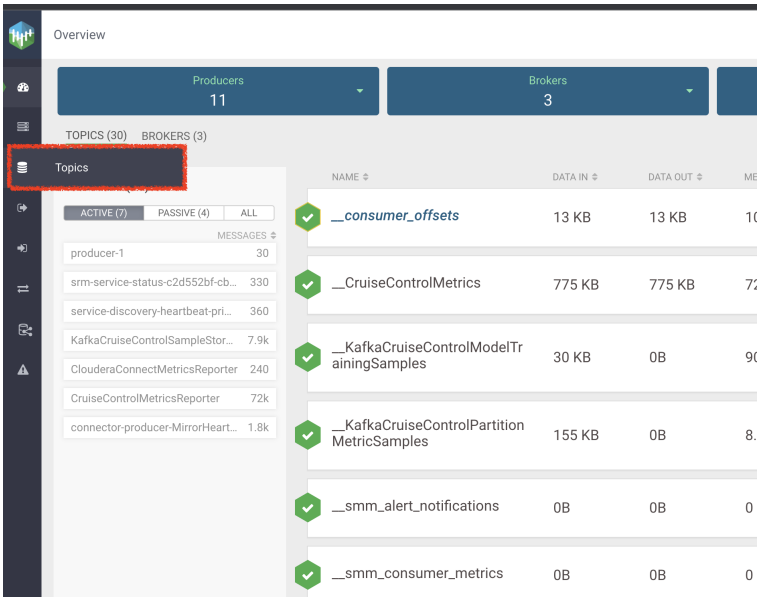
2. Pre-requisites

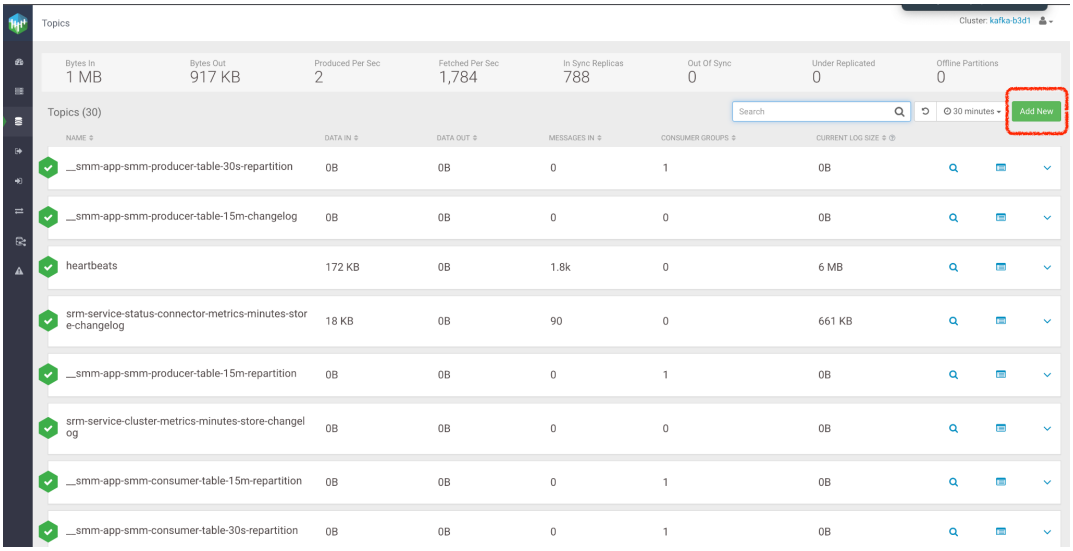
2.1. Create a Kafka Topic

1. Login to Streams Messaging Manager by clicking the appropriate hyperlink in the Streams Messaging Datahub (workshop-asean-streams-mess)

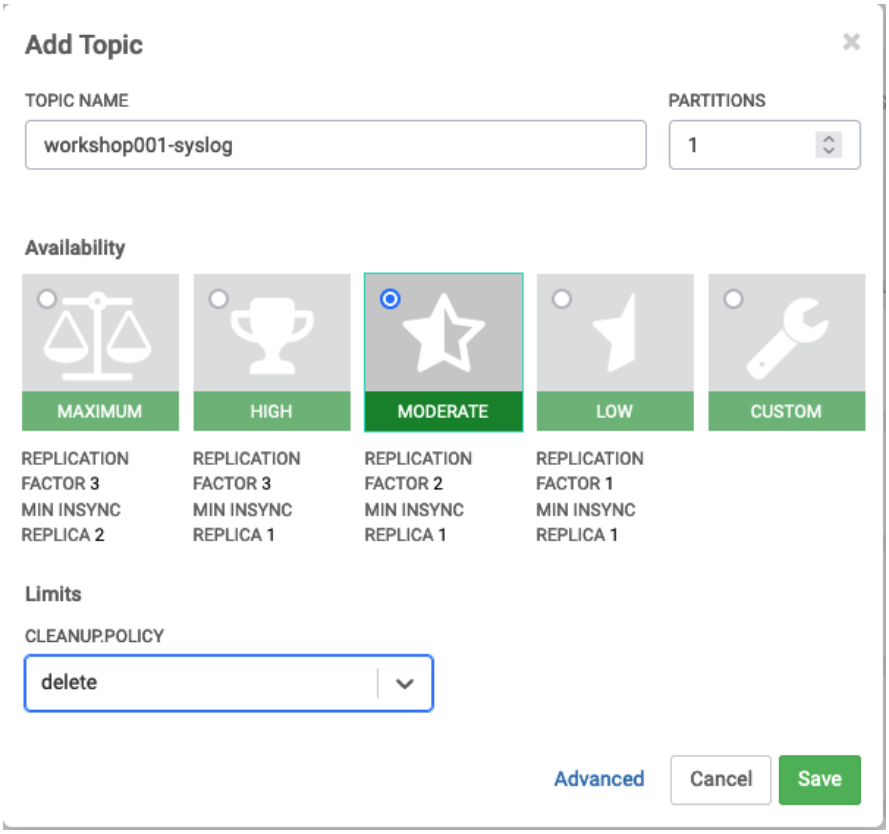


2. Click on Topics in the left tab





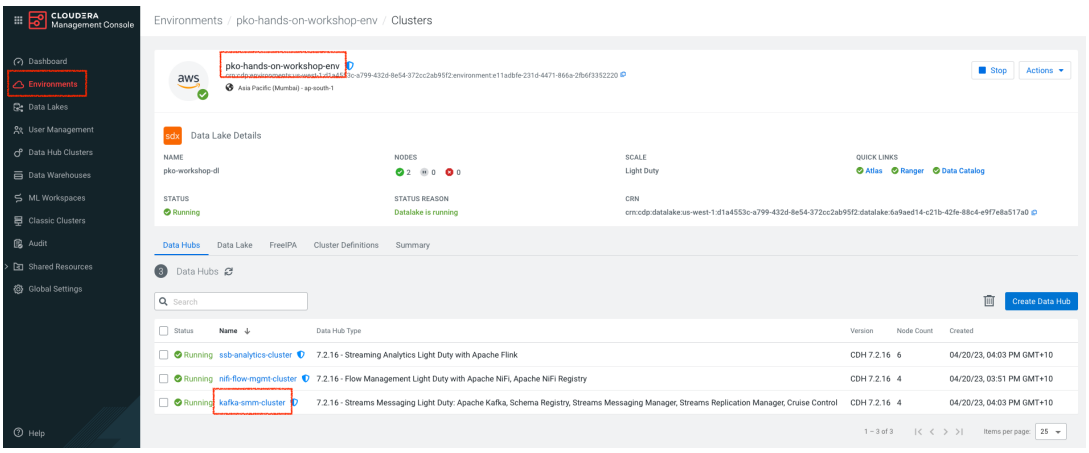
4. Create a Topic with the following parameters then click **Save**:
    - **Name**:<username>-syslog
    - **Partitions**: 1
    - **Availability**: Moderate
    - **Cleanup Policy**: Delete

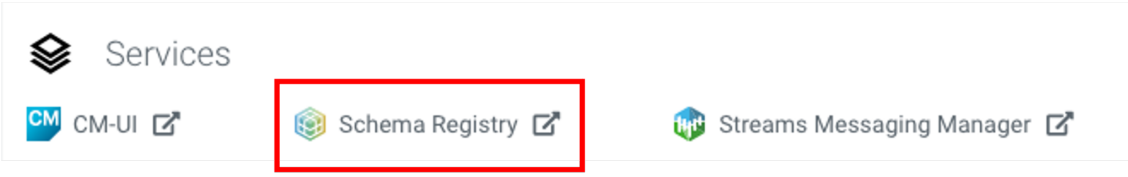


Note: The Flow will not work if you set the Cleanup Policy to anything other than **Delete**. This is because we are not specifying keys when writing to Kafka.

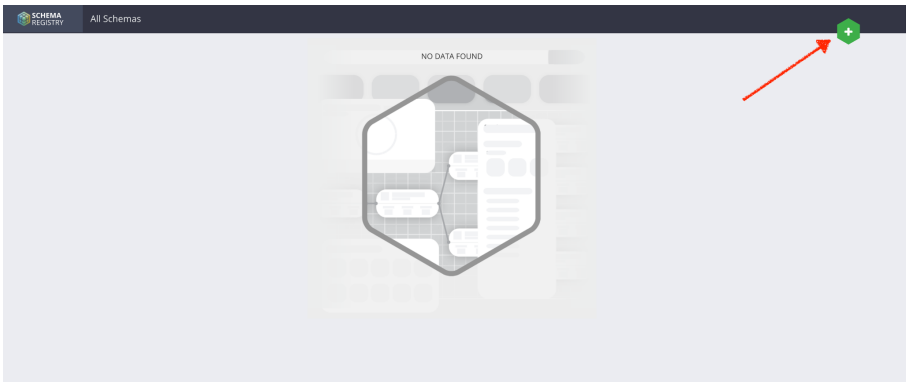
2.2. Create a Schema in Schema Registry

1. Login to Schema Registry by clicking the appropriate hyperlink in the Streams Messaging Datahub(`kafka-smm-cluster`)





2. Click on the + button on the top right to create a new schema.



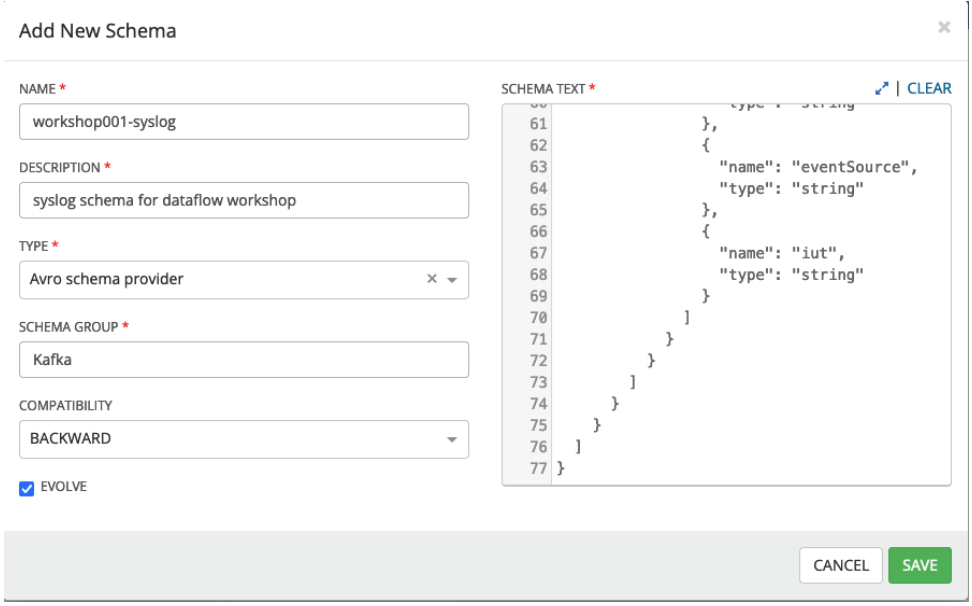
3. Create a new schema with the following information:
    - **Name**: <username>-syslog
    - **Description**: syslog schema for dataflow workshop
    - **Type**: Avro schema provider
    - **Schema Group**: Kafka
    - **Compatibility**: Backward
    - **Evolve**: True
    - **Schema Text**: Copy and paste the schema text below into the “Schema Text” field

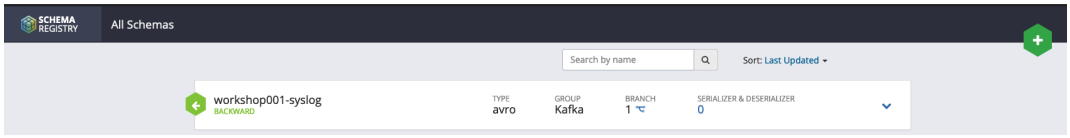
```
{
  "name": "syslog",
  "type": "record",
  "namespace": "com.cloudera",
  "fields": [
    {
      "name": "priority",
      "type": "int"
    },
    {
      "name": "severity",
      "type": "int"
    },
    {
      "name": "facility",
      "type": "int"
    },
    {
      "name": "version",
      "type": "int"
    },
    {
      "name": "timestamp",
      "type": "long"
    },
    {
      "name": "hostname",
      "type": "string"
    },
    {
      "name": "body",
      "type": "string"
    },
    {
      "name": "appName",
      "type": "string"
    },
```

```
    {
      "name": "procid",
      "type": "string"
    },
    {
      "name": "messageid",
      "type": "string"
    },
    {
      "name": "structuredData",
      "type": {
        "name": "structuredData",
        "type": "record",
        "fields": [
          {
            "name": "SDID",
            "type": {
              "name": "SDID",
              "type": "record",
              "fields": [
                {
                  "name": "eventId",
                  "type": "string"
                },
                {
                  "name": "eventSource",
                  "type": "string"
                },
                {
                  "name": "iut",
                  "type": "string"
                }
              ]
            }
          }
        ]
      }
    }
  ]
}
```

Note: The name of the Kafka Topic you previously created and the Schema Name must be the same.

Click on **SAVE**.

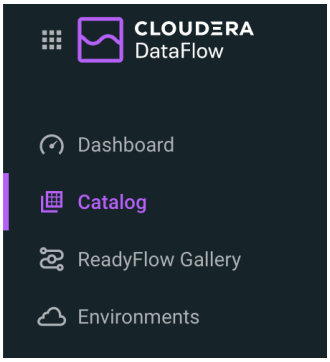

Add New Schema
NAME *
workshop001-syslog
DESCRIPTION *
syslog schema for dataflow workshop
TYPE *
Avro schema provider
SCHEMA GROUP *
Kafka
COMPATIBILITY
BACKWARD
EVOLVE
SCHEMA TEXT *
CLEAR
},
{
"name": "eventSource",
"type": "string"
},
{
"name": "iut",
"type": "string"
}
]
}
}
]
}
}
]
}
CANCEL
SAVE


SCHEMA REGISTRY
All Schemas
Search by name
Sort: Last Updated
workshop001-syslog
BACKWARD
TYPE
avro
GROUP
Kafka
BRANCH
1
SERIALIZER & DESERIALIZER
0

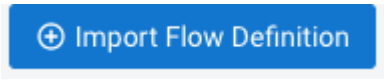
Lab 3 : Operationalizing Externally Developed Data Flows with CDF-PC

1. Import the Flow into the CDF-PC Catalog

a. Open the CDF-PC data service and click on Catalog in the left tab.



b. Select Import Flow Definition on the Top Right

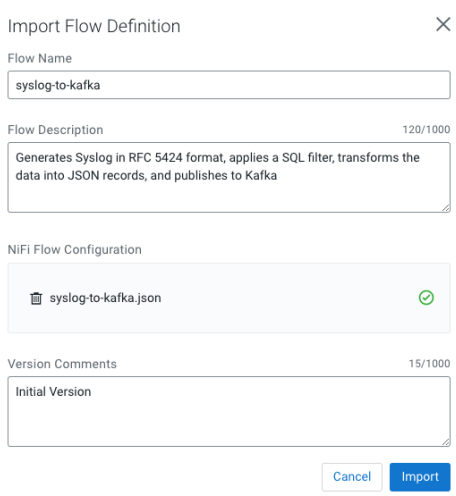


c. Add the following information:

- **Flow Name:** <username>-syslog-to-kafka
- **Flow Description:**

Reads Syslog in RFC 5424 format, applies a SQL filter, transforms the data into JSON records, and publishes to Kafka

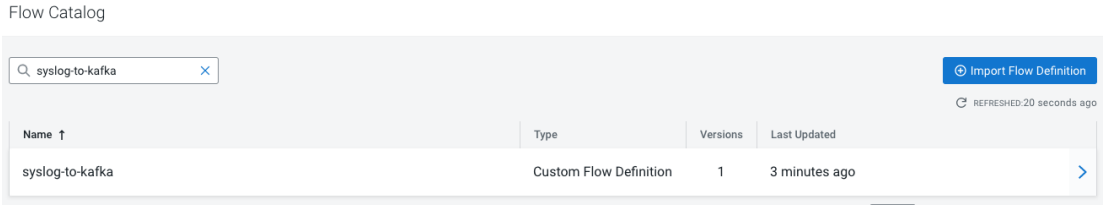
- **NiFi Flow Configuration:** syslog-to-kafka.json (From the resources downloaded earlier)
- **Version Comments:** Initial Version



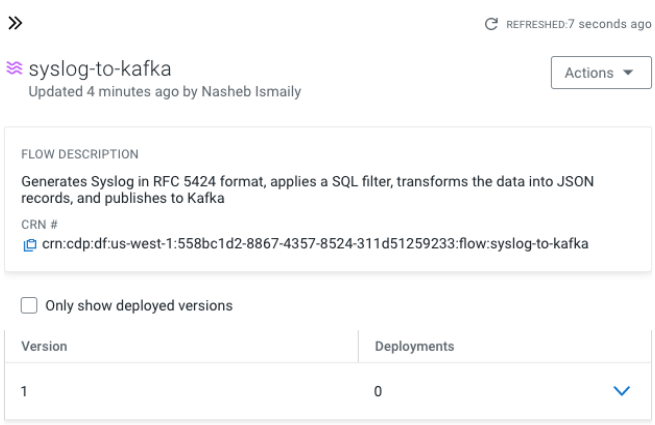
Click **IMPORT**

2. Deploy the Flow in CDF-PC

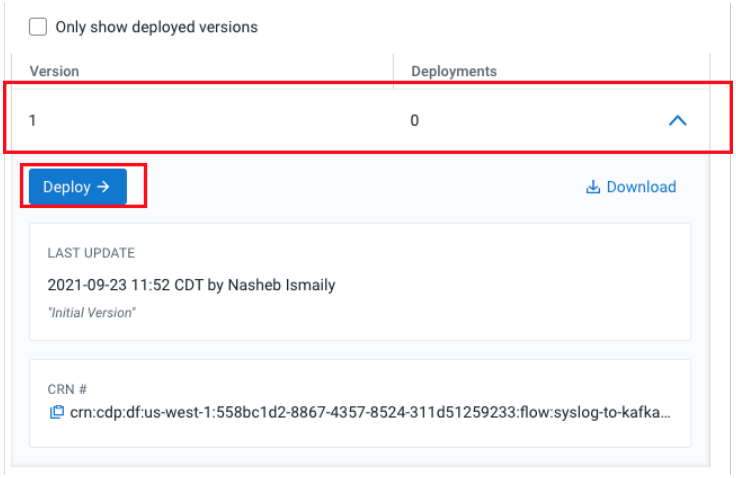
1. Search for the flow in the Flow Catalog



2. Click on the Flow, you should see the following:

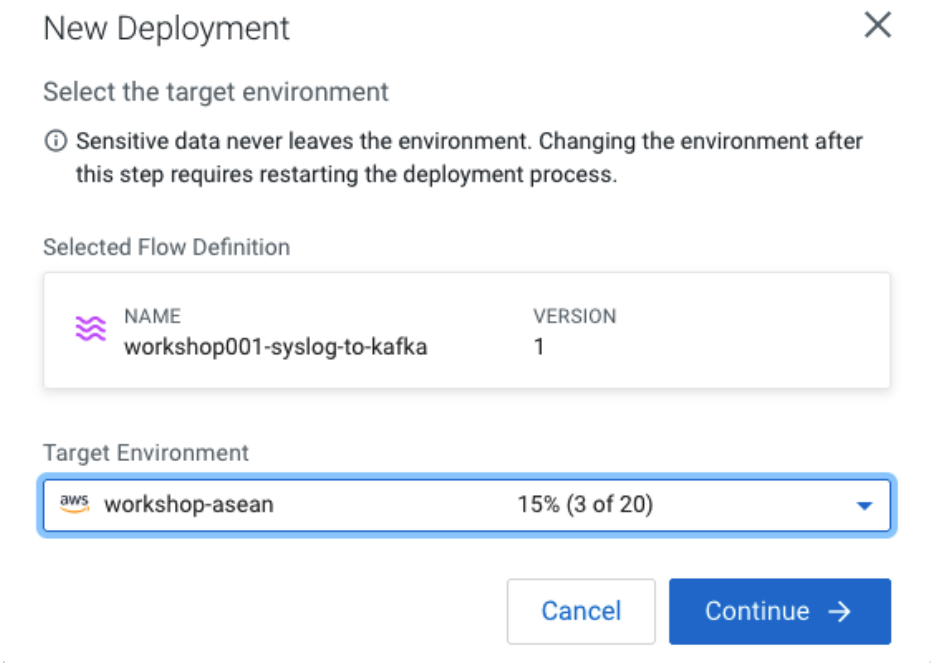


3. Click on **Version 1**, you should see a **Deploy** Option appear shortly. Then click on **Deploy**.



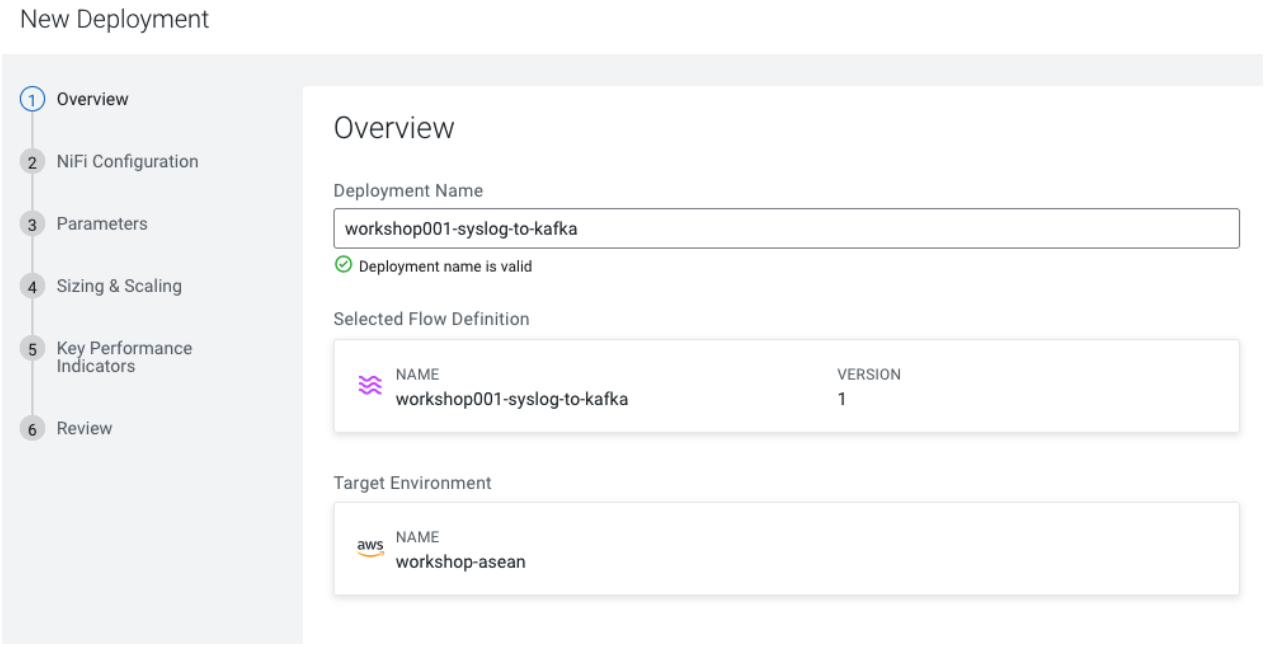
4. Select the CDP environment where this flow will be deployed, then click **Continue**.

   NOTE: THE NAME OF THE ENVIRONMENT WILL BE SHARED BY THE INSTRUCTOR

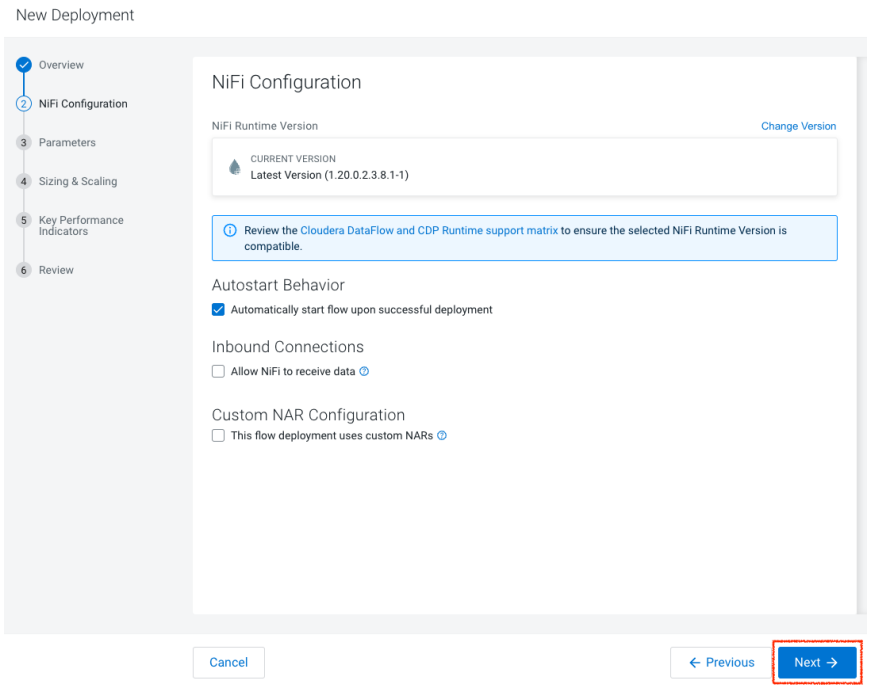


5. Give the deployment a unique name, then click **Next**.

   Example : {user_id}-syslog-to-kafka



6. In the NiFi Configuration screen, click **Next**.

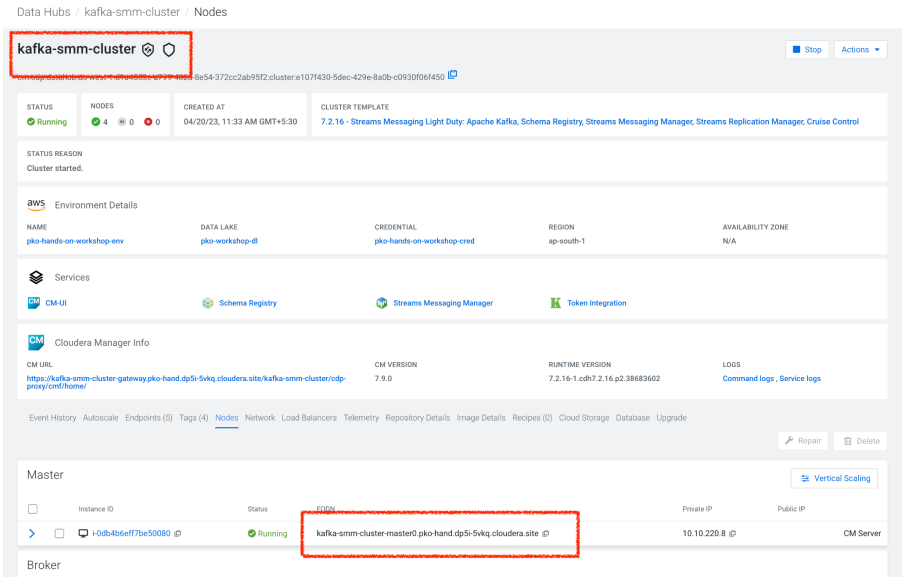


7. Add the Flow Parameters as below, then click **Next**.
    - **CDP Workload User** - The workload username for the current user
        - Example : workshop001
    - **CDP Workload Password** - The workload password for the current user P@ssw0rd@2023
    - **Filtre Rule -** `SELECT * FROM FLOWFILE`
    - **Kafka Broker Endpoint** - A comma separated list of Kafka Brokers.

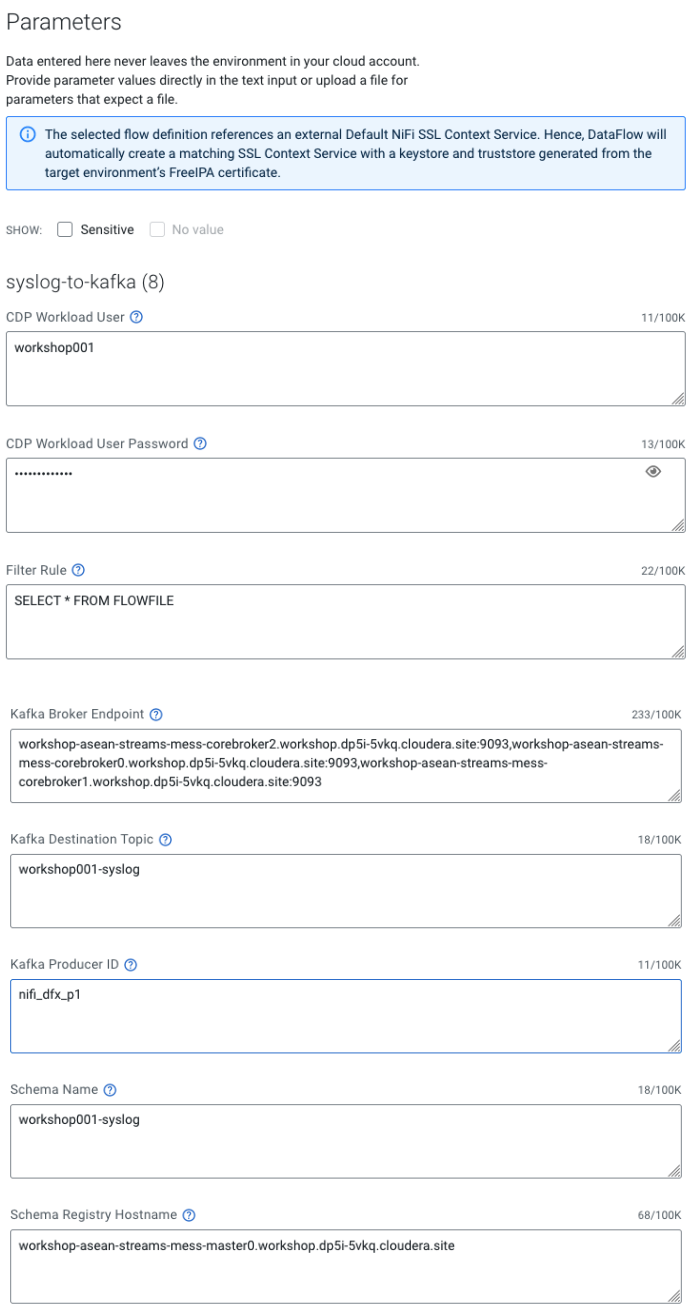
      `[Obtained in Lab 0, section 4]`

      `Example:` workshop-asean-streams-mess-corebroker2.workshop.dp5i-5vkq.cloudera.site:9093,workshop-asean-streams-mess-corebroker0.workshop.dp5i-5vkq.cloudera.site:9093,workshop-asean-streams-mess-corebroker1.workshop.dp5i-5vkq.cloudera.site:9093

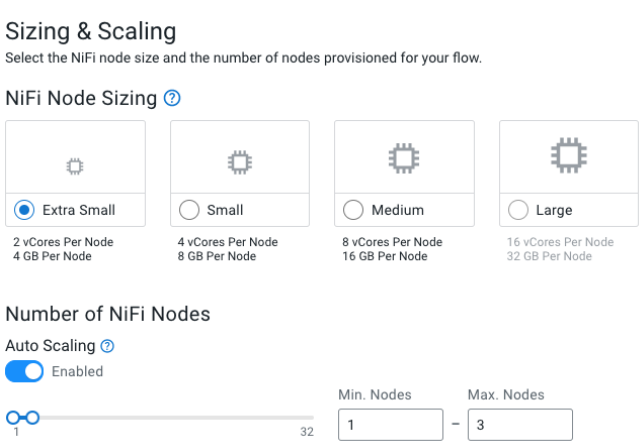
    - **Kafka Destination Topic -** <username>-syslog (Ex: workshop001-syslog)
    - ***Kafka Producer ID*** *- nifi_dfx_p1*
    - **Schema Name** - <username>-syslog (Ex: workshop001-syslog)
    - **Schema Registry Hostname** - The hostname of the master server in the Kafka Datahub(`kafka-smm-cluster`)[Refer screenshot below]



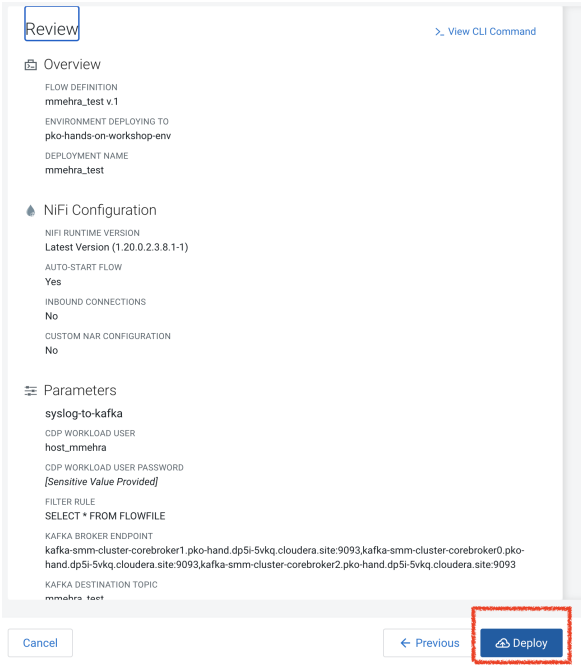
Example:workshop-asean-streams-mess-master0.workshop.dp5i-5vkq.cloudera.site



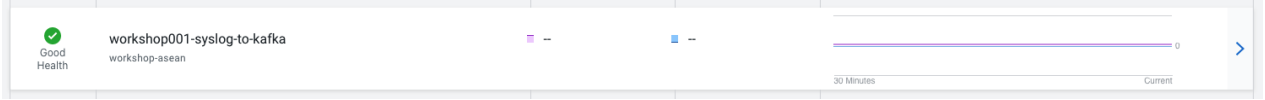
8. On the next page, define the Sizing and Scaling as follows, then click **Next**.
    - **Size:** Extra Small
    - **Enable Auto Scaling:** True
    - **Min Nodes:** 1
    - **Max Nodes:** 3



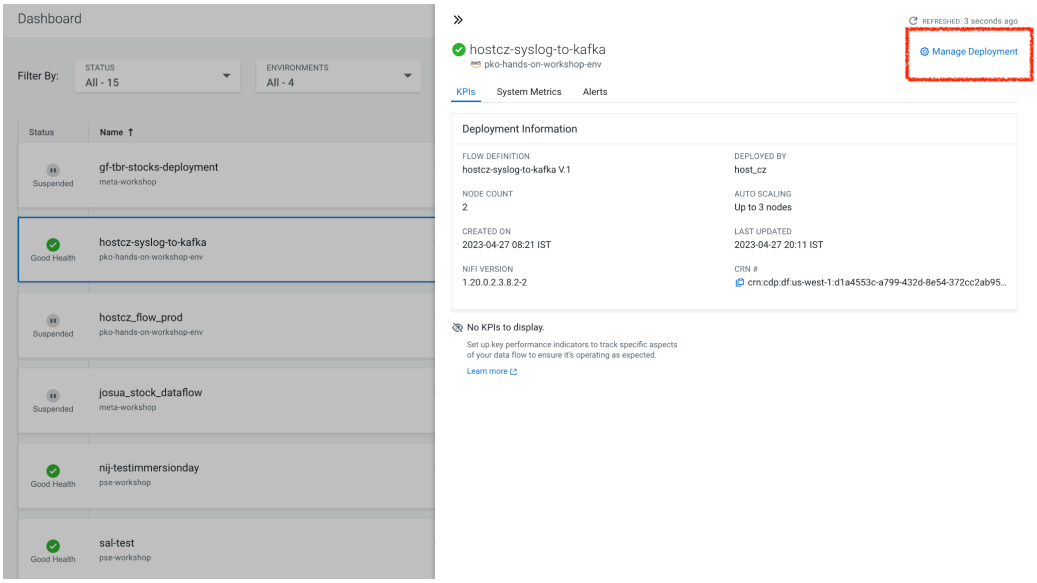
9. Skip the KPI page by clicking **Next** and Review your deployment. Then Click **Deploy**.



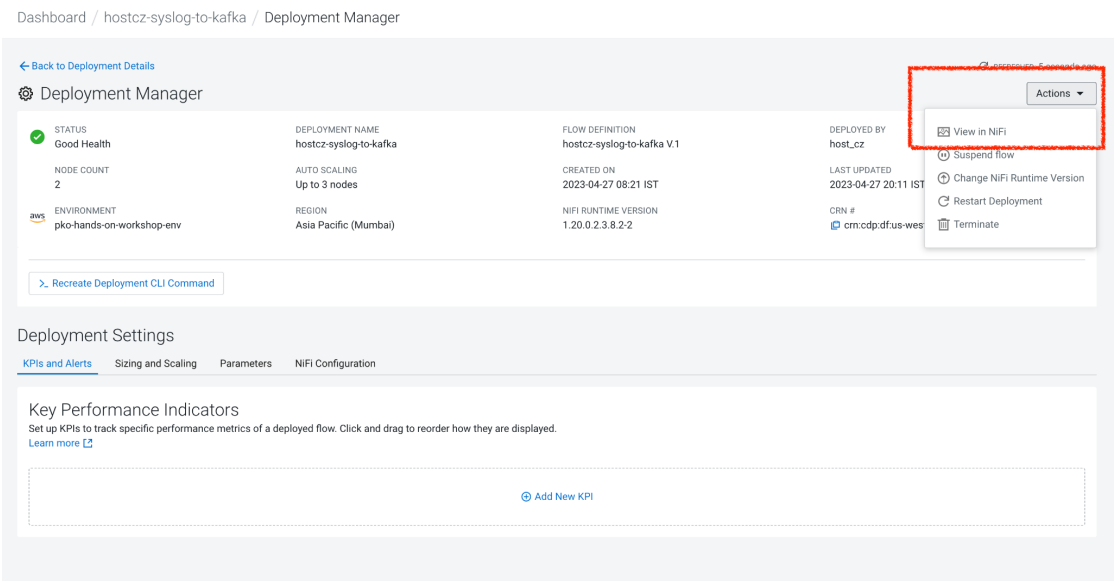
10. Proceed to the CDF-PC Dashboard and wait for your flow deployment to complete, which might take a few minutes. A Green Check Mark will appear once complete, which might take a few minutes.



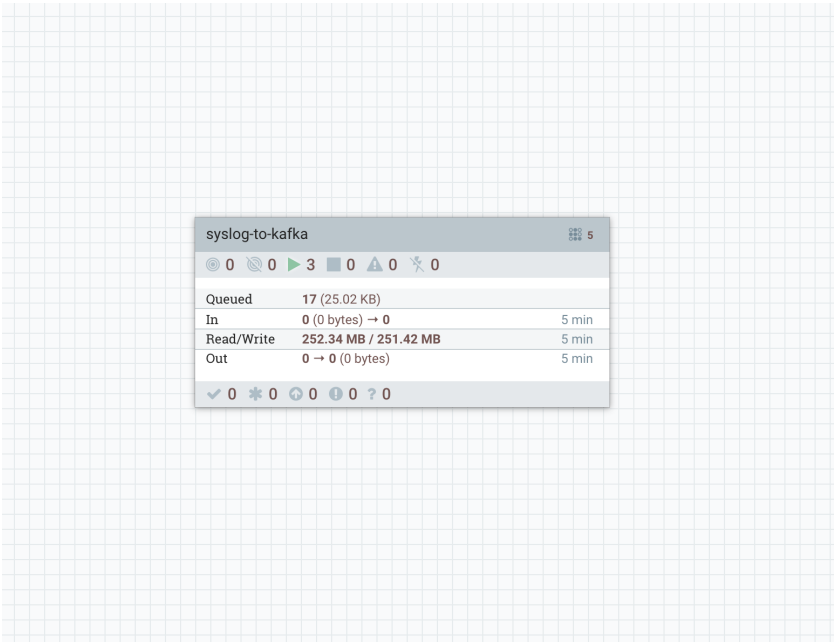
11. Click into your deployment and then click **Manage Deployment** on the top right to view your flow in NiFi.



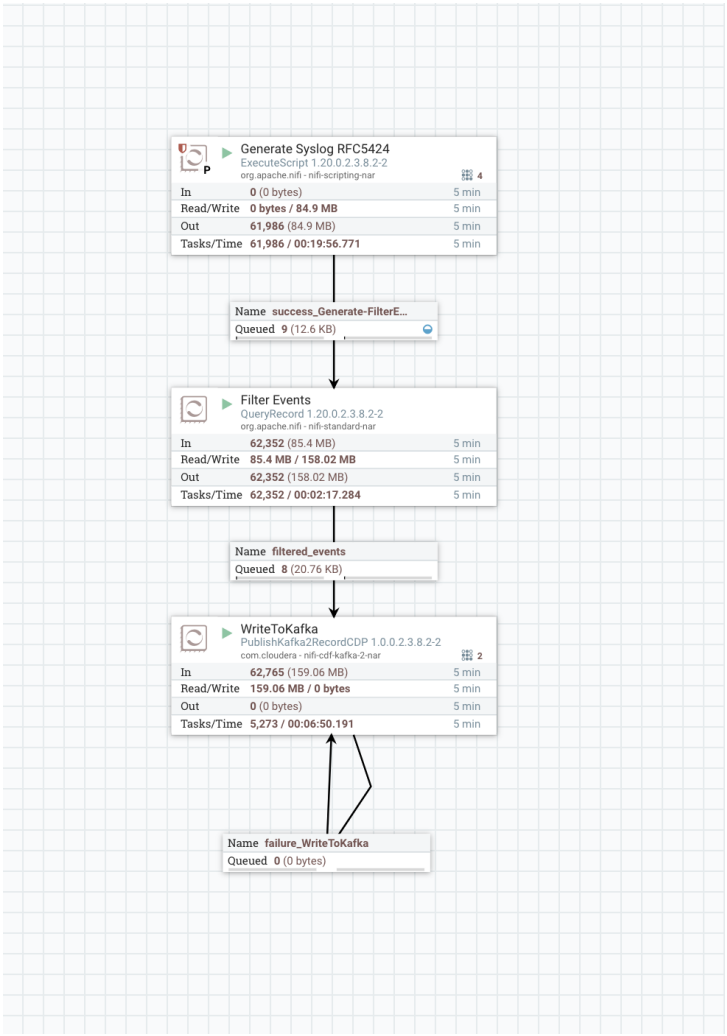
Now click on **ACTIONS** and select ***View in NiFi***



The flow that you just deployed will look something like this on NiFi



Double click on the Process Group to see the flow


Generate Syslog RFC5424
ExecuteScript 1.20.0.2.3.8.2-2
org.apache.nifi - nifi-scripting-nar
4
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 84.9 MB 5 min
Out 61,986 (84.9 MB) 5 min
Tasks/Time 61,986 / 00:19:56.771 5 min
Name success_Generate-FilterE...
Queued 9 (12.6 KB)
Filter Events
QueryRecord 1.20.0.2.3.8.2-2
org.apache.nifi - nifi-standard-nar
In 62,352 (85.4 MB) 5 min
Read/Write 85.4 MB / 158.02 MB 5 min
Out 62,352 (158.02 MB) 5 min
Tasks/Time 62,352 / 00:02:17.284 5 min
Name filtered_events
Queued 8 (20.76 KB)
WriteToKafka
PublishKafka2RecordCDP 1.0.0.2.3.8.2-2
com.cloudera - nifi-cdf-kafka-2-nar
2
In 62,765 (159.06 MB) 5 min
Read/Write 159.06 MB / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 5,273 / 00:06:50.191 5 min
Name failure_WriteToKafka
Queued 0 (0 bytes)

Lab 4 : SQL Stream Builder

1. Overview

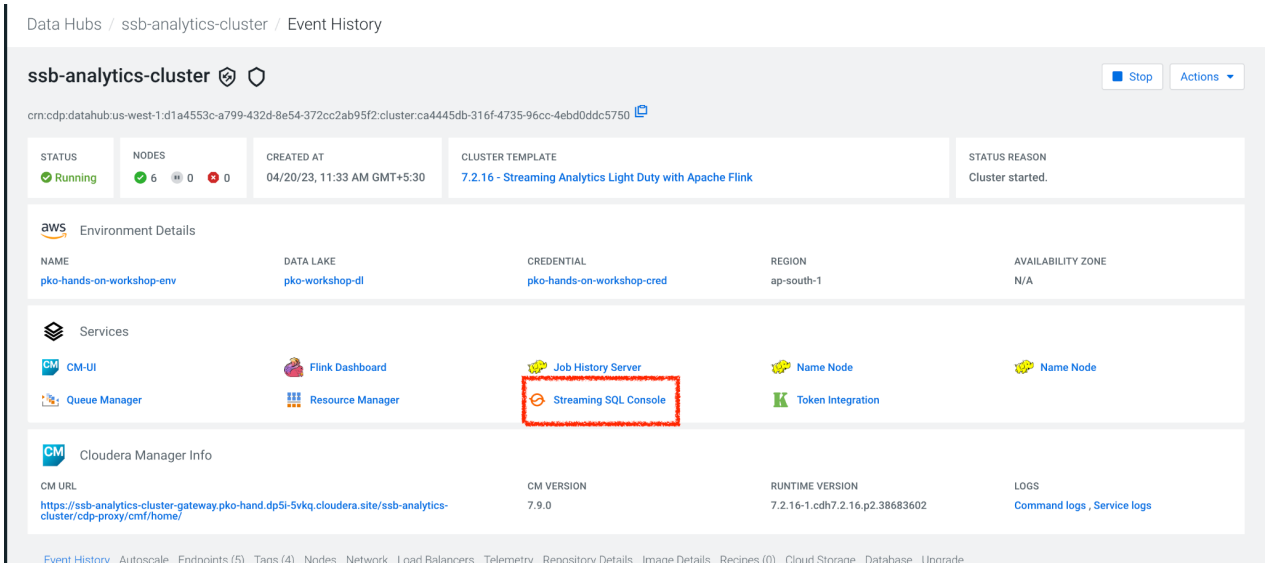
The purpose of this workshop is to demonstrate streaming analytic capabilities using Cloudera SQL Stream Builder. We will leverage the NiFi Flow deployed in CDF-PC from the previous lab and demonstrate how to query live data and subsequently sink it to another location. The SQL query will leverage the existing syslog schema in Schema Registry.

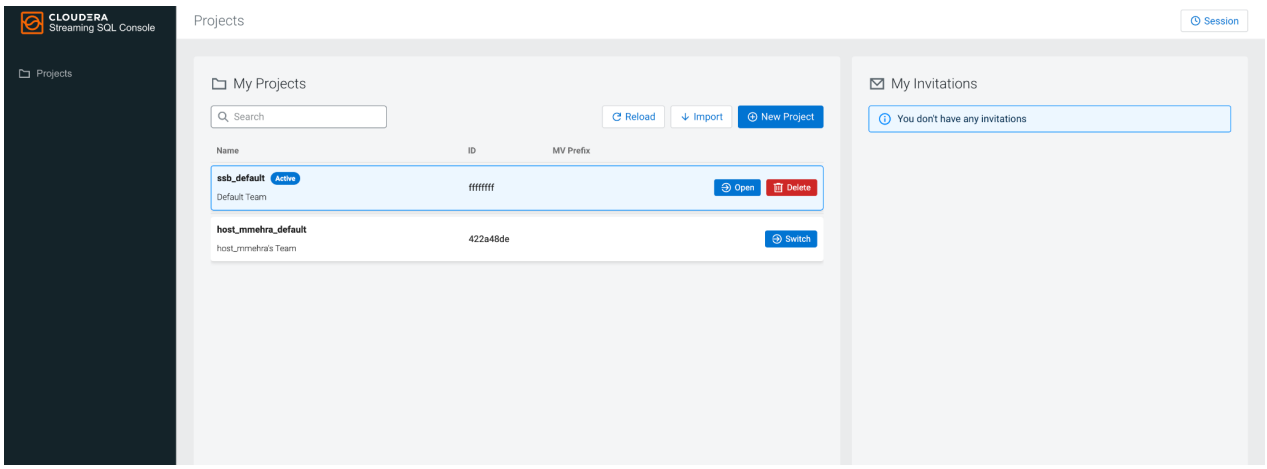
2. Creating a Project

Step 1: Go to the SQL Stream Builder UI

SSB Interface can be reached from the Data Hub that is running the Streams Analytics, in our case - *ssb-analytics-cluster*

Within the Data Hub, click on **Streaming SQL Console**



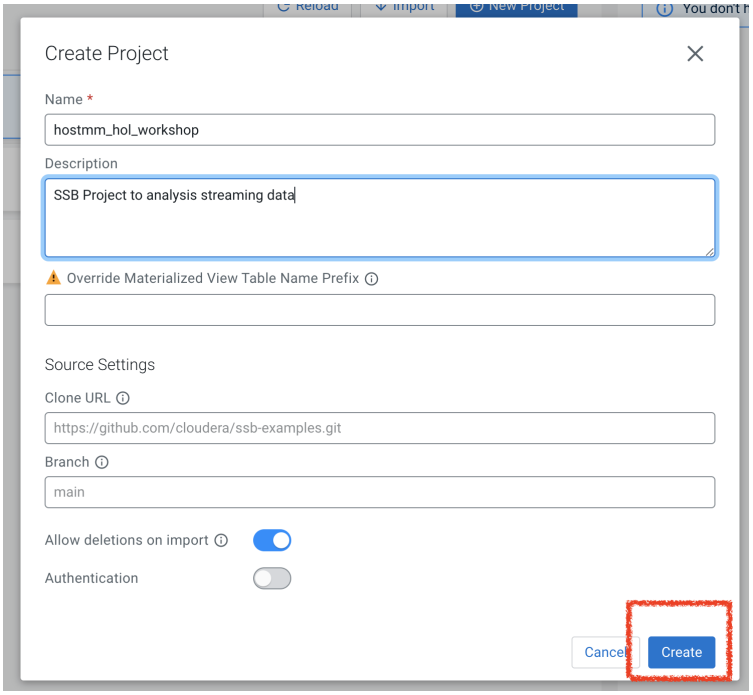


Step 2: Creation of a Project

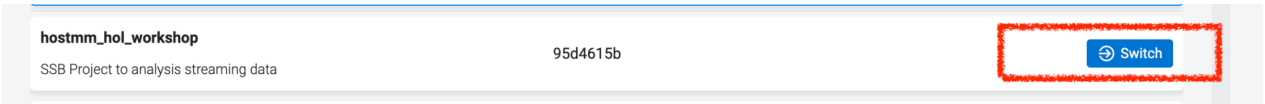
Create a SSB Project by clicking **"New Project"** using the following details and click "**Create**"

Name : {user-id}_hol_workshop

Description : SSB Project to analyze streaming data

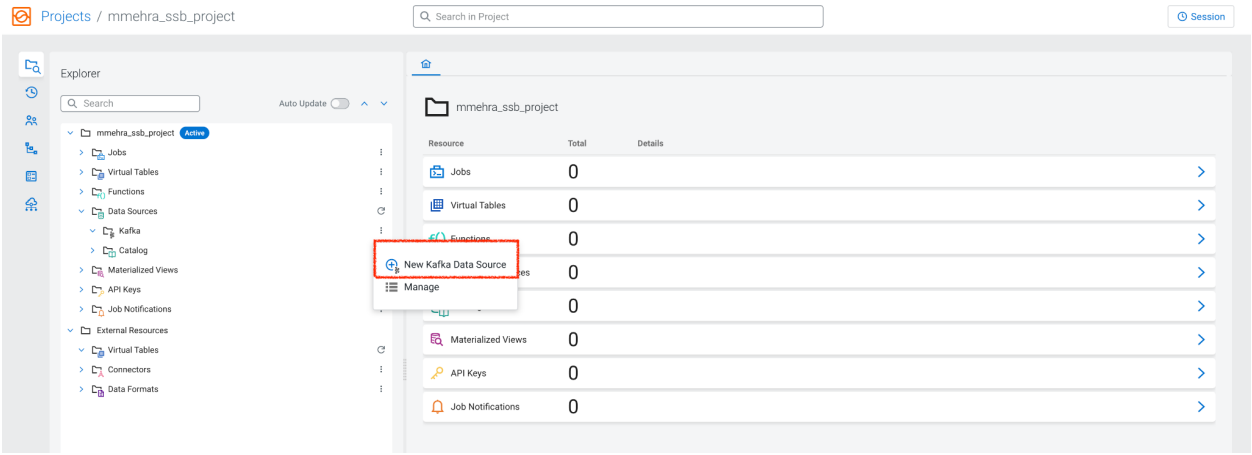


Switch to the created project. Click on **Switch**



Step 3 : Create Kafka Data Store

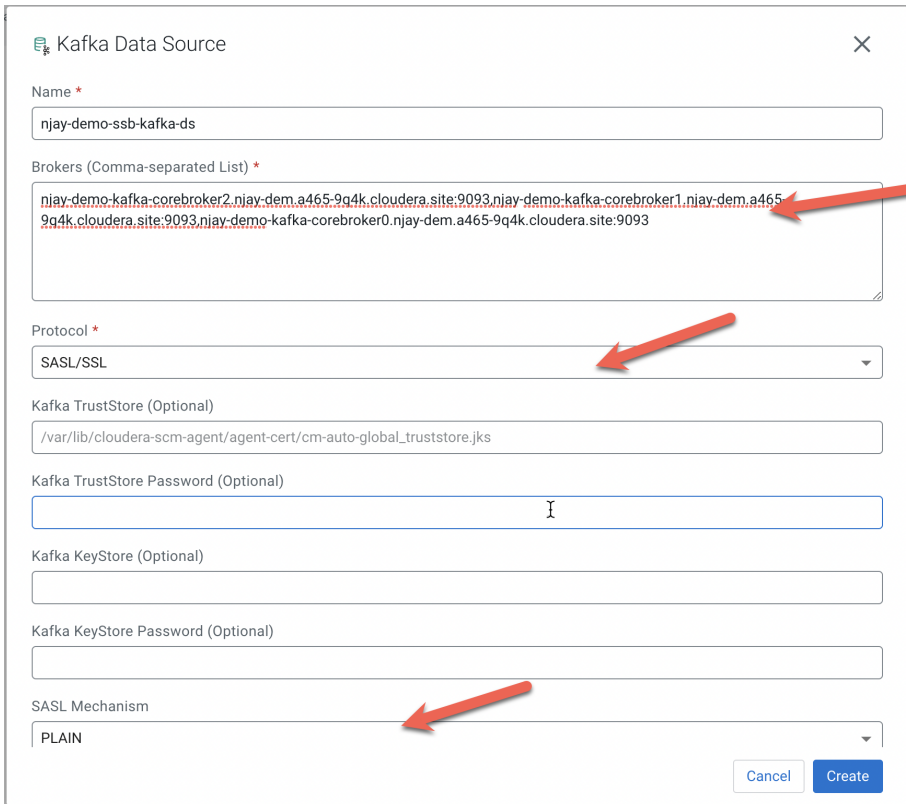
Create Kafka Data Store by selecting “**Data Sources”** in the left pane, clicking on the three-dotted icon next to “**Kafka”**, then selecting **“New Kafka Data Source”**.

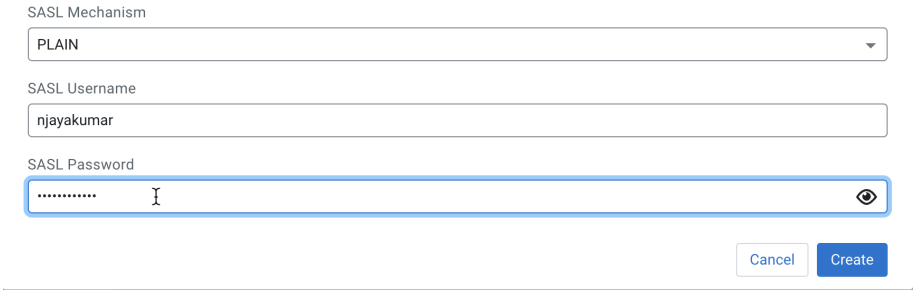


Name : {user-id}_cdp_kafka
Brokers (Comma-separated List)

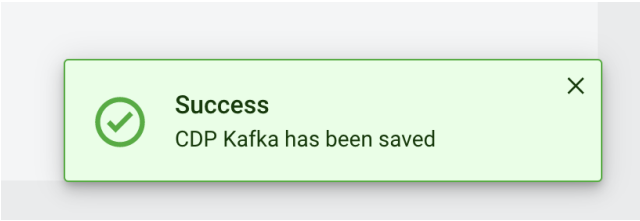
workshop-asean-streams-mess-corebroker2.workshop.dp5i-5vkq.cloudera.site:9093,workshop-asean-streams-mess-corebroker0.workshop.dp5i-5vkq.cloudera.site:9093,workshop-asean-streams-mess-corebroker1.workshop.dp5i-5vkq.cloudera.site:9093

Protocol : SASL/SSL
SASL Username : <workload-username>
Example : workshop001
SASL Password : <Set in Lab 0 Section 3> P@ssw0rd@2023
SASL Mechanism : PLAIN



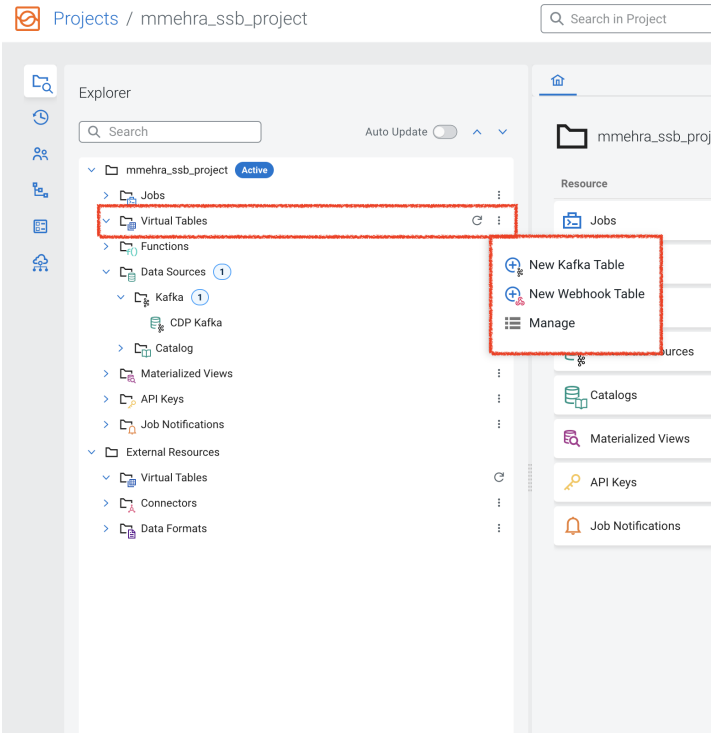


Click on **VALIDATE** to test the connections once successful click on **CREATE**



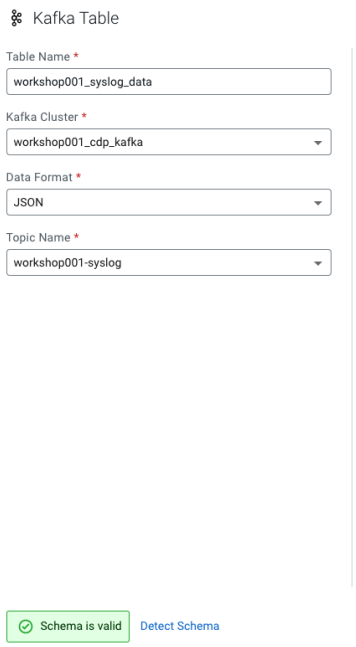
Step 4: Create Kafka Table

Create Kafka Table, by selecting **“Virtual Tables”** in the left pane, clicking on the three-dotted icon next to it, then clicking on **“New Kafka Table”**.



Step 5: Configure the Kafka Table

1. Enter the following details in the Kafka Table dialog box:
    - Table Name: **{user-id}_syslog_data**
    - Kafka Cluster: **<select the Kafka data source you created previously>**
    - Data Format: **JSON**
    - Topic Name: **<select the topic created in Schema Registry>**

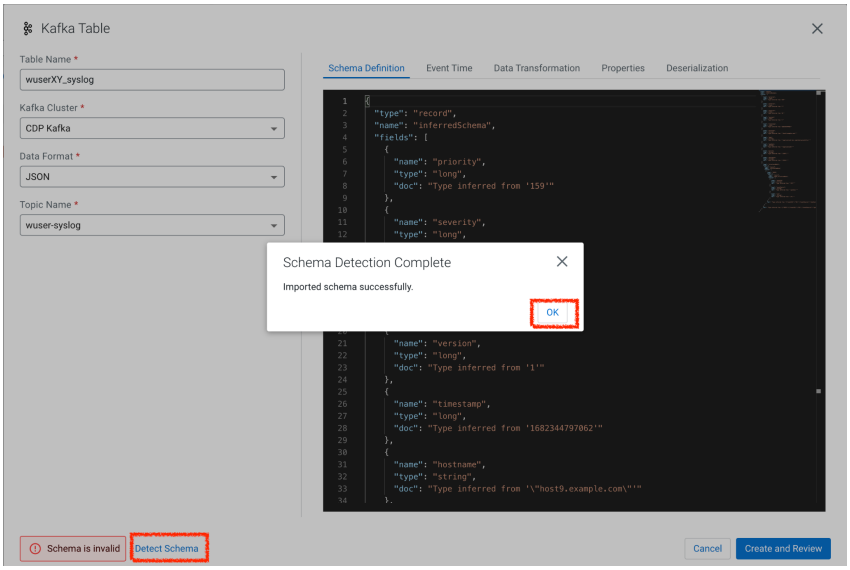


2. When you select Data Format as AVRO, you must provide the correct Schema Definition when creating the table for SSB to be able to successfully process the topic data.

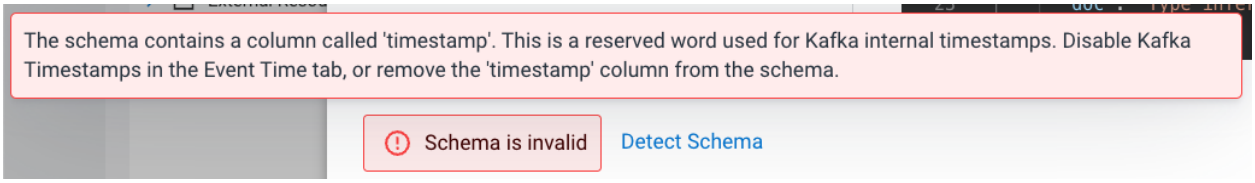
    For JSON tables, though, SSB can look at the data flowing through the topic and try to infer the schema automatically, which is quite handy at times. Obviously, there must be data in the topic already for this feature to work correctly.

    Note: SSB tries its best to infer the schema correctly, but this is not always possible and sometimes data types are inferred incorrectly. You should always review the inferred schemas to check if it's correctly inferred and make the necessary adjustments.

    Since you are reading data from a JSON topic, go ahead and click on **Detect Schema** to get the schema inferred. You should see the schema be updated in the **Schema Definition** tab.



3. You will also notice that a "Schema is invalid" message appears upon the schema detection. If you hover the mouse over the message it shows the reason:

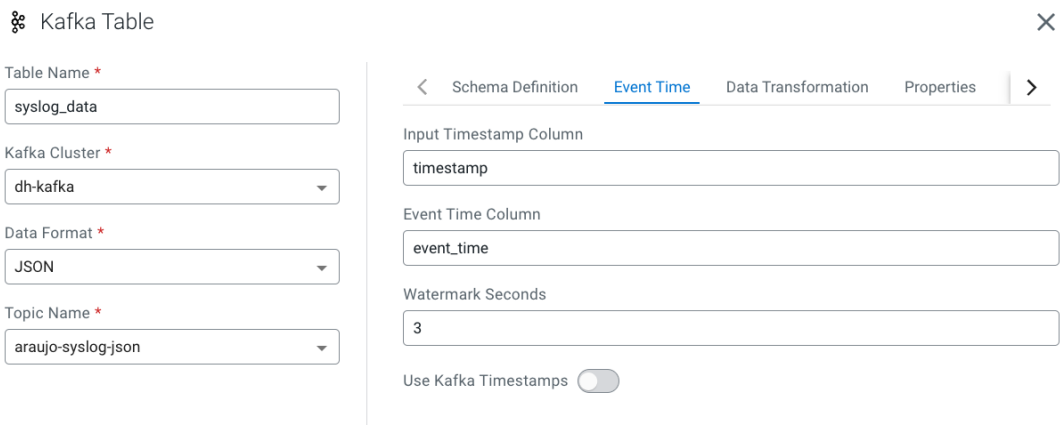


   You will fix this in the next step.

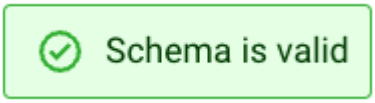
4. Each record read from Kafka by SSB has an associated timestamp column of data type TIMESTAMP ROWTIME. By default, this timestamp is sourced from the internal timestamp of the Kafka message and is exposed through a column called eventTimestamp.

   However, if your message payload already contains a timestamp associated with the event (event time), you may want to use that instead of the Kafka internal timestamp.

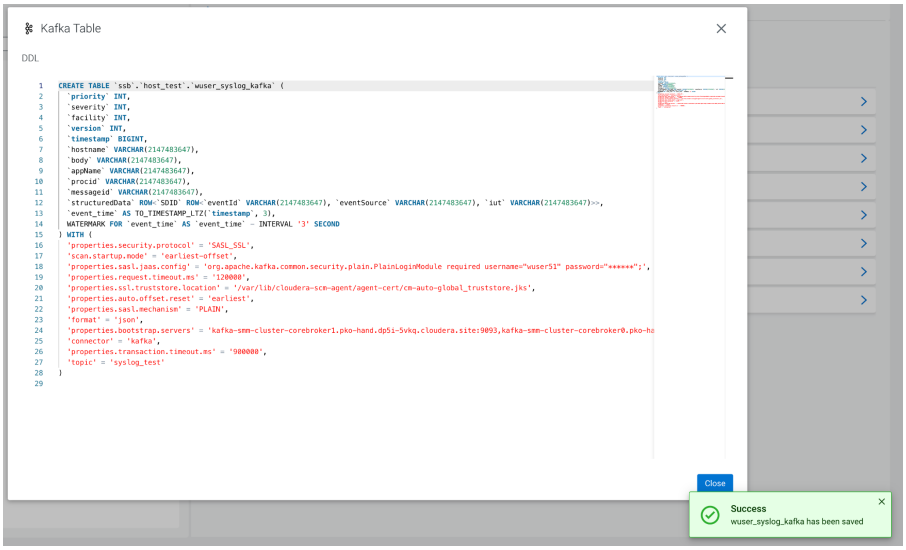
   In this case, the syslog message has a field called "**timestamp**" that contains the timestamp you should use. You want to expose this field as the table's "**event_time**" column. To do this, click on the Event Time tab and enter the following properties:
   - Use Kafka Timestamps: **Disable**
   - Input Timestamp Column: **timestamp**
   - Event Time Column: **event_time**
   - Watermark Seconds: **3**



5. Now that you have configured the event time column, click on **Detect Schema** again. You should see the schema turn valid:



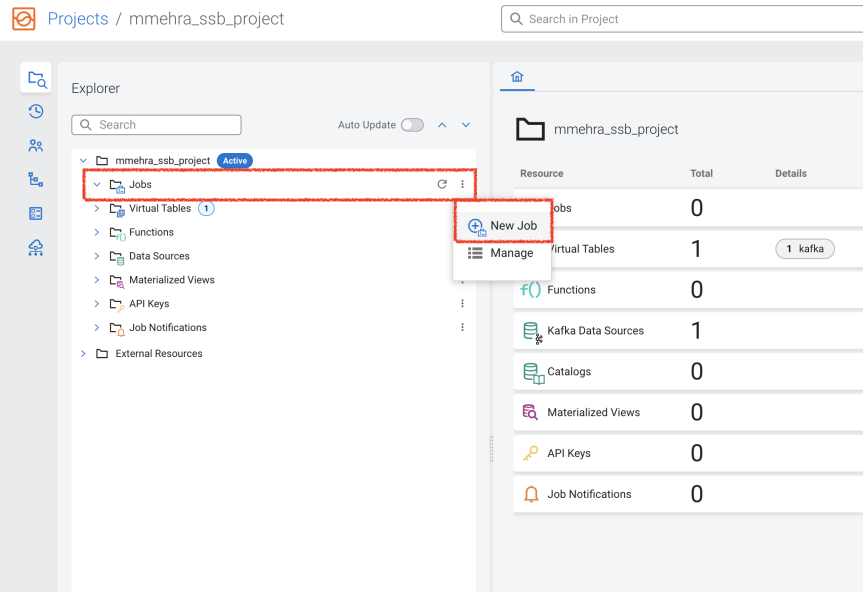
6. Click the **Create and Review** button to create the table.



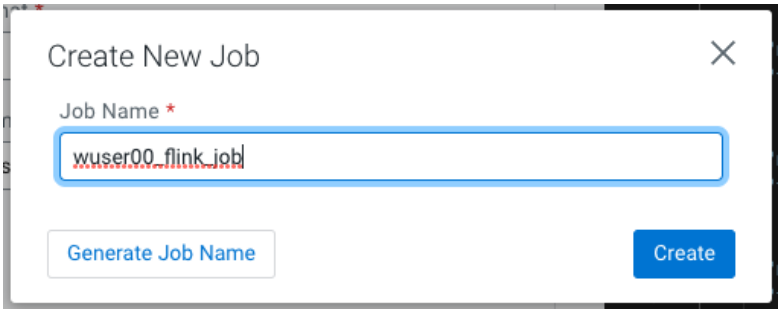
Review the table's DDL and click **Close**.

Step 6: Create a Flink Job

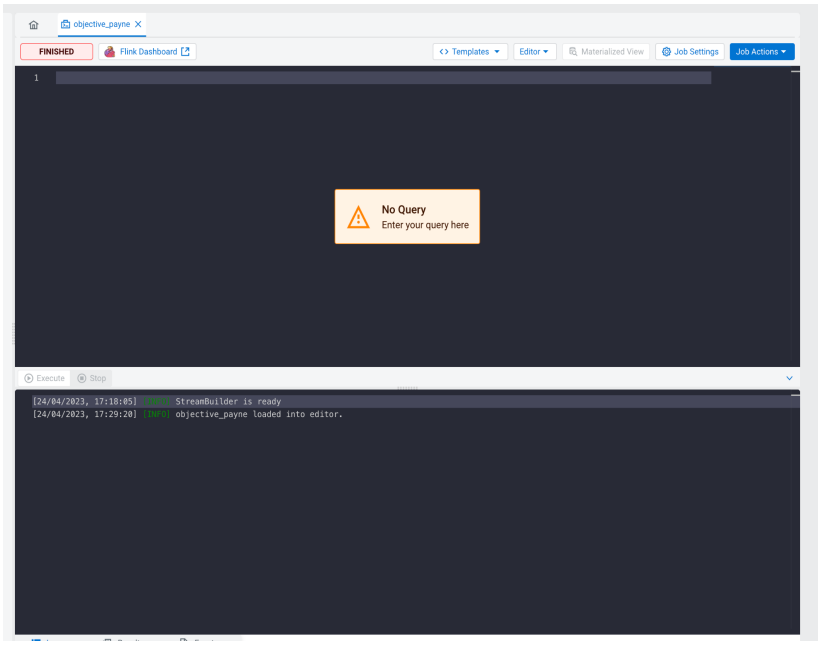
Create a Flink Job, by selecting **“Jobs”** in the left pane, clicking on the three-dotted icon next to it, then clicking on **“New Job”**.



Give a job name and click **CREATE**



The Query Editor should now show up

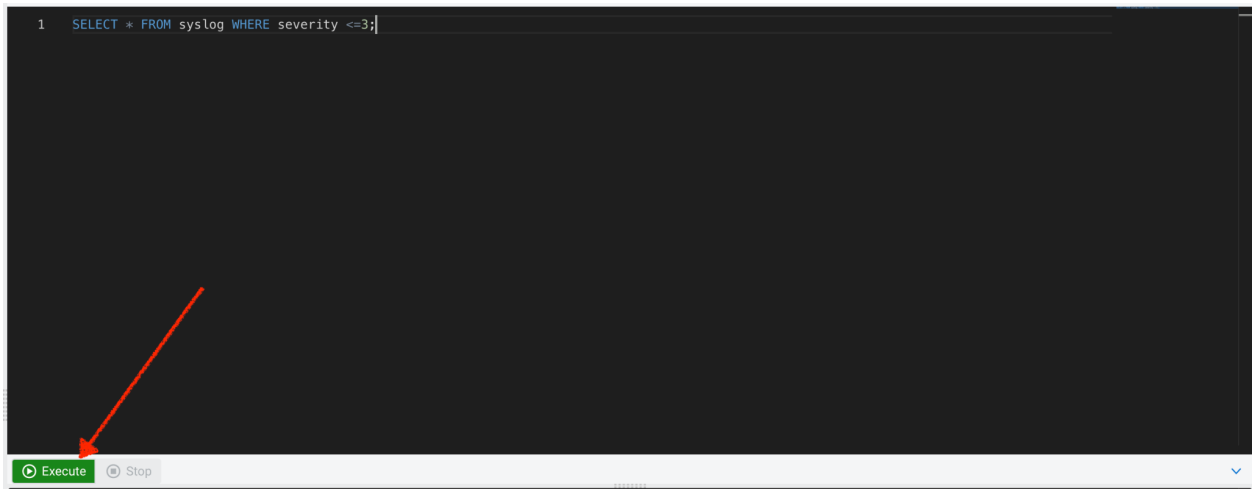


Add the following SQL Statement in the Editor

SELECT * FROM **{user-id}_syslog_data** WHERE severity <=3

NOTE : Replace {user-id} with your assigned username

Run the Streaming SQL Job by clicking **Execute**. Also, ensure your {user_id}-syslog-to-kafka flow is running in CDF-PC.



In the Results tab, you should see syslog messages with severity levels <=3