Informatica™

BDM Training 2018

BDM, BDS Lab Guide

BDM 10.2.1 on AWS HDP 2.6

Partner Solution Consultants team

Table of content

1. Introduction

1.1 Purpose

This lab guide provides information on how to quickly demo the following as a “take away” on BDM 10.2.1:

- Lab00: BDM: Unit test HDFS pass-through mapping in native, hive, blaze & spark modes
- Lab01: Show new 10.2.1 MIS (Mass Ingestion Service) capability
- Lab02: dynamic mapping unique capability (oracle to HIVE in Blaze hadoop push down)
- Lab03: Multiple Ingestion from Oracle to Hive in native, jdbc, sqoop in native, hive & blaze
- Lab04: Load & Profile a hive table in native and hadoop modes
- Lab05: cleansing capabilities in hadoop (blaze) mode
- Lab06: masking capabilities in hadoop (blaze) mode
- Lab07: H2R capabilities parsing complex files in Blaze mode
- Lab08: Intelligent Structure Discovery & parsing complex files in Spark mode
- Lab09: Show AWS RedShift & S3 integration in native & push down
- Lab10a/b: Realtime data collection with EDS and distributed processing with BDS

1.2 Prerequisites

- BDM 10.2.1 Client & Server installed & configured
- CCO, HDFS, HIVE, HADOOP; HBASE as (default) connection names
- Informatica Domain is started and all Informatica services are running (for BDM)
- Hadoop Platform is up and Hadoop services (Zookeeper, HDFS, Hive, YARN, HBase) are running
- EDS 2.3.2 installed and configured
- Elastic Search 6.2.x and Kibana 6.2.x installed and running on the server machine
- Python 3 installed on the server machine, with the following modules:
  - Kafka
  - cx_Oracle (http://cx-oracle.readthedocs.io/en/latest/installation.htmlmodule)

1.3 Before you Start

The labs will use the 10.2.1 Developer but also following interfaces you should open upfront:

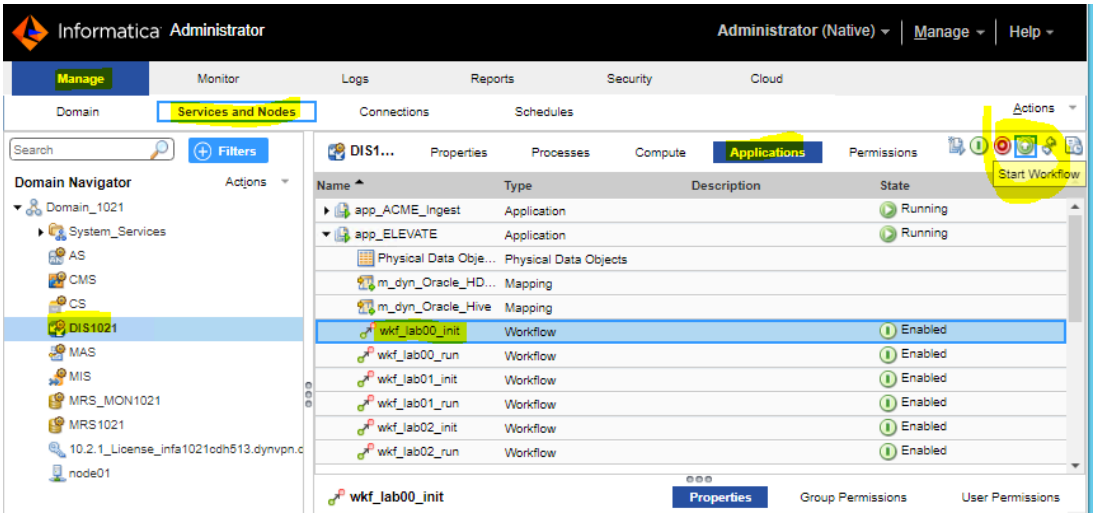
1.3.1 Informatica Administration Console

The **Informatica Administration Console** will be used to **start workflows**

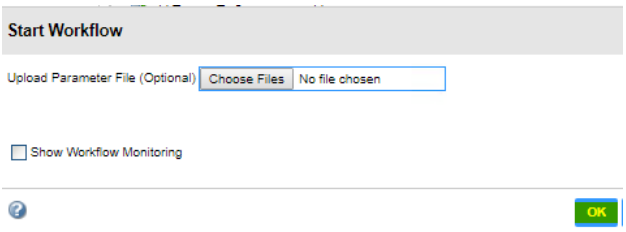
You will be asked LATER in the labs to start a workflow, to do so:

- Go to http://bdm.localdomain:6008/administrator and log in as Administrator/infa

- Navigate to Manage > Services and Nodes > DIS 1021 > Applications



- select one then click the Start Workflow icon (see above)
- leave default (we will no use parameter file) and click OK

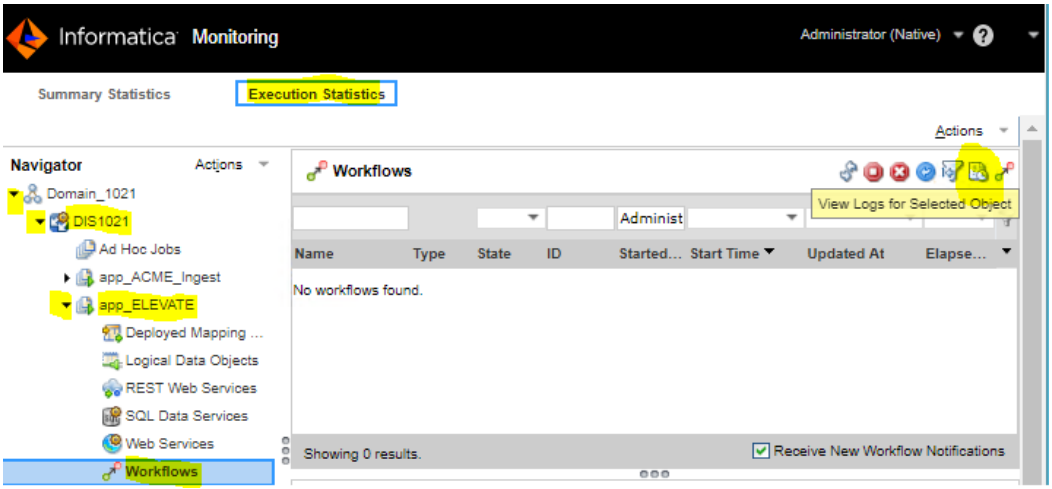


1.3.2 Informatica Monitoring Console

The **Informatica Monitoring Console** will be used to **monitor workflows & mappings**

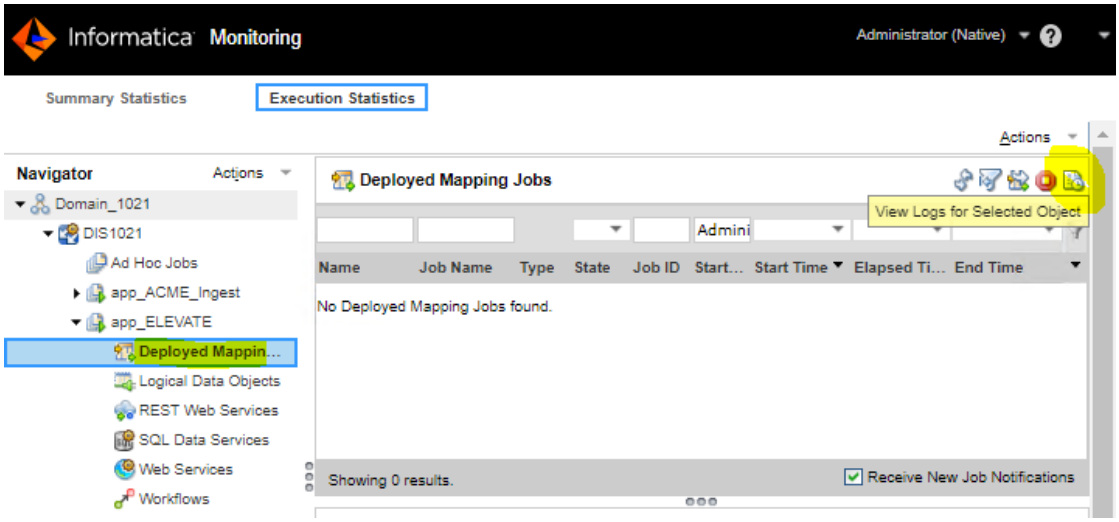
You will LATER use it in the labs to monitor workflows, to do so:

- Go to http://bdm.localdomain:6008/monitoring and log in as Administrator/infa
- Navigate to Execution Statistics > DIS 1021 > app_ELEVATE > Workflows



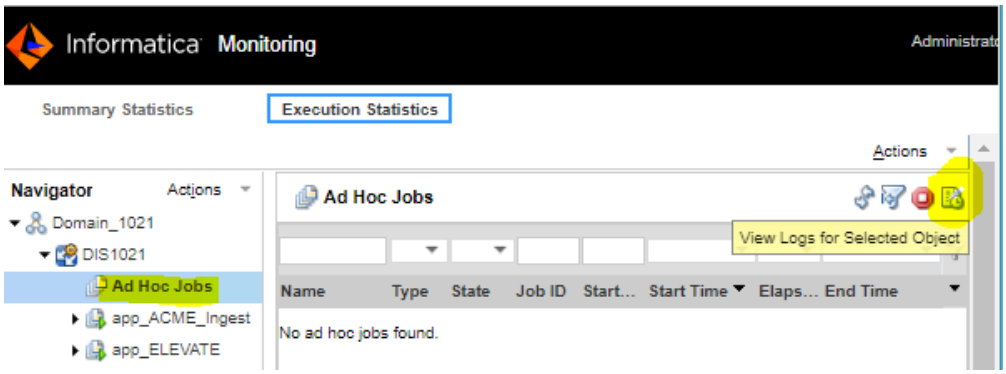
You will be able from there to view workflow status
Note: you can get logs from there using the icon as shown above

- Navigate to Execution Statistics > DIS 1021 > app_ELEVATE > Deployed mappings



You will be able from there to view the dynamic mapping status
Note: you can get logs from there using the icon as shown above

- Navigate to Execution Statistics > DIS 1021 > app_ELEVATE > Deployed mappings
- Navigate to Execution Statistics > DIS 1021 > AdHoc Jobs



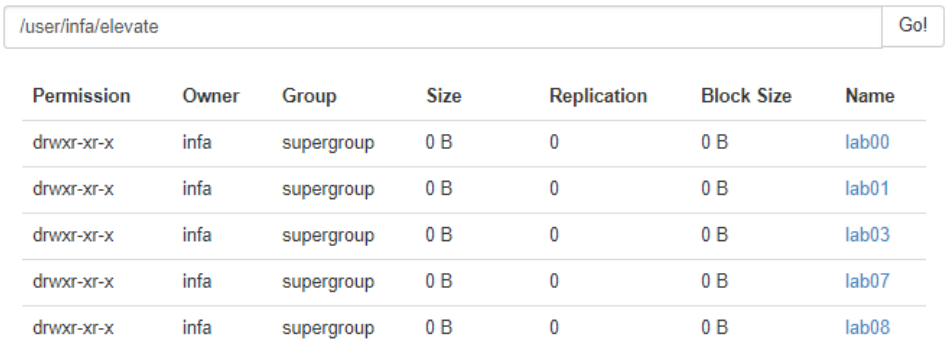
You will be able from there to view the mapping status you ran from Developer tool
Note: you can get logs from there using the icon as shown above

1.3.3 HDFS Web Browser

The **Hadoop HDFS Web Console** will be used to **view HDFS files & Hive tables**

You will LATER use it in the labs **to see HDFS outputs**, to do so:

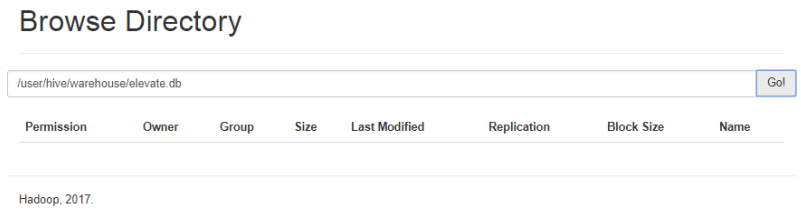
Go to http://bdm.localdomain:50070/explorer.html#/user/infa/elevate



Then browse to the specific lab folder to see result files

You will LATER also use it in the labs **to see HIVE outputs**, to do so:

Go to http://bdm.localdomain:50070/explorer.html#/user/hive/warehouse/elevate.db



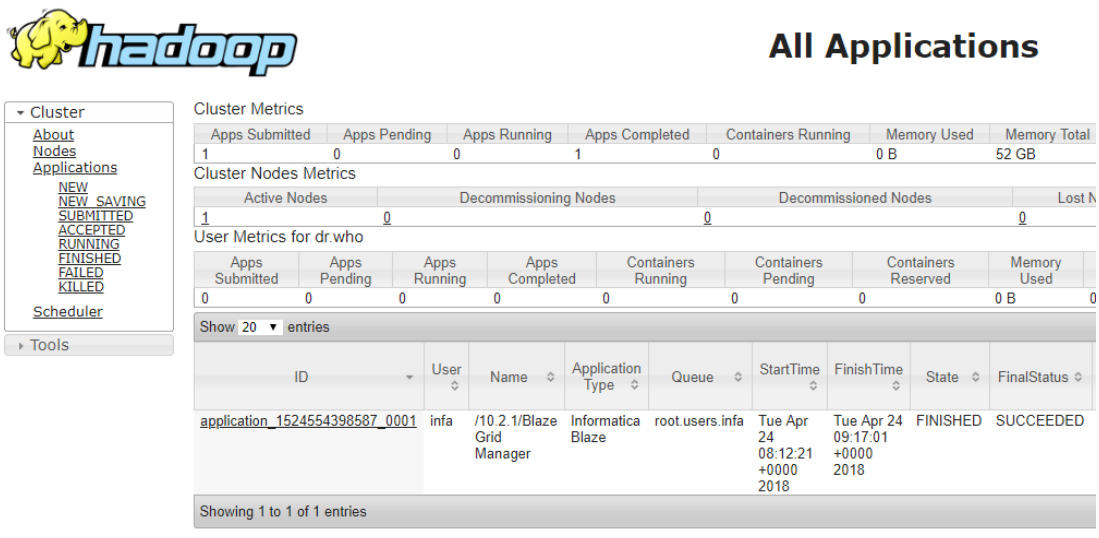
You will see there the hive table created.

1.3.4 YARN Monitor Console

The **YARN Monitor Web Console** will be used to **view BDM pushed down YARN executions**

You will LATER use it in the labs **to see mapreduce and spark jobs**, to do so:

Go to http://bdm.localdomain:8088/cluster/apps



From there you will be able to browse to yarn application logs etc.

1.3.5 Blaze Monitor Console

The **Blaze Monitoring Console** will be used to **view BDM Blaze executions jobs**

You will LATER use it in the labs **to see Blaze execution jobs**, to do so:

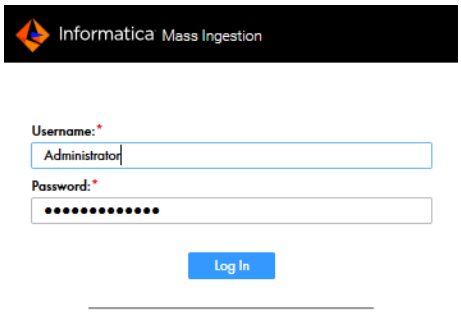
Go to http://bdm.localdomain:9080/Blaze

1.3.6 Informatica Mass Ingestion Service

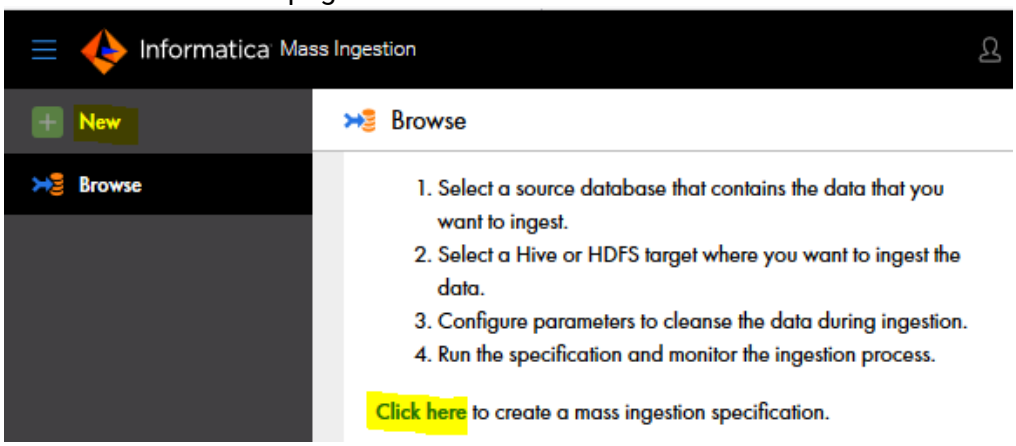
The **Informatica Mass Ingestion Service** will be used to **mass ingest data**

You will be asked LATER in the labs to create Mass Ingestion Services, to do so:

Go to http://bdm.localdomain:9050/mi/login and log in as Administrator/infa



You should see initial page



To create LATER a MIS project, you will either click “NEW” or “Click here”

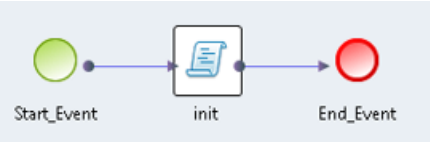
2. Lab00 - Unit test BDM abstraction layer

2.1 Purpose

Unit test and run HDFS pass-through mapping in native, hive, blaze and spark modes

2.2 Initialize Lab

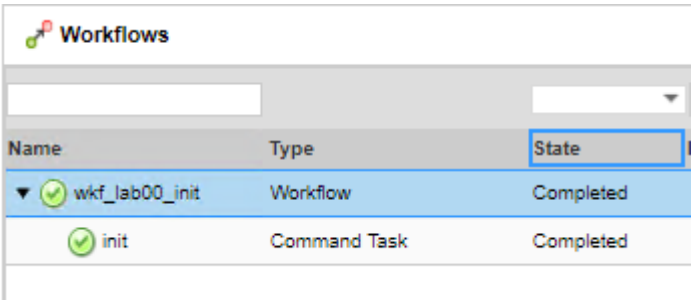
Workflow **wkf_lab00_init** calls a script to reinitialize the lab



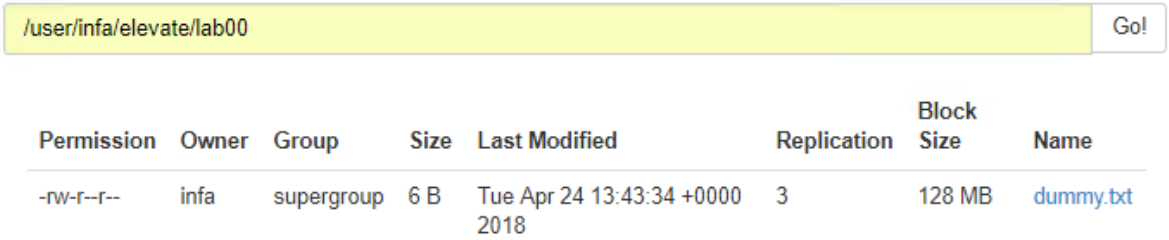
The **init** task command will create HDFS dummy file, clean target files

To initialize the lab, go to Informatica Administration Console and run the workflow

You should see in Informatica Monitoring Console:

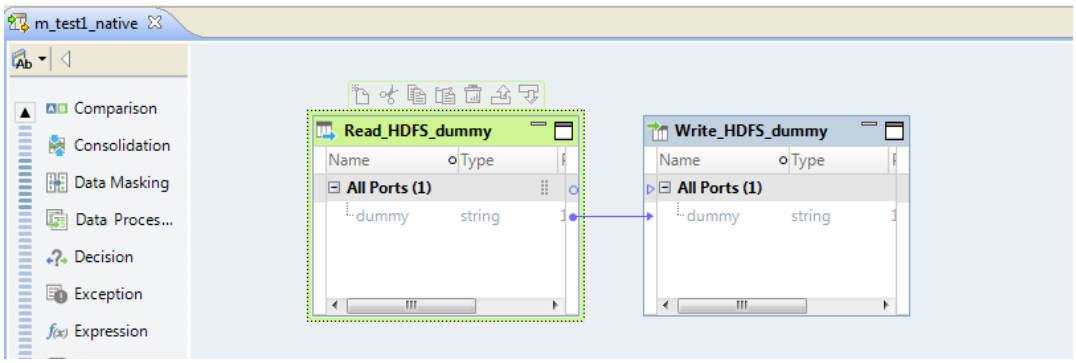


Use HDFS Web Browser and navigate to /user/infa/elevate/lab00, you should see:



2.3 Review Lab Content

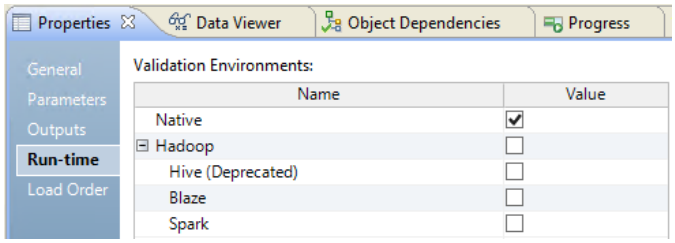
Open Developer, go to ELEVATE/BDM_Labs/lab00_* folder and open the m_test1_native mapping. You should see:



You can run preview on the source:

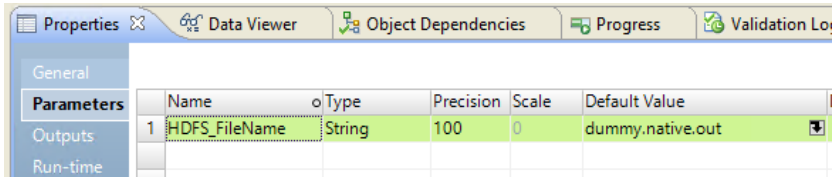
	dummy
1	dummy

This mapping is only validated against Native mode:



It is set for native mode Execution:

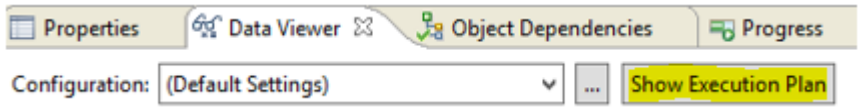
This mapping use file name parameter:



For hive, blaze and spark mappings:

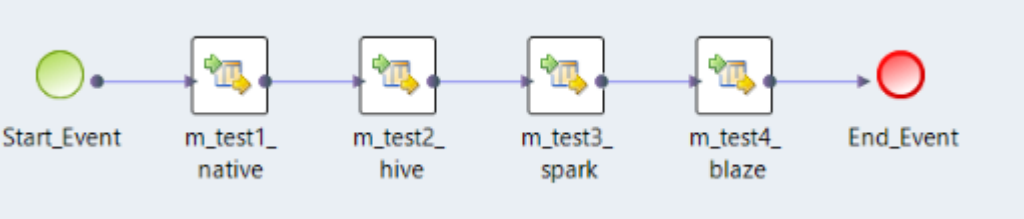
Review parameter and Execution Validation & Environment

Click on Data Viewer, '**Show Execution Plan**' to see code sent to YARN at run time.



2.4 Run Lab

Workflow **wkf_lab00_run** runs all unit test one after another



To run unit test in native, hive, spark, blaze go to Informatica Administration Console and run the workflow

You should see in Informatica Monitoring Console:

wkf_lab00_run	Workflow	Completed
m_test4_blaze	Mapping Task	Completed
m_test3_spark	Mapping Task	Completed
m_test2_hive	Mapping Task	Completed
m_test1_native	Mapping Task	Completed

You should see in YARN Monitor Console:

Show 20 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus
application_1524554398587_0004	infa	/10.2.1/Blaze Grid Manager	Informatica Blaze	root.users.infa	Tue Apr 24 13:54:48 +0000 2018	N/A	RUNNING	UNDEFINED
application_1524554398587_0003	infa	m_test3_spark	SPARK	root.users.infa	Tue Apr 24 13:53:51 +0000 2018	Tue Apr 24 13:54:24 +0000 2018	FINISHED	SUCCEEDED
application_1524554398587_0002	infa	INSERT OVERWRITE TABL...s_dummy_m_test2_hive(Stage-0)	MAPREDUCE	root.users.infa	Tue Apr 24 13:53:10 +0000	Tue Apr 24 13:53:28 +0000 2018	FINISHED	SUCCEEDED

You see in Blaze Monitor Console:

Name	Start Time	End Time	Elapsed Time	State	Grid Segment	Running	Succeeded	Failed
gtid-4-1-78133079-1	Tue Apr 24 2018 1:55:47 PM	Tue Apr 24 2018 1:55:56 PM	0:0:8	Succeeded	1	0	1	0

Use HDFS Web Browser and navigate to /user/infa/elevate/lab00, you should see:

/user/infa/elevate/lab00 Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	infa	supergroup	6 B	Tue Apr 24 13:53:28 +0000 2018	3	128 MB	dummy.hive.out-m-00000
-rw-r--r--	infa	supergroup	6 B	Tue Apr 24 13:52:54 +0000 2018	3	128 MB	dummy.native.out
-rw-r--r--	infa	supergroup	6 B	Tue Apr 24 13:54:24 +0000 2018	3	128 MB	dummy.spark.out-m-00000
-rw-r--r--	infa	supergroup	6 B	Tue Apr 24 13:43:34 +0000 2018	3	128 MB	dummy.txt
-rw-r--r--	infa	supergroup	6 B	Tue Apr 24 13:55:56 +0000 2018	3	128 MB	dummy1.blaze.out

3. Lab01 - PIM Mass Ingestion Service

3.1 Purpose

Show new 10.2.1 MIS (Mass Ingestion Service) capability

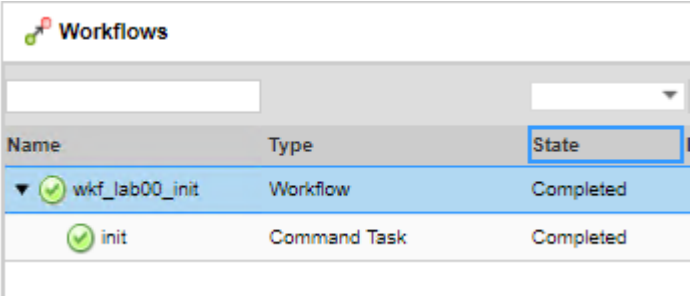
3.2 Initialize Lab

Workflow **wkf_lab01_init** calls a script to reinitialize the lab

The **init** task command will clean target HDFS folder

To initialize the lab, go to Informatica Administration Console and run the workflow

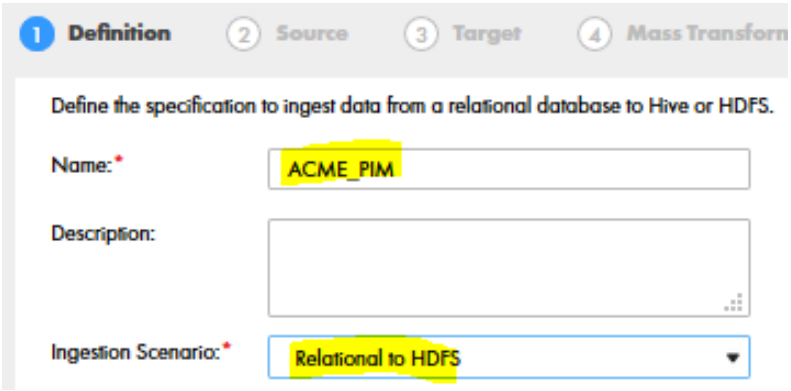
You should see in Informatica Monitoring Console:



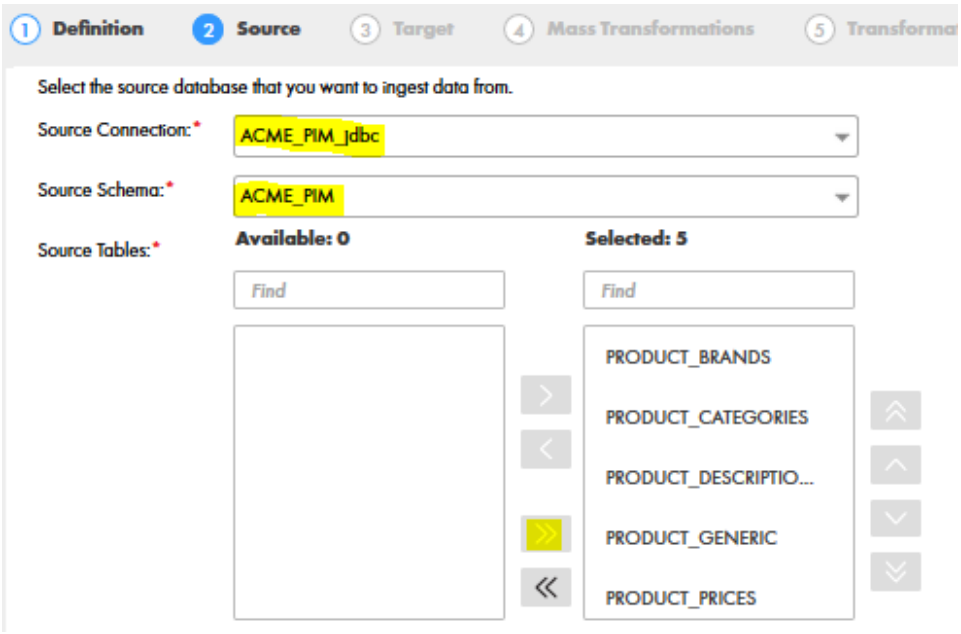
Use HDFS Web Browser and navigate to /user/infa/elevate/lab01: folder should be **empty**

3.3 Create MIS project

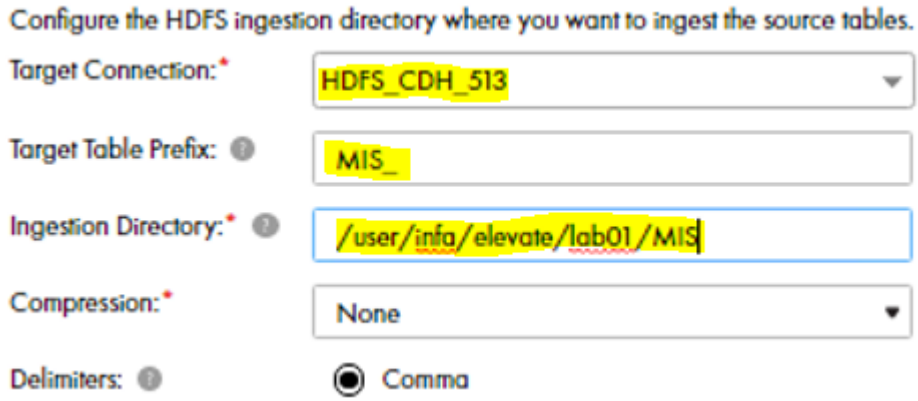
Go to Informatica Mass Ingestion Services and create a new project



Enter **ACME_PIM** as project Name, **select Relational to HDFS** and hit **Next**



Select **ACME_PIM_jdbc** as Source Connection, **ACME_PIM** as Source Schema
Then select all tables (**>>**) and hit **Next**



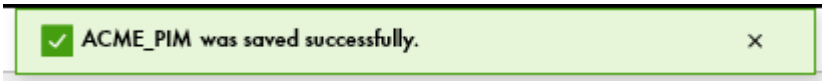
Select **HDFS** as Target Connection, enter **MIS_** as Target Prefix,
/user/infa/elevate/lab01/MIS **as Ingestion Directory**
Leave rest as default and hit **Next**

Leave all blank as default and hit **Next**

Ingestion Objects (5)

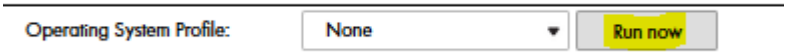
Sources	Filter By	Drop Columns	Trim	Convert to Upp...	Convert to Low...	Replace Columns
PROD...						
PROD...						
PROD...						
PROD...						
PROD...						

Leave all blank as default and hit **Save**



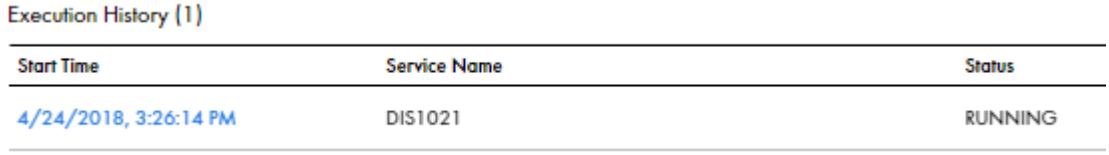
3.4 Deploy & Run MIS project

Select the **DIS** and the **HADOOP** connection and hit **Deploy**

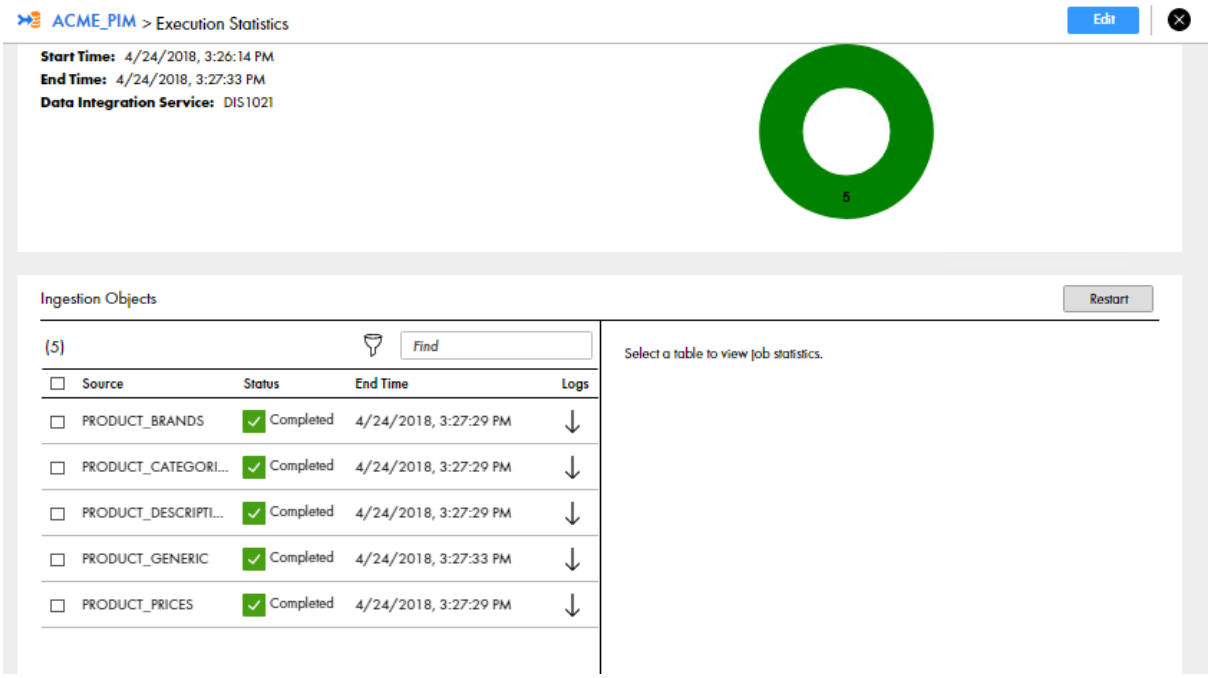


Once deployed, Hit **Run Now**

A RUNNING task should appear below Execution History

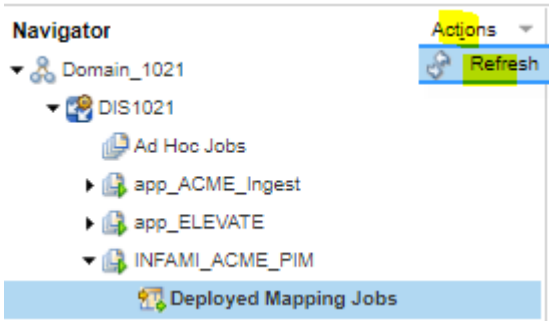


Hit the link, you should see:



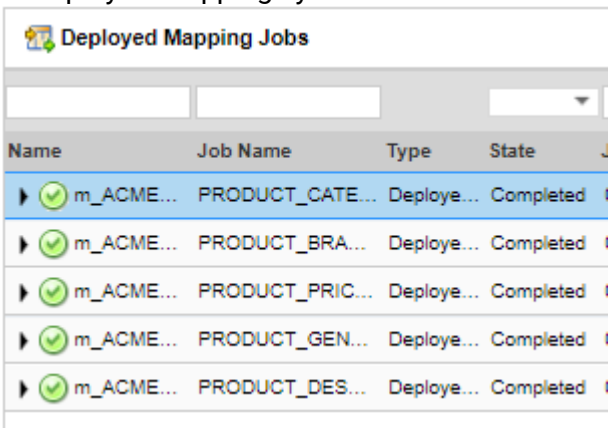
After a while you will see successful completion tasks

In Informatica Monitoring you can see application INFAMI_PIM_CRM was deployed:



You may need to refresh the applications

In Deployed mappings you should see:



Use HDFS Web Browser and navigate to /user/infa/elevate/lab01/MIS, you should see:

/user/infa/elevate/lab01/MIS Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	infa	supergroup	0 B	Tue Apr 24 15:27:28 +0000 2018	0	0 B	MIS_PRODUCT_BRANDS
drwxr-xr-x	infa	supergroup	0 B	Tue Apr 24 15:27:28 +0000 2018	0	0 B	MIS_PRODUCT_CATEGORIES
drwxr-xr-x	infa	supergroup	0 B	Tue Apr 24 15:27:28 +0000 2018	0	0 B	MIS_PRODUCT_DESCRIPTIONS
drwxr-xr-x	infa	supergroup	0 B	Tue Apr 24 15:27:33 +0000 2018	0	0 B	MIS_PRODUCT_GENERIC
drwxr-xr-x	infa	supergroup	0 B	Tue Apr 24 15:27:28 +0000 2018	0	0 B	MIS_PRODUCT_PRICES

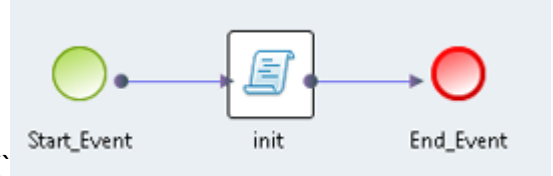
3.5 Extra Content (Optional)

3.5.1 Purpose

Show dynamic mapping unique capability (oracle to HDFS in Blaze hadoop push down)

3.5.2 Initialize Lab

Workflow **wkf_lab01_init** calls a script to reinitialize the lab

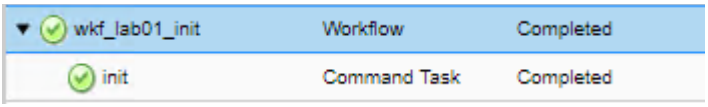


X`

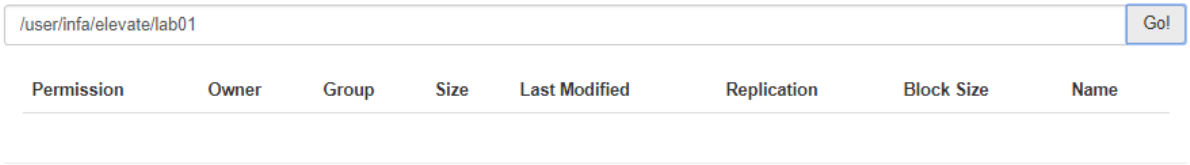
The init task command will clean all HDFS target data

To initialize the lab, go to Informatica Administration Console and run the workflow

You should see in Informatica Monitoring Console:



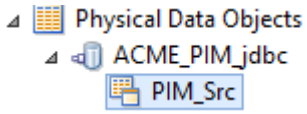
Use HDFS Web Browser and navigate to /user/infa/elevate/lab01, you should see:



3.5.3 Review Lab Content

In Developer, connect to MRS and open folder ELEVATE/BDM_Labs/lab01_*

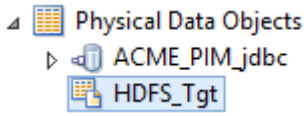
Observe the source object:



The **source** 'PIM_Src' uses 'ACME_PIM_jdbc' connection and 'fake' columns:

	Name	Native Type	Precisi...	Scale	Primar...	Nullable
1	dummyString	varchar	10	0	☐	☐
2	dummyString1	varchar	10	0	☐	☐
3	dummyString2	varchar	10	0	☐	☐

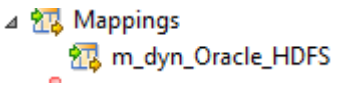
Observe the **target** object:



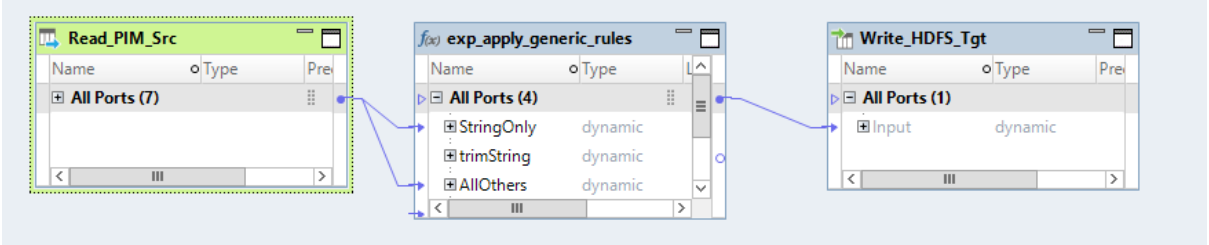
Open the Object and see in Advanced > Run-time Write output folder and parameter filename

Output file directory	/user/infa/elevate/lab01
Output file name	HDFS_Filename (Parameter)

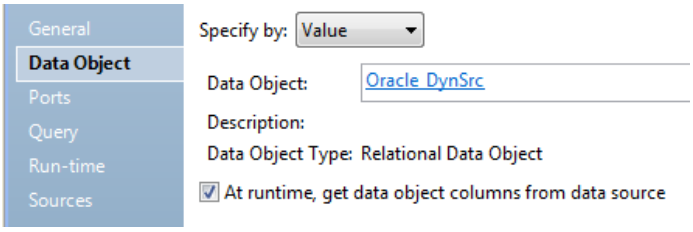
Observe the mapping:



The **mapping** is using dynamic ports



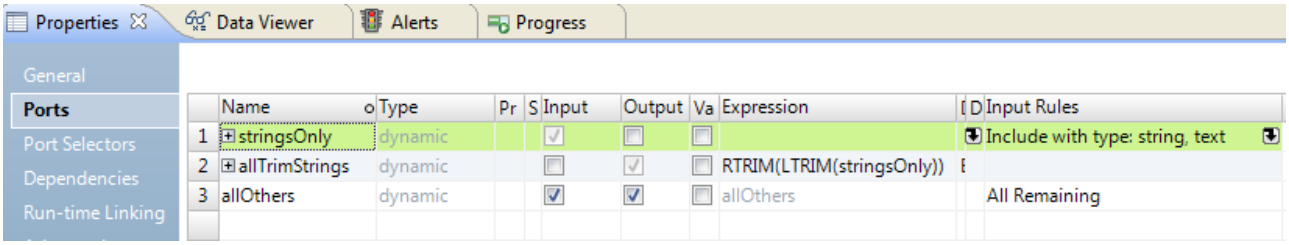
In mapping **Read Data Object** observe that it uses 'At runtime, get data object columns from data source'



In mapping **Read** Object '**Run-time**' properties observe that it uses **parameter** for **table name**

Name	Value
Connection	ACME_PIM_jdbc
Owner	
Resource	ORCL_TableName (Parameter)

In mapping Expression observe the port configuration



Input port 'stringsOnly' is defined as dynamic, input only and includes all string and text ports

Output port 'allTrimStrings is defined as dynamic, output only, doing RTRIM and LTRIM on port 'stringsOnly'

Output port 'allOthers" is defined as dynamic, both input and output and include all remaining ports

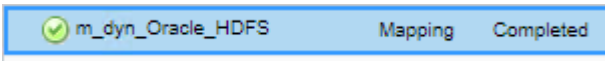
In the mapping Parameters table observe the configured parameters:

Name	Type	Pr...	S...	Default Value
ORCL_TableName	String	10...	0	PRODUCT_BRANDS
HDFS_Filename	String	10...	0	product_brands.csv

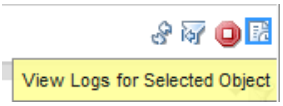
The mapping will use these default values as Input table and HDFS target file

You can now **run manually** the mapping

You should see in Informatica Monitoring Console:



You can download the mapping log



Check that it has taken the above parameters as below:

```
The Integration Service uses the default value [PRODUCT_BRANDS] for the mapping parameter
[ORCL_TableName].

The Integration Service uses the default value [product_brands.csv] for the mapping parameter
[HDFS_Filename].
```

In next section, we will automate the ingestion of all POS tables into HIVE via a workflow

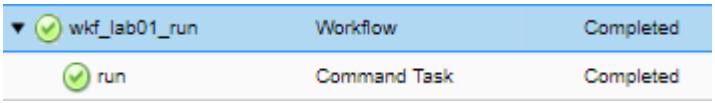
3.5.4 Run Lab

Workflow **wkf_lab01_run** calls a script to run the lab

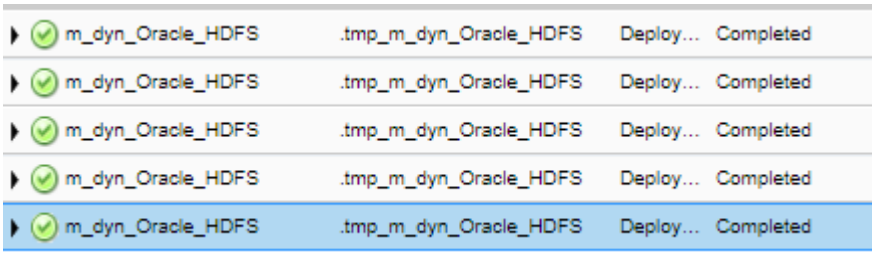
The run task will get all PIM tables, create a parameter file and execute run mapping command for each one.

To run the lab, go to Informatica Administration Console and run the workflow

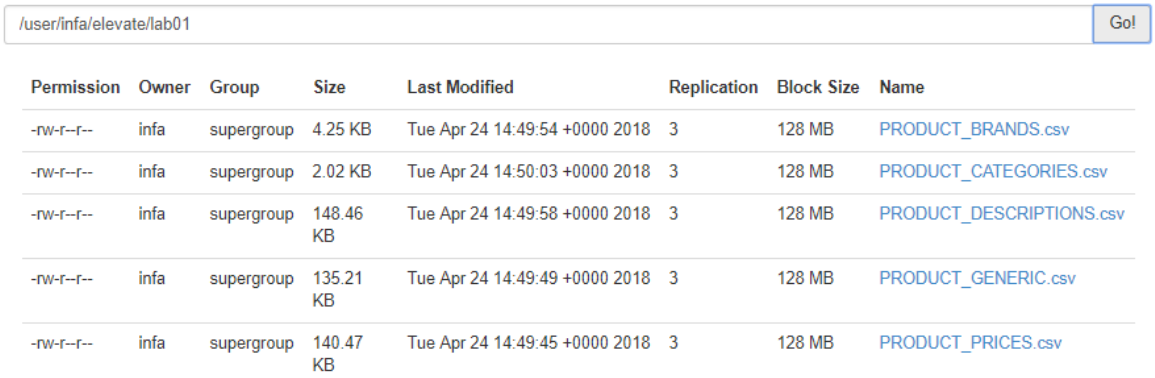
You should see in Informatica Monitoring Console:



And, in the navigation pane, go to deployed mapping:



Use HDFS Web Browser and navigate to /user/infa/elevate/lab01, you should see:



Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	infa	supergroup	4.25 KB	Tue Apr 24 14:49:54 +0000 2018	3	128 MB	PRODUCT_BRANDS.csv
-rw-r--r--	infa	supergroup	2.02 KB	Tue Apr 24 14:50:03 +0000 2018	3	128 MB	PRODUCT_CATEGORIES.csv
-rw-r--r--	infa	supergroup	148.46 KB	Tue Apr 24 14:49:58 +0000 2018	3	128 MB	PRODUCT_DESCRIPTIONS.csv
-rw-r--r--	infa	supergroup	135.21 KB	Tue Apr 24 14:49:49 +0000 2018	3	128 MB	PRODUCT_GENERIC.csv
-rw-r--r--	infa	supergroup	140.47 KB	Tue Apr 24 14:49:45 +0000 2018	3	128 MB	PRODUCT_PRICES.csv

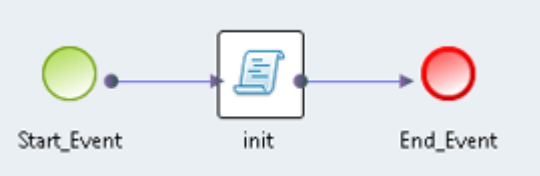
4. Lab02 - POS Mass Ingest with Dynamic mappings

4.1 Purpose

Show dynamic mapping unique capability (oracle to HIVE in Blaze hadoop push down)

4.2 Initialize Lab

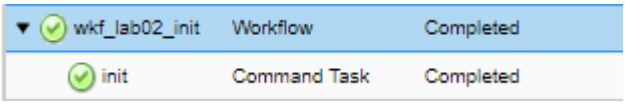
Workflow **wkf_lab02_init** calls a script to reinitialize the lab



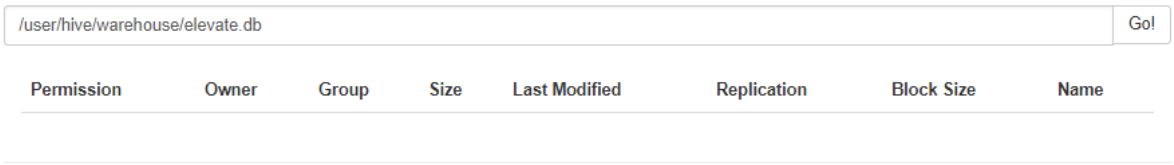
The init task command will clean all HIVE target data

To initialize the lab, go to Informatica Administration Console and run the workflow

You should see in Informatica Monitoring Console:



Use HDFS Web Browser and navigate to /user/hive/warehouse/elevate.db:

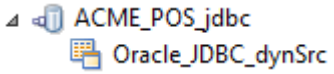


You should see **no hive tables**

4.3 Review Lab Content

In Developer, connect to MRS and open folder ELEVATE/BDM_Labs/lab02_*

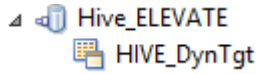
Observe the **source** object:



The **source** 'Oracle_JDBC_dynSrc' uses 'ACME_POS_jdbc' connection and 'fake' columns:

	Name	Native Type	Precisi...	Scale	Primar...	Nullable
1	String	varchar	10	0	☐	☐
2	Others	varchar	10	0	☐	☐

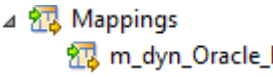
Observe the **target** object:



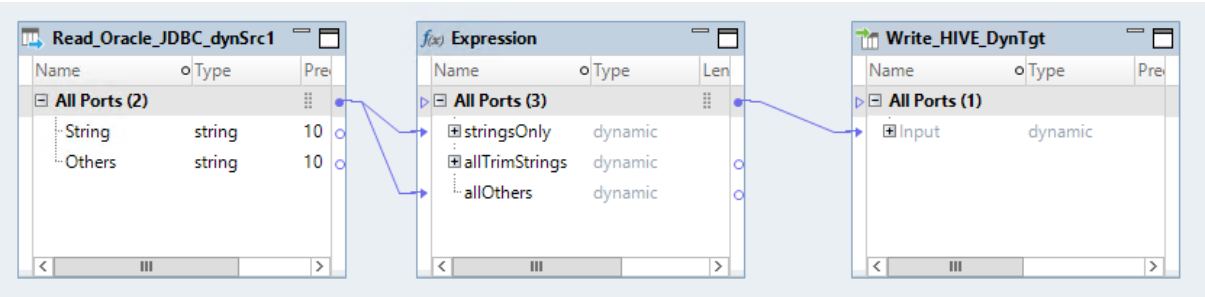
The **source** 'HIVE_DynTgt' uses 'Hive_ELEVATE' connection and 'fake' columns:

	Name	Native Type	Precisi...
1	dyn_col	string	10

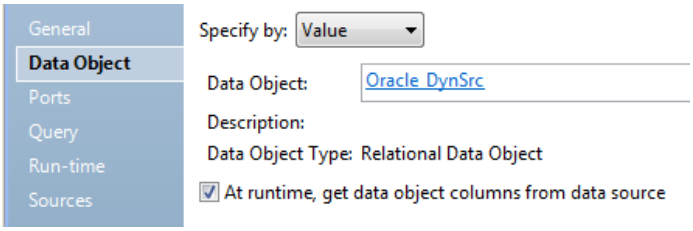
Observe the mapping:



The **mapping** is using dynamic ports



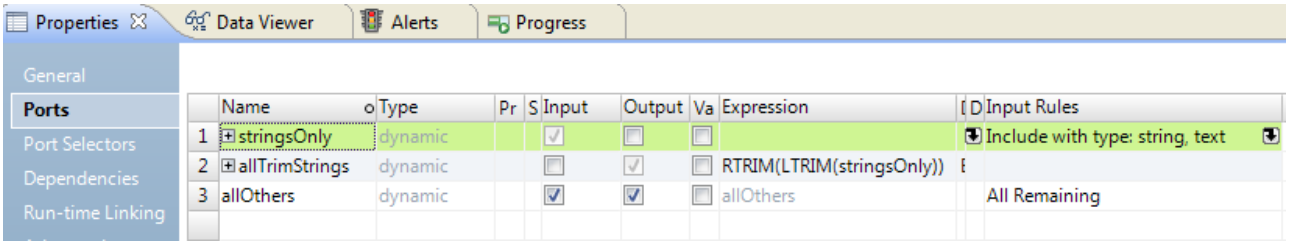
In mapping **Read Data Object** observe that it uses 'At runtime, get data object columns from data source'



In mapping **Read** Object '**Run-time**' properties observe that it uses **parameter** for **table name**

Name	Value
Connection	ACME_POS_jdbc
Owner	
Resource	Oracle_Table (Parameter)

In mapping Expression observe the port configuration



'**stringsOnly**' port is defined as dynamic, input only and includes all string and text ports

'**allTrimStrings**' is defined as dynamic, output only, doing RTRIM and LTRIM on 'stringsOnly'

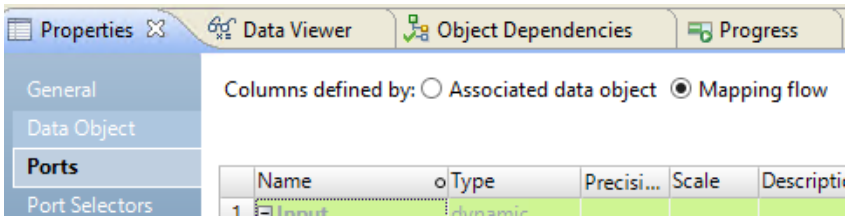
'**allOthers**' port is defined as dynamic, both input and output and include all remaining ports

In the mapping **Parameters** table observe the configured parameters:

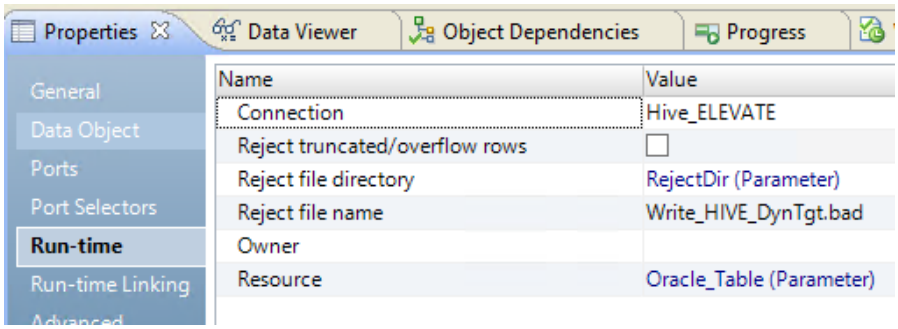
	Name	Type	Precision	Scale	Default Value	Description
1	Oracle_Table	String	100	0	STORES	
2	Hive_Format	String	10	0	AVRO	

The mapping will use these default values as Input table and hive format target

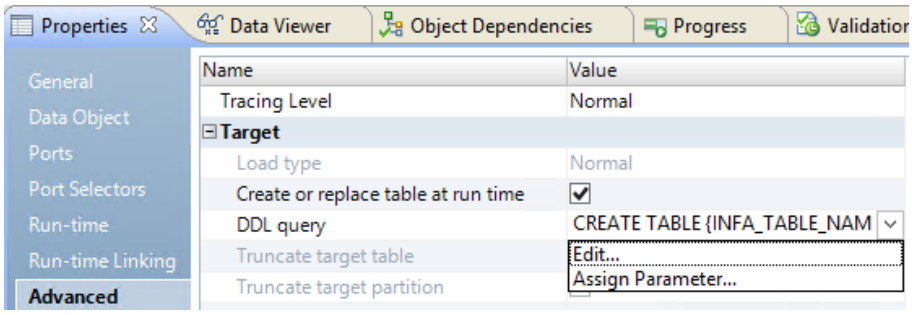
In mapping **Write Data Object** observe that it uses 'Mapping flow'



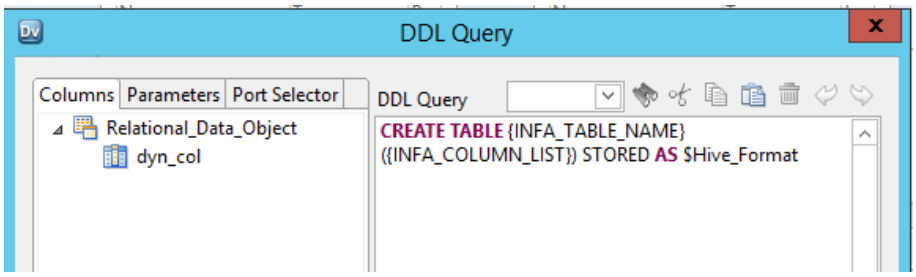
In 'Run-time' observe it will use the same mapping parameter for Hive target name



In 'Advanced' properties see that 'create or replace table at run time' is checked



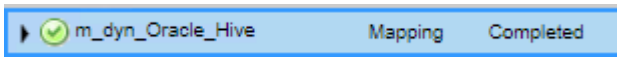
Edit the DDL Query



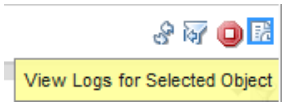
Observe the specific syntax and that it uses the parameter Hive_Format

You can now **run manually** the mapping

You should see in Informatica Monitoring Console:



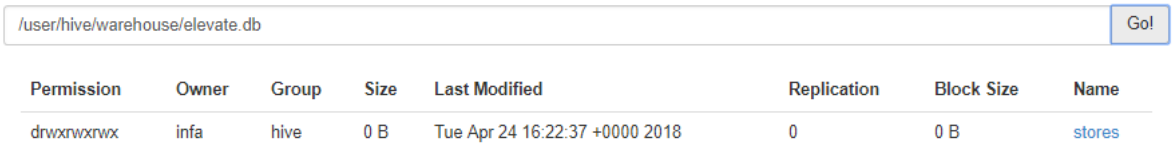
You can download the mapping log



Check that it has taken the above parameters as below:

```
Integration Service uses the default value [STORES] for the mapping parameter [Oracle_Table].
Integration Service uses the default value [AVRO] for the mapping parameter [Hive_Format].
```

Use HDFS Web Browser and navigate to /user/hive/warehouse/elevate.db:



You should see **stores** hive table was created

You can download the underlying HDFS file and see it is in AVRO format

```
Objavro.schemaÎ       {"type":"record","name":"InfaAvroScheam","f ….
...
```

In next section, we will automate the ingestion of all POS tables into HDFS via a workflow

Before we need to reinitialize the lab, running workflow wkf_lab02_init again

Note: you should get otherwise: Failure to execute Query CREATE TABLE `ORDERS`

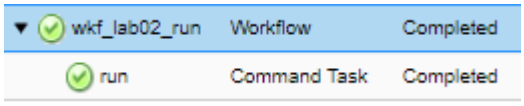
4.4 Run Lab

Workflow **wkf_lab02_run** calls a script to run the lab

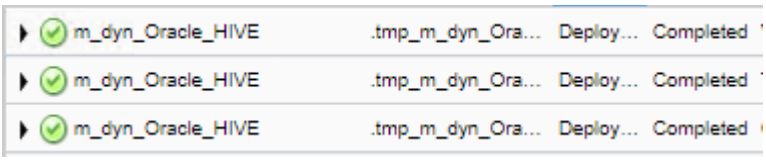
The run task will get all PIM tables, create a parameter file and execute run mapping command for each one.

To run the lab, go to Informatica Administration Console and run the workflow

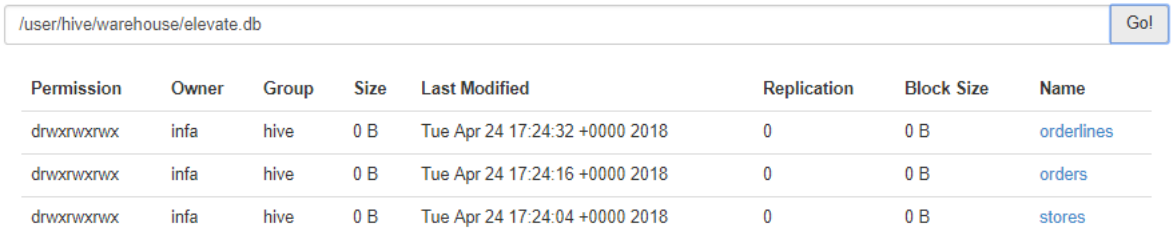
You should see in Informatica Monitoring Console:



And, in the navigation pane, go to deployed mapping:



Use HDFS Web Browser and navigate to /user/hive/warehouse/elevate.db:



You should see that hive tables orderlines, orders and stores were created

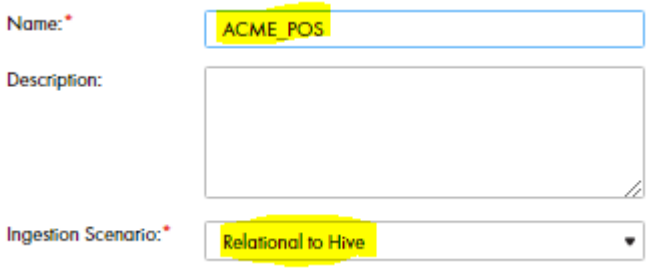
4.5 Extra Steps (Optional)

4.5.1 Purpose

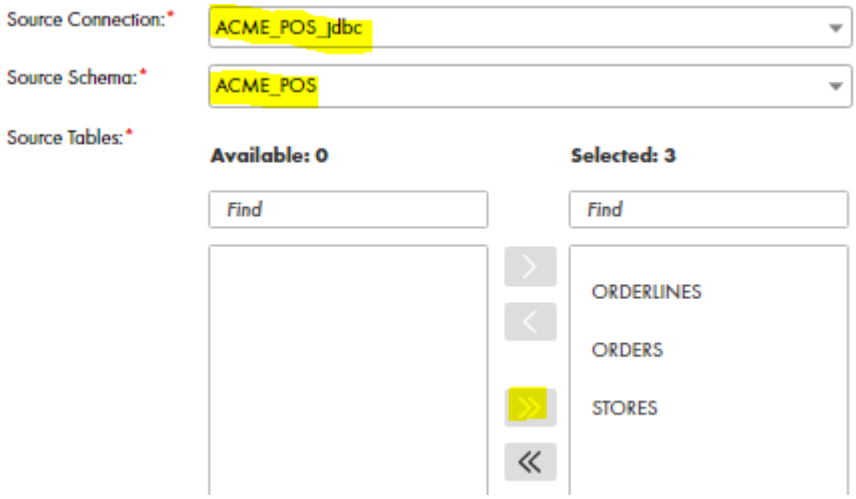
Show new 10.2.1 MIS (Mass Ingestion Service) capability

4.5.2 Create MIS project

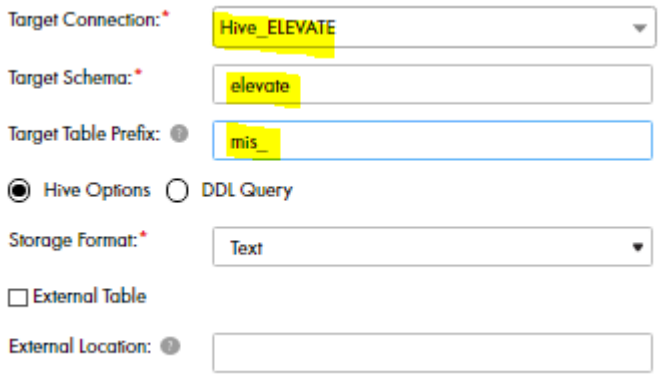
Go to Informatica Mass Ingestion Services and create a new project



Enter **ACME_POS** as project Name, **select Relational to Hive** and hit **Next**

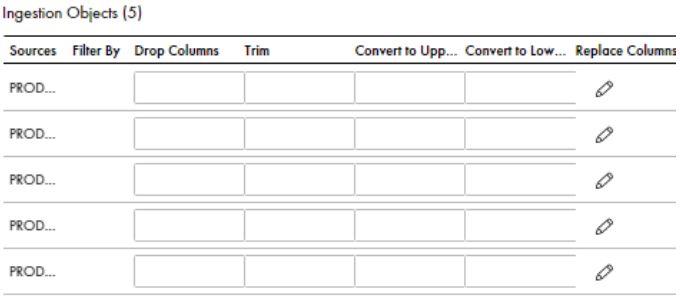


Select **ACME_POS_jdbc** as Source Connection, **ACME_POS** as Source Schema
Then select all tables (**>>**) and hit Next

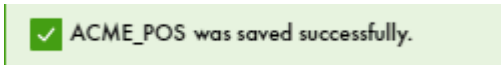


Select **Hive_ELEVATE** as Target Connection and **elevate** as Target Schema
Enter **mis_** as Target Table Prefix
Leave rest as default and hit **Next**

Leave all blank as default and hit **Next**

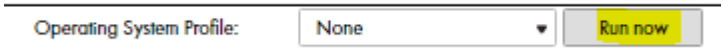


Leave all blank as default and hit **Save**



4.5.3 Deploy & Run MIS project

Select the **DIS1021** and the **HADOOP_CDH_513** connection and hit **Deploy**



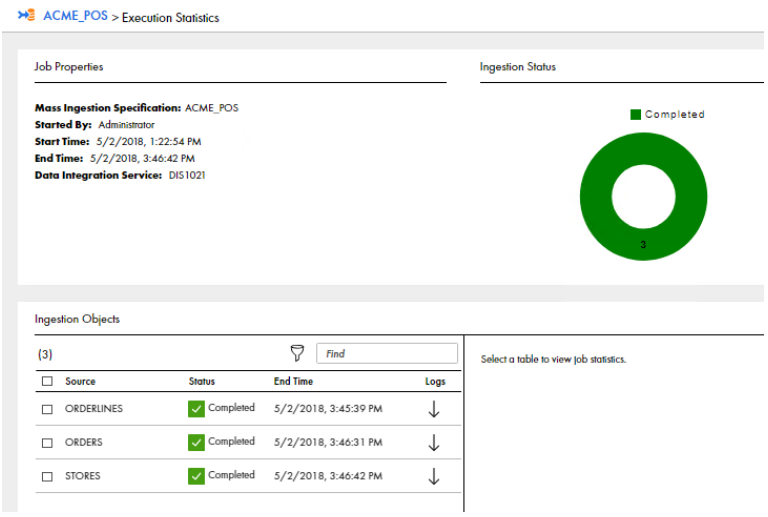
Once deployed, Hit **Run Now**

A RUNNING task should appear below Execution History

Execution History (1)

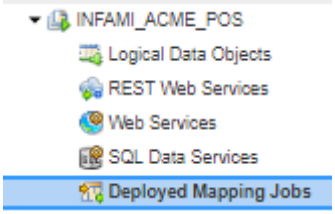
Start Time	Service Name	Status
4/24/2018, 3:26:14 PM	DIS1021	RUNNING

Hit the link



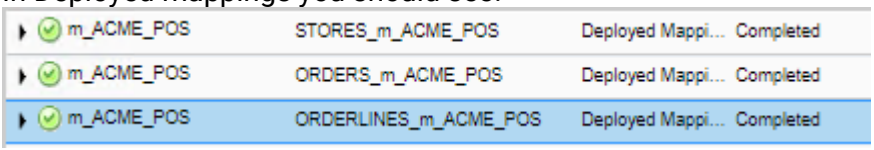
After a while you will see successful completion tasks

In Informatica Monitoring you can see application INFAMI_POS_CRM was deployed:

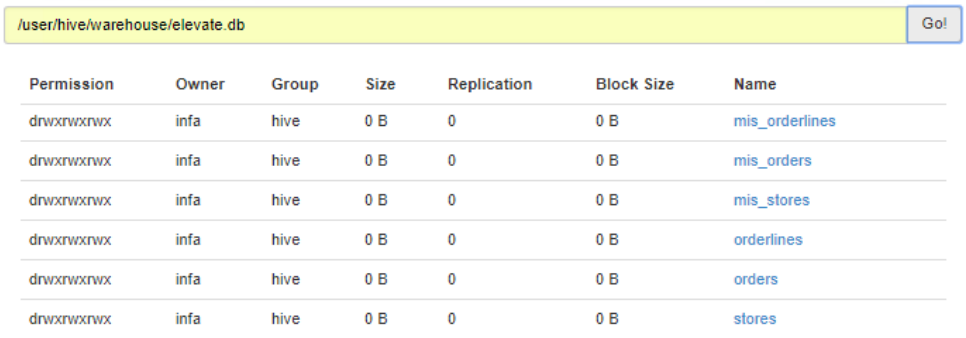


Note: You may need to refresh the applications

In Deployed mappings you should see:



Use HDFS Web Browser and navigate to /user/hive/warehouse/elevate.db:



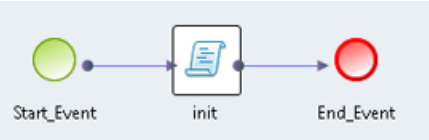
5. Lab03 – CRM Multiple Ingest (Demo only)

5.1 Purpose

Shows Multiple Ingestion from Oracle to Hive using native, jdbc, sqoop in native, hive & blaze

5.2 Initialize Lab

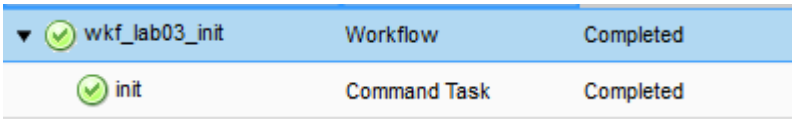
Workflow **wkf_lab03_init** calls a script to reinitialize the lab



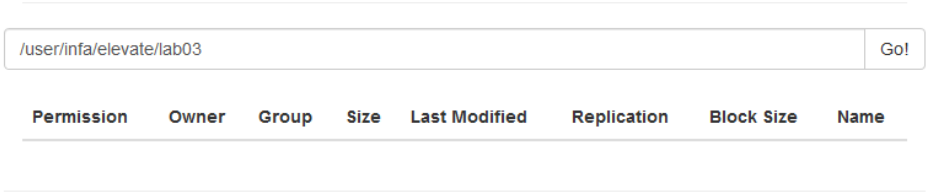
The **init** task command will create HDFS dummy file, clean target files

To initialize the lab, go to Informatica Administration Console and run the workflow

You should see in Informatica Monitoring Console:



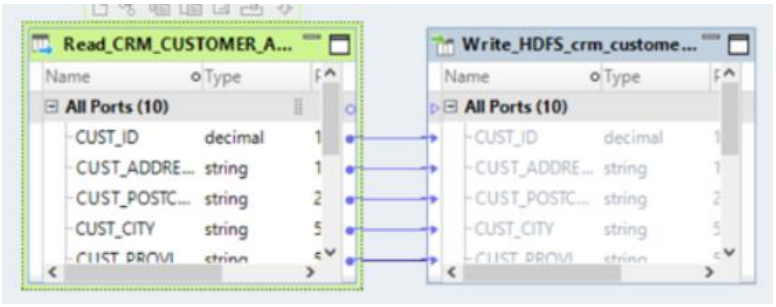
Use HDFS Web Browser and navigate to /user/infa/elevate/lab03, you should see:



5.3 Review Lab Content

Open Developer, go to ELEVATE/BDM_Labs/lab03_* folder

open the mapping m_load_HDFS_customer_address_oracle_native.

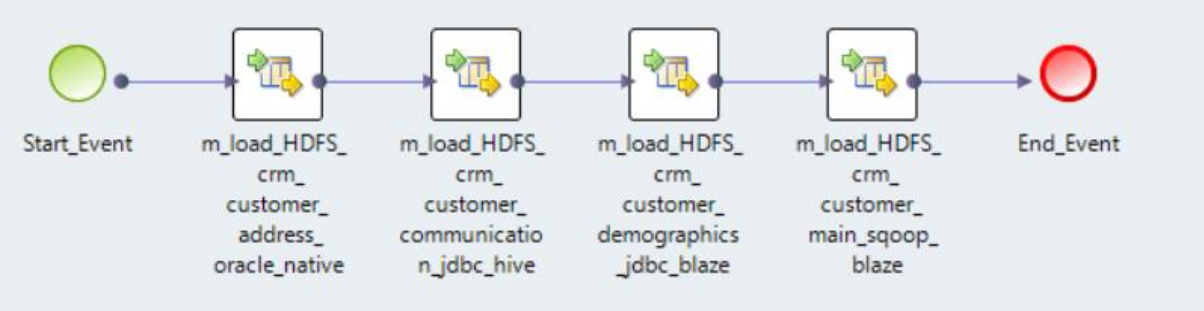


This passthrough mapping is using oracle native connection and native mode

Observe other mappings

5.4 Run Lab

Workflow **wkf_lab03_run** runs all unit test one after another



TO test multiple ingestion, go to Informatica Administration Console and run the workflow

You should see in Informatica Monitoring Console:

wkf_lab03_run	Workflow	Completed
m_load_HDFS_crm_customer_main_sqoop_blaze	Mapping Task	Completed
m_load_HDFS_crm_customer_demographics_jdbc_blaze	Mapping Task	Completed
m_load_HDFS_crm_customer_communication_jdbc_hive	Mapping Task	Completed
m_load_HDFS_crm_customer_address_oracle_native	Mapping Task	Completed

You should see in YARN Monitor Console:

application_1524754196789_0006	infa	CRM_CUSTOMER_MAIN.jar	MAPREDUCE	default	0	Thu Apr 26 17:39:41 +0200 2018	Thu Apr 26 17:40:02 +0200 2018	FINISHED	SUCCEEDED	N/A
application_1524754196789_0005	infa	/10.2.1/Blaze Grid Manager	Informatica Blaze	default	0	Thu Apr 26 17:38:07 +0200 2018	N/A	RUNNING	UNDEFINED	5
application_1524754196789_0004	infa	INSERT OVERWRITE TABL...munication_jdbc_hive(Stage-1)	MAPREDUCE	default	0	Thu Apr 26 17:37:28	Thu Apr 26 17:37:45	FINISHED	SUCCEEDED	N/A

You can see that sqoop blaze launches an extra mapreduce task

Use HDFS Web Browser and navigate to /user/infa/elevate/lab03, you should see:

/user/infa/elevate/lab03 Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	infa	hdfs	0 B	29/04/2018 à 20:45:37	0	0 B	customer_address
drwxr-xr-x	infa	hdfs	0 B	29/04/2018 à 20:47:08	0	0 B	customer_communication
drwxr-xr-x	infa	hdfs	0 B	29/04/2018 à 20:48:40	0	0 B	customer_demographics
drwxr-xr-x	infa	hdfs	0 B	29/04/2018 à 20:49:20	0	0 B	customer_main

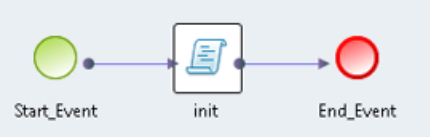
6. Lab04 - CRM Discover data

6.1 Purpose

Load & Profile a hive table in native and hadoop modes

6.2 Initialize Lab

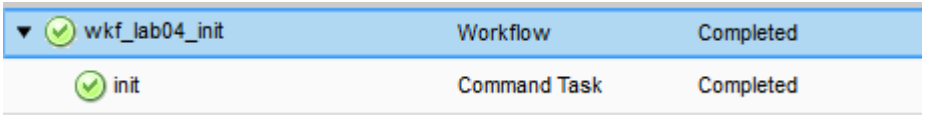
Workflow **wkf_lab04_init** calls a script to reinitialize the lab



The **init** task command will drop hive target table

To initialize the lab, go to Informatica Administration Console and run the workflow

You should see in Informatica Monitoring Console:



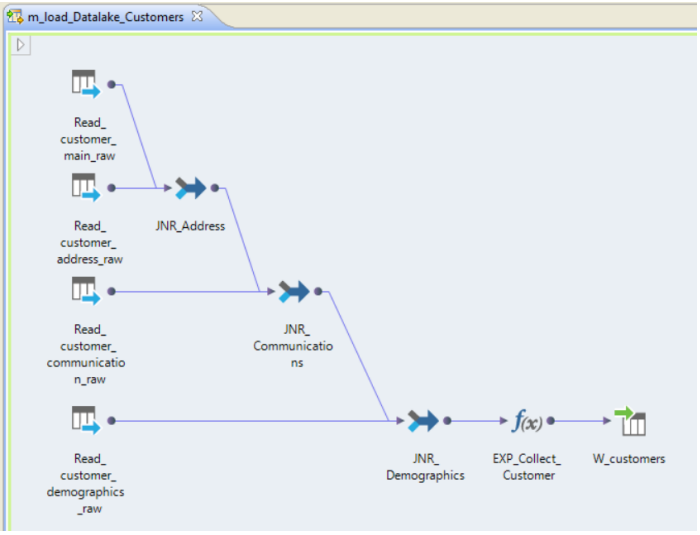
Using HDFS Web Browser go to /apps/hive/warehouse/elevate.db

You should see no **customer** table

6.3 Review Lab Content

The goal of this lab is to load then profile customer hive table using native & hadoop modes

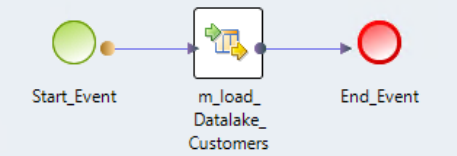
In Developer, go to lab04_* folder and open mapping **m_load_Datalake_Customers**:



This mapping joins all CRM tables (from previous lab) into Hive

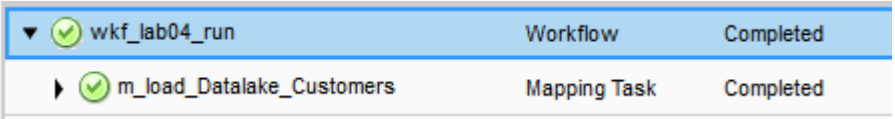
6.4 Run Lab

Workflow **wkf_lab04_run** will load the hive target



Go to Informatica Administration Console and run the workflow

You should see in Informatica Monitoring Console:

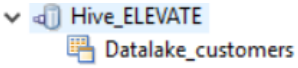


Using HDFS Web Browser go to /apps/hive/warehouse/elevate.db

You should see **customer** hive table was created

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxrwxrwx	infa	hdfs	0 B	29/04/2018 à 20:54:45	0	0 B	customers

To review the hive content, in Developer, lab04_* folder, open **Datalake_customers**



Run Data Preview, you should see:

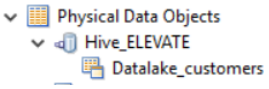
Name: Datalake_customers

	custid	cust_name	cust_firstname	cu	cust_lastname	cust_gender	cust_dob
1	166	MATTINZ	KARIMA	<	KARIMA MAT...	F	09/28/2045 0...
2	167	BETTELLI	EDDY	<	EDDY BETTELLI	1	12/09/1969 0...
3	57	TURBIN	GWENN	<	GWENN TUR...	F	10/15/1957 0...
4	58	GONCALEZ	MELDA	<	MELDA GON...	F	02/25/1959 0...

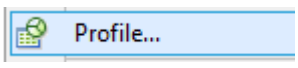
We will now create & run profiling on this table

6.4.1 Create & Run Profile in native mode – (Optional)

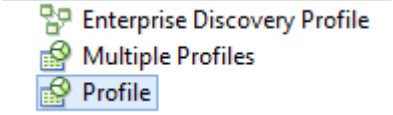
In Developer, open INFA/lab04_* and select **Datalake_customers** Data Object



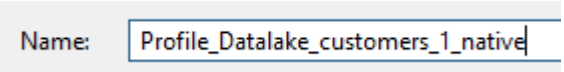
Right click and chose Profile



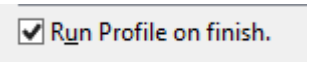
Chose Simple Profile an hit Next



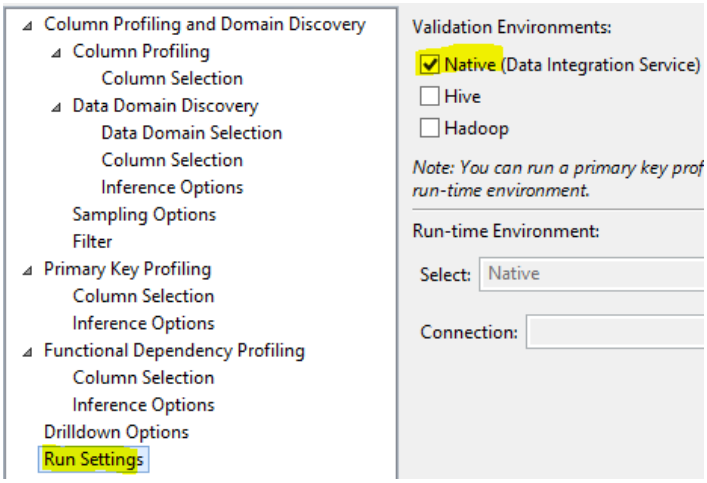
Rename it Profile_Datalake_customers_1_native



Leave the Run Profile option checked and hit Next



Go to **Run Settings** and leave default **Native** execution mode



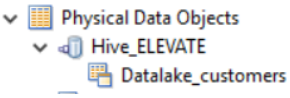
Hit Finish

Wait until the profile is finished monitoring in Developer **Progress** tab

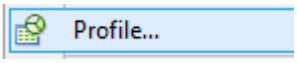
In our lab environment, this should take **~20 seconds** for completion

6.4.2 Create & Run Profile in blaze mode

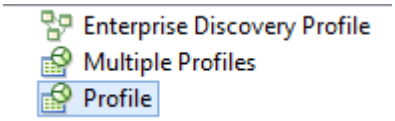
In Developer, open INFA/lab04_* and select **Datalake_customers** Data Object



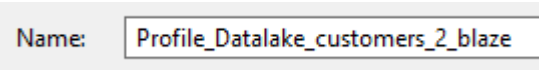
Right click and chose Profile



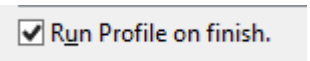
Chose Simple Profile and hit Next



Rename it Profile_Datalake_customers_2_blaze

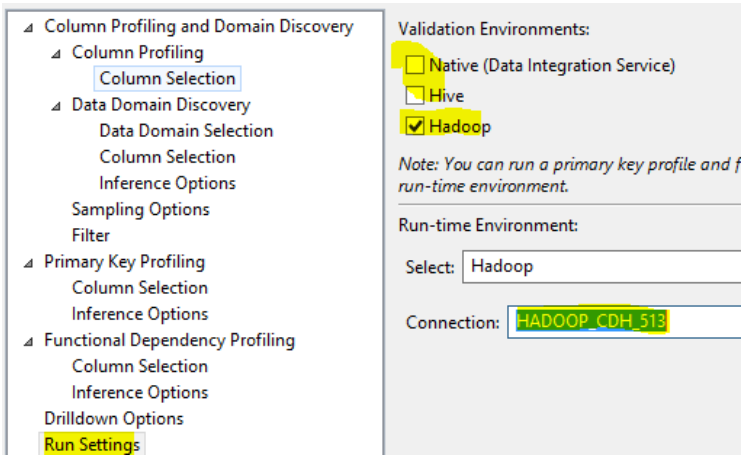


Leave the Run Profile option checked and hit Next



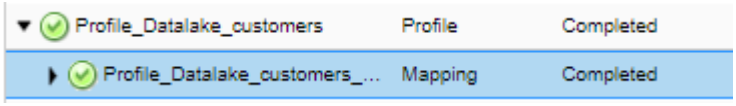
Go to **Run Settings** and **uncheck Native** and check Hadoop execution mode

Then Browse and set the Hadoop connection:



Hit Finish

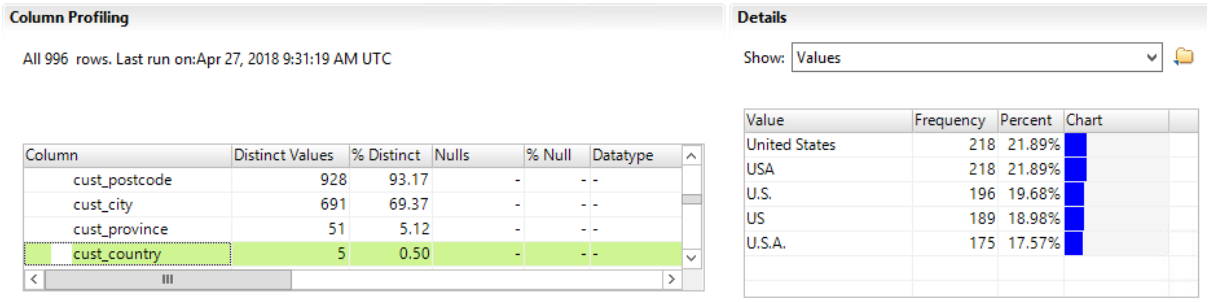
You should see in Informatica Monitoring Console:



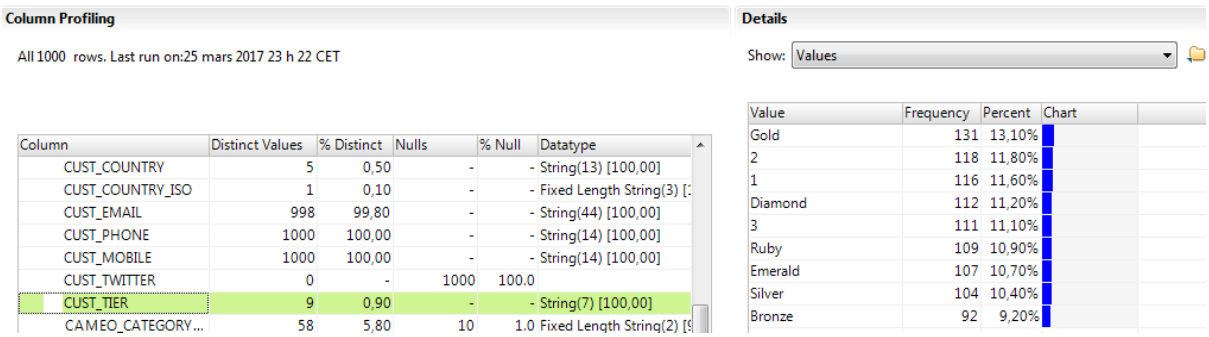
In our lab environment, this should take **~40 seconds** for completion

6.5 Review Profiling results for GENDER and TIER columns

In Developer, go to the profile, results and see the **COUNTRY column**:



In Developer, go to the profile, results and see the **TIER column**:



In next lab exercise we will **cleanse** those **both columns** using Data Quality Transformations.

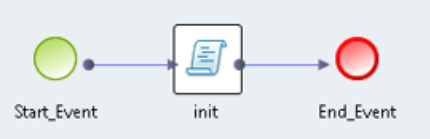
7. Lab05 - CRM Cleanse & Validate

7.1 Purpose

Show cleansing capabilities in hadoop (blaze) mode

7.2 Initialize Lab

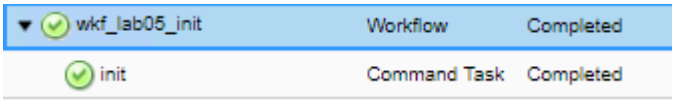
Workflow **wkf_lab05_init** calls a script to reinitialize the lab



The **init** task command will empty hive target table

To initialize the lab, go to Informatica Administration Console and run the workflow

You should see in Informatica Monitoring Console:



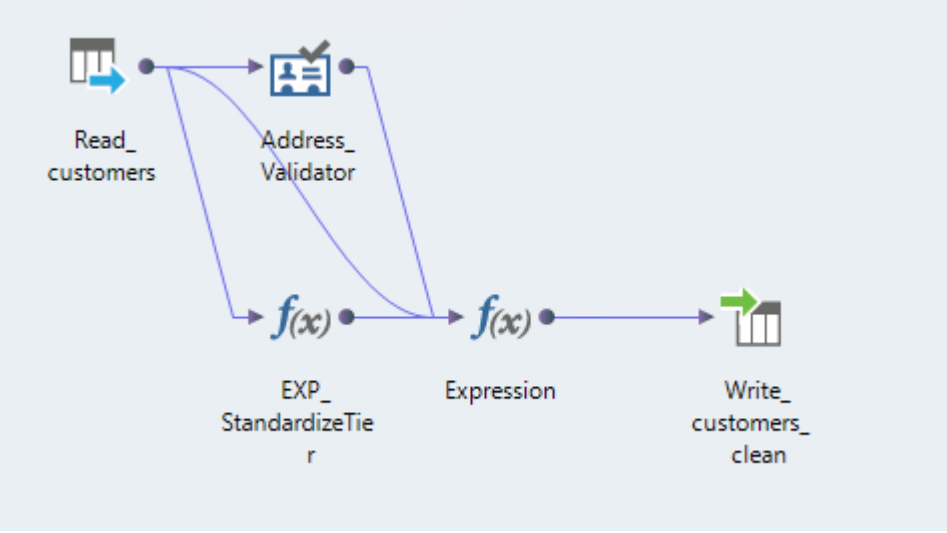
Using HDFS Web Browser go to /apps/hive/warehouse/elevate.db

You should see no **customer_clean** table

7.3 Review Lab Content

Open Developer, go to ELEVATE/BDM_Labs/lab05_* folder

open the mapping **m_cleanse_customer**



This mapping uses Data Quality Standardization rule (on Customer TIER) and Address Validation transformations.

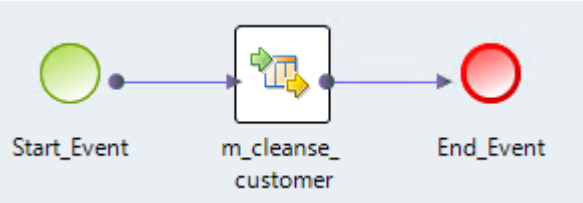
Select the **Expression** object and run **Data Preview**

You should see that columns **cust_tier** and **cust_country_iso** now contains cleansed & validated data.

We will now run this mapping in Hadoop Blaze mode.

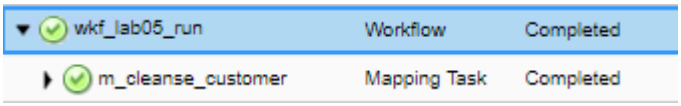
7.4 Run Lab

Workflow **wkf_lab05_run** will execute this mapping



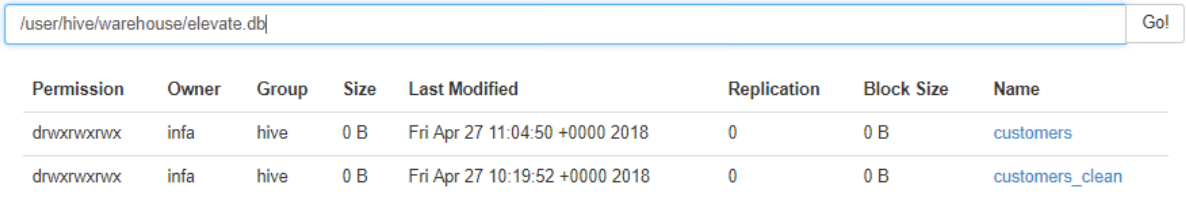
Go to Informatica Administration Console and run the workflow

You should see in Informatica Monitoring Console:

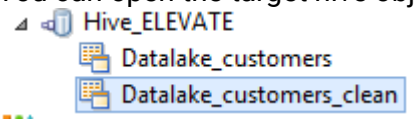


Using HDFS Web Browser go to /apps/hive/warehouse/elevate.db

You should see **customer_clean** hive table was created



You can open the target hive object



And Preview the cleansed Data

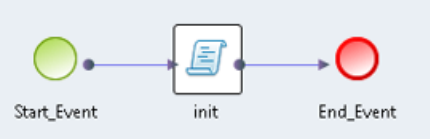
8. Lab06 - CRM Secure

8.1 Purpose

Show masking capabilities in hadoop (blaze) mode

8.2 Initialize Lab

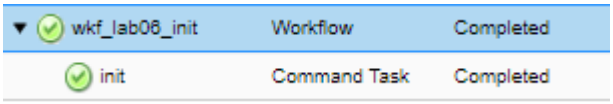
Workflow **wkf_lab06_init** calls a script to reinitialize the lab



The **init** task command will empty hive target table

To initialize the lab, go to Informatica Administration Console and run the workflow

You should see in Informatica Monitoring Console:



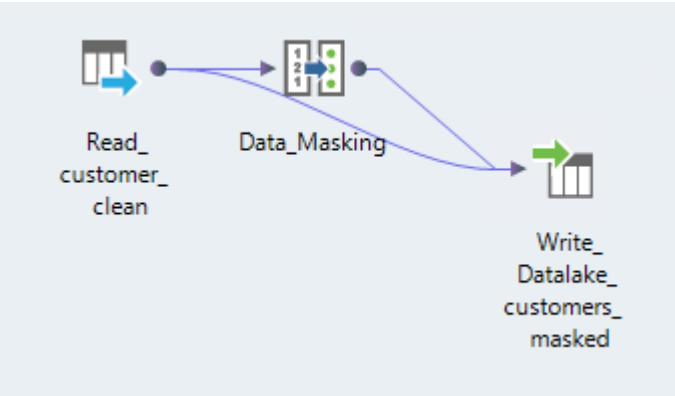
Using HDFS Web Browser go to /apps/hive/warehouse/elevate.db

You should see no **customers_masked** table

8.3 Review Lab Content

Open Developer, go to ELEVATE/BDM_Labs/lab06_* folder

open the mapping **m_mask_customer**



This mapping uses Data Masking transformations.

Select the **Data_Masking** object and run **Data Preview**

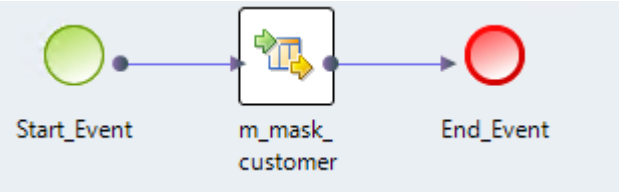
	cust_phone	cust_mobile	out_cust_phone	out_cust_mob...
1	95686683	956688193	20486310	558953365
2	619-422-687	619-252-939	361-033-013	020-953-095
3	339580555	339743737	679551024	926515700
4	+1204340655	+1204869743	+1396810008	+3823665678
5	402929363	402375562	201455680	973088763

You should see that columns **cust_phone** and **cust_mobile** will be masked.

We will now run this mapping in Hadoop Blaze mode.

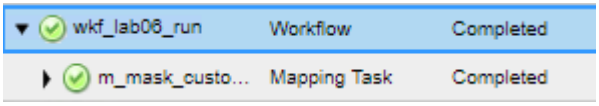
8.4 Run Lab

Workflow **wkf_lab06_run** will execute this mapping



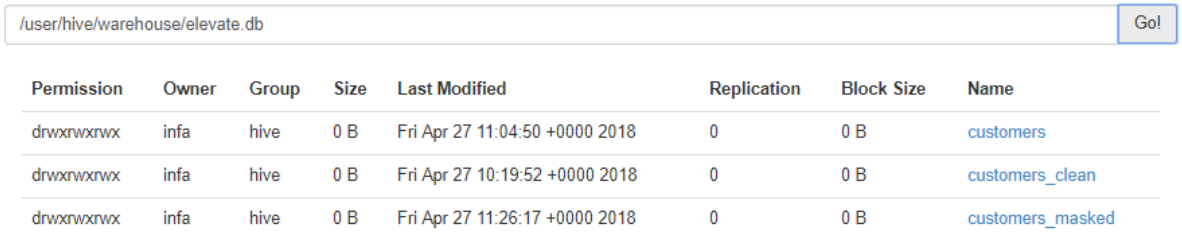
Go to Informatica Administration Console and run the workflow

You should see in Informatica Monitoring Console:

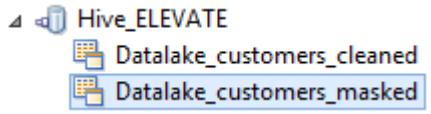


Using HDFS Web Browser go to /apps/hive/warehouse/elevate.db

You should see **customer_masked** hive table was created



You can open the target hive object



And Preview the masked data

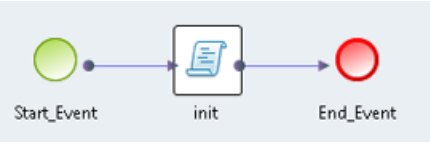
9. Lab07 - SIS Complex File DP & H2R

9.1 Purpose

Show H2R capabilities parsing complex files (xml files with same structure) in Blaze mode

9.2 Initialize Lab

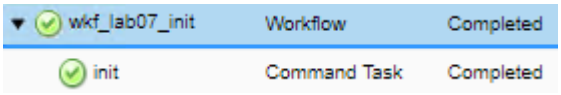
Workflow **wkf_lab07_init** calls a script to reinitialize the lab



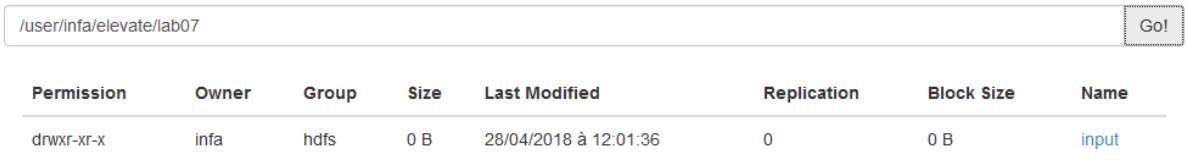
The **init** task command will clean target files

To initialize the lab, go to Informatica Administration Console and run the workflow

You should see in Informatica Monitoring Console:



Use HDFS Web Browser and navigate to /user/infa/elevate/lab07, you should see:

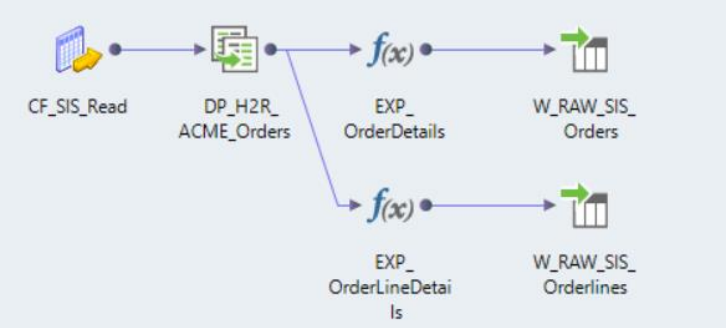


You can browse to the input folder and see all HDFS XML files.

9.3 Review Lab Content

Open Developer, go to ELEVATE/BDM_Labs/lab07_* folder

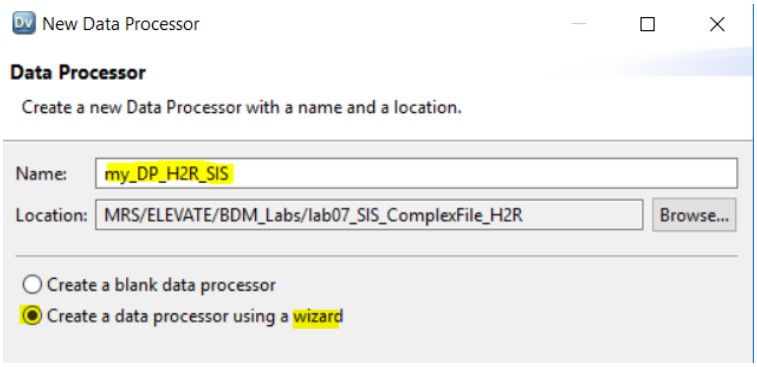
open the mapping **m_load_RAW_StoreInStore_Orders**



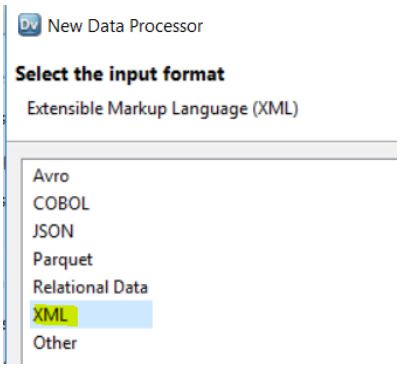
This mapping is using H2R transformation to parse complex files

Note: should you want to create same H2R Transformation follow the steps:

- Create new Data Processor Object and use Wizard option



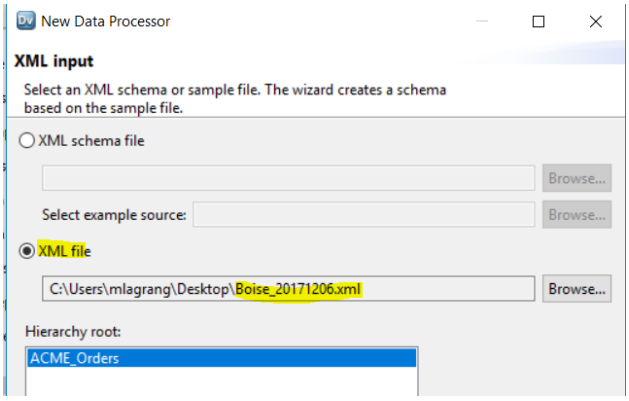
- Select Input format as XML



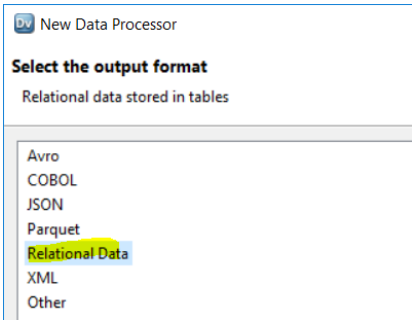
- Copy to your Desktop and unzip the below XML sample


Boise_20171206.zip

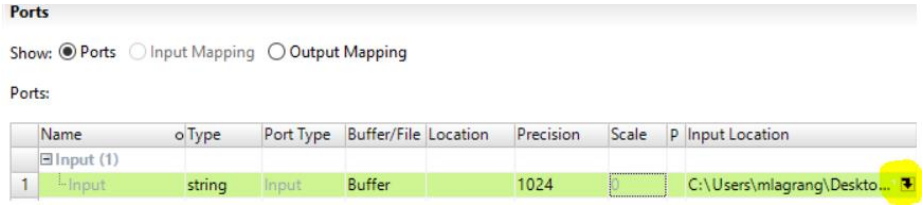
- Choose XML and Browse to your Desktop unzipped XML sample



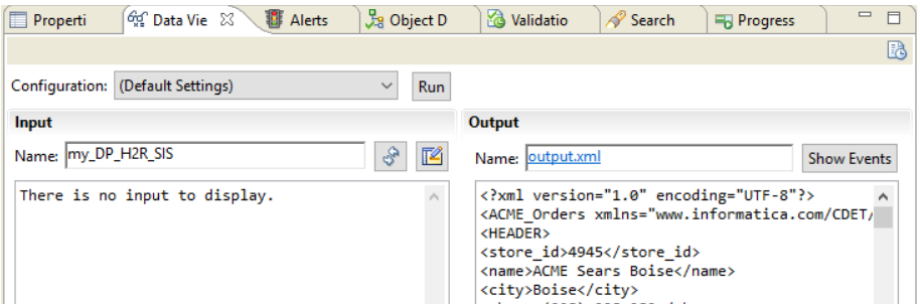
- Chose Output 'Relational Data' format, hit Finish



- Set the input location (Browse to your sample file)

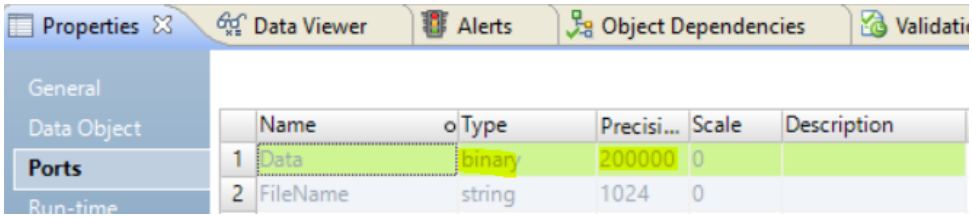


- Run Data Preview to test it



Select the mapping **CF_SIS_Read** Read Object

Observe that the 'Data' Port is of type binary with a 'large enough' size



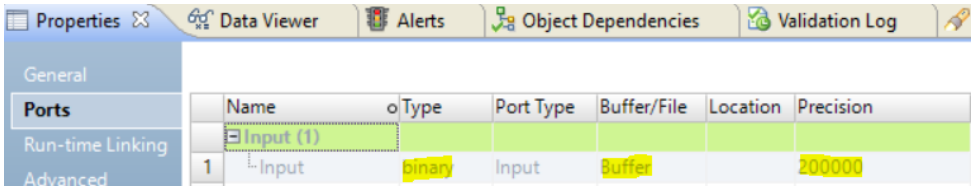
Run the Data Preview, you should see:

	Data	FileName
1	<binary>	hdfs://cos7.hdp26.local:8020/user/infa/elevate/lab07/Boise_20171206.xml
2	<binary>	hdfs://cos7.hdp26.local:8020/user/infa/elevate/lab07/Chula_Vista_20171206.xml
3	<binary>	hdfs://cos7.hdp26.local:8020/user/infa/elevate/lab07/Denver_20171206.xml
4	<binary>	hdfs://cos7.hdp26.local:8020/user/infa/elevate/lab07/Durham_20171206.xml
5	<binary>	hdfs://cos7.hdp26.local:8020/user/infa/elevate/lab07/Fort Wayne 20171206.xml

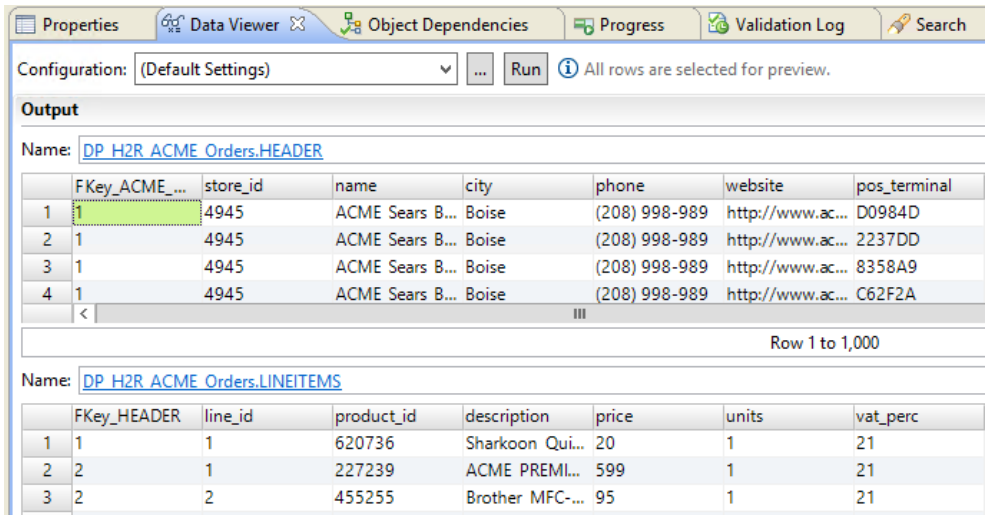
Note: 'one line a file' means the input size is 'large enough' (would otherwise be 'chunked' in multiple – and invalid xml structure- of xml files…)

Go to the DP_H2R_ACME_Orders Transformation

Observe that we use binary format, Buffer input and same 'large enough' input size:



Run Data Preview, you should see:

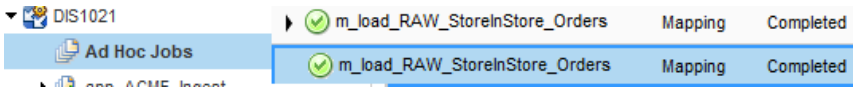


We will now run this mapping in native, hive & Blaze mode.

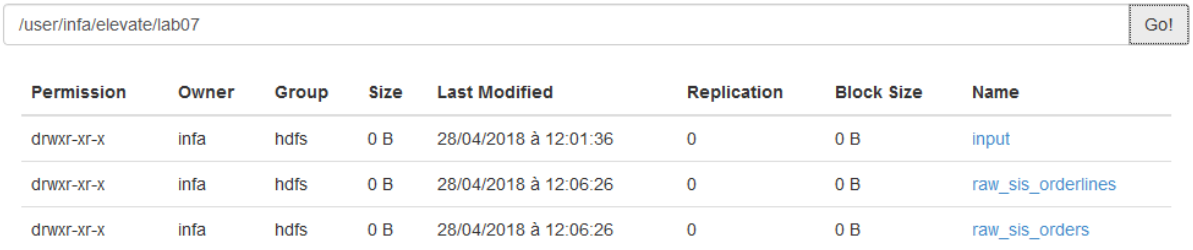
9.4 Run Lab

Configure the mapping to run in native & Blaze mode, Run it and compare executions.

You should see in Informatica Monitoring Console:



Use HDFS Web Browser and navigate to /user/infa/elevate/lab07, you should see:



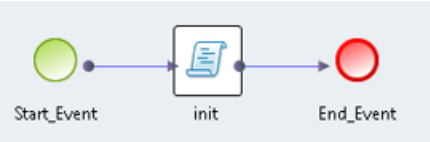
10. Lab08 - Call Center Unstructured File ISD

10.1 Purpose

Show ISD (Intelligent Structure Discovery) capabilities parsing complex files in Spark mode

10.2 Initialize Lab

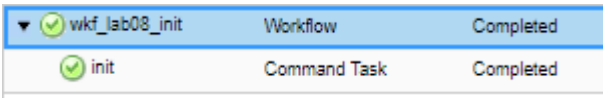
Workflow **wkf_lab08_init** calls a script to reinitialize the lab



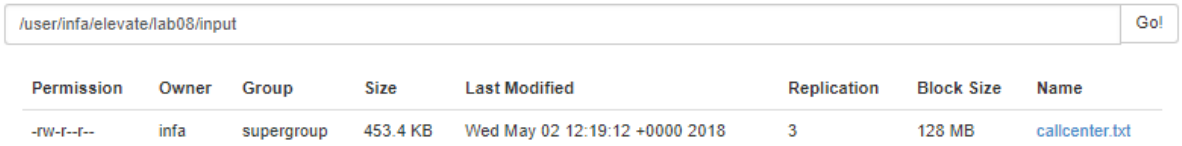
The **init** task command will clean target files

To initialize the lab, go to Informatica Administration Console and run the workflow

You should see in Informatica Monitoring Console:



Use HDFS Web Browser and navigate to /user/infa/elevate/lab08/input, you should see:

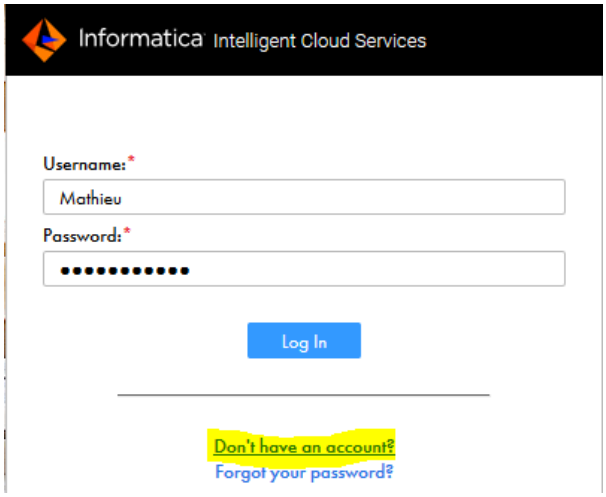


10.3 Build & Export an ISD Model from Cloud

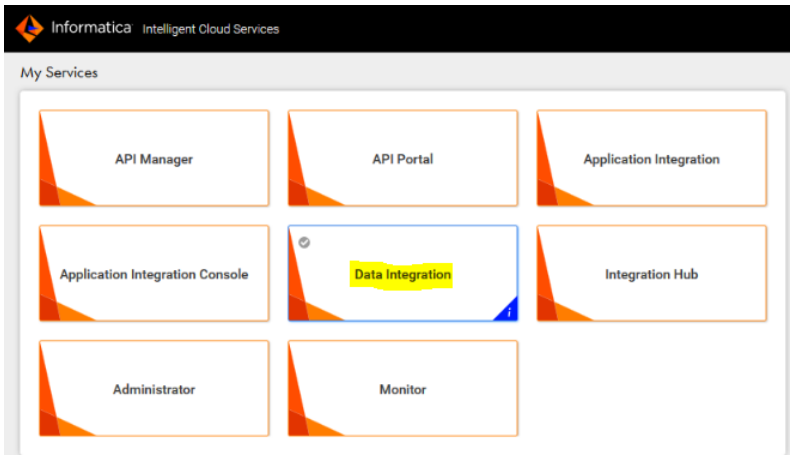
Copy to your Desktop the sample file we will be working with:

raw_callcenter.csv

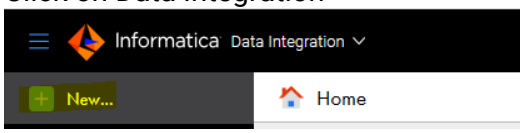
Go to Informatica Cloud: https://dm-us.informaticacloud.com/identity-service/home



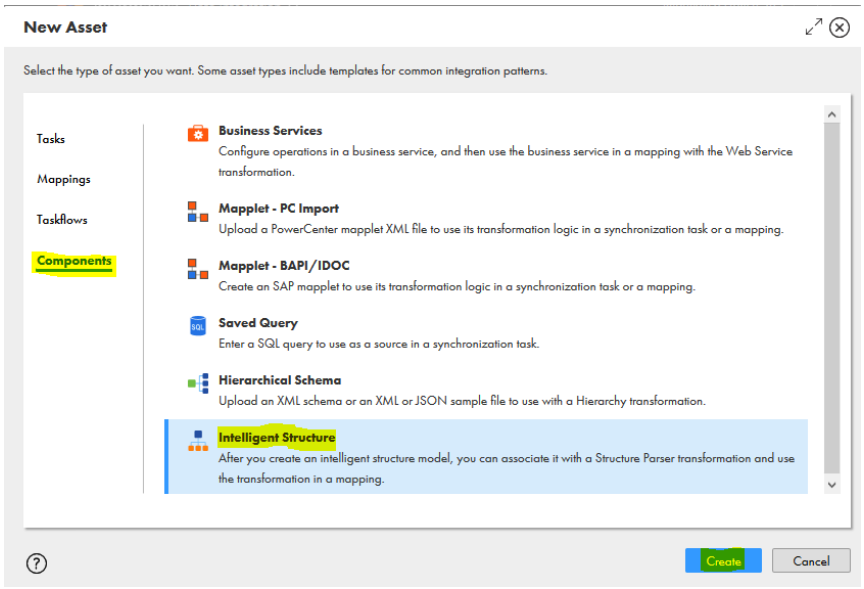
Log with your credentials if you have already (or register for Trial period)



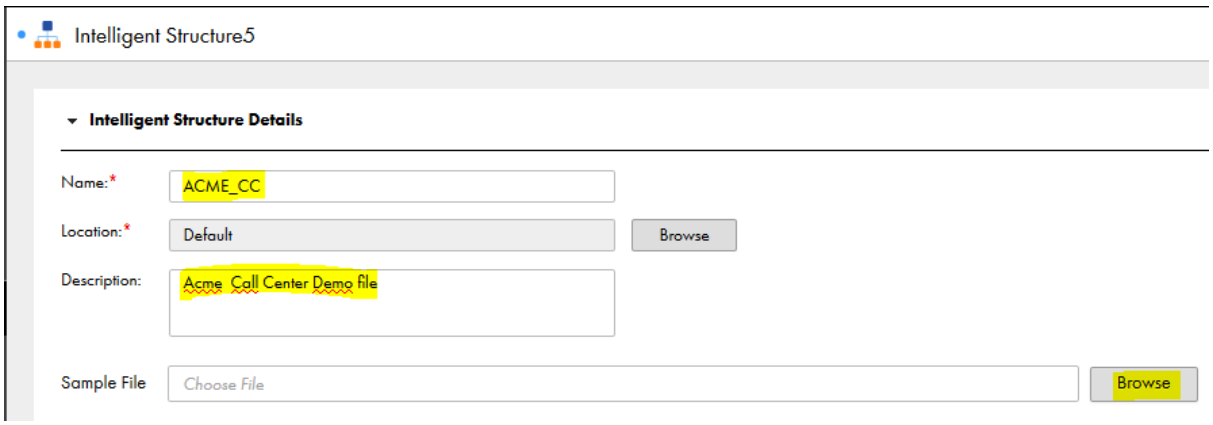
Click on Data Integration



Click on New

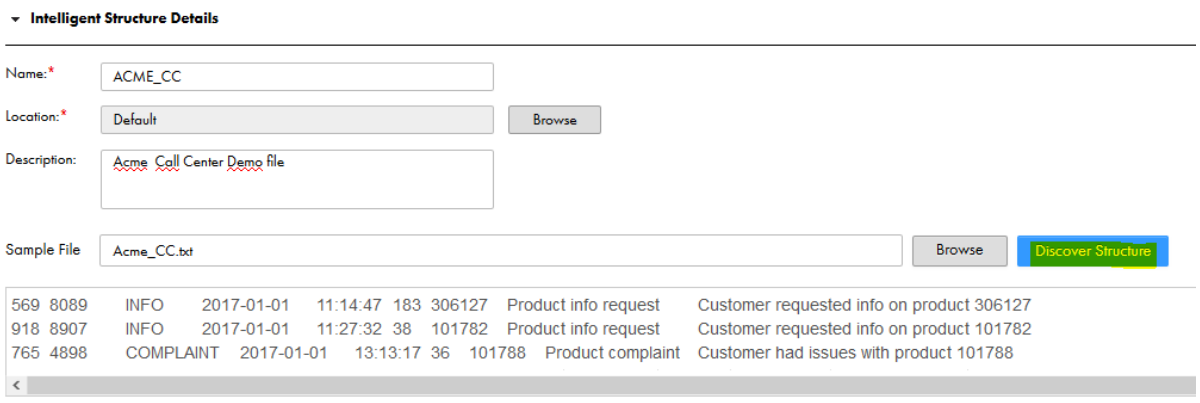


Select **Components**, then **Intelligent Structure** and Hit Create

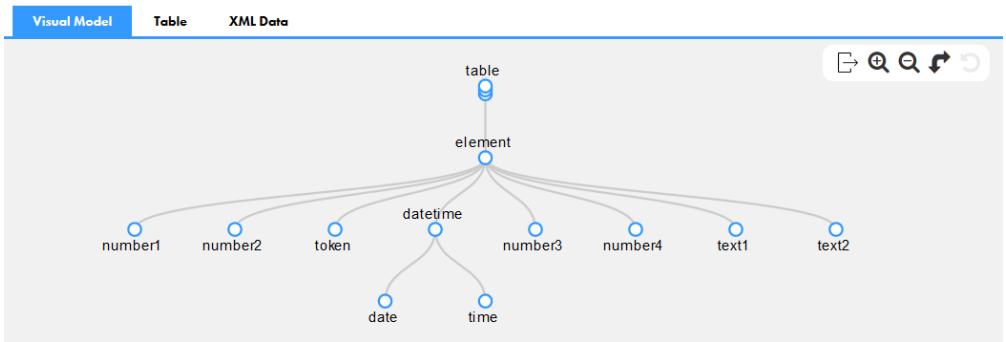


Enter ACME as Name, add comment and Hit the second Browse button to choose a file

Select Acme_Demo.txt from your Desktop

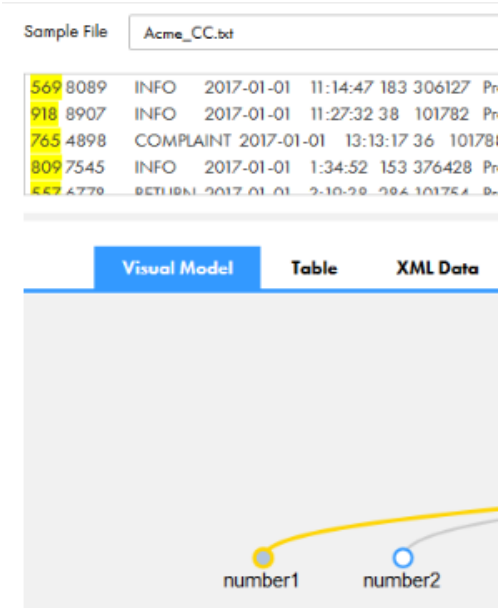


The Data is shown
Hit Discover Structure



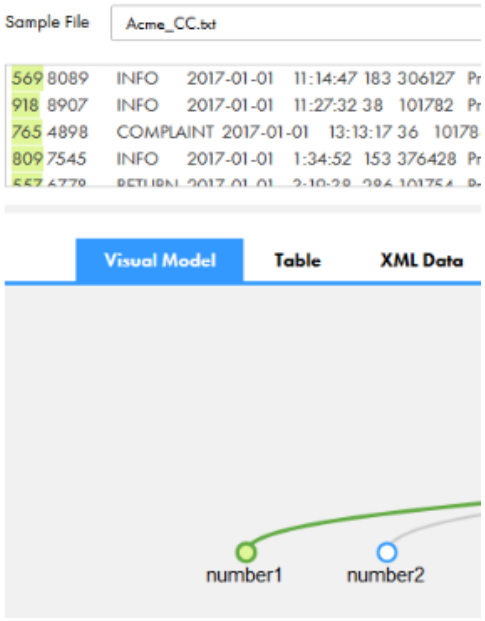
After a short while a model is presented

Move your mouse to first element: ‘number1’



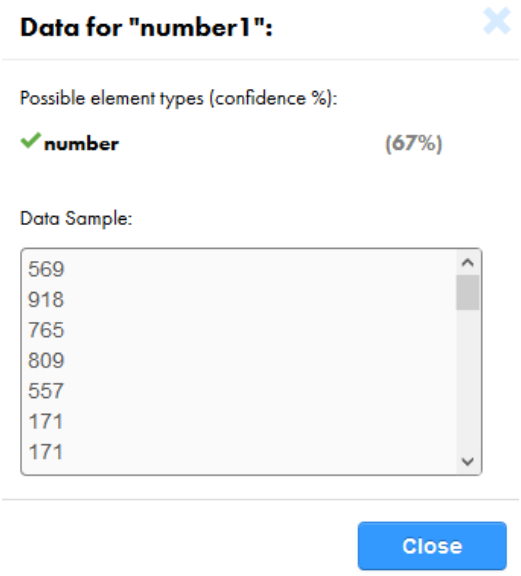
You can see which data it corresponds to

Click on first element: ‘number1’



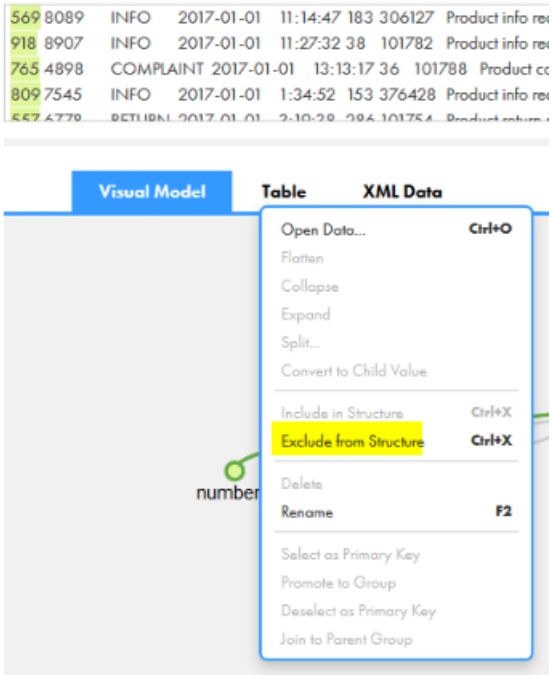
It has selected in green the branch – should you want to keep this element

Click again (or double-click) on first element ‘number1’

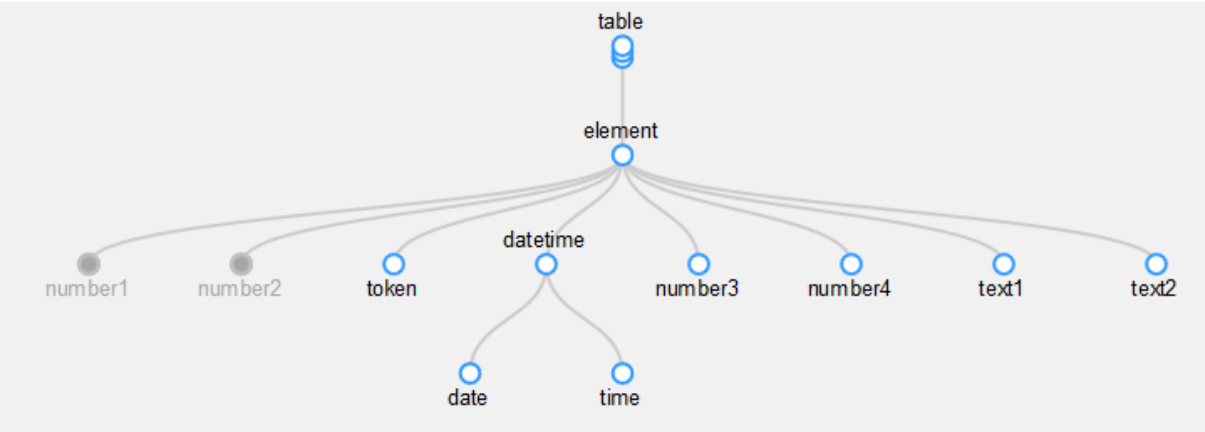


You can see with which confidence the underlying machine learning algorithm made his decision to make this element an 'number'

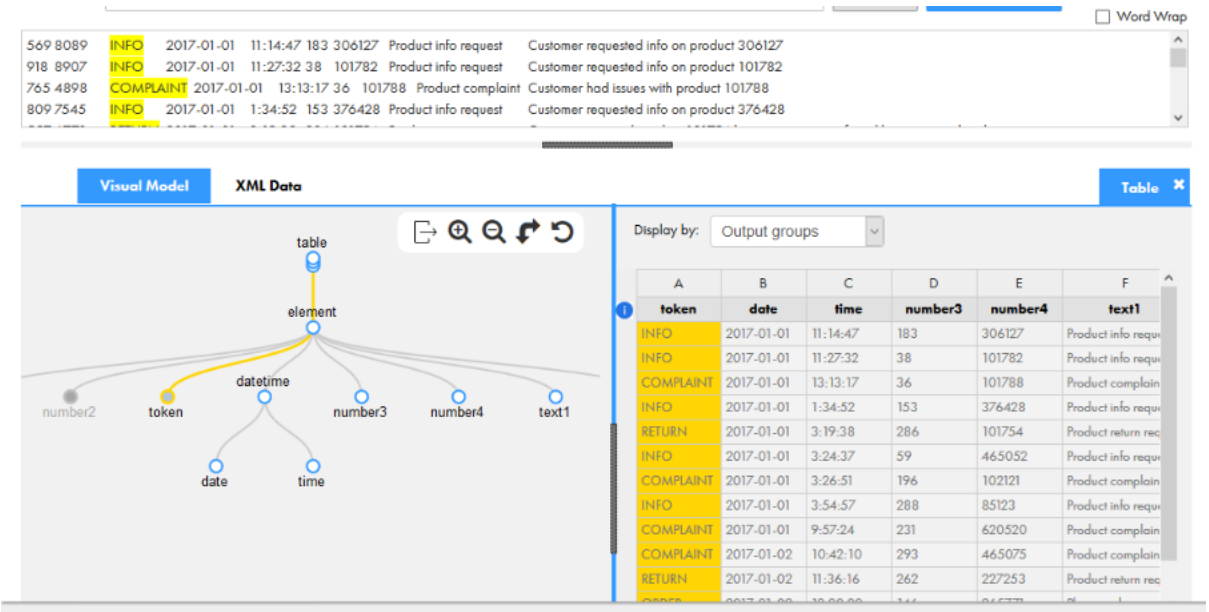
Right click on IP element



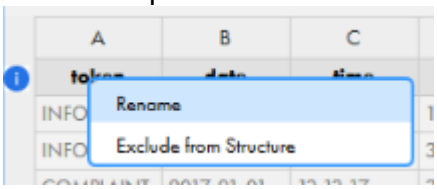
You can see all operations such as exclude rename etc…
As we don't know what it corresponds to: let's choose "**Exclude from Structure**"

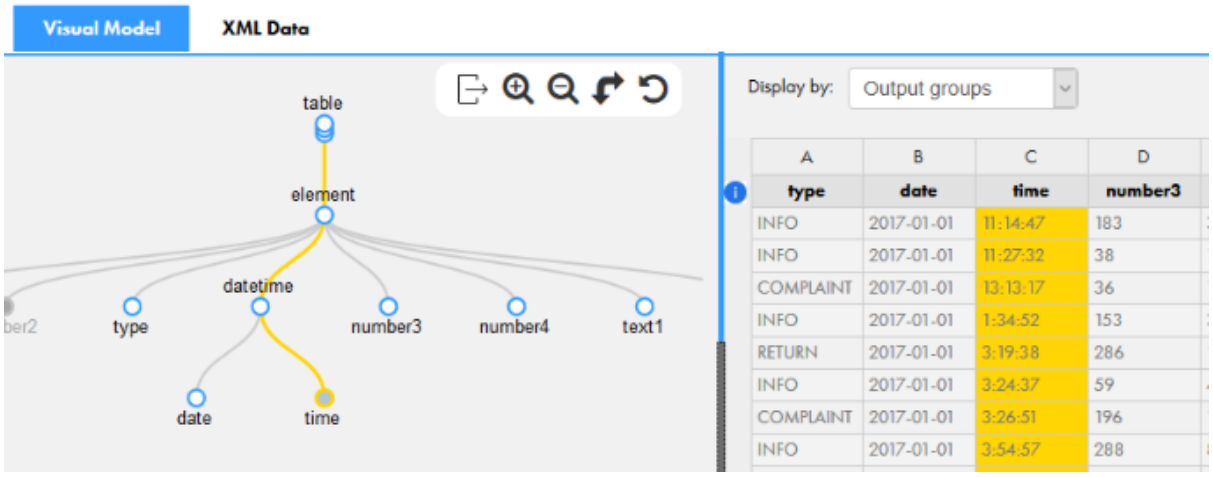


Click on Table then navigate to column A: 'token'



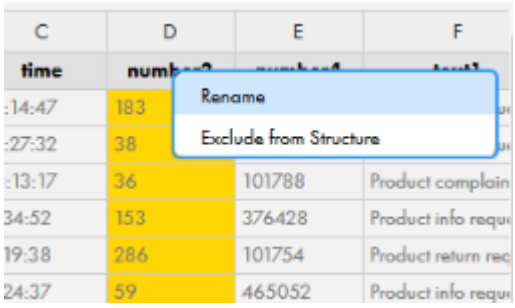
This correspond to a TYPE of Call Center Request, so let's rename it **type**



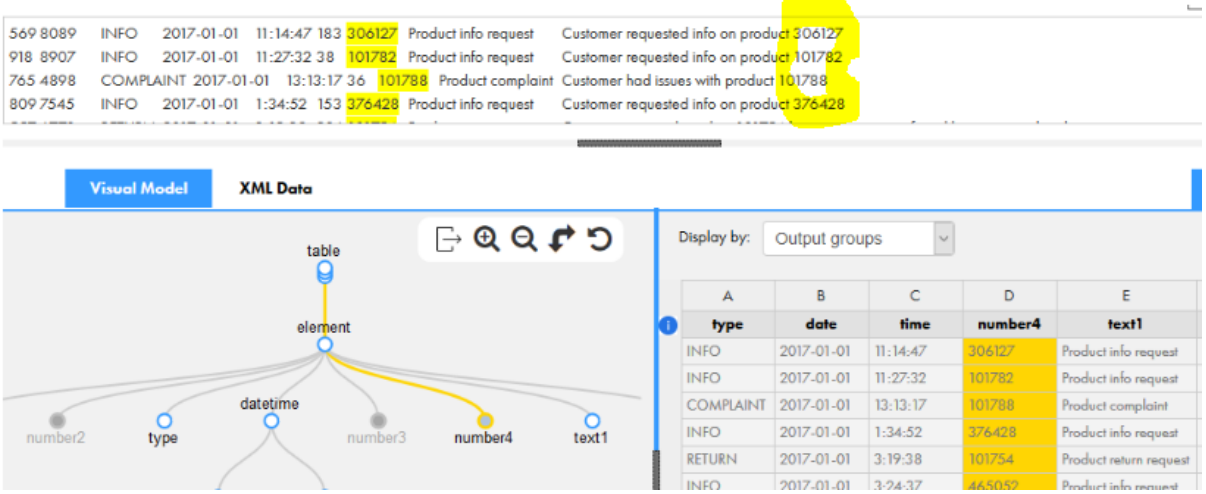


You can observe that **date** and **time** have been **correctly identified** by the **machine learning** algorithm

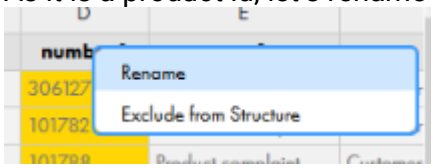
As column D: 'number3' doesn't yet make sense as a number, let's remove it by clicking **'Exclude from Structure'**



For next item 'number4' (now column D) we are lucky: the text tells us it is a product



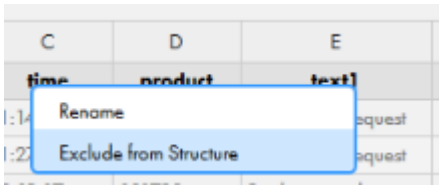
As it is a product id, let's rename it **product**



Select column E: 'text1'

A	B	C	D	E
type	date	time	product	text1
INFO	2017-01-01	11:14:47	306127	Product info request
INFO	2017-01-01	11:27:32	101782	Product info request
COMPLAINT	2017-01-01	13:13:17	101788	Product complaint
INFO	2017-01-01	1:34:52	376428	Product info request
RETURN	2017-01-01	3:19:38	101754	Product return request
INFO	2017-01-01	3:24:37	465052	Product info request

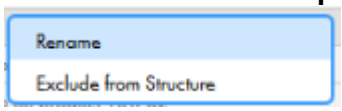
As it seems a description of type (not adding more info) let's remove it (text1)as well



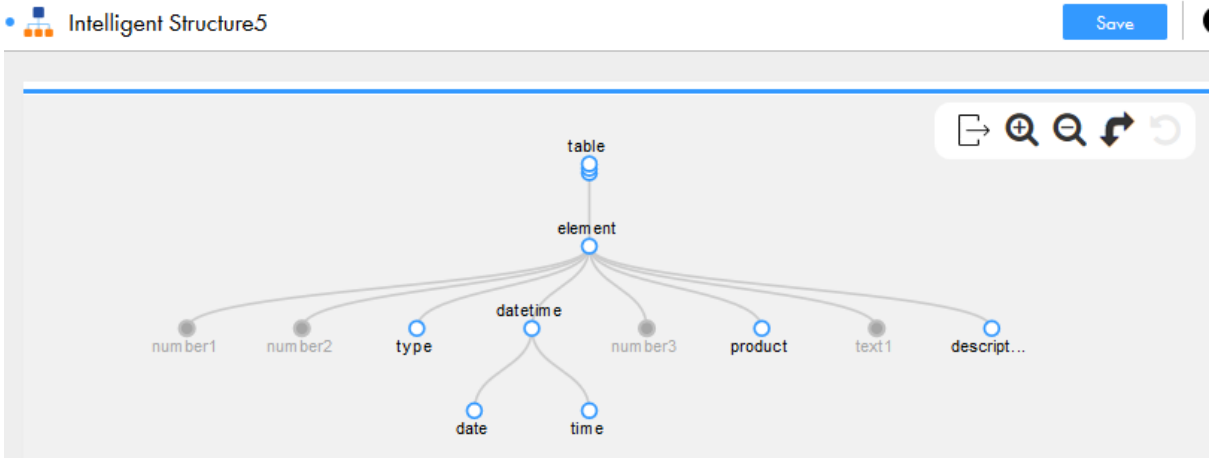
For now last columns (now E): 'text1'

D	E
product	text2
306127	Customer requested info on product 306127
101782	Customer requested info on product 101782
101788	Customer had issues with product 101788
376428	Customer requested info on product 376428
101754	Customer returned product 101754 because customer found better p...
465052	Customer requested info on product 465052
102121	Customer had issues with product 102121
85123	Customer requested info on product 85123

It seems to provide a more detailed description of the call reason
So let's rename it **description**

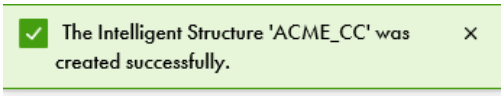


Close the Table view, you should see:

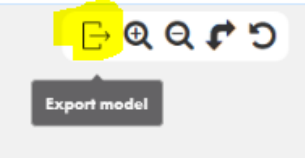


Click **Save**

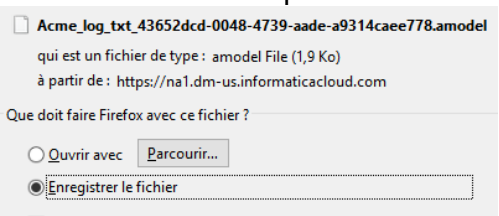
You should see:



Let's now export the model:



You will be asked to open or save the file like for example:



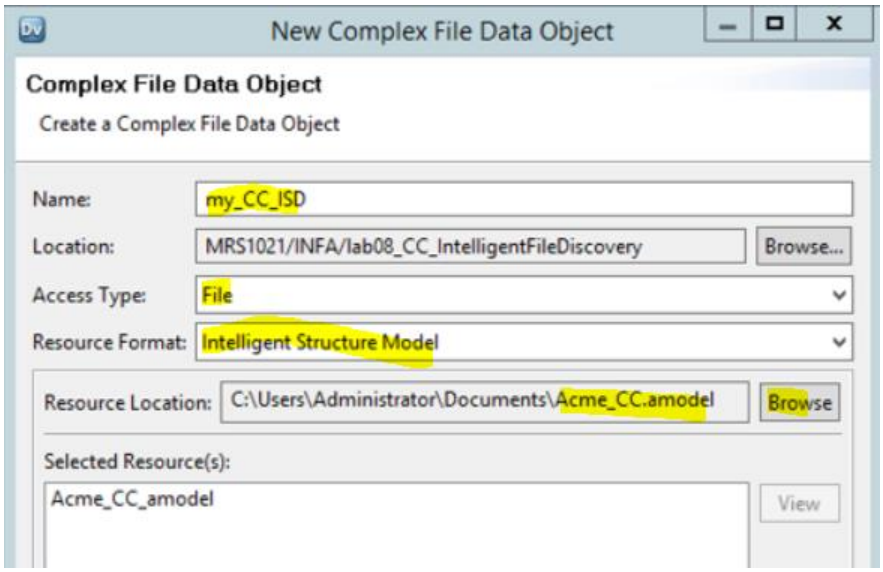
Save it to your Desktop and rename it to **Acme_CC.amodel**

For your convenience you can also Copy to your Desktop the resulting model:

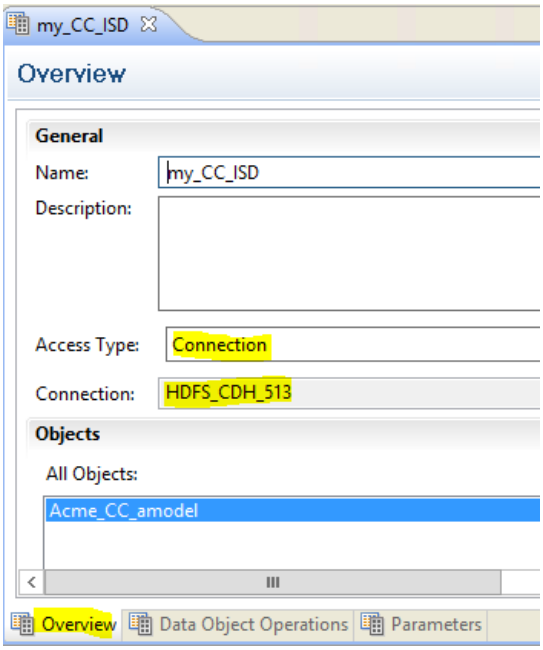
raw_callcenter_csv.amodel

10.4 Import & Run ISD Model into Developer

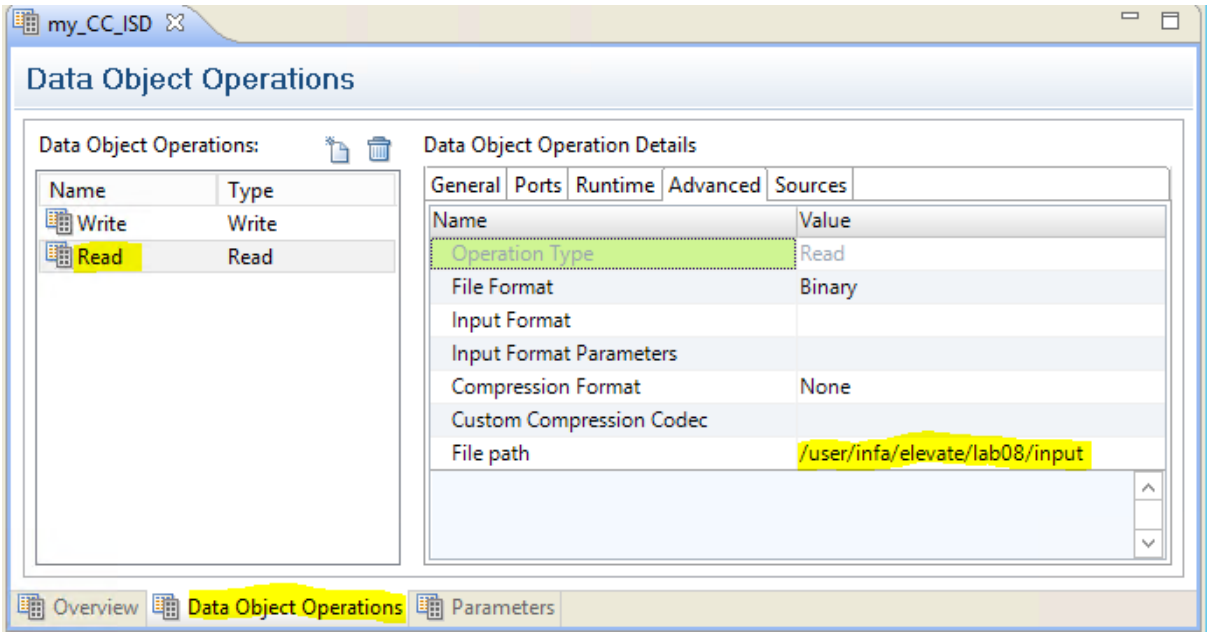
Open Developer, and go to lab8*/**my_folder**, create a new **Complex File Data Object**



Name it **my_CC_ISD**, choose **File** for Access type and **Intelligent Structure Model** for Resource Format and **Browse** to your Desktop/**Acme_CC.amodel**
Hit **Finish**

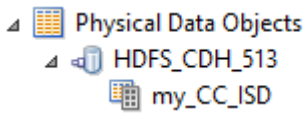


In the **Overview**, set Access Type as **Connection** and Browse to connection **HDFS_CDH_513**

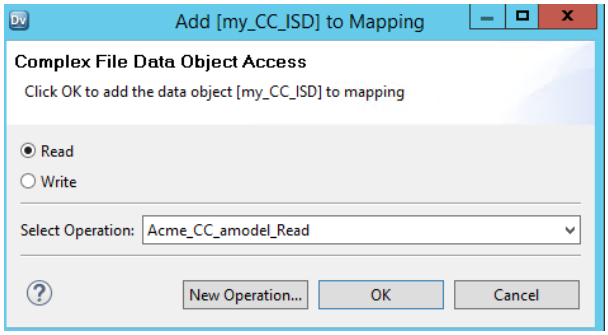


We need to position the Complex File to read to the HDFS Call center file.
To do so: go to **Data Object Operations**, select the **Read** operation then **Advanced** and enter the path: **/user/infa/elevate/lab08/input**

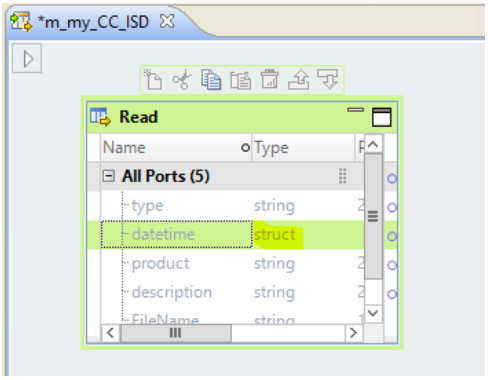
You should see:



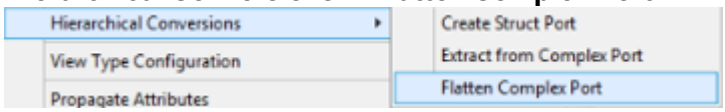
Create a mapping **m_my_CC_ISD** and Drag & drop **my_CC_ISD** Complex File as **Read** Object



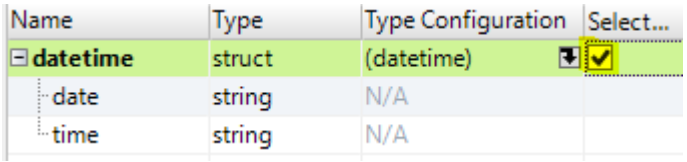
You should see:



Observe that datetime is of type struct (HType). Select datetime port, **right click** and chose **Hierarchical Conversions** -> **Flatten Complex Port**

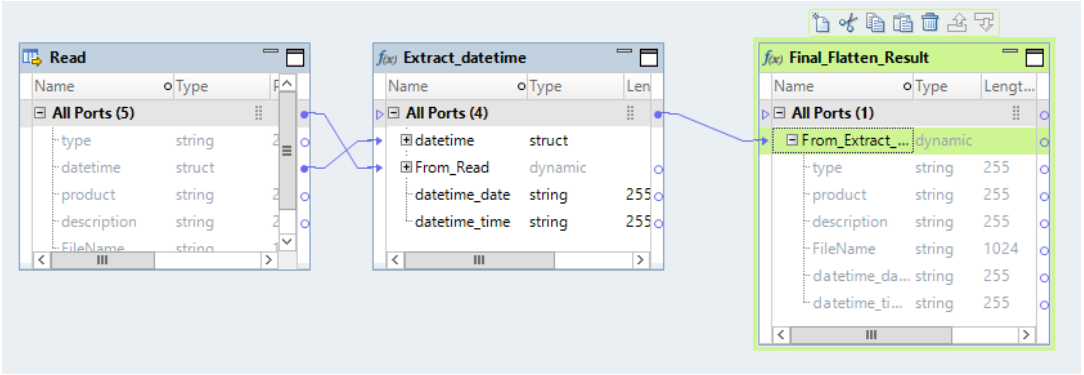


Then select datetime

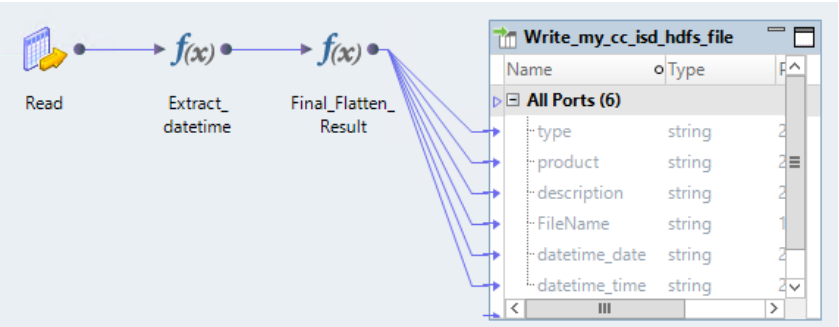


and hit Finish.

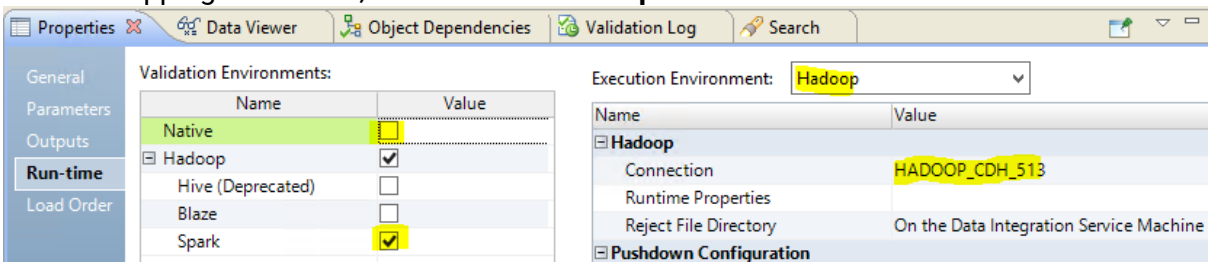
You should see:



Drag and Drop as **Write** Object existing HDFS target **my_cc_isd_hdfs_file** and auto link from last Expression. You should see:



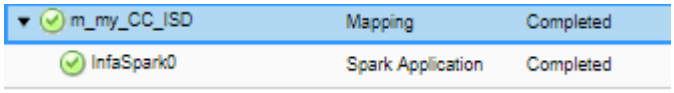
For the mapping to validate, we need to set it in **spark** mode:



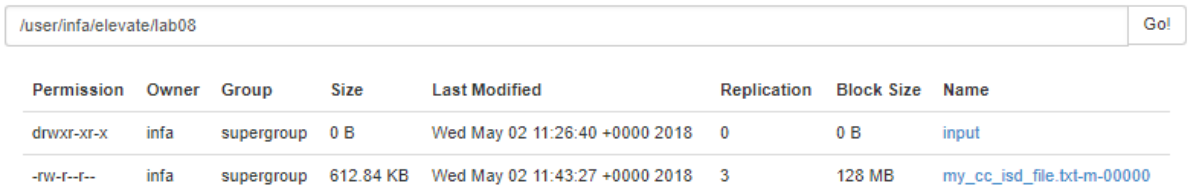
In the Run-time properties, unselect the Native Validation, select Spark and set the Hadoop Connection

Save & Run the mapping

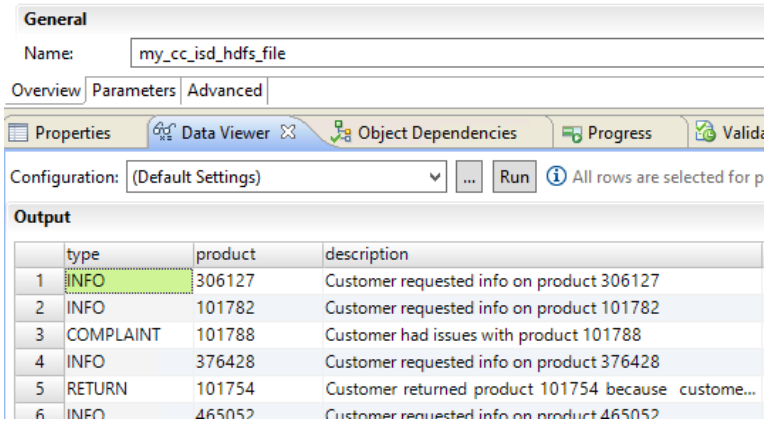
You should see in Informatica Monitoring Console:



Use HDFS Web Browser and navigate to /user/infa/elevate/lab08, you should see:



Run Data Viewer on target object, you should see:



11. Lab09 – Weblogs Integrate AWS RedShift & S3 (optional)

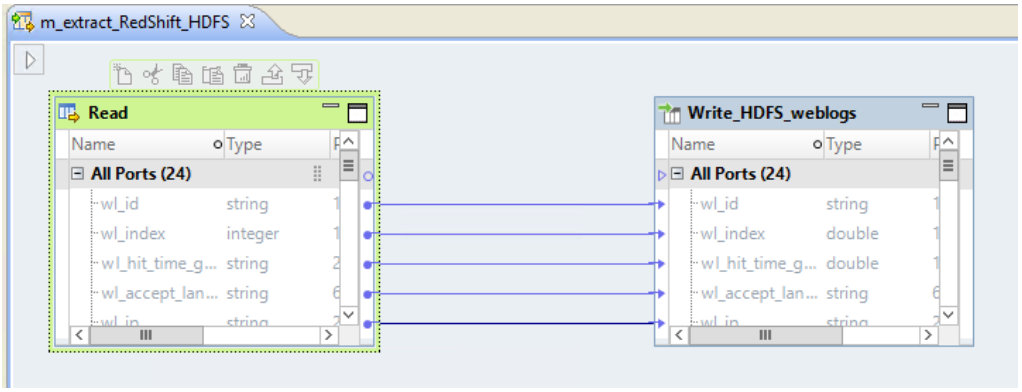
11.1 Purpose

Show AWS RedShift & S3 integration in native & push down

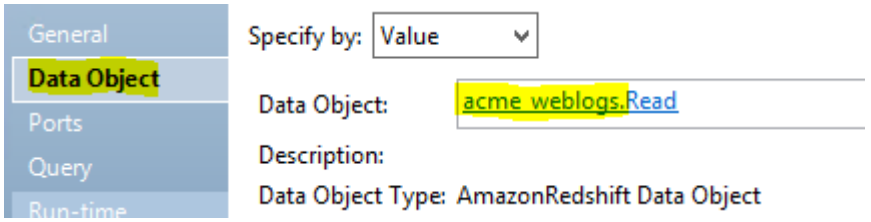
11.2 Review Lab Content

In Developer, go to ELEVATE/BDM_Labs/lab09_* folder

Open the passthrough mapping **m_extract_RedShift_HDFS**

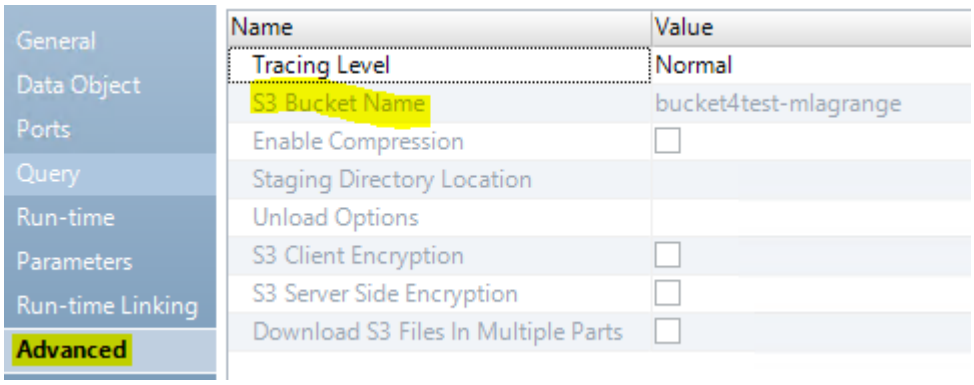


Select the **Read** object and go to Data Object



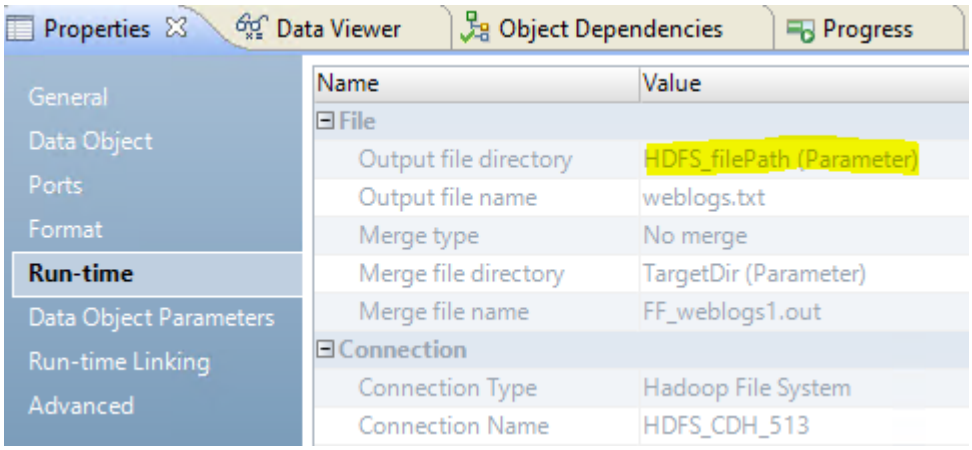
The source is a Redshift table called acme_weblogs

Go to **Advanced**



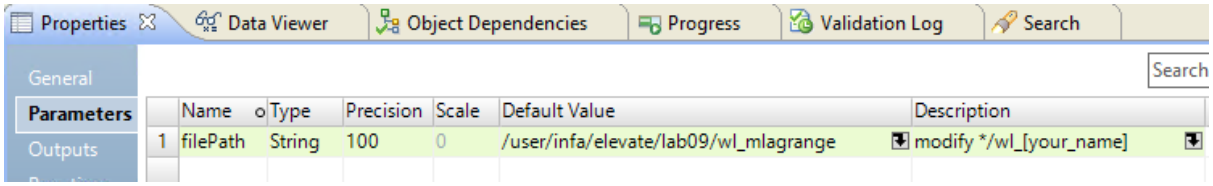
Note: for extracting the RedShift data we will store the data into the specified bucket name

Select the **Write_HDFS_weblogs** target object and go to **Run-time**



It uses a parameter for HDFS file path

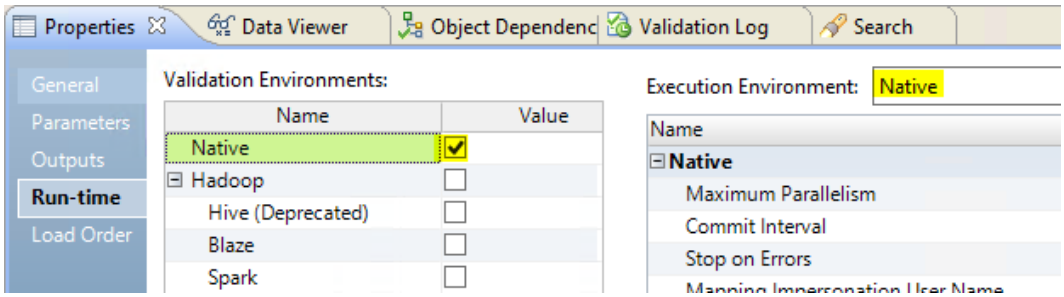
Go the **mapping parameters**



And change the default value /user/infa/elevate/lab09/wl_mlagrange to:

/user/infa/elevate/lab09/wl_your_name

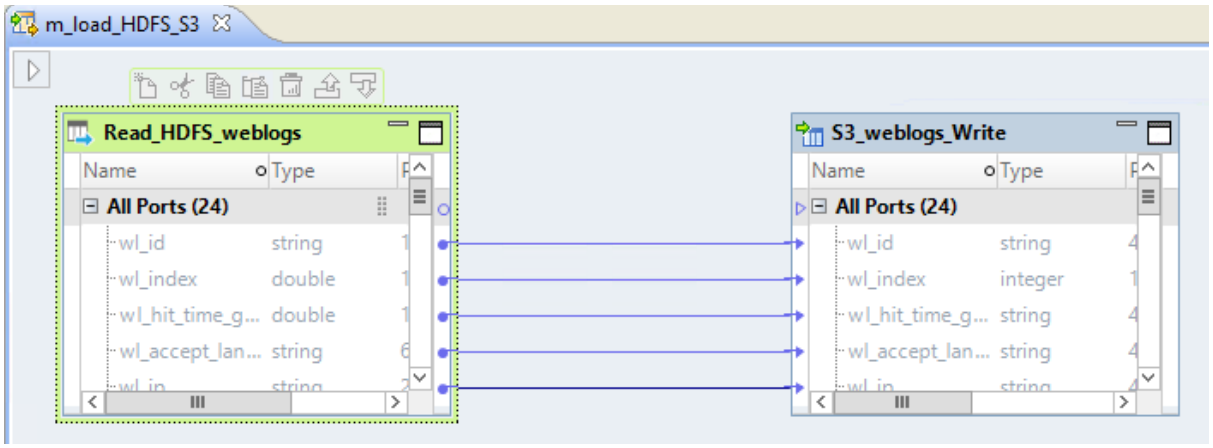
Go to the **Run-time** properties



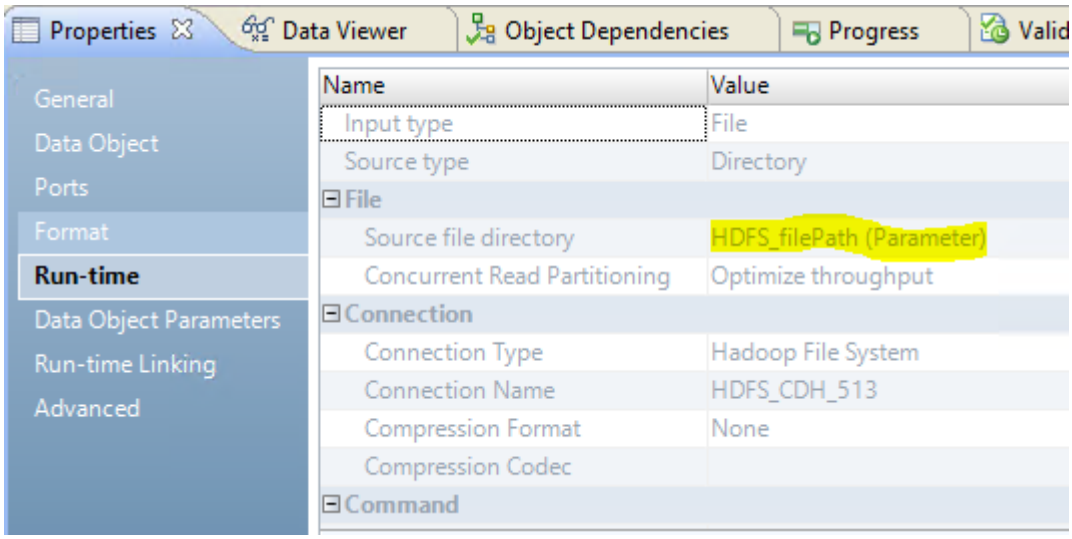
We will run this mapping in native mode.

Save the mapping

Open second mapping **m_load_HDFS_S3**

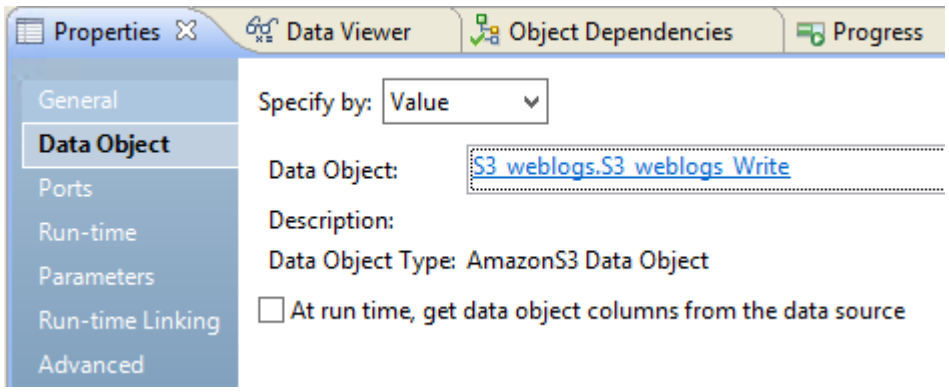


Select the **Read_HDFS_weblogs** source Object and go to **Run-time** properties



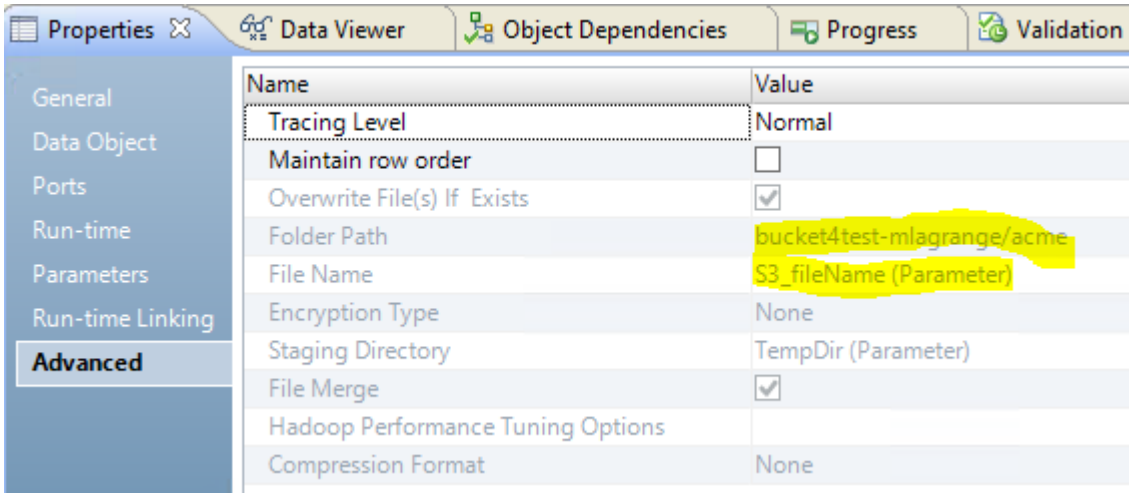
HDFS file path is using a parameter

Select the **S3_weblogs_Write** target Object and go to **Data Object**



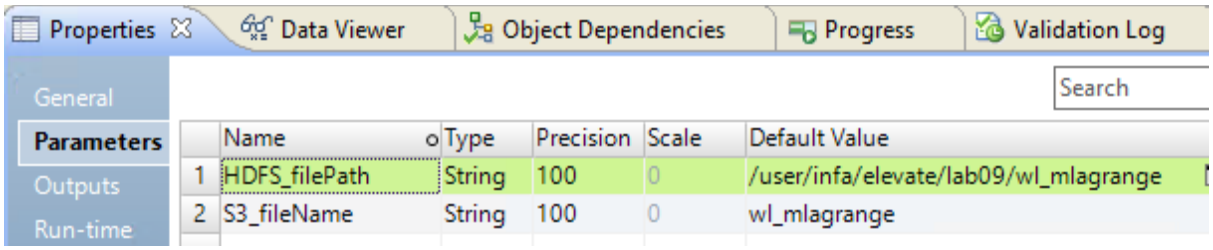
We will write to AWS S3 Object

Go to the **Advanced** properties



Observe the target S3 Bucket selected and note that the S3 File Name is a parameter

Go the **mapping parameters**

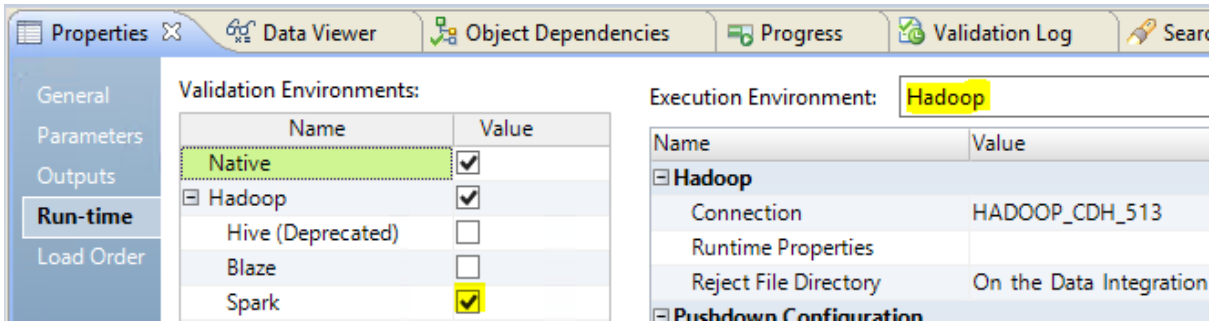


And change the default values to

HDFS_filePath: /user/infa/elevate/lab09/wl_your_name

S3_fileName: wl_your_name

Go to the **Run-time** properties



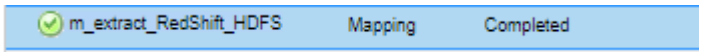
We will run this mapping in **spark** mode.

Save the mapping

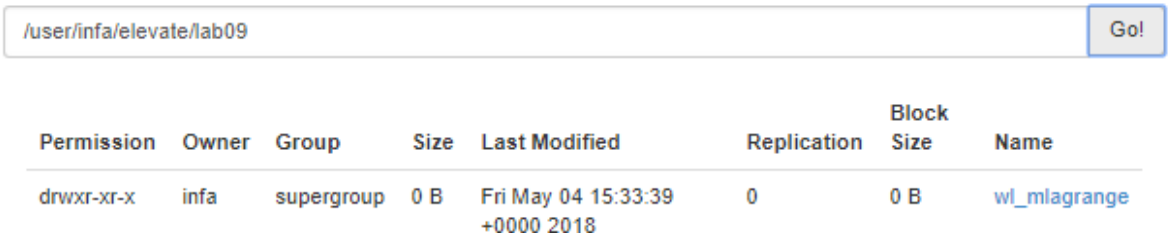
11.3 Run Lab

Run the mapping **m_extract_RedShift_HDFS**

You should see in Informatica Monitoring Console:

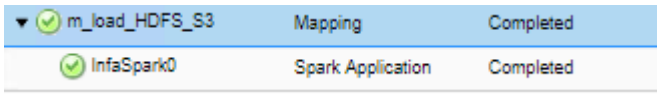


Use HDFS Web Browser and navigate to /user/infa/elevate/lab09, you should see:

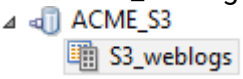


Run the mapping **m_load_HDFS_S3**

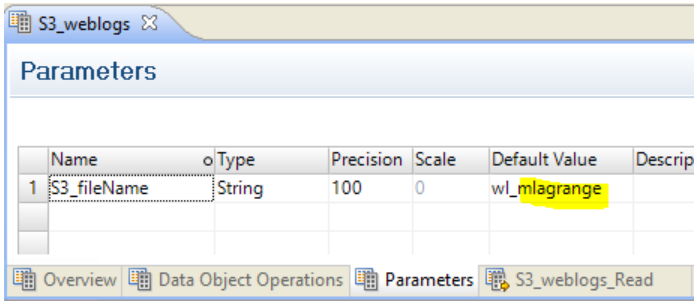
You should see in Informatica Monitoring Console:



Open the S3_weblogs object

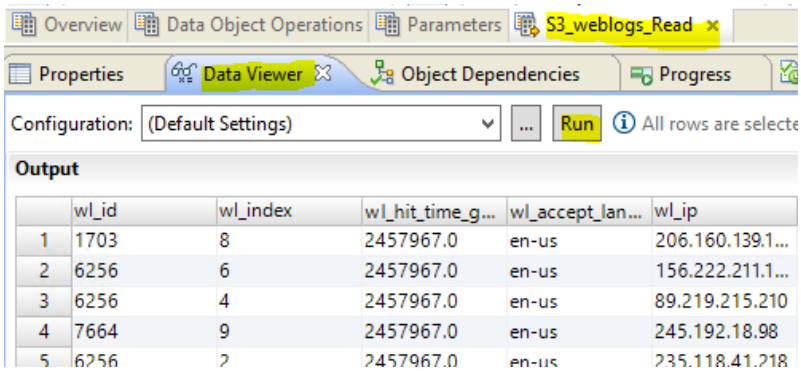


Go to parameters



Change the default value to: wl_your_name and **save** the object

Go to **S3_weblogs_Read** and Run the **Data Viewe**r, you should see:



12. Lab10a – BDS weblogs real-time processing

12.1 Purpose

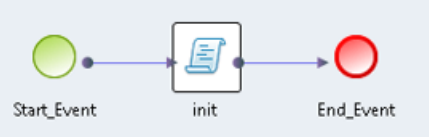
Real-time collection and processing of weblogs data.

ACME wants to be able to capture/analyze clickstreams of customer's interactions with their website and derive some useful real-time statistics:

- Monitor in real-time how many products are being browsed/ordered/returned
- Detect, for each product, if the ratio of daily returns vs orders is above 30%, and raise an alert if it does.
- Ingest the alert events explained above into the Data Lake for further offline historical analytics

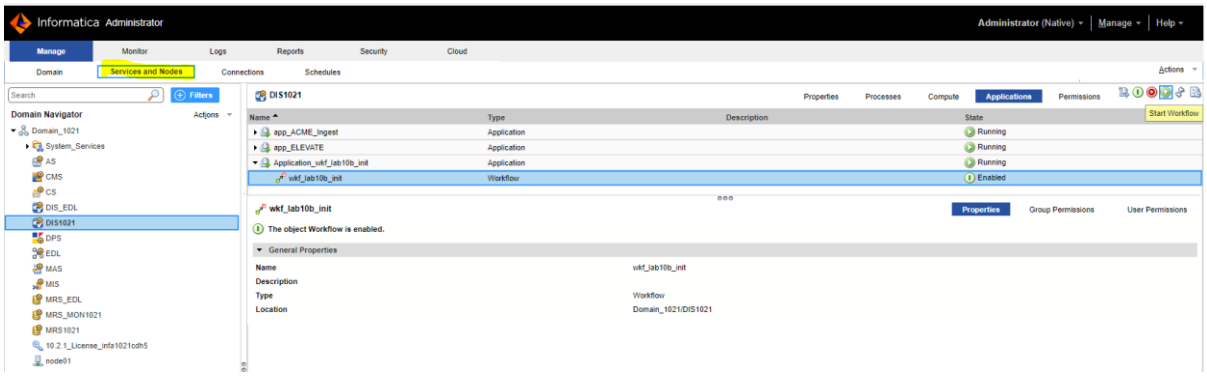
12.2 Initialize Lab

Workflow **wkf_lab10a_init** calls a script to reinitialize the lab

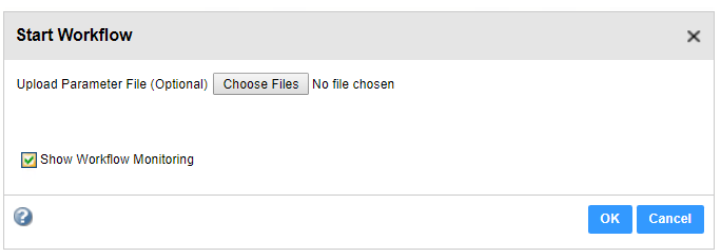


The **init** task command will reset target HBASE tables, create the required Kafka topics and clean Elastic Search indexes (for Kibana dashboarding).

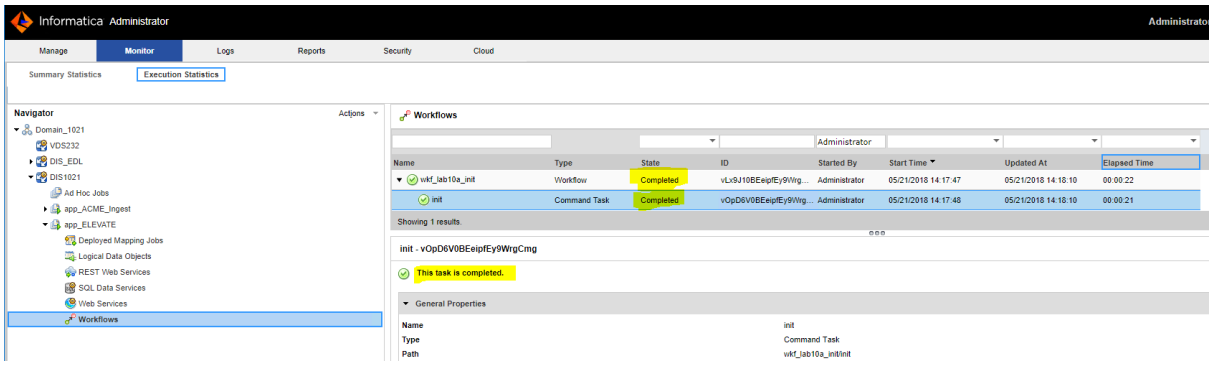
To initialize the lab, log on to the admin console and run the associated **wkf_lab10a_init** workflow; navigating under ***Manage -> Services and Nodes -> DIS1021*** and selecting the '**Applications**' tab view (expand the **app_ELEVATE** application from the list):



Do not upload any parameter file; check '**Show Workflow Monitoring**' and click **OK**



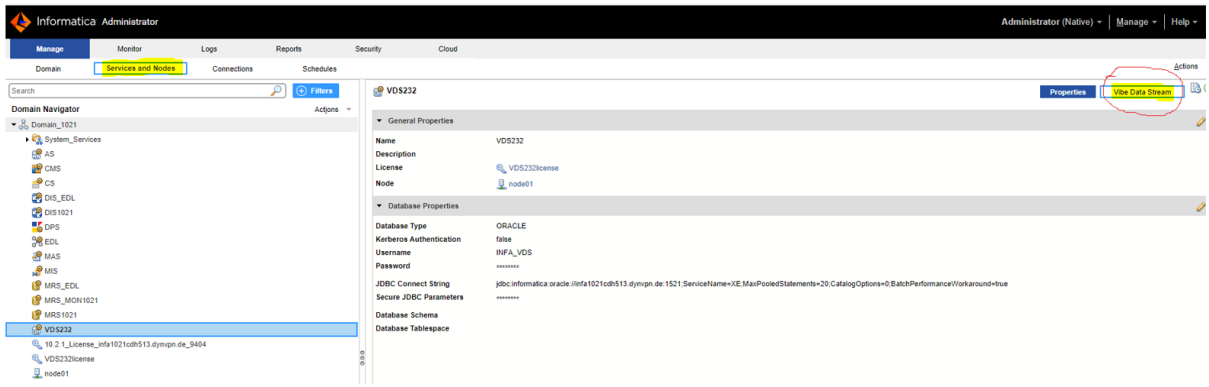
You should see:



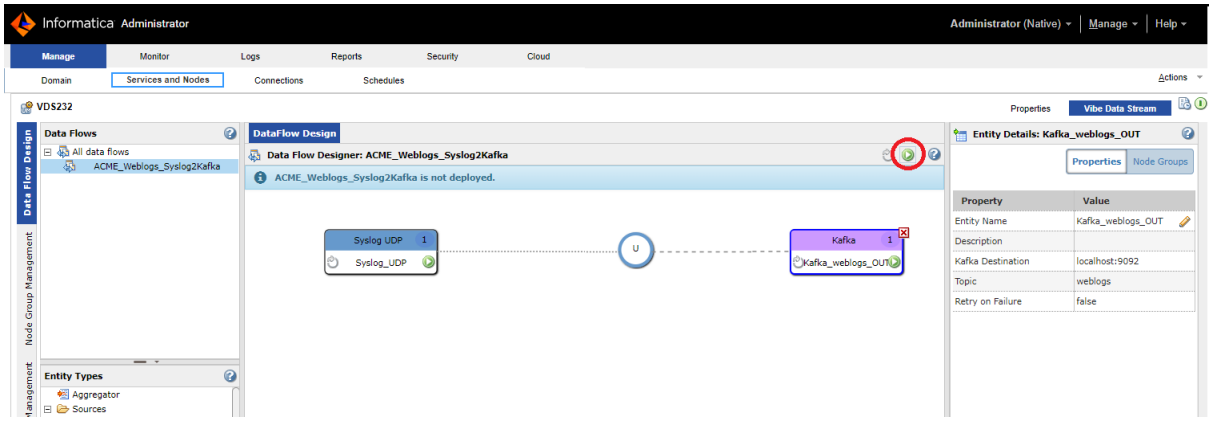
12.3 Start the Edge Data Streaming Flow

Informatica Edge Data Streaming (EDS) will be used to collect weblogs data published on Syslog protocol, format on-the-fly the unstructured weblog data message into JSON and publish it to the 'weblogs' Kafka topic (for BDS input).

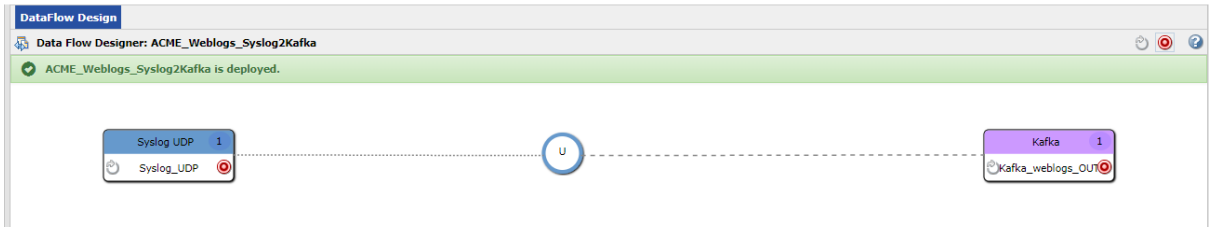
To deploy the EDS data flow, log into the Admin console and navigate to ***Manage -> Services and Nodes -> Vibe Data Stream*** (this is the former name of EDS):



You should see:

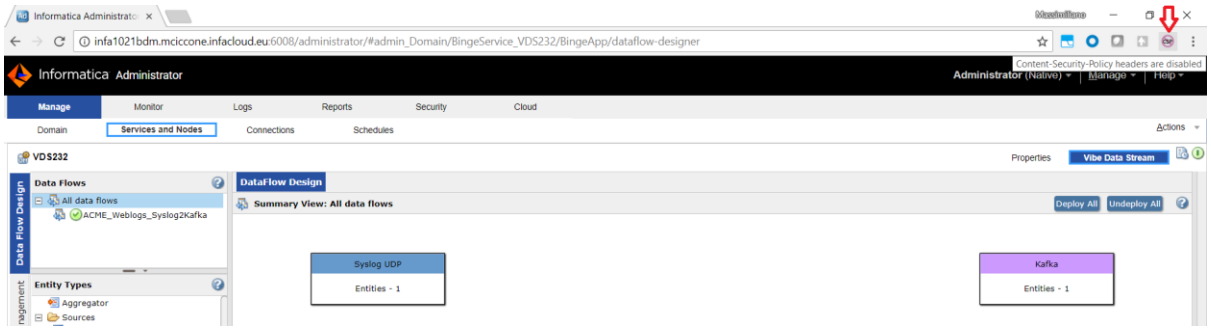


Deploy the EDS data flow (click top-right icon). You should see:

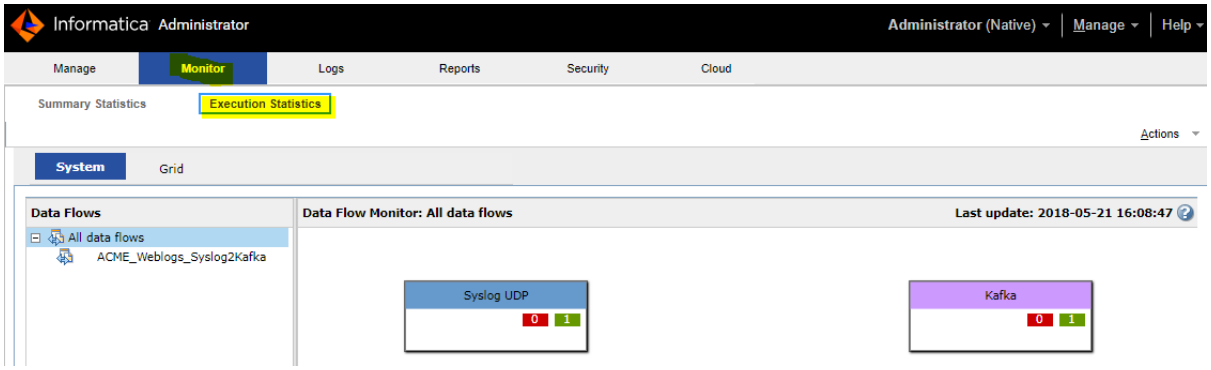


Now you can monitor the EDS data flow run-time statistics.

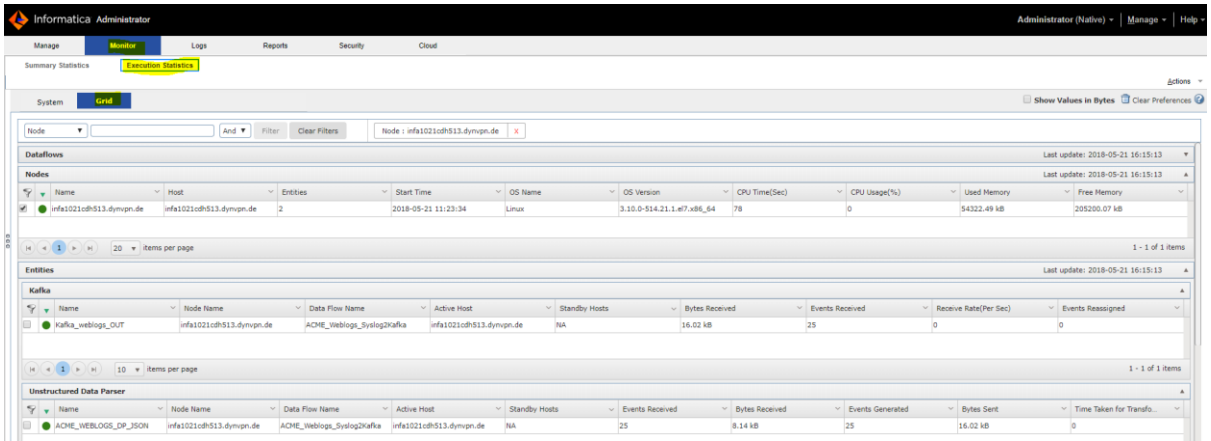
First, ensure that the Content-Security-Policy headers are disabled by clicking the Chrome plug-in icon on the top-right corner of the browser:



Then navigate to ***Monitor -> Execution Statistics***:



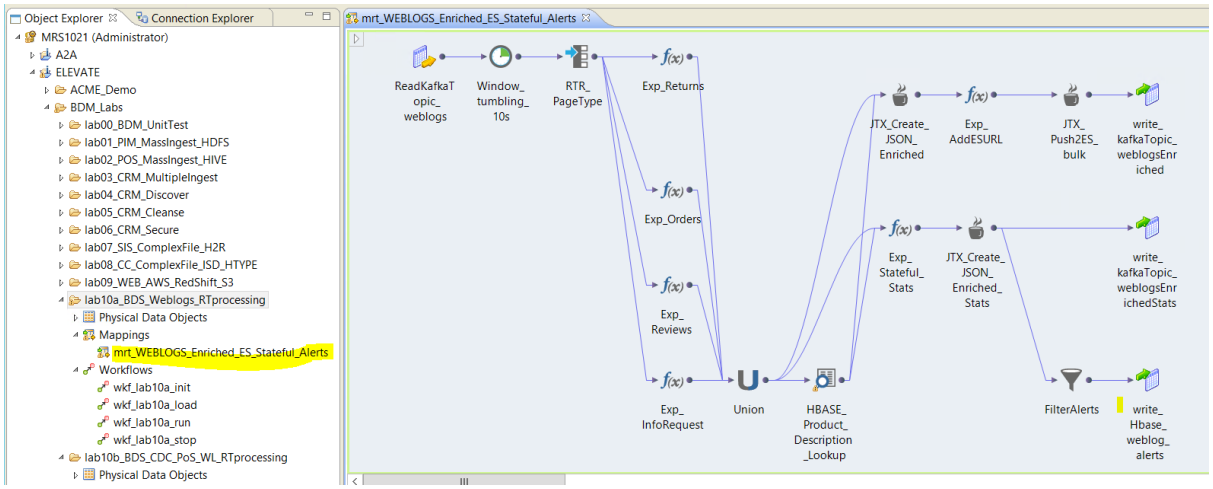
When selecting the 'Grid' view, you can monitor EDS runtime statistics on collected real-time data:



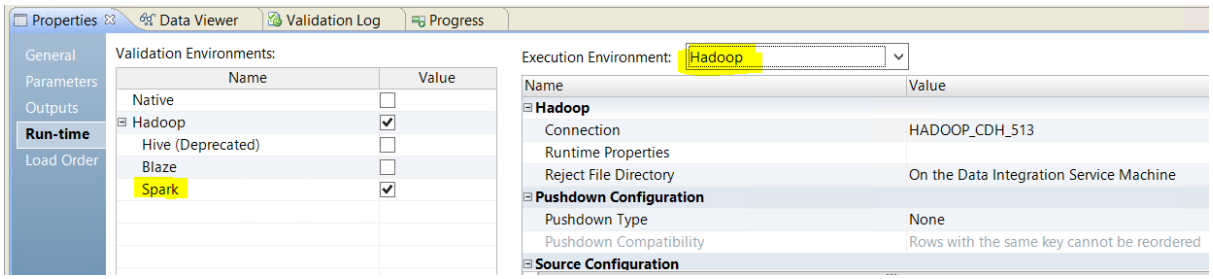
12.4 Review Lab Content

Open Developer, connect to MRS and open folder **ELEVATE/BDM_Labs/lab10a_BDS_Weblogs_RTprocessing**.

Then open the **mrt_WEBLOGS_Enriched_ES_Stateful_Alerts** mapping. You should see:

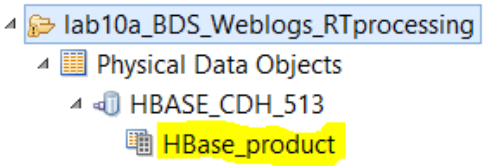


From the Run-time properties, you can see that this mapping is set for **Hadoop-Spark** Streaming execution mode:



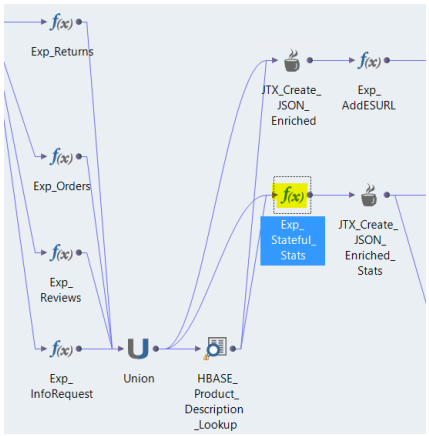
Which means that BDS will automatically translate it into Scala code for being executed by the Spark Streaming real-time processing engine.

In Developer, open the **HBASE_CDH_513** folder and observe the **source** HBase object:

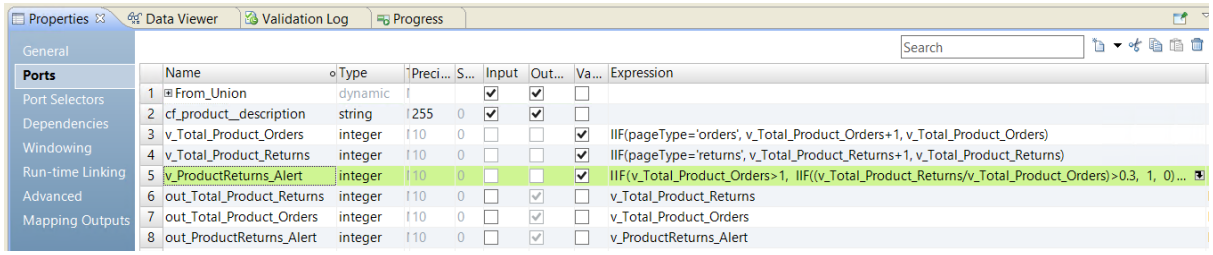


The **source** '**HBase_product**' is a look-up table used to fetch in real-time the description of weblog product IDs (the raw input weblog stream does not contain descriptions for products).

Now, observe the '**Exp_Stateful_Stats**' expression in the mapping:

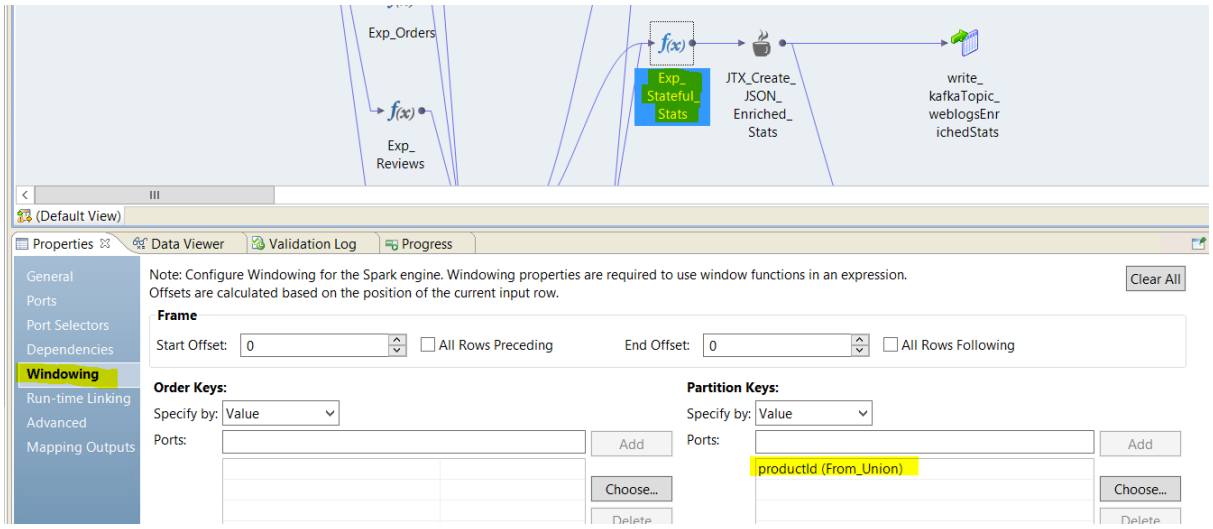


Select the ***'Properties->Ports***' view tab:



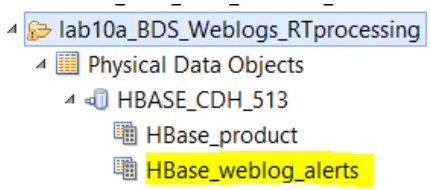
The **stateful variable** '**v_ProductReturns_Alert**' keeps track of the ratio of Orders vs returns for a specific product; if the returns are more than 30% of the orders an alert is raised (i.e. published to a Kafka topic and written to HBase).

Observe the **Windowing** of the stateful expression:



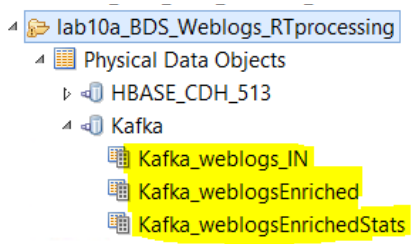
This is to group the state of variables by product id.

Observe the **target** HBase Data Object:



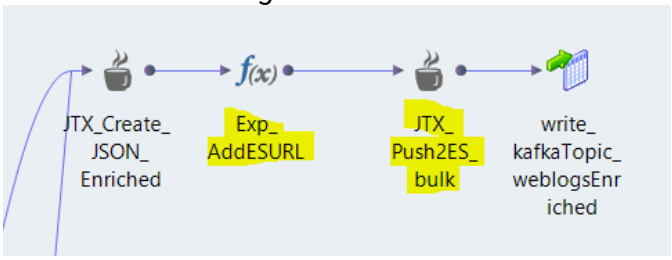
The **target 'HBase_weblog_alerts'** Data Object describes a table used to store the weblogs related alerts generated by the mapping.

Observe the Kafka Data Objects:



These Data Objects describe Kafka topics used by the real-time mapping for both input (i.e. weblogs) and output (i.e. weblogsEnriched, weblogsEnrichedStats).

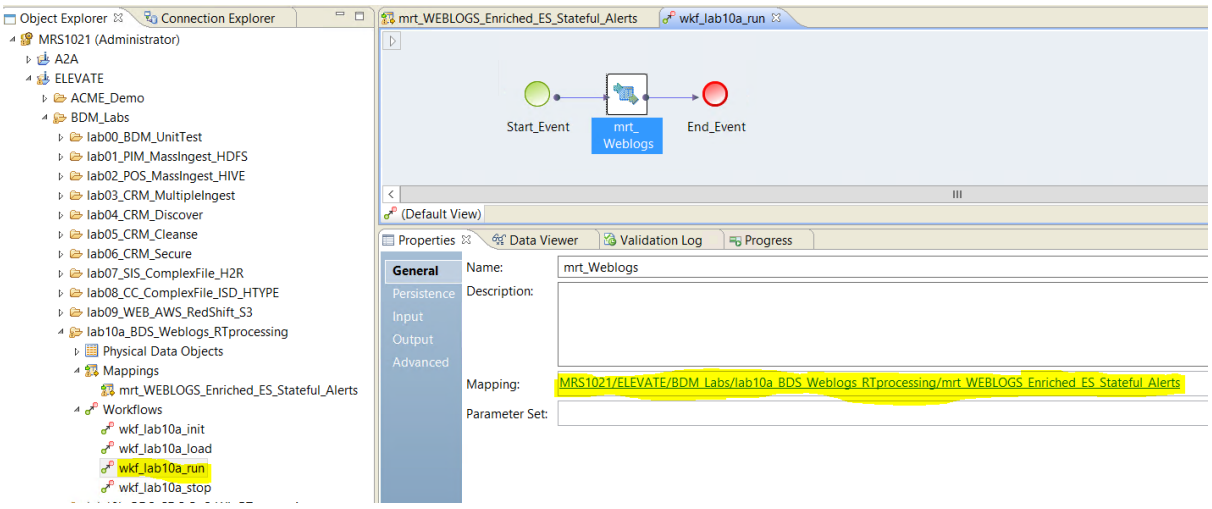
The enriched weblogs will also be indexed in real-time in **Elastic Search**:



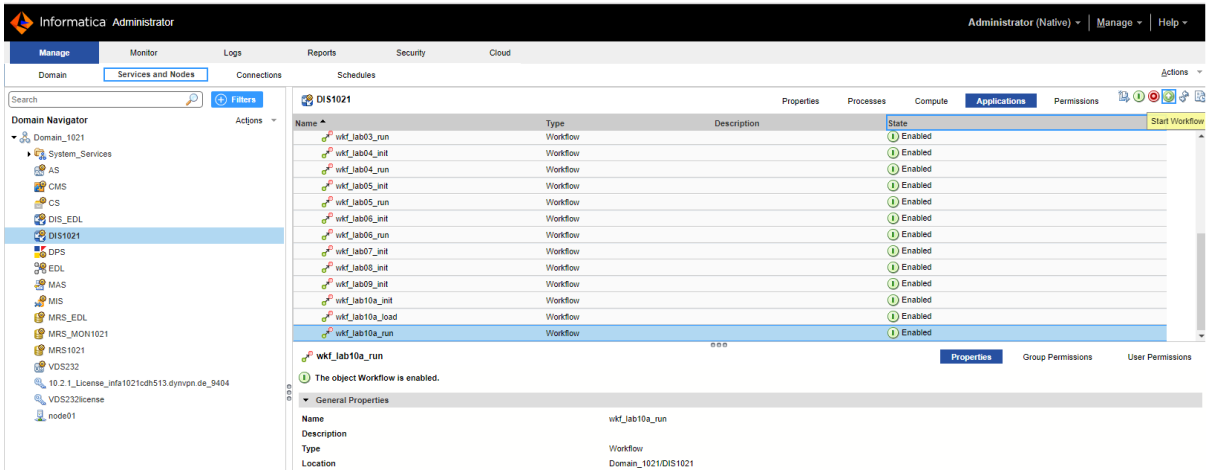
for being then displayed in a **Kibana Dashboard.** (see 'Observe Outcome' section below)

12.5 Run the Lab

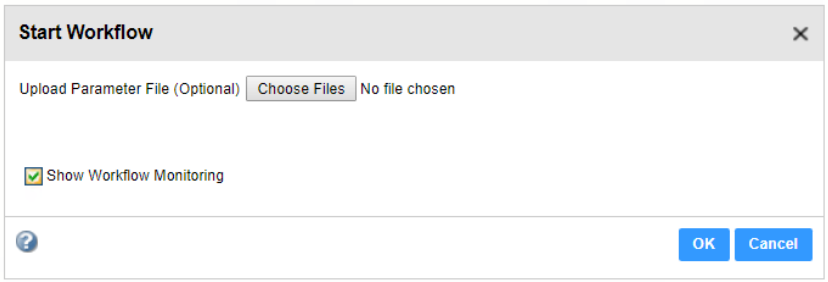
Workflow **wkf_lab10a_run** runs the **mrt_WEBLOGS_Enriched_ES_Stateful_Alerts** mapping.



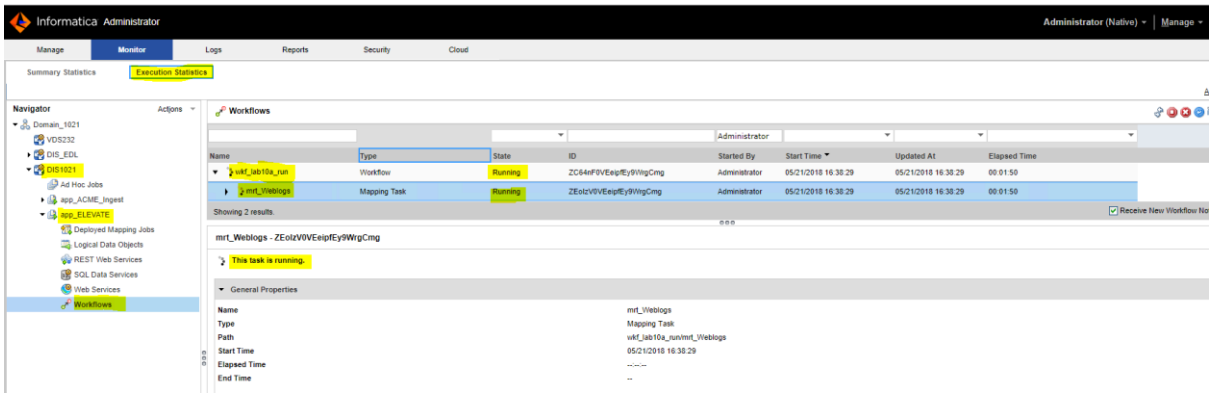
To run **lab10a** mapping log on to the admin console and run the **wkf_lab10a_run** workflow navigating under ***Manage -> Services and Nodes -> DIS1021*** and selecting the '**Applications**' tab view (expand the **app_ELEVATE** application from the list):



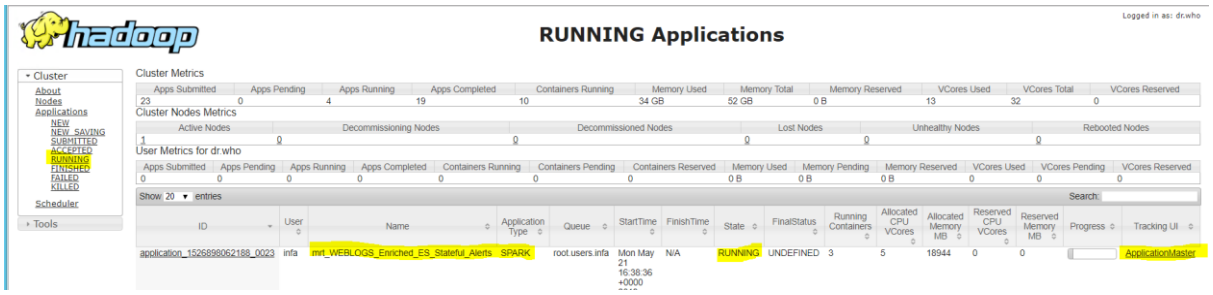
Click the Start Workflow icon on the top-right, check ***Show Monitoring*** and click **OK**:



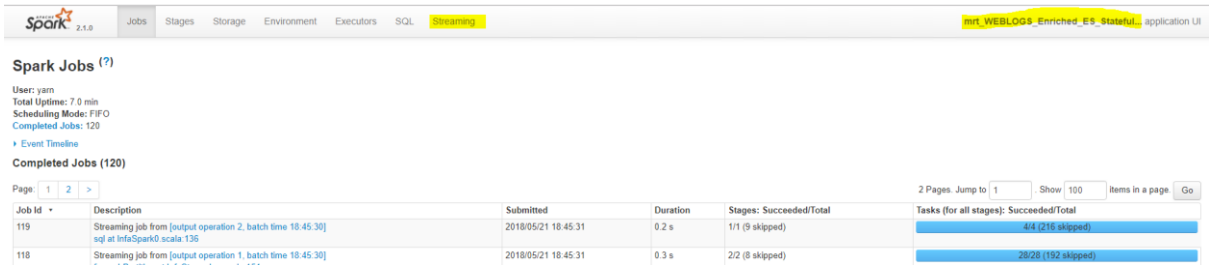
Under ***Monitor->Execution Statistics->DIS1021->app_ELEVATE->workflows***, you should see:



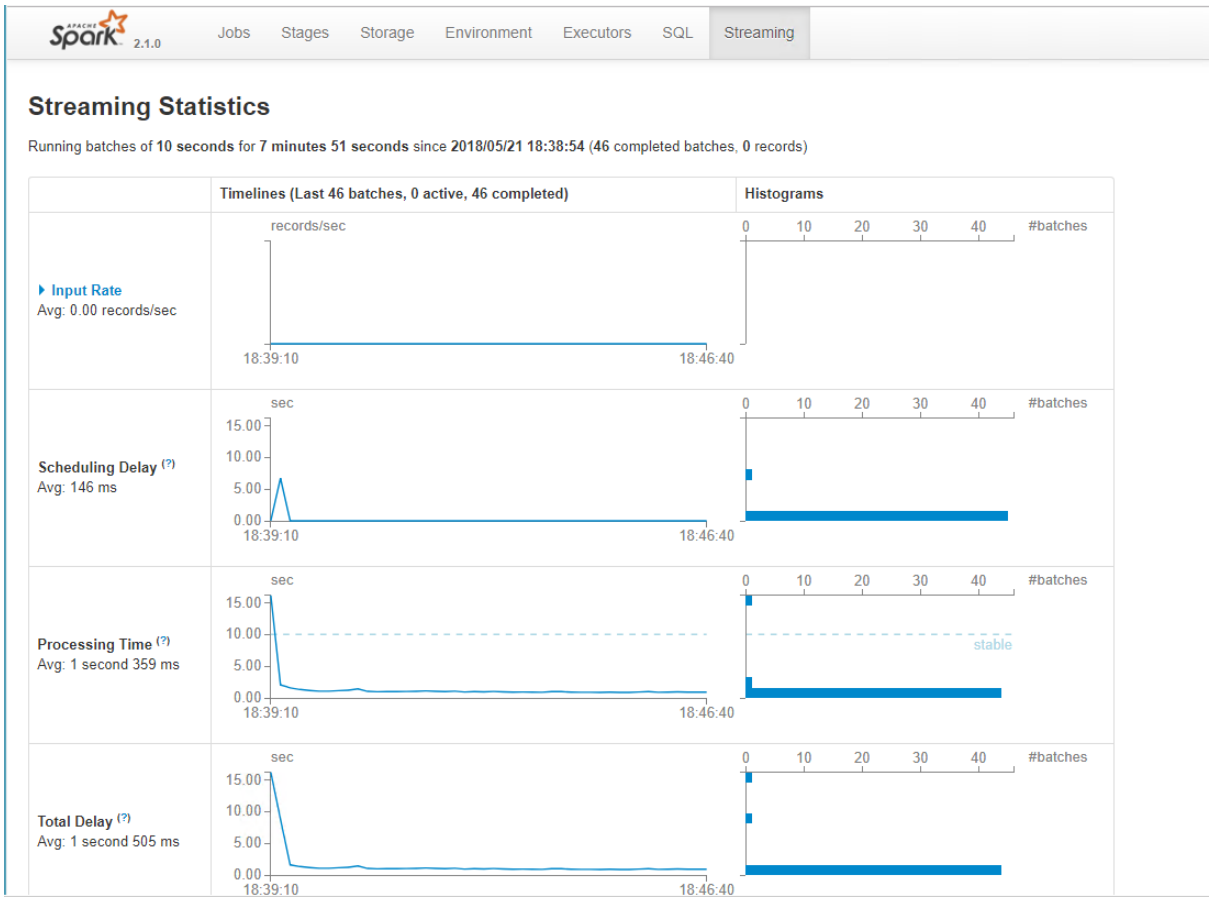
In YARN Monitor you can view the runtime statistics of the Spark Streaming Application execution (http://bdm.localdomain:8088/cluster/apps/RUNNING):



Note: It can take few minutes before the Spark Streaming job is deployed on the cluster for execution. Once the job is in RUNNING state, click **ApplicationMaster** on the right; you should see:



Click the '**Streaming**' tab on the top right; you should see all the runtime statistics for the Spark Streaming job associated to the BDS mapping logic:

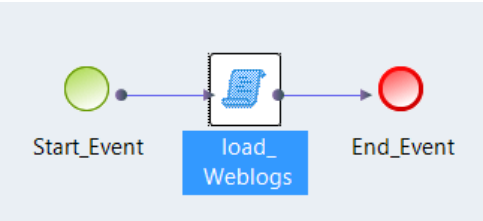


Note: It may take couple of minutes for the Streaming tab to appear.

The **Lab10a**'s BDS mapping is now ready to ingest and process real-time data.

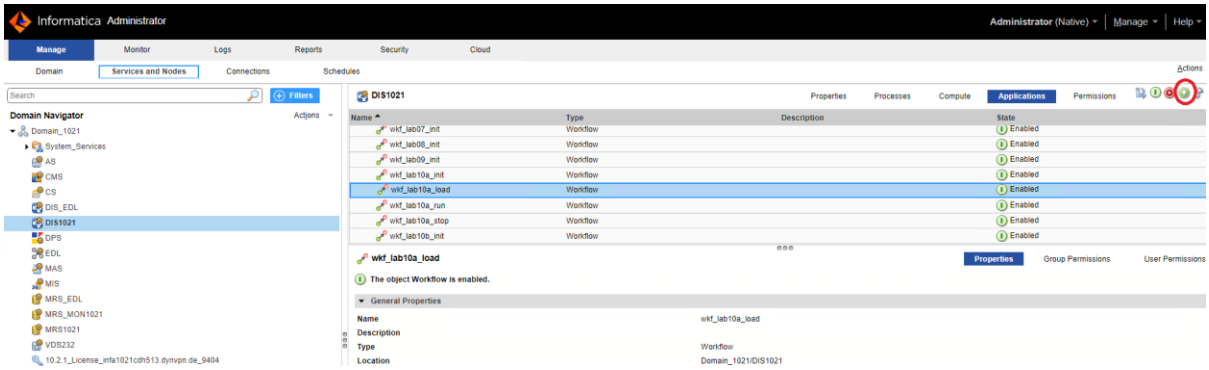
12.6 Load Real-time Mapping Data

Workflow **wkf_lab10a_load** loads real-time data streams to the **mrt_WEBLOGS_Enriched_ES_Stateful_Alerts** mapping.

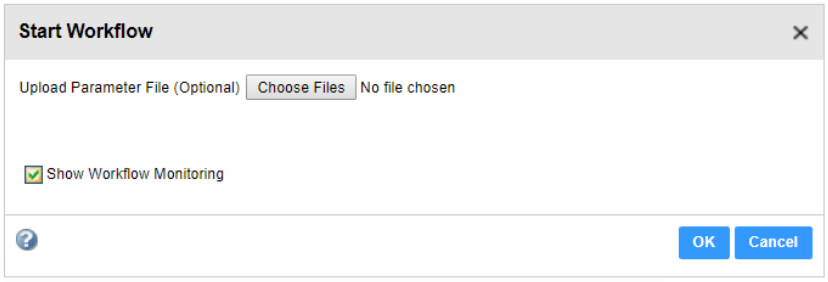


It will start a Python script that will produce a real-time data stream of raw weblogs data over a Syslog protocol (for EDS input); the EDS data flow will then produce a Kafka output stream on the '**weblogs**' topic (for BDS input).

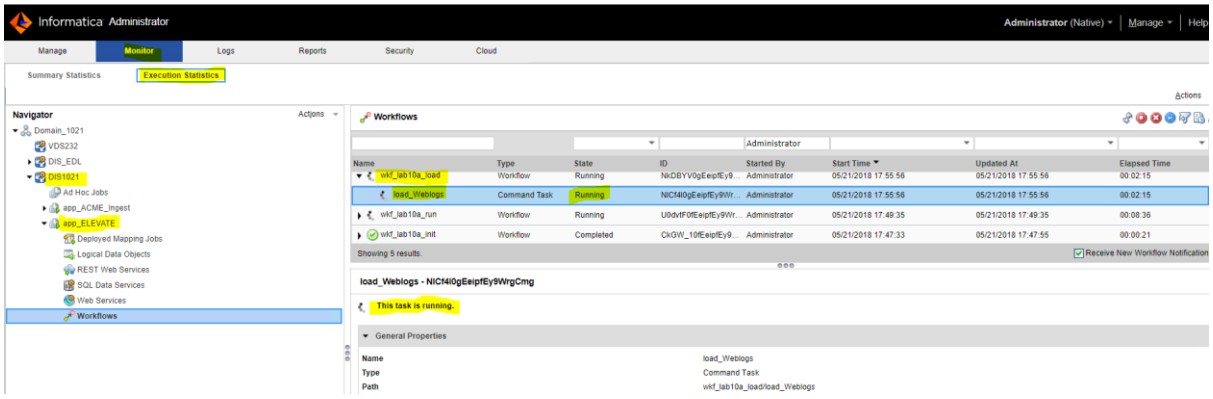
To load data to **lab10a** mapping log to the admin console and run the **wkf_lab10a_load** workflow:



Do not upload any parameter file; check **Show Workflow Monitoring** and click **OK**:

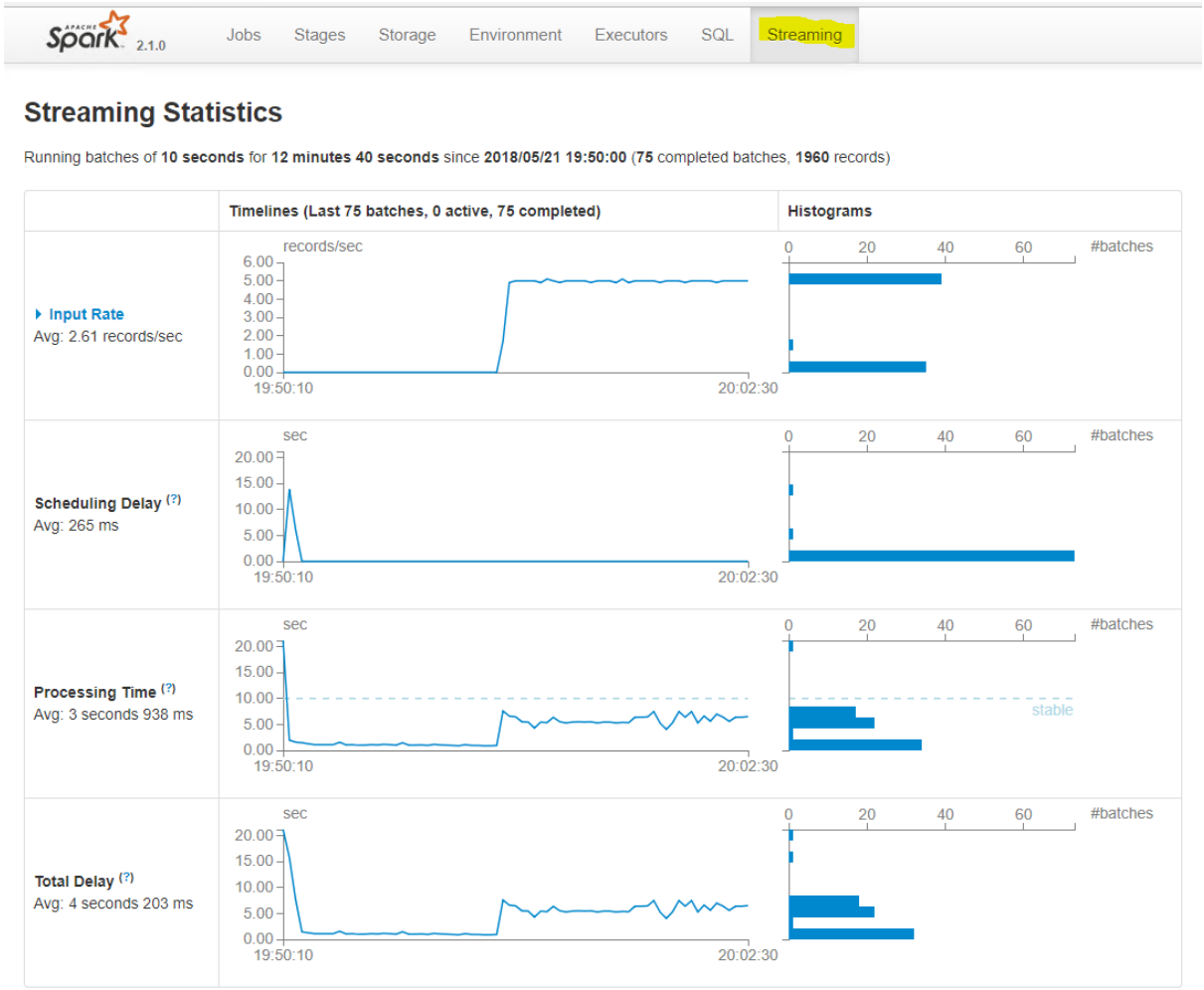


Under ***Monitor->Execution Statistics->DIS1021->app_ELEVATE->workflows***, you should see:



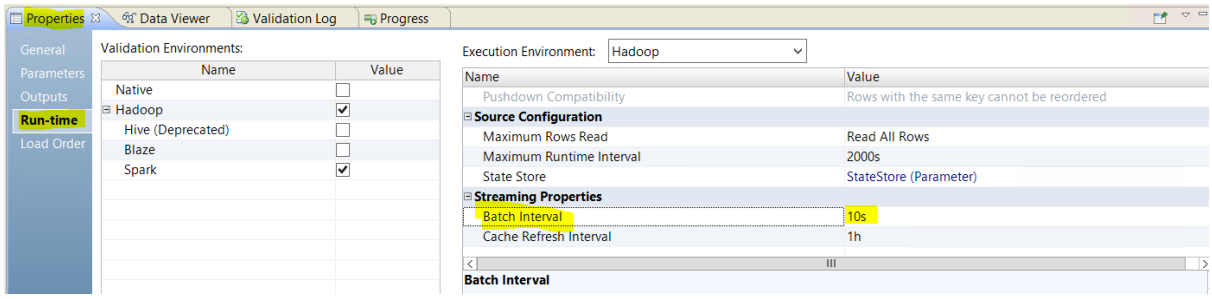
Real-time Data should be now flowing into EDS and consequently into BDS.

You can refresh the runtime statistics of the Spark Streaming Application associated to the BDS mapping by clicking the '**Streaming**' tab in the YARN application web UI:

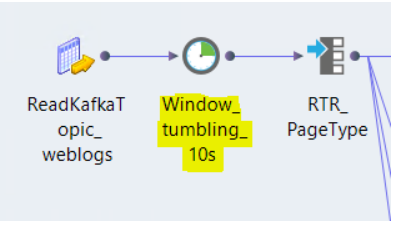


Note: The Processing Time should be constantly below the mapping batch interval (i.e. 10 seconds). This will avoid the Scheduling Delay of processing new record batches to increase and eventually, on the long run, to cause the Spark Streaming job to fail. Therefore, it is a good rule of thumb to set the BDS mapping batch interval to a long enough period for a single records batch processing to finish.

The BDS mapping's **Batch Interval** can be observed under the ***Properties->Run-Time*** view tab:



This is different from the **windowing period** used for the streaming mapping (e.g. either Tumbling or Sliding), which further groups streaming records before processing them (and must be a multiple of the Run-Time batch interval):



12.7 Observe Outcome

The '**weblog_alerts**' HBASE table is storing all the alerts coming from the stateful statistics computed on the weblogs stream in the BDS mapping.
It can be viewed using the following HBase shell command:

```
$ hbase shell
> scan 'weblog_alerts'
```

```
[infa@infa1021cdh513 ~]$ hbase shell
Java HotSpot(TM) 64-Bit Server VM warning: Using incremental CMS is deprecated and will likely be removed in a future release
18/05/21 17:58:43 INFO Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.0-cdh5.13.1, rUnknown, Thu Nov  9 08:47:05 PST 2017

hbase(main):001:0> scan 'weblog_alerts'
ROW                                   COLUMN+CELL
 0487b520-6d43-4501-ad0e-407f8d4dcd06 column=cf_weblog_alerts:Alert, timestamp=1526918206244, value={"pageType":"orders","productId":"455250","Pr
                                      oductDescription":"Brother Laserprinter HL-2130","Total_Product_Returns":1,"Total_Product_Orders":2,"Produc
                                      tReturnsAlertID":1}
 249e336d-0d02-4227-a864-30f733592192 column=cf_weblog_alerts:Alert, timestamp=1526918235108, value={"pageType":"orders","productId":"414073","Pr
                                      oductDescription":"Wacom Bamboo Pen & Touch","Total_Product_Returns":1,"Total_Product_Orders":3,"ProductRet
                                      urnsAlertID":1}
 57230c64-1586-4f72-a6be-d4c71cc8789c column=cf_weblog_alerts:Alert, timestamp=1526918166372, value={"pageType":"orders","productId":"950349","Pr
                                      oductDescription":"Zotac ZBOX AD02 Barebone","Total_Product_Returns":1,"Total_Product_Orders":2,"ProductRet
                                      urnsAlertID":1}
 701dc920-cc76-4855-b48e-e9d30aa128b6 column=cf_weblog_alerts:Alert, timestamp=1526918235107, value={"pageType":"orders","productId":"102022","Pr
                                      oductDescription":"Razer Goliathus Speed Extended XL","Total_Product_Returns":1,"Total_Product_Orders":3,"P
                                      roductReturnsAlertID":1}
 721d822f-90ef-469a-9094-2aed5fae64a5 column=cf_weblog_alerts:Alert, timestamp=1526918226146, value={"pageType":"orders","productId":"102022","Pr
                                      oductDescription":"Razer Goliathus Speed Extended XL","Total_Product_Returns":1,"Total_Product_Orders":2,"P
                                      roductReturnsAlertID":1}
 8bbfc052-6ee5-4c47-8560-f161bc568da6 column=cf_weblog_alerts:Alert, timestamp=1526918318157, value={"pageType":"returns","productId":"620793","P
                                      roductDescription":"SteelSeries QcK Diablo 3 Witch Doctor Edition","Total_Product_Returns":1,"Total_Product
                                      _Orders":2,"ProductReturnsAlertID":1}
 9a2190da-0942-42be-bd1e-9564ddf9f831 column=cf_weblog_alerts:Alert, timestamp=1526918116317, value={"pageType":"orders","productId":"253214","Pr
                                      oductDescription":"Acer PC Aspire X1430 E-300/4/500/W7HP","Total_Product_Returns":1,"Total_Product_Orders":
                                      3,"ProductReturnsAlertID":1}
 ab9b32fa-7dd3-47ee-b4f0-a96d24ce184b column=cf_weblog_alerts:Alert, timestamp=1526918346130, value={"pageType":"orders","productId":"950349","Pr
                                      oductDescription":"Zotac ZBOX AD02 Barebone","Total_Product_Returns":1,"Total_Product_Orders":3,"ProductRet
                                      urnsAlertID":1}
```

Weblog alerts are generated when, for a specific product, the number of returns is above 30% of orders.

The **weblogsEnrichedStats** Kafka topic contains the outcome of the stateful statistics computations of the per-product orders vs returns (for bad products monitoring).
You can monitor the content of this topic (while the BDS mapping is running) by executing the following shell alias command on the server machine:

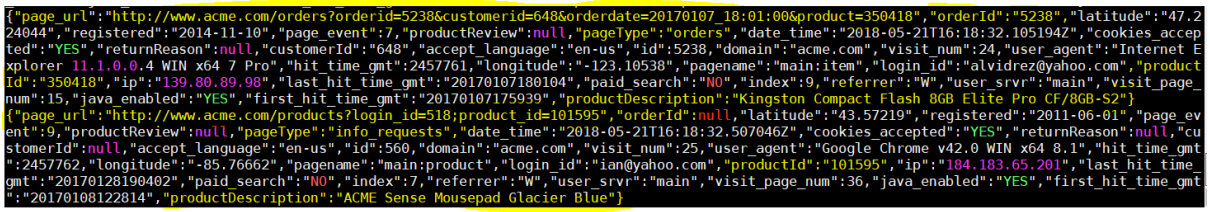
```
$ kafkaConsumer weblogsEnrichedStats
```

```
{"productId":"253290","ProductDescription":"Asus Netbook R051BX-BLK033S C60/10.1'/1/320/W7S","pageType":"orders","Total_Product_Returns":0
,"Total_Product_Orders":1,"ProductReturns_alertID":0}
{"productId":"253218","ProductDescription":"Asus PC CP6230-i3/4/1TB/W7HP/6471-1GB","pageType":"orders","Total_Product_Returns":0,"Total_Pr
oduct_Orders":2,"ProductReturns_alertID":0}
{"productId":"101891","ProductDescription":"Logitech Mouse M125 White","pageType":"orders","Total_Product_Returns":0,"Total_Product_Orders
":1,"ProductReturns_alertID":0}
{"productId":"231311","ProductDescription":"Targus Notebook Skin 16'' Impax Neoprene Blue","pageType":"orders","Total_Product_Returns":0,"
Total_Product_Orders":1,"ProductReturns_alertID":0}
{"productId":"181152","ProductDescription":"Sweex Notebook Mouse Silver USB .MI102.","pageType":"orders","Total_Product_Returns":0,"Total_
Product_Orders":1,"ProductReturns_alertID":0}
{"productId":"455250","ProductDescription":"Brother Laserprinter HL-2130","pageType":"orders","Total_Product_Returns":2,"Total_Product_Ord
ers":3,"ProductReturns_alertID":1}
{"productId":"253246","ProductDescription":"LG 23' D2342P-PN LED 3D met gratis bril","pageType":"orders","Total_Product_Returns":0,"Total_
Product_Orders":1,"ProductReturns_alertID":0}
{"productId":"227239","ProductDescription":"ACME PREMIUM i3-2100","pageType":"orders","Total_Product_Returns":0,"Total_Product_Orders":2,"
ProductReturns_alertID":0}
```

Also, the **weblogsEnriched** Kafka topic contains the weblogs records enriched by parsing the raw input stream URLs, producing some additional fields and looking up the product

descriptions. It can be viewed in real-time (while the BDS mapping is running) with the following shell alias command on the infa server machine:

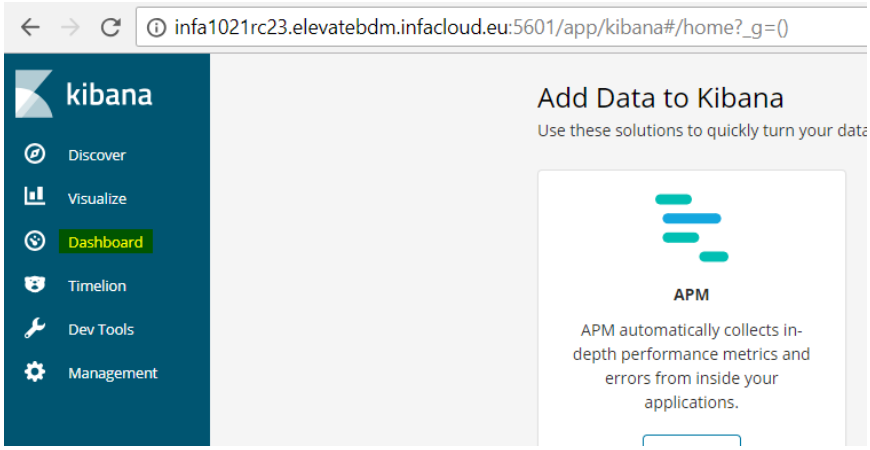
```
$ kafkaConsumer weblogsEnriched
```



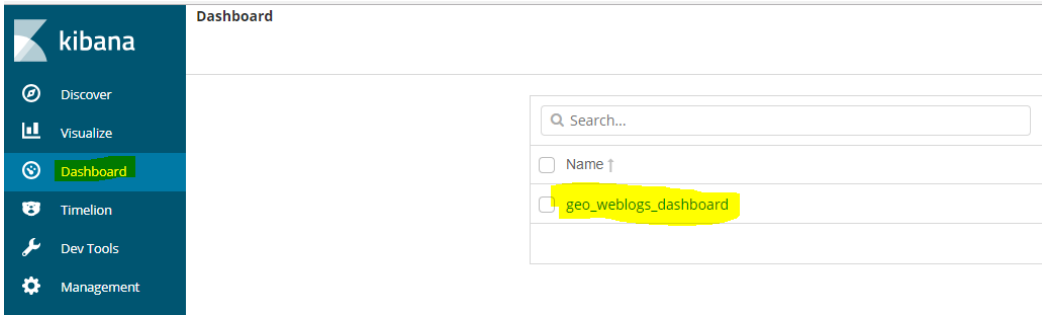
Above is a sample of two kafka messages containing the enriched weblogs with the additional fields resulting from the real-time enrichment (e.g. pageType, orderId, productid, productDescription etc.).

To visualize the **Kibana dashboard** being populated in real-time by the output of the BDS mapping, connect to the Kibana WEB UI at:
http://infa1021rc23.elevatebdm.infacloud.eu:5601

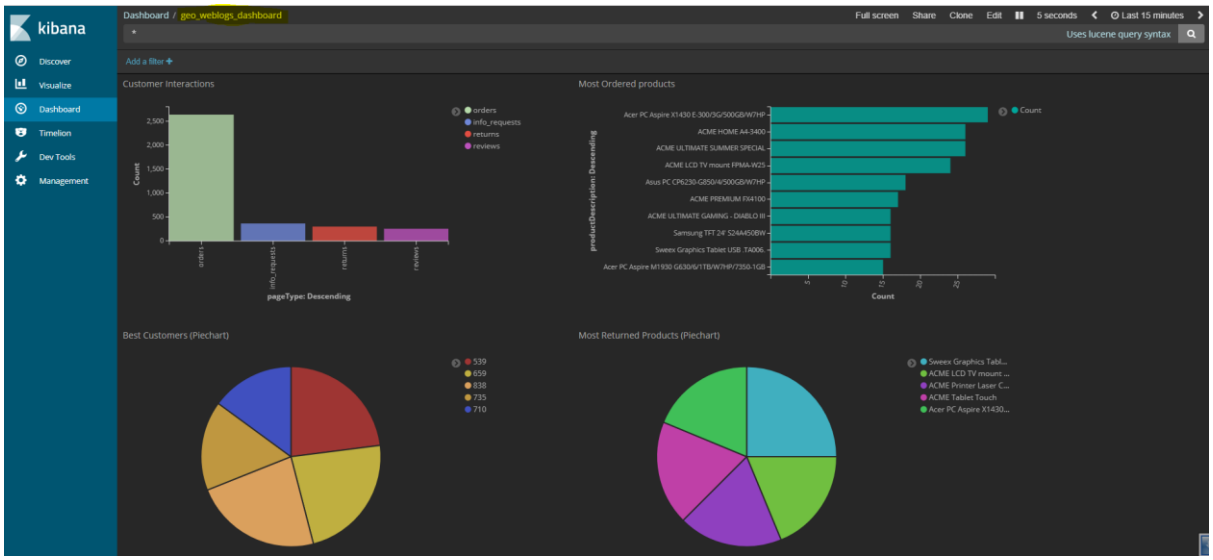
Then select Dashboard from the left side menu:



Now click the '**geo_weblogs_dashboard**' link:



You should see:

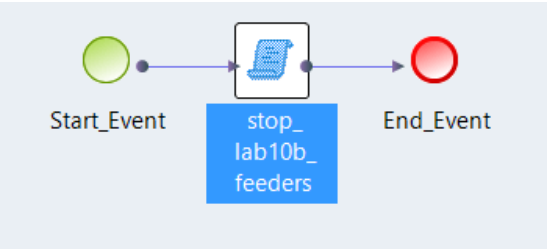


Take your time to observe and understand the dashboard's Visualizations.

Note: the top-right and bottom-right charts are showing the **product description** obtained by the real-time HBase lookup of the BDS mapping

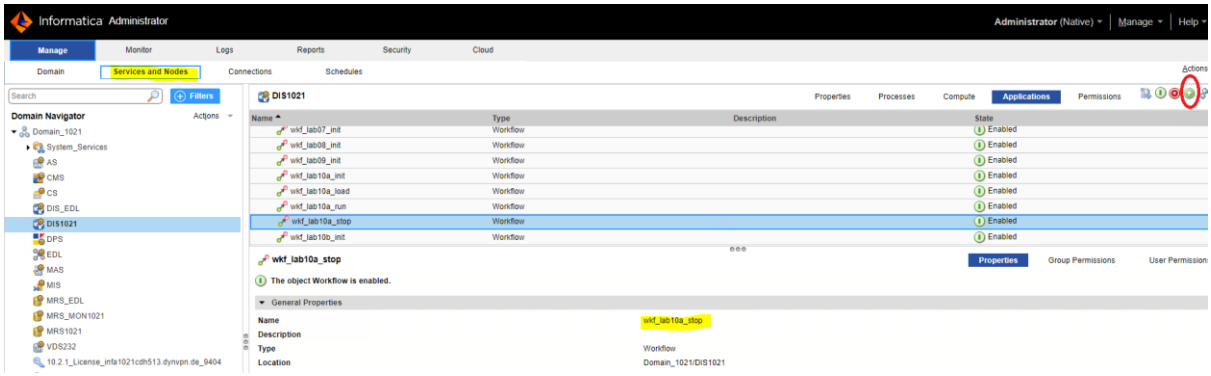
12.8 Stop the Lab

Workflow **wkf_lab10a_stop** stops the real-time data feeder to the **BDS** mapping.

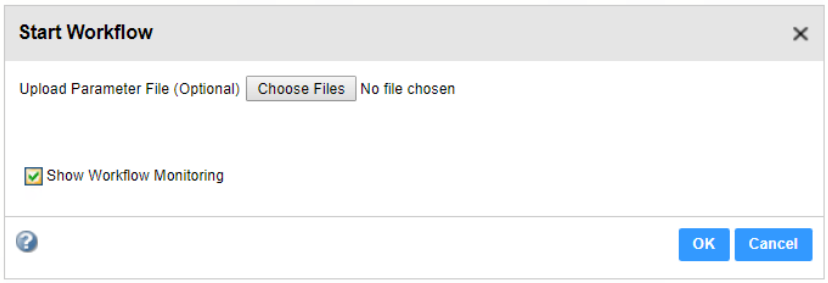


It will stop the same Python script that was started by the loading workflow.

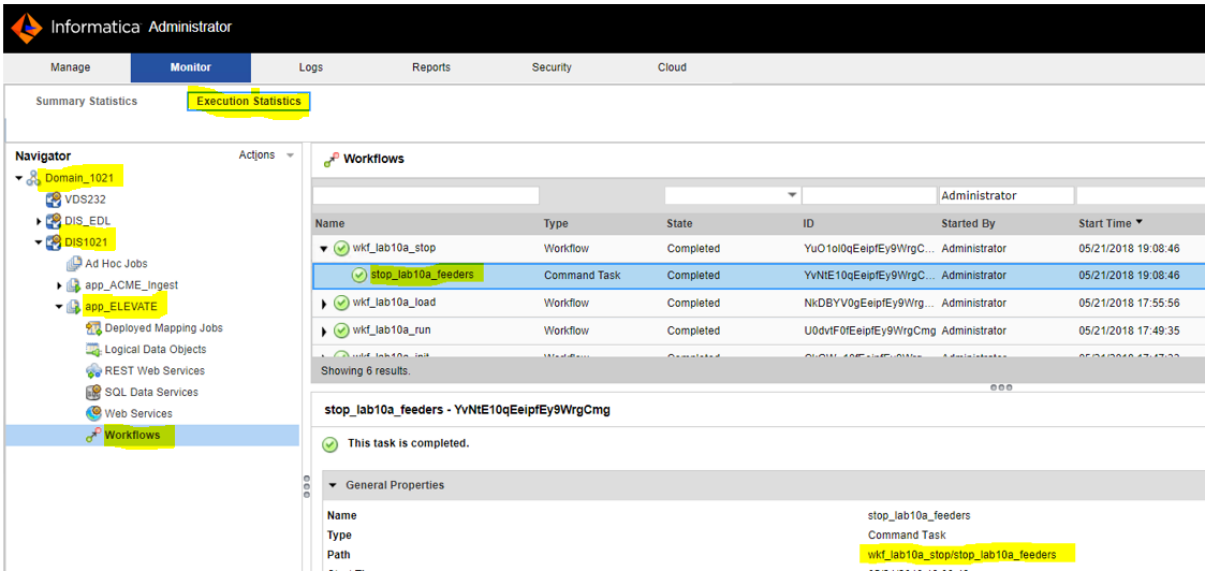
To **stop** the data feeder of **lab10a** mapping log on to the admin console and run the associated **wkf_lab10a_stop** workflow navigating under ***Manage -> Services and Nodes -> DIS1021*** and selecting the '**Applications**' tab view (expand the **app_ELEVATE** application from the list):



Click the Start Workflow icon on the top-right, check **Show Workflow Monitoring** and click **OK**:



Under ***Monitor->Execution Statistics->DIS1021->app_ELEVATE->workflows***, you should see:



This concludes BDS lab10a.

13. Lab10b – BDS weblogs real-time processing (Optional)

13.1 Purpose

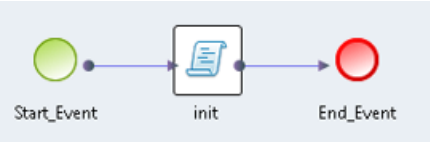
Real-time collection and processing of weblogs and CDC PoS data.

ACME now wants increase insights by also capturing and analyzing real-time Point of Sale (PoS) transactions data coming from brick-and-mortar stores, as well as clickstreams of customer's interactions with their website, and derive some additional useful real-time statistics:

- Monitor in real-time how many daily product items are being purchased at brick-and-mortar stores
- Detect, for each product, whether the amount of total daily purchased items (both online and in-store) goes above a certain threshold and raise an alert if it does.
- Ingest the alert events explained above into the Data Lake for further offline historical analytics

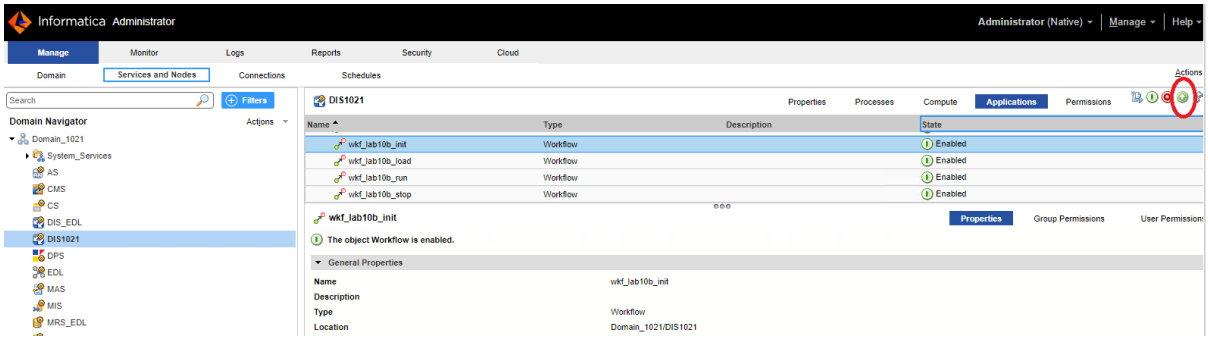
13.2 Initialize Lab

Workflow **wkf_lab10b_init** calls a script to reinitialize the lab

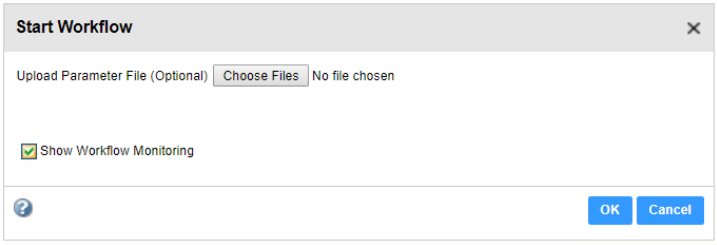


The **init** task command will reset target HBASE tables, create the required Kafka topics and clean Elastic Search indexes (for Kibana dashboarding).

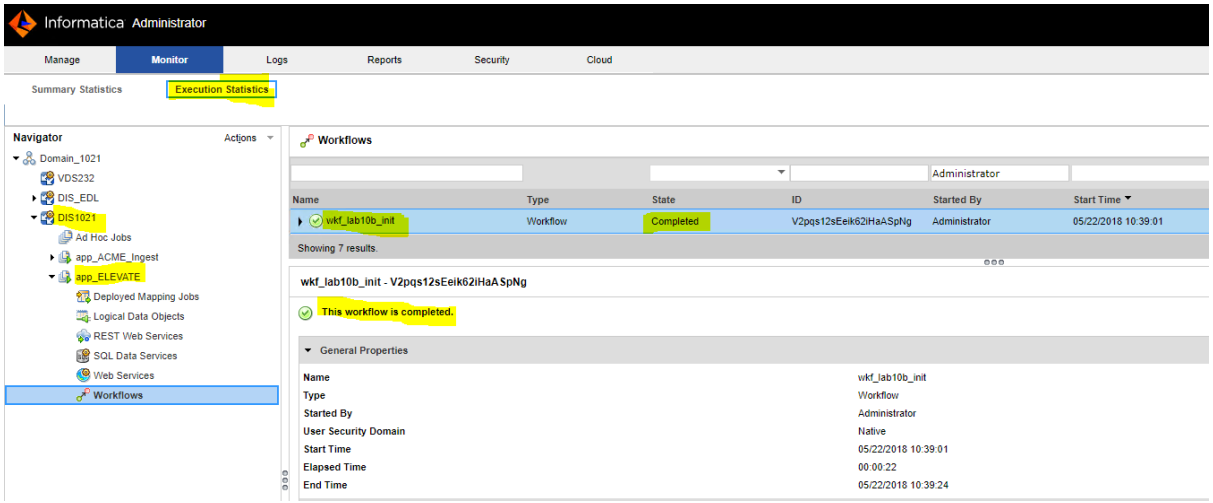
To initialize the lab, log on to the admin console and run the associated **wkf_lab10b_init** workflow navigating under ***Manage -> Services and Nodes -> DIS1021*** and selecting the '**Applications**' tab view (expand the **app_ELEVATE** application from the list):



Do not upload any parameter file; check **Show Workflow Monitoring** and click **OK**.



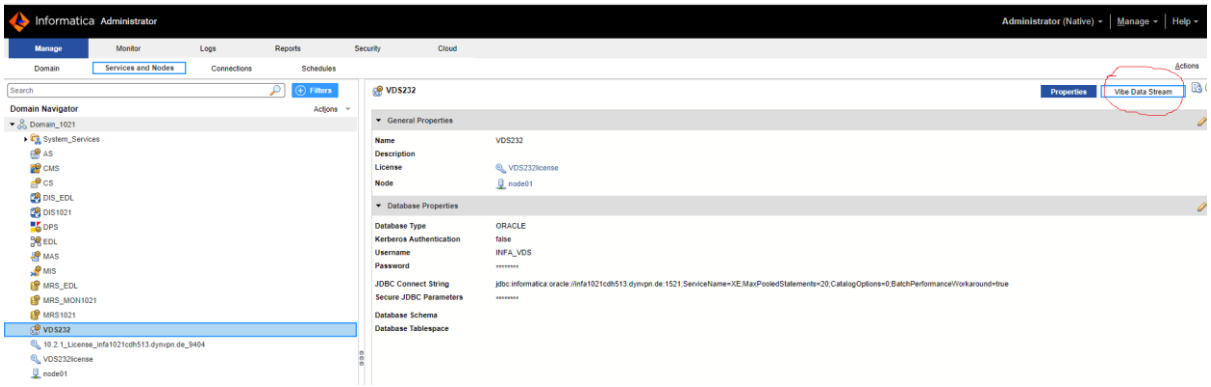
You should see:



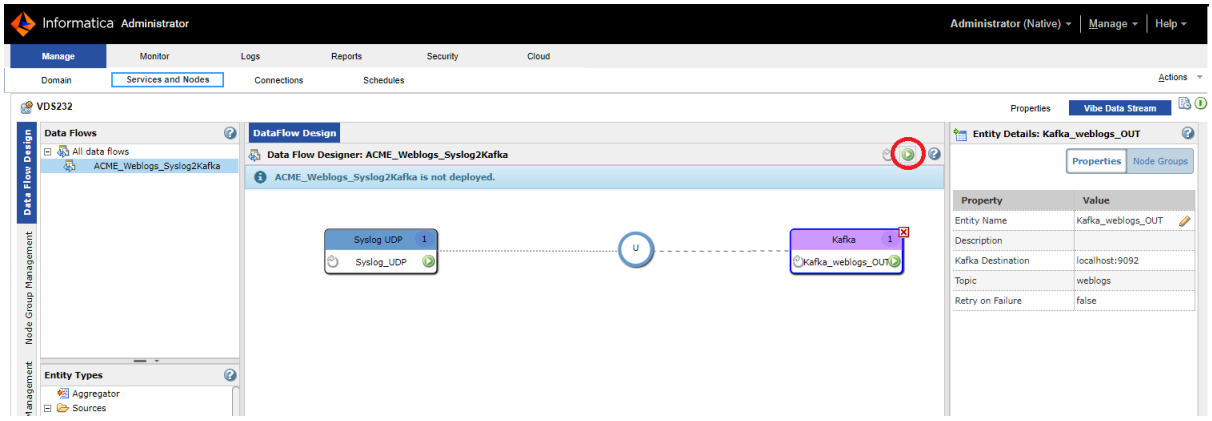
13.3 Start the Edge Data Streaming Flow

Informatica Edge Data Streaming (EDS) will be used to collect weblogs data published on Syslog protocol, format on-the-fly the unstructured weblog data message into JSON and publish it to the **weblogs** Kafka topic (for BDS input).

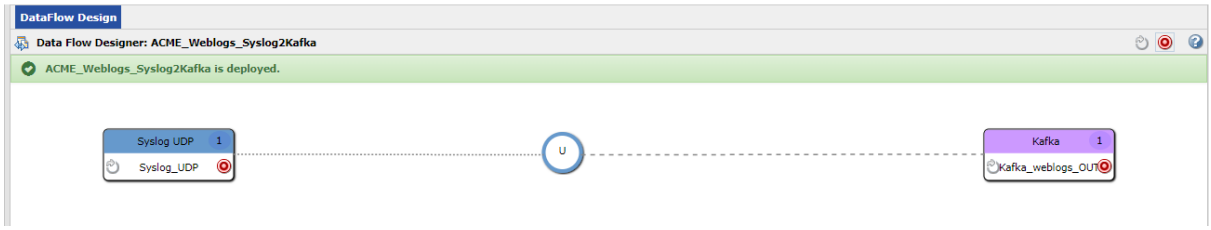
To deploy the EDS data flow, log into the Admin console and navigate to ***Manage -> Services and Nodes -> Vibe Data Stream*** (this is the former name of EDS):



You should see:

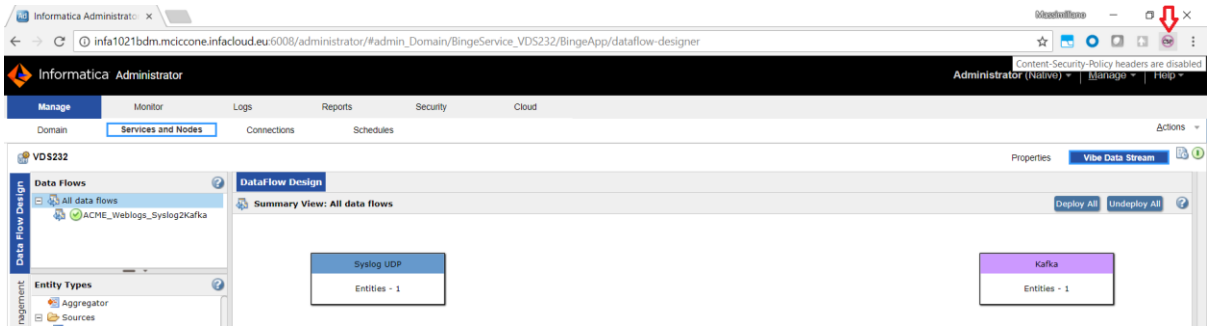


Deploy the EDS data flow (click top-right icon). You should see:

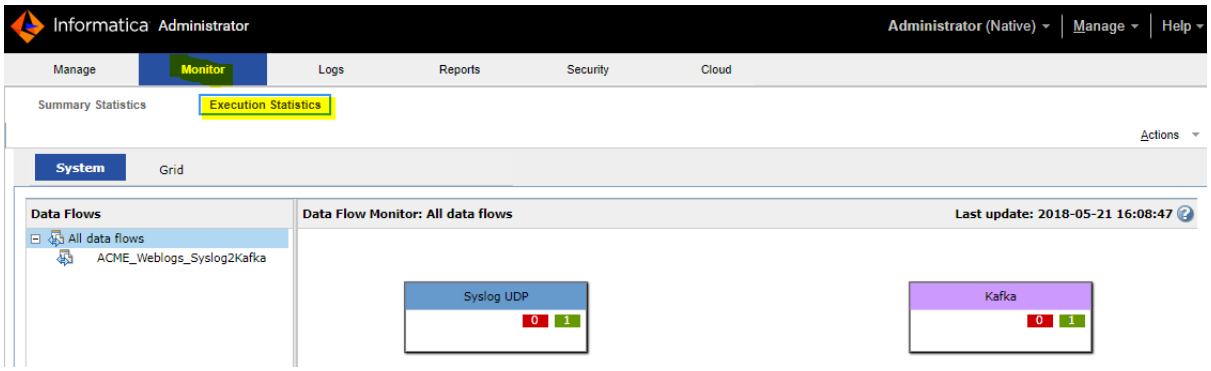


Now you can monitor the EDS data flow run-time statistics.

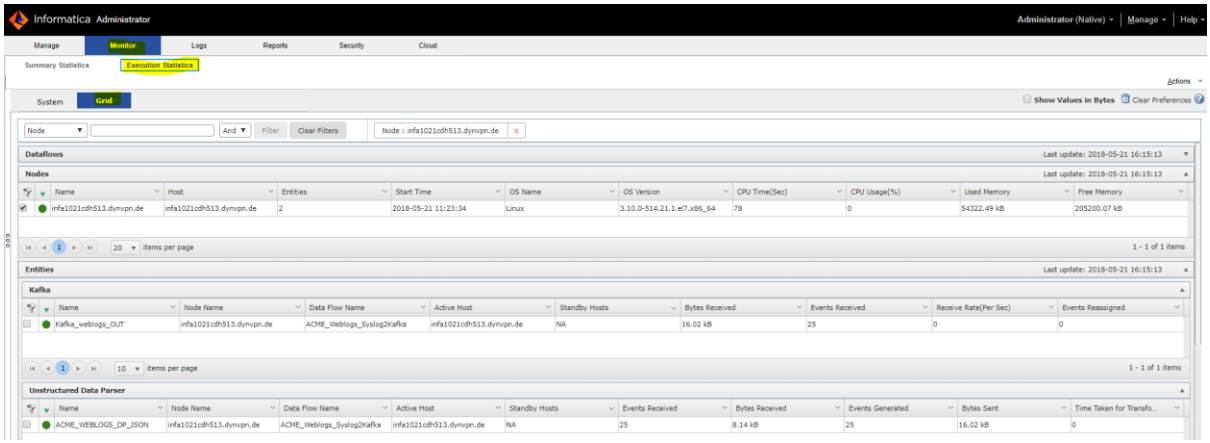
First, ensure that the **Content-Security-Policy** headers are disabled by clicking the Chrome plug-in icon on the top-right corner of the browser:



Then navigate to ***Monitor -> Execution Statistics***:

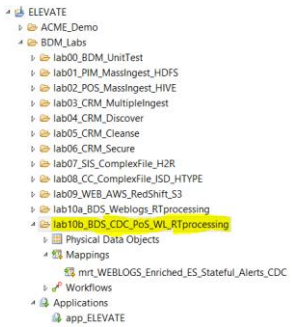


When selecting the **Grid** view, you can monitor EDS runtime statistics on collected real-time data:



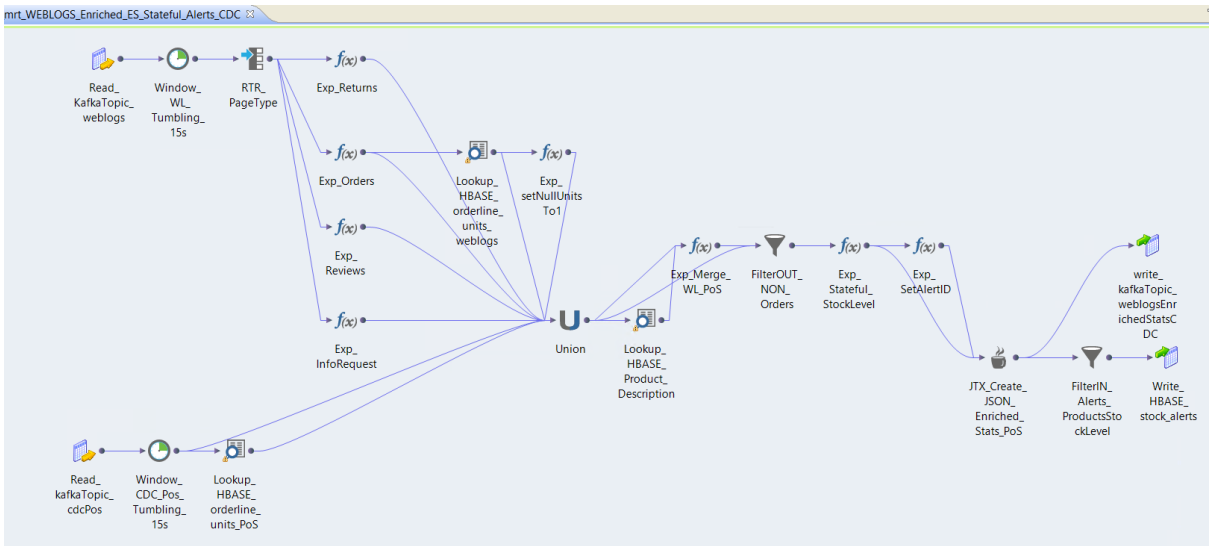
13.4 Review Lab Content

Open Developer and navigate to **lab10b_BDS_CDC_PoS_WL_RTprocessing** folder:

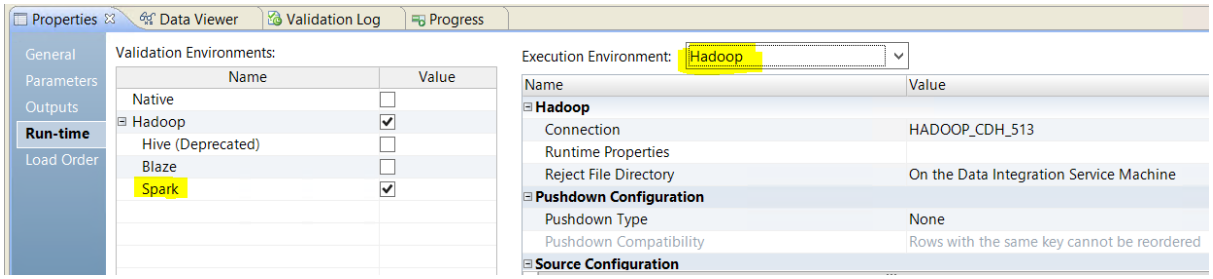


and open the **mrt_WEBLOGS_Enriched_ES_Stateful_Alerts_CDC** mapping.

You should see:

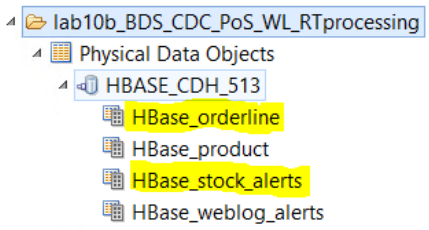


Also this mapping is set for **Hadoop-Spark** Streaming execution mode:



Which means that it will translate into Scala code for the Spark Streaming real-time processing engine.

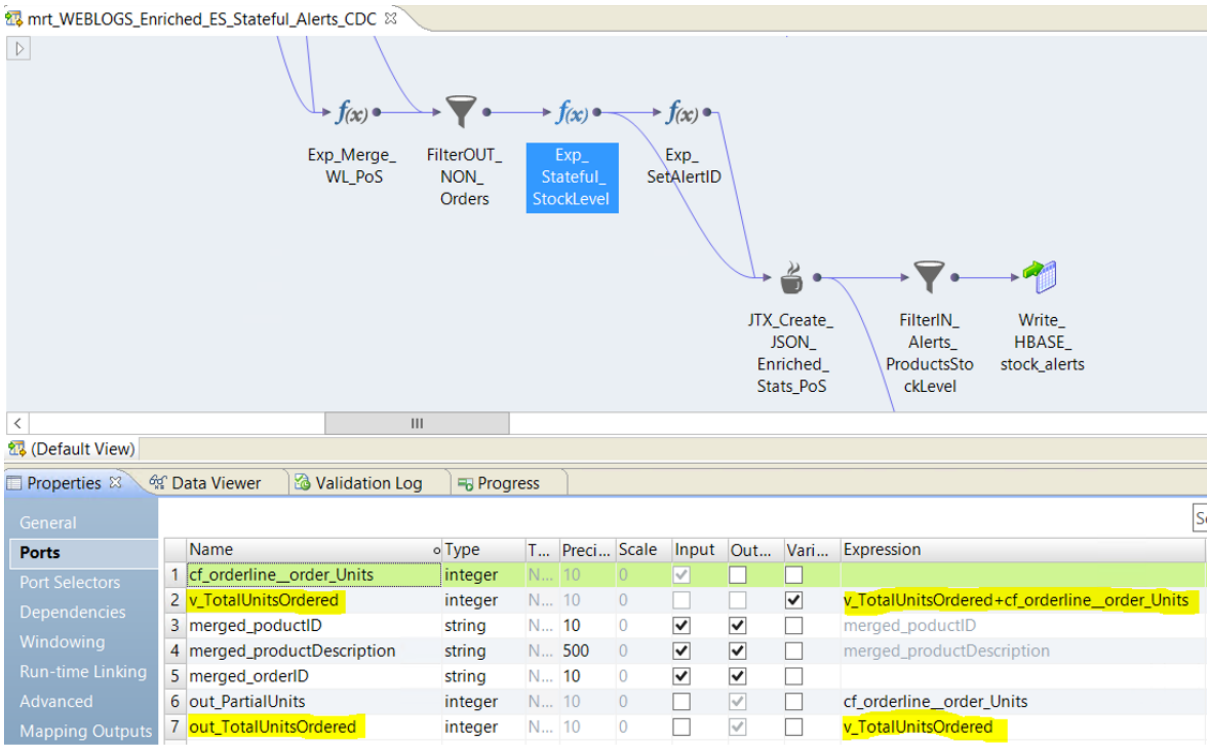
In Developer, observe the two additional HBase objects (with respect to Lab10a):



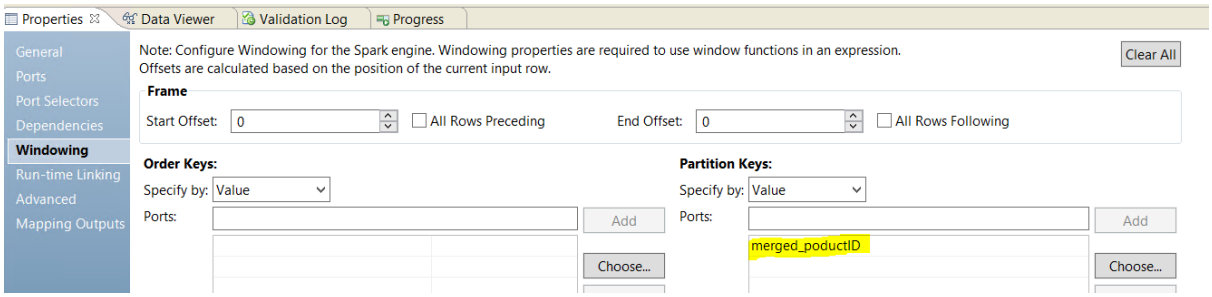
The **source** '**HBase_orderline**' is a **look-up table** used to fetch real-time additional information on PoS transactions (i.e. number of product items purchased and price).

The **target** '**HBase_stock_alerts**' Data Object describes a table used to store the product stock level alerts generated by the mapping.

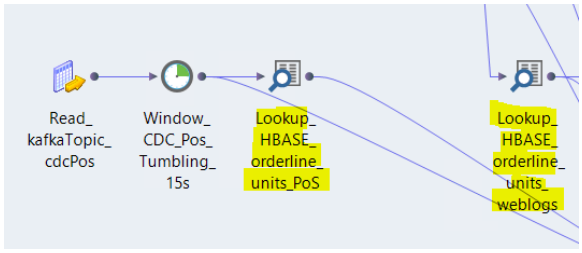
Observe the '**Exp_Stateful_StockLevel**' expression in the mapping and select the '**Ports**' view tab:



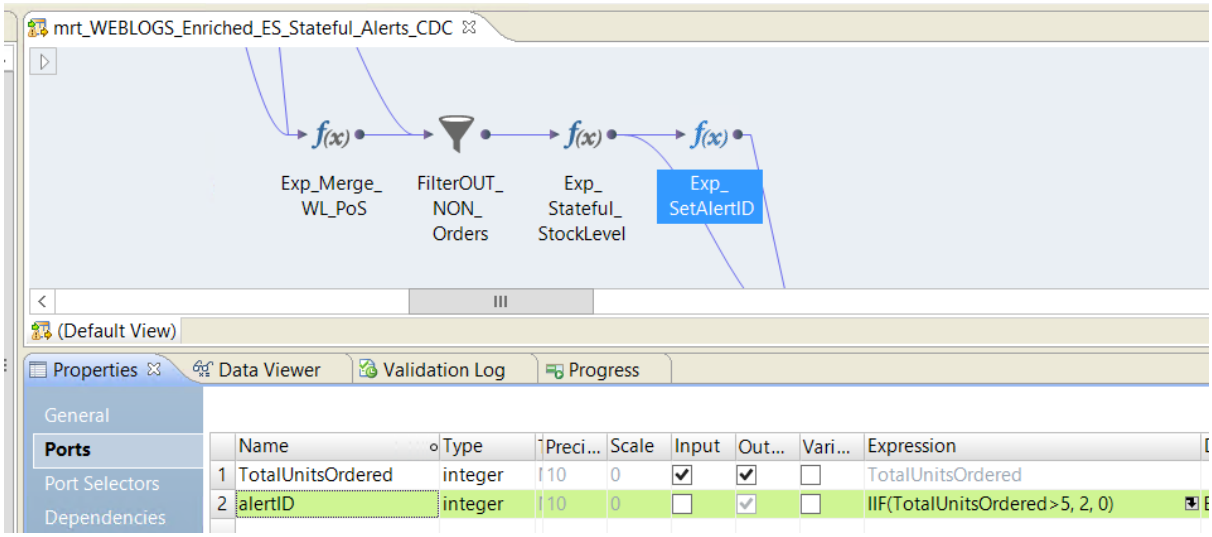
Observe also the **Windowing** of the stateful expression:



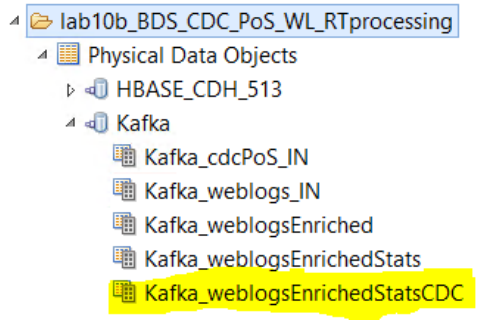
The **stateful variable** '**v_TotalUnitsOrdered**' keeps track of the daily total amount of items ordered for a specific product, for both online and in-store purchases, by adding up the number of product units ordered resulting from the real-time look-up to the '**orderline**' HBase table (using the PoS transaction id or weblog order id as a look-up key):



If the total daily number of items purchased for a product goes above 5 units, then an alert is raised (i.e. written to HBase and published to a Kafka topic):



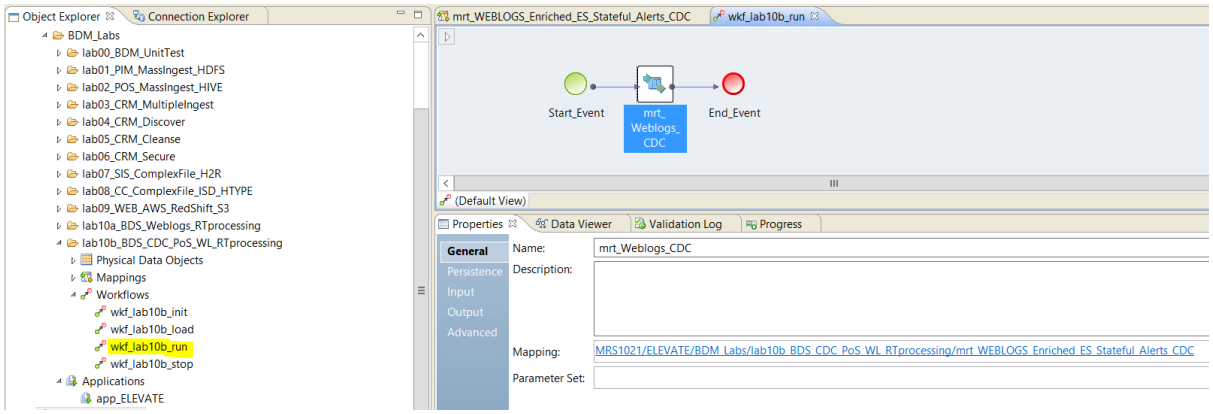
Observe the additional **Kafka** Data Object:



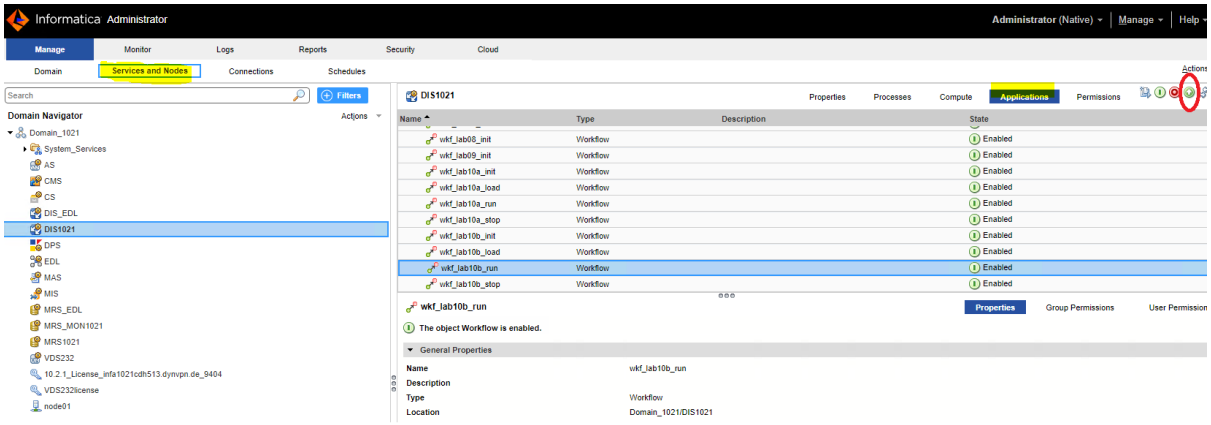
This Data Object describes the Kafka topic '**weblogsEnrichedStatsCDC**' used for product stock-level alerts output.

13.5 Run the Lab

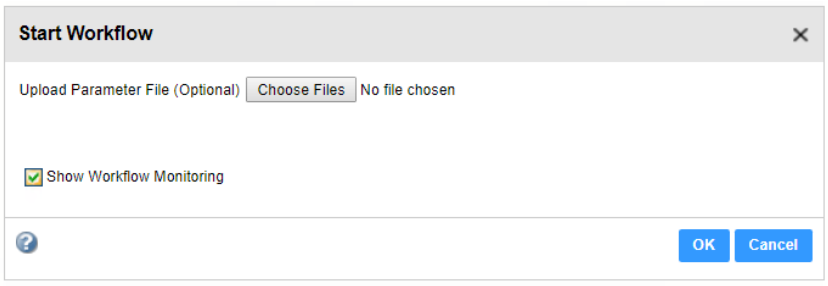
Workflow **wkf_lab10b_run** runs the **mrt_WEBLOGS_Enriched_ES_Stateful_Alerts_CDC** mapping.



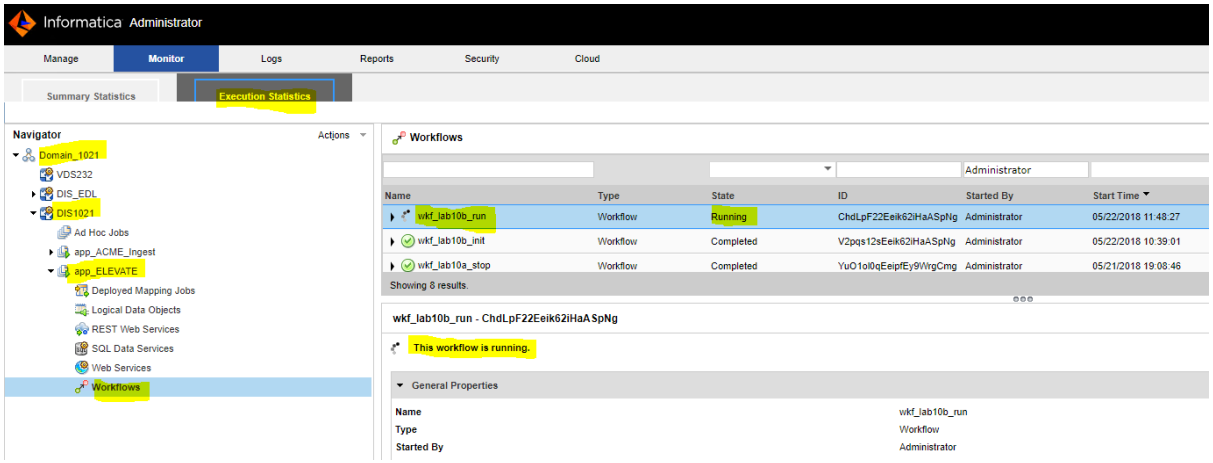
To run **lab10b** mapping log on to the admin console and run the associated **wkf_lab10b_run** workflow navigating under ***Manage -> Services and Nodes ->DIS1021*** and selecting the '**Applications**' tab view (expand the **app_ELEVATE** application from the list):



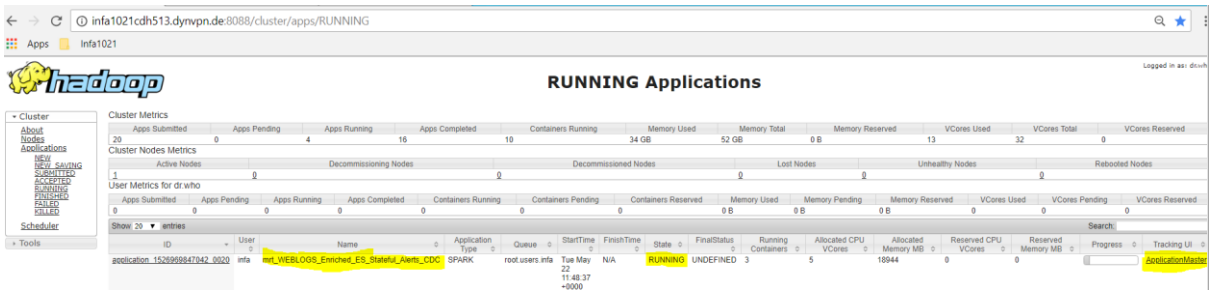
Click Start Workflow (top-right), check Show Monitoring and click OK:



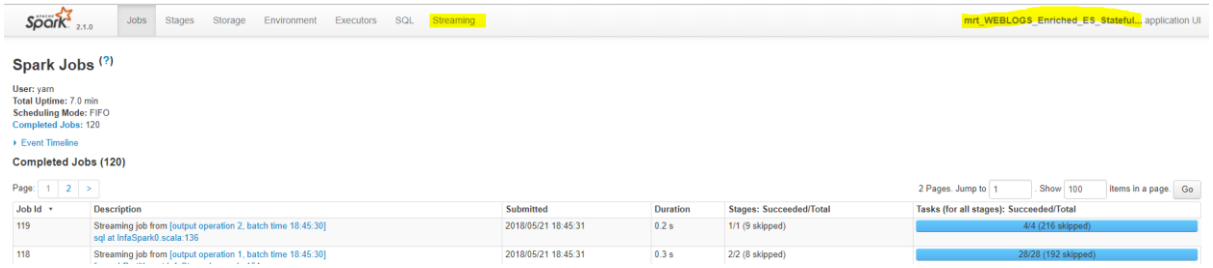
Under ***Monitor->Execution Statistics->DIS1021->app_ELEVATE->workflows***, you should see:



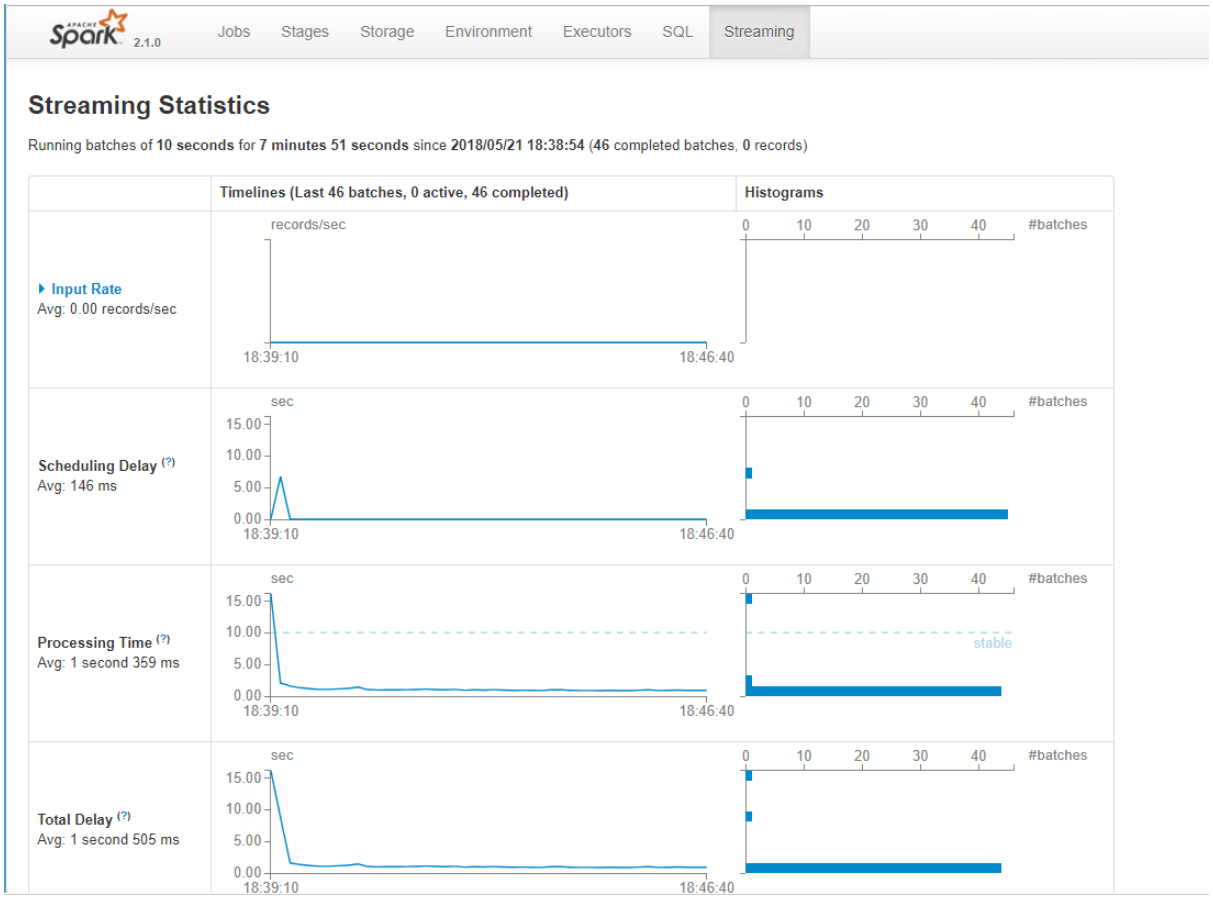
In YARN Monitor you can view the runtime statistics of the associated Spark Streaming Application execution (http://bdm.localdomain:8088/cluster/apps/RUNNING):



Click **ApplicationMaster** on the right; you should see:



Click the **Streaming** tab on the top right; you should see all the runtime statistics for the Spark Streaming job associated to the BDS mapping logic:

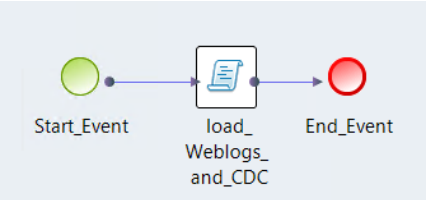


Note: It may take couple of minutes for the Streaming tab to appear.

The **Lab10b**'s BDS mapping is now ready to ingest and process real-time data.

13.6 Load Real-time Mapping Data

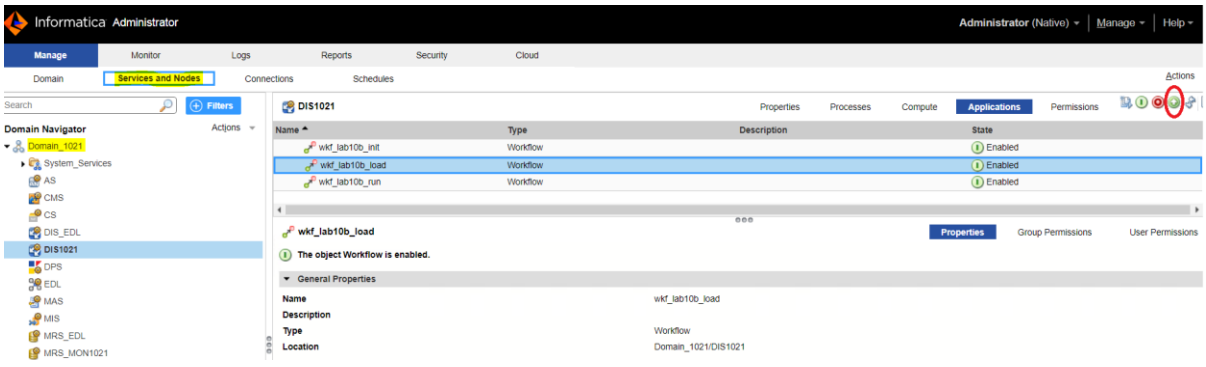
Workflow **wkf_lab10b_load** loads real-time data streams to the **mrt_WEBLOGS_Enriched_ES_Stateful_Alerts_CDC** mapping.



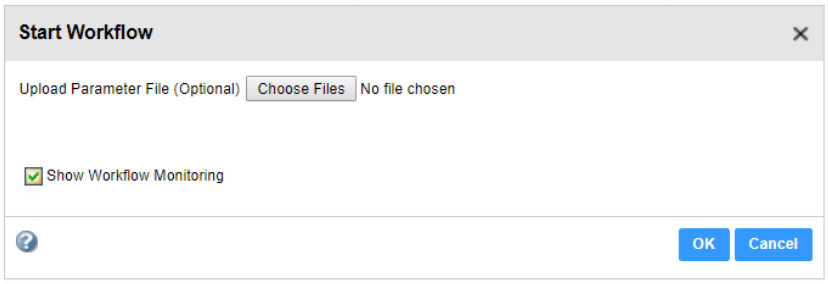
It will start 2 Python scripts that will produce the following real-time data streams:

1) Raw weblogs data over a Syslog protocol (for EDS input); the EDS data flow will then produce a Kafka output stream on the '**weblogs**' topic (for BDS input).
2) A simulation of real-time CDC data capturing changes on the ACME **orders** table, for simulated PoS transactions executed at brick-and-mortar shops; this will be sent over a **cdcPos** kafka topic stream directly to BDS (i.e. no previous EDS data collection flow).

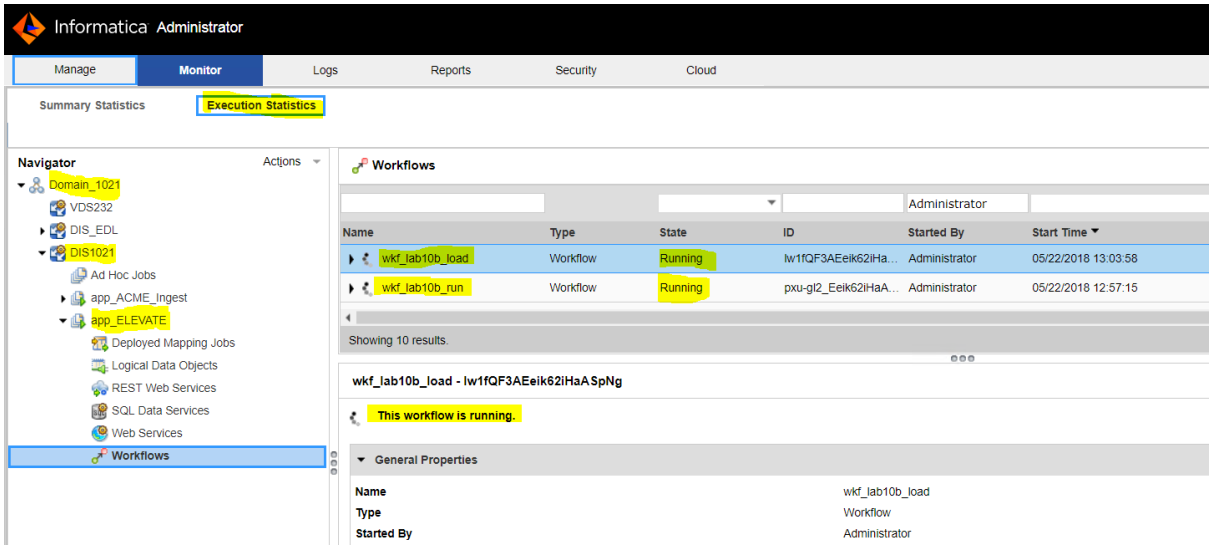
To load data to **lab10b** mapping log on to the admin console and run the associated **wkf_lab10b_load** workflow navigating under ***Manage -> Services and Nodes ->DIS1021*** and selecting the '**Applications**' tab view (expand the **app_ELEVATE** application from the list):



Click Start Workflow, check **Show Workflow Monitoring** and click **OK**:



Under ***Monitor->Execution Statistics->DIS1021->app_ELEVATE->workflows***, you should see:



Real-time data should be now flowing into Lab10b mapping from 2 different streaming channels:

1) Weblogs from Syslog into EDS and consequently into BDS.
2) CDC PoS transaction data from Kafka into BDS

13.7 Observe Outcome

The **weblogsEnrichedStatsCDC** Kafka topic contains the outcome of the stateful stats computations of the per-product total number of units ordered (for products stock level monitoring).
You can monitor the content of this topic (while the BDS mapping is running) by executing the following shell alias command on the server machine:

```
$ kafkaConsumer weblogsEnrichedStatsCDC
```


```
{"ProductID":"101788","OrderID":"3807","ProductDescription":"Logitech Keyboard Cordless K340 USB","Partial_orderUnits":1,"TotalProductUnitsOrdere
d":2,"ProductStockLevel_alertID":0}
{"ProductID":"414073","OrderID":"2038","ProductDescription":"Wacom Bamboo Pen & Touch","Partial_orderUnits":1,"TotalProductUnitsOrdered":8,"Produ
ctStockLevel_alertID":2}
{"ProductID":"465061","OrderID":"3856","ProductDescription":"Dymo LabelManager PnP","Partial_orderUnits":1,"TotalProductUnitsOrdered":1,"ProductS
tockLevel_alertID":0}
{"ProductID":"101882","OrderID":"9086","ProductDescription":"Logitech Keyboard Media K200","Partial_orderUnits":1,"TotalProductUnitsOrdered":1,"P
roductStockLevel_alertID":0}
{"ProductID":"253245","OrderID":"8192","ProductDescription":"LG 23' E2351T-BN","Partial_orderUnits":1,"TotalProductUnitsOrdered":4,"ProductStockL
evel_alertID":0}
{"ProductID":"253170","OrderID":"2038","ProductDescription":"Acer PC Aspire X1430 E-300/3G/500GB/W7HP","Partial_orderUnits":1,"TotalProductUnitsO
rdered":7,"ProductStockLevel_alertID":2}
{"ProductID":"80571","OrderID":"9086","ProductDescription":"Microsoft Windows 10 Pro 32bit SP1","Partial_orderUnits":1,"TotalProductUnitsOrdered"
:1,"ProductStockLevel_alertID":0}
```


The '**stock_alerts**' HBASE table is storing all the alerts coming from the stateful statistics computed on the combined weblogs and CDC PoS stream in the BDS mapping. It can be viewed using the following HBase shell command:

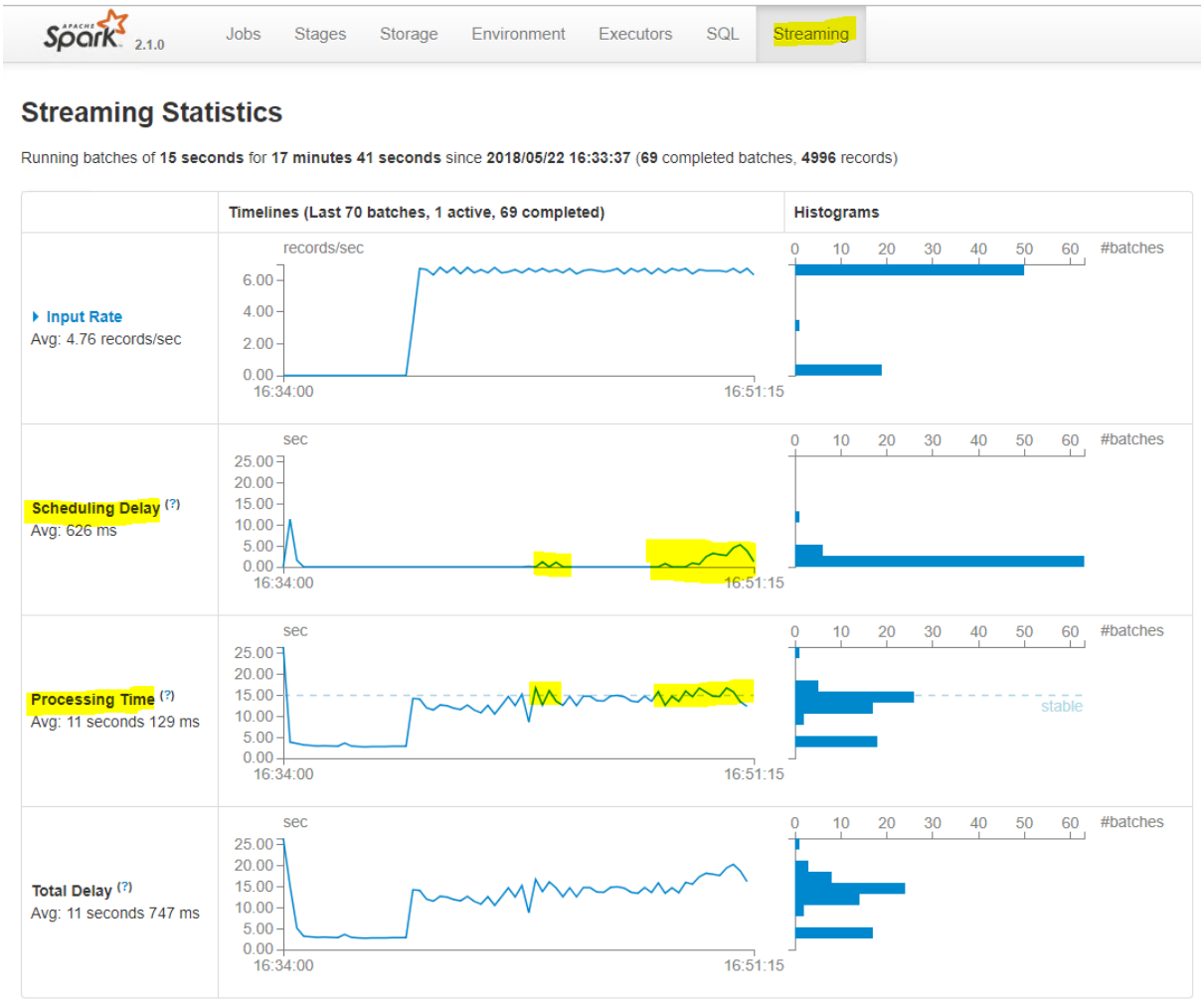
```
> scan 'stock_alerts'
```


```
hbase(main):028:0> scan 'stock_alerts'
ROW                                   COLUMN+CELL
 101881                               column=cf_stock_alerts:Alert, timestamp=1526996361746, value={"ProductID":"101881","OrderID":"559","Product
                                      Description":"Logitech Desktop Cordless MK320","Partial_orderUnits":1,"TotalProductUnitsOrdered":7,"Product
                                      StockLevel_alertID":2}
 102022                               column=cf_stock_alerts:Alert, timestamp=1526996453635, value={"ProductID":"102022","OrderID":"9940","Produc
                                      tDescription":"Razer Goliathus Speed Extended XL","Partial_orderUnits":1,"TotalProductUnitsOrdered":7,"Prod
                                      uctStockLevel_alertID":2}
 102170                               column=cf_stock_alerts:Alert, timestamp=1526996391598, value={"ProductID":"102170","OrderID":"2478","Produc
                                      tDescription":"Rapoo keyboard bluetooth for iPad Blade Black","Partial_orderUnits":1,"TotalProductUnitsOrde
                                      red":7,"ProductStockLevel_alertID":2}
 121873                               column=cf_stock_alerts:Alert, timestamp=1526996406744, value={"ProductID":"121873","OrderID":"9148","Produc
                                      tDescription":"Wacom Bamboo Fun Pen & Touch Small","Partial_orderUnits":1,"TotalProductUnitsOrdered":7,"Pro
                                      ductStockLevel_alertID":2}
 171618                               column=cf_stock_alerts:Alert, timestamp=1526996466435, value={"ProductID":"171618","OrderID":"9376","Produc
                                      tDescription":"ACME LCD Arm FPMA-D600 Black Clamp Type","Partial_orderUnits":1,"TotalProductUnitsOrdered":9
                                      ,"ProductStockLevel_alertID":2}
 171819                               column=cf_stock_alerts:Alert, timestamp=1526996346694, value={"ProductID":"171819","OrderID":"7801","Produc
                                      tDescription":"ACME LCD Arm FPMA-D965 Black","Partial_orderUnits":1,"TotalProductUnitsOrdered":6,"ProductSt
                                      ockLevel_alertID":2}
```


The '**stock_alerts**' table contains the latest alert (if generated) for a product ID (the product Id is used as an HBASE Row ID). An Alert is generated when the total of daily ordered units (both online and in-store) for a product is above 5.

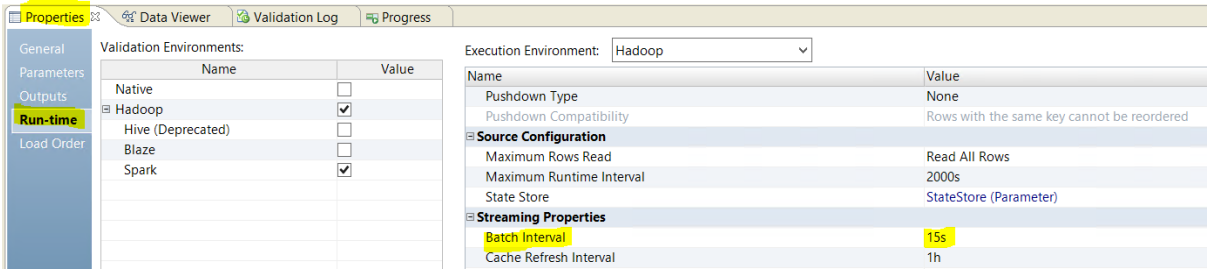
Note: for each product id, only the latest stock-level alert will be visible, this is because the product id is used as a row key.

You can refresh the runtime statistics of the Spark Streaming job associated to the BDS mapping by clicking the **Streaming** tab in the YARN application WEB UI:

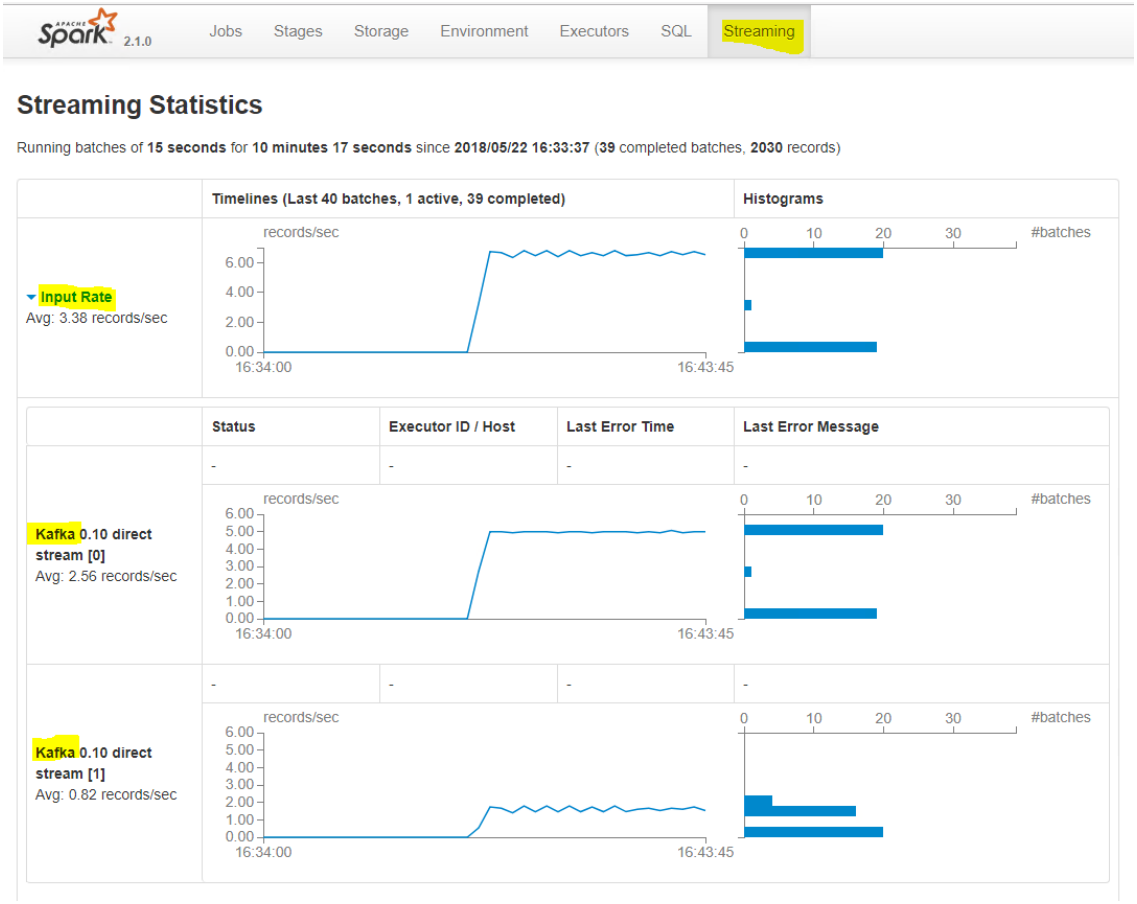


Note: The processing time for this Lab's mapping is slightly higher than for Lab10a (due to additional targets and more complex real-time computations). For a short period, it also went above 15 seconds, which is the same as the mapping 'Batch Interval' (RDD). This is ok if it is just a transitory event but, if constant, can lead to the scheduling delay steadily increasing and to the mapping eventually failing.

You can view the BDS mapping's Batch Interval under the Run-time properties (click on the mapping canvas first):

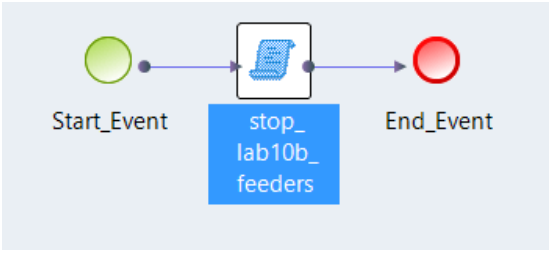


You can also expand the input rate chart and view the two Kafka streams singularly:



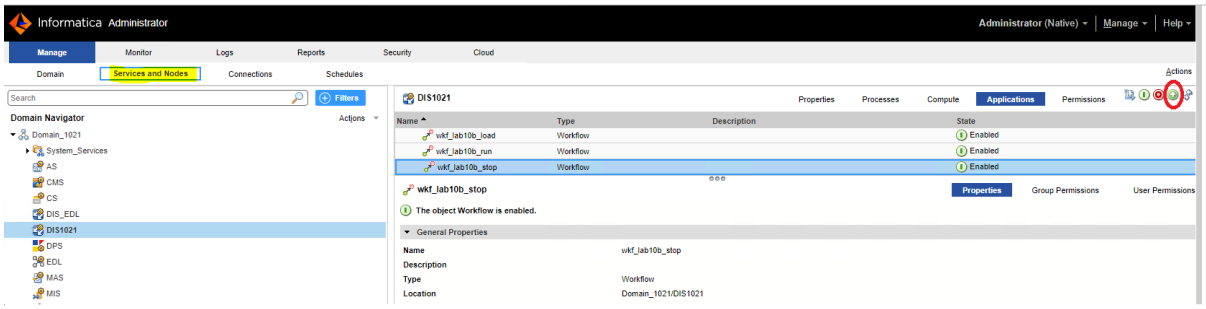
13.8 Stop the Lab

Workflow **wkf_lab10b_stop** stops the real-time data feeder to the **BDS** mapping.

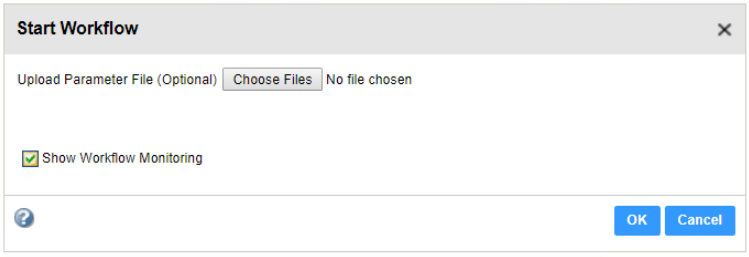


It will stop the same Python scripts that were started by the loading workflow.

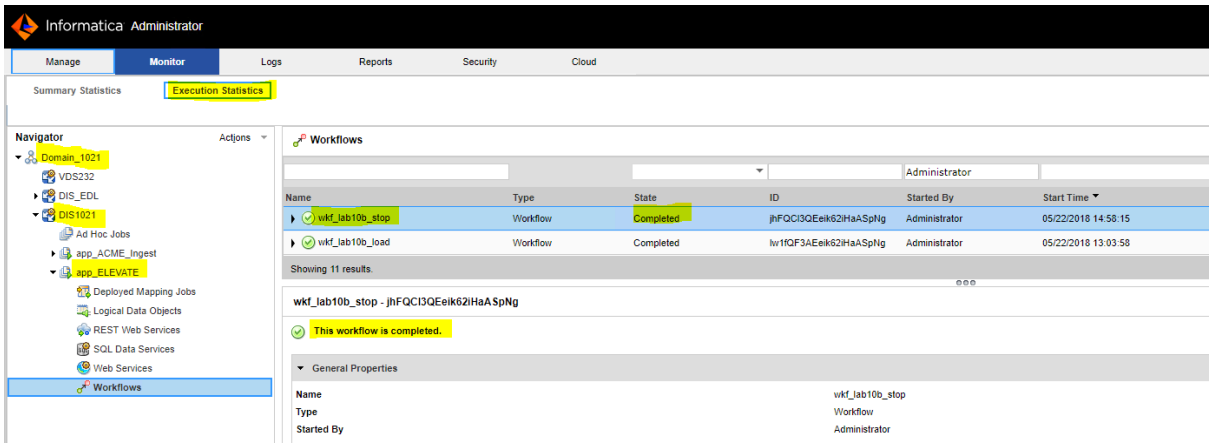
To **stop** the data feeders of **lab10b** mapping log on to the admin console and run the associated **wkf_lab10b_stop** workflow navigating under ***Manage -> Services and Nodes -> DIS1021*** and selecting the **Applications** tab view (expand the **app_ELEVATE** application from the list):



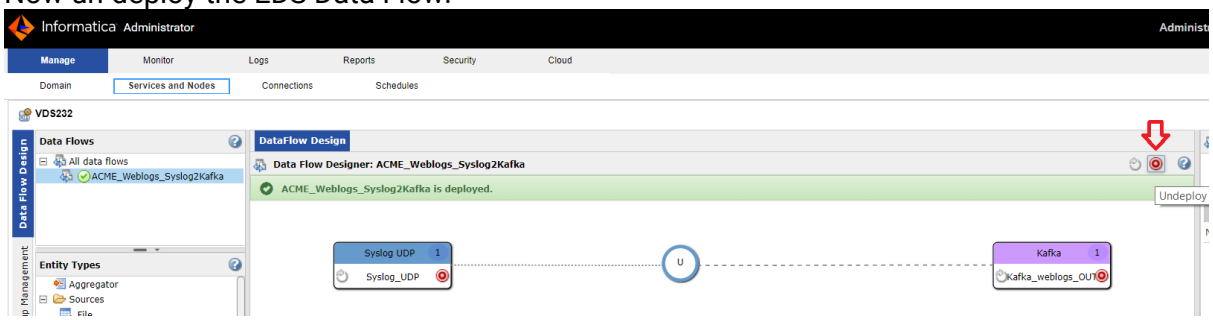
Click Start Workflow, check **Show Workflow Monitoring** and click **OK**:



Under ***Monitor->Execution Statistics->DIS1021->app_ELEVATE->workflows***, you should see:



Now un-deploy the EDS Data Flow:



This concludes BDS lab10b.