



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.....	5
1.1	PURPOSE	5
1.2	PREREQUISITES	5
1.3	BEFORE YOU START	5
1.3.1	Informatica Administration Console	5
1.3.2	Informatica Monitoring Console.....	6
1.3.3	HDFS Web Browser.....	7
1.3.4	YARN Monitor Console.....	8
1.3.5	Blaze Monitor Console	8
1.3.6	Informatica Mass Ingestion Service.....	9
2.	LAB00 - UNIT TEST BDM ABSTRACTION LAYER.....	10
2.1	PURPOSE	10
2.2	INITIALIZE LAB	10
2.3	REVIEW LAB CONTENT.....	11
2.4	RUN LAB.....	12
3.	LAB01 - PIM MASS INGESTION SERVICE.....	13
3.1	PURPOSE	13
3.2	INITIALIZE LAB	13
3.3	CREATE MIS PROJECT.....	13
3.4	DEPLOY & RUN MIS PROJECT	16
3.5	EXTRA CONTENT (OPTIONAL)	17
3.5.1	Purpose	17
3.5.2	Initialize Lab	17
3.5.3	Review Lab Content.....	18
3.5.4	Run Lab.....	20
4.	LAB02 - POS MASS INGEST WITH DYNAMIC MAPPINGS.....	21
4.1	PURPOSE	21
4.2	INITIALIZE LAB	21
4.3	REVIEW LAB CONTENT.....	21
4.4	RUN LAB.....	24
4.5	EXTRA STEPS (OPTIONAL)	25
4.5.1	Purpose	25
4.5.2	Create MIS project	25
4.5.3	Deploy & Run MIS project.....	26

5.	LAB03 – CRM MULTIPLE INGEST (DEMO ONLY)	28
5.1	PURPOSE	28
5.2	INITIALIZE LAB	28
5.3	REVIEW LAB CONTENT.....	28
5.4	RUN LAB.....	29
6.	LAB04 - CRM DISCOVER DATA.....	30
6.1	PURPOSE	30
6.2	INITIALIZE LAB	30
6.3	REVIEW LAB CONTENT.....	30
6.4	RUN LAB.....	31
6.4.1	Create & Run Profile in native mode – (Optional).....	31
6.4.2	Create & Run Profile in blaze mode	32
6.5	REVIEW PROFILING RESULTS FOR GENDER AND TIER COLUMNS	33
7.	LAB05 - CRM CLEANSE & VALIDATE	34
7.1	PURPOSE	34
7.2	INITIALIZE LAB	34
7.3	REVIEW LAB CONTENT.....	34
7.4	RUN LAB.....	35
8.	LAB06 - CRM SECURE	36
8.1	PURPOSE	36
8.2	INITIALIZE LAB	36
8.3	REVIEW LAB CONTENT.....	36
8.4	RUN LAB.....	37
9.	LAB07 - SIS COMPLEX FILE DP & H2R	38
9.1	PURPOSE	38
9.2	INITIALIZE LAB	38
9.3	REVIEW LAB CONTENT.....	38
9.4	RUN LAB.....	41
10.	LAB08 - CALL CENTER UNSTRUCTURED FILE ISD.....	42
10.1	PURPOSE	42
10.2	INITIALIZE LAB	42
10.3	BUILD & EXPORT AN ISD MODEL FROM CLOUD	42
10.4	IMPORT & RUN ISD MODEL INTO DEVELOPER	50
11.	LAB09 – WEBLOGS INTEGRATE AWS REDSHIFT & S3 (OPTIONAL)	54
11.1	PURPOSE	54
11.2	REVIEW LAB CONTENT.....	54
11.3	RUN LAB.....	58
12.	LAB10A – BDS WEBLOGS REAL-TIME PROCESSING	59
12.1	PURPOSE	59

12.2	INITIALIZE LAB	59
12.3	START THE EDGE DATA STREAMING FLOW	60
12.4	REVIEW LAB CONTENT.....	62
12.5	RUN THE LAB	64
12.6	LOAD REAL-TIME MAPPING DATA.....	66
12.7	OBSERVE OUTCOME	70
12.8	STOP THE LAB.....	72
13.	LAB10B – BDS WEBLOGS REAL-TIME PROCESSING (OPTIONAL)	74
13.1	PURPOSE	74
13.2	INITIALIZE LAB	74
13.3	START THE EDGE DATA STREAMING FLOW	75
13.4	REVIEW LAB CONTENT.....	77
13.5	RUN THE LAB	80
13.6	LOAD REAL-TIME MAPPING DATA.....	82
13.7	OBSERVE OUTCOME	84
13.8	STOP THE LAB.....	86

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

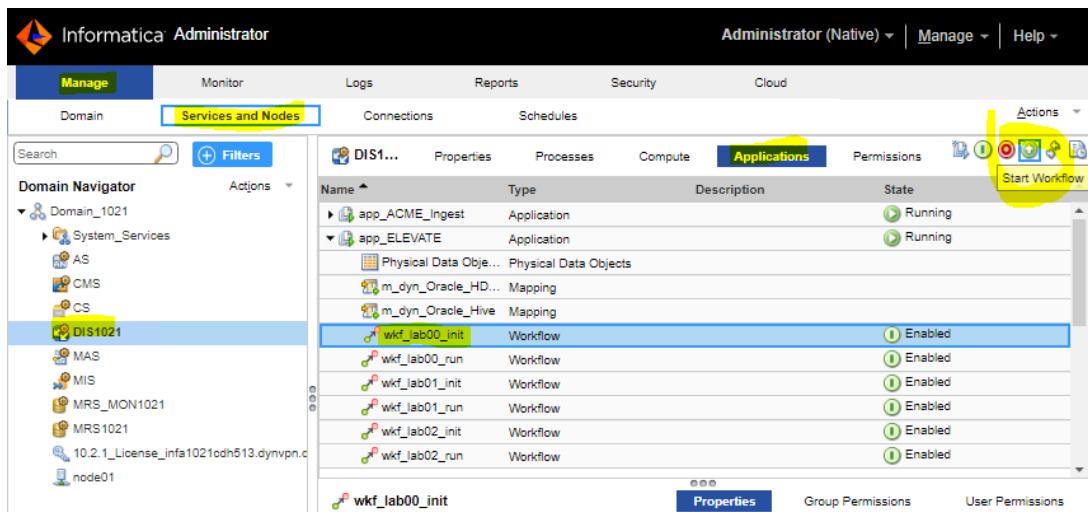
The labs will use the 10.2.1 Developer but also following interfaces you should open upfront:

1.3.1 Informatica Administration Console

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

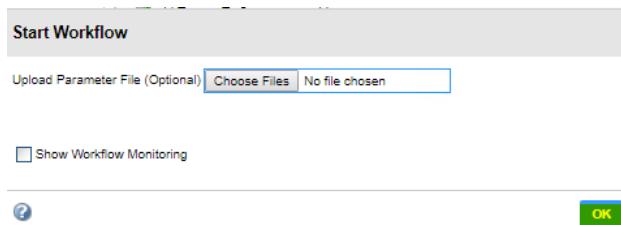
You will be asked LATER in the labs to start a workflow, to do so:

- Go to <http://bdm.localdomain:6008/administrator> and log in as Administrator/infa
- Navigate to Manage > Services and Nodes > DIS 1021 > Applications



The screenshot shows the Informatica Administrator interface. The top navigation bar includes 'Administrator (Native) | Manage | Help'. The main menu has tabs for 'Manage', 'Monitor', 'Logs', 'Reports', 'Security', and 'Cloud'. Under 'Manage', there are sub-tabs for 'Domain', 'Services and Nodes', 'Connections', 'Schedules', 'Applications', 'Permissions', and 'Actions'. The 'Actions' tab is highlighted. On the left, the 'Domain Navigator' shows a tree structure with 'Domain_1021' expanded, showing 'System_Services', 'AS', 'CMS', 'CS', and 'DIS1021'. 'DIS1021' is selected and highlighted in blue. The central pane displays a table of objects under the 'Applications' tab. The table columns are 'Name', 'Type', 'Description', and 'State'. The 'wkf_lab00_init' workflow is selected and highlighted in blue. The 'Actions' column for this row contains a yellow circle highlighting the 'Start Workflow' icon.

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

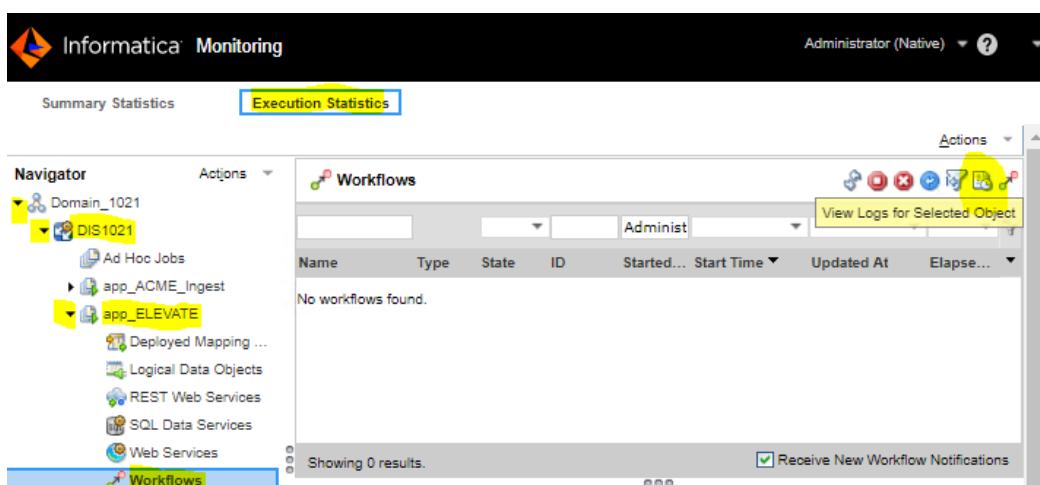


1.3.2 Informatica Monitoring Console

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

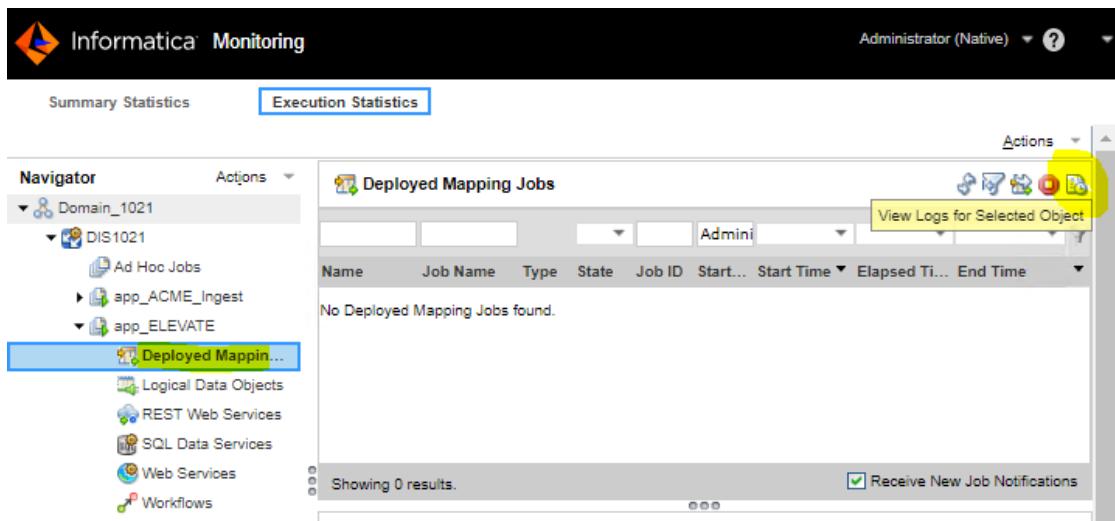


This is a screenshot of the Informatica Monitoring interface. The top navigation bar includes 'Administrator (Native) | ?'. The main menu has tabs for 'Summary Statistics' and 'Execution Statistics'. The 'Execution Statistics' tab is highlighted. On the left, the 'Navigator' pane shows a tree structure with 'Domain_1021' expanded, showing 'DIS1021' and its sub-components: 'Ad Hoc Jobs', 'app_ACME_Ingest', and 'app_ELEVATE'. 'app_ELEVATE' is selected and highlighted in blue. The central pane displays a table titled 'Workflows'. The table columns are 'Name', 'Type', 'State', 'ID', 'Started...', 'Start Time', 'Updated At', and 'Elapsed...'. The table body contains the message 'No workflows found.' At the bottom right of the table area is a yellow circle highlighting the 'View Logs for Selected Object' button.

You will be able from there to view workflow status

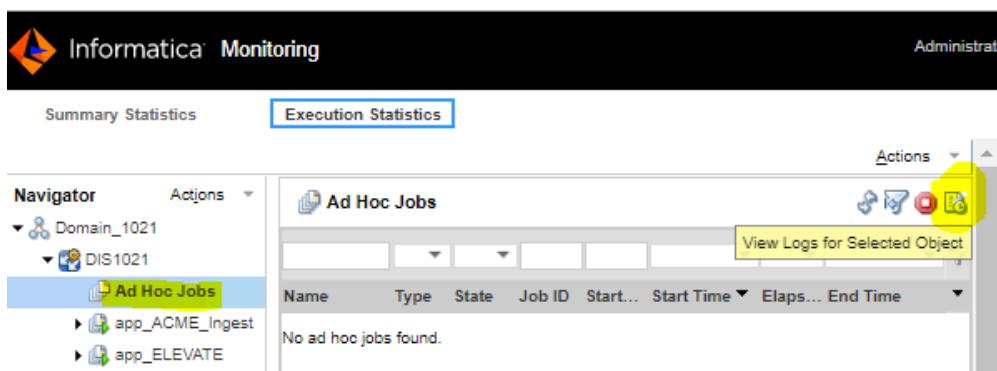
Note: you can get logs from there using the icon as shown above

- 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>

/user/infa/elevate	<input type="button" value="Go!"/>
--------------------	------------------------------------

Permission	Owner	Group	Size	Replication	Block Size	Name
drwxr-xr-x	infa	supergroup	0 B	0	0 B	lab00
drwxr-xr-x	infa	supergroup	0 B	0	0 B	lab01
drwxr-xr-x	infa	supergroup	0 B	0	0 B	lab03
drwxr-xr-x	infa	supergroup	0 B	0	0 B	lab07
drwxr-xr-x	infa	supergroup	0 B	0	0 B	lab08

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>

Browse Directory

/user/hive/warehouse/elevate.db	<input type="button" value="Go!"/>						
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
Hadoop, 2017.							

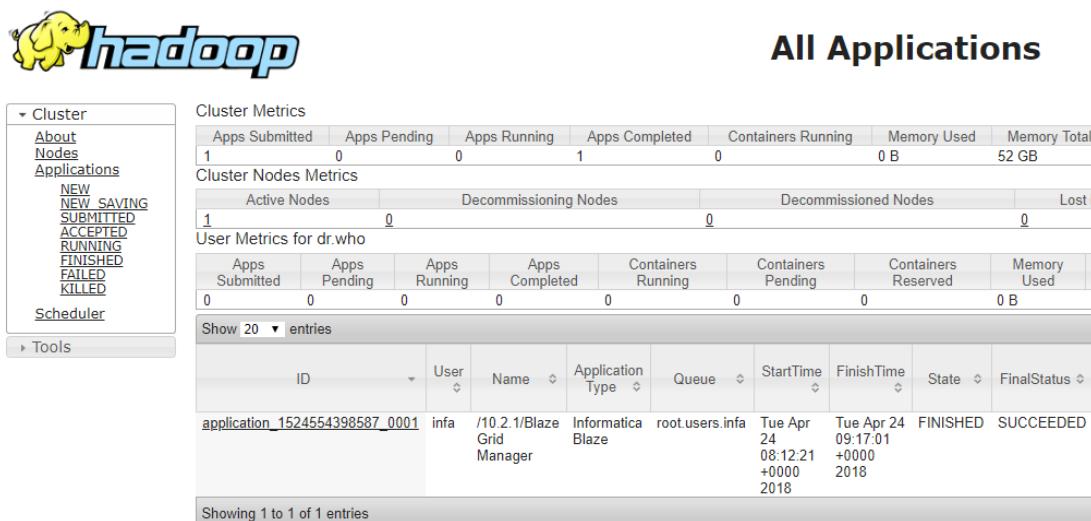
You will see there the hive table created.

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>



The screenshot shows the Hadoop YARN Monitor Console interface. On the left, there's a sidebar with navigation links like Cluster, About, Nodes, Applications, Scheduler, and Tools. The main area is titled "All Applications". It displays various metrics and a table of running applications.

Cluster Metrics:

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total
1	0	0	1	0	0 B	52 GB

Cluster Nodes Metrics:

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes
1	0	0	0

User Metrics for dr.who:

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used
0	0	0	0	0	0	0 B	0

Applications Table:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus
application_1524554398587_0001	infa	/10.2.1/Blaze Grid Manager	Informatica Blaze	root.users.infa	Tue Apr 24 08:12:21 +0000 2018	Tue Apr 24 09:17:01 +0000 2018	FINISHED	SUCCEEDED

Showing 1 to 1 of 1 entries

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

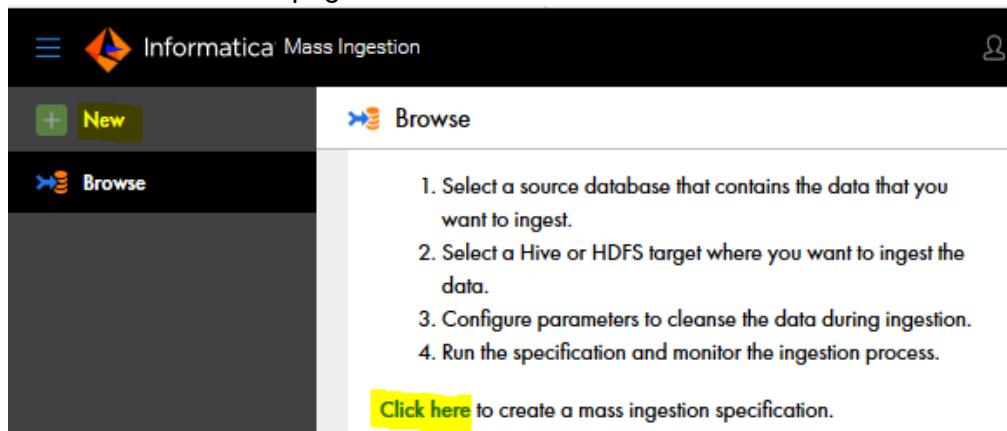
The **Informatica Mass Ingestion Service** will be used to **mass ingest data**

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



1. Select a source database that contains the data that you want to ingest.
2. Select a Hive or HDFS target where you want to ingest the data.
3. Configure parameters to cleanse the data during ingestion.
4. Run the specification and monitor the ingestion process.

Click here to create a mass ingestion specification.

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

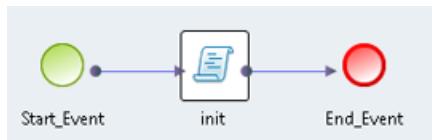
2. Lab00 - Unit test BDM abstraction layer

2.1 Purpose

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

2.2 Initialize Lab

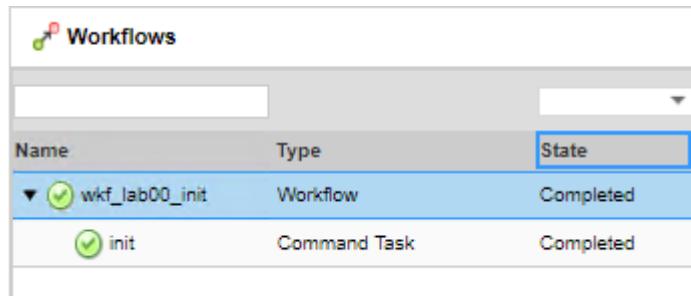
Workflow **wkf_lab00_init** calls a script to reinitialize the lab



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

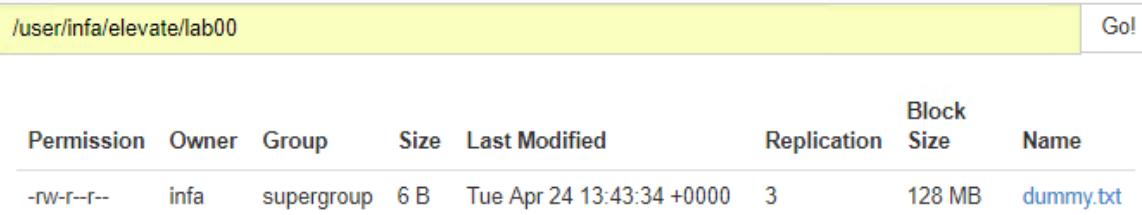
To initialize the lab, go to [Informatica Administration Console](#) and run the workflow

You should see in [Informatica Monitoring Console](#):



Name	Type	State
wkf_lab00_init	Workflow	Completed
init	Command Task	Completed

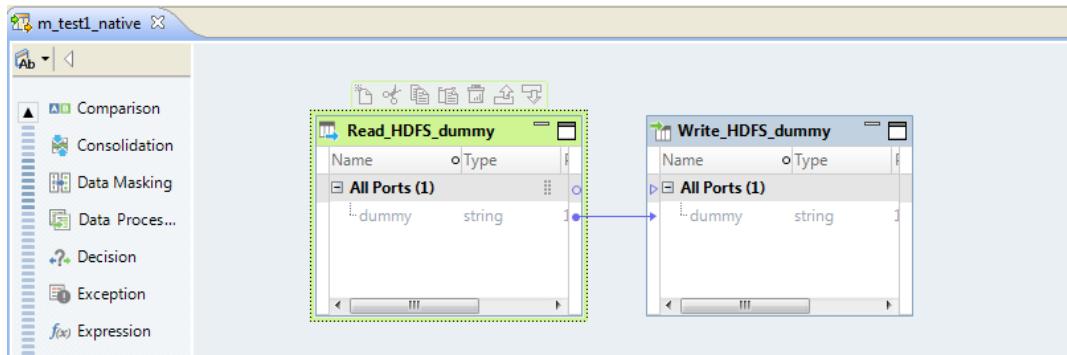
Use [HDFS Web Browser](#) and navigate to /user/infa/elevate/lab00, you should see:



Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	infa	supergroup	6 B	Tue Apr 24 13:43:34 +0000 2018	3	128 MB	dummy.txt

2.3 Review Lab Content

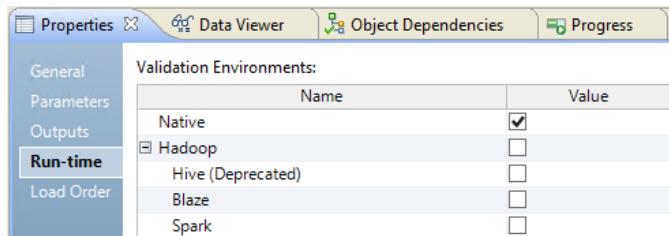
Open Developer, go to ELEVATE/BDM_Labs/lab00_* folder and open the m_test1_native mapping. You should see:



You can run preview on the source:

dummy	
1	dummy

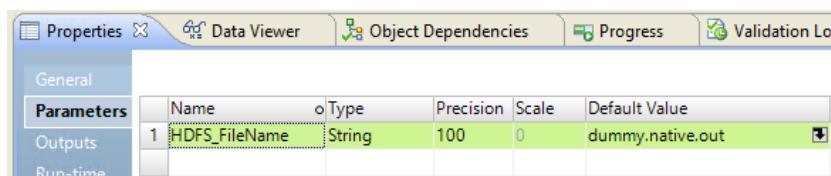
This mapping is only validated against Native mode:



It is set for native mode Execution:

Execution Environment: **Native**

This mapping use file name parameter:



For hive, blaze and spark mappings:

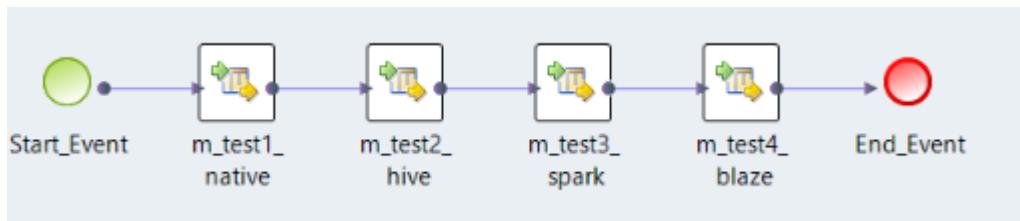
Review parameter and Execution Validation & Environment

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](#):

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 TABLE...s_dummy_m_test2_hive(Stage-0)	MAPREDUCE	root.users.infa	Tue Apr 24 13:53:10 +0000 2018	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	

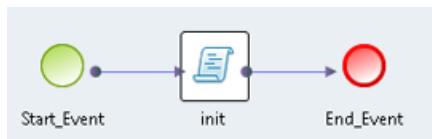
3. Lab01 - PIM Mass Ingestion Service

3.1 Purpose

Show new 10.2.1 MIS (Mass Ingestion Service) capability

3.2 Initialize Lab

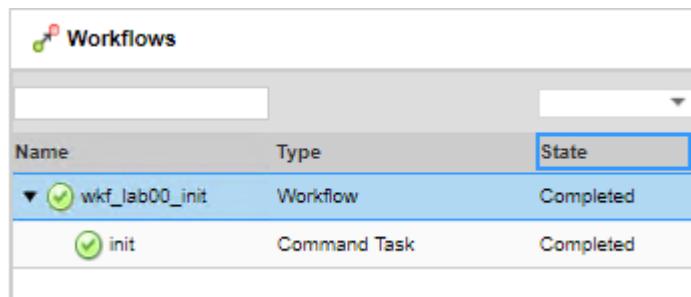
Workflow **wkf_lab01_init** calls a script to reinitialize the lab



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](#):

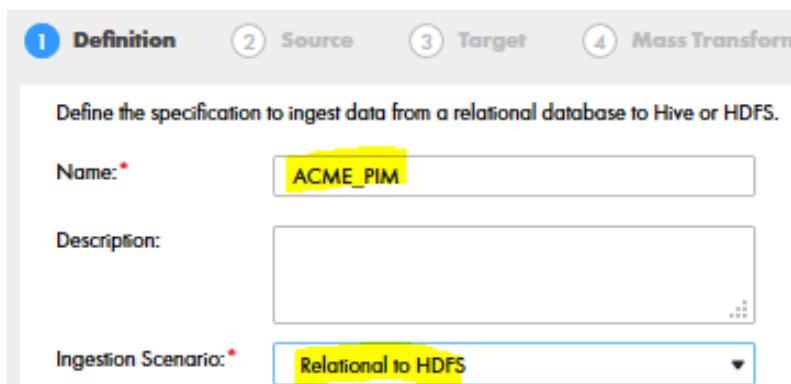


Name	Type	State
wkf_lab01_init	Workflow	Completed
init	Command Task	Completed

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



1 Definition 2 Source 3 Target 4 Mass Transform

Define the specification to ingest data from a relational database to Hive or HDFS.

Name: *

Description:

Ingestion Scenario: *

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

1 Definition 2 Source 3 Target 4 Mass Transformations 5 Transformation

Select the source database that you want to ingest data from.

Source Connection: * **ACME_PIM_jdbc**

Source Schema: * **ACME_PIM**

Source Tables: * Available: 0 Selected: 5

Find	Find
<input type="text"/>	<input type="text"/>
PRODUCT_BRANDS	<input type="button" value="<"/>
PRODUCT_CATEGORIES	<input type="button" value="^"/>
PRODUCT_DESCRIPTIONS	<input type="button" value="^"/>
PRODUCT_GENERIC	<input type="button" value="v"/>
PRODUCT_PRICES	<input type="button" value="v"/>

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

Configure the HDFS ingestion directory where you want to ingest the source tables.

Target Connection: * **HDFS_CDH_513**

Target Table Prefix: **MIS_**

Ingestion Directory: * **/user/infa/elevate/lab01/MIS**

Compression: * **None**

Delimiters: **Comma**

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

Filter By:

Drop Columns:

Trim:

Convert to Uppercase:

Convert to Lowercase:

Replace Columns:

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

 ACME_PIM was saved successfully.



3.4 Deploy & Run MIS project

Deploy

Deploy on:	DIS	Hadoop Connection:	HADOOP	Deploy
------------	-----	--------------------	--------	--------

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

Operating System Profile: None **Run now**

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, you should see:

ACME_PIM > Execution Statistics

Start Time: 4/24/2018, 3:26:14 PM
End Time: 4/24/2018, 3:27:33 PM
Data Integration Service: DIS1021

Ingestion Objects

(5)	Source	Status	End Time	Logs
PRODUCT_BRANDS	Completed	4/24/2018, 3:27:29 PM	↓	
PRODUCT_CATEGORI...	Completed	4/24/2018, 3:27:29 PM	↓	
PRODUCT_DESCRIPTI...	Completed	4/24/2018, 3:27:29 PM	↓	
PRODUCT_GENERIC	Completed	4/24/2018, 3:27:33 PM	↓	
PRODUCT_PRICES	Completed	4/24/2018, 3:27:29 PM	↓	

Select a table to view job statistics.

After a while you will see successful completion tasks

In [Informatica Monitoring](#) you can see application INFAMI_PIM_CRM was deployed:

Navigator

- Domain_1021
 - DIS1021
 - Ad Hoc Jobs
 - app_ACME_Ingest
 - app_ELEVATE
 - INFAMI_ACME_PIM

Actions

Refresh

Deployed Mapping Jobs

You may need to refresh the applications

In Deployed mappings you should see:

Deployed Mapping Jobs				
Name	Job Name	Type	State	
▶  m_ACME...	PRODUCT_CATE...	Deploye...	Completed	
▶  m_ACME...	PRODUCT_BRA...	Deploye...	Completed	
▶  m_ACME...	PRODUCT_PRIC...	Deploye...	Completed	
▶  m_ACME...	PRODUCT_GEN...	Deploye...	Completed	
▶  m_ACME...	PRODUCT_DES...	Deploye...	Completed	

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	

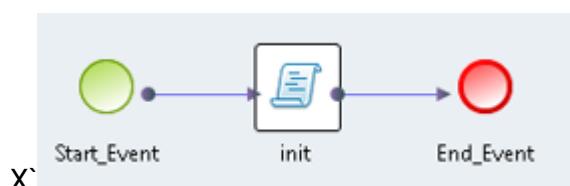
3.5 Extra Content (Optional)

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



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](#):

 wkf_lab01_init	Workflow	Completed
 init	Command Task	Completed

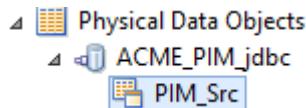
Use [HDFS Web Browser](#) and navigate to /user/infa/elevate/lab01, you should see:

/user/infa/elevate/lab01								Go!
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	

3.5.3 Review Lab Content

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

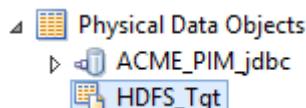
Observe the source object:



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

Name	Native Type	Precision	Scale	Primary Key	Nullable	Index
1 dummyString	varchar	10	0	<input type="checkbox"/>	<input type="checkbox"/>	
2 dummyString1	varchar	10	0	<input type="checkbox"/>	<input type="checkbox"/>	
3 dummyString2	varchar	10	0	<input type="checkbox"/>	<input type="checkbox"/>	

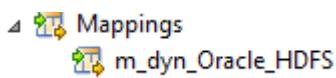
Observe the **target** object:



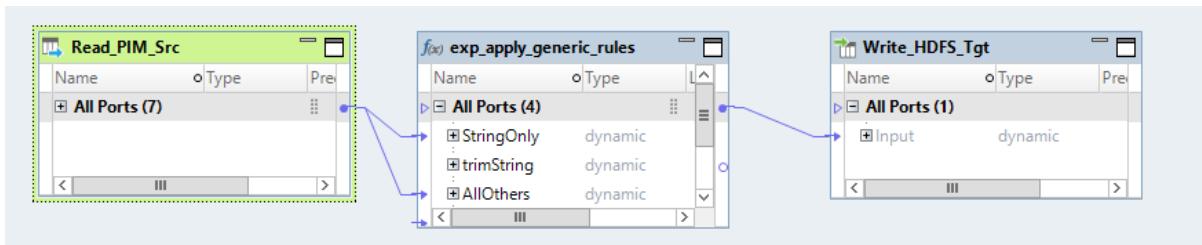
Open the Object and see in Advanced > Run-time Write output folder and parameter filename

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

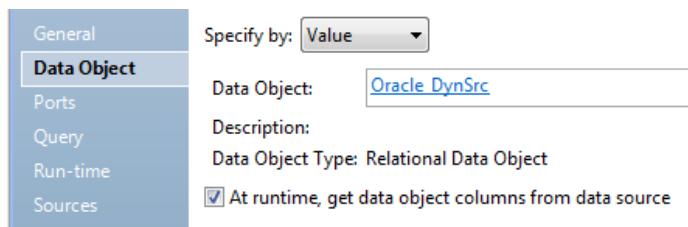
Observe the mapping:



The **mapping** is using dynamic ports



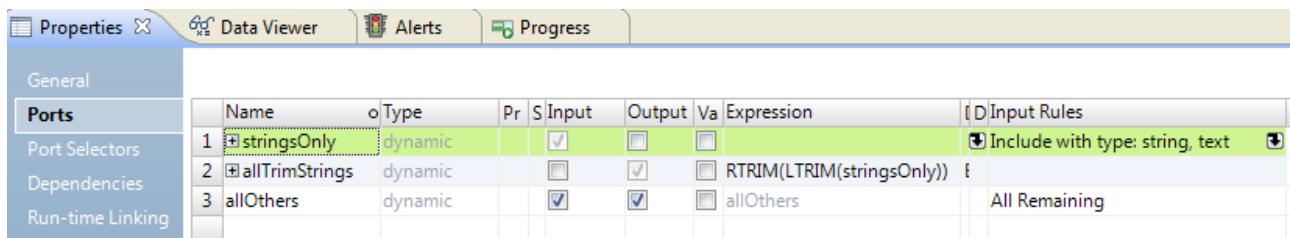
In mapping **Read Data Object** observe that it uses 'At runtime, get data object columns from data source'



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

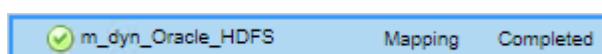
In the mapping Parameters table observe the configured parameters:

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

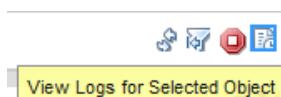
The mapping will use these default values as Input table and HDFS target file

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:

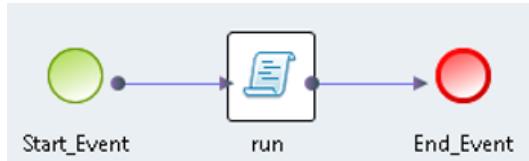
The Integration Service uses the default value [PRODUCT_BRANDS] for the mapping parameter [ORCL_TableName].

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](#):

	wkf_lab01_run	Workflow	Completed
	run	Command Task	Completed

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

	m_dyn_Oracle_HDFS	.tmp_m_dyn_Oracle_HDFS	Deploy...	Completed
	m_dyn_Oracle_HDFS	.tmp_m_dyn_Oracle_HDFS	Deploy...	Completed
	m_dyn_Oracle_HDFS	.tmp_m_dyn_Oracle_HDFS	Deploy...	Completed
	m_dyn_Oracle_HDFS	.tmp_m_dyn_Oracle_HDFS	Deploy...	Completed
	m_dyn_Oracle_HDFS	.tmp_m_dyn_Oracle_HDFS	Deploy...	Completed

Use [HDFS Web Browser](#) and navigate to /user/infa/elevate/lab01, you should see:

/user/infa/elevate/lab01								Gol!
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	

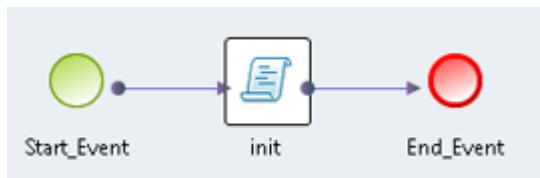
4. Lab02 - POS Mass Ingest with Dynamic mappings

4.1 Purpose

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

4.2 Initialize Lab

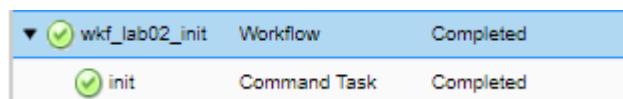
Workflow **wkf_lab02_init** calls a script to reinitialize the lab



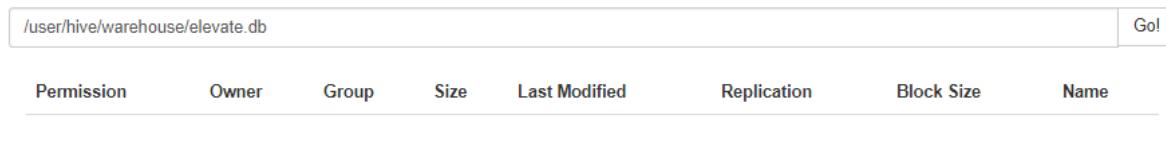
The init task command will clean all HIVE target data

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:



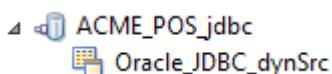
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
							/user/hive/warehouse/elevate.db

You should see **no hive tables**

4.3 Review Lab Content

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

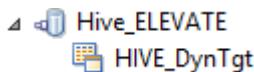
Observe the **source** object:



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

	Name	Native Type	Precision	Scale	Primary	Nullable
1	String	varchar	10	0	<input type="checkbox"/>	<input type="checkbox"/>
2	Others	varchar	10	0	<input type="checkbox"/>	<input type="checkbox"/>

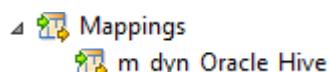
Observe the **target** object:



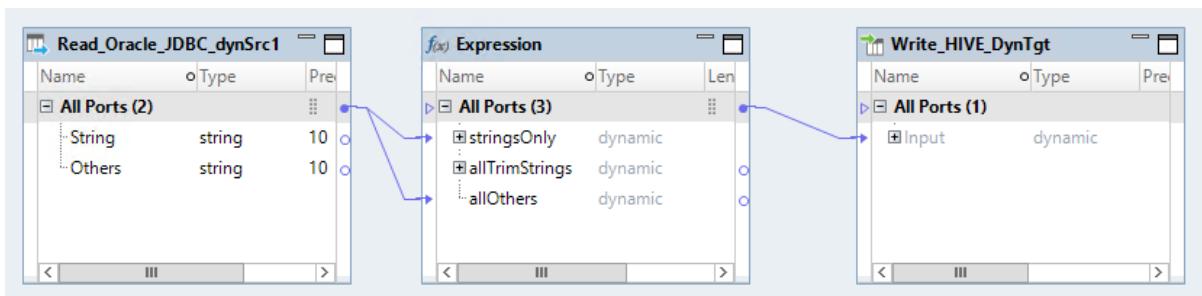
The **source** 'HIVE_DynTgt' uses 'Hive_ELEVATE' connection and 'fake' columns:

	Name	Native Type	Precision
1	dyn_col	string	10

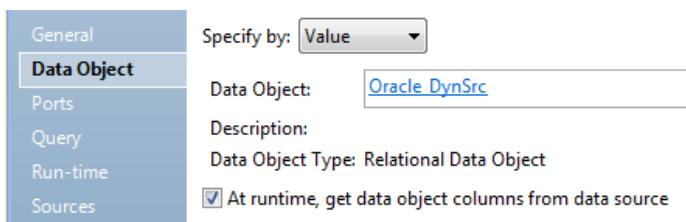
Observe the mapping:



The **mapping** is using dynamic ports



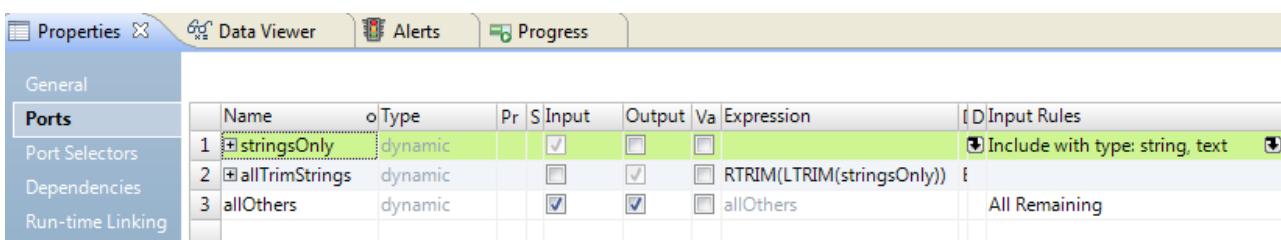
In mapping **Read Data Object** observe that it uses 'At runtime, get data object columns from data source'



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'

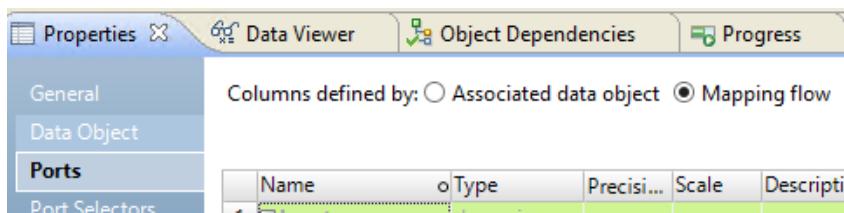
'allOthers' port is defined as dynamic, both input and output and include all remaining ports

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

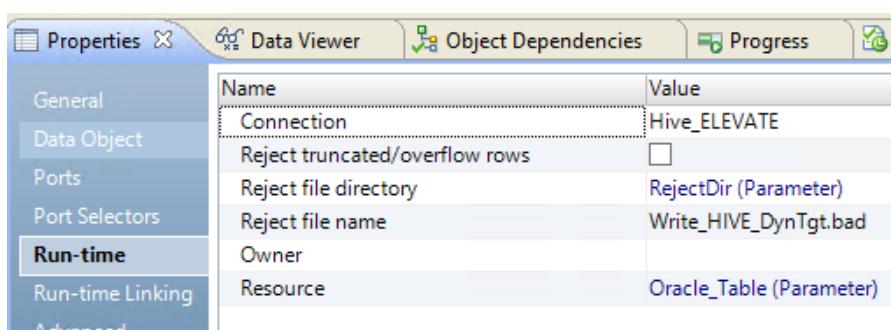
	Name	Type	Precision	Scale	Default Value	Description
1	Oracle_Table	String	100	0	STORES	
2	Hive_Format	String	10	0	AVRO	

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

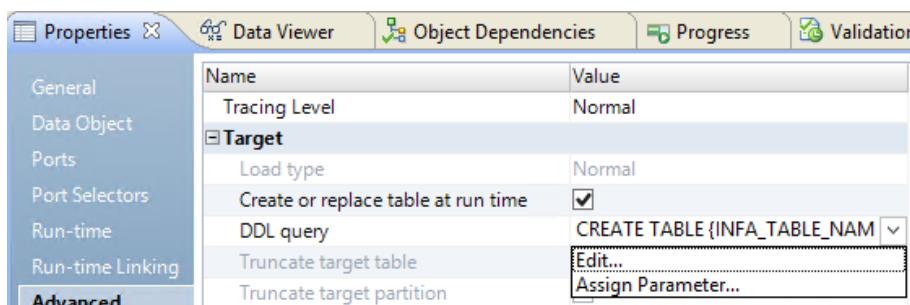
In mapping **Write Data Object** observe that it uses 'Mapping flow'



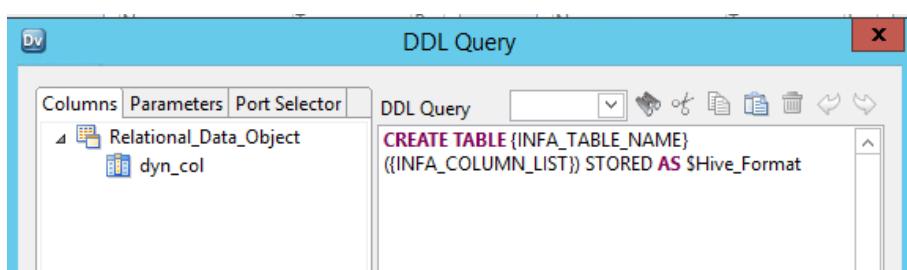
In 'Run-time' observe it will use the same mapping parameter for Hive target name



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



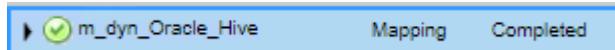
Edit the DDL Query



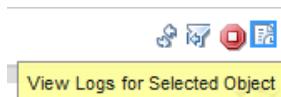
Observe the specific syntax and that it uses the parameter Hive_Format

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:

/user/hive/warehouse/elevate.db								Go!
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
drwxrwxrwx	infa	hive	0 B	Tue Apr 24 16:22:37 +0000 2018	0	0 B	stores	

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

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

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

```

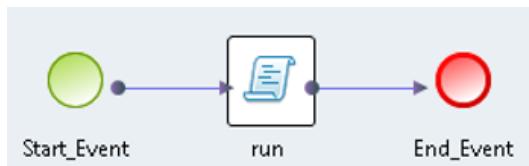
In next section, we will automate the ingestion of all POS tables into HDFS via a workflow

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

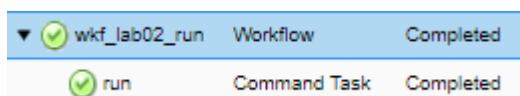
Workflow **wkf_lab02_run** calls a script to run the lab



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:

 m_dyn_Oracle_HIVE	.tmp_m_dyn_Ora...	Deploy...	Completed
 m_dyn_Oracle_HIVE	.tmp_m_dyn_Ora...	Deploy...	Completed
 m_dyn_Oracle_HIVE	.tmp_m_dyn_Ora...	Deploy...	Completed

Use [HDFS Web Browser](#) and navigate to /user/hive/warehouse/elevate.db:

/user/hive/warehouse/elevate.db								Go!
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
drwxrwxrwx	infa	hive	0 B	Tue Apr 24 17:24:32 +0000 2018	0	0 B	orderlines	
drwxrwxrwx	infa	hive	0 B	Tue Apr 24 17:24:16 +0000 2018	0	0 B	orders	
drwxrwxrwx	infa	hive	0 B	Tue Apr 24 17:24:04 +0000 2018	0	0 B	stores	

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

Name:*	<input type="text" value="ACME_POS"/>
Description:	<input type="text"/>
Ingestion Scenario:*	<input type="text" value="Relational to Hive"/>

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

Source Connection:*	<input type="text" value="ACME_POS_jdbc"/>												
Source Schema:*	<input type="text" value="ACME_POS"/>												
Source Tables:*	<table border="0"> <tr> <td style="vertical-align: top;"> Available: 0 </td> <td style="vertical-align: top;"> Selected: 3 </td> </tr> <tr> <td><input type="text" value="Find"/></td> <td><input type="text" value="Find"/></td> </tr> <tr> <td><input type="button" value=">"/></td> <td><input type="button" value="ORDERLINES"/></td> </tr> <tr> <td><input type="button" value="<"/></td> <td><input type="button" value="ORDERS"/></td> </tr> <tr> <td><input type="button" value=">>"/></td> <td><input type="button" value="STORES"/></td> </tr> <tr> <td><input type="button" value="<<"/></td> <td></td> </tr> </table>	Available: 0	Selected: 3	<input type="text" value="Find"/>	<input type="text" value="Find"/>	<input type="button" value=">"/>	<input type="button" value="ORDERLINES"/>	<input type="button" value="<"/>	<input type="button" value="ORDERS"/>	<input type="button" value=">>"/>	<input type="button" value="STORES"/>	<input type="button" value="<<"/>	
Available: 0	Selected: 3												
<input type="text" value="Find"/>	<input type="text" value="Find"/>												
<input type="button" value=">"/>	<input type="button" value="ORDERLINES"/>												
<input type="button" value="<"/>	<input type="button" value="ORDERS"/>												
<input type="button" value=">>"/>	<input type="button" value="STORES"/>												
<input type="button" value="<<"/>													

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

Target Connection: * **Hive_ELEVATE**

Target Schema: * **elevate**

Target Table Prefix: **mis_**

Hive Options DDL Query

Storage Format: * **Text**

External Table

External Location: **mis_**

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

Filter By: **mis_**

Drop Columns: **mis_**

Trim: **mis_**

Convert to Uppercase: **mis_**

Convert to Lowercase: **mis_**

Replace Columns: **mis_**

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**

ACME_POS was saved successfully.

4.5.3 Deploy & Run MIS project

Deploy

Deploy on: **DIS1021**

Hadoop Conne...

HADOOP_CDH

Deploy

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

Operating System Profile: **None**

Run now

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

ACME_POS > Execution Statistics

Job Properties

Mass Ingestion Specification: ACME_POS
 Started By: Administrator
 Start Time: 5/2/2018, 1:22:54 PM
 End Time: 5/2/2018, 3:46:42 PM
 Data Integration Service: DIS1021

Ingestion Status



Completed

Ingestion Objects

(3)	Source	Status	End Time	Logs
ORDERLINES		Completed	5/2/2018, 3:45:39 PM	↓
ORDERS		Completed	5/2/2018, 3:46:31 PM	↓
STORES		Completed	5/2/2018, 3:46:42 PM	↓

Select a table to view job statistics.

After a while you will see successful completion tasks

In [Informatica Monitoring](#) you can see application INFAMI_POS_CRM was deployed:

- ▼  INFAMI_ACME_POS
 -  Logical Data Objects
 -  REST Web Services
 -  Web Services
 -  SQL Data Services
 -  Deployed Mapping Jobs

Note: You may need to refresh the applications

In Deployed mappings you should see:

 m_ACME_POS	STORES_m_ACME_POS	Deployed Mappi... Completed
 m_ACME_POS	ORDERS_m_ACME_POS	Deployed Mappi... Completed
 m_ACME_POS	ORDERLINES_m_ACME_POS	Deployed Mappi... Completed

Use [HDFS Web Browser](#) and navigate to /user/hive/warehouse/elevate.db:

/user/hive/warehouse/elevate.db							Go!
Permission	Owner	Group	Size	Replication	Block Size	Name	
drwxrwxrwx	infa	hive	0 B	0	0 B	mis_orderlines	
drwxrwxrwx	infa	hive	0 B	0	0 B	mis_orders	
drwxrwxrwx	infa	hive	0 B	0	0 B	mis_stores	
drwxrwxrwx	infa	hive	0 B	0	0 B	orderlines	
drwxrwxrwx	infa	hive	0 B	0	0 B	orders	
drwxrwxrwx	infa	hive	0 B	0	0 B	stores	

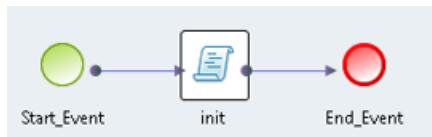
5. Lab03 – CRM Multiple Ingest (Demo only)

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



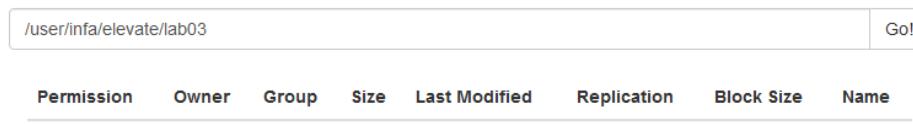
The **init** task command will create HDFS dummy file, clean target files

To initialize the lab, go to [Informatica Administration Console](#) and run the workflow

You should see in [Informatica Monitoring Console](#):

wkf_lab03_init	Workflow	Completed
init	Command Task	Completed

Use [HDFS Web Browser](#) and navigate to /user/infa/elevate/lab03, you should see:

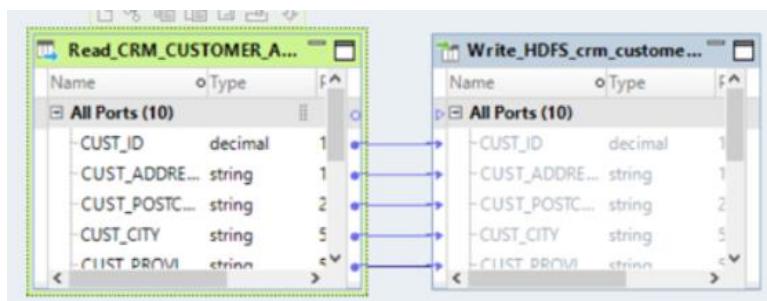


Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
							/user/infa/elevate/lab03

5.3 Review Lab Content

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

open the mapping m_load_HDFS_customer_address_oracle_native.

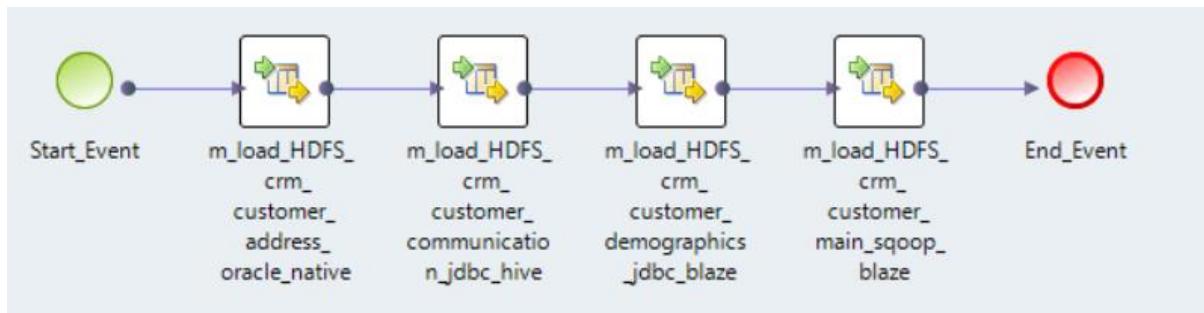


This passthrough mapping is using oracle native connection and native mode

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 TABLE...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

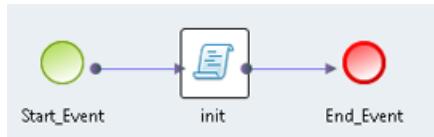
6. Lab04 - CRM Discover data

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](#):

	wkf_lab04_init	Workflow	Completed
	init	Command Task	Completed

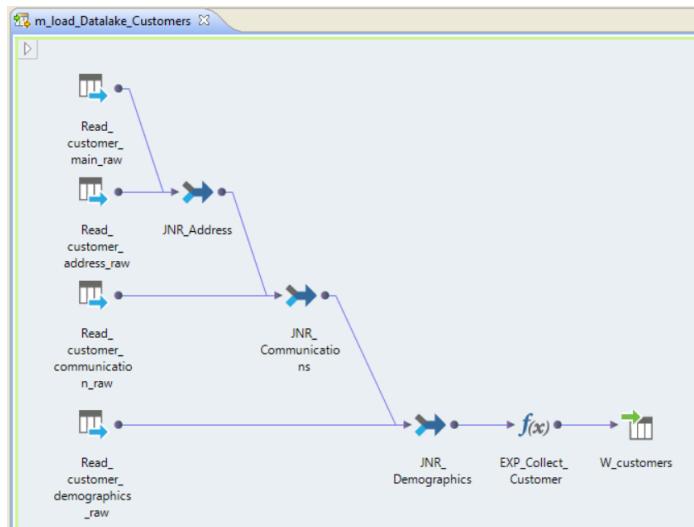
Using [HDFS Web Browser](#) go to /apps/hive/warehouse/elevate.db

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

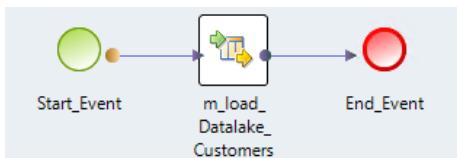
In Developer, go to lab04_* folder and open mapping **m_load_Datalake_Customers**:



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

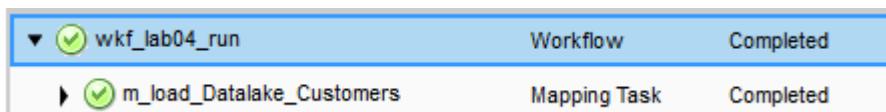
6.4 Run Lab

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](#):

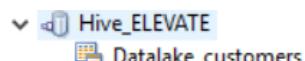


Using [HDFS Web Browser](#) go to /apps/hive/warehouse/elevate.db

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**



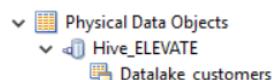
Run Data Preview, you should see:

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...

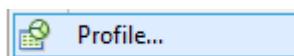
We will now create & run profiling on this table

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

- Enterprise Discovery Profile
- Multiple Profiles
- Profile

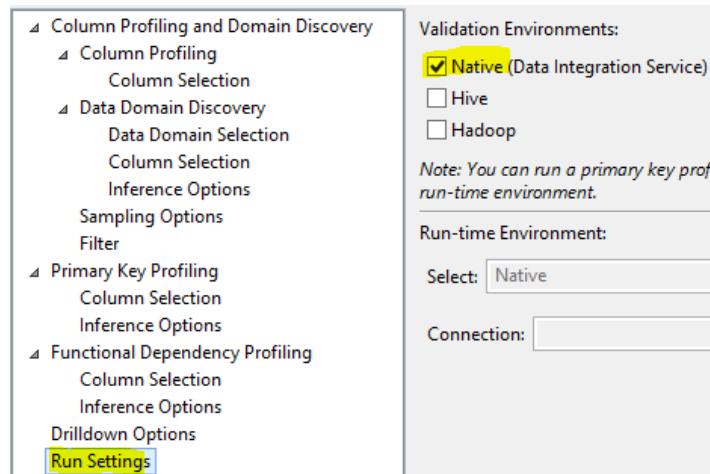
Rename it Profile_Datalake_customers_1_native

Name: Profile_Datalake_customers_1_native

Leave the Run Profile option checked and hit Next

Run Profile on finish.

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



Column Profiling and Domain Discovery

- Column Profiling
 - Column Selection
- Data Domain Discovery
 - Data Domain Selection
 - Column Selection
 - Inference Options
- Sampling Options
- Filter

Primary Key Profiling

- Column Selection
- Inference Options

Functional Dependency Profiling

- Column Selection
- Inference Options

Drilldown Options

Run Settings

Validation Environments:

Native (Data Integration Service)

Hive

Hadoop

Note: You can run a primary key profile in a run-time environment.

Run-time Environment:

Select: Native

Connection:

Hit Finish

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

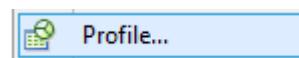
In our lab environment, this should take **~20 seconds** for completion

6.4.2 Create & Run Profile in blaze mode

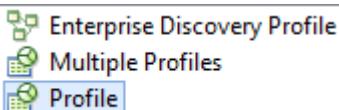
In Developer, open INFA/lab04_* and select **Datalake_customers** Data Object



Right click and chose Profile



Chose Simple Profile and hit Next



Rename it Profile_Datalake_customers_2_blaze

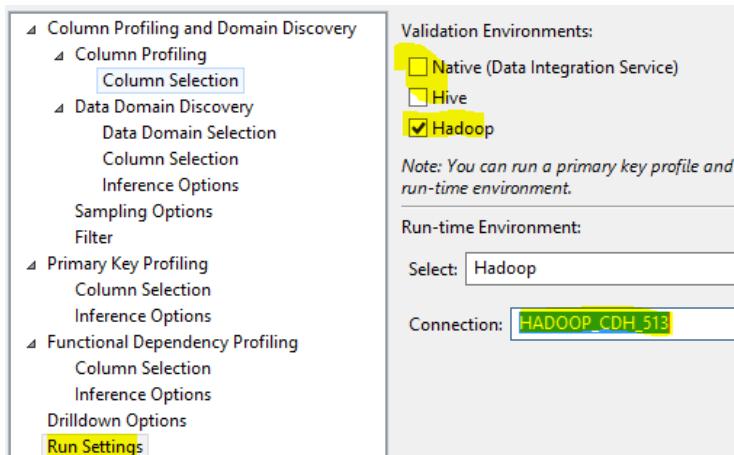
Name: Profile_Datalake_customers_2_blaze

Leave the Run Profile option checked and hit Next

Run Profile on finish.

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

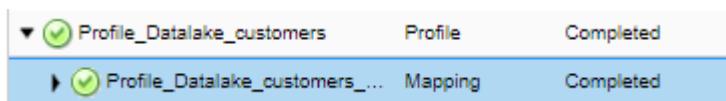
Then Browse and set the Hadoop connection:



The screenshot shows the 'Column Profiling and Domain Discovery' section. Under 'Validation Environments', 'Hadoop' is selected. In the 'Run-time Environment' section, 'Select:' is set to 'Hadoop' and the 'Connection:' dropdown is set to 'HADOOP_CDH_513'.

Hit Finish

You should see in [Informatica Monitoring Console](#):

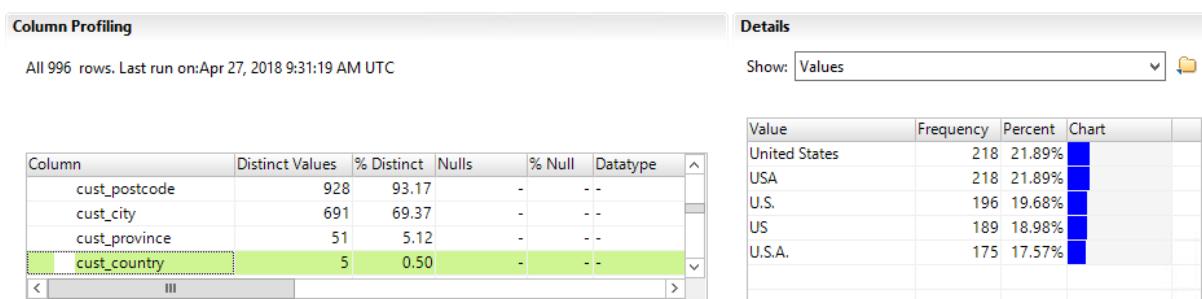


The monitoring console shows two completed profile tasks: 'Profile_Datalake_customers' and 'Profile_Datalake_customers_...'. Both tasks are marked as 'Completed'.

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

6.5 Review Profiling results for GENDER and TIER columns

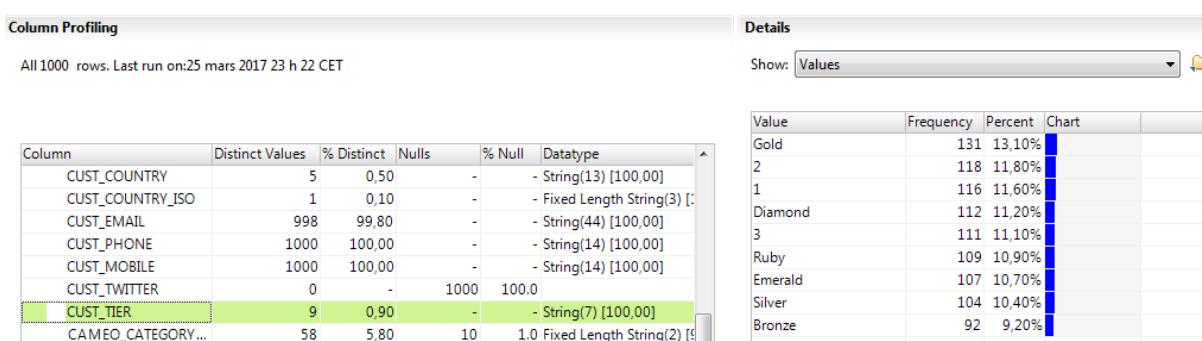
In Developer, go to the profile, results and see the **COUNTRY column**:



The screenshot shows the 'Column Profiling' interface for the 'cust_country' column. The 'Details' panel on the right displays the following data:

Value	Frequency	Percent	Chart
United States	218	21.89%	
USA	218	21.89%	
U.S.	196	19.68%	
US	189	18.98%	
U.S.A.	175	17.57%	

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



The screenshot shows the 'Column Profiling' interface for the 'CUST_TIER' column. The 'Details' panel on the right displays the following data:

Value	Frequency	Percent	Chart
Gold	131	13.10%	
2	118	11.80%	
1	116	11.60%	
Diamond	112	11.20%	
3	111	11.10%	
Ruby	109	10.90%	
Emerald	107	10.70%	
Silver	104	10.40%	
Bronze	92	9.20%	

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

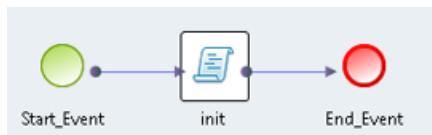
7. Lab05 - CRM Cleanse & Validate

7.1 Purpose

Show cleansing capabilities in hadoop (blaze) mode

7.2 Initialize Lab

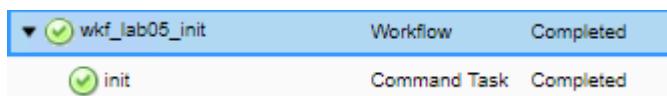
Workflow **wkf_lab05_init** calls a script to reinitialize the lab



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](#):



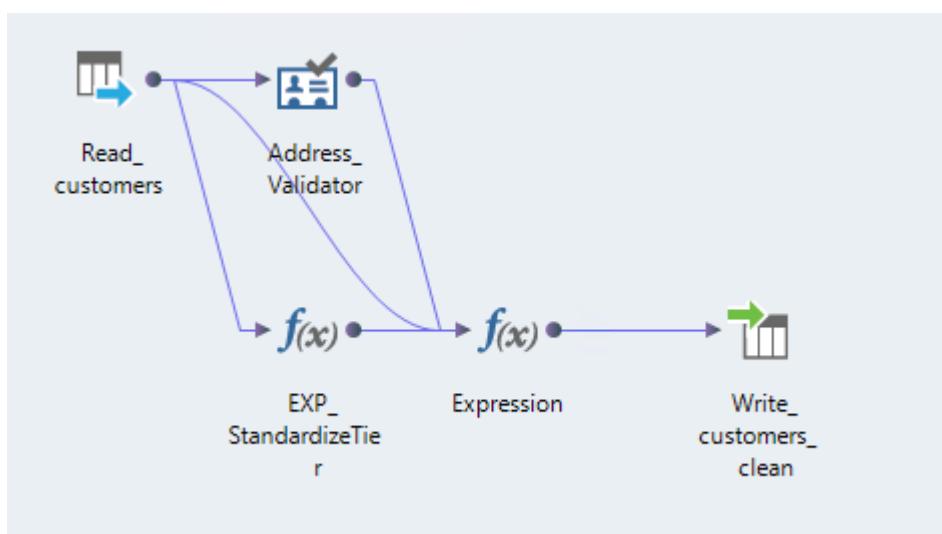
Using [HDFS Web Browser](#) go to /apps/hive/warehouse/elevate.db

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](#):

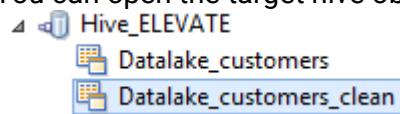


Using [HDFS Web Browser](#) go to /apps/hive/warehouse/elevate.db

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

/user/hive/warehouse/elevate.db								Go!
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
drwxrwxrwx	infa	hive	0 B	Fri Apr 27 11:04:50 +0000 2018	0	0 B	customers	
drwxrwxrwx	infa	hive	0 B	Fri Apr 27 10:19:52 +0000 2018	0	0 B	customers_clean	

You can open the target hive object



And Preview the cleansed Data

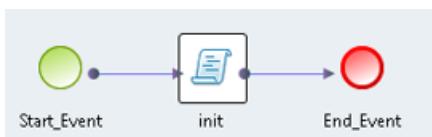
8. Lab06 - CRM Secure

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](#):

wfk_lab06_init	Workflow	Completed
init	Command Task	Completed

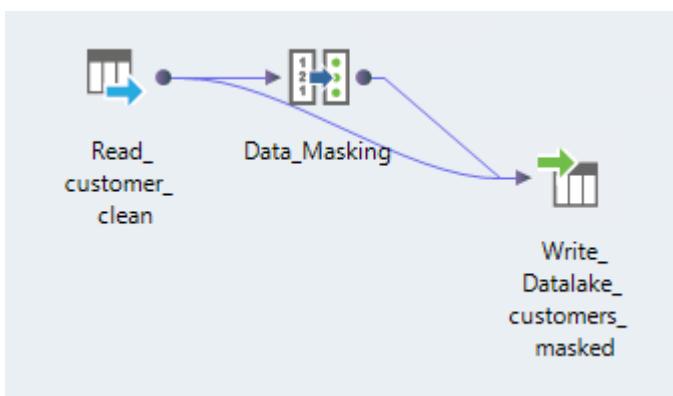
Using [HDFS Web Browser](#) go to /apps/hive/warehouse/elevate.db

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**

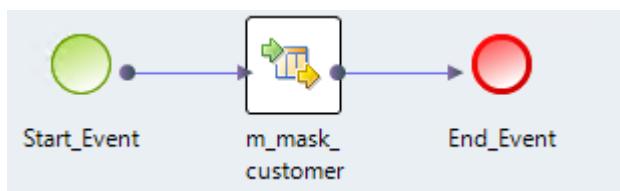
	cust_phone	cust_mobile	out_cust_phone	out_cust_mob...
1	95686683	956688193	20486310	558953365
2	619-422-687	619-252-939	361-033-013	020-953-095
3	339580555	339743737	679551024	926515700
4	+1204340655	+1204869743	+1396810008	+3823665678
5	402929363	402375562	201455680	973088763

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

We will now run this mapping in Hadoop Blaze mode.

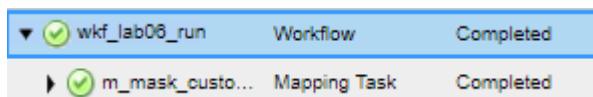
8.4 Run Lab

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

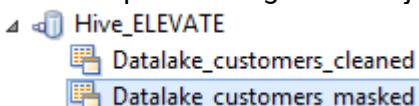
You should see **customer_masked** hive table was created

/user/hive/warehouse/elevate.db

Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxrwxrwx	infa	hive	0 B	Fri Apr 27 11:04:50 +0000 2018	0	0 B	customers
drwxrwxrwx	infa	hive	0 B	Fri Apr 27 10:19:52 +0000 2018	0	0 B	customers_clean
drwxrwxrwx	infa	hive	0 B	Fri Apr 27 11:26:17 +0000 2018	0	0 B	customers_masked

You can open the target hive object



And Preview the masked data

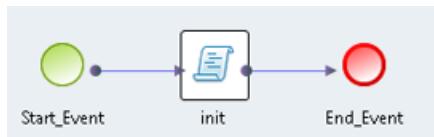
9. Lab07 - SIS Complex File DP & H2R

9.1 Purpose

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

9.2 Initialize Lab

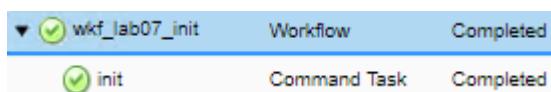
Workflow **wkf_lab07_init** calls a script to reinitialize the lab



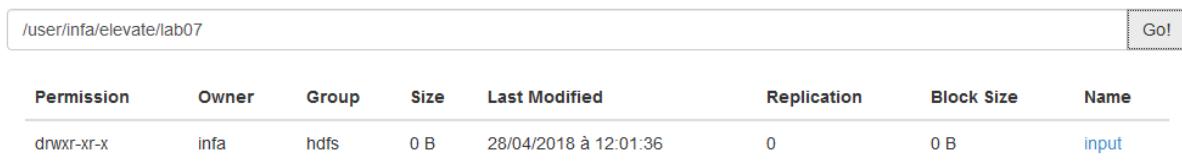
The **init** task command will clean target files

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:



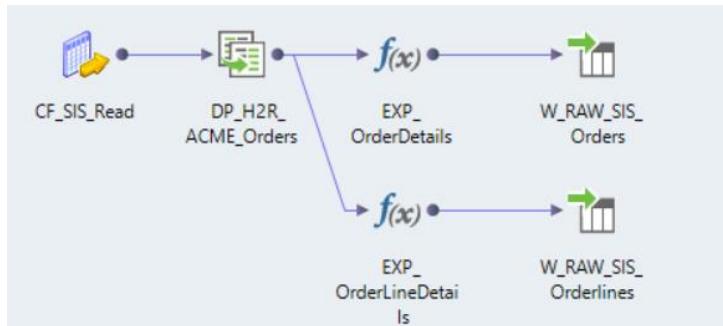
/user/infa/elevate/lab07							
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	infa	hdfs	0 B	28/04/2018 à 12:01:36	0	0 B	input

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

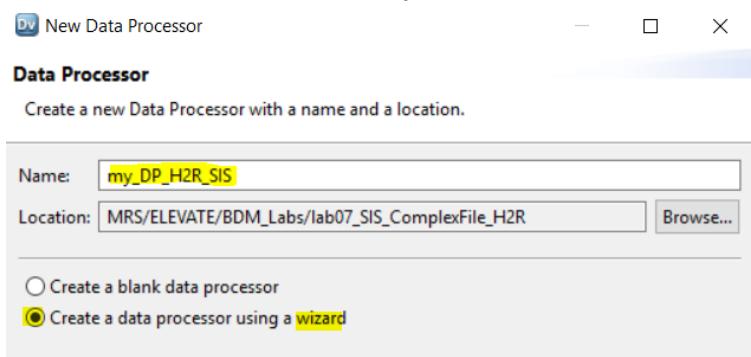
open the mapping **m_load_RAW_StoreInStore_Orders**



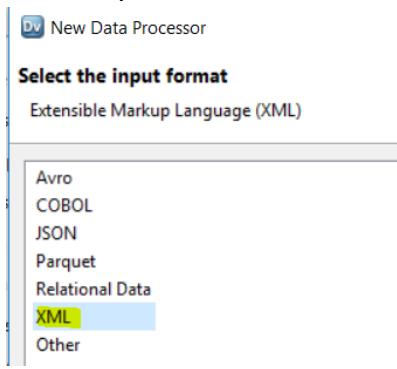
This mapping is using H2R transformation to parse complex files

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

- Create new Data Processor Object and use Wizard option



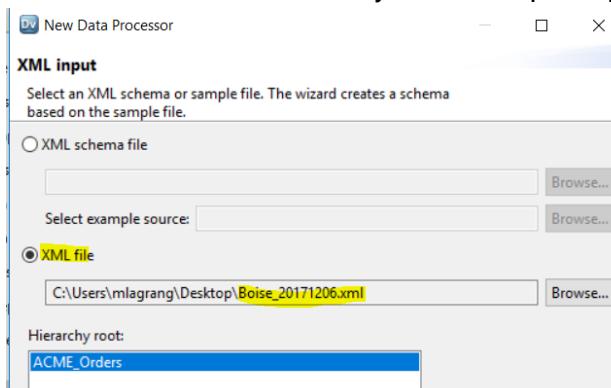
- Select Input format as XML



- [Copy to your Desktop](#) and unzip the below XML sample



- Choose XML and Browse to your Desktop unzipped XML sample



- Chose Output 'Relational Data' format, hit Finish

New Data Processor

Select the output format

Relational data stored in tables

Avro
COBOL
JSON
Parquet
Relational Data
XML
Other

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

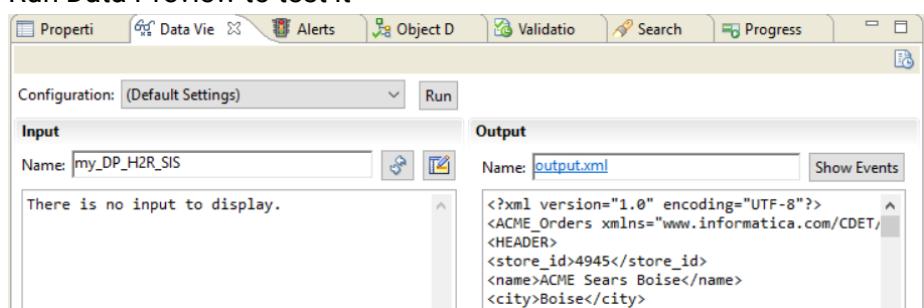
Ports

Show: Ports Input Mapping Output Mapping

Ports:

Name	Type	Port Type	Buffer/File	Location	Precision	Scale	P	Input Location
1 Input (1)	L-Input	string	Input	Buffer	1024	0		C:\Users\mlagrang\Desktop\output.xml

- Run Data Preview to test it



The screenshot shows the Data Preview window with two tabs: 'Input' and 'Output'.
Input: Configuration: (Default Settings) Run
 Name: my_DP_H2R_SIS
 There is no input to display.
Output: Name: output.xml Show Events
 XML content:
 <?xml version="1.0" encoding="UTF-8"?>
<ACME_Orders xmlns="www.informatica.com/CDET/">
<HEADER>
<store_id>4945</store_id>
<name>ACME Sears Boise</name>
<city>Boise</city>

Select the mapping **CF_SIS_Read** Read Object

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



Properties

General
Data Object
Ports
Run-time

Name	Type	Precisi...	Scale	Description
1 Data	binary	200000	0	
2 FileName	string	1024	0	

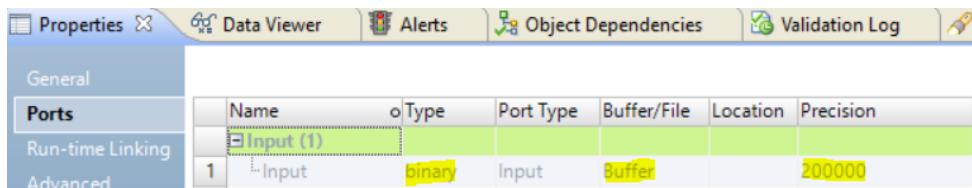
Run the Data Preview, you should see:

	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...)

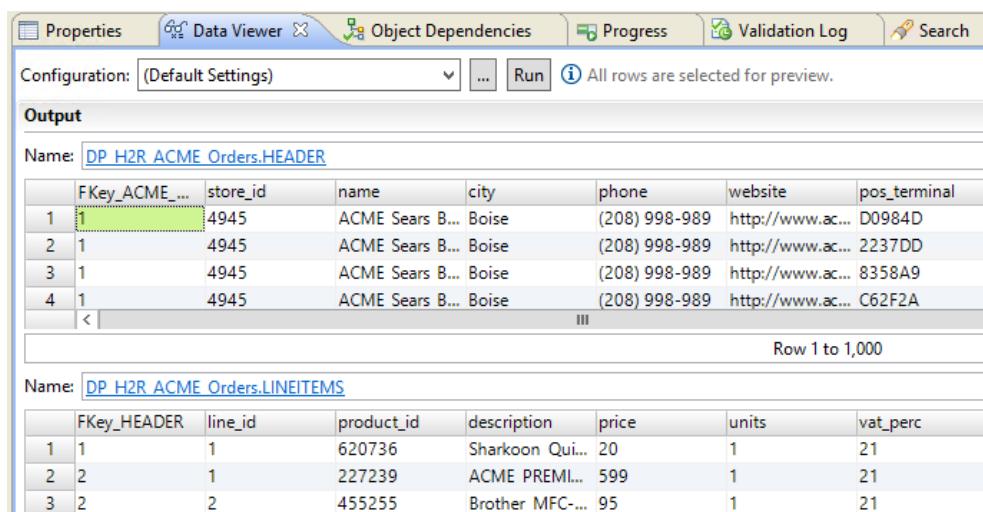
Go to the DP_H2R_ACME_Orders Transformation

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



The screenshot shows the Informatica Data Preview interface. The top navigation bar includes tabs for Properties, Data Viewer, Alerts, Object Dependencies, Validation Log, and a pencil icon. Below the tabs, there's a 'General' section with tabs for Ports, Run-time Linking, and Advanced. A table titled 'Ports' is displayed, showing a single row labeled 'Input (1)' with columns for Name, Type, Port Type, Buffer/File, Location, and Precision. The 'Name' column has a tooltip 'Input (1)', 'Type' is 'binary', 'Port Type' is 'Input', 'Buffer/File' is 'Buffer', 'Location' is 'zo0000', and 'Precision' is '200000'.

Run Data Preview, you should see:



The screenshot shows the Informatica Data Preview interface with two tables displayed. The first table is named 'DP_H2R_ACME_Orders.HEADER' and contains 4 rows of data. The second table is named 'DP_H2R_ACME_Orders.LINEITEMS' and contains 3 rows of data. Both tables have columns such as FKey_ACME..., store_id, name, city, phone, website, pos_terminal, FKey_HEADER, line_id, product_id, description, price, units, and vat_perc.

	FKey_ACME...	store_id	name	city	phone	website	pos_terminal
1	1	4945	ACME Sears B...	Boise	(208) 998-989	http://www.ac...	D0984D
2	1	4945	ACME Sears B...	Boise	(208) 998-989	http://www.ac...	2237DD
3	1	4945	ACME Sears B...	Boise	(208) 998-989	http://www.ac...	8358A9
4	1	4945	ACME Sears B...	Boise	(208) 998-989	http://www.ac...	C62F2A

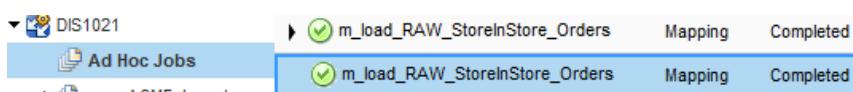
	FKey_HEADER	line_id	product_id	description	price	units	vat_perc
1	1	1	620736	Sharkoon Qui...	20	1	21
2	2	1	227239	ACME PREMI...	599	1	21
3	2	2	455255	Brother MFC-...	95	1	21

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

9.4 Run Lab

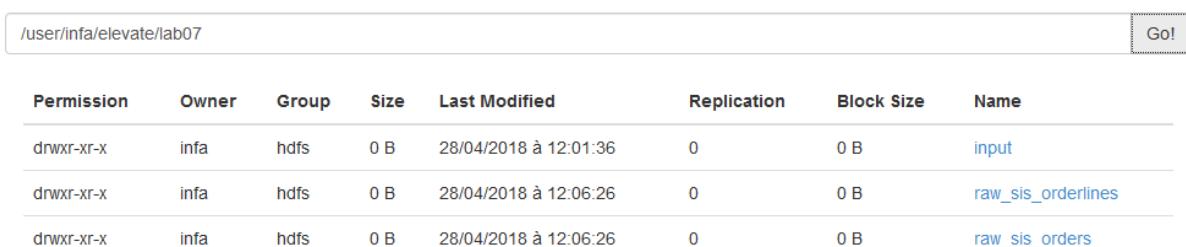
Configure the mapping to run in native & Blaze mode, Run it and compare executions.

You should see in [Informatica Monitoring Console](#):



The screenshot shows the Informatica Monitoring Console. It displays a tree structure under 'DIS1021'. Under 'Ad Hoc Jobs', there are two entries: 'm_load_RAW_StoreInStore_Orders' and another 'm_load_RAW_StoreInStore_Orders'. Both entries are marked with a green checkmark and the status 'Completed'.

Use [HDFS Web Browser](#) and navigate to /user/infa/elevate/lab07, you should see:



The screenshot shows the HDFS Web Browser interface. The URL bar shows '/user/infa/elevate/lab07'. The main area displays a table of files with columns: Permission, Owner, Group, Size, Last Modified, Replication, Block Size, and Name. There are three files listed: 'input' (drwxr-xr-x, infa, hdfs, 0 B, 28/04/2018 à 12:01:36, 0, 0 B, input), 'raw_sis_orderlines' (drwxr-xr-x, infa, hdfs, 0 B, 28/04/2018 à 12:06:26, 0, 0 B, raw_sis_orderlines), and 'raw_sis_orders' (drwxr-xr-x, infa, hdfs, 0 B, 28/04/2018 à 12:06:26, 0, 0 B, raw_sis_orders).

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	infa	hdfs	0 B	28/04/2018 à 12:01:36	0	0 B	input
drwxr-xr-x	infa	hdfs	0 B	28/04/2018 à 12:06:26	0	0 B	raw_sis_orderlines
drwxr-xr-x	infa	hdfs	0 B	28/04/2018 à 12:06:26	0	0 B	raw_sis_orders

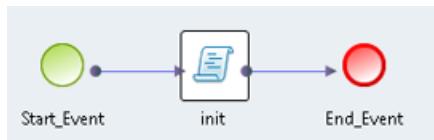
10. Lab08 - Call Center Unstructured File ISD

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](#):

	wkf_lab08_init	Workflow	Completed
	init	Command Task	Completed

Use [HDFS Web Browser](#) and navigate to /user/infa/elevate/lab08/input, you should see:

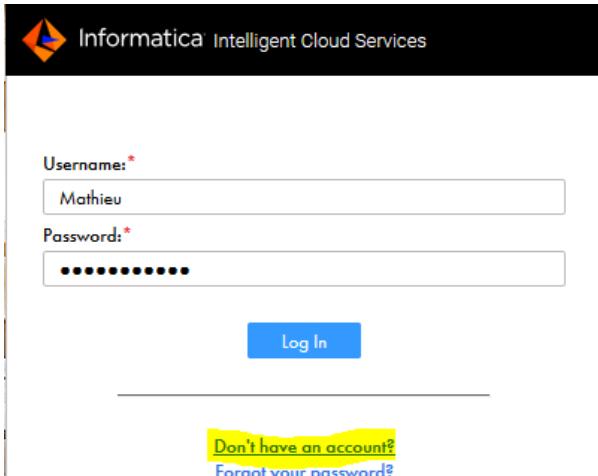
/user/infa/elevate/lab08/input								Go!
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
-rw-r--r--	infa	supergroup	453.4 KB	Wed May 02 12:19:12 +0000 2018	3	128 MB	callcenter.txt	

10.3 Build & Export an ISD Model from Cloud

[Copy to your Desktop](#) the sample file we will be working with:

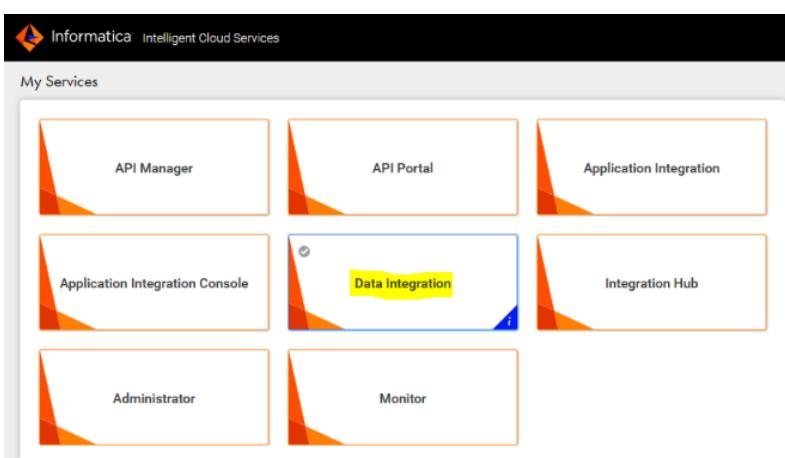


Go to Informatica Cloud: <https://dm-us.informaticacloud.com/identity-service/home>



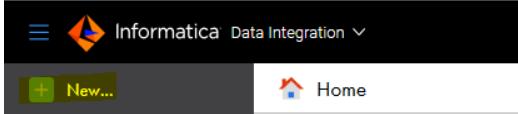
The screenshot shows the Informatica Intelligent Cloud Services login page. It features a black header with the Informatica logo and the text "Intelligent Cloud Services". Below the header is a form with two input fields: "Username:" containing "Mathieu" and "Password:" containing a series of dots. A blue "Log In" button is positioned below the password field. At the bottom of the form, there are two links: "Don't have an account?" and "Forgot your password?".

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



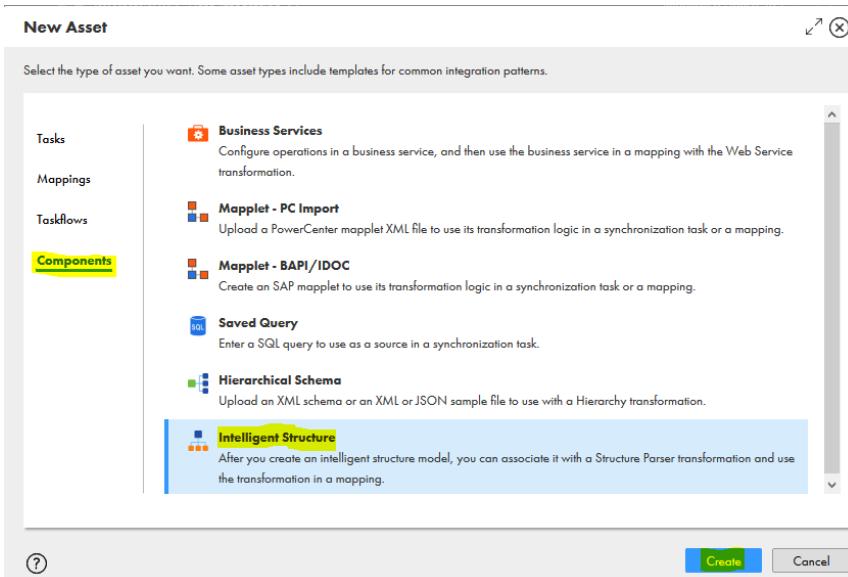
The screenshot shows the Informatica Intelligent Cloud Services home screen under the "My Services" section. It displays several service tiles: API Manager, API Portal, Application Integration, Application Integration Console, Data Integration (which is highlighted with a yellow box), Integration Hub, Administrator, and Monitor.

Click on Data Integration



The screenshot shows a sub-menu for "Data Integration". It includes a "New..." button and a "Home" link.

Click on New



The screenshot shows the "New Asset" dialog box. The left sidebar lists "Tasks", "Mappings", "Taskflows", and "Components" (which is selected and highlighted with a yellow box). The main pane displays asset types under "Business Services": "Mapplet - PC Import" (with a sub-note about PowerCenter mapplet XML file), "Mapplet - BAPI/IDOC" (with a sub-note about SAP mapplet transformation logic), "Saved Query" (with a sub-note about SQL query as a source), "Hierarchical Schema" (with a sub-note about XML or JSON schema), and "Intelligent Structure" (which is highlighted with a blue box and has a sub-note about creating an intelligent structure model for mapping). At the bottom right of the dialog are "Create" and "Cancel" buttons.

Select Components, then Intelligent Structure and Hit Create

Intelligent Structure5

Intelligent Structure Details

Name:*	ACME_CC	
Location:*	Default	Browse
Description:	Acme Call Center Demo file	
Sample File	Choose File	Browse

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

Select Acme_Demo.txt from your Desktop

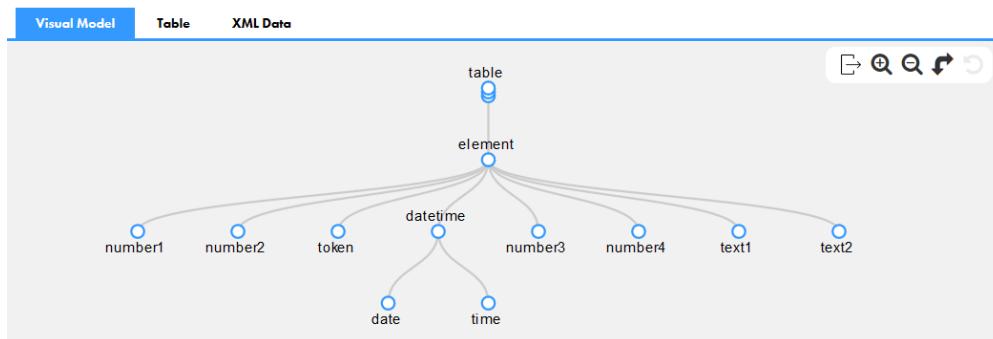
Intelligent Structure Details

Name:*	ACME_CC		
Location:*	Default	Browse	
Description:	Acme Call Center Demo file		
Sample File	Acme_CC.txt	Browse	Discover Structure

```

569 8089 INFO 2017-01-01 11:14:47 183 306127 Product info request Customer requested info on product 306127
918 8907 INFO 2017-01-01 11:27:32 38 101782 Product info request Customer requested info on product 101782
765 4898 COMPLAINT 2017-01-01 13:13:17 36 101788 Product complaint Customer had issues with product 101788
  
```

The Data is shown
Hit Discover Structure



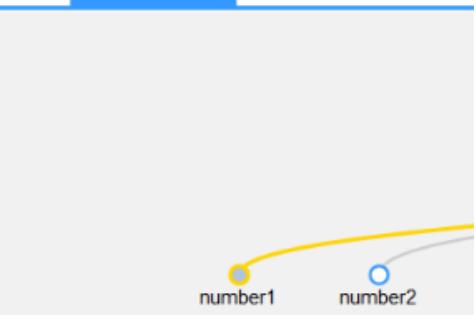
After a short while a model is presented

Move your mouse to first element: 'number1'

Sample File Acme_CC.txt

569	8089	INFO	2017-01-01	11:14:47	183	306127	Pr
918	8907	INFO	2017-01-01	11:27:32	38	101782	Pr
765	4898	COMPLAINT	2017-01-01	13:13:17	36	101781	
809	7545	INFO	2017-01-01	1:34:52	153	376428	Pr
569	4770	RETURNED	2017-01-01	2:10:20	394	101754	Pr

Visual Model Table XML Data



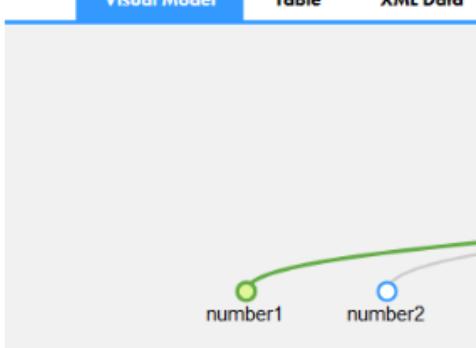
You can see which data it corresponds to

Click on first element: 'number1'

Sample File Acme_CC.txt

569	8089	INFO	2017-01-01	11:14:47	183	306127	Pr
918	8907	INFO	2017-01-01	11:27:32	38	101782	Pr
765	4898	COMPLAINT	2017-01-01	13:13:17	36	101781	
809	7545	INFO	2017-01-01	1:34:52	153	376428	Pr
569	4770	RETURNED	2017-01-01	2:10:20	394	101754	Pr

Visual Model Table XML Data



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

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

Data for "number1":

Possible element types (confidence %):

- ✓ **number** (67%)

Data Sample:

```

569
918
765
809
557
171
171

```

Close

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

569 8089	INFO	2017-01-01	11:14:47	183 306127	Product info re
918 8907	INFO	2017-01-01	11:27:32	38 101782	Product info re
765 4898	COMPLAINT	2017-01-01	13:13:17	36 101788	Product cc
809 7545	INFO	2017-01-01	1:34:52	153 376428	Product info re
557 4770	DETERMINE	2017-01-01	9:10:20	394 101754	Product return

Visual Model Table XML Data

number

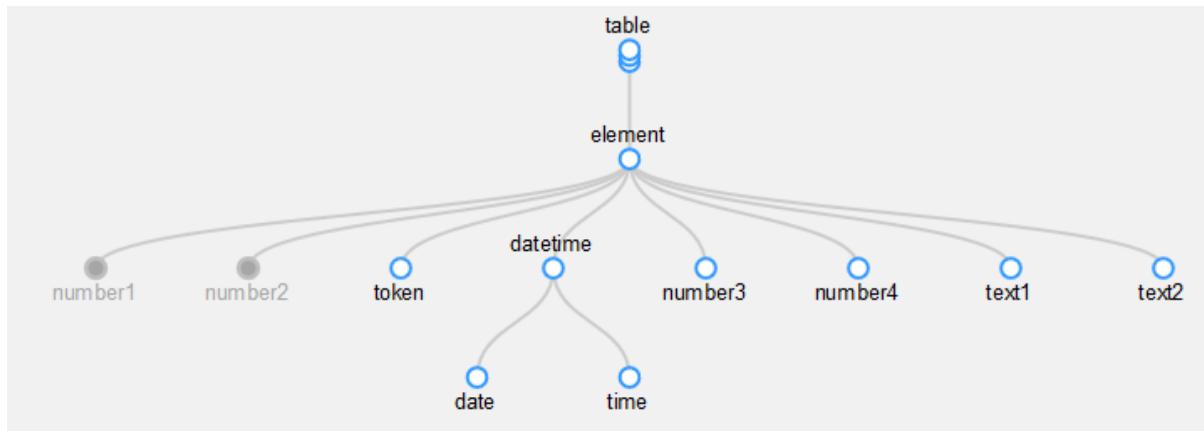
Open Data... **Ctrl+O**
 Flatten
 Collapse
 Expand
 Split...
 Convert to Child Value

Include in Structure **Ctrl+X**
Exclude from Structure **Ctrl+X**

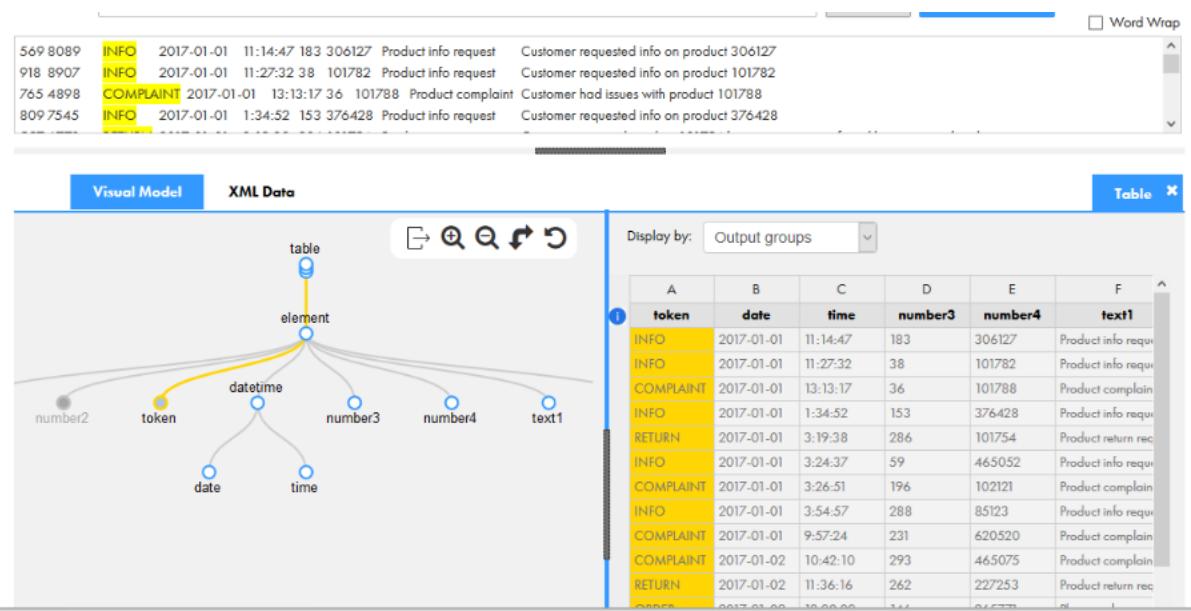
Delete
 Rename **F2**

Select as Primary Key
 Promote to Group
 Deselect as Primary Key
 Join to Parent Group

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**"

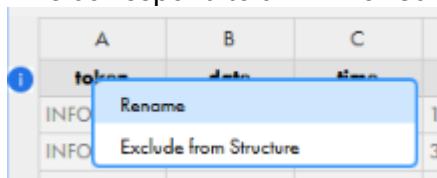


Click on Table then navigate to column A: 'token'

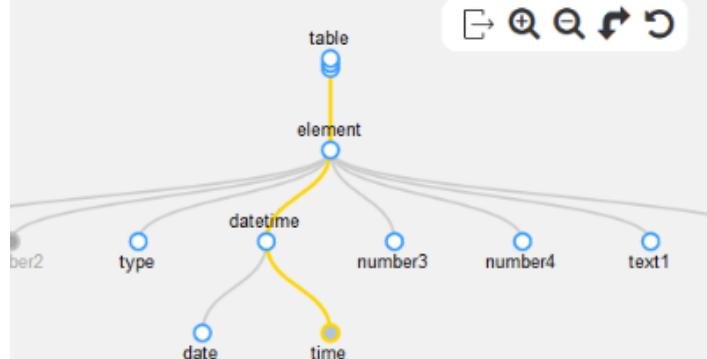


A	B	C	D	E	F
INFO	2017-01-01	11:14:47	183	306127	Product info request
INFO	2017-01-01	11:27:32	38	101782	Product info request
COMPLAINT	2017-01-01	13:13:17	36	101788	Customer had issues with product 101788
INFO	2017-01-01	1:34:52	153	376428	Product info request
INFO	2017-01-01	3:19:38	286	101754	Product return request
INFO	2017-01-01	3:24:37	59	465052	Product info request
COMPLAINT	2017-01-01	3:26:51	196	102121	Product complaint
INFO	2017-01-01	3:54:57	288	85123	Product info request
COMPLAINT	2017-01-01	9:57:24	231	620520	Product complaint
COMPLAINT	2017-01-02	10:42:10	293	465075	Product complaint
RETURN	2017-01-02	11:36:16	262	227253	Product return request
INFO	2017-01-02	19:00:00	344	946738	Product info request

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



Visual Model XML Data



Display by: Output groups

A	B	C	D
type	date	time	number3
INFO	2017-01-01	11:14:47	183
INFO	2017-01-01	11:27:32	38
COMPLAINT	2017-01-01	13:13:17	36
INFO	2017-01-01	1:34:52	153
RETURN	2017-01-01	3:19:38	286
INFO	2017-01-01	3:24:37	59
COMPLAINT	2017-01-01	3:26:51	196
INFO	2017-01-01	3:54:57	288

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**'

C	D	E	F
time	number3	number4	
11:14:47	183		Rename
11:27:32	38		Exclude from Structure
13:13:17	36	101788	Product complain
1:34:52	153	376428	Product info requ
3:19:38	286	101754	Product return req
3:24:37	59	465052	Product info requ

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

569 8089	INFO	2017-01-01	11:14:47	183	306127	Product info request	Customer requested info on product 306127
918 8907	INFO	2017-01-01	11:27:32	38	101782	Product info request	Customer requested info on product 101782
765 4898	COMPLAINT	2017-01-01	13:13:17	36	101788	Product complaint	Customer had issues with product 101788
809 7545	INFO	2017-01-01	1:34:52	153	376428	Product info request	Customer requested info on product 376428

Visual Model XML Data



Display by: Output groups

A	B	C	D	E
type	date	time	number4	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 is a product id, let's rename it **product**

number	306127	Rename
	101782	Exclude from Structure
	101788	Product info request

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
COMPLAINT	2017-01-01	3:24:51	102121	Product complaint

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

C	D	E
time	product	text1
I:14	Rename	request
I:21	Exclude from Structure	request

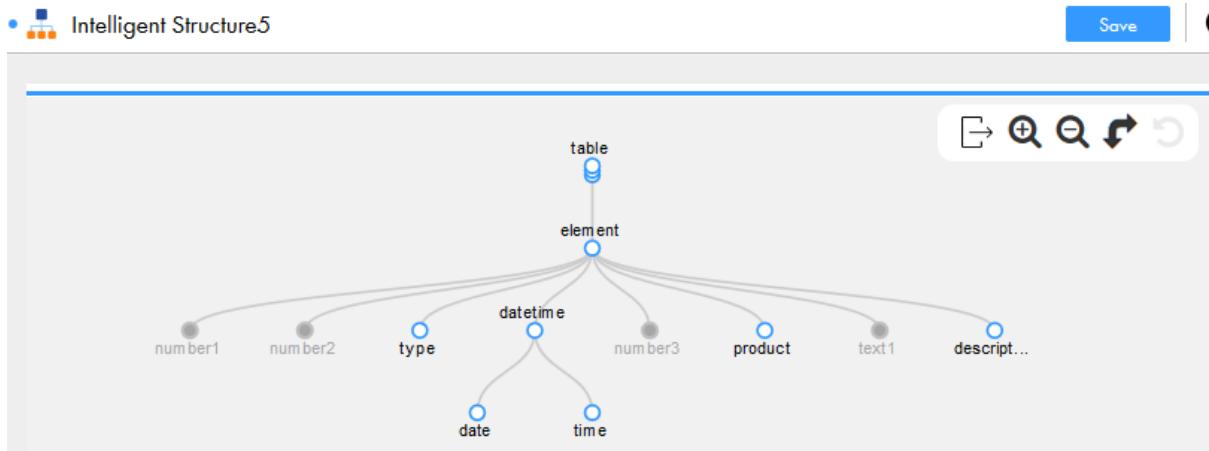
For now last columns (now E): 'text1'

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
420520	Customer had issues with product 420520

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

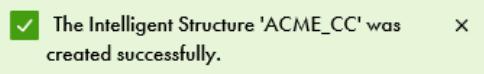
Rename
Exclude from Structure

Close the Table view, you should see:

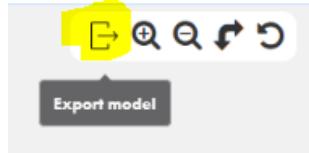


Click **Save**

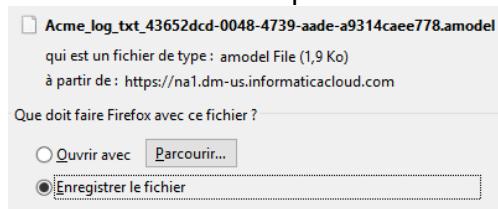
You should see:



Let's now export the model:



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



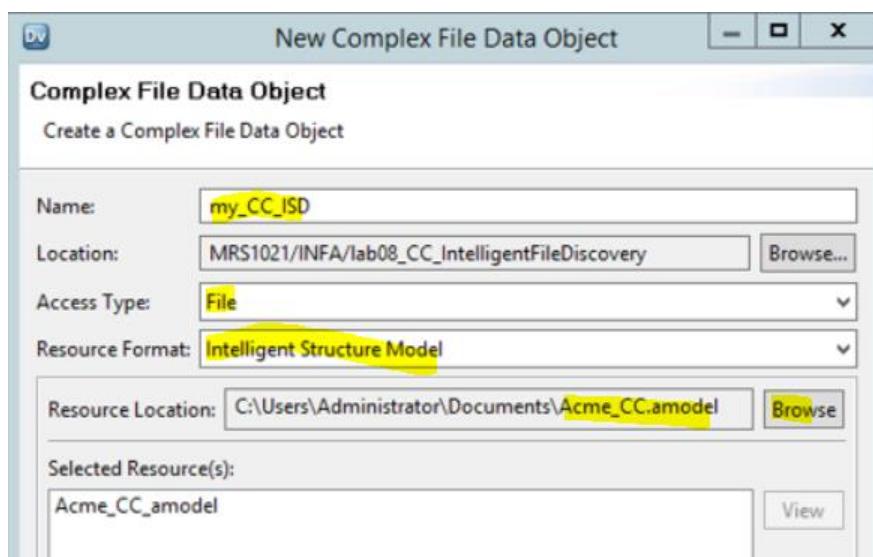
Save it to your Desktop and rename it to **Acme_CC.amodel**

For your convenience you can also [Copy to your Desktop](#) the resulting model:

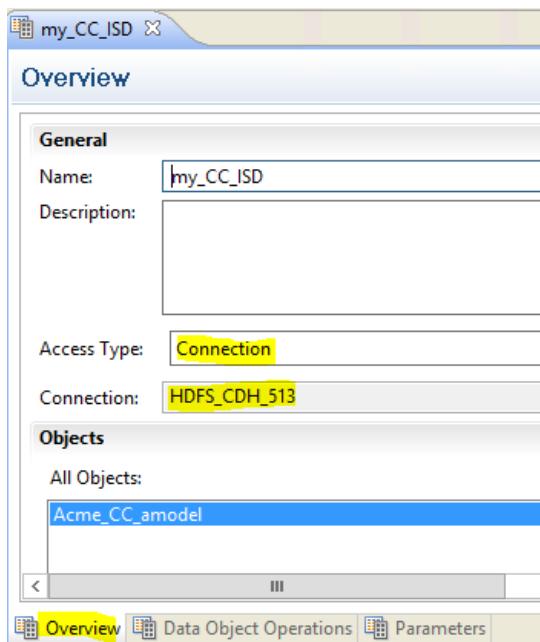


10.4 Import & Run ISD Model into Developer

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**



Overview

General

Name: my_CC_ISD

Description:

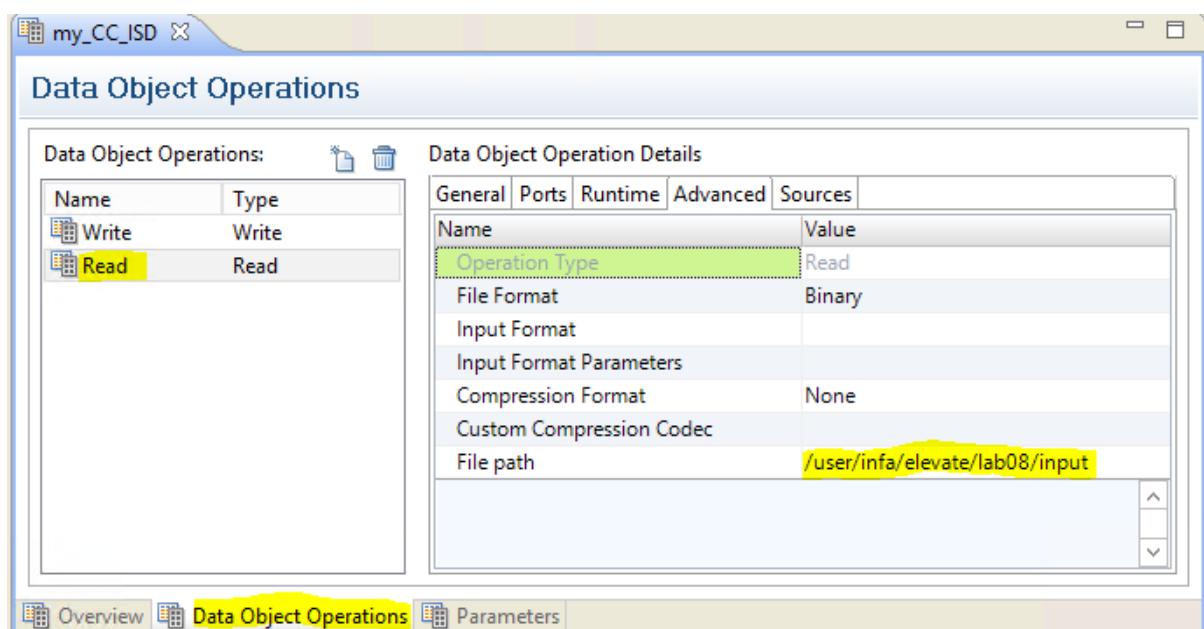
Access Type: Connection

Connection: HDFS_CDH_513

Objects

All Objects: Acme_CC_amodel

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



Data Object Operations

Name	Type
Write	Write
Read	Read

Data Object Operation Details

General		Ports	Runtime	Advanced	Sources
Name	Value				
Operation Type	Read				
File Format	Binary				
Input Format					
Input Format Parameters					
Compression Format	None				
Custom Compression Codec					
File path	/user/infa/elevate/lab08/input				

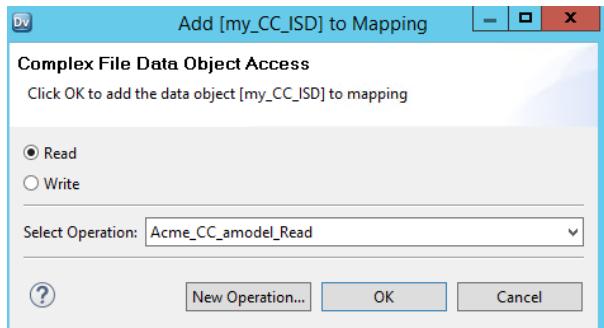
We need to position the Complex File to read to the HDFS Call center file.

To do so: go to **Data Object Operations**, select the **Read** operation then **Advanced** and enter the path: **/user/infa/elevate/lab08/input**

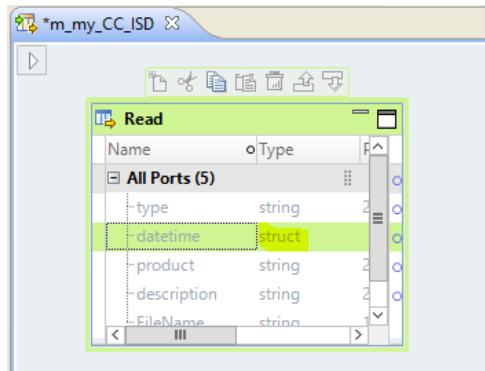
You should see:

-  Physical Data Objects
-  HDFS_CDH_513
-  my_CC_ISD

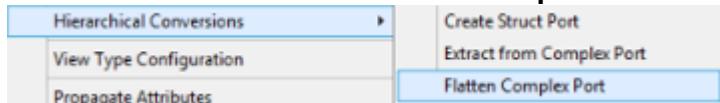
Create a mapping **m_my_CC_ISD** and Drag & drop **my_CC_ISD** Complex File as **Read** Object



You should see:



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

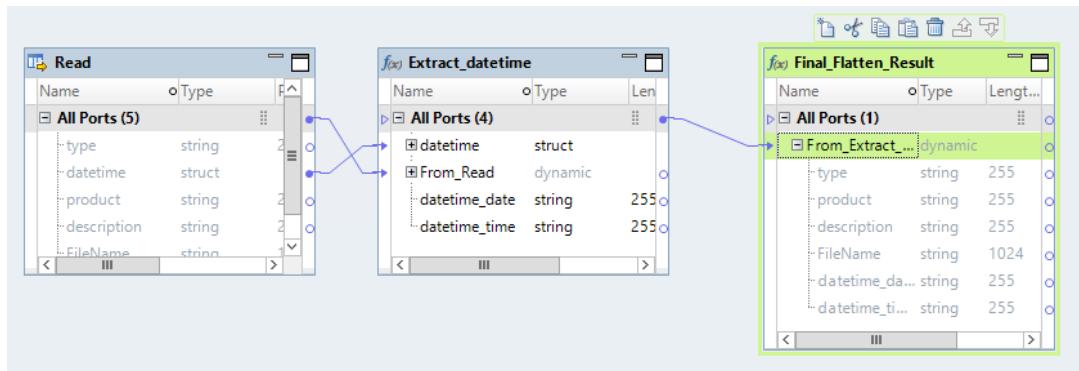


Then select datetime

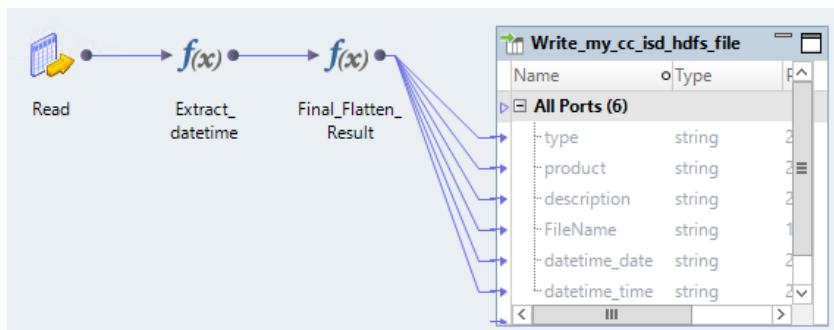
Name	Type	Type Configuration	Select...
datetime	struct	(datetime)	<input checked="" type="checkbox"/>
date	string	N/A	
time	string	N/A	

and hit Finish.

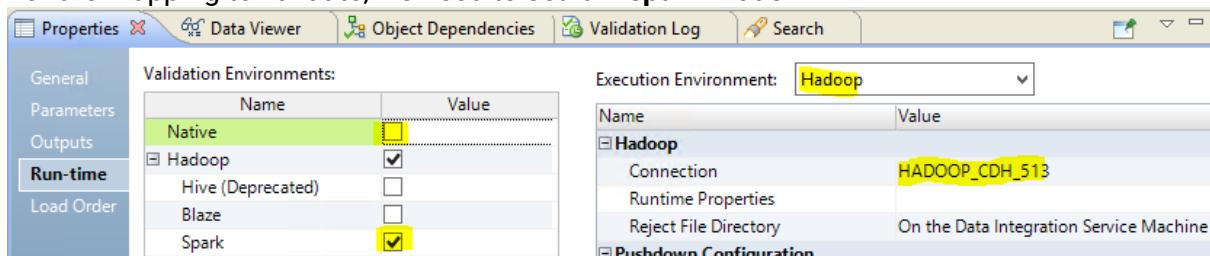
You should see:



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



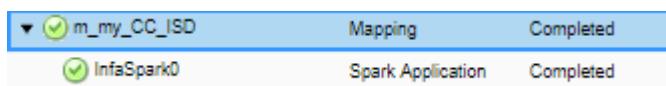
For the mapping to validate, we need to set it in **spark** mode:



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:

/user/infa/elevate/lab08									Go!
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name		
drwxr-xr-x	infa	supergroup	0 B	Wed May 02 11:26:40 +0000 2018	0	0 B	input		
-rw-r--r--	infa	supergroup	612.84 KB	Wed May 02 11:43:27 +0000 2018	3	128 MB	my_cc_isd_file.txt-m-00000		

Run Data Viewer on target object, you should see:

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

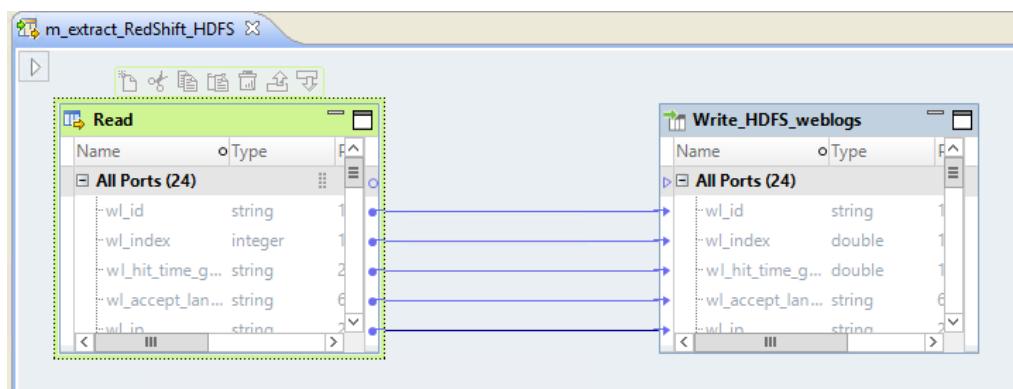
11.1 Purpose

Show AWS RedShift & S3 integration in native & push down

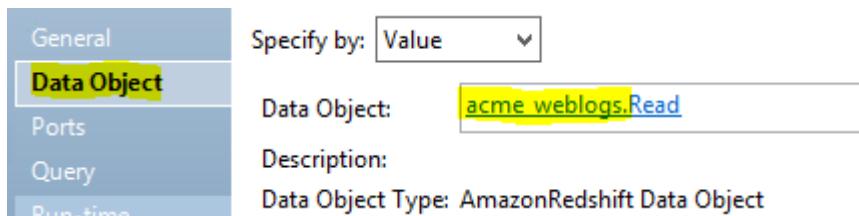
11.2 Review Lab Content

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

Open the passthrough mapping **m_extract_RedShift_HDFS**

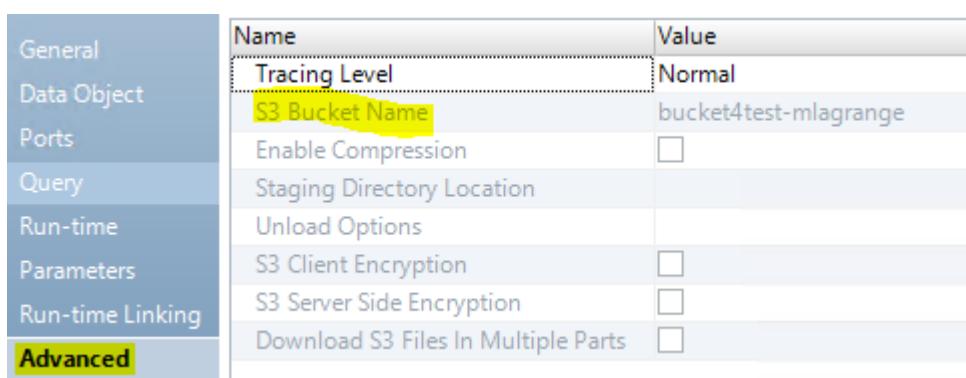


Select the **Read** object and go to Data Object



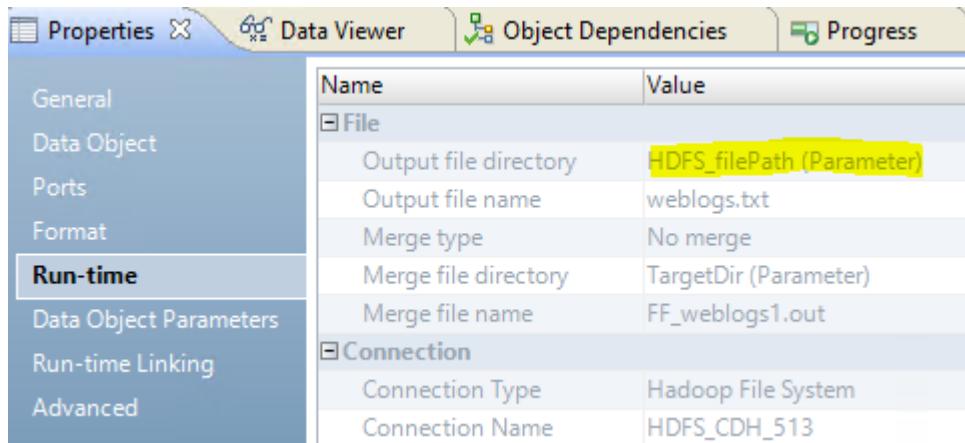
The source is a Redshift table called acme_weblogs

Go to Advanced



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

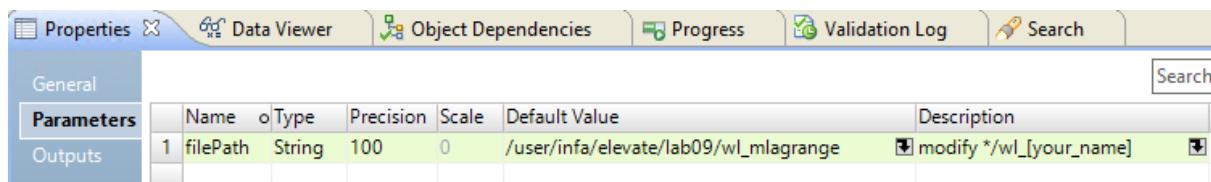
Select the **Write_HDFS_weblogs** target object and go to **Run-time**



Name	Value
File	
Output file directory	HDFS_filePath (Parameter)
Output file name	weblogs.txt
Merge type	No merge
Merge file directory	TargetDir (Parameter)
Merge file name	FF_weblogs1.out
Connection	
Connection Type	Hadoop File System
Connection Name	HDFS_CDH_513

It uses a parameter for HDFS file path

Go the **mapping parameters**

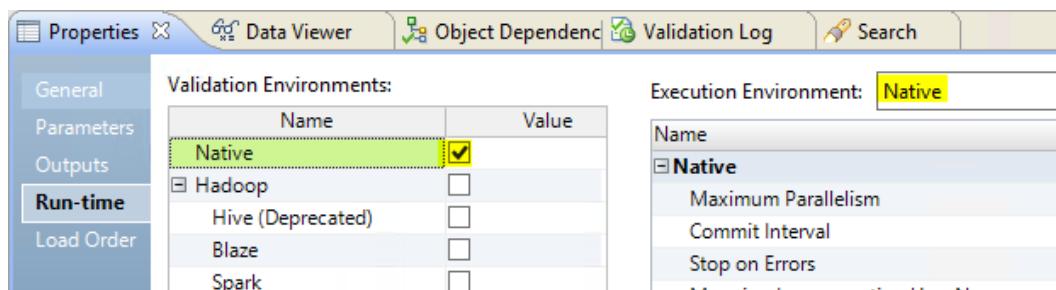


Name	Type	Precision	Scale	Default Value	Description
1 filePath	String	100	0	/user/infa/elevate/lab09/wl_mlagrange	modify */wl_[your_name]

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

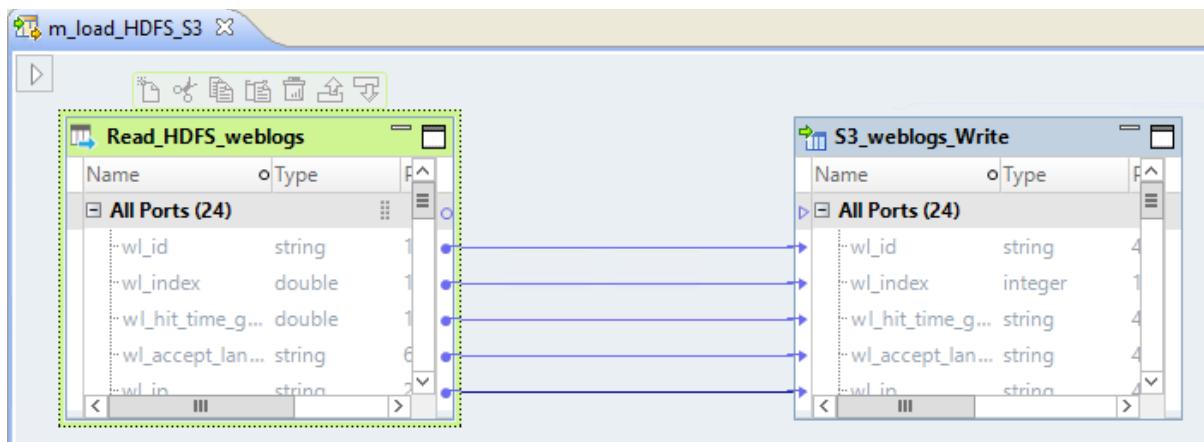


Name	Value
Native	<input checked="" type="checkbox"/>
Hadoop	<input type="checkbox"/>
Hive (Deprecated)	<input type="checkbox"/>
Blaze	<input type="checkbox"/>
Spark	<input type="checkbox"/>

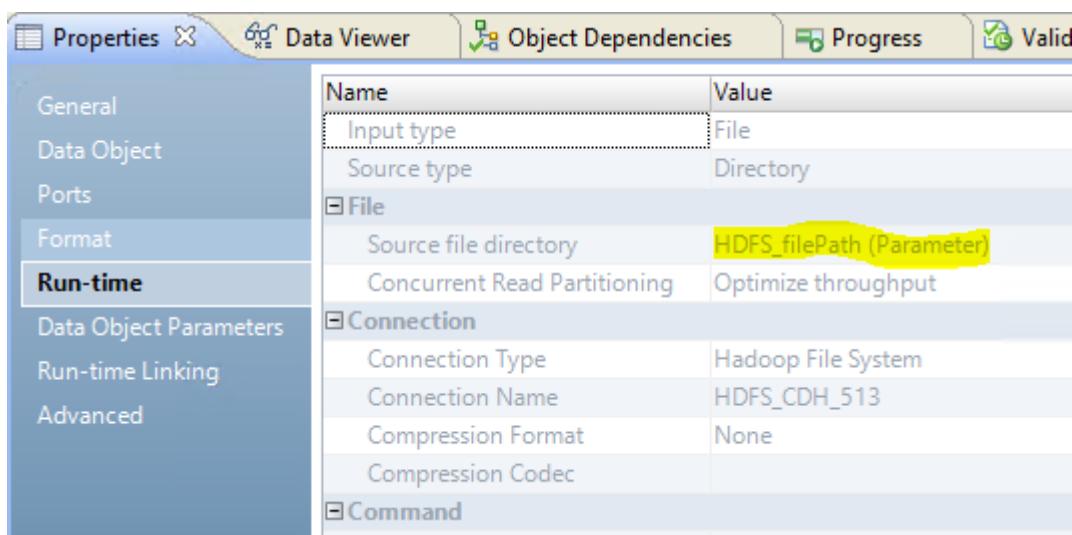
We will run this mapping in native mode.

Save the mapping

Open second mapping **m_load_HDFS_S3**



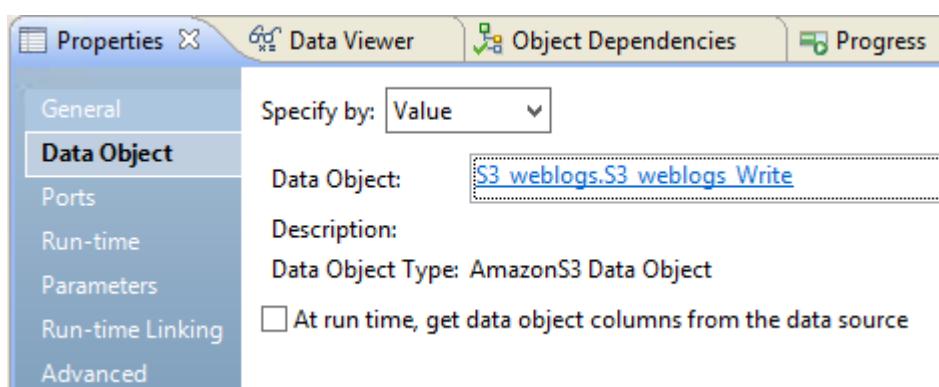
Select the **Read_HDFS_weblogs** source Object and go to **Run-time** properties



Name	Value
Input type	File
Source type	Directory
Source file directory	HDFS_filePath (Parameter)
Concurrent Read Partitioning	Optimize throughput
Connection Type	Hadoop File System
Connection Name	HDFS_CDH_513
Compression Format	None
Compression Codec	

HDFS file path is using a parameter

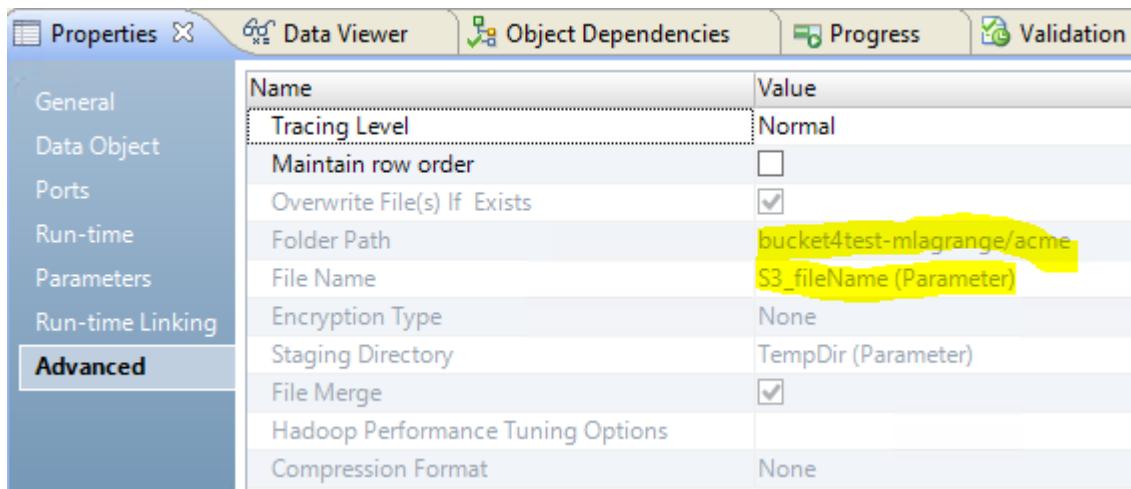
Select the **S3_weblogs_Write** target Object and go to **Data Object**



Specify by:	Value
Data Object:	S3 weblogs.S3 weblogs Write
Description:	Data Object Type: AmazonS3 Data Object
<input type="checkbox"/> At run time, get data object columns from the data source	

We will write to AWS S3 Object

Go to the **Advanced** properties

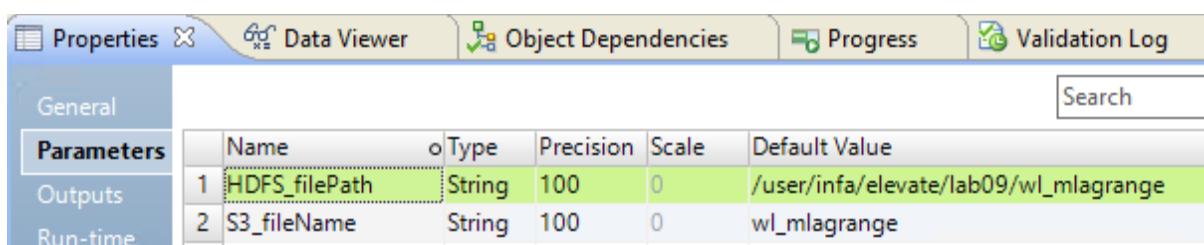


The screenshot shows the Informatica Properties window with the 'Advanced' tab selected. The configuration includes:

Name	Value
Tracing Level	Normal
Maintain row order	<input type="checkbox"/>
Overwrite File(s) If Exists	<input checked="" type="checkbox"/>
Folder Path	bucket4test-mlagrange/acme
File Name	S3_fileName (Parameter)
Encryption Type	None
Staging Directory	TempDir (Parameter)
File Merge	<input checked="" type="checkbox"/>
Hadoop Performance Tuning Options	
Compression Format	None

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

Go to the **mapping parameters**



The screenshot shows the Informatica Properties window with the 'Parameters' tab selected. It lists two parameters:

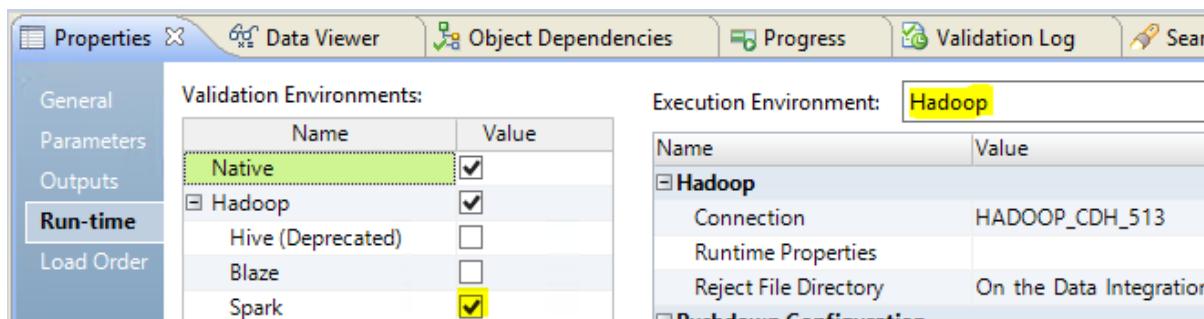
Name	Type	Precision	Scale	Default Value
1 HDFS_filePath	String	100	0	/user/infa/elevate/lab09/wl_mlagrange
2 S3_fileName	String	100	0	wl_mlagrange

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



The screenshot shows the Informatica Properties window with the 'Run-time' tab selected. It displays the Validation Environments and Execution Environment sections:

Validation Environments:

Name	Value
Native	<input checked="" type="checkbox"/>
Hadoop	<input checked="" type="checkbox"/>
Hive (Deprecated)	<input type="checkbox"/>
Blaze	<input type="checkbox"/>
Spark	<input checked="" type="checkbox"/>

Execution Environment: Hadoop

Name	Value
Connection	HADOOP_CDH_513
Runtime Properties	
Reject File Directory	On the Data Integration

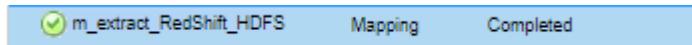
We will run this mapping in **spark** mode.

Save the mapping

11.3 Run Lab

Run the mapping **m_extract_RedShift_HDFS**

You should see in [Informatica Monitoring Console](#):

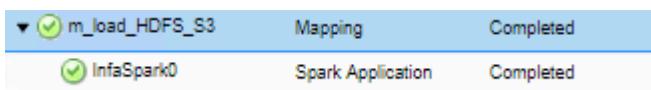


Use [HDFS Web Browser](#) and navigate to /user/infa/elevate/lab09, you should see:

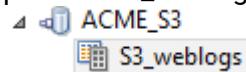


Run the mapping **m_load_HDFS_S3**

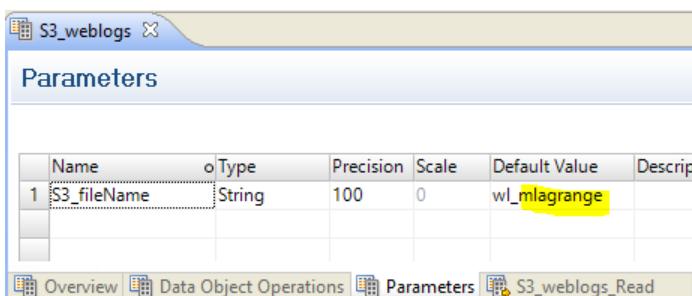
You should see in [Informatica Monitoring Console](#):



Open the S3_weblogs object



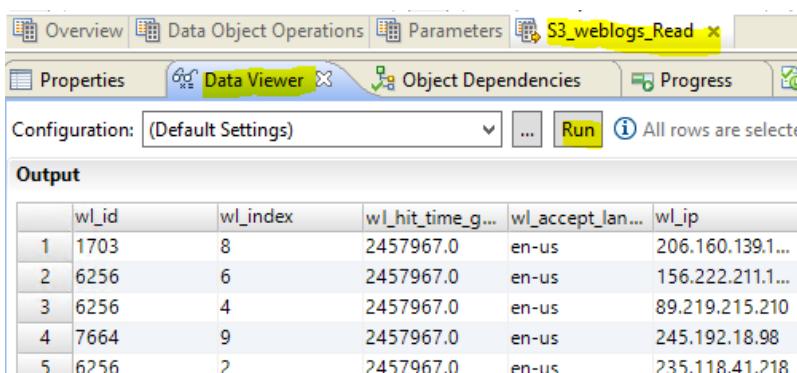
Go to parameters



Name	Type	Precision	Scale	Default Value	Description
S3_fileName	String	100	0	wl_mlagrange	

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

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



wl_id	wl_index	wl_hit_time_g...	wl_accept_lan...	wl_ip
1	1703	8	2457967.0	en-us 206.160.139.1...
2	6256	6	2457967.0	en-us 156.222.211.1...
3	6256	4	2457967.0	en-us 89.219.215.210
4	7664	9	2457967.0	en-us 245.192.18.98
5	6256	2	2457967.0	en-us 235.118.41.218

12. Lab10a – BDS weblogs real-time processing

12.1 Purpose

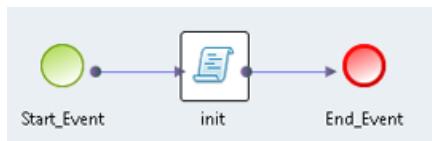
Real-time collection and processing of weblogs data.

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

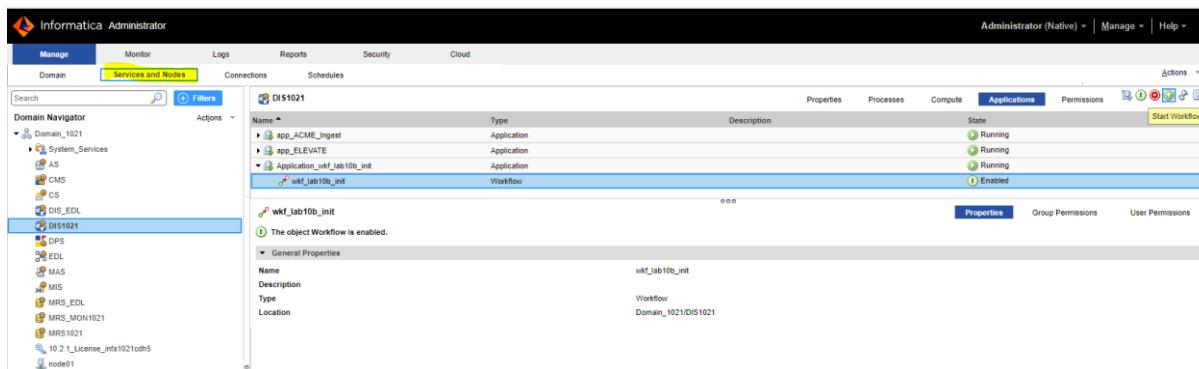
12.2 Initialize Lab

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



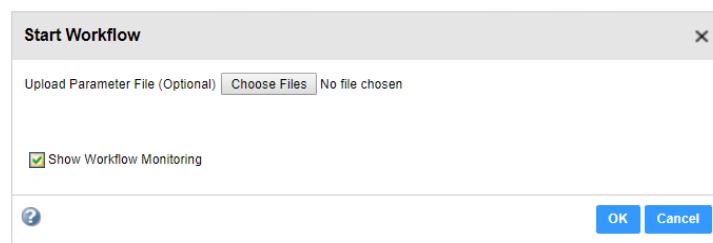
The **init** task command will reset target HBASE tables, create the required Kafka topics and clean Elastic Search indexes (for Kibana dashboarding).

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):

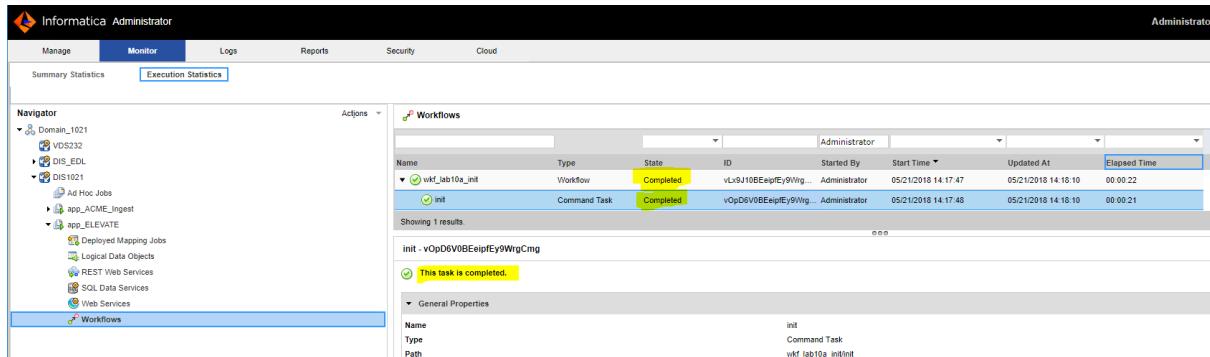


Name	Type	Description	State
app_ACME_Ingest	Application		Running
app_ELEVATE	Application		Running
Application_wkf_lab10b_init	Application		Running
wkf_lab10b_init	Workflow		Enabled

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



You should see:



Name	Type	State	ID	Started By	Start Time	Updated At	Elapsed Time
vlf_lab10a_init	Workflow	Completed	vLx910BEeipEy9Wrg...	Administrator	05/21/2018 14:17:47	05/21/2018 14:18:10	00:00:22

Showing 1 results.

init - vOpD6V0BEeipEy9WrgCmg

This task is completed.

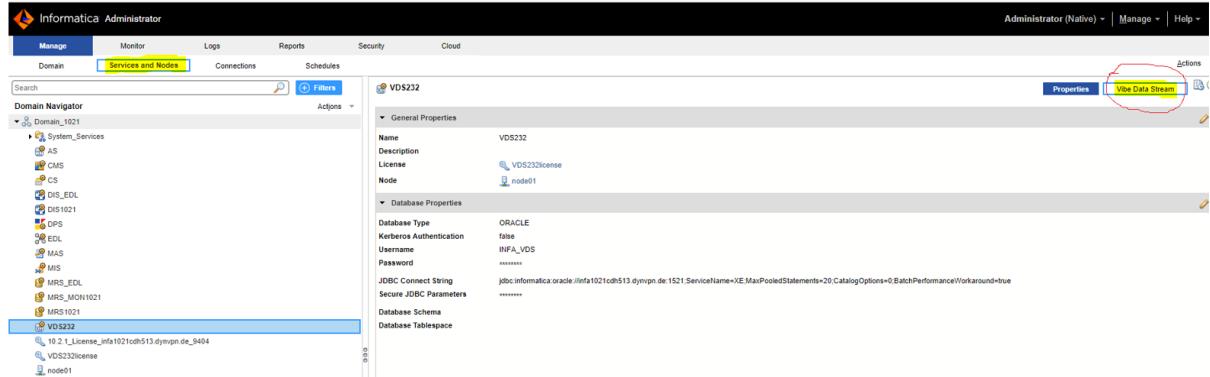
General Properties

Name	init
Type	Command Task
Path	vlf_lab10a_init\init

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):



VDS232

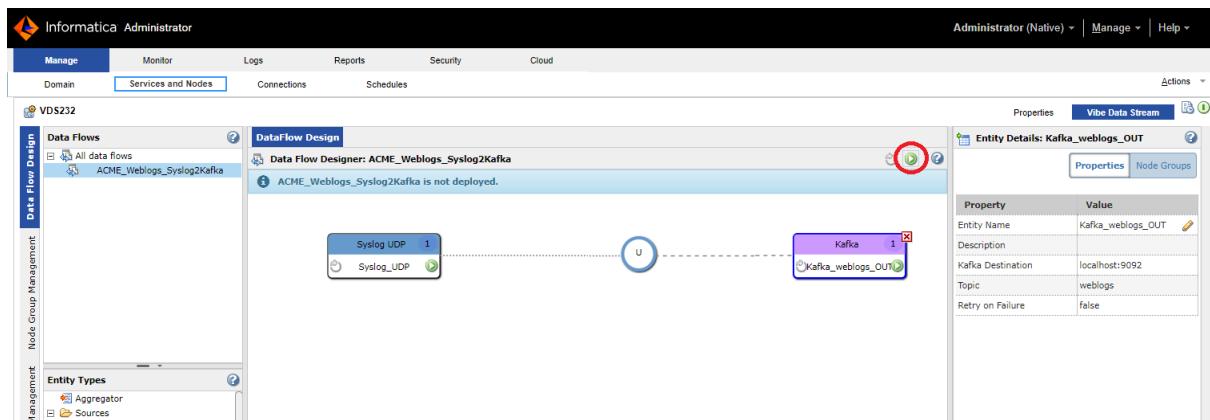
General Properties

- Name: VDS232
- Description: VDS232license
- Node: mode01

Database Properties

- Database Type: ORACLE
- Kerberos Authentication: false
- Username: INF_A_VDS
- Password: *****
- JDBC Connect String: jdbc:informatica.oracle://ifa1021cdh513.dympn.de:1521/ServiceName=XE.MaxPooledStatements=20.CatalogOptions=0.BatchPerformance=0.Orakround=True
- Secure JDBC Parameters: *****
- Database Schema: *****
- Database Tablespace: *****

You should see:



Data Flow Design

Entity Details: Kafka_weblogs_OUT

Property	Value
Entity Name	Kafka_weblogs_OUT
Description	
Kafka Destination	localhost:9092
Topic	weblogs
Retry on Failure	false

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:

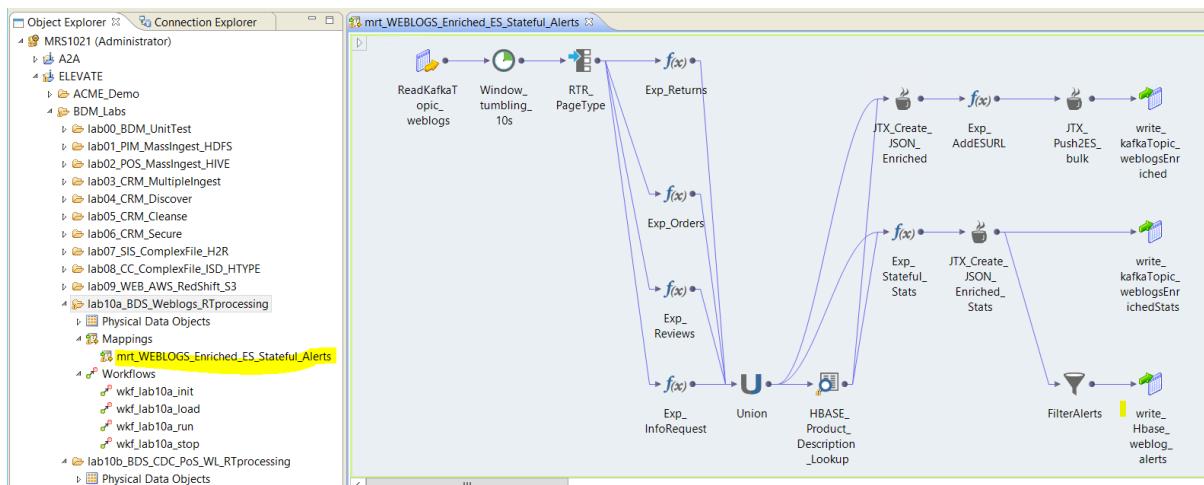
Then navigate to **Monitor -> Execution Statistics**:

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

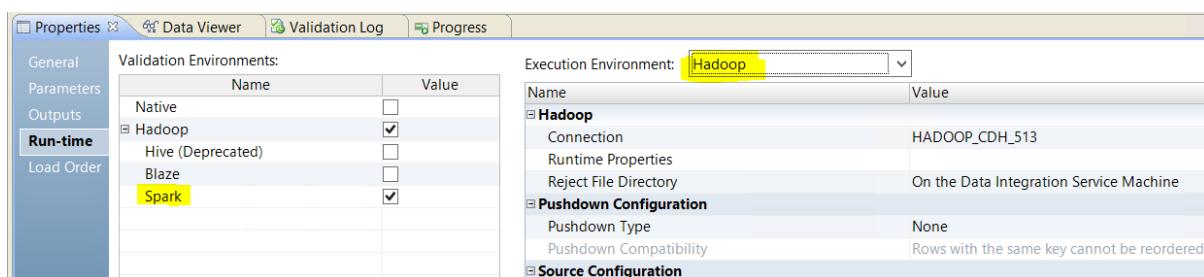
12.4 Review Lab Content

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:

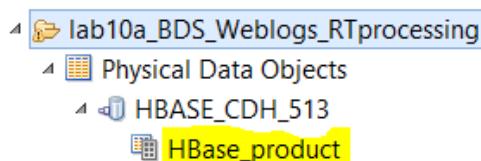


From the Run-time properties, you can see that this mapping is set for **Hadoop-Spark** Streaming execution mode:



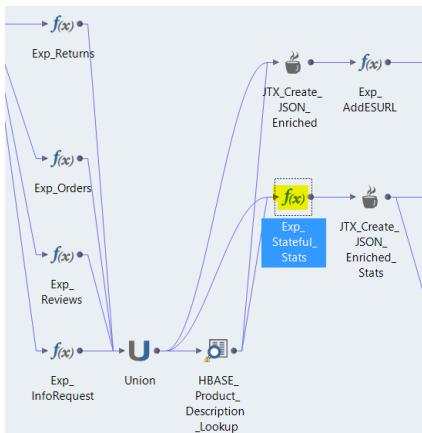
Which means that BDS will automatically translate it into Scala code for being executed by the Spark Streaming real-time processing engine.

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

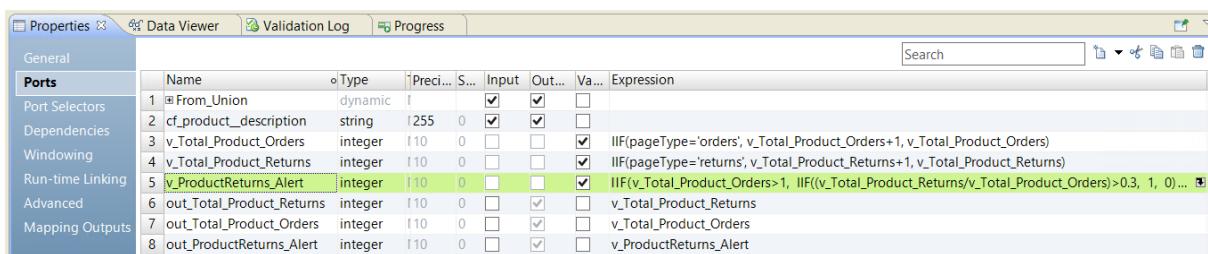


The **source 'HBase_product'** is a look-up table used to fetch in real-time the description of weblog product IDs (the raw input weblog stream does not contain descriptions for products).

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



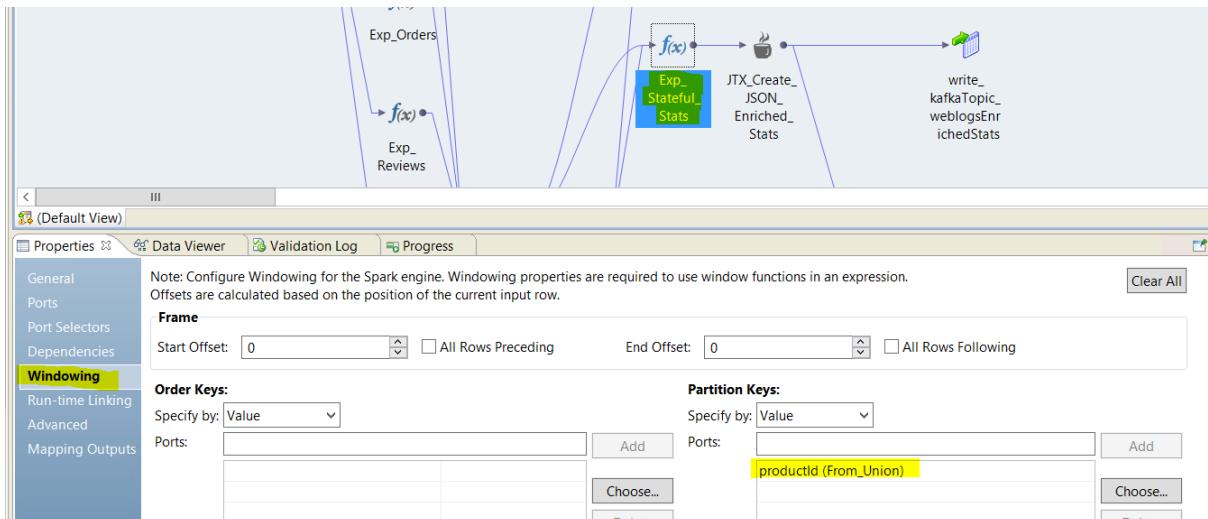
Select the '**Properties->Ports**' view tab:



Name	Type	Precision	Scale	Input	Output	Value	Expression
From_Union	dynamic	1255	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
cf_product_description	string	1255	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	IIF(pageType='orders', v_Total_Product_Orders+1, v_Total_Product_Orders)
v_Total_Product_Orders	integer	110	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	IIF(pageType='returns', v_Total_Product_Returns+1, v_Total_Product_Returns)
v_ProductReturns_Alert	integer	110	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	IIF(v_Total_Product_Orders>1, IIF(v_Total_Product_Returns/v_Total_Product_Orders>0.3, 1, 0)...) <input style="width: 20px; height: 15px;" type="button" value="..."/>
out_Total_Product_Returns	integer	110	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	v_Total_Product_Returns
out_Total_Product_Orders	integer	110	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	v_Total_Product_Orders
out_ProductReturns_Alert	integer	110	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	v_ProductReturns_Alert

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).

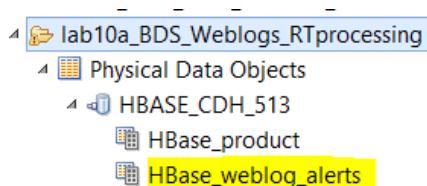
Observe the **Windowing** of the stateful expression:



The windowing configuration is set up for the 'v_ProductReturns_Alert' stateful variable. The 'Frame' section specifies 'Start Offset: 0' and 'End Offset: 0'. The 'Order Keys' section uses 'Value' for both 'Specify by:' and 'Ports:' fields. The 'Partition Keys' section also uses 'Value' for both 'Specify by:' and 'Ports:' fields, with 'productId (From_Union)' highlighted in yellow.

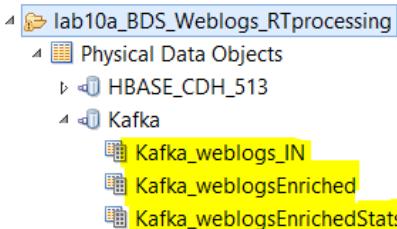
This is to group the state of variables by product id.

Observe the **target HBase Data Object**:



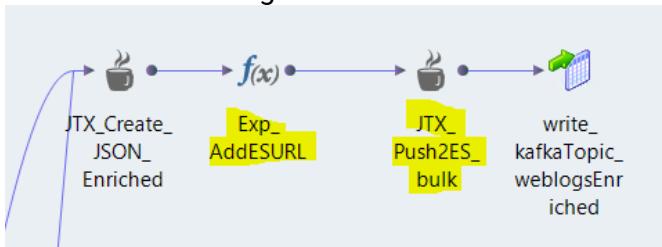
The target '**HBase_weblog_alerts**' Data Object describes a table used to store the weblogs related alerts generated by the mapping.

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).

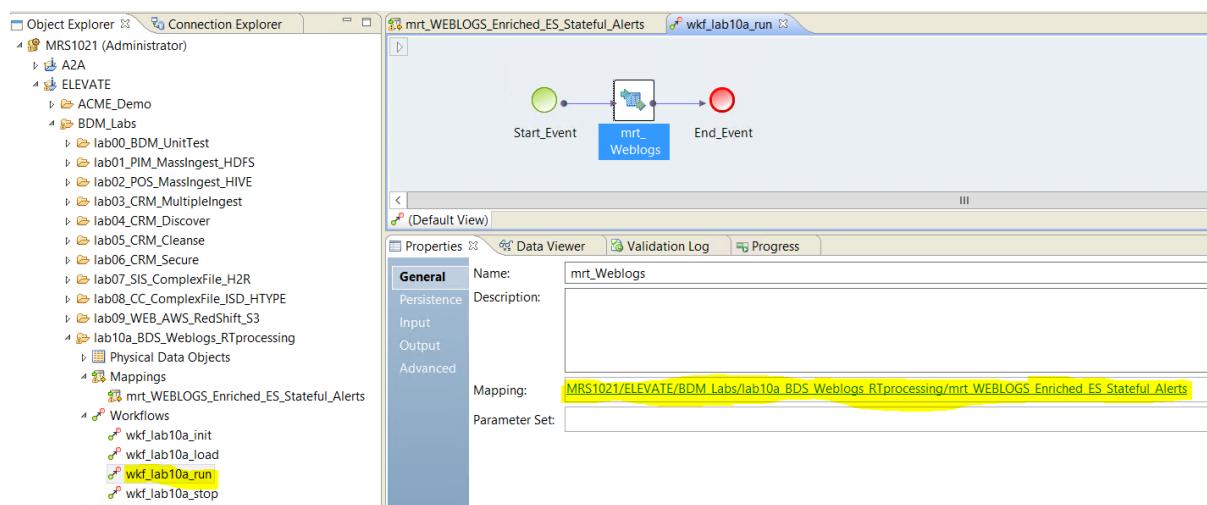
The enriched weblogs will also be indexed in real-time in **Elastic Search**:



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

12.5 Run the Lab

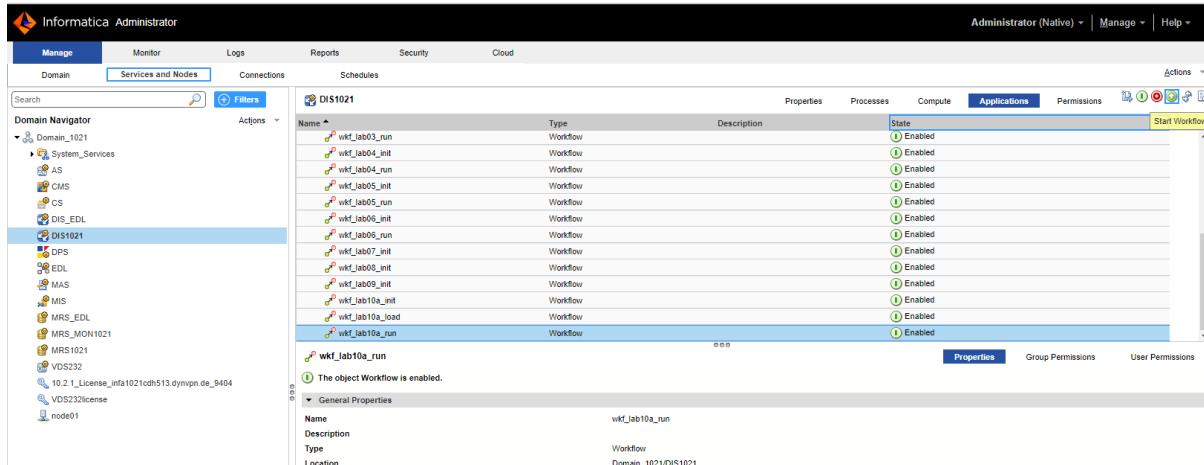
Workflow **wkf_lab10a_run** runs the **mrt_WEBLOGS_Enriched_ES_Stateful_Alerts** mapping.



The screenshot shows the 'wfk_lab10a_run' workflow in the Workflow view. The main pane displays a simple state machine with 'Start_Event', a 'mrt_Weblogs' state, and 'End_Event'. The properties pane for the 'mrt_Weblogs' state shows the following details:

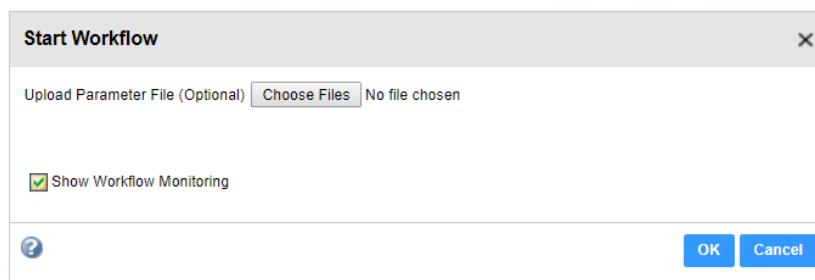
- Name:** mrt_Weblogs
- Description:** (empty)
- Mapping:** MRS1021/ELEVATE/BDM Labs/lab10a_BDS_Weblogs_RTprocessing/mrt_WEBLOGS_Enriched_ES_Stateful_Alerts (highlighted with a yellow box)
- Parameter Set:** (empty)

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):

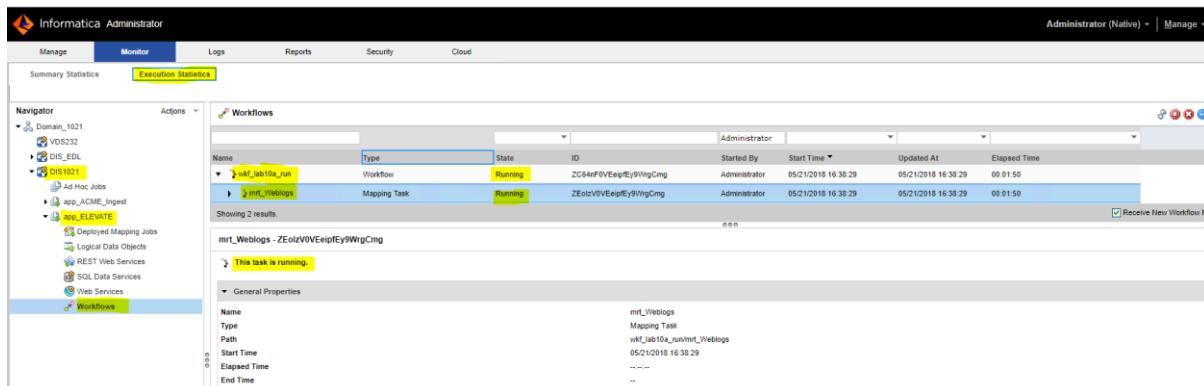


The screenshot shows the Informatica Administrator interface. The left sidebar has a 'Domain Navigator' section with various nodes like Domain_1021, AS, CMS, CS, DIS_EDL, DPS, EDL, MAS, MIS, MRS_EDL, MRS_MON1021, MRS_1021, VDS322, and node01. The main content area is titled 'DIS1021' and lists several workflows. One workflow, 'wkf_lab10a_run', is highlighted. At the bottom, there are tabs for 'Properties', 'Group Permissions', and 'User Permissions'.

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:



The screenshot shows the 'Execution Statistics' tab for the 'DIS1021' domain. It lists various workflows and tasks. One task, 'mrt_Weblogs - ZEolzV0V6EipEipEy9WrgCmg', is highlighted with a yellow background and labeled 'This task is running.'. The 'General Properties' panel on the right provides detailed information about this specific task.

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



The screenshot shows the 'RUNNING Applications' page in the Hadoop YARN Monitor. It lists various applications with their details. One application, 'mrt_WEBLOGS_Enriched_ES_Stateful_Aggregations_SPARK', is selected and highlighted with a yellow background. The 'ApplicationMaster' link is also highlighted in yellow.

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:

Apache Spark 2.1.0

- Jobs
- Stages
- Storage
- Environment
- Executors
- SQL
- Streaming**

User: yarn
Total Uptime: 7.0 min
Scheduling Mode: FIFO
Completed Jobs: 120

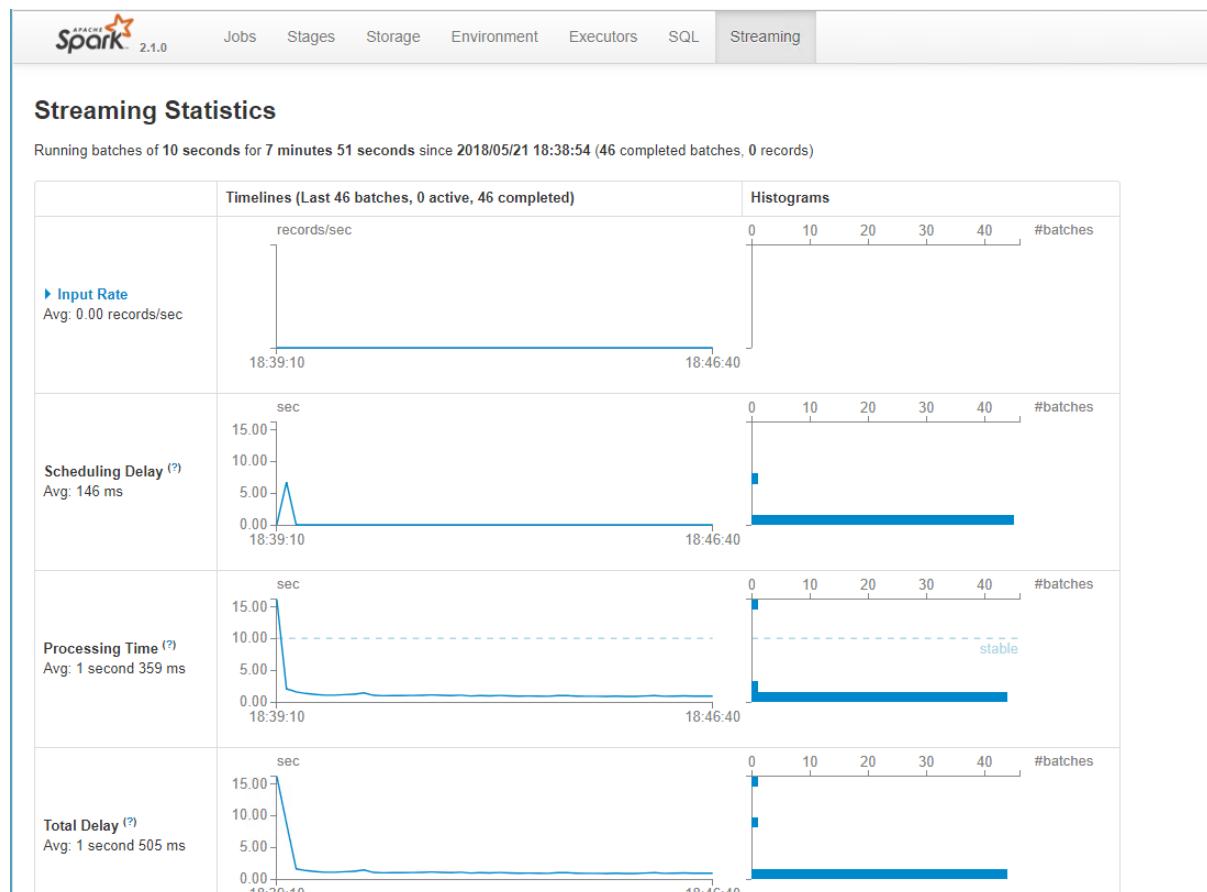
Event Timeline

Completed Jobs (120)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
119	Streaming job from [output operation 2, batch time 18:45:30] sql at InfaSpark0 scala:T36	2018/05/21 18:45:31	0.2 s	1/1 (9 skipped)	4/4 (216 skipped)
118	Streaming job from [output operation 1, batch time 18:45:30]	2018/05/21 18:45:31	0.3 s	2/2 (8 skipped)	28/28 (192 skipped)

2 Pages. Jump to 1 Show 100 Items in a page Go

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:

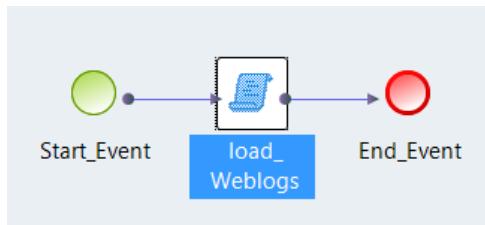


Note: It may take couple of minutes for the Streaming tab to appear.

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

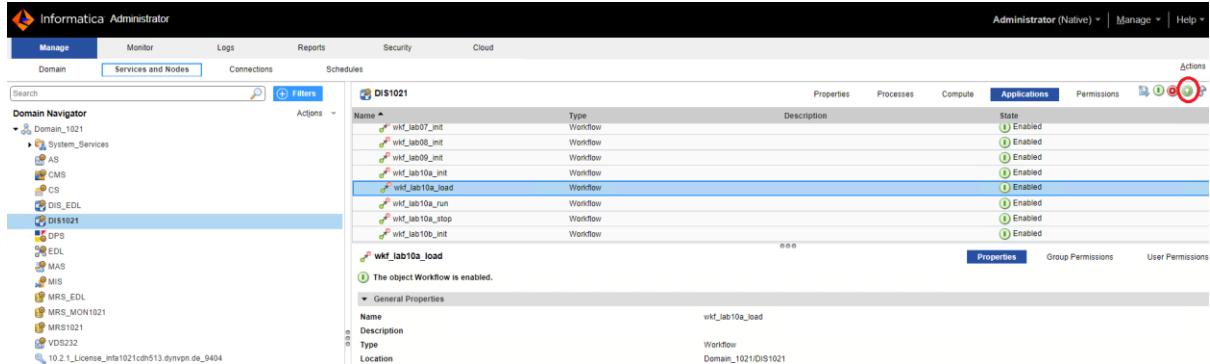
12.6 Load Real-time Mapping Data

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



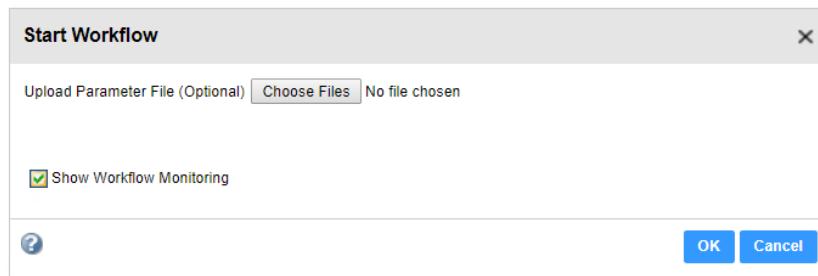
It will start a Python script that will produce a real-time data stream of raw weblogs data over a Syslog protocol (for EDS input); the EDS data flow will then produce a Kafka output stream on the '**weblogs**' topic (for BDS input).

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

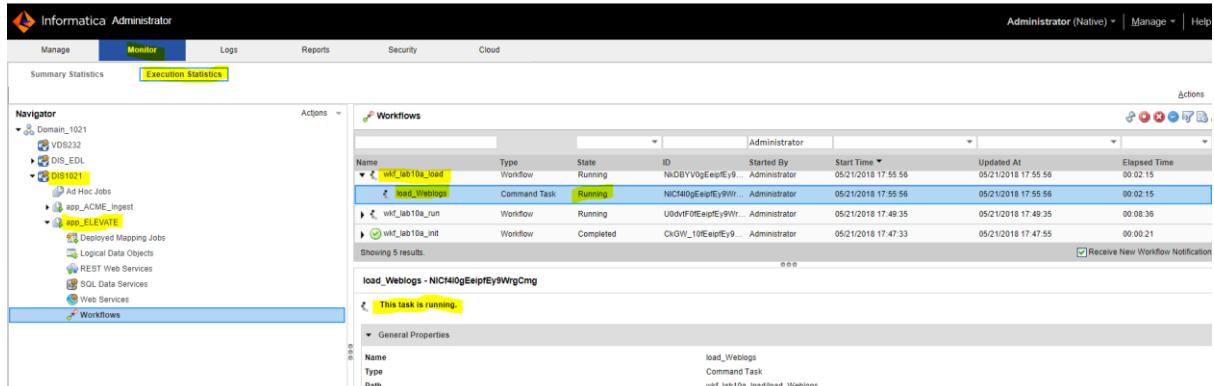


The screenshot shows the Informatica Administrator interface. The top navigation bar includes Manage, Monitor, Logs, Reports, Security, and Cloud. The left sidebar is the Domain Navigator, showing various domains like Domain_1021, DIS1021, and VDS232. The main content area is titled 'DIS1021' and shows a list of workflows under the 'Applications' tab. One workflow, 'wkf_lab10a_load', is highlighted. A tooltip indicates it is enabled. Below the list is a 'General Properties' section for 'wkf_lab10a_load'. The bottom right of the screen has tabs for Properties, Group Permissions, and User Permissions.

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



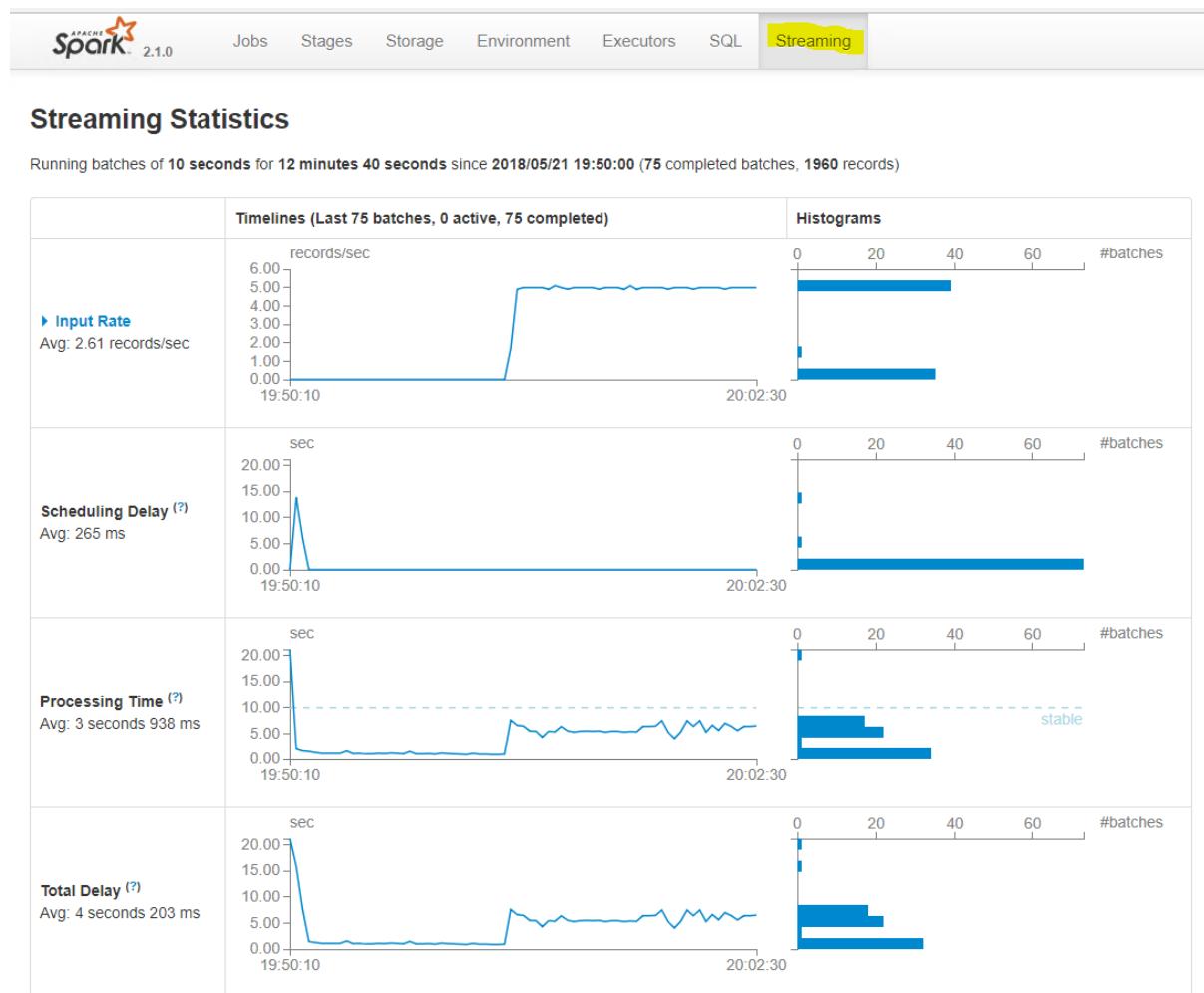
Under **Monitor->Execution Statistics->DIS1021->app_ELEVATE->workflows**, you should see:



The screenshot shows the Informatica Administrator interface with the 'Monitor' tab selected. In the 'Execution Statistics' section, the 'Workflows' table is displayed for the 'DIS1021' domain. One workflow, 'wkf_lab10a_load', is shown as 'Running'. Below the table, a detailed view of the 'load_Weblogs' command task is shown, indicating it is 'Running'. The 'General Properties' section for this task shows the name is 'load_Weblogs', type is 'Command Task', and path is 'wkf_lab10a_load/load_Weblogs'.

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

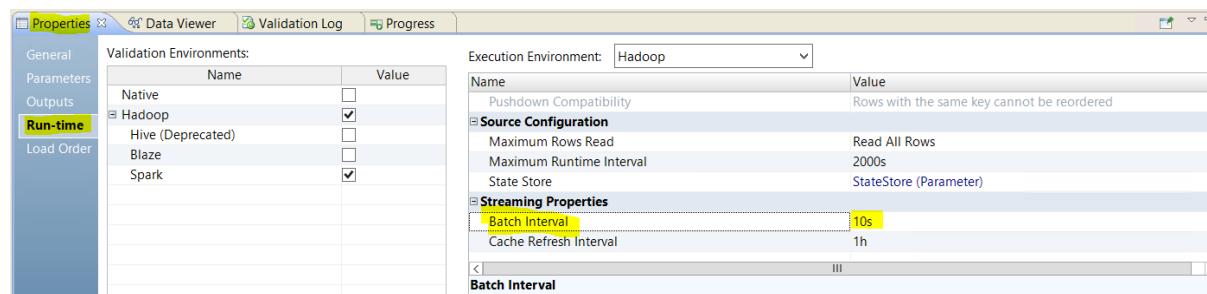
You can refresh the runtime statistics of the Spark Streaming Application associated to the BDS mapping by clicking the '**Streaming**' tab in the YARN application web UI:



Active Batches (0)

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'
```

```
[java@infal021cdh513 ~]$ 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-407f8d4dc06 column=cf_weblog_alerts:Alert, timestamp=1526918206244, value={"pageType":"orders", "productId":455250, "ProductDescription":"Brother Laserprinter HL-2130", "Total_Product_Returns":1, "Total_Product_Orders":2, "ProductReturnsAlertID":1}
249e336d-0d02-4227-a864-30f733592192 column=cf_weblog_alerts:Alert, timestamp=1526918235108, value={"pageType":"orders", "productId":414073, "ProductDescription":"Wacom Bamboo Pen & Touch", "Total_Product_Returns":1, "Total_Product_Orders":3, "ProductReturnsAlertID":1}
57230c64-1586-4f72-a6be-d4c71cc8789c column=cf_weblog_alerts:Alert, timestamp=1526918166372, value={"pageType":"orders", "productId":950349, "ProductDescription":"Zotac ZBOX AD02 Barebone", "Total_Product_Returns":1, "Total_Product_Orders":2, "ProductReturnsAlertID":1}
701dc920-cc76-4855-b48e-e9d30aa128b6 column=cf_weblog_alerts:Alert, timestamp=1526918235107, value={"pageType":"orders", "productId":102022, "ProductDescription":"Razer Goliathus Speed Extended XL", "Total_Product_Returns":1, "Total_Product_Orders":3, "ProductReturnsAlertID":1}
721d822f-90ef-469a-9094-2aed5fae64a5 column=cf_weblog_alerts:Alert, timestamp=1526918226146, value={"pageType":"orders", "productId":102022, "ProductDescription":"Razer Goliathus Speed Extended XL", "Total_Product_Returns":1, "Total_Product_Orders":2, "ProductReturnsAlertID":1}
8bbfc052-6ee5-4c47-8560-f161bc568da6 column=cf_weblog_alerts:Alert, timestamp=1526918318157, value={"pageType":"returns", "productId":620793, "ProductDescription":"SteelSeries QcK Diablo 3 Witch Doctor Edition", "Total_Product_Returns":1, "Total_Product_Orders":2, "ProductReturnsAlertID":1}
9a2190da-0942-42be-bd1e-9564ddff9f831 column=cf_weblog_alerts:Alert, timestamp=1526918116317, value={"pageType":"orders", "productId":253214, "ProductDescription":"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, "ProductDescription":"Zotac ZBOX AD02 Barebone", "Total_Product_Returns":1, "Total_Product_Orders":3, "ProductReturnsAlertID":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_Product_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_Orders":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
```

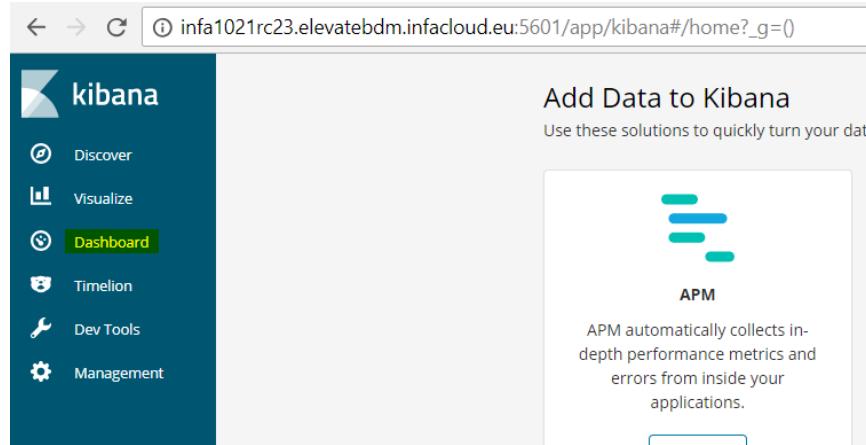
```
{"page_url":"http://www.acme.com/orders?orderid=5238&customerid=648&orderdate=20170107_18:01:00&product=350418","orderId":5238,"latitude":47.224044,"registered":2014-11-10,"page_event":7,"productReview":null,"pageType":"orders","date_time":2018-05-21T16:18:32.105194Z,"cookies_accepted":YES,"returnReason":null,"customerId":648,"accept_language":en-us,"id":5238,"domain":acme.com,"visit_num":24,"user_agent":Internet Explorer 11.1.0.0.4 WIN x64 7 Pro,"hit_time_gmt":2457761,"longitude":-123.10538,"pagename":main:item,"login_id":alvidrez@yahoo.com,"productId":350418,"ip":139.80.89.98,"last_hit_time_gmt":20170107180104,"paid_search":NO,"index":9,"referrer":W,"user_srvr":main,"visit_page_num":15,"java_enabled":YES,"first_hit_time_gmt":20170107175930,"productDescription":Kingston Compact Flash 8GB Elite Pro CF/8GB-S2} {"page_url":"http://www.acme.com/products?login_id=518;product_id=101595","orderId":null,"latitude":43.57219,"registered":2011-06-01,"page_event":9,"productReview":null,"pageType":info_requests,"date_time":2018-05-21T16:18:32.507046Z,"cookies_accepted":YES,"returnReason":null,"customerId":null,"accept_language":en-us,"id":560,"domain":acme.com,"visit_num":25,"user_agent":Google Chrome v42.0 WIN x64 8.1,"hit_time_gmt":2457762,"longitude":-85.76662,"pagename":main:product,"login_id":ian@yahoo.com,"productId":101595,"ip":184.183.65.201,"last_hit_time_gmt":20170128190402,"paid_search":NO,"index":7,"referrer":W,"user_srvr":main,"visit_page_num":36,"java_enabled":YES,"first_hit_time_gmt":20170108122814,"productDescription":ACME Sense Mousepad Glacier Blue"}
```

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:

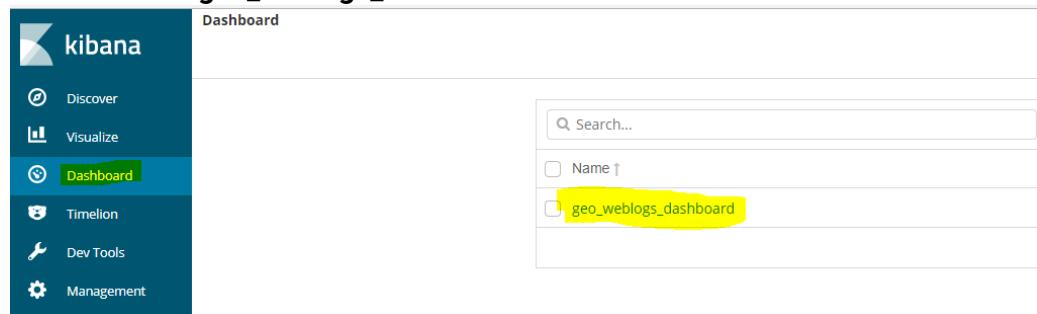
<http://infa1021rc23.elevatebdm.infacloud.eu:5601>

Then select Dashboard from the left side menu:



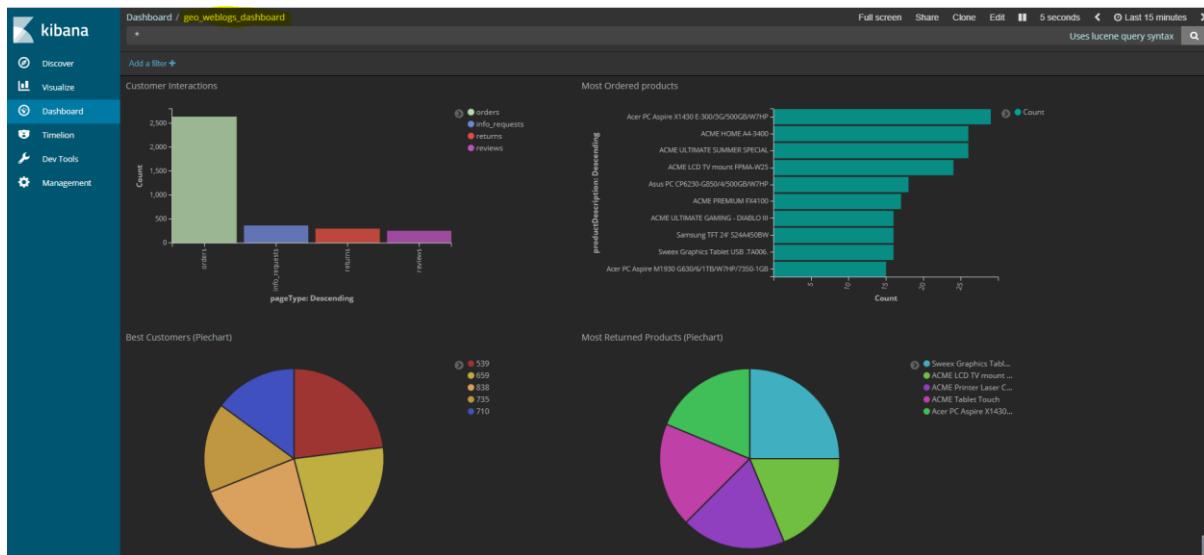
The screenshot shows the Kibana interface. On the left, there is a dark sidebar with the Kibana logo and several options: Discover, Visualize, **Dashboard** (which is highlighted in green), Timelion, Dev Tools, and Management. To the right, there is a light-colored panel titled "Add Data to Kibana" with the sub-section "APM". It contains a small icon of three horizontal bars, the word "APM", and a brief description: "APM automatically collects in-depth performance metrics and errors from inside your applications." Below this section, there is a search bar and a list of items, one of which is "geo_weblogs_dashboard", which is also highlighted in yellow.

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



This screenshot shows the same Kibana interface as the previous one, but the "geo_weblogs_dashboard" link in the sidebar has been clicked, and it is now highlighted in green. The rest of the interface remains the same, with the APM section visible on the right.

You should see:

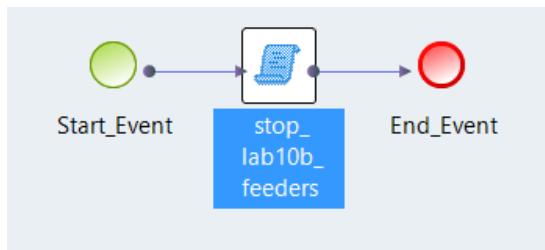


Take your time to observe and understand the dashboard's Visualizations.

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

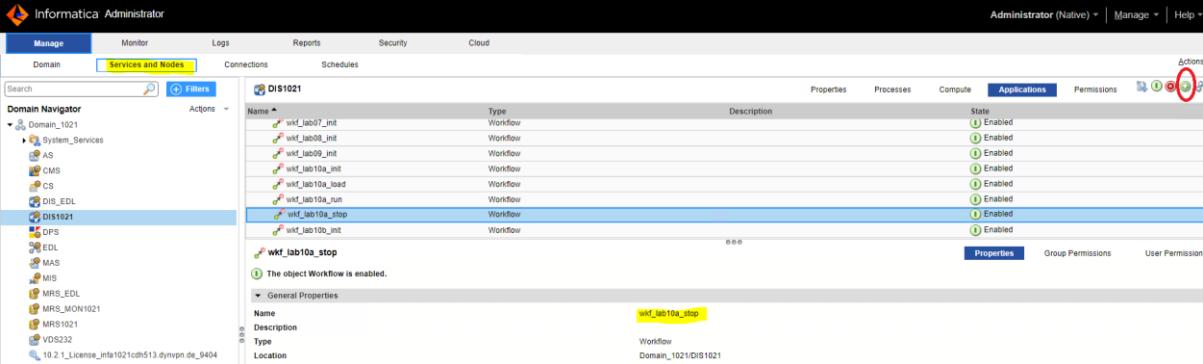
12.8 Stop the Lab

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



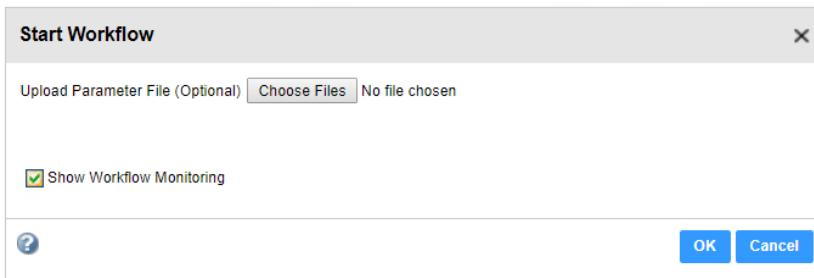
It will stop the same Python script that was started by the loading workflow.

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):

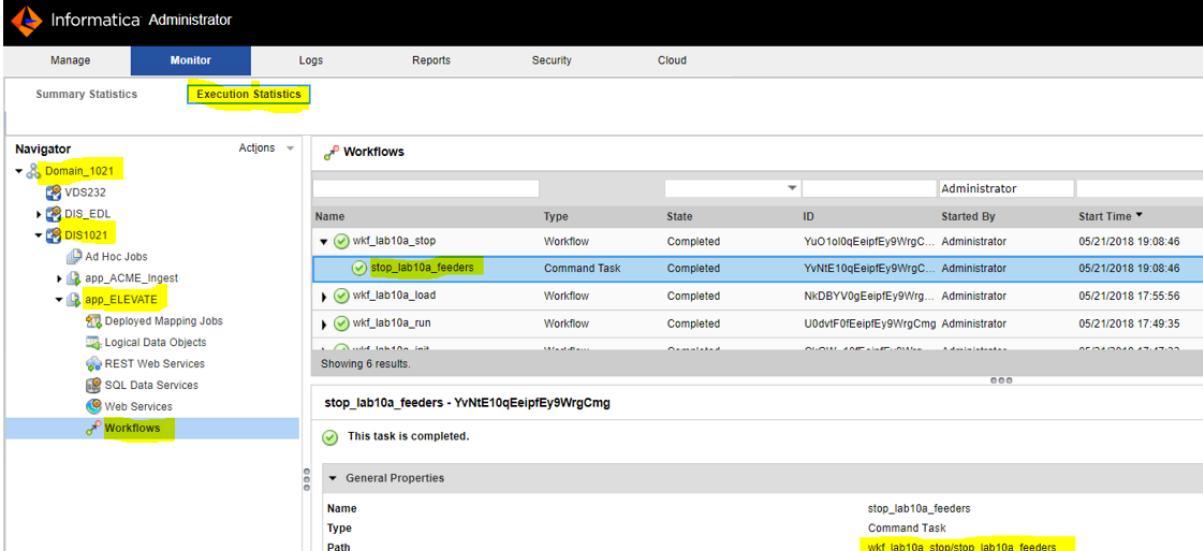


The screenshot shows the Informatica Administrator interface. The top navigation bar includes Manage, Monitor, Logs, Reports, Security, Cloud, and the current tab, Services and Notes. On the left, the Domain Navigator lists domains such as System_Services, AS, CMS, CS, DIS_EDL, and DIS1021. The main content area displays a table for the domain DIS1021, listing various workflows like 'wkf_lab07_init', 'wkf_lab08_init', etc. One workflow, 'wkf_lab10a_stop', is highlighted. At the bottom of the table, it says 'The object Workflow is enabled.' Below the table, there's a 'General Properties' section with fields for Name, Description, Type, and Location. The 'Actions' bar at the top right has several icons, with the start workflow icon circled in red.

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:



The screenshot shows the Informatica Monitor interface. The top navigation bar includes Manage, Monitor, Logs, Reports, Security, and Cloud, with the current tab being 'Execution Statistics'. On the left, the Navigator shows the domain DIS1021 and its sub-nodes like VDS232, DIS_EDL, and app_ELEVATE. The app_ELEVATE node is selected. The main content area displays a table titled 'Workflows' showing tasks like 'stop_lab10a_feeder', 'wkf_lab10a_load', and 'wkf_lab10a_run'. The 'stop_lab10a_feeder' task is highlighted. Below the table, a details pane shows the task is completed with the message 'This task is completed.' and a 'General Properties' section where the name is set to 'stop_lab10a_feeder'.

This concludes BDS lab10a.

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

13.1 Purpose

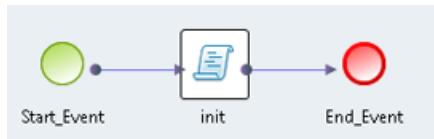
Real-time collection and processing of weblogs and CDC PoS data.

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

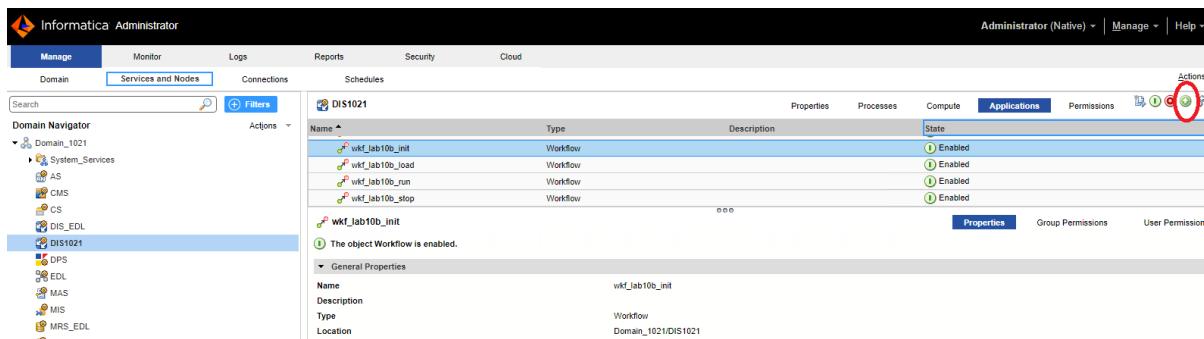
13.2 Initialize Lab

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):

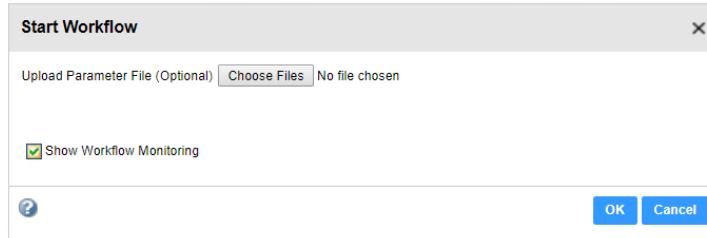


Name	Type	Description	State
wkf_lab10b_init	Workflow		Enabled
wkf_lab10b_load	Workflow		Enabled
wkf_lab10b_run	Workflow		Enabled
wkf_lab10b_stop	Workflow		Enabled
wkf_lab10b_init	Workflow		Enabled

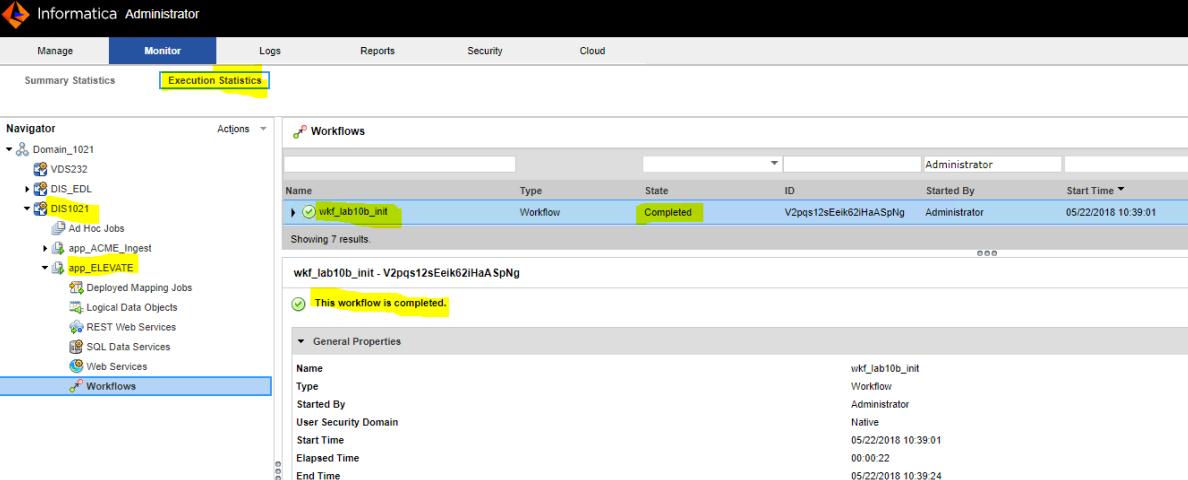
General Properties:

Name	wkf_lab10b_init
Description	
Type	Workflow
Location	Domain_1021/DIS1021

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



You should see:

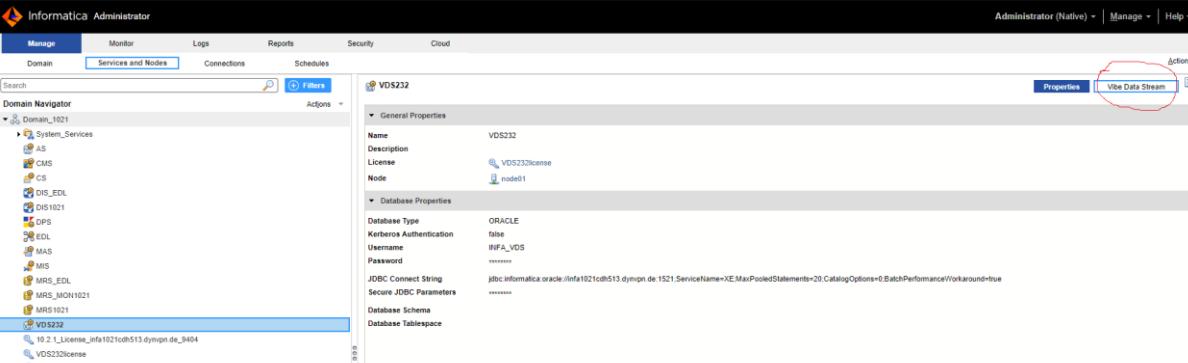


Name	Type	State	ID	Started By	Start Time
wkf_lab10b_init - V2pqsl2sEek62lHaAspNg	Workflow	Completed	V2pqsl2sEek62lHaAspNg	Administrator	05/22/2018 10:39:01

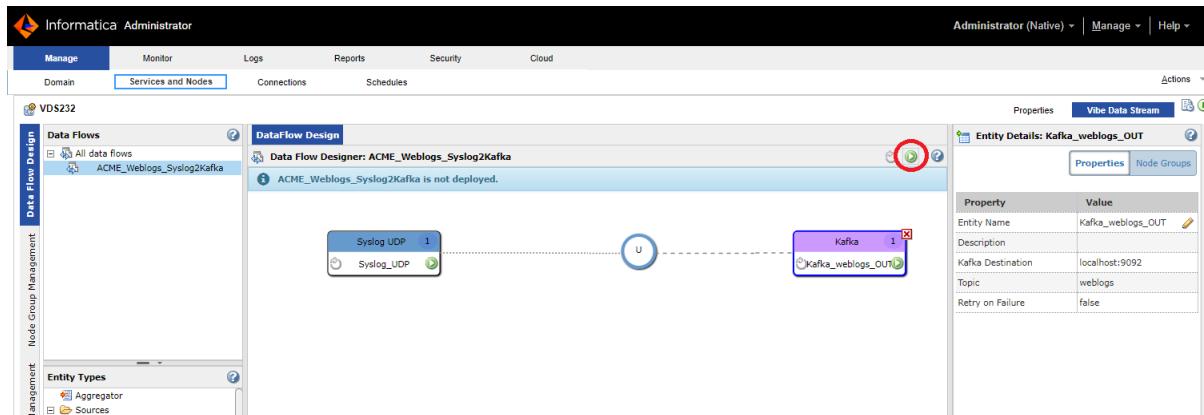
13.3 Start the Edge Data Streaming Flow

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:



The screenshot shows the Informatica Administrator interface with the 'Data Flow Design' tab selected. On the left, the 'Data Flows' tree shows 'All data flows' and 'ACME_Weblogs_Syslog2kafka'. The main panel displays a data flow diagram with a 'Syslog UDP' source node connected to a 'Kafka' destination node via a union node ('U'). A tooltip indicates 'ACME_Weblogs_Syslog2kafka is not deployed.' To the right, the 'Entity Details' pane shows properties for 'Kafka_weblogs_OUT': Entity Name (Kafka_weblogs_OUT), Description (localhost:9092), Topic (weblogs), and Retry on Failure (false). The 'Actions' bar at the top right includes 'Properties', 'Vibe Data Stream', and other icons.

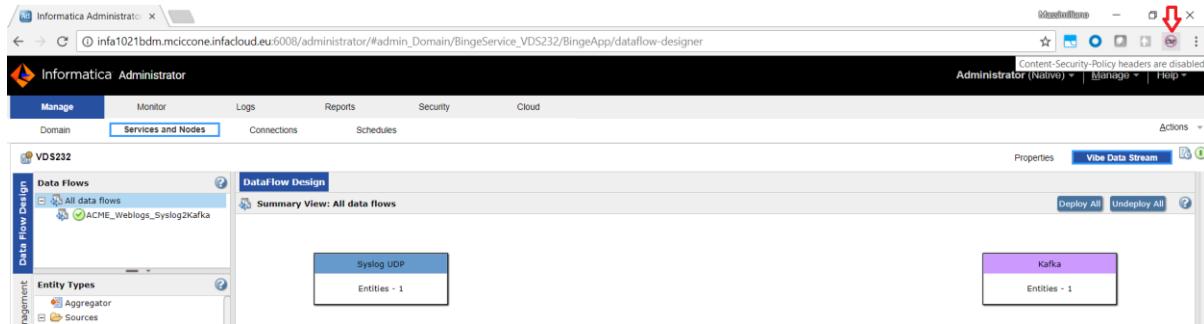
Deploy the EDS data flow (click top-right icon). You should see:



The screenshot shows the same Data Flow Designer interface after deployment. The status bar now displays 'ACME_Weblogs_Syslog2kafka is deployed.' and the 'Deploy' icon in the top right is now greyed out. The data flow diagram remains the same, with the nodes and connections visible.

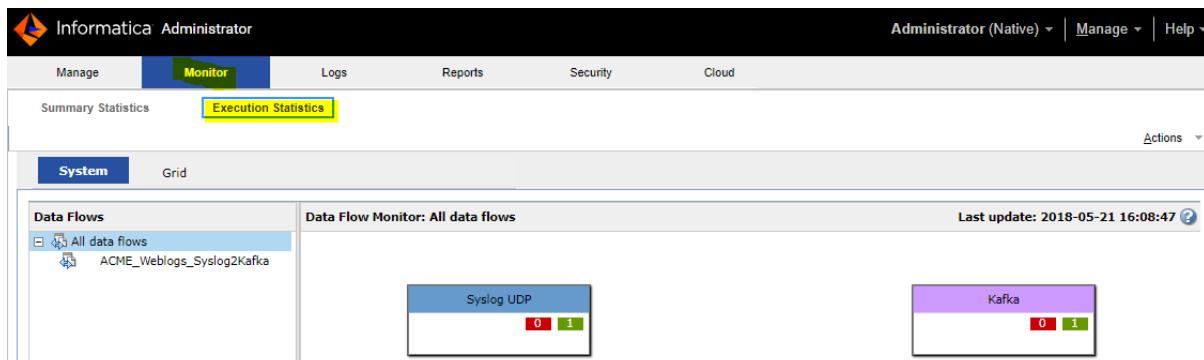
Now you can monitor the EDS data flow run-time statistics.

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:



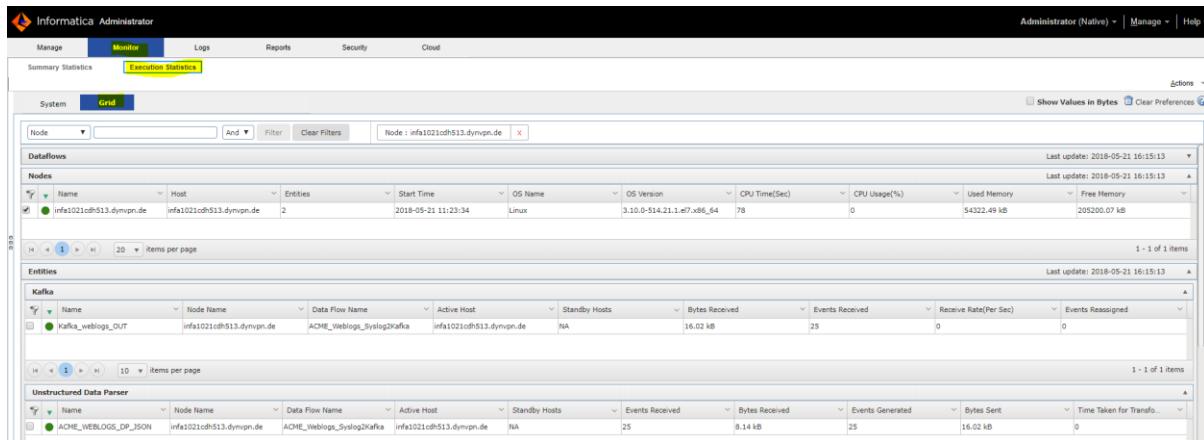
The screenshot shows a browser window with the URL 'infa1021bdm.miccone.infacloud.eu:6008/administrator/#admin_Domain/BingeService_VDS232/BingeApp/dataflow-designer'. The status bar at the bottom right of the browser window indicates 'Content-Security-Policy headers are disabled'. The browser toolbar includes icons for back, forward, search, and refresh, along with a red square icon.

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



The screenshot shows the Informatica Administrator interface with the 'Monitor' tab selected. Under 'Execution Statistics', the 'Grid' view is active. The left sidebar shows 'Data Flows' with 'All data flows' and 'ACME_Weblogs_Syslog2kafka'. The main panel displays two nodes: 'Syslog UDP' and 'Kafka', each with a grid showing real-time runtime statistics. The last update time is shown as 'Last update: 2018-05-21 16:08:47'.

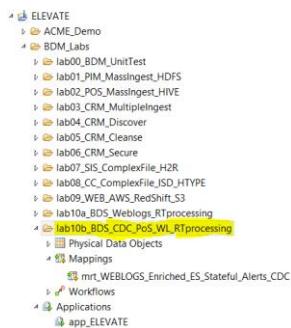
When selecting the **Grid** view, you can monitor EDS runtime statistics on collected real-time data:



The screenshot shows the Informatica Administrator interface with the 'Monitor' tab selected. The 'Execution Statistics' section displays data for a node named 'inf1021cdh513.dymvpn.de'. The table includes columns for Name, Host, Entities, Start Time, OS Name, OS Version, CPU Time(Sec), CPU Usage(%), Used Memory, and Free Memory. Below this, there are sections for Dataflows, Entities (Kafka), and Unstructured Data Parser.

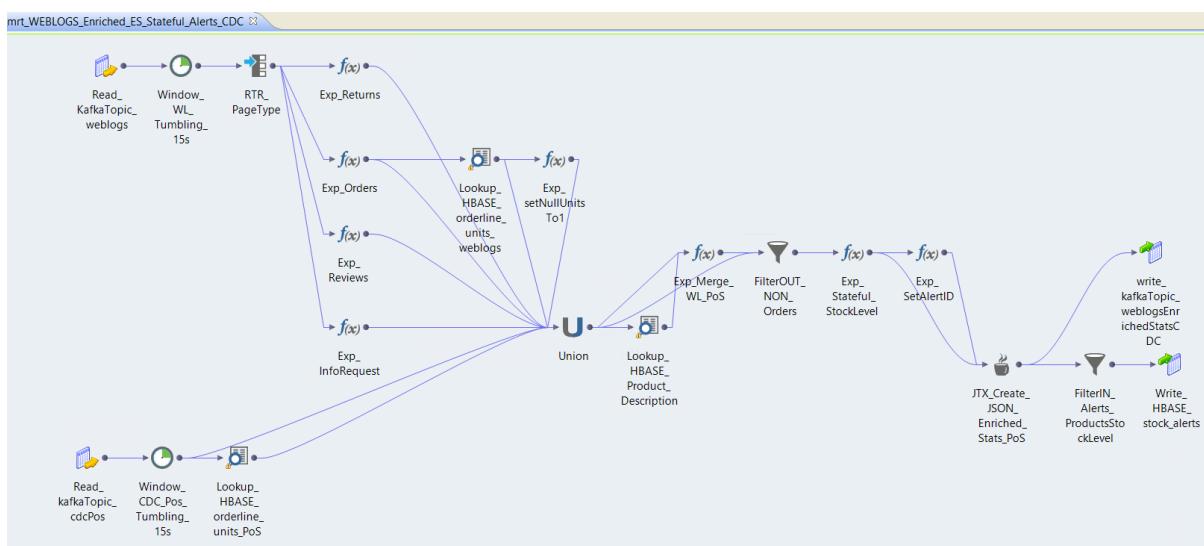
13.4 Review Lab Content

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

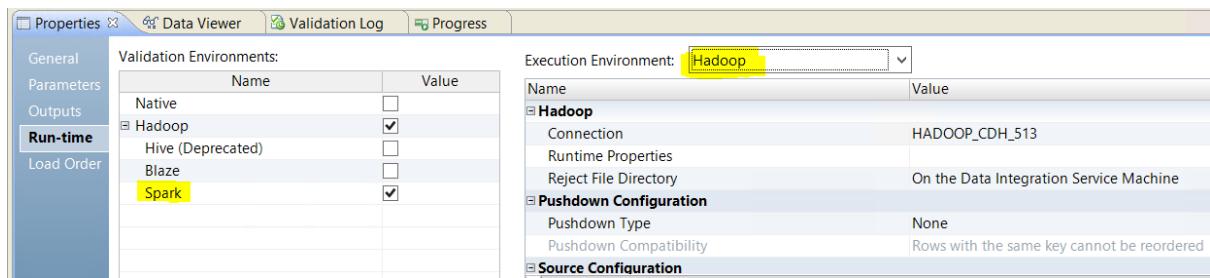


and open the **mrt_WEBLOGS_Enriched_ES_Stateful_Alerts_CDC** mapping.

You should see:

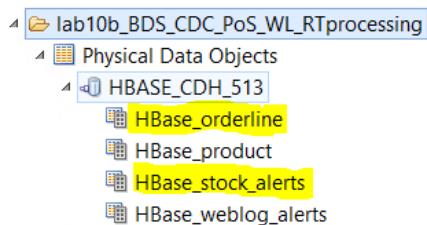


Also this mapping is set for **Hadoop-Spark Streaming** execution mode:



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

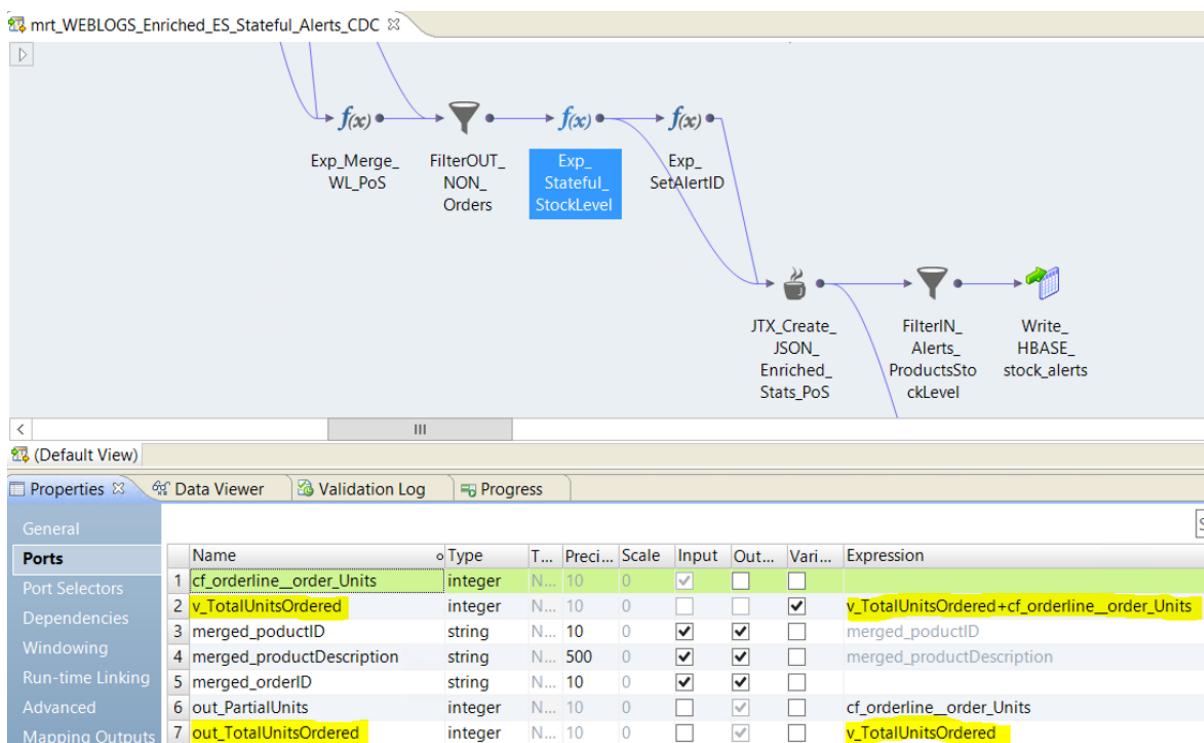
In Developer, observe the two additional HBase objects (with respect to Lab10a):



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:

Properties Data Viewer Validation Log Progress

General

Note: Configure Windowing for the Spark engine. Windowing properties are required to use window functions in an expression. Offsets are calculated based on the position of the current input row.

Ports

Port Selectors

Dependencies

Windowing (selected)

Run-time Linking Advanced Mapping Outputs

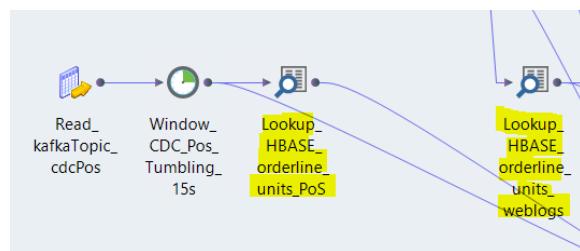
Frame

Start Offset: 0 All Rows Preceding End Offset: 0 All Rows Following

Order Keys: Specify by: Value Ports: Choose...

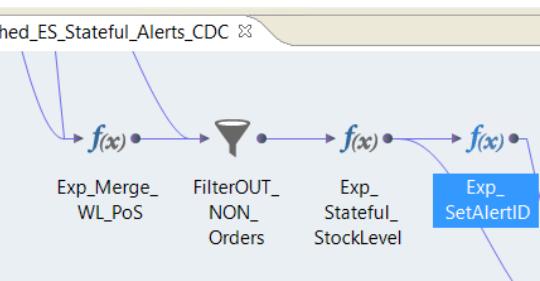
Partition Keys: Specify by: Value Ports: merged_productID Choose...

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):

mrt_WEBLOGS_Enriched_ES_Stateful_Alerts_CDC



```

graph LR
    A[Exp_Merge_WL_PoS] --> B[FilterOUT_NON_Orders]
    B --> C[Exp_Stateful_StockLevel]
    C --> D[Exp_SetAlertID]
  
```

The diagram shows a stateful processing flow. It starts with an enrichment step (Exp_Merge_WL_PoS), followed by a filter step (FilterOUT_NON_Orders) to remove non-orders. Then, it performs a stateful update (Exp_Stateful_StockLevel) to track stock levels. Finally, it triggers an alert (Exp_SetAlertID) when the total units ordered exceed 5.

Properties Data Viewer Validation Log Progress

General

Ports

Name	Type	Preci...	Scale	Input	Out...	Vari...	Expression
1 TotalUnitsOrdered	integer	10	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	TotalUnitsOrdered
2 alertID	integer	10	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	IIF(TotalUnitsOrdered>5, 2, 0)

Observe the additional **Kafka** Data Object:

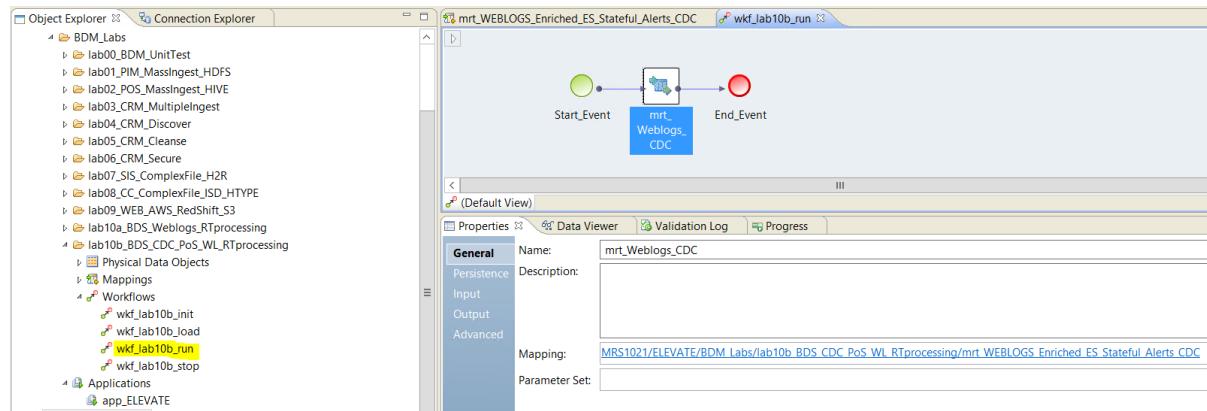
lab10b_BDS_CDC_PoS_WL_RTprocessing

- Physical Data Objects
 - HBASE_CDH_513
 - Kafka
 - Kafka_cdcPoS_IN
 - Kafka_weblogs_IN
 - Kafka_weblogsEnriched
 - Kafka_weblogsEnrichedStats
 - Kafka_weblogsEnrichedStatsCDC** (highlighted)

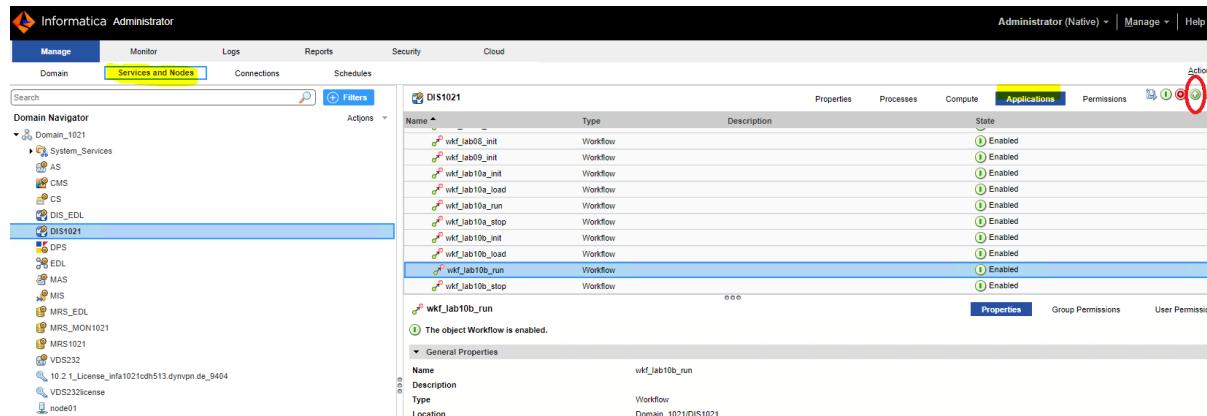
This Data Object describes the Kafka topic '**weblogsEnrichedStatsCDC**' used for product stock-level alerts output.

13.5 Run the Lab

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

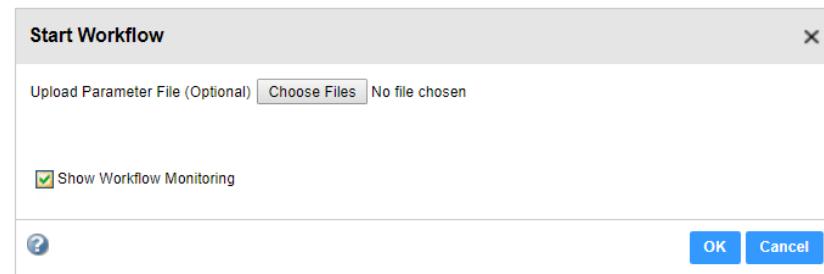


To run **lab10b** mapping log on to the admin console and run the associated **wkf_lab10b_run** workflow navigating under **Manage -> Services and Nodes ->DIS1021** and selecting the '**Applications**' tab view (expand the **app_ELEVATE** application from the list):

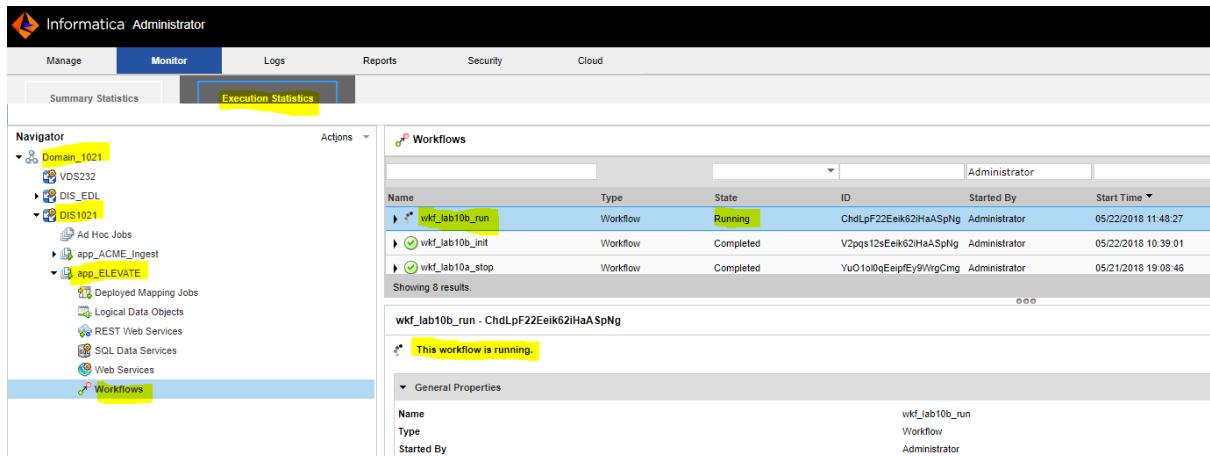


The screenshot shows the Informatica Administrator interface with the 'Services and Nodes' tab selected. The left sidebar shows the Domain Navigator with 'Domain_1021' expanded, revealing 'System_Services', 'AS', 'CMS', 'CS', 'DIS_EDL', and 'DIS1021'. Under 'DIS1021', 'GPS', 'EDL', 'MAS', 'MIS', 'MRS_EDL', 'MRS_MON1021', 'MRS1021', 'VDS232', 'VDS232license', and 'node01' are listed. The main pane shows a table of workflows for 'DIS1021'. A blue row highlights the 'wfk_lab10b_run' workflow. The 'Properties' tab is selected at the bottom. A red circle highlights the 'Actions' button in the top right corner of the table header. The table columns include Name, Type, Description, and State (all enabled).

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

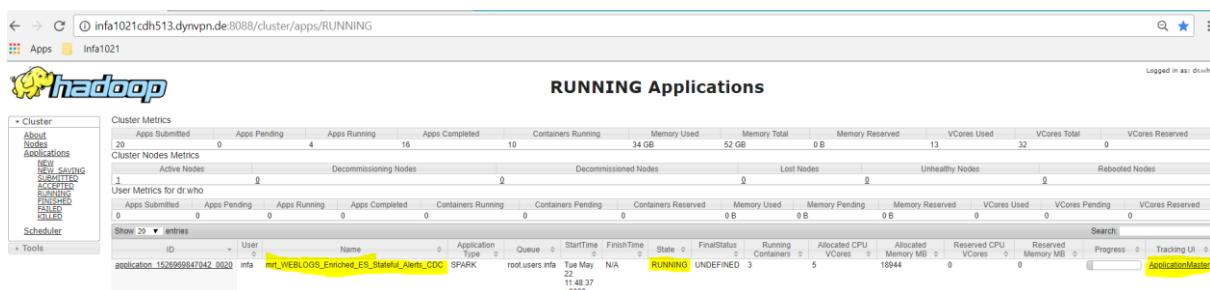


Under **Monitor->Execution Statistics->DIS1021->app_ELEVATE->workflows**, you should see:



The screenshot shows the Informatica Administrator interface with the 'Execution Statistics' tab selected. The left sidebar shows a tree view of domains and various services like VDS232, DIS_EDL, and DIS1021. Under DIS1021, there are sections for Ad Hoc Jobs, Deployed Mapping Jobs, Logical Data Objects, REST Web Services, SQL Data Services, and Web Services. The 'Workflows' section is currently selected. A list of workflows is displayed, including 'wlf_lab10b_run' (Running), 'wlf_lab10b_int' (Completed), and 'wlf_lab10a_stop' (Completed). The 'General Properties' panel on the right shows details for the running workflow.

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



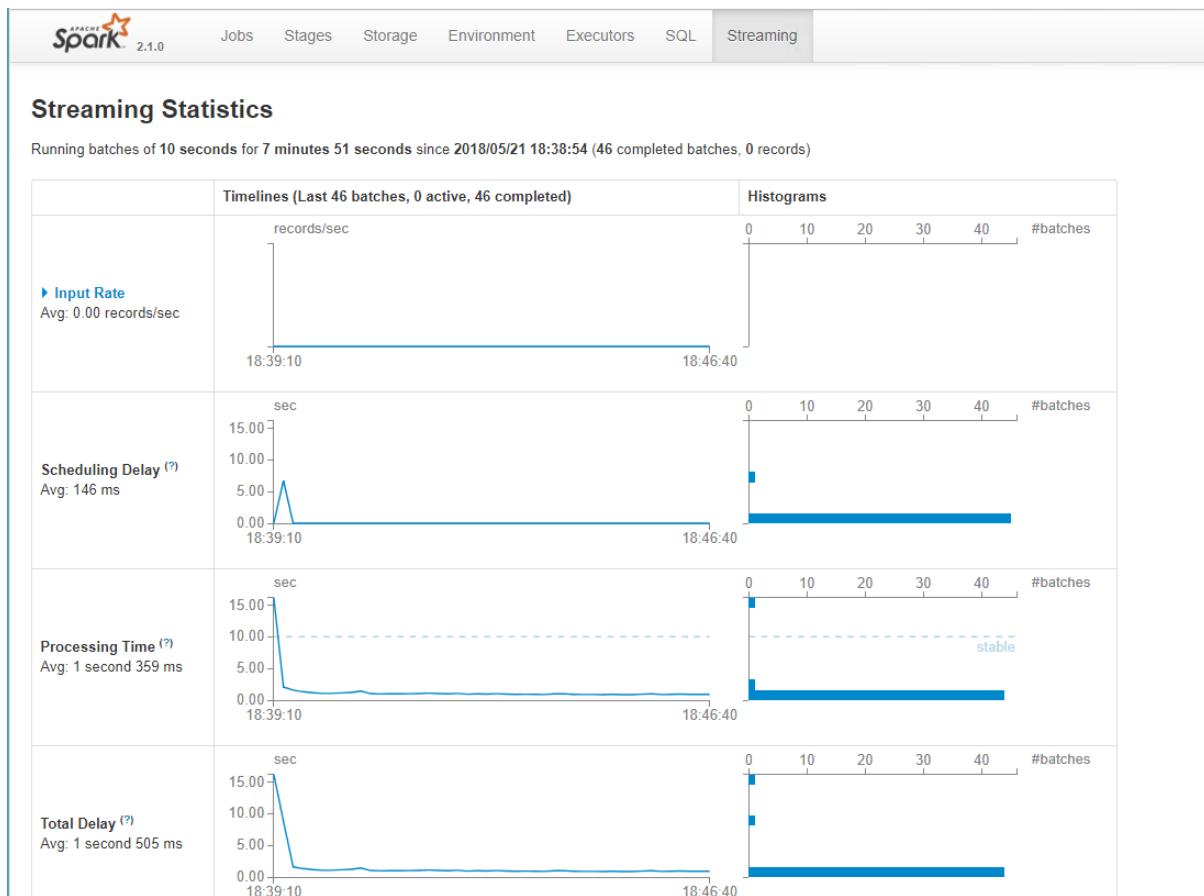
The screenshot shows the YARN Monitor interface with the 'RUNNING Applications' tab selected. It displays a table of running applications with columns for ID, User, Name, Application Type, Queue, Start Time, Finish Time, State, Final Status, Running, Allocated CPU, Allocated Memory MB, Reserved CPU, Reserved Memory MB, Progress, and Tracking UI. One application, 'application_1526959847042_0920', is highlighted.

Click ApplicationMaster on the right; you should see:



The screenshot shows the Spark 2.1.0 interface with the 'Streaming' tab selected. It displays a table of completed jobs with columns for Page, Job Id, Description, Submitted, Duration, Stages: Succeeded/Total, and Tasks (for all stages): Succeeded/Total. Two jobs are listed: job 119 and job 118.

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:

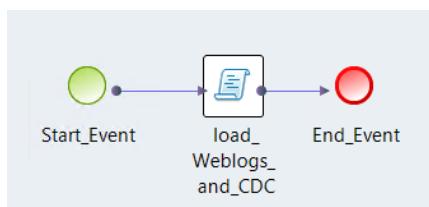


Note: It may take couple of minutes for the Streaming tab to appear.

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

13.6 Load Real-time Mapping Data

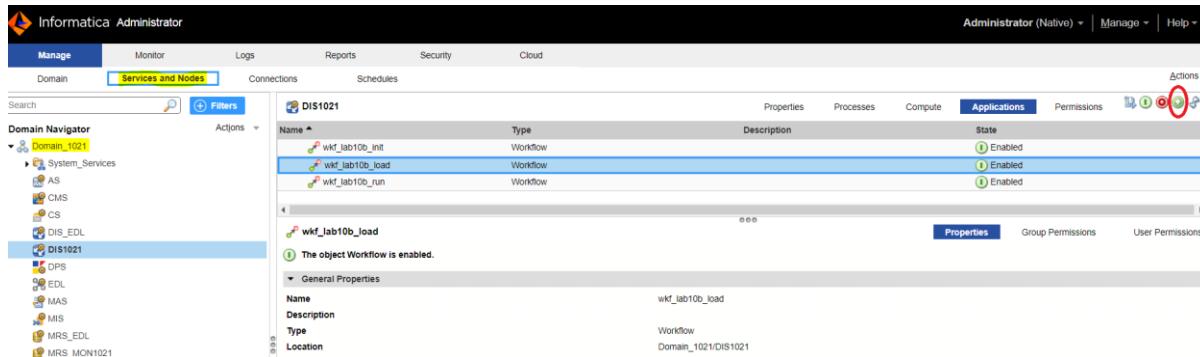
Workflow **wkf_lab10b_load** loads real-time data streams to the **mrt_WEBLOGS_Enriched_ES_Stateful_Alerts_CDC** mapping.



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

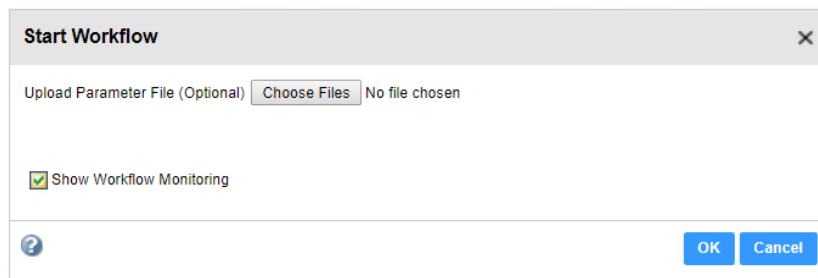
- 1) Raw weblogs data over a Syslog protocol (for EDS input); the EDS data flow will then produce a Kafka output stream on the '**weblogs**' topic (for BDS input).
- 2) A simulation of real-time CDC data capturing changes on the ACME **orders** table, for simulated PoS transactions executed at brick-and-mortar 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):

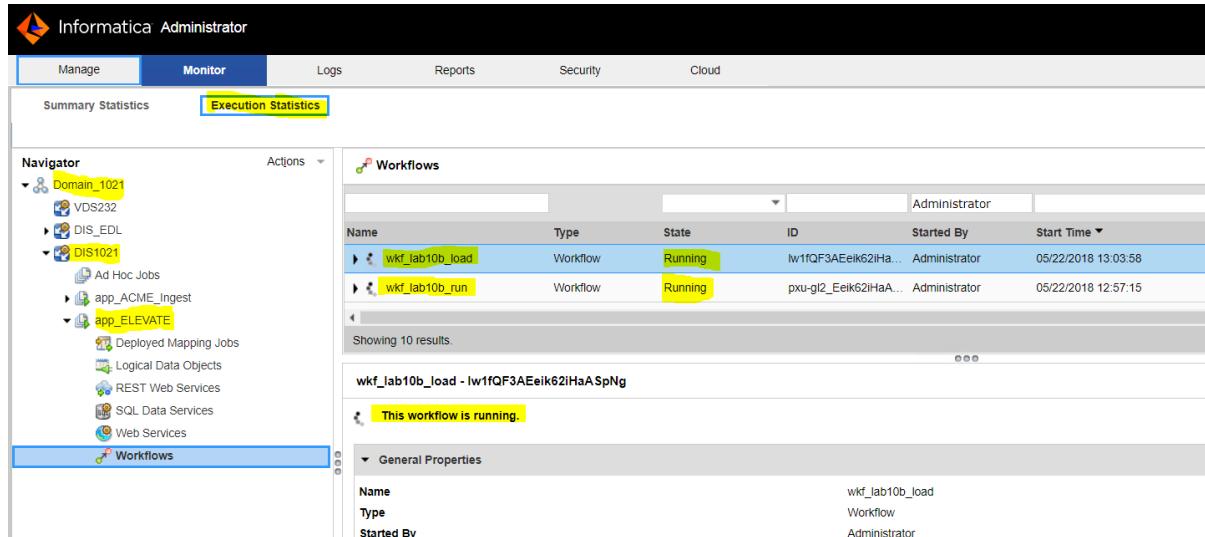


Name	Type	Description	State
wkf_lab10b_int	Workflow		Enabled
wkf_lab10b_load	Workflow		Enabled
wkf_lab10b_run	Workflow		Enabled

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



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



Name	Type	State	ID	Started By	Start Time
wkf_lab10b_load	Workflow	Running	Iw1fQF3AEeik62iHaA...	Administrator	05/22/2018 13:03:58
wkf_lab10b_run	Workflow	Running	pxu-gI2_Eeik62iHaA...	Administrator	05/22/2018 12:57:15

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, "TotalProductUnitsOrdered": 2, "ProductStockLevel_alertID": 0}
{"ProductID": "414073", "OrderID": "2038", "ProductDescription": "Wacom Bamboo Pen & Touch", "Partial_orderUnits": 1, "TotalProductUnitsOrdered": 8, "ProductStockLevel_alertID": 2}
{"ProductID": "465061", "OrderID": "3856", "ProductDescription": "Dymo LabelManager PnP", "Partial_orderUnits": 1, "TotalProductUnitsOrdered": 1, "ProductStockLevel_alertID": 0}
{"ProductID": "101882", "OrderID": "9086", "ProductDescription": "Logitech Keyboard Media K200", "Partial_orderUnits": 1, "TotalProductUnitsOrdered": 1, "ProductStockLevel_alertID": 0}
{"ProductID": "253245", "OrderID": "8192", "ProductDescription": "LG 23' E2331T-BN", "Partial_orderUnits": 1, "TotalProductUnitsOrdered": 4, "ProductStockLevel_alertID": 0}
{"ProductID": "253170", "OrderID": "2038", "ProductDescription": "Acer PC Aspire X1430 E-300/3G/500GB/W7HP", "Partial_orderUnits": 1, "TotalProductUnitsOrdered": 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:

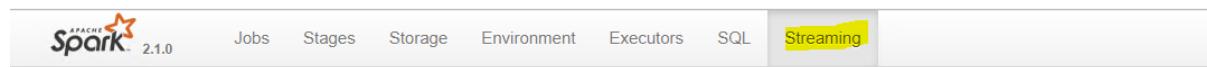
```
> scan 'stock_alerts'
```

```
hbase(main):028:0> scan 'stock_alerts'
ROW                                     COLUMN+CELL
101881                                  column=cf_stock_alerts:Alert, timestamp=1526996361746, value={"ProductID": "101881", "OrderID": "559", "ProductDescription": "Logitech Desktop Cordless MK320", "Partial_orderUnits": 1, "TotalProductUnitsOrdered": 7, "ProductStockLevel_alertID": 2}
102022                                  column=cf_stock_alerts:Alert, timestamp=1526996453635, value={"ProductID": "102022", "OrderID": "19940", "ProductDescription": "Razer Goliathus Speed Extended XL", "Partial_orderUnits": 1, "TotalProductUnitsOrdered": 7, "ProductStockLevel_alertID": 2}
102170                                  column=cf_stock_alerts:Alert, timestamp=1526996391598, value={"ProductID": "102170", "OrderID": "2478", "ProductDescription": "Rapoo keyboard bluetooth for iPad Blade Black", "Partial_orderUnits": 1, "TotalProductUnitsOrdered": 7, "ProductStockLevel_alertID": 2}
121873                                  column=cf_stock_alerts:Alert, timestamp=1526996406744, value={"ProductID": "121873", "OrderID": "9148", "ProductDescription": "Wacom Bamboo Fun Pen & Touch Small", "Partial_orderUnits": 1, "TotalProductUnitsOrdered": 7, "ProductStockLevel_alertID": 2}
171618                                  column=cf_stock_alerts:Alert, timestamp=1526996466435, value={"ProductID": "171618", "OrderID": "9376", "ProductDescription": "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", "ProductDescription": "ACME LCD Arm FPMA-D965 Black", "Partial_orderUnits": 1, "TotalProductUnitsOrdered": 6, "ProductStockLevel_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:



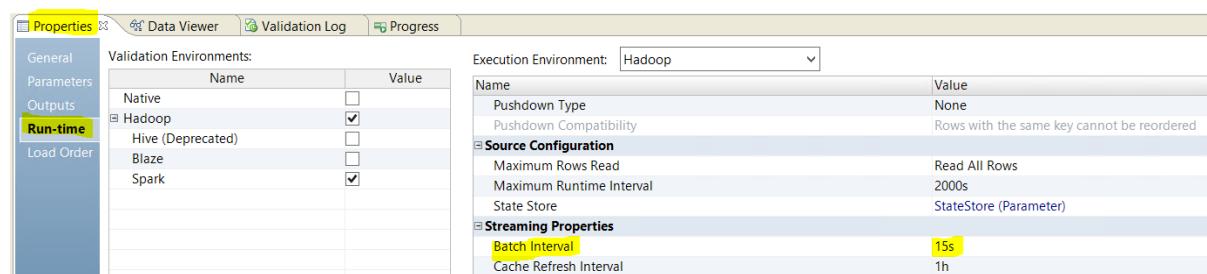
Streaming Statistics

Running batches of 15 seconds for 17 minutes 41 seconds since 2018/05/22 16:33:37 (69 completed batches, 4996 records)



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):



Properties panel showing Run-time configuration:

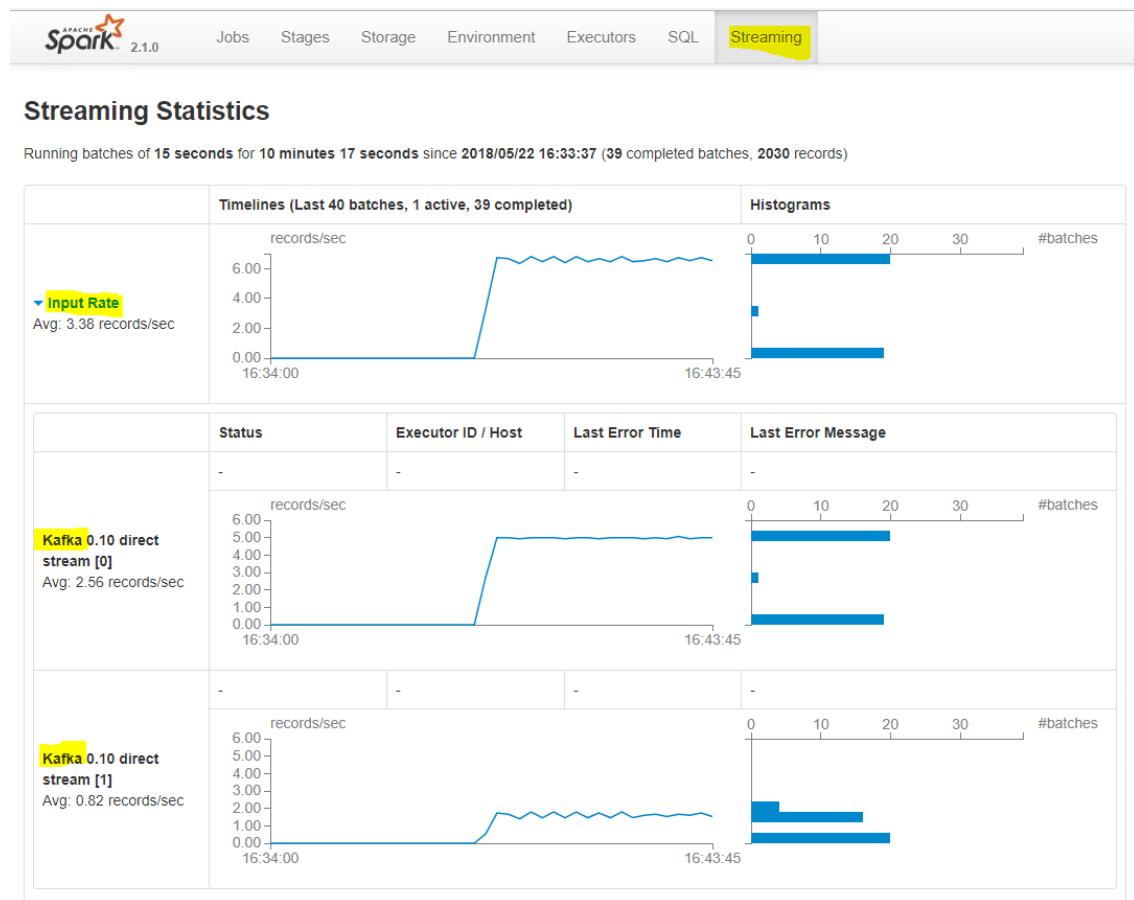
Name	Value
Native	<input type="checkbox"/>
Hadoop	<input checked="" type="checkbox"/>
Hive (Deprecated)	<input type="checkbox"/>
Blaze	<input type="checkbox"/>
Spark	<input checked="" type="checkbox"/>

Execution Environment: Hadoop

Streaming Properties:

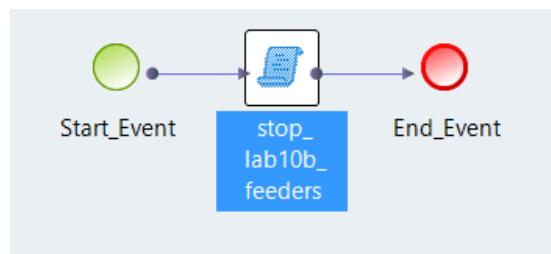
- Batch Interval: 15s
- Cache Refresh Interval: 1h

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



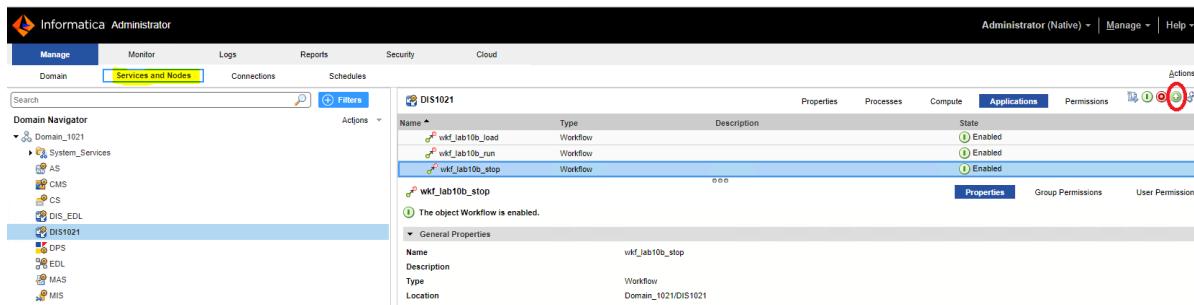
13.8 Stop the Lab

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



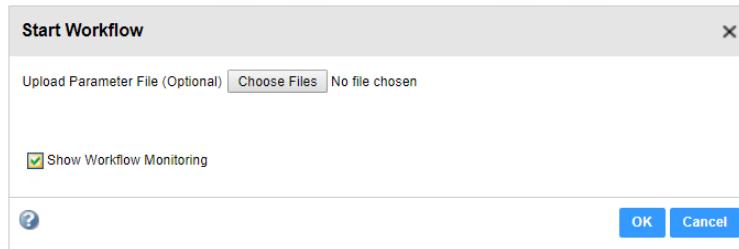
It will stop the same Python scripts that were started by the loading workflow.

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):

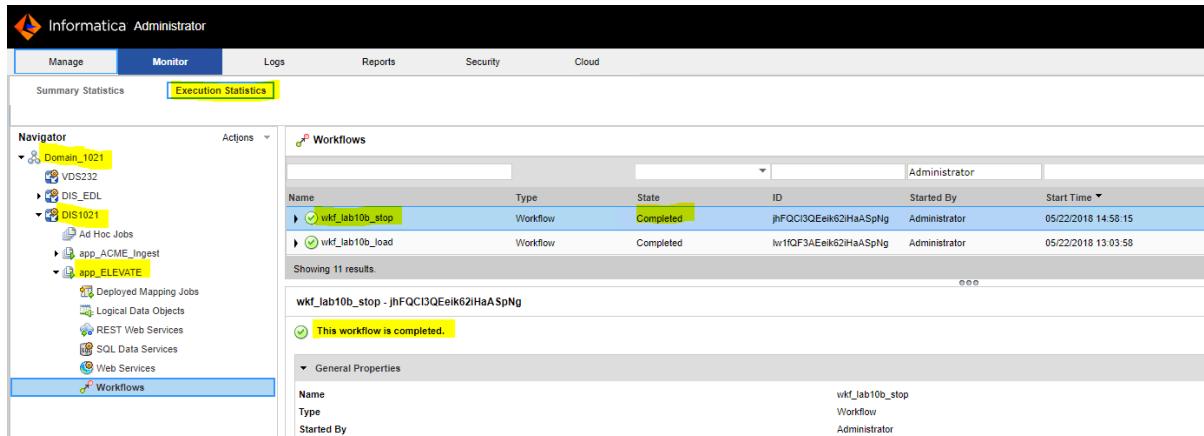


The screenshot shows the Informatica Administrator interface. The top navigation bar includes Manage, Monitor, Logs, Reports, Security, Cloud, and tabs for Domain, Services and Nodes, Connections, and Schedules. The Services and Nodes tab is selected. On the left, the Domain Navigator shows domains VDS232, DIS_EDL, and DIS1021. The main pane displays the 'DIS1021' domain with three workflows: wfk_lab10b_load, wfk_lab10b_run, and wfk_lab10b_stop. The wfk_lab10b_stop workflow is highlighted. The 'Actions' column for this workflow contains a red circle around the 'Start Workflow' button.

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

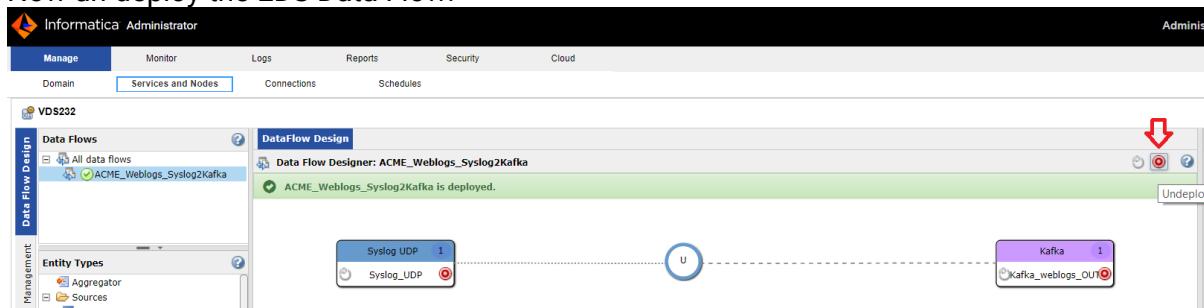


Under **Monitor->Execution Statistics->DIS1021->app_ELEVATE->workflows**, you should see:



The screenshot shows the Informatica Administrator interface under the Monitor > Execution Statistics > DIS1021 > app_ELEVATE > workflows path. The left sidebar shows the Navigator with domains VDS232, DIS_EDL, and DIS1021. The main pane displays a list of workflows under 'wfk_lab10b_stop', 'wfk_lab10b_load', and 'wfk_lab10b_stop'. The 'wfk_lab10b_stop' entry is highlighted with a red circle. Below the list, a message says 'Showing 11 results.' and 'wfk_lab10b_stop - jhFQCI3QEeik62iHaAspNg'. A red circle highlights the 'General Properties' section, which shows the workflow is completed.

Now un-deploy the EDS Data Flow:



The screenshot shows the Informatica Administrator interface under the Services and Nodes tab for VDS232. The left sidebar shows the Data Flow Design panel with a deployed data flow named 'ACME_Weblogs_Syslog2Kafka'. The right side of the screen shows the data flow design with components like 'Syslog UDP' and 'Kafka'. A red arrow points to the 'Undeploy' button located on the right side of the interface.

This concludes BDS lab10b.