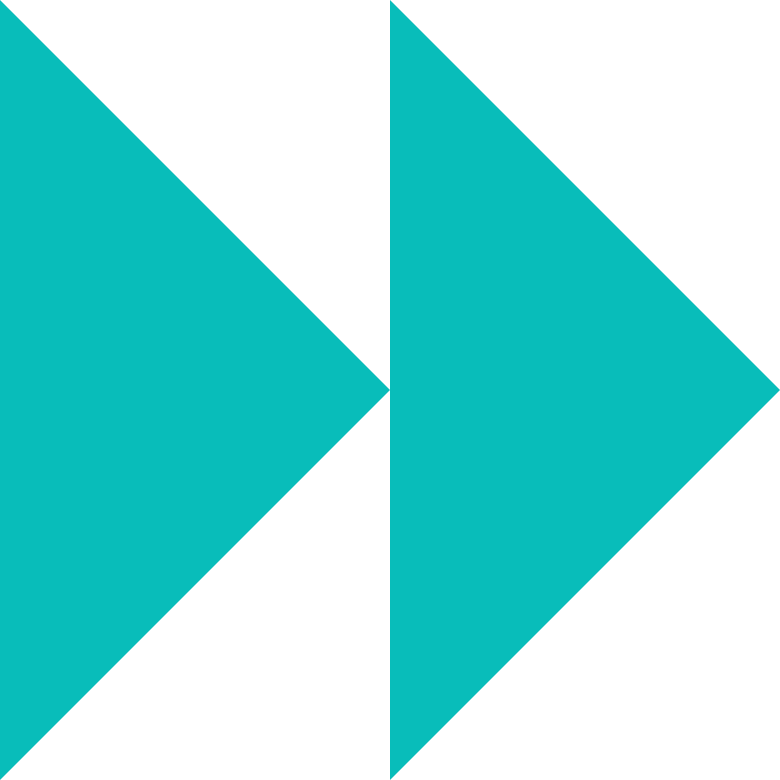
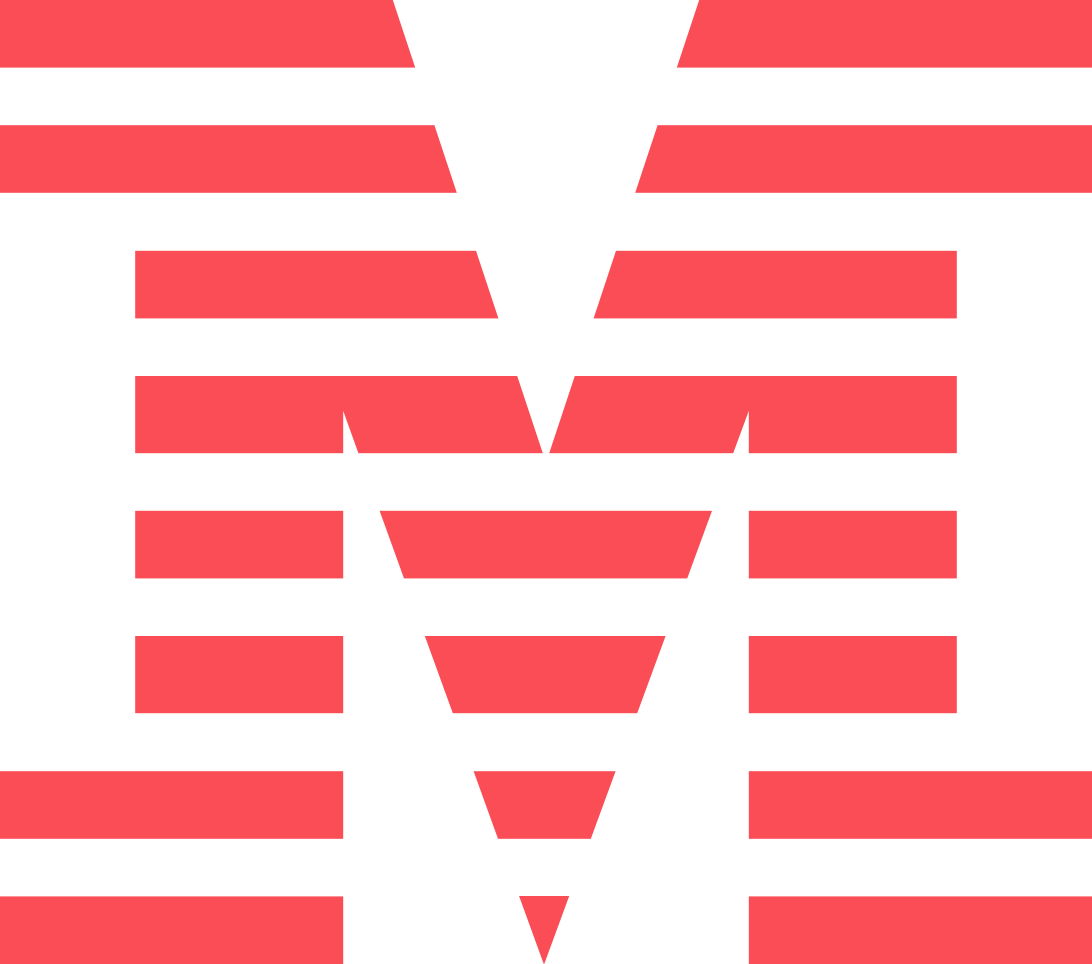
IBM StreamSets

Hands-on Lab Guide



IBM Tech Sales

Table of Contents

[1 Introduction 5](#_Toc197435733)

[1.1 About this hands-on lab 5](#_Toc197435734)

[2 Log-in the Lab Environment 6](#_Toc197435735)

[2.1 Register for the StreamSets Platform Account 6](#_Toc197435736)

[3 Create simple pipeline to process JSON data 7](#_Toc197435737)

[3.1 Creating a new pipeline 7](#_Toc197435738)

[3.2 Adding a pipeline origin 9](#_Toc197435739)

[3.3 Adding a Field Flattener Processor 10](#_Toc197435740)

[3.4 Adding a Field Masker Processor 10](#_Toc197435741)

[3.5 Adding a pipeline destination 11](#_Toc197435742)

[3.6 Preview the pipeline 13](#_Toc197435743)

[3.7 Run the pipeline 15](#_Toc197435744)

[3.8 Checking In the Pipeline 16](#_Toc197435745)

[3.9 Adding/reusing an existing fragment (modularity) 16](#_Toc197435746)

[3.1 Version Control 20](#_Toc197435747)

[4 Geolocation Fraud Detection 21](#_Toc197435748)

[4.1 Creating a new pipeline 21](#_Toc197435749)

[4.1 Adding a Kafka origin 23](#_Toc197435750)

[4.2 SingleStore JDBC Lookup 24](#_Toc197435751)

[4.3 Preform in Flight Calculations based on Business Rules 26](#_Toc197435752)

[4.4 Write Transactions to SingleStore 27](#_Toc197435753)

[4.5 Validate, Preview, and Run the real time pipeline 29](#_Toc197435754)

Notices and disclaimers

© 2024 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information.

This document is distributed “as is” without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted per the terms and conditions of the agreements under which they are provided. The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

IBM products are manufactured from new parts or new and used parts.   
In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

Notices and disclaimers (Continued)

It is the customer’s responsibility to ensure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer’s business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM’s products. IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at

[Learn more →](http://www.ibm.com/legal/copytrade.shtml)

# Introduction

IBM StreamSets is a next-generation data integration platform designed for real-time, continuous data movement across hybrid and multi-cloud environments.

Modern businesses rely on fast-moving, high-volume data—but moving it reliably is often complex. With IBM StreamSets, it’s never been easier to build smart, resilient data pipelines that connect your enterprise systems, cloud applications, and big data platforms.

IBM StreamSets empowers users to design, test, and deploy data pipelines visually—connecting sources like Kafka, Snowflake, AWS, Azure, and on-premise databases with drag-and-drop ease. It ensures your data flows continuously and accurately, without requiring deep technical expertise or custom code.

StreamSets offers end-to-end visibility into your data pipelines with real-time monitoring, smart error handling, and built-in data drift detection. Proactively identify bottlenecks or schema changes before they impact your business, and maintain high-quality data delivery at all times.

Use IBM StreamSets to accelerate digital transformation initiatives, reduce integration time and cost, and empower teams to build, operate, and scale real-time data pipelines with confidence—whether on-prem, in the cloud, or across both.

## About this hands-on lab

Welcome to this hands-on lab, where you’ll gain practical experience in building and managing data pipelines using IBM StreamSets. In today’s data-driven world, the ability to efficiently move, transform, and store data is critical. In this lab, you will work on two key pipelines that showcase different aspects of data engineering using IBM StreamSets:

Pipeline 1: Data pipelines build with intuitive UI

* Objective: Learn how to ingest, transform, and store data in popular file formats. You’ll work with various IBM StreamSets processors to mask and flatten JSON data.
* Key Skills: Pipeline design, data masking and flattening.

2. Pipeline 2: Fraud Detection made using pre-configured components (“fragments”)

* Objective: Learn how to rapidly create pipelines through the use of pre-configured, reusable connections
* Key Skills: Rapid assembly of pipelines using pre-built connections
* Key Skills: Create a real time pipeline with inflight enrichment

# Log-in the Lab Environment

You'll have been registered already; look in your email for an invitation to the StreamSets Control Hub environment and follow the instructions in the email to accept the invitation and gain access.

When signing in, use your IBM Single-Sign-On credentials.

# Create simple pipeline to process JSON data

Data pipelines often require that one read data from different data sources, including data in JSON or XML format. Some of these formats can include data in hierarchical/nested structures. While these nested data structures are great at improving readability, there is a need to flatten them to store them in a relational database and/or doing reporting/analytics with the data.

## Creating a new pipeline

We will create a new pipeline call “My simple pipeline - <YOUR NAME>” by clicking on the hamburger menu on the top left, selecting “Build” menu items and clicking on the “Pipelines”. This will present a list of pipelines you have access to. Now follow the steps below to create a new pipeline.

Connect to StreamSets platform at https://cloud.login.streamsets.com/login using a browser on your local

machine (so you can copy and paste from this document into your browser).

* Click the “Create a pipeline” button
* Give your pipeline a name and description, select next.A screenshot of a computer

  AI-generated content may be incorrect.
* Select your data plane instance (aka engine), click on “Save and next”.

A screenshot of a computer

AI-generated content may be incorrect.

* Ensure the current access control list (ACL) is valid. You should see just yourself with read/write permissions.

Select “Save & Open in Canvas

In this first pipeline, we will leverage a simple yet powerful set of capabilities including:

1) Adding a pipeline origin (Dev Raw Data Source) to simulate some test data originating from a source.

2) Adding a Field Masker Processor to mask an “ssn” field

3) Adding a Field Flattener to flatten the complex JSON data

4) Adding a pipeline destination (Local FS) to write the output of the pipeline to a local CSV file.

5) Connecting the above 4 stages in the above order to finalize the pipeline

## Adding a pipeline origin

We will start by adding a pipeline origin that simulates a data source.

* Click “Add stage”, search for and select “Dev Raw Data Source”. Click on the Dev Raw Data Source tile to add it to the pipeline canvas
* Select the newly added “Dev Raw Data Source” stage, update its Configuration and change the Raw Data value to the below JSON content:

{

"USER": {

"ID": 123,

"NAME": "JOHN DOE",

"SSN": "123-45-6789",

"ADDRESS": {

"STREET": "123 MAIN ST",

"CITY": "SPRINGFIELD",

"STATE": "IL"

}

},

"ORDERS": [

{"ID": 1, "AMOUNT": 250.0},

{"ID": 2, "AMOUNT": 125.5}

]

}

* Set the “Stop After First Batch” checkbox

A screenshot of a computer

AI-generated content may be incorrect.

## Adding a Field Flattener Processor

In this step we’ll add a Field Flattener Processor to flatten the complex JSON payload structure.

* Click on “Add stage” and select the “Field Flattener” stage.

A screenshot of a web page

AI-generated content may be incorrect.

* Select the newly added Field Flattener stage, and inspect its default configuration which will flatten the entire record with “.”as a name separator:

A screenshot of a computer

AI-generated content may be incorrect.

## Adding a Field Masker Processor

In this step we’ll add a Field Masker Processor to mask the “/USER/SSN” field

* Click on “Add stage” and select the “Field Masker” stage:

A screenshot of a computer

AI-generated content may be incorrect.

* Select the newly added Field Masker stage, update the Configuration and add the value “/USER.SSN” to the Fields to Mask property (without quotes) and click the Enter key so the value tokenizes, like this:

A white rectangular object with a black stripe

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Adding a pipeline destination

The results of a pipeline can be written to one or more destinations. IBM StreamSets is centered around a hybrid approach both from an execution environment and pipeline target/destination so one can choose to write the results of the pipeline to either a file, database or any target that is designed to accept such data. In this example we will write a delimited file, grouped into folders organized by current date.

* Click on “Add Stage” and search/select “Local FS”. • Update properties for “Local FS”
  + Under “Data Format”, select “Delimited” as the “Data Format” and “With Header Line”:

A screenshot of a computer

AI-generated content may be incorrect.

* + Under “Output Files”, accept the default Directory template location which is: /tmp/out/${YYYY()}-${MM()}-${DD()}-${hh()}

This location is on the local file system of the StreamSets engine and makes use of StreamSets functions to get the current year, month, day, and hour.

* + Under “Output Files”, also set the File Suffix to “csv” (without quotes).

A screenshot of a computer

AI-generated content may be incorrect.

Your pipeline should now look like this:

A screenshot of a computer

AI-generated content may be incorrect.

## Preview the pipeline

Pipeline Preview allows the user to inspect data as it moves through the pipeline, validating the source data and the results of all transforms before the data is written to a target.

* Click on the Preview button and select “Run Preview”:

A screenshot of a computer

AI-generated content may be incorrect.

* You should see the source data:

A screenshot of a computer

AI-generated content may be incorrect.

* Click on the Field Masker stage and inspect the data before and after masking is appliedA screenshot of a computer

  AI-generated content may be incorrect.
* Click on the Field Flattener stage and inspect the data before and after flattening is applied:

A screenshot of a computer

AI-generated content may be incorrect.

* Close the preview by clicking on the Close Preview button:

A screenshot of a cell phone

AI-generated content may be incorrect.

## Run the pipeline

* Click on the Draft Run button and select “Start Pipeline”

A screenshot of a diagram

AI-generated content may be incorrect.

* You should see metrics like this:

A screenshot of a computer

AI-generated content may be incorrect.

Congratulation you have successfully built your first pipeline

## Checking In the Pipeline

A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Adding/reusing an existing fragment (modularity)

Select the Edit Button at the top of the page to edit the pipeline and create a version 2

A screen shot of a computer

AI-generated content may be incorrect.

IBM StreamSets promotes reuse of existing code fragments for better modularity and reusability. In this example we will select and use an existing pipeline fragment called “Mask and flatten JSON”. This fragment will mask and flatten the JSON payload.

A diagram of a diagram

AI-generated content may be incorrect.

After select both of the stages at once, this will prompt the screen below

A screenshot of a computer

AI-generated content may be incorrect.

Select Create Pipeline Fragment

A screenshot of a computer

AI-generated content may be incorrect.

Next Screen will ask to give your Fragment a name, for the purposes of the lab you can give it a name like “Mask and Flatten JSON”

A screenshot of a computer

AI-generated content may be incorrect.

We will use the authoring engine set up for the lab

A screenshot of a computer

AI-generated content may be incorrect.

Next hit, Save and Publish, to save the fragment

A screenshot of a computer

AI-generated content may be incorrect.

Check In the Fragment and hit publish and close

A white rectangular sign with a black and white text

AI-generated content may be incorrect.

Your pipeline should now look like this with your new fragment

A screenshot of a computer

AI-generated content may be incorrect. A diagram of a diagram

AI-generated content may be incorrect.

We can expand the fragment and check if it matches

A screenshot of a computer

AI-generated content may be incorrect.

Preview the pipeline again

## Version Control

StreamSets support built in versioning control for pipelines and fragments

A screenshot of a computer

AI-generated content may be incorrect.

You will select the V-2 Draft button at the top of the screen and then hit the 3 Dots to compare with the Previous version of the pipeline that we checked in

A white grid with black dots

AI-generated content may be incorrect.

We can see what changed between version 1 and version 2

A diagram of a document

AI-generated content may be incorrect. A white rectangular sign with a puzzle piece

AI-generated content may be incorrect.

We created a new fragment with the Field Flattener and Field Masker Stage and used that fragment in the V-2 of the pipeline

Congratulations you have created your first pipeline with IBM StreamSets

# Geolocation Fraud Detection

Data pipelines often require that one read data from different data sources, including data in JSON or XML format. Some of these formats can include data in hierarchical/nested structures. While these nested data structures are great at improving readability, there is a need to flatten them to store them in a relational database and/or doing reporting/analytics with the data.

Real-time data Streaming from Kafka using IBM StreamSets Pipelines can be used to filter and enrich before writing to one or more destinations. In this pipeline we will read real time data from credit card transactions, preform a SingleStore lookup for last transaction, use business rules to flag if suspicious, and then write to multiple destinations.

A diagram of a company

AI-generated content may be incorrect.

## Creating a new pipeline

We will create a new pipeline call “Transaction Streaming - <YOUR NAME>” by clicking on the hamburger menu on the top left, selecting “Build” menu items and clicking on the “Pipelines”. Click the “Create a pipeline” button

* Give your pipeline a name and description, select next. A screenshot of a computer

  AI-generated content may be incorrect.
* Select your data plane instance (aka engine), click on “Save and next”.

A screenshot of a computer

AI-generated content may be incorrect.

* Ensure the current access control list (ACL) is valid. You should see just yourself with read/write permissions.

Select “Save & Open in Canvas

## Adding a Kafka origin

A close-up of a tag

AI-generated content may be incorrect.

We will start by adding a Kafka origin to stream financial transactions

A screenshot of a phone

AI-generated content may be incorrect.



A grey rectangular object with blue border

AI-generated content may be incorrect.

* We have a global connection for Kafka, so we will use the drop down menu to use the pre-configured connection to Kafka
* Change the Consumer Group to your name

A white line on a white surface

AI-generated content may be incorrect.

A white lined paper with black lines

AI-generated content may be incorrect.

* Now that we have a predefined connection to Kafka, we want to listen to a particular Kafka topic, in the case of this lab we will use the below value
  + cc\_transactions

A screenshot of a computer

AI-generated content may be incorrect.

* Within the Data Format tab in the Kafka Origin, select JSON as the Data Format

## SingleStore JDBC Lookup

A logo of a server

AI-generated content may be incorrect.

A white square with black and blue text

AI-generated content may be incorrect.

* We will click on the add stage button to continue building out our pipeline

A screenshot of a computer

AI-generated content may be incorrect.

* Search for JDBC Lookup and select the stage

A white background with black lines

AI-generated content may be incorrect.

* We have a predefined connection to SingleStore that we will use to lookup the last known time and location of a particular customer transaction
* Select the SingleStore Transaction within the drop down menu

A screenshot of a computer code

AI-generated content may be incorrect.

* We will now use a SQL query to selection the location from the previous transaction of the card for fraud detection. Please use the below SQL Query

WITH prev AS (

SELECT location, timestamp

FROM transactions

WHERE account\_id = ${record:value('/account\_id')} AND suspicious = FALSE

ORDER BY transaction\_id DESC

LIMIT 1

)

SELECT

GEOGRAPHY\_DISTANCE(prev.location, "${record:value('/location')}") as delta\_x,

TIMESTAMPDIFF(SECOND, prev.timestamp, "${record:value('/timestamp')}") as delta\_t

FROM prev;

## Preform in Flight Calculations based on Business Rules

In this stage we will use the information pulled from the SingleStore lookup to create a new field in our record called suspicious. This will be created based on a calculation if the last two transactions time and location could be accomplished

A screenshot of a computer

AI-generated content may be incorrect.

* Select the Add Stage button and search Expression Evaluator

A screenshot of a computer

AI-generated content may be incorrect.

* Under output field we will create a new field called suspicious
* This field will be determined by using the below field expression.

${record:exists('/delta\_x') and record:exists('/delta\_t') and record:value('/delta\_x') / record:value('/delta\_t') > 268.224}

* Let’s break down this expression to find out what each individual part is doing within the expression
  + ${record:exists('/delta\_x’) and record:exists(‘/delta\_t')
    - This makes sure that the field that’s we enriched our records with in the previous stage with a SingleStore lookup
  + record:value('/delta\_x') / record:value('/delta\_t') > 268.224
    - This takes the record value from our record for distance and time. This divides the distance by the time and if it is above in terms of meters. 268.224 meters per second is 600 mph or the cruising speed of an airplane
    - We have used this value to flag suspicions records for fraud if the transaction was made quicker than an airplane could travel in the time between transactions
* Add another field to the record to track the user running the pipeline:



* This is really only needed as we have many people writing to the same table in our target database!

## Write Transactions to SingleStore

In this final stage of creating a streaming pipeline we will write the data to SingleStore for all the transactions. This will allow us to continue enriching records with real time data for future transactions.

A screenshot of a computer

AI-generated content may be incorrect.

* Select the Add Stage Button

A screenshot of a computer

AI-generated content may be incorrect.

* Select the SingleStore Destination

A screenshot of a login

AI-generated content may be incorrect.



* Again, we will use a provided global connection to connect to Single Store

A white background with black lines

AI-generated content may be incorrect.

* For schema we will use finance and table name transactions
  + finance
  + transactions

## Validate, Preview, and Run the real time pipeline

A screenshot of a computer

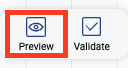
AI-generated content may be incorrect.

* Now we will validate the pipeline to make sure we can connect to the Kafka origin, SingleStore lookup processor, and the SingleStore Destination

A green text on a white background

AI-generated content may be incorrect.

* We will get the following message if we are able to connect to the origin, processors, and destination



* Now let’s preview the pipeline to make sure each stage is working as we intended for development purpose before we write to the destination

Kafka Origin

A close-up of a puzzle

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

Here is an example of the data how we receive it as a Kafka message for a credit card transaction

A logo of a server

AI-generated content may be incorrect.

Within our JDBC lookup to SingleStore, we have now pulled the information for the difference in location calculated by meters, as the new field delta\_x and the difference in time in second, as the new field delta\_t

A white rectangular object with a black border

AI-generated content may be incorrect.

A white rectangle with yellow and orange letters and a blue circle with black text

AI-generated content may be incorrect.

In the expression evaluator we will calculate on the fly if the speed of the transaction (distance/time) is faster than 268.224 meters per second (600 miles per hour) or faster than a commercial airplane

A screenshot of a computer

AI-generated content may be incorrect.

In this case we can calculate that our speed is ~421 meters or 941.75 miles per hour. This is faster than our expression of 600 miles per hour so the record is flagged as suspicious

A logo on a tile

AI-generated content may be incorrect.

In the last stage we write the record back to SingleStore to be able to tracking these transactions and flag transactions in a real time mode

Now Lets Run the Pipeline

A screenshot of a phone

AI-generated content may be incorrect.

Hit Draft Run, then Start Pipeline

A screenshot of a graph

AI-generated content may be incorrect.

This will run until we tell it to stop, running as a real time pipeline

A screenshot of a phone

AI-generated content may be incorrect.

To stop the pipeline, click on the stop button to gracefully shut down the pipeline

Congratulations you have created your first real time streaming pipeline with IBM StreamSets