# Creating Weather pipeline with StreamSets and MariaDB.

## Introduction.

This is an instruction on how to create a simple ETL with StreamSets.

For the data source <https://openweathermap.org/current> will be utilized.

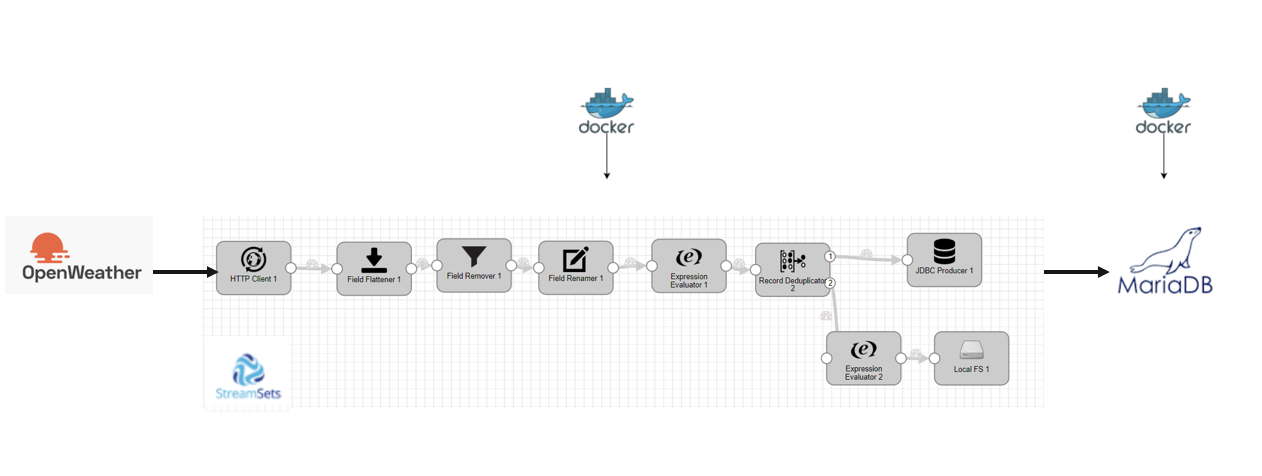
You will need to register and Sign in to get your own api key.

As target database MariaDB will be used. Environment will be created using docker containers and docker-compose utility.

This instruction will lead you through the basic steps that need to be taken to create and configure pipeline that will do the following:

1. Poll <https://openweathermap.org/> and obtain current weather for a certain city
2. Transform the incoming data in order to get rid of useless information, duplicates, present temperature in Celsius
3. Write the result to MariaDB database and write duplicated values to csv file in local folder

Pipeline architecture



## Environment preparation

1 ) As the environment will spin-up with docker you need to install it, register and sign in.

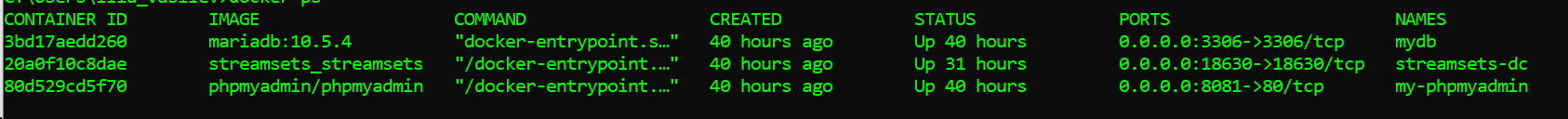
<https://www.docker.com/get-started>

2)Once docker desktop is up and running, download [archive](https://epam-my.sharepoint.com/:u:/r/personal/alexey_girin_epam_com/Documents/Big%20Data%20QA%20stack/SPb%20Big%20Data%20Course/12.%20Data%20pipeline%20management/Exercises/streamsets.rar?csf=1&web=1&e=pkGcms) with docker files

3)Open command line -> go to the directory where you’ve unarchived the docker files and execute the following commands(wait till completion before executing the next command):

* docker-compose build
* docker-compose up
* docker ps

In the end you should see the following



4)Services will be available by the following paths:

Streamsets  http://localhost:18630/ user: Admin Password: Admin

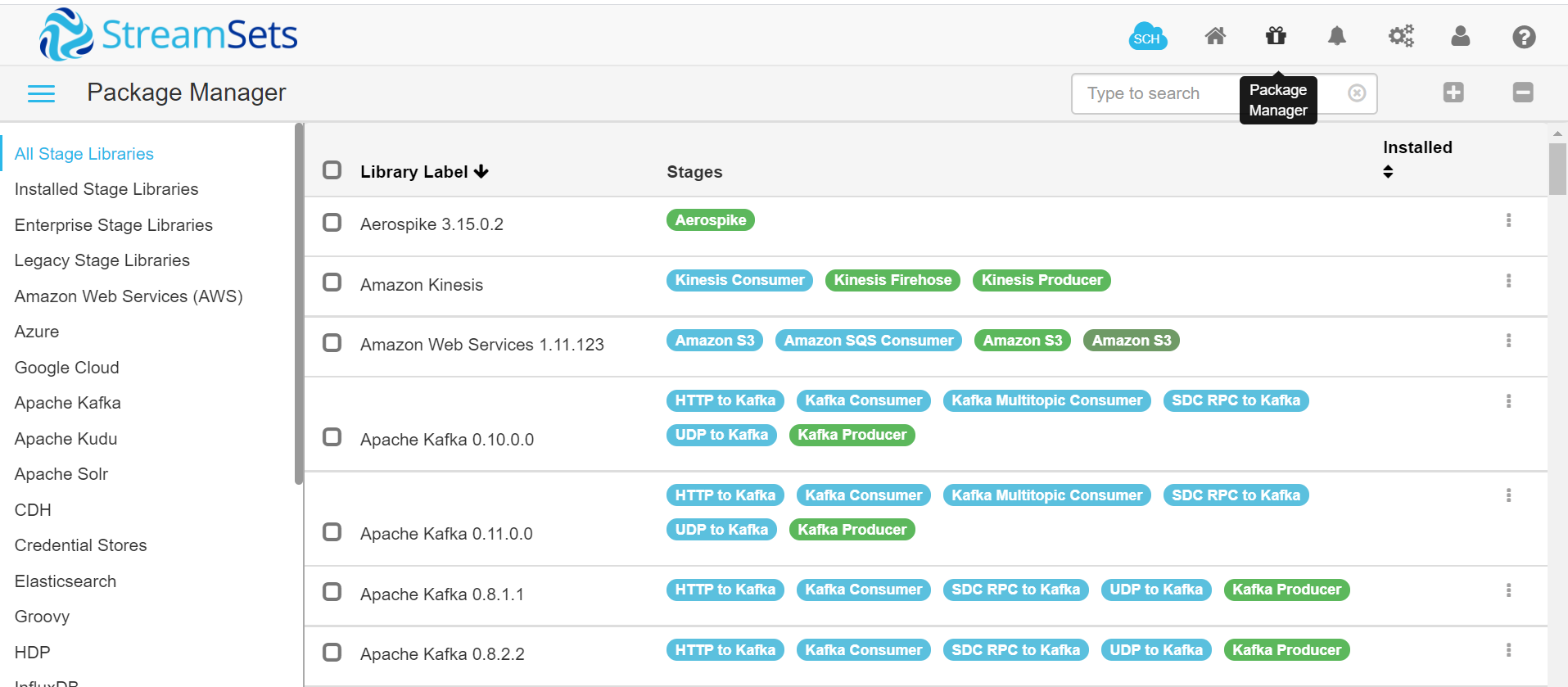
MariaDB http://localhost:3306/ user: root password: mypass123

phpmydamin: <http://localhost:8081/>

Please note: if the ports mentioned above are in use, you’ll need to change them in docker-compose.yml and repeat point 3.

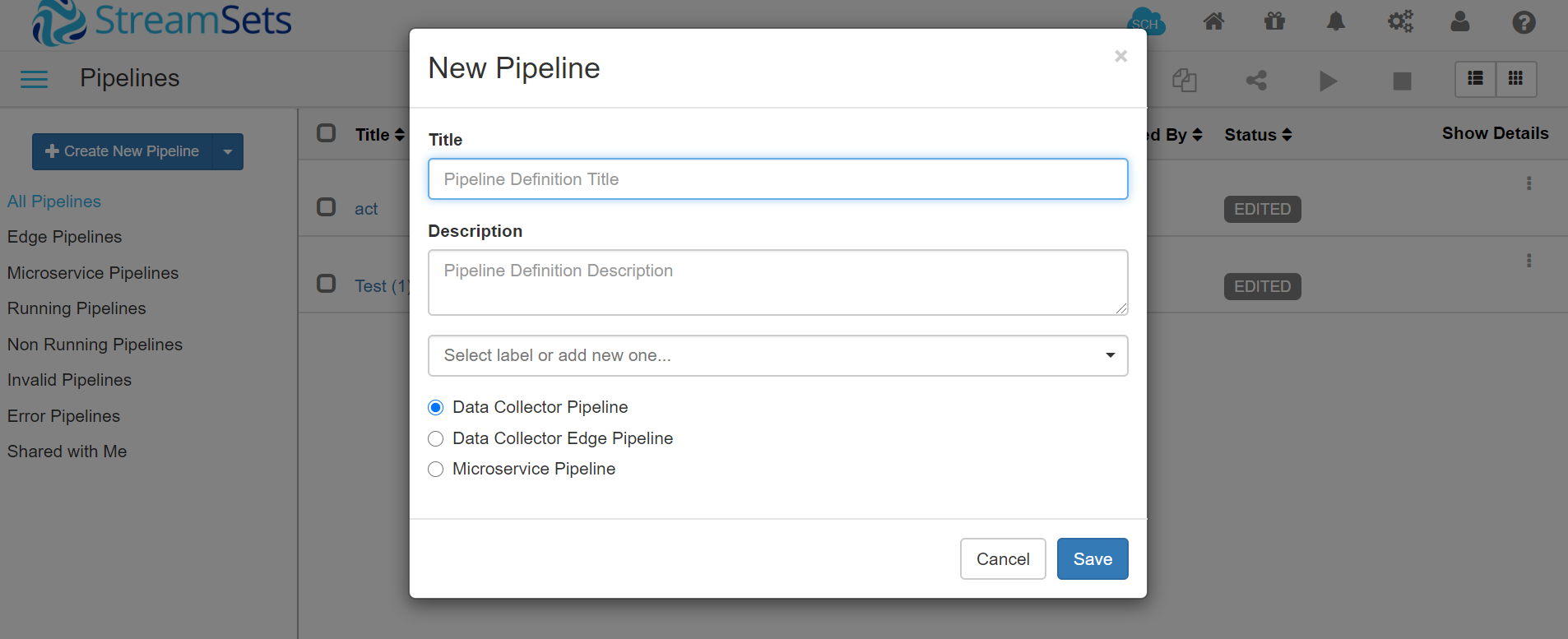
5)In order to be able to write data to MariaDB you need to install JDBC driver for MariaDB in StreamSets

* + Login streamsets http://localhost:18630/ user: Admin Password: Admin
  + Go to Package Manager -> Scroll the left pain to the very bottom to External Libraries
  + And install MariaDB driver (mariadb-java-client-2.6.0.jar is in archive with docker files)
  + restart StreamSets Data Collector
  + Go to External libraries in Package manager and check that MariaDB driver is installed



## Pipeline creation.

On Home screen press Create new Pipeline button.



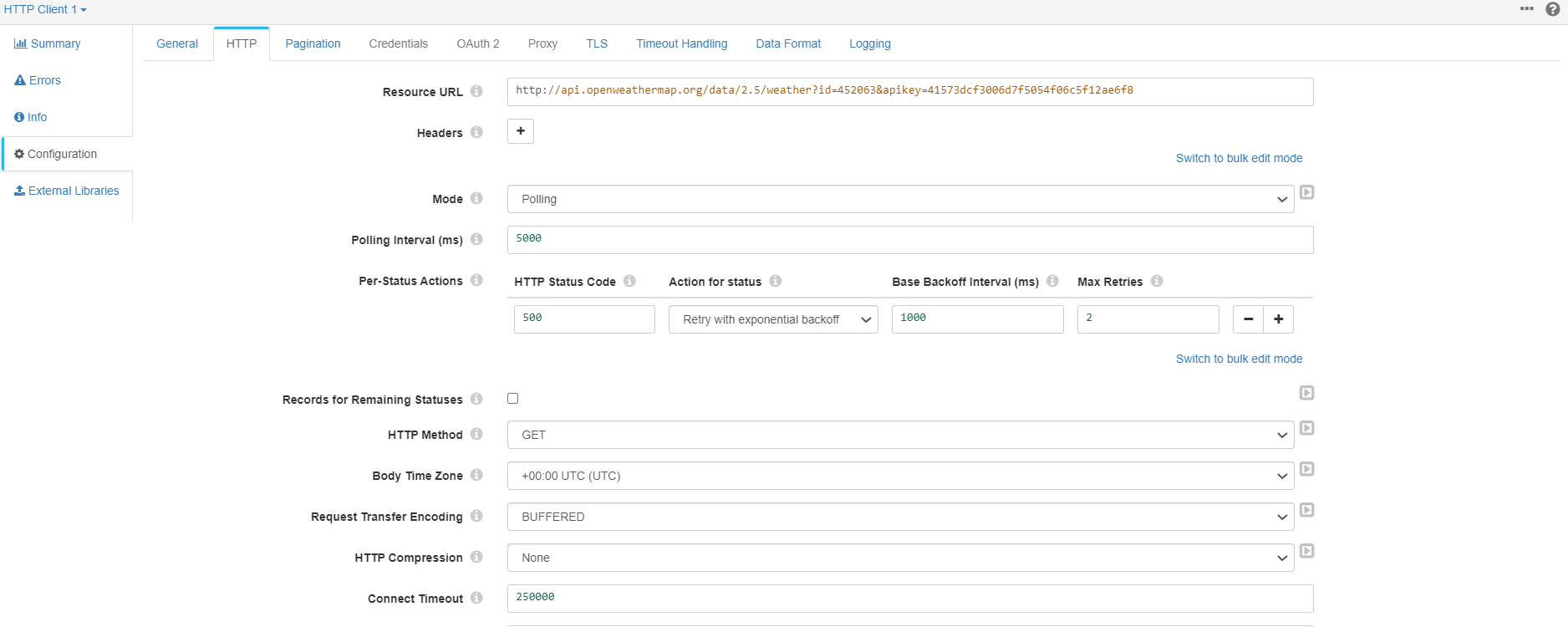
Specify the pipeline name, description and press Save.

### HTTP Client

Then empty pipeline is opened and you need to add the Origin that will communicate with openweather service and get the data.

**HTTP client** should be used, add it to the pipeline and configure as described below:

For resource <https://openweathermap.org/current> By City ID is used



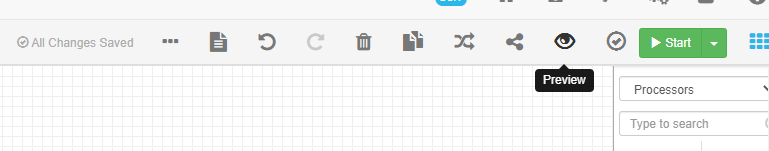
### Field Flattener

As weather API returns response in JSON format like this

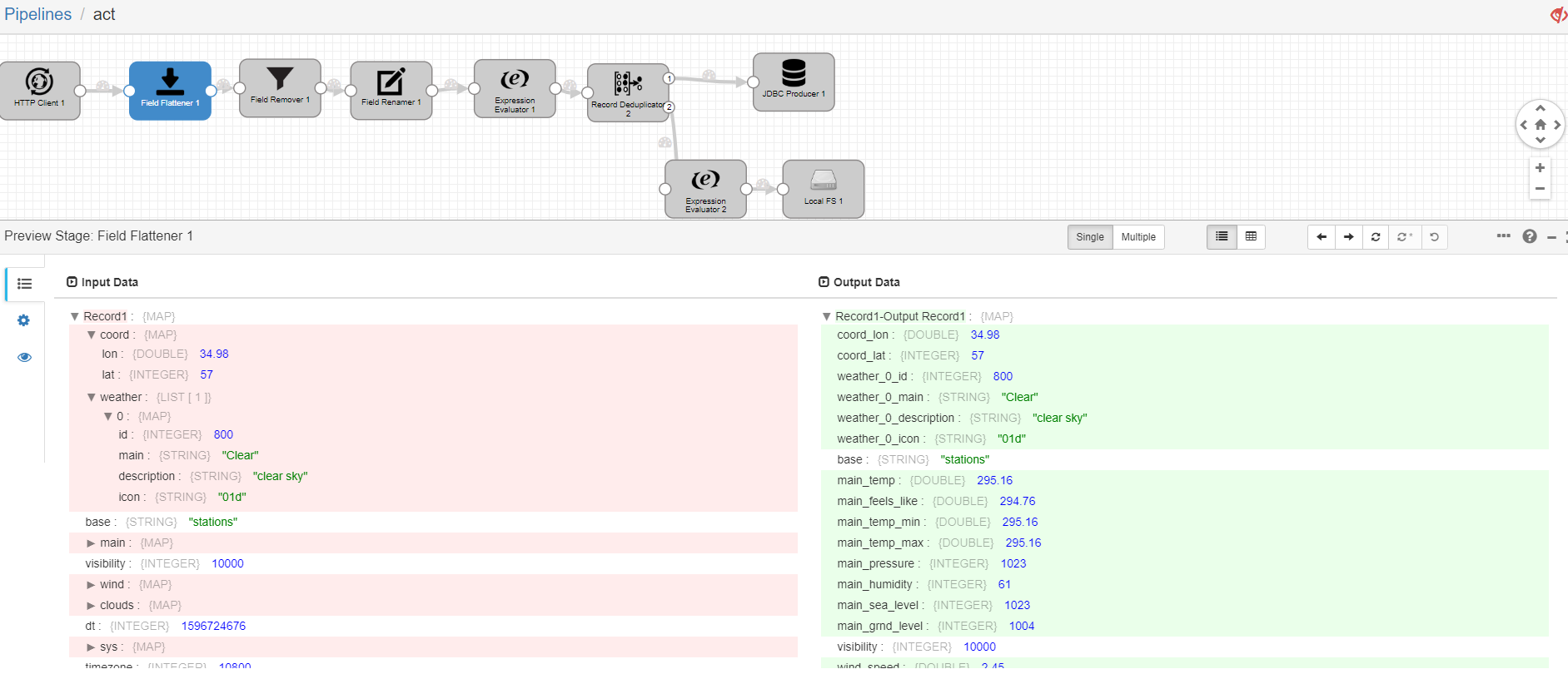
{"coord":{"lon":34.98,"lat":57},"weather":[{"id":800,"main":"Clear","description":"clear sky","icon":"01d"}],"base":"stations","main":{"temp":295.16,"feels\_like":294.76,"temp\_min":295.16,"temp\_max":295.16,"pressure":1023,"humidity":61,"sea\_level":1023,"grnd\_level":1004},"visibility":10000,"wind":{"speed":2.45,"deg":314},"clouds":{"all":0},"dt":1596724676,"sys":{"country":"RU","sunrise":1596678566,"sunset":1596735748},"timezone":10800,"id":452063,"name":"Dubrovka","cod":200}

You need to Flatten json with the help of **Field Flattener** processor.

You can test the output by clicking the preview button



Then click Run Preview, then click the Field Flattener and check the result.

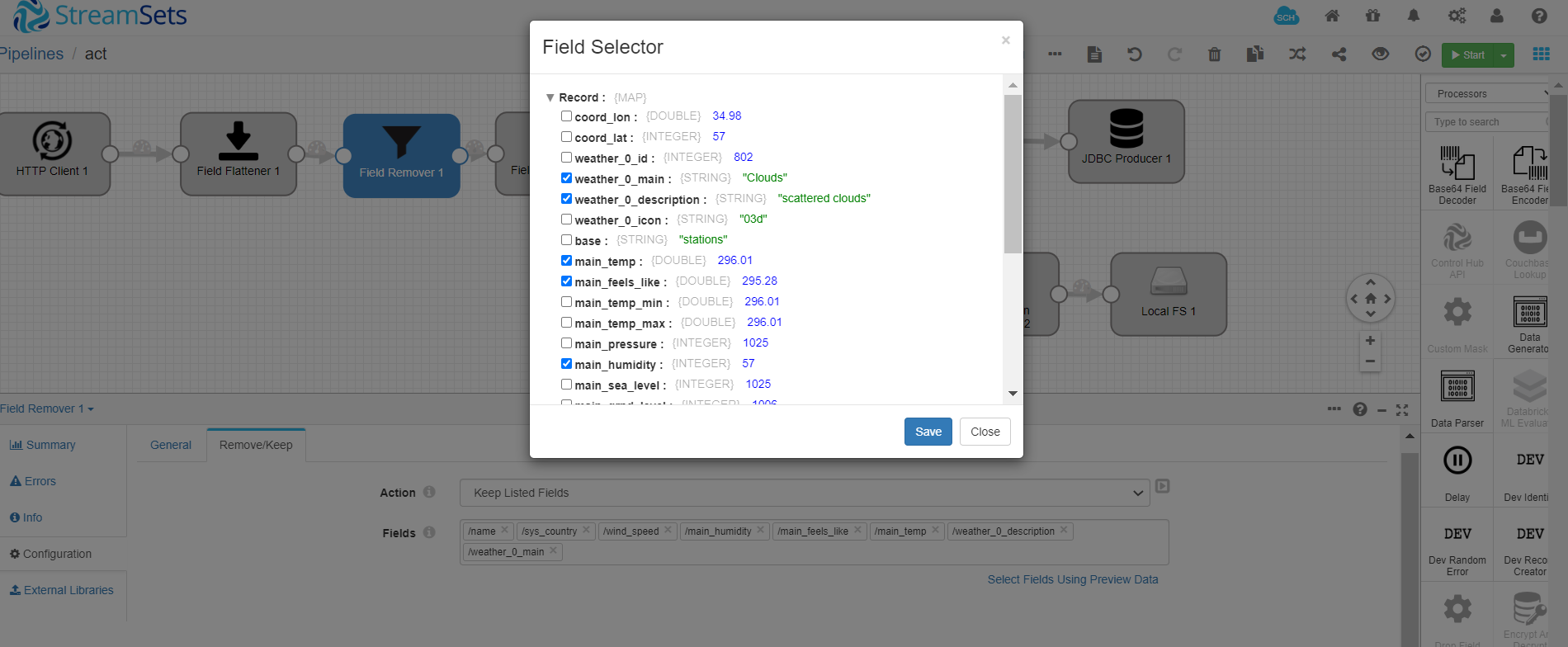


Note: you can preview any stage of your pipeline by choosing it. In case something doesn’t work or configured incorrectly you will get an exception instead.

### Field Remover

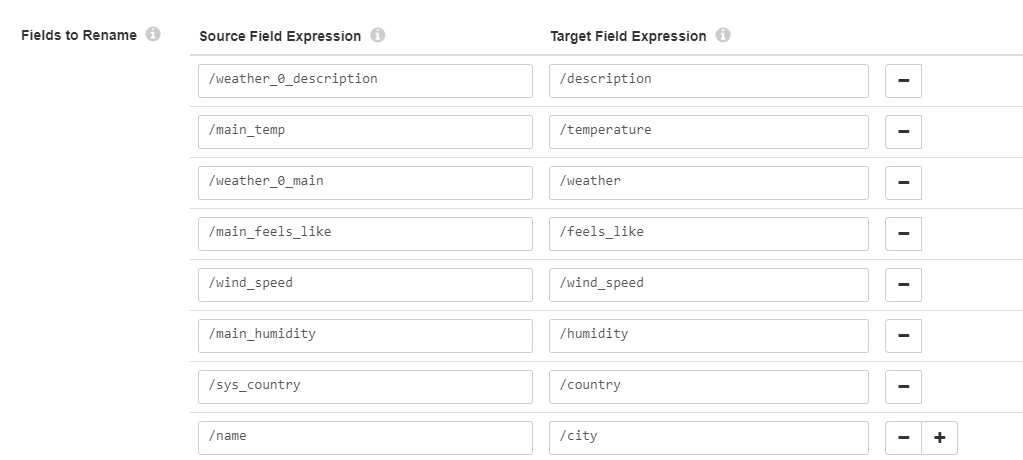
After that you will need to get rid of useless fields with the help of the **Field Remover** processor.

Field that should remain can be specified in Field Selector menu or listed in Fields field.



### Field Renamer

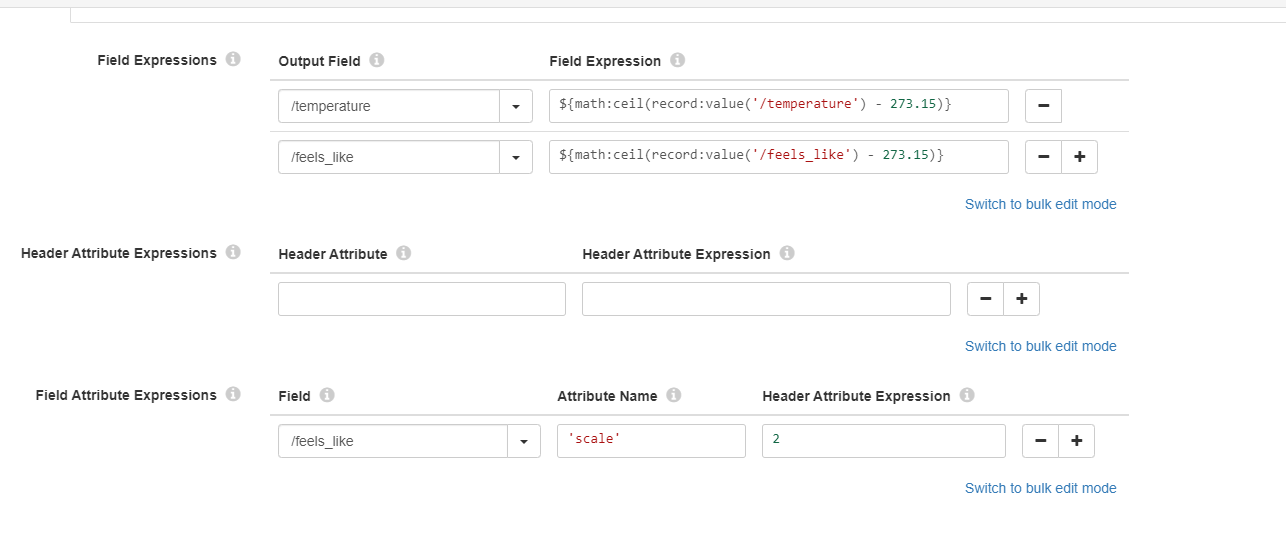
Since the field names are taken from API response it doesn’t match column names in the target table of MariaDB, so “Field Renamer” should be utilized.

The set of fields can be specified using UI menu, regular expressions are also supported, it is handy to use if common rule of renaming can be applied for several fields. 

### Expression Evaluator

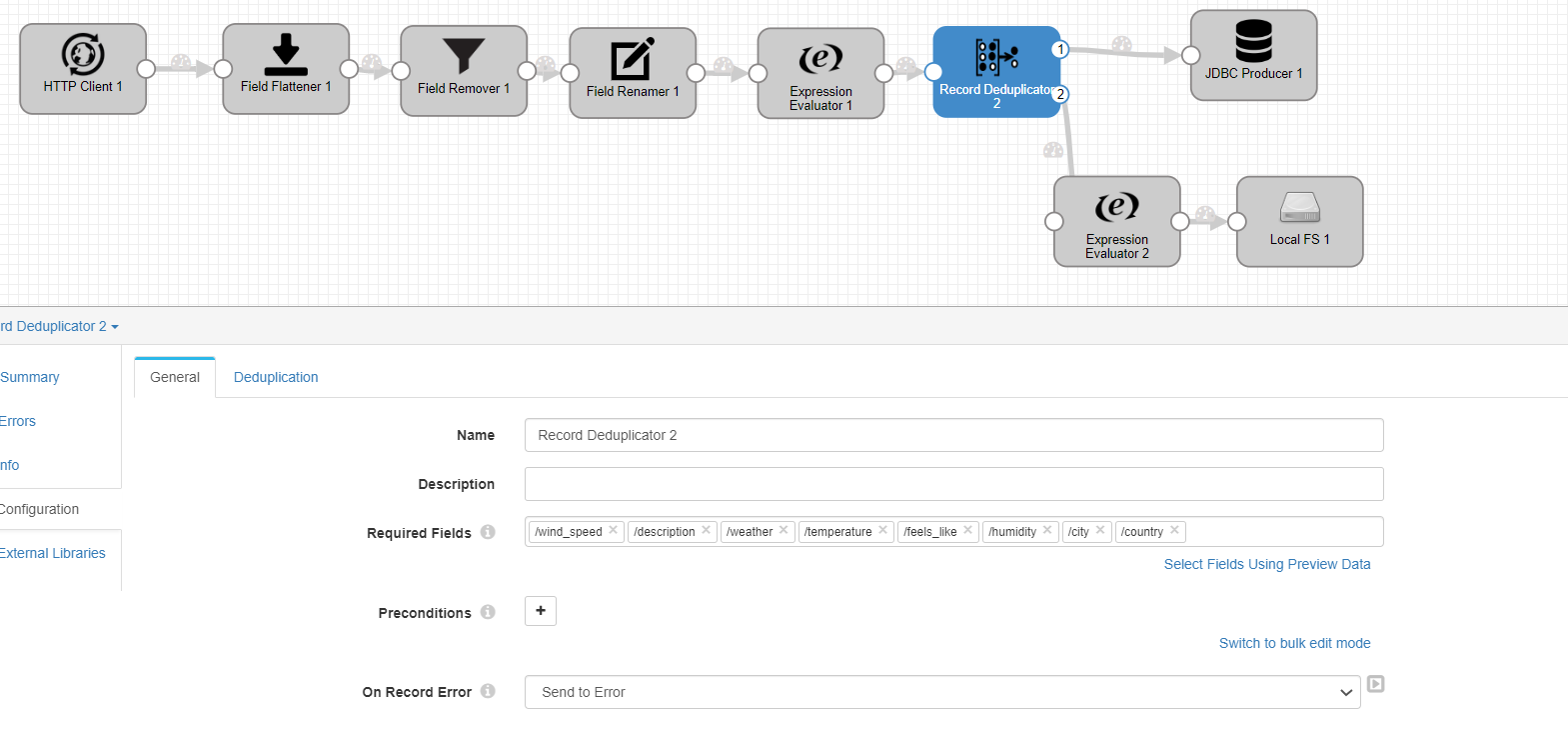
As we need to store the temperature in Celsius, then we need to add an “Expression Evaluator” to convert temperature values from K to C.

For each field we need to define an expression and proper attribute for scaling and rounding.



### Record Deduplicator

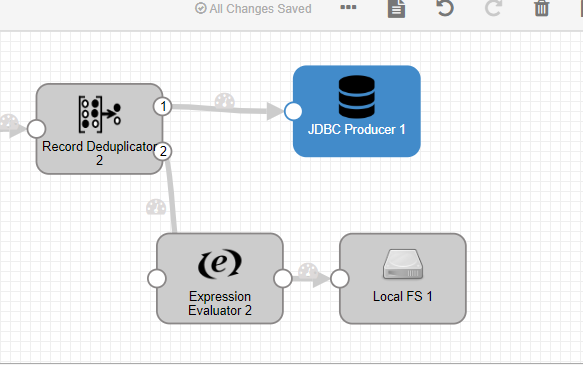
Since we our HTTP client is set up in Polling mode, it may return duplicate records that we don’t need to store. To get rid of we should add “Record Deduplicator”



## Destinations.

De-duplicated records should be stored in our MariaDB table and duplicates should go to a CSV file.

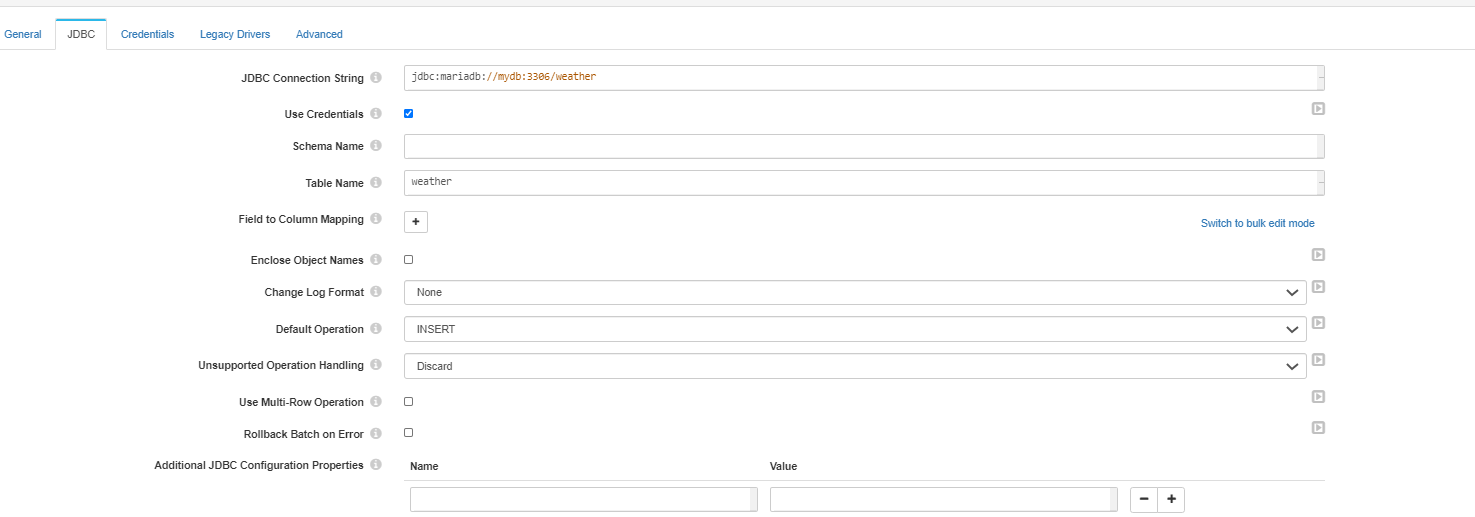
So 2 destinations should be added correspondingly.



### JDBC Producer

To setup JDBC producer you need to specify connection settings and database credentials on a Credentials tab. See Environment preparation section for credentials.

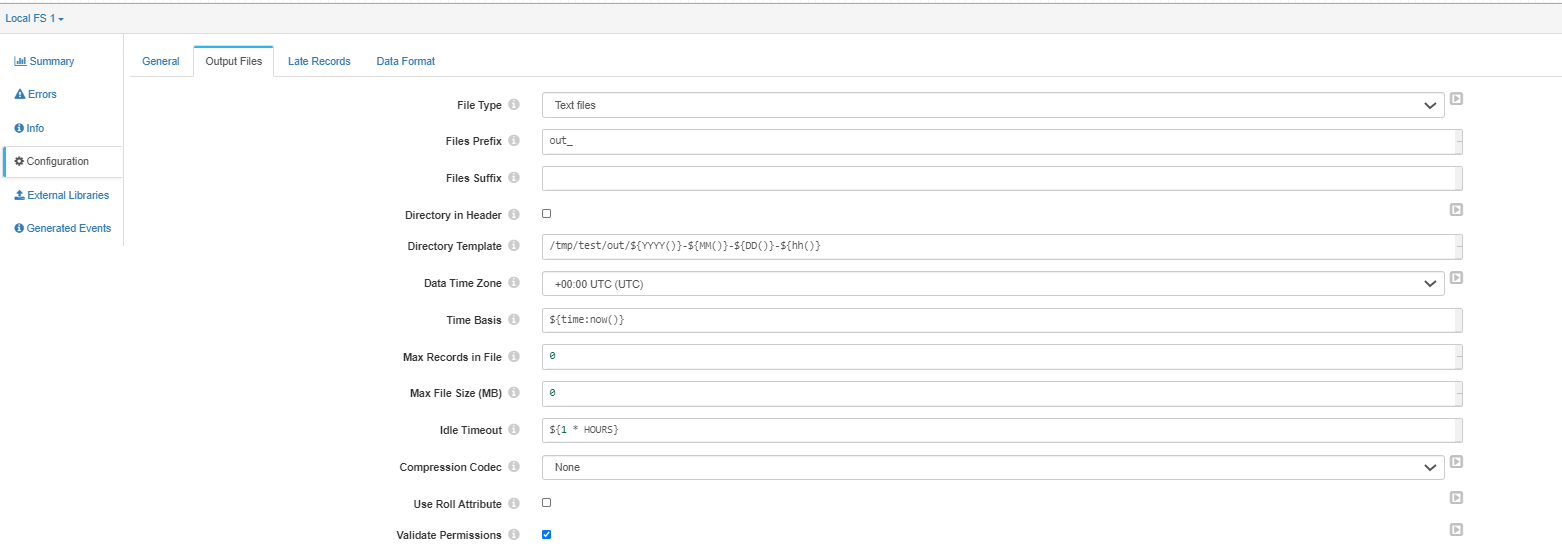
Specify table name “weather” and Default Operation “INSERT”

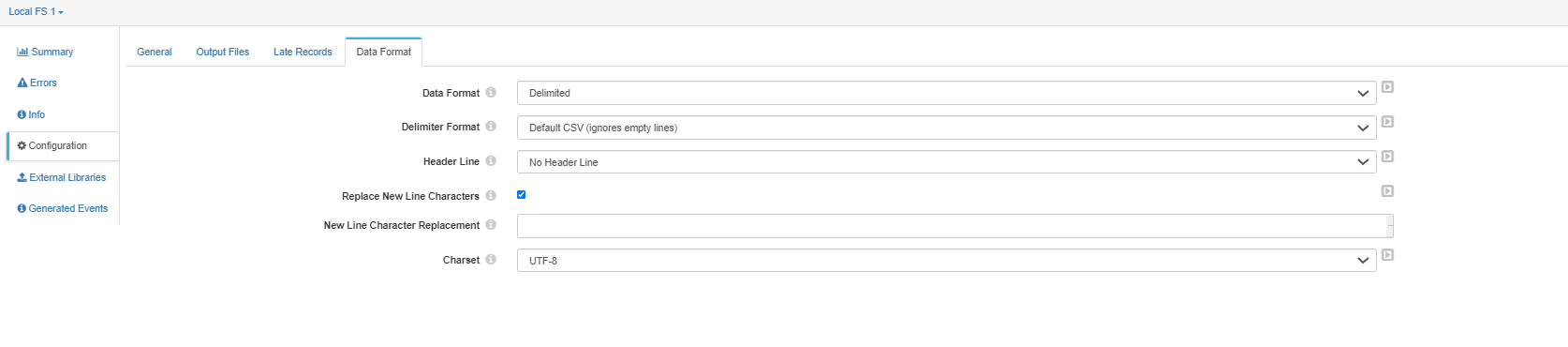


### Local FS

To store duplicates in a CSV file “Local FS” can be used

Setup the directory template, file prefix and Data format





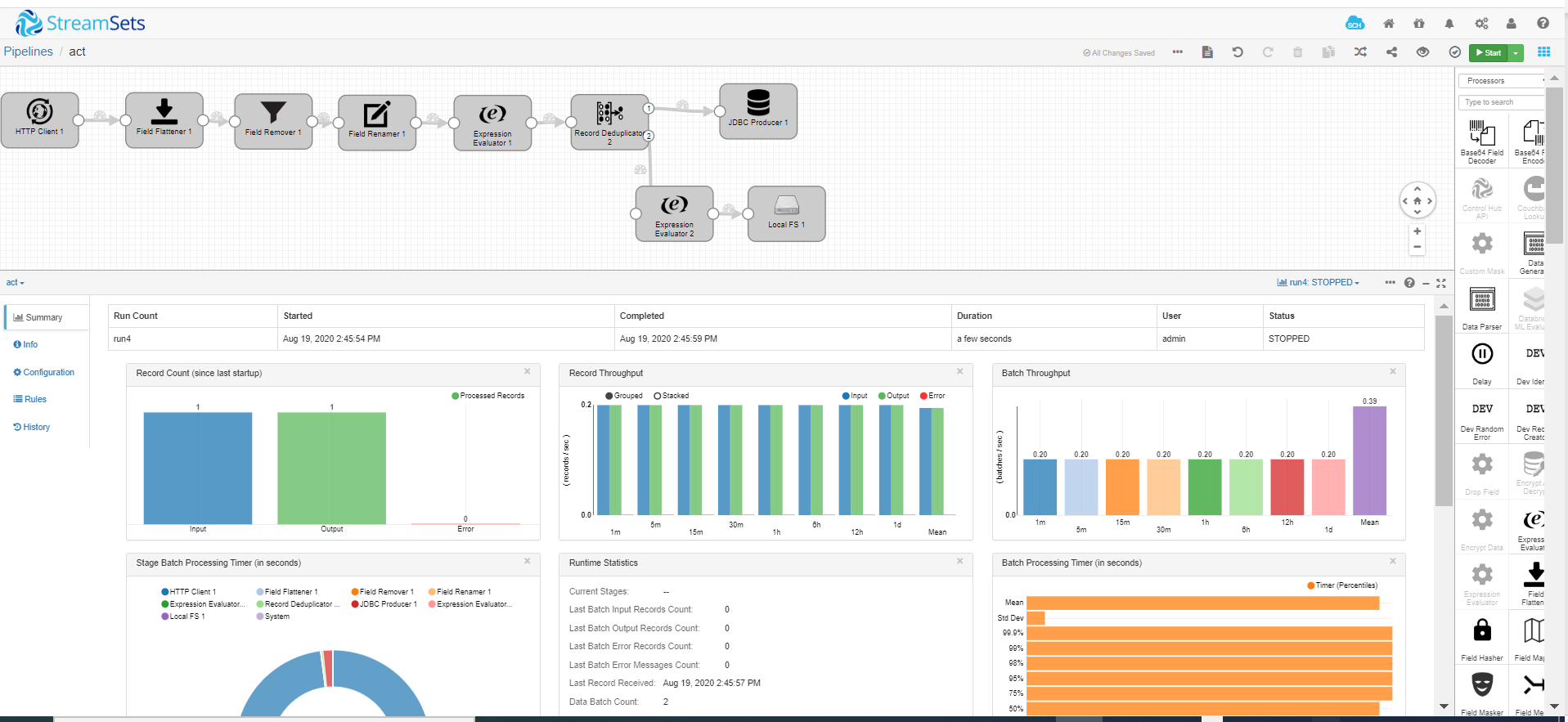
### Running the pipeline.

Once all stages are added, make sure all ins and outs are bonded with arrows correctly and test the pipeline using “Preview” (as specified in Field Flattener section).

Once all errors are fixed (if any) press “Start” button and let the pipeline work for some time.

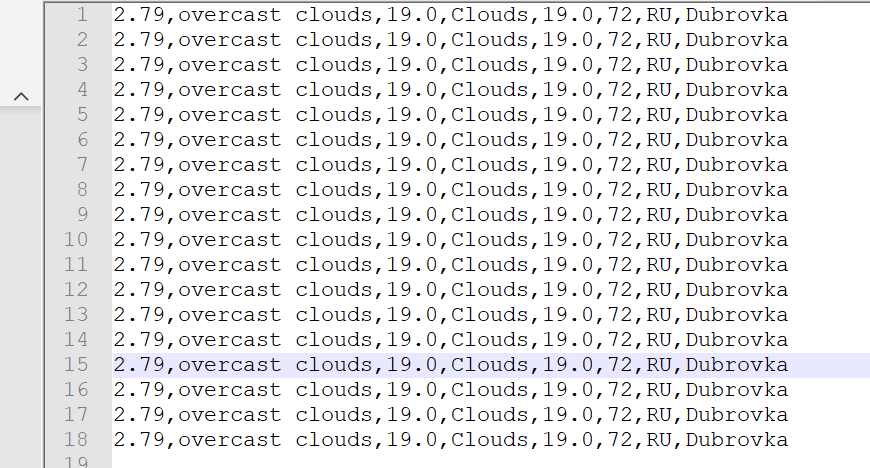
Then stop it and check the results.

Below the pipeline you will see various execution statistics as on the screenshot.



Check the existence of CSV file with duplicates in the directory that you’ve specified. Copy the file to your local drive.

The output should look like this.



Log in to MariaDB with phpMyAdmin (runs in docker image see environment preparation section) or use any other tool to connect to the database (e.g. DBeaver or any other)

The data should look like this

