Data Pipeline Project

Soheila Sadeghiram

Sl.sadeqi@gmail.com
soheila.sadeghiram@contactenergy.co.nz

Sunday 26th June, 2022

Contents

1 Introduction

In this project, I designed a data pipeline that gets data from an online database, populates required fields, and stores into an s3 bucket on AWS.

I use AWS Lambda for this project. After creating the script in Python, I deploy it to AWS Lambda. The lambda function will execute the logic and end.

Instead of using AWS Lambda, there is also another method to complete this project, which is using a cron job. Both methods have been explained in Section 4.

2 Inputs and Outputs

This original data includes the following fields: id, date_time, year, month, mdate, day, time, sensor_id, sensor_name, hourly_counts.

There are two instances of the data. The main dataset has 4320533 records:

`https://data.melbourne.vic.gov.au/api/views/b2ak-trbp/rows.csv?accessType=DOWNLOAD&api_foundry=true`.

As a default in my Lambda function, I have used an instance of the main dataset which was present at:

`https://data.melbourne.vic.gov.au/resource/b2ak-trbp.csv`

which is a smaller csv file with only 1000 records, and therefore, has a faster execution time. Using this data, you can run the project in only few minutes to see how it works. Note that you can easily replace the link in the python script (lambda_function.py) with the main data link. Instructions are given in the python script as well.

As an output, an S3 bucket will be created. I have used soheila_s3_bucket as the default name; however, it could be easily customised to get the bucket name from the command line before creating any buckets (can be done upon request).

After the project is executed, two output files will be created and stored in the bucket: a) Top_10_month.csv

This includes: *row_id*, *year*, *month*, *sensor_id*, *sensor_name*, *pedestrain_counts* and *rank*. *row_id* is the unique incremental row identifier. *pedestrian_counts* is the total number of pedestrian for that combination of *year* and *month* recorded by the sensor. *rank* shows the rank of the sensor (suburb) within the group of year and month.

b) Top_10_day.csv This is similar to Top_10_month.csv except that it shows the pedestrian counts per day.

3 Files Included in Project Folder

1. **Lambda_function.py**: the python script which extracts the hourly number of pedestrian data from the corresponding Website. This data is for Melbourne suburbs, obtained by sensors. The path to this file is lambda_s3/datapull/lambda_function.py.

   Further, the script transform the data so that we get the top 10 suburbs with the highest pedestrian count per month and per day separately. Each function has been explained in the script.

   Finally, the script uploads results to S3.

2. **setup_infra.sh**: this is a bash script where we do the following tasks:

   (a) Packages our code

   (b) Creates a bucket (soheila_s3_bucket) that I use to store the top 10 suburbs with the most pedestrian per day and month.

   (c) Creates a Policy

   (d) Creates a Role

(e) Attaches the Policy from step 3 to the Role from step 4

(f) Sleep for 15 s to allow the attachment to complete

(g) Uploads my code (lambda_function.py) and creates a lambda function, with a max run time of 60 min

(h) Sets up a schedule for the lambda function to run every 60 minutes

(i) Writes out the output of each command to a file called setup.log in the directory.

3. **tear_down_infra.sh**: this script is to tear down the infrastructure that was set up using setup_infra.sh. Make sure to use the same bucket name which in my is soheila_s3_bucket (as mentioned above, it can be customised easily.)

4. **Lambda_function_local.py**: this is an extra script which extracts and transforms data, but stores the results locally in files *out_month.csv* and *out_day.csv*. The default source for this script is the main dataset with 4M records. Aside from the Lambda method explained earlier, there is another way to perform this project, which is to set a *cron* job in the Unix terminal. Note that you will have a limited time for Lambda to complete (the current maximum execution time for Lambda is 15min). Although I did not experience that problem in the steps using Lambda, there is always a possibility, in particular when working with larger datasets, that your Lambda times out.

   The reason I included this function in the project was that I am going to implement the second method (cron job) as well instead of using Lambda. I will happily be able to complete this task, upon request, within one more day.

4 Executing the Project

To execute the data pipeline and see the output in the corresponding bucket, please follow the steps below:

1. Download and lambda_s3 folder to your computer. The download link is on GitHub:

2. Open a command line interface and navigate to lambda_s3 folder. Afterwards, run the commands below one by one:

   chmod 755 ./setup_infra.sh

   ./setup_infra.sh soheila_s3_bucket

   The first command is to grant permission to setup_infra.sh and the second one is to run setup_infra.sh.

3. Monitor the runs: you can monitor the lambda runs by going to AWS Cloudwatch UI. You can also check the files in your soheila_s3_bucket using the command below:

   aws s3 ls s3://soheila_s3_bucket/ –recursive

4. tear_down.sh: finally, run the command below to remove all the settings created by setup_infra.sh:

   chmod 755 tear_down_infra.sh

   ./tear_down_infra.sh soheila_s3_bucket

5 Some other Ideas

As mentioned in the previous section, I was using a cron job to do the same project but in a different way. This can also be completed if needed (e.g., when the Lambda function times out for the large dataset). At the moment, my project ran flawlessly using the default settings.

Another Idea that I loved to implement but could not catch due to time limit (which might not be very related to the data engineering role either but usually required in any organisation) was to visualise the output files in Power BI. I believe that the outcome will be very interesting for businesses due to the nature of this data. For example, there can be filters on the top n records, days or months. I had created a Power BI dashboard for another project at my current workplace where the data seemed very similar to your data. These are shown in Figures 1 and 2.