**Belong Data Engineer Coding Exercise**

This exercise seeks to use an Amazon Web Services, specifically S3 (for storage of the CSV data) as well as Athena (Analytical SQL tool) to create a database for Pedestrian Count system in Melbourne from 2009 to current for querying. The Python codes found in the repository utilises AWS SDK, Boto3, to make use of the Amazon services through scripting.

**Data sources used:**

* [**https://data.melbourne.vic.gov.au/Transport/Pedestrian-Counting-System-2009-to-Present-counts-/b2ak-trbp**](https://data.melbourne.vic.gov.au/Transport/Pedestrian-Counting-System-2009-to-Present-counts-/b2ak-trbp)
* [**https://data.melbourne.vic.gov.au/Transport/Pedestrian-Counting-System-Sensor-Locations/h57g-5234**](https://data.melbourne.vic.gov.au/Transport/Pedestrian-Counting-System-Sensor-Locations/h57g-5234)

**The extracted stats expected:**

1. Top 10 (most pedestrians) locations by day
2. Top 10 (most pedestrians) locations by month

Below is the process to extract statistics and allow for further queries to be run:

**Approach**

The datasets contain hourly pedestrian counts since 2009 to present from sensor devices around Melbourne CBD. The data is collected monthly and the two datasets can be merged or joined by Sensor ID columns to provide more information such as location and status from additional set.

To query selected data to extract the statistics that are expected, it was chosen to use AWS S3 web interface to store the data in a bucket. This was all downloaded and uploaded through a script using boto3 SDK for Python.

The data is then connected using AWS Athena which is a serverless interactive query system. This works on the CSV data that is structured and stored in S3 buckets. It also uses the standard SQL language. As this is a small data set, no servers are needed to perform the querying. The AWS Athena console or Lambda scripting console could be used here to automate or interact with the data, however, boto3 SDK in python was again utilised locally to interact with the service as this data is only updated every month.

Once the tables are created in Athena with our data in S3 bucket the data is queried to extract expected statistics mentioned above. This is again done locally with boto3 and the JSON results are wrangled into a CSV file that is outputted in ‘/results’ of the script folder for review.

**OVERVIEW**

* **SIMPLE ARCHITECTURE**

REPORTING

TRANSFORM

INGEST

Boto3 queries.py Python Script

1.

Boto3 schemas.py Python Script

1.

Boto3 injest\_s3.py Python Script

1.

QUERIED CSVs - /results

Source 1

Source 2

* **DATABASE DESIGN - DBD**

Graphical user interface, application

Description automatically generated

**How to run the application?**

1. You need the all the requirements in your python environment to run the application.
   1. To install all the packages required run the command line from the cloned repository folder where requirements.txt is located

*pip install -r requirements.txt*

Alternatively install boto3: pip install boto3   
 and pip install requests

1. To run the application on the test account please unzip the provided secrets.py file and replace the holding secretys.py file in the base file location.
   1. If this is created on a new AWS a range of alternative steps are required. See below for steps:
      1. Set up a new AWS account
      2. Create a new AWS S3 bucket. Save the name and replace the bucket names in all at the Python scripts in the base folder
      3. Create a new AWS IAM credentials for the root user to allow programmatic access for amazon services. Make sure the credential has full access linked to the user.
         1. Copy the users ‘access key’ and ‘secret access key’ and replace the holding values in Python script secret.py with the new keys.
2. In a command prompt at the base of the repository folder run the first python script with the command  
     
    python 1\_ingest\_s3.py

Running this script will take several minutes as it will download the latest pedestrian counts by hour as a CSV. It will then upload this file to AWS S3 bucket. When all is completed, it will return a “### Complete! ###” statement. This data is only updated every and as such only requires updating as needed.

1. In a command prompt run the second script  
     
    python 2\_athena\_schemas.py

This will create all the tables in AWS Athena using the data downloaded and stored in S3.

1. In a command prompt run the last script

Python 3\_athena\_queries.py

This script will run queries to a default database for each day and month returning a range of queried results that are wrangled into a new CSV file in file location /result. The layout of the results can eaily be changed with a new query and modified for new row layouts

Please contact me for any questions of the code and any modifications that are required.