DataEng: Data Maintenance In-class Assignment

This week you will construct a data archiver that compresses, encrypts and stores pipelined data into a low-cost, high-capacity GCP Storage Bucket.

Submit: Make a copy of this document and use it to record your responses and results (use colored highlighting when recording your responses/results). Store a PDF copy of the document in your git repository along with your code before submitting for this week.

Develop a new python PubSub subscriber similar to the subscribers that you have created multiple times for this class. This new subscriber (archive.py) will receive data from a PubSub topic, compress the data, encrypt the data and store the resulting data into a GCP Storage Bucket.

A. [MUST] Discussion Questions

When archiving data for a data pipeline we could (a) compress, (b) encrypt and/or (c) reduce the data. Here, “reducing the data” refers to the process of interpolating or aggregating detailed data, such as 5 second breadcrumbs for all buses on all trips, into coarser data. For example, we could aggregate 5-second breadcrumbs into 30-second breadcrumbs.

Under what circumstances might each of these transformations (compress, encrypt, reduce) be desirable for data archival?

Would it make sense to combine these transformations?

Record your responses here and then discuss them with your work group.

Compression: for reducing storage space and speeding up data transmission. Useful when dealing with large volumes of data.

Encryption: for ensuring data security and confidentiality, especially when dealing with sensitive or personal information.

Data Reduction: for optimizing storage and improving processing efficiency, particularly when retaining only essential information is sufficient for analysis.

Combining these transformations can be beneficial. For instance, compressing encrypted data can minimize storage requirements while maintaining security. Similarly, reducing data before encryption can enhance efficiency and security simultaneously.

B. [MUST] Create Test Pipeline

Create a new PubSub topic called “archivetest” or something similar. Create a new subscriber program (call it archiver.py) that subscribes to the topic, receives the data and (for now) discards it.

To produce test data, copy/reuse the publisher program used for your class project, and alter it to publish to the new archivetest topic. Run this test publisher manually to gather data (1 day and 100 vehicles) from busdata.cs.pdx.edu and test the archivetest topic and your archiver.py program.

As always, you can/should test your code with smaller data sets first. Try it with just one bus or one trip, and then when everything is working, run it with 100 vehicles.

C. [MUST] Store Data to GCP Storage Bucket

Modify archiver.py to store all received data to a GCP Storage Bucket. You will need to create and configure a Storage Bucket for this purpose. We recommend using the Nearline Storage class for this assignment though you are free to choose any of the offered classes of service. Be sure to remove the bucket at the end of the week to reduce GCP credit usage.

How much bucket space (in KiBs) does it take to store 1 day of breadcrumbs for 100 vehicles?

81552.08 KiBs

D. [SHOULD] Compress

Modify archiver.py to compress the data before it stores the data to the storage bucket. Use zlib compression which is provided by default by python. How large is the archived data compared to the original?

How much bucket space (in KiBs) does it take to store the compressed data?
55321.77 KiBs

E. [SHOULD] Encrypt

Modify archiver.py to encrypt the data prior to writing it to the Storage Bucket. Your archive.py program should encrypt after compressing the data. Use RSA encryption as described here: link
There is no need to manage your private encryption keys securely for this assignment, and you may keep your private key in a file or within your python code.

Be sure to test your archiver by decrypting and decompressing the data stored in the Storage Bucket. We suggest that you create a separate python program for this purpose.

How much bucket space (in KiBs) does it take to store the encrypted, compressed data?
92245.64 KiBs