Event-Based Data Pipeline in GCP

Business Overview

There is a continual need to give huge volumes of data to teams in a data-driven company. Many tools are available to assist you with your requirements and wants. Choosing the appropriate mechanism may be complicated and overwhelming at times. The key idea to remember is that there is no one-size-fits-all tool or architecture, and it all depends on your needs.
For data ingestion, many businesses are opting for Event-Driven Architecture (EDA). EDA is a software design paradigm used in application development. It enables enterprises to track and recognize significant business moments and then act on them immediately. EDA differs from a conventional request-driven system in that services must wait for a response before moving on to the next job.
Many application architectures are increasingly event-driven in a modern economy fuelled by high digital transaction volume.
These pipelines can provide the agility, scalability, context, and responsiveness required for performant digital business applications. The event-driven design also needs low coupling, making it a suitable match for distributed application architectures.
There are producers and consumers in event-driven architecture. Producers unearth events and craft a message. The event is subsequently transmitted to event consumers through an event channel, where it is processed. An event processing system then performs a response to the event message, which results in creating an activity downstream.
This project will create an event-driven data pipeline using the Google Cloud Platform's serverless features. It will be based on some of the development methods that are frequently used in businesses.

Dataset Description

The dataset used in this project will be a Covid-19 dataset(COVID-19 Cases.csv) from data.world, which consists of a few of the following attributes:

- people_positive_cases_count
- county_name
- case_type
- data_source

Tech Stack
➔ Language: Python3.7
➔ Services: Cloud Composer, Google Cloud Storage (GCS), Pub-Sub, Cloud Functions, BigQuery, BigTable

Cloud Composer
Cloud Composer is a workflow orchestration service based on Apache Airflow that is completely managed. Cloud Composer pipelines are easily constructed as directed acyclic graphs (DAGs) using Python. One-click deployment provides quick access to an extensive library of connectors and numerous graphical representations of your process in operation, making debugging a breeze. Your jobs will stay on track if your directed acyclic graphs are automatically synchronized.

Google Cloud Storage
Google Cloud Storage is a web-based file storage service that allows you to store and access files on the Google Cloud Platform infrastructure. The service combines Google's cloud's speed and scalability with sophisticated security and sharing features. It provides Infrastructure as a Solution (IaaS), similar to Amazon's S3 online storage service.

Pub/Sub
To ingest and disseminate data, Pub/Sub is utilized in streaming analytics and data integration pipelines. It works equally well as messaging-oriented middleware for service integration or as a queue to parallelize operations.
Pub/Sub allows you to construct event producers and consumers systems, known as publishers and subscribers. Publishers connect with subscribers asynchronously by disseminating events rather than synchronous remote procedure calls (RPCs). It is comparable to Apache Kafka.

Cloud Functions
Google Cloud Functions is a serverless execution environment used to develop and connect cloud applications. Cloud Functions allows you to build simple, one-time functions related to events generated by your cloud infrastructure and services. When an event being monitored fires, your function is called. Your code runs in a completely controlled environment. There is no need to set up Infrastructure or manage servers.

BigQuery
BigQuery is a fully managed corporate data warehouse that lets you collect and analyze your data with built-in capabilities such as machine learning, geographic analysis, and business analytics. BigQuery's serverless design enables you to do research using SQL queries while managing no infrastructure. The scalable, distributed analytical engine in BigQuery allows you to query terabytes in seconds and petabytes in minutes.

BigTable

Cloud Bigtable is a sparsely filled table with the ability to expand to billions of rows and thousands of columns, allowing you to store terabytes or even petabytes of data. Each row has a single value that is indexed, and this value is known as the row key. Bigtable is suitable for storing massive volumes of single-keyed data with extremely low latency. It has a high read and write throughput with minimal latency, making it an excellent data source for MapReduce processes.

Key Takeaways

- Understanding the problem statement
- Understanding the Project Architecture
- Overview of Dataset
- Installing and setting up Cloud SDK
- Installing Dependencies
- Understanding Pub/Sub
- Introduction to BigQuery
- Creating and Publishing to Pub/Sub topics
- Using Cloud Functions to transfer Data
- Introduction to Cloud Composer and Airflow
- Creating Composer instance and GCS buckets
- Creating SSH and SFTP connection from Composer
- Transferring data from VM to GCS
- Understanding Apache Beam with Dataflow Runner
- Introduction and Working with BigTable
- Analyzing different console metrics
- Creating Dashboards on Cloud Studio

Architecture