Image Processing Data Architecture

By Andri Renardi Lauw

Client Requirements

1. Able to retain image data for at least 7 days
2. Able to have business intelligence on key statistics for the image data (e.g. number and types of image processed, by which customer, etc)

Client Current State

The client’s team currently have

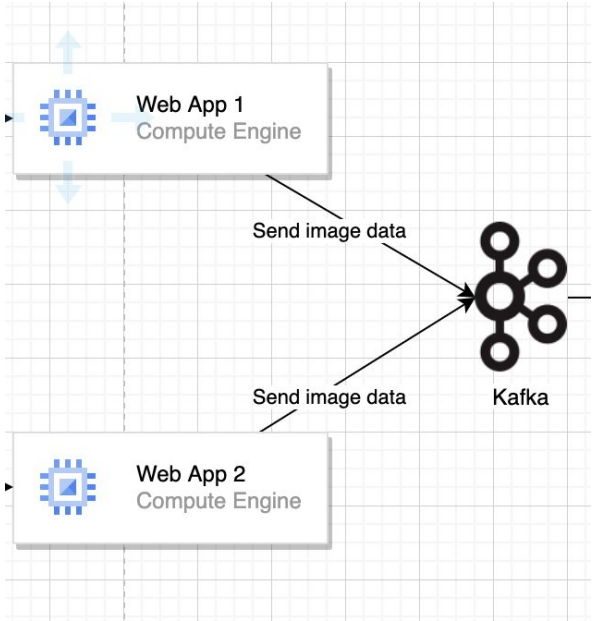
1. Web application to collect customer uploaded images
2. Web application to stream images via Kafka Stream
3. Image processing logic

Assumptions

Based on the client's current state, we can assume several things:

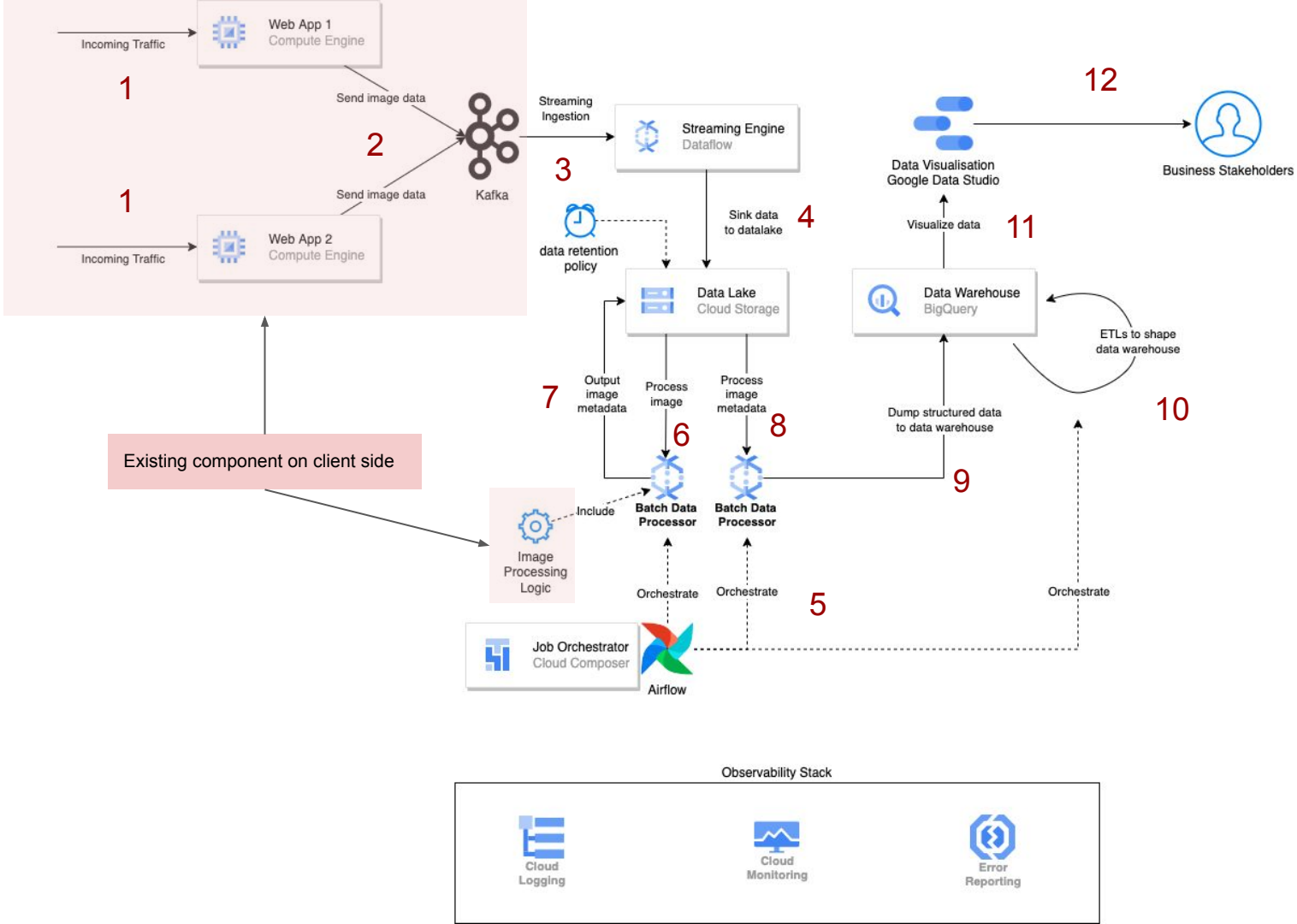
1. We assume **the client use Google Cloud Platform (GCP) as its cloud service provider**, although used minimally
2. **The web applications are up and running**, hosted in some compute machines. **For now we will assume they are hosted in Google Compute Engine (GCE)**, which is nothing more than a Virtual Machine
3. Since one of the web applications use Kafka stream to stream image data, **we assume a Kafka cluster is also up and running**. **We assume it is hosted in GCE as well**
4. The image processing logic exist, but hasn't been deployed elsewhere and only used in ad-hoc manner

Assumptions



Proposed Architecture

1. We will leverage more cloud services to build the complete data architecture to satisfy the client's requirement
2. Some notable components:
    a. Google Cloud Dataflow for hosting streaming and batch jobs to process data
    b. Google Cloud Storage for main data lake storage
    c. Google Cloud Bigquery for data warehouse storage and processing engine
    d. Google Cloud Composer, managed Airflow service for ETL jobs orchestrator
    e. Google Data Studio for data visualisation
3. Observability Components
    a. Google Cloud Logging for logging solution
    b. Google Cloud Monitoring for monitoring solution
    c. Google Cloud Error Reporting for alerting solution


Incoming Traffic
Web App 1
Compute Engine
1
Send image data
2
Incoming Traffic
Web App 2
Compute Engine
1
Send image data
Kafka
Streaming Ingestion
Streaming Engine
Dataflow
3
Sink data to datalake
4
data retention policy
Data Lake
Cloud Storage
Data Warehouse
BigQuery
Visualize data
11
Data Visualisation
Google Data Studio
12
Business Stakeholders
ETLs to shape data warehouse
10
Output image metadata
7
Process image
6
Process image metadata
8
Dump structured data to data warehouse
9
Existing component on client side
Include
Batch Data Processor
Batch Data Processor
Image Processing Logic
Orchestrate
Orchestrate
5
Orchestrate
Job Orchestrator
Cloud Composer
Airflow
Observability Stack
Cloud Logging
Cloud Monitoring
Error Reporting

System Flow

We are now going to explain the system and data flow based on the cue number attached in architecture diagram in the previous slide

1. External traffic hit the web applications (e.g. customers uploading images to the web applications)
2. Web applications send image data to Kafka
3. Streaming job which is hosted in Google Cloud Dataflow will consume the data from Kafka by subscribing to the corresponding Kafka topics
4. Streaming job dump the image data to Google Cloud Storage. Data is sorted and partitioned based on certain property (e.g. ingestion timestamp)

System Flow

5. Google Cloud Composer, which is a managed Airflow job orchestrator, will orchestrate a periodical batch job to process data. This step will be responsible to schedule step 6 to step 10

6. A batch job, which incorporate existing image processing logic, written in Apache Beam and hosted in Google Cloud Dataflow for serverless distributed compute engine, will process the image data and produce information regarding the image in periodical manner

7. The image metadata is stored back in Google Cloud Storage alongside the corresponding image in a structured manner

System Flow

8. Another batch job hosted in Google Cloud Dataflow will fetch and process these image metadata

9. The image metadata is processed and stored in Google Cloud Bigquery as a data warehouse in a structured fashion in a tabular format

10. A subsequent Bigquery jobs will be invoked to further process the data, cleaned and shaped properly inside the data warehouse, and deliver it to data mart where it can be consumed easily by the BI team

11. BI team can leverage Google Data Studio to create visualisation from the data mart in Bigquery, providing insight to business stakeholders

12. Business stakeholders can consume the insight generated by BI team via visualization in Google Data Studio

Pros and Cons

Pros:

+ Most of the components are serverless, which means less effort and resource to operate and maintain the operations, the infrastructure work are offloaded to GCP team which is world class standard.
+ Fully scalable and highly available system end-to-end, which can adapt quickly to various volume of data traffic and workloads.
+ Since most infrastructure work are offloaded to cloud provider, the team can focus mainly on the developing the components which matter to business.
+ The project can be done on a lean team structure.
+ Pay only what you use, and scale up only when needed.
+ No change to existing components. The data architecture extends the existing components.

Pros and Cons

Cons:

- Need to be familiar with GCP services which comes with a certain learning curve.
- Leveraging cloud services means you have to be very diligent on maintaining cost efficiency, since any excess resource will come into your billing.
- Leveraging native cloud services such as Bigquery also means you are vendor lock-in. Means that in a very unlikely event that Google went out-of-business, there’s gonna be significant migration and re-architecture work.