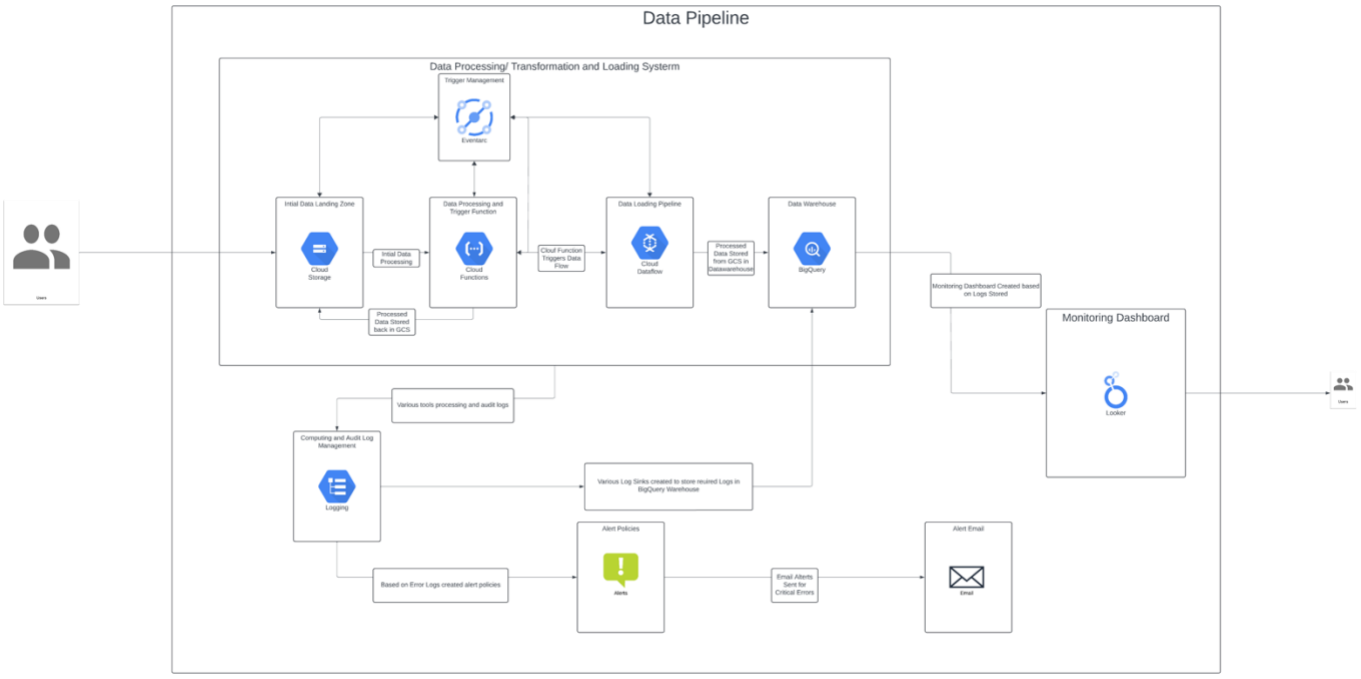
1. Project Overview

This document outlines the development of a data monitoring system for an Extract, Transform, Load (ETL) pipeline. The system is designed to extract data from a Google Cloud Storage (GCS) bucket, transform it, and subsequently load it into a Google BigQuery table. The solution incorporates a monitoring dashboard and an email alerting mechanism to address any operational issues within the pipeline.

2. System Architecture

- Data Source: Google Cloud Storage
- ETL Process: Managed by Google Cloud Functions
- Data Loading: Facilitated by Google DataFlow
- Data Warehousing: Implemented via Google BigQuery
- Monitoring: Conducted through Looker
- Alerting: Managed by Google Cloud Monitoring with notification capabilities



3. Overview of Design

3.1 Data Extraction

- Tools Used: Python scripts leveraging Google Cloud libraries
- Process: Scripts are event-triggered and extract CSV files from a designated GCS bucket.

3.2 Data Transformation

- Tools Used: Python, utilizing the Pandas library for data manipulation.
- Process: Scripts transform data by cleansing, filtering, and structuring it for loading into BigQuery.

3.3 Data Loading

- Tools Used: Google DataFlow using legacy templates
- Process: Transformed data is batch-loaded into BigQuery, optimizing API call efficiency and performance.

4. 4. Monitoring Dashboard

- Tools Used: Looker Studio for reporting.
- Design: Dashboards are designed to display key metrics such as load times, data volumes, and operational efficiency.

5. Alerting System

- Tools Used: Google Cloud Monitoring for anomaly and failure detection, with event-triggered notifications to initiate email alerts.
- Process: Utilizes custom metrics and logs to inform stakeholders of issues via email, facilitated inbuilt notification system.

6. Future Scope Criteria

6.1. Enhancements:

- Integration of multiple data sources to enhance the robustness of the data warehouse.
- Implementation of Google Cloud Composer to utilize Airflow for scheduling and managing jobs, enabling real-time or periodic data streaming.
- Development of modular, systematic triggers to enhance pipeline execution reliability.

7. Learning from the coding challenge

- Exploration of Google Cloud Platform (GCP) and its array of tools for robust data pipeline creation.
- Utilization of legacy templates for efficient data loading.
- Recognition of the importance of environment variables in cloud platforms.
- Insights on cost reduction and simplification of setup by using tools within the same platform.

8. Conclusion

This system has been meticulously designed to be robust, scalable, and efficient, utilizing the comprehensive data management and monitoring services provided by Google Cloud. The pipeline effectively addresses all the requirements of the coding challenge, demonstrating its capability to handle complex data processing tasks with high efficiency.