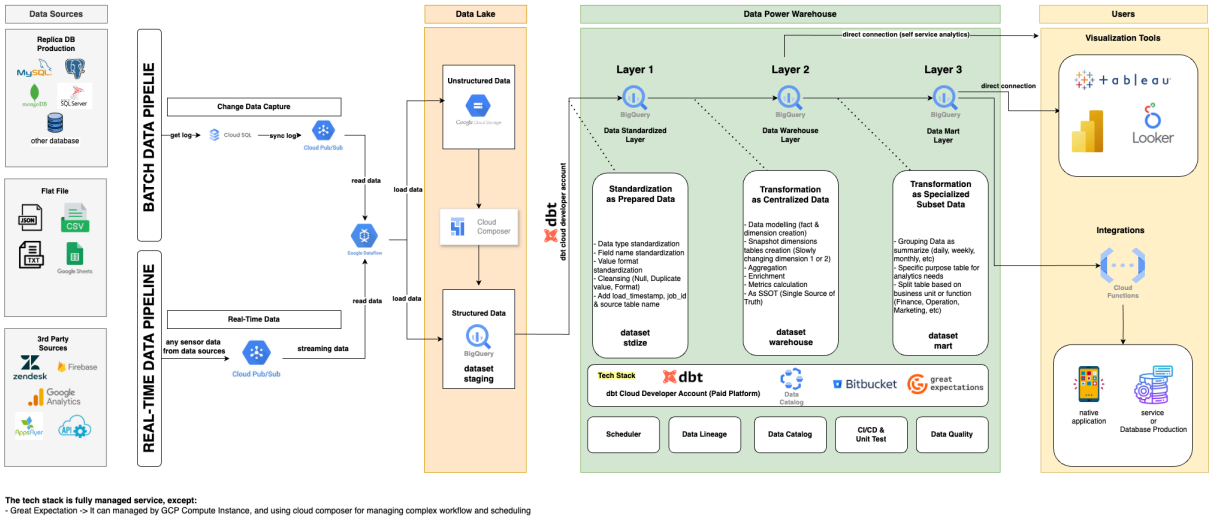
Name : Noverdy Safrizal

Data pipeline Architecture

1. Fully Managed Service Tech Stack

End-to-End Data Pipeline Architecture using any Fully Managed Services Tech Stack



Link [here]

Detailed Explanation

1. The data pipeline process is categorized into two types: batch processing and real-time data processing.
    a. Batch Processing:
        - Utilizes Cloud SQL audit logs to read logs from data sources.
        - Pub/Sub is used to synchronize logs across topics, which are defined per table.
        - Dataflow then reads data from Pub/Sub, loads it, and stores it either in Google Cloud Storage (GCS) for unstructured data or structured data in BigQuery.
    b. Real-time Event Processing:
        - Uses Pub/Sub and Kubernetes to stream event data per log activity.
        - Dataflow serves as the pipeline orchestrator, facilitating storage in BigQuery and GCS.
2. Data Orchestration & Transformation
   Cloud Composer is employed as an orchestration tool for ingesting files from GCS to BigQuery's staging layer.
3. ELT Process. Data transformation is managed using DBT Cloud, which includes a built-in scheduler and is able to integrate with Bitbucket for CI/CD pull request processes. Data modeling is performed at each layer, ensuring separation based on

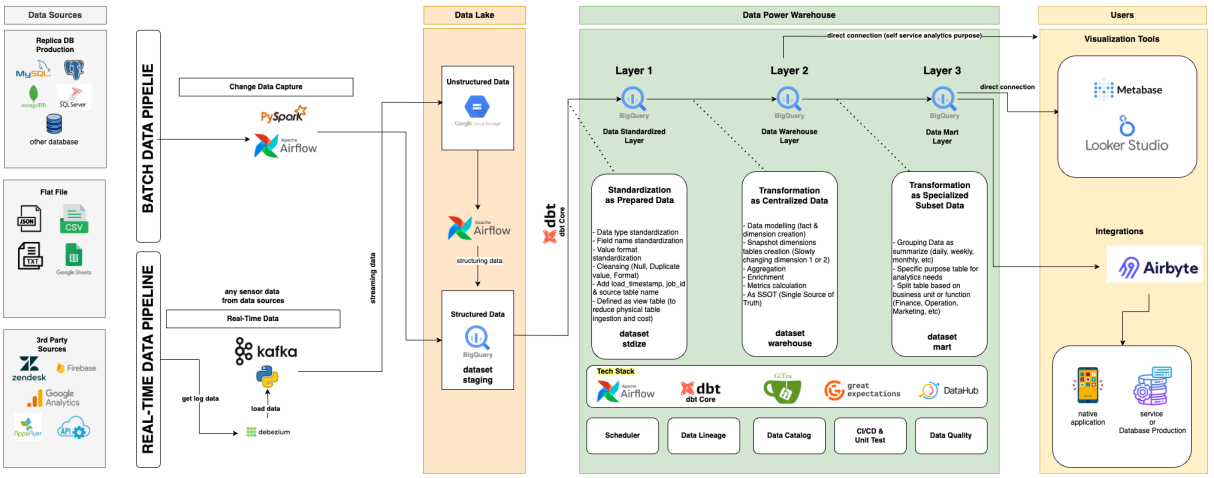
the details outlined above. The transformation process incorporates key data warehouse best practices, including:

Data Lineage	Helps users, particularly BI teams, explore data source paths when generating insight reports or conducting ad-hoc analysis. ***Tech stack: DBT***
Data Catalog	Facilitates data exploration by providing column definitions, data types, and identifying PII (Personally Identifiable Information), leading to improved efficiency and a better customer experience. ***Tech stack: cloud data catalog***
Data Quality	Ensures data governance through data quality checks, which are critical for maintaining high standards. Great Expectations is one of the tools used for data quality validation, which can be deployed on GCP Compute Engine or Cloud Composer as the base code for quality checks and scheduling. ***Tech stack: Great Expectation***
CI/CD & Unit Testing	Enhances BI team development workflows by ensuring data marts meet standards through unit testing before deployment to production. ***Tech stack: Bitbucket***

4. Data Accessibility & API Integration
    a. Business users can perform self-service queries on properly structured data warehouses.
    b. Data marts created by the BI team serve as open-sources for BI tools, supporting reporting and dashboarding needs.
    c. Cloud Functions are utilized to develop APIs that provide downstream applications and databases with access to warehouse data.

2. Using a Single Linux Server

End-to-End Data Pipeline Architecture Using a Single Linux Server



The tech stack is self-installed using Docker on a single Linux server.
- Debezium + Apache Kafka
- Airflow
- Python + Pyspark
- dbt Core
- GiTea
- Great Expectation
- Datahub
- Airbyte
- Metabase

Link here

Detailed Explanation

1. The data pipeline process is categorized into two types: batch processing and real-time data processing.
    a. Batch Processing
        - Batch processing is implemented using custom Python code and PySpark, orchestrated with Airflow to load Change Data Capture (CDC).
        - The ETL process includes necessary transformations, ensuring that data is well-structured as a physical table in the output.
        - The system is deployed on a server using Docker images and containers, hosting both Airflow and PySpark.
    b. Real-time Event Processing:
        - Debezium is required to detect data changes in the database.
        - Kafka stores table-level changes in separate topics, which are consumed by Google Cloud Storage (GCS) as the target storage.
2. Unstructured Data Transformation
   This process is executed using the same Airflow instance installed on the server. Custom Python code is used for the ETL process, converting unstructured data from GCS (JSON/Parquet files) into structured physical tables in the staging dataset.
3. ELT Process
   Data modeling is performed using dbt Core, connected to BigQuery. The dbt Core environment is installed on the same server and is orchestrated by Airflow as the

scheduler. The transformation process follows key data warehouse best practices, including:

Data Lineage	Helps users, particularly BI teams, explore data source paths when generating insight reports or conducting ad-hoc analysis. ***Tech stack: dbt***
Data Catalog	Facilitates data exploration by providing column definitions, data types, and identifying PII (Personally Identifiable Information), leading to improved efficiency and a better customer experience. ***Tech stack: dbt***
Data Quality	Ensures data governance through data quality checks, which are critical for maintaining high standards. Great Expectations is one of the tools used for data quality validation, which can be deployed on linux using docker and docker compose as the base code for quality checks and scheduling. ***Tech stack: Great Expectation***
CI/CD & Unit Testing	Enhances BI team development workflows by ensuring data marts meet standards through unit testing before deployment to production. ***Tech stack: GiTea***

4. Data Accessibility & API Integration
    a. Business users can perform self-service queries on properly structured data warehouses.
    b. Data marts created by the BI team serve as open-sources for BI tools, supporting reporting and dashboarding needs. Consider using paid platform BI-Tools like looker. Looker has great capability to self-service analytics.
    c. Airbyte is utilized to develop APIs that provide downstream applications and databases with access to warehouse data. That can be installed with Docker and Docker Compose.

Additional:

The entire architecture must ensure data freshness by implementing daily alerts to notify the team of any errors or issues within the pipeline. These alerts can be sent to Slack, Telegram, Google Chat, or other messaging platforms.

Additionally, monitoring tools are required to ensure that the server remains operational during data ingestion (ETL/ELT) and streaming processes. It is also essential to define thresholds for query consumption for BigQuery to manage costs effectively and keep them within the allocated budget.