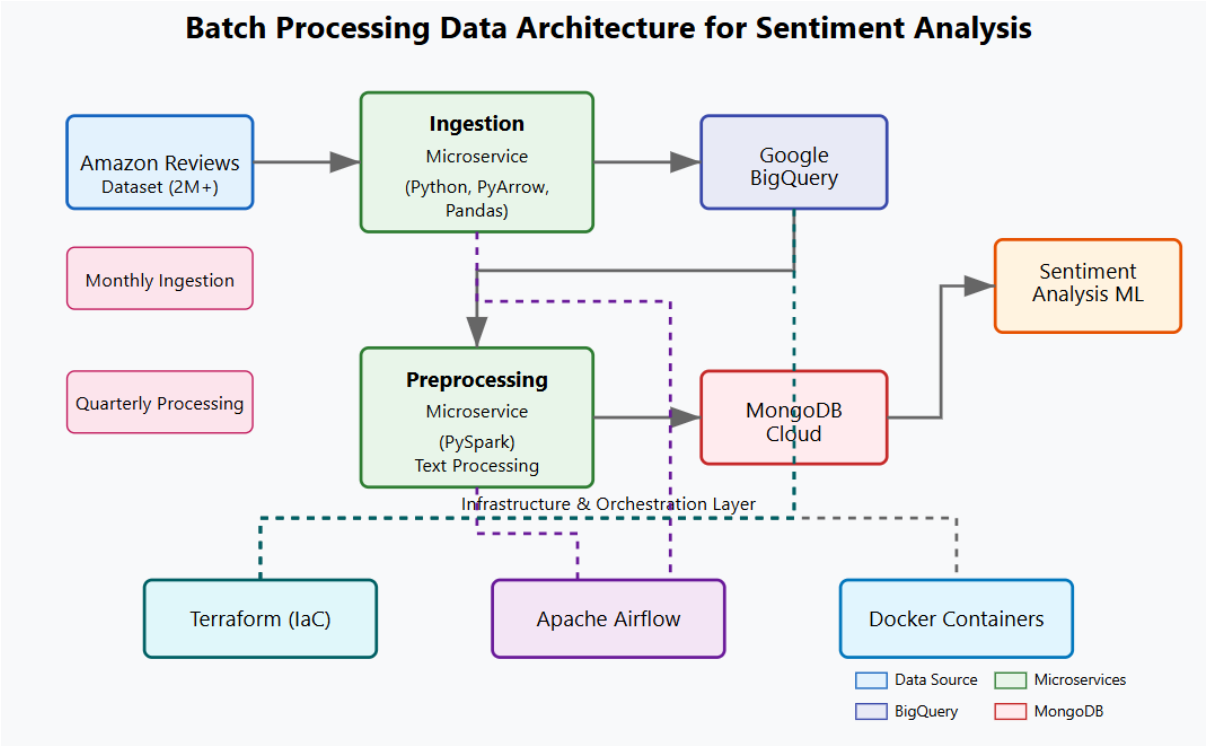
Task1: Conception Phase for a batch-processing-based data architecture for Sentiment Analysis ML Application

1. Overview

This project focuses on developing a batch-processing data pipeline for a sentiment analysis machine learning (ML) application. The system processes the Amazon Reviews dataset, which contains over 2 million records, ensuring efficient ingestion, transformation, and storage. The architecture leverages Google BigQuery for storage, PySpark for large-scale data preprocessing, and MongoDB Cloud for flexible data storage. The entire workflow is containerized using Docker and automated with Terraform for infrastructure provisioning and Apache Airflow for orchestration.

2. System Architecture

The data pipeline consists of two microservices: one dedicated to ingestion and another to preprocessing. These microservices work in tandem to ensure a seamless ETL process.



2.1 Data Ingestion

The ingestion microservice is responsible for extracting raw data from a local server and storing it in Google BigQuery. It is implemented using Python, PyArrow, and Pandas, allowing efficient handling of data before storage. Terraform is utilized to automate infrastructure deployment, ensuring reproducibility and ease of scaling.

2.2 Data Preprocessing

The preprocessing microservice retrieves data from BigQuery, applies transformations using PySpark, and saves the cleaned dataset into MongoDB Cloud. This step ensures data

consistency and prepares it for further ML model training. The processing pipeline is orchestrated with Apache Airflow, which schedules batch jobs at predefined intervals.

2.3 Data Storage and Delivery

- Google BigQuery is used for storing raw and structured data due to its scalable and efficient querying capabilities.
- MongoDB Cloud serves as the destination for processed data, offering flexibility for machine learning applications to retrieve structured and semi-structured information.

2.4 Reliability, Scalability, and Maintainability

To ensure high system reliability, Airflow task retries, logging mechanisms, and fault-tolerant workflows are integrated. Scalability is achieved through serverless computing in BigQuery and distributed processing with PySpark. Maintainability is enhanced through modular microservices, Docker containerization, and version control using GitHub.

2.5 Security, Governance, and Protection

Security measures include IAM roles for access control, TLS/SSL encryption for data in transit, and MongoDB authentication policies. Audit logs from BigQuery, MongoDB, and Airflow ensure governance and compliance with data protection standards. Data governance practices include schema validation, metadata management, and role-based access control.

2.6 Containerization and Infrastructure Automation

Both microservices are containerized using Docker, ensuring consistent deployments across environments. Terraform provisions cloud infrastructure, automating the setup of BigQuery datasets and MongoDB storage. Pre-built Python and Apache Spark Docker images are customized with required dependencies to optimize performance.

2.7 Processing Frequency

The system follows a batch-processing approach, with data ingestion scheduled monthly and preprocessing executed quarterly. This ensures that updated, cleaned data is always available for ML model training.

3. Conclusion

This data pipeline is designed to be scalable, reliable, and secure, supporting the sentiment analysis ML application by providing high-quality processed data. The integration of Google BigQuery, PySpark, and MongoDB Cloud, alongside Docker, Terraform, GitHub, and Apache Airflow, ensures a structured and automated workflow. By leveraging modern data engineering techniques, this architecture provides a robust foundation for data-driven sentiment analysis.