Development Phase for a Batch-Processing-Based Data Architecture for Sentiment Analysis ML Application

1. Implementation Overview This phase focuses on translating the conception-phase blueprint into a fully functional data pipeline. The system is developed to handle large-scale sentiment analysis on Amazon Reviews data, ensuring efficient data ingestion, transformation, and storage. The development process is structured into microservices, containerized for deployment efficiency, and automated with Terraform and Apache Airflow.

2. Cloud Storage Configuration To store both raw and preprocessed datasets, I configured cloud storage solutions using Google BigQuery and MongoDB. The setup involved creating authentication files to enable automated login and seamless data access. This ensures efficient handling of large datasets and streamlined data flow between services.

3. Development Environment Setup

- Installed VS Code as the primary development environment.
- Created a structured project folder containing:
    - **Microservices** (for ingestion and preprocessing logic).
    - **Terraform** (for infrastructure automation).
    - **Airflow** (dags, plugins, and shared data for task scheduling).
    - **Documents** (for project documentation and version control).

4. GitHub Repository Initialization

- Created a GitHub repository to manage version control and collaboration.
- Initialized the Git repository within the project folder.
- Pushed the initial project structure to GitHub.

5. Microservices Development

- Developed microservices using Python, leveraging libraries such as Pandas PyArrow
- The ingestion microservice extracts data from raw sources and loads it into BigQuery.
- The preprocessing microservice cleans and transforms data with PySpark before storing it in MongoDB.

6. Infrastructure Automation with Terraform

- Created Terraform scripts to automate the provisioning of BigQuery datasets and MongoDB collections.
- Ensured repeatable and scalable cloud resource management.

7. Containerization and Orchestration

- Installed Docker Desktop to containerize the microservices.
- Created Docker images for both ingestion and preprocessing services.
- Developed a **Docker Compose** file to manage multi-container deployment.
- Integrated Apache Airflow into the workflow for scheduling and monitoring data processing tasks.

8. Execution and Workflow Automation

- Running docker-compose up initializes the batch processing pipeline.
- Airflow DAGs orchestrate data ingestion and preprocessing on a scheduled basis.
- Logs and performance metrics can be monitored in real-time within Airflow.

Conclusion The development phase successfully builds the foundation for a batch-processing sentiment analysis pipeline. The integration of BigQuery, PySpark, MongoDB Cloud, Docker, Terraform, Airflow, and Docker Compose ensures a scalable, reliable, and secure data pipeline capable of handling large-scale processing efficiently.

GitHub Repository:
https://github.com/MostafaMoeiniML/Bigquery-Pyspark-MongoDB/

Big Dataset:

https://drive.google.com/drive/u/4/folders/16qyKNRDmMrKSCbH7vv9_51Ajmy5Byf_A