Final Project: Building an End-to-End Data Engineering Pipeline

Objective:

The goal of this project is to apply the concepts and tools learned throughout the course—including Hadoop, HDFS, PySpark, Kafka, Airflow, and Superset—to build a complete data pipeline. Students will design, implement, and present an end-to-end solution that integrates various data engineering components, mimicking real-world challenges.

Project Overview

Use Case:

Retail Analytics and Reporting Platform
A retail company wants to modernize its data platform to gain insights into sales, inventory, and customer behavior. The company operates both online and offline stores and collects data from multiple sources, including:

- **Transactional Databases**: Contains sales and inventory data.
- **Web APIs**: Fetches customer reviews and product ratings.
- **CSV Files**: Weekly reports on marketing campaign performance.
- **Semi-structured Data**: JSON logs from e-commerce website interactions.

Your task is to design a pipeline that:

1. Ingests data from the above sources into a data lake.
2. Processes and transforms the data for storage in a data warehouse.
3. Implements real-time data processing for monitoring website activity.
4. Visualizes key metrics in a reporting dashboard.

Requirements:

Step 1: Data Ingestion

- **Transactional Databases**: Use tools like Python or Spark JDBC to extract data from MySQL or PostgreSQL databases. Students may either set up the databases themselves or use the provided mock data to streamline this step.
- **Web APIs**: Write Python scripts to fetch data from a public API such as Fake Store API (to simulate external API calls relevant to retail scenarios).
- **CSV Files**: Load static files into HDFS using Hadoop commands.
- **JSON Logs**: Use Kafka to simulate streaming log data from the e-commerce website.

Step 2: Data Storage

- **Data Lake**: Store raw data in HDFS in its original formats (CSV, JSON, etc.).
- **Hive**: Create external Hive tables to query the raw data stored in HDFS.

Step 3: Data Processing

- **Batch Processing**:
  - Use PySpark to clean and transform sales and inventory data.
  - Implement data aggregation (e.g., total sales by region, top-selling products).
- **Real-Time Processing**:
  - Use Spark-Streaming or Flink to process JSON logs streamed via Kafka, prioritizing metrics such as user engagement, active sessions, and error rates.
  - Calculate metrics like the number of active users and popular product views in real-time.

Step 4: Data Automation and Orchestration

- **Apache Airflow**:
  - Automate the end-to-end pipeline, including data ingestion, transformation, and loading into the data warehouse.
  - Set up DAGs with monitoring and error-handling capabilities.

Step 5: Data Visualization

- **Reporting Dashboards**:
  - Use Superset or Power BI to create dashboards for:
    - Sales trends over time.
    - Real-time website activity.
    - Inventory status and alerts for low stock.

Deliverables

1. **Technical Documentation**: Provide a detailed report explaining:
   - Data sources and ingestion methods.
   - Transformation and processing steps.
   - Orchestration workflows.
   - Visualization design.
2. **Code Repository**: Submit all scripts and configuration files in a GitHub repository.
3. **Presentation**: Prepare a 10-minute presentation showcasing:
   - Pipeline architecture.
   - Key challenges and solutions.
   - Insights derived from the data.

Dataset Details

- **Transactional Data**:
  - Source: Mock MySQL database.
  - Schema:
    - `sales`: `order_id`, `product_id`, `customer_id`, `quantity`, `total_price`, `order_date`
    - `inventory`: `product_id`, `product_name`, `category`, `stock_quantity`, `price`
- **Marketing Data**:
  - Source: CSV file.
  - Columns: `campaign_id`, `channel`, `clicks`, `conversions`, `spend`, `date`
- **Customer Reviews**:
  - Source: Use a public API such as Fake Store API.
- **Website Logs**:
  - Source: Simulated JSON data pushed to Kafka.
  - Example schema: `{"user_id": "123", "action": "view", "product_id": "456", "timestamp": "2024-01-01T12:00:00Z"}`

Tools and Technologies

- **Data Lake**: HDFS
- **Batch Processing**: PySpark
- **Real-Time Processing**: Kafka + Spark-Streaming/Flink
- **Data Warehouse**: Apache Hive
- **Orchestration**: Apache Airflow
- **Visualization**: Superset/Power BI

Grading Criteria Basis

- **Pipeline Design**: Well-structured and efficient workflows.
- **Technical Implementation**: Accurate use of tools and technologies.
- **Documentation and Presentation**: Clear and professional deliverables.
- **Innovation and Problem-Solving**: Creative approaches to challenges.

Submission Guidelines

- **Subbmission**: Upload your GitHub repository link and documentation to the course platform.