Mini Data Platform in Docker Containers – Semester Project Guide

Objective

Develop a mini data platform using Docker containers that simulates a business process, ingests data into PostgreSQL, captures changes with Debezium, streams data through Kafka, processes it in Spark and stores it in MinIO in Delta format.

Task-by-Task Breakdown & Deliverables

Task 1: Simulating a Business Process with Python

Topics

* Writing a Python script to simulate a business process.
* Reading data from three CSV files.
* Creating tables in PostgreSQL dynamically.
* Inserting data into PostgreSQL.

Deliverables

* Python script that:
  + Reads three CSV files.
  + Creates PostgreSQL tables dynamically.
  + Inserts data into PostgreSQL.
* Updated docker-compose.yml with PostgreSQL

Task 2: Connecting Debezium to Capture Changes in PostgreSQL

Topics

* Setting up Debezium to monitor PostgreSQL changes.
* Configuring Debezium as a Kafka connector.
* Enabling AVRO or JSON serialization for messages sent to Kafka.

Deliverables

* Configured Debezium service in docker-compose.yml.
* PostgreSQL configured with logical replication enabled.
* Debezium configured to use AVRO or JSON format for messages.
* Verified that changes in PostgreSQL are captured by Debezium and sent to Kafka in AVRO format.

Task 3: Kafka Setup & Streaming Events in AVRO or JSON Format

Topics

* Setting up a Kafka cluster in Docker.
* Configuring Kafka topics to use AVRO or JSON format.
* Setting up Schema Registry for AVRO or JSON message serialization.
* Writing a Kafka consumer to validate incoming messages.

Deliverables

* Running Kafka cluster in docker-compose.yml.
* Kafka configured with AVRO or JSON serialization.
* Schema Registry set up to manage AVRO or JSON schemas.
* Kafka consumer script that reads AVRO or JSON messages and decodes them.

Task 4: Integrating Spark with Kafka for Data Processing

Topics

* Introduction to Apache Spark and its role in big data processing.
* Connecting Spark Structured Streaming to Kafka.
* Deserializing AVRO or JSON messages from Kafka in Spark.
* Performing basic transformations on streamed data.

Deliverables

* Spark container added to docker-compose.yml.
* Spark job that reads AVRO or JSON messages from Kafka.
* Basic transformation performed on incoming data.

Task 5: Storing Processed Data in MinIO using Delta Lake

Topics

* Introduction to MinIO as an S3-compatible object store.
* Writing Spark DataFrames in Delta format.
* Configuring MinIO for Spark access.

Deliverables

* MinIO configured in docker-compose.yml.
* Spark job that writes processed data to MinIO in Delta format.
* Documentation on how to retrieve data from MinIO.

Task 6: Automating Deployment & Ensuring Reliability

Topics

* Automating the deployment process.
* Using Docker volumes and networks for better integration.
* Handling failures and ensuring data consistency.

Deliverables

* Fully automated docker-compose.yml with all services.
* Updated documentation on how to deploy the project from scratch.