Hello Heart

Senior Data Engineer

Home Assessment

Objective

The purpose of this task is to assess your ability to build a data pipeline that ingests, processes, de-identifies, stores, and joins health-related data using Docker and LocalStack to simulate S3. The output should be stored in **Parquet format**, and the two resulting Parquet tables should be joined and displayed using **PySpark**.

Overview

You are tasked with building a batch ETL pipeline that processes patient health data from multiple sources. You will de-identify sensitive information and then transform and store the data in **Parquet format** in a simulated **S3** environment using **LocalStack**. The final step is to join the two Parquet files (patient data and appointment data) and print the joined table using **PySpark**.

Data Sources

You will work with two datasets:

1. **Patient Data (CSV format):** Contains patient details such as patient_id, name, age, address, phone_number, and diagnosis.
2. **Appointment Data (CSV format):** Contains details of patient appointments, including patient_id, appointment_date, and doctor.

Steps to Complete

1. **Data Ingestion:**
    - Ingest data from the two provided CSV files. One contains patient details, and the other contains appointment data.
2. **De-identification:**
    - De-identify sensitive patient data such as name, address, and phone_number using an anonymization technique (e.g., hashing).
    - Ensure that the data can still be linked across the two datasets via a secure common identifier (e.g., patient_id).
3. **Data Transformation:**
    - Clean and transform the data. Ensure that:

    - Phone numbers and addresses are in a consistent format.
    - Data is deduplicated based on patient_id.
    - Join the two datasets using patient_id to create a single view of the patient and their appointment history.
4. **Data Storage:**
    - Set up **LocalStack** to simulate an S3 environment.
    - Store the de-identified and transformed data as **Parquet** files in the LocalStack S3 bucket.
    - Ensure your solution uses **Docker** to run the pipeline in a containerized environment.
5. **Data Join with PySpark:**
    - Use **PySpark** to load the two Parquet tables (patient data and appointment data) from the LocalStack S3 bucket.
    - Join the two tables on patient_id and print the resulting joined dataset.
6. **Documentation:**
    - Provide brief documentation outlining your approach to the ETL pipeline, the transformations applied, the de-identification process, how you set up Docker and LocalStack, and any assumptions made during the task.

Deliverables

1. **ETL Code:** Provide the code for your pipeline (Python, PySpark, etc.).
2. **De-identified and Transformed Data:** Store the final dataset as Parquet files in the LocalStack S3 bucket.
3. **Docker Setup:** Ensure that your solution includes a **Dockerfile** and instructions for running your pipeline using Docker and LocalStack.
4. **PySpark Join:** Demonstrate the join between the two Parquet tables using PySpark, and print the result.
5. **Documentation:** Provide a README file with instructions on how to run your code, including Docker, LocalStack setup, and PySpark.

Please compress files to a .zip and upload using the link shared from our recruiting team.

Additional Notes

- Ensure that your code follows best practices for error handling, logging, and modularity.
- LocalStack should simulate S3, and the output should be stored as Parquet files.
- The final step should show the joined dataset using PySpark.
- If you need clarification, please ask questions to the recruiting team through email.