An ELT pipeline with Airbyte and DBT

Objective

The goal of this take-home assignment is to assess your ability to design and implement an ELT (Extract, Load, Transform) pipeline for processing cryptocurrency data as well as your ability to understand usual concepts of a Python project.

You will be required to extract raw data (price of a coin at a certain timestamp) from the CoinGecko API, load it into DuckDB, and transform the data using dbt to generate daily candlestick data (opening, closing, minimum, and maximum prices).

We advise you not to spend more than 3 hours on this test, we understand that this is already a long assignment **and we don't expect you to be able to fulfill every single part of the project in due time.** So do not worry when submitting your work if it’s not finished, but do mention how you imagined the rest.

Requirements:

Repo Setup:

- We want you to submit a **Git** repo (either as a standalone archive, either as a Github or Gitlab link)
- Use a README to walk us through your solution! Explain the design decisions made during the implementation process.
- Use a Makefile
- Use a dependency management tool (Poetry, PDM, UV)**

**It's unlikely that in production you would share the same Python environment for all components of the pipeline. Here you may use the same if you choose this option, it's really up to you.

Extraction:

- Use PyAirbyte to extract raw data from the CoinGecko API.
- Retrieve historical price data for a specific cryptocurrency (e.g., Bitcoin) for a given time range with a granularity of ~1 hour. Do not try to retrieve the entire history of the due to API rate limitation. Instead, retrieve data for a reasonable time range (e.g., 1 week).
- Ensure that the extracted data includes timestamp, price and any other relevant fields.

Loading:

- Load the extracted raw data into DuckDB, a lightweight and in-process SQL database.
- Design an appropriate schema for storing the raw data in DuckDB.

Transformation:

- Use dbt (Data Build Tool) for the transformation layer of the pipeline.

- Write a dbt model to transform the raw data into a model that let's you compute daily candlestick data, including the opening price, closing price, minimum price, and maximum price for each day.

Orchestration:

- Use either Airflow, Dagster or Prefect to build a DAG to run your ELT pipeline.
- You may or may not use Docker (and docker compose) to bootstrap a local scheduler.
    - NB: some schedulers like Airflow or Dagster have a standalone in-process** version that you can use as well without Docker.
- Configure the necessary tasks for data extraction, loading, and transformation.

Bonus Points:

- Implement error handling and data quality checks within the pipeline.
- Your dag is idempotent and can be run multiple times without causing issues.
- Your transformation layer is optimized for performance and scalability (incremental).
- Optimize the performance of the pipeline by considering factors such as data partitioning and indexing.
- Provide a simple data visualization or analysis using the transformed candlestick data.

Evaluation Criteria:

Code quality, readability, and adherence to best practices. Clear and concise documentation.

Please submit your assignment within the given timeframe. If you have any questions or need further clarification, feel free to reach out to both paul.messinesi@lydia-app.com and adrien.nouvellet@lydia-app.com

Good luck with the assignment!