Assignment: Dockerized Data Pipeline with Airflow/DagsterObjective

Develop a Dockerized data pipeline using either Airflow or Dagster to automatically fetch, parse, and store stock market data.

The pipeline should:

- **Fetch Data:** Retrieve JSON stock market data from a free API (e.g., Alpha Vantage, Yahoo Finance) on a scheduled basis (hourly or daily).
- **Process and Store:** Parse the JSON response and update an existing PostgreSQL table with the extracted information.
- **Ensure Robustness:** Include comprehensive error handling and logic to manage missing data gracefully.

Requirements

To achieve the objective, the pipeline must meet the following technical specifications:

- **Docker Compose:** Utilize Docker Compose for seamless building and deployment of the entire pipeline with a single command.
- **Orchestration:** Select either Airflow or Dagster as the data orchestrator, deploying it within a Docker container.
- **Pipeline Logic:** Implement a DAG (for Airflow) or a job (for Dagster) that performs the following steps:
    - **API Interaction:** Fetch data from the chosen stock market API using a Python `requests` library.
    - **Data Extraction:** Parse the JSON response and extract all relevant data points.
    - **Database Update:** Update the designated PostgreSQL table with the extracted data.
    - **Error Management:** Incorporate `try-except` blocks and conditional logic to handle errors and missing data effectively.
- **Security:** Manage sensitive information such as API keys and database credentials using environment variables.
- **Scalability & Resilience:** Design the pipeline to be scalable and capable of handling potential failures.

Deliverables:

The successful completion of this assignment requires the submission of the following:

- **`docker-compose.yml`:** A Docker Compose file to build and run the entire data pipeline.
- **Orchestrator Logic:** A DAG (Airflow) or a job (Dagster) file containing the core data pipeline logic.
- **Data Fetching Script:** A Python script responsible for fetching data from the stock market API and updating the PostgreSQL table.
- **`README.md`:** A comprehensive `README.md` file providing clear instructions on how to

build and run the pipeline.

Evaluation Criteria

The project will be evaluated based on the following criteria:

- **Correctness:** Accuracy of data fetching and database updates.
- **Error Handling:** Effectiveness and robustness of the implemented error and missing data handling.
- **Scalability:** The pipeline's ability to accommodate increased data volume or processing frequency.
- **Code Quality:** The organization, readability, and maintainability of the codebase.
- **Dockerization:** The proper implementation of Dockerization, enabling execution via Docker Compose.