**Project Plan: Daily Market Data Analytics Pipeline**

**1. Executive Summary & Objectives**

This project's goal is to design, build, and deploy a robust, automated data pipeline that ingests daily stock market data from a public API. The raw data will be processed and stored in a cloud data warehouse, where it will be modeled into an analytics-ready format. This foundational pipeline will enable historical market analysis and serve as a reliable data source for future business intelligence or data science initiatives.

**Key Objectives:**

* **Objective 1: Automate Data Ingestion:** Develop a reliable, scheduled process to extract daily stock market data for a predefined list of tickers, eliminating manual data collection.
* **Objective 2: Create a Centralized Data Warehouse:** Establish a clean, well-documented, and performant set of tables in a cloud data warehouse, serving as the "single source of truth" for market data.
* **Objective 3: Ensure Data Quality & Reliability:** Implement automated checks and transformations to ensure the data is accurate, consistent, and ready for analysis.

**2. Technology Stack**

This project will utilize a modern, cloud-native data stack.

* **Cloud Provider:** Google Cloud Platform (GCP) or Amazon Web Services (AWS)
* **Data Ingestion:** **Python 3.x** with the requests library
* **Data Source:** **Alpha Vantage API** (Time Series Stock Data)
* **Orchestration:** **Apache Airflow** (running locally via Docker)
* **Data Lake (Storage):** **Google Cloud Storage (GCS)** or **Amazon S3**
* **Data Warehouse:** **Google BigQuery** or **Snowflake**
* **Transformation & Modeling:** **dbt (Data Build Tool)**
* **Containerization:** **Docker** & Docker Compose

**3. Project Milestones & Timeline**

This project will be executed in four one-week sprints.

**Milestone 1: Environment Setup & Local Extraction (Target: End of Week 1)**

The focus of this phase is to establish the development environment and validate the data extraction logic.

* **Tasks:**
  + 1.1: Create cloud project/account on GCP or AWS.
  + 1.2: Obtain an API key from Alpha Vantage.
  + 1.3: Develop a Python script (extract.py) that successfully fetches daily data for at least five stock tickers and saves the output locally as JSON files.
  + 1.4: Set up a local Apache Airflow environment using the official Docker Compose file.
* **Deliverable:** A functional Python script and a running local Airflow instance.

**Milestone 2: Cloud Integration & Orchestration (Target: End of Week 2)**

This phase focuses on moving data to the cloud and automating the extraction process.

* **Tasks:**
  + 2.1: Provision a data lake bucket in GCS/S3 and a data warehouse instance in BigQuery/Snowflake.
  + 2.2: Modify the Python script to upload the raw JSON files directly to the data lake, partitioned by date (e.g., gs://market-data-lake/raw/prices/YYYY-MM-DD/).
  + 2.3: Create an initial Airflow DAG (market\_data\_dag.py) that runs the Python extraction script on a weekday schedule (0 8 \* \* 1-5).
* **Deliverable:** Raw JSON files appearing automatically in the cloud data lake every weekday.

**Milestone 3: Data Transformation & Modeling (Target: End of Week 3)**

The focus here is to transform the raw, semi-structured data into clean, structured tables ready for analysis.

* **Tasks:**
  + 3.1: Initialize a dbt project and connect it to the data warehouse.
  + 3.2: Configure dbt to read the raw JSON files from the data lake.
  + 3.3: Build dbt models to create the following tables:
    - stg\_daily\_prices: Cleans and casts data types from the raw source.
    - dim\_tickers: A dimension table with ticker symbols and other metadata.
    - fct\_market\_daily: A final fact table with daily OHLCV (Open, High, Low, Close, Volume) data, properly keyed and documented.
  + 3.4: Add dbt data quality tests (e.g., not\_null for closing price, unique keys) to ensure data integrity.
* **Deliverable:** Clean, tested, and queryable tables in the data warehouse.

**Milestone 4: End-to-End Pipeline Automation & Documentation (Target: End of Week 4)**

This final phase integrates the transformation step into the automated workflow and prepares the project for showcase.

* **Tasks:**
  + 4.1: Add a dbt run task to the Airflow DAG, making it dependent on the successful completion of the extraction task.
  + 4.2: Perform a full end-to-end run of the pipeline to validate that data flows from the API to the final dbt models automatically.
  + 4.3: Create a comprehensive README.md file in the project's GitHub repository, explaining the project's purpose, architecture, and how to set it up.
* **Deliverable:** A fully automated, production-style data pipeline and a well-documented project repository.