

Stock Price Prediction for Amazon (AMZN) and Meta (META) using Airflow, yfinance, and Snowflake

Danish Waseem (019101511), Srinidhi Jaya Revanth Srirangarajapally (019123143)

Department of Applied Data Science,

College of Information, Data, and Society,

San Jose State University, San Jose, California, USA 95192

Email: *danish.waseem@sjsu.edu, srinidhijayarevanth.srirangarajapally@sjsu.edu*

Abstract

This paper presents a daily automated system that downloads stock prices for Amazon (AMZN) and Meta (META), stores them in Snowflake, and predicts prices for the next seven days using Snowflake ML Forecast. The workflows are built and scheduled using Apache Airflow and monitored through its web interface. This report describes the motivation, requirements, workflow design, database schema, and implementation details of the system. Screenshots, code repositories, and outputs are referenced inline and in the Appendix.

I. TEAM INTRODUCTION

Danish Waseem led ETL pipeline development, Snowflake schema, and orchestration.

Srinidhi Jaya Revanth Srirangarajapally implemented forecasting DAG, SQL modeling in Snowflake ML, and documentation.

Both collaborated on system design, testing, and report writing.

II. PROBLEM STATEMENT

Stock price forecasting is a key application in data analytics that requires processing large volumes of time-series data. Manual data collection and modeling are error-prone and time-consuming. This project automates the entire workflow: data extraction using yfinance, transformation and loading through Airflow, and predictive modeling using Snowflake ML. The system collects 180 days of OHLCV (Open, High, Low, Close, Volume) data for AMZN and META, stores it in Snowflake, and automatically forecasts prices for the next seven days.

III. DATASET(S)

We use **Yahoo Finance** via the `yfinance` Python package:

- **Symbols:** AMZN, META
- **Fields:** Open, High, Low, Close, Volume
- **Horizon:** Most recent 180 trading days (daily interval)
- **Access:** Public API wrapper; no keys required

IV. SYSTEM DIAGRAM

Figure 1 depicts the end-to-end flow from extraction to prediction. The system ensures data integrity, modularity, and full automation from raw data ingestion to forecast output.

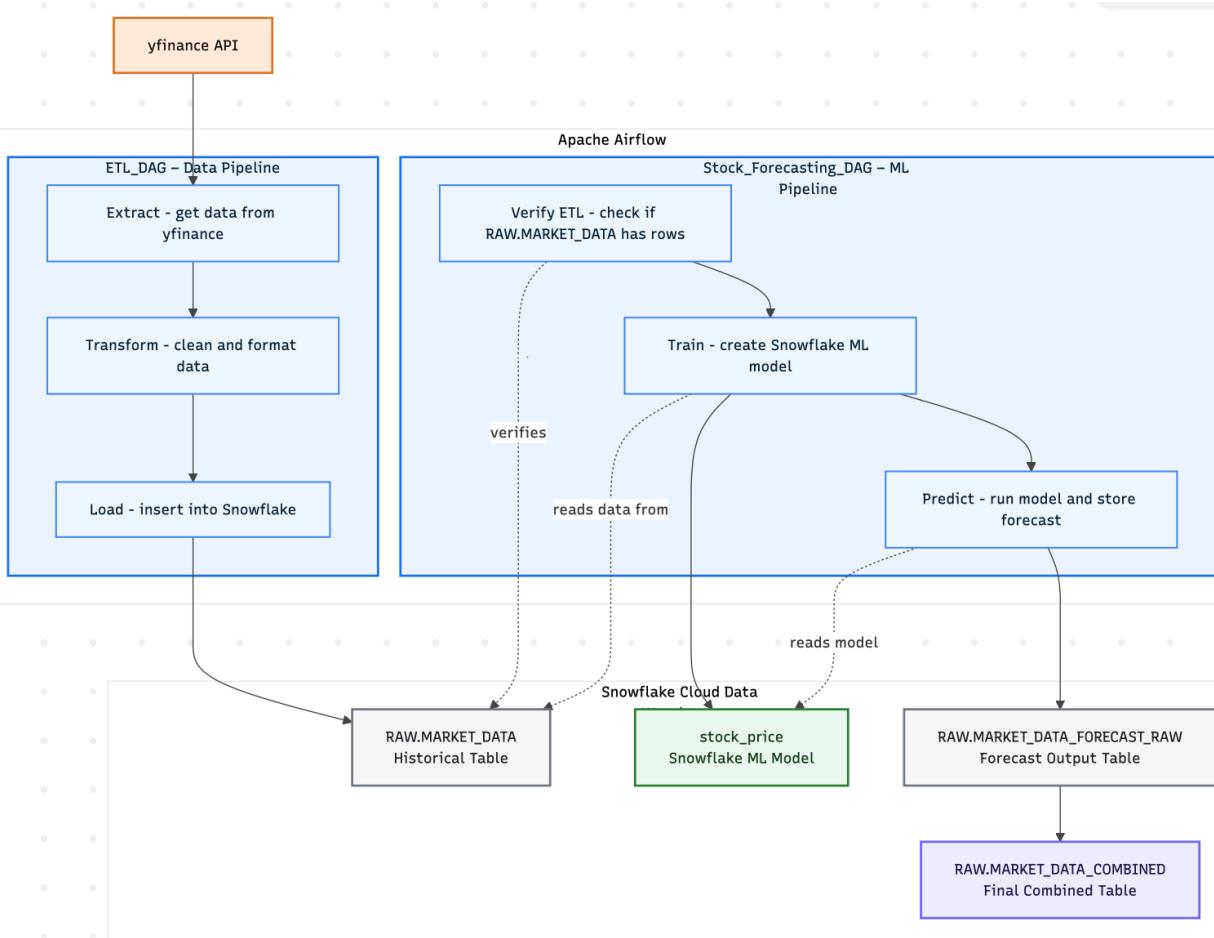


Fig. 1: System Architecture: *yfinance* → Airflow ETL DAG → Snowflake (*RAW.MARKET_DATA*) → Airflow Forecasting DAG → Final Output.

V. TABLES

A. *RAW.MARKET_DATA* (*Historical Data*)

TABLE I: Snowflake Table: *RAW.MARKET_DATA*

Column	Type	Null	Constraints / Notes
DATE	DATE	No	Trading day (UTC). Logical uniqueness with SYMBOL.
OPEN	FLOAT	No	Open price for the day.
HIGH	FLOAT	No	High price for the day.
LOW	FLOAT	No	Low price for the day.
CLOSE	FLOAT	No	Close price (fallback to Adj Close if needed).
VOLUME	BIGINT	No	Total traded volume for the day.
SYMBOL	VARCHAR (10)	No	Ticker symbol (AMZN, META).
CREATED_AT	TIMESTAMP_NTZ	Yes	Ingestion timestamp; default CURRENT_TIMESTAMP () .

Logical Primary Key: (SYMBOL, DATE).

A populated view of *RAW.MARKET_DATA* is shown in Figure 2.

B. *RAW.MARKET_DATA_COMBINED* (*History ∪ Forecast*)

A sample of the final combined output is shown in Figure 3.

The screenshot shows a Snowflake session window titled "2025-10-06 3:04pm". The table "RAW.MARKET_DATA" is displayed with the following schema:

	DATE	OPEN	HIGH	LOW	CLOSE	VOLUME	SYMBOL	CREATED_AT
172	2025-09-24	757.5	761.099985352	752.53002927	760.659973145	8828200	META	2025-10-06 15:37:56.930
173	2025-09-25	753.450012207	756.770019531	744.549987793	748.909973145	10591100	META	2025-10-06 15:37:57.548
174	2025-09-26	750	751.929992676	737.349975586	743.75	9696300	META	2025-10-06 15:37:58.250
175	2025-09-29	748.719970703	750.780029297	738.150024414	743.400024414	9246800	META	2025-10-06 15:37:58.733
176	2025-09-30	742.25	742.969970703	726.299987793	734.380004883	16228800	META	2025-10-06 15:37:59.323
177	2025-10-01	721.49990234	721.849975586	710.200012207	717.340026855	20419800	META	2025-10-06 15:38:00.027
178	2025-10-02	722.58001709	727.770019531	718.140014648	727.049987793	11415300	META	2025-10-06 15:38:00.623
179	2025-10-03	729.830004883	731	710.179992676	710.559997559	16110000	META	2025-10-06 15:38:01.018
180	2025-10-06	705	718.880004883	690.659973145	715.659973145	21357735	META	2025-10-06 15:38:01.841
181	2025-01-17	225.839998338	226.509994507	223.080001831	225.940002441	42370100	AMZN	2025-10-06 15:38:02.514
182	2025-01-21	228.899993896	231.779998779	226.940002441	230.710006714	39951500	AMZN	2025-10-06 15:38:02.972
183	2025-01-22	232.020004272	235.440002441	231.190002441	235.009994507	41446200	AMZN	2025-10-06 15:38:04.717
184	2025-01-23	234.100006104	235.520004272	231.509994507	235.419998169	28044000	AMZN	2025-10-06 15:38:05.539
185	2025-01-24	234.5	236.399993896	232.929992676	234.850006104	25890700	AMZN	2025-10-06 15:38:06.041
186	2025-01-27	226.210006714	235.61000061	225.86000061	235.419998169	49428300	AMZN	2025-10-06 15:38:06.541
187	2025-01-28	234.289993286	241.770004272	233.379995728	238.149993896	41587200	AMZN	2025-10-06 15:38:07174
188	2025-01-29	239.020004272	240.38999393	236.149993896	237.070007324	26091700	AMZN	2025-10-06 15:38:07884
189	2025-01-30	23713999939	237.949996948	232.220001221	234.639999393	32020700	AMZN	2025-10-06 15:38:08.626
190	2025-01-31	236.5	240.289993286	236.410003662	237.679992676	36110200	AMZN	2025-10-06 15:38:09.317
191	2025-02-03	234.059997559	239.25	232.899993896	237.419998169	37285900	AMZN	2025-10-06 15:38:09.840
192	2025-02-04	239.009994507	242.520004272	238.029998779	242.059997559	29713800	AMZN	2025-10-06 15:38:09.840
193	2025-02-05	237.020004272	238.320007324	235.199996948	236.169998169	38727300	AMZN	2025-10-06 15:38:09.840

Fig. 2: Snowflake table RAW.MARKET_DATA populated by ETL_DAG.

TABLE II: Snowflake Table: RAW.MARKET_DATA_COMBINED

Column	Type	Null	Constraints / Notes
SYMBOL	VARCHAR (10)	No	Ticker symbol.
DATE	DATE	No	History date or forecasted date (cast from timestamp).
ACTUAL	FLOAT	Yes	Close price for history rows; NULL for forecasts.
FORECAST	FLOAT	Yes	Model forecast for forecast rows; NULL for history.
LOWER_BOUND	FLOAT	Yes	Lower bound of 95% interval.
UPPER_BOUND	FLOAT	Yes	Upper bound of 95% interval.

Row Semantics: Exactly one of {ACTUAL, FORECAST} is non-NULL.

VI. IMPLEMENTATION DETAILS

A. Scheduling and Orchestration

Airflow schedules two DAGs:

- **ETL_DAG** at 02:30 daily (Extract → Transform → Load).
- **Stock_Forecasting_DAG** at 03:00 daily (Verify → Train → Predict).

B. ETL_DAG Pipeline

Extract uses `yfinance.Ticker().history()` for a clean DataFrame. **Transform** drops invalid rows. **Load** executes DDL and idempotent upserts (implemented as delete+insert) inside a transaction. Figure 4 shows the ETL DAG in the Airflow UI.

C. Forecasting Pipeline

Verify checks that RAW.MARKET_DATA has rows. **Train** creates MARKET_DATA_v1 and the model stock_price. **Predict** calls !FORECAST, stores the raw output, creates a clean view, and builds the combined table. Figure 5 shows the Forecasting DAG.

SYMBOL	DATE	ACTUAL	FORECAST	LOWER_BOUND	UPPER_BOUND
358 AMZN	2025-10-02	222.410003662	null	null	null
359 AMZN	2025-10-03	219.509994507	null	null	null
360 AMZN	2025-10-06	220.899993896	null	null	null
361 META	2025-10-07	null	723.283589609	695.070817929	751.49639458
362 META	2025-10-08	null	726.302968271	687.253432031	765.352565546
363 META	2025-10-09	null	725.531270273	678.063458784	772.999120601
364 META	2025-10-10	null	724.714925058	670.113051722	779.316809491
365 META	2025-10-13	null	726.336201913	685.430788862	787.241642708
366 META	2025-10-14	null	727.30409421	660.689230834	793.919019495
367 META	2025-10-15	null	729.214528329	657.34245613	801.086650467
368 META	2025-10-16	null	730.414632089	653.844606632	807.164707484
369 META	2025-10-17	null	730.540517099	649.168853434	811.914202959
370 META	2025-10-20	null	730.864644296	645.134242485	816.595067902
371 META	2025-10-21	null	733.7509358	643.874780863	823.627146224
372 META	2025-10-22	null	734.909474618	641.070526674	828.748439208
373 META	2025-10-23	null	738.401021249	640.780034293	836.042074789
374 META	2025-10-24	null	733.49802247	632.195573183	834.798521695
375 AMZN	2025-10-07	null	220.798302739	211.361003454	230.235602024
376 AMZN	2025-10-08	null	220.798302739	208.029575648	233.56702983
377 AMZN	2025-10-09	null	220.798302739	205.402808452	236.193697026
378 AMZN	2025-10-10	null	220.798302739	203.16322579	231.77555742
379 AMZN	2025-10-13	null	220.798302739	201.177555742	234.177555742

Fig. 3: Forecast output in RAW.MARKET_DATA_COMBINED (history \cup forecast).

D. Key SQL (Snowflake)

Model creation:

```
CREATE OR REPLACE SNOWFLAKE.ML.FORECAST stock_price(
INPUT_DATA => SYSTEM$REFERENCE('VIEW', 'MARKET_DATA_v1'),
SERIES_COLNAME => 'SYMBOL',
TIMESTAMP_COLNAME => 'DATE_v1',
TARGET_COLNAME => 'CLOSE',
CONFIG_OBJECT => { 'ON_ERROR': 'SKIP' }
);
```

Forecast execution and capture:

```
BEGIN
CALL stock_price!FORECAST(
FORECASTING_PERIODS => 14,
CONFIG_OBJECT => {'prediction_interval': 0.95}
);
LET x := SQLID;
CREATE OR REPLACE TABLE RAW.MARKET_DATA_FORECAST_RAW AS
SELECT * FROM TABLE(RESULT_SCAN(:x));
END;
```

Final union:

```
CREATE OR REPLACE TABLE RAW.MARKET_DATA_COMBINED AS
SELECT SYMBOL, DATE, CLOSE AS ACTUAL, NULL AS FORECAST, NULL AS LOWER_BOUND, NULL AS
UPPER_BOUND
FROM RAW.MARKET_DATA
WHERE SYMBOL IN ('META', 'AMZN')
```

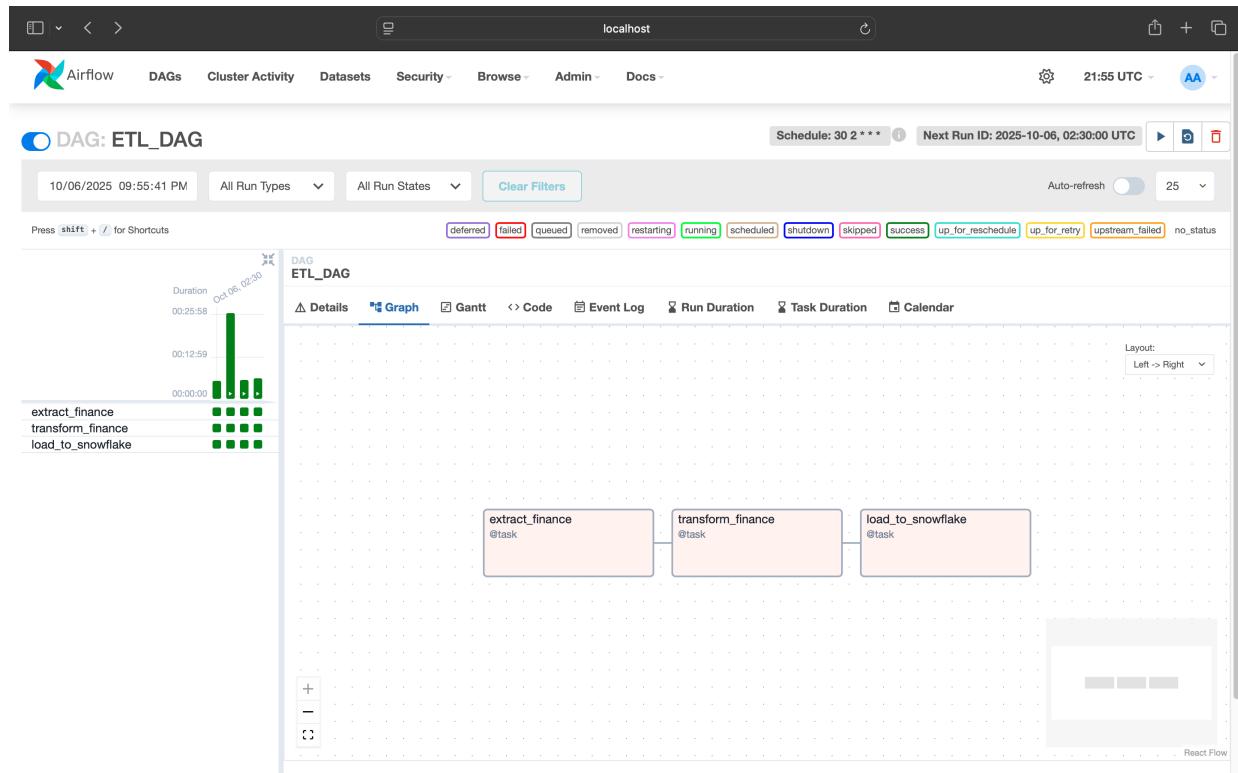


Fig. 4: Airflow ETL DAG: Extract → Transform → Load.

UNION ALL

```
SELECT SYMBOL, CAST(DATE_v1 AS DATE), NULL AS ACTUAL, FORECAST, LOWER_BOUND,
       UPPER_BOUND
FROM MARKET_DATA_FORECAST_CLEAN;
```

VII. LESSONS

- Using `Ticker().history()` avoids MultiIndex issues from `yf.download()` for single symbols.
- Always wrap Snowflake writes in transactions for safe retries.
- Validate data before model training (our Verify task prevents empty-train runs).
- Keep DAGs modular; it simplifies debugging and re-runs.

VIII. FUTURE WORK

- Add more symbols and longer forecasting windows.
- Include technical indicators (moving averages, Bollinger Bands) for trend analysis.

IX. CONCLUSION

Two Airflow DAGs automate stock ingestion and forecasting for AMZN and META. Airflow provides scheduling/observability, while Snowflake ML enables fast model training and inference. The Verify–Train–Predict flow maintains data quality and reproducibility.

ACKNOWLEDGMENT

The authors thank Professor Keeyong Han and SA Jeff Chong for their valuable feedback and guidance throughout the project.

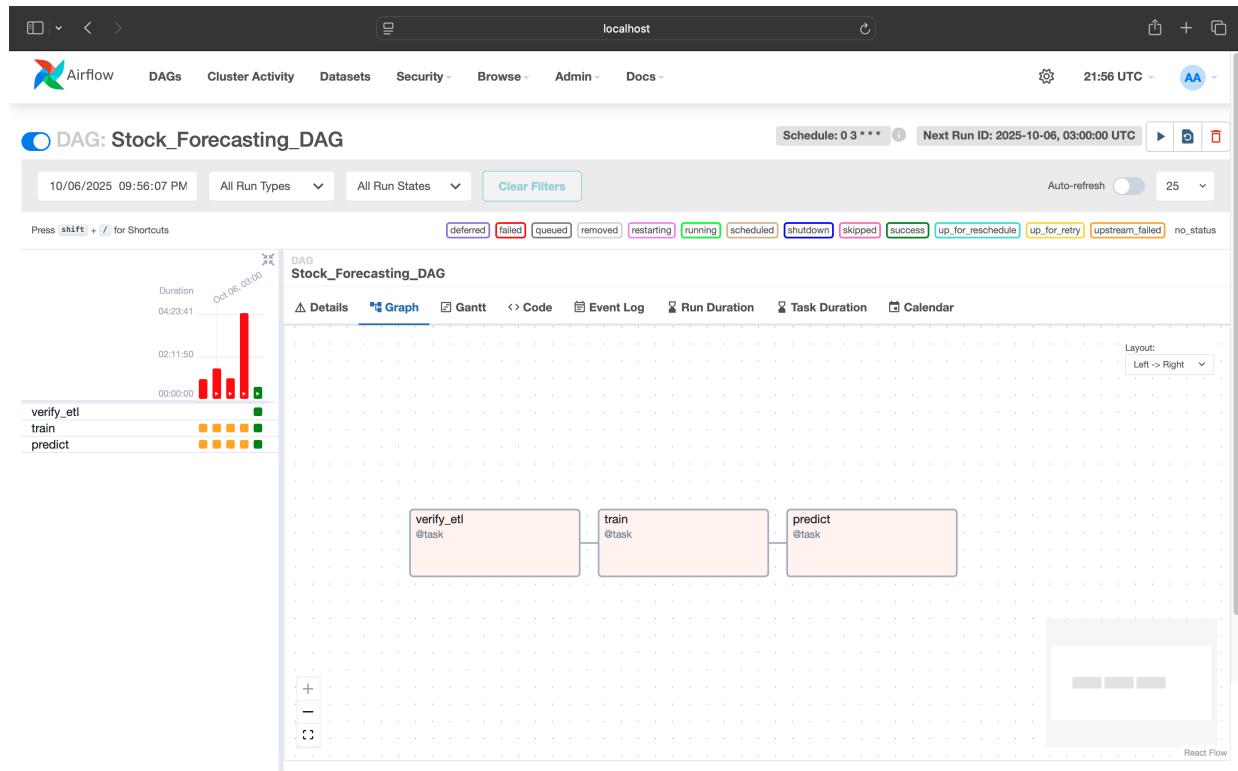


Fig. 5: Airflow Stock Forecasting DAG: Verify → Train → Predict.

REFERENCES

- 1) yfinance: <https://pypi.org/project/yfinance/>
- 2) Snowflake ML Forecast: <https://docs.snowflake.com/en/sql-reference/ml-forecast>
- 3) Apache Airflow: <https://airflow.apache.org/docs/>

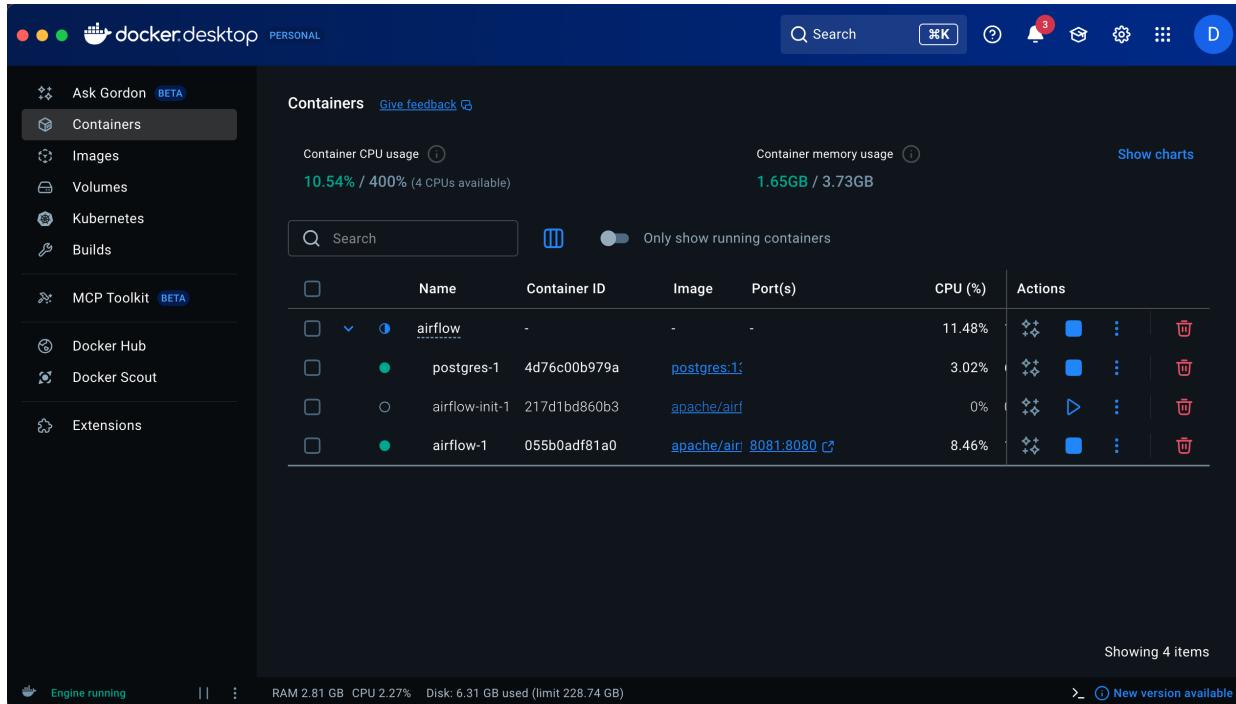


Fig. 6: Docker Desktop

APPENDIX: SOURCE CODE AND REPOSITORIES

All Python DAGs, SQL scripts, and supplementary files are hosted on GitHub:

- Danish Waseem (Main Repo): <https://github.com/danwaseem/SJSU-DATA226/tree/main/LAB1>
- Srinidhi Jaya Revanth Srirangarajapally (Main Repo): <https://github.com/Revanth0211/DATA-226-Lab-1->
- Danish Waseem (Airflow DAGs): <https://github.com/danwaseem/SJSU-DATA226/tree/main/Airflow>
- Srinidhi Jaya Revanth Srirangarajapally (Airflow DAGs): <https://github.com/Revanth0211/Stock-Prices-ETL-Airflo>

The screenshot shows the Airflow web interface at the 'DAGs' page. At the top, there are navigation links: Airflow, DAGs, Cluster Activity, Datasets, Security, Browse, Admin, Docs, and a search bar. The top right corner shows the time as 19:52 UTC and a user icon. A message banner at the top states: "Triggered ETL_DAG with new Run ID manual_2025-10-07T19:52:03.140158+00:00, it should start any moment now." Below the banner, the title "DAGs" is displayed. A filter bar allows selecting "All" (3), "Active" (3), or "Paused" (0) DAGs, and shows counts for "Running" (1) and "Failed" (1). There are also buttons for "Filter DAGs by tag" and "Search DAGs". An "Auto-refresh" button and a refresh icon are on the far right.

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions
AlphaVantage_to_Snowflake ETL StockAPI	airflow	3 (1 Failed)	30 2 * * *	2025-10-06, 02:30:00	2025-10-07, 02:30:00	2 (1 Failed)	[Run, View, Stop]
ETL_DAG ETL StockAPI	airflow	3 (1 Failed)	30 2 * * *	2025-10-07, 19:52:03	2025-10-07, 02:30:00	2 (1 Failed)	[Run, View, Stop]
Stock_Forecasting_DAG Forecast SnowflakeML	airflow	3 (5 Failed)	0 3 * * *	2025-10-06, 21:12:17	2025-10-07, 03:00:00	3 (1 Failed)	[Run, View, Stop]

Pagination controls and a message "Showing 1-3 of 3 DAGs" are at the bottom. A footer bar at the very bottom displays the version information: "Version: v2.10.1" and "Git Version: .release:854173176f372f6509800ed446286c32cb75045e".

Fig. 7: DAGS

The screenshot shows the Airflow web interface at the 'Connections' page. The top navigation bar includes links for Airflow, DAGs, Cluster Activity, Datasets, Security, Browse, Admin, and Docs. The top right corner shows the time as 19:52 UTC and a user icon. A message banner at the top states: "Version: v2.10.1" and "Git Version: .release:854173176f372f6509800ed446286c32cb75045e". Below the banner, the title "List Connection" is displayed. A search bar and a "Record Count: 1" indicator are present. A table lists the connections:

Conn Id	Conn Type	Description	Host	Port	Is Encrypted	Is Extra Encrypted
snowflake_conn	snowflake				False	False

Fig. 8: Snowflake Connection in Airflow

Schema Details Tables Views Stages File Formats

7 Tables

NAME ↑	TYPE	CLASSIFICATION	OWNER	ROWS	BYTES	CREATED
MARKET_DATA	Table	—	TRAINING_R...	360	12.0KB	2 weeks ago
MARKET_DATA_COMBINED	Table	—	TRAINING_R...	388	5.0KB	22 hours ago
MARKET_DATA_FORECAST_RAW	Table	—	TRAINING_R...	28	3.0KB	22 hours ago

Fig. 9: Snowflake Tables

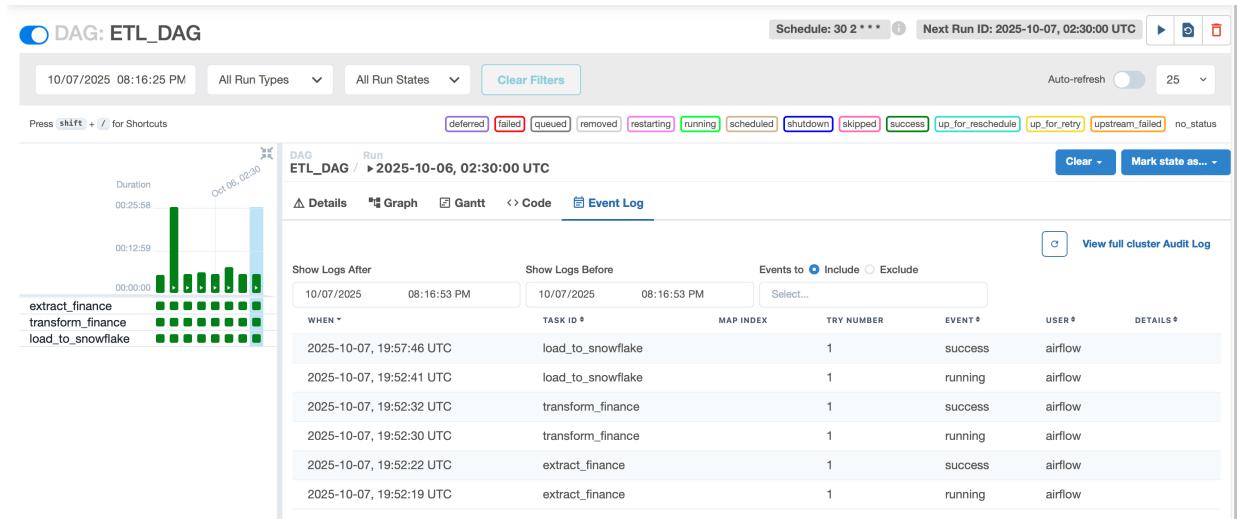


Fig. 10: ETL DAG Event Log

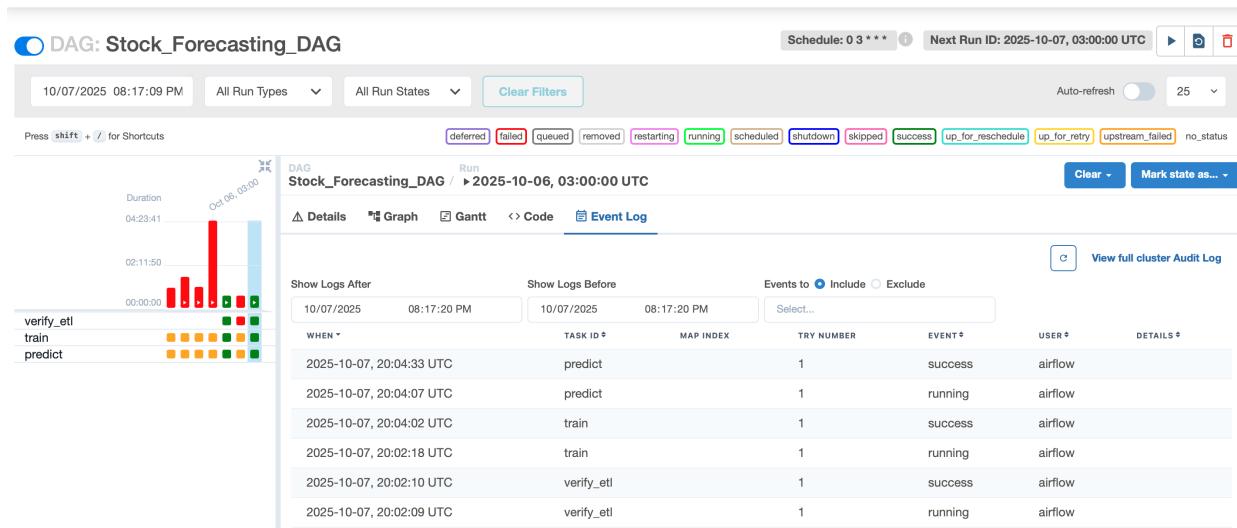


Fig. 11: Stock Forecasting DAG Event Log