

## DATA 226- DATAWAREHOUSE

### Homework 5

Name : Vimalanandhan Sivanandham

SJSU ID: 017596436

#### Porting homework #4 to Airflow (13 pts)

- (+2) Create tasks using @task decorator (refer to [GitHub linkLinks to an external site.](#))
  - You can use as many tasks as you want
  - Schedule the tasks properly (task dependency)

```
from airflow import DAG
from airflow.models import Variable
from airflow.decorators import task
import snowflake.connector
from datetime import datetime
import requests
import logging

def return_snowflake_conn():
    """Initialize Snowflake connection manually using `snowflake.connector`."""
    # Retrieve credentials from Google Colab
    snowflake_user = Variable.get("SNOWFLAKE_USER")
    snowflake_password = Variable.get("SNOWFLAKE_PASSWORD")
    snowflake_account = Variable.get("SNOWFLAKE_ACCOUNT")

    # Connect to Snowflake
    conn = snowflake.connector.connect(
        user=snowflake_user,
        password=snowflake_password,
        account=snowflake_account
    )
    return conn.cursor()

@task
def extract():
    """Extract AAPL stock data from Alpha Vantage API"""
    api_key = Variable.get("ALPHA_VANTAGE_API_KEY") # Get API key from Airflow Variables
    symbol = "AAPL"
    url = f"https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol={symbol}&apikey={api_key}&outputsize=compact"

    try:
        response = requests.get(url)
        response.raise_for_status()
        data = response.json().get("Time Series (Daily)", {})
        logging.info(f"Extracted {len(data)} records")
        return data # XCom push
    except Exception as e:
        logging.error(f"Error in extract: {str(e)}")
        raise
```

```

@task
def extract():
    """Extract AAPL stock data from Alpha Vantage API"""
    api_key = Variable.get("ALPHA_VANTAGE_API_KEY") # Get API key from Airflow Variables
    symbol = "AAPL"
    url = f"https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol={symbol}&apikey={api_key}&outputsize=compact"

    try:
        response = requests.get(url)
        response.raise_for_status()
        data = response.json().get("Time Series (Daily)", {})
        logging.info(f"Extracted {len(data)} records")
        return data # XCom push
    except Exception as e:
        logging.error(f"Error in extract: {str(e)}")
        raise

@task
def transform(data):
    """Transform extracted stock data into structured format"""
    records = []
    for date, values in data.items():
        records.append([
            date, float(values["1. open"]), float(values["2. high"]),
            float(values["3. low"]), float(values["4. close"]),
            int(values["5. volume"])
        ])
    logging.info(f"Transformed {len(records)} records")
    return records # XCom push

@task
def load(records):
    """Load transformed data into Snowflake"""
    cur = return_snowflake_conn()
    try:
        cur.execute("BEGIN;")
        cur.execute("""
            CREATE TABLE IF NOT EXISTS stock_data.raw.stock_data (
                date DATE PRIMARY KEY,
                open FLOAT,
                high FLOAT,
                low FLOAT,
                close FLOAT,
                volume INT
            );
        """)
        cur.execute("DELETE FROM stock_data.raw.stock_data;") # Full refresh
    
```

- (+1) Set up a variable for Alpha Vantage API key
  - Use the variable in your code (Variable.get)
  - Capture the Admin -> Variables screenshot (an example will be provided ①)

```
from airflow import DAG
from airflow.models import Variable
from airflow.decorators import task
import snowflake.connector
from datetime import datetime
import requests
import logging

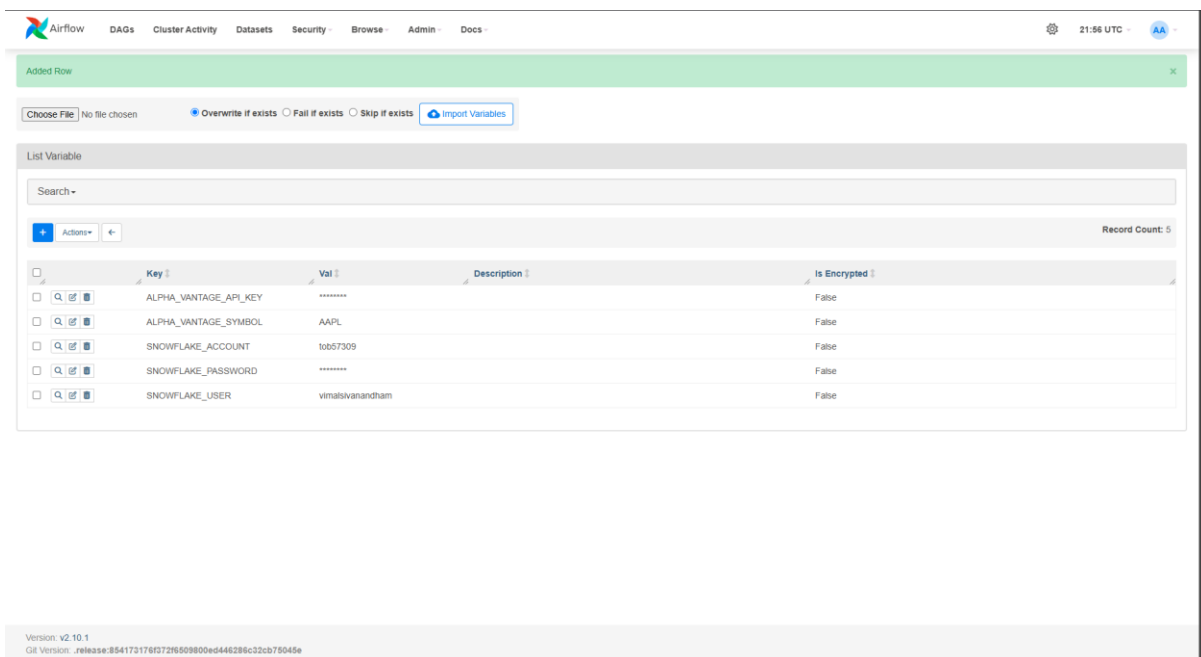
def return_snowflake_conn():
    """Initialize Snowflake connection manually using `snowflake.connector`."""

    # Retrieve credentials from Google Colab
    snowflake_user = Variable.get("SNOWFLAKE_USER")
    snowflake_password = Variable.get("SNOWFLAKE_PASSWORD")
    snowflake_account = Variable.get("SNOWFLAKE_ACCOUNT")

    # Connect to Snowflake
    conn = snowflake.connector.connect(
        user=snowflake_user,
        password=snowflake_password,
        account=snowflake_account
    )
    return conn.cursor()

@task
def extract():
    """Extract AAPL stock data from Alpha Vantage API"""
    api_key = Variable.get("ALPHA_VANTAGE_API_KEY") # Get API key from Airflow Variables
    symbol = "AAPL"
    url = f"https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol={symbol}&apikey={api_key}&outputsize=compact"

    try:
        response = requests.get(url)
        response.raise_for_status()
        data = response.json().get("Time Series (Daily)", {})
        logging.info(f"Extracted {len(data)} records")
        return data # XCom push
    except Exception as e:
        logging.error(f"Error in extract: {str(e)}")
        raise
```



The screenshot shows the Airflow Admin interface. At the top, there's a navigation bar with links like DAGs, Cluster Activity, Datasets, Security, Browse, Admin, and Docs. Below this, there's a green banner that says "Added Row". Underneath, there's a section for "List Variable" with a search bar and a table of variables.

	Key	Val	Description	Is Encrypted
<input type="checkbox"/>	ALPHA_VANTAGE_API_KEY	*****		False
<input type="checkbox"/>	ALPHA_VANTAGE_SYMBOL	AAPL		False
<input type="checkbox"/>	SNOWFLAKE_ACCOUNT	to657309		False
<input type="checkbox"/>	SNOWFLAKE_PASSWORD	*****		False
<input type="checkbox"/>	SNOWFLAKE_USER	vimalivanandham		False

Record Count: 5

Version: v2.10.1  
Git Version: release:854173176f572f6509800e446286c32cb75645e

- (+2) Set up Snowflake Connection (refer to [GitHub link](#)[Links to an external site.](#))
  - Use the connection in your code
  - Capture the Connection detail page screenshot (an example will be provided ②)

```
from airflow import DAG
from airflow.models import Variable
from airflow.decorators import task
import snowflake.connector
from datetime import datetime
import requests
import logging

def return_snowflake_conn():
    """Initialize Snowflake connection manually using `snowflake.connector`."""

    # Retrieve credentials from Google Colab
    snowflake_user = Variable.get("SNOWFLAKE_USER")
    snowflake_password = Variable.get("SNOWFLAKE_PASSWORD")
    snowflake_account = Variable.get("SNOWFLAKE_ACCOUNT")

    # Connect to Snowflake
    conn = snowflake.connector.connect(
        user=snowflake_user,
        password=snowflake_password,
        account=snowflake_account
    )
    return conn.cursor()

@task
def extract():
    """Extract AAPL stock data from Alpha Vantage API"""
    api_key = Variable.get("ALPHA_VANTAGE_API_KEY") # Get API key from Airflow Variables
    symbol = "AAPL"
    url = f"https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol={symbol}&apikey={api_key}&outputsize=compact"

    try:
        response = requests.get(url)
        response.raise_for_status()
        data = response.json().get("Time Series (Daily)", {})
        logging.info(f"Extracted {len(data)} records")
        return data # XCom push
    except Exception as e:
        logging.error(f"Error in extract: {str(e)}")
        raise
```

Connection successfully tested

21:49 UTC

45

Connection successfully tested

Edit Connection

Connection id \* snowflake\_conn

Connection Type \* Snowflake  
Connection Type missing? Make sure you've installed the corresponding Airflow Provider Package.

Description

Schema snowflake schema

Login snowflakeuser@sn

Password

Extra

```
{
  "account": "usdt1000",
  "warehouse": "compute-wh",
  "database": "db",
  "schema": "role",
  "password": "role",
  "warehouse_role": "role"
}
```

Account usdt1000

Warehouse compute-wh

Database db

Region snowflake hosted region

Role snowflake role

Private key (Path) Path of snowflake private key (PEM Format)

Private key (Text)

- (+5) Ensure the overall DAG is implemented properly and runs successfully
  - A github link with the entire code needs to be submitted (2 pts)
  - Implement the same full refresh using SQL transaction (3 pts)

<https://github.com/Vimalanandhan/DATA-226---DATAWAREHOUSE/tree/main/Homework/Homework5>

```
@task
def extract():
    """Extract AAPL stock data from Alpha Vantage API"""
    api_key = Variable.get("ALPHA_VANTAGE_API_KEY") # Get API key from Airflow Variables
    symbol = "AAPL"
    url = f"https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol={symbol}&apikey={api_key}&outputsize=compact"

    try:
        response = requests.get(url)
        response.raise_for_status()
        data = response.json().get("Time Series (Daily)", {})
        logging.info(f"Extracted {len(data)} records")
        return data # XCom push
    except Exception as e:
        logging.error(f"Error in extract: {str(e)}")
        raise

@task
def transform(data):
    """Transform extracted stock data into structured format"""
    records = []
    for date, values in data.items():
        records.append([
            date, float(values["1. open"]), float(values["2. high"]),
            float(values["3. low"]), float(values["4. close"]),
            int(values["5. volume"])
        ])

    logging.info(f"Transformed {len(records)} records")
    return records # XCom push

@task
def load(records):
    """Load transformed data into Snowflake"""
    cur = return_snowflake_conn()
    try:
        cur.execute("BEGIN;")
        cur.execute("""
            CREATE TABLE IF NOT EXISTS stock_data.raw.stock_data (
                date DATE PRIMARY KEY,
                open FLOAT,
                high FLOAT,
                low FLOAT,
                close FLOAT,
                volume INT
            );
        """)
        cur.execute("DELETE FROM stock_data.raw.stock_data;") # Full refresh
```

```

def load(records):
    cur.execute("BEGIN;")
    cur.execute("""
        CREATE TABLE IF NOT EXISTS stock_data.raw.stock_data (
            date DATE PRIMARY KEY,
            open FLOAT,
            high FLOAT,
            low FLOAT,
            close FLOAT,
            volume INT
        );
    """)
    cur.execute("DELETE FROM stock_data.raw.stock_data;") # Full refresh

    for record in records:
        # Debug: Print each record before inserting
        print(f"Inserting record: {record}")

        # Check for NULL values and replace them
        cleaned_record = [
            value if value is not None else 0 # Replace NULLs with 0
            for value in record
        ]

        sql = f"""
            INSERT INTO stock_data.raw.stock_data (date, open, high, low, close, volume)
            VALUES ('{cleaned_record[0]}', {cleaned_record[1]}, {cleaned_record[2]}, {cleaned_record[3]}, {cleaned_record[4]}, {cleaned_record[5]});
        """
        cur.execute(sql)

    cur.execute("COMMIT;")
    logging.info("Data successfully loaded into Snowflake")
    except Exception as e:
        cur.execute("ROLLBACK;")
        logging.error(f"Error in load: {str(e)}")
        raise

# Define the Airflow DAG
with DAG(
    dag_id='AAPL_Stock_ETL',
    start_date=datetime(2024, 9, 21),
    catchup=False,
    tags=['ETL', 'Stock Data'],
    schedule_interval='30 2 * * *' # Run daily at 2:30 AM UTC
) as dag:
    data = extract()
    transformed_data = transform(data)
    load(transformed_data)

```

- (+2) Capture two screenshot of your Airflow Web UI (examples to follow)
  - One with the Airflow homepage showing the DAG (③)
  - The other with the log screen of the DAG (④)

Airflow DAGs Cluster Activity Datasets Security Browse Admin Docs

19:42 PST (-08:00)

### DAGs

Active 1 Paused 1

Running 0 Failed 0

Filter DAGs by tag

Search DAGs

Auto-refresh

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
AAPL_Stock_ETL <a href="#">ETL</a> <a href="#">Stock Data</a>	airflow		30 2 ***	2025-03-01, 17:40:29	2025-02-28, 18:30:00			...
HelloWorld <a href="#">example</a>	visual		0 2 ***	2025-02-28, 22:36:15	2025-02-28, 18:00:00			...

Showing 1-2 of 2 DAGs

Version: v2.10.1  
Git Version: .release.854173176f372650980ed446286c32cb75045e

Airflow DAGs Cluster Activity Datasets Security Browse Admin Docs

19:43 PST (-08:00)

### DAG: AAPL\_Stock\_ETL

Schedule: 30 2 \*\*\* Next Run ID: 2025-03-01, 18:30:00 PST

01-03-2025 07:43:04 PM All Run Types All Run States Clear Filters

Press **ctrl** + **J** for Shortcuts

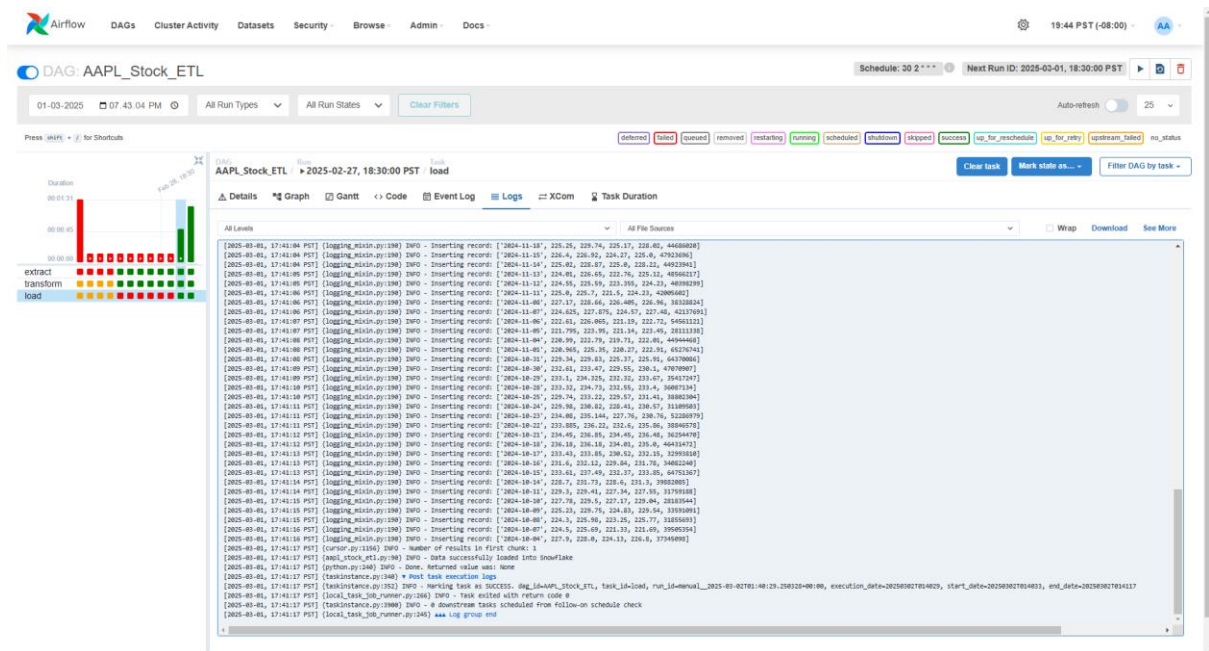
Duration: 00:01:34 Feb-28, 18:30

extract transform load

Details Graph Gantt Code Event Log Logs XCom Task Duration

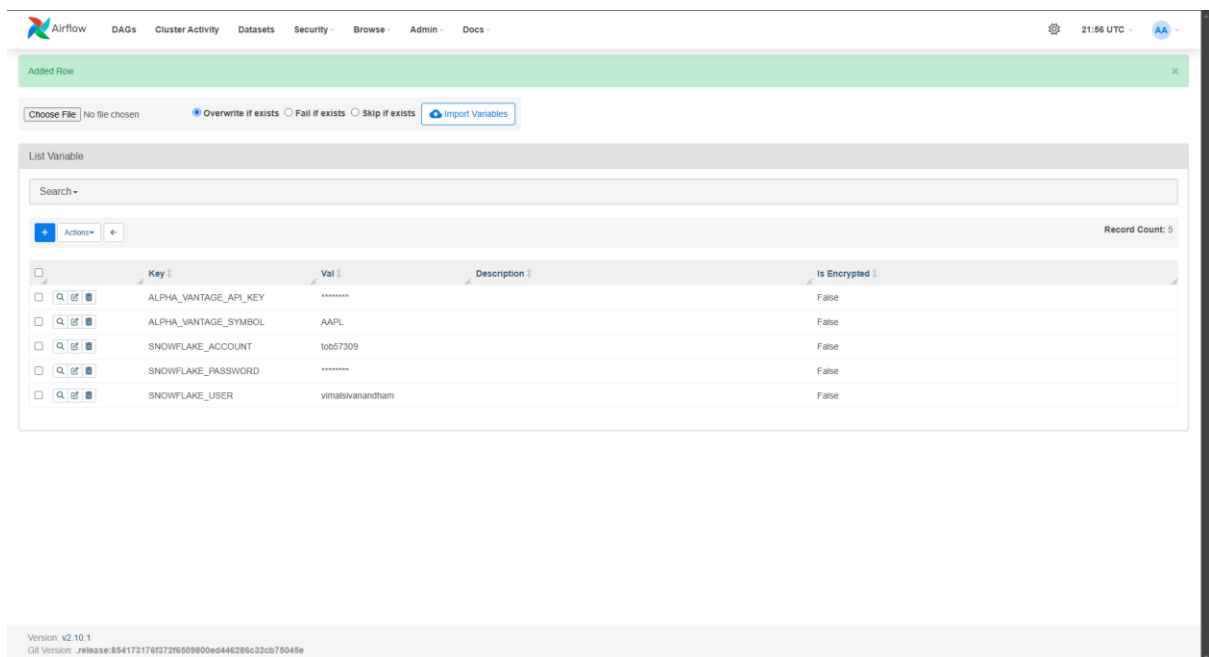
All Levels All File Sources

```
[2025-03-01, 17:43:04 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-18", 225.25, 229.74, 225.17, 228.82, 44608000]
[2025-03-01, 17:43:04 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-19", 226.4, 226.92, 224.27, 225.9, 47920200]
[2025-03-01, 17:43:04 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-20", 225.45, 226.87, 225.9, 228.22, 44620400]
[2025-03-01, 17:43:05 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-21", 224.81, 226.45, 222.74, 225.12, 44564217]
[2025-03-01, 17:43:06 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-22", 224.55, 225.59, 223.85, 224.23, 40960200]
[2025-03-01, 17:43:06 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-23", 225.0, 225.7, 221.5, 224.23, 42065600]
[2025-03-01, 17:43:06 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-26", 227.17, 228.66, 226.48, 226.96, 38208200]
[2025-03-01, 17:43:06 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-27", 224.62, 227.87, 224.57, 227.45, 42337000]
[2025-03-01, 17:43:07 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-28", 221.61, 226.48, 221.19, 222.72, 54061112]
[2025-03-01, 17:43:07 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-29", 221.79, 223.99, 221.94, 223.49, 42333300]
[2025-03-01, 17:43:08 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-29", 222.79, 223.79, 223.71, 222.81, 44044400]
[2025-03-01, 17:43:08 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-29", 226.94, 226.95, 226.27, 222.91, 45270700]
[2025-03-01, 17:43:08 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 223.84, 225.83, 225.37, 225.76, 44070600]
[2025-03-01, 17:43:09 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 223.61, 223.47, 229.59, 228.1, 47070600]
[2025-03-01, 17:43:09 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 223.1, 224.26, 222.37, 223.47, 34471200]
[2025-03-01, 17:43:10 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 223.32, 224.73, 222.65, 223.4, 34687100]
[2025-03-01, 17:43:10 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 229.74, 233.22, 229.57, 231.41, 34682200]
[2025-03-01, 17:43:11 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 229.85, 230.82, 228.41, 230.57, 31899000]
[2025-03-01, 17:43:11 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 229.85, 229.144, 227.76, 228.78, 32280970]
[2025-03-01, 17:43:11 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 223.85, 224.22, 223.6, 225.46, 34044100]
[2025-03-01, 17:43:12 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 224.46, 226.85, 224.46, 226.46, 34254400]
[2025-03-01, 17:43:12 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 226.16, 226.16, 224.84, 225.8, 44614400]
[2025-03-01, 17:43:13 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 223.43, 223.85, 226.52, 222.59, 32283300]
[2025-03-01, 17:43:13 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 223.6, 222.32, 229.84, 221.76, 34682200]
[2025-03-01, 17:43:13 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 223.43, 227.49, 222.37, 223.45, 44712400]
[2025-03-01, 17:43:14 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 228.7, 221.79, 226.5, 221.3, 34682200]
[2025-03-01, 17:43:14 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 222.3, 228.41, 227.34, 227.35, 31799800]
[2025-03-01, 17:43:15 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 229.23, 229.75, 224.83, 229.54, 33580800]
[2025-03-01, 17:43:15 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 224.3, 225.89, 223.26, 225.77, 31850600]
[2025-03-01, 17:43:16 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 224.5, 225.49, 221.33, 221.63, 39060300]
[2025-03-01, 17:43:16 PST] [logging_mixin.py:190] INFO - Inserting record: ["2024-11-30", 227.9, 228.4, 224.13, 224.6, 37349800]
[2025-03-01, 17:43:17 PST] [cursor.py:1134] INFO - Number of results in first chunk: 1
[2025-03-01, 17:43:17 PST] [aapl_stock_etl.py:38] INFO - data successfully loaded into Snowflake
[2025-03-01, 17:43:17 PST] [python.py:260] INFO - Done. Returned value: None
[2025-03-01, 17:43:17 PST] [taskinstance.py:1940] INFO - Post task execution logs
[2025-03-01, 17:43:17 PST] [taskinstance.py:1952] INFO - Having task as SUCCESS. dag_id=AAPL_Stock_ETL, task_id=load, run_id=manual_2025-03-01T18:30:00-08:00, execution_date=20250301T18:30:00, start_date=20250301T18:30:00, end_date=20250301T18:30:00
[2025-03-01, 17:43:17 PST] [taskinstance.py:1952] INFO - Task task_executor:260 - Task failed with return code 0
[2025-03-01, 17:43:17 PST] [taskinstance.py:1952] INFO - downstream tasks scheduled from follow-on schedule check
[2025-03-01, 17:43:17 PST] [taskinstance.py:1952] INFO - Task task_executor:260 - Task failed with return code 0
```



- (+1) Overall formatting

4 screenshot examples are in the lecture notes (from slides 62 to 64)





Contention successfully tested

© 2014 Pearson Education, Inc. or its affiliate(s). All rights reserved.

 Search DAGs

Showing 1-2 of 2 DAGs

**Airflow** DAGs Cluster Activity Datasets Security Browse Admin Docs

**DAG: AAPL\_Stock\_ETL** Schedule: 30 2 \* \* \* Next Run ID: 2025-03-01, 18:30:00 PST

01-03-2025 07:43:04 PM All Run Types All Run States Clear Filters

Press **shift** + **f** for Shortcuts

**AAPL\_Stock\_ETL** Run: 2025-02-27, 18:30:00 PST Task: load

Details Graph Gantt Code Event Log Logs XCOM Task Duration

All Levels All File Sources

2025-03-01, 17:43:04 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-18, 225.25, 229.74, 225.17, 228.46, 44680090]  
2025-03-01, 17:43:04 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-15, 226.4, 226.92, 224.27, 225.8, 47923980]  
2025-03-01, 17:43:04 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-14, 225.02, 226.87, 225.49, 225.22, 44923941]  
2025-03-01, 17:43:05 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-13, 224.48, 226.45, 222.74, 225.12, 44946217]  
2025-03-01, 17:43:05 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-12, 224.55, 225.59, 223.55, 224.23, 44946299]  
2025-03-01, 17:43:06 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-11, 225.4, 225.7, 225.5, 224.32, 42065692]  
2025-03-01, 17:43:06 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-08, 227.17, 228.66, 226.46, 226.96, 38328242]  
2025-03-01, 17:43:06 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-07, 224.625, 227.875, 224.57, 227.49, 41237001]  
2025-03-01, 17:43:07 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-06, 222.41, 226.465, 221.19, 222.72, 54661121]  
2025-03-01, 17:43:07 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-05, 222.795, 227.95, 221.34, 223.46, 38311383]  
2025-03-01, 17:43:08 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-04, 228.49, 222.79, 229.71, 222.46, 44944461]  
2025-03-01, 17:43:08 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-01, 226.965, 225.15, 226.27, 222.91, 46370743]  
2025-03-01, 17:43:08 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-31, 223.34, 229.43, 225.37, 223.46, 34877061]  
2025-03-01, 17:43:09 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-30, 232.41, 233.47, 229.55, 228.1, 47979097]  
2025-03-01, 17:43:09 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-29, 231.1, 234.25, 232.32, 233.47, 39477017]  
2025-03-01, 17:43:10 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-28, 233.32, 234.73, 232.65, 233.4, 34687134]  
2025-03-01, 17:43:10 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-25, 229.74, 232.22, 229.57, 231.41, 34682184]  
2025-03-01, 17:43:11 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-24, 229.85, 230.82, 228.41, 230.57, 31499081]  
2025-03-01, 17:43:11 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-23, 234.06, 235.144, 227.76, 230.76, 52389793]  
2025-03-01, 17:43:11 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-22, 231.055, 236.22, 232.4, 235.46, 38469793]  
2025-03-01, 17:43:12 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-21, 234.46, 234.85, 234.45, 234.46, 34254793]  
2025-03-01, 17:43:12 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-18, 234.16, 234.16, 234.46, 235.4, 46434712]  
2025-03-01, 17:43:13 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-17, 231.43, 231.85, 230.42, 231.85, 31639145]  
2025-03-01, 17:43:13 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-16, 231.4, 232.12, 229.84, 231.76, 34682184]  
2025-03-01, 17:43:13 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-15, 233.41, 237.49, 232.37, 233.46, 64751267]  
2025-03-01, 17:43:14 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-14, 228.7, 231.73, 226.4, 231.3, 39828093]  
2025-03-01, 17:43:14 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-13, 229.3, 229.43, 227.34, 227.35, 31759181]  
2025-03-01, 17:43:15 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-10, 227.78, 229.5, 227.19, 229.46, 34359461]  
2025-03-01, 17:43:15 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-09, 225.23, 229.75, 224.83, 229.54, 35939891]  
2025-03-01, 17:43:15 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-08, 224.3, 225.56, 223.25, 225.71, 31639145]  
2025-03-01, 17:43:16 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-07, 224.5, 225.49, 221.33, 221.69, 35939891]  
2025-03-01, 17:43:16 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-04, 227.9, 228.4, 224.15, 226.8, 37494993]  
2025-03-01, 17:43:17 PST [logging\_mysql.py:190] INFO - data successfully loaded into Snowflake  
2025-03-01, 17:43:17 PST [python.py:190] INFO - Done. Returned value set: None  
2025-03-01, 17:43:17 PST [taskinstance.py:190] INFO - Mapping task as SUCCESS. dag\_id=AAPL\_Stock\_ETL, task\_id=load, run\_id=manual\_2025-03-01-18:30:00-00, execution\_date=2025-03-01T18:30:00, start\_date=2025-03-01T18:30:00, end\_date=2025-03-01T18:30:00  
2025-03-01, 17:43:17 PST [taskinstance.py:190] INFO - Task exited with return code 0  
2025-03-01, 17:43:17 PST [taskinstance.py:190] INFO - # downstream tasks scheduled from follow-on schedule check  
2025-03-01, 17:43:17 PST [taskinstance.py:190] INFO - Log group end

**Airflow** DAGs Cluster Activity Datasets Security Browse Admin Docs

**DAG: AAPL\_Stock\_ETL** Schedule: 30 2 \* \* \* Next Run ID: 2025-03-01, 18:30:00 PST

01-03-2025 07:43:04 PM All Run Types All Run States Clear Filters

Press **shift** + **f** for Shortcuts

**AAPL\_Stock\_ETL** Run: 2025-02-27, 18:30:00 PST Task: load

Details Graph Gantt Code Event Log Logs XCOM Task Duration

All Levels All File Sources

2025-03-01, 17:43:04 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-18, 225.25, 229.74, 225.17, 228.46, 44680090]  
2025-03-01, 17:43:04 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-15, 226.4, 226.92, 224.27, 225.8, 47923980]  
2025-03-01, 17:43:04 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-14, 225.02, 226.87, 225.49, 225.22, 44923941]  
2025-03-01, 17:43:05 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-13, 224.48, 226.45, 222.74, 225.12, 44946217]  
2025-03-01, 17:43:05 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-12, 224.55, 225.59, 223.55, 224.23, 44946299]  
2025-03-01, 17:43:06 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-11, 225.4, 225.7, 225.5, 224.32, 42065692]  
2025-03-01, 17:43:06 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-08, 227.17, 228.66, 226.46, 226.96, 38328242]  
2025-03-01, 17:43:06 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-07, 224.625, 227.875, 224.57, 227.49, 41237001]  
2025-03-01, 17:43:07 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-06, 222.41, 226.465, 221.19, 222.72, 54661121]  
2025-03-01, 17:43:07 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-05, 222.795, 227.95, 221.34, 223.46, 38311383]  
2025-03-01, 17:43:08 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-04, 228.49, 222.79, 229.71, 222.46, 44944461]  
2025-03-01, 17:43:08 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-11-01, 226.965, 225.15, 226.27, 222.91, 46370743]  
2025-03-01, 17:43:08 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-31, 223.34, 229.43, 225.37, 223.46, 34877061]  
2025-03-01, 17:43:09 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-30, 232.41, 233.47, 229.55, 228.1, 47979097]  
2025-03-01, 17:43:09 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-29, 231.1, 234.25, 232.32, 233.47, 39477017]  
2025-03-01, 17:43:10 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-28, 233.32, 234.73, 232.65, 233.4, 34687134]  
2025-03-01, 17:43:10 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-25, 229.74, 232.22, 229.57, 231.41, 34682184]  
2025-03-01, 17:43:11 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-24, 229.85, 230.82, 228.41, 230.57, 31499081]  
2025-03-01, 17:43:11 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-23, 234.06, 235.144, 227.76, 230.76, 52389793]  
2025-03-01, 17:43:11 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-22, 231.055, 236.22, 232.4, 235.46, 38469793]  
2025-03-01, 17:43:12 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-21, 234.46, 234.85, 234.45, 234.46, 34254793]  
2025-03-01, 17:43:12 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-18, 234.16, 234.16, 234.46, 235.4, 46434712]  
2025-03-01, 17:43:13 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-17, 231.43, 231.85, 230.42, 231.85, 31639145]  
2025-03-01, 17:43:13 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-16, 231.4, 232.12, 229.84, 231.76, 34682184]  
2025-03-01, 17:43:14 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-15, 233.41, 237.49, 232.37, 233.46, 64751267]  
2025-03-01, 17:43:14 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-14, 228.7, 231.73, 226.4, 231.3, 39828093]  
2025-03-01, 17:43:14 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-13, 229.3, 229.43, 227.34, 227.35, 31759181]  
2025-03-01, 17:43:15 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-10, 227.78, 229.5, 227.19, 229.46, 34359461]  
2025-03-01, 17:43:15 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-09, 225.23, 229.75, 224.83, 229.54, 35939891]  
2025-03-01, 17:43:15 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-08, 224.3, 225.56, 223.25, 225.71, 31639145]  
2025-03-01, 17:43:16 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-07, 224.5, 225.49, 221.33, 221.69, 35939891]  
2025-03-01, 17:43:16 PST [logging\_mysql.py:190] INFO - Inserting record: [2024-10-04, 227.9, 228.4, 224.15, 226.8, 37494993]  
2025-03-01, 17:43:17 PST [logging\_mysql.py:190] INFO - data successfully loaded into Snowflake  
2025-03-01, 17:43:17 PST [python.py:190] INFO - Done. Returned value set: None  
2025-03-01, 17:43:17 PST [taskinstance.py:190] INFO - Mapping task as SUCCESS. dag\_id=AAPL\_Stock\_ETL, task\_id=load, run\_id=manual\_2025-03-01-18:30:00-00, execution\_date=2025-03-01T18:30:00, start\_date=2025-03-01T18:30:00, end\_date=2025-03-01T18:30:00  
2025-03-01, 17:43:17 PST [taskinstance.py:190] INFO - Task exited with return code 0  
2025-03-01, 17:43:17 PST [taskinstance.py:190] INFO - # downstream tasks scheduled from follow-on schedule check  
2025-03-01, 17:43:17 PST [taskinstance.py:190] INFO - Log group end