

# Event Driven Data Platform with Kafka, Spark, and Airflow

This project demonstrates a local ETL pipeline for processing event data from simulated sources (e.g., CRM, ERP, website, and app) using an event-driven architecture. The pipeline leverages Kafka for data ingestion, Spark for transformation, PostgreSQL/JSON for storage, and Apache Airflow for orchestration.

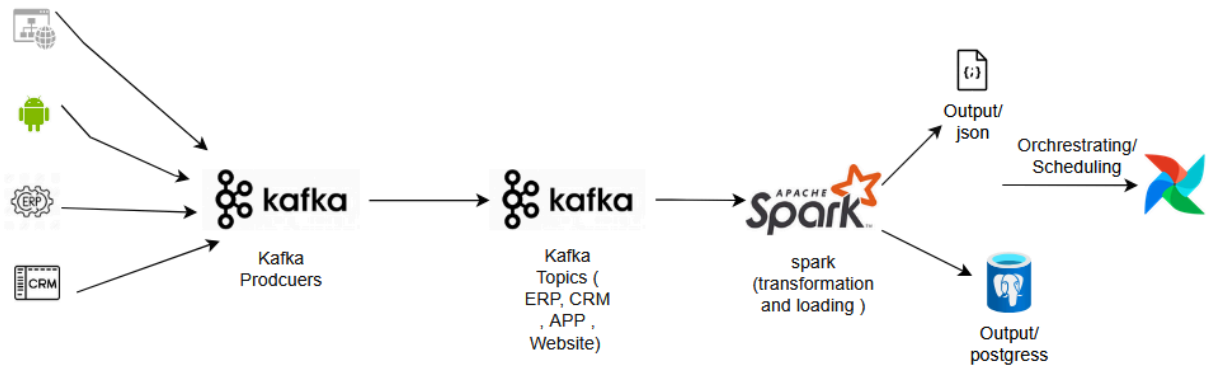
## Overview

### Goal :

To simulate how real-time and batch data pipelines work in a modern data stack by:

- Capturing raw event streams from mock producers
- Processing the events with PySpark streaming
- Persisting the cleaned data in JSON format
- Automating the entire pipeline with Apache Airflow

## Architecture



## Components:

- **Kafka:** Acts as the message broker receiving events from producers.
- **Spark Structured Streaming:** Consumes Kafka topics, applies transformations, and writes structured output.
- **PostgreSQL :** Stores summary or metadata.
- **Airflow:** Orchestrates the end-to-end flow.
- **Docker:** Containers for Airflow, Kafka, Redis, PostgreSQL.

## Workflow:

1. Mock producers generate and publish JSON events to Kafka topics.(didn't use an actual API)
2. Spark consumes and processes the events based on schemas.

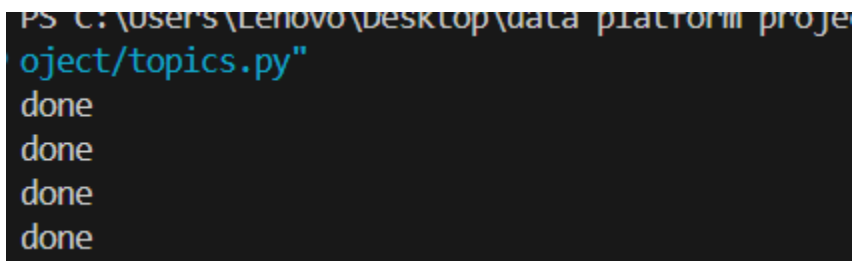
3. Cleaned data is saved as JSON files locally.
4. Airflow schedules and monitors the daily ETL run.

## Project Structure

```
project-root/
├── dags/
│   ├── etl_pipeline_dag.py          # Airflow DAG definition
│   ├── producers/
│   │   └── producer.py             # Generates Kafka events
│   └── spark_jobs/
│       └── spark_consumer.py        # Spark Structured Streaming job
├── output/                          # Cleaned JSON output
├── archive/                         # Timestamped archives of previous runs
├── docker-compose.yml               # All services defined here
└── README.md
```

## Screenshots :

### 1- [topics.py](#) output








```
PS C:\users\Lenovo\Desktop\data platform\project> python object/topics.py
done
done
done
done
```

## 2- producers.py output :

```
● Producers.py"
sent
sent
sent
sent
sent
sent
sent
sent
sent
sent
sent
sent
sent
sent
sent
sent
sent
```

## 3- Airflow ( manual triggering of the ETL pipeline) :

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions
 local_kafka_spark_etl etl kafka spark	data-engineering		0 14 * * * *	2025-07-15, 10:26:46	2025-07-15, 12:00:00		 

## SPark output (sampels) :

Spark automatically load the events data to tables on postgresql and also make on prem version as json file

On prem json output samples :

## App events :

```
{
  "event_type": "properties",
  "data": {
    "property_id": "3436df8d-5343-4fd4-96e1-3a6fcfccc2f6",
    "type": "apartment",
    "transaction_type": "rent",
    "title": "Luxury 2BR Apartment - Downtown Dubai",
    "price": "120000",
    "location": {
      "latitude": 25.2048,
      "longitude": 55.2708,
      "address": "Burj Khalifa Tower, Downtown Dubai"
    },
    "details": {
      "bedrooms": 2,
      "bathrooms": 2,
      "size_sqft": 1450,
      "amenities": [
        "pool",
        "gym",
        "concierge"
      ],
      "views": 47,
      "saves": 8,
      "status": "available",
      "last_updated": "2025-06-22T10:12:33Z"
    },
    "timestamp": "2025-07-02T12:04:42.658200"
  }
},
{
  "event_type": "properties",
  "data": {
    "property_id": "c80ec64d-4d9b-4700-8528-9a6b11ab2c48",
    "type": "villa",
    "transaction_type": "sale",
    "title": "Beachfront Villa - Palm Jumeirah",
    "price": "8500000",
    "location": {
      "latitude": 25.1123,
      "longitude": 55.1387,
      "address": "Palm Jumeirah, Frond A"
    },
    "details": {
      "bedrooms": 5,
      "bathrooms": 6,
      "size_sqft": 7800,
      "amenities": [
        "private beach",
        "maid's room",
        "smart home"
      ],
      "views": 112,
      "saves": 23,
      "status": "under_offer",
      "last_updated": "2025-06-22T10:07:45Z"
    },
    "timestamp": "2025-07-02T12:04:44.685808"
  }
},
{
  "event_type": "user_activity",
  "data": {
    "event_id": "a2091ea1-ed26-44b2-a522-a5327bfc8db7",
    "user_id": "user_7892",
    "event_type": "property_view",
    "property_id": "properties[0].property_id",
    "duration_sec": 87,
    "device_info": {
      "os": "Android 14",
      "model": "Samsung Galaxy S24",
      "screen_res": "1440x3088"
    },
    "timestamp": "2025-06-22T10:05:12Z"
  },
  "timestamp": "2025-06-22T10:05:12Z"
},
{
  "event_type": "user_activity",
  "data": {
    "event_id": "ef5bbebd-63cf-44d1-a18f-3a22caae3793",
    "user_id": "user_7892",
    "event_type": "property_saved",
    "property_id": "properties[0].property_id",
    "timestamp": "2025-06-22T09:30:00Z"
  },
  "timestamp": "2025-06-22T09:30:00Z"
},
{
  "event_type": "transactions",
  "data": {
    "transaction_id": "9f38c1a8-74f4-4d4d-9db3-0b472ad0be3f",
    "user_id": "user_7892",
    "property_id": "properties[0].prop"
  }
}
```

```
erty_id","type":"reservation_deposit","amount":"5000","payment_method":"ap  
ple_pay","status":"completed","receipt_url":"https://payments.realestateco  
rp.com/r_789234","timestamp":"2025-06-21T16:45:00Z"},"timestamp":"2025-06-  
21T16:45:00Z"}  
  
{"event_type":"users","data":{"user_id":"user_7892","name":"Ahmed  
Mohammed","email":"ahmed.m@example.ae","phone":"+971501234567","preference  
s":{"\locations\":[\Downtown Dubai\","\Palm  
Jumeirah\","\min_bedrooms\":2,\budget_rent\":{"\min\":80000,\max\":1500  
00},\budget_buy\":{"\min\":3000000,\max\":10000000}}},"account_status":"  
verified","registration_date":"2025-01-15T11:20:34Z"},"timestamp":"2025-07  
-02T12:04:48.712670"}  
  
{"event_type":"properties","data":{"property_id":"45b07077-98c0-4259-831c-  
554ccc7e4a83","type":"apartment","transaction_type":"rent","title":"Luxury  
2BR Apartment - Downtown  
Dubai","price":"120000","location":{"\latitude\":25.2048,\longitude\":55  
.2708,\address\":"Burj Khalifa Tower, Downtown  
Dubai"},"details":{"\bedrooms\":2,\bathrooms\":2,\size_sqft\":1450,\  
amenities\":[\pool\","\gym\","\concierge\"]},"views":"47","saved":"8","s  
tatus":"available","last_updated":"2025-06-22T10:12:33Z"},"timestamp":"202  
5-07-02T12:28:26.329112"}  
  
{"event_type":"properties","data":{"property_id":"33d148f0-c0a6-48e2-8a8f-  
41b29882bc7e","type":"villa","transaction_type":"sale","title":"Beachfront  
Villa - Palm  
Jumeirah","price":"8500000","location":{"\latitude\":25.1123,\longitude\  
":55.1387,\address\":"Palm Jumeirah, Frond  
A"},"details":{"\bedrooms\":5,\bathrooms\":6,\size_sqft\":7800,\amen  
ities\":[\private_beach\","\maid's room\","\smart  
home\"]},"views":"112","saved":"23","status":"under_offer","last_updated"  
:"2025-06-22T10:07:45Z"},"timestamp":"2025-07-02T12:28:27.350381"}  
  
{"event_type":"user_activity","data":{"event_id":"b21c4424-00b9-4950-bf06-  
029e9d206471","user_id":"user_7892","event_type":"property_view","property  
_id":"properties[0].property_id","duration_sec":"87","device_info":{"\os\  
":"Android 14","\model\":"Samsung Galaxy  
S24","\screen_res\":"1440x3088"},"timestamp":"2025-06-22T10:05:12Z"},"tim  
estamp":"2025-06-22T10:05:12Z"}  
  
{"event_type":"user_activity","data":{"event_id":"835b833e-c501-4acc-bb31-  
c1fcfc23fbdb","user_id":"user_7892","event_type":"property_saved","propert  
y_id":"properties[0].property_id","timestamp":"2025-06-22T09:30:00Z"},"tim  
estamp":"2025-06-22T09:30:00Z"}
```

```
{"event_type": "transactions", "data": {"transaction_id": "98e07707-fad2-4a3e-ae36-a615fffff8da", "user_id": "user_7892", "property_id": "properties[0].property_id", "type": "reservation_deposit", "amount": "5000", "payment_method": "apple_pay", "status": "completed", "receipt_url": "https://payments.realestatecorp.com/r_789234", "timestamp": "2025-06-21T16:45:00Z"}, "timestamp": "2025-06-21T16:45:00Z"}

{"event_type": "users", "data": {"user_id": "user_7892", "name": "Ahmed Mohammed", "email": "ahmed.m@example.ae", "phone": "+971501234567", "preferences": {"\locations\": [\"Downtown Dubai\", \"Palm Jumeirah\"], \"min_bedrooms\": 2, \"budget_rent\": {\"min\": 80000, \"max\": 150000}, \"budget_buy\": {\"min\": 3000000, \"max\": 10000000}}, \"account_status\": \"verified\", \"registration_date\": \"2025-01-15T11:20:34Z\"}, \"timestamp\": \"2025-07-02T12:28:31.376544\"}
```

## CRM events :

```
{"event": "lead_created", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T10:00:00"}

{"event": "call_logged", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T12:00:00"}

{"event": "lead_created", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T10:00:00"}

{"event": "call_logged", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T12:00:00"}

{"event": "lead_created", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T10:00:00"}

{"event": "call_logged", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T12:00:00"}

{"event": "lead_created", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T10:00:00"}

{"event": "call_logged", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T12:00:00"}

{"event": "lead_created", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T10:00:00"}

{"event": "call_logged", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T12:00:00"}
```

```
{ "event": "lead_created", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T10:00:00" }
{ "event": "call_logged", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T12:00:00" }
{ "event": "lead_created", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T10:00:00" }
{ "event": "call_logged", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T12:00:00" }
{ "event": "lead_created", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T10:00:00" }
{ "event": "call_logged", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T12:00:00" }
{ "event": "lead_created", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T10:00:00" }
{ "event": "call_logged", "lead_id": 101, "agent": "Mona", "timestamp": "2025-06-22T12:00:00" }
```

## ERP events :

```
{ "event_type": "property_listing_created", "property_id": "RE-2025-1001", "address": "123 Palm Boulevard, Dubai Marina", "listing_price": 4500000.0, "currency": "AED", "agent": "Ahmed Al-Farsi", "timestamp": "2025-06-01T09:15:00Z", "sap_transaction_id": "SAP-RE-847392" }
{ "event_type": "buyer_interest_registered", "property_id": "RE-2025-1001", "agent": "Mona Khalid", "buyer_id": "VIP-7892", "buyer_name": "Li Wei Investments LLC", "timestamp": "2025-06-10T14:30:00Z", "sap_transaction_id": "SAP-RE-849573" }
{ "event_type": "sales_contract_initiated", "property_id": "RE-2025-1001", "contract_value": 4350000.0, "status": "under_review", "legal_team": "Smith & Partners", "timestamp": "2025-06-18T11:45:00Z", "sap_transaction_id": "SAP-RE-850294" }
{ "event_type": "deposit_received", "property_id": "RE-2025-1001", "amount": 1000000.0, "payment_method": "bank_transfer", "transaction_ref": "UAE-RE-2025-789234", "timestamp": "2025-06-20T16:20:00Z", "sap_transaction_id": "SAP-RE-851023" }
```



```
{
  "event_type": "property_listing_created",
  "property_id": "RE-2025-1001",
  "address": "123 Palm Boulevard, Dubai Marina",
  "listing_price": 4500000.0,
  "currency": "AED",
  "agent": "Ahmed Al-Farsi",
  "timestamp": "2025-06-01T09:15:00Z",
  "sap_transaction_id": "SAP-RE-847392"
}

{
  "event_type": "buyer_interest_registered",
  "property_id": "RE-2025-1001",
  "agent": "Mona Khalid",
  "buyer_id": "VIP-7892",
  "buyer_name": "Li Wei Investments LLC",
  "timestamp": "2025-06-10T14:30:00Z",
  "sap_transaction_id": "SAP-RE-849573"
}

{
  "event_type": "sales_contract_initiated",
  "property_id": "RE-2025-1001",
  "contract_value": 4350000.0,
  "status": "under_review",
  "legal_team": "Smith & Partners",
  "timestamp": "2025-06-18T11:45:00Z",
  "sap_transaction_id": "SAP-RE-850294"
}

{
  "event_type": "deposit_received",
  "property_id": "RE-2025-1001",
  "amount": 1000000.0,
  "payment_method": "bank_transfer",
  "transaction_ref": "UAE-RE-2025-789234",
  "timestamp": "2025-06-20T16:20:00Z",
  "sap_transaction_id": "SAP-RE-851023"
}

{
  "event_type": "property_listing_created",
  "property_id": "RE-2025-1001",
  "address": "123 Palm Boulevard, Dubai Marina",
  "listing_price": 4500000.0,
  "currency": "AED",
  "agent": "Ahmed Al-Farsi",
  "timestamp": "2025-06-01T09:15:00Z",
  "sap_transaction_id": "SAP-RE-847392"
}

{
  "event_type": "buyer_interest_registered",
  "property_id": "RE-2025-1001",
  "agent": "Mona Khalid",
  "buyer_id": "VIP-7892",
  "buyer_name": "Li Wei Investments LLC",
  "timestamp": "2025-06-10T14:30:00Z",
  "sap_transaction_id": "SAP-RE-849573"
}

{
  "event_type": "sales_contract_initiated",
  "property_id": "RE-2025-1001",
  "contract_value": 4350000.0,
  "status": "under_review",
  "legal_team": "Smith & Partners",
  "timestamp": "2025-06-18T11:45:00Z",
  "sap_transaction_id": "SAP-RE-850294"
}

{
  "event_type": "deposit_received",
  "property_id": "RE-2025-1001",
  "amount": 1000000.0,
  "payment_method": "bank_transfer",
  "transaction_ref": "UAE-RE-2025-789234",
  "timestamp": "2025-06-20T16:20:00Z",
  "sap_transaction_id": "SAP-RE-851023"
}

{
  "event_type": "property_listing_created",
  "property_id": "RE-2025-1001",
  "address": "123 Palm Boulevard, Dubai Marina",
  "listing_price": 4500000.0,
  "currency": "AED",
  "agent": "Ahmed Al-Farsi",
  "timestamp": "2025-06-01T09:15:00Z",
  "sap_transaction_id": "SAP-RE-847392"
}
```

```
{ "event_type": "buyer_interest_registered", "property_id": "RE-2025-1001", "agent": "Mona Khalid", "buyer_id": "VIP-7892", "buyer_name": "Li Wei Investments LLC", "timestamp": "2025-06-10T14:30:00Z", "sap_transaction_id": "SAP-RE-849573" }

{ "event_type": "sales_contract_initiated", "property_id": "RE-2025-1001", "contract_value": 4350000.0, "status": "under_review", "legal_team": "Smith & Partners", "timestamp": "2025-06-18T11:45:00Z", "sap_transaction_id": "SAP-RE-850294" }

{ "event_type": "deposit_received", "property_id": "RE-2025-1001", "amount": 1000000.0, "payment_method": "bank_transfer", "transaction_ref": "UAE-RE-2025-789234", "timestamp": "2025-06-20T16:20:00Z", "sap_transaction_id": "SAP-RE-851023" }

{ "event_type": "property_listing_created", "property_id": "RE-2025-1001", "address": "123 Palm Boulevard, Dubai Marina", "listing_price": 4500000.0, "currency": "AED", "agent": "Ahmed Al-Farsi", "timestamp": "2025-06-01T09:15:00Z", "sap_transaction_id": "SAP-RE-847392" }

{ "event_type": "buyer_interest_registered", "property_id": "RE-2025-1001", "agent": "Mona Khalid", "buyer_id": "VIP-7892", "buyer_name": "Li Wei Investments LLC", "timestamp": "2025-06-10T14:30:00Z", "sap_transaction_id": "SAP-RE-849573" }

{ "event_type": "sales_contract_initiated", "property_id": "RE-2025-1001", "contract_value": 4350000.0, "status": "under_review", "legal_team": "Smith & Partners", "timestamp": "2025-06-18T11:45:00Z", "sap_transaction_id": "SAP-RE-850294" }

{ "event_type": "deposit_received", "property_id": "RE-2025-1001", "amount": 1000000.0, "payment_method": "bank_transfer", "transaction_ref": "UAE-RE-2025-789234", "timestamp": "2025-06-20T16:20:00Z", "sap_transaction_id": "SAP-RE-851023" }
```

## Website events :

```
{ "log_id": "WEB-20250625-001", "event_type": "user_registration", "user_id": "user_4821", "timestamp": "2025-06-25T08:12:45Z" }

{ "log_id": "WEB-20250625-002", "event_type": "property_view", "user_id": "user_4821", "timestamp": "2025-06-25T08:15:33Z" }

{ "log_id": "WEB-20250625-003", "event_type": "search_query", "user_id": "user_4821", "timestamp": "2025-06-25T08:14:02Z" }
```

```
{ "log_id": "WEB-20250625-004", "event_type": "contact_agent", "user_id": "user_4821", "timestamp": "2025-06-25T08:18:21Z" }
{ "log_id": "SYS-20250625-001", "event_type": "property_listing_update", "timestamp": "2025-06-25T09:30:15Z" }
{ "log_id": "WEB-20250625-001", "event_type": "user_registration", "user_id": "user_4821", "timestamp": "2025-06-25T08:12:45Z" }
{ "log_id": "WEB-20250625-002", "event_type": "property_view", "user_id": "user_4821", "timestamp": "2025-06-25T08:15:33Z" }
{ "log_id": "WEB-20250625-003", "event_type": "search_query", "user_id": "user_4821", "timestamp": "2025-06-25T08:14:02Z" }
{ "log_id": "WEB-20250625-004", "event_type": "contact_agent", "user_id": "user_4821", "timestamp": "2025-06-25T08:18:21Z" }
{ "log_id": "SYS-20250625-001", "event_type": "property_listing_update", "timestamp": "2025-06-25T09:30:15Z" }
{ "log_id": "WEB-20250625-001", "event_type": "user_registration", "user_id": "user_4821", "timestamp": "2025-06-25T08:12:45Z" }
{ "log_id": "WEB-20250625-002", "event_type": "property_view", "user_id": "user_4821", "timestamp": "2025-06-25T08:15:33Z" }
{ "log_id": "WEB-20250625-003", "event_type": "search_query", "user_id": "user_4821", "timestamp": "2025-06-25T08:14:02Z" }
{ "log_id": "WEB-20250625-004", "event_type": "contact_agent", "user_id": "user_4821", "timestamp": "2025-06-25T08:18:21Z" }
{ "log_id": "SYS-20250625-001", "event_type": "property_listing_update", "timestamp": "2025-06-25T09:30:15Z" }
```

## How to Run the Project

### Prerequisites

- Docker & Docker Compose
- Python 3.10+

- Java 8+ (for Spark)
- Apache Spark installed locally
- Kafka running locally (or using a mock container)
- Optional: .env file for secrets like Gmail credentials

### **Step 1: Clone the Repo**

```
git clone https://github.com/your-name/data-platform
cd data-platform
```

### **Step 2: Add Environment Variables (Optional but Recommended)**

Create a `.env` file in the root directory:

```
EMAIL_APP_PASSWORD=your_app_password_here
```

### **Step 3: Start the Services**

```
docker-compose up --build
```

This will spin up:

- PostgreSQL
- Redis
- Airflow (Webserver, Scheduler, Init)

Airflow UI will be available at: <http://localhost:8080>

Login credentials:

- Username: `airflow`
- Password: `airflow`

#### Step 4: Place DAG Files

Ensure your `etl_pipeline_dag.py` and its folders are mounted correctly under the `./dags` volume inside Docker.

Airflow loads DAGs from:

`/opt/airflow/dags/`

#### Step 5: Trigger the Pipeline

- Go to Airflow UI
- Unpause the DAG: `local_kafka_spark_etl`
- Click **Trigger DAG**  to test it manually

If successful:

- Output files will appear under `/output`
- Archived versions will appear under `/archive`

## Scheduling

The DAG is scheduled to run **daily at 2:00 PM Cairo time** (UTC+3).

```
schedule="0 14 * * *"
```

```
start_date=pendulum.datetime(2025, 7, 15, 14, 0, tz="Africa/Cairo")
```

## Email Notification Setup

You can configure Airflow to send Gmail alerts on DAG failure.

Update `docker-compose.yml` under `airflow-webserver`:

environment:

- AIRFLOW\_\_SMTP\_\_SMTP\_HOST=smtp.gmail.com
- AIRFLOW\_\_SMTP\_\_SMTP\_STARTTLS=True
- AIRFLOW\_\_SMTP\_\_SMTP\_SSL=False
- AIRFLOW\_\_SMTP\_\_SMTP\_PORT=587
- AIRFLOW\_\_SMTP\_\_SMTP\_MAIL\_FROM=your\_email@gmail.com
- AIRFLOW\_\_SMTP\_\_SMTP\_USER=your\_email@gmail.com
- 

```
AIRFLOW__SMTP__SMTP_PASSWORD=${GMAIL_APP_PASSWORD}
```

## Production Tips & Developer Notes

### What Worked

- Using simple BashOperators to call Spark/Python scripts was straightforward and reliable.
- Volume mounting into Airflow container ensured DAG visibility without extra syncing.
- Archiving JSON output with timestamps provided clear version control of daily runs.

## Common Issues & How to Avoid Them

Issue	Fix
DAG not showing in UI	Ensure the <code>.py</code> file is inside the correct <code>./dags/</code> folder mounted into <code>/opt/airflow/dags</code>
Spark crash with memory error	Restart laptop or limit partitions. Reduce workload per batch
DAG time mismatch	Use <code>pendulum.datetime()</code> with correct time zone like <code>Africa/Cairo</code>
Email not sending	Enable App Passwords in Gmail and use <code>.env</code> to inject secrets
Kafka topic data not visible	Ensure your producers run and publish to the expected topic names

## Debugging DAGs

- Use the Airflow logs tab to trace where failures happen.

You can SSH into the container and manually run the Spark script if needed:

```
docker exec -it airflow-webserver bash
spark-submit dags/spark_jobs/spark_consumer.py
```

## **Final Notes**

This project was built without using cloud resources to simulate a production-like data pipeline locally. You can expand it further by connecting real APIs, logging, alerting, and integrating it with front-end dashboards.