











# A Proclamation Regarding the Noble Data Engineering Books ETL Pipeline Project!

## Table of Contents

1.  Of Overview and Noble Purpose
2.  Of Purpose, Intention, and Worthy Usage
3.  BEHOLD! The Diagrammatic Depiction of the ETL Pipeline
4.  Features of Noble Craft
5.  The Grand Architecture
  -  Components of Ye ETL Pipeline
6.  For the Journeyman Getting Started
  -  Preparations for Thy Quest
  -  Commencement of Deployment
7.  Usage of This Mechanism
8.  Customize to Thy Liking
9.  Project Structure
10.  Known Issues and Their Vanquishment
11.  A Roadmap of Future Glories
12.  The Spirit of Fellowship
13.  License
14.  For the Unversed in Antiquity's Tongue

## Of Overview and Noble Purpose

Greetings, kind scholars and brave data wranglers! Lend thy ears and open thine eyes, for I shall regale thee with the tale of a most wondrous endeavor: the Books ETL Pipeline Project. In this hallowed pursuit, we dost weave together the intricate threads of data extraction, transformation, and loading to uncover knowledge most profound.

This grand mechanism, devised by tireless toil and wisdom, doth unite the realms of Python, Docker, PostgreSQL, and Airflow. By its might, one may harvest bookly treasures from the vast libraries of OpenLibrary and Google Books, cleanse and refine them, and store them in databanks for enlightenment and analysis.

Lo, this project is not merely a tool but a masterwork that doth exemplify the art and science of data engineering. Scholars, practitioners, and seekers of wisdom alike may find value herein, as it is both a tome of learning and a marvel of modern craft.

Thus, embark, good reader, upon this journey of discovery, and let the annals of data yield their secrets unto thee!

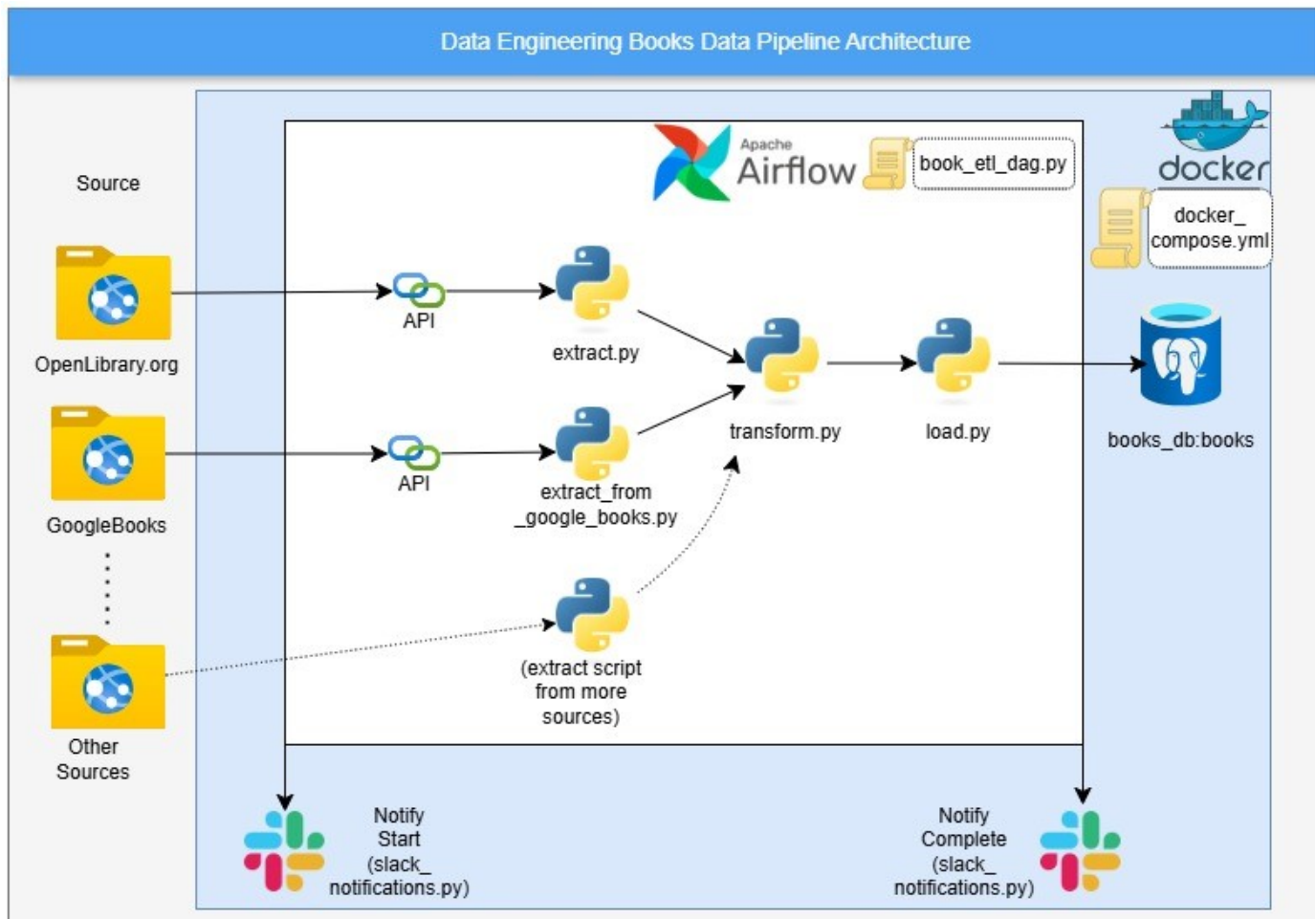
## 📖 Of Purpose, Intention, and Worthy Usage 📖

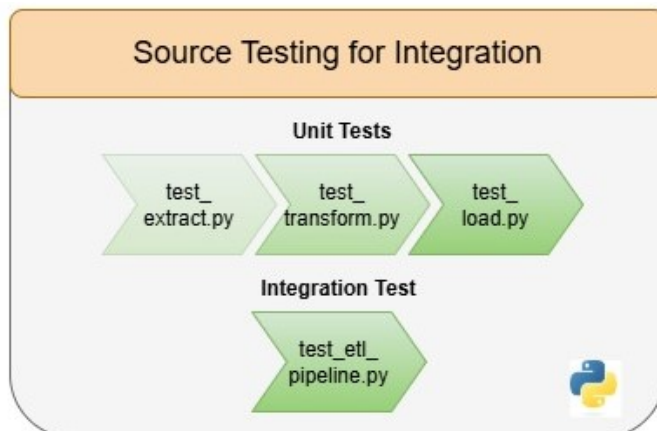
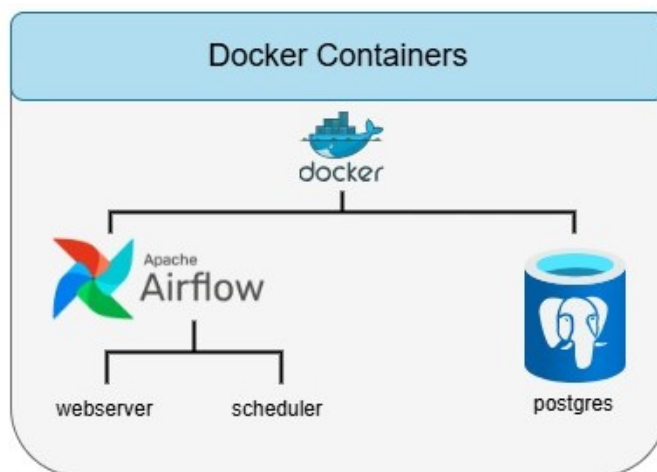
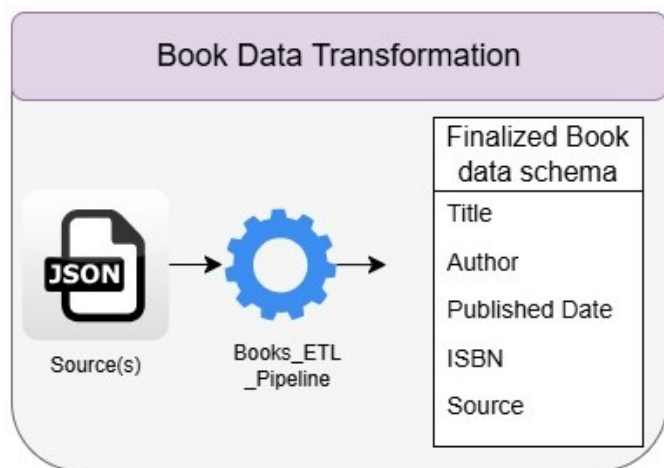
Hark! This noble endeavor is fashioned to fetch and hold knowledge, tracking the comings and goings of books upon the digital shelves. But lo! Its utility extendeth far beyond the boundaries of this humble purpose. Prithree know, fair user, that **thou mayest adapt its workings to suit thine own curiosities**. By a simple **tweak of query**, *thou mayest turn this engine toward thine own pursuits*—be it tracking wares, scrolls, or other matters of great import. Wield this tool as thy will decrees, and may it serve thee well in thy noble quests!

## 🖼️ BEHOLD! The Diagrammatic Depiction of the ETL Pipeline 🖼️

Hear ye, hear ye! Gather thy gaze upon this most wondrous depiction of the grand ETL pipeline!

Within its bounds, thou shalt witness the harmonious interplay of myriad parts, each a vital cog in this celestial mechanism. From Security Sanctuaries to ensure the sacred safety of thine operations, to the Testing Grounds whereupon thy code is proven and hardened, this diagram illustrates the majestic flow of data, transformed from its humble JSON origins into a regal table of fields—fit for analysis and insight.





**Security:** Lo, the bastions of access control and protection, ensuring no ill-begotten hand may meddle with the data's purity.

**Testing:** Prithee, regard this as the proving grounds where robustness is forged, where bugs are vanquished, and the pipeline stands resilient.

**Docker Enclosure:** Witness the orchestration of containers, wherein each component dwelleth in isolation yet communicateth with precision, making the entire pipeline agile and portable.

**Data Extraction:** Here lieth the cradle of our endeavor, whence data is lifted from its JSON confines and set forth upon its transformative journey.

**Data Transformation:** The alchemy of the pipeline! Fields are cleansed, shaped, and readied for their destined purpose. Here, titles, authors, years, and sources are refined into their final glorious forms.

**Final Table:** The culmination of all labors! Behold the tabular majesty, wherein the fruits of thy efforts—titles, authors, publication years, and more—stand ready to enlighten thy endeavors.

**Airflow Sorcery:** Marvel at the enchanted scheduler, tirelessly orchestrating the pipeline's every step with grace and precision.



## DAG: book\_etl\_dag

success

Schedule: @daily

Next Run: 2024-12-04, 00:00:00

Grid

Graph

Calendar

Task Duration

Task Tries

Landing Times

Gantt

Details

&lt;&gt; Code

Audit Log



2024-12-03T13:49:29-05:00 Runs 25 Run manual\_\_2024-12-03T18:49:28.405602+00:00

Layout Left &gt; Right

Update

Find Task...

PythonOperator

deferred

failed

queued

removed

restarting

running

scheduled

shutdown

skipped

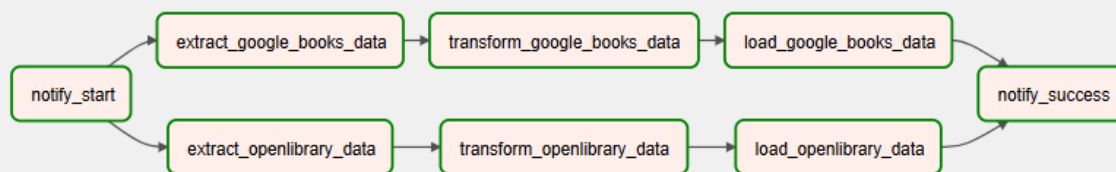
success

up\_for\_reschedule

up\_for\_retry

upstream\_failed no\_status

Auto-refresh



Here, in this tableau of wisdom, the ETL process cometh alive. Gaze upon its intricacies, for herein liest not just a method but a marvel, where chaos is tamed and knowledge is borne.

## 🌟 Features of Noble Craft

- 🛠️ **Extraction of Many Founts:** Gathers knowledge from the OpenLibrary and Google Books APIs, like a wise scholar pulling treasures from ancient tomes.
- 🧹 **Purification of Data:** Cleanseth and enriches the raw information, ensuring it is fair and fit for study.
- 📊 **Integration with the Repository of Postgres:** Deposits the bounty into a steadfast database for safekeeping and recall.
- 🛡️ **Defenses and Logging of Errors:** Implements vigilant sentinels to guard against mishaps and record the chronicles of the pipeline.

Airflow DAGs Datasets Security Browse Admin Docs 14:23 EST (-05:00) AA

DAG: book\_etl\_dag Schedule: False

Grid Graph Calendar Task Duration Landing Times Gantt Details Code Audit Log

Task Instance: load\_google\_books\_data at 2024

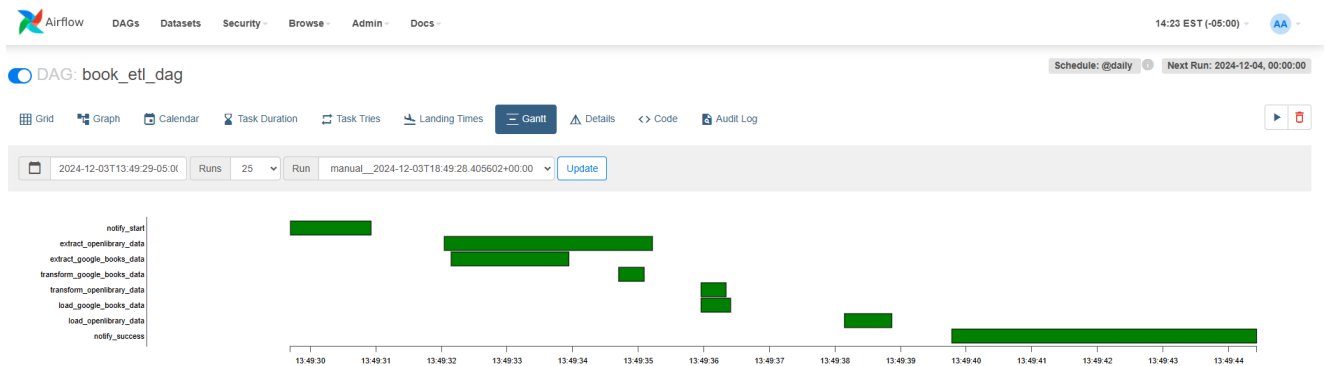
Task Instance Details Rendered Template Log XCom

Log by attempts

```
1
*** Log file does not exist: /opt/airflow/logs/dag_id=book_etl_dag/run_id=manual_2024-12-03T18:49:28.405602+00:00/task_id=load_google_books_data/attempt=1.log
*** Fetching from: http://6ace03a8ee2:8793/log/dag_id=book_etl_dag/run_id=manual_2024-12-03T18:49:28.405602+00:00/task_id=load_google_books_data/attempt=1.log

[2024-12-03, 13:49:35 EST] [taskinstance.py:1087] INFO - Dependencies all met for <TaskInstance: book_etl_dag.load_google_books_data manual_2024-12-03T18:49:28.405602+00:00 [queued]>
[2024-12-03, 13:49:35 EST] [taskinstance.py:1087] INFO - Dependencies all met for <TaskInstance: book_etl_dag.load_google_books_data manual_2024-12-03T18:49:28.405602+00:00 [queued]>
[2024-12-03, 13:49:35 EST] [taskinstance.py:1283] INFO -
-----
[2024-12-03, 13:49:35 EST] [taskinstance.py:1284] INFO - Starting attempt 1 of 2
[2024-12-03, 13:49:35 EST] [taskinstance.py:1285] INFO -
-----
[2024-12-03, 13:49:36 EST] [taskinstance.py:1304] INFO - Executing <Task(PythonOperator): load_google_books_data> on 2024-12-03 18:49:28.405602+00:00
[2024-12-03, 13:49:36 EST] [standard_task_runner.py:55] INFO - Started process 278 to run task
[2024-12-03, 13:49:36 EST] [standard_task_runner.py:82] INFO - Running: ['***', 'tasks', 'run', 'book_etl_dag', 'load_google_books_data', 'manual_2024-12-03T18:49:28.405602+00:00', '--job-id', '12', '--raw', '--subdir', 'DAGS_FOLDER/book_etl_dag.py', '--cfg-
[2024-12-03, 13:49:36 EST] [standard_task_runner.py:85] INFO - Job ID: Subtask load_google_books_data
[2024-12-03, 13:49:36 EST] [loggingixin.py:137] WARNING - /home/***/local/lib/python3.7/site-packages/****/settings.py:249 DeprecationWarning: The sqlalchemy_conn option in [core] has been moved to the sqlalchemy_conn option in [database] - the old setting
[2024-12-03, 13:49:36 EST] [task_command.py:389] INFO - Running <TaskInstance: book_etl_dag.load_google_books_data manual_2024-12-03T18:49:28.405602+00:00 [running]> on host 6ace03a8ee2
[2024-12-03, 13:49:36 EST] [taskinstance.py:1513] INFO - Exporting the following env vars:
AIRFLOW_CTX_DAG_OWNER=***
AIRFLOW_CTX_DAG_ID=book_etl_dag
AIRFLOW_CTX_TASK_ID=load_google_books_data
AIRFLOW_CTX_EXECUTION_DATE=2024-12-03T18:49:28.405602+00:00
AIRFLOW_CTX_TRY_NUMBER=1
AIRFLOW_CTX_DAG_RUN_ID=manual_2024-12-03T18:49:28.405602+00:00
[2024-12-03, 13:49:36 EST] [load.py:26] INFO - Starting the data loading process...
[2024-12-03, 13:49:36 EST] [load.py:30] INFO - Ensuring the 'books' and 'metadata' tables exist.
[2024-12-03, 13:49:36 EST] [load.py:56] INFO - Inserting new books into the 'books' table.
```

- 🕒 **Automation of Timely Tasks:** Employeth the magic of Airflow to schedule thy tasks, ensuring they commence with precision.



- 📧 **Slack Heraldry:** Dispatches messengers to announce the state of thine efforts in real-time.

data engineering books pipeline Search data engineering books pipeline

# all-data-engineering-books-pipeline

Messages Company Handbook

Wednesday, November 27th

ETL Pipeline: Process completed successfully

data engineering books etl pipeline ETL Pipeline: Starting the process

data engineering books etl pipeline ETL Pipeline: Starting the process

data engineering books etl pipeline ETL Pipeline: Starting the process

data engineering books etl pipeline ETL Pipeline: Failed! Error: '>' not supported between instances of 'str' and 'int'

Monday, December 2nd

data engineering books etl pipeline ETL Pipeline: Starting the process

data engineering books etl pipeline ETL Pipeline: Failed! Error: '>' not supported between instances of 'str' and 'int'

data engineering books etl pipeline ETL Pipeline: Starting the process

data engineering books etl pipeline ETL Pipeline: Process completed successfully

data engineering books etl pipeline ETL Pipeline: Starting the process

data engineering books etl pipeline ETL Pipeline: Process completed successfully

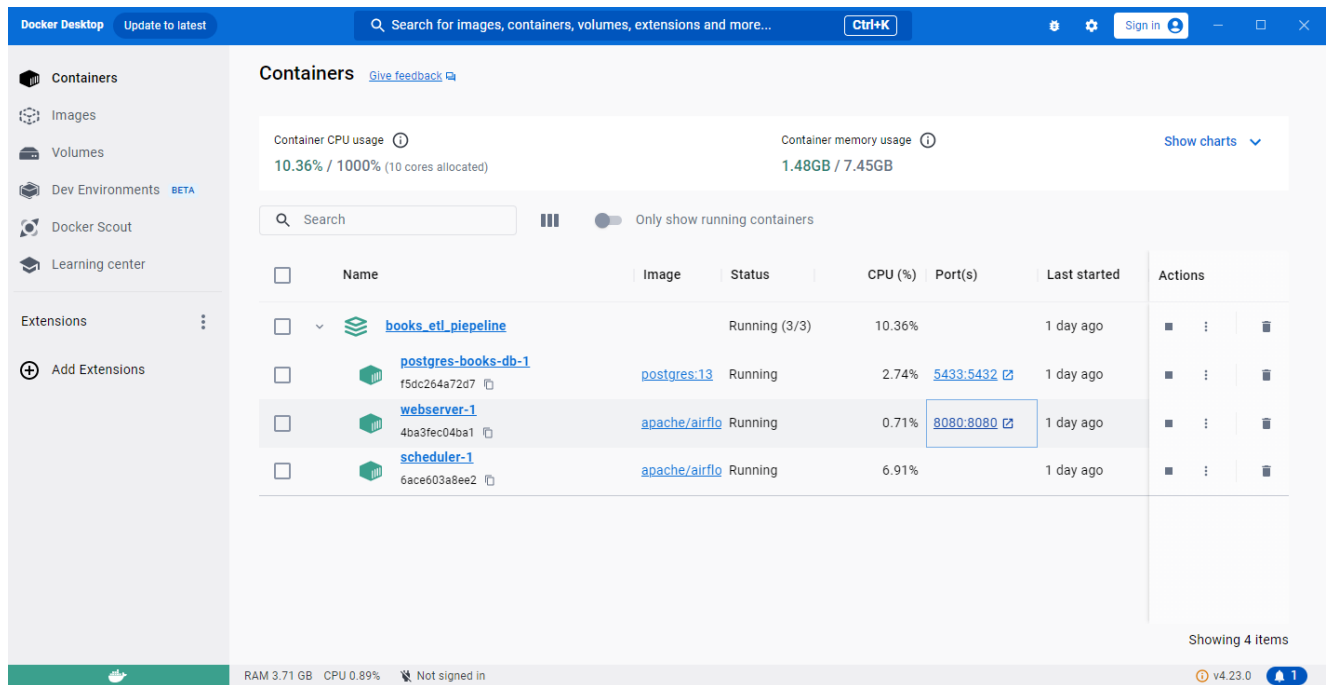
Yesterday

data engineering books etl pipeline ETL Pipeline: Starting the process

B I Link Embed Table Code View

Message #all-data-engineering-books-pipeline


- 🐳 **Encasement in Docker's Vessel:** Encircles the pipeline in the aegis of Docker for deployment and scaling to lands far and wide.




# The Grand Architecture

## Components of Ye ETL Pipeline


### 1. Extraction:

- Summoneth data from OpenLibrary and Google Books .
- Handles peculiarities of pagination and rate limits, like a skilled juggler with flaming torches.



### 2. Transformation:

- Cleanseth and standardizes the records .
- Resolves missing fields and maketh the data ready for usage.



### 3. Loading:

- Deposits the enriched bounty into Postgres' eternal vaults .
- Employeth conflict resolution to smite duplicate entries.

### 4. Orchestration:

- Commands the dance of tasks through an Airflow DAG .
- Schedules and retries with the wisdom of experience .

### 5. Containerization:

- Packages all components within Docker's might vessel .
- Uses Docker Compose to steer the ships .

## 6. 🛎️ **Monitoring:**

- Announceth pipeline statuses via Slack 📱.
- Airflow's interface reveals all activity 📄.

# For the Journeyman Getting Started

## ⚙️ **Preparations for Thy Quest**

- 🐳 Docker & Docker Compose
- 🐍 Python 3.8 or above
- 📄 requirements.txt should provide thee with required Python incantations
- 🗝️ Slack Token, should thou seek notifications
- 🛠️ Basic wit in SQL and Python

## **Commencement of Deployment**

### 1. **Cloneth the repository:**

```
git clone https://github.com/VBlackie/books_etl.git
cd Books_ETL_Pipeline
```

### 2. **Declare Thy Secrets:** Create a .env file with:

```
POSTGRES_USER=airflow
POSTGRES_PASSWORD=airflow
POSTGRES_DB=books_db
AIRFLOW_WEBSERVER__SECRET_KEY=<your_secret_key>
AIRFLOW_ADMIN_USERNAME=admin
AIRFLOW_ADMIN_PASSWORD=admin
SLACK_CHANNEL=<your-slack-channel>
SLACK_API_TOKEN=<your-slack-api-token>
GOOGLE_BOOKS_API_KEY=<your-google-books-api-key>
```

### 3. **Raise Thy Docker Containers:**

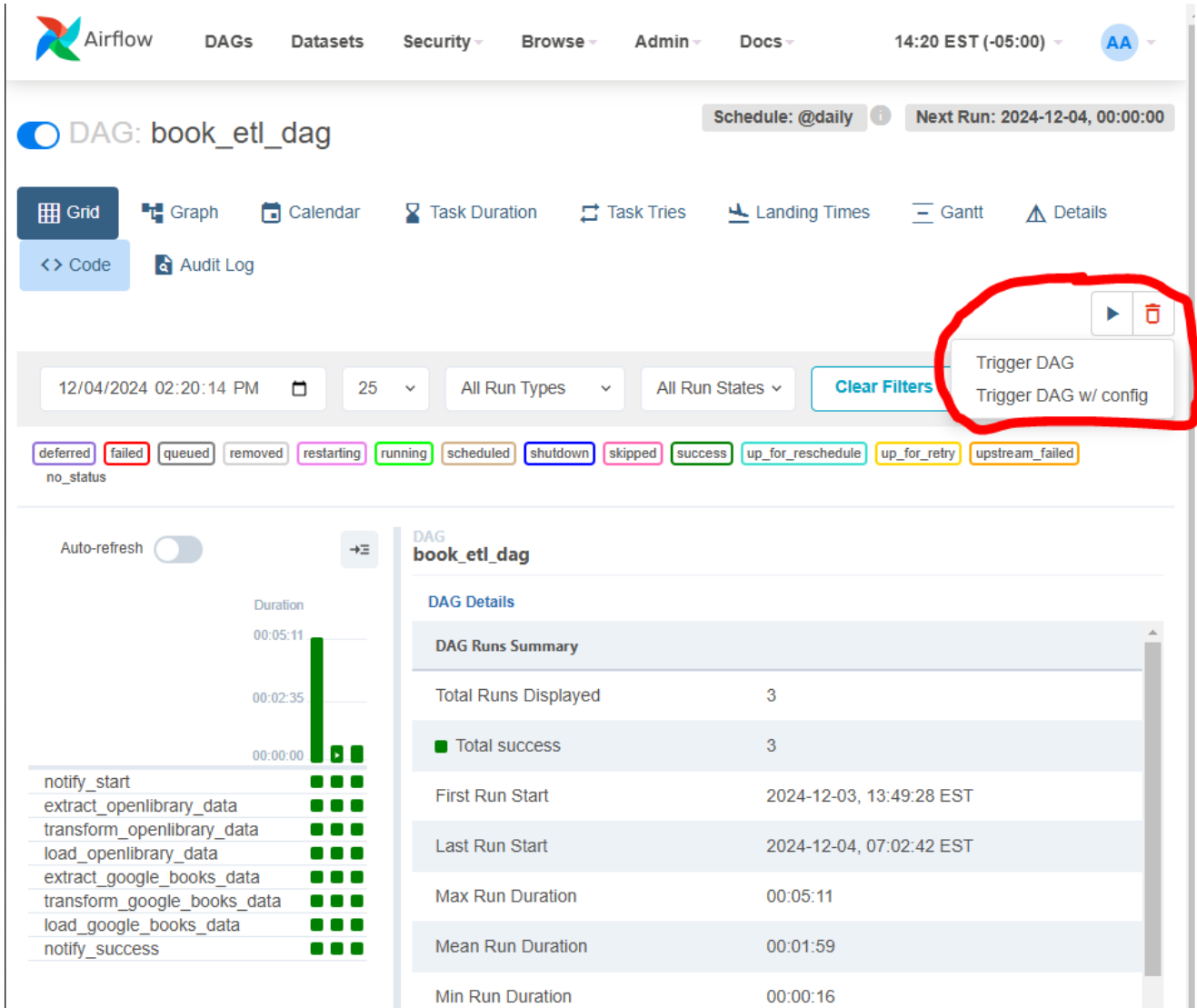
```
docker-compose up --build
```

### 4. **Enter the Interface of Airflow:**

- Navigate to <http://localhost:8080>.
- Credentials: Username: admin Password: admin

# Usage of This Mechanism

- Command Thy Pipeline through the hallowed Airflow UI .



**Airflow** DAGs Datasets Security Browse Admin Docs 14:20 EST (-05:00) AA

**DAG: book\_etl\_dag** Schedule: @daily Next Run: 2024-12-04, 00:00:00

Grid Graph Calendar Task Duration Task Tries Landing Times Gantt Details

<> Code Audit Log

12/04/2024 02:20:14 PM 25 All Run Types All Run States Clear Filters

Trigger DAG Trigger DAG w/ config

deferred failed queued removed restarting running scheduled shutdown skipped success up\_for\_reschedule up\_for\_retry upstream\_failed no\_status

Auto-refresh


**DAG book\_etl\_dag**


**DAG Details**

**DAG Runs Summary**

Total Runs Displayed	3
Total success	3
First Run Start	2024-12-03, 13:49:28 EST
Last Run Start	2024-12-04, 07:02:42 EST
Max Run Duration	00:05:11
Mean Run Duration	00:01:59
Min Run Duration	00:00:16



- 
[DAGs](#)
[Datasets](#)
[Security](#)
[Browse](#)
[Admin](#)
[Docs](#)
14:21 EST (-05:00)
AA


DAG: book\_etl\_dag

Schedule: @daily
Next Run: 2024-12-04, 00:00:00

[Grid](#)
[Graph](#)
[Calendar](#)
[Task Duration](#)
[Task Tries](#)
[Landing Times](#)
[Gantt](#)
[Details](#)

[Code](#)
[Audit Log](#)

This view displays selected events and operations that have been taken on this dag. The included and excluded events are set in the Airflow configuration, which by default remove view only actions. For a full list of events regarding this DAG, click [here](#).

Time ⌵	Task ID ⌵	Event ⌵	Logical Date ⌵	Owner ⓘ	Details
2024-12-04, 07:03:11	notify_success	success	2024-12-03 05:00:00+00:00	airflow	None
2024-12-04, 07:03:10	notify_success	cli_task_run	None	airflow	{"host_name": "6ace603a8ee2", "full_command": "[/home/airflow/.local/bin/airflow', 'scheduler']"]}
2024-12-04, 07:03:10	notify_success	running	2024-12-03 05:00:00+00:00	airflow	None
2024-12-04, 07:03:09	notify_success	cli_task_run	None	airflow	{"host_name": "6ace603a8ee2", "full_command": "[/home/airflow/.local/bin/airflow', 'scheduler']"]}
2024-12-04,	load_openlibrary_data	success	2024-12-03	airflow	None

- The screenshot shows the pgAdmin 4 web interface. On the left is a sidebar with a tree view of the database structure. The main area is divided into a query editor and a results pane. The query editor contains a SQL query that selects all records from the 'public.books' table, ordered by 'id' in ascending order. The results pane displays the output of this query as a table with four columns: 'author', 'published\_date', 'isbn', and 'source'. The table contains 25 rows of data, representing various books and their publication details.

	author	published_date	isbn	source
83	Xiaofei He, Xinbo Gao, Yanning Zhang, Zhi-Hua Zhou, Zhi-Yong Liu	2015	9783192398997	OpenLibrary
84	IEEE Computer Society Technical Committee on Data Engineering	1994	0818654007	OpenLibrary
85	Kevin Beaver, Kevin M. Beaver	2004	9780470602683	OpenLibrary
86	Martin Atzmueller, Samia Oussena, Thomas Roth-Berghofer	2016	9781522502937	OpenLibrary
87	Paco Nathan	2020	9781098115043	OpenLibrary
88	Jagdish Chandra Patni	2024	9788981138407	OpenLibrary
89	Mark Hinders	2024	9781394271245	OpenLibrary
90	Michele Pinto, Sammy El Khammal	2023	9781804616024	OpenLibrary
91	Kent Graziano	2015	1796584992	OpenLibrary
92	Ford Lumban Gaol	2013	9783642288074	OpenLibrary
93	International Conference on Data Engineering (4th 1988 Los Angeles, Calif.)	1988	9780818688270	OpenLibrary
94	Dan Sullivan	2020	1119618436	OpenLibrary
95	Ford Lumban Gaol	2014	97836424441330	OpenLibrary
96	Vikrant Bhatnaja, Lai Khin Wee, Jerry Chun-Wei Lin, Suresh Chandra Satapathy, T. M. Rajesh	2022	9789811915581	OpenLibrary
97	Turkey) International Conference on Data Engineering (23rd 2007 Istanbul	2007	1424408024	OpenLibrary
98	Florida National University	2010	9781609604509	OpenLibrary
99	P. S. Yu	1995	9780818669101	OpenLibrary
100	Lhoussaine Alla, Aziz Hmioui, Badr Bentaha	2024	97893693931729	OpenLibrary
101	Niranjan N. Chiplunkar, Takaroni Fukao	2020	9811535132	OpenLibrary
102	and Data Engineer... Nabendu Chaki, Nagargur Devarakonda, Anirban Sarkar, Narayan C. Debnath	2018	9819906083	OpenLibrary
103	Tobias Macey	2021	9781492062417	OpenLibrary
104	Santi Caballá, Jordi Conesa	2018	9783319463171	OpenLibrary

- Connect unto the database with the following credentials:

- Host: localhost
- Port: 5433
- Username: airflow
- Password: airflow
- Database Name: books\_db

Alternatively, should thou be inclined to use the command line:

```
psql -h localhost -p 5433 -U airflow -d books_db
```

## Customize to Thy Liking

### Modify the Query to Suit Thy Quest

Dost thou seek knowledge beyond data engineering? Fear not, for the script is designed to be molded to thy whims! Within the sacred function `extract_books_data` in `extract.py`, thou shalt find the query:

```
def extract_books_data():
    url = 'https://openlibrary.org/search.json?q=data+engineering' # Focused query on data engineering
```

Replace 'data+engineering' with the essence of thy pursuit. Forsooth, be it "philosophy", "alchemy", or any subject dear to thee, and lo, the knowledge shall be fetched accordingly.

### Add Thine Own Sources

Should thee wish to extend the reach of this mechanism, thou mayst craft a new script for extracting data. To ensure thy creation aligns with the grand pipeline, thou must honor the sacred format of the `transform.py` script. The records must be transformed thusly:

```
transformed_data.append({
    'title': book['title'],
    'author': book['author'],
    'published_date': book['published_date'],
    'isbn': book['isbn'],
    'source': book['source']
})
```

This ensures the data from thy new source melds seamlessly with the rest of the enriched tome of knowledge.

### Test for Compatibility and Righteousness

Ere thou dost deploy thy customizations, ensure thy worketh withstands the trials of unit tests. Use the tests provided within the tests/ realm to confirm compatibility. The command to summon the trials is:

```
pytest tests/
```








Run this incantation within thy project's sanctuary to verify thy changes passeth all scrutiny.

## Heed These Words

By following these steps, thou canst tailor this repository to serve thy most peculiar pursuits. Modify, expand, and test—this pipeline shall bend to thy will whilst retaining its elegance and might.

## Project Structure

 Books\_ETL\_Pipeline/

```
|—  dags/
| |— book_etl_dag.py # DAG of Airflow
| |— extract.py # Gatherer of Data from OpenLibrary
| |— extract_from_google.py # Gatherer of Data from GoogleBooks
| |— transform.py # Purifier of Records
| |— load.py # Depositor of Information
| |— slack_notifications.py # Herald of Notifications
|—  logs/ # Chronicles of Airflow
|—  plugins/ # Custom Enhancements
|—  tests/ # Realm of Testing and Validation
| |— test_extract.py # Examiner of Gatherer Logic
| |— test_transform.py # Scrutinizer of Data Purification
| |— test_load.py # Overseer of Data Deposition
| |— test_etl_pipeline.py # Examiner of Integrity
|—  docker-compose.yml # Configuration of the Fleet
|—  requirements.txt # The Scroll of Dependencies
|—  .env # Hidden Secrets
```

## Known Issues and Their Vanquishment

### 1. Scheduler Heartbeat Falters .

- Ensure Airflow volumes are intact.
- Use docker system prune -f to cleanse thy setup.

### 2. SQL Insert Woes .

- Ensure table schema matches the load.py script.

### 3. Log Vanish into the Ether 🙈:

- Verify mappings in docker-compose.yml

### 4. The Goblins of Slumber Delay Thy Database 🐉:

- At first run, the database machinery doth refuse to awaken promptly, for the goblins within linger in slumber.
- Prithee, restart thy services twice, and lo, the machinery shall spring to life!

## A Roadmap of Future Glories

- 🌐 Extend support to Goodreads or others.
- 📊 Bind the pipeline with Metabase for noble visualization.
- 📈 Enhance metadata reporting.
- 🚀 Embrace CI/CD for automated testing.

## The Spirit of Fellowship

Contributions are welcome! Sharpen thy code and submit thy Pull Requests. Together, let us make this project legendary ✨.

## License

This project is bestowed under the MIT License. It is free to use, modify, and cherish.

## For the Unversed in Antiquity's Tongue

A Glossary of Ye Olde Terms

Fear not, gentle reader, should the flowery language of this proclamation confound thee! Below is a humble guide to the more curious words thou mayst encounter within this hallowed text:

- Alas! – A cry of sorrow or regret, used to express lamentation. Example: "Alas! The goblins of slumber delay thy database!"
- Anon – Soon, shortly, in a little while. Example: "Deploy thy pipeline anon and uncover treasures untold!"
- Behold! – Look upon this with awe and wonder! Example: "Behold! The Diagrammatic Depiction of the ETL Pipeline!"
- Doth – An archaic form of 'does,' used for emphasis. Example: "Lo, this project doth exemplify the art of data engineering."

- Hark! – Pay heed! Listen well, for what follows is of utmost importance. Example: "Hark! This noble endeavor is fashioned to fetch and hold knowledge!"
- Hear ye! Hear ye! – An announcement or proclamation, commanding attention. Example: "Hear ye, hear ye! Gather thy gaze upon this most wondrous depiction!"
- Lo! – Behold! A word to draw attention to something noteworthy. Example: "Lo, this pipeline is not merely a tool but a masterwork!"
- Methinks – I believe, I consider, or it seems to me. Example: "Methinks this endeavor shall serve thee well in thy noble quest!"
- Prithee – I entreat thee, or I ask of thee. Example: "Prithee know, fair user, that thou mayest adapt its workings."
- Thou/Thy/Thee/Thine – You/Your/To You/Yours (respectively). Example: "Command thy pipeline and monitor thy logs with diligence."
- Verily – Truly, indeed, without a doubt. Example: "Verily, this mechanism is a marvel of data engineering!"