

# Data Pipeline Report

## Team Members:

Serik Bakhrayev 20B030585

Sadykova Yasmin 22B22B1540

Birlizhanova Aruzhan 22B030329

### - Data Source (API) Source:

API - <https://api.eia.gov/v2/electricity/rto/fuel-type-data/data/>

Public API with power generation data API returns data in JSON format.

Sample data:

- period — date and hour
- respondent — electricity market operator
- fueltype — fuel type
- value — electricity volume

The API is read-only and requires an API key for access.

### - Virtual environment and libraries

A Python virtual environment was created to isolate project dependencies:

python -m venv venv

venv/bin/activate (Windows: venv\Scripts\activate)

All required libraries were installed from the requirements.txt file:

pip install -r requirements.txt

The requirements.txt file contains all dependencies necessary to run

Kafka producers/consumers, data processing with Pandas, and Airflow DAGs.

### - Kafka (data streaming)

Kafka launch Kafka was installed via Homebrew and launched:  
brew services start kafka  
Creating a Kafka topic kafka-topics --create \ --topic raw\_electricity  
\ --bootstrap-server localhost:9092 \ --partitions 1 \ --replication-factor 1  
Kafka Verification  
kafka-console-consumer \ --topic raw\_electricity \ --from-beginning \ --bootstrap-server  
localhost:9092

**- Job 1 — Ingestion** (API → Kafka)  
File src/job1\_producer.py  
What does \* Makes a request to the API \* Receives JSON \* Sends each record to Kafka topic raw\_electricity Manual verification python src/job1\_producer.py If consumer shows JSON— ingestion works.

**- Job 2 — Cleaning & Storage** (Kafka → SQLite)  
Files src/job2\_cleaner.py src/db\_utils.py  
What does \* Reads data from Kafka \* Clears data: \* converts dates \* converts value to a number \* deletes incorrect values \* Creates an SQLite database \* Saves data to the events table Manual launch python src/job2\_cleaner.py Checking the database sqlite3 data/app.db  
SELECT COUNT(\*) FROM events;

**- Job 3 — Analytics** (SQLite → Aggregation)  
File src/job3\_analytics.py  
What does \* Reads data from the events table \* Groups by: \* date \* type of fuel \* Counts: \* average value \* maximum \* the amount \* Saves the result to the daily\_summary table Manual launch python src/job3\_analytics.py

- **Airflow (orchestration)** DAG files airflow/dags/job1\_ingestion\_dag.py  
 airflow/dags/job2\_clean\_store\_dag.py airflow/dags/job3\_daily\_summary\_dag.py DAG 1 — job1\_ingestion Purpose: continuous data loading Schedule: None (starts manually) Reason: ingestion is a continuous pipeline (by assignment condition). DAG 2 — job2\_clean\_store Purpose: cleaning and preservation Schedule: 0 \* \* \* \* (every hour) DAG 3 — job3\_daily\_summary Purpose: Schedule Analytics: 0 0 \* \* \* (once a day)

- **The launch of Airflow** Initialization airflow db init Launch airflow standalone Web UI <http://localhost:8080>

- **Checking in the Airflow UI** Checked in the UI:  
 \* DAGs are visible  
 \* job2 and job3 are running on schedule  
 \* job1 starts manually  
 \* the data is successfully traversing the entire pipeline

	id	period	respondent	fueltype
2	2	2025-12-17T07:00:00...	AVA	OTH
3	3	2025-12-17T07:00:00...	AVA	SUN
4	4	2025-12-17T07:00:00...	AVA	WAT
5	5	2025-12-17T07:00:00...	AVA	WND
6	6	2025-12-17T07:00:00...	AVRN	NG
7	7	2025-12-17T07:00:00...	AVRN	SUN
8	8	2025-12-17T07:00:00...	AVRN	WAT
9	9	2025-12-17T07:00:00...	AVRN	WND
10	10	2025-12-17T07:00:00...	BANC	NG
11	11	2025-12-17T07:00:00...	BANC	OTH
12	12	2025-12-17T07:00:00...	BANC	SUN

SQLite database — events table with cleaned data.

data > app.db

Search tables... Reset Filters Records: 24 Search 24 records...

Tables (3)

- > events
- > sqlite\_sequence
- > daily\_summary

	date	fueltype	avg_value	max_value	total_value
1	025-12-17	BAT	7	14	28
2	025-12-17	COL	1619.8333333333...	9252	9719
3	025-12-17	GEO	589.25	1167	2357
4	025-12-17	NG	2313	9055	32382
5	025-12-17	NUC	3054.8	11087	15274
6	025-12-17	OIL	143.6666666666...	770	862
7	025-12-17	OTH	461.3636363636364	1498	5075
8	025-12-17	PS	0	0	0
9	025-12-17	SNB	0.4	1	2
10	025-12-17	SUN	2.88888888888889	26	26
11	025-12-17	WAT	1243.625	9061	19898
12	025-12-17	WND	807.3636363636364	2222	8881
13	025-12-18	BAT	0.333333333333...	1	1
14	025-12-18	COL	1463.1666666666...	8278	8779
15	025-12-18	GEO	588	1164	2352
16	025-12-18	NG	1850	7594	25900
17	025-12-18	NUC	2898.6	11094	14493
18	025-12-19	OTH	1463.1666666666...	8278	8779

Try SQLite Viewer in the browser ↗

SQLite database — daily\_summary table with aggregated analytics results.

Dags Runs Task Instances

Search Dags Advanced Search ✎

All Failed Running Success Enabled Filter by tag × Reset 1 filter Sort by Latest Run Start Date...

3 Dags

job1\_ingestion

Schedule Latest Run Next Run  
2025-12-18, 23:43:44 ⏱️

job3\_daily\_summary

Schedule Latest Run Next Run  
0 0 \* \* \* 2025-12-18, 23:43:47 ⏱️ 2025-12-19, 05:00:00 ⏱️

job2\_clean\_store

Schedule Latest Run Next Run  
0 \* \* \* \* 2025-12-18, 23:43:49 ⏱️ 2025-12-19, 00:00:00 ⏱️

Launchpad

Dags Runs Task Instances

Search Dags Advanced Search ✎

All Failed Running Success All Filter by tag × Reset 1 filter Sort by Latest Run Start Date...

3 Dags

job1\_ingestion

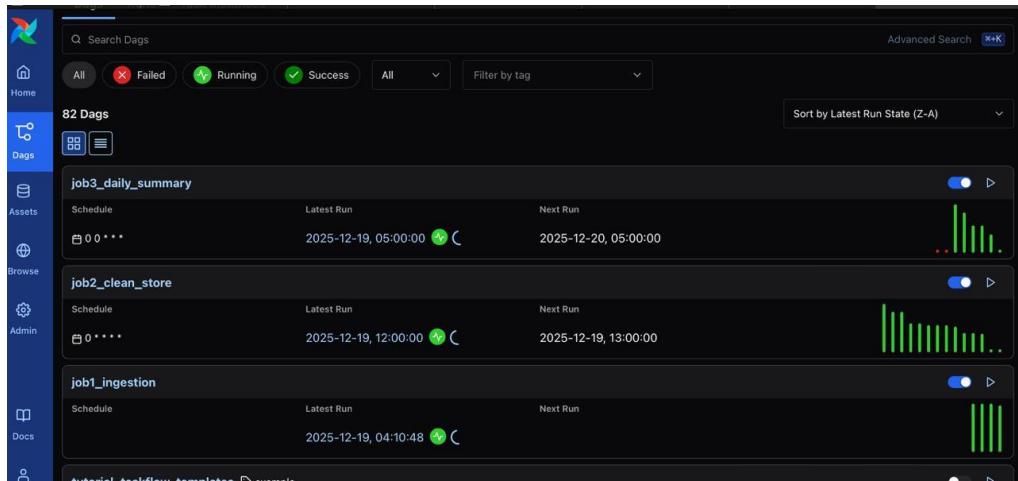
Schedule Latest Run Next Run  
2025-12-18, 18:01:27 ⏱️

job3\_daily\_summary

Schedule Latest Run Next Run  
0 0 \* \* \* 2025-12-18, 18:01:32 ⏱️ 2025-12-19, 05:00:00 ⏱️

job2\_clean\_store

Schedule Latest Run Next Run  
0 \* \* \* \* 2025-12-18, 23:00:00 ⏱️ 2025-12-19, 00:00:00 ⏱️



Airflow UI showing all DAGs and their successful runs

Successful DAG execution in Airflow

Successful task log — data cleaning and storage

Successful task log — daily analytics aggregation