

Департамент образования города Москвы
Государственное автономное образовательное учреждение
высшего образования города Москвы
«Московский городской педагогический университет»

Институт цифрового образования
Департамент информатики, управления и технологий

ДИСЦИПЛИНА:

«Проектный практикум по разработке ETL-решений »

Практическая работа 21.03

Выполнила:

Студентка группы АДЭУ-211

Кравцова Алёна Евгеньевна

Руководитель:

Босенко Т.М

Москва

2025

Задание 1.3. Запуск кейса umbrella

Перейдем на сайт `weather.api` для получения личного ключа (Рис. 1) и далее добавим его в код `real_umbrella.py` (Рис. 2).

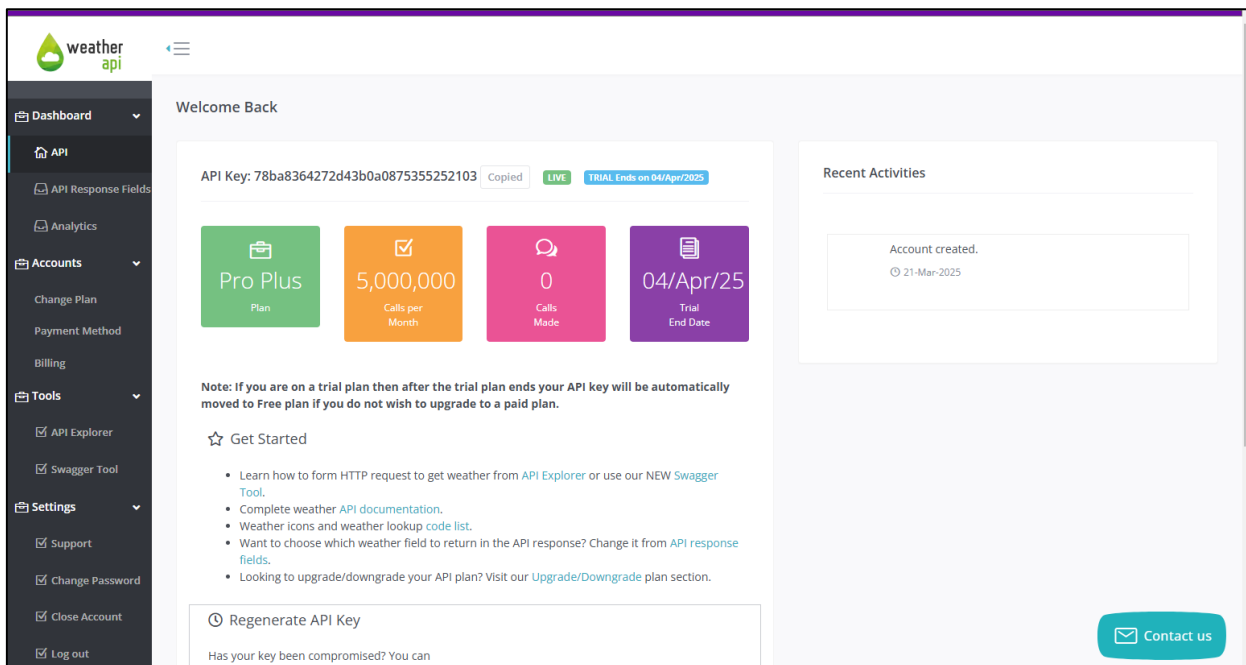


Рис. 1 – Weather api

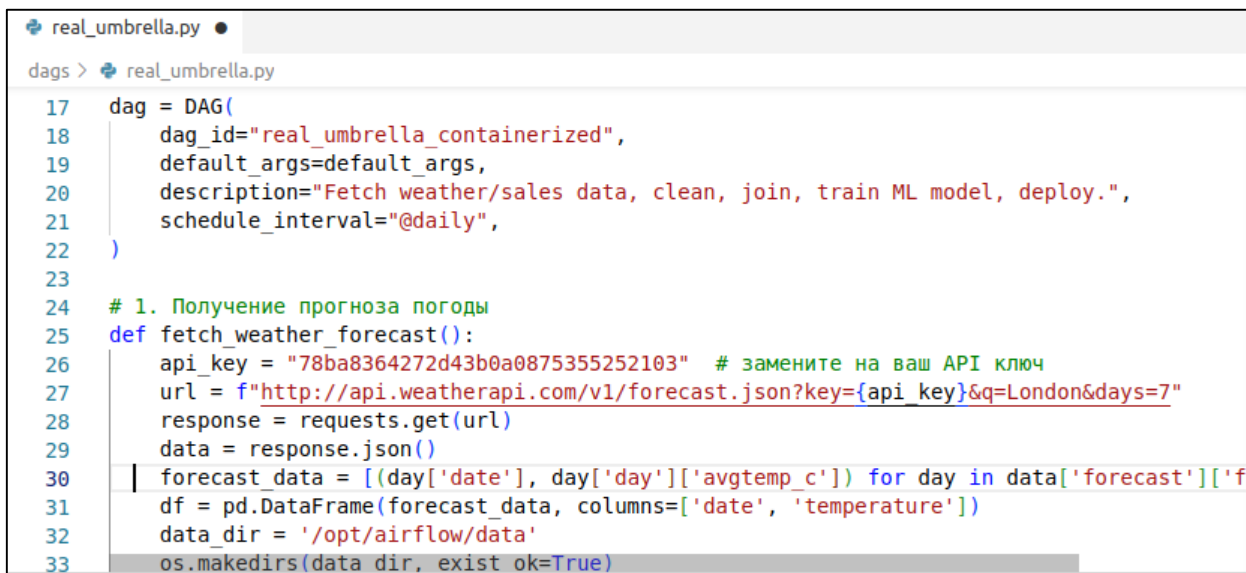


Рис. 2 – Добавление api ключа

Далее проверим, что никакие контейнеры не запущены и соберем образ (Рис. 3). После успешной сборки запустим контейнеры (Рис. 4).

```

● mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_umbrella_25$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
● mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_umbrella_25$ sudo docker build -t custom-airflow:2.0.0-py3.8 .
[+] Building 71.8s (8/8) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile                0.8s
=> => transferring dockerfile: 491B                                0.1s
=> [internal] load metadata for docker.io/apache/airflow:2.0.0-python3.8 0.0s
=> [internal] load .dockerignore                                  0.4s
=> => transferring context: 2B                                       0.0s
=> [1/4] FROM docker.io/apache/airflow:2.0.0-python3.8          4.6s
=> [2/4] RUN pip install --no-cache-dir pandas scikit-learn joblib requests 50.5s
=> [3/4] RUN pip install azure-storage-blob==12.8.1              8.5s
=> [4/4] RUN mkdir -p /opt/airflow/data /opt/airflow/logs && chown -R airflow: /opt/airflow/dat 0.9s
=> exporting to image                                             5.1s
=> => exporting layers                                              5.0s
=> => writing image sha256:a7cfbd4aacc72f5836433ec8fa25f86286c1db588a94533e25f5676a2af468f1 0.1s
=> => naming to docker.io/library/custom-airflow:2.0.0-python3.8 0.0s

```

Рис. 3 – Сборка образа

```

no such service: --build
● mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_umbrella_25$ sudo docker compose up --build
[+] Running 7/7
✓ Network business_case_umbrella_25_default                    C...           0.3s
✓ Volume "business_case_umbrella_25_logs"                      Cre...         0.0s
✓ Volume "business_case_umbrella_25_postgres_data"             Created        0.1s
✓ Container business_case_umbrella_25-postgres-1              Created        2.4s
✓ Container business_case_umbrella_25-init-1                   Created        0.7s
✓ Container business_case_umbrella_25-webserver-1              Created        0.9s
✓ Container business_case_umbrella_25-scheduler-1              Created        1.0s
Attaching to init-1, postgres-1, scheduler-1, webserver-1
postgres-1 | The files belonging to this database system will be owned by user "postgres".
postgres-1 | This user must also own the server process.
postgres-1 |
postgres-1 | The database cluster will be initialized with locale "en_US.utf8".
postgres-1 | The default database encoding has accordingly been set to "UTF8".
postgres-1 | The default text search configuration will be set to "english".
postgres-1 |
postgres-1 | Data page checksums are disabled.
postgres-1 |
postgres-1 | fixing permissions on existing directory /var/lib/postgresql/data ... ok
postgres-1 | creating subdirectories ... ok

```

Рис. 4 – Запуск контейнеров

Далее зайдем по <http://localhost:8080/> и откроется Airflow, что говорит об успешном запуске контейнера (Рис. 5).

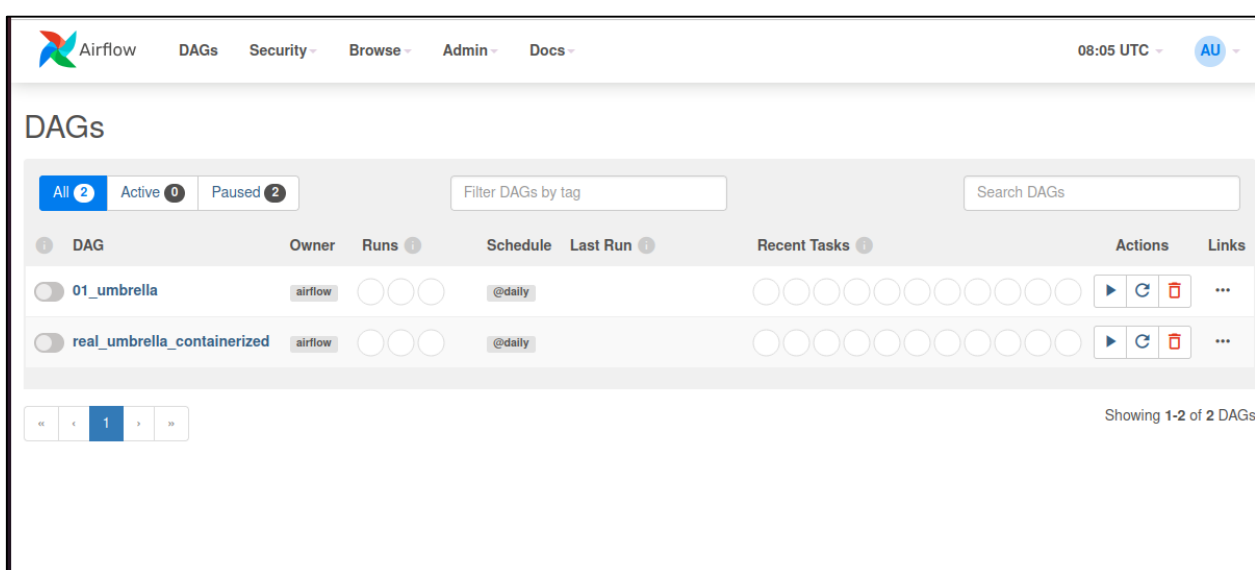


Рис. 5 – Airflow

На странице DAGs представлены имеющиеся даги (см. Рис. 5).

Для того чтобы перейти в dag необходимо нажать на синее кликабельное название dag, после чего откроется страница, в которой содержится перечень задач данного dag. Для его запуска необходимо воспользоваться кнопкой «запуск» (Рис. 6).

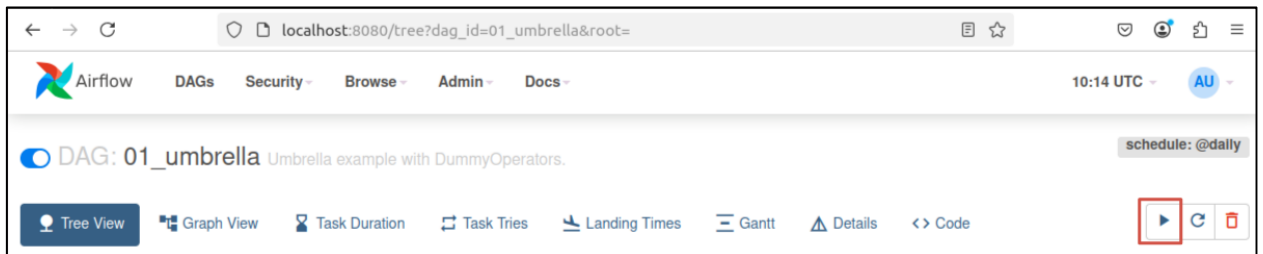


Рис. 6 – Запуск dag

Представление задач dag доступно как в виде дерева на вкладке «Tree View» (Рис. 7).

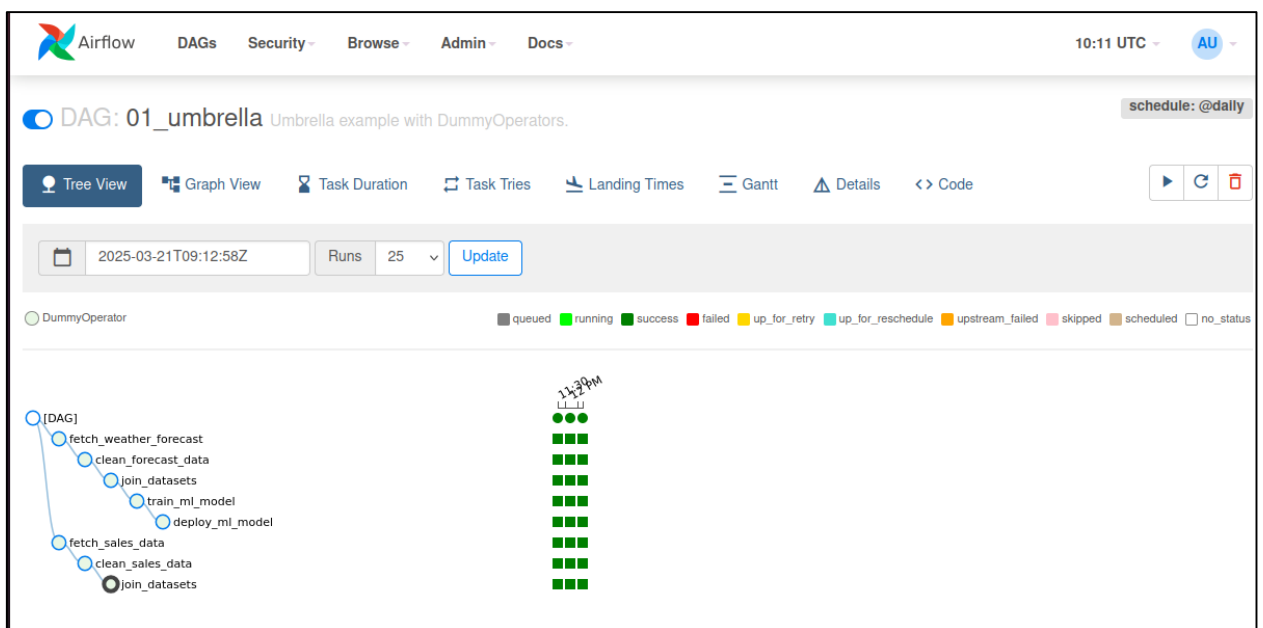


Рис. 7 – Представление в виде дерева

Также можно посмотреть представление в виде графа посредством перехода на вкладку «Graph View» (Рис. 8).

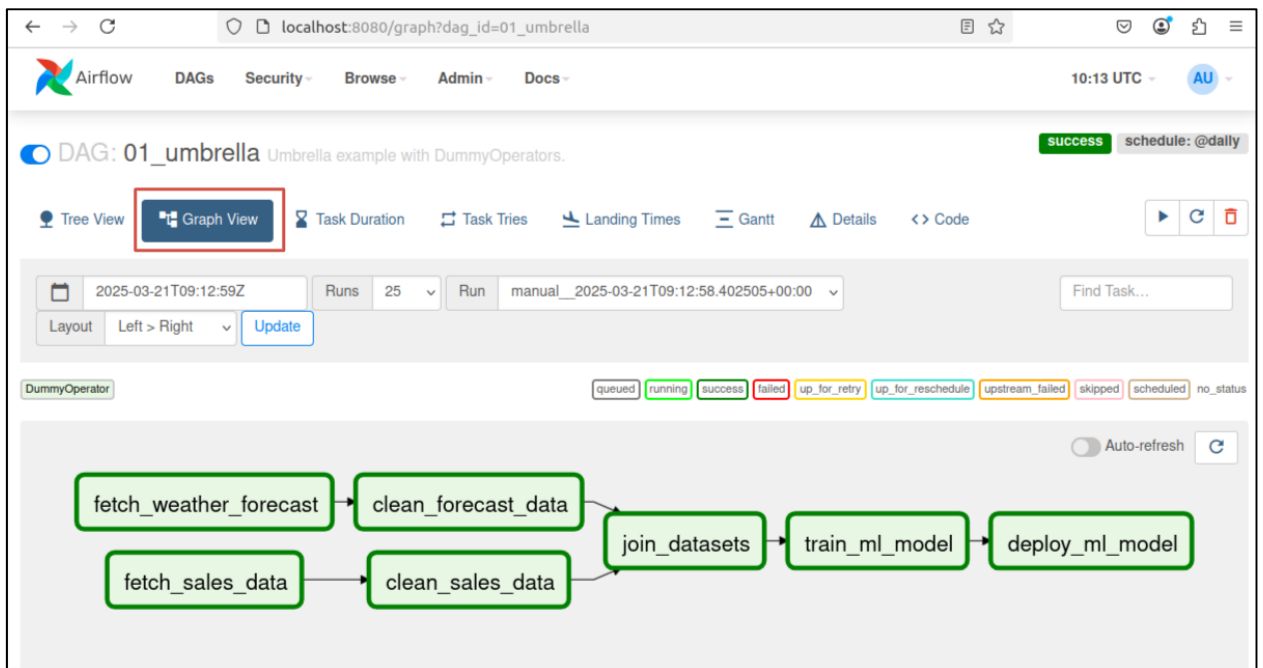


Рис. 8 – Представление в виде графа

На данной страницы также представлены статусы выполнения задач. При наведении на конкретный элемент dag отразиться информация о времени запуска, окончания, а также статус выполнения (Рис. 9).

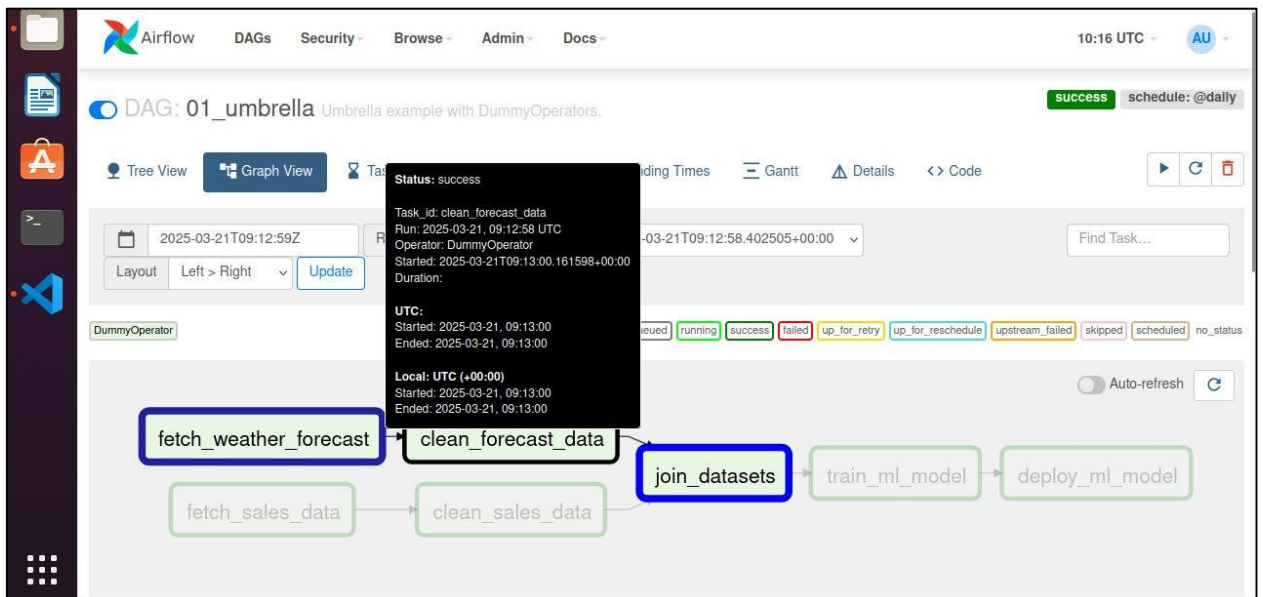


Рис. 9 – Детализированные сведения

Итак, после успешного запуска dag (Рис. 10) в папке data появятся файлы (Рис. 11).

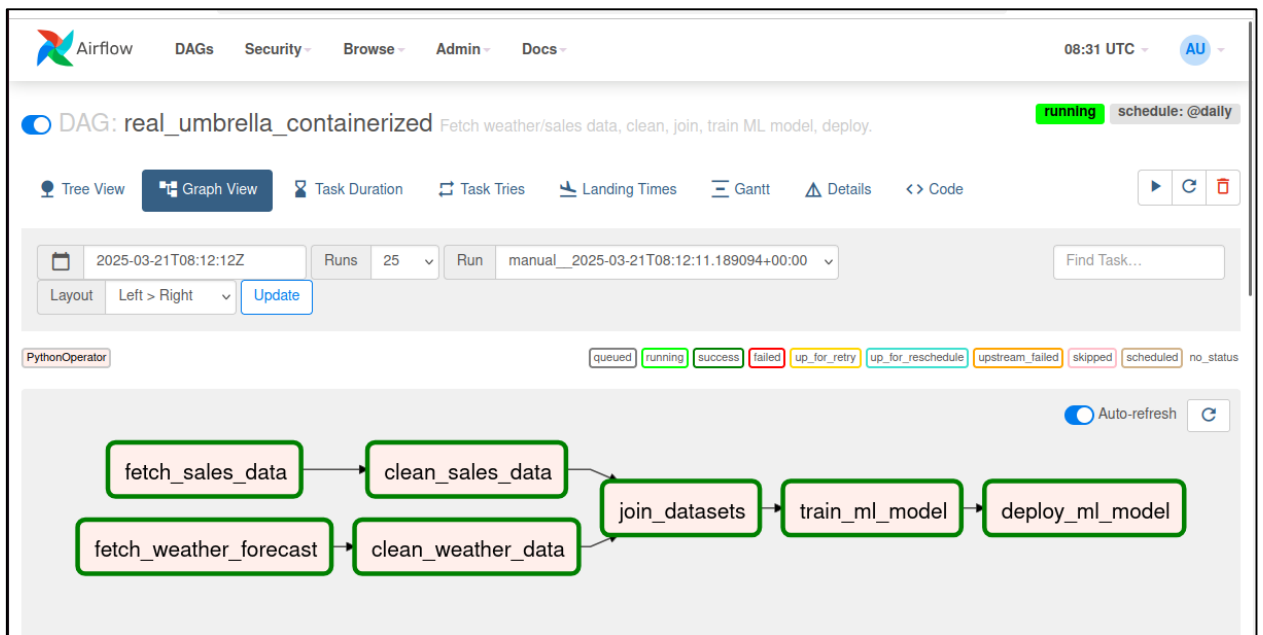


Рис. 10 – Успешный запуск real_umbrella_containerized

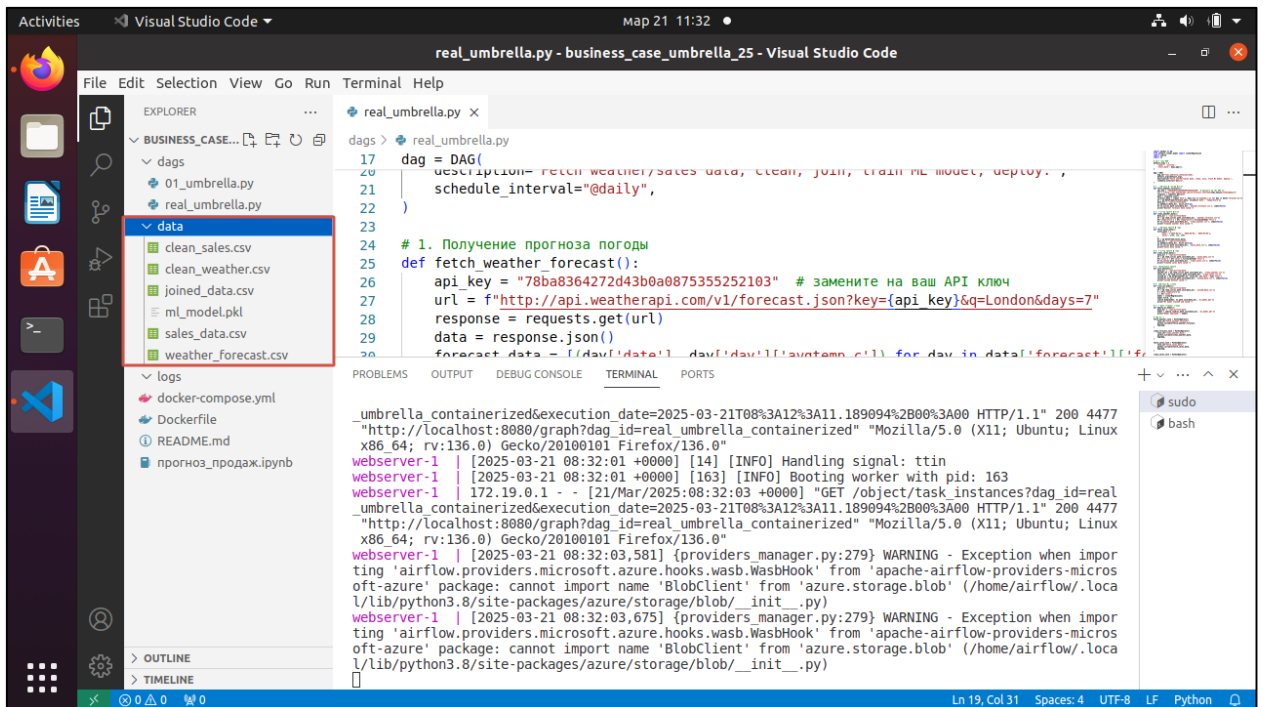


Рис. 11 – Выгруженные файлы

Задание 1.4. Спроектировать верхнеуровневую архитектуру аналитического решения задания Бизнес кейс Umbrella в draw.io.

В качестве слоя источника данных выступает weather.api. Слой хранения данных – это PostgreSQL. Получившаяся архитектура представлена на рисунке 12.

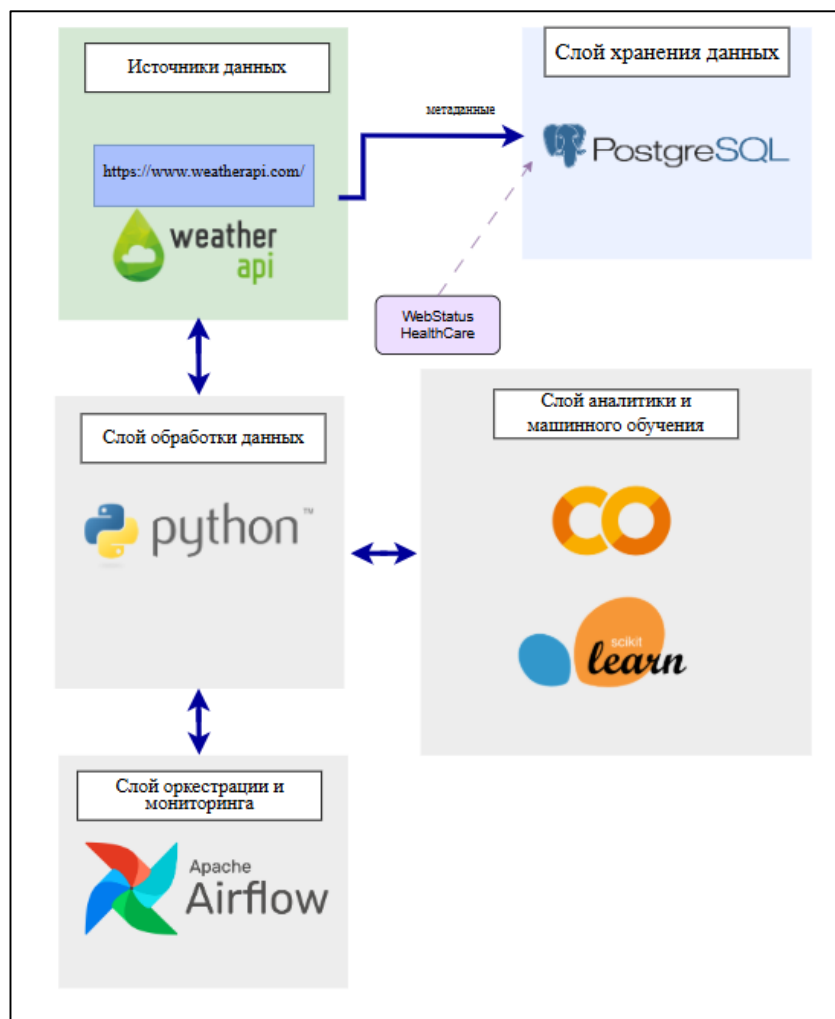


Рис. 12 – Архитектура

Индивидуальное задание. Вариант 6.

Получить прогноз в Риме на 5 дней	Заменить пропуски средним значением	Построить bar chart
-----------------------------------	-------------------------------------	---------------------

Для того, чтобы получить прогноз погоды в Риме на 5 дней надо прописать данные параметры в запросе следующим образом (Рис. 13).

```
# 1. Получение прогноза погоды
def fetch_weather_forecast():
    api_key = "78ba8364272d43b0a0875355252103" # замените на ваш API ключ
    url = f"http://api.weatherapi.com/v1/forecast.json?key={api_key}&q=Rome&days=5"
```

Рис. 13 – Запрос на получение данных о погоде в Риме

Далее сохраним изменения и перезапустим контейнер, а после и сам dag. Dag был успешно выполнен (Рис. 14) и были загружены данные, которые содержат сведения о прогнозе на 5 дней (Рис. 15).

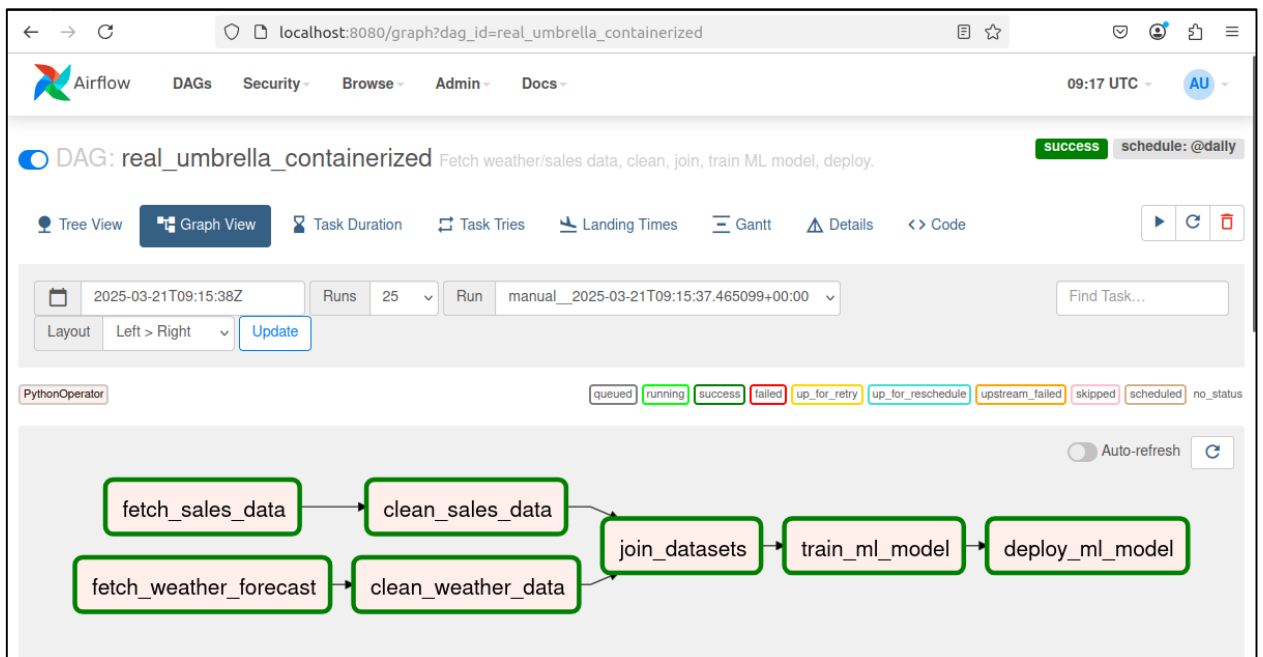


Рис. 14 – Выполнение real_umbrella_containerize

data >	weather_forecast.csv
1	date,temperature
2	2025-03-21,11.5
3	2025-03-22,12.4
4	2025-03-23,13.5
5	2025-03-24,12.7
6	2025-03-25,11.4
7	

Рис. 15 – Полученный датасет

Для того, чтобы пропуски заменить на среднее значение температуры необходимо в файл umbrella.py добавить условие:

```
mean_temp = df['temperature'].mean()
df['temperature'] = df['temperature'].fillna(mean_temp)
```

```
5
6
7 # 2. Очистка данных погоды
8 def clean_weather_data():
9     data_dir = '/opt/airflow/data'
10    df = pd.read_csv(os.path.join(data_dir, 'weather_forecast.csv'))
11    mean_temp = df['temperature'].mean()
12    df['temperature'] = df['temperature'].fillna(mean_temp)
13
14    df.to_csv(os.path.join(data_dir, 'clean_weather.csv'), index=False)
15    print("Cleaned weather data saved.")
```

Рис. 16 – Замена пропусков

Далее необходимо построить bar chart. На оси X – день, а по оси Y – температура в этот день. График представлен на рисунке 17.

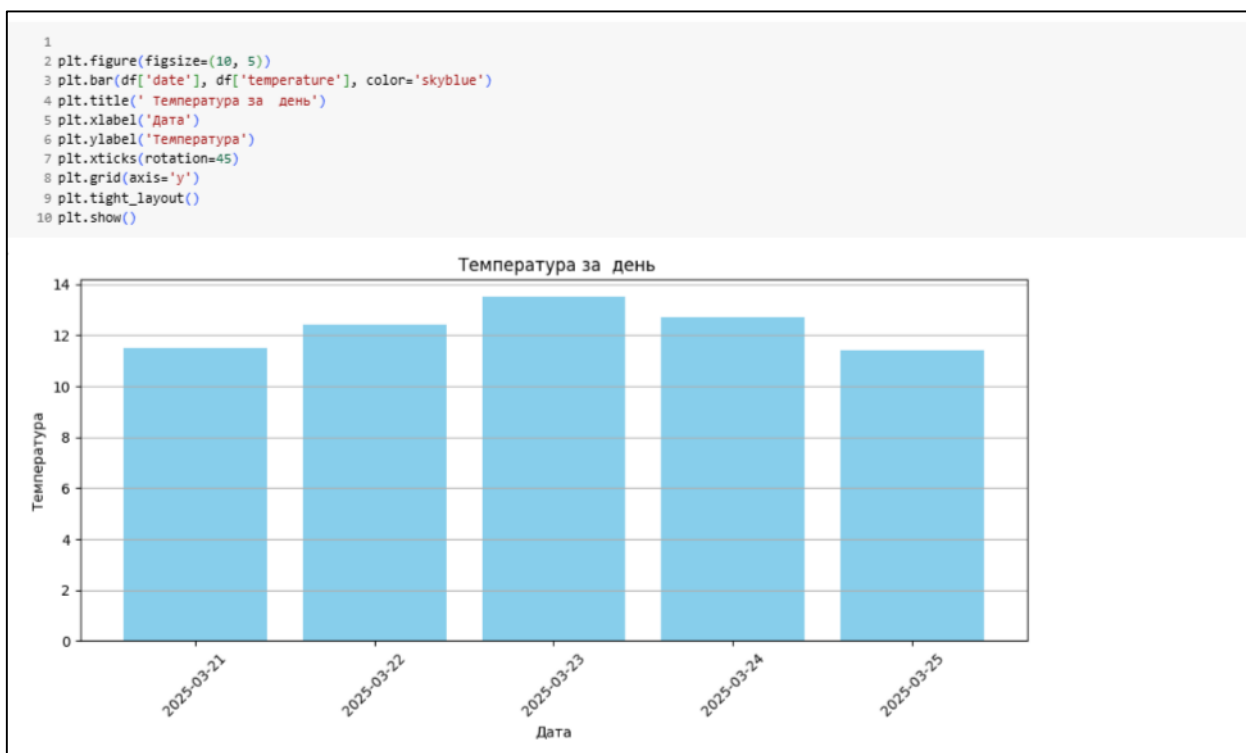


Рис. 17 – Bar chart

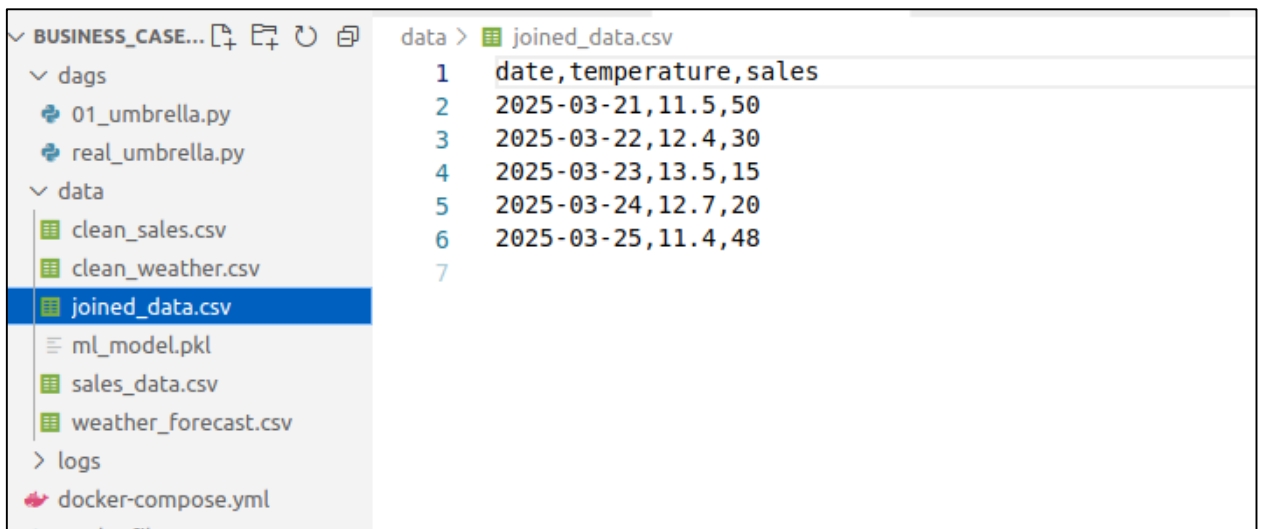
Модель машинного обучения.

Необходимо скорректировать данные о продажах по 5 дням (вместо изначальных 3 дат, которые были) и даты должны совпадать с теми, которые получаем по api (Рис. 18).

```
sales_data = {
    'date': ['2025-03-21', '2025-03-22', '2025-03-23', '2025-03-24', '2025-03-25'],
    'sales': [50, 30, 15, 20, 48]
}
```

Рис. 18 – Данные по датам

Далее видим, что в выгруженном датасете стоят корректные даты и продажи (Рис. 19).



	date	temperature	sales
1	2025-03-21	11.5	50
2	2025-03-22	12.4	30
3	2025-03-23	13.5	15
4	2025-03-24	12.7	20
5	2025-03-25	11.4	48

Рис. 19 – Полученный csv

Далее запустим модель и посмотрим количество зонтов, которое необходимо произвести при средней температуре в 10 градусов (Рис. 20).

```
[1] 1 from google.colab import files
    2 uploaded = files.upload()

Выбрать файлы Число файлов: 2
• ml_model.pkl(n/a) - 925 bytes, last modified: 21.03.2025 - 100% done
• weather_forecast.csv(text/csv) - 97 bytes, last modified: 21.03.2025 - 100% done
Saving ml_model.pkl to ml_model (1).pkl
Saving weather_forecast.csv to weather_forecast (1).csv

[2] 1 !pip install dill

Requirement already satisfied: dill in /usr/local/lib/python3.11/dist-packages (0.3.9)

[8] 1 import joblib
    2 model = joblib.load("ml_model.pkl")
    3 import pandas as pd
    4 print(model.predict(pd.DataFrame({'temperature': [10]}))) # Например, прогноз продаж при 10°C

[73.33856209]
```

Рис. 20 – Прогнозирование производства при температуре 10 градусов

При уменьшении температуры до 0 градусов, количество зонтов увеличивается (Рис. 21).

```
[13] 1 import joblib
     2 model = joblib.load("ml_model.pkl")
     3 import pandas as pd
     4 print(model.predict(pd.DataFrame({'temperature': [0]}))) # Например, прогноз продаж при 0°C

[250.4627451]
```

Рис. 21 – Прогнозирование производства при температуре 0 градусов

Заключение: Таким образом, в рамках практической работы на занятии были закреплены знания по запуску и настройке Airflow, также был получен

опыт получения данных по api через запуск dag. В ходе выполнения удалось выгрузить полученные данные и обучить модель ML.