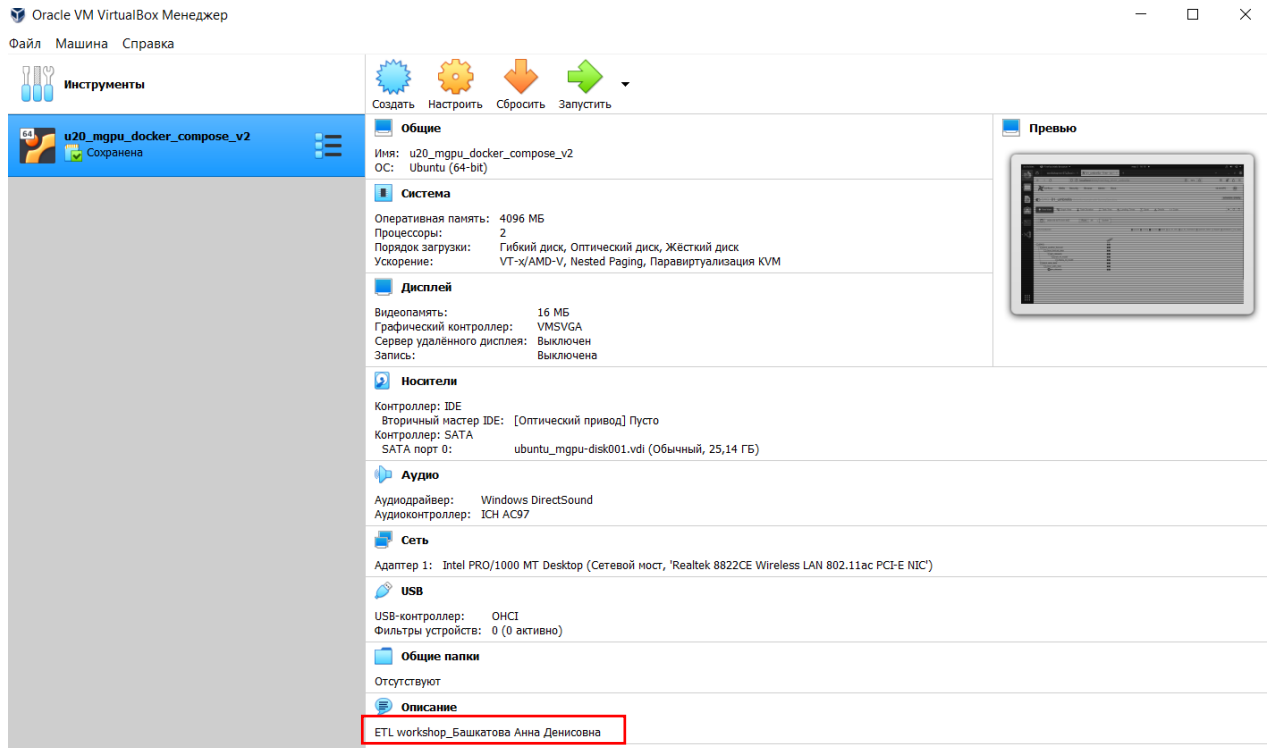


Практическая работа 4. Часть 1

Задание 4.1. Бизнес кейс «Umbrella»

Задание 4.1.1. Развернуть VM ubuntu_mgpu.ova в VirtualBox.



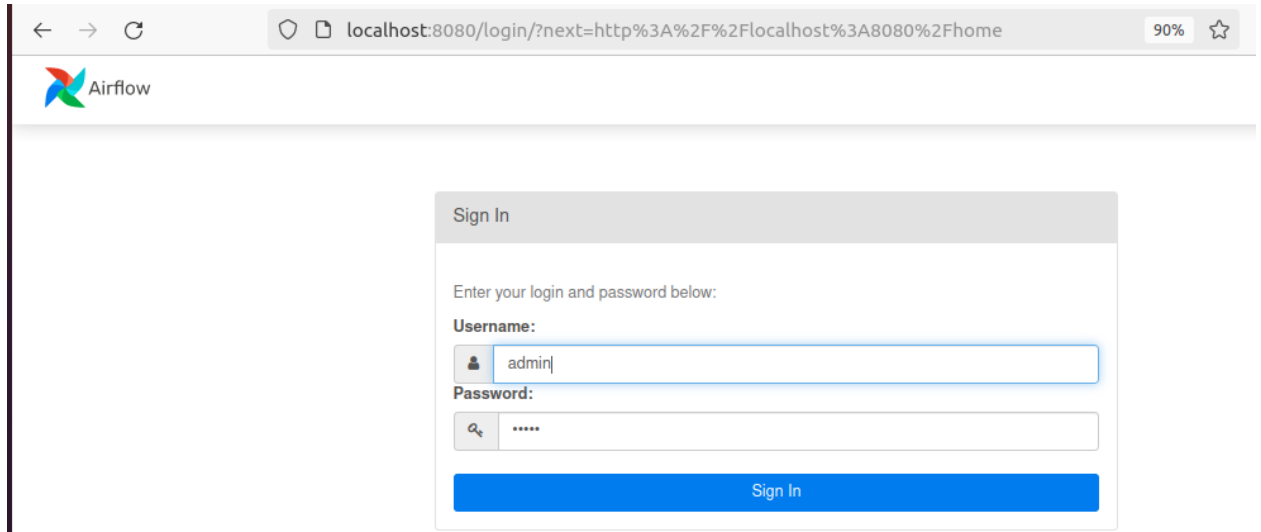
Задание 4.1.2. Клонировать на ПК задание Бизнес кейс Umbrella в домашний каталог VM.

```
mgpu@mgpu-VirtualBox:~$ git clone https://github.com/BosenkoTM/workshop-on-ETL.git
Cloning into 'workshop-on-ETL'...
remote: Enumerating objects: 54, done.
remote: Counting objects: 100% (54/54), done.
remote: Compressing objects: 100% (51/51), done.
remote: Total 54 (delta 11), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (54/54), 15.46 KiB | 1.19 MiB/s, done.
mgpu@mgpu-VirtualBox:~$ ls
Desktop  Downloads  Pictures  snap      thinclient_drives  workshop-on-ETL
Documents Music      Public   Templates Videos
mgpu@mgpu-VirtualBox:~$ cd workshop-on-ETL/
mgpu@mgpu-VirtualBox:~/workshop-on-ETL$ ls
business_case_umbrella  README.md
mgpu@mgpu-VirtualBox:~/workshop-on-ETL$ bashkatova anna
```

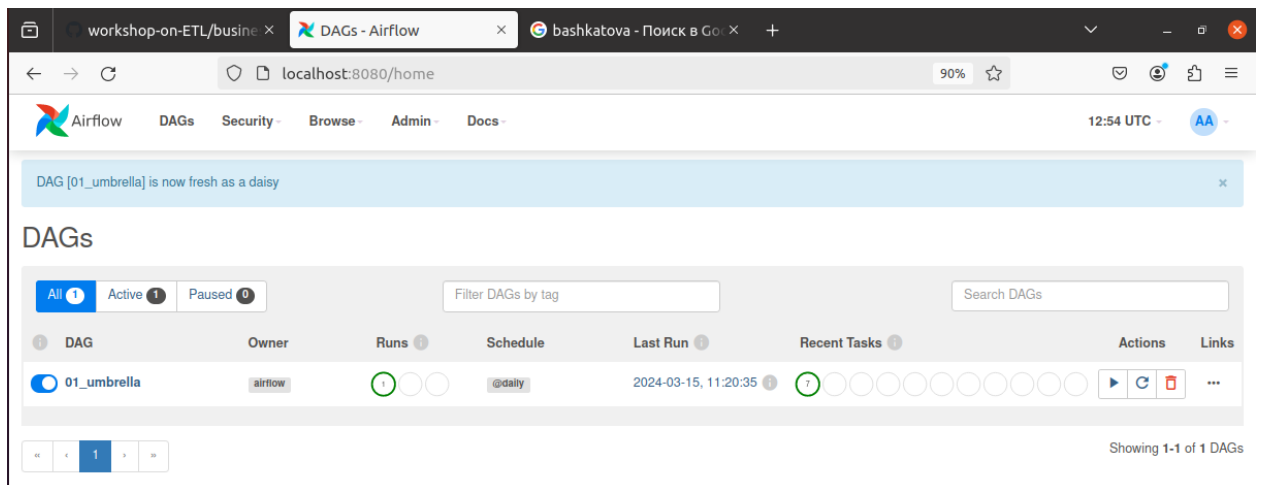
Задание 4.1.3. Запустить контейнер с кейсом, изучить и описать основные элементы интерфейса Apache Airflow.

```
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_umbrella$ sudo docker compose up -d
[+] Running 4/5
  ⌘ Network business_case_umbrella default      Created
  ✓ Container business_case_umbrella-postgres-1 Started
  ✓ Container business_case_umbrella-init-1     Started
  ✓ Container business_case_umbrella-scheduler-1 Started
  ✓ Container business_case_umbrella-webserver-1 Started
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_umbrella$ bashkatova anna
```

Переходим по линку `http://localhost:8080/` и входим в систему.



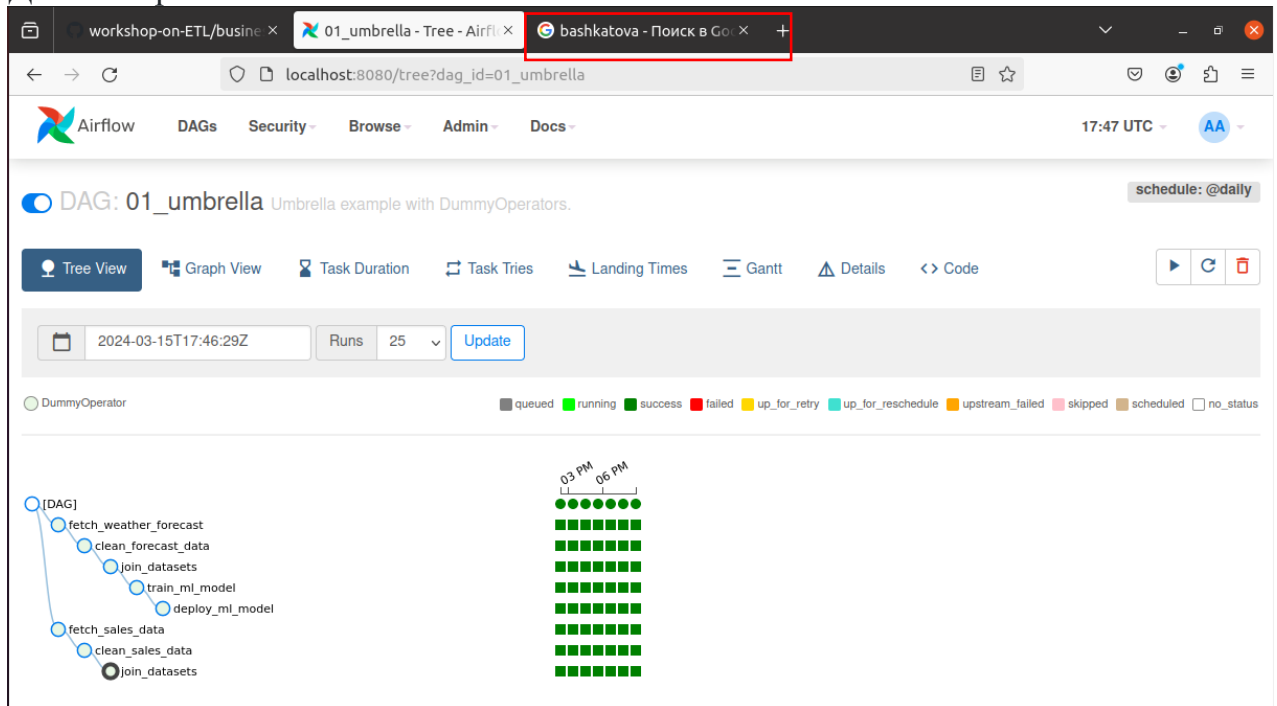
После входа представлен DAG Umbrella.



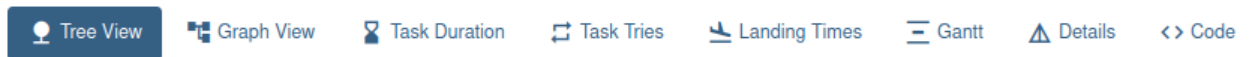
Веб-интерфейс:

- **Ползунок Pause/Unpause DAG** — переключатель включение/выключение DAG. По умолчанию все новые DAG — остановлены, для запуска DAG необходимо предварительно включить.
- **Owner** — владелец/автор DAG.
- **Runs** — состояние запусков прошлых DAG. 3 состояния:
 - Success: успешно выполнен
 - Running: выполняется
 - Failed: есть ошибки при выполнении
- **Schedule** — периодичность запуска DAG.
- **Last Run** — дата и время последнего запуска DAG.
- **Recent Tasks** — текущее состояние последних запусков DAG
- **Actions** — запуск DAG вручную, обновление или удаление DAG.
- **Links** — список быстрого доступа к просмотру кода DAG, деталей выполнения, просмотру в виде графа или диаграммы Ганта и т.д.

Далее открываем DAG:

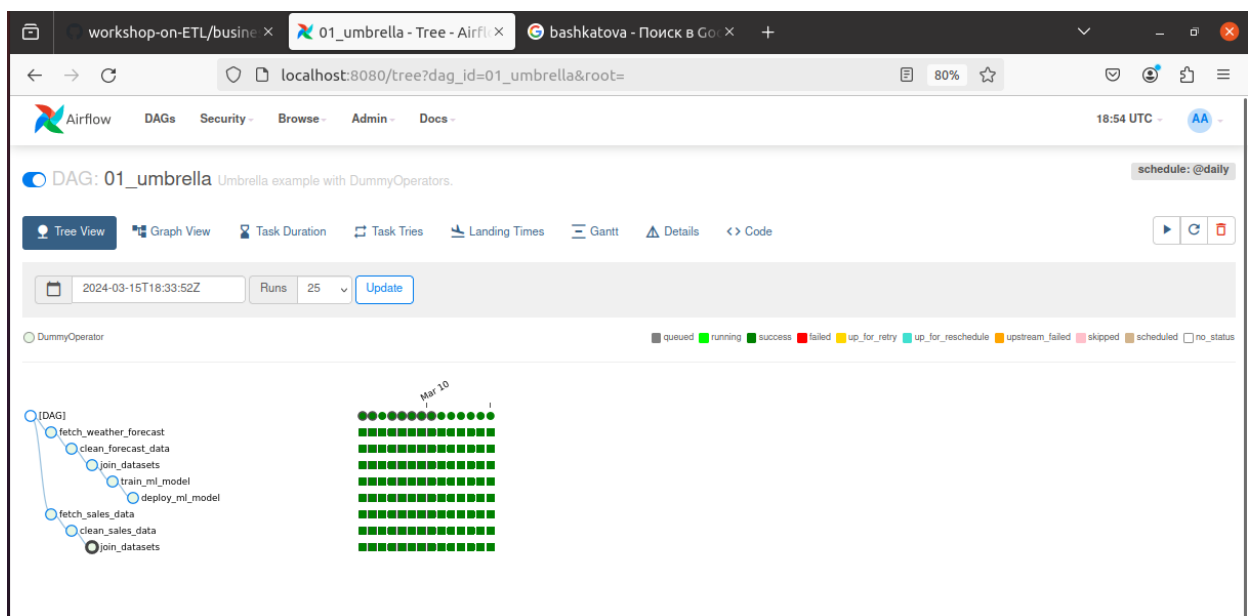


Рассмотрим и опишем основные элементы интерфейса:



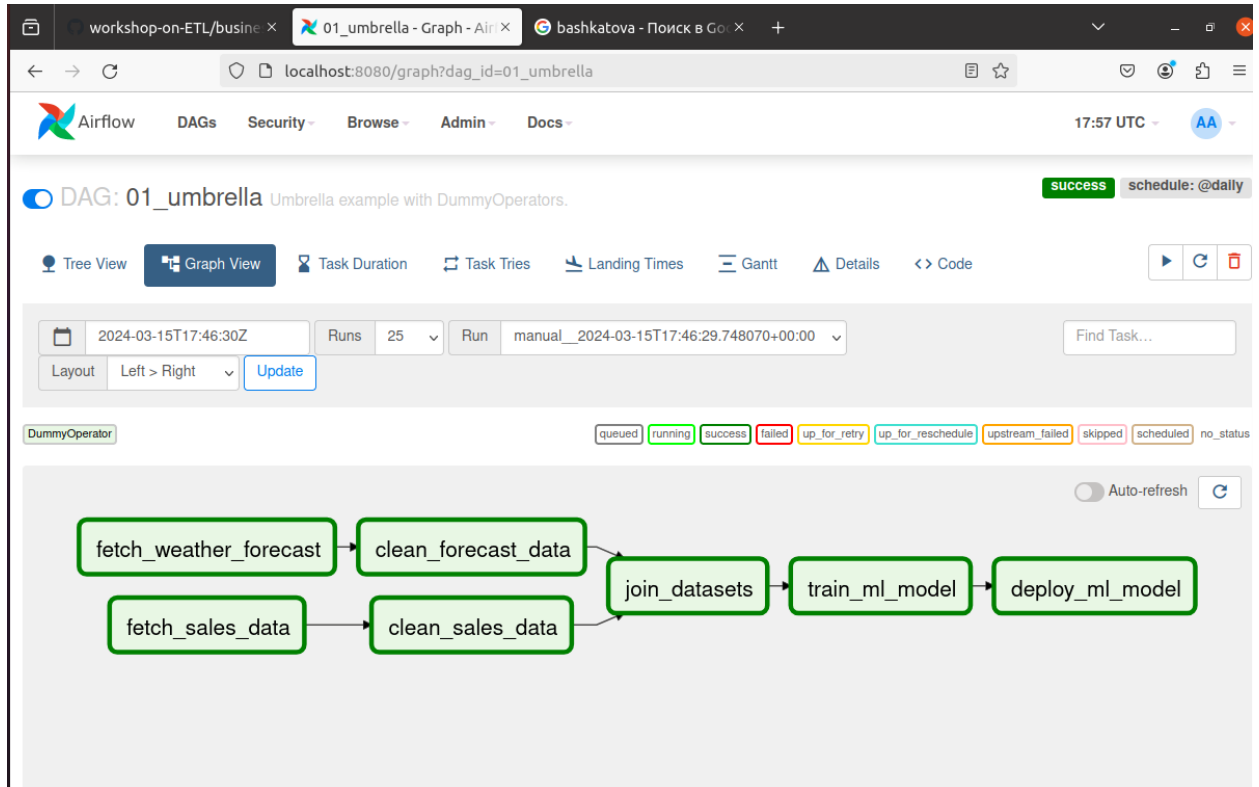
1. Вкладка Tree View

Позволяет просмотреть DAG и зависимости между операторами в виде дерева. Справа находится статус выполнения всех операторов, входящих в DAG с легендой. Круги представляют операторы. Квадраты – задачи. На статусы можно кликать, чтобы посмотреть более детальную информацию.



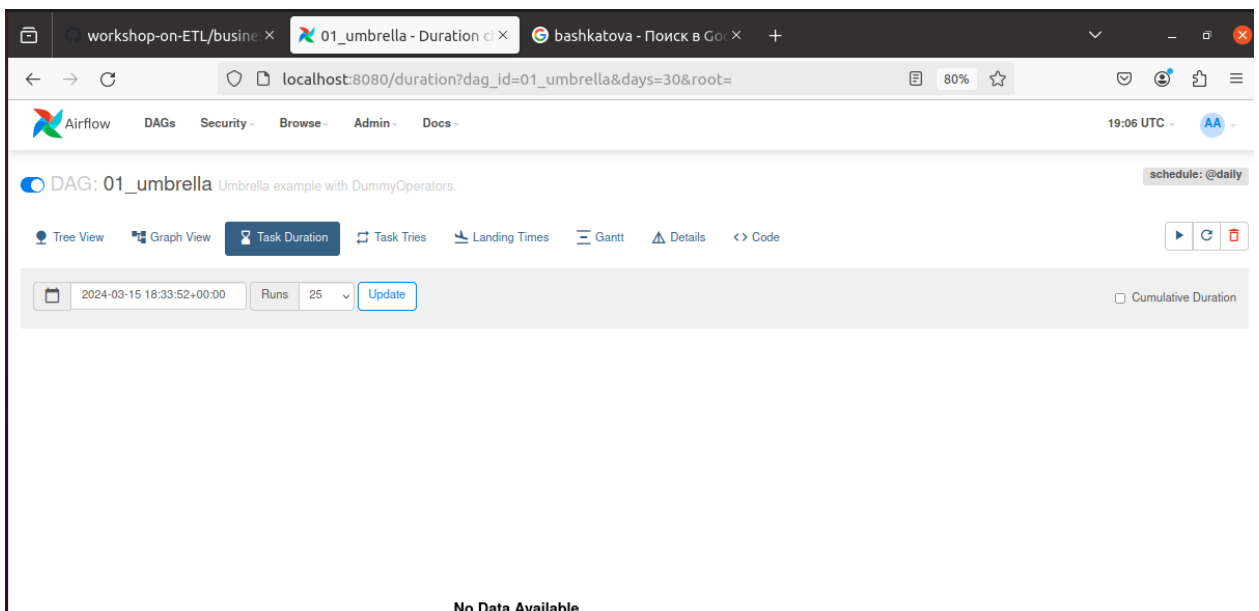
2. Вкладка Graph View

Здесь DAG представлен в виде графа, чтобы визуально просматривать зависимости между задачами в рамках DAG, а также для получения статуса задач для последнего запуска DAG. Здесь прямоугольники соответствуют задачам, а цвета границ соответствуют статусу задач.



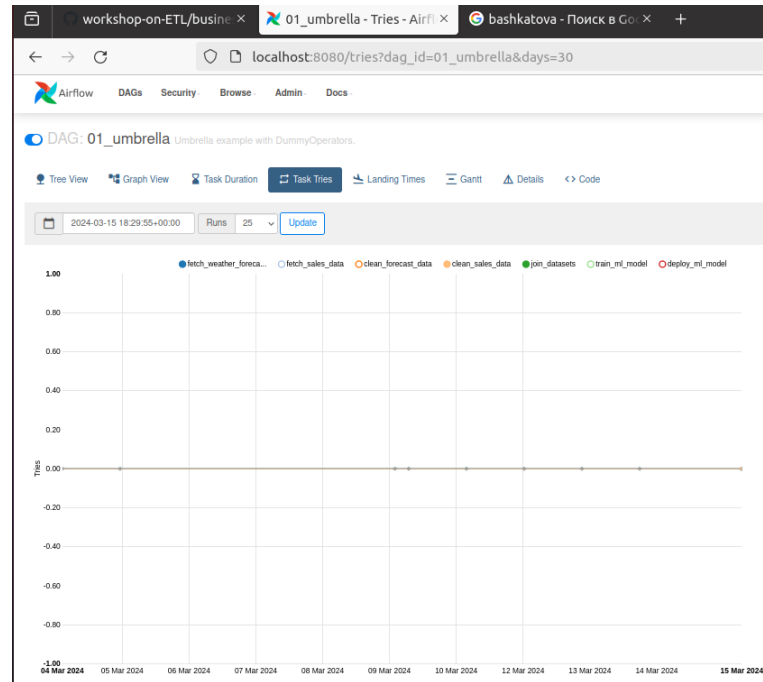
3. Вкладка Task Duration

Общее представление о продолжительности выполнения задач за последние N запусков, это позволяет находить отклонения и быстро понимать, на что тратится время в работе за много запусков.



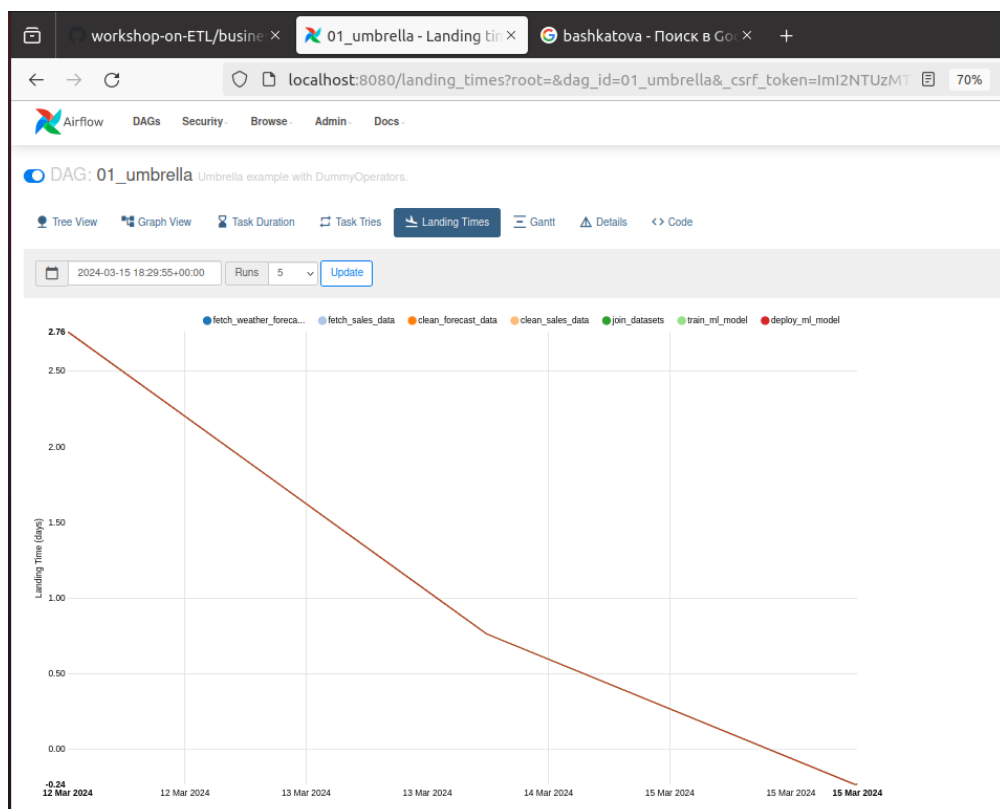
4. Вкладка Task Tries

Показывает график повторных запусков операторов по периодам.



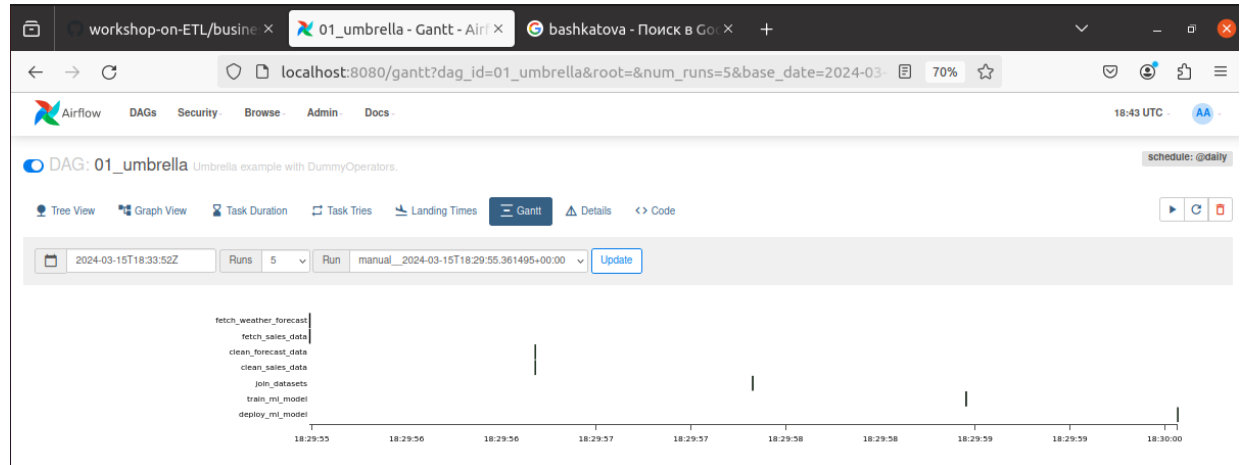
5. Вкладка Landing Times

График времени выполнения задачи в DAG показывает задержку с момента планирования задачи до ее завершения, и его можно отслеживать с течением времени, чтобы оценить эффективность изменений. Он создает картину происходящего, давая нам представление, необходимое для начала устранения неполадок.



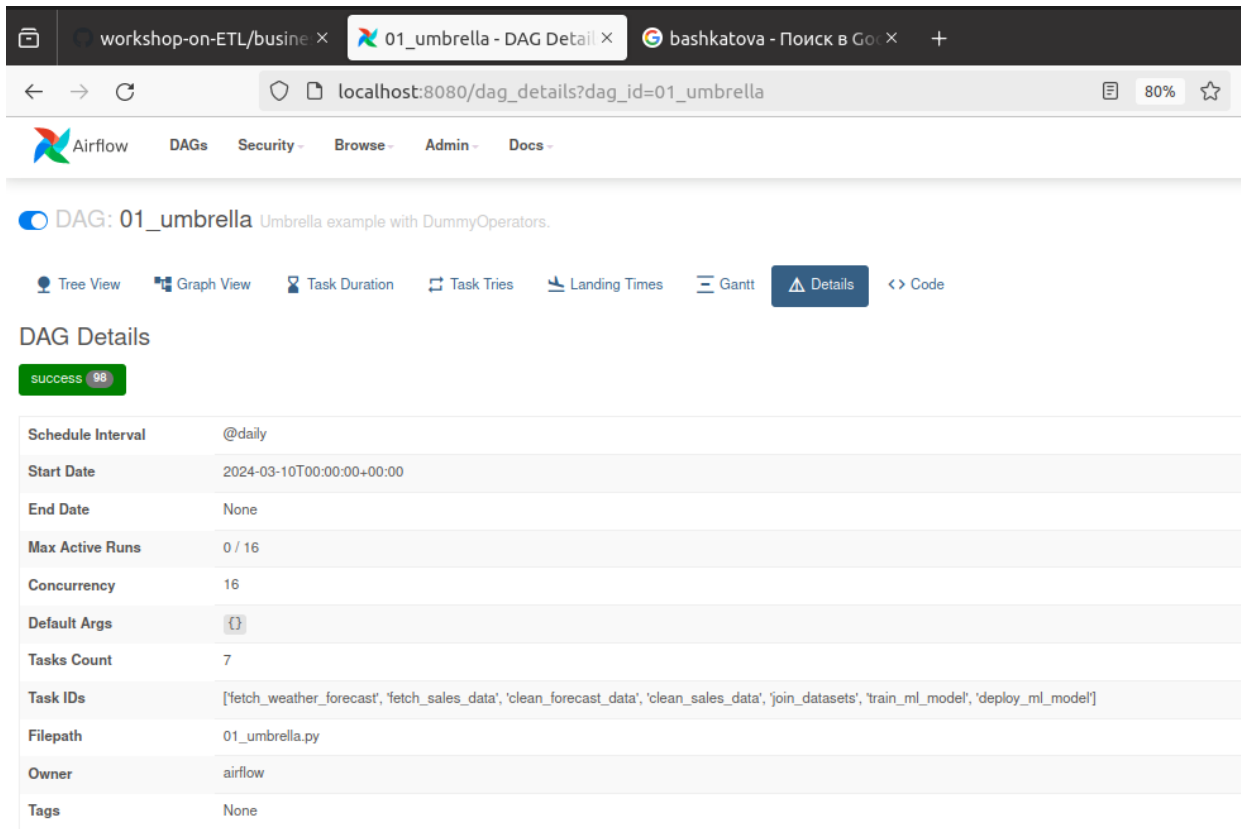
6. Вкладка Gantt

Отображает историю выполнения DAG в виде Диаграммы Ганта. Представление Ганта позволяет анализировать продолжительность задачи, а также совпадения. Взглянув на представление Ганта, можно быстро определить узкие места, а также то, где на конкретную диаграмму тратится большая часть времени. Чем больше прямоугольник в представлении Ганта, тем больше времени требуется для выполнения задачи



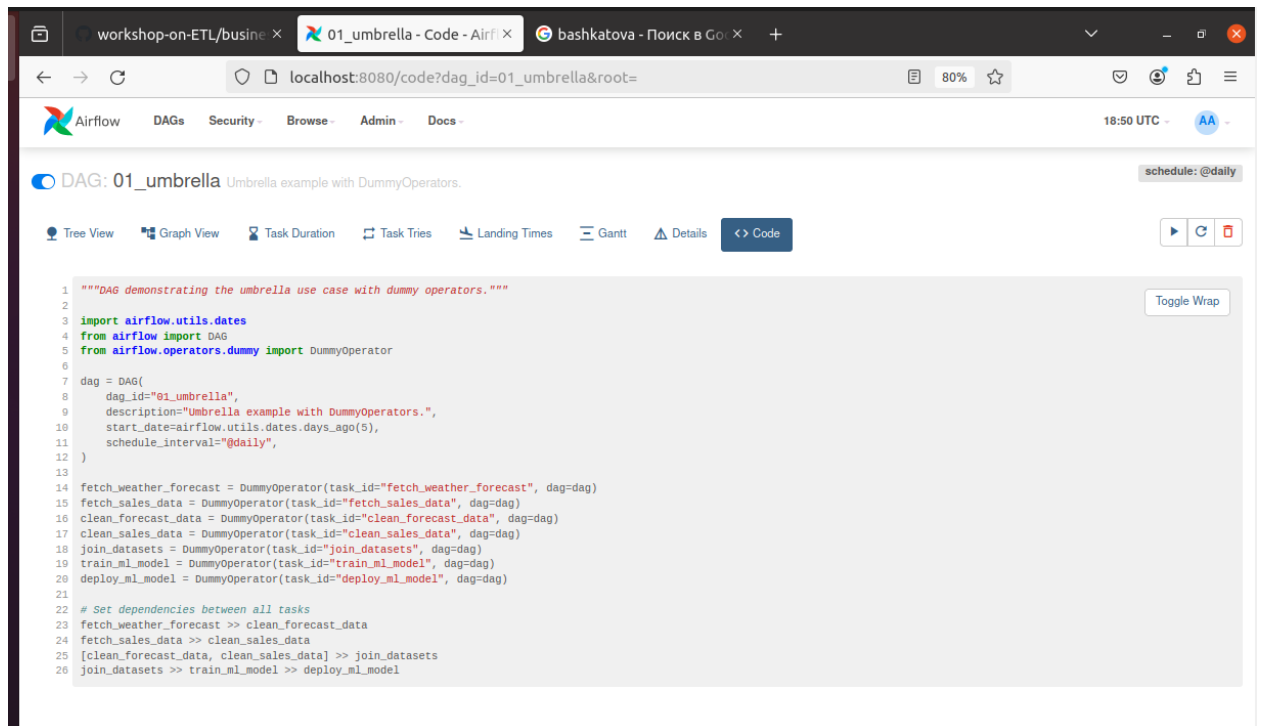
7. Вкладка Details

Детальная информация по DAG, включая интервал запуска, дата начала, дата окончания, максимальное количество активных запусков, число задач выполняемых параллельно, количество задач, идентификаторы задач, путь к файлу, владелец, тэги.



8. Вкладка Code

Дает возможность просмотреть код конвейера данных.



The screenshot shows the Apache Airflow web interface in a browser. The address bar indicates the URL is `localhost:8080/code?dag_id=01_umbrella&root=`. The interface displays the 'Code' tab for a DAG named '01_umbrella'. The code is written in Python and defines a DAG with several tasks: `fetch_weather_forecast`, `fetch_sales_data`, `clean_forecast_data`, `clean_sales_data`, `join_datasets`, `train_ml_model`, and `deploy_ml_model`. The tasks are connected in a sequence, with dependencies between them. The code is displayed in a syntax-highlighted editor with line numbers. A 'Toggle Wrap' button is visible on the right side of the code editor.

Задание 4.1.4. Верхнеуровневая архитектура аналитического решения задания Бизнес кейс Umbrella в [draw.io](#).

