

Департамент образования и науки 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 № 4 Тема:

«Проектирование сквозного конвейера ETL на Python и Airflow».

Выполнила: Нестратова А.М., АДЭУ-201

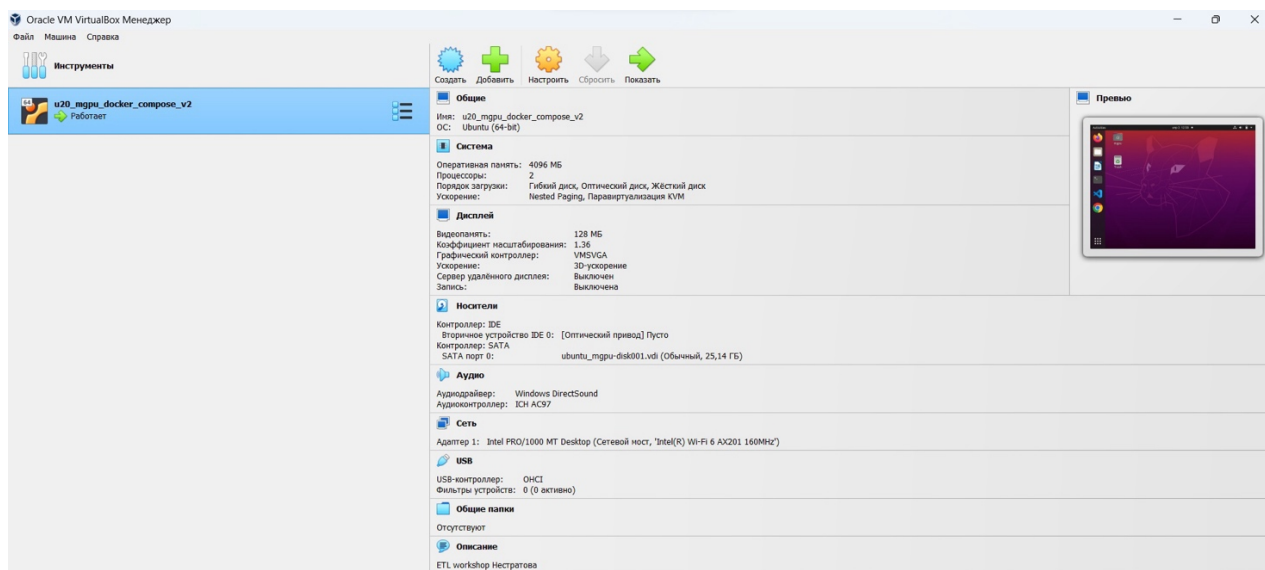
Преподаватель: Босенко Т.М.

Москва

2024

Задание 4.1. Бизнес кейс «Umbrella»

4.1.1. Развернуть VM ubuntu_mgpu.ova в VirtualBox.



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

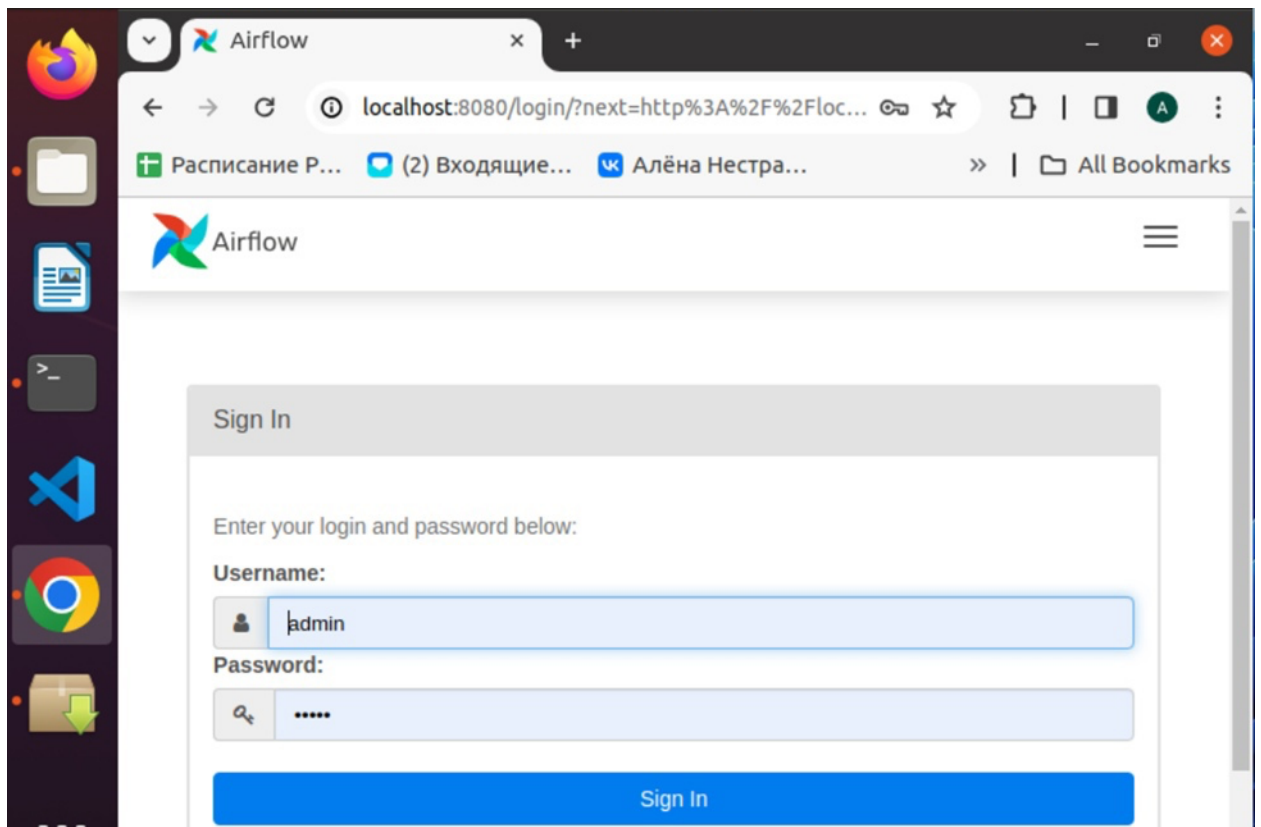
`git clone https://github.com/BosenkoTM/workshop-on-ETL.git`



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
[sudo] password for mgpu:
[+] Running 4/5
  :: Network business_case_umbrella_default      Cre...      1.5s
  ✓ Container business_case_umbrella-postgres-1 Started      0.5s
  ✓ Container business_case_umbrella-scheduler-1 Started      1.3s
  ✓ Container business_case_umbrella-webserver-1 Started      1.3s
  ✓ Container business_case_umbrella-init-1      St...      1.2s
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_umbrella$
```

Переходим по линку ``http://localhost:8080/``.



После входа представлен DAG Umbrella.

Описание интерфейса:

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. У него есть 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.

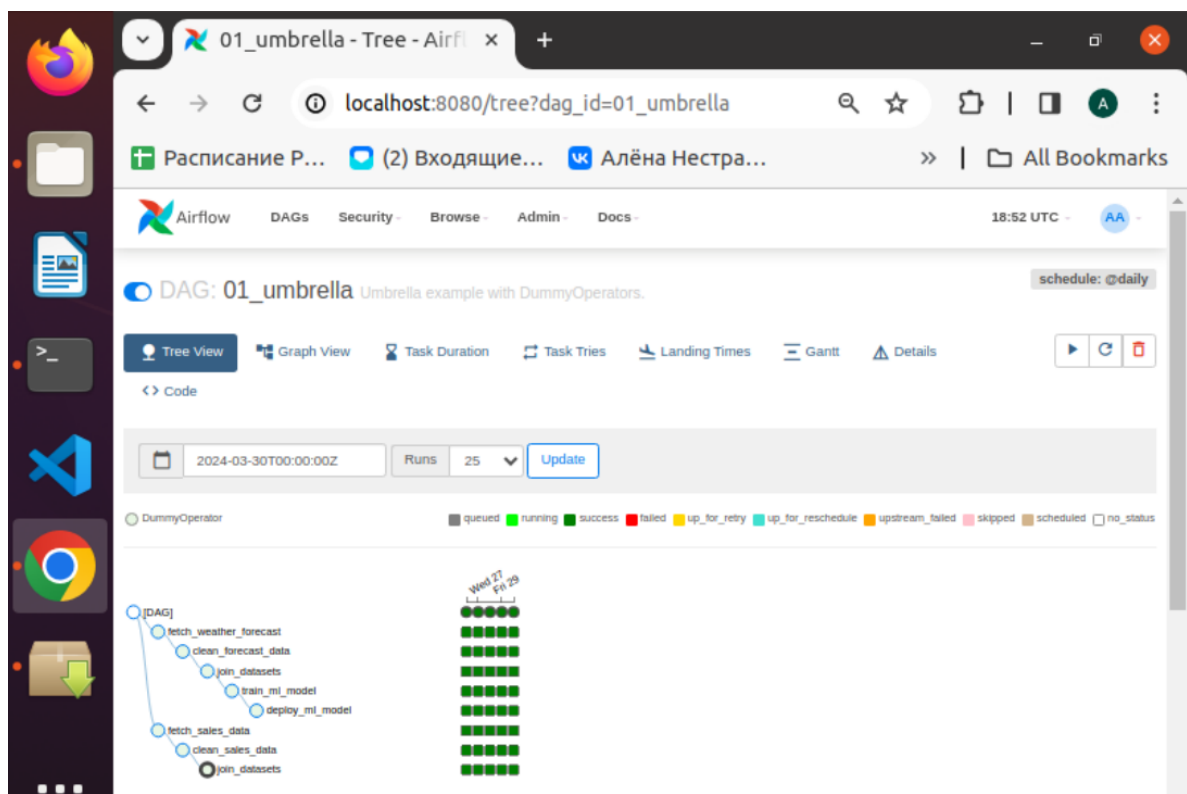
Links — список быстрого доступа к просмотру кода DAG, деталей выполнения, просмотру в виде графа или диаграммы Ганта и т.д.

DAGs (Directed Acyclic Graphs) - Графы направленного ациклического связывания:

- Список всех определенных и загруженных DAG.
- Возможность управления и контроля за запуском и остановкой DAG.
- Просмотр статуса выполнения каждой конкретной DAG.

Tree View (Представление в виде дерева):

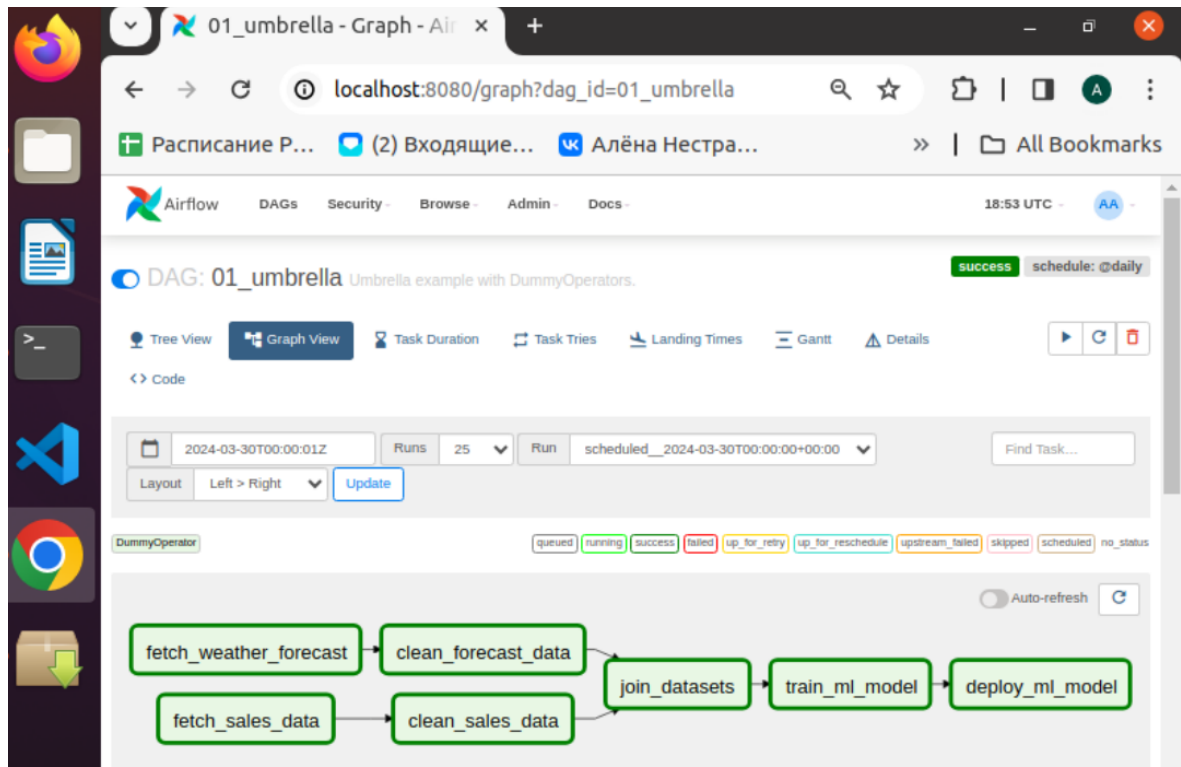
Интерактивное дерево с иерархией задач и их статусом. Позволяет легко наблюдать и управлять задачами и их зависимостями.



Graph View (Визуальное представление):

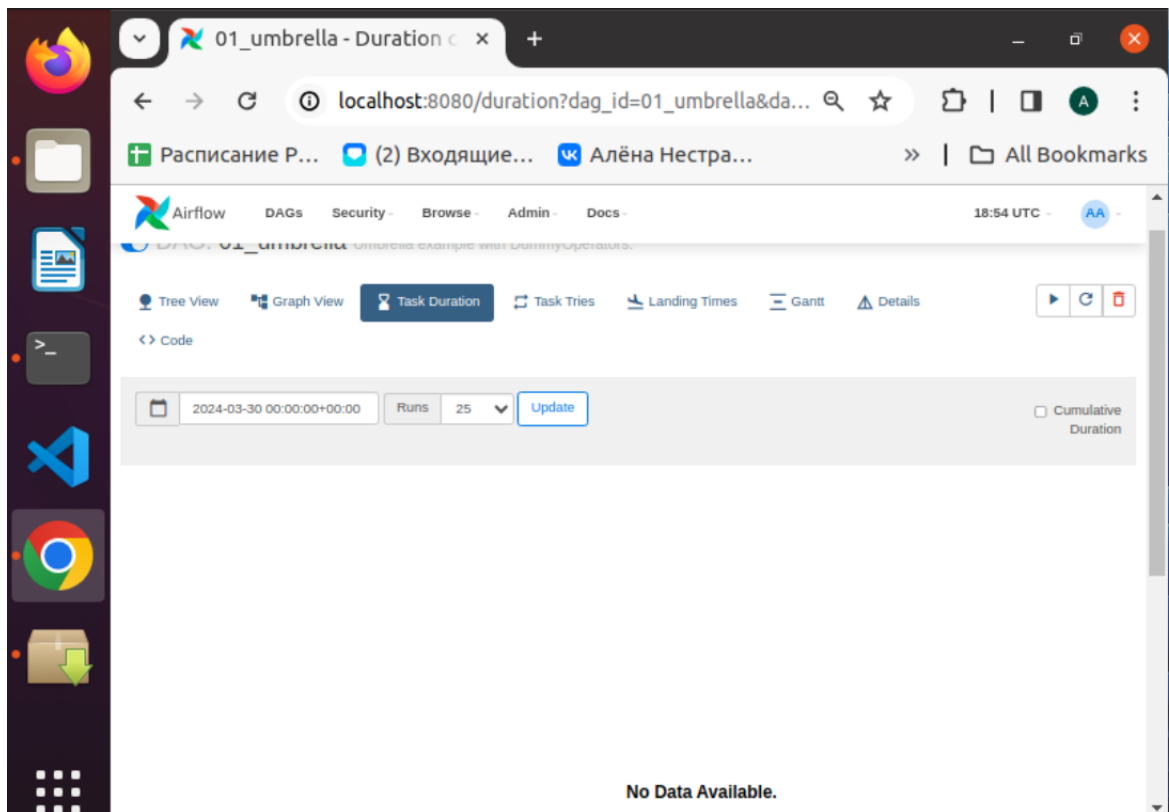
- Визуализация структуры DAG в виде графа с зависимостями между задачами.

- Позволяет легко отслеживать поток выполнения задач с учетом зависимостей.

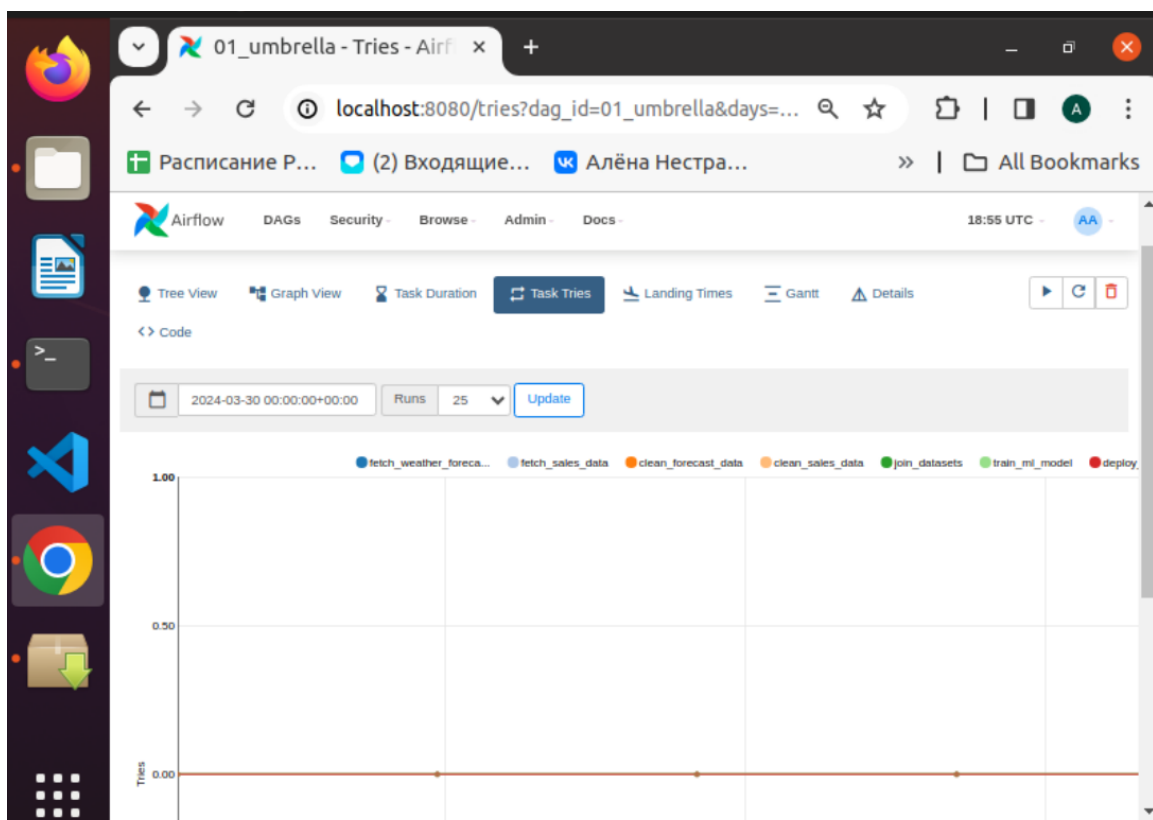


Task duration (продолжительность задачи)

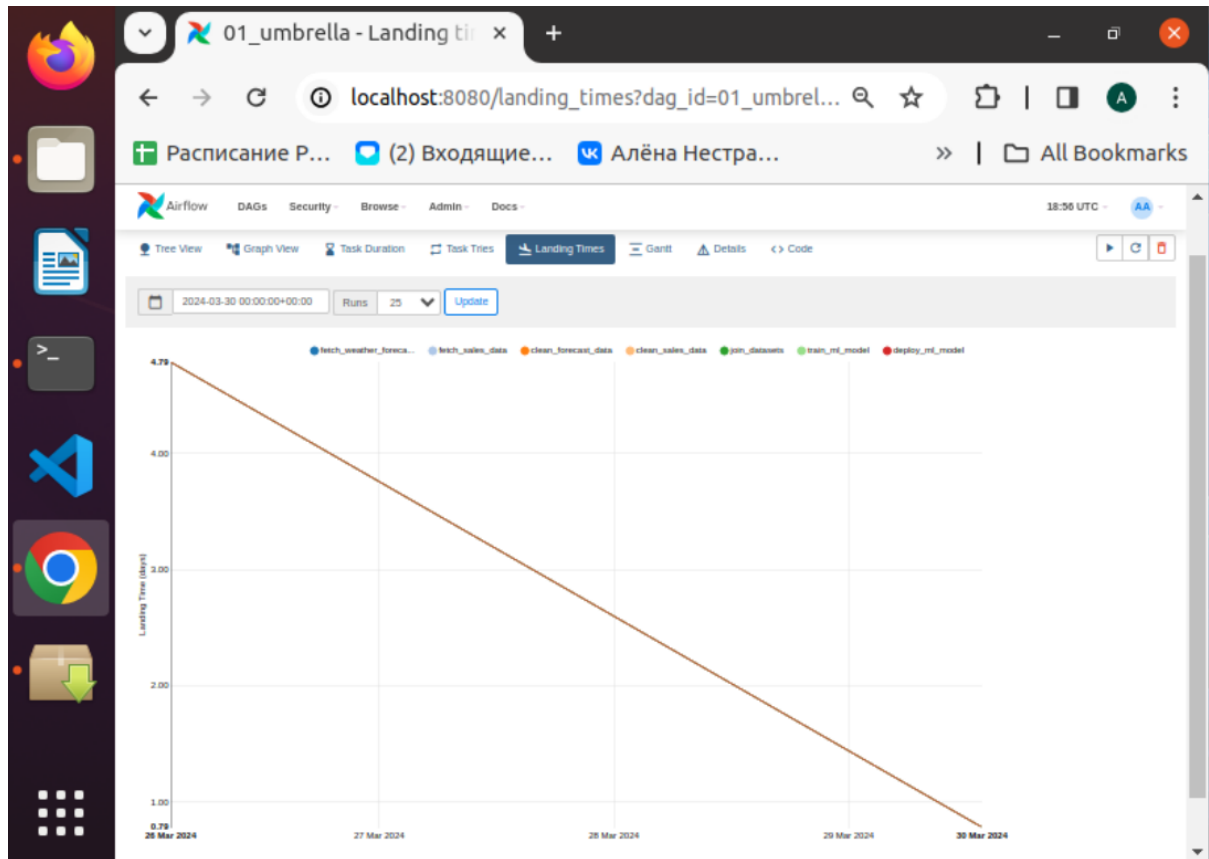
Продолжительность выполнения различных задач за последние N запусков. Это представление позволяет находить выбросы и быстро понимать, на что тратится время в вашей группе обеспечения доступности баз данных за многие прогоны.



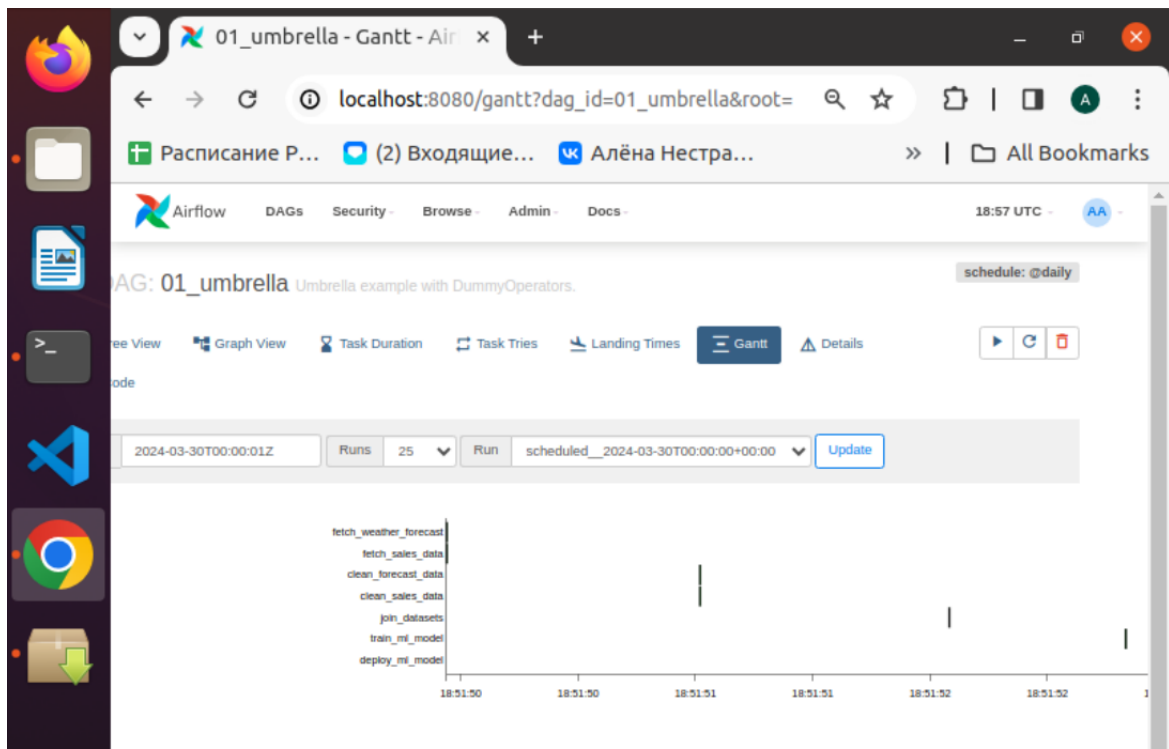
Task Tries отображает количество попыток выполнения конкретной задачи (Task) в рамках DAG (Directed Acyclic Graph). Каждая строка в этой вкладке представляет одну попытку выполнения задачи.



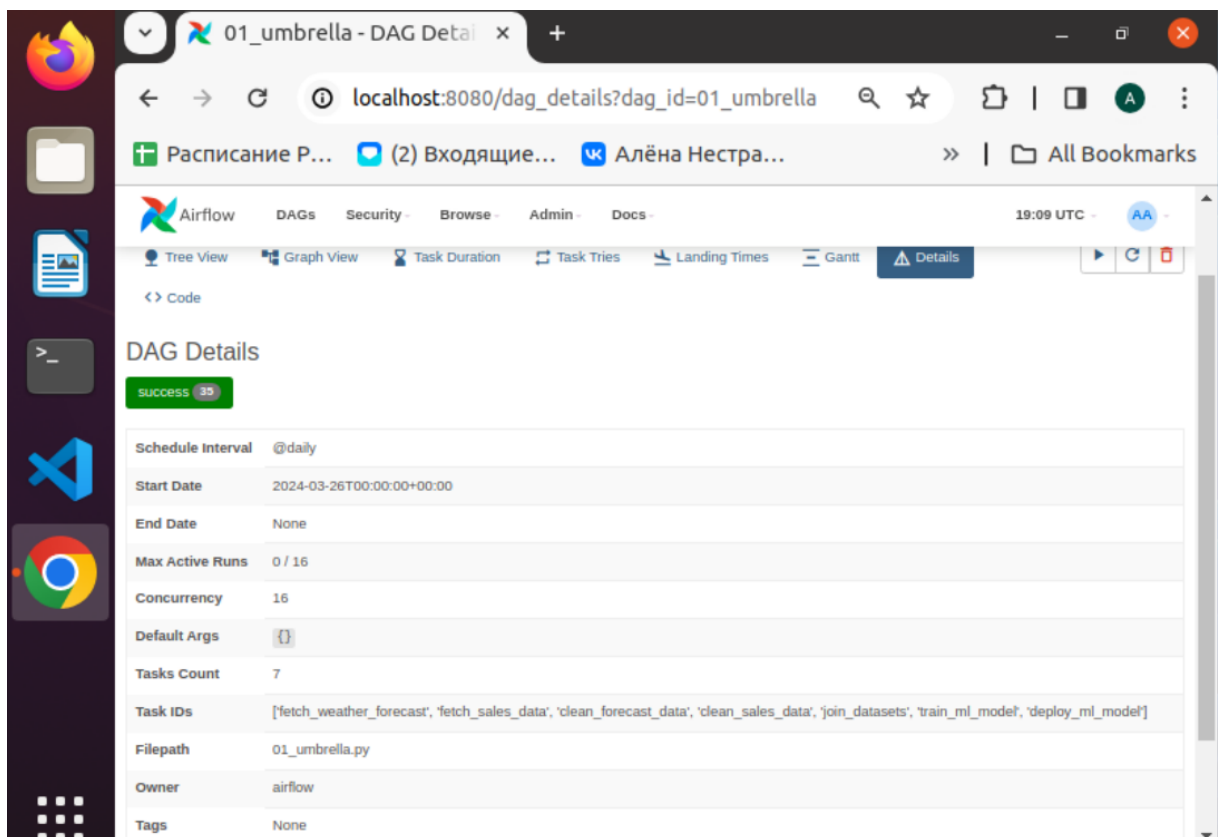
Landing Times отображает информацию о времени посадки задачи (Task) в очередь исполнения. Это время указывает на момент, когда задача была добавлена в очередь Airflow и готова к выполнению.



Gantt. Диаграмма Ганта позволяет анализировать длительность и перекрытие задач. Вы можете быстро определить узкие места и места, на которые тратится большая часть времени при выполнении конкретных запусков группы обеспечения доступности баз данных.



Details (детали) предоставляет дополнительную информацию о задаче (Task) или даге (DAG). В этой вкладке можно найти различные аспекты и параметры, касающиеся задачи или DAG, что обеспечивает более глубокое понимание и контроль над их выполнением.



Code. Дает возможность посмотреть код конвейера данных.

The screenshot shows a web browser window displaying the Apache Airflow interface. The browser's address bar shows the URL `localhost:8080/code?dag_id=01_umbrella&root=`. The Airflow interface has a top navigation bar with links for DAGs, Security, Browse, Admin, and Docs. The main header shows the DAG name '01_umbrella' with a description 'Umbrella example with DummyOperators.' and a schedule of '@daily'. Below the header, there are tabs for different views: Tree View, Graph View, Task Duration, Task Tries, Landing Times, Gantt, and Details. A 'Code' button is highlighted. The code editor displays the following Python code:

```
1 """DAG demonstrating the umbrella use case with dummy operators."""
2
3 import airflow.utils.dates
4 from airflow import DAG
5 from airflow.operators.dummy import DummyOperator
6
7 dag = DAG(
8     dag_id="01_umbrella",
9     description="Umbrella example with DummyOperators.",
10    start_date=airflow.utils.dates.days_ago(5),
11    schedule_interval="@daily",
12 )
13
14 fetch_weather_forecast = DummyOperator(task_id="fetch_weather_forecast", dag=dag)
15 fetch_sales_data = DummyOperator(task_id="fetch_sales_data", dag=dag)
16 clean_forecast_data = DummyOperator(task_id="clean_forecast_data", dag=dag)
17 clean_sales_data = DummyOperator(task_id="clean_sales_data", dag=dag)
18 join_datasets = DummyOperator(task_id="join_datasets", dag=dag)
19 train_ml_model = DummyOperator(task_id="train_ml_model", dag=dag)
20 deploy_ml_model = DummyOperator(task_id="deploy_ml_model", dag=dag)
21
```

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

