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

Лабораторная работа №4.1

Тема:

Проектирование сквозного конвейера ETL Airflow

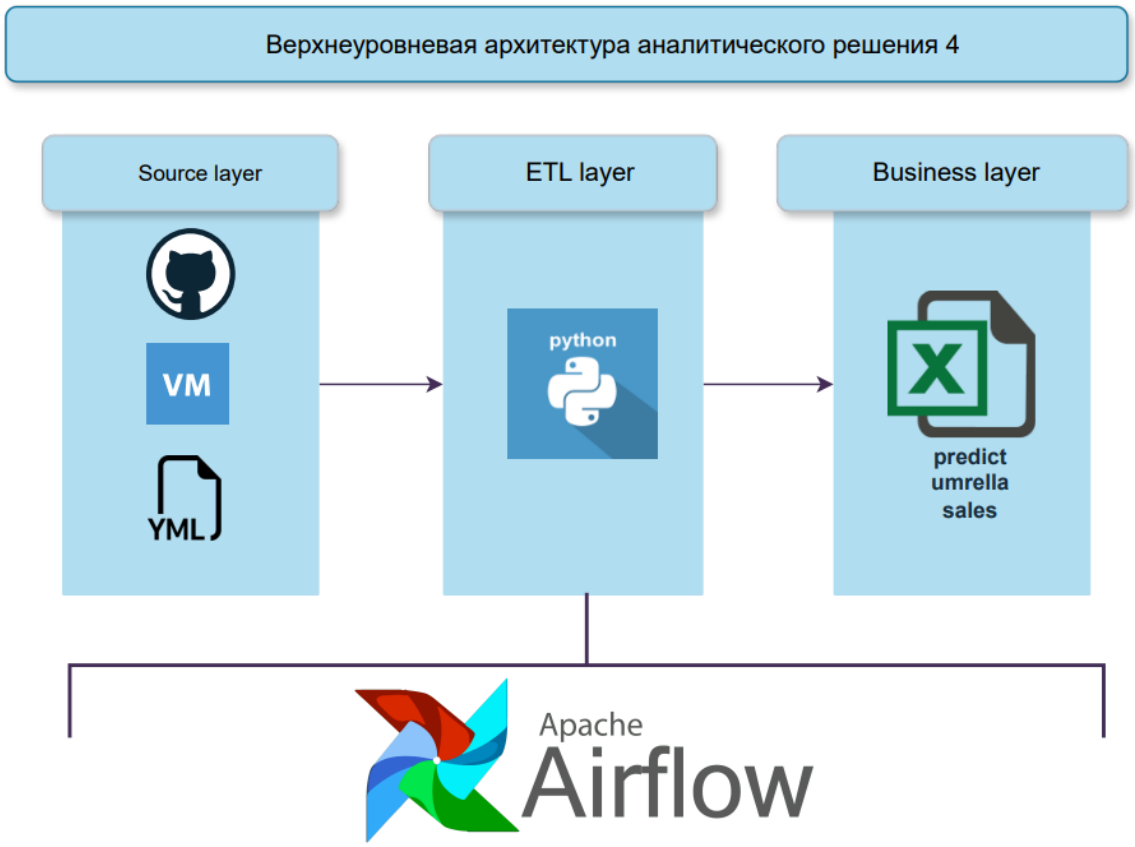
Выполнила: Олифир А. А., группа: АДЭУ-201

Преподаватель: Босенко Т. М.

Москва

2024

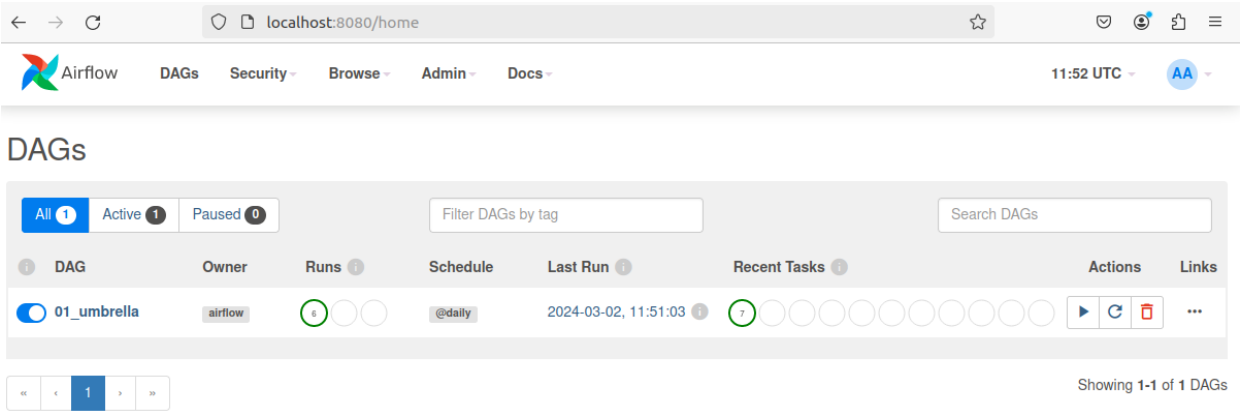
Верхнеуровневая архитектура аналитического решения 4



Описание элементов интерфейса Apache Airflow

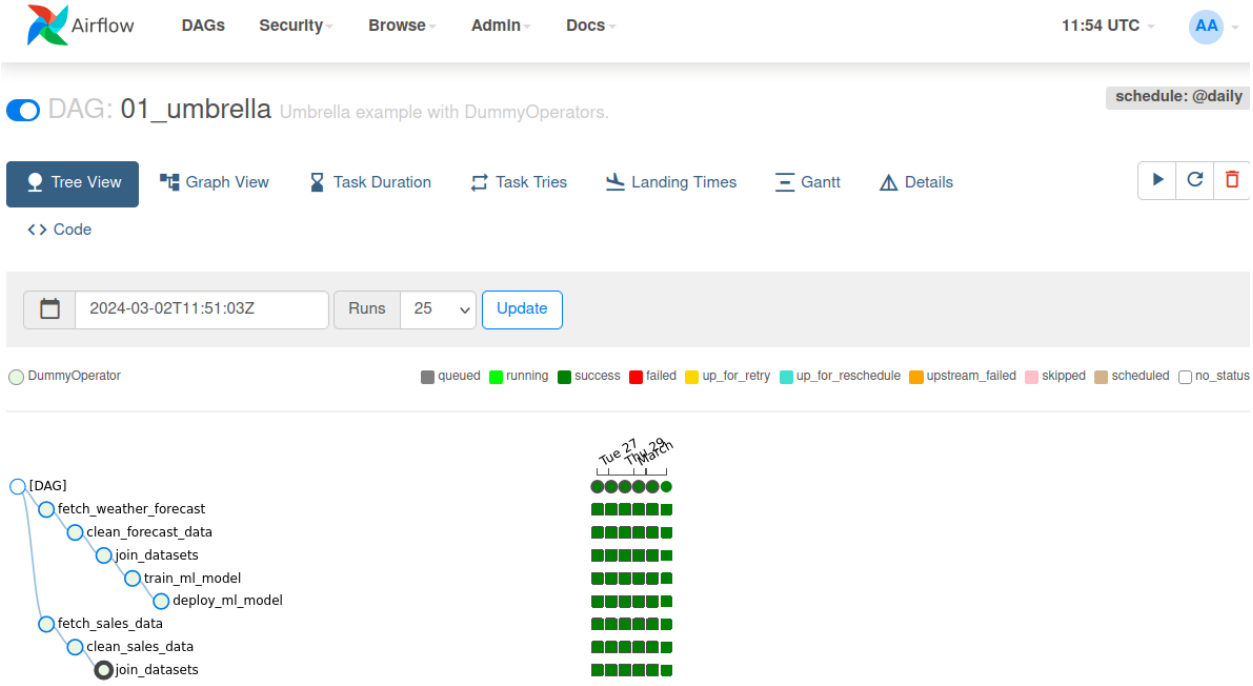
После входа на стартовой странице представлен набор DAGs, предназначенный для демонстрации функциональности Airflow, а также краткая информация по ним.

Просмотреть список DAGов можно по кнопке DAGs верхнего меню. При нажатии на наименование DAGa открывается экран его свойств.



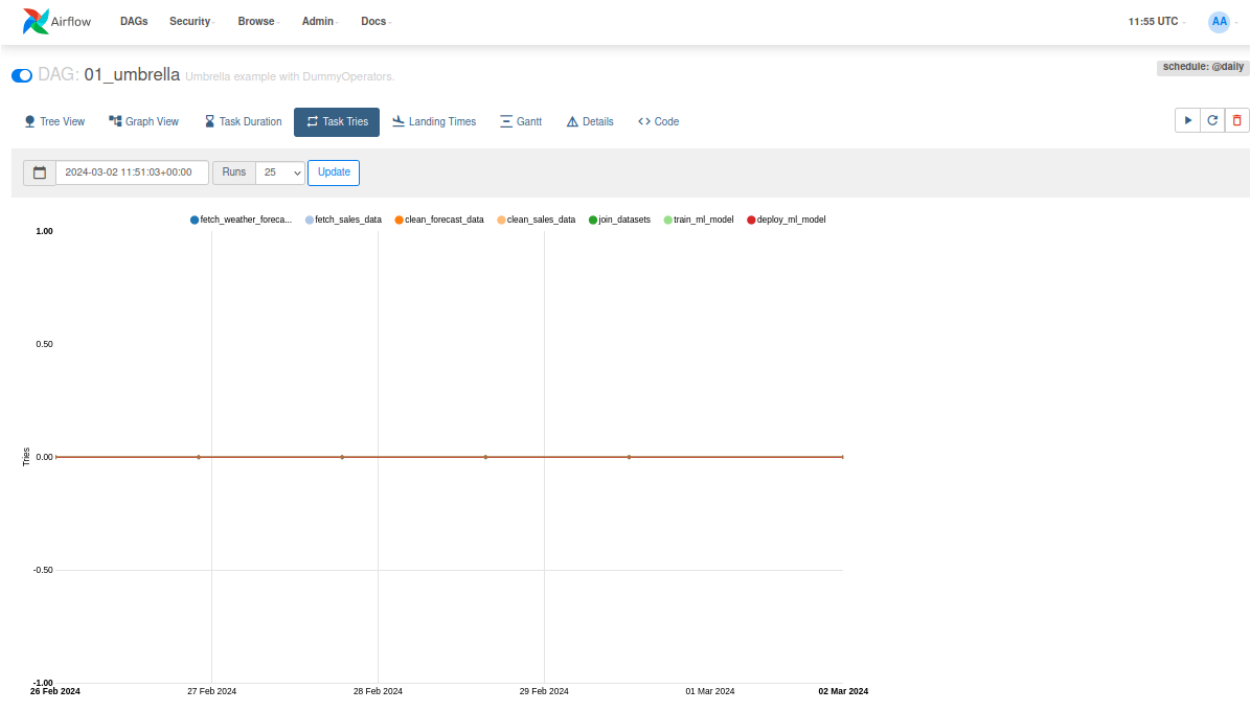
На первой вкладке «Tree View» показана структура направленного ациклического графика (DAG), в котором задачи представлены узлами. Справа от структуры отображаются метрики выполнения всего DAG и индивидуальных

задач, сгруппированные по датам. Каждая колонка метрик соответствует одному запуску DAG. Цветовое кодирование, указанное в легенде в верхнем правом углу, отражает состояния выполнения задач. При наведении курсора на задачу отображаются ее свойства, при клике на задачу можно получить более подробную информацию и список доступных действий.

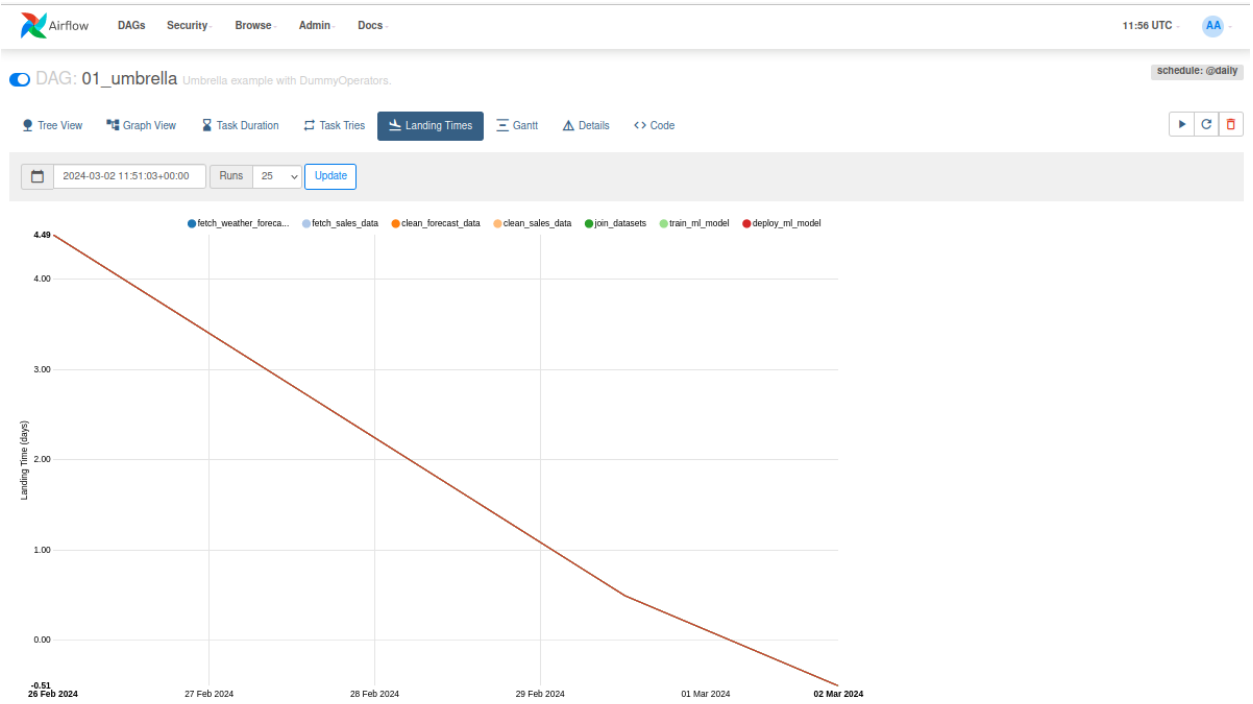


На второй вкладке «Graph View» отображается графическое представление DAGa с возможностью пройти в свойства составляющих его задач.

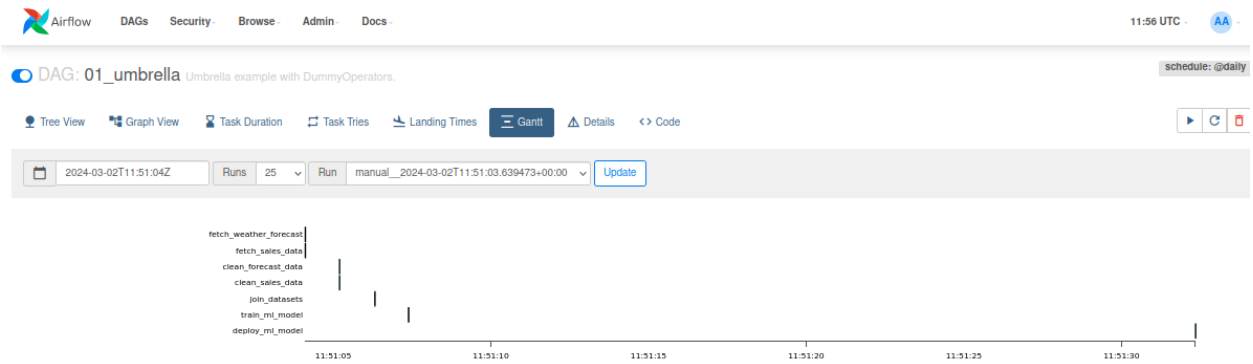
Вкладка «Task Tries» отображает график провалившихся попыток выполнить задачи.



Вкладка «Landing Times» отображает время завершения задачи после ее старта.



Вкладка «Gantt» - график, показывающий распределение времени по задачам относительно времени выполнения всего DAGa.



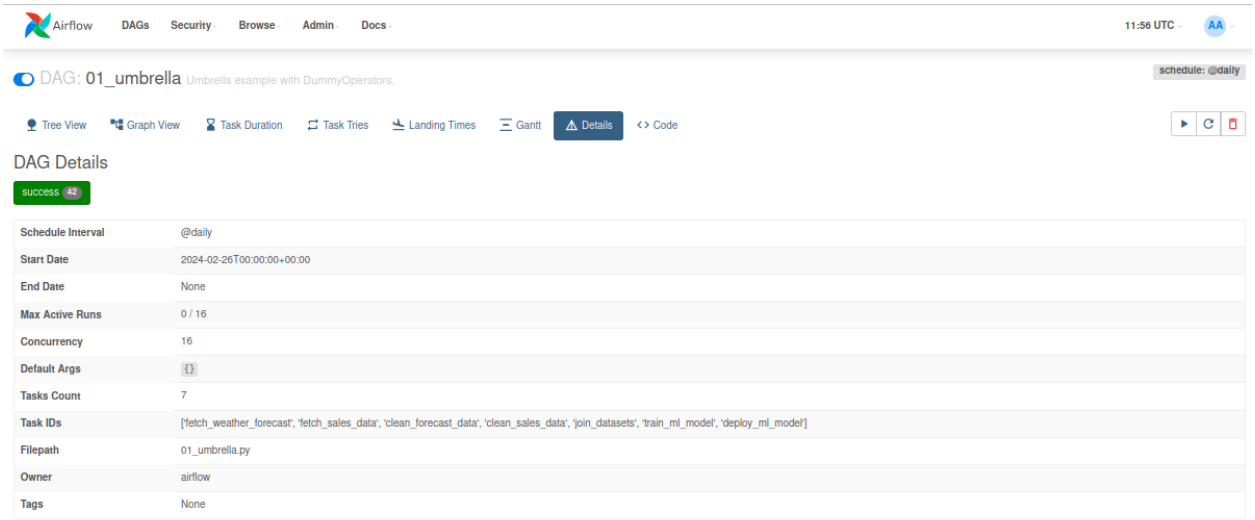
«Details» - основные параметры DAGa, например:

schedule_interval: Расписание выполнения DAG, указывает на частоту запуска DAG.

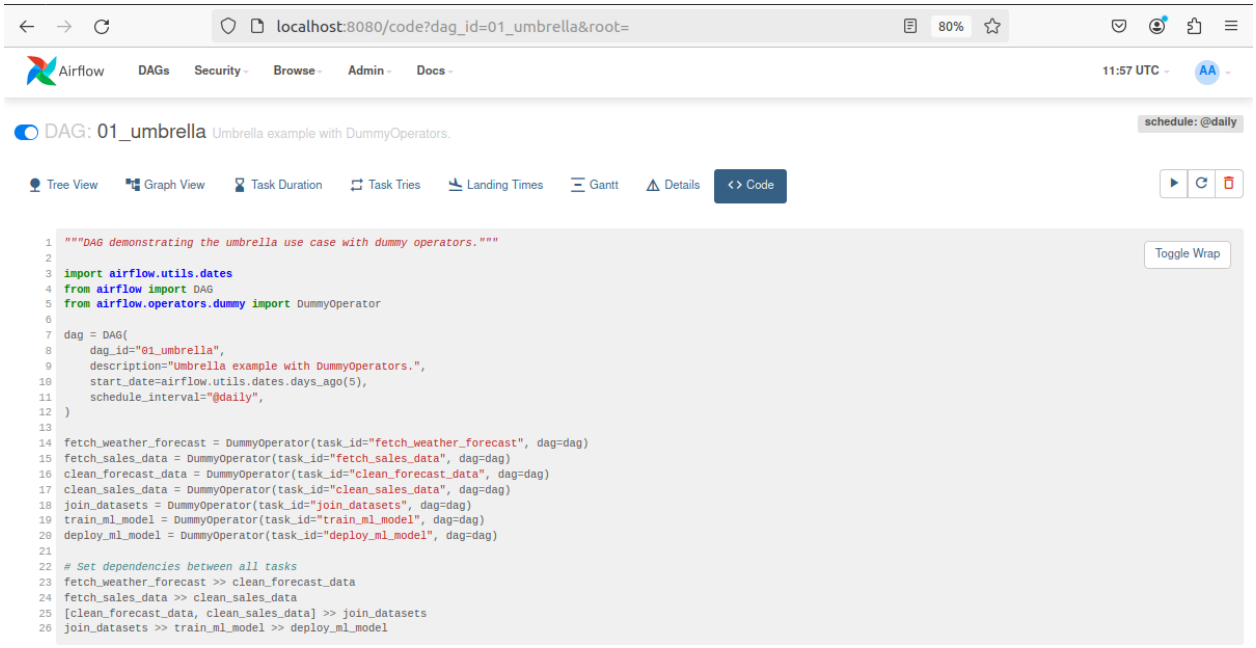
start_date: Дата и время, с которой начинается выполнение DAG.

max_active_runs: Максимальное количество одновременно активных выполнений DAG.

concurrency: Максимальное количество задач, которые могут выполняться одновременно в DAG.



<>code – вкладка, на которой можно посмотреть код DAGa на языке Python.


localhost:8080/code?dag_id=01_umbrella&root=
80%
Airflow
DAGs
Security
Browse
Admin
Docs
11:57 UTC
AA
DAG: 01_umbrella
Umbrella example with DummyOperators.
schedule: @daily
Tree View
Graph View
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Toggle Wrap
"""DAG demonstrating the umbrella use case with dummy operators."""
import airflow.utils.dates
from airflow import DAG
from airflow.operators.dummy import DummyOperator
dag = DAG(
dag_id="01_umbrella",
description="Umbrella example with DummyOperators.",
start_date=airflow.utils.dates.days_ago(5),
schedule_interval="@daily",
)
fetch_weather_forecast = DummyOperator(task_id="fetch_weather_forecast", dag=dag)
fetch_sales_data = DummyOperator(task_id="fetch_sales_data", dag=dag)
clean_forecast_data = DummyOperator(task_id="clean_forecast_data", dag=dag)
clean_sales_data = DummyOperator(task_id="clean_sales_data", dag=dag)
join_datasets = DummyOperator(task_id="join_datasets", dag=dag)
train_ml_model = DummyOperator(task_id="train_ml_model", dag=dag)
deploy_ml_model = DummyOperator(task_id="deploy_ml_model", dag=dag)
# Set dependencies between all tasks
fetch_weather_forecast >> clean_forecast_data
fetch_sales_data >> clean_sales_data
[clean_forecast_data, clean_sales_data] >> join_datasets
join_datasets >> train_ml_model >> deploy_ml_model