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 5.2

Разработка алгоритмов для трансформации данных. Airflow DAG

Выполнила: Сачкова Г.Г., группа: АДЭУ-211

Преподаватель: Босенко Т.М.

Москва

2025

Задачи:

1. Запустить контейнер с кейсом, изучить основные элементы DAG в Apache Airflow.

2. Создать исполняемый файл с расширением .sh, который автоматизирует выгрузку данных из контейнера в основную ОС данных, полученные в результате работы DAG в Apache Airflow.

3. Спроектировать верхнеуровневую архитектуру аналитического решения задания Бизнес-кейса «Rocket» и архитектуру DAG Бизнес-кейса «Rocket» в draw.io.

4. Построить диаграмму Ганта работы DAG в Apache Airflow.

Ход решения:

Прежде чем начать выполнение задания, надо клонировать репозиторий workshop-on-ETL, это продемонстрировано на рисунке 1.

```
Cloning into 'workshop-on-ETL'...
remote: Enumerating objects: 563, done.
remote: Counting objects: 100% (453/453), done.
remote: Compressing objects: 100% (394/394), done.
remote: Total 563 (delta 222), reused 59 (delta 32), pack-reused 110 (from 1)
Receiving objects: 100% (563/563), 5.82 MiB | 5.40 MiB/s, done.
Resolving deltas: 100% (260/260), done.
```

Рисунок 1 – Клонирование репозитория workshop-on-ETL

На рисунке 2 показан запуск контейнеров

```
WARN[0000] /home/dev/Downloads/workshop-on-ETL/business_case_rocket/docker-compose.yml: the attribute `v
 is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 35/35
 ✔ postgres Pulled
 ✔ init Pulled
 ✔ webserver Pulled
 ✔ scheduler Pulled
[+] Running 6/6
 ✔ Network business_case_rocket_default          Created
 ✔ Volume "business_case_rocket_logs"            Created
 ✔ Container business_case_rocket-postgres-1     Started
 ✔ Container business_case_rocket-init-1         Started
 ✔ Container business_case_rocket-webserver-1    Started
 ✔ Container business_case_rocket-scheduler-1    Started
```

Рисунок 2 – Запуск контейнеров

На рисунках 3-6 показано выполнение всех дагов

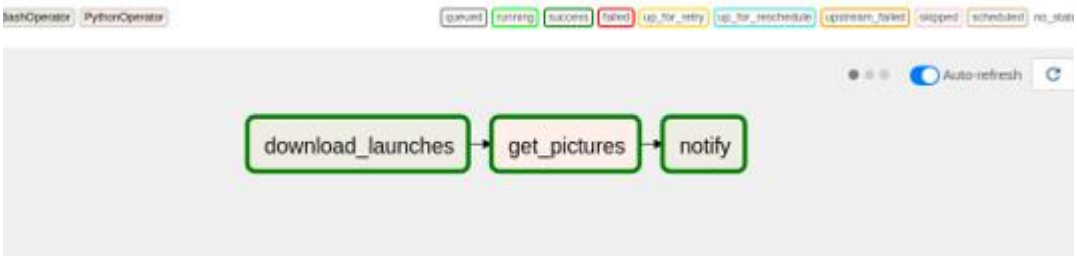

Рисунок 3 – Выполнение первого дага

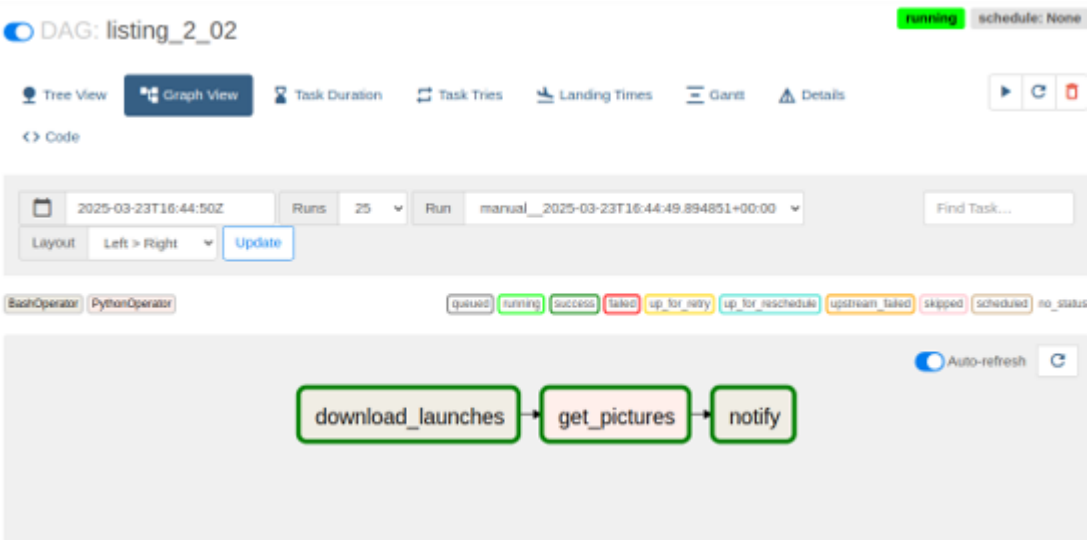

Рисунок 4 – Выполнение второго дага

Рисунок 5 – Выполнение третьего дага

Рисунок 6 – Выполнение четвертого дага

Диаграмма Ганта изображена на рисунке 7.

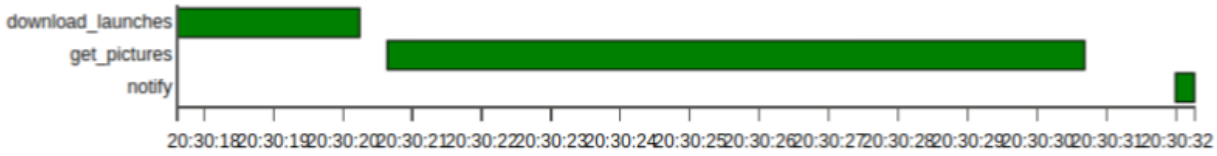

Рисунок 7 – Диаграмма Ганта

На рисунке 8 показан файл .sh.

```
#!/bin/bash
CONTAINER_NAME="business_case_rocket-scheduler-1"
CONTAINER_PATH="/tmp/launches.json"
HOST_PATH="/home/dev/Downloads/workshop-on-ETL/Sachkova"
mkdir -p "$HOST_PATH"
docker cp "$CONTAINER_NAME":"$CONTAINER_PATH" "$HOST_PATH"
echo "The data has been uploaded"
```

Рисунок 8 – Файл SH

На рисунке 9 показано выполнение файла ./export.sh. Как можно заметить, он успешно выполнился.

```
dev@dev-vm:~/Downloads/workshop-on-ETL$ ./export.sh
Successfully copied 24.1kB to /home/dev/Downloads/workshop-on-ETL/Sachkova
The data has been uploaded
```

Рисунок 9 – Выполнение файла ./export.sh

На рисунке 10 продемонстрировано, что файл копировался.

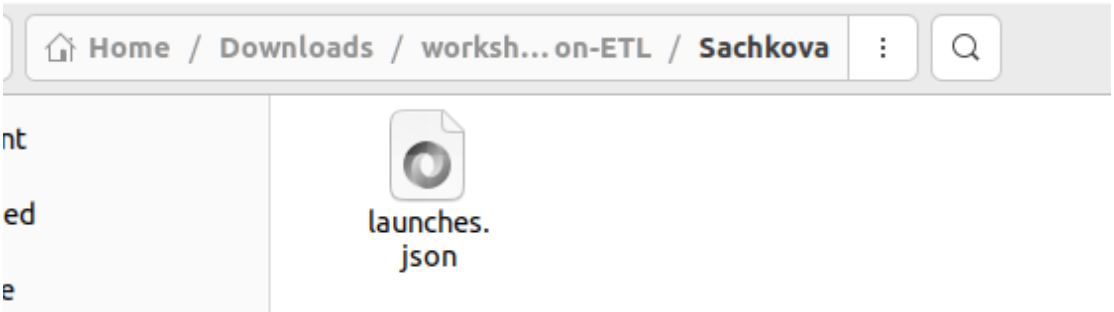

Рисунок 10 – Файл копировался

На рисунке 11 показана верхнеуровневая архитектура аналитического решения Rocket.

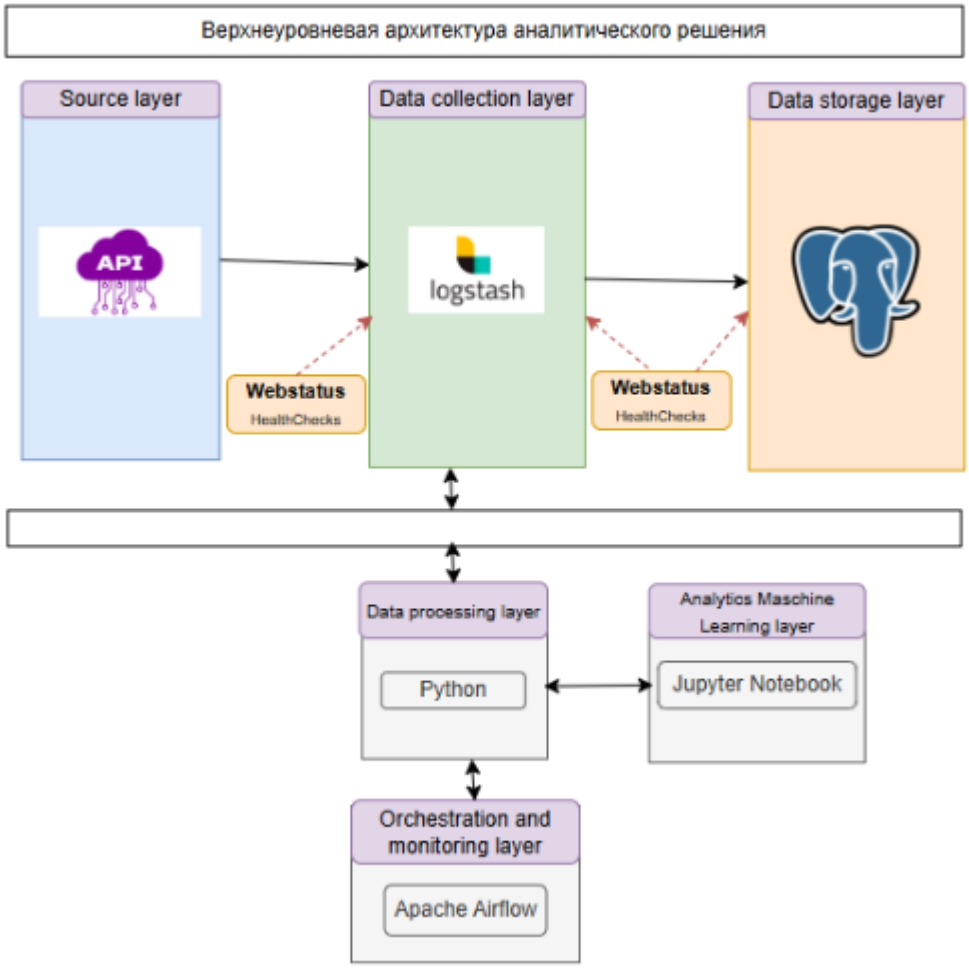

Рисунок 11 – Верхнеуровневая архитектура (Rocket)

Общий процесс

1. Сбор данных: API Rocket извлекает данные.
2. Сбор и обработка данных: Logstash обрабатывает и отправляет данные в PostgreSQL.
3. Хранение данных: Данные сохраняются в PostgreSQL.
4. Обработка данных: Python обрабатывает данные из PostgreSQL.
5. Анализ и машинное обучение: Jupyter Notebook используется для анализа и обучения моделей.
6. Оркестрация и мониторинг: Apache Airflow управляет всем процессом и отслеживает выполнение задач.

На рисунке 12 показана архиетектура DAG.

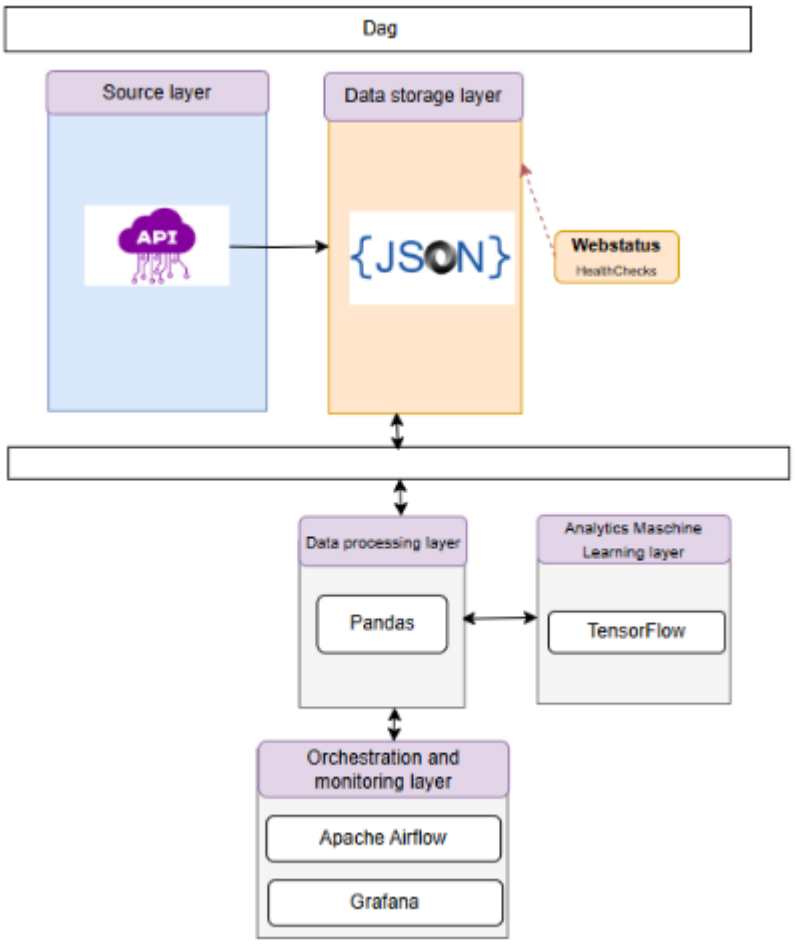

Рисунок 12 – Архитектура DAG

Общий процесс

1. Сбор данных: API извлекает данные.
2. Хранение данных: Данные сохраняются в формате JSON.
3. Обработка данных: Pandas обрабатывает данные из JSON.
4. Анализ и машинное обучение: TensorFlow используется для обучения моделей на обработанных данных.
5. Оркестрация и мониторинг: Apache Airflow управляет всем процессом, а Grafana визуализирует метрики и статус выполнения задач.

Выводы:

1. Запущен контейнер с кейсом, изучить основные элементы DAG в Apache Airflow.

2. Создан исполняемый файл с расширением .sh, который автоматизирует выгрузку данных из контейнера в основную ОС данных, полученные в результате работы DAG в Apache Airflow.

3. Спроектирована верхнеуровневую архитектуру аналитического решения задания Бизнес-кейса «Rocket» и архитектуру DAG Бизнес-кейса «Rocket» в draw.io.

4. Построена диаграмма Ганта работы DAG в Apache Airflow.