# 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

# Вебинар №3

Тема:

«Бизнес кейс «Rocket»

Выполнил(а): Морозова Валерия АДЭУ-211

Преподаватель:

Москва

2025

Развернута Конфигурация BM в VirtualBox.

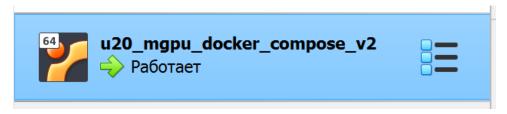


Рисунок 1. Образ развернут

Клонировано на ПК задание Бизнес-кейс «Rocket» в домашний каталог ВМ.

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

```
mgpu@mgpu-VirtualBox:~$ git clone https://github.com/BosenkoTM/workshop-on-ETL. git
Cloning into 'workshop-on-ETL'...
remote: Enumerating objects: 502, done.
remote: Counting objects: 100% (392/392), done.
remote: Compressing objects: 100% (337/337), done.
remote: Total 502 (delta 184), reused 59 (delta 32), pack-reused 110 (from 1)
Receiving objects: 100% (502/502), 5.77 MiB | 4.79 MiB/s, done.
Resolving deltas: 100% (222/222), done.
mgpu@mgpu-VirtualBox:~$ Morozova V
```

Рисунок 2. Задание клонировано в домашний каталог

Запуск контейнера с кейсом, изучение основных элементов DAG в Apache Airflow.

Прежде чем запустить контейнер необходимо остановить все контейнеры Docker.

```
mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business_case_rocket_25$ sudo docker stop $(su
do docker ps -q)
[sudo] password for mgpu:
46692c13c2e2
95cd6c7dcce1
b9e7590beb35
```

Рисунок 3. Остановка всех контейнеров

```
mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business_case_rocket_25$ sudo docker rm $(sudo docker ps -a -q)
46692c13c2e2
95cd6c7dcce1
eab25ddb7335
4a5116148fa8
fd8defe93e21
7738d43e7e56
cbf378713957
b9e7590beb35
mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business case rocket 25$
```

Рисунок 4. Удаление запущенных контейнеров

```
mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business case rocket 25$ sudo docker build -t
  custom-airflow:slim-2.8.1-python3.11 .
   [+] Building 1.4s (7/7) FINISHED
                                                                                                                                             docker:default
    => [internal] load build definition from Dockerfile
                                                                                                                                                               0.0s
    => => transferring dockerfile: 568B
                                                                                                                                                               0.05
    => [internal] load metadata for docker.io/apache/airflow:slim-2.8.1-python3.11
                                                                                                                                                               1.3s
    => [internal] load .dockerignore
                                                                                                                                                               0.0s
    => => transferring context: 2B
                                                                                                                                                               0.0s
    => [1/3] FROM docker.io/apache/airflow:slim-2.8.1-python3.11@sha256:751babd58a83e44ae23
                                                                                                                                                               0.0s
                                                                                        pandas
                                                                                                                                             joblib
    => CACHED [2/3] RUN pip install --no-cache-dir
                                                                                                               scikit-learn
                                                                                                                                                               0.05
    => CACHED [3/3] RUN mkdir -p /opt/airflow/data /opt/airflow/logs
                                                                                                                           && chown -R airflo
                                                                                                                                                               0.0s
                                                                                                                                                               0.0s
    => exporting to image
    => => exporting layers
                                                                                                                                                               0.0s
    => => writing image sha256:b27eadb226ef294cc74800de710609fcd30e2eldf47a1400fb103ba9c73b 0.0s
                         Рисунок 5. Сборка Docker образа с указанием тэга
 mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business case rocket 25$ sudo cho
    wn -R 50000:50000 ./data
    [sudo] password for mgpu:
 mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business case rocket 25$ curl -L
    "https://ll.thespacedevs.com/2.0.0/launch/upcoming" -o data space.txt
                              % Received % Xferd Average Speed
        % Total
                                                                                                    Time
                                                                                                                     Time
                                                                                                                                       Time Current
                                                                                                   Total
                                                                     Dload Upload
                                                                                                                     Spent
                                                                                                                                      Left Speed
        0
                    0
                              0
                                          0
                                                    0
                                                                0
                                                                                           0 --:--:--
                                                                                                                   0:00:01 --:--:-
                                                                             0
    100 25908 100 25908
                                                    0
                                                               0
                                                                       7787
                                                                                           0 0:00:03 0:00:03 --:-- 47191
 o mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business case rocket 25$
                                          Рисунок 6. Настройка прав доступа
    o mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business case rocket 25$ sudo doc
       ker compose up --build
       [sudo] password for mgpu:
       [+] Running 4/2
         ✓ Container business case rocket 25-postgres-1
                                                                                                                                                          0.1s
        ✓ Container business_case_rocket 25-init-1
                                                                                                                                                         0.1s
                                                                                                      Created
        ✓ Container business case rocket 25-webserver-1
                                                                                                                                                         0.1s
                                                                                                     Created
         ✓ Container business case rocket 25-scheduler-1 Created
                                                                                                                                                         0.1s
       Attaching to init-1, postgres-1, scheduler-1, webserver-1
       postgres-1
                                   PostgreSQL Database directory appears to contain a database; Skippi
       postgres-1
       ng initialization
       postgres-1
                                2025-03-29 00:33:19.782 UTC [1] LOG: starting PostgreSQL 12.18 on
       postgres-1
                                               Рисунок 7. Запуск контейнеров
                                                                                                                         Schedule: @daily Next Run: 2025-03-28, 00:00:00

    DAG: download rocket local Download rocket pictures of recently launched rockets

                                                                                                                                                               ▶ □
             Task Tries ★ Landing Times 🔁 Gantt 🛕 Details 🔸 Code
                                                                                                                       Audit Log
     03/28/2025, 10:59:36 AM 🗎 25 🗸 All Run Types 🗸 All Run States 🗸 Clear Filters
   Press shift + / for Shortcut
                                                                 deferred failed queued removed restarting running scheduled skipped success up_or_reschedule up_for_retry upstream_failed no_starting running scheduled skipped success up_or_reschedule up_for_retry upstream_failed no_starting running scheduled skipped skipped success up_or_reschedule up_for_retry upstream_failed no_starting running scheduled skipped skipped scheduled skipped skipped scheduled 
                                                                                                                                               Clear - Mark state as... .
                                         download rocket local / ▶ 2025-03-28, 00:00:00 UTC
                                         ↑ Details • Graph ☑ Gantt <> Code
                                                                                                                                                       Left -> Right 🗸
   download launches
```

Рисунок 8. DAG выполнен успешно

get\_pictures

notify

download launches

get pictures

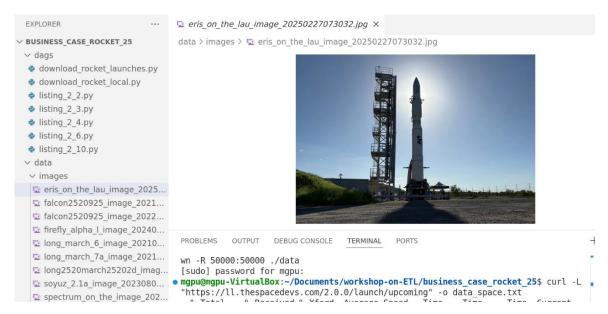


Рисунок 9. Изображения выгрузились в папку

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

```
mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business_case_rocket_25$ chmod +x export_images.sh
mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business_case_rocket_25$ ./export_images.sh
C6 29 мар 2025 04:19:49 МSK: Копирование изображений из контейнера...
Права доступа установлены (755)

Рисунок 10. Предоставление доступа
```

```
$ export_images.sh

1  # Конфигурация

2  CONTAINER_DATA_PATH="/home/mgpu/Documents/workshop-on-ETL/business_case_rocket_25/data/im

3  LOCAL_TARGET_DIR="home/mgpu/Downloads/rocket_images" # Целевая папка

4  

5  # Копируем файлы
6  echo "$(date): Копирование изображений из контейнера"
7  cp -r "$CONTAINER_DATA_PATH"/* "$LOCAL_TARGET_DIR"

8  

9  

10  # Устанавливаем корректные права
11  chmod -R 755 "$LOCAL_TARGET_DIR"

12  echo -e "\nПрава доступа установлены (755)"
```

Рисунок 11. Файл с расширением .sh

1.1 Спроектировать верхнеуровневую архитектуру аналитического решения задания **Бизнес-кейса** «**Rocket**» в draw.io. Необходимо использовать:

```
Source Layer - слой источников данных.

Storage Layer - слой хранения данных.

Business Layer - слой для доступа к данным пользователей.
```

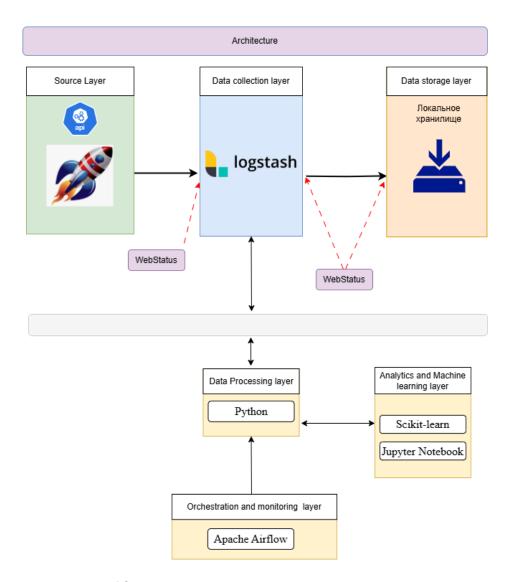


Рисунок 12. Архитектура аналитического решения задания Бизнескейса «Rocket»

1.2 Спроектировать архитектуру DAG **Бизнес-кейса** «**Rocket**» в draw.io. Необходимо использовать:

Source Layer - слой источников данных.

Storage Layer - слой хранения данных.

Business Layer - слой для доступа к данным пользователей.

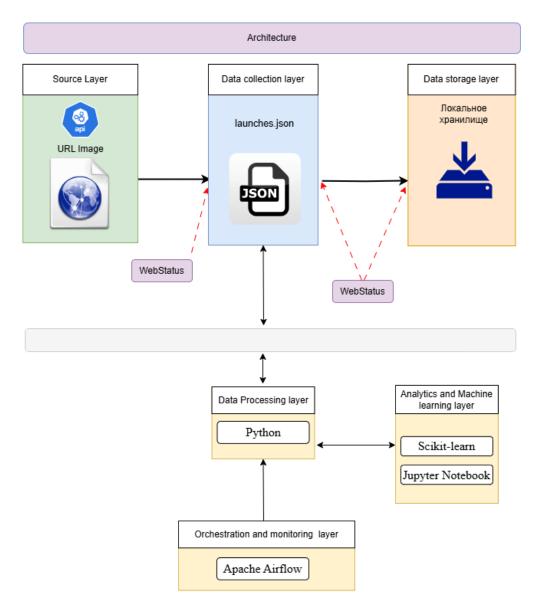


Рисунок 13. Архитектура DAG Бизнес-кейса «Rocket»

1.3 Построить диаграмму Ганта работы DAG в Apache Airflow.

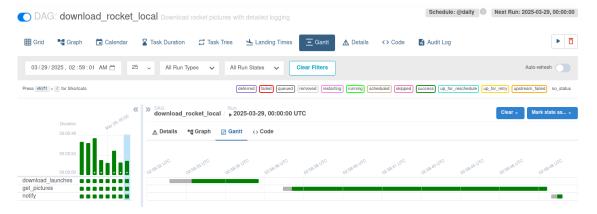


Рисунок 14. Диаграмма Ганта

### Индивидуальное задание

Настроить файл логирования для загрузки изображений в процессе работы DAG

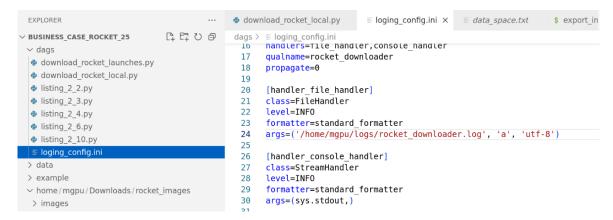


Рисунок 15. Создать файл формата ini

```
≡ loging_config.ini
download_rocket_local.py ×
                                              ≡ data_space.txt
                                                                $ export_in
import logging
  3
      from datetime import timedelta
  4
  5
      import airflow.utils.dates
  7
      import requests
  8
      import requests.exceptions as requests exceptions
  9
      from airflow import DAG
 10
      from airflow.operators.bash import BashOperator
 11
      from airflow.operators.python import PythonOperator
 12
 13
 14
      # Настройка логгера
 15
      log = logging.getLogger( name )
 16
      logging.basicConfig(
 17
          level=logging.INFO,
 18
          format='%(asctime)s - %(name)s - %(levelname)s - %(message)s',
 19
              logging.FileHandler('/opt/airflow/logs/rocket downloader.log
 20
              logging.StreamHandler()
 21
 22
          1
 23
```

Рисунок 16. Настройка логгера

### Вывод:

Запуск DAG и отображение диаграммы Ганта для отслеживания времени выполнения каждого этапа позволяет выявлять места для доработки и совершенствования. Бизнесу в свою очередь важно видеть, на что больше всего уходит время.

Автоматическое копирование и перенос выгруженных файлов на локальную ОС так же сокращает рабочее время и бюджет, выделенный на задачи подобного рода (в случаях, когда оплата по ставке за час).

Внедрение системы логирования и метрик скорости загрузки изображений в DAG Apache Airflow не просто улучшает техническую сторону процесса, но и напрямую влияет на финансовые показатели бизнеса. Снижение затрат на хостинг и трафик за счет оптимизации размера изображений. Логи ошибок (ConnectionError, Timeout) позволяют быстро находить и устранять проблемы с API или внешними серверами. Ускорение загрузки на 100–500 мс может увеличить конверсию на 1–5%. Можно заранее масштабировать серверы или кэшировать изображения, избегая аварийных ситуаций.