

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**

**№ 5 Тема:**

«Airflow DAG».

Выполнила: Нестратова А.М., АДЭУ-201

Преподаватель: Босенко Т.М.

Москва

2024

## Постановка задачи

5.1.1. Развернуть ВМ ubuntu\_mgpu.ova в VirtualBox.

5.1.2. Клонировать на ПК задание **Бизнес-кейс «Rocket»** в домашний каталог ВМ.

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

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

- Создать DAG согласно алгоритму, который предоставит преподаватель.
- Изучить логи, выполненного DAG. Скачать логи из контейнера на основную ОС, используя команду:

```
docker cp <container_hash>:/path/to/zip/file.zip /path/on/host/new_name.zip
```

- Выгрузить полученный результат работы DAG в основной каталог ОС, используя команду:

```
docker cp -r <containerId>:/path/to/directory /path/on/host
```

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

5.1.5. 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.

5.1.6. 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.

5.1.7. Построить диаграмму Ганта работы DAG в Apache Airflow.

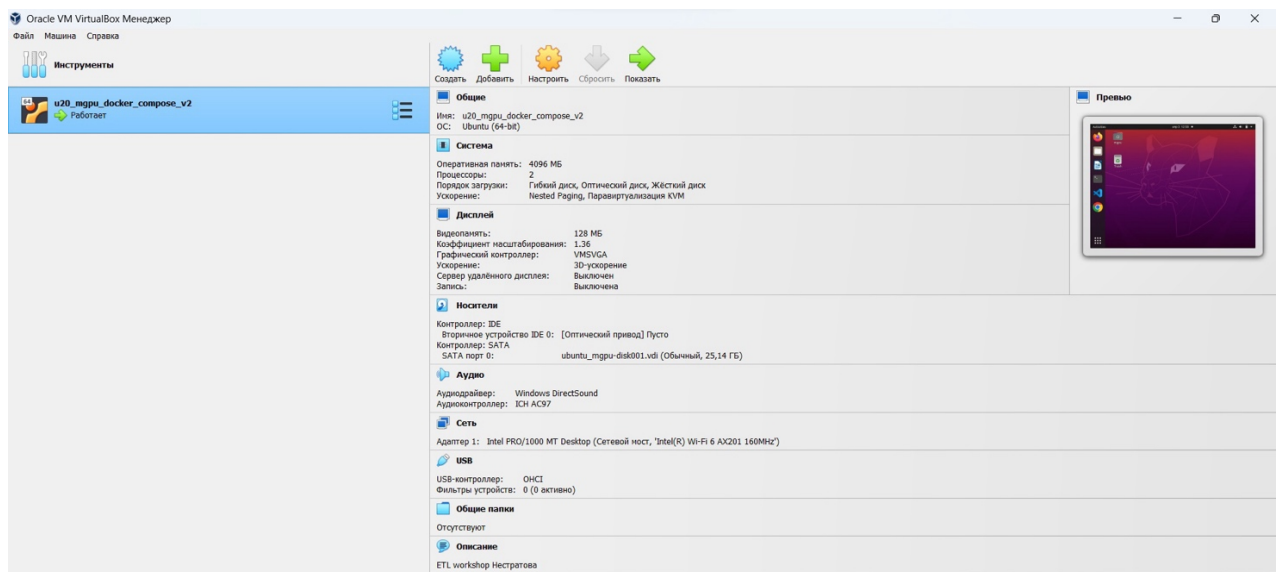
5.1.8. Результаты исследований представить в виде файла ФИО-05.pdf, в котором отражены следующие результаты:

- постановка задачи;
- исходный код всех DAGs, которые требовались для решения задачи, а также представить граф DAG в Apache Airflow;
- верхнеуровневая архитектура задания **Бизнес-кейса «Rocket»**, выполненная в draw.io;
- архитектура DAG **Бизнес-кейса «Rocket»** , выполненная в draw.io;
- скрин лог-файла результатов работы DAGs в Apache Airflow;
- диаграмма Ганта DAG в Apache Airflow.

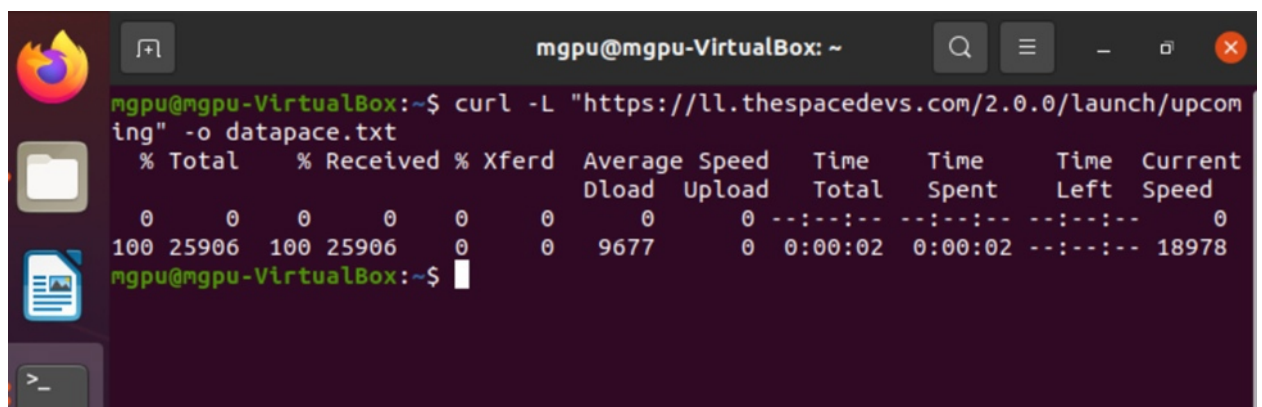
архитектура DAG Бизнес-кейса «Rocket», выполненная в draw.io; скрин лог-файла результатов работы DAGs в Apache Airflow; диаграмма Ганта DAG в Apache Airflow.

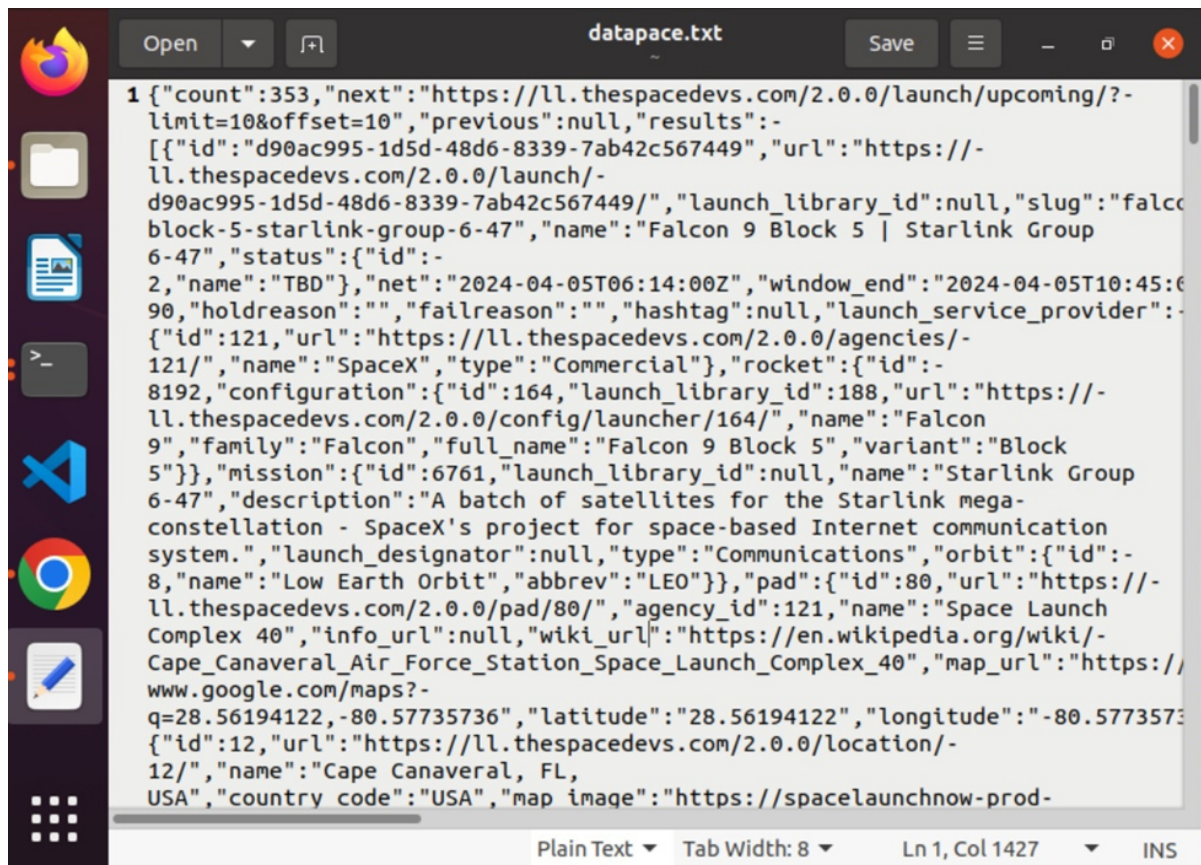
## Решение задачи

Начало работы в VirtualBox.



Используем утилиту `curl` для загрузки данных с заданного URL-адреса "<https://ll.thespacedevs.com/2.0.0/launch/upcoming>" с автоматическим перенаправлением (-L) и сохранение этих данные в файл с именем `data_space.txt`.





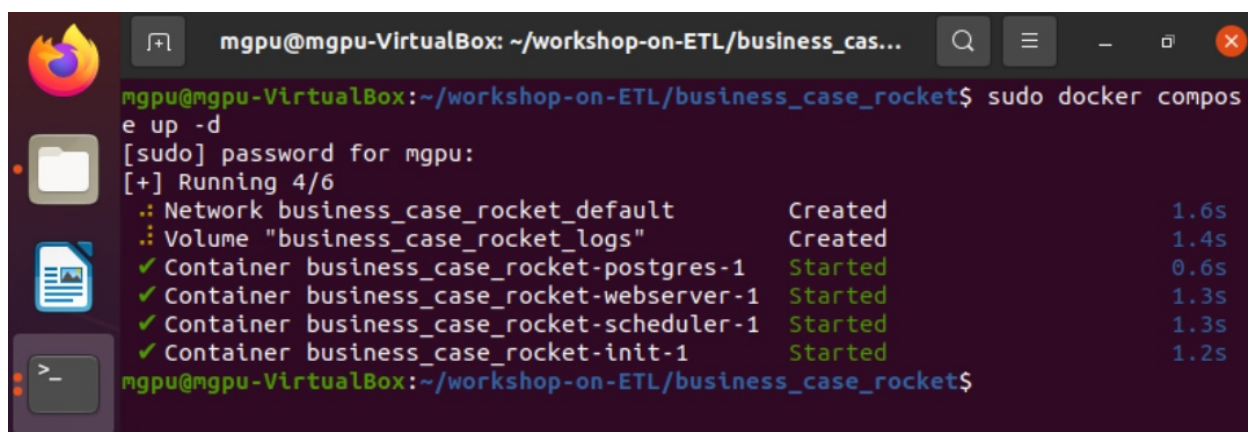
A screenshot of a text editor window titled "datapace.txt". The editor shows a JSON array with one object representing a launch. The object includes details like count (353), next/previous URLs, launch library ID, slug ("falcon-9-block-5-starlink-group-6-47"), name ("Falcon 9 Block 5 | Starlink Group 6-47"), status, net ("2024-04-05T06:14:00Z"), window\_end ("2024-04-05T10:45:00Z"), holdreason, failreason, hashtag, launch\_service\_provider, rocket (SpaceX), configuration (Falcon 9), mission (Starlink Group 6-47), description, launch\_designator, type (Communications), orbit (Low Earth Orbit), pad (Cape Canaveral, FL), and map image URL. The status is "TBD".

```
1 [{"count":353,"next":"https://ll.thespacedevs.com/2.0.0/launch/upcoming/?-limit=10&offset=10","previous":null,"results":-[{"id":"d90ac995-1d5d-48d6-8339-7ab42c567449","url":"https://-ll.thespacedevs.com/2.0.0/launch/-d90ac995-1d5d-48d6-8339-7ab42c567449/","launch_library_id":null,"slug":"falcon-9-block-5-starlink-group-6-47","name":"Falcon 9 Block 5 | Starlink Group 6-47","status":{"id":-2,"name":"TBD"},"net":"2024-04-05T06:14:00Z","window_end":"2024-04-05T10:45:00Z","holdreason":"","failreason":"","hashtag":null,"launch_service_provider":{"id":121,"url":"https://ll.thespacedevs.com/2.0.0/agencies/-121/","name":"SpaceX","type":"Commercial"},"rocket":{"id":-8192,"configuration":{"id":164,"launch_library_id":188,"url":"https://-ll.thespacedevs.com/2.0.0/config/launcher/164/","name":"Falcon 9","family":"Falcon","full_name":"Falcon 9 Block 5","variant":"Block 5"},"mission":{"id":6761,"launch_library_id":null,"name":"Starlink Group 6-47","description":"A batch of satellites for the Starlink mega-constellation - SpaceX's project for space-based Internet communication system.","launch_designator":null,"type":"Communications","orbit":{"id":-8,"name":"Low Earth Orbit","abbrev":"LEO"},"pad":{"id":80,"url":"https://-ll.thespacedevs.com/2.0.0/pad/80/","agency_id":121,"name":"Space Launch Complex 40","info_url":null,"wiki_url":"https://en.wikipedia.org/wiki/-Cape_Canaveral_Air_Force_Station_Space_Launch_Complex_40","map_url":"https://www.google.com/maps?-q=28.56194122,-80.57735736","latitude":"28.56194122","longitude":"-80.57735736"},"location":{"id":12,"url":"https://ll.thespacedevs.com/2.0.0/location/-12/","name":"Cape Canaveral, FL, USA","country code":"USA","map image":"https://spacelaunchnow-prod-
```

Необходимо клонировать репозиторий в каталог VM



Запускаем контейнер

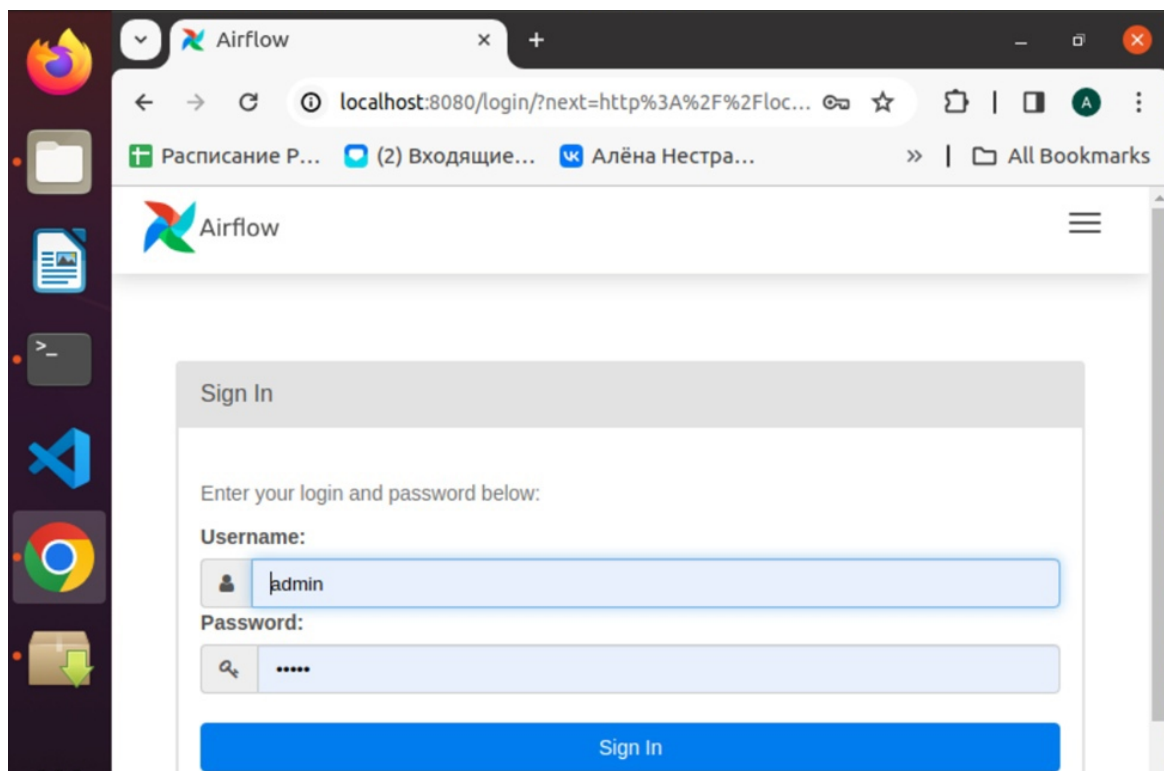


A screenshot of a terminal window showing the output of the "docker compose up -d" command. The output indicates that the containers were successfully started. The command prompt is "mgpu@mgpu-VirtualBox: ~/workshop-on-ETL/business\_case\_rocket\$".

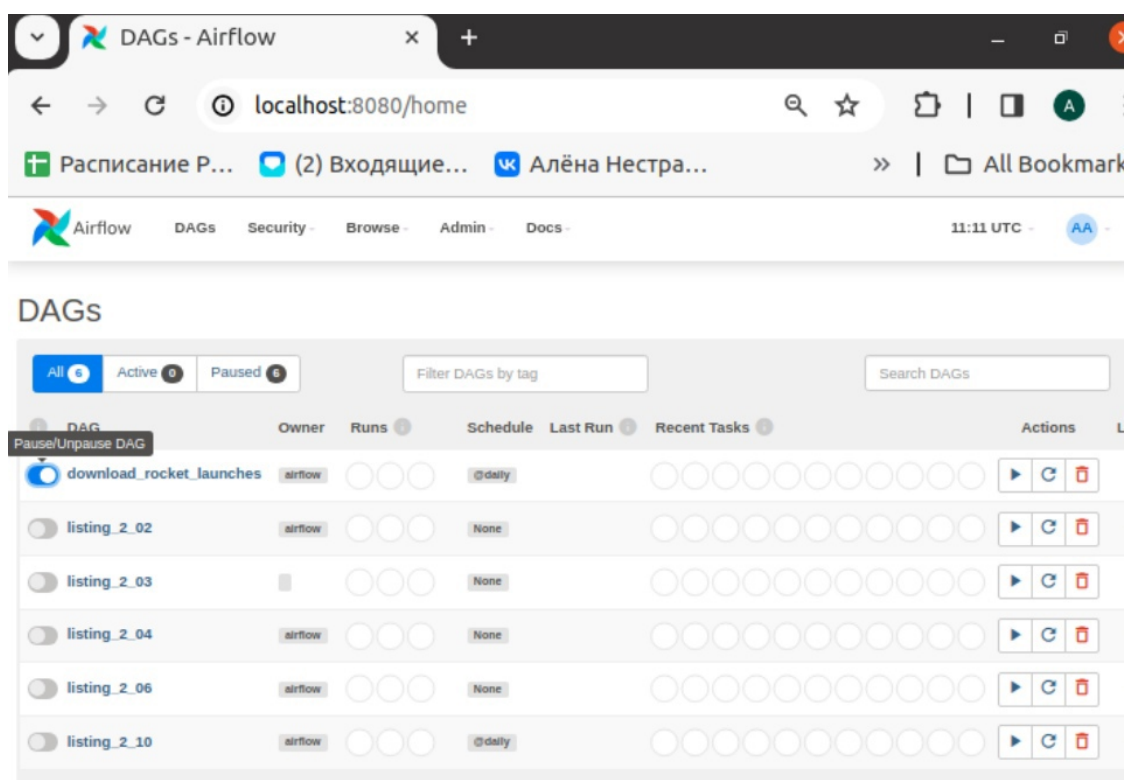
```
mgpu@mgpu-VirtualBox: ~/workshop-on-ETL/business_case_rocket$ sudo docker compose up -d
[sudo] password for mgpu:
[+] Running 4/6
  :: Network business_case_rocket_default          Created           1.6s
  :: Volume "business_case_rocket_logs"           Created           1.4s
  ✓ Container business_case_rocket-postgres-1     Started           0.6s
  ✓ Container business_case_rocket-webserver-1    Started           1.3s
  ✓ Container business_case_rocket-scheduler-1    Started           1.3s
  ✓ Container business_case_rocket-init-1         Started           1.2s
mgpu@mgpu-VirtualBox: ~/workshop-on-ETL/business_case_rocket$
```

Запускаем DAG в Airflow и проверяем пользовательский интерфейс Airflow, перейдя по линку

`http://localhost:8080/`.

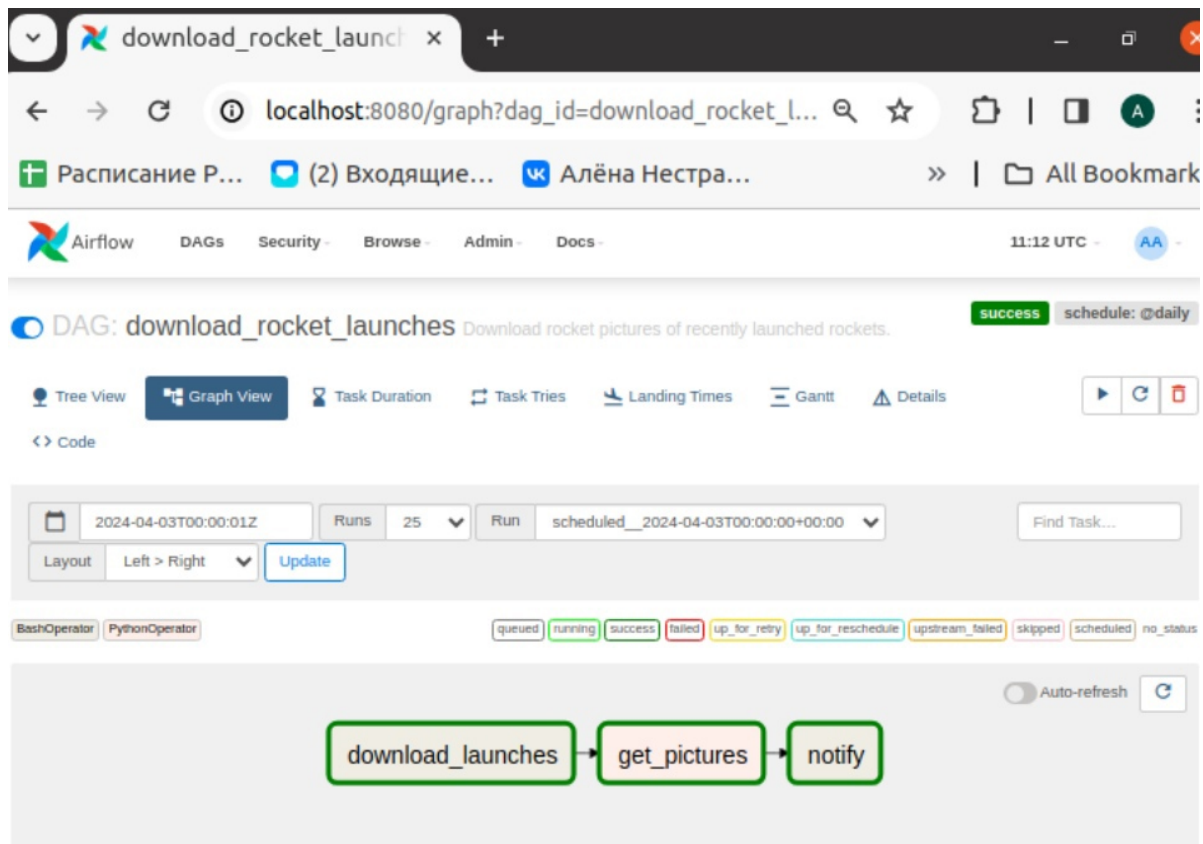


Запускаем DAG

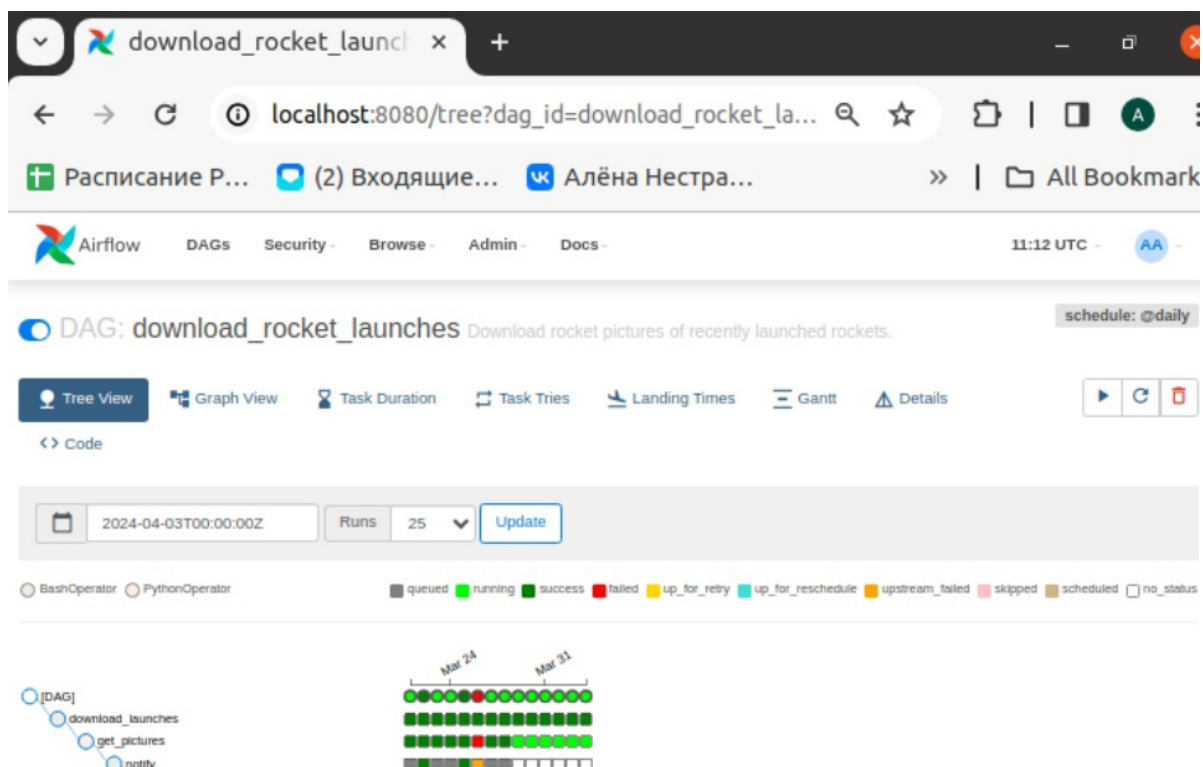




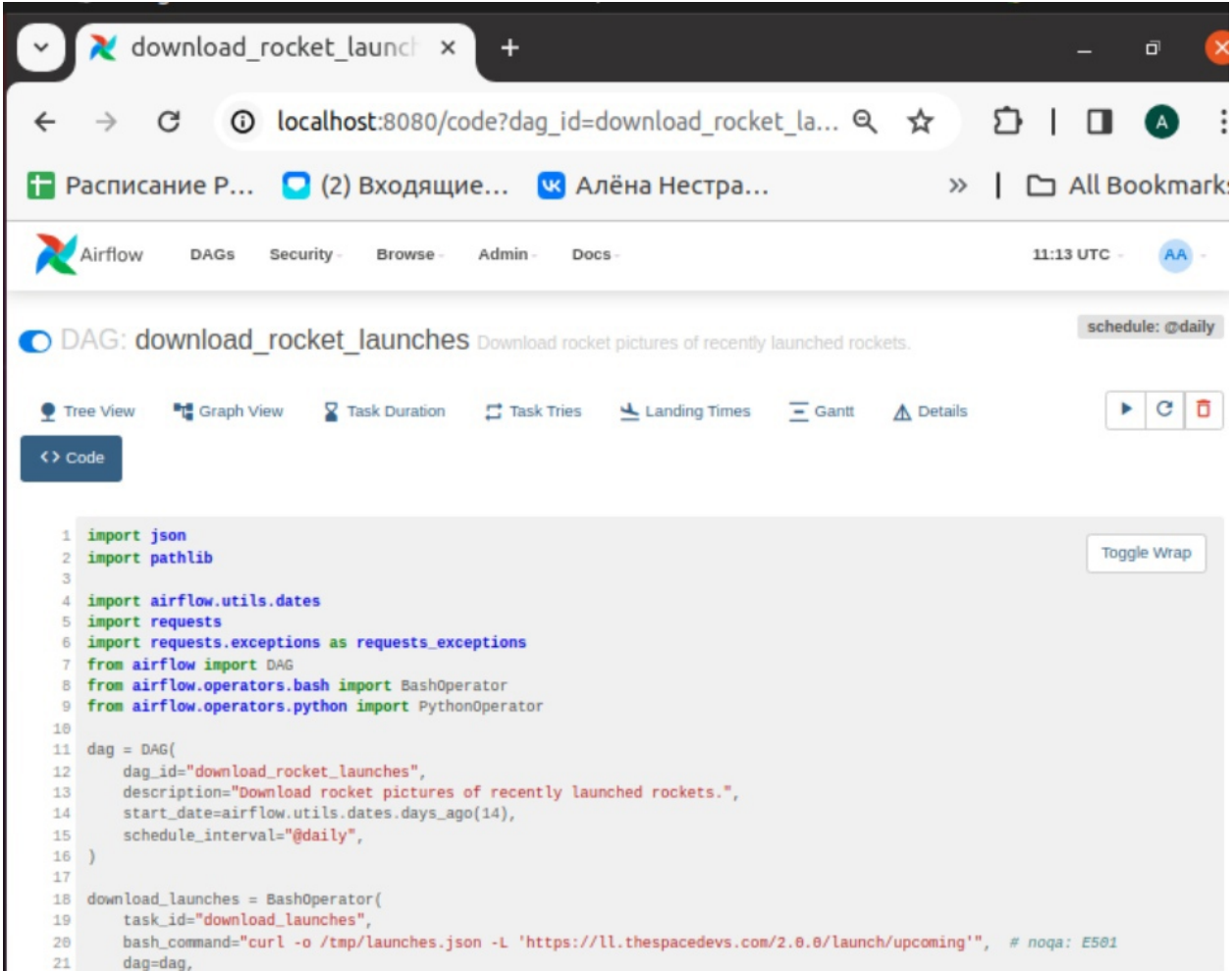
После того как DAG запущен необходимо посмотреть его граф в Apache Airflow перейдя на вкладку Graph View.



Дополнительно необходимо посмотреть диаграмму Ганта DAG в Apache Airflow на вкладке Gantt.



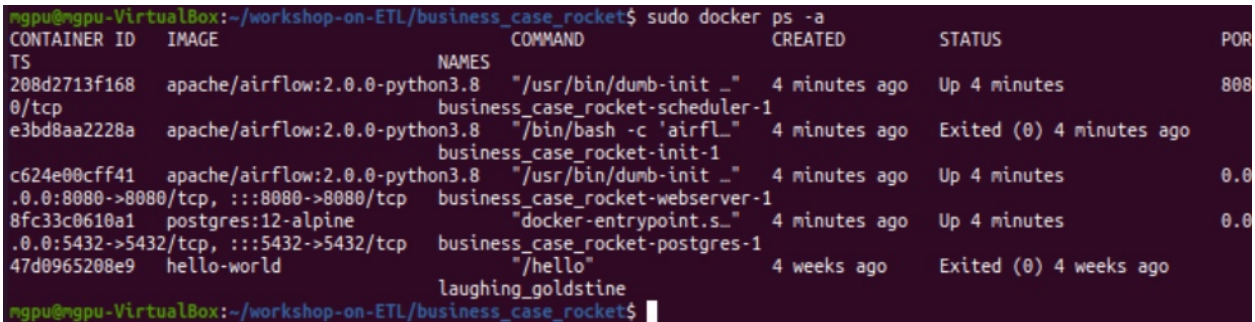
На вкладке Code необходимо просмотреть исходный код DAG.



The screenshot shows the Apache Airflow web interface in a browser. The URL is `localhost:8080/code?dag_id=download_rocket_la...`. The page title is "DAG: download\_rocket\_launches" with a description "Download rocket pictures of recently launched rockets." and a schedule of "@daily". The interface includes tabs for Tree View, Graph View, Task Duration, Task Tries, Landing Times, Gantt, and Details. A "Code" button is visible, and the code editor shows the following Python code:

```
1 import json
2 import pathlib
3
4 import airflow.utils.dates
5 import requests
6 import requests.exceptions as requests_exceptions
7 from airflow import DAG
8 from airflow.operators.bash import BashOperator
9 from airflow.operators.python import PythonOperator
10
11 dag = DAG(
12     dag_id="download_rocket_launches",
13     description="Download rocket pictures of recently launched rockets.",
14     start_date=airflow.utils.dates.days_ago(14),
15     schedule_interval="@daily",
16 )
17
18 download_launches = BashOperator(
19     task_id="download_launches",
20     bash_command="curl -o /tmp/launches.json -L 'https://ll.thespacedevs.com/2.0.0/launch/upcoming'", # noqa: E501
21     dag=dag,
```

Необходимо определить номера контейнера, в котором выполнен DAG. Таким образом, хеш контейнера `208d2713f168`, а имя контейнера `business_case_rocket-scheduler-1`.



The screenshot shows a terminal window with the command `sudo docker ps -a` executed. The output is a table of Docker containers:

CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
208d2713f168	apache/airflow:2.0.0-python3.8	business_case_rocket-scheduler-1	"/usr/bin/dumb-init _"	4 minutes ago	Up 4 minutes	8080/tcp
e3bd8aa2228a	apache/airflow:2.0.0-python3.8	business_case_rocket-init-1	"/bin/bash -c 'airfl_'"	4 minutes ago	Exited (0) 4 minutes ago	
c624e00cfff41	apache/airflow:2.0.0-python3.8	business_case_rocket-webserver-1	"/usr/bin/dumb-init _"	4 minutes ago	Up 4 minutes	0.0.0.0:8080->8080/tcp
8fc33c0610a1	postgres:12-alpine	business_case_rocket-postgres-1	"docker-entrypoint.s_"	4 minutes ago	Up 4 minutes	0.0.0.0:5432->5432/tcp
47d0965208e9	hello-world	laughing_goldstine	"/hello"	4 weeks ago	Exited (0) 4 weeks ago	

Проводим проверку наличия логов в контейнере, предварительно войдя в контейнер.



```

mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_rocket$ sudo docker exec -it business_case_rocket-scheduler-1 /bin
/bash
airflow@208d2713f168:/opt/airflow$ ls
airflow.cfg  dags  logs  unittests.cfg  webserver_config.py
airflow@208d2713f168:/opt/airflow$ cd logs
airflow@208d2713f168:/opt/airflow/logs$ ls
dag_processor_manager  download_rocket_launches  scheduler
airflow@208d2713f168:/opt/airflow/logs$ cd download_rocket_launches
airflow@208d2713f168:/opt/airflow/logs/download_rocket_launches$ ls
download_launches  get_pictures  notify
airflow@208d2713f168:/opt/airflow/logs/download_rocket_launches$ cd notify
airflow@208d2713f168:/opt/airflow/logs/download_rocket_launches/notify$ ls
2024-03-21T00:00:00+00:00  2024-03-25T00:00:00+00:00  2024-03-30T00:00:00+00:00  2024-04-03T00:00:00+00:00
2024-03-22T00:00:00+00:00  2024-03-27T00:00:00+00:00  2024-03-31T00:00:00+00:00
2024-03-23T00:00:00+00:00  2024-03-28T00:00:00+00:00  2024-04-01T00:00:00+00:00
2024-03-24T00:00:00+00:00  2024-03-29T00:00:00+00:00  2024-04-02T00:00:00+00:00
airflow@208d2713f168:/opt/airflow/logs/download_rocket_launches/notify$ cd 2024-04-03T00:00:00+00:00
airflow@208d2713f168:/opt/airflow/logs/download_rocket_launches/notify/2024-04-03T00:00:00+00:00$ ls
1.log
airflow@208d2713f168:/opt/airflow/logs/download_rocket_launches/notify/2024-04-03T00:00:00+00:00$

```

Далее происходит выгрузка логов из контейнера в основную ОС

```

mgpu@mgpu-VirtualBox:~$ sudo docker cp 208:/opt/airflow/logs/download_rocket_launches/notify/2024-04-03T00:00:00+00:00/
1.log Downloads/logs_alena.log
Successfully copied 4.61kB to /home/mgpu/Downloads/logs_alena.log
mgpu@mgpu-VirtualBox:~$

```

```

mgpu@mgpu-VirtualBox:~$ sudo docker cp 208:/opt/airflow/logs/download_rocket_launches/download_launches/2024-04-03T00:00:00+00:00/1.log Downloads/logs_alena1.log
[sudo] password for mgpu:
Successfully copied 5.12kB to /home/mgpu/Downloads/logs_alena1.log
mgpu@mgpu-VirtualBox:~$

```

```

mgpu@mgpu-VirtualBox:~$ sudo docker cp 208:/opt/airflow/logs/download_rocket_launches/get_pictures/2024-04-03T00:00:00+00:00/1.log Downloads/logs_alena2.log
Successfully copied 6.66kB to /home/mgpu/Downloads/logs_alena2.log
mgpu@mgpu-VirtualBox:~$

```

После выгрузки проверяются в каталоге основной ОС файлы логов, которые были сохранены.

1. Файл notify
2. Файл download
3. Файл pictures

Open

logs\_alena.log  
~/Downloads

Save

```
1 [2024-04-04 11:12:15,927] {taskinstance.py:826} INFO - Dependencies all met
  for <TaskInstance: download_rocket_launches.notify
    2024-04-03T00:00:00+00:00 [queued]>
2 [2024-04-04 11:12:15,944] {taskinstance.py:826} INFO - Dependencies all met
  for <TaskInstance: download_rocket_launches.notify
    2024-04-03T00:00:00+00:00 [queued]>
3 [2024-04-04 11:12:15,944] {taskinstance.py:1017} INFO -
4 -----
5 [2024-04-04 11:12:15,944] {taskinstance.py:1018} INFO - Starting attempt 1
  of 1
6 [2024-04-04 11:12:15,944] {taskinstance.py:1019} INFO -
7 -----
8 [2024-04-04 11:12:15,955] {taskinstance.py:1038} INFO - Executing
  <Task(BashOperator): notify> on 2024-04-03T00:00:00+00:00
9 [2024-04-04 11:12:15,957] {standard_task_runner.py:51} INFO - Started
  process 11540 to run task
10 [2024-04-04 11:12:15,964] {standard_task_runner.py:75} INFO - Running:
    ['airflow', 'tasks', 'run', 'download_rocket_launches', 'notify',
     '2024-04-03T00:00:00+00:00', '--job-id', '42', '--pool', 'default_pool', '--
     raw', '--subdir', 'DAGS_FOLDER/download_rocket_launches.py', '--cfg-path',
     '/tmp/tmpej37kmm4']
11 [2024-04-04 11:12:15,966] {standard_task_runner.py:76} INFO - Job 42:
    Subtask notify
12 [2024-04-04 11:12:16,028] {logging_mixin.py:103} INFO - Running
  <TaskInstance: download_rocket_launches.notify 2024-04-03T00:00:00+00:00
    [running]> on host 208d2713f168
13 [2024-04-04 11:12:16,093] {taskinstance.py:1230} INFO - Exporting the
```

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS

Open

logs\_alena1.log  
~/Downloads

Save

```
1 [2024-04-04 11:12:01,259] {taskinstance.py:826} INFO - Dependencies all met
  for <TaskInstance: download_rocket_launches.download_launches
    2024-04-03T00:00:00+00:00 [queued]>
2 [2024-04-04 11:12:01,306] {taskinstance.py:826} INFO - Dependencies all met
  for <TaskInstance: download_rocket_launches.download_launches
    2024-04-03T00:00:00+00:00 [queued]>
3 [2024-04-04 11:12:01,306] {taskinstance.py:1017} INFO -
4 -----
5 [2024-04-04 11:12:01,306] {taskinstance.py:1018} INFO - Starting attempt 1
  of 1
6 [2024-04-04 11:12:01,306] {taskinstance.py:1019} INFO -
7 -----
8 [2024-04-04 11:12:01,333] {taskinstance.py:1038} INFO - Executing
  <Task(BashOperator): download_launches> on 2024-04-03T00:00:00+00:00
9 [2024-04-04 11:12:01,335] {standard_task_runner.py:51} INFO - Started
  process 10686 to run task
10 [2024-04-04 11:12:01,364] {standard_task_runner.py:75} INFO - Running:
    ['airflow', 'tasks', 'run', 'download_rocket_launches',
     'download_launches', '2024-04-03T00:00:00+00:00', '--job-id', '26', '--
     pool', 'default_pool', '--raw', '--subdir', 'DAGS_FOLDER/-
     download_rocket_launches.py', '--cfg-path', '/tmp/tmpw3c6i2ap']
11 [2024-04-04 11:12:01,364] {standard_task_runner.py:76} INFO - Job 26:
    Subtask download_launches
12 [2024-04-04 11:12:01,522] {logging_mixin.py:103} INFO - Running
  <TaskInstance: download_rocket_launches.download_launches
    2024-04-03T00:00:00+00:00 [running]> on host 208d2713f168
13 [2024-04-04 11:12:01,815] {taskinstance.py:1230} INFO - Exporting the
```

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS



```
logs_alena2.log
~/Downloads

1 [2024-04-04 11:12:04,287] {taskinstance.py:826} INFO - Dependencies all met
  for <TaskInstance: download_rocket_launches.get_pictures
  2024-04-03T00:00:00+00:00 [queued]>
2 [2024-04-04 11:12:04,316] {taskinstance.py:826} INFO - Dependencies all met
  for <TaskInstance: download_rocket_launches.get_pictures
  2024-04-03T00:00:00+00:00 [queued]>
3 [2024-04-04 11:12:04,316] {taskinstance.py:1017} INFO -
4 -----
5 [2024-04-04 11:12:04,316] {taskinstance.py:1018} INFO - Starting attempt 1
  of 1
6 [2024-04-04 11:12:04,316] {taskinstance.py:1019} INFO -
7 -----
8 [2024-04-04 11:12:04,327] {taskinstance.py:1038} INFO - Executing
  <Task(PythonOperator): get_pictures> on 2024-04-03T00:00:00+00:00
9 [2024-04-04 11:12:04,330] {standard_task_runner.py:51} INFO - Started
  process 10846 to run task
10 [2024-04-04 11:12:04,336] {standard_task_runner.py:75} INFO - Running:
  ['airflow', 'tasks', 'run', 'download_rocket_launches', 'get_pictures',
  '2024-04-03T00:00:00+00:00', '--job-id', '29', '--pool', 'default_pool', '--
  raw', '--subdir', 'DAGS_FOLDER/download_rocket_launches.py', '--cfg-path',
  '/tmp/tmpm3st7sx5']
11 [2024-04-04 11:12:04,337] {standard_task_runner.py:76} INFO - Job 29:
  Subtask get_pictures
12 [2024-04-04 11:12:04,431] {logging_mixin.py:103} INFO - Running
  <TaskInstance: download_rocket_launches.get_pictures
  2024-04-03T00:00:00+00:00 [running]> on host 208d2713f168
13 [2024-04-04 11:12:04,558] {taskinstance.py:1230} INFO - Exporting the

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS
```

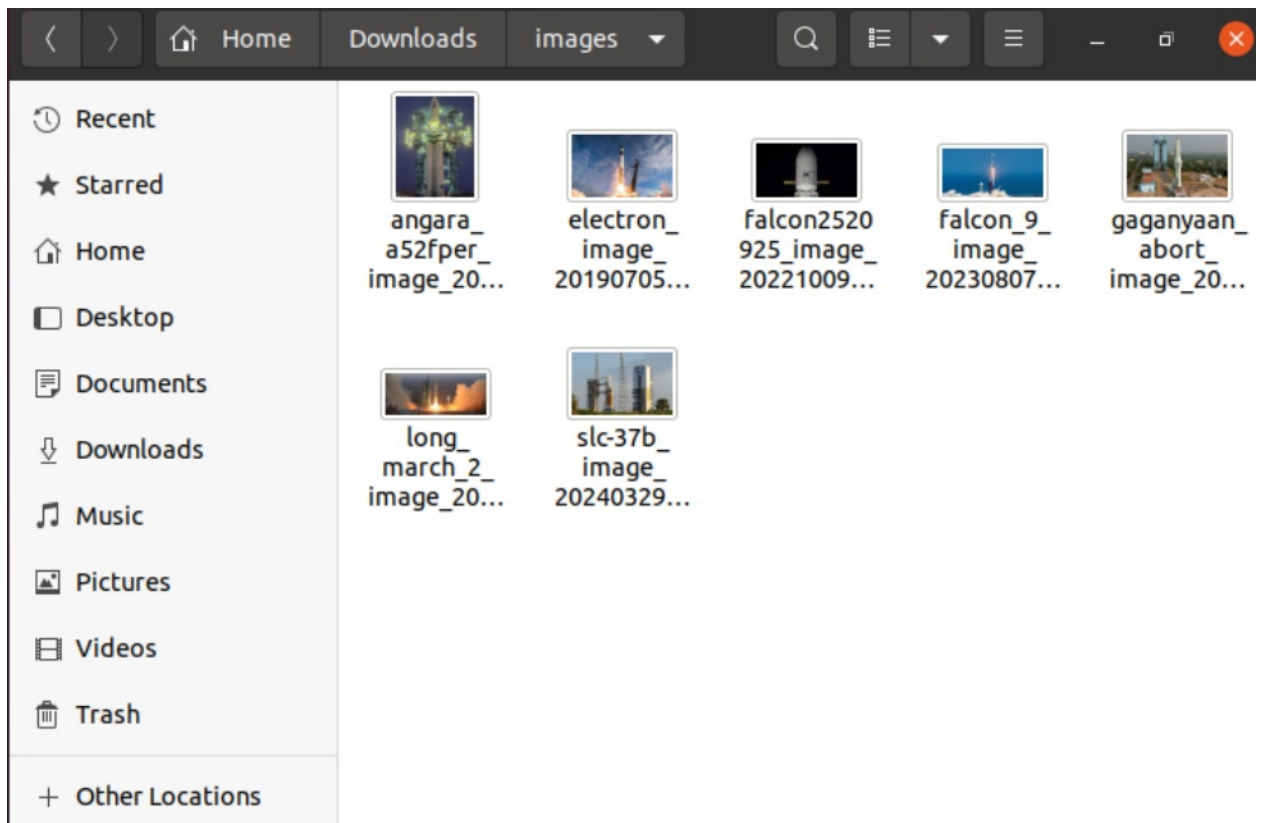
Также необходимо выгрузить полученный результат работы DAG в основной каталог ОС. Для начала проверяем наличие результатов в контейнере.

```
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_rocket$ sudo docker exec -it business_case_rocket-scheduler-1 /bin
/bash
airflow@208d2713f168:/opt/airflow$ cd /tmp/images
airflow@208d2713f168:/tmp/images$ ls
angara_a52fper_image_20211227193648.jpg      falcon_9_image_20230807133459.jpeg          slc-37b_image_20240329083424.jpeg
electron_image_20190705175640.jpeg          gaganyaan_abort_image_20231021132156.jpeg
falcon2520925_image_20221009234147.png      long_march_2_image_20210908195835.jpeg
airflow@208d2713f168:/tmp/images$
```

Далее необходимо скопировать все файлы в основную ОС.

```
mgpu@mgpu-VirtualBox:~$ sudo docker cp 208:/tmp/images Downloads/images
Successfully copied 2.02MB to /home/mgpu/Downloads/images
mgpu@mgpu-VirtualBox:~$
```

После этого необходимо проверить их в каталоге основной ОС.



Создадим скрипт исполняемого файла с расширением .sh, который автоматизирует выгрузку данных из контейнера в основную ОС данных, полученные в результате работы DAG в Apache Airflow.

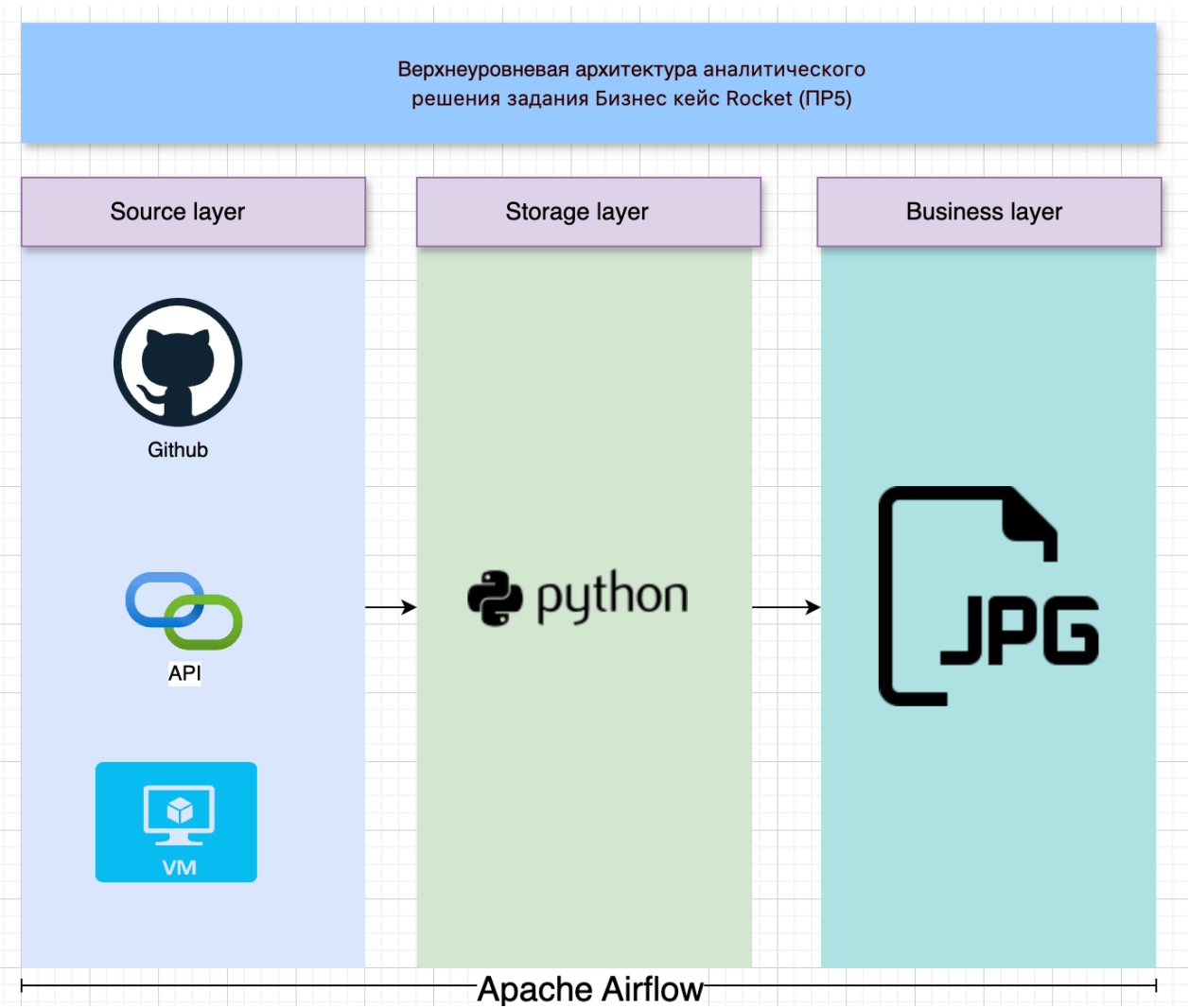
```
GNU nano 4.8                               alena.sh
#!/bin/bash

CONTAINER_ID=$(sudo docker ps --filter "name=business_case_rocket-scheduler-1" -q)
sudo docker cp --archive $CONTAINER_ID:/tmp/images /home/mgpu/Downloads/imaging
```

По итогам запуска исполняемого файла alena.sh выгрузка данных, полученных в результате работы DAG, из контейнера в основную ОС выполнена успешно.

```
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_rocket$ nano alena.sh
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_rocket$ sh alena.sh
Successfully copied 2.02MB to /home/mgpu/Downloads/imaging
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_rocket$
```

Верхнеуровневая архитектура аналитического решения. Бизнес-кейса «Rocket»



## Архитектура DAG Бизнес-кейса «Rocket».

