

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: Алексейчук А.А., АДЭУ-201

Преподаватель: Босенко Т.М.

Москва

2024

Оглавление

Постановка задачи.....	3
Исходный код всех dags	5
Граф DAG в Apache Airflow.....	6
Верхнеуровневая архитектура задания Бизнес-кейса «Rocket», выполненная в draw.io	7
Архитектура DAG Бизнес-кейса «Rocket» , выполненная в draw.io	8
Скрин лог-файла результаов работы dags в Apache Airflow	8
Диаграмма Ганта DAG в Apache Airflow.....	12
Заключение	14

Постановка задачи.

Самостоятельная работа

5.1.1. Развернуть VM ubuntu_mgpu.ova в VirtualBox.

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

```
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.

Проверка ответа URL-адреса с помощью Curl из командной строки (Рисунок 1). Значения в фигурных скобках относятся к одному запуску ракеты. Информация об идентификаторе ракеты, времени начала и окончания окна запуска ракеты, URL-адресе изображения запускаемой ракеты.

```
hops@hops-VirtualBox:~$ curl -L "https://ll.thespacedevs.com/2.0.0/launch/upcoming"
{"count":362,"next":"https://ll.thespacedevs.com/2.0.0/launch/upcoming/?limit=10&offset=10","previous":null,"results":[{"id":"827fac66-147f-4afc-9489-1b1d736ee989","url":"https://ll.thespacedevs.com/2.0.0/launch/827fac66-147f-4afc-9489-1b1d736ee989/","launch_library_id":null,"slug":"falcon-9-block-5-starlink-group-6-44","name":"Falcon 9 Block 5 | Starlink Group 6-44","status":{"id":3,"name":"Success"},"net":"2024-03-16T08:21:00Z","window_end":"2024-03-16T02:39:00Z","window_start":"2024-03-15T22:39:00Z","inhold":false,"tbdline":false,"tbddate":false,"probability":null,"holdreason":"","failreason":"","hsttag":null,"launch_service_provider":{"id":121,"url":"https://ll.thespacedevs.com/2.0.0/agencies/121/","name":"SpaceX","type":"Commercial"},"rocket":{"id":8189,"configuration":{"id":264,"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":6758,"launch_library_id":null,"name":"Starlink Group 6-44","description":"A batch of 23 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":87,"url":"https://ll.thespacedevs.com/2.0.0/pad/87/","agency_id":121,"name":"Launch Complex 39A","info_url":null,"wiki_url":"https://en.wikipedia.org/wiki/Kennedy_Space_Center_Launch_Complex_39A","map_url":"https://www.google.com/maps?q=-28.60822681,-80.60428186","latitude":28.60822681,"longitude":-80.60428186},"location":{"id":27,"url":"https://ll.thespacedevs.com/2.0.0/location/27/","name":"Kennedy Space Center, FL, USA","country_code":"USA","map_image":"https://spacelaunchnow-prod-east.nyc3.digitaloceanspaces.com/media/launch_images/location_27_20200803142447.jpg","total_launch_count":231,"total_landing_count":0},"map_image":"https://spacelaunchnow-prod-east.nyc3.digitaloceanspaces.com/media/launch_images/pad_87_20200803143537.jpg","total_launch_count":173},"webcast_live":false,"image":"https://spacelaunchnow-prod-east.nyc3.digitaloceanspaces.com/media/images/falcon2520925_image_2022100923417.png","infographic":null,"program":{"id":25,"url":"https://ll.thespacedevs.com/2.0.0/program/25/","name":"Starlink","description":"Starlink is a satellite internet constellation operated by American aerospace company SpaceX","agencies":[{"id":121,"url":"https://ll.thespacedevs.com/2.0.0/agencies/121/","name":"SpaceX","type":"Commercial"},"image_url":"https://spacelaunchnow-prod-east.nyc3.digitaloceanspaces.com/media/images/starlink_program_20231228154508.jpeg","start_date":"2018-02-22T14:17:00Z","end_date":null,"info_url":"https://starlink.com","wiki_url":"https://en.wikipedia.org/wiki/Starlink"}]}{"id":"2323baef-19f9-45fc-bc3d-3d34f7021f82","url":"https://ll.thespacedevs.com/2.0.0/launch/2323baef-19f9-45fc-bc3d-3d34f7021f82/","launch_library_id":null,"slug":"falcon-9-block-5-starlink-group-7-16"}
```

Рисунок 1- Проверка ответа URL

Исходный код всех dags

download_rocket_launches.py (Рисунок 2).

```
download_rocket_launches.py 4 X
C: > Users > Nastya > Downloads > download_rocket_launches.py > ...
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://1l.thespacedevs.com/2.0.0/launch/upcoming'", # noqa: E501
21     dag=dag,
22 )
23
24
25 def _get_pictures():
26     # Ensure directory exists
27     pathlib.Path("/tmp/images").mkdir(parents=True, exist_ok=True)
28
29     # Download all pictures in launches.json
30     with open("/tmp/launches.json") as f:
31         launches = json.load(f)
32         image_urls = [launch["image"] for launch in launches["results"]]
33         for image_url in image_urls:
34             try:
35                 response = requests.get(image_url)
36                 image_filename = image_url.split("/")[-1]
37                 target_file = f"/tmp/images/{image_filename}"
38                 with open(target_file, "wb") as f:
39                     f.write(response.content)
40                 print(f"Downloaded {image_url} to {target_file}")
41             except requests_exceptions.MissingSchema:
42                 print(f"{image_url} appears to be an invalid URL.")
43             except requests_exceptions.ConnectionError:
44                 print(f"Could not connect to {image_url}.")
45
46
47 get_pictures = PythonOperator(
48     task_id="get_pictures", python_callable=_get_pictures, dag=dag
49 )
50
51 notify = BashOperator(
52     task_id="notify",
53     bash_command='echo "There are now $(ls /tmp/images/ | wc -l) images."',
54     dag=dag,
55 )
56
57 download_launches >> get_pictures >> notify
58
```

Рисунок 2 - Исходный код

Граф DAG в Apache Airflow

После запуска образа на странице localhost:8080 запускаем DAG(Рисунок 3).

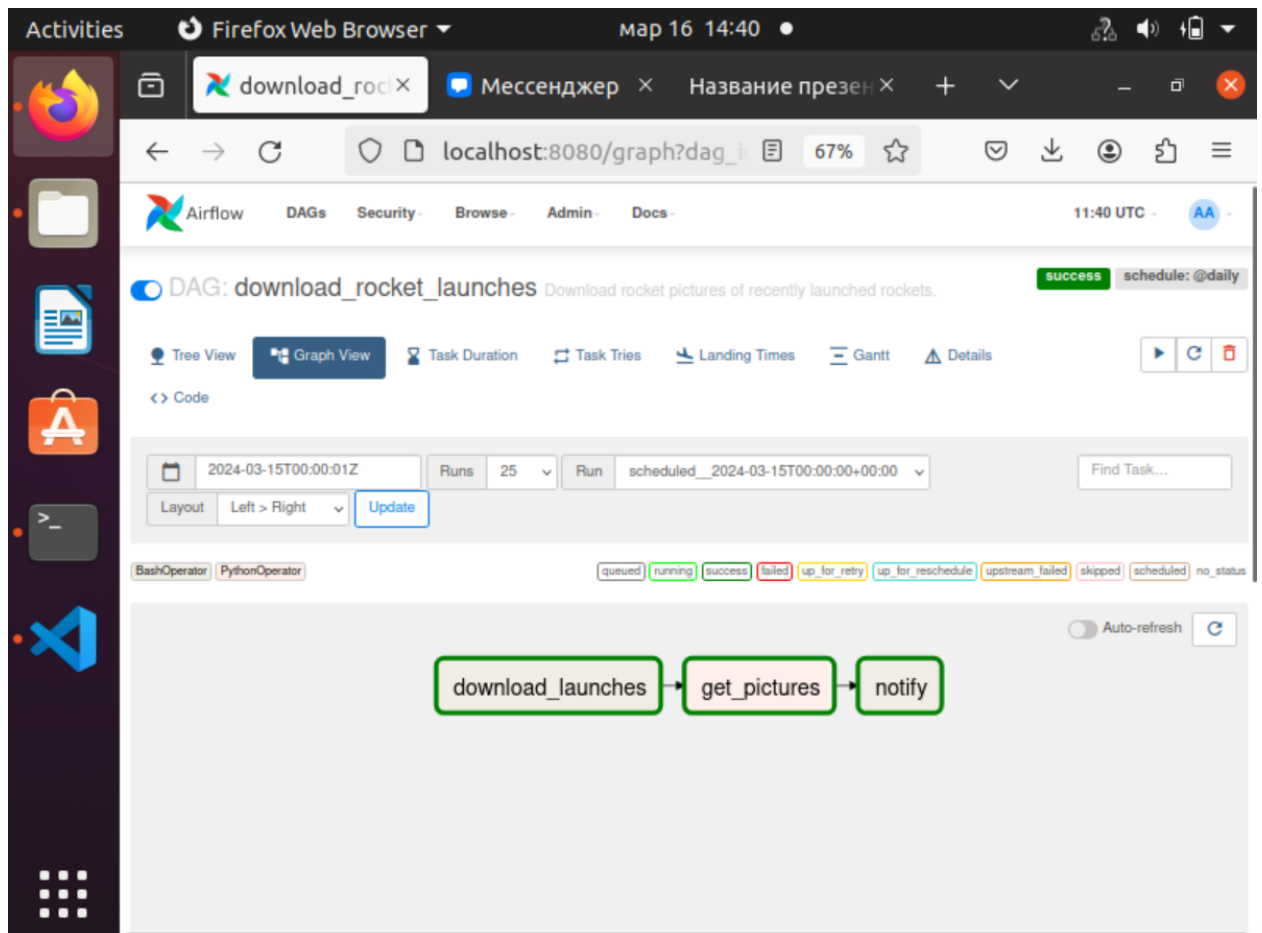


Рисунок 3 - Граф DAG

**Верхнеуровневая архитектура задания Бизнес-кейса «Rocket»,
выполненная в draw.io**

<https://drive.google.com/file/d/12RS59WPzPVeUAYMhRs-eN78pJXu8fEsH/view?usp=sharing> (Рисунок 4).

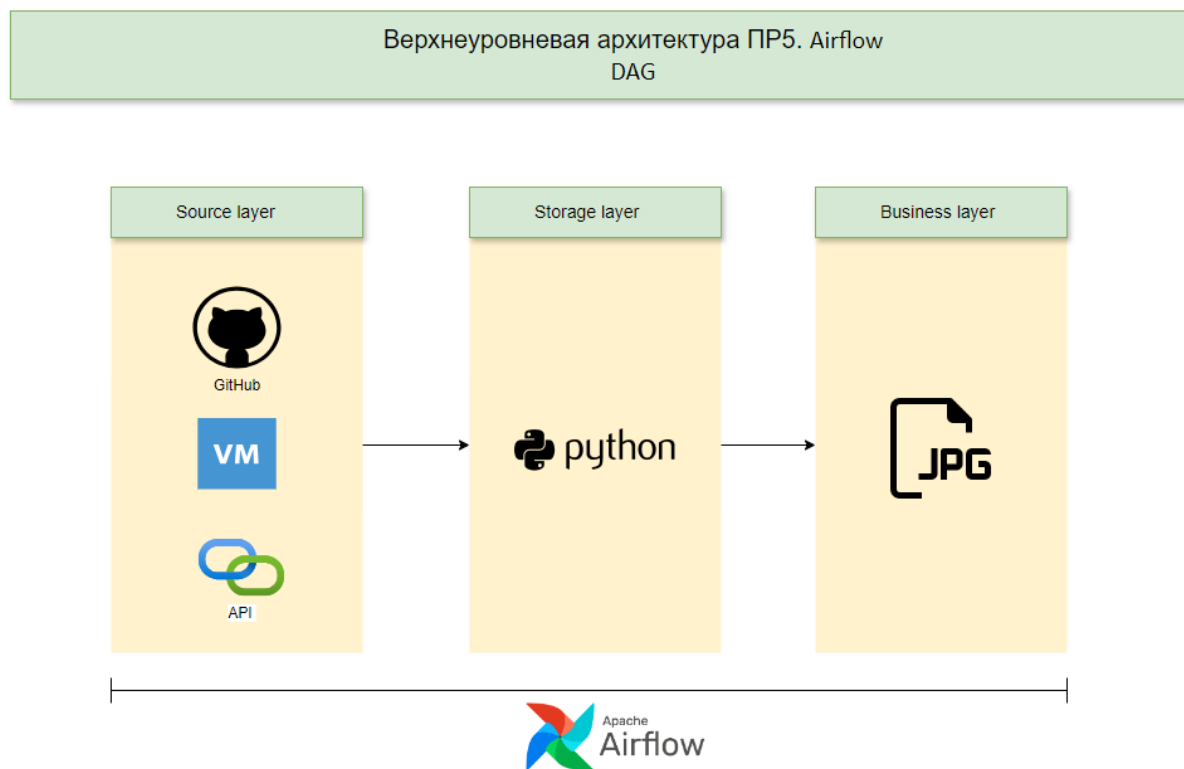


Рисунок 4 - Верхнеуровневая архитектура

Архитектура DAG Бизнес-кейса «Rocket» , выполненная в draw.io

<https://drive.google.com/file/d/1hkI594CqEVUDXKf9RxkcCDh2Mx2Hkz9s/view?usp=sharing>

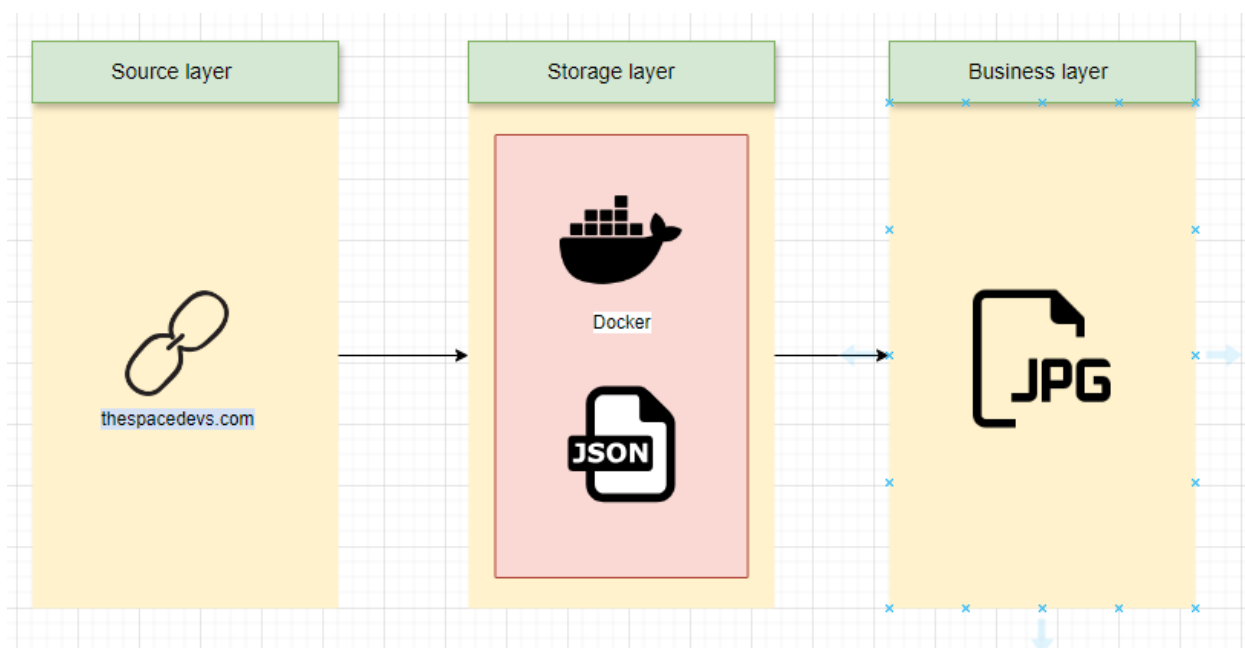


Рисунок 5 - Архитектура DAG

Скрин лог-файла результаов работы dags в Apache Airflow

Проверка файлов логов.

Проверка файла лога download_launches_logs (Рисунок 6).


```
logs.log
logs.log
1 [2024-03-16 10:39:52,561] {taskinstance.py:826} INFO - Dependencies all met for <TaskInstance: download_rocket
2 [2024-03-16 10:39:52,673] {taskinstance.py:826} INFO - Dependencies all met for <TaskInstance: download_rocket
3 [2024-03-16 10:39:52,673] {taskinstance.py:1017} INFO -
4 -----
5 [2024-03-16 10:39:52,673] {taskinstance.py:1018} INFO - Starting attempt 1 of 1
6 [2024-03-16 10:39:52,673] {taskinstance.py:1019} INFO -
7 -----
8 [2024-03-16 10:39:52,763] {taskinstance.py:1038} INFO - Executing <Task(BashOperator): download_launches> on 2
9 [2024-03-16 10:39:52,774] {standard_task_runner.py:51} INFO - Started process 20891 to run task
10 [2024-03-16 10:39:52,802] {standard_task_runner.py:75} INFO - Running: ['airflow', 'tasks', 'run', 'download_r
11 [2024-03-16 10:39:52,803] {standard_task_runner.py:76} INFO - Job 5: Subtask download_launches
12 [2024-03-16 10:39:53,102] {logging_mixin.py:103} INFO - Running <TaskInstance: download_rocket_launches.downlc
13 [2024-03-16 10:39:53,566] {taskinstance.py:1230} INFO - Exporting the following env vars:
14 AIRFLOW_CTX_DAG_OWNER=airflow
15 AIRFLOW_CTX_DAG_ID=download_rocket_launches
16 AIRFLOW_CTX_TASK_ID=download_launches
17 AIRFLOW_CTX_EXECUTION_DATE=2024-03-02T00:00:00:00
18 AIRFLOW_CTX_DAG_RUN_ID=scheduled_2024-03-02T00:00:00:00
19 [2024-03-16 10:39:53,567] {bash.py:135} INFO - Tmp dir root location:
20 /tmp
21 [2024-03-16 10:39:53,568] {bash.py:158} INFO - Running command: curl -o /tmp/launches.json -L 'https://11.thes
22 [2024-03-16 10:39:53,659] {bash.py:169} INFO - Output:
23 [2024-03-16 10:39:54,009] {bash.py:173} INFO - % Total % Received % Xferd Average Speed Time Time
24 [2024-03-16 10:39:54,022] {bash.py:173} INFO - Dload Upload Total Spent
25 [2024-03-16 10:39:55,579] {bash.py:173} INFO -
26 0 0 0 0 0 0 0 0 0 0:00:00 0:00:00 0:00:00 0
27 0 0 0 0 0 0 0 0 0 0:00:00 0:00:00 0:00:00 0
28 0 0 0 0 0 0 0 0 0 0:00:01 0:00:01 0:00:01 0
29 0 0 0 0 0 0 0 0 0 0:00:01 0:00:01 0:00:01 0
30 [2024-03-16 10:39:56,346] {bash.py:173} INFO -
31 65 27139 65 17744 0 0 7641 0 0:00:03 0:00:02 0:00:01 7641
32 100 27139 100 27139 0 0 11617 0 0:00:02 0:00:02 0:00:00 705k
33 [2024-03-16 10:39:56,349] {bash.py:177} INFO - Command exited with return code 0
34 [2024-03-16 10:39:56,836] {taskinstance.py:1135} INFO - Marking task as SUCCESS. dag_id=download_rocket_launch
35 [2024-03-16 10:39:57,070] {taskinstance.py:1195} INFO - 1 downstream tasks scheduled from follow-on schedule c
36 [2024-03-16 10:39:57,083] {local_task_job.py:118} INFO - Task exited with return code 0
37
```

Рисунок 6 - download_launches_logs

Проверка файла лога get_pictures_logs (Рисунок 7).

```
logs1.log X
logs1.log
1 [2024-03-16 10:40:03,677] {taskinstance.py:826} INFO - Dependencies all met for <TaskInstance: download_rock
2 [2024-03-16 10:40:04,058] {taskinstance.py:826} INFO - Dependencies all met for <TaskInstance: download_rock
3 [2024-03-16 10:40:04,058] {taskinstance.py:1017} INFO -
4 -----
5 [2024-03-16 10:40:04,058] {taskinstance.py:1018} INFO - Starting attempt 1 of 1
6 [2024-03-16 10:40:04,059] {taskinstance.py:1019} INFO -
7 -----
8 [2024-03-16 10:40:04,493] {taskinstance.py:1038} INFO - Executing <Task(PythonOperator): get_pictures> on 20
9 [2024-03-16 10:40:04,514] {standard_task_runner.py:51} INFO - Started process 21019 to run task
10 [2024-03-16 10:40:04,532] {standard_task_runner.py:75} INFO - Running: ['airflow', 'tasks', 'run', 'download
11 [2024-03-16 10:40:04,634] {standard_task_runner.py:76} INFO - Job 17: Subtask get_pictures
12 [2024-03-16 10:40:06,110] {logging_mixin.py:103} INFO - Running <TaskInstance: download_rocket_launches.get_
13 [2024-03-16 10:40:07,355] {taskinstance.py:1230} INFO - Exporting the following env vars:
14 AIRFLOW_CTX_DAG_OWNER=airflow
15 AIRFLOW_CTX_DAG_ID=download_rocket_launches
16 AIRFLOW_CTX_TASK_ID=get_pictures
17 AIRFLOW_CTX_EXECUTION_DATE=2024-03-02T00:00:00+00:00
18 AIRFLOW_CTX_DAG_RUN_ID=scheduled__2024-03-02T00:00:00+00:00
19 [2024-03-16 10:40:08,732] {logging_mixin.py:103} INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.dig
20 [2024-03-16 10:40:10,007] {logging_mixin.py:103} INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.dig
21 [2024-03-16 10:40:11,253] {logging_mixin.py:103} INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.dig
22 [2024-03-16 10:40:12,238] {logging_mixin.py:103} INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.dig
23 [2024-03-16 10:40:13,913] {logging_mixin.py:103} INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.dig
24 [2024-03-16 10:40:14,875] {logging_mixin.py:103} INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.dig
25 [2024-03-16 10:40:16,186] {logging_mixin.py:103} INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.dig
26 [2024-03-16 10:40:17,428] {logging_mixin.py:103} INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.dig
27 [2024-03-16 10:40:18,827] {logging_mixin.py:103} INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.dig
28 [2024-03-16 10:40:19,838] {logging_mixin.py:103} INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.dig
29 [2024-03-16 10:40:19,838] {python.py:118} INFO - Done. Returned value was: None
30 [2024-03-16 10:40:19,908] {taskinstance.py:1135} INFO - Marking task as SUCCESS. dag_id=download_rocket_laun
31 [2024-03-16 10:40:21,303] {local_task_job.py:169} WARNING - State of this instance has been externally set t
32 [2024-03-16 10:40:21,318] {process_utils.py:95} INFO - Sending Signals.SIGTERM to GPID 21019
33 [2024-03-16 10:40:23,440] {taskinstance.py:1214} ERROR - Received SIGTERM. Terminating subprocesses.
34 [2024-03-16 10:40:23,503] {process_utils.py:61} INFO - Process psutil.Process(pid=21019, status='terminated'
35 [2024-03-16 10:40:23,504] {local_task_job.py:118} INFO - Task exited with return code 1
36
```

Рисунок 7 - get_pictures_logs

Проверка файла лога notify_logs (Рисунок 8).

```
logs2.log x
logs2.log
1 [2024-03-16 10:40:29,290] {taskinstance.py:826} INFO - Dependencies all met for <TaskInstance: download_rocket
2 [2024-03-16 10:40:29,678] {taskinstance.py:826} INFO - Dependencies all met for <TaskInstance: download_rocket
3 [2024-03-16 10:40:29,678] {taskinstance.py:1017} INFO -
4 -----
5 [2024-03-16 10:40:29,678] {taskinstance.py:1018} INFO - Starting attempt 1 of 1
6 [2024-03-16 10:40:29,679] {taskinstance.py:1019} INFO -
7 -----
8 [2024-03-16 10:40:30,193] {taskinstance.py:1038} INFO - Executing <Task(BashOperator): notify> on 2024-03-02T00:00:00+00:00
9 [2024-03-16 10:40:30,204] {standard_task_runner.py:51} INFO - Started process 21188 to run task
10 [2024-03-16 10:40:30,371] {standard_task_runner.py:75} INFO - Running: ['airflow', 'tasks', 'run', 'download_rocket_launches', 'notify']
11 [2024-03-16 10:40:30,372] {standard_task_runner.py:76} INFO - Job 31: Subtask notify
12 [2024-03-16 10:40:31,579] {logging_mixin.py:103} INFO - Running <TaskInstance: download_rocket_launches.notify>
13 [2024-03-16 10:40:32,302] {taskinstance.py:1230} INFO - Exporting the following env vars:
14 AIRFLOW_CTX_DAG_OWNER=airflow
15 AIRFLOW_CTX_DAG_ID=download_rocket_launches
16 AIRFLOW_CTX_TASK_ID=notify
17 AIRFLOW_CTX_EXECUTION_DATE=2024-03-02T00:00:00+00:00
18 AIRFLOW_CTX_DAG_RUN_ID=scheduled_2024-03-02T00:00:00+00:00
19 [2024-03-16 10:40:32,362] {bash.py:135} INFO - Tmp dir root location:
20 /tmp
21 [2024-03-16 10:40:32,362] {bash.py:158} INFO - Running command: echo "There are now $(ls /tmp/images/ | wc -l) images."
22 [2024-03-16 10:40:32,865] {bash.py:169} INFO - Output:
23 [2024-03-16 10:40:32,936] {bash.py:173} INFO - There are now 8 images.
24 [2024-03-16 10:40:32,937] {bash.py:177} INFO - Command exited with return code 0
25 [2024-03-16 10:40:33,390] {taskinstance.py:1135} INFO - Marking task as SUCCESS. dag_id=download_rocket_launches, task_id=notify
26 [2024-03-16 10:40:35,577] {local_task_job.py:169} WARNING - State of this instance has been externally set to SUCCESS
27 [2024-03-16 10:40:35,582] {process_utils.py:95} INFO - Sending Signals.SIGTERM to GPID 21188
28 [2024-03-16 10:40:35,665] {taskinstance.py:1214} ERROR - Received SIGTERM. Terminating subprocesses.
29 [2024-03-16 10:40:35,665] {bash.py:185} INFO - Sending SIGTERM signal to bash process group
30 [2024-03-16 10:40:35,928] {process_utils.py:61} INFO - Process psutil.Process(pid=21188, status='terminated', ...)
31 [2024-03-16 10:40:35,930] {local_task_job.py:118} INFO - Task exited with return code 1
```

Рисунок 8 - notify_logs

Выгрузка папки с изображениями из Apache Airflow (Рисунок 9).

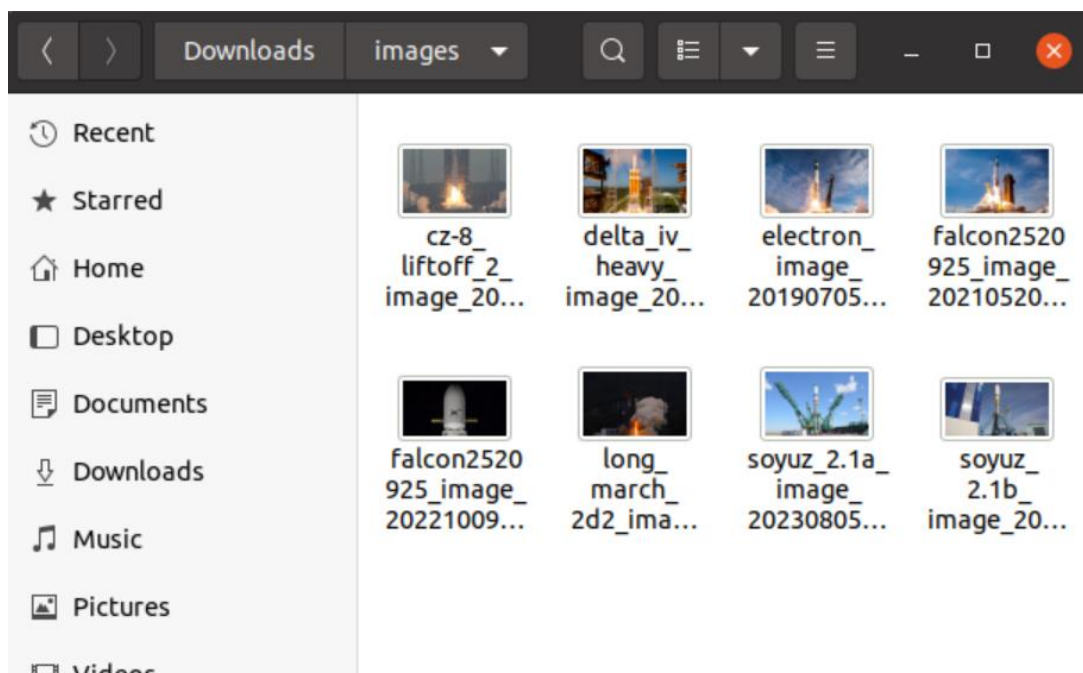


Рисунок 9 - Выгрузка картинок

Диаграмма Ганта DAG в Apache Airflow

Диаграмма Ганта DAG (Рисунок 10).

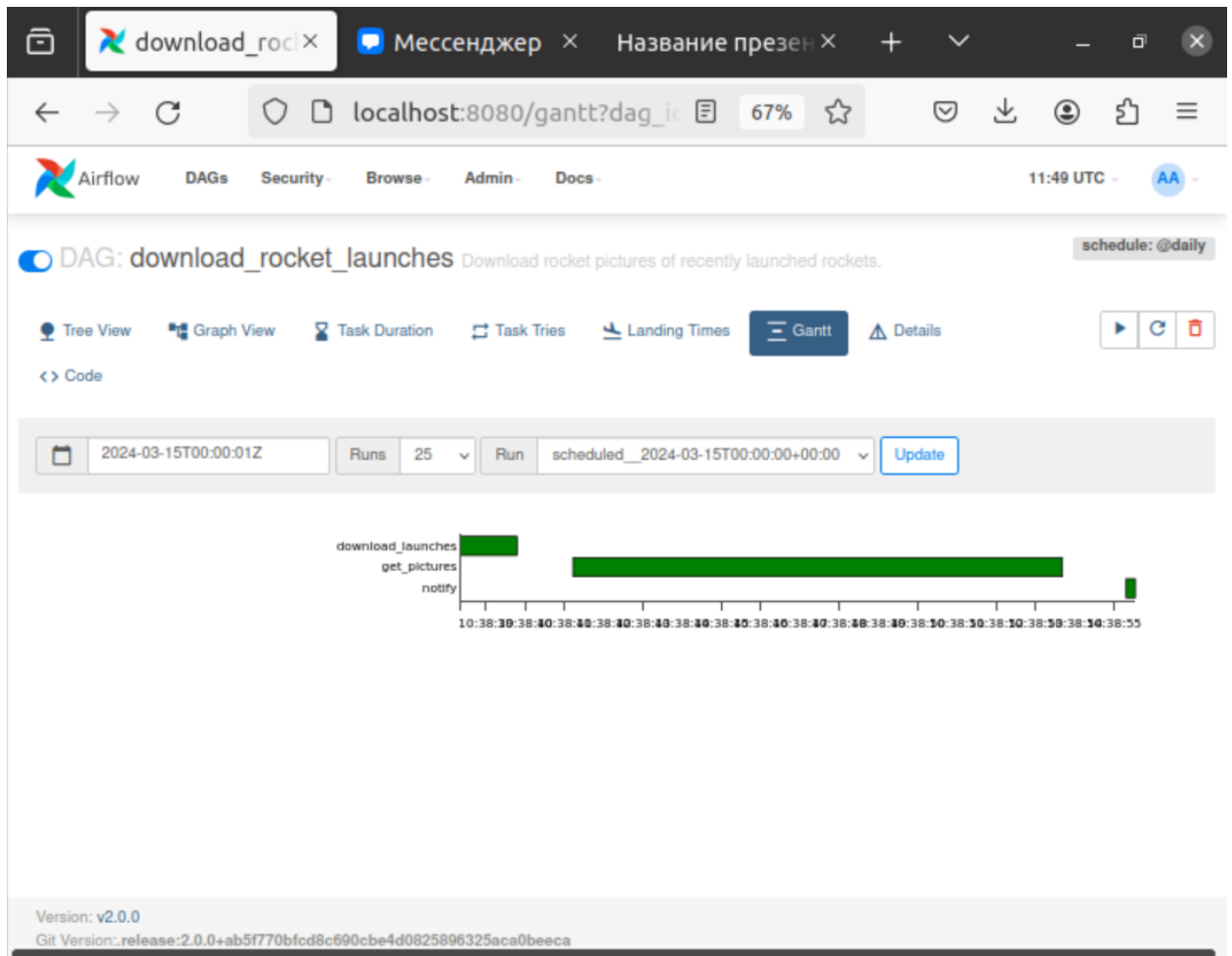


Рисунок 10 - Диаграмма Ганта

Исполняемый файл 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 export_data.sh
export_dir=/home/mgpu/Downloads
sudo docker cp 8f7:/opt/airflow/logs/download_rocket_launches/notify/2024-03-0
echo "Data exported to $export_dir"
```

Рисунок 11 - Исполняемый файл .sh

```
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_rocket$ touch export_data.sh
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_rocket$ sudo nano export_data.sh
[sudo] password for mgpu:
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_rocket$ chmod +x export_data.sh
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_rocket$ ./export_data.sh
Successfully copied 4.61kB to /home/mgpu/Downloads/logs.log
Data exported to /home/mgpu/Downloads
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_rocket$
```

Рисунок 12 - Успешный запуск файла

Заключение

После выполнения работы было освоено:

1. Запуск контейнера и создание DAG в Apache Airflow:

Были изучены основные элементы и функционал Apache Airflow, создали и запущены Directed Acyclic Graph (DAG), изучили логи выполнения и скачали их для анализа.

2. Проектирование верхнеуровневой архитектуры аналитического решения:

Была спроектирована верхнеуровневая архитектура для аналитического решения задания «Rocket», включая слои источников данных, хранения данных и доступа к данным пользователей.

3. Проектирование архитектуры DAG:

Была разработана архитектура DAG для кейса «Rocket», учитывая слои источников, хранения и бизнес-логики доступа к данным.

4. Построение диаграммы Ганта работы DAG:

Была построена диаграмма Ганта для работы DAG в Apache Airflow, что позволяет визуализировать и планировать последовательность выполнения задач в графическом виде.

В итоге, выполнение приведенных шагов позволило освоить процессы создания и анализа DAG в Apache Airflow, а также разработки архитектуры данных и построения диаграмм работы DAG.