Департамент образования города Москвы

Государственное автономное образовательное учреждение
высшего образования города Москвы
«Московский городской педагогический университет»

Институт цифрового образования
Департамент информатики, управления и технологий

Практическая работа на вебинаре 28.03.2025
по дисциплине «Проектный практикум по разработке ETL-решений»

Направление подготовки 38.03.05 – бизнес-информатика
Профиль подготовки «Аналитика данных и эффективное управление»
(очная форма обучения)

Выполнила:
St_88

Москва
2025

ВАРИАНТ 1

Вариант	Задание 1	Задание 2	Задание 3
1	Провести анализ данных о стартах ракет из JSON-файла и составить отчет о количестве запусков за последний месяц	Разработать способ обработки исключений для недоступных данных в JSON	Создать отчет по количеству успешно скачанных изображений с URL

ХОД РАБОТЫ

1. В подготовленной виртуальной машине открываем проект «Business_case_rocket_25». Далее в файл дага, планируемого к запуску, внесем изменения в соответствии с вариантом индивидуального задания.

Во-первых, необходимо собрать данные о запусках за последний месяц. Для этого необходимо применить фильтры к API:

```
'https://ll.thespacedevs.com/2.0.0/launch/?net__gte=2025-03-01&net__lte=2025-03-31'"
```

Во-вторых, решением 2 и 3 варианта может являться запись данных о выгруженных картинках в csv-файл “error_urls.csv”. В данном файле предполагается хранение как url, по которым получилось получить изображение, также и тех, по которым возникли ошибки. Таким образом, исключения недоступных данных не будут теряться, и их можно будет отсмотреть вручную при необходимости, а также данный файл csv будет готов для дальнейшей обработки и созданию отчёта по количеству успешно скачанных изображений.

download_rocket_local.py ●

dags > download_rocket_local.py

```python
def _get_pictures():
    # Обеспечиваем существование директории для изображений
    images_dir = "/opt/airflow/data/images2"
    pathlib.Path(images_dir).mkdir(parents=True, exist_ok=True)
    error_file = "/opt/airflow/data/error_urls.csv"

    # Загружаем данные из файла JSON
    with open("/opt/airflow/data/launches.json") as f:
        launches = json.load(f)
        image_urls = [launch["image"] for launch in launches["results"]]

    # Открываем CSV-файл и записываем ошибки
    with open(error_file, "w", newline="", encoding="utf-8") as error_log:
        csv_writer = csv.writer(error_log, delimiter=",")
        csv_writer.writerow(["URL", "Message"])  # Заголовки

        for image_url in image_urls:
            try:
                response = requests.get(image_url)
                response.raise_for_status()  # Проверяем ошибки HTTP

                image_filename = image_url.split("/")[-1]
                target_file = f"{images_dir}/{image_filename}"

                with open(target_file, "wb") as img_file:
                    img_file.write(response.content)

                print(f"Downloaded {image_url} to {target_file}")
                csv_writer.writerow([image_url, "Good URL"])
            except requests_exceptions.MissingSchema:
                csv_writer.writerow([image_url, "Invalid URL"])
                print(f"Invalid URL: {image_url}")
            except requests_exceptions.ConnectionError:
                csv_writer.writerow([image_url, "ConnectionError"])
                print(f"Could not connect to {image_url}")
            except requests_exceptions.HTTPError as e:
                csv_writer.writerow([image_url, f"HTTP Error {e.response.status_code}"])
                print(f"HTTP Error {e.response.status_code} for {image_url}")
```

Далее определяем права доступа к папке data:

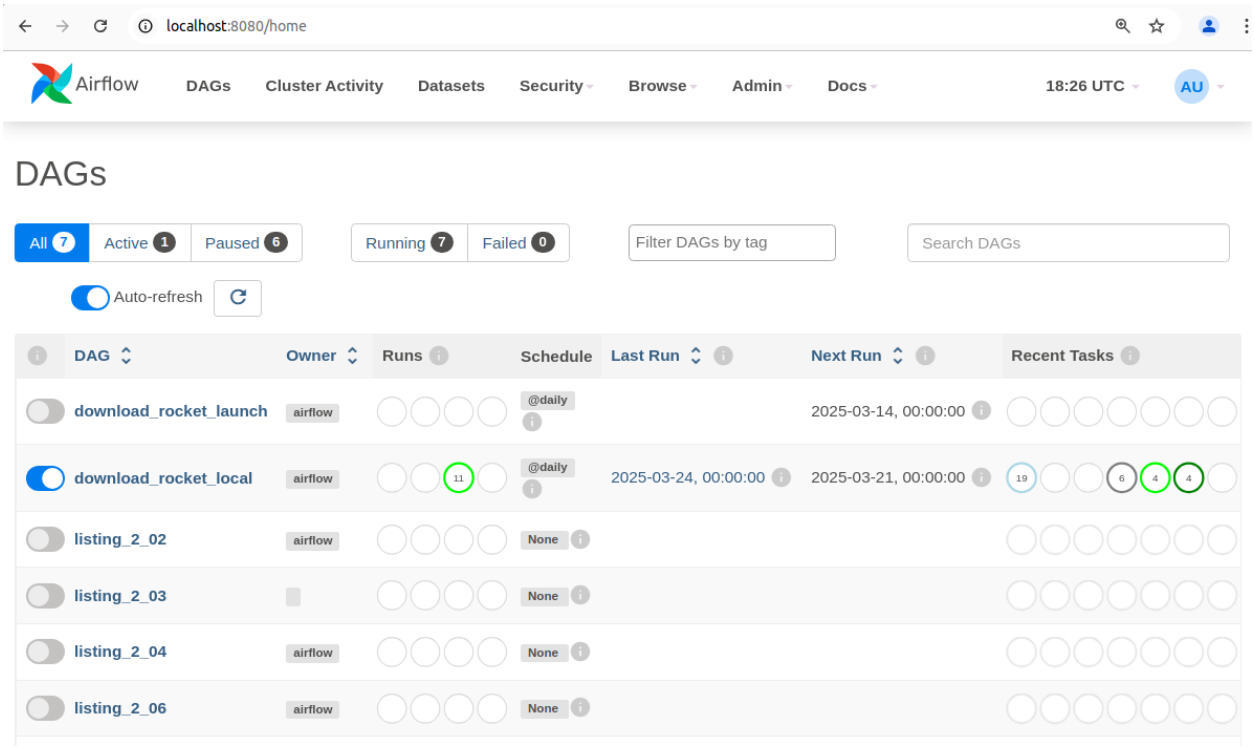
```
dev@dev-vm:~/28_03_rocket/business_case_rocket_25$ sudo chown -R 50000:50000 ./data
[sudo] password for dev:
dev@dev-vm:~/28_03_rocket/business_case_rocket_25$
```

2. Собираем и запускаем контейнеры:

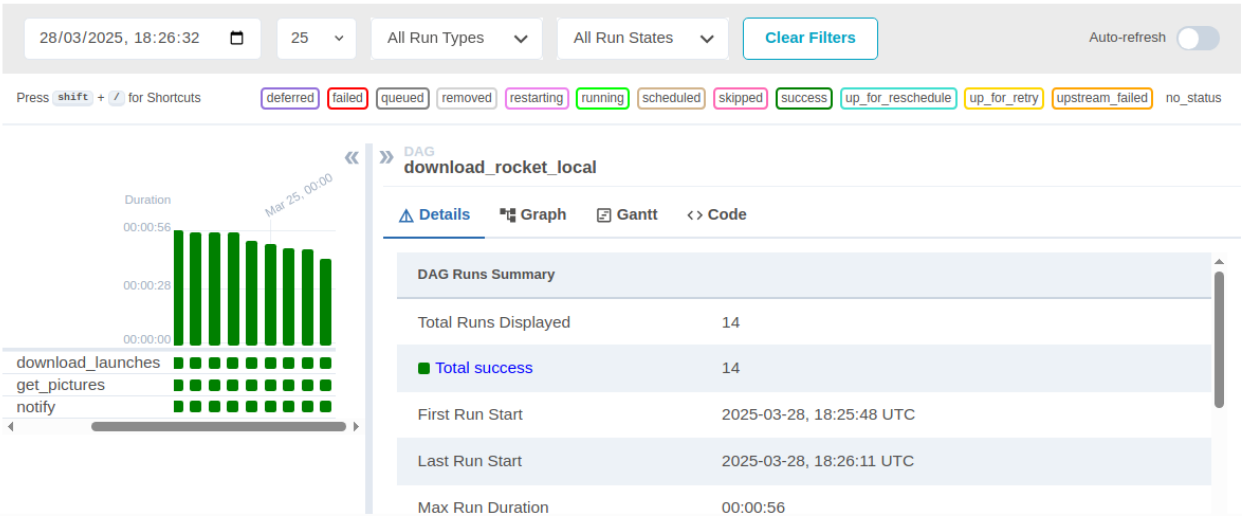
```
dev@dev-vm:~/28_03_rocket/business_case_rocket_25$ sudo docker build -t custom-airflow:slim-2.8.1-python3.11 .
2025/03/28 21:21:41 in: []string{}
2025/03/28 21:21:41 Parsed entitlements: []
[+] Building 4.9s (7/7) FINISHED                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                       0.1s
 => => transferring dockerfile: 568B                                                                       0.1s
 => [internal] load metadata for docker.io/apache/airflow:slim-2.8.1-python3.11                            4.0s
 => [internal] load .dockerignore                                                                          0.1s
 => => transferring context: 2B                                                                            0.0s
 => [1/3] FROM docker.io/apache/airflow:slim-2.8.1-python3.11@sha256:751babd58a83e44ae23c393fe1552196c25f3e  0.0s
 => CACHED [2/3] RUN pip install --no-cache-dir     pandas     scikit-learn     joblib     requests     azu  0.0s
 => CACHED [3/3] RUN mkdir -p /opt/airflow/data /opt/airflow/logs     && chown -R airflow: /opt/airflow/dat  0.0s
 => exporting to image                                                                                     0.1s
 => => exporting layers                                                                                    0.0s
 => => writing image sha256:2590566c41fb0a53daa56b82b144d68bfe41f1478e4fcf3eee081b66821fa1b3               0.0s
 => => naming to docker.io/library/custom-airflow:slim-2.8.1-python3.11                                    0.0s

dev@dev-vm:~/28_03_rocket/business_case_rocket_25$ sudo docker compose up --build
[+] Running 4/4
 ✔ Container business_case_rocket_25-postgres-1   Running                                                  0.0s
 ✔ Container business_case_rocket_25-init-1       Created                                                  0.0s
 ✔ Container business_case_rocket_25-scheduler-1  Running                                                  0.0s
 ✔ Container business_case_rocket_25-webserver-1  Running                                                  0.0s
Attaching to init-1, postgres-1, scheduler-1, webserver-1
```

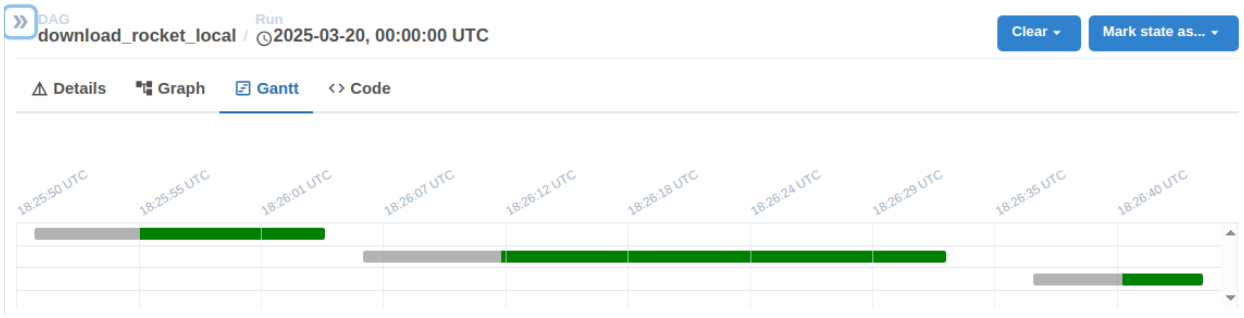
Проверяем доступность Airflow (порт localhost:8080):



Переходим к ДАГу и запускаем его:



ДАГ успешно отработал. На диаграмме Ганта отображается время выполнения кода:



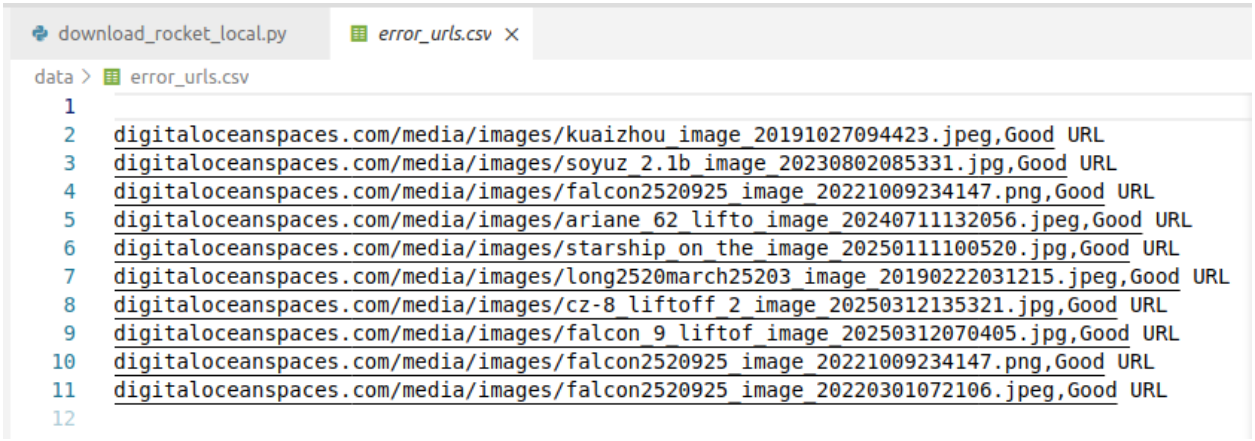
Функция загрузки данных заняла 8 секунд;

Функция получения изображений – 20 секунд;

Функция уведомлений – 3 секунды.

Просмотр загруженных изображений:

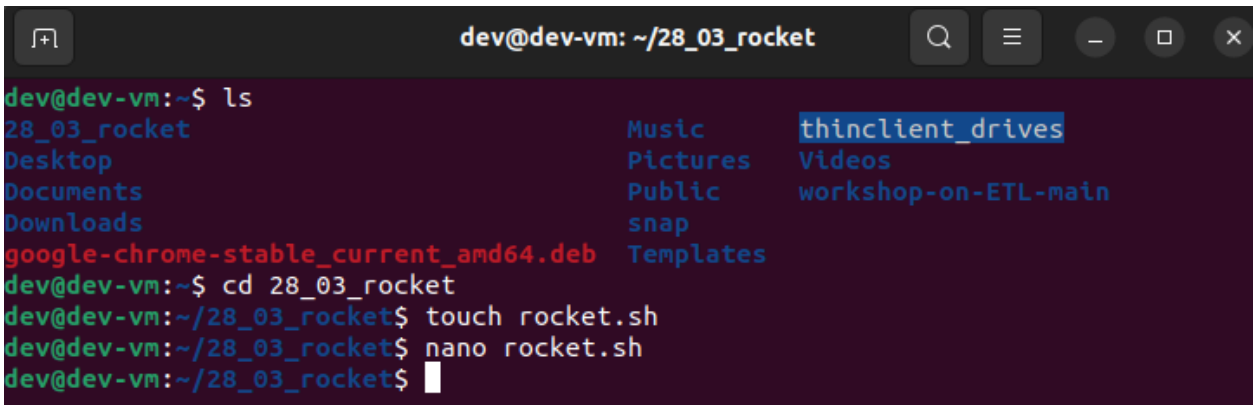
Просмотр файла с информацией о всех полученных URL:



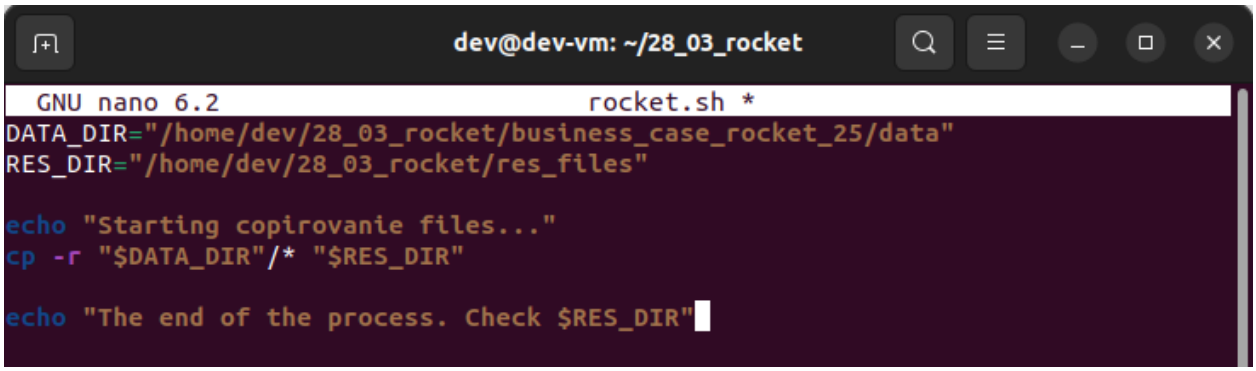
Как видно, все полученные URL доступны, и ни по одному из них ошибок не было.

(Пыталась разными вариантами получить ситуацию, когда URLбыли бы с ошибками, однако, очень хороший Launch Linbrary, там таких нет)

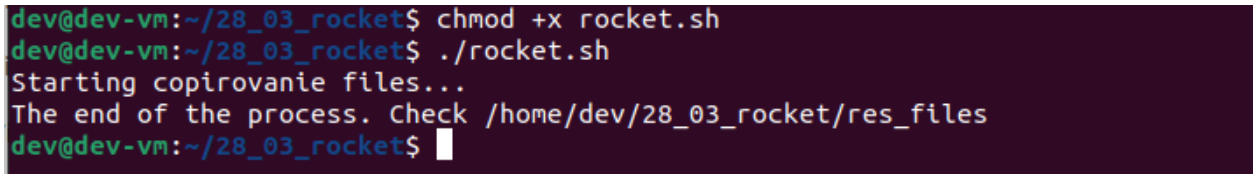
3. Далее необходимо создать исполняемый файл с расширением sh, который автоматизирует выгрузку данных из контейнера в основную ОС данных, полученные в результате работы DAG в Apache Airflow.



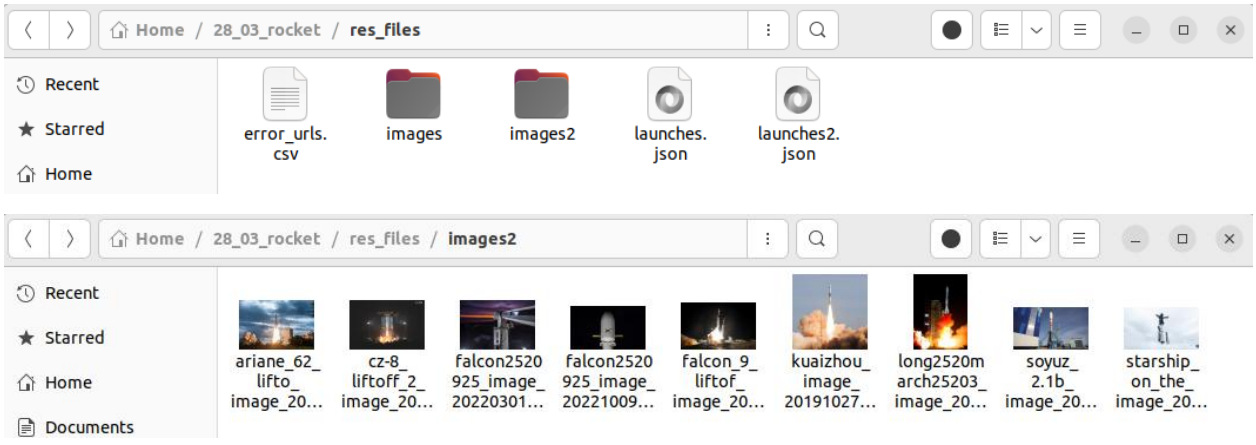
Листинг файла:



Настройка прав и запуск полученного файла:



В результате получили файлы в основной ОС:



4. Далее необходимо провести анализ данных о стартах ракет из JSON-файла и составить отчет о количестве запусков за последний месяц (март 2025).

Для этого воспользуемся Google Colab.

Загружаем файл json и преобразовываем его в удобочитаемый Датафрейм:

Индивидуальное задание 1

Провести анализ данных о стартах ракет из JSON-файла и составить отчет о количестве запусков за последний месяц

```
[1]  1 from google.colab import files
     2 uploaded = files.upload()
```

```
Choose files launches2.json
• launches2.json(application/json) - 27846 bytes, last modified: 28/03/2025 - 100% done
Saving launches2.json to launches2.json
```

```
[2]  1 import pandas as pd
     2 from datetime import datetime
     3
     4 df = pd.read_json('launches2.json')
```

```
1 df
```

	count	next	previous	results
0	29	https://ll.thespacedevs.com/2.0.0/launch/?limi...	NaN	{'id': 'f640a809-8402-46f9-8787-5ac954a9c17e',...
1	29	https://ll.thespacedevs.com/2.0.0/launch/?limi...	NaN	{'id': '75d43ce8-6d8d-4e9f-9348-4f51902e63c2',...
2	29	https://ll.thespacedevs.com/2.0.0/launch/?limi...	NaN	{'id': '21fa8a9b-d2c6-4434-b510-96afb6ca1dd1',...

```
[4]  1 launches = df['results']
```

```
[5]  1 flattened_data = []
```

```
1 #Обработка каждого запуска и вычленение этого всего в структуру
2 for launch in launches:
3     flattened = {
4         'id': launch.get('id'),
5         'name': launch.get('name'),
6         'status': launch.get('status', {}).get('name'),
7         'net': launch.get('net'),
8         'window_start': launch.get('window_start'),
9         'window_end': launch.get('window_end'),
10        'probability': launch.get('probability'),
11        'provider_id': launch.get('launch_service_provider', {}).get('id'),
12        'provider_name': launch.get('launch_service_provider', {}).get('name'),
13        'provider_type': launch.get('launch_service_provider', {}).get('type'),
14        'rocket_id': launch.get('rocket', {}).get('id'),
15        'rocket_name': launch.get('rocket', {}).get('configuration', {}).get('name'),
16        'rocket_family': launch.get('rocket', {}).get('configuration', {}).get('family'),
17        'rocket_full_name': launch.get('rocket', {}).get('configuration', {}).get('full_name'),
18        'mission_id': launch.get('mission', {}).get('id'),
19        'mission_name': launch.get('mission', {}).get('name'),
20        'mission_type': launch.get('mission', {}).get('type'),
21        'mission_description': launch.get('mission', {}).get('description'),
22        'orbit_name': launch.get('mission', {}).get('orbit', {}).get('name'),
23        'pad_id': launch.get('pad', {}).get('id'),
24        'pad_name': launch.get('pad', {}).get('name'),
```

```
1 df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 28 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   id                   10 non-null     object
 1   name                 10 non-null     object
 2   status               10 non-null     object
 3   net                  10 non-null     object
 4   window_start         10 non-null     object
 5   window_end           10 non-null     object
 6   probability          4 non-null      float64
 7   provider_id          10 non-null     int64
 8   provider_name        10 non-null     object
 9   provider_type        10 non-null     object
 10  rocket_id            10 non-null     int64
 11  rocket_name          10 non-null     object
 12  rocket_family        10 non-null     object
 13  rocket_full_name     10 non-null     object
 14  mission_id           10 non-null     int64
 15  mission_name         10 non-null     object
 16  mission_type         10 non-null     object
 17  mission_description  10 non-null     object
 18  orbit_name           10 non-null     object
 19  pad_id               10 non-null     int64
 20  pad_name             10 non-null     object
 21  latitude             10 non-null     object
 22  longitude            10 non-null     object
 23  country_code         10 non-null     object
 24  location_name        10 non-null     object
```

```
[8]  1 #Преобразованит столбцов с датами в тип данных дата
     2 for time_field in ['net', 'window_start', 'window_end']:
     3     df2[time_field] = pd.to_datetime(df2[time_field], format='%Y-%m-%dT%H:%M:%SZ', errors='coerce')
```

```
[9]  1 #Сортировка по дате запуска
     2 df2 = df2.sort_values('net')
```

```
1 df2
```

	id	name	status	net	window_start	window_end	probability	provider_id	provider_name	provider_type	...	orbit_name	pad_id	pad_name	latitude	lon
0	f640a809-8402-46f9-8787-5ac954a9c17e	Kuaizhou-1A | Unknown Payload	Failure	2025-03-01 10:00:00	2025-03-01 09:51:00	2025-03-01 10:37:00	NaN	194	ExPace	Commercial	...	Unknown	21	Launch Area 95A	40.969117	100.
1	75d43ce8-6d8d-4e9f-9348-4f51902e63c2	Soyuz 2.1b/Fregat-M | Glonass-K2 No. 14 (Kosmo...	Success	2025-03-02 22:22:17	2025-03-02 22:00:00	2025-03-03 00:00:00	NaN	193	Russian Space Forces	Government	...	Medium Earth Orbit	36	43/3 (43L)	62.9273	
2	21fa8a9b-d2c6-4434-b510-96afb6ca1dd1	Falcon 9 Block 5 | Starlink Group 12-20	Success	2025-03-03 02:24:00	2025-03-03 02:24:00	2025-03-03 05:21:20	90.0	121	SpaceX	Commercial	...	Low Earth Orbit	80	Space Launch Complex 40	28.56194122	-80.57
3	abb6612f-3fc4-4e9c-96ed-71c12f9e9086	Ariane 62 | CSO-3	Success	2025-03-06 16:24:26	2025-03-06 16:24:26	2025-03-06 16:24:26	NaN	115	Arianespace	Commercial	...	Sun-Synchronous Orbit	67	Ariane Launch Area 4	5.256319	-52.
4	a2b94add-d1e8-4b1d-9f25-afe7904580d3	Starship | Flight 8	Failure	2025-03-06 23:30:00	2025-03-06 23:30:00	2025-03-07 00:30:00	NaN	121	SpaceX	Commercial	...	Suborbital	188	Orbital Launch Mount A	25.9962	-97.

Далее оставляем только нужные столбцы:

```
#Оставляем только интересующие нас столбцы
df2_cleaned=df2[["name","status","net",'provider_name','provider_type','rocket_name','mission_type','country_code','image_url']]
df2_cleaned
```

	name	status	net	provider_name	provider_type	rocket_name	mission_type	country_code	image_url
0	Kuaizhou-1A | Unknown Payload	Failure	2025-03-01 10:00:00	ExPace	Commercial	Kuaizhou	Unknown	CHN	https://thespacedevs-prod.nyc3.digitaloceanspa...
1	Soyuz 2.1b/Fregat-M | Glonass-K2 No. 14 (Kosmo...	Success	2025-03-02 22:22:17	Russian Space Forces	Government	Soyuz 2.1b/Fregat-M	Navigation	RUS	https://thespacedevs-prod.nyc3.digitaloceanspa...
2	Falcon 9 Block 5 | Starlink Group 12-20	Success	2025-03-03 02:24:00	SpaceX	Commercial	Falcon 9	Communications	USA	https://thespacedevs-prod.nyc3.digitaloceanspa...
3	Ariane 62 | CSO-3	Success	2025-03-06 16:24:26	Arianespace	Commercial	Ariane 62	Government/Top Secret	GUF	https://thespacedevs-prod.nyc3.digitaloceanspa...
4	Starship | Flight 8	Failure	2025-03-06 23:30:00	SpaceX	Commercial	Starship	Test Flight	USA	https://thespacedevs-prod.nyc3.digitaloceanspa...
5	Long March 3B/E | TJSW-15	Success	2025-03-09 17:17:00	China Aerospace Science and Technology Corpora...	Government	Long March 3	Government/Top Secret	CHN	https://thespacedevs-prod.nyc3.digitaloceanspa...
6	Long March 8 | G60 Polar Group 05	Success	2025-03-11 16:38:00	China Aerospace Science and Technology Corpora...	Government	Long March 8	Communications	CHN	https://thespacedevs-prod.nyc3.digitaloceanspa...
7	Falcon 9 Block 5 | SPHEREx & PUNCH	Success	2025-03-12 03:10:12	SpaceX	Commercial	Falcon 9	Astrophysics	USA	https://thespacedevs-prod.nyc3.digitaloceanspa...
8	Falcon 9 Block 5 | Starlink Group 12-21	Success	2025-03-13 02:35:30	SpaceX	Commercial	Falcon 9	Communications	USA	https://thespacedevs-prod.nyc3.digitaloceanspa...

В итоге, видим, что в марте 2025 было 10 запусков:

```
1 print(f"За март 2025 было {len(df2_cleaned)} запусков")
```

```
За март 2025 было 10 запусков
```

```
[ ] 1 df2_cleaned.to_csv('launches_processed.csv', index=False)
```

Выгружаем датафрейм в файл CSV, чтобы далее создать более наглядную визуализацию.


```
1 print(f"За март 2025 было {len(df2_cleaned)} запусков")
```

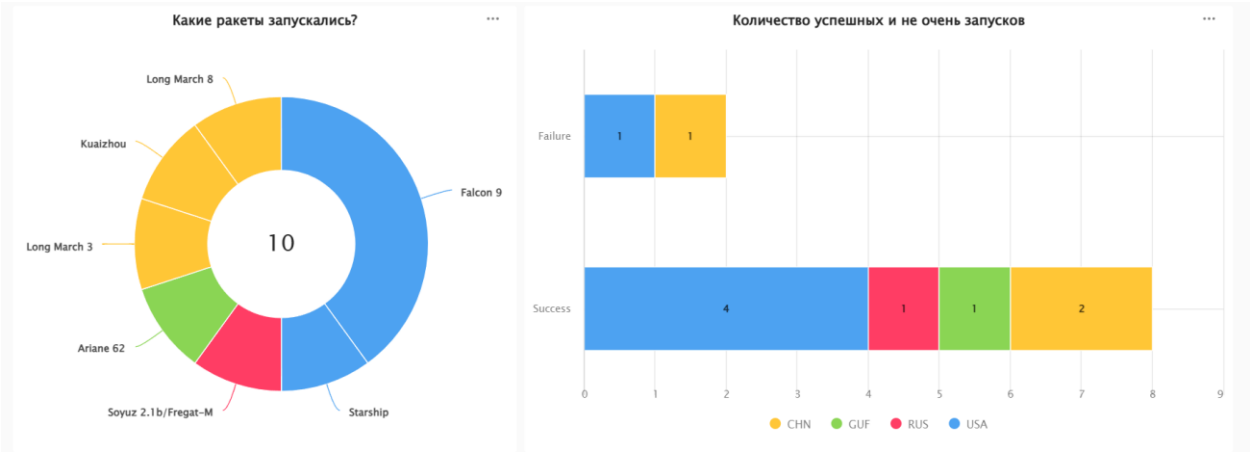
```
За март 2025 было 10 запусков
```

```
[ ] 1 df2_cleaned.to_csv('launches_processed.csv', index=False)
```


Полученная визуализация в Yandex Datalens:

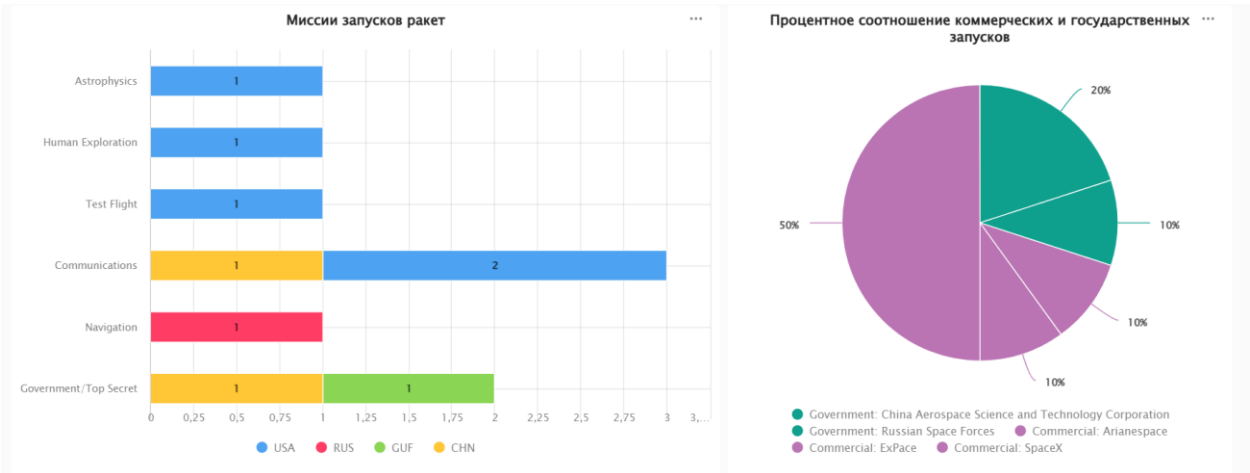
https://datalens.yandex.cloud/gm3wcfto0kk82

Больше всего запусков (5) было у США, а именно компанией-провайдером SpaceX, также в Китае было целых 3 запуска. В России и Французская Гвиана по 1 запуску было в марте.



Также стоит отметить, что 8 из 10 запусков были успешных.

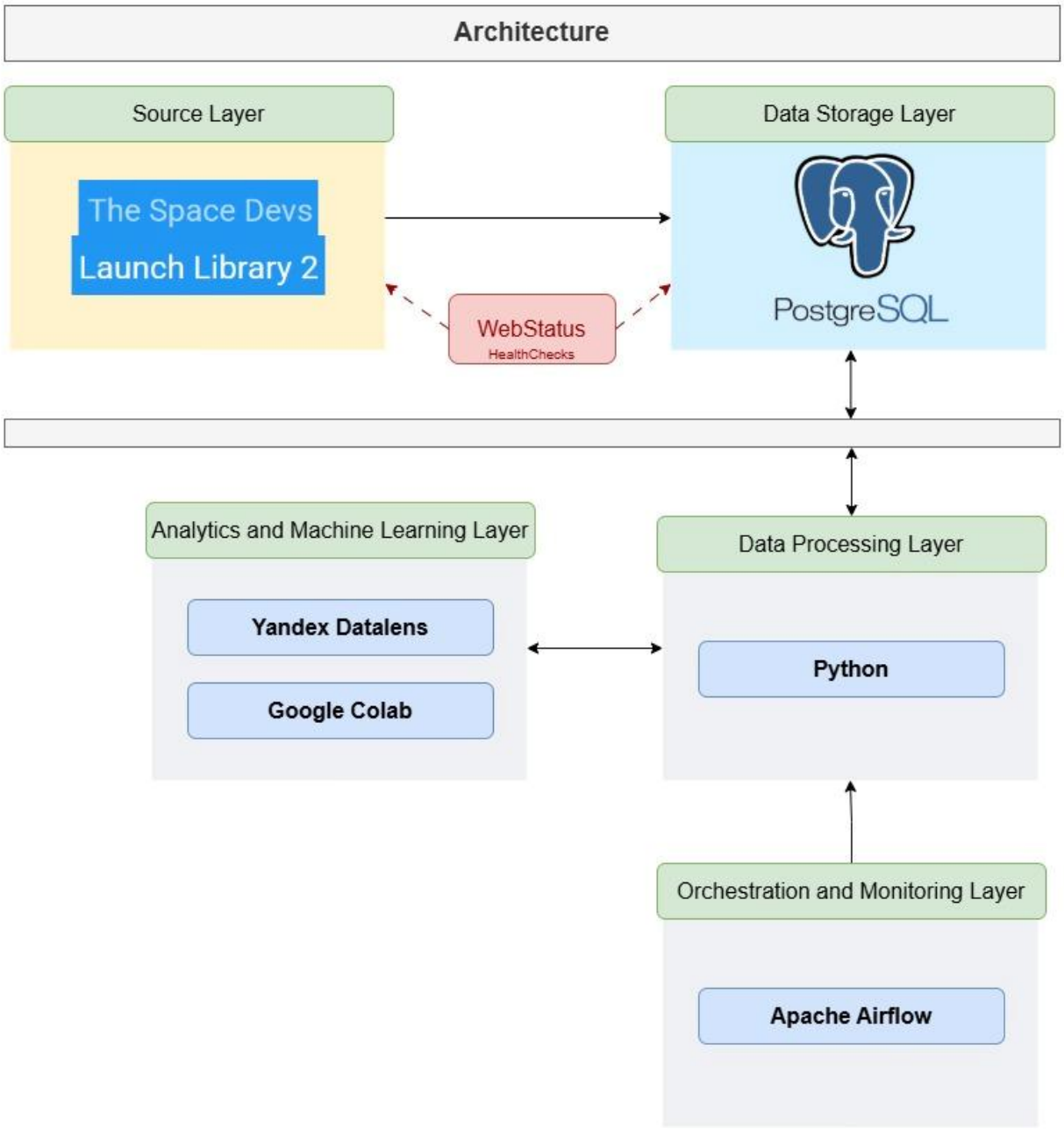
2 запуска в США и Китае были неудачными.



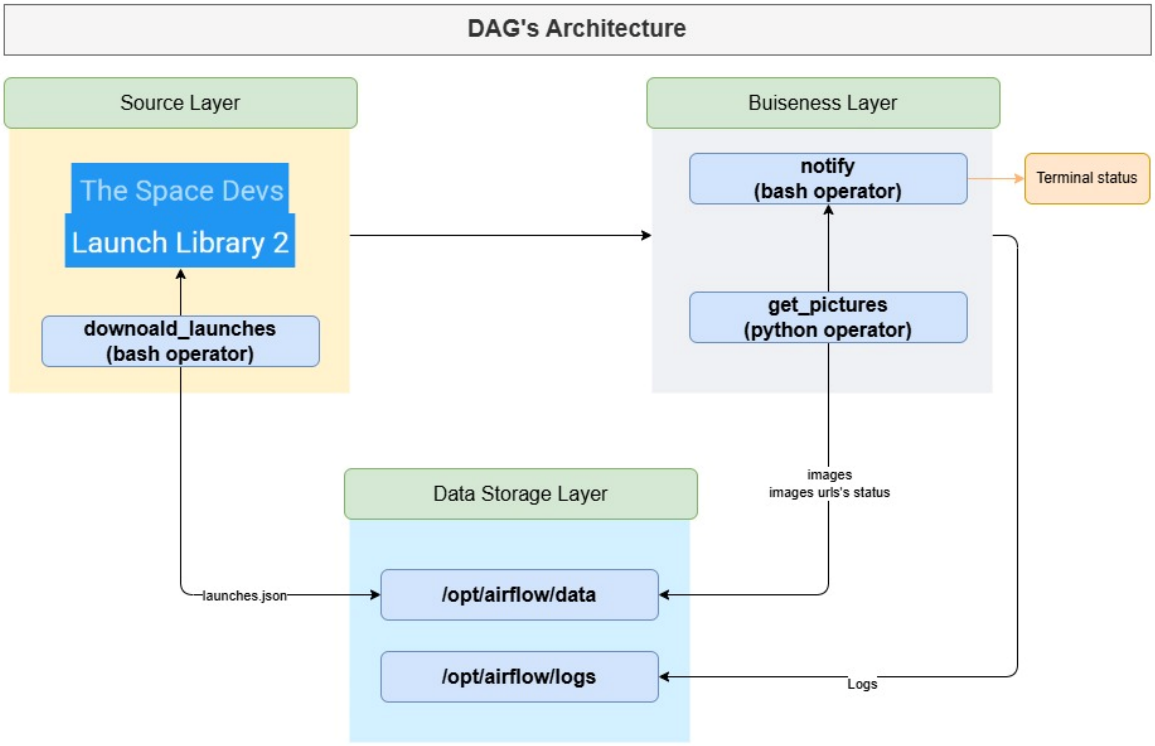
В основном в марте 2025 запускали ракеты с миссией коммуникации.

А также, 70% запусков были сделаны коммерческими компаниями, и лишь 30% государственными.

ВЕРХНЕУРОВНЕВАЯ АРХИТЕКТУРА РЕШЕНИЯ



АРХИТЕКТУРА DAGa



Bash оператор Download_launches по API получает данные о запусках ракет из Launch Library 2 в формате файла json.

Оператор python Get_pictures отбирает из файла json url изображений запусков и проверяет из на доступность (корректный адрес изображения, или есть проблемы). Все url и статусы их доступности записываются в файл «error_urls.scv» (в хранилище /opt/airflow/data).

Помимо этого, оператор Get_pictures изображения, которые удалось получить по url, записывает в хранилище /opt/airflow/data/images.

Bash оператор Notify уведомляет через терминал об успешности отработанных ранее операторов и отображает количество записанных изображений в результате выполнения ОАГа.

ВЫВОДЫ

Таким образом, в ходе выполнения работы были сделаны следующие задачи:

1. был изменен код ДАГа в соответствии с индивидуальными заданиями, благодаря чему удалось получить данные о запусках за март 2025 года, а также записывать все url, включая те, по которым не удалось получить изображение в следствие ошибки;
2. был создан исполняемый файл с расширением .sh для автоматизации выгрузки данных из контейнера в основную систему;
3. был предоставлен отчет с визуализацией о количестве запусков в марте 2025 года;
4. были построены верхнеуровневая архитектура бизнес-кейса, а также архитектура исполняемого ОАГа.

Благодаря предложенному решению по фиксации недоступных данных, а именно url, теперь при отработке ДАГа в случае ошибок, пользователь сможет ознакомиться с ними и вручную (либо при создание автоматизированного решения) попробовать решить ошибку. То есть, повышается шанс получить изображения даже по некорректным url (ранее они просто терялись, точнее отображались просто уведомлением в терминале, откуда довольно тяжело их вычленить для последующего анализа).

К тому же, теперь в результате отработки ОАГа также данные о количестве успешно загруженных изображений фиксируются в удобном формате csv файла, что может быть полезным для подготовки отчета работы ДАГа за длительный промежуток времени.

Помимо этого, по условию одного из заданий необходимо было реализовать отчет о стартах ракет за последний месяц. Полученные результаты отчета могут быть использованы для компаний, планирующих

спонсировать космические миссии. Из отчета, можно увидеть, количество успешных и неуспешных запусков различных компаний, проанализировать, какие типы ракет успешно или не успешно удалось запустить, и, исходя из этого, далее сделать соответствующие выводы о том, у какие провайдеры лидируют, у каких есть шансы и амбиции к увеличению количества успешных запусков и т.д.