

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: Соколова М. С., группа: АДЭУ-201

Преподаватель: Босенко Т. М.

Москва

2024

Цель работы: изучение структуры Apache Airflow, выполнение задач DAG, настройка извлечения данных из контейнера.

Задачи работы:

- Клонировать бизнес-кейс «Rocket» в домашний каталог ВМ;
- Запустить контейнер с кейсом Apache Airflow;
- Выгрузить полученный результат работы DAG в основной каталог ОС;
- Создать файл с расширением .sh для автоматической выгрузки данных из контейнера в основную ОС данных, полученные в результате работы DAG;
- Спроектировать верхнеуровневую архитектуру аналитического решения задания Бизнес-кейса «Rocket»;
- Спроектировать архитектуру DAG Бизнес-кейса «Rocket»;
- Построить диаграмму Ганта работы DAG в Apache Airflow.

Для начала работы необходимо проверить доступность источника.

Верхнеуровневая архитектура аналитического решения задания Бизнес-кейса «Rocket» показана на рисунке 1.

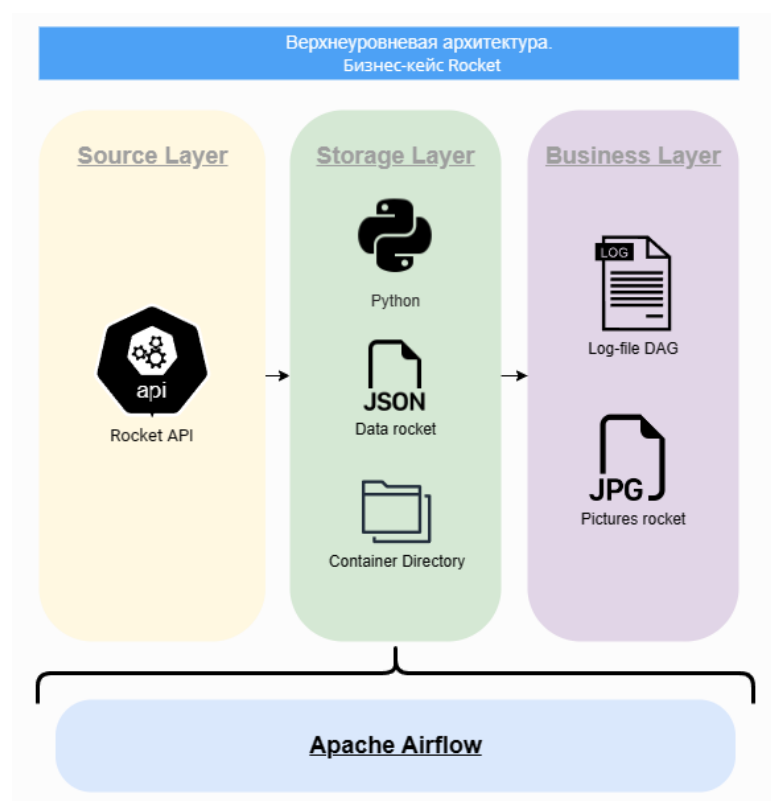


Рисунок 1 – Верхнеуровневая архитектура аналитического решения задания

Архитектуру DAG Бизнес-кейса «Rocket» представлена на рисунке 2.

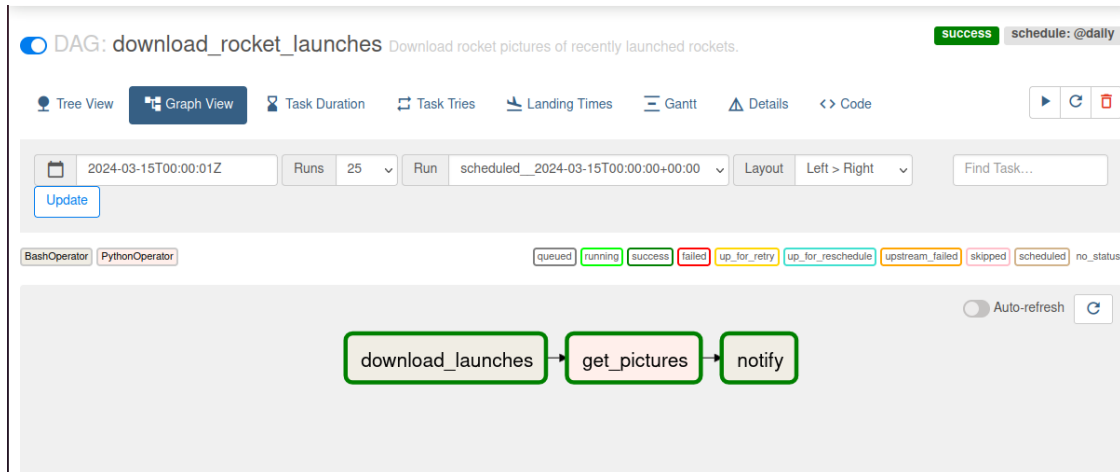


Рисунок 2 – Архитектуру DAG Бизнес-кейса «Rocket»

Описание процесса выполнения работы:

Проверка ответа URL-адреса с помощью Curl из командной строки продемонстрирована на рисунке 3. Ответ представляет собой документ JSON. Квадратные скобки обозначают список. Все значения в этих фигурных скобках относятся к одному запуску ракеты. В ответе содержится информация о предстоящих космических запусках, включая:

- Название запуска и его статус (успех, планируется, и т. д.);
- Даты начала и окончания окна запуска;
- Информация о ракете и миссии;
- Информация о пусковом комплексе;
- Ссылки на изображения и другие ресурсы.

```
mgpu@mgpu-VirtualBox:~$ ls
Desktop  Downloads  Pictures  snap      thinclient_drives  workshop-on-ETL
Documents Music      Public   Templates Videos
mgpu@mgpu-VirtualBox:~$ sudo rm -R workshop-on-ETL
[sudo] password for mgpu:
mgpu@mgpu-VirtualBox:~$ ls
mgpu@mgpu-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":"2024-03-16T00:21:00Z","window_end":"2024-03-16T02:39:00Z","window_start":"2024-03-15T22:39:00Z","inhold":false,"tbdttime":false,"tbddate":false,"probability":null,"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":8189,"configuration":{"id":164,"launch_library_id":188,"url":"https://ll.thespacedevs.com/2.0.0/config/launches/164/","name":"Falcon 9","family":"Falcon","full_name":"Falcon 9 Block 5","variant":"Block 5"}},"mission":{"f
```

Рисунок 3 – Проверка ответа URL-адреса

Команда, представленная на рисунке 4, отображает запись результата запроса будет сохранен в файле под названием data_space.txt

```

mgpu@mgpu-VirtualBox:~$ curl -L "https://ll.thespacedevs.com/2.0.0/launch/upcoming" -o data_space.txt
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
  0     0    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
100 27139 100 27139    0     0 38331    0  --:--:-- --:--:-- --:--:-- 7837k

```

Рисунок 4 - Запись результата в файл

Для запуска контейнера необходимо клонировать и войти в каталог с файлом `docker-compose.yml`. Этот процесс отображен на рисунке 5.

```

mgpu@mgpu-VirtualBox:~$ curl -L "https://ll.thespacedevs.com/2.0.0/launch/upcoming" -o data_space.txt
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
  0     0    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
100 27139 100 27139    0     0 38331    0  --:--:-- --:--:-- --:--:-- 7837k
mgpu@mgpu-VirtualBox:~$ git clone https://github.com/BosenkoTM/workshop-on-ETL
fatal: destination path 'workshop-on-ETL' already exists and is not an empty directory.
mgpu@mgpu-VirtualBox:~$ ls
data_space.txt  Desktop  Downloads  Music  Public  Templates  Videos
mgpu@mgpu-VirtualBox:~$ cd Downloads
mgpu@mgpu-VirtualBox:~/Downloads$ git clone https://github.com/BosenkoTM/workshop-on-ETL
Cloning into 'workshop-on-ETL'...
remote: Enumerating objects: 147, done.
remote: Counting objects: 100% (37/37), done.
remote: Compressing objects: 100% (37/37), done.
remote: Total 147 (delta 18), reused 0 (delta 0), pack-reused 110
Receiving objects: 100% (147/147), 46.24 KiB | 493.00 KiB/s, done.
Resolving deltas: 100% (56/56), done.
mgpu@mgpu-VirtualBox:~/Downloads$ ls
workshop-on-ETL
mgpu@mgpu-VirtualBox:~/Downloads$ cd workshop-on-ETL
mgpu@mgpu-VirtualBox:~/Downloads/workshop-on-ETL$ ls
business_case_rocket  business_case_stocksense  business_case_umbrella  README.md
mgpu@mgpu-VirtualBox:~/Downloads/workshop-on-ETL$ cd business_case_rocket
mgpu@mgpu-VirtualBox:~/Downloads/workshop-on-ETL/business_case_rocket$ ls
dags  docker-compose.yml  example  README.md  scripts
mgpu@mgpu-VirtualBox:~/Downloads/workshop-on-ETL/business_case_rocket$

```

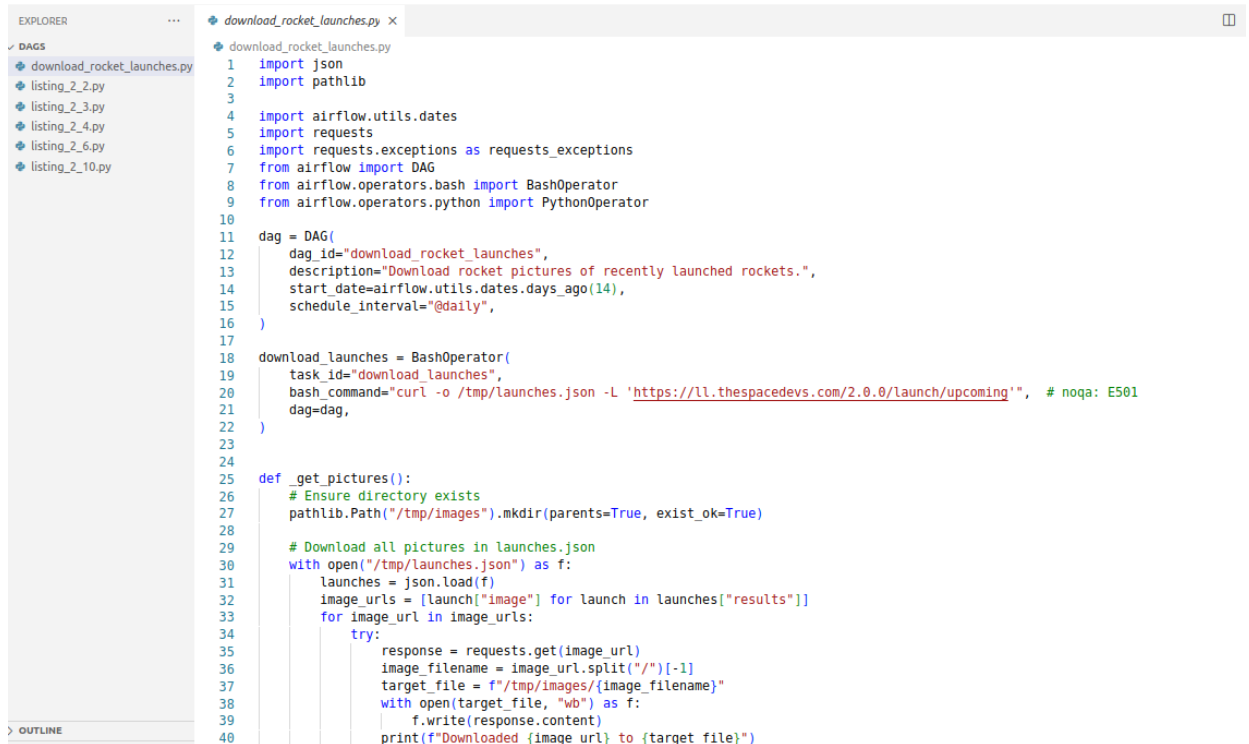
Рисунок 5 – Клонирование репозитория

Файл, фрагмент которого продемонстрирован на рисунке 6, описывает настройку пайплайна для загрузки изображений ракетных запусков. Сначала определяется DAG с идентификатором «`download_rocket_launches`», который описывает задачи и их зависимости. Есть три оператора задач:

- `download_launches`: используется для загрузки информации о предстоящих запусках ракет из API и сохранения данных о запусках в файл «`/tmp/launches.json`».
- `get_pictures`: оператор Python, который загружает изображения ракет из JSON-файла, охраненного в предыдущем шаге, и сохраняет изображения в каталог «`/tmp/images`».
- `notify`: оператор bash, который выводит количество изображений, находящихся в каталоге «`/tmp/images`».

Есть зависимости между задачами: `download_launches` зависит от DAG, `get_pictures` зависит от `download_launches`, а `notify` зависит от `get_pictures`, что

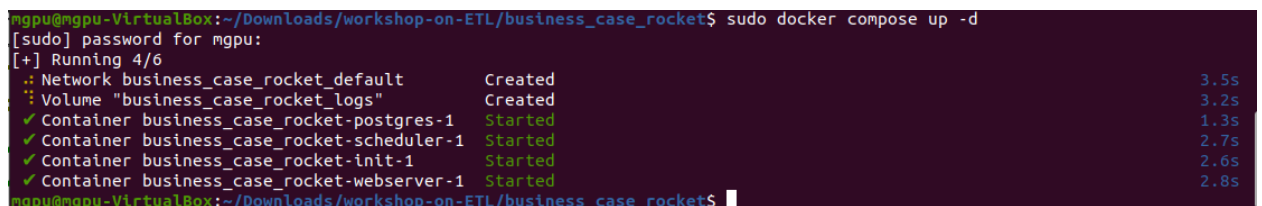
образует цепь выполнения задач. Download_launches использует BashOperator, а get_pictures и notify используют PythonOperator и BashOperator соответственно для выполнения соответствующих действий. DAG запускается ежедневно с учетом последних двух недель.



```
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,
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}")
```

Рисунок 6 – Фрагмент кода DAG

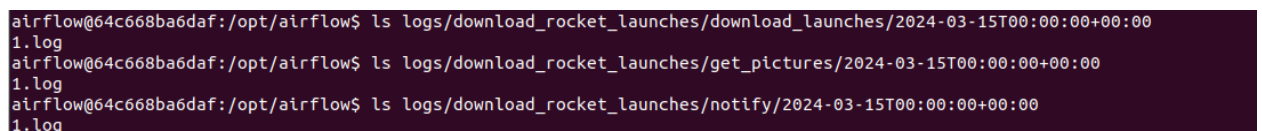
Запуск контейнера Apache Airflow продемонстрирован на рисунке 7.



```
mgpu@mgpu-VirtualBox:~/Downloads/workshop-on-ETL/business_case_rocket$ sudo docker compose up -d
[sudo] password for mgpu:
[+] Running 4/6
  ⚙ Network business_case_rocket_default      Created           3.5s
  ⚙ Volume "business_case_rocket_logs"       Created           3.2s
  ✓ Container business_case_rocket-postgres-1 Started           1.3s
  ✓ Container business_case_rocket-scheduler-1 Started           2.7s
  ✓ Container business_case_rocket-init-1     Started           2.6s
  ✓ Container business_case_rocket-webserver-1 Started           2.8s
mgpu@mgpu-VirtualBox:~/Downloads/workshop-on-ETL/business_case_rocket$
```

Рисунок 7 – Запуск контейнера

Для загрузки лог-файлов необходимо убедиться, что они сохранились в каталогах. Команда и результат представлены на рисунке 8.



```
airflow@64c668ba6daf:/opt/airflow$ ls logs/download_rocket_launches/download_launches/2024-03-15T00:00:00+00:00
1.log
airflow@64c668ba6daf:/opt/airflow$ ls logs/download_rocket_launches/get_pictures/2024-03-15T00:00:00+00:00
1.log
airflow@64c668ba6daf:/opt/airflow$ ls logs/download_rocket_launches/notify/2024-03-15T00:00:00+00:00
1.log
```

Рисунок 8 – Проверка наличия лог-файлов в папках контейнера

```
mgpu@mgpu-VirtualBox:~$ sudo docker cp 64c:/opt/airflow/logs/download_rocket_launches/download_launches/2024-03-15T00:00:00:00/1.log /home/mgpu/Downloads/download_launches_logs.log
Successfully copied 5.12kB to /home/mgpu/Downloads/download_launches_logs.log

mgpu@mgpu-VirtualBox:~$ sudo docker cp 64c:/opt/airflow/logs/download_rocket_launches/get_pictures/2024-03-15T00:00:00:00/1.log /home/mgpu/Downloads/get_pictures_logs.log
Successfully copied 6.66kB to /home/mgpu/Downloads/get_pictures_logs.log

mgpu@mgpu-VirtualBox:~$ sudo docker cp 64c:/opt/airflow/logs/download_rocket_launches/notify/2024-03-15T00:00:00:00/1.log /home/mgpu/Downloads/notify_logs.log
Successfully copied 4.61kB to /home/mgpu/Downloads/notify_logs.log

mgpu@mgpu-VirtualBox:~$ sudo docker cp 64c:/tmp/images /home/mgpu/Downloads/images
Successfully copied 2.76MB to /home/mgpu/Downloads/images
```

Проверка корректности сохранения скопированных файлов представлена на рисунке 10.

Фрагмент лог-файла задачи `download launches` показан на рисунке 11.

```
EXPLORER    ...
└─ LOGS_AND_IMAGES
   └─ download_launches_logs.log
      ├── getPictures_logs.log
      └── notify_logs.log
```

```
# download_launches_logs.log
1 [2024-03-16 10:38:45,670] {taskinstance.py:826} INFO - Dependencies all met for <TaskInstance: download_rocket_launches.download_laun
2 [2024-03-16 10:38:45,707] {taskinstance.py:826} INFO - Dependencies all met for <TaskInstance: download_rocket_launches.download_laun
3 [2024-03-16 10:38:45,744] {taskinstance.py:1017} INFO - 
4 .....
5 [2024-03-16 10:38:45,707] {taskinstance.py:1018} INFO - Starting attempt 1 of 1
6 [2024-03-16 10:38:45,707] {taskinstance.py:1019} INFO - 
7 .....
8 [2024-03-16 10:38:45,729] {taskinstance.py:1038} INFO - Executing <Task(BashOperator): download_launches> on 2024-03-15T08:00:00+00:00
9 [2024-03-16 10:38:45,732] {standard_task_runner.py:51} INFO - Started process 45577 to run task
10 [2024-03-16 10:38:45,744] {standard_task_runner.py:73} INFO - Running: ['airflow', '--tasks', 'run', 'download_rocket_launches', 'down
11 [2024-03-16 10:38:45,744] {standard_task_runner.py:73} INFO - Job 24: Subtask download_launches
12 [2024-03-16 10:38:45,908] {logging_mixin.py:103} INFO - Running <TaskInstance: download_rocket_launches.download_launches 2024-03-15
13 [2024-03-16 10:38:46,361] {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=2024-03-15T08:00:00+00:00
18 AIRFLOW_CTX_DAG_RUN_ID=scheduled_2024-03-15T08:00:00+00:00
19 [2024-03-16 10:38:46,361] {bash.py:135} INFO - Tmp dir root location:
20 /tmp
21 [2024-03-16 10:38:46,362] {bash.py:158} INFO - Running command: curl -o /tmp/launches.json -L "https://ll.tspacedevs.com/2.0.0/Launches
22 [2024-03-16 10:38:46,525] {bash.py:169} INFO - Output:
23 [2024-03-16 10:38:46,671] {bash.py:173} INFO - % Total % Received % Xferd Average Speed Time Time Time Current
24 [2024-03-16 10:38:46,671] {bash.py:173} INFO - Dload Upload Total Spent Left Speed
25 [2024-03-16 10:38:46,671] {bash.py:173} INFO -
26 0 0 0 0 0 0 0 0 --::--::--::--::-- 0
27 0 0 0 0 0 0 0 0 --::--::--::--::-- 0
28 0 0 0 0 0 0 0 0 --::--::~::~~::~ 0
29 [2024-03-16 10:38:47,250] {bash.py:173} INFO -
30 100 27139 100 27139 0 0 46791 0 --::--::~::~~::~ 46791
31 [2024-03-16 10:38:47,255] {bash.py:177} INFO - Command exited with return code 0
32 [2024-03-16 10:38:47,308] {taskinstance.py:1135} INFO - Marking task as SUCCESS. dag_id=download_rocket_launches, task_id=download_laun
33 [2024-03-16 10:38:47,519] {taskinstance.py:1195} INFO - 1 downstream tasks scheduled from follow-on schedule check
34 [2024-03-16 10:38:47,543] {local_task_job.py:118} INFO - Task exited with return code 0
35 .....
```

Рисунок 11 – Фрагмент лог-файла задачи download launches

Фрагмент лог-файла задачи get_pictures представлен на рисунке 12.

```
EXPLORER
download_launches_logs.log
get_pictures_logs.log
notify_logs.log

1 [2024-03-16 10:38:49,477] (taskinstance.py:826) INFO - Dependencies all met for <TaskInstance: download_rocket_launches.get_pictures
2 [2024-03-16 10:38:49,511] (taskinstance.py:826) INFO - Dependencies all met for <TaskInstance: download_rocket_launches.get_pictures
3 [2024-03-16 10:38:49,511] (taskinstance.py:1017) INFO - .....
4 .....
5 [2024-03-16 10:38:49,511] (taskinstance.py:1018) INFO - Starting attempt 1 of 1
6 [2024-03-16 10:38:49,511] (taskinstance.py:1019) INFO - .....
7 .....
8 [2024-03-16 10:38:49,533] (taskinstance.py:1038) INFO - Executing <Task(PythonOperator): get_pictures> on 2024-03-15T00:00:00+00:00
9 [2024-03-16 10:38:49,536] (standard_task_runner.py:51) INFO - Started process 45654 to run task
10 [2024-03-16 10:38:49,541] (standard_task_runner.py:75) INFO - Running: ['airflow', 'tasks', 'run', 'download_rocket_launches', 'get_pictures']
11 [2024-03-16 10:38:49,542] (standard_task_runner.py:76) INFO - Job 29: Subtask get_pictures
12 [2024-03-16 10:38:49,668] (logging_mixin.py:183) INFO - Running <TaskInstance: download_rocket_launches.get_pictures 2024-03-15T00:00:00+00:00>
13 [2024-03-16 10:38:49,954] (taskinstance.py:1238) 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=2024-03-15T00:00:00+00:00
18 AIRFLOW_CTX_DAG_RUN_ID=scheduled_2024-03-15T00:00:00+00:00
19 [2024-03-16 10:38:50,954] (logging_mixin.py:183) INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.digitaloceanspaces.com/media/
20 [2024-03-16 10:38:52,011] (logging_mixin.py:183) INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.digitaloceanspaces.com/media/
21 [2024-03-16 10:38:53,204] (logging_mixin.py:183) INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.digitaloceanspaces.com/media/
22 [2024-03-16 10:38:54,307] (logging_mixin.py:183) INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.digitaloceanspaces.com/media/
23 [2024-03-16 10:38:55,587] (logging_mixin.py:183) INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.digitaloceanspaces.com/media/
24 [2024-03-16 10:38:57,030] (logging_mixin.py:183) INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.digitaloceanspaces.com/media/
25 [2024-03-16 10:38:58,252] (logging_mixin.py:183) INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.digitaloceanspaces.com/media/
26 [2024-03-16 10:38:59,950] (logging_mixin.py:183) INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.digitaloceanspaces.com/media/
27 [2024-03-16 10:39:01,828] (logging_mixin.py:183) INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.digitaloceanspaces.com/media/
28 [2024-03-16 10:39:02,878] (logging_mixin.py:183) INFO - Downloaded https://spacelaunchnow-prod-east.nyc3.digitaloceanspaces.com/media/
29 [2024-03-16 10:39:02,923] (python.py:118) INFO - Done. Returned value was: None
30 [2024-03-16 10:39:03,604] (taskinstance.py:1135) INFO - Marking task as SUCCESS. dag_id=download_rocket_launches, task_id=get_pictures
31 [2024-03-16 10:39:05,872] (local_task_job.py:169) WARNING - State of this instance has been externally set to success. Terminating it
32 [2024-03-16 10:39:05,874] (process_utils.py:95) INFO - Sending Signals.SIGTERM to GPID 45654
33 [2024-03-16 10:39:06,241] (taskinstance.py:1214) ERROR - Received SIGTERM. Terminating subprocesses.
34 [2024-03-16 10:39:06,292] (process_utils.py:61) INFO - Process psutil.Process(pid=45654, status='terminated', exitcode=1, started='16/03/2024 10:38:49')
35 [2024-03-16 10:39:06,297] (local_task_job.py:118) INFO - Task exited with return code 1
36
```

Рисунок 12 – Фрагмент лог-файла задачи get_pictures

Фрагмент лог-файла задачи get_pictures представлен на рисунке 13.

```
EXPLORER
logs_and_images
download_launches_logs.log
get_pictures_logs.log
notify_logs.log

1 [2024-03-16 10:39:11,791] (taskinstance.py:826) INFO - Dependencies all met for <TaskInstance: download_rocket_launches.notify 2024-03-15T00:00:00+00:00>
2 [2024-03-16 10:39:12,096] (taskinstance.py:826) INFO - Dependencies all met for <TaskInstance: download_rocket_launches.notify 2024-03-15T00:00:00+00:00>
3 [2024-03-16 10:39:12,096] (taskinstance.py:1017) INFO - .....
4 .....
5 [2024-03-16 10:39:12,096] (taskinstance.py:1018) INFO - Starting attempt 1 of 1
6 [2024-03-16 10:39:12,096] (taskinstance.py:1019) INFO - .....
7 .....
8 [2024-03-16 10:39:12,264] (taskinstance.py:1038) INFO - Executing <Task(BashOperator): notify> on 2024-03-15T00:00:00+00:00
9 [2024-03-16 10:39:12,338] (standard_task_runner.py:51) INFO - Started process 45894 to run task
10 [2024-03-16 10:39:12,467] (standard_task_runner.py:75) INFO - Running: ['airflow', 'tasks', 'run', 'download_rocket_launches', 'notify']
11 [2024-03-16 10:39:12,468] (standard_task_runner.py:76) INFO - Job 41: Subtask notify
12 [2024-03-16 10:39:13,213] (logging_mixin.py:183) INFO - Running <TaskInstance: download_rocket_launches.notify 2024-03-15T00:00:00+00:00>
13 [2024-03-16 10:39:13,555] (taskinstance.py:1238) 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=2024-03-15T00:00:00+00:00
18 AIRFLOW_CTX_DAG_RUN_ID=scheduled_2024-03-15T00:00:00+00:00
19 [2024-03-16 10:39:13,556] (bash.py:135) INFO - Tmp dir root location:
20 /tmp
21 [2024-03-16 10:39:13,556] (bash.py:158) INFO - Running command: echo "There are now $(ls /tmp/images/ | wc -l) images."
22 [2024-03-16 10:39:13,684] (bash.py:169) INFO - Output:
23 [2024-03-16 10:39:13,785] (bash.py:173) INFO - There are now 8 Images.
24 [2024-03-16 10:39:13,785] (bash.py:177) INFO - Command exited with return code 0
25 [2024-03-16 10:39:13,877] (taskinstance.py:1135) INFO - Marking task as SUCCESS. dag_id=download_rocket_launches, task_id=notify, ex
26 [2024-03-16 10:39:14,118] (taskinstance.py:1195) INFO - 0 downstream tasks scheduled from follow-on schedule check
27 [2024-03-16 10:39:14,181] (local_task_job.py:118) INFO - Task exited with return code 0
28
```

Рисунок 13 – Фрагмент лог-файла задачи notify

Скопированные файлы из контейнера с изображениями успешно выгружены. Результат продемонстрирован рисунке 14.

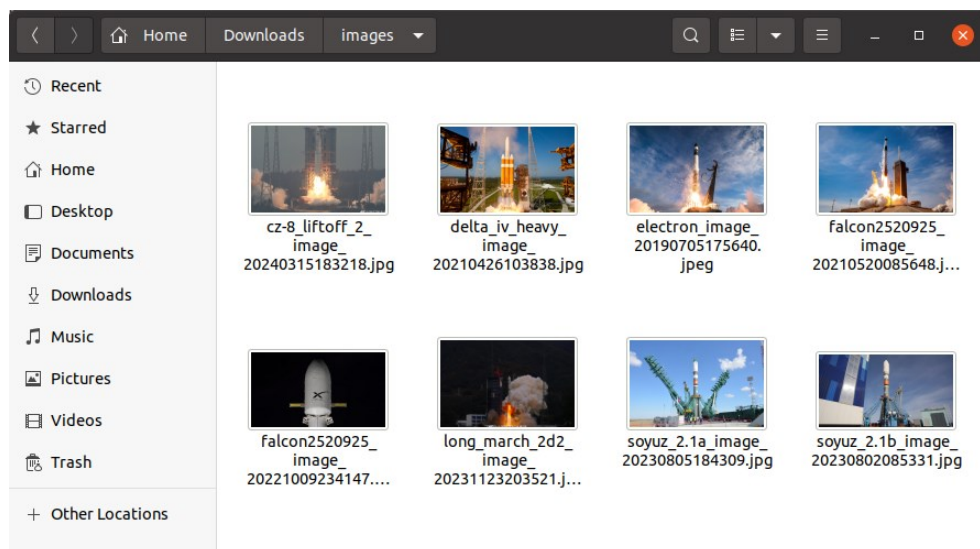


Рисунок 14 – Скопированные файлы

Исполняемый файл с расширением .sh, который автоматизирует выгрузку данных из контейнера в основную ОС данных, полученные в результате работы DAG в Apache Airflow, представлен на рисунке 15.

```
$ get_image.sh x
home > mgpu > Downloads > $ get_image.sh
1  #!/bin/bash
2
3  # Найти ID контейнера по его имени "business_case_rocket-scheduler-1"
4  CONTAINER_ID=$(sudo docker ps -aqf "name=business_case_rocket-scheduler-1")
5
6  sudo docker cp $CONTAINER_ID:/tmp/images /home/mgpu/Downloads/images
```

Рисунок 15 – Файл с расширением .sh

Результат запуска файла с расширением .sh продемонстрирован на рисунке 16.

```
mgpu@mgpu-VirtualBox:~/Downloads$ chmod +x get_image.sh
mgpu@mgpu-VirtualBox:~/Downloads$ ./get_image.sh
[sudo] password for mgpu:
Successfully copied 1.98MB to /home/mgpu/Downloads/images
mgpu@mgpu-VirtualBox:~/Downloads$ CONTAINER_ID=$(sudo docker ps -aqf "name=business_case_rocket-scheduler-1")
mgpu@mgpu-VirtualBox:~/Downloads$ echo $CONTAINER_ID
64c668ba6daf
mgpu@mgpu-VirtualBox:~/Downloads$
```

Рисунок 16 – Запуск файла с расширением .sh

Диаграмма Ганта DAG в Apache Airflow показана на рисунке 17.

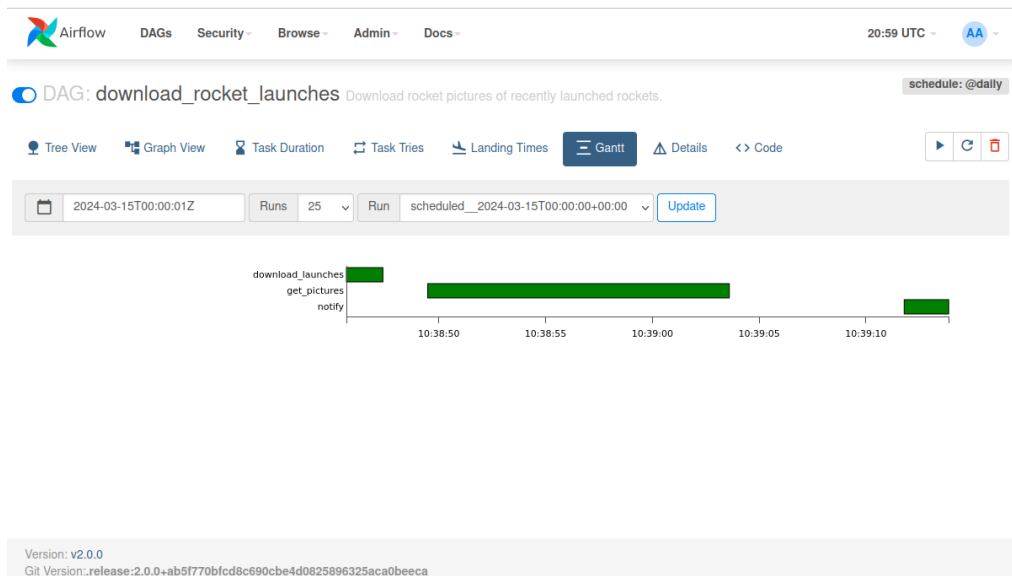


Рисунок 17 – Диаграмма Ганта DAG в Apache Airflow