

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

Лабораторная работа №5

Тема:

Создание ETL-конвейеров

Выполнила: Олифир А. А., группа: АДЭУ-201

Преподаватель: Босенко Т. М.

Москва

2024

В рамках практической работы №5 необходимо реализовать следующие задачи:

Необходимо осуществить проверку ответа URL-адреса с помощью Curl из командной строки (рисунок 1). Ответ представляет собой документ JSON. Квадратные скобки обозначают список. Все значения в этих фигурных скобках относятся к одному запуску ракеты. Можно увидеть информацию об идентификаторе ракеты, а также время начала и окончания окна запуска ракеты, URL-адрес изображения запускаемой ракеты

Рисунок 1 – Проверка ответа URL-адреса

```
ngpu@ngpu-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 | 520.00 KiB/s, done.
Resolving deltas: 100% (56/56), done.
ngpu@ngpu-VirtualBox:~/Downloads$ ls
workshop-on-ETL
ngpu@ngpu-VirtualBox:~/Downloads$ cd workshop-on-ETL
ngpu@ngpu-VirtualBox:~/Downloads/workshop-on-ETL$ ls
business_case_rocket  business_case_stocksense  business_case_umbrella  README.md
ngpu@ngpu-VirtualBox:~/Downloads/workshop-on-ETL$ cd business_case_rocket
ngpu@ngpu-VirtualBox:~/Downloads/workshop-on-ETL/business_case_rocket$ ls
days  docker-compose.yml  example  README.md  scripts
ngpu@ngpu-VirtualBox:~/Downloads/workshop-on-ETL/business_case_rocket$
```

Необходимо запустить образ DAG (рисунок 3).

```
mgpu@mgpu-VirtualBox:~/Downloads/workshop-on-ETL/business_case_rocket$ sudo docker compose up -d
[+] Running 4/4
✓ Container business_case_rocket-postgres-1   Started
✓ Container business_case_rocket-scheduler-1   Started
✓ Container business_case_rocket-init-1        Started
✓ Container business_case_rocket-webserver-1    Started
```

После запуска образа на странице localhost:8080 нужно запустить DAG и посмотреть графическое представление DAG (рисунок 4).

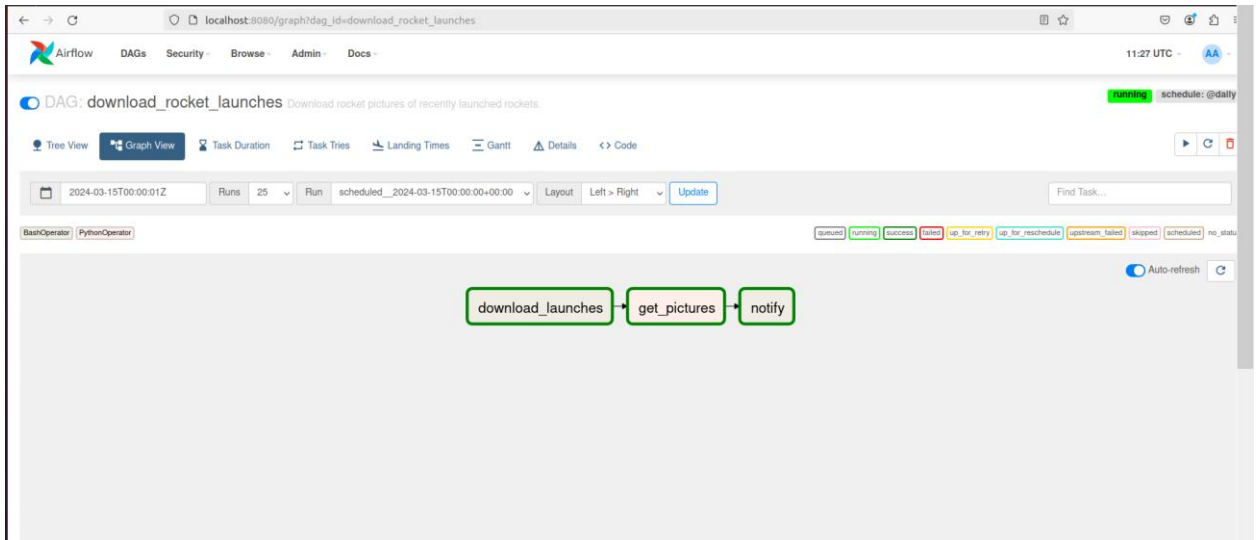


Рисунок 4 – Графическое представление DAG

Диаграмма Ганта для DAG отражена на вкладке Gantt (рисунок 5).

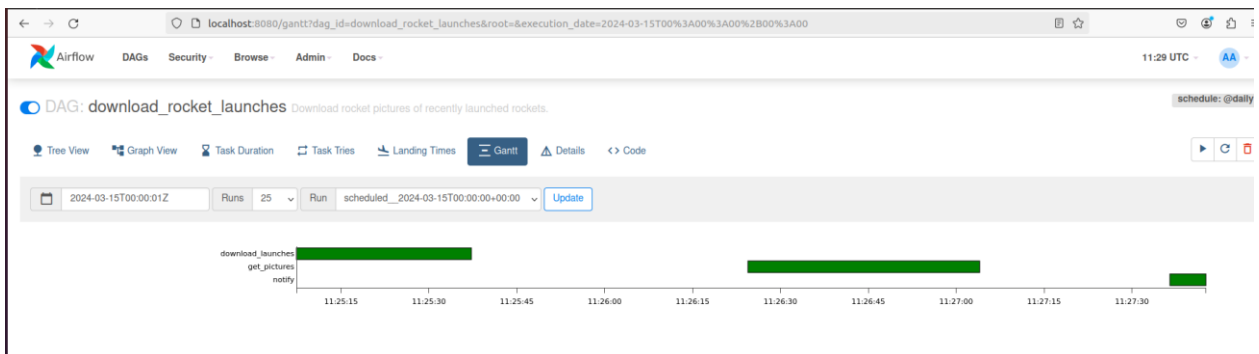


Рисунок 5 – Диаграмма Ганта для DAG

Далее осуществляется проверка наличия логов в контейнере, предварительно необходимо войти в контейнер (рисунок 6).

```
mgpu@mgpu-VirtualBox:~$ sudo docker exec -it business_case_rocket-scheduler-1 /bin/bash
[sudo] password for mgpu:
airflow@64c668ba6daf:/opt/airflow$ ls
airflow.cfg  dags  logs  unittests.cfg  webserver_config.py
airflow@64c668ba6daf:/opt/airflow$ cd logs
airflow@64c668ba6daf:/opt/airflow/logs$ ls
dag_processor_manager  download_rocket_launches  scheduler
airflow@64c668ba6daf:/opt/airflow/logs$ cd download_rocket_launches
airflow@64c668ba6daf:/opt/airflow/logs/download_rocket_launches$ ls
download_launches  get_pictures  notify
airflow@64c668ba6daf:/opt/airflow/logs/download_rocket_launches$ cd notify
airflow@64c668ba6daf:/opt/airflow/logs/download_rocket_launches/notify$ ls
2024-03-02T00:00:00+00:00  2024-03-07T00:00:00+00:00  2024-03-11T00:00:00+00:00  2024-03-15T00:00:00+00:00
2024-03-04T00:00:00+00:00  2024-03-08T00:00:00+00:00  2024-03-12T00:00:00+00:00
2024-03-05T00:00:00+00:00  2024-03-09T00:00:00+00:00  2024-03-13T00:00:00+00:00
2024-03-06T00:00:00+00:00  2024-03-10T00:00:00+00:00  2024-03-14T00:00:00+00:00
airflow@64c668ba6daf:/opt/airflow/logs/download_rocket_launches/notify$ cd 2024-03-15T00:00:00+00:00
airflow@64c668ba6daf:/opt/airflow/logs/download_rocket_launches/notify/2024-03-15T00:00:00+00:00$ ls
1.log
```

Рисунок 6 – Проверка наличия логов в контейнере

Далее осуществляется выгрузка логов из контейнера в основную ОС (рисунки 7-8).

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

Рисунок 7 – Выгрузка логов notify_logs и download_launches_logs

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

Рисунок 8 – Выгрузка лога get_pictures_logs

На рисунке 9 продемонстрирована выгрузка изображений.

```
mgpu@mgpu-VirtualBox:~$ sudo docker cp 64c:/tmp/images /home/mgpu/Downloads/images
Successfully copied 1.98MB to /home/mgpu/Downloads/images
```

Рисунок 9 – Выгрузка изображений ракет

На рисунке 10 показана проверка выгрузки логов и папки с изображениями.

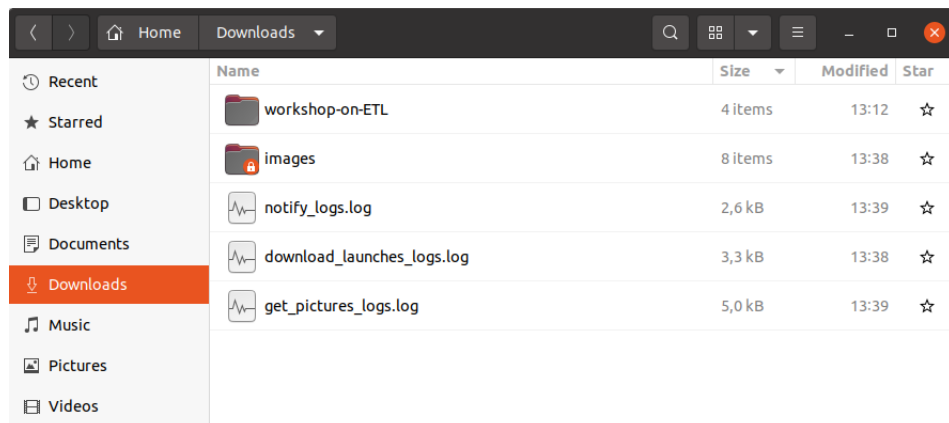


Рисунок 10 – Проверка выгрузки логов и папки с изображениями

В каталоге основной ОС осуществляется проверка файла логов (рисунки 11-13).

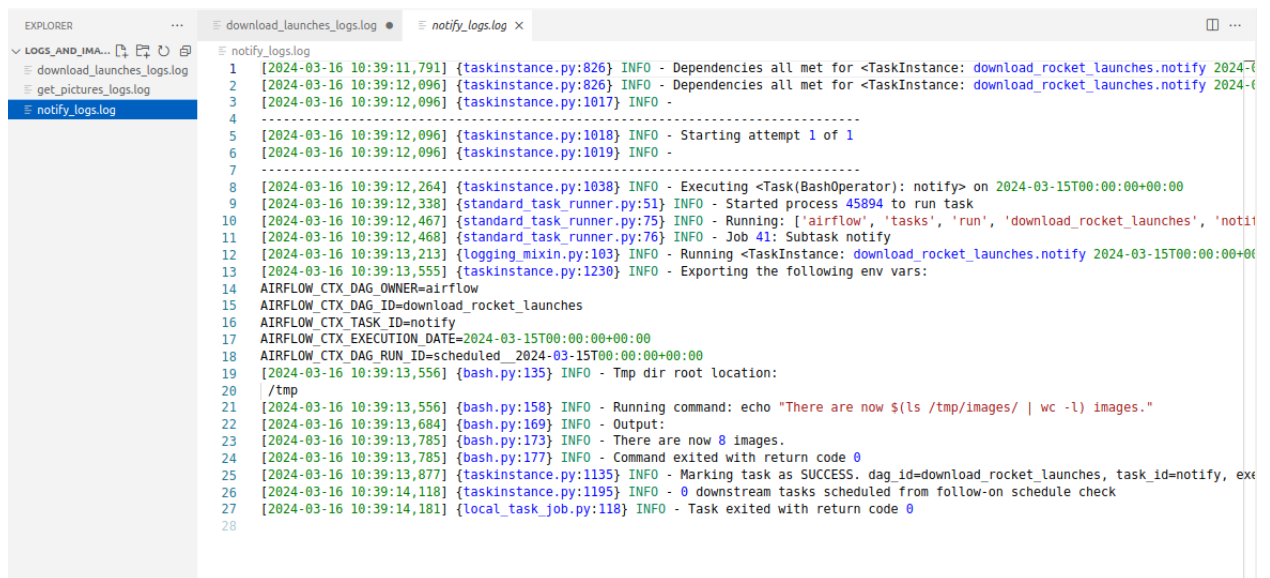


Рисунок 13 – Проверка файла лога notify_logs

На рисунке 14 представлены изображения, выгруженные из контейнера в основную ОС.

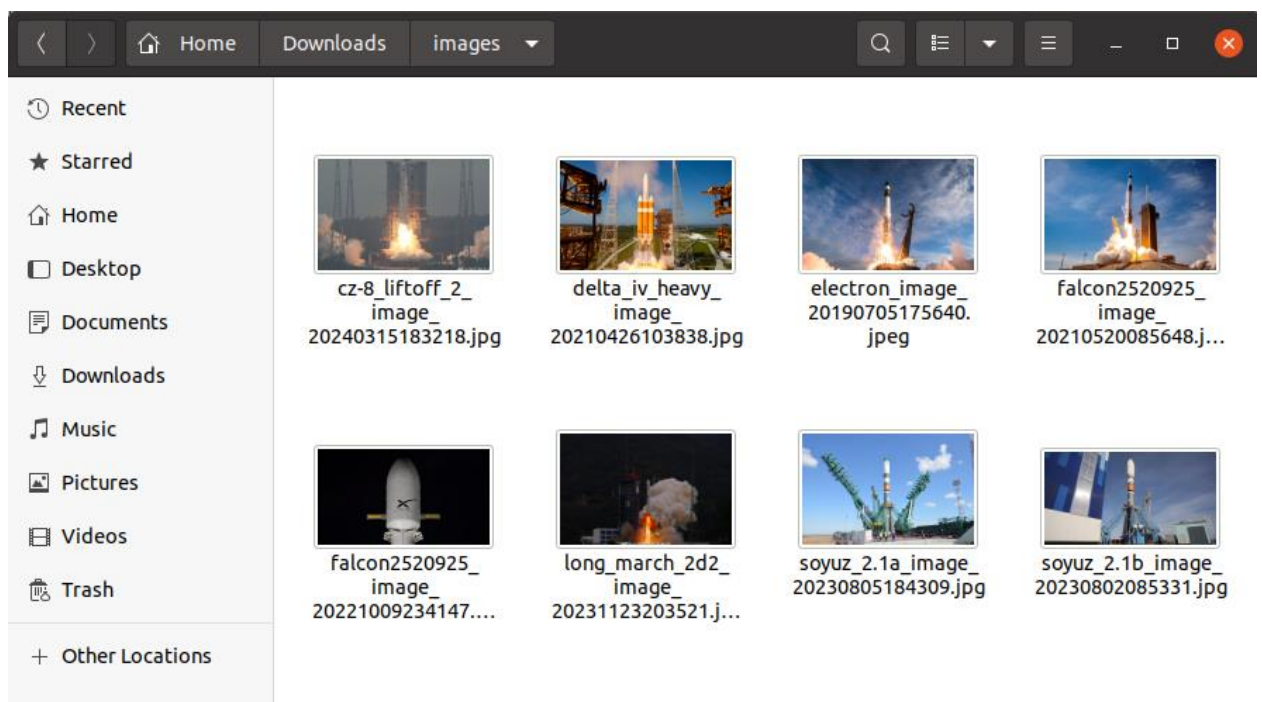


Рисунок 14 – Изображения ракет, выгруженные из контейнера в основную ОС

На рисунке 15 представлена верхнеуровневая архитектура аналитического решения

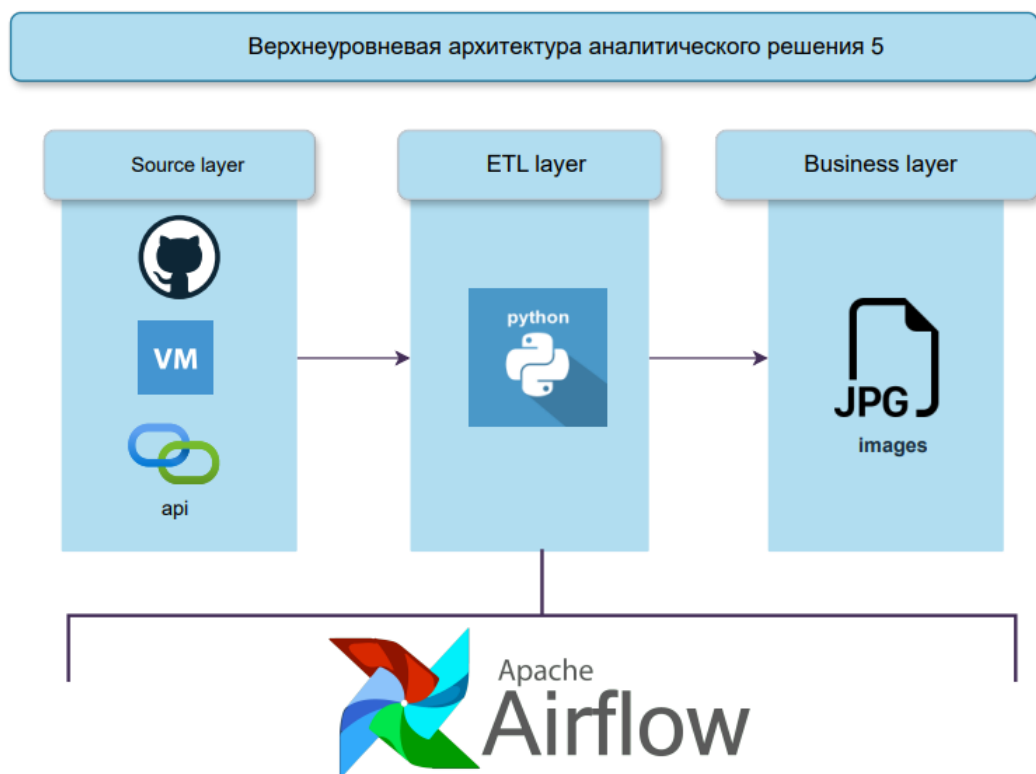


Рисунок 15 – Верхнеуровневая архитектура аналитического решения