

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

Вебинар №3

Тема:

«Бизнес кейс «Rocket»

Выполнил(а): Морозова Валерия АДЭУ-211

Преподаватель:

Москва

2025

Развернута Конфигурация ВМ в VirtualBox.

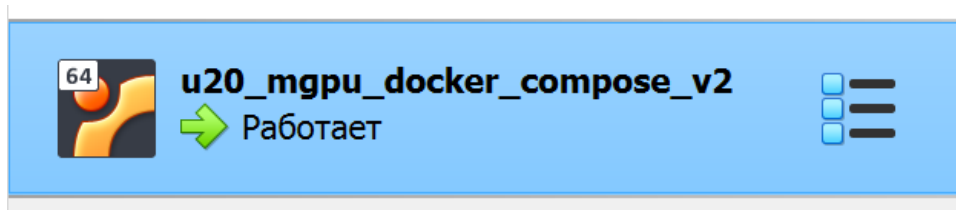


Рисунок 1. Образ развернут

Клонировано на ПК задание Бизнес-кейс «Rocket» в домашний каталог ВМ.

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

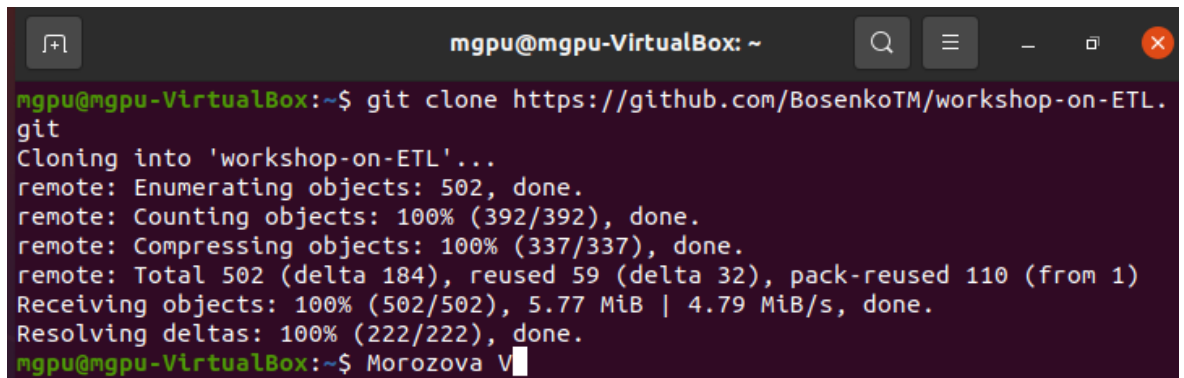


Рисунок 2. Задание клонировано в домашний каталог

Запуск контейнера с кейсом, изучение основных элементов DAG в Apache Airflow.

Прежде чем запустить контейнер необходимо остановить все контейнеры Docker.

```
● mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business_case_rocket_25$ sudo docker stop $(sudo  
do docker ps -q)  
[sudo] password for mgpu:  
46692c13c2e2  
95cd6c7dcce1  
b9e7590beb35
```

Рисунок 3. Остановка всех контейнеров

```
● mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business_case_rocket_25$ sudo docker rm $(sudo  
do docker ps -a -q)  
46692c13c2e2  
95cd6c7dcce1  
eab25ddb7335  
4a5116148fa8  
fd8defe93e21  
7738d43e7e56  
cbf378713957  
b9e7590beb35  
○ mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business_case_rocket_25$
```

Рисунок 4. Удаление запущенных контейнеров

```

● mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business_case_rocket_25$ sudo docker build -t
custom-airflow:slim-2.8.1-python3.11 .
[+] Building 1.4s (7/7) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 568B                             0.0s
=> [internal] load metadata for docker.io/apache/airflow:slim-2.8.1-python3.11 1.3s
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                    0.0s
=> [1/3] FROM docker.io/apache/airflow:slim-2.8.1-python3.11@sha256:751babd58a83e44ae23 0.0s
=> CACHED [2/3] RUN pip install --no-cache-dir pandas scikit-learn joblib 0.0s
=> CACHED [3/3] RUN mkdir -p /opt/airflow/data /opt/airflow/logs && chown -R airflow 0.0s
=> exporting to image                                           0.0s
=> exporting layers                                             0.0s
=> => writing image sha256:b27eadb226ef294cc74800de710609fcd30e2e1df47a1400fb103ba9c73b 0.0s

```

Рисунок 5. Сборка Docker образа с указанием тэга

```

● mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business_case_rocket_25$ sudo cho
wn -R 50000:50000 ./data
[sudo] password for mgpu:
● mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business_case_rocket_25$ 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  --:--:--  0:00:01 --:--:--    0
100 25908 100 25908    0     0  7787     0  0:00:03  0:00:03 --:--:-- 47191
○ mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business_case_rocket_25$ █

```

Рисунок 6. Настройка прав доступа

```

○ mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business_case_rocket_25$ sudo doc
ker compose up --build
[sudo] password for mgpu:
[+] Running 4/2
 ✓ Container business_case_rocket_25-postgres-1    Created           0.1s
 ✓ Container business_case_rocket_25-init-1         Created           0.1s
 ✓ Container business_case_rocket_25-webserver-1    Created           0.1s
 ✓ Container business_case_rocket_25-scheduler-1    Created           0.1s
Attaching to init-1, postgres-1, scheduler-1, webserver-1
postgres-1 |
postgres-1 | PostgreSQL Database directory appears to contain a database; Skippi
ng initialization
postgres-1 |
postgres-1 | 2025-03-29 00:33:19.782 UTC [1] LOG:  starting PostgreSQL 12.18 on

```

Рисунок 7. Запуск контейнеров

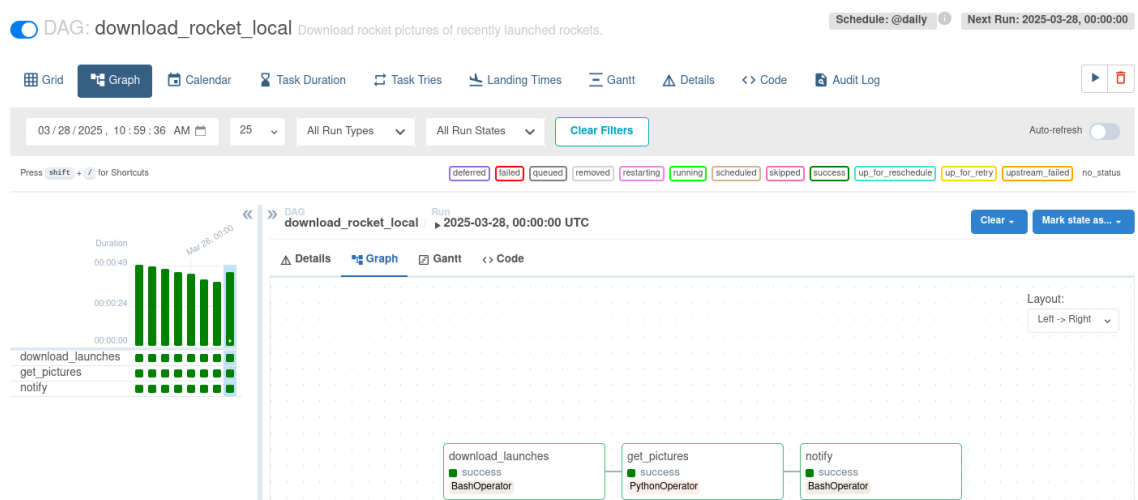


Рисунок 8. DAG выполнен успешно

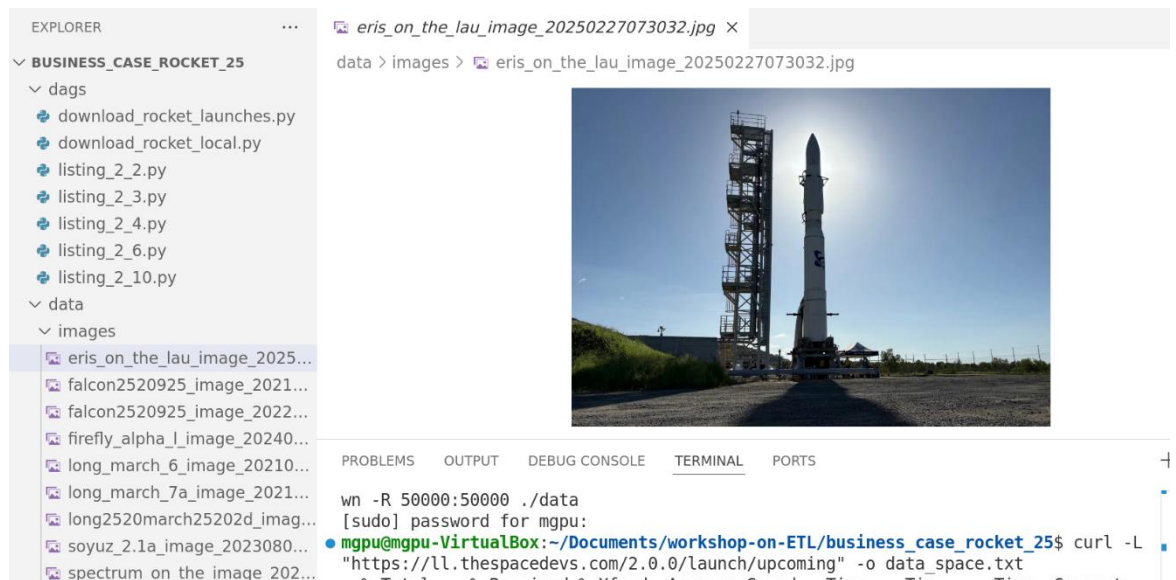


Рисунок 9. Изображения выгрузились в папку

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

```
mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business_case_rocket_25$ chmod +x export_images.sh
mgpu@mgpu-VirtualBox:~/Documents/workshop-on-ETL/business_case_rocket_25$ ./export_images.sh
Сб 29 мар 2025 04:19:49 MSK: Копирование изображений из контейнера...

Права доступа установлены (755)
```

Рисунок 10. Предоставление доступа

```
$ export_images.sh
1 # Конфигурация
2 CONTAINER_DATA_PATH="/home/mgpu/Documents/workshop-on-ETL/business_case_rocket_25/data/images"
3 LOCAL_TARGET_DIR="/home/mgpu/Downloads/rocket_images" # Целевая папка
4
5 # Копируем файлы
6 echo "$(date): Копирование изображений из контейнера"
7 cp -r "$CONTAINER_DATA_PATH"/* "$LOCAL_TARGET_DIR"
8
9
10 # Устанавливаем корректные права
11 chmod -R 755 "$LOCAL_TARGET_DIR"
12 echo -e "\nПрава доступа установлены (755)"
```

Рисунок 11. Файл с расширением **.sh**

1.1 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.

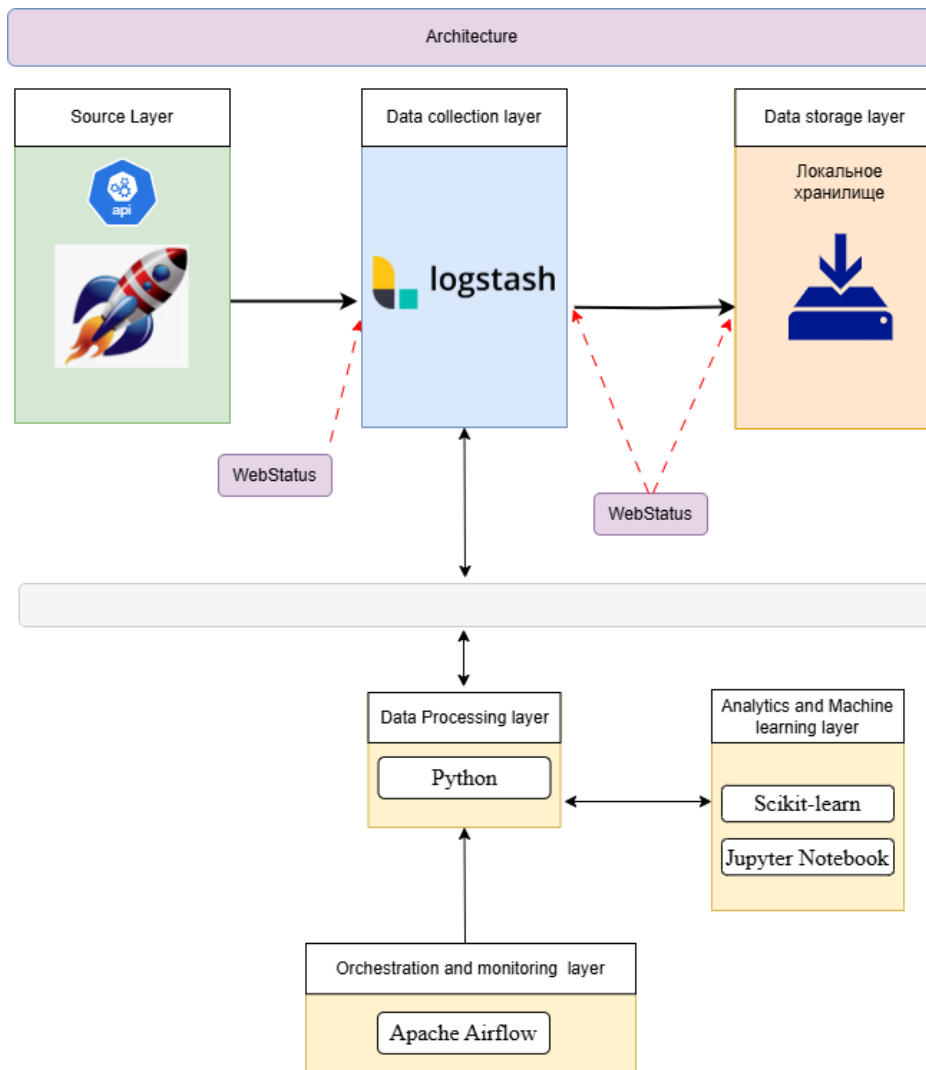


Рисунок 12. Архитектура аналитического решения задания Бизнес-кейса «Rocket»

1.2 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.

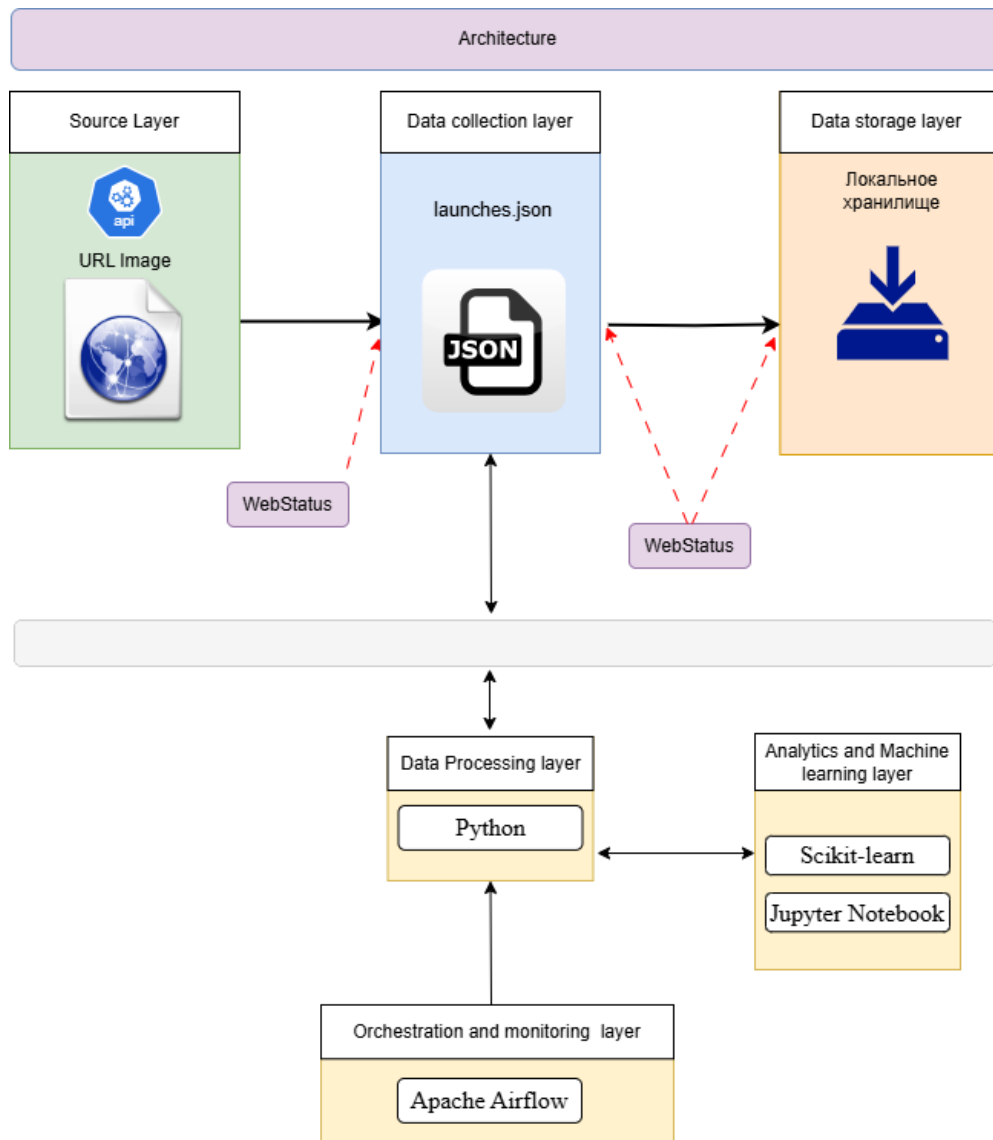


Рисунок 13. Архитектура DAG Бизнес-кейса «Rocket»

1.3 Построить диаграмму Ганта работы DAG в Apache Airflow.

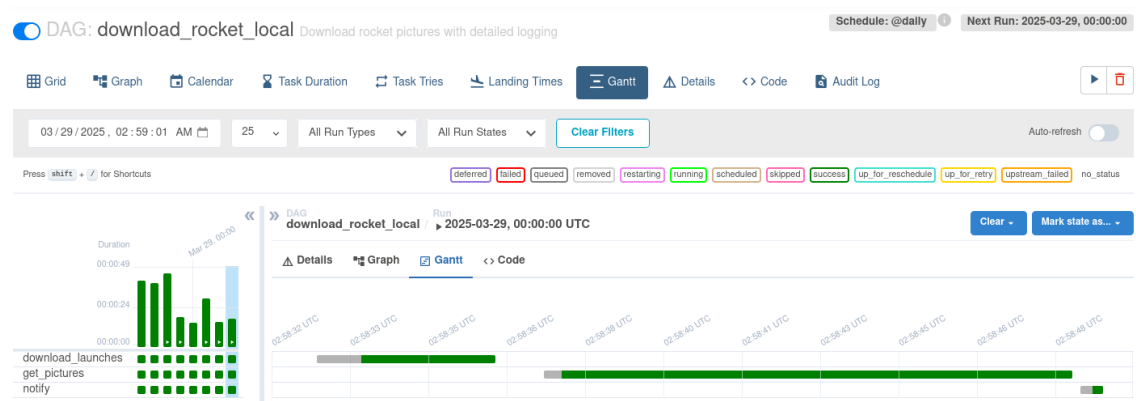


Рисунок 14. Диаграмма Ганта

Индивидуальное задание

Настроить файл логирования для загрузки изображений в процессе работы DAG

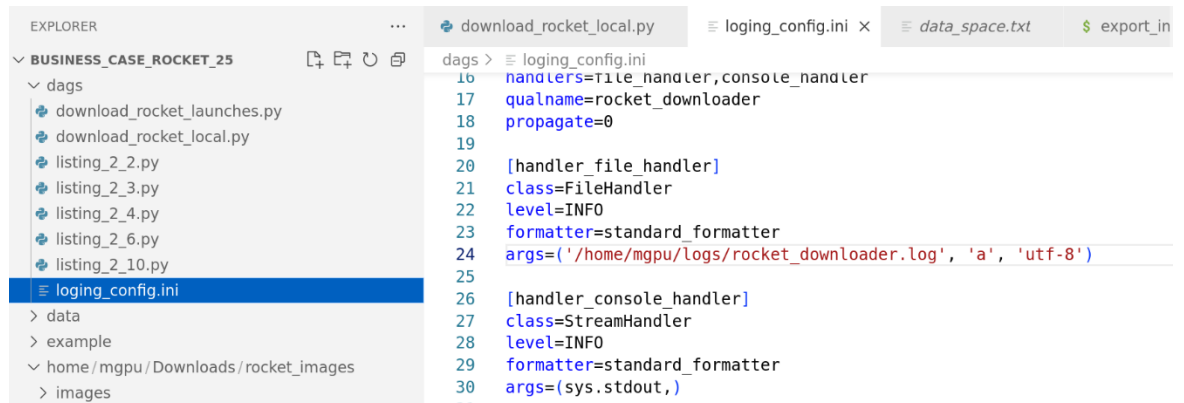


Рисунок 15. Создать файл формата ini

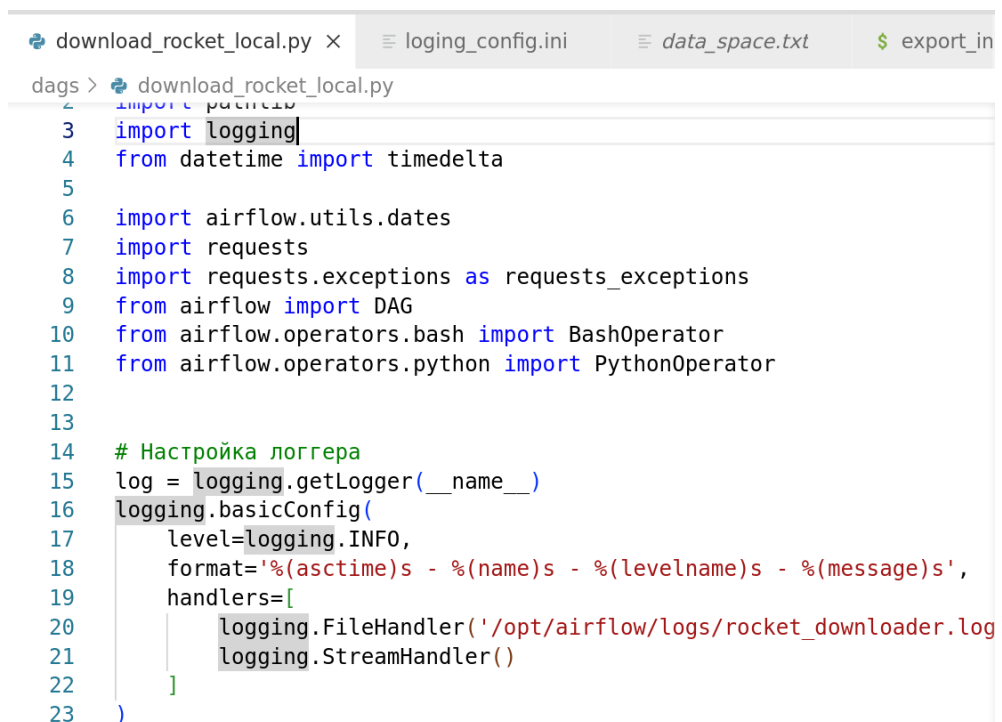


Рисунок 16. Настройка логгера

Вывод:

Запуск DAG и отображение диаграммы Ганта для отслеживания времени выполнения каждого этапа позволяет выявлять места для доработки и совершенствования. Бизнесу в свою очередь важно видеть, на что больше всего уходит время.

Автоматическое копирование и перенос выгруженных файлов на локальную ОС так же сокращает рабочее время и бюджет, выделенный на задачи подобного рода (в случаях, когда оплата по ставке за час).

Внедрение системы логирования и метрик скорости загрузки изображений в DAG Apache Airflow не просто улучшает техническую сторону процесса, но и напрямую влияет на финансовые показатели бизнеса. Снижение затрат на хостинг и трафик за счет оптимизации размера изображений. Логи ошибок (ConnectionError, Timeout) позволяют быстро находить и устранять проблемы с API или внешними серверами. Ускорение загрузки на 100–500 мс может увеличить конверсию на 1–5%. Можно заранее масштабировать серверы или кэшировать изображения, избегая аварийных ситуаций.