

# ETL: Финал

ФИО: Курепин Андрей Дмитриевич

Группа: МИНДА241

Факультет: Инженерия данных

## Задание 1

Разработаем приложение генератор данных на Python, полный код представлен в папке генератора `get_data/*`. Запустим код и сгенерируем 1 миллион записей, Рисунки 1-2.

```
final_2 > task_1 > gen_data > main.py > ...  
1  from models import Transactions  
2  
3  if __name__ == '__main__':  
4      transaction = Transactions()  
5      transaction_data = transaction.gen_array_data(1000 * 1000, True)  
6      transaction.export_to_txt(transaction_data)  
7
```

Рисунок 1 Код для генерации данных

EXPLORER

- HSE\_HW\_ETL
  - final
  - final\_2
    - task\_1
      - data
        - Transactions\_data.csv U
      - gen\_data
        - \_\_pycache\_\_
        - \_\_init\_\_.py U
        - main.py U
        - models.py U
        - requirements.txt U
      - sql

main.py ...\gen\_data U

models.py ...\gen\_data U

Transactions\_data.csv U

final\_2 > task\_1 > data > Transactions\_data.csv

```
1  transaction_id,user_id,amount,currency,transaction_date,is_fraud  
2  1,3325,-7387,EUR,2024-11-05 03:39:22,false  
3  2,4934,-1887,USD,2024-07-24 03:39:26,true  
4  3,7366,5589,GBP,2023-05-11 12:49:10,true  
5  4,4871,-3348,RUB,2022-01-20 03:18:40,false  
6  5,3322,400,USD,2022-03-10 17:43:11,true  
7  6,7030,-2275,EUR,2024-12-07 03:47:43,true  
8  7,4518,-8516,EUR,2024-07-18 11:29:14,false  
9  8,8916,-220,JPY,2023-01-07 14:28:15,true  
10 9,8012,-5798,EUR,2023-10-07 05:20:24,true  
11 10,7116,-7574,EUR,2022-11-24 15:14:33,false  
12 11,2588,-8126,JPY,2023-07-17 21:29:52,false  
13 12,4580,-2736,USD,2022-01-13 06:06:45,true  
14 13,9932,-8101,USD,2024-09-10 05:08:56,false
```

Рисунок 2 Сгенерированные данные

Создадим базу данных YDB, Рисунки 3-4. SQL код приведен в папке sql/\*.

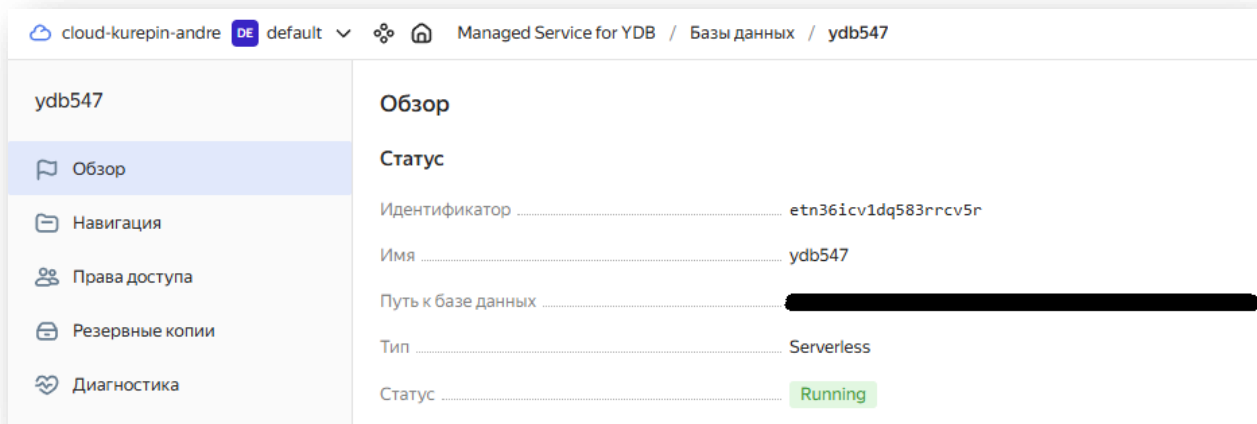
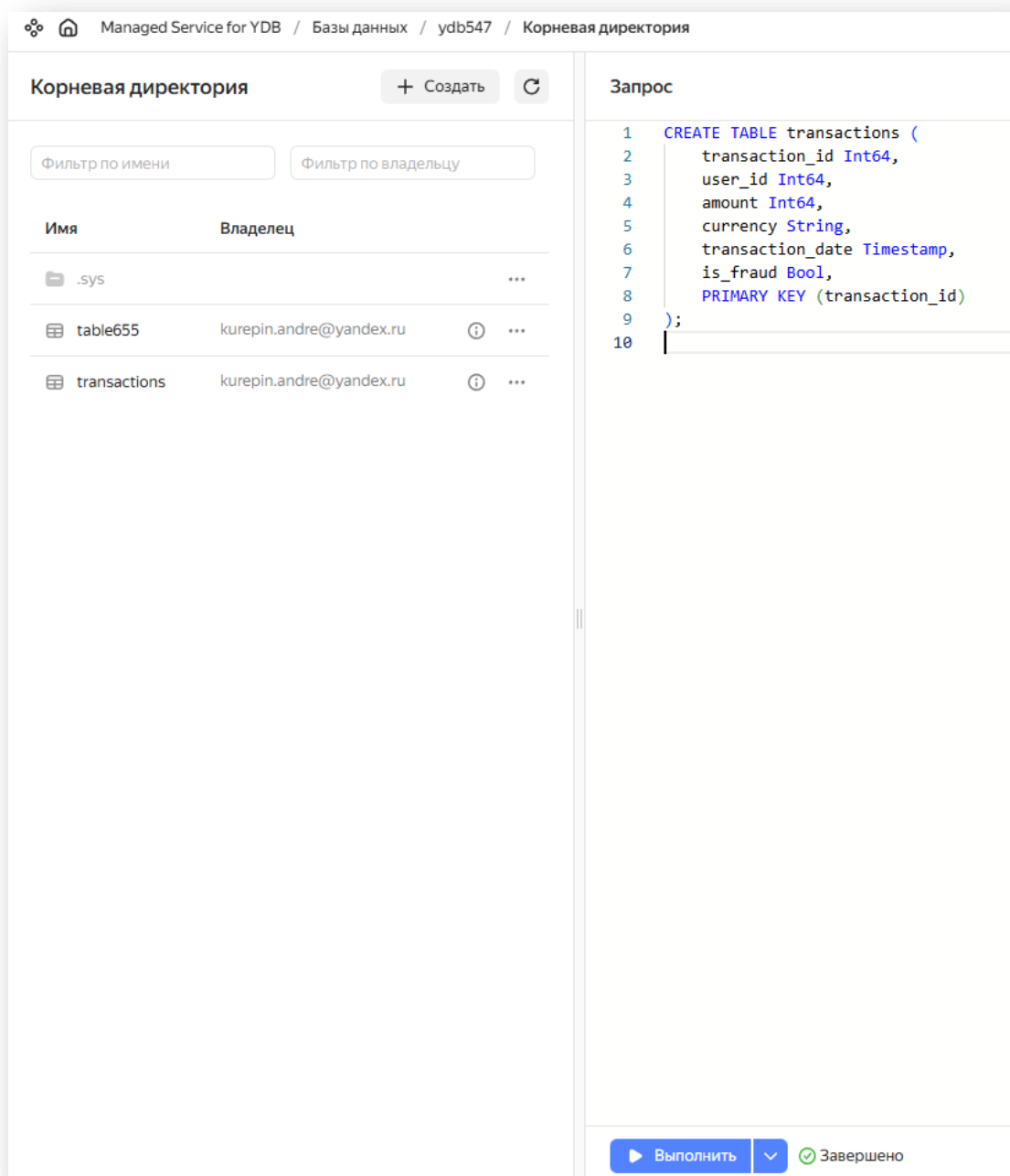


Рисунок 3 Создание БД



**Рисунок 4** Создание таблицы transactions

Выполним вставку сгенерированных данных в созданную таблицу, Рисунок 5-6. Проверим размер таблицы, Рисунок 7.

```
PS D:\HSE\etl|hse_hw_etl> ydb --verbose `
>> --endpoint "██████████"
>> --database "██████████"
>> --sa-key-file "██████████"
>> import file csv `
>> --path "transactions" `
>> --delimiter ";" `
>> --skip-rows 1 `
>> --null-value "" `
>> --progress `
>> "final_2\task_1\data\Transactions_dat
Using service account key file provided wi
2% |тцитцйтцитцстцстцстцстцстцстцстцстц
```

### Рисунок 5 Загрузка данных

[illegible]

### Рисунок 6 Успешная загрузка данных

Managed Service for YDB / Базы данных / ydb547 / Корневая директория / transactions

transactions

Строковая таблица

Показаны только первые 100 записей

#	transaction_id	user_id	amount	currency	transac
1	1	3325	-7387	"EUR"	2024-11 05T03:3
2	2	4934	-1887	"USD"	2024-07 24T03:3
3	3	7366	5589	"GBP"	2023-05 11T12:4
4	4	4871	-3348	"RUB"	2022-01 20T03:1
5	5	3322	400	"USD"	2022-03 10T17:4
6	6	7030	-2275	"EUR"	2024-12 07T03:4
7	7	4518	-8516	"EUR"	2024-07 18T11:2
8	8	8916	-220	"JPY"	2023-01 07T14:2
9	9	8012	-5798	"EUR"	2023-10 07T05:2

Запрос

1 select count(\*) from transactions;

Выполнить

Завершено

Результат #1

#	column0
1	1000000

**Рисунок 7** Размер таблицы transactions

Выполним задание, создадим трансфер из YDB в Object Storage, Рисунок 8. Результаты работы трансфера представлены на Рисунках 9-10.

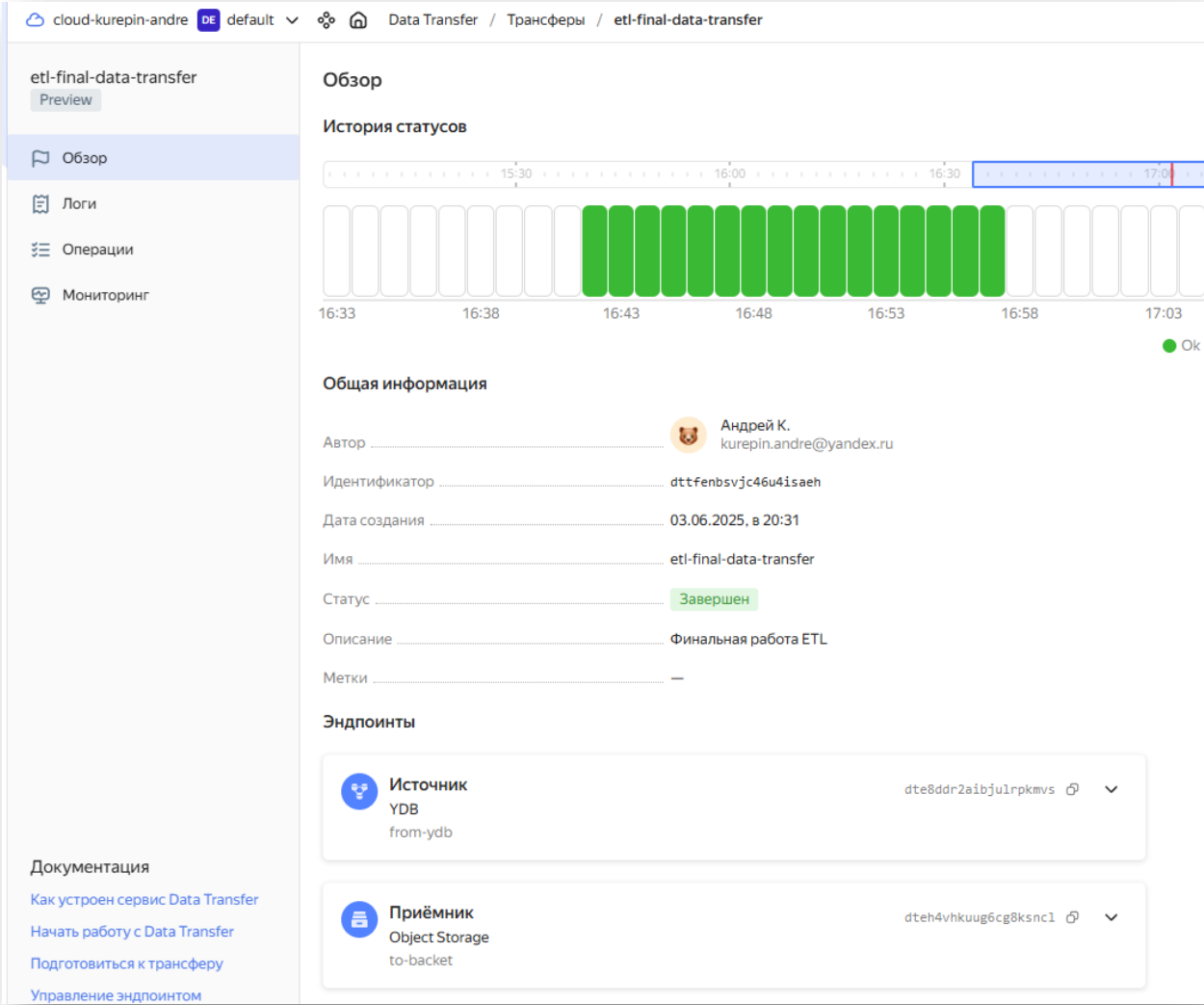


Рисунок 8 Трансфер данных

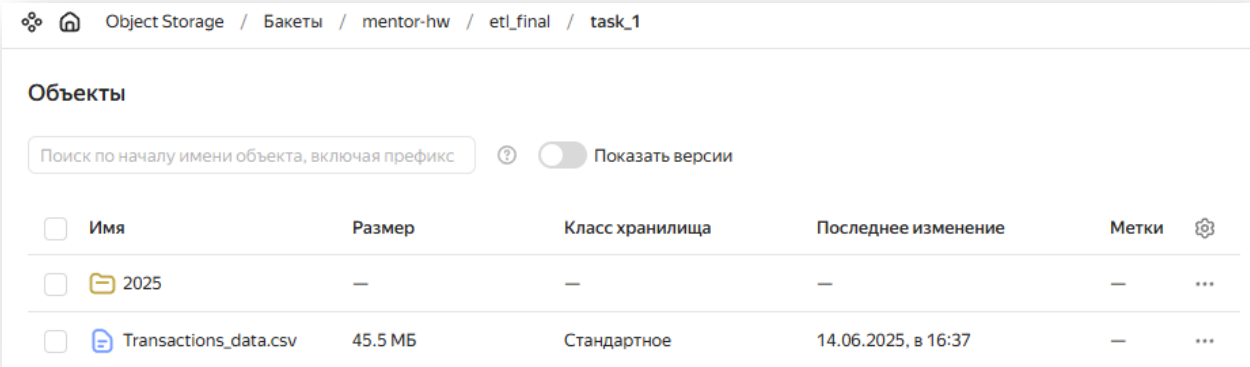


Рисунок 9 Папка с данными

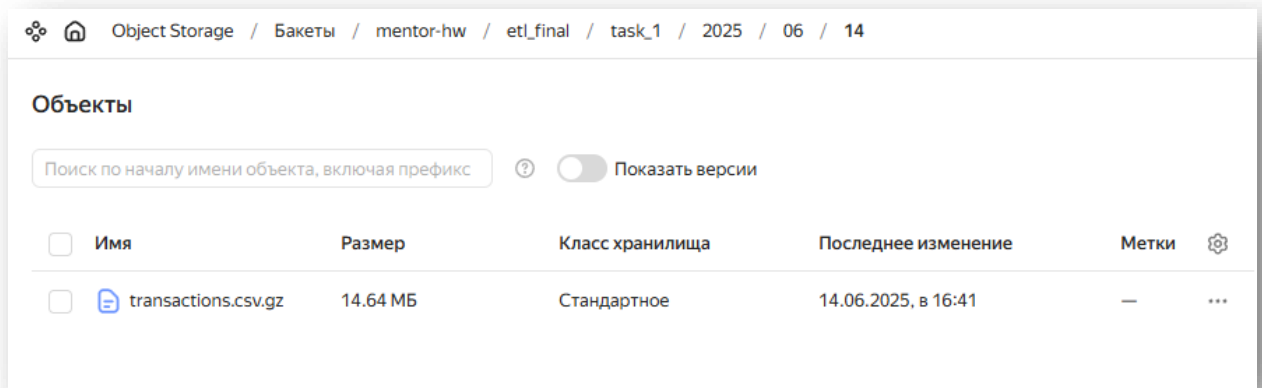


Рисунок 10 Результирующий файл

## Задание 2

Концепция – развернут Airflow, который запускает по расписанию DAG для анализа статистики по мошенническим транзакциям. На первом этапе поднимается Data Processing, на втором выполняется Spark задание, после вычислительный DP кластер удаляется. Spark задание берет файл полученный в результате выполнения первого задания.

Разработаем DAG и Spark скрипты, загрузим их в storage Рисунок 11 и 12. Полный код представлен в файлах *DP-fraud-DAG.py* и *prepare\_fraud\_info.py*.

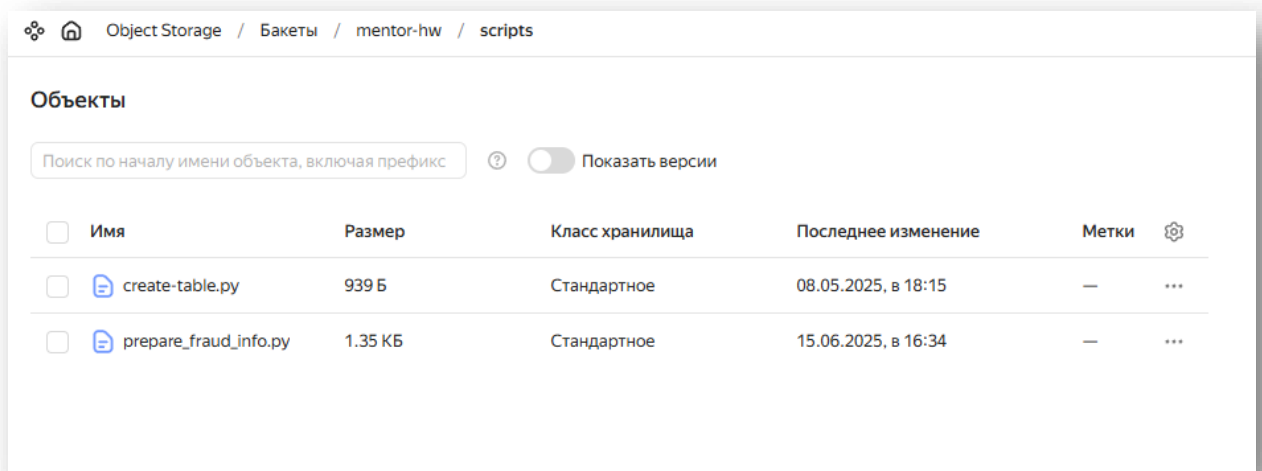
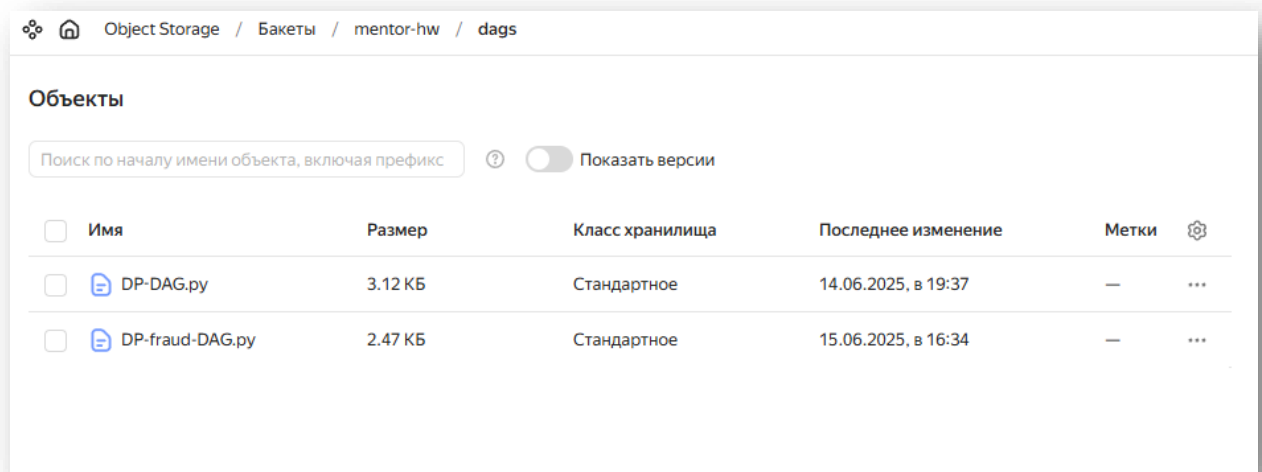


Рисунок 11 Spark скрипт



Object Storage / Бакеты / mentor-hw / dags

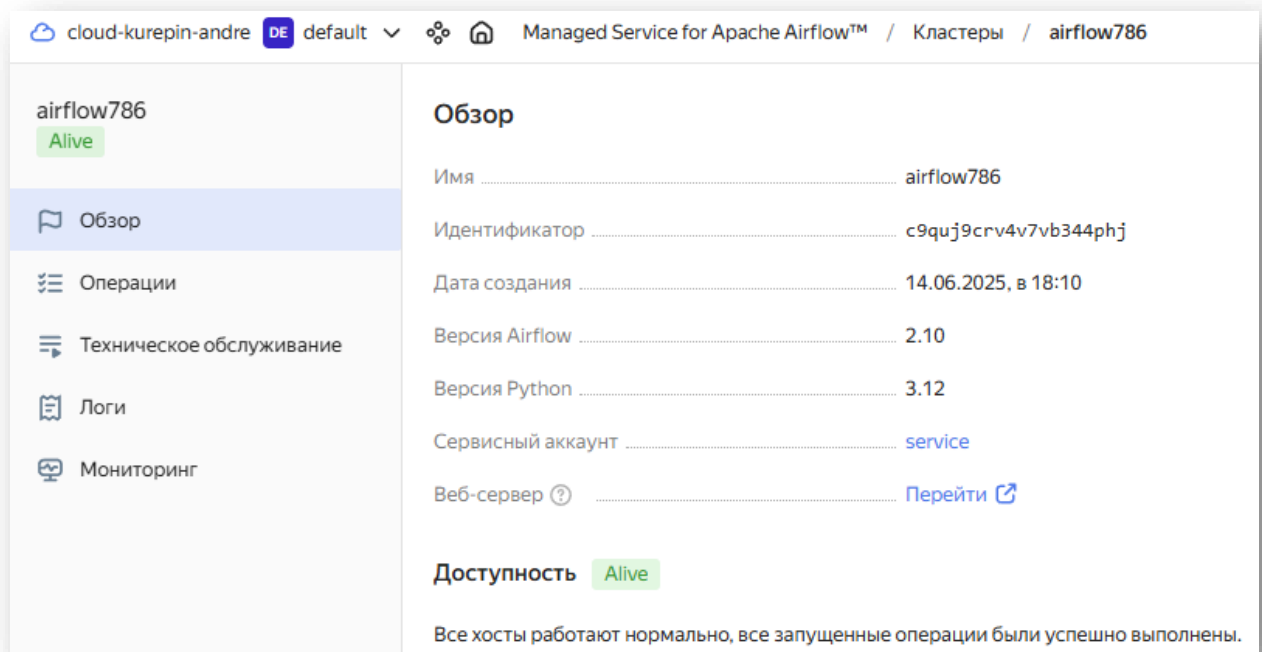
Объекты

Поиск по началу имени объекта, включая префикс ? ☐ Показать версии

<input type="checkbox"/> Имя	Размер	Класс хранилища	Последнее изменение	Метки	
<input type="checkbox"/> DP-DAG.py	3.12 КБ	Стандартное	14.06.2025, в 19:37	—	...
<input type="checkbox"/> DP-fraud-DAG.py	2.47 КБ	Стандартное	15.06.2025, в 16:34	—	...

Рисунок 12 DAG скрипт

Поднимем кластер Airflow, Рисунок 13. Откроем admin панель Airflow, в списке DAG-ов отображается my-final-hw-dag Рисунок 14.



cloud-kurepin-andre DE default Managed Service for Apache Airflow™ / Кластеры / airflow786

airflow786

Alive

Обзор

Операции

Техническое обслуживание

Логи

Мониторинг

### Обзор

Имя ..... airflow786

Идентификатор ..... c9quj9crv4v7vb344phj

Дата создания ..... 14.06.2025, в 18:10

Версия Airflow ..... 2.10

Версия Python ..... 3.12

Сервисный аккаунт ..... [service](#)

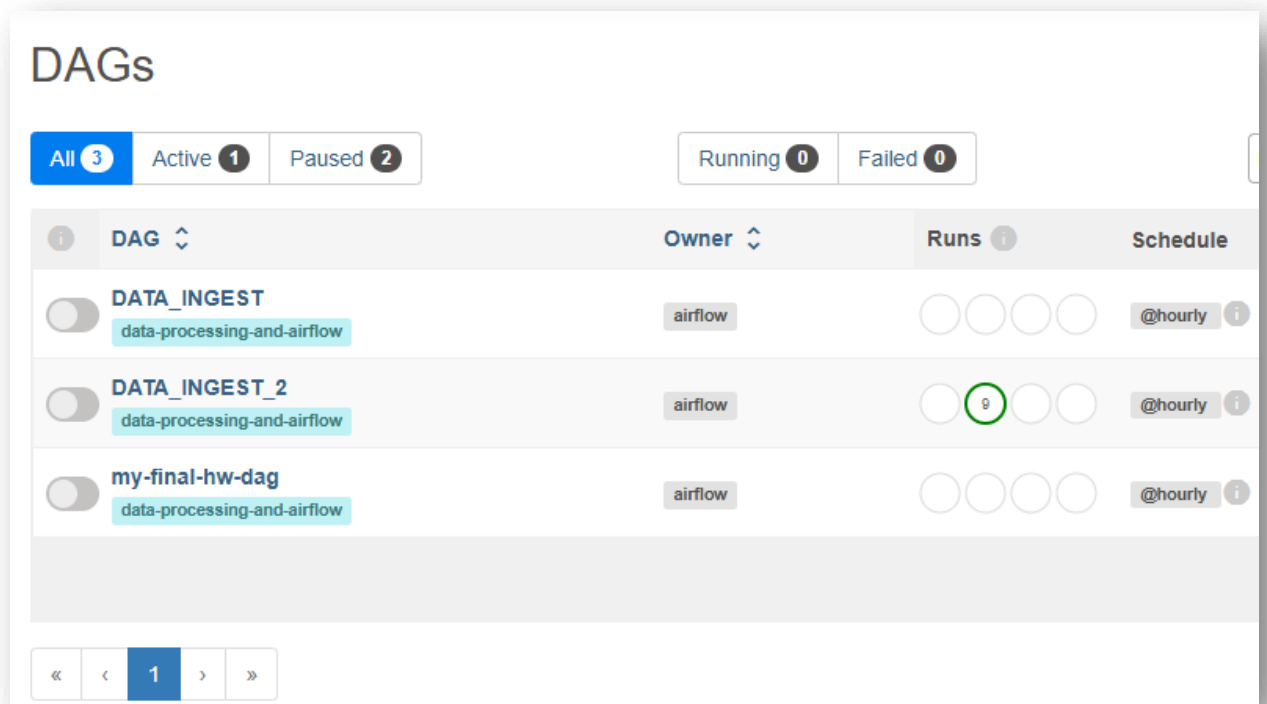
Веб-сервер ? ..... [Перейти](#)

**Доступность** Alive

Все хосты работают нормально, все запущенные операции были успешно выполнены.

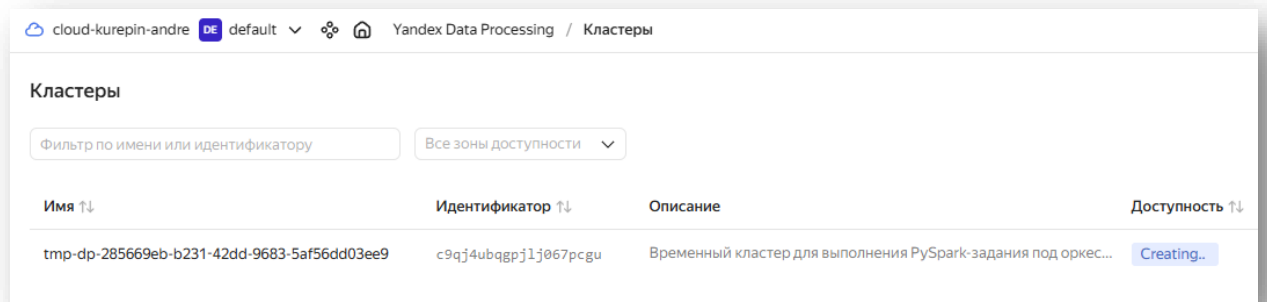
Рисунок 13 Кластер Airflow





**Рисунок 14** Веб интерфейс Airflow

Запустим данный DAG, на первом шаге будет создан вычислительный кластер, Рисунок 15. Успешное выполнение DAG, Рисунок 16. После успешного выполнения DAG сформировал файл, Рисунок 17-18.



**Рисунок 15** Создание вычислительного кластера

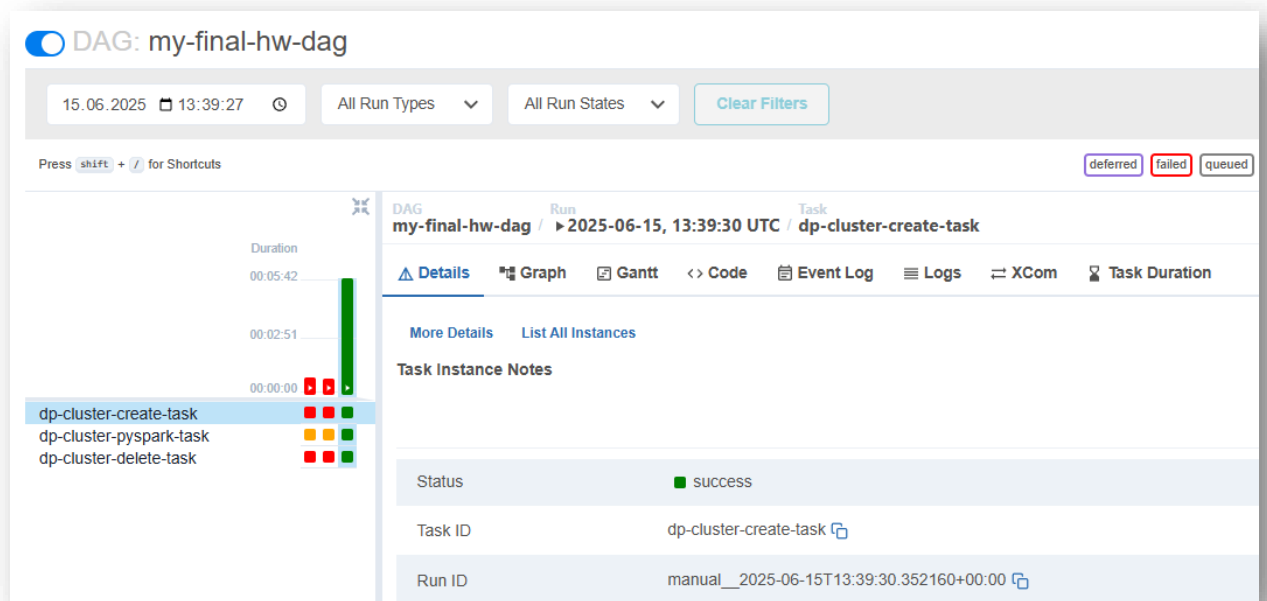


Рисунок 16 Успешное выполнение DAG-а

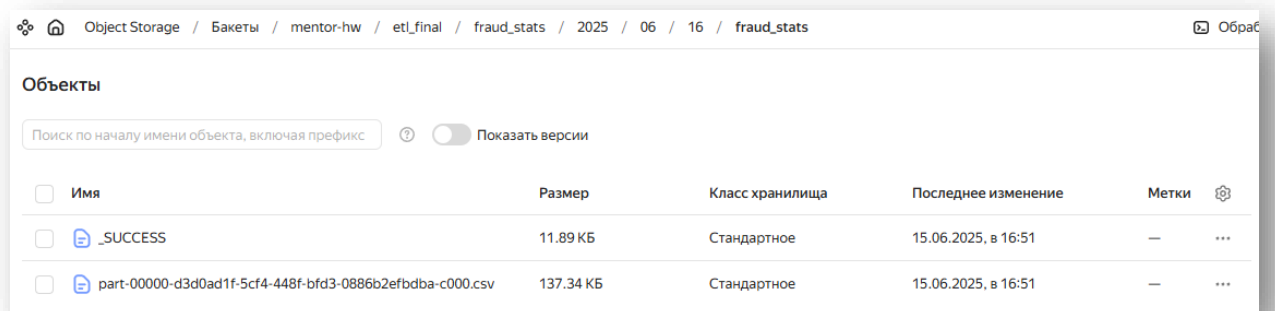


Рисунок 17 Результаты работы DAG-а

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	user_id,fraud_transactions_sent,total_fraud_amount_sent													
2	2866,45,-34954													
3	1959,53,-18992													
4	833,51,-6388													
5	7554,47,-27983													
6	4935,51,105294													
7	6466,53,91805													
8	3794,45,63160													
9	463,38,33759													
10	496,50,9444													
11	7754,51,8004													
12	6357,54,46124													
13	1342,57,-20012													
14	1591,50,-3553													

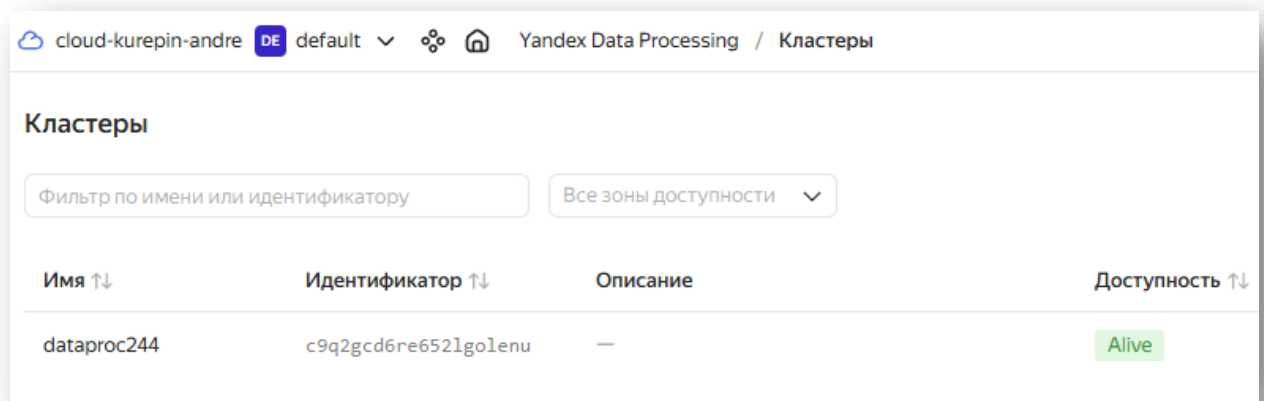
Рисунок 18 Полученный файл

### Задание 3

Для выполнения задания создадим кластер Kafka и кластер Data processing, Рисунки 19 и 20.

Имя	Идентификатор	Описание	Доступность
kafka171	c9qo2fieah5u8va2ub9p	—	Alive

Рисунок 19 Созданный кластер Kafka

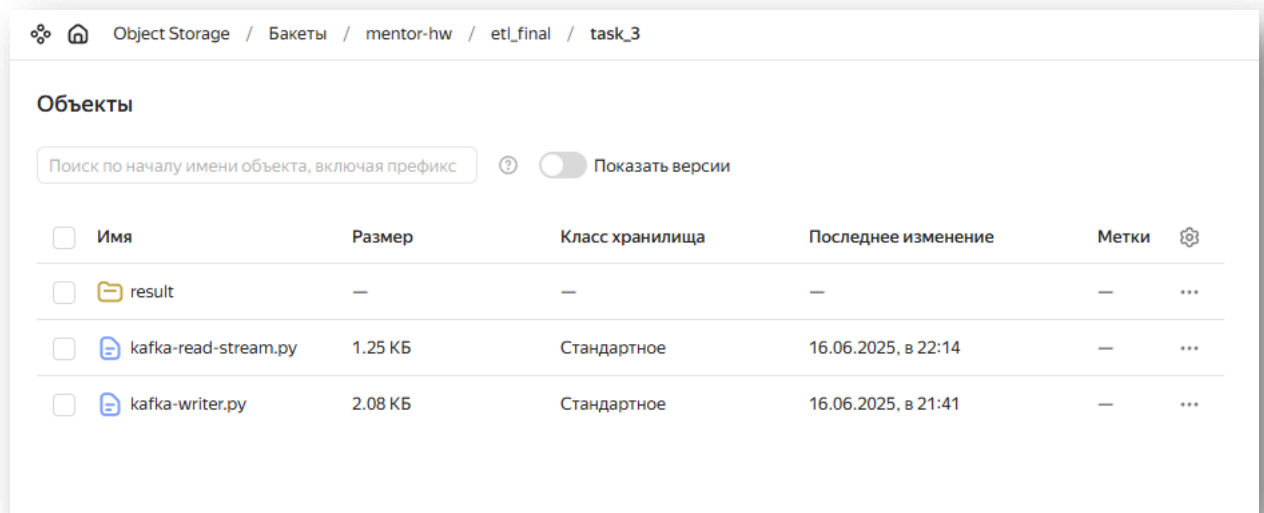


**Рисунок 20** Созданный кластер Data processing

Разработаем Python-скрипт *kafka-writer.py* для записи данных в топик кафки. Скрипт записывает в топик информация о транзакциях, Рисунок 21.

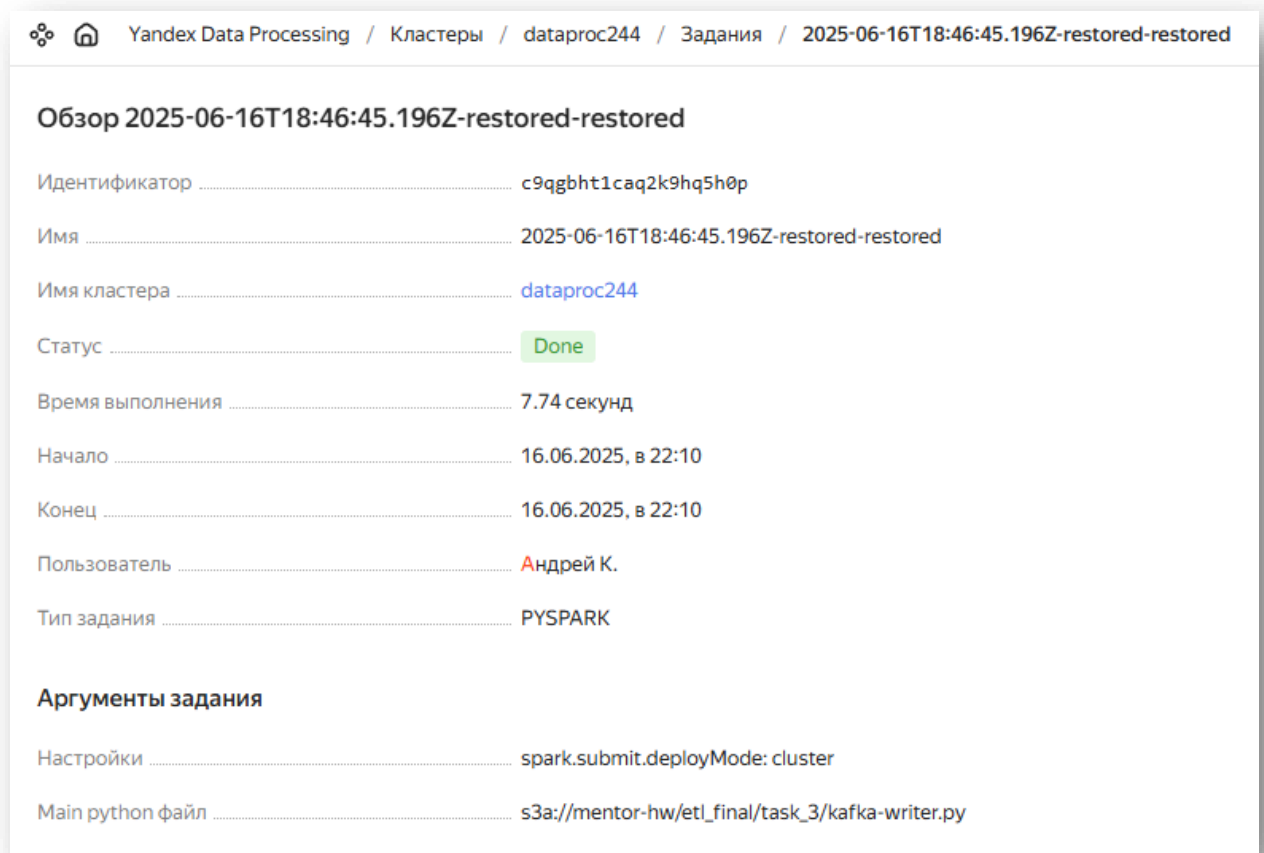
Разработаем Python-скрипт *kafka-read-stream.py* для потокового чтения из топика кафки. Скрипт считывает информацию из топика и записывает ее в Storage, Рисунок 21.

Полный код представлен в файлах *kafka-writer.py* и *kafka-read-stream.py*.



**Рисунок 21** Скрипты для запись/чтения в/из топика кафки

Создадим задание в кластере DP для записи данных в топик и запустим его, результат на Рисунке 22.



The screenshot displays the Yandex Data Processing console interface. At the top, the breadcrumb navigation shows the path: Yandex Data Processing / Кластеры / dataproc244 / Задания / 2025-06-16T18:46:45.196Z-restored-restored. Below this, the title 'Обзор 2025-06-16T18:46:45.196Z-restored-restored' is followed by a list of task details. The details include the identifier 'c9qgbht1caq2k9hq5h0p', the name '2025-06-16T18:46:45.196Z-restored-restored', the cluster name 'dataproc244', and a status of 'Done' indicated by a green box. Other details include a completion time of 7.74 seconds, start and end times of 16.06.2025 at 22:10, the user 'Андрей К.', and the task type 'PYSARK'. A section titled 'Аргументы задания' (Task Arguments) lists the configuration 'spark.submit.deployMode: cluster' and the main Python file 's3a://mentor-hw/etl\_final/task\_3/kafka-writer.py'.

Идентификатор	c9qgbht1caq2k9hq5h0p
Имя	2025-06-16T18:46:45.196Z-restored-restored
Имя кластера	dataproc244
Статус	Done
Время выполнения	7.74 секунд
Начало	16.06.2025, в 22:10
Конец	16.06.2025, в 22:10
Пользователь	Андрей К.
Тип задания	PYSARK

**Аргументы задания**

Настройки	spark.submit.deployMode: cluster
Main python файл	s3a://mentor-hw/etl_final/task_3/kafka-writer.py

**Рисунок 22** Успешное выполнение задания на запись в топик

Создадим задание в кластере DP для чтения данных из топика и запустим его, результат на Рисунке 23.

Yandex Data Processing / Кластеры / dataproc244 / Задания / 2025-06-15T18:35:25.607Z-restored-restored-restored

Обзор 2025-06-15T18:35:25.607Z-restored-restored-restored

Идентификатор

c9qk1cu0a66s6i65h1oh

Имя

2025-06-15T18:35:25.607Z-restored-restored-restored

Имя кластера

dataproc244

Статус

Done

Время выполнения

15.98 секунд

Начало

16.06.2025, в 22:14

Конец

16.06.2025, в 22:15

Пользователь

Андрей К.

Тип задания

PYSPARK

Аргументы задания

Настройки

spark.submit.deployMode: cluster

Main python файл

s3a://mentor-hw/etl\_final/task\_3/kafka-read-stream.py

Рисунок 23 Успешное выполнение задания на чтение из топика

В Storage успешно записались данные, Рисунок 24 и 25.

Object Storage / Бакеты / mentor-hw / etl\_final / task\_3 / result

Объекты

Поиск по началу имени объекта, включая префикс

Показать версии



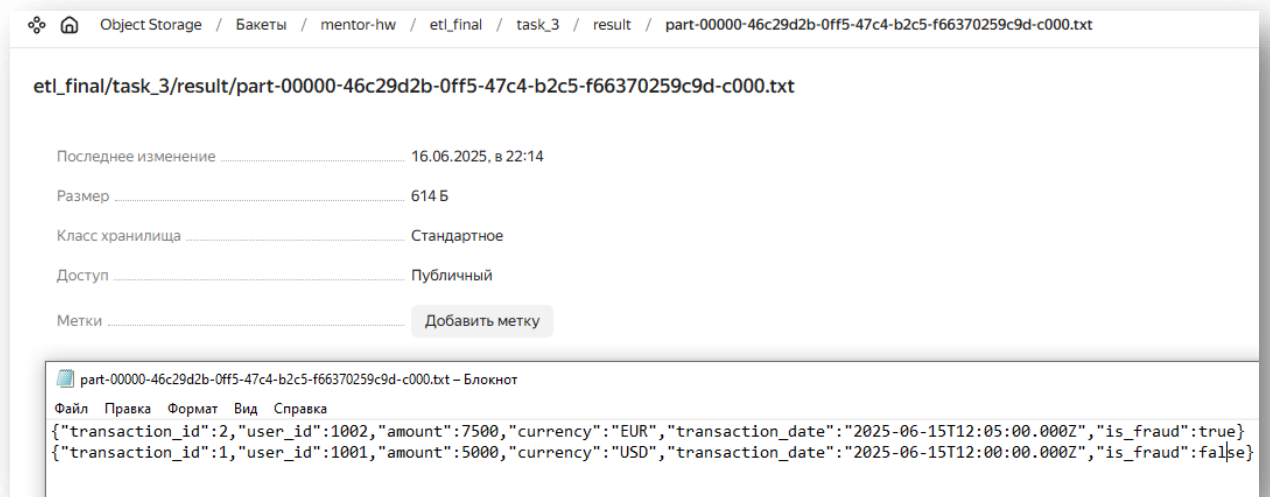
<input type="checkbox"/> Имя	Размер	Класс хранилища	Последнее изменение
<input type="checkbox"/>  _SUCCESS	0 Б	Стандартное	16.06.2025, в 22:14
<input type="checkbox"/>  part-00000-46c29d2b-0ff5-47c4-b2c5-f66370259c9d-c000.txt	614 Б	Стандартное	16.06.2025, в 22:14

Рисунок 24 Полученный файл



**Рисунок 25** Полученный файл