

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Самарский национальный исследовательский университет  
имени академика С.П. Королева»

Институт информатики и кибернетики  
Кафедра технической кибернетики

Лабораторная работа № 1  
по курсу «Инженерия данных»

Студент: Бурлаков А.Е.  
Группа: 6231-010402D

Самара 2025

В ходе выполнения данной лабораторной работы были использованы следующие инструменты:

- Perfect – управление пайплайна,
- MinIO – хранение необработанных данных,
- ClickHouse – база данных.

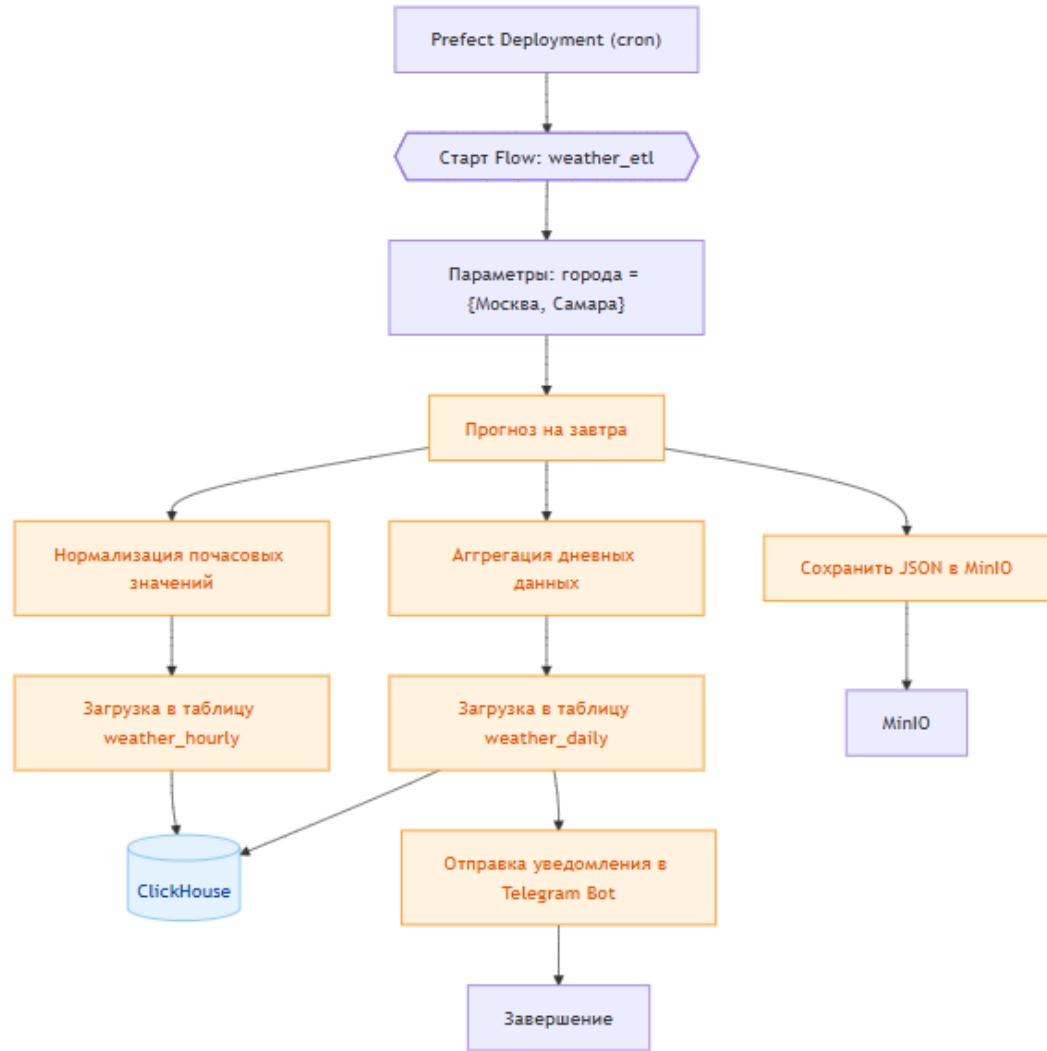


Рисунок 1. Пайплайн работы.

Источник данных.

В качестве источника данных использовался публичный сервис Open-Meteo API, предоставляющий почасовые и дневные прогнозы погоды по заданным географическим координатам.

Из API запрашивались параметры: температура воздуха, количество осадков, скорость ветра, направление ветра. Прогноз формировался для двух городов: Москва и Самара.

Разработанный пайплайн реализован в виде ETL-процесса и включает следующие компоненты:

1. Prefect — оркестрация задач и управление выполнением пайплайна.
2. MinIO — объектное хранилище для сохранения сырых (raw) JSON-данных.
3. ClickHouse — аналитическая база данных для хранения структурированных и агрегированных данных.
4. Telegram Bot API — канал оповещения пользователя о прогнозе погоды.

Для работы с ClickHouse реализована функция `ch_client()`, которая возвращает подключение к базе данных.

Перед запуском пайплайна создаются две таблицы: `weather_hourly` — для хранения почасовых значений погоды и `weather_daily` — для хранения агрегированных дневных показателей. Функция `fetch_forecast()` выполняет запрос к Open-Meteo API, задавая координаты городов и формируя HTTP-запрос с параметрами прогноза. Для каждого города создаётся отдельный в хранилище MinIO.

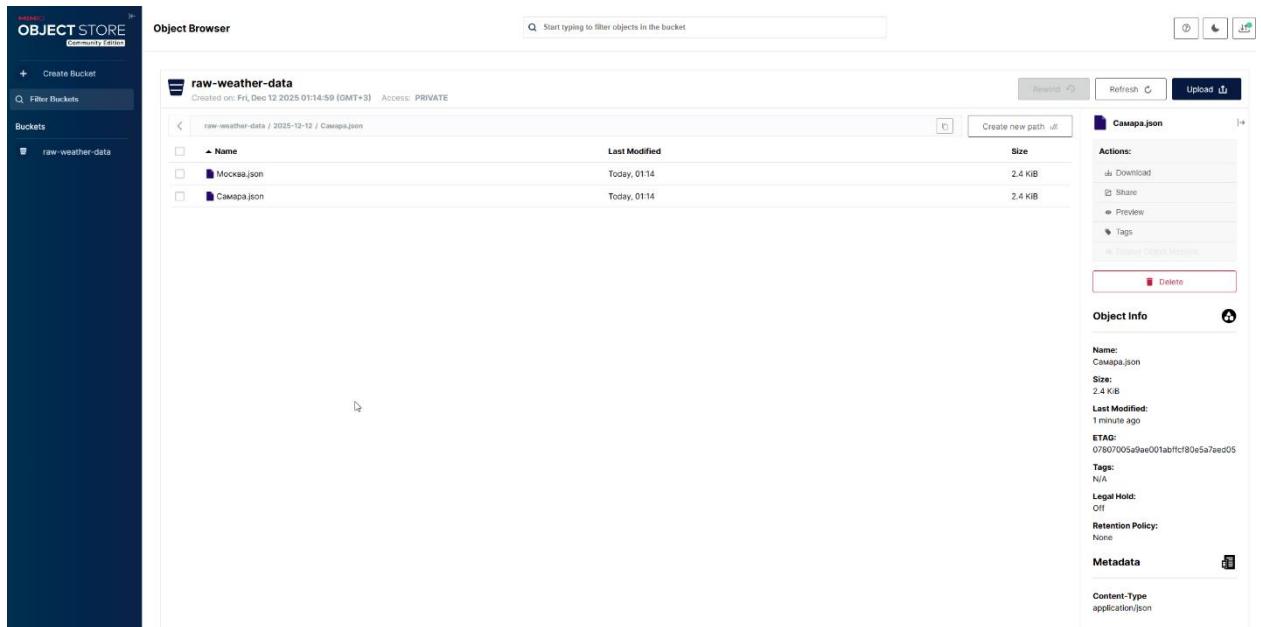


Рисунок 2. Объекты в MinIO.

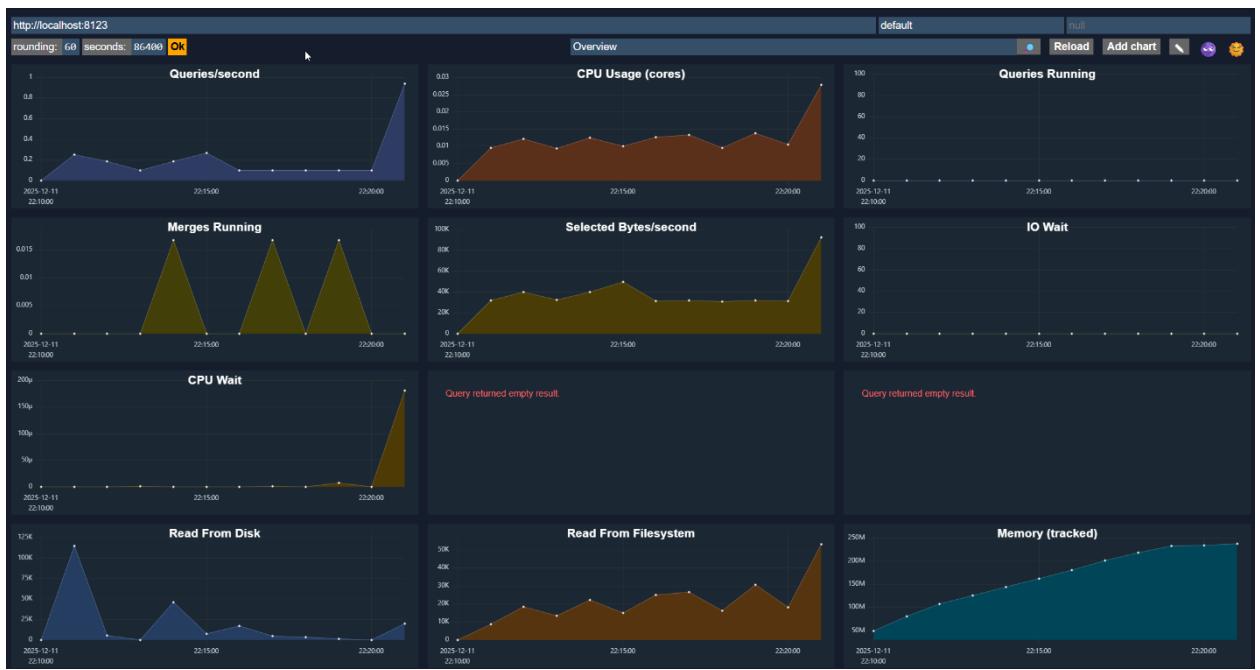


Рисунок 3. Данные с Open-Meteo-API.

Функция `build_hourly()` выполняет обработку сырых данных и формирует pandas.DataFrame с почасовыми значениями погоды только за завтрашний день. В таблицу включаются: город, временная метка, температура, осадки, скорость и направление ветра. После сформированный DataFrame загружается в таблицу базы данных ClickHouse.

#	city	timestamp	temperature	precipitation	wind_speed	wind_direction
1	Москва	2025-12-13 00:00:00	-5	0	15.6	341
2	Москва	2025-12-13 01:00:00	-5	0	15.7	340
3	Москва	2025-12-13 02:00:00	-5	0	18.7	332
4	Москва	2025-12-13 03:00:00	-5	0	18.8	331
5	Москва	2025-12-13 04:00:00	-5	0	15	325
6	Москва	2025-12-13 05:00:00	-5.1	0	15.2	324
7	Москва	2025-12-13 06:00:00	-5.4	0	16.5	328
8	Москва	2025-12-13 07:00:00	-5.7	0	15.3	319
9	Москва	2025-12-13 08:00:00	-5.9	0	15.4	323
10	Москва	2025-12-13 09:00:00	-6.2	0	16.2	323
11	Москва	2025-12-13 10:00:00	-6.2	0	15.3	321
12	Москва	2025-12-13 11:00:00	-6.2	0	15.6	320
13	Москва	2025-12-13 12:00:00	-6.1	0	15.8	319
14	Москва	2025-12-13 13:00:00	-6	0	15.5	318
15	Москва	2025-12-13 14:00:00	-5.6	0	16	316
16	Москва	2025-12-13 15:00:00	-5.4	0	15.3	315
17	Москва	2025-12-13 16:00:00	-5.4	0	16.1	318
18	Москва	2025-12-13 17:00:00	-5.5	0	14.8	315
19	Москва	2025-12-13 18:00:00	-5.7	0	13	309
20	Москва	2025-12-13 19:00:00	-5.8	0	13.2	313
21	Москва	2025-12-13 20:00:00	-5.9	0	13.2	315
22	Москва	2025-12-13 21:00:00	-5.9	0	12.5	314
23	Москва	2025-12-13 22:00:00	-6.1	0	12.5	309
24	Москва	2025-12-13 23:00:00	-6.1	0	12.8	310
25	Самара	2025-12-13 00:00:00	-0.6	0.2	8.4	211
26	Самара	2025-12-13 01:00:00	-0.5	0	8.6	213
27	Самара	2025-12-13 02:00:00	-0.5	0	7.6	199
28	Самара	2025-12-13 03:00:00	-0.4	0	8.4	200
29	Самара	2025-12-13 04:00:00	-0.4	0	9.4	198
30	Самара	2025-12-13 05:00:00	-0.4	0	9.4	198
31	Самара	2025-12-13 06:00:00	-0.3	0	8.5	192

Рисунок 4. База данных в ClickHouse.

The screenshot shows a terminal window for ClickHouse. At the top, it displays the URL `http://localhost:8123`, the query `SELECT * FROM weather_daily`, the ClickHouse version `v25.6.13.41, uptime 15 min`, the database `default`, and the password prompt. Below the query, the results are shown in a table:

#	city	date	temp_min	temp_max	temp_avg	precip_total
1	Москва	2025-12-13	-6.2	-5	-5.633333	0
2	Самара	2025-12-13	-7.6	-0.2	-2.5916667	14.1

At the bottom right of the terminal, there is a status message: `100.0%, Read 2 rows, 78.00 B`.

Рисунок 5. Обработанные данные.

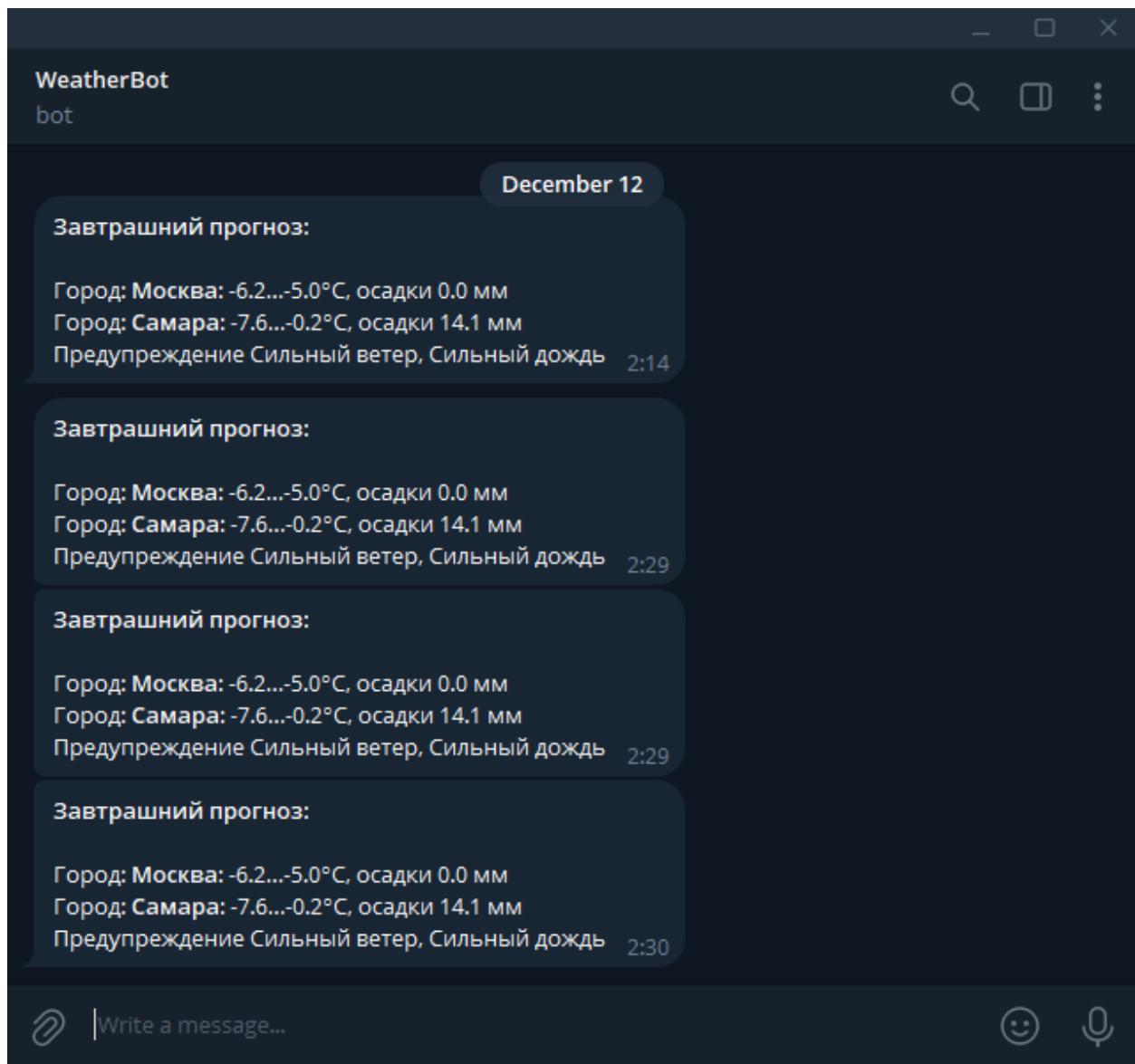


Рисунок 6. Автоматическая отправка уведомления в Telegram с кратким прогнозом на завтра.

```

PS C:\Users\engut\IdeaScripts\lab1> & C:/Users/engut/AppData/Local/Programs/Python/Python313/python.exe c:/Users/engut/IdeaScripts/lab1/weather_pipeline.py
01:29:32.658 | INFO | prefect - Starting temporary server on http://127.0.0.1:8693
See https://docs.prefect.io/v3/concepts/server#how-to-guides for more information on running a dedicated Prefect server.
Your flow 'Weather Pipeline' is being served and polling for scheduled runs!

To trigger a run for this flow, use the following command:

$ prefect deployment run 'Weather Pipeline/weather-job'

01:29:48.686 | INFO | prefect.flow_runs.runner - Runner 'weather-job' submitting flow run '7a0059e3-6b38-450f-bf60-5a8a7d6208be'
01:29:50.807 | INFO | prefect.flow_runs.runner - Completed submission of flow run '7a0059e3-6b38-450f-bf60-5a8a7d6208be'
01:29:52.427 | INFO | prefect - Starting temporary server on http://127.0.0.1:8112
See https://docs.prefect.io/v3/concepts/server#how-to-guides for more information on running a dedicated Prefect server.
01:29:57.534 | INFO | Flow run 'smart-buzzard' - Beginning flow run 'smart-buzzard' for flow 'Weather Pipeline'
01:29:58.358 | INFO | Task run 'Получение прогноза на завтра-e64' - Загружено данных: 2 городов.
01:29:58.362 | INFO | Task run 'Получение прогноза на завтра-e64' - Finished in state Completed()
01:29:58.659 | INFO | Task run 'Сохранение исходников в MinIO-28d' - MinIO saved: 2025-12-12/Москва.json
01:29:58.703 | INFO | Task run 'Сохранение исходников в MinIO-28d' - MinIO saved: 2025-12-12/Самара.json
01:29:58.705 | INFO | Task run 'Сохранение исходников в MinIO-28d' - Finished in state Completed()
01:29:58.956 | INFO | Task run 'Формирование почасового DataFrame-5cf' - Hour DF built: 48 rows
01:29:58.957 | INFO | Task run 'Формирование почасового DataFrame-5cf' - Finished in state Completed()
01:29:59.189 | INFO | Task run 'Загрузка почасовых значений в ClickHouse-7a8' - Finished in state Completed()
01:29:59.406 | INFO | Task run 'Агрегация по дням-24d' - Daily aggregated: 2 records
01:29:59.408 | INFO | Task run 'Агрегация по дням-24d' - Finished in state Completed()
01:29:59.659 | INFO | Task run 'Загрузка дневной статистики в ClickHouse-468' - Finished in state Completed()
01:30:00.395 | INFO | Task run 'Отправка прогноза в Telegram-6aa' - Finished in state Completed()
01:30:00.415 | INFO | Flow run 'smart-buzzard' - Finished in state Completed()
01:30:00.427 | INFO | prefect - Stopping temporary server on http://127.0.0.1:8112
01:30:00.930 | INFO | prefect.flow_runs.runner - Process for flow run 'smart-buzzard' exited cleanly.
01:30:01.560 | INFO | prefect.flow_runs.runner - Runner 'weather-job' submitting flow run 'd5a633ac-fc22-484a-845a-241e5b5e2615'
01:30:03.759 | INFO | prefect.flow_runs.runner - Completed submission of flow run 'd5a633ac-fc22-484a-845a-241e5b5e2615'
01:30:05.468 | INFO | prefect - Starting temporary server on http://127.0.0.1:8010
See https://docs.prefect.io/v3/concepts/server#how-to-guides for more information on running a dedicated Prefect server.
01:30:10.035 | INFO | Flow run 'stereotyped-oarfish' - Beginning flow run 'stereotyped-oarfish' for flow 'Weather Pipeline'
01:30:10.809 | INFO | Task run 'Получение прогноза на завтра-015' - Загружено данных: 2 городов.
01:30:10.812 | INFO | Task run 'Получение прогноза на завтра-015' - Finished in state Completed()
01:30:11.084 | INFO | Task run 'Сохранение исходников в MinIO-9ef' - MinIO saved: 2025-12-12/Москва.json
01:30:11.120 | INFO | Task run 'Сохранение исходников в MinIO-9ef' - MinIO saved: 2025-12-12/Самара.json
01:30:11.123 | INFO | Task run 'Сохранение исходников в MinIO-9ef' - Finished in state Completed()
01:30:11.337 | INFO | Task run 'Формирование почасового DataFrame-5cf' - Hour DF built: 48 rows
01:30:11.339 | INFO | Task run 'Формирование почасового DataFrame-5cf' - Finished in state Completed()
01:30:11.578 | INFO | Task run 'Загрузка почасовых значений в ClickHouse-4df' - Finished in state Completed()
01:30:11.791 | INFO | Task run 'Агрегация по дням-557' - Daily aggregated: 2 records
01:30:11.793 | INFO | Task run 'Агрегация по дням-557' - Finished in state Completed()
01:30:12.023 | INFO | Task run 'Загрузка дневной статистики в ClickHouse-48b' - Finished in state Completed()
01:30:12.681 | INFO | Task run 'Отправка прогноза в Telegram-84c' - Finished in state Completed()
01:30:12.712 | INFO | Flow run 'stereotyped-oarfish' - Finished in state Completed()
01:30:12.720 | INFO | prefect - Stopping temporary server on http://127.0.0.1:8010
01:30:13.159 | INFO | prefect.flow_runs.runner - Process for flow run 'stereotyped-oarfish' exited cleanly.
01:31:02.937 | INFO | prefect.flow_runs.runner - Runner 'weather-job' submitting flow run 'dfbbc473-ad33-4b7a-b642-bc9de3fc6848'
01:31:05.401 | INFO | prefect.flow_runs.runner - Completed submission of flow run 'dfbbc473-ad33-4b7a-b642-bc9de3fc6848'

```

Рисунок 7. Логирование из Perfect.

## Вывод.

В рамках лабораторной работы была спроектирована и реализована система автоматизированной обработки метеорологических данных, основанная на принципах ETL. Реализованный процесс охватывает получение прогноза погоды из внешнего источника Open-Meteo, сохранение исходных данных в объектном хранилище MinIO, преобразование и структурирование почасовых значений, вычисление агрегированных показателей за сутки и последующую загрузку результатов в аналитическую базу данных ClickHouse. Интеграция и управление этапами обработки выполнены с использованием оркестратора Prefect, что обеспечило контроль выполнения задач, ведение журналов и повторное выполнение операций при возникновении ошибок. Создание схем хранения данных в ClickHouse выполняется автоматически на этапе инициализации пайплайна.

Наибольшие затруднения в процессе выполнения работы были обусловлены необходимостью освоения новых технологий и инструментов, включая Docker, Prefect, MinIO и ClickHouse, а также особенностями обработки ответов внешнего API, содержащих неполные данные.