

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное
образовательное учреждение высшего образования
«Самарский национальный исследовательский университет
имени академика С.П. Королева»
(Самарский университет)

Институт информатики и кибернетики

Отчет по лабораторной работе №1

Дисциплина: «Инженерия данных»

Выполнил: Морозов Д. А

Группа: 6233-010402D

Самара 2025

1. Кратко об архитектуре: схема пайплайна, какие инструменты и почему.

Пайплайн реализован в виде ETL-процесса с использованием оркестратора Prefect, который управляет выполнением задач, логированием и обработкой ошибок. Архитектура построена по модульному принципу и включает этапы извлечения данных, трансформации и загрузки результатов в аналитическое хранилище.

В качестве источника данных используется публичный API Open-Meteo, предоставляющий погодные прогнозы. Сырые ответы API сохраняются в MinIO, который используется как объектное хранилище для хранения неизменённых данных и обеспечения воспроизводимости обработки.

Когда я создавал свои тг. Решения, то не использовал подробный инструментарий. Поработал и понял:

1. Очень удобно хранить исходники в Метео (Я возможно начну с ним работать)
2. Perfect по сути работу упрощает, сокращает код, которые нужно написать + организует бизнес-логику (точнее следует ей)
3. Кликхаус единственной проблемой стал – я потратил 2 дня на то, чтобы отчет сформировать из-за постоянных проблем с доступом/несоответствием полей (такое вообще впервые вижу())

2. Источник данных: эндпоинт Open-Meteo, параметры запроса.

В качестве источника данных используется публичный погодный API Open-Meteo, удобно можно забирать данные необходимые (С парсингом новостей +- так же, взял по вашему совету)

Эндпоинт - <https://api.open-meteo.com/v1/forecast>

Параметры запроса:

1. latitude, longitude — координаты города
2. timezone — временная зона города
3. start_date, end_date — дата прогноза (завтра)
4. hourly — список почасовых показателей

3. Extract → Transform → Load: по 2–3 предложения на этап.

1) Extract

На этапе Extract = запрос к API Open-Meteo для получения прогноза погоды на следующий день по Самаре+Москва. Запрос формируется с указанием координат города, временной зоны и списка почасовых метеопараметров (вот это подсмотрел). Полученный JSON-ответ сохраняется в объектное хранилище MinIO, сырой файл.

2) Transform

На этапе Transform почасовые данные нормализуются в табличный формат: извлекаются временные метки, температура, осадки, скорость и направление ветра. Дополнительно рассчитываются агрегированные дневные показатели: минимальная, максимальная и средняя температура, сумма осадков и максимальная скорость ветра. В

процессе трансформации выполняется базовая фильтрация некорректных данных (удаление записей без временной метки).

3) Load

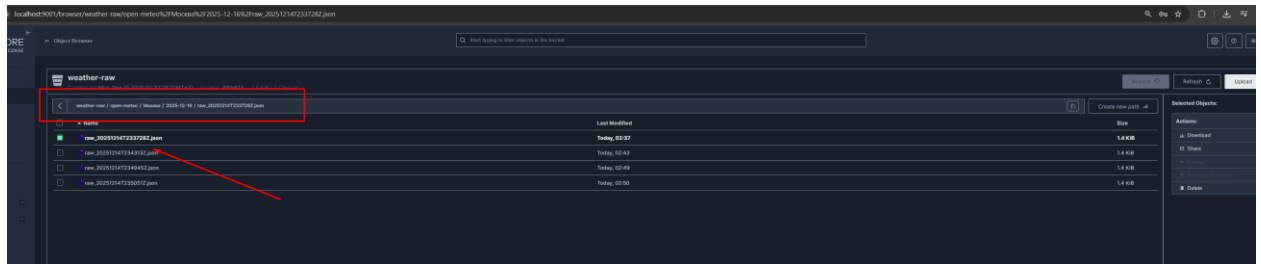
На этапе Load преобразованные данные загружаются в аналитическое хранилище ClickHouse. И оттуда с ними уже можно работать (Самый проблемный момент, опять же. Я, используя нейронку, три раза за 3 разных дня ПЕРЕПИСЫВАЛ все параметры хранилища)

4. Extract → Transform → Load: по 2–3 предложения на этап.

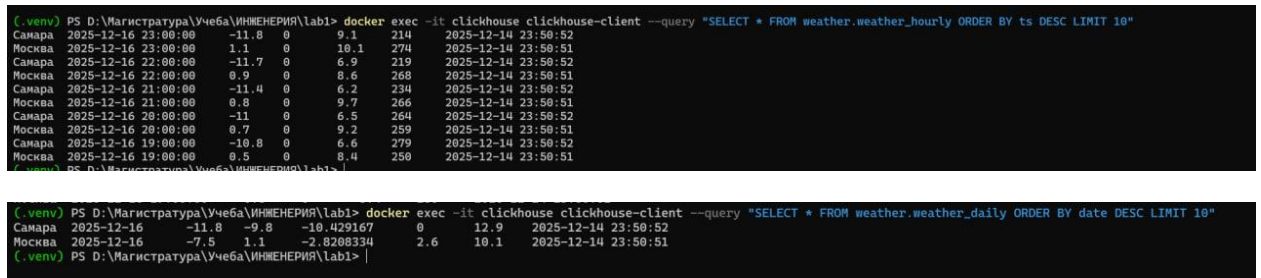
Контроль успешности HTTP-ответа источника данных (`raise_for_status`), фильтрация записей без временной метки при трансформации и сохранение сырых JSON-ответов в MinIO для трассировки и повторной обработки. Устойчивость пайплайна повышена за счёт повторных попыток выполнения задач Extract и Load в Prefect.

Основные точки сбоя: недоступность API Open-Meteo(Ни разу не было), ошибки подключения и прав доступа в ClickHouse (Главная проблема, ее не решил, ее по сути и не решить. Только избавиться от ошибок), а также несоответствие схемы данных таблицам хранилища. Наиболее проблемным этапом стала настройка хранилища и прав доступа, потребовавшая многократной переработки конфигурации.

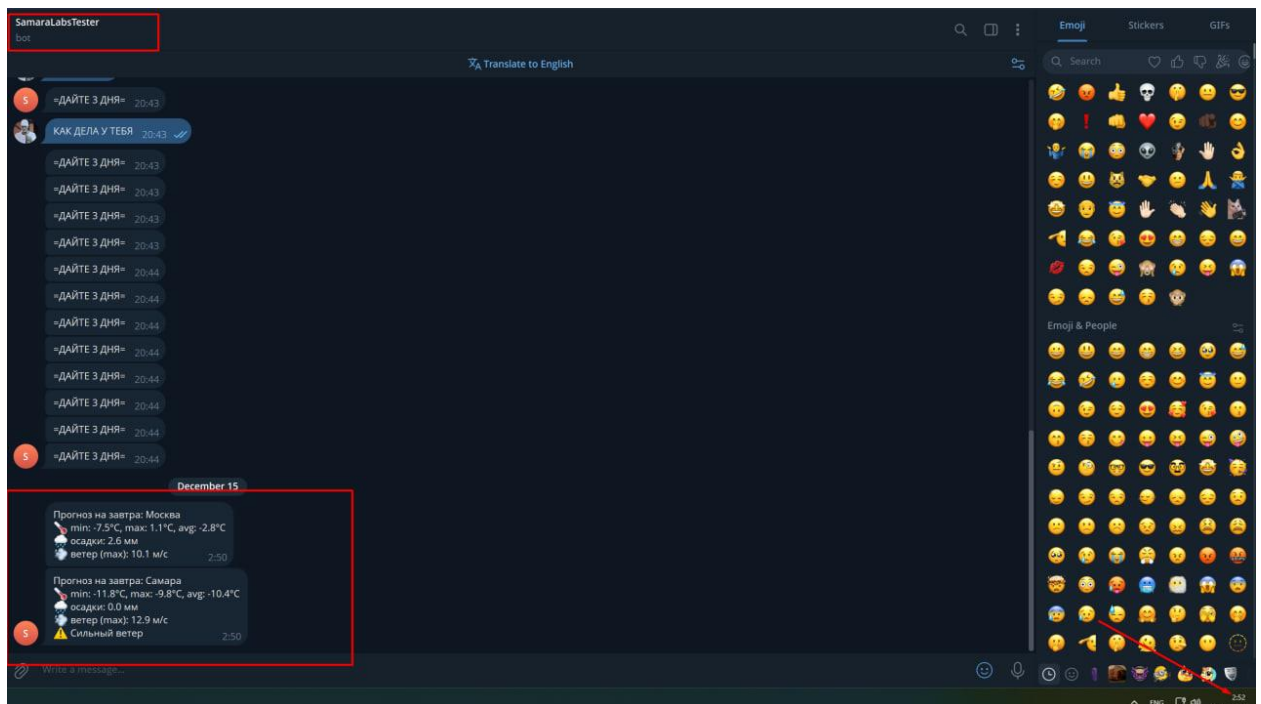
Рисунок с хранилищем мино



А так же уже сохраненные после в кликхаус данные в двух таблицах



Отправленная статистика в телеграмм (с датами)



6. Выводы: что было сложным, что бы улучшили.

По сути лабораторная очень прикладная и полезная, ничего сложного не было, единственное – кликхаус стал очень проблемным.

Если бы было время, я бы подумал над тем, как его ошибки обработать, т.к я, даже с ЧатГПТ и интернетом потратил часы на исправление банальных ошибок