

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, математики и электроники

Факультет информатики

Кафедра технической кибернетики

**Лабораторная работа №1
по дисциплине «Инженерия данных»**

Студент Соболев И.В.

Группа 6231-010402D

Преподаватель Парингер Р.А.

При выполнении лабораторной работы согласно заданию были использованы следующие инструменты:

- Prefect – для управления и выполнения ETL пайплайна,
- MinIO – для хранения необработанных данных,
- ClickHouse – для базы данных.

По заданию данной лабораторной работы было необходимо реализовать пайплайн изображенный на рисунке 1.

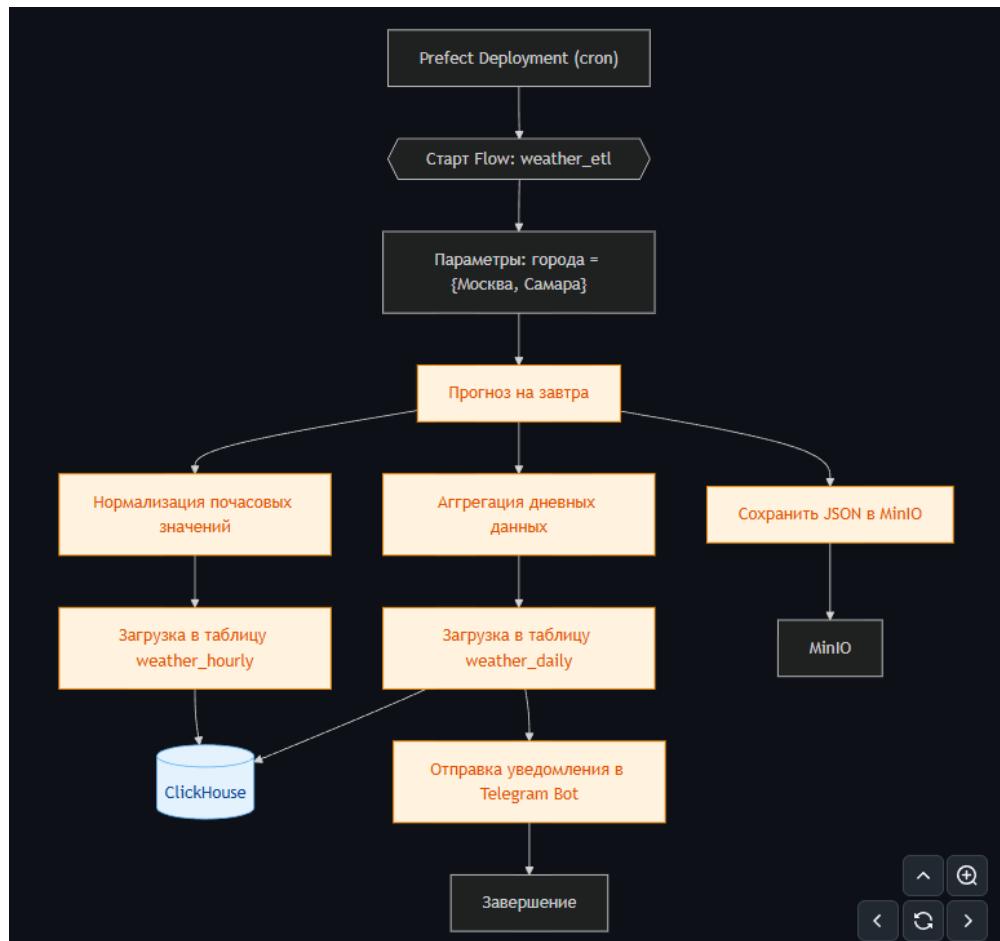


Рисунок 1 – ETL пайплайн

В качестве источника погодных данных был использован Open-Meteo API.

Параметры запроса представлены на рисунке 2.

```

params = {
    "latitude": lat,
    "longitude": lon,
    "daily": ["temperature_2m_max", "temperature_2m_min", "precipitation_sum", "windspeed_10m_max"],
    "hourly": ["temperature_2m", "precipitation", "windspeed_10m", "winddirection_10m"],
    "timezone": "Europe/Moscow",
    "start_date": TOMORROW,
    "end_date": TOMORROW,
}

```

Рисунок 2 – Параметры запроса

На рисунке 3 представлен flow в Prefect.

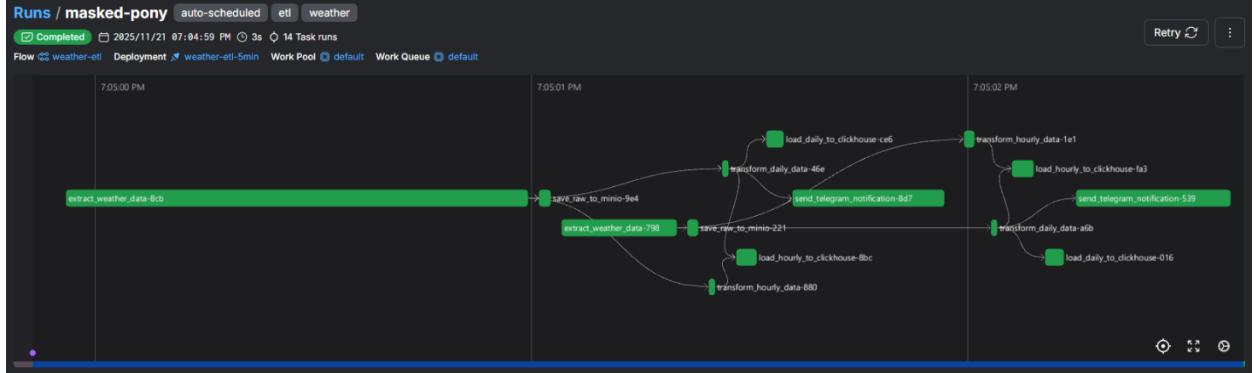


Рисунок 3 – Flow в Prefect

В работе были реализованы следующие ETL этапы:

Extract: Получение прогноза погоды на завтра по параметрам, указанным на рисунке 2, необработанные данные сохраняются в MinIO;

Transform: Подготовка данных (нормализация) для часовых прогнозов weather_hourly и агрегация данных для дневных прогнозов weather_daily;

Load: Загрузка преобразованных данных в таблицы ClickHouse;

Telegram: Отправка уведомления пользователю через Telegram Bot с прогнозом на завтра.

На рисунке 4 представлен результат сохранения данных в MinIO.

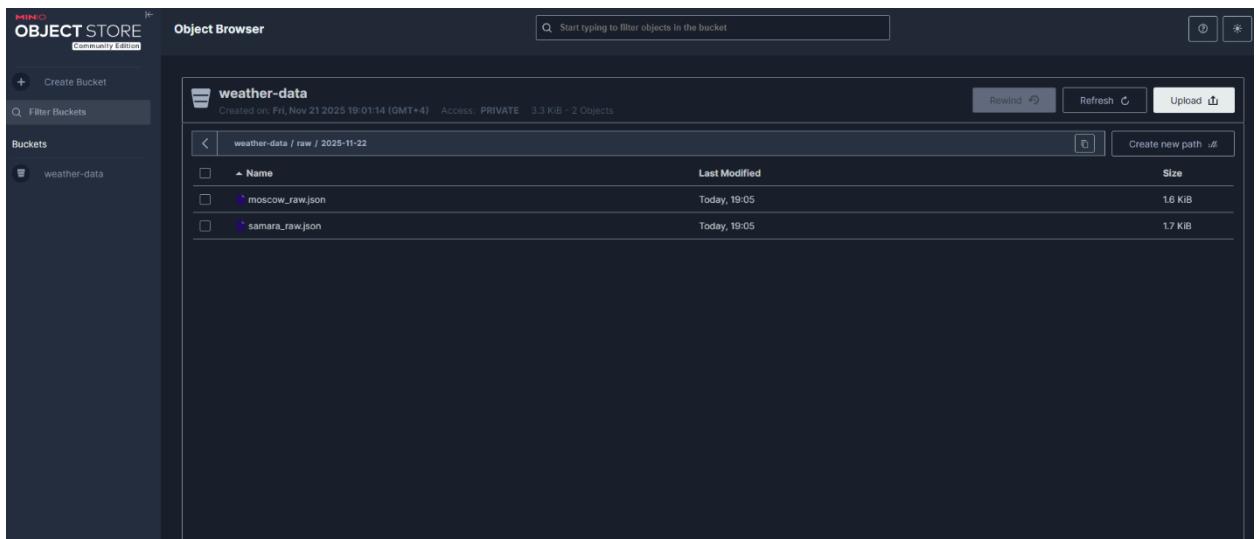


Рисунок 4 – Хранилище данных в MinIO

На рисунке 5 представлен результат отправки уведомления пользователю с помощью Telegram Bot.



Рисунок 5 – Уведомление в Telegram

Во время выполнения данной работы самым сложным оказалось следить за правильным использованием переменных окружения .env. Было

необходимо внимательно следить за корректностью вызовов этих переменных в docker-compose.yml. Также мне усложнили выполнение работы сервисы на моём компьютере, которые заняли стандартные и не самые стандартные порты, закрывая мне возможность легкой настройки инструментов по стоковым портам. Этую проблему было не так сложно решить поиском приложений по занятых портам.

Ввиду того, что я фанат усложнения задачи, я бы предложил улучшить Telegram бота путём добавления дополнительных взаимодействий с пользователем, например, помимо автоматической отправки уведомлений, я бы добавил возможность пользователю выбирать город для прогноза, а также выбирать на какой день или период необходим прогноз.