

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
Кафедра технической кибернетики

Отчет по лабораторной работе № 1

Дисциплина: «Инженерия данных»

Выполнил: Миролюбов В.Б.

Группа: 6233-010402D

Самара 2025

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

Схема пайплайна:

Prefect (оркестратор) → Open-Meteo API (источник) → MinIO (сырые данные) → ClickHouse (агрегированные таблицы) → Telegram Bot (уведомления).

Какие инструменты и почему:

- Prefect 3.x — для оркестрации и расписания (гибкость, UI, управление зависимостями).
- MinIO — S3-совместимое хранилище для сырых JSON-ответов (дешево, просто, локально).
- ClickHouse — колоночная СУБД для аналитики (быстрые агрегации, масштабируемость).
- Telegram Bot API — для push-уведомлений (удобно, бесплатно, современно).

Источник данных: эндпоинт Open-Meteo, параметры запроса.

Используется бесплатный API Open-Meteo (<https://open-meteo.com/>).

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

Параметры запроса:

- latitude, longitude — координаты Москвы и Самары, определены примерно.
- hourly=temperature_2m, precipitation, wind_speed_10m, wind_direction_10m — нужные метрики для анализа.
- forecast_days=2 — чтобы гарантированно получить завтрашний день.
- timezone=Europe/Moscow — для корректного времени.

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

Extract: Ежедневно в 23:15 Prefect запускает flow, который делает HTTP-запросы к Open-Meteo API для указанных городов. Сырые JSON-ответы сохраняются в MinIO для аудита и отладки.

Transform: Почасовые данные нормализуются в плоскую структуру для таблицы weather_hourly. Для weather_daily рассчитываются агрегаты: min/max/avg температура,

сумма осадков, и генерируются предупреждения при сильном ветре (>15 м/с) или осадках (>10 мм).

Load: Преобразованные данные загружаются в ClickHouse с помощью clickhouse-connect. Отправляется краткое текстовое уведомление в Telegram с прогнозом и алertsами.

Качество данных: какие проверки качества данных реализованы и возможные точки сбоя.

Проверки:

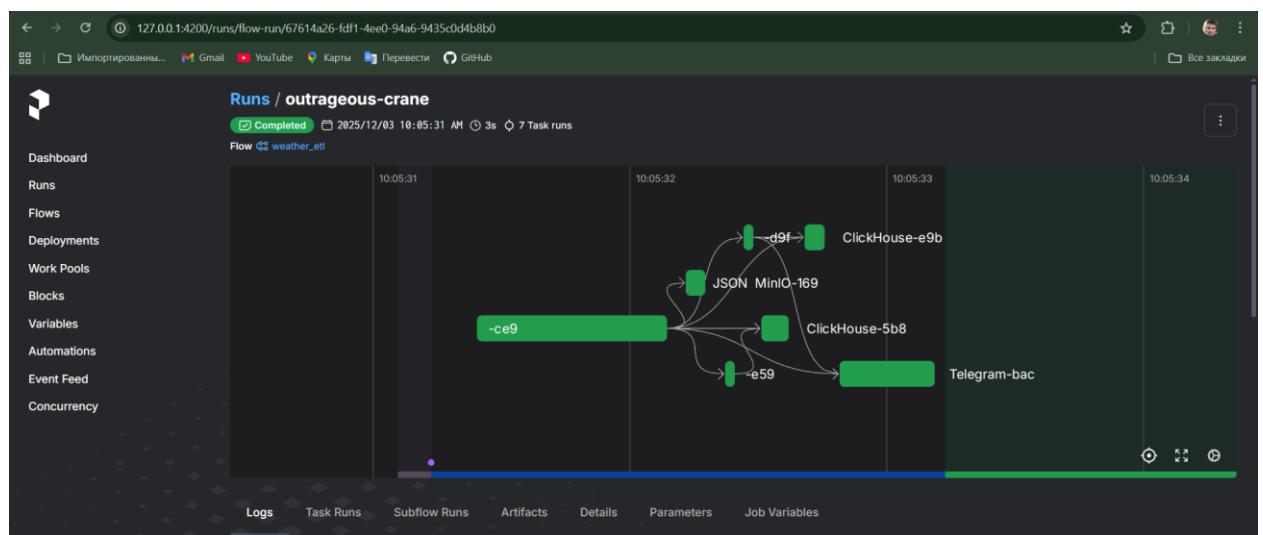
- Валидация HTTP-статуса (200) при запросе к API.
- Проверка наличия координат для города.
- Проверка, что для завтрашнего дня есть хотя бы одна почасовая запись.
- Обработка None значений при нормализации.

Точки сбоя:

- Недоступность Open-Meteo API (обрабатывается через retries=3 в Prefect).
- Неверные/отсутствующие секреты (MinIO, CH, Telegram) — хранятся в Prefect Blocks.

Результаты работы пайплайна:

1. Логи Prefect



```

Dec 3rd, 2025
INFO Beginning flow run 'outrageous-crane' for flow 'weather_etl'
INFO Запуск ETL пайплайна погоды
INFO Finished in state Completed()
INFO Сырые данные для Москва сохранены в MinIO: raw/Москва_20251203_100532.json
INFO Сырые данные для Самара сохранены в MinIO: raw/Самара_20251203_100532.json
INFO Finished in state Completed()
INFO Загружено 48 почасовых записей в ClickHouse
INFO 10:05:31 AM
prefect.flow_runs
INFO 10:05:31 AM
prefect.flow_runs
INFO 10:05:32 AM
Получить прогноз погоды
prefect.task_runs
INFO 10:05:32 AM
Сохранить JSON в MinIO-
prefect.task_runs
INFO 10:05:32 AM
Сохранить JSON в MinIO-
prefect.task_runs
INFO 10:05:32 AM
Сохранить JSON в MinIO-
prefect.task_runs
INFO 10:05:32 AM
Нормализовать почасовые
prefect.task_runs
INFO 10:05:32 AM
Агрегировать дневные даты
prefect.task_runs
INFO 10:05:32 AM
Загрузить почасовые данные
prefect.task_runs

```

```

INFO Finished in state Completed()
INFO 10:05:32 AM
prefect.task_runs
INFO Загружено 2 дневных записей в ClickHouse
INFO 10:05:32 AM
prefect.task_runs
INFO Finished in state Completed()
INFO 10:05:32 AM
prefect.task_runs
INFO Уведомление отправлено в Telegram
INFO 10:05:32 AM
Отправить уведомление
prefect.task_runs
INFO Finished in state Completed()
INFO 10:05:33 AM
prefect.task_runs
INFO Пайплайн завершён успешно
INFO Активация Windows
prefect.flow_runs
INFO Finished in state Completed()
INFO 10:05:33 AM
prefect.flow_runs
Чтобы активировать Windows, перейдите в раздел
"Параметры".
INFO 10:05:33 AM
prefect.flow_runs

```

Рисунки 1, 2, 3 – Логи работы Prefect

2. Содержимое бакета в MinIO

| Name | Last Modified | Size |
|-----------------------------|---------------|---------|
| Москва_20251203_100532.json | Today, 10:05 | 3.8 Kib |
| Самара_20251203_100532.json | Today, 10:05 | 3.8 Kib |

Рисунок 4 – Логи работы Prefect

3. Содержимое в Click

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query editor contains the following SQL code:

```
SELECT * FROM weather.weather_daily;
SELECT * FROM weather.weather_hourly;
```

The results grid, titled "Результат 1", displays the following data for the weather_daily table:

| | city | date | temp_min | temp_max | temp_avg | total_precipitation | wind_alert |
|---|--------|-----------------|----------|----------|------------|---------------------|------------|
| 1 | Москва | 2025-12-04 GMT+ | 0,3 | 2,8 | 1,6833333 | 0 | 0 |
| 2 | Самара | 2025-12-04 GMT+ | -0,3 | 1 | 0,34166667 | 0 | 0 |

Рисунок 5 – Содержимое таблицы weather_daily

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query editor contains the following SQL code:

```
SELECT * FROM weather.weather_hourly;
```

The results grid, titled "Результат 1", displays the following data for the weather_hourly table:

| | city | datetime | temperature | precipitation | wind_speed | wind_direction |
|----|--------|-----------------------|-------------|---------------|------------|----------------|
| 1 | Москва | 2025-12-04 00:00:00 G | 0,3 | 0 | 2,5 | 188 |
| 2 | Москва | 2025-12-04 01:00:00 G | 0,4 | 0 | 3,3 | 221 |
| 3 | Москва | 2025-12-04 02:00:00 G | 0,6 | 0 | 2,6 | 196 |
| 4 | Москва | 2025-12-04 03:00:00 G | 0,7 | 0 | 3,5 | 204 |
| 5 | Москва | 2025-12-04 04:00:00 G | 0,8 | 0 | 3,1 | 216 |
| 6 | Москва | 2025-12-04 05:00:00 G | 0,9 | 0 | 3,1 | 216 |
| 7 | Москва | 2025-12-04 06:00:00 G | 1 | 0 | 2,9 | 210 |
| 8 | Москва | 2025-12-04 07:00:00 G | 1,1 | 0 | 2,8 | 220 |
| 9 | Москва | 2025-12-04 08:00:00 G | 1,2 | 0 | 2,4 | 207 |
| 10 | Москва | 2025-12-04 09:00:00 G | 1,2 | 0 | 2,3 | 162 |
| 11 | Москва | 2025-12-04 10:00:00 G | 1,4 | 0 | 2,6 | 164 |
| 12 | Москва | 2025-12-04 11:00:00 G | 1,8 | 0 | 3,5 | 156 |
| 13 | Москва | 2025-12-04 12:00:00 G | 2,1 | 0 | 4,2 | 160 |
| 14 | Москва | 2025-12-04 13:00:00 G | 2,4 | 0 | 3,8 | 163 |
| 15 | Москва | 2025-12-04 14:00:00 G | 2,6 | 0 | 4,1 | 165 |
| 16 | Москва | 2025-12-04 15:00:00 G | 2,8 | 0 | 4,1 | 165 |
| 17 | Москва | 2025-12-04 16:00:00 G | 2,8 | 0 | 4,7 | 176 |
| 18 | Москва | 2025-12-04 17:00:00 G | 2,7 | 0 | 4,7 | 180 |
| 19 | Москва | 2025-12-04 18:00:00 G | 2,6 | 0 | 5 | 180 |
| 20 | Москва | 2025-12-04 19:00:00 G | 2,5 | 0 | 5,7 | 198 |
| 21 | Москва | 2025-12-04 20:00:00 G | 2,3 | 0 | 5,5 | 191 |
| 22 | Москва | 2025-12-04 21:00:00 G | 2,2 | 0 | 5,2 | 192 |
| 23 | Москва | 2025-12-04 22:00:00 G | 2,1 | 0 | 5,7 | 198 |
| 24 | Москва | 2025-12-04 23:00:00 G | 1,9 | 0 | 5,4 | 200 |
| 25 | Самара | 2025-12-04 00:00:00 G | 0 | 0 | 5,8 | 270 |

Рисунок 6 – Содержимое таблицы weather_hourly

4. Уведомления в Telegram

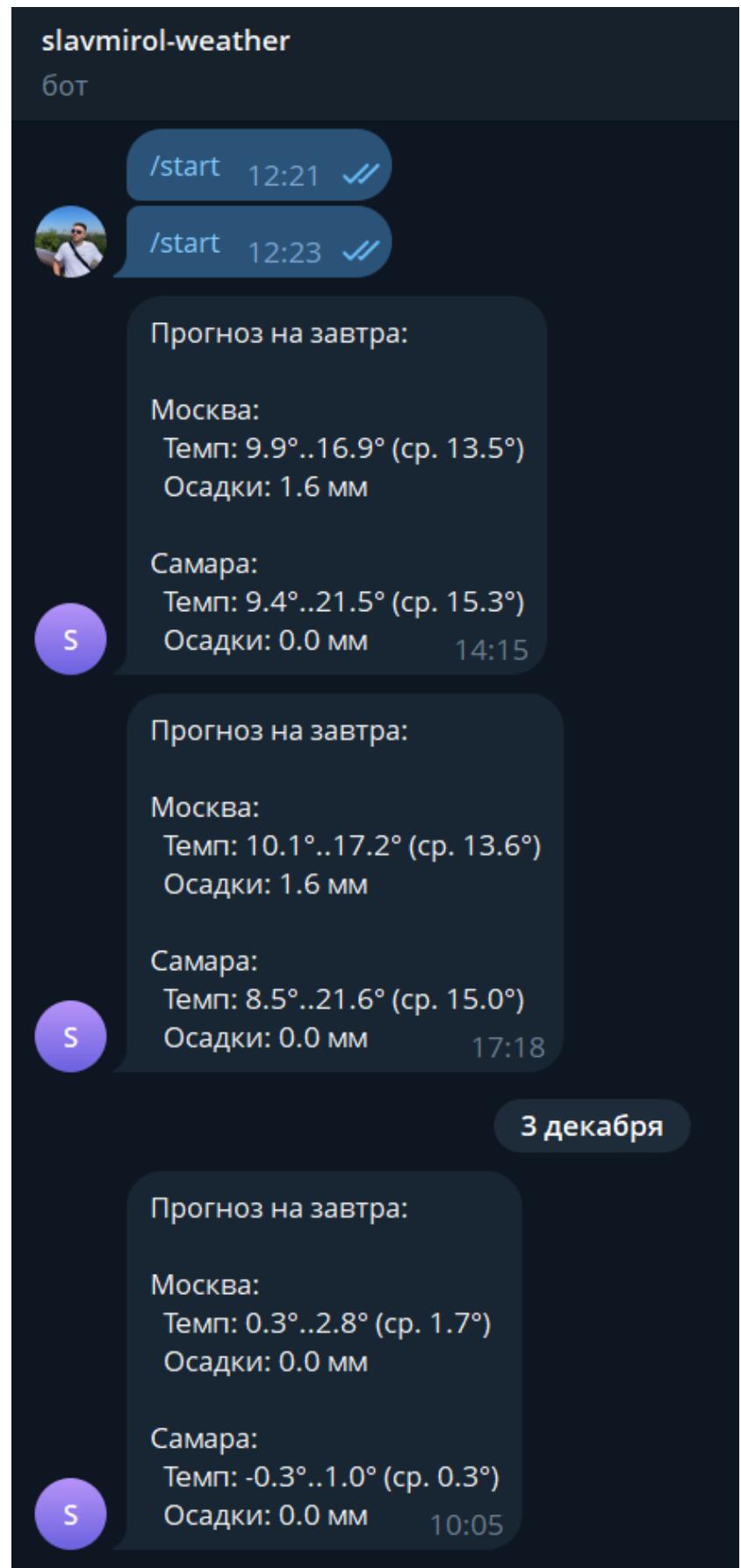


Рисунок 7 – Уведомления в телеграм

Выводы: что было сложным, что бы улучшили.

Сложным было разобраться со структурой Prefect: сервер, деплой, work-pool, процесс.

Что бы улучшил?

- Реализовал бы алERTы при сбое flow (Slack/email).
- Добавил бы Grafana-дашборд для визуализации.