

Отчёт о внедрении мониторинга и логирования в системе учёта посещаемости

Введение

Данный отчет описывает практики мониторинга и логирования, внедренные в проекте системы учета посещаемости. Система разработана с использованием современного стека технологий и развернута с применением контейнеризации и оркестрации для обеспечения высокой доступности, масштабируемости и простоты обслуживания.

Архитектура системы

Система состоит из следующих компонентов:

- **Backend-сервис (attendance-server):**
 - Node.js/Fastify-приложение на TypeScript - Реализует REST API для управления посещаемостью - Экспортирует метрики Prometheus через эндпоинт `/metrics`
- **База данных:**
 - MongoDB для хранения данных - Redis для кэширования и улучшения производительности
- **Система мониторинга:**
 - Prometheus для сбора и хранения метрик - Node Exporter для системных метрик - Grafana для визуализации метрик
- **Инфраструктура:**
 - Docker для контейнеризации - Kubernetes для оркестрации - ConfigMaps для конфигурации

Внедрённые практики мониторинга

1. Метрики приложения

В системе внедрены следующие метрики уровня приложения:

- **Метрики запросов:**
 - Счетчики запросов для всех эндпоинтов (`*RequestCounter`) - Счетчики ошибок для всех эндпоинтов (`*RequestErrors`) - Гистограммы длительности запросов (`*RequestDuration`)
- **Метрики базы данных:**
 - Метрики длительности операций с MongoDB (`dbOperationDuration*`) - Метрики длительности операций с Redis (`redisOperationDuration*`)

2. Система сбора и хранения метрик

Prometheus настроен для сбора метрик с трех источников:

- Собственные метрики Prometheus (localhost:9090)
 - Метрики Node Exporter (node-exporter:9100)
 - Метрики сервера посещаемости (attendance-server:3002)

Интервал сбора метрик настроен на 5 секунд для оперативного реагирования на проблемы.

3. Визуализация метрик

Grafana используется для визуализации метрик. Настроены дашборды для отслеживания:

- Активности сервера посещаемости
 - Производительности базы данных
 - Состояния системы
 - Показателей использования ресурсов

Внедрённые практики логирования

Система логирования построена на базе встроенного логгера Fastify:

- **Уровни логирования:**
 - Настройки логов зависят от окружения (`process.env.SIRIUS_X_ATTENDANCE_PROJECT_STATUS`)
 - В продакшен-среде - выборочное

логирование для снижения объема логов
- В тестовой и разработческой среде -
детальное логирование

- **Обработка ошибок:**
 - Настроен глобальный обработчик ошибок (`setErrorHandler`) -
 - Логирование ошибок валидации и иных исключений - Корректные HTTP-ответы с информативными сообщениями
- **Логирование жизненного цикла:**
 - Логирование запуска и остановки сервера -
 - Логирование процесса инициализации базы данных (seeding) - Корректное завершение работы при получении сигналов SIGINT и SIGTERM

Развертывание в Kubernetes

Система развернута в Kubernetes-кластере с использованием следующих ресурсов:

- **Deployments:**
 - `attendance-server-deployment.yaml` - Backend-сервис
 - `attendance-db-deployment.yaml` - MongoDB
 - `redis-deployment.yaml` - Redis
 - `prometheus-deployment.yaml` - Prometheus
 - `node-exporter-deployment.yaml` - Node Exporter
 - `grafana-deployment.yaml` - Grafana
- **Services:**
 - `attendance-server-service.yaml` -
 - `attendance-db-service.yaml` -
 - `redis-service.yaml` -
 - `prometheus-service.yaml` -
 - `node-exporter-service.yaml` -
 - `grafana-service.yaml` -
- **ConfigMaps:**
 - `attendance-env-configmap.yaml` - Переменные окружения для attendance-server
 - `prometheus-config.yaml` - Конфигурация Prometheus

Результаты внедрения

Внедрение системы мониторинга и логирования позволило:

■ **Повысить
наблюдаемость
системы:**

- Своевременное обнаружение проблем - Детальный анализ производительности - Выявление узких мест

■ **Улучшить
надежность:**

- Мониторинг состояния компонентов - Автоматические перезапуски при сбоях - Контроль ресурсов

■ **Облегчить
отладку:**

- Информативные логи - Метрики производительности - Визуализация данных

Обоснование выбора инструментов

- **Prometheus + Grafana:**
 - Стандарт индустрии для мониторинга
 - Высокая масштабируемость и надежность
 - Богатые возможности интеграции
 - Выразительный язык запросов (PromQL)

- **Node Exporter:**
 - Стандартное решение для сбора системных метрик

- Широкий набор метрик хоста

- Легкая интеграция с Prometheus

- **Kubernetes:**
 - Автоматическое управление контейнерами
 - Самовосстановление при сбоях
 - Централизованное управление конфигурацией
 - Масштабируемая

Заключение

Внедренные практики мониторинга и логирования обеспечивают полную наблюдаемость системы, позволяя оперативно реагировать на проблемы и оптимизировать производительность. Использование контейнеризации и оркестрации с помощью Kubernetes обеспечивает надежность и масштабируемость системы.

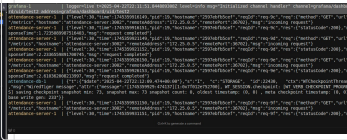
Приложение

- Скриншоты дашборда Grafana
 - Скриншот метрик Prometheus
 -

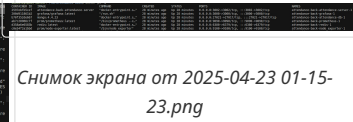
Приложение: Скриншоты



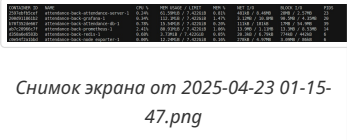
Снимок экрана от 2025-04-23 01-13-08.png



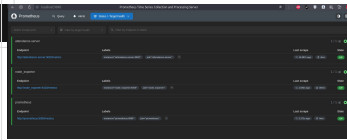
Снимок экрана от 2025-04-23 01-14-25.png



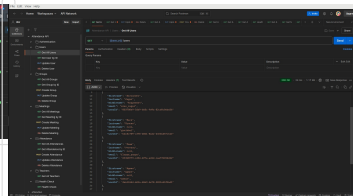
Снимок экрана от 2025-04-23 01-15-23.png



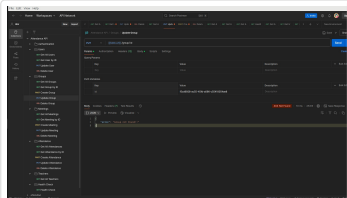
Снимок экрана от 2025-04-23 01-15-47.png



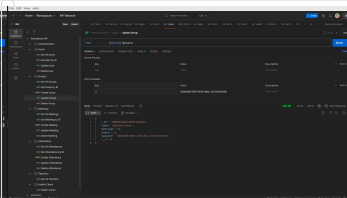
Снимок экрана от 2025-04-23 01-16-15.png



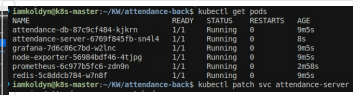
Снимок экрана от 2025-04-23 01-26-18.png



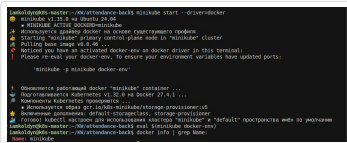
Снимок экрана от 2025-04-23 01-26-59.png



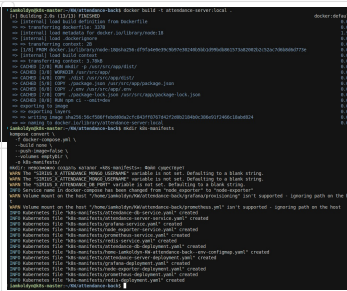
Снимок экрана от 2025-04-23 01-27-28.png



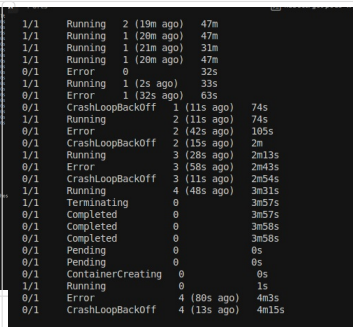
Снимок экрана от 2025-04-23 03-00-36.png



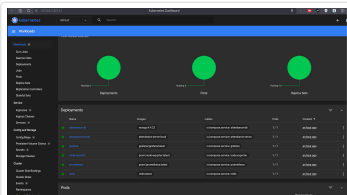
Снимок экрана от 2025-04-23 11-27-00.png



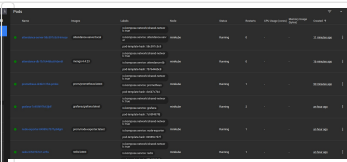
Снимок экрана от 2025-04-23 11-27-58.png



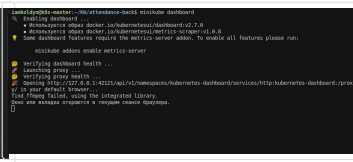
Снимок экрана от 2025-04-23 13-01-36.png



Снимок экрана от 2025-04-23 13-17-06.png



Снимок экрана от 2025-04-23 13-17-30.png



Снимок экрана от 2025-04-23 13-18-14.png