

## Лабораторная работа №10

### Метрики, логи, трейсы.

**Цель работы:** Научиться создавать распределённые системы для фиксации агрегированных, событийных действий и сохранять их полный путь.

### Задание

1. Поднимите Prometheus, elasticsearch и jaegertracing

services:

prometheus:

image: prom/prometheus:latest

container\_name: prometheus

volumes:

- "./prometheus.yml:/etc/prometheus/prometheus.yml"

ports:

- "9090:9090"

prometheus-push:

image: prom/pushgateway:master

container\_name: prom-push-gateway

ports:

- "9091:9091"

elasticsearch:

container\_name: elasticsearch

image: elasticsearch:9.2.2

environment:

- discovery.type=single-node

- xpack.security.enabled=false

- "ES\_JAVA\_OPTS=-Xms512m -Xmx512m"

ports:

- "9200:9200"

volumes:

- "./esdata:/usr/share/elasticsearch/data"

jaeger:

image: jaegertracing/all-in-one:1.76.0

ports:

- "16686:16686" # Web UI
- "14268:14268" # Collector HTTP
- "14250:14250" # Collector gRPC
- "6831:6831/udp" # Agent UDP (compact thrift)
- "6832:6832/udp" # Agent UDP (binary thrift)
- "5778:5778" # Config HTTP
- "5775:5775/udp" # Agent UDP (compact thrift)
- "4317:4317" # OTLP gRPC
- "4318:4318" # OTLP HTTP
- "9411:9411" # Zipkin

environment:

- COLLECTOR\_OTLP\_ENABLED=true
- LOG\_LEVEL=debug

2. Выбрал сервис своего приложения. Athlete-monitoring .

3. Перепишите приложения под свою предметную область, которые вы будете выводить и отслеживать, учитывая ошибки, разные уровни логов и трейсов:

5. Система мониторинга здоровья спортсменов

Сбор данных с фитнес-трекеров, анализ нагрузок, предсказание травм, персонализированные тренировочные планы.

**Что отслеживать:**

- Метрики: пульс спортсменов, уровень усталости, риск травм, прогресс тренировок
- Логи: данные с датчиков, рекомендации системы, инциденты (превышение нагрузок)
- Трейсы: путь данных "датчик → анализ → рекомендация → тренеру"

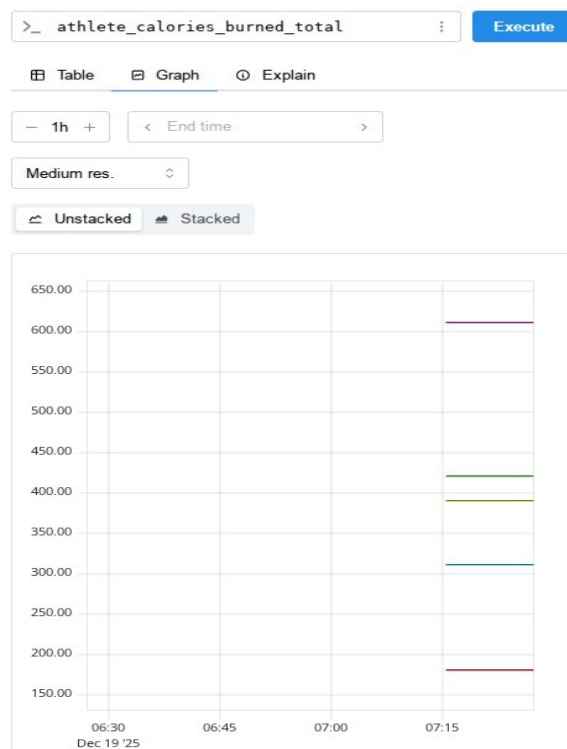


Рис. 1 — итоговое количество потраченных калорий спортсменами.

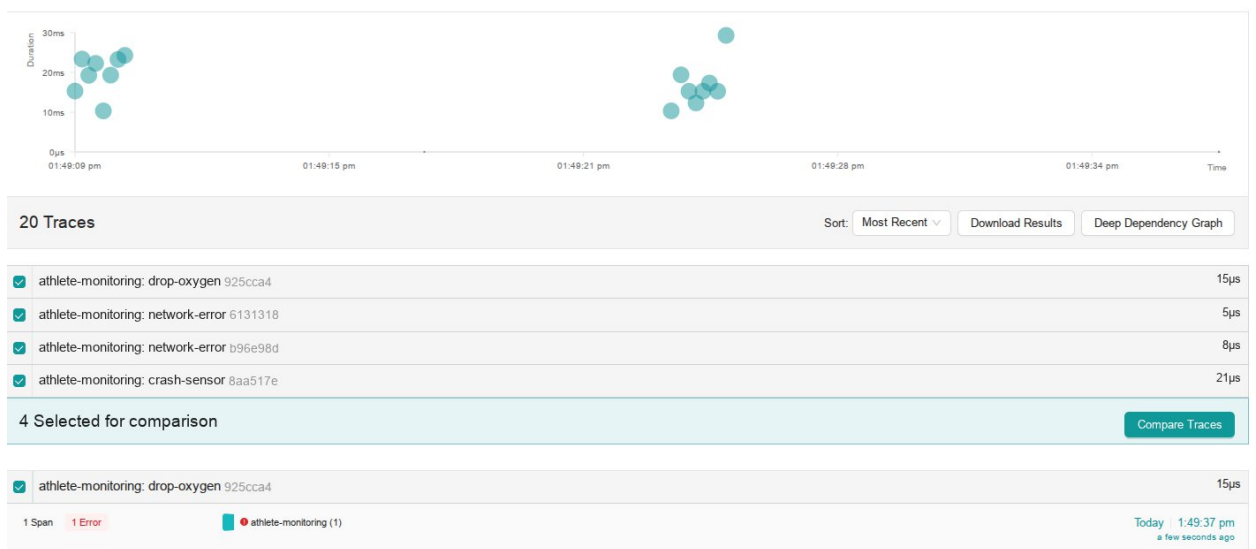


Рис. 2 — Полученные трейсы от приложения.

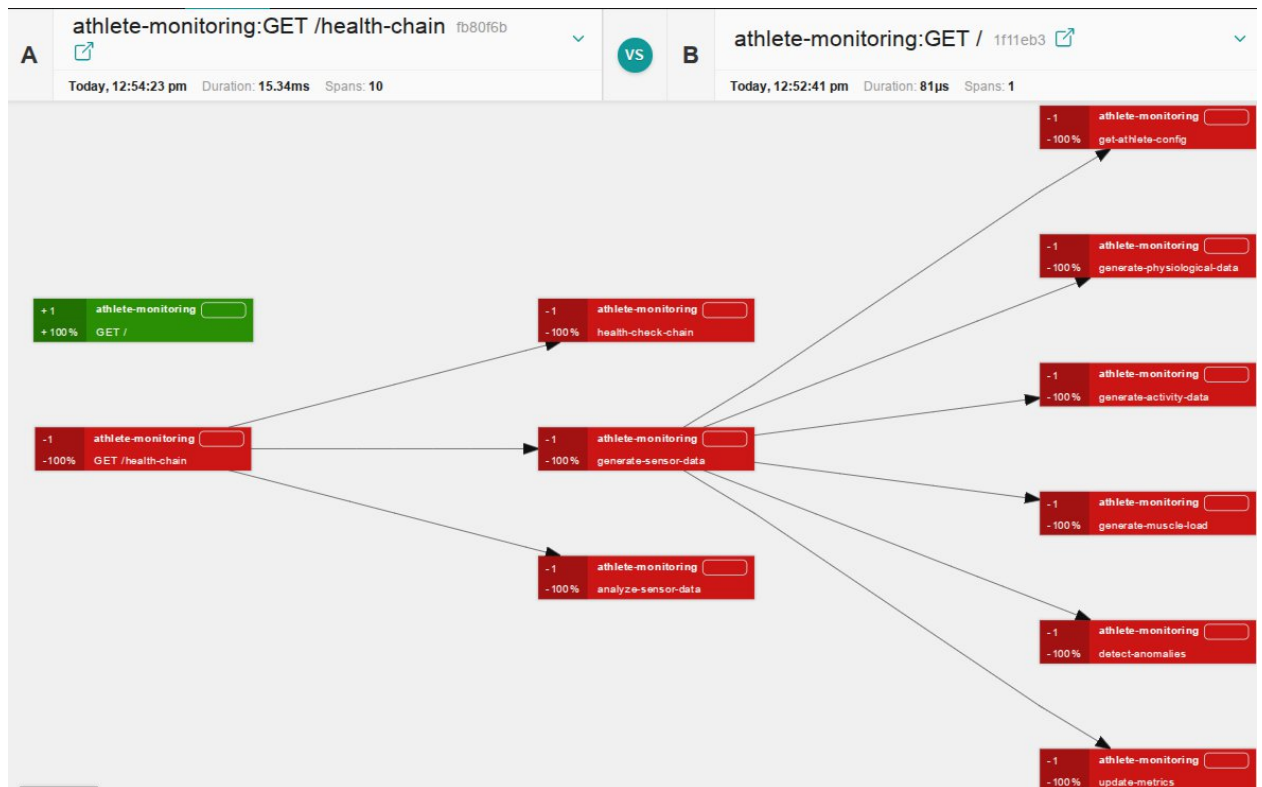


Рис. 3 — Зависимости запросов.

### Контрольные вопросы

1. Что такое метрика? Количественный показатель системы
2. Что такое логи? Текстовое сообщение о конкретном событии
3. Что такое трейсы? Путь запроса
4. Кому в компании необходимы метрики? всем
5. Для кого необходимы логи?

Разработчику

6. Что позволяет отследить трейсы?

Путь запроса.

7. Какие бывают трейсы?

Стек-трейс (Stack Trace), Распределенный трейсинг (Distributed Tracing), HTTP-трейсинг, Системный трейсинг (System Tracing) .

8. Сделайте диаграмму описывающую взаимосвязь приложения с сервисами для сборов лог, метрик и трейсов, используя draw.io

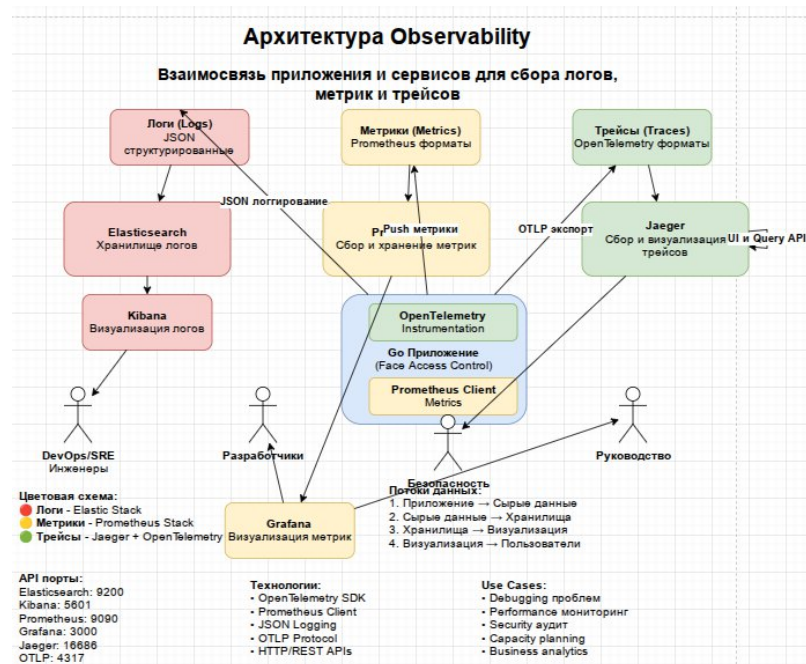


Рис. 4 — Диаграмма.

**Вывод:** Я научился создавать распределённые системы для фиксации агрегированных, событийных действий и сохранять их полный путь.