

Obserwowalność

Metryki, logi i ślady w jednej diagnozie

mgr inż. Jakub Woźniak

Zarządzanie Systemami Rozproszonymi · Semestr zimowy 2026/27

Pomiar	09:55	10:05
Żądania API / s	100	100
Błędy API	0,1%	0,1%
p95 czasu API	180 ms	2400 ms
CPU API	35%	38%

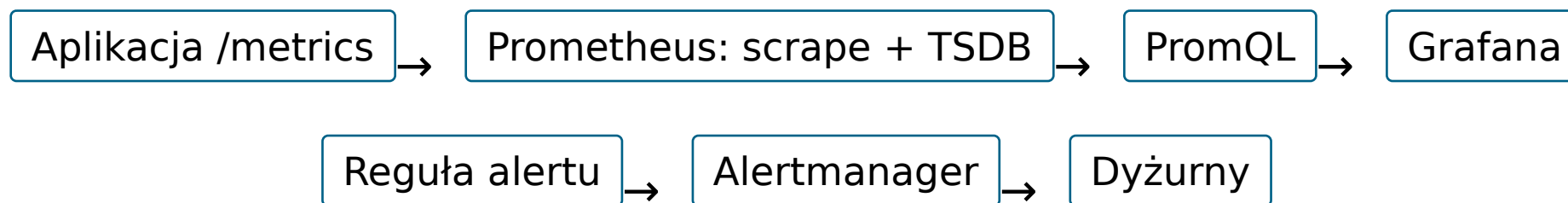
Pytanie diagnostyczne

Czy niski CPU i brak 5xx oznaczają poprawne działanie? Jak oddzielić czas pracy od czasu oczekiwania?

Sygnał	Jednostka	Zastosowanie
Metryka	Szereg czasowy	Trend, agregacja, alert
Log	Zdarzenie z kontekstem	Szczegół błędu i identyfikator
Trace	Spany powiązanej operacji	Czas i zależności na ścieżce

- Metryka zawęża okres i usługę; trace wskazuje wolny etap; log opisuje zdarzenie.

OpenTelemetry — Signals



- Pull: Prometheus okresowo odpytuje endpointy metryk.
- Scrape niedostępny $\neq$ wartość metryki równa zero.
- Grafana prezentuje wyniki zapytań; nie zastępuje instrumentacji.

Prometheus — Overview

```
orders_completed_total{result="ok"} 12340  
queue_oldest_age_seconds{queue="orders"} 42  
http_request_duration_seconds_bucket{le="0.5"} 950  
http_request_duration_seconds_bucket{le="1"} 990  
http_request_duration_seconds_bucket{le="+Inf"} 1000
```

- Counter: rośnie; może się wyzerować po restarcie.
- Gauge: bieżący stan, może rosnać i maleć.
- Histogram klasyczny: kumulatywne liczniki przedziałów.

Prometheus — Metric types

```
sum(rate(http_requests_total{job="api"}[5m]))  
  
sum(rate(http_requests_total{job="api",code=~"5.."}[5m]))  
/  
sum(rate(http_requests_total{job="api"}[5m]))
```

- rate: tempo na sekundę, z obsługą resetu countera.
- Najpierw rate na szeregu, potem suma po instancjach.
- Okno 5 min wygładza; udział błędów bez ruchu wymaga osobnej interpretacji.

Prometheus — rate()

```
histogram_quantile(  
    0.95,  
    sum by (le) (  
        rate(http_request_duration_seconds_bucket{job="api"}[5m])  
    )  
)
```

- Sumujemy bucket counters; zachowujemy granicę `le`.
- Wynik przybliżony: dokładność zależy od granic przedziałów.
- Średnia z p95 instancji nie daje p95 całej usługi.

Prometheus — Histograms and summaries

Przykład dydaktyczny

10 tras $\times$ 5 kodów odpowiedzi $\times$ 20 instancji = do 1000 kombinacji etykiet jednej metryki.

- Dodatkowo `user_id` o 100 000 wartościach: do 100 mln kombinacji.
- Rzeczywista liczba zależy od występujących kombinacji; histogram mnoży serie przez buckety.
- Identyfikator zamówienia: `log` lub `trace` zamiast labela metryki.

Prometheus — Naming and labels



Oś: 0-2400 ms; przykład jednego wolnego żądania. Pozostały czas: inne etapy API.

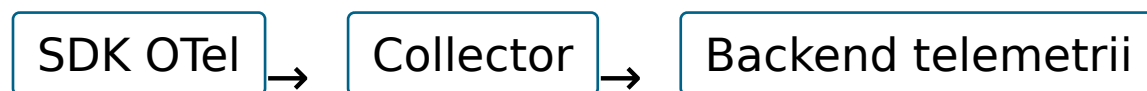
- Span oczekiwania na połączenie DB dominuje nad wykonaniem SQL.
- W kolejce kontekst wymaga przeniesienia w wiadomości; możliwe relacje links.

OpenTelemetry — Traces

```
{  
  "time": "10:05:12.210",  
  "service": "orders-api", "version": "v2",  
  "trace_id": "4bf92f3577b34da6a3ce929d0e0e4736",  
  "event": "db_pool_wait",  
  "wait_ms": 2100, "pool_max": 5, "pool_in_use": 5  
}
```

- Metryka: skala zjawiska. Trace: miejsce oczekiwania. Log: konfiguracja puli.
- Bez tokenów, haseł i pełnych danych zamówienia w logu.

Log syntetyczny; nazwy pól aplikacyjne.



- Collector: odbiór, przetwarzanie, batching i eksport; także kolejki i limity.
- Backend, np. Azure Monitor: przechowywanie i zapytania według jego modelu.
- Redakcja danych przed eksportem; awaria backendu nie powinna blokować zamówień.

Decyzja	Zysk	Ograniczenie
Krótsza retencja logów	Mniej przechowywanych danych	Mniej historii diagnozy
Sampling śladów	Mniej eksportu	Możliwe pominięcie rzadkiej awarii
Agregacja metryk	Mniej serii	Utrata części wymiarów

- SLI i rachunek błędów opierać na odpowiednim pełnym pomiarze.
- Próbkowanie pomaga diagnozować; nie daje automatycznie bezstronnej statystyki.

```
# Zmiana konfiguracji orders-api
- DB_POOL_MAX: "20"
+ DB_POOL_MAX: "5"

# W tym samym okresie
DB active connections: 50 / limit 200
DB CPU: 25%
API replicas: 10; traffic: bez zmiany
```

Decyzja

Połącz diff, trace i log. Zaproponuj zmianę, pomiar potwierdzający i jedną alternatywną hipotezę.

- [Prometheus — Overview](#)
- [Prometheus — Query functions](#)
- [Prometheus — Histograms](#)
- [Prometheus — Naming](#)
- [OpenTelemetry — Traces](#)
- [OpenTelemetry — Collector](#)

Dokumentacja sprawdzona: 26.09.2026. Dane przypadków: przykłady dydaktyczne.