Analisando o comportamento de aplicações:

DRAGONBD

NE0B1TS

KuberLink

API4COM

dati

Os 3! pilares usados pelas maiores empresas do mundo para fazer observabilidade

Code Island CLOUD 2025

George Tavares

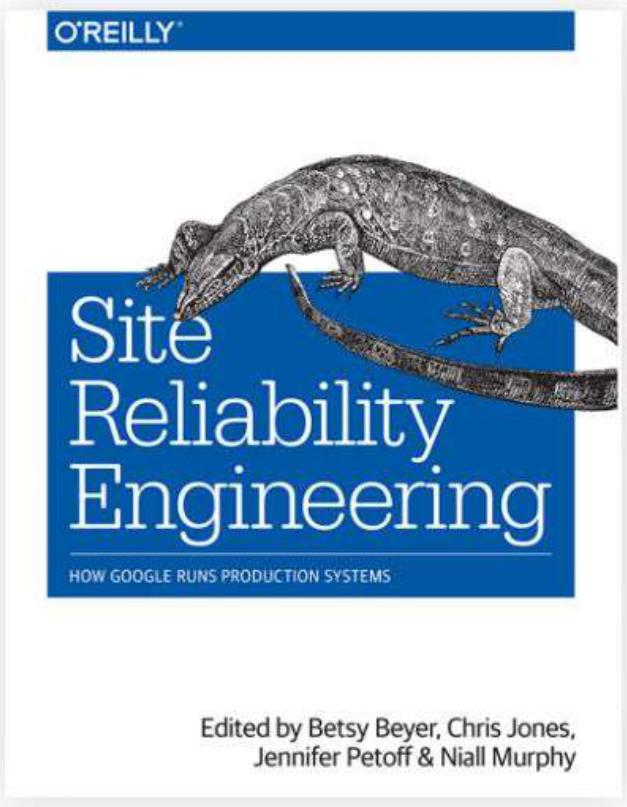
Observabilidade - DevOps

Golden Signals
- Rate
- Errors
- Duration
- Saturation

- SLI
- SLO
- SLA

Metodologia USE - Recursos
- Utilization
- Saturation
- Error

Metodologia RED - Serviços
- Rate
- Error
- Duration



Observabilidade Opentelemetry

Métricas

counter,gauge,histogram,summary

Pull (prometheus)

Push

Traces

Span, events

Opentelemetry (opencensus+opentracing)

Instrumentação por agentes

Log

Baggage

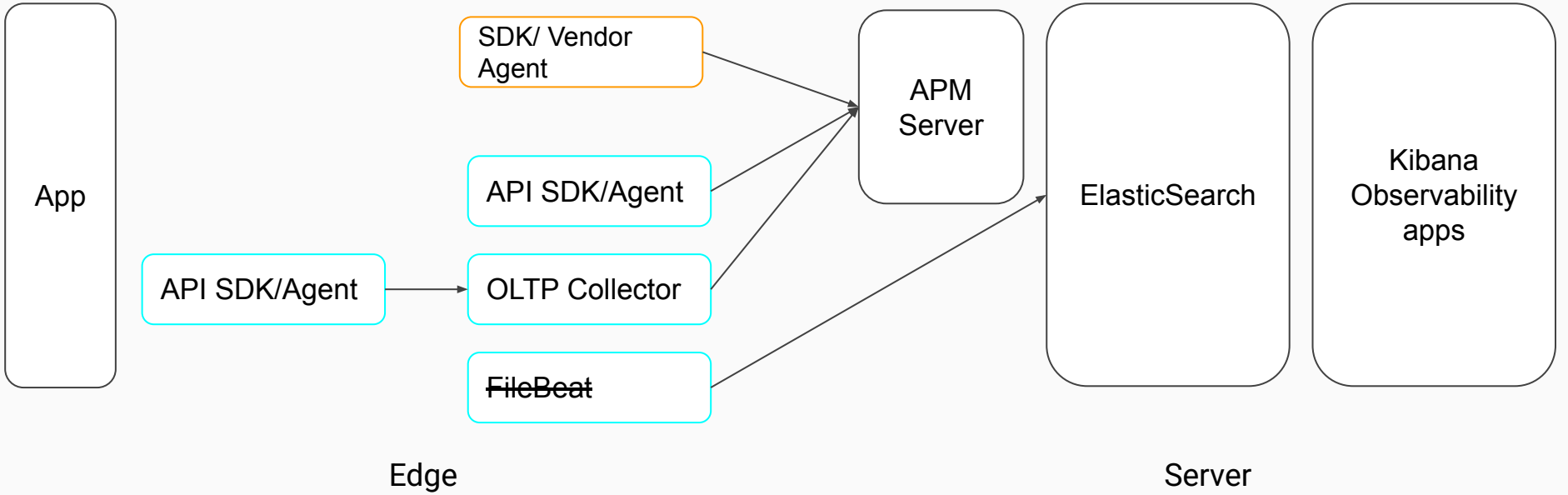
Collector

OTTL

Alertas??

Profiling

Opentelemetry Demo Setup



Semantic Convention

- Define nomes comuns aos nomes dos atributos, métricas propriedades
- Cada provider tinha seu próprio nome
- Prometheus teve que mudar bastante para suportar isso

HTTP server

Metric: `http.server.request.duration`

This metric is required.

When this metric is reported alongside an HTTP server span, the metric value SHOULD be the same as the HTTP server span duration.

This metric SHOULD be specified with ExplicitBucketBoundaries of `[ 0.005, 0.01, 0.025, 0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 1, 2.5, 5, 7.5, 10 ]`.

Naming conventions

Meter names should use dots to separate segments, `a.name.like.this`. Micrometer applies naming conventions to convert registered meter names to match the expectations of backend monitoring systems.

Given the following declaration of a timer: `registry.timer("http.server.requests")`, applied naming conventions will emit the following metrics for different monitoring systems:

- Prometheus: `http_server_requests_duration_seconds`
- Atlas: `httpServerRequests`
- Graphite: `http.server.requests`
- InfluxDB: `http_server_requests`

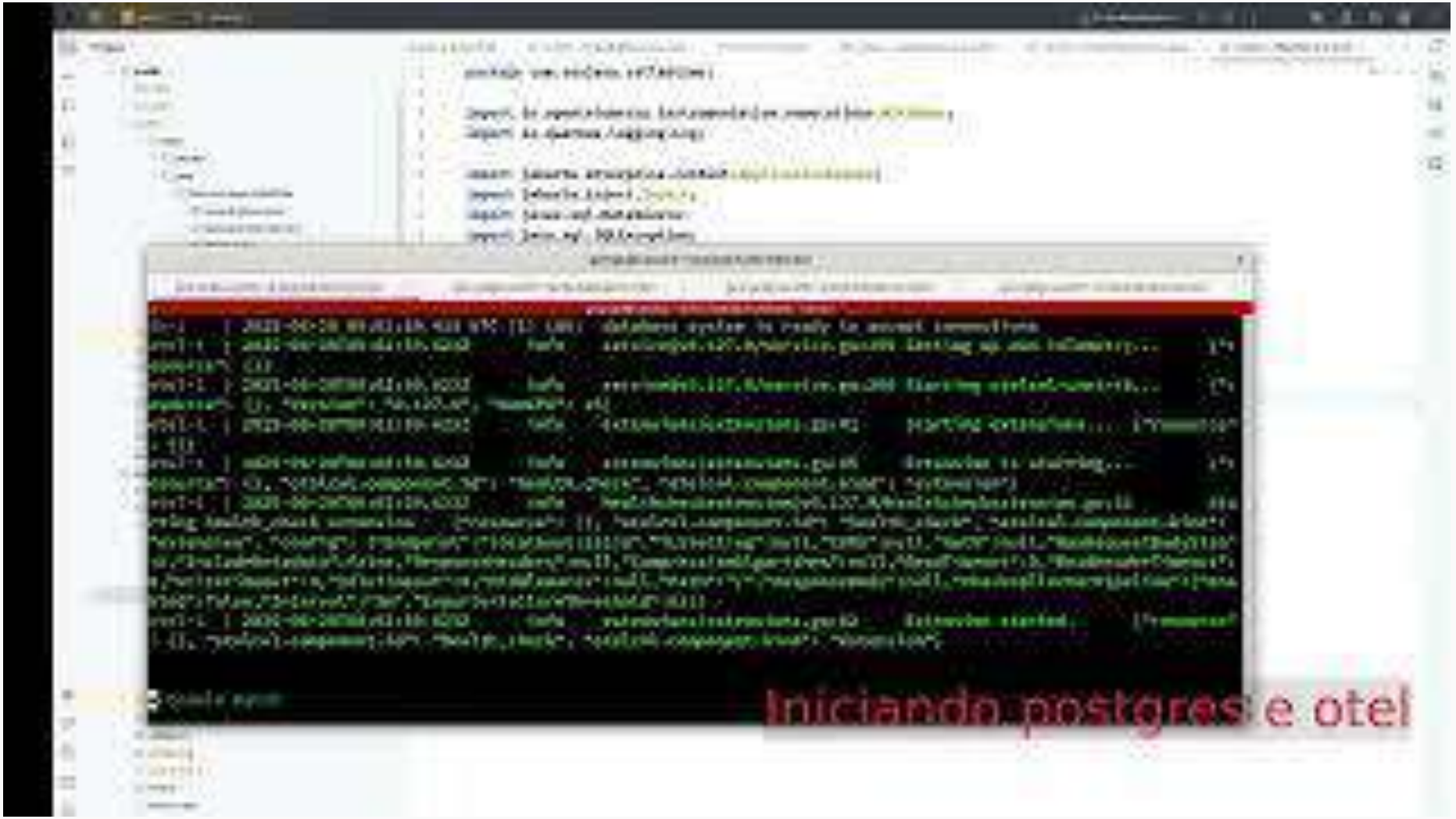
Demo Application

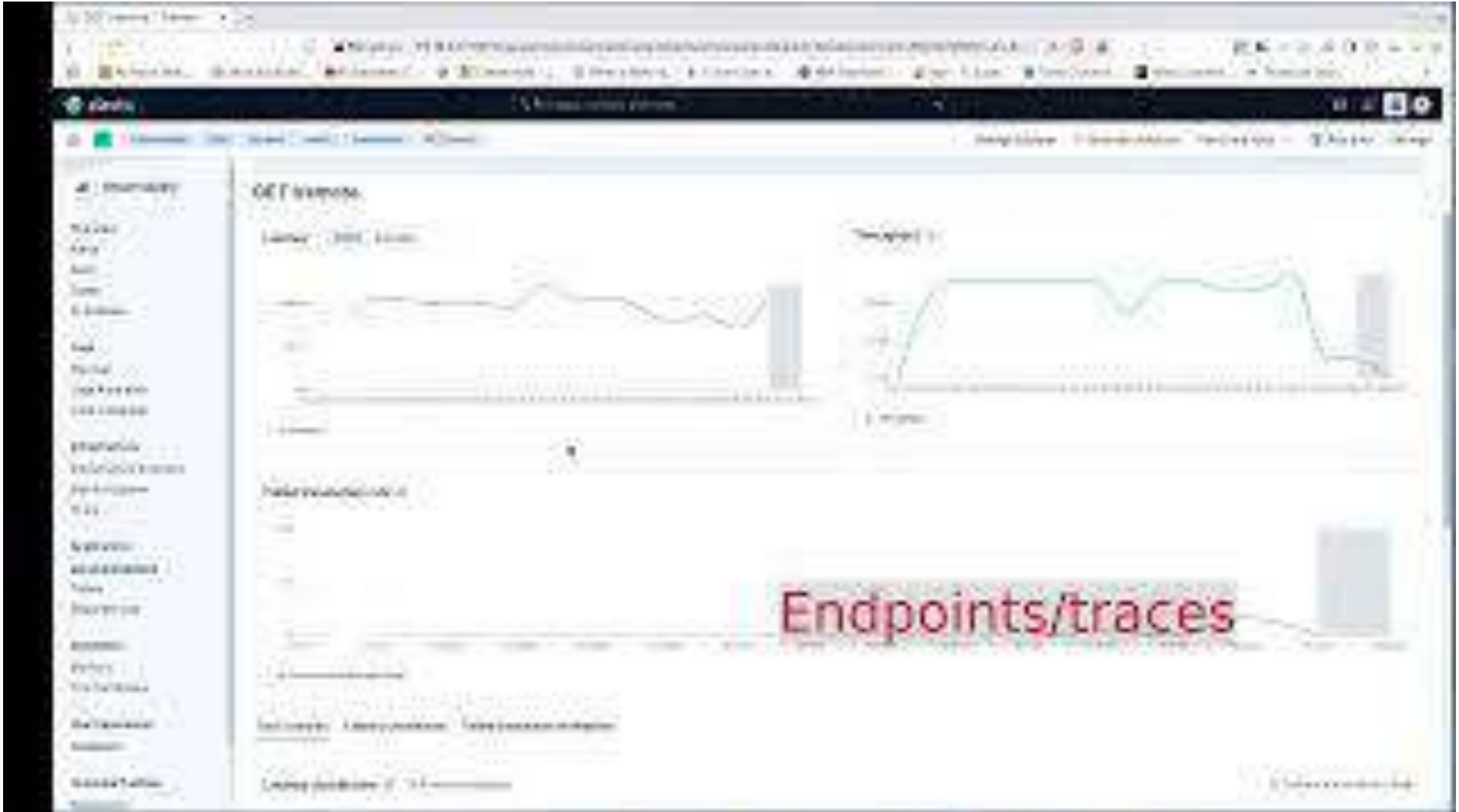
* Services:

** code1
Quarkus, opentelemetry, metric otlp exporter
- hello: call wait
- error: call wait, throw exception
- remote: call wait, remote call code2 -> hello

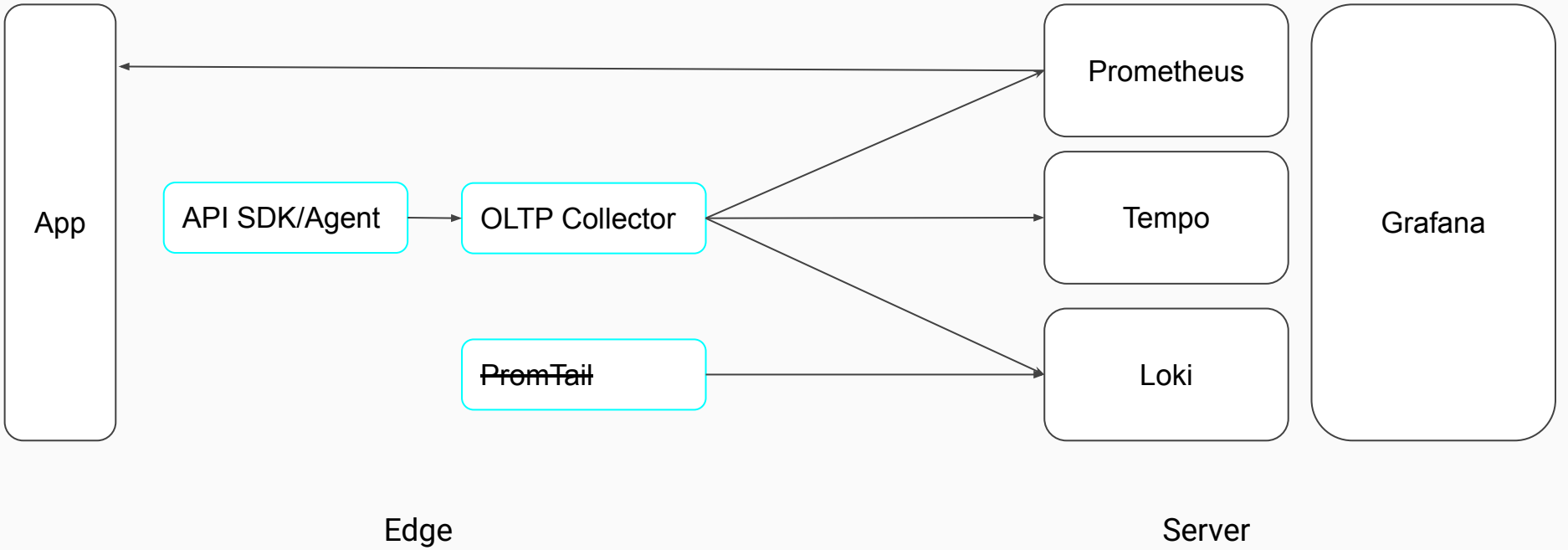
** code2
Quarkus, opentelemetry, metric otlp exporter, jdbc addon
- hello: call wait (request to postgres pg_sleep)

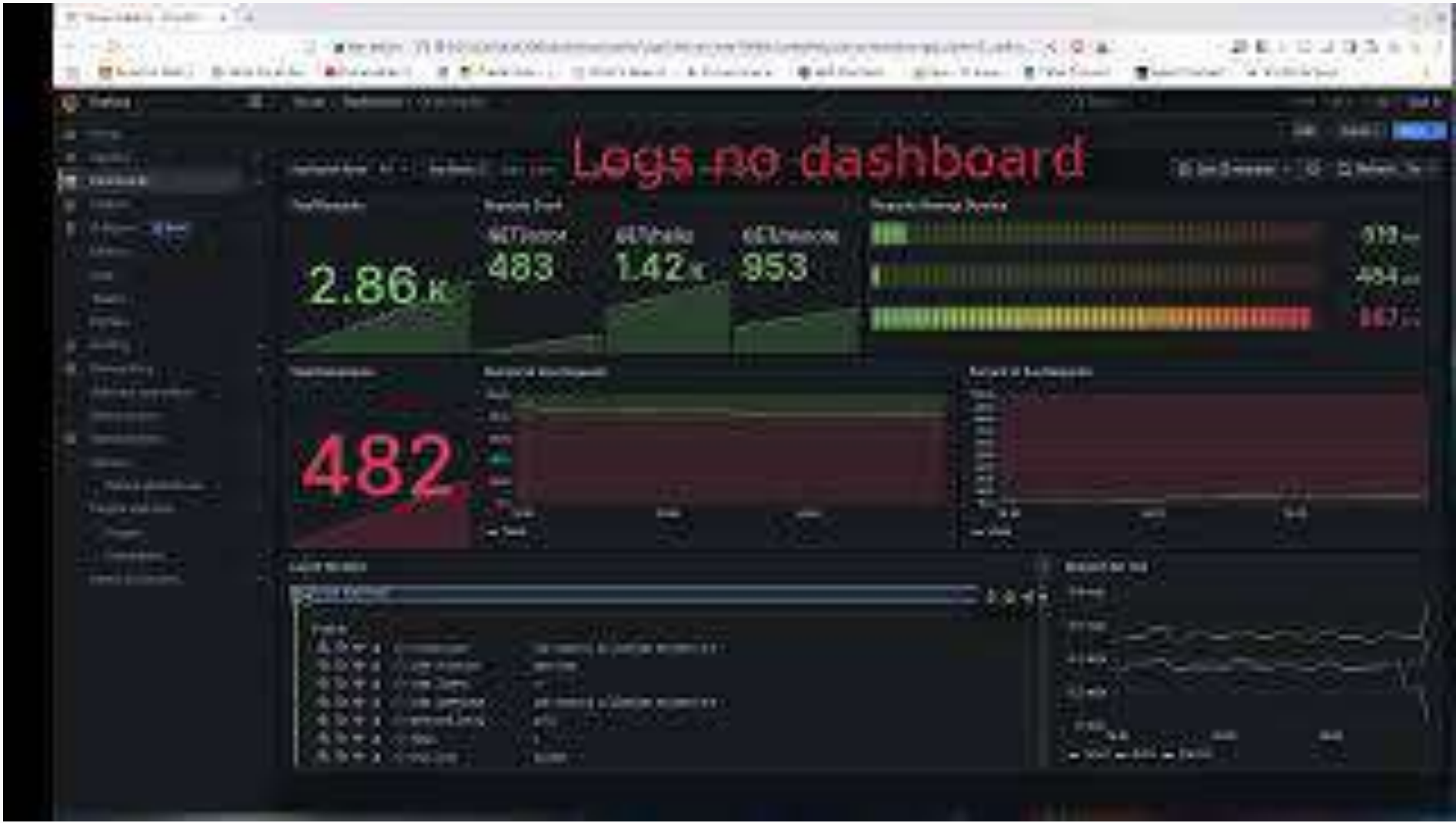
GraalVM native compiled - AOT


Iniciando postgres e otel


Endpoints/traces

Grafana Tempo Loki Prometheus Demo Setup




Logs no dashboard
2.86 K
483
1.42 K
953
482

Futuro

Continuos Profiling

- Grafana comprou o Pyroscope
- Elastic comprou Optimyze e está doando para o opentelemetry
- Permite identificar os hot spots de gargalo de performance
- Não é uma visão de timeline, e sim uma visão de agregados de tempo de execução
- eBPF


Trace d buido

George Tavares

- Desenvolvedor na Noson
- Entusiasta de tecnologia, adoro estudar infraestrutura e devops
- Tiro certificações nas horas vagas: 4xAWS, 4xAZURE, CKA, CKS, ~~Elastic Engeneering, Elastic Observability~~ ...
- Vencedor do 1o K8S ULC da Linuxtips

https://github.com/tanquetav/lab-observabilidade

https://www.linkedin.com/in/tavaresgeorge/

Perguntas ?