



# Plataforma de Observabilidade: Auto instrumentação e Gerenciamento de Telemetria em Ambientes Kubernetes

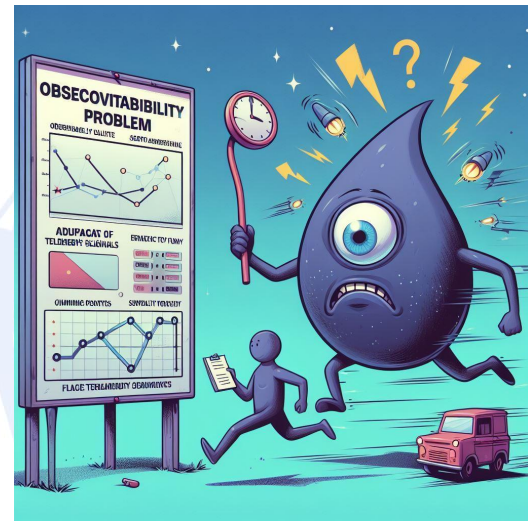


# Ezzio Moreira

- Casado com Luciana.
- Graduado em Redes de computadores, pós-graduado em arquitetura de software.
- Possuo algumas certificações.
- Membro de:
  - DevOps Days Fortaleza
  - DevOps Ceará
  - Mentoria DevOps
  - Dose na Nuvem
- SRE na Dr. Cash.
- Aprendendo Observabilidade, OpenTelemetry e Golang.

# Contextualizando o Problema

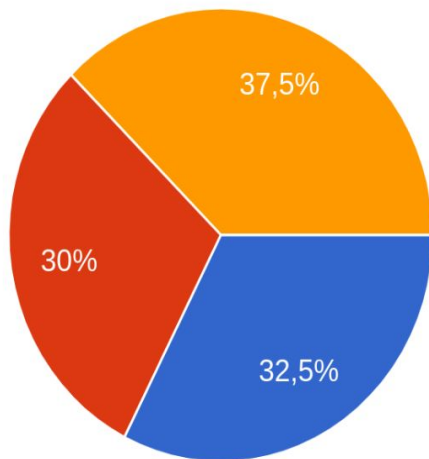
- Times só considera a observabilidade durante ou depois do desenvolvimento do software.
- Sistemas são implantados com poucos sinais de telemetria.
- A ausência de sinais de telemetria adequados.
- Demora na implementação da observabilidade.



# Resultado da Pesquisa

Como a observabilidade é incorporada ao ciclo de desenvolvimento de software?

40 respostas

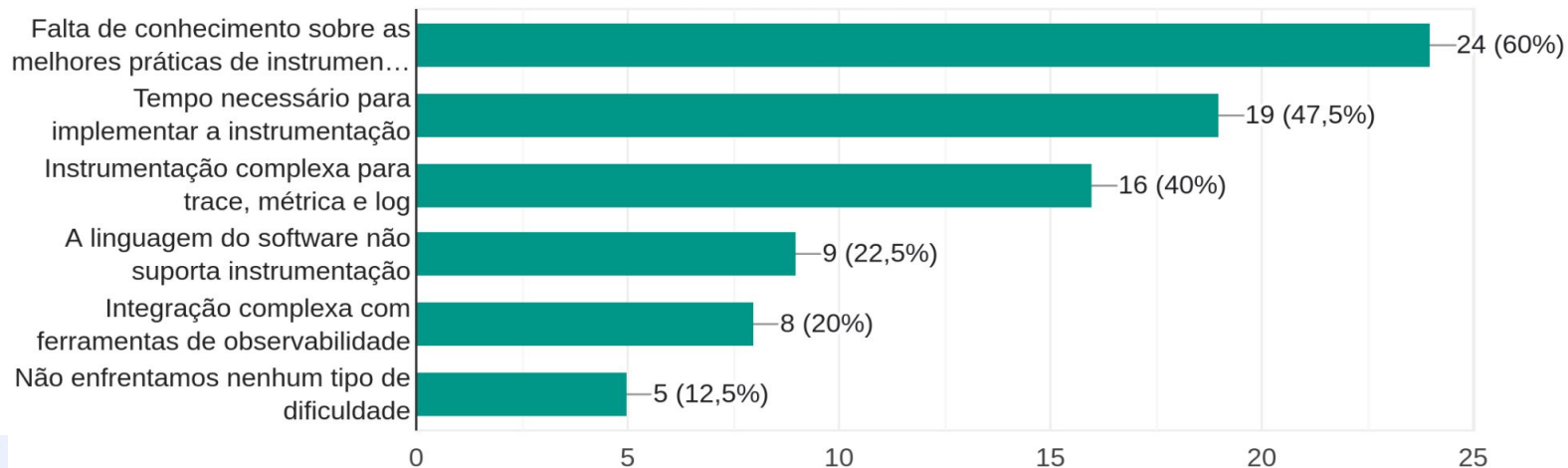


- A observabilidade é considerada após o desenvolvimento
- A observabilidade é considerada durante o desenvolvimento
- A observabilidade é considerada desde o início do planejamento

# Resultado da Pesquisa

Os times enfrentam dificuldades ao implantar sinais de observabilidade em suas aplicações? Se sim, quais?

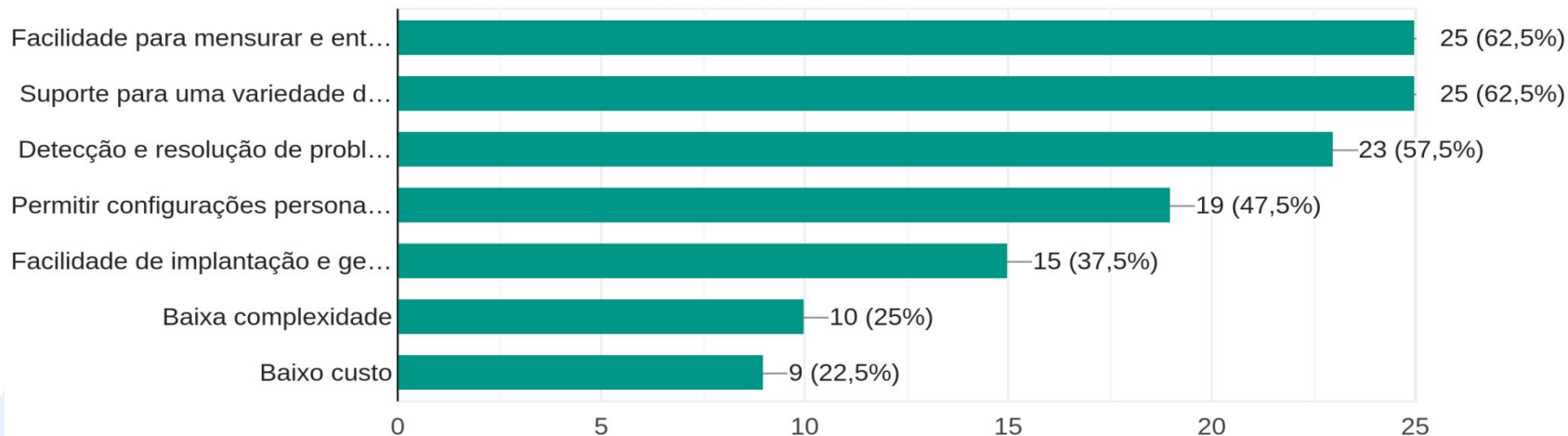
40 respostas



# Resultado da Pesquisa

Quais são suas expectativas em relação a uma plataforma de observabilidade eficaz para ambiente Kubernetes?

40 respostas





# Os principais problemas detectados

- A falta de um padrão bem definido.
- Tempo necessário para implantar a telemetria.
- Complexidade da instrumentação.



# A proposta de solução

Uma plataforma de observabilidade que seja capaz de implementar instrumentação automática e gerenciamento de telemetria para ambientes Kubernetes, com intuito de reduzir a complexidade e maximizando a eficiência operacional dos times.



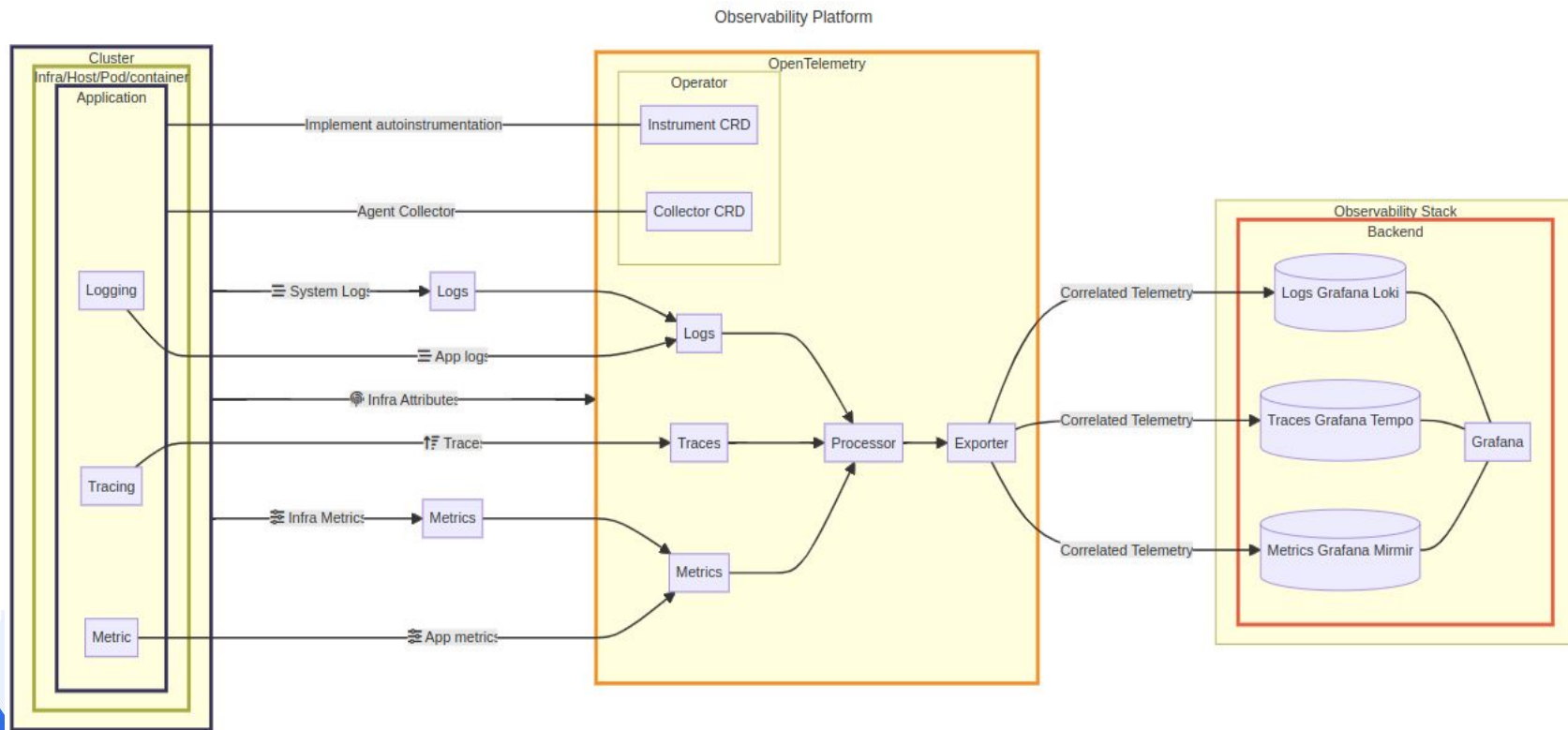


# OpenTelemetry

- Padronização.
- Instrumentação Automática.
- Interoperabilidade.
- Suporte a múltiplas linguagens.
- Cloud Native.
- Comunidade Ativa.

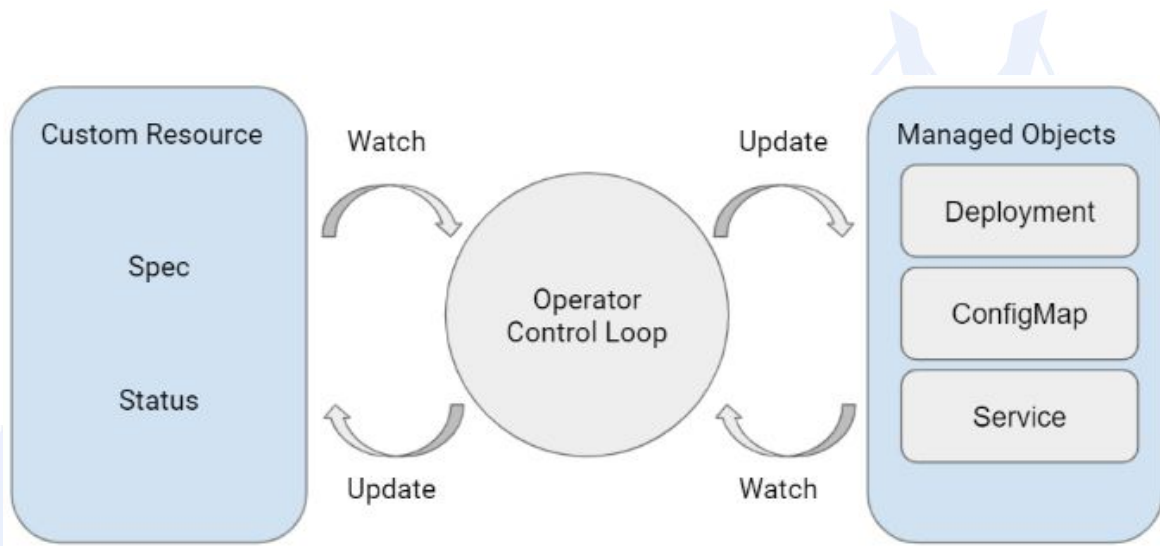


# Arquitetura da Plataforma



# Principais Componentes

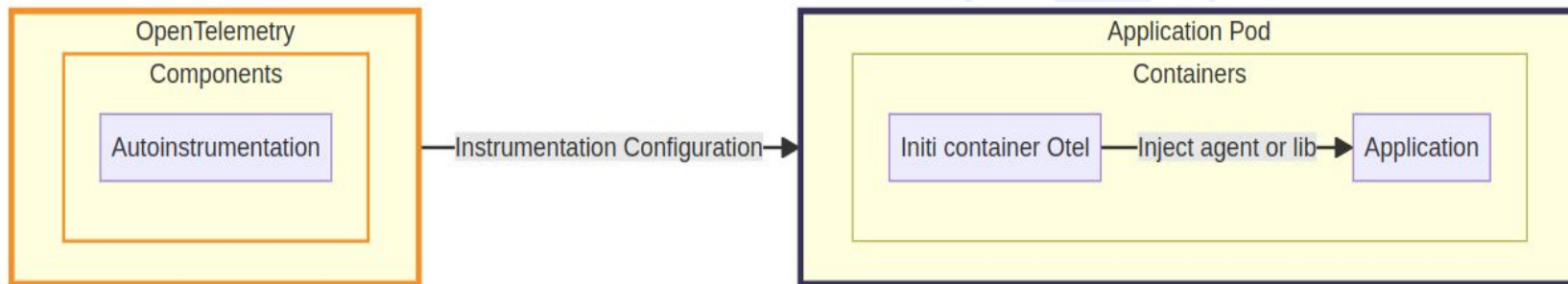
- **Operator:**
  - Gerencia recursos OpenTelemetry no cluster Kubernetes.



# Principais Componentes

- **Instrumentation:**

- Implementa instrumentação de rastros (trace) automaticamente.



# Instrumentation

```
1  apiVersion: opentelemetry.io/v1alpha1
2  kind: Instrumentation
3  metadata:
4    name: python-instrumentation
5  spec:
6    exporter:
7      endpoint: http://platform-agent-collector.observability.svc.cluster.local:4318
8    propagators:
9      - tracecontext
10     - baggage
11     - b3
12     - b3multi
13     - jaeger
14     - ottrace
15    sampler:
16      type: parentbased_traceidratio
17      argument: "1"
```

# Instrumentation

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: petclinic # Nome do deployment
spec:
  selector:
    matchLabels:
      app: petclinic # Seleciona pods com o rótulo "app: petclinic"
  replicas: 1 # Número de réplicas desejadas
  template:
    metadata:
      labels:
        app: petclinic # Rótulo aplicado aos pods
      annotations:
        instrumentation.opentelemetry.io/inject-java: "java-instrumentation"
    spec:
      containers:
```

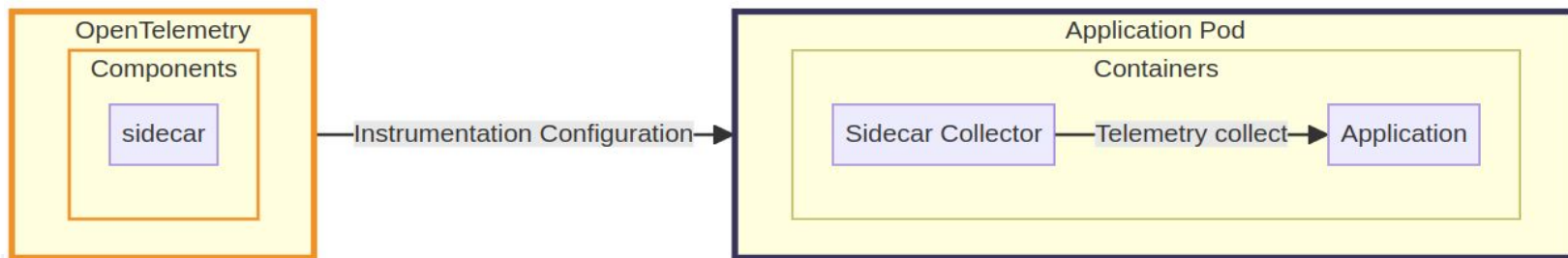




# Principais Componentes


- **Sidecar:**

- Implementa um segundo contêiner no Pod da aplicação que será responsável pela coleta e transporte dos dados de telemetria.



# Sidecar

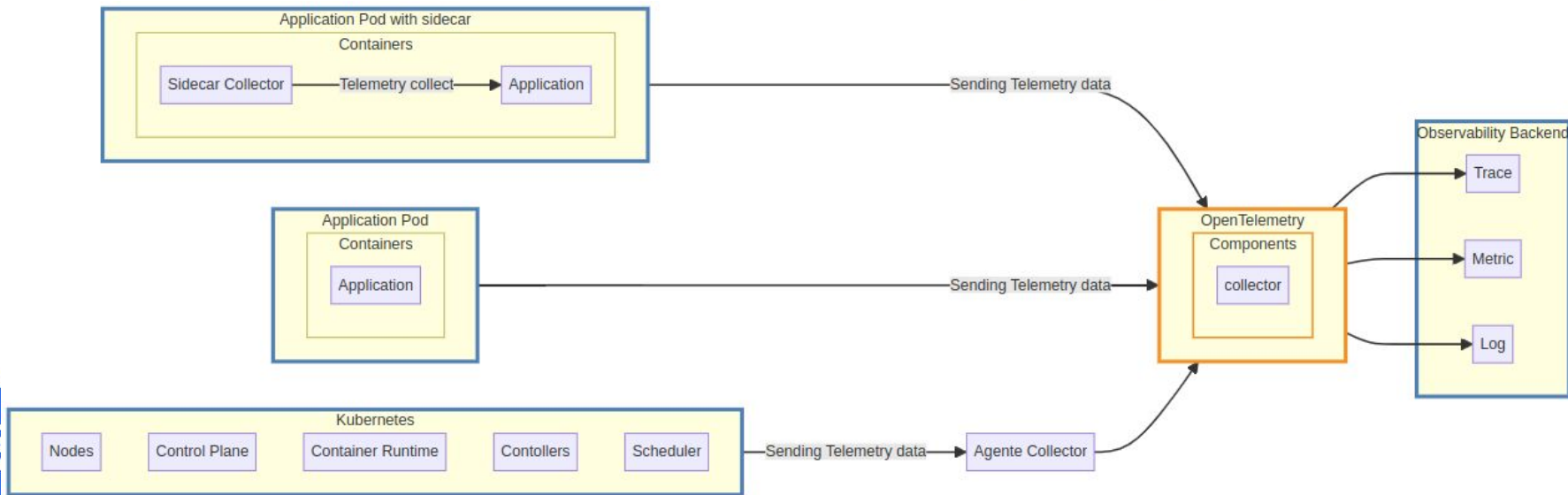
```
apiVersion: v1
kind: Pod
metadata:
  name: myapp
  annotations:
    sidecar.opentelemetry.io/inject: "true"
spec:
  containers:
    - name: myapp
      image: jaegertracing/vertx-create-span:operator-e2e-tests
      ports:
        - containerPort: 8080
          protocol: TCP
```



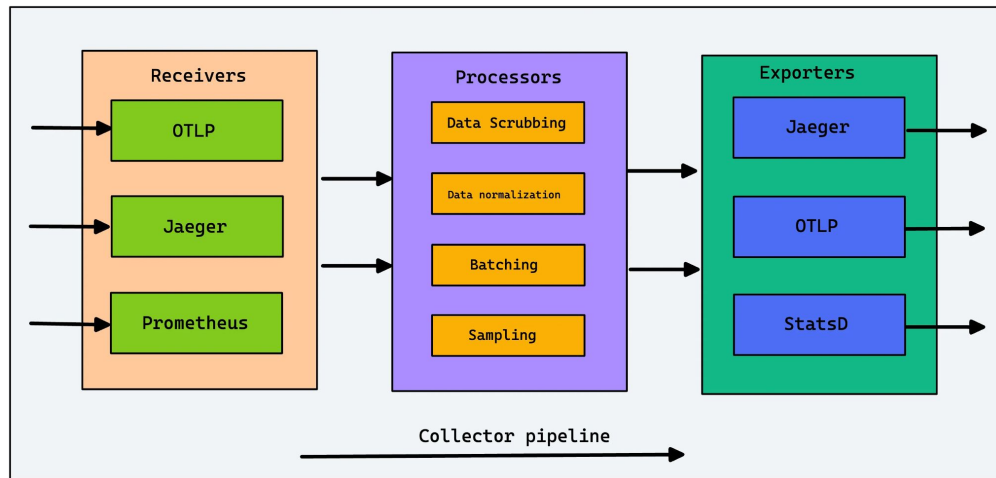
# Principais Componentes

- **Collector:**

- Agentes responsáveis por receber, processar e enviar dados de telemetria.



# Collector



```
receivers:
  otlp:
    protocols:
      grpc:
        endpoint: 0.0.0.0:4317
      http:
        endpoint: 0.0.0.0:4318
processors:
  batch:

exporters:
  otlp:
    endpoint: otelcol:4317

extensions:
  health_check:
  pprof:
  zpages:

service:
  extensions: [health_check, pprof, zpages]
  pipelines:
    traces:
      receivers: [otlp]
      processors: [batch]
      exporters: [otlp]
    metrics:
      receivers: [otlp]
      processors: [batch]
      exporters: [otlp]
    logs:
      receivers: [otlp]
      processors: [batch]
      exporters: [otlp]
```

# Collector

```
1  apiVersion: opentelemetry.io/v1alpha1
2  kind: OpenTelemetryCollector
3  metadata:
4    name: platform-agent
5    namespace: observability
6    labels:
7      monitoring: prometheus
8  spec:
9    image: otel/opentelemetry-collector-contrib-dev:latest
10   mode: "daemonset"
11   serviceAccount: platform-agent
12   hostNetwork: true
13   env:
14     - name: KUBE_NODE_NAME
15       valueFrom:
16         fieldRef:
17           apiVersion: v1
18           fieldPath: status.hostIP
19     - name: POD_IP
20       valueFrom:
21         fieldRef:
22           fieldPath: status.podIP
23     - name: OTEL_SERVICE_NAME
24       value: "platform-agent"
25     - name: TOKEN
26       valueFrom:
27         secretKeyRef:
28           name: grafana-secret
29           key: token
```

# O que a solução tem de especial?

- A plataforma automatiza e simplifica a implantação, coleta, processamento e envio de dados de telemetria.
- Reduz a carga operacional, facilitará a entrega de software com maior confiabilidade.
- Garante um mínimo de observabilidade para os sistemas.
- A disseminação da cultura de observabilidade entre os times.
- Proporciona maior flexibilidade e liberdade aos usuários.





# Atenção

- A auto instrumentação é uma porta de entrada para a observar seu sistema. É uma maneira eficiente de começar a monitorar e entender o comportamento do sistema.
- A instrumentação manual combinada com a automática resulta em dados de telemetria mais ricos e detalhados.
- Essa abordagem híbrida maximiza o potencial de observabilidade do sistema.

## Contato



[github.com/EzzioMoreira](https://github.com/EzzioMoreira)



[linkedin.com/in/ezzio-moreira](https://linkedin.com/in/ezzio-moreira)



[t.me/dosedetelemetria](https://t.me/dosedetelemetria)



[mentoriadevops.io](https://mentoriadevops.io)

# Links e Referências

- Build in Observability While Developing - Jamie Danielson, Honeycomb.io
  - [https://www.youtube.com/watch?v=TXO\\_d594Ri8](https://www.youtube.com/watch?v=TXO_d594Ri8)
- MVP GitHub - observability-platform
  - <https://github.com/EzzioMoreira/observability-platform>
- OpenTelemetry Operator
  - <https://opentelemetry.io/docs/kubernetes/operator/>
- Projeto Aplicado
  - [https://docs.google.com/document/d/1mUWeTiMe925Zvst0DstMT2\\_0ctUcNGbA/edit?usp=sharing&oid=103595219062593706643&rtpof=true&sd=true](https://docs.google.com/document/d/1mUWeTiMe925Zvst0DstMT2_0ctUcNGbA/edit?usp=sharing&oid=103595219062593706643&rtpof=true&sd=true)

