



Observabilidade

Além do alcance

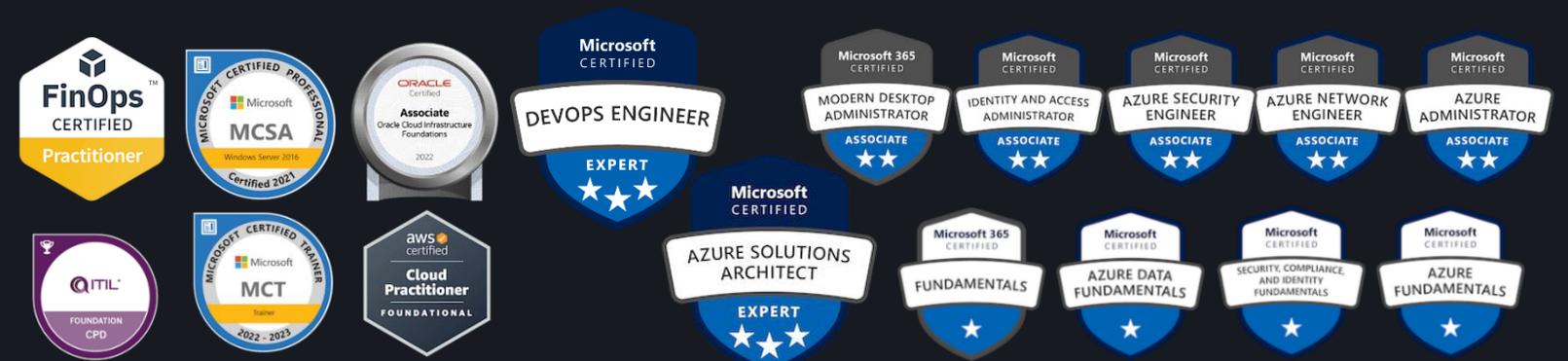
Mentoria 3.0 Canal da Cloud
Trilha DevOps

Nível: Básico - Intermediário

WHOIAM

Rafael Martin Alves Ferreira

- 10+ anos atuando com tecnologia
- Arquiteto de Soluções Cloud @ CI&T
- Algumas Certificações técnicas
- Bacharel em Ciências da Computação
- Geek, Gamer 🎮
- Filmes 🎬 séries 📺
- Pai de uma golden 🐕

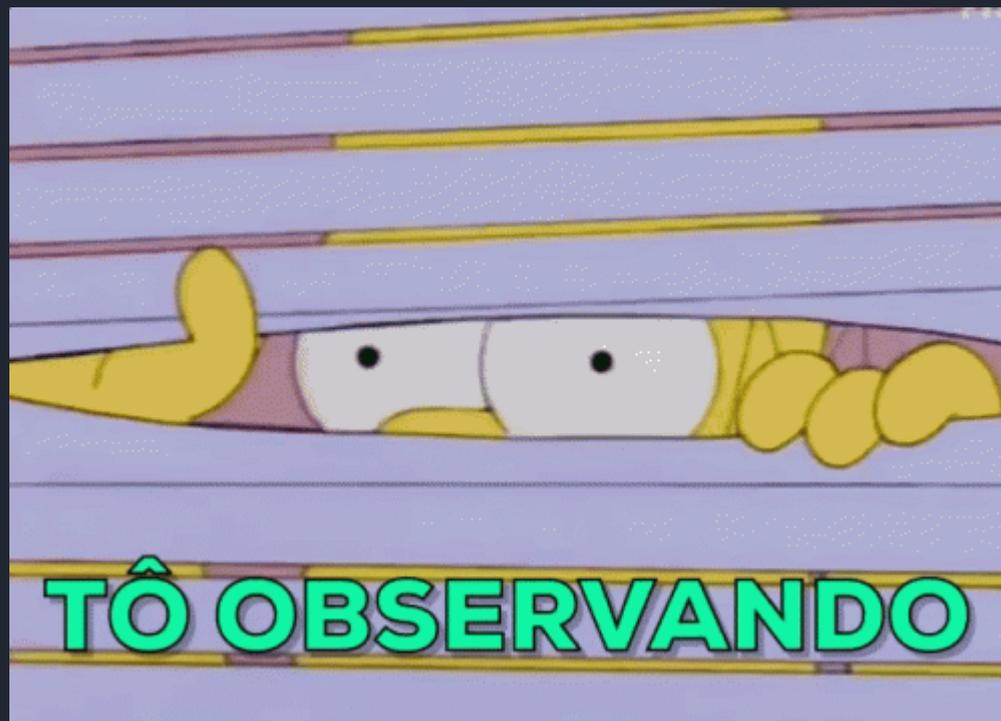


Monitoramento



O monitoramento é o ato de coletar, processar e exibir métricas e dados quantitativos

Observabilidade



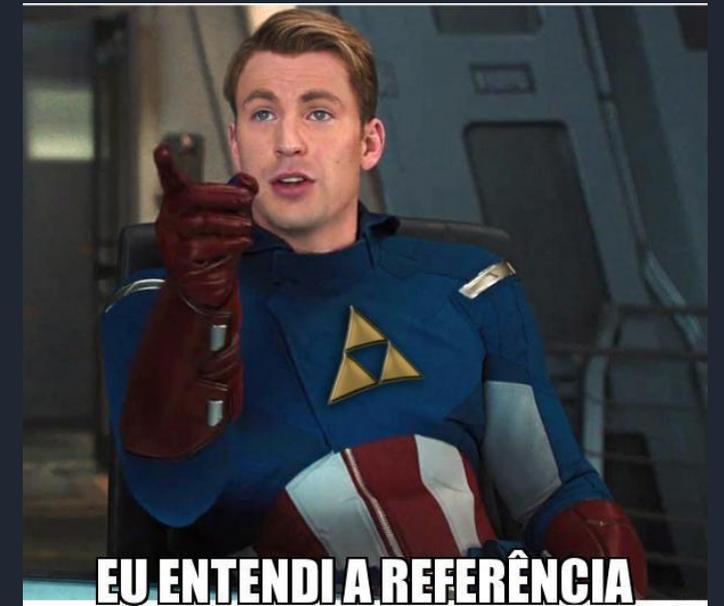
- Não limita-se apenas a métricas e dados coletados
- Evolução do monitoramento 2.0
- Nos permite compreender um sistema a partir do exterior, permitindo fazer perguntas sobre o sistema sem conhecer o seu funcionamento interno.

Observabilidade



Monitoramento

Observabilidade



Então não preciso de monitoramento se tenho observabilidade? Preciso
Sim

Eles se somam? Sim

Monitoramento

Pilares da Observabilidade

1

Métricas

Métricas ajudam a avaliar o desempenho da aplicação e ou host
CPU, memória latência

2

Tracing

Rastrear requisições e transações em seus microsserviços
Permite uma melhor resolução de problemas e depuração

3

Logs

Logs para revelam padrões ocultos, troubleshooting de resoluções de problemas
Sem logs, você não é ninguém

Métricas: Medindo o Sucesso do seu Sistema

As métricas são fundamentais para entender o desempenho e eficácia do seu sistema. Métricas coletadas, basta analisar para tomar decisões afim de otimizar a performance do seu Sistema ou host (VM ou Container)

Guia CNCF Landscape

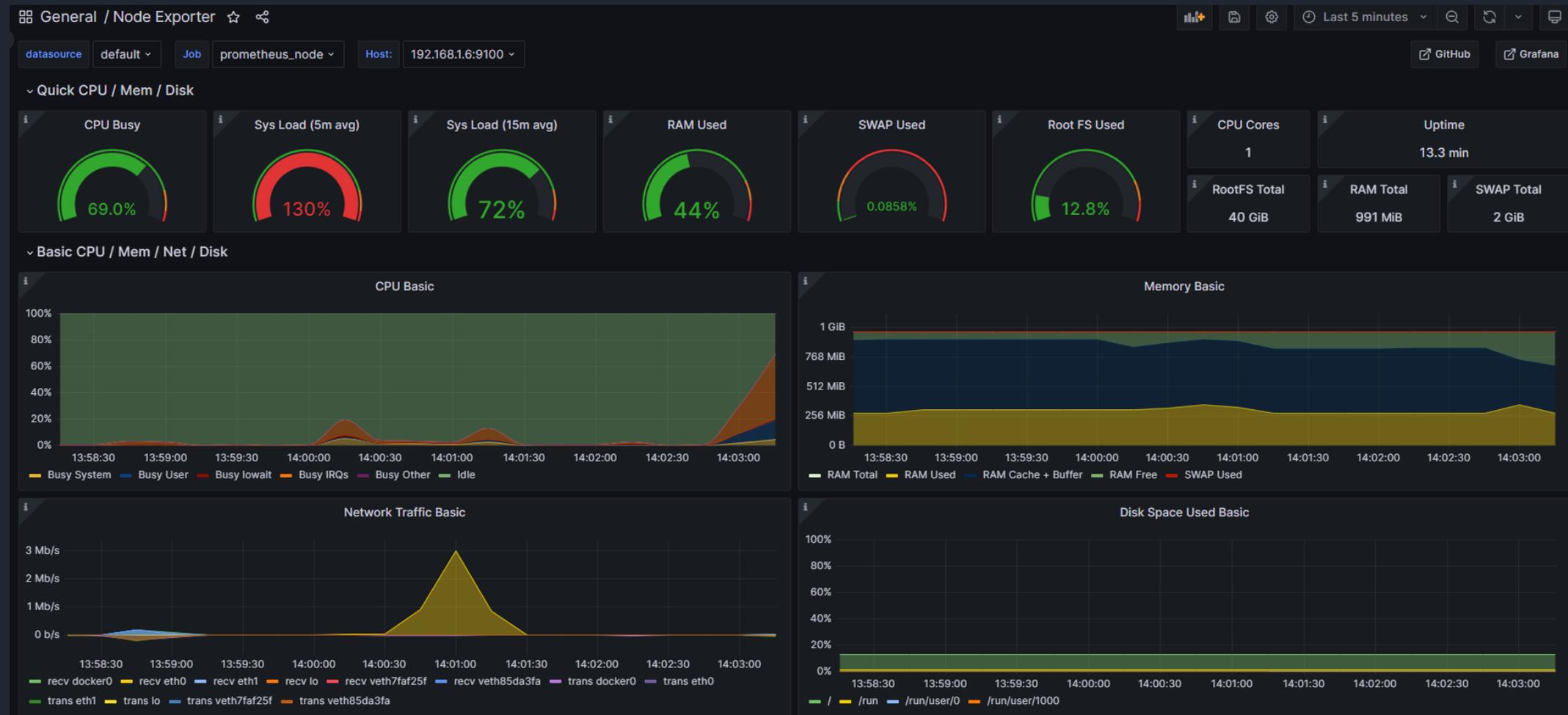
Cloud Native Computing Foundation

Metric

S

 Prometheus CNCF GRADUATED	 cortex CNCF INCUBATING	 OPENMETRICS CNCF INCUBATING	 Thanos CNCF INCUBATING	 Alibaba Cloud Application Real-Time Monitoring Service	 Amazon CloudWatch	 APPDYNAMICS	 Application High Availability Service	 ManageEngine Applications Manager	 AppNeta	 appoptics	 AppSignal	 Aternity	 Azure Monitor	 beats	 bluemator				
				 botkube	 catchpoint	 centreon	 checkmk	 chronosphere	 监控宝	 DATADOG	 DeepFlow	 dynatrace	 epsagon	 falcon	 FLOWMILL				
 FONIO	 foresight	 Google Stackdriver	 Gradle	 Grafana	 Grafana Mimir	 Grafana Phlare	 graphite	 观测云	 HEADLAMP	 Honeybadger	 HUBBLE	 ICINGA	 influxdata	 INSPEKTOR GADGET	 INSTANA	 IRONdb	 K8SGPT	 Keep	 KEPLER
 kiali	 kuberhealthy	 Last9	 LeanIX	 LinDB	 LogicMonitor	 logz.io	 M3	 mackerel	 MICROMETER	 Nagios	 NETDATA	 Netis	 New Relic	 NexClipper	 Nightingale	 NODESOURCE	 OPENTSDb	 opstrace	 OverOps
 PIXIE	 Promscale	 replex	 ROOKOUT	 Sensu	 SENTRY	 SIDEKICK	 SignalFx	 Skooner	 SOSIVIO	 StackState	 sysdig	 听云	 TRaaS BOS	 trickster	 VECTOR	 VICTORIA METRICS	 virtasant	 vmware Aria Operations	 CloudHealth
 weave scope	 WhaTap	 ZABBIX																	

Metrics



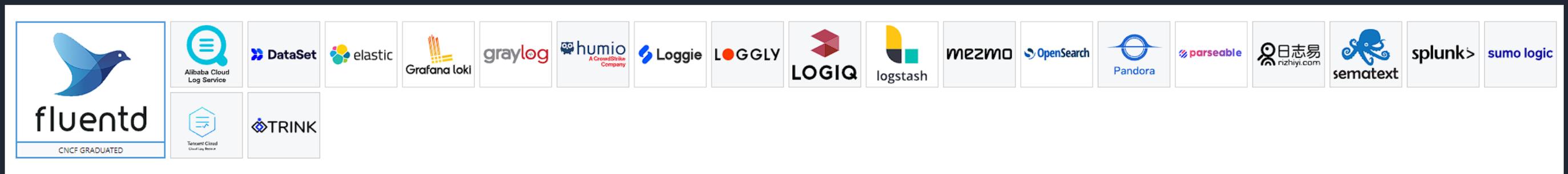
Análise de Logs: Desvendando os Segredos do seu Sistema

Os logs capturam vários eventos que acontecem no sistema, como ações com falha ou sucesso. Como por exemplo um diário

Guia CNCF Landscape

Cloud Native Computing Foundation

Logging



Logging

```
2015-10-17 15:45:11,258 INFO [main] org.apache.hadoop.metrics2.impl.MetricsConfig: loaded properties from hadoop-metrics2.properties
2015-10-17 15:45:11,399 INFO [main] org.apache.hadoop.metrics2.impl.MetricsSystemImpl: Scheduled snapshot period at 10 second(s).
2015-10-17 15:45:11,399 INFO [main] org.apache.hadoop.metrics2.impl.MetricsSystemImpl: MapTask metrics system started
2015-10-17 15:45:11,430 INFO [main] org.apache.hadoop.mapred.YarnChild: Executing with tokens:
2015-10-17 15:45:11,430 INFO [main] org.apache.hadoop.mapred.YarnChild: Kind: mapreduce.job, Service: job_1445062781478_0015, Ident: (org.apache$
2015-10-17 15:45:11,602 INFO [main] org.apache.hadoop.mapred.YarnChild: Sleeping for 0ms before retrying again. Got null now.
2015-10-17 15:45:12,196 INFO [main] org.apache.hadoop.mapred.YarnChild: mapreduce.cluster.local.dir for child: /tmp/hadoop-msrabi/nm-local-dir/u$
2015-10-17 15:45:12,711 INFO [main] org.apache.hadoop.conf.Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session$
2015-10-17 15:45:13,602 INFO [main] org.apache.hadoop.yarn.util.ProcfsBasedProcessTree: ProcfsBasedProcessTree currently is supported only on Li$
2015-10-17 15:45:13,618 INFO [main] org.apache.hadoop.mapred.Task: Using ResourceCalculatorProcessTree : org.apache.hadoop.yarn.util.WindowsBas$
2015-10-17 15:45:14,008 INFO [main] org.apache.hadoop.mapred.MapTask: Processing split: hdfs://msra-sa-41:9000/pageinput2.txt:402653184+134217728
2015-10-17 15:45:14,102 INFO [main] org.apache.hadoop.mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2015-10-17 15:45:14,102 INFO [main] org.apache.hadoop.mapred.MapTask: mapreduce.task.io.sort.mb: 100
2015-10-17 15:45:14,102 INFO [main] org.apache.hadoop.mapred.MapTask: soft limit at 83886080
2015-10-17 15:45:14,102 INFO [main] org.apache.hadoop.mapred.MapTask: bufstart = 0; bufvoid = 104857600
2015-10-17 15:45:14,102 INFO [main] org.apache.hadoop.mapred.MapTask: kvstart = 26214396; length = 6553600
2015-10-17 15:45:14,118 INFO [main] org.apache.hadoop.mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuf$
2015-10-17 15:45:17,305 INFO [main] org.apache.hadoop.mapred.MapTask: Spilling map output
2015-10-17 15:45:17,305 INFO [main] org.apache.hadoop.mapred.MapTask: bufstart = 0; bufend = 48271024; bufvoid = 104857600
2015-10-17 15:45:17,305 INFO [main] org.apache.hadoop.mapred.MapTask: kvstart = 26214396(104857584); kvend = 17310640(69242560); length = 890375$
2015-10-17 15:45:17,305 INFO [main] org.apache.hadoop.mapred.MapTask: (EQUATOR) 57339776 kvi 14334940(57339760)
2015-10-17 15:45:26,696 INFO [SpillThread] org.apache.hadoop.mapred.MapTask: Finished spill 0
2015-10-17 15:45:26,696 INFO [main] org.apache.hadoop.mapred.MapTask: (RESET) equator 57339776 kv 14334940(57339760) kvi 12140764(48563056)
2015-10-17 15:45:30,603 INFO [main] org.apache.hadoop.mapred.MapTask: Spilling map output
2015-10-17 15:45:30,603 INFO [main] org.apache.hadoop.mapred.MapTask: bufstart = 57339776; bufend = 743078; bufvoid = 104857600
2015-10-17 15:45:30,603 INFO [main] org.apache.hadoop.mapred.MapTask: kvstart = 14334940(57339760); kvend = 5428644(21714576); length = 8906297/$
2015-10-17 15:45:30,603 INFO [main] org.apache.hadoop.mapred.MapTask: (EQUATOR) 9811814 kvi 2452948(9811792)
2015-10-17 15:45:39,525 INFO [SpillThread] org.apache.hadoop.mapred.MapTask: Finished spill 1
2015-10-17 15:45:39,525 INFO [main] org.apache.hadoop.mapred.MapTask: (RESET) equator 9811814 kv 2452948(9811792) kvi 244148(976592)
2015-10-17 15:45:43,307 INFO [main] org.apache.hadoop.mapred.MapTask: Spilling map output
2015-10-17 15:45:43,307 INFO [main] org.apache.hadoop.mapred.MapTask: bufstart = 9811814; bufend = 58036090; bufvoid = 104857600
2015-10-17 15:45:43,307 INFO [main] org.apache.hadoop.mapred.MapTask: kvstart = 2452948(9811792); kvend = 19751904(79007616); length = 8915445/6$
2015-10-17 15:45:43,307 INFO [main] org.apache.hadoop.mapred.MapTask: (EQUATOR) 67104842 kvi 16776204(67104816)
```



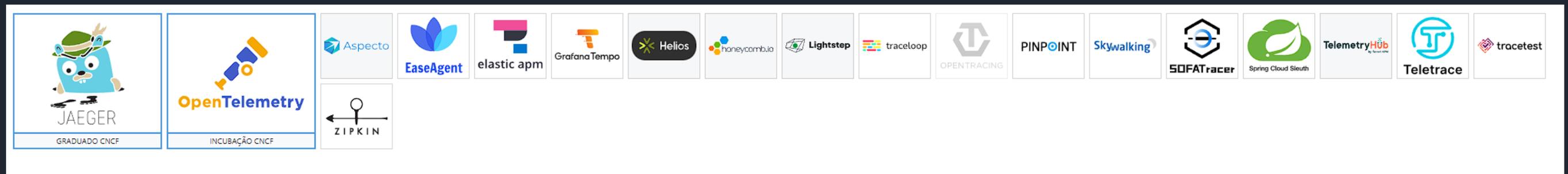
Tracing: Seguindo o Fluxo do seu Sistema

A rastreabilidade é fundamental para entender como um sistema funciona e como melhorá-lo. Coletar e analisar informações de tracing, podemos encontrar gargalos e otimizar a performance do seu sistema.

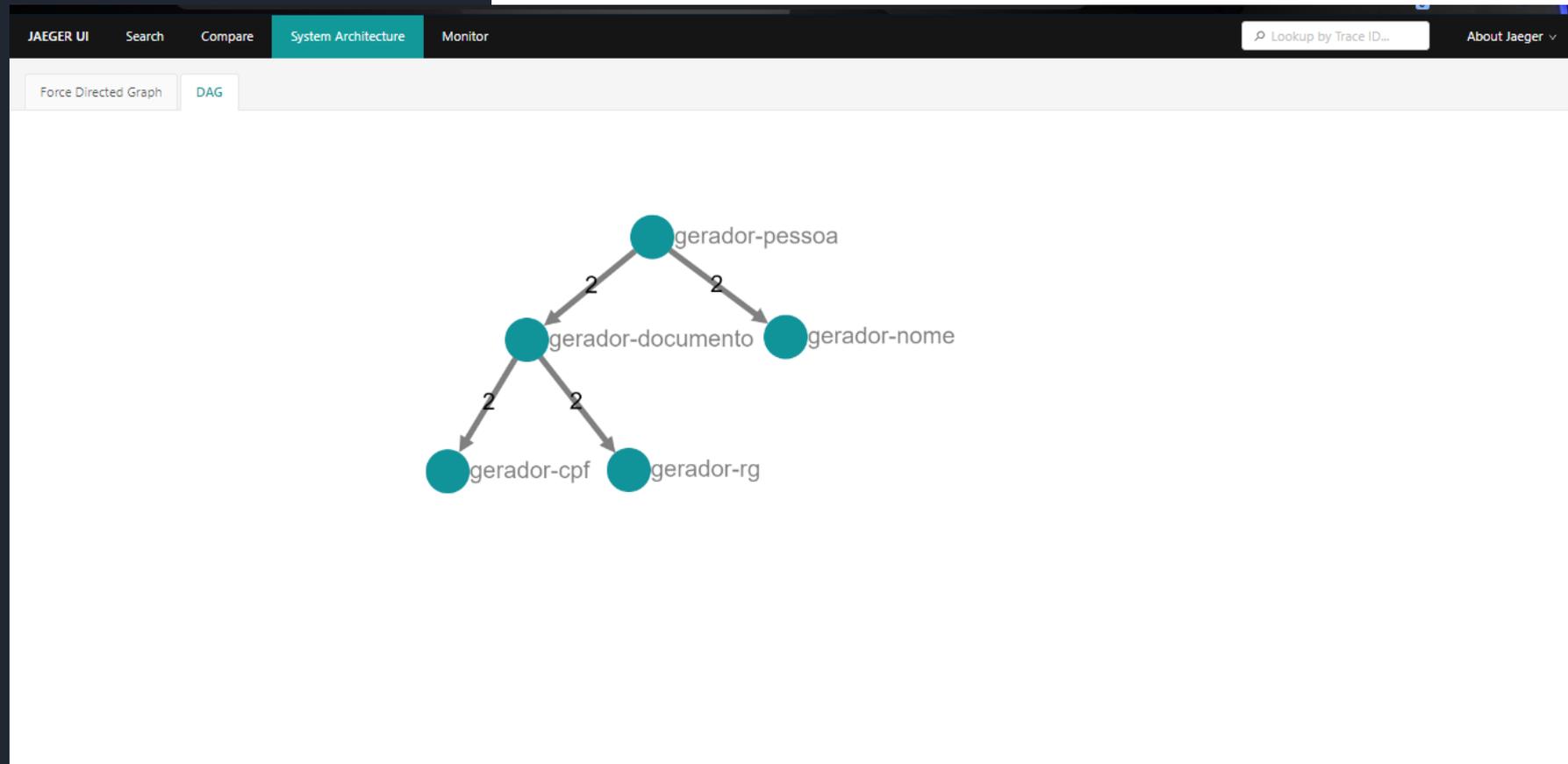
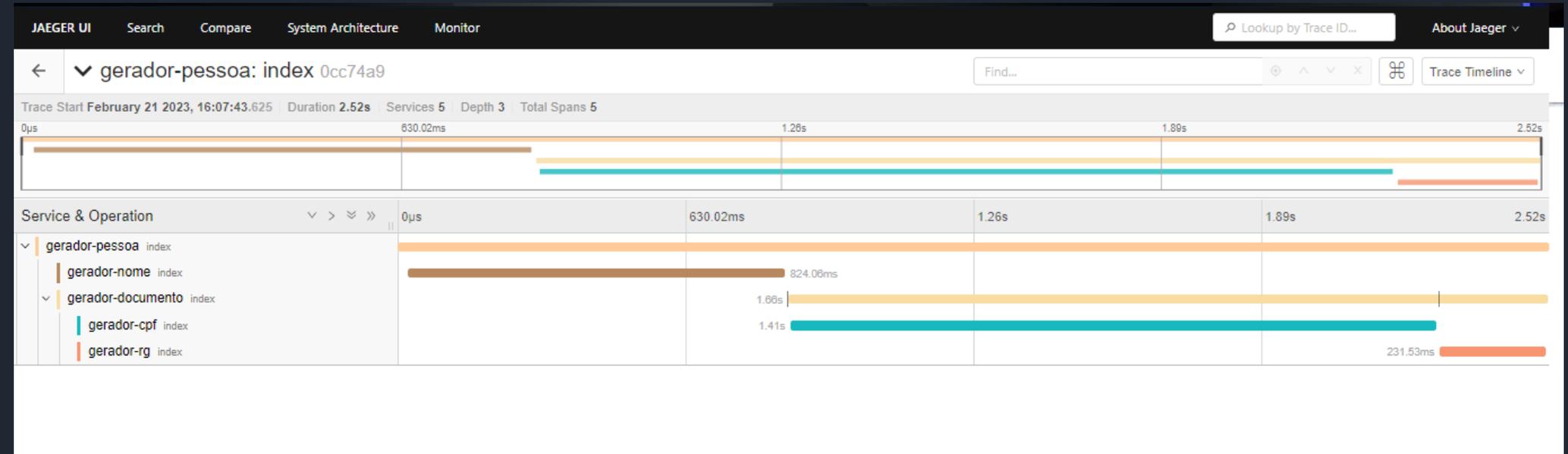
Guia CNCF Landscape

Cloud Native Computing Foundation

Tracing



Tracing





kube-prometheus

Components included in this package:

- The Prometheus Operator
- Highly available Prometheus
- Highly available Alertmanager
- Prometheus node-exporter
- Prometheus Adapter for Kubernetes Metrics APIs
- kube-state-metrics
- Grafana





OpenTelemetry

O que é OpenTelemetry ?

OpenTelemetry é um framework e kit de ferramentas de observabilidade projetado para criar e gerenciar dados de telemetria, como rastreamentos , métricas e logs .

Por que OpenTelemetry ?

No contexto do software, isso significa ser capaz de compreender o estado interno de um sistema examinando os dados que incluem rastreamentos, métricas e logs.

O que não é OpenTelemetry?

Não é um back-end de observabilidade como Jaeger, Prometheus. Seu foco é na geração, coleta, gerenciamento e exportação de dados de telemetria. O armazenamento e a visualização desses dados são intencionalmente deixados para outras ferramentas.

SLA, SLO e SLI

1

SLA (Service Level Agreement)

Acordo entre o provedor e o cliente, estabelecendo os níveis de serviço que devem ser cumpridos. Contrato com cláusulas legais e financeiras

2

SLO (Service Level Objective)

Meta específica de desempenho que o provedor de serviços se compromete alcançar para atender ao SLA.

3

SLI (Service Level Indicator)

O SLI é uma métrica quantitativa que mede o desempenho de um serviço em relação ao SLO estabelecido.

Revelando The Four Golden Signals

Latência

Tempo de resposta de uma requisição. Crítico, pois é percebido diretamente pelo usuário.

Erros

Explore por que ocorrem erros e falhas, e descubra estratégias eficazes para detectá-los, categorizá-los e resolvê-los.

Tráfego

Volume de demandas que uma aplicação ou serviço recebe. Ajuda a entender padrões de uso.

Saturação

Mede o quão sobrecarregado um serviço está em relação aos recursos disponíveis. Índice de necessidade de escalabilidade.

Revelando The Four Golden Signals

Tráfego

- Mede usando a quantidade de requisições por segundo.
- Separa requisições em grupos específicos para análise mais detalhada.

Latência

- Mede o tempo de respostas das requisições.
- Separa latência de requisições bem e malsucedidas.

Erros

- Categoriza códigos de status HTTP (404, 500, etc.)
- Identifica exceções lançadas pela aplicação.
- Importância de seguir boas práticas de tratamento e propagação de erros.

Saturação

- Mede sobrecarga de recursos (CPU, memória, disco, rede).
- Indica possíveis gargalos antes que se tornem problemas visíveis para o usuário.
- Pode ser um indicador de necessidade de escalabilidade.

Importância dos SLs

- São essenciais para medir e estabelecer metas claras de qualidade.
- Alinham o desempenho e a disponibilidade do produto com as expectativas dos usuários.
- Diferenciam um serviço "bom" de um serviço "excelente" no contexto de DevOps.

Como Começar

- Identificar o que é mais importante para os usuários (tempo de carregamento, disponibilidade, etc.).
- Começar com duas ou três métricas como ponto de partida.
- Envolvimento de toda a equipe (Dev, Infraestrutura, QA) na definição de SLIs e SLOs
- Definir metas realistas reconhecendo que 100% de excelência pode ser difícil de alcançar.

Alertas e Respostas a Incidentes

Configurar alertas com base nos dados observados para detectar e responder rapidamente a incidentes.

É o famoso evitar do meu cliente me avisar que o sistema caiu ou está lento

nunca devemos criar alertas com logs



Receiver integration

- <discord_config>
- <email_config>
- <msteams_config>
- <opsgenie_config>
- <pagerduty_config>
- <pushover_config>
- <slack_config>
- <sns_config>
- <telegram_config>
- <victorops_config>
- <webhook_config>
- <wechat_config>
- <webex_config>

Alertas e Respostas a Incidentes



Azure

Descubra as ferramentas de observabilidade no Azure para obter insights valiosos sobre seus sistemas distribuídos baseados em nuvem.

The screenshot displays the Azure Observability tool catalog, organized into three main sections: Detection, triage, and diagnosis; Insights; and Monitoring tools. Each tool is represented by an icon and a label, with some including a 'PARTNER' or 'PREVIEW' badge.

- Detection, triage, and diagnosis**
 - Alerts
 - Diagnostic settings
 - Metrics
 - Azure Workbooks
 - Autoscale
 - Log Analytics workspaces
 - Azure Monitor workspaces
 - Change Analysis
 - Managed Prometheus
 - Azure Native Dynatrace Service **PARTNER**
- Insights**
 - Application Insights
- Monitoring tools**
 - Activity log
 - Log Analytics query packs **PREVIEW**
 - Azure Monitors for SAP solutions
 - Service Health
 - Datadog - An Azure Native ISV Service **PARTNER**
 - Azure Managed Grafana
 - Azure Native New Relic Service **PARTNER**
 - Elastic Cloud (Elasticsearch) – An Azure Native ISV... **PARTNER**
 - Monitor ★
 - Network Watcher

AWS

Explore as ferramentas de observabilidade da AWS para compreender e gerenciar as interações complexas em um cluster Kubernetes.

Serviços relacionados



Amazon CloudWatch

Capacidade de observação dos seus recursos da AWS e aplicativos na AWS e no local



AWS X-Ray

Execute o rastreamento distribuído em várias aplicações e sistemas para ajudar a encontrar a latência em um sistema e direcioná-lo para melhorias.



Amazon Managed Grafana

Visualização de dados escalável, segura e altamente disponível para suas métricas operacionais



Amazon Managed Service for Prometheus

Monitoramento gerenciado, seguro e altamente disponível para seus contêineres

GCP

Aprenda como a observabilidade no GCP permite capturar e analisar métricas em uma arquitetura serverless para obter maior eficiência.

Operações

[Saiba mais](#)

Cloud Debugger

Inspeção de estado do aplicativo em tempo real e depuração na produção.

Cloud Logging

Gerenciamento de registros de auditoria, plataforma e aplicativos no Google Cloud.

Cloud Monitoring

Integridade da infraestrutura e do aplicativo com métricas avançadas.

Cloud Profiler

Criador de perfil de alocação heap e de CPU para analisar o desempenho do aplicativo.

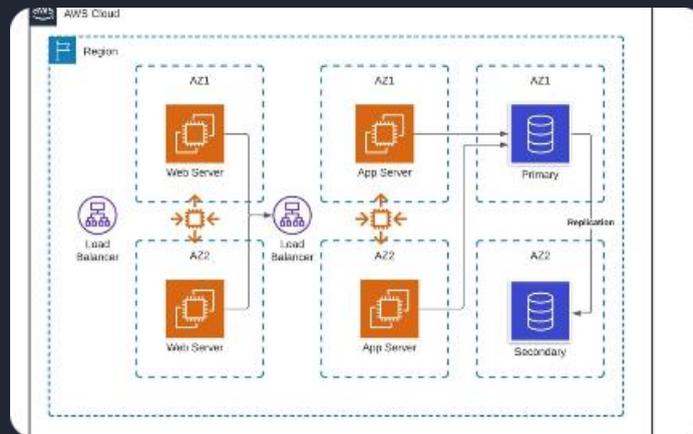
Cloud Trace

Sistema de rastreamento que coleta dados de latência de aplicativos.

Error Reporting

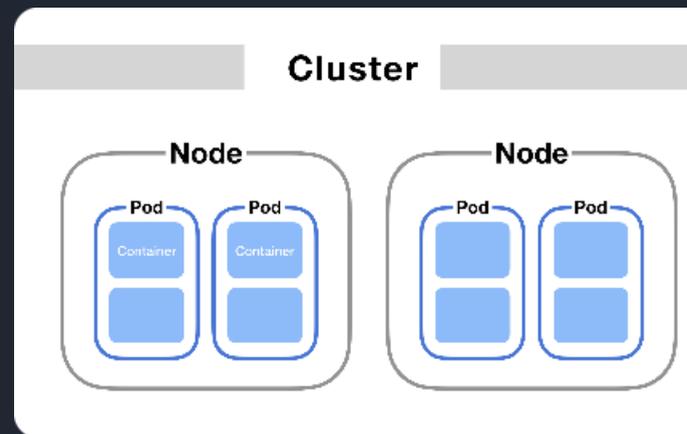
Monitoramento e alertas de exceções em tempo real.

Abraçando a Observabilidade em Arquitetura Cloud Native



Microservices

Torna-se ainda mais crucial no ambiente distribuído de microservices em nuvem.



Orquestração de Contêineres

ajuda a compreender e gerenciar as interações complexas em um cluster Kubernetes.



Computação Serverless

permite capturar e analisar métricas em uma arquitetura serverless para maior eficiência.

Sem observabilidade, não existe DevOps

- Observabilidade é um ciclo infinito
- Se sua aplicação não tem observabilidade, ela não é importante, não é vital pro negócio
- É necessário ter uma equipe dedicada a cada aplicação (**IDEAL**)
- Quem não mede, não gerencia!
- É preciso maturidade para implantar observabilidade
- Não é fácil implantar observabilidade



Cultura de Observabilidade

Importância de uma cultura que valoriza a observabilidade.
Incentive a colaboração entre desenvolvedores, operações e equipes de QA para resolver problemas rapidamente.

Evitar



Computação verde

- ❖ Prática que tem como objetivo minimizar o impacto ambiental associado às atividades de processamento de dados, armazenamento e infraestrutura tecnológica.
- ❖ Os datacenters gastam 2% de **TODA** energia do mundo
- ❖ Explorando esse ponto entre Observabilidade e Computação Verde, práticas conscientes de monitoramento, não apenas melhoram o desempenho e a confiabilidade dos sistemas, mas também contribuem para a redução do impacto ambiental associado à infraestrutura de tecnologia da informação.

1. Otimização de Recursos
2. Detecção de padrões de Anomalias
3. Eficiência Operacional
4. Planejamento de Capacidade

Microsoft testa com **sucesso** datacenter mergulhado no mar





Polêmica

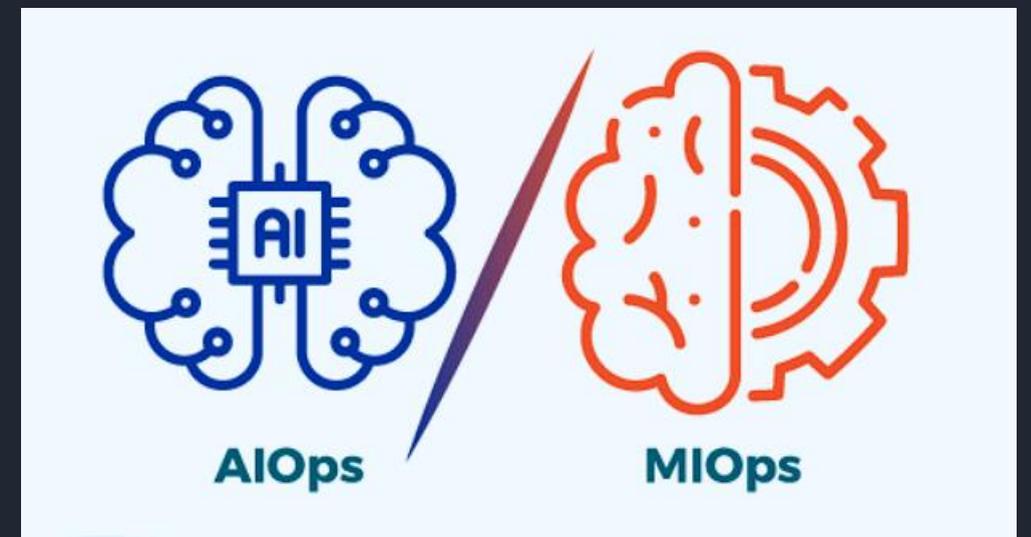
Colocar ou não colocar rate limits?

Ir contra a maré?

MLOps e AIOps: Impulsionando Eficiência e Inteligência

MLOps e AIOps estão revolucionando operações, aproveitando o aprendizado de máquina e a inteligência artificial. Automatizando os sistemas, melhora o desempenho e obtem informações valiosas para tomadas de decisões proativas.

Mantenha-se à frente da concorrência com essas tecnologias de ponta.



Ex: prometheus-anomaly-detector

APM: Monitorando e Otimizando o Desempenho do seu Sistema

o APM (Application Performance Monitoring) ajuda a monitorar e otimizar o desempenho do seu sistema. Identificar gargalos, analisar métricas e rastrear transações tudo em um único lugar garantindo uma experiência de usuário excepcional.



Conclusão

Incorporar práticas de observabilidade em seus fluxos de trabalho tem como objetivo de otimizar proativamente o desempenho do sistema, aprimorando a confiabilidade e resolver rapidamente problemas.

clique



“Trabalhe duro até que seus heróis sejam seus companheiros”

Conecte-se comigo para colaborar em diversas plataformas!

Vocês podem me encontrar em



rafaelmaferreira

A horizontal row of six icons on a dark background. From left to right: a white infinity symbol on a white square background; the LinkedIn logo (blue square with white 'in'); the GitHub logo (white octocat on a white circle); the Telegram logo (white speech bubble on a black square); and the Instagram logo (white camera on a multi-colored square).