

Embarcadero Conference 2024

Inovação faz parte do nosso DNA!

Monitorando sua aplicação a partir de métricas.

{Alessandro Medeiros

Exploreemos como implementar monitoramento eficaz em aplicações Delphi usando métricas. Abordaremos as principais ferramentas e técnicas para capturar, analisar e visualizar dados, garantindo que sua aplicação esteja sempre performando ao máximo. Ideal para desenvolvedores que buscam melhorar a observabilidade e resolver problemas antes que eles afetem os usuários.

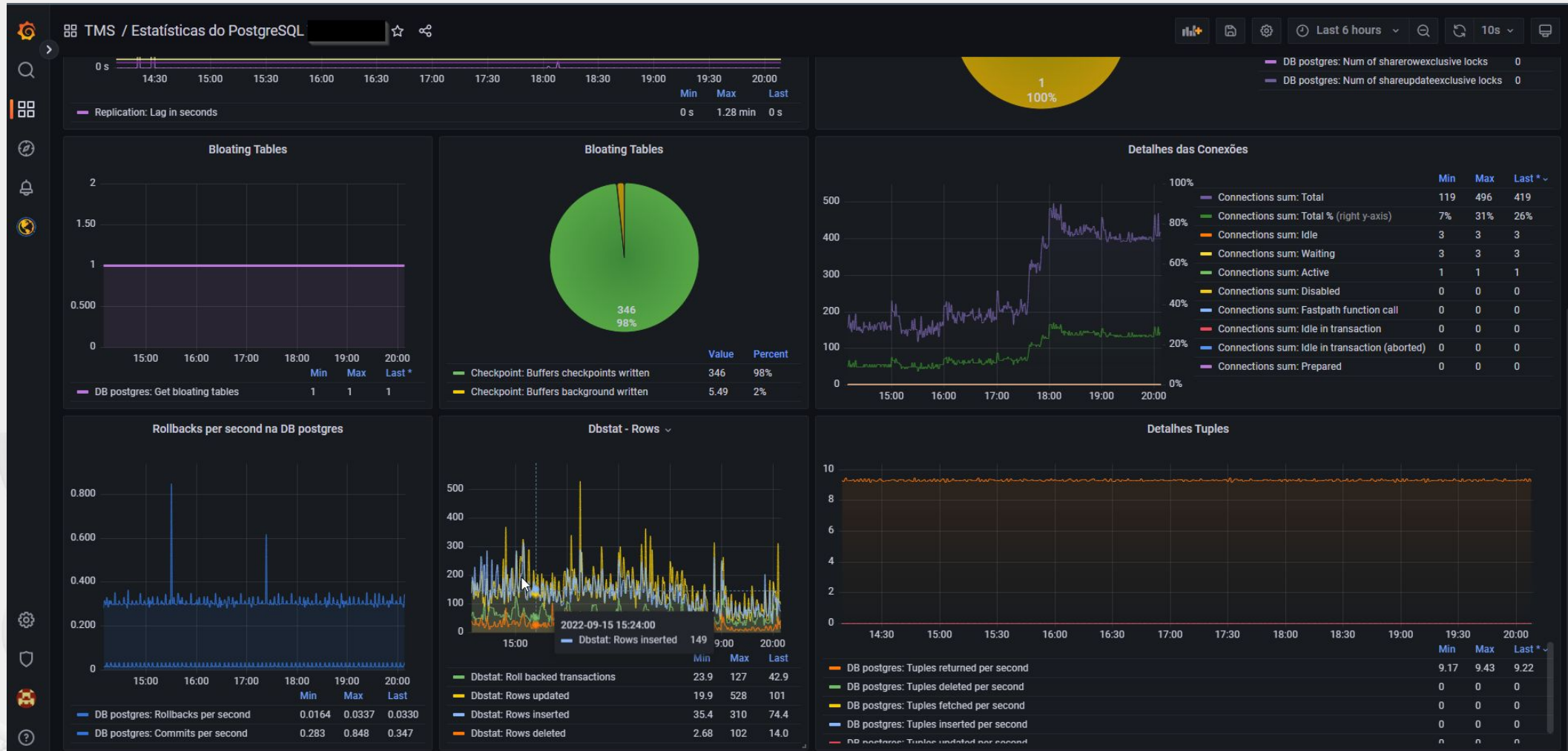


Observabilidade

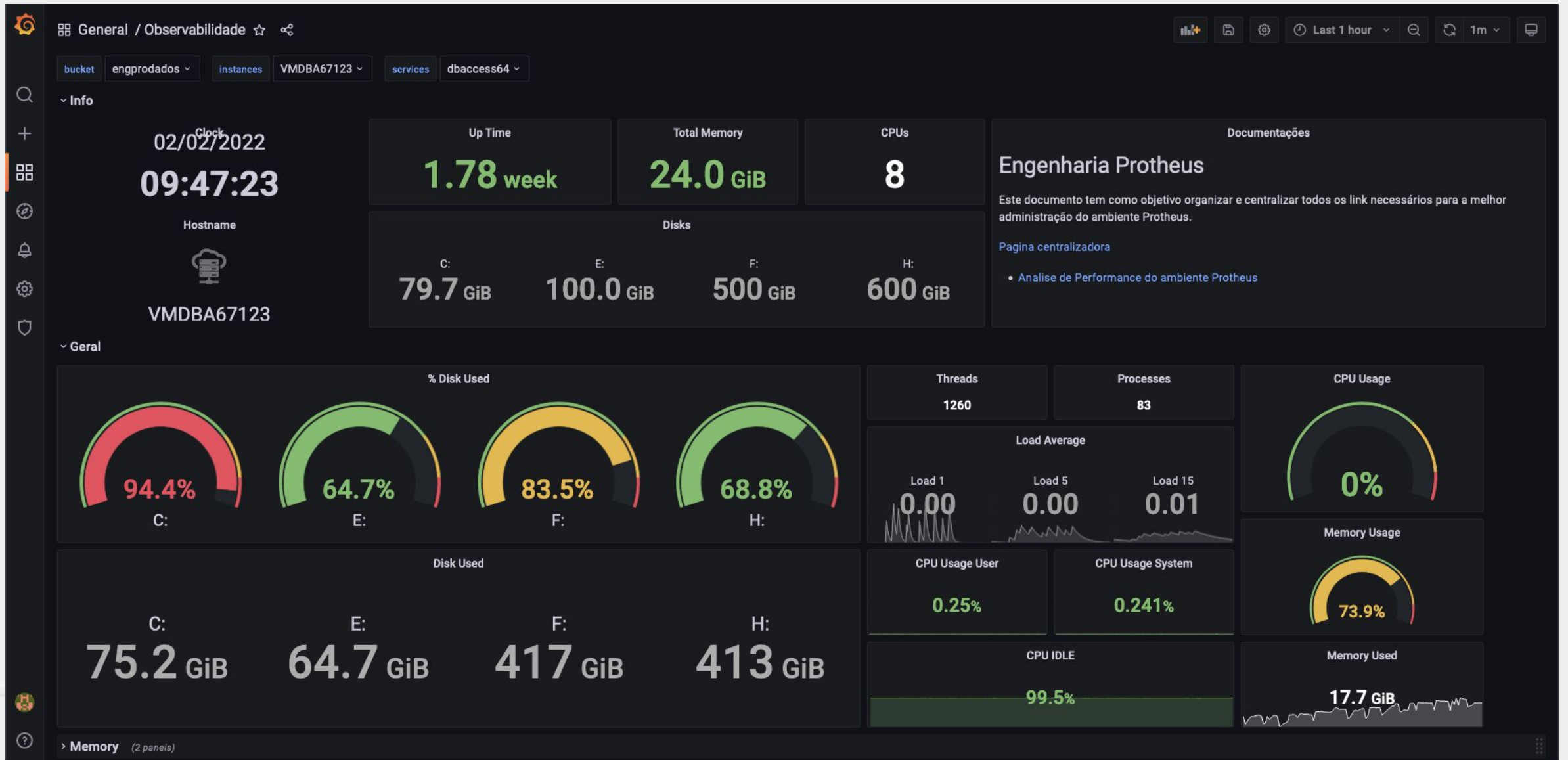
- Observabilidade é a capacidade de uma aplicação ser monitorada de forma que, ao coletar e analisar dados, seja possível entender seu comportamento interno sem precisar alterar o código.
- Vai além do monitoramento tradicional, ao englobar não só a coleta de dados, mas também a capacidade de interpretá-los para prever e solucionar problemas.
- Destaca-se pela importância em ambientes complexos, onde múltiplas dependências e integrações precisam ser continuamente monitoradas.



Monitoramento



Observabilidade



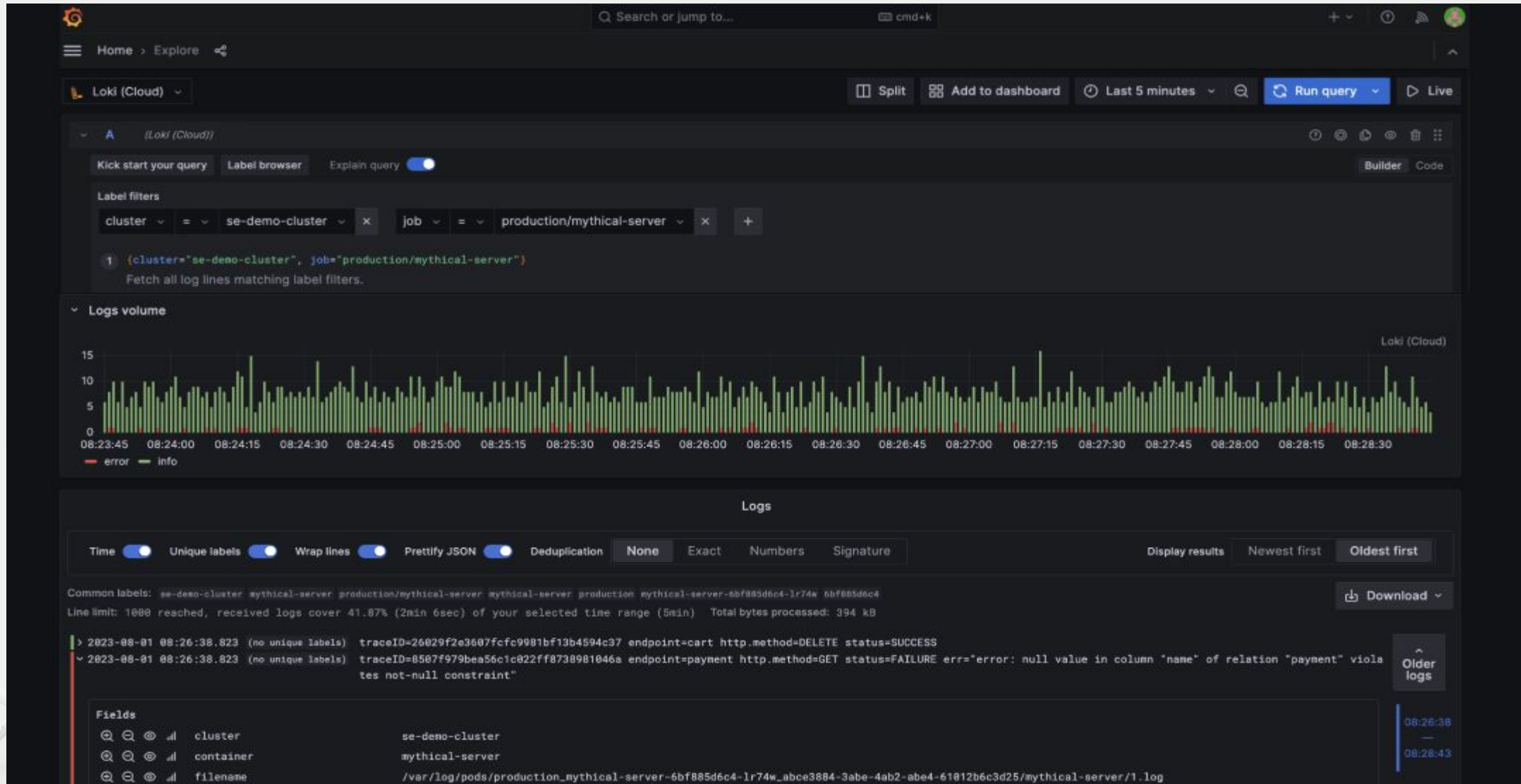
Componentes Essenciais da Observabilidade

- **Logs:** Registros detalhados de eventos que ocorrem na aplicação, permitindo rastrear problemas e atividades.
- **Métricas:** Valores numéricos que representam o estado e o desempenho de diferentes aspectos da aplicação, como o uso de CPU, memória, tempo de resposta, etc.
- **Traces:** Rastreamento das execuções na aplicação, permitindo identificar o fluxo de chamadas entre diferentes partes do sistema e localizar gargalos.



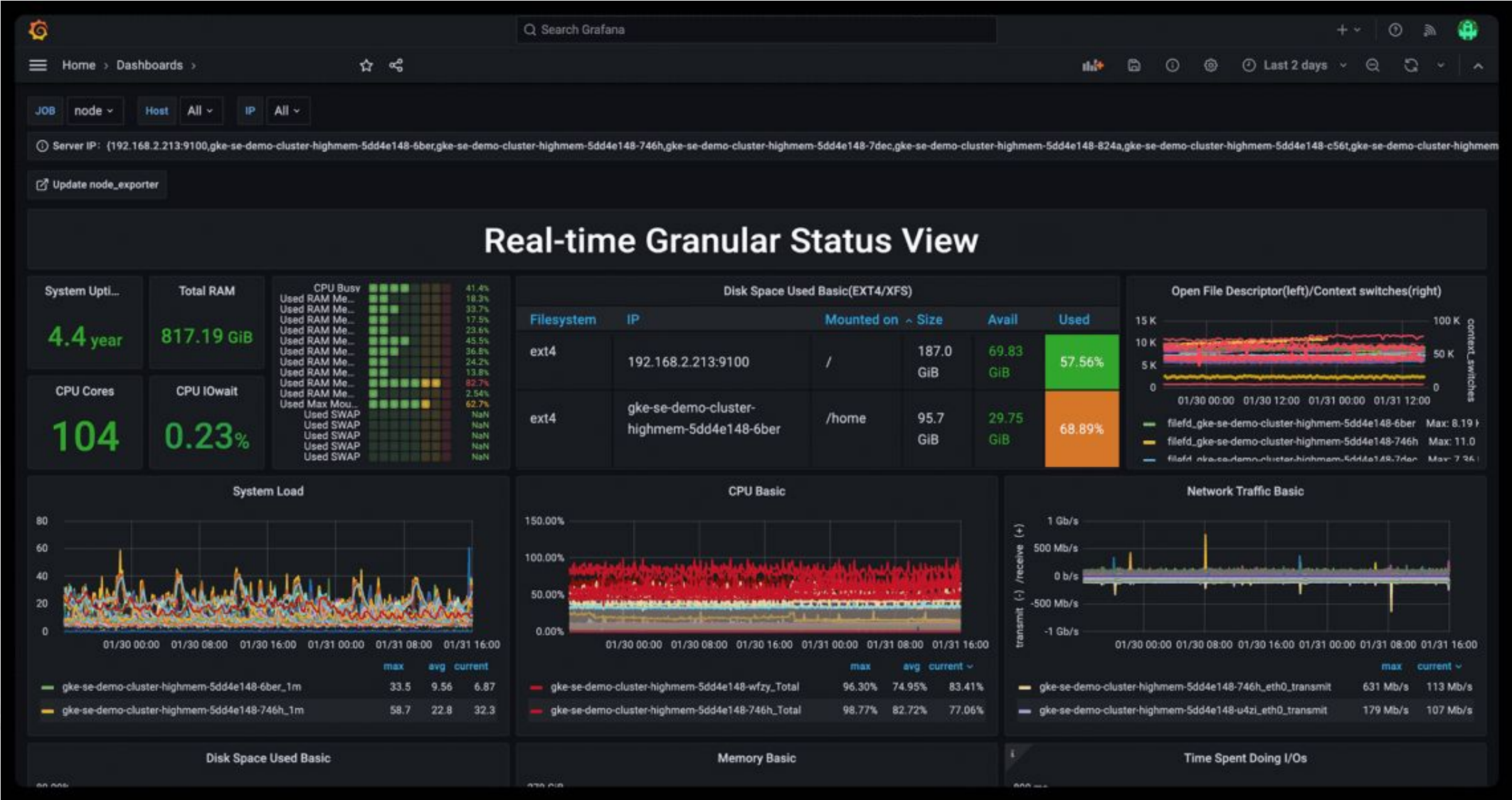
Componentes Essenciais da Observabilidade

Logs



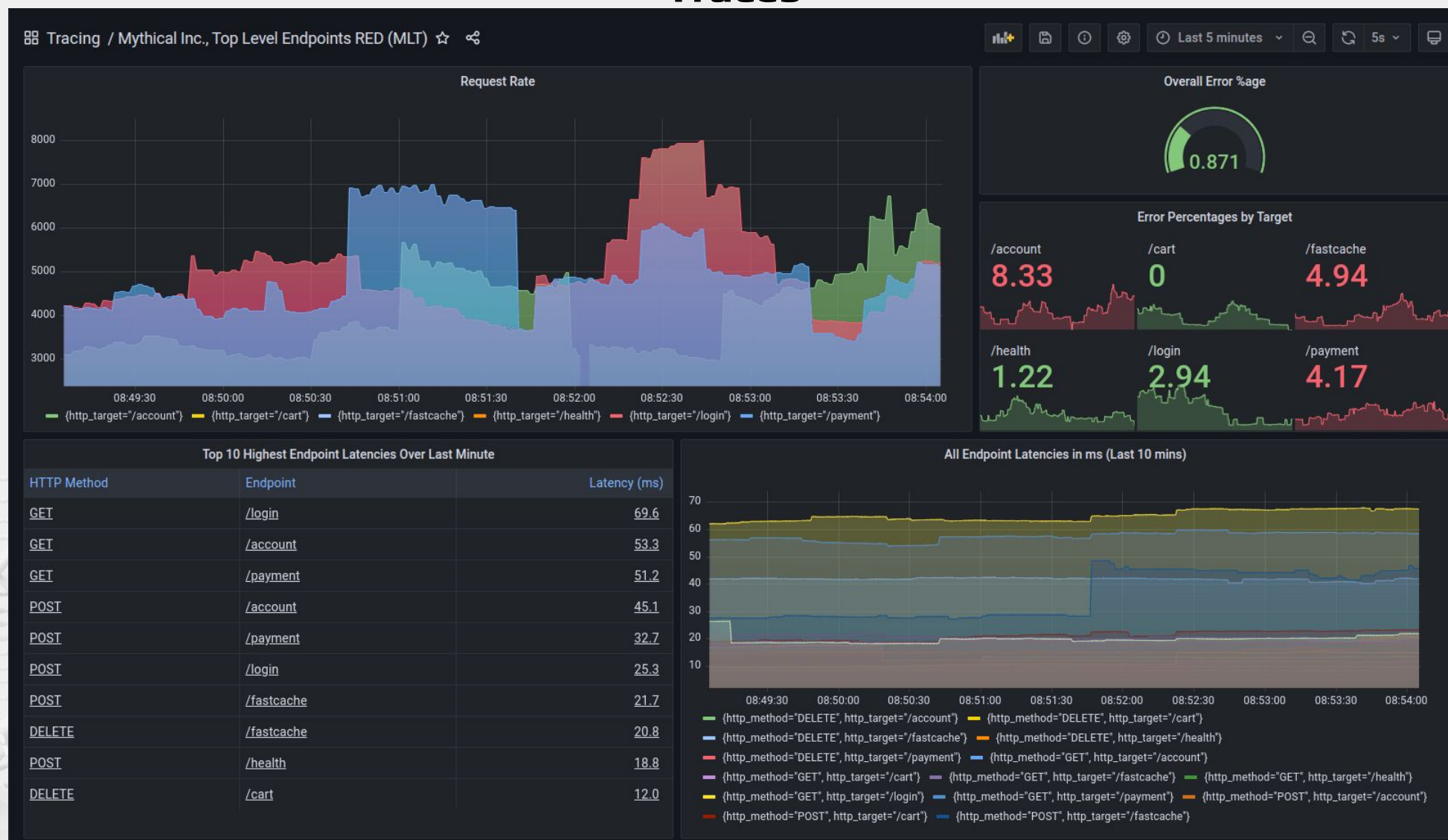
Componentes Essenciais da Observabilidade

Métricas



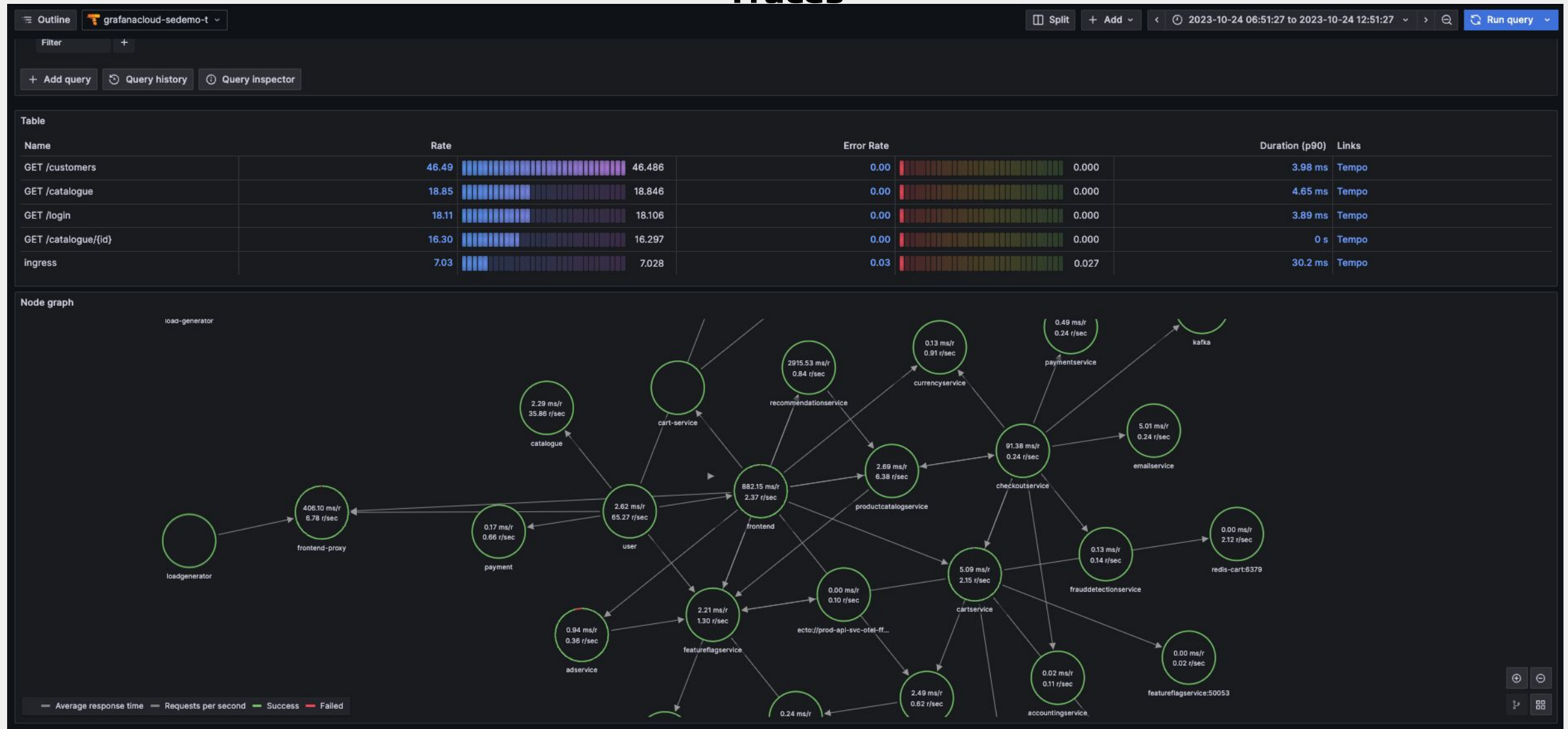
Componentes Essenciais da Observabilidade

Traces



Componentes Essenciais da Observabilidade

Traces

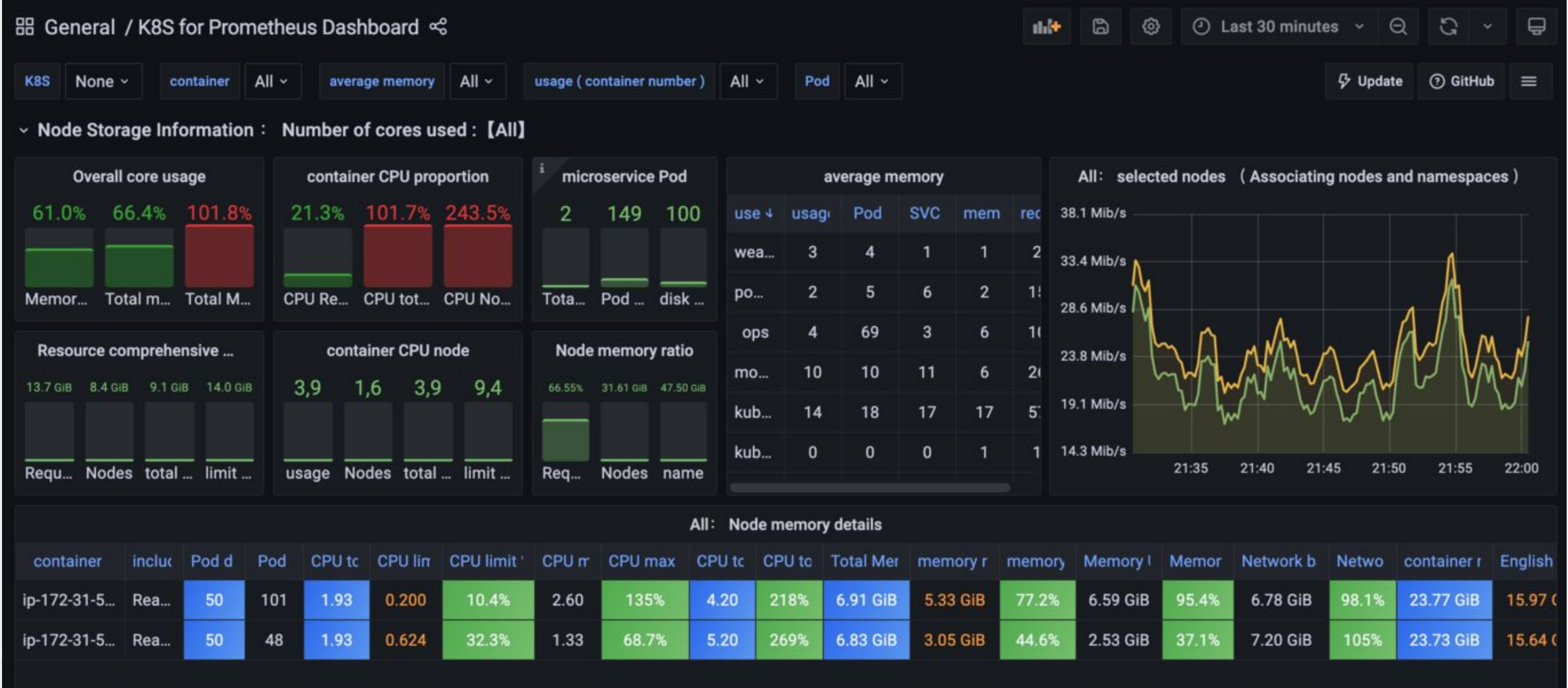


Observabilidade em Aplicações Delphi

- Aplicações Delphi, por serem frequentemente utilizadas em sistemas legados e em áreas como automação comercial, enfrentam desafios específicos para implementar observabilidade.
- Ferramentas disponíveis para Delphi podem ser limitadas, mas existem soluções como o Grafana Loki para logs e a integração com Prometheus e Grafana para métricas e visualização.
- O uso dessas ferramentas permite criar um ambiente monitorado, onde os desenvolvedores podem rapidamente identificar e corrigir problemas.



Monitoramento com Métricas em Delphi



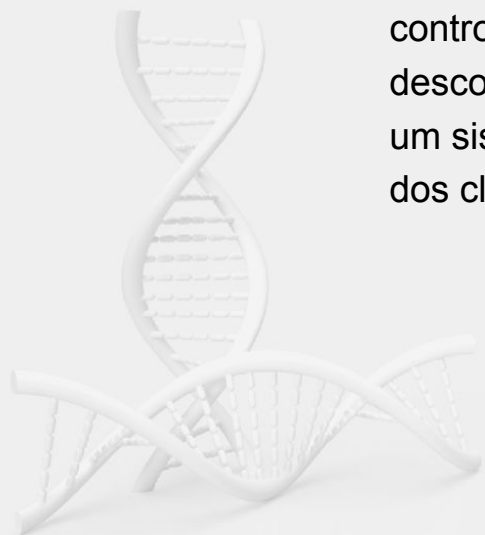
Estudos de Caso e Melhores Práticas

1. Caso 1: Otimização de Desempenho em um Sistema de Automação Comercial

- Uma empresa de software enfrentava lentidão e instabilidade em seu sistema de automação comercial desenvolvido em Delphi. A equipe implementou a observabilidade com logs usando Grafana Loki e métricas com Prometheus. Ao monitorar o uso de CPU e memória, identificaram gargalos causados por consultas SQL ineficientes. Após ajustes, o tempo de resposta foi reduzido em 40%, melhorando a experiência do usuário e reduzindo os custos operacionais.

2. Caso 2: Prevenção de Falhas em um Sistema de Controle de Estoque

- Uma startup de logística, também usando Delphi, estava lidando com falhas intermitentes em seu sistema de controle de estoque. Com a observabilidade ativa, eles utilizaram traces para rastrear a origem do problema, descobrindo que ocorria durante integrações com um serviço externo. Com essa informação, implementaram um sistema de alertas e failover, que reduziu as falhas em 90%, garantindo maior confiabilidade e confiança dos clientes.



Estudos de Caso e Melhores Práticas

- **Melhores Práticas:**
 - **Configuração Proativa:** Configure logs detalhados e monitore métricas essenciais desde o início do projeto.
 - **Dashboards Personalizados:** Utilize Grafana para criar dashboards que mostrem as informações mais críticas para o seu sistema.
 - **Revisão Contínua:** Estabeleça uma rotina de revisão dos dados coletados para ajustes contínuos.

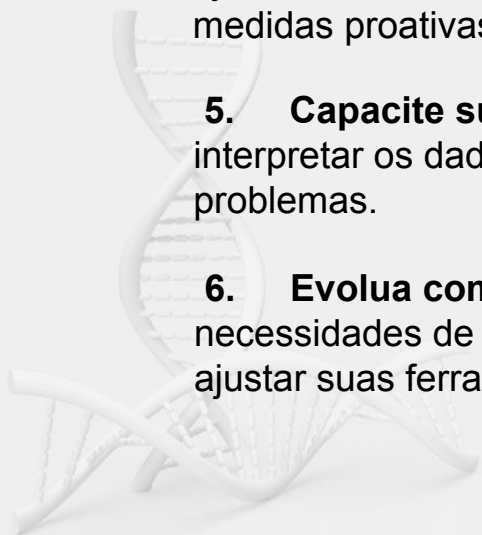


Delphi, na prática



Conclusão e Recomendações

1. **Inicie com o Básico:** Configure logs detalhados usando Grafana Loki e métricas básicas com Prometheus desde o início do desenvolvimento. Isso estabelece uma base sólida para monitoramento contínuo.
2. **Personalize seus Dashboards:** Use o Grafana para criar dashboards que atendam às necessidades específicas do seu projeto. Foco em métricas críticas como tempo de resposta, uso de CPU, e taxa de erros pode fazer a diferença na detecção precoce de problemas.
3. **Implemente Alertas Inteligentes:** Configure alertas que sejam acionados com base em limiares críticos de desempenho e estabilidade. Alertas inteligentes ajudam a prevenir falhas antes que afetem os usuários finais.
4. **Monitore e Ajuste Continuamente:** A observabilidade não é uma tarefa única; exige monitoramento constante e ajustes com base nos dados coletados. Revisar regularmente os logs, métricas e traces permite identificar padrões e tomar medidas proativas.
5. **Capacite sua Equipe:** Garanta que todos na equipe entendam a importância da observabilidade e saibam como interpretar os dados coletados. Treinamento e boas práticas compartilhadas fortalecem a capacidade de resposta da equipe a problemas.
6. **Evolua com as Necessidades do Projeto:** À medida que seu projeto cresce e se torna mais complexo, suas necessidades de observabilidade também aumentarão. Esteja preparado para escalar suas soluções de monitoramento e ajustar suas ferramentas conforme necessário.



Conclusão e Recomendações

Bibliotecas usadas:

Client Prometheus: <https://github.com/marcobreveglieri/prometheus-client-delphi>

Middleware:

<https://github.com/marcobreveglieri/horse-prometheus-metrics>

<https://github.com/dliocode/datalogger>



Sessão de Perguntas e Respostas.



Embarcadero Conference 2024


Inovação faz parte do nosso DNA!



Quer me ver na
#ECON25?
Acesse o QRCode
e avalie minha palestra!




Alessandro Medeiros

 [@ialemedeiros](#) [@academiadocodigo](#)

 [@alepmedeiros](#)

 contato@academiadocodigo.com.br

 [\(22\) 9 8149-8212](tel:(22)98149-8212)

