



# Observabilidade e Rastreabilidade

Entendendo sistemas complexos e distribuídos

Indo além dos logs

# O problema

- Sistemas distribuídos são caixas-pretas
- Logs isolados não explicam falhas complexas
- Precisamos enxergar o caminho da informação/dado

# O que é Observabilidade?

- Capacidade de entender o estado interno do sistema
- Baseado apenas nos dados que ele expõe
- Vai além de monitoramento tradicional

# O que é Rastreabilidade?

- Acompanhar uma requisição ponta a ponta
- Entender dependências e gargalos
- Ver como o sistema realmente funciona

# Os três pilares

- Logs: eventos detalhados
- Métricas: comportamento agregado
- Tracing: o caminho percorrido

# Por que isso importa?

- Debug mais rápido (MTTR menor)
- Menos suposições, mais dados
- Decisões técnicas mais seguras

# Exemplo prático

- Usuário percebe lentidão
- Trace mostra onde a requisição para
- O problema deixa de ser invisível

# Ferramentas – AWS

- CloudWatch: métricas e logs
- X-Ray: rastreamento distribuído
- Integração nativa e rápida

# Ferramentas – Open Source

- *OpenTelemetry*: padrão da indústria
- *Prometheus*: métricas
- *Grafana*: visualização unificada
- Elastic Stack (ElasticSearch, Logstash e Kibana)

# Desafios reais

- Custo e volume de dados
- Instrumentação correta
- Evitar ruído e fadiga de alerta

# O alerta do monolito

- Rastrear tudo gera ruído
- O foco deve ser em fronteiras críticas
- Menos é mais

# Observabilidade é cultura

- Não é responsabilidade só do SRE
- Faz parte do Definition of Done
- Código observável é código saudável

# Conclusão

- Observabilidade traz previsibilidade
- Menos estresse em produção
- Mais confiança para evoluir sistemas

# Próximos passos

- Escolher um serviço como piloto
- Instrumentar com as ferramentas definidas
- Aprender com dados reais

# Q&A

*Vulgo dúvidas?*