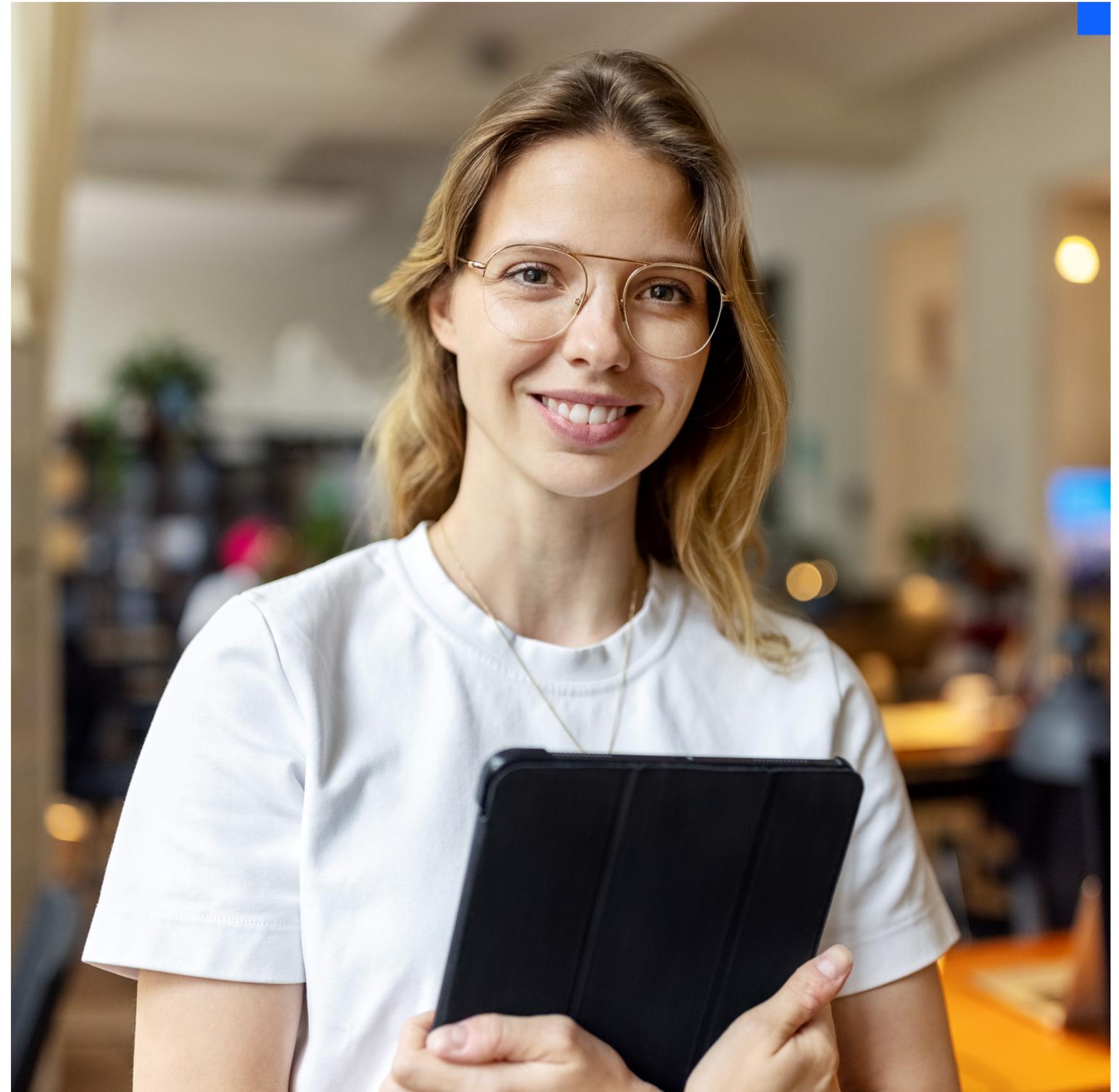


Observabilidade para desenvolvedores

O que é observabilidade?



Conteúdos

01 →
Introdução

02 →
O que é
observabilidade?

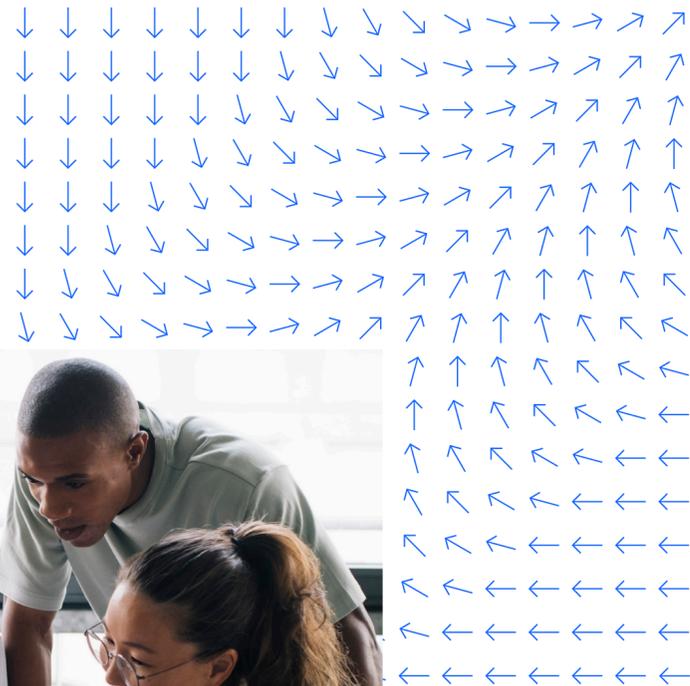
03 →
Como alcançar a
observabilidade total
de ponta a ponta

04 →
Padrões de
observabilidade
e código aberto

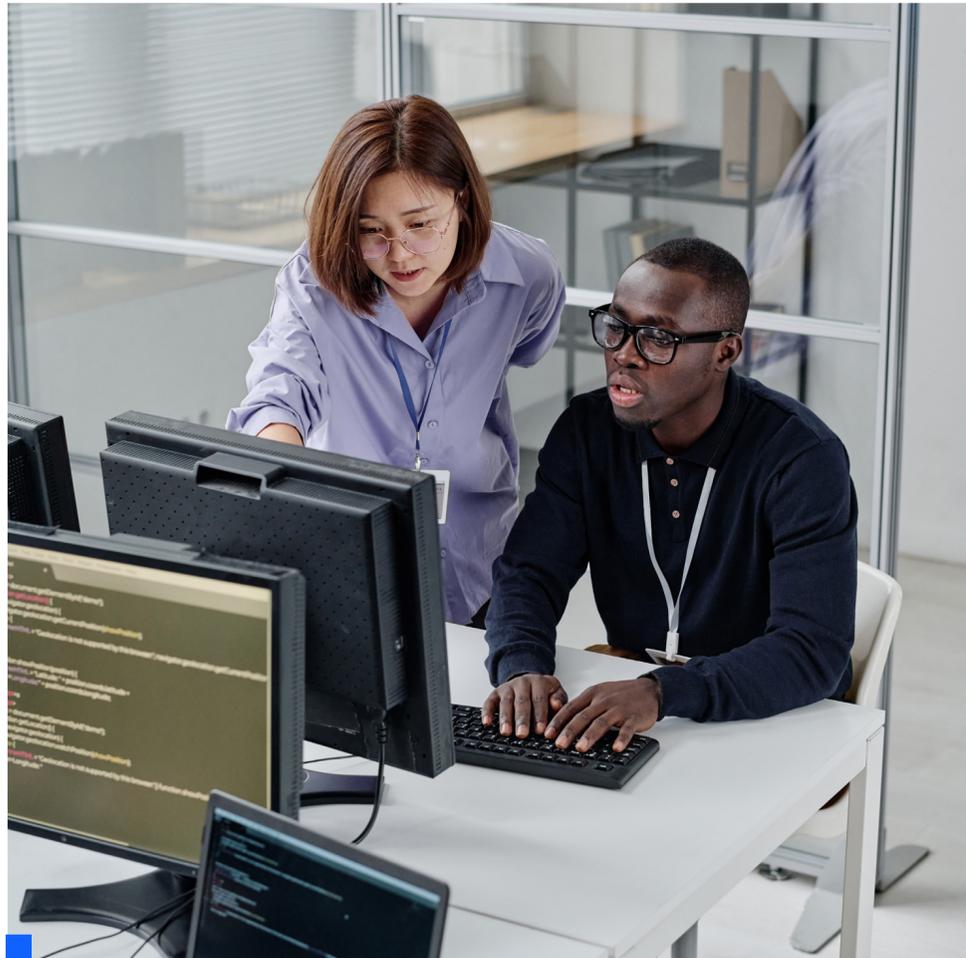
05 →
Metodologia SLO

06 →
Além do código aberto

07 →
O IBM Instana é
adequado para você?



Introdução



Os desenvolvedores enfrentam um crescente desafio: como solucionar problemas de software quando ele é composto por muitos serviços diferentes que são executados em diversas linguagens e plataformas? Como podemos perceber mudanças críticas, analisar a caixa preta de nossos serviços e discernir as verdadeiras causas dos erros?

Não faz muito tempo que, para depurar um programa, normalmente era necessário ficar pesquisando por logs de erros. Essa abordagem era adequada para equipes pequenas que rodavam programas simples e locais com um pequeno número de instâncias.

Mas a TI está sempre mudando. Os paradigmas de arquitetura de software evoluíram de monólitos a microsserviços. As responsabilidades dos desenvolvedores mudaram com o surgimento do DevOps, o que representa uma mudança na forma como os desenvolvedores se responsabilizam pelos programas após a entrega.

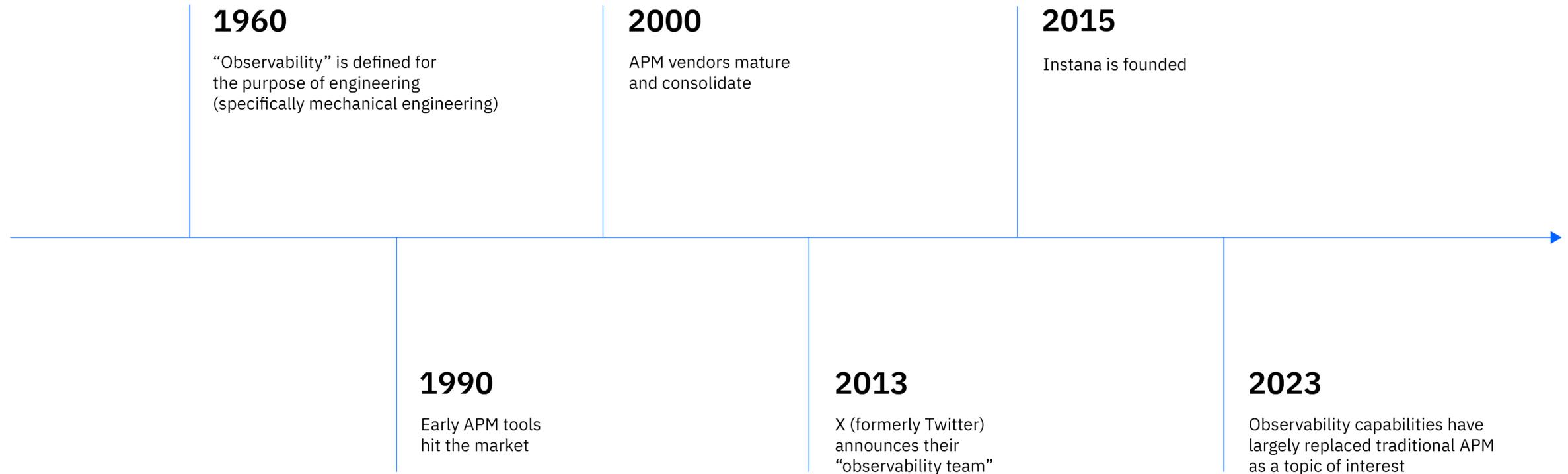
As origens da observabilidade

O termo observabilidade foi definido pela primeira vez para fins de engenharia no artigo de RE Kalman de 1960 intitulado “Sobre a teoria geral de sistemas de controle.”¹ O termo observabilidade, no que se refere à engenharia mecânica, foi definido como sendo a capacidade de entender o estado interno de um sistema medindo seus resultados.

Cinquenta anos depois, os desenvolvedores e profissionais de software ainda monitoravam seus sistemas utilizando ferramentas de instrumentação tediosas, se é que monitoravam mesmo. Já em 2013, a mudança para sistemas distribuídos já estava em curso. Foi quando o X (antes chamado de Twitter) anunciou que estava criando uma nova “equipe de observabilidade” para centralizar e padronizar a coleta de dados de telemetria entre centenas de serviços do Twitter.² O termo observabilidade ganha notoriedade.

Observabilidade hoje

Mais de uma década depois, equipes de TI continuam a lidar com serviços ainda mais detalhados e responsabilidades de trabalho mais multifuncionais. Os microsserviços estão dando lugar para a computação sem servidor, com o DevOps levando à engenharia de confiabilidade local (SRE). A observabilidade continua importante como sempre foi e o escopo do desafio está crescendo exponencialmente.



O que é observabilidade?



Deixando de lado as definições científicas, observabilidade no desenvolvimento de software é observar o que está acontecendo dentro de um aplicativo ou sistema, e identificar o que estamos vendo. Em suma, observabilidade é visibilidade somado com entendimento.

Para os modernos sistemas de software distribuídos, as ferramentas de observabilidade são totalmente obrigatórias. Simplesmente não há outra maneira do desenvolvedor monitorar constantemente a complexidade que surge com a divisão dos aplicativos em muitos pedaços minúsculos. Por sorte, essa arquitetura complexa também é o que viabiliza a existência das ferramentas de observabilidade.

Os sistemas de software viabilizam a observabilidade unificada porque os serviços normalmente têm um plano de controle universal e utilizam linguagens comuns, como HTTP, RPC ou outra qualquer, para se comunicarem entre si. Além disso, seria difícil imaginar ter que depender de logs desatualizados para manter as operações de TI funcionando normalmente.

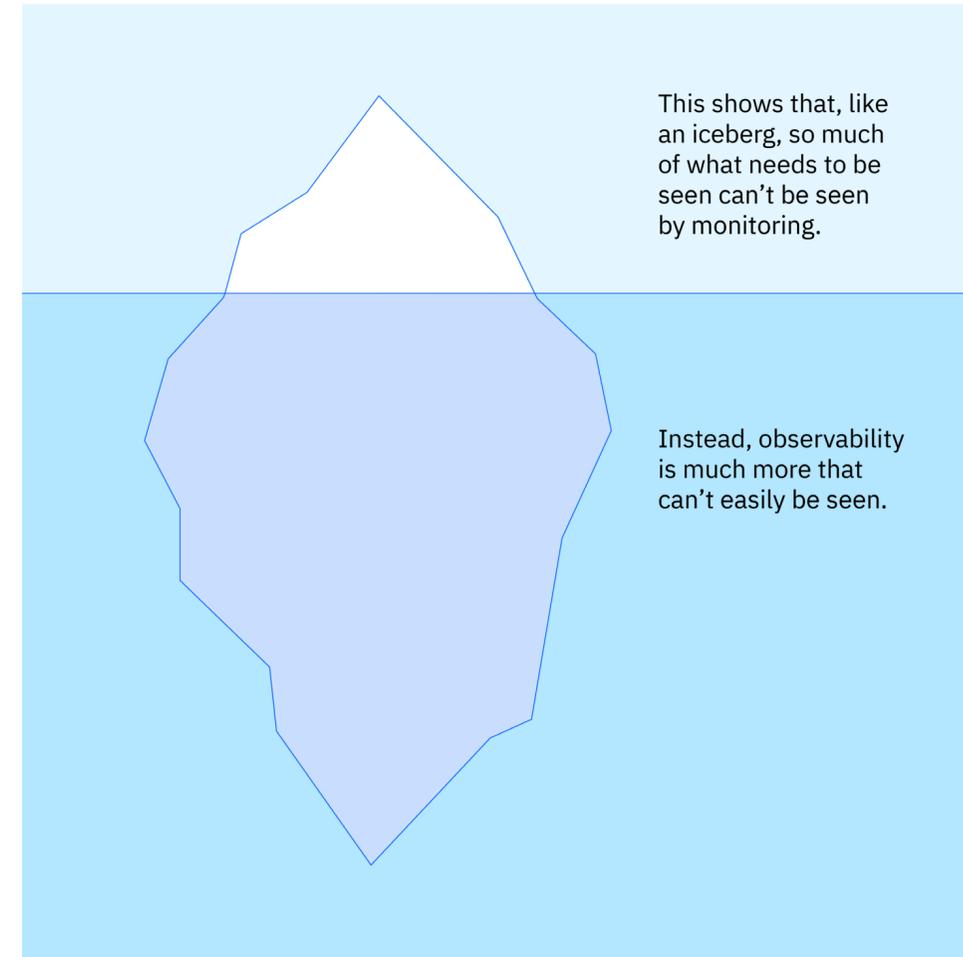
O que é observabilidade?

Como a observabilidade difere do monitoramento?

Não é possível ter uma imagem verdadeira de onde se está indo sem saber onde se esteve. O monitoramento de desempenho do aplicativo (APM) é um conjunto importante de recursos que coexistem com a observabilidade moderna.

O monitoramento tradicional concentra-se em medir aspectos predefinidos de componentes conhecidos. Isto era ótimo na década de 1980, mas as arquiteturas distribuídas de hoje contam com componentes de aplicativo que estão sempre mudando, às vezes a todo segundo. As ferramentas de APM tradicionais não conseguem acompanhar esse ambiente dinâmico e em rápida evolução.

Obviamente, o pior momento para descobrir que está faltando alguma métrica crucial é durante a triagem de uma falha de produção. É neste momento que a observabilidade se torna crítica. Em vez de predeterminar o que medir, as ferramentas de observabilidade veem tudo o que acontece entre os serviços e até mesmo dentro deles. Esta funcionalidade traz respostas a questões que você nem poderia prever, como quando você enfrenta situações desconhecidas.





O monitoramento pode ser útil. Às vezes. Ferramentas de monitoramento e alertas baseados em métricas são ótimos para responder a questões básicas como:

- Meu serviço está on-line e disponível para os clientes?
- Qual porcentagem de solicitações apresenta erros?
- Qual é a latência média de uma solicitação?
- Quanta memória o Redis está usando?
- Preciso realizar uma ação de escala linear?

No entanto, no decorrer da manutenção de integridade dos aplicativos, frequentemente precisamos conseguir responder a questões mais profundas como as seguintes:

- Os usuários estão enfrentando algum erro durante os caminhos críticos para os negócios?
- Por que a taxa de erro é bem maior para este subconjunto específico de clientes?
- Onde está o gargalo no sistema que causa latência em um endpoint específico?
- O que mais mudou no meu sistema que poderia estar causando esse erro?
- Qual serviço preciso atualizar para corrigir esse erro?

Como alcançar observabilidade total de ponta a ponta

Para responder a essas questões mais profundas é necessário desenvolver um sistema que monitore todo o ciclo de vida da solicitação do usuário, do cliente à camada de persistência. Para os desenvolvedores de aplicativos, isso significa instrumentar clientes e back-ends para emitir dados de telemetria úteis.

Quando essa telemetria estiver disponível, é necessário coletá-la, processá-la, armazená-la e analisá-la para transformar os dados em insights úteis e práticos. Normalmente, o agente ou coletor reúne os sinais de telemetria e os encaminha para um banco de dados para armazenamento. Os bancos de dados de séries temporais e colunares desempenham um papel muito importante no armazenamento eficiente e na consulta de dados de métricas.

Por fim, para análise e insights, é necessário ter uma interface de usuário para consultar o banco de dados. Isso permitirá que os dados sejam exibidos como gráficos e dashboards. É possível também configurar alertas para notificar rapidamente os desenvolvedores caso ocorra algum problema, incluindo a análise automatizada de causa raiz e alertas inteligentes.

Os sinais de telemetria

São os formatos de dados utilizados para coletar informações de serviços de aplicativos.

- **Rastreios:** cronogramas avançados com escopo de solicitação definido
- **Eventos:** logs estruturados para rastreamento de mudanças externas como implementações
- **Métricas:** números agregáveis sobre eventos ou sistemas

- **Perfis:** métricas do nível do tempo de execução
- **Logs:** registros de eventos com data e hora
- **Exceções:** logs estruturados para rastreamento de erros

É importante lembrar que nenhum sinal ou conjunto de sinais constitui observabilidade por si só. No entanto, o rastreamento distribuído é indiscutivelmente o sinal mais importante na observabilidade.

O que são rastreios distribuídos?

Quase todos os desenvolvedores conhecem os rastreios de stack. Eles aparecem todos os dias nos logs de erros. Mas o rastreio distribuído é o que você recebe quando as chamadas de função de um único serviço são substituídas pelas chamadas de rede de um aplicativo distribuído.

Os rastreios distribuídos são compostos de “intervalos”, um derivado dos períodos de tempo. Todo intervalo possui um horário de início, um horário de término, qualquer número de atributos customizados e uma referência ao seu intervalo pai. Sempre que um serviço processa uma solicitação, ele cria um intervalo que faz referência ao intervalo do serviço que está fazendo o chamado. Isso é recebido por meio dos cabeçalhos da solicitação seguindo a especificação W3C Trace Context.

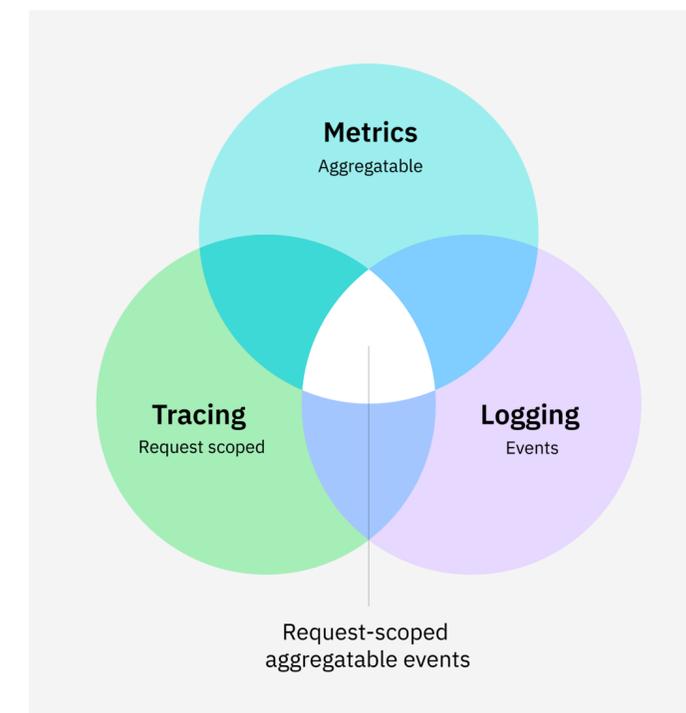
É possível criar também intervalos personalizados para adicionar mais granularidade ao rastreio em um serviço específico. Por exemplo, se houver uma função que faz muito processamento depois de receber resultados de um banco de dados, pode ser útil incluir essa função em um intervalo personalizado.

Mas os rastreios são mais do que simples linhas do tempo. Envolvem também muito mais do que apenas latência. Com o rastreio do caminho de solicitações individuais, é possível contextualizar logs, métricas e outros sinais de uma forma que ajude a responder questões de um ponto de vista centralizado no usuário (com escopo da solicitação). Os rastreios incluem uma estrutura de dados que ajuda você a:

- Entender os fluxos de solicitação em todo o sistema.
- Visualizar imediatamente a topologia do seu sistema.
- Extrair métricas da variedade dos metadados de rastreio.
- Enriquecer os logs com contexto anexando-os a um intervalo específico.

Reunindo tudo

Embora esses sinais possam ser úteis isoladamente, você pode extrair respostas mais valiosas quando todos esses dados puderem ser correlacionados e pesquisados de maneira relevante. Peter Bourgon, o bem conhecido engenheiro de software, expôs com elegância o ponto crucial para a informação onde métricas, rastreios e registros se sobrepõem.³ Os resultados são eventos agregáveis com escopo de solicitação.



Padrões de observabilidade e código aberto

Em 2019, os projetos OpenTracing e OpenCensus se fundiram e se tornaram o OpenTelemetry. Este projeto de código aberto da Computing Native Computing Foundation (CNCF) está estabelecendo rapidamente padrões abertos para telemetria que são amplamente adotados por ferramentas de código aberto e também por plataformas de observabilidade de fornecedores que incluem o IBM® Instana.

O projeto OpenTelemetry trabalha em três áreas para simplificar a coleta de dados de telemetria: especificações e padrões, ferramentas de instrumentação e ferramentas de pipeline.

As especificações incluem:

- OTLP — um formato binário para representar com eficiência os rastreios e logs de métricas.
- Especificação do W3C Trace Context para propagação de um traceId pai para serviços de recebimento de dados.
- APIs de linguagem que podem ser chamadas pelo seu código de biblioteca ou aplicativo.

As ferramentas de instrumentação incluem:

- SDKs de linguagem que conectam as APIs de linguagem a implementações específicas. Podem ser os chamados padrões ou plugins fornecidos por uma comunidade ou fornecedor e usados para processar e exportar telemetria.
- Bibliotecas de instrumentação automática que chamam automaticamente as APIs de linguagem quando ocorrem eventos relevantes em sua framework ou biblioteca.

E, por fim, para processamento e transmissão, existe o OpenTelemetry Collector. Essa ferramenta expansiva pode atuar como um agente em seu nó para coletar e encaminhar todos os dados de telemetria. Existe também um grande ecossistema de receptores, processadores e exportadores disponíveis como plugins.

Back-ends de observabilidade de código aberto

O projeto OpenTelemetry não oferece um back-end para armazenar e analisar seus dados de monitoramento, mas diversas ferramentas de código aberto são compatíveis com os sinais de OTLP.

O projeto de demo do OpenTelemetry oferece um exemplo de aplicativo de microsserviços que pode ser executado no docker ou no Kubernetes. Inclui três bancos de dados de telemetria de código aberto: Prometheus para métricas e logs e Jaeger para rastreios. Inclui também o Grafana para visualização de algumas das métricas do Prometheus.

Metodologia SLO



Até agora, você provavelmente já ouviu falar muito sobre objetivos de nível de serviço (SLOs) e indicadores de nível de serviço (SLIs).

A metodologia SLO representa uma nova maneira de pensar sobre o desempenho e o funcionamento do software que anda de mãos dadas com a observabilidade. A observabilidade como conceito vem da teoria do controle, que se concentra nas operações ótimas, não perfeitas, das máquinas. A metodologia SLO ensina que as equipes de TI que buscam a perfeição geralmente alcançam resultados piores do que quando estabelecem metas realistas.

Tudo começa com o indicador de nível de serviço. O SLI é qualquer métrica ou estatística que pode ser convertida em porcentagem. No contexto da execução de software, SLIs são, por exemplo, a porcentagem de solicitações que foram atendidas com sucesso ou a porcentagem de solicitações que tiveram latência aceitável.

Um SLO é um destino vinculado a um SLI. Os SLOs costumam ser expressos como um certo “número de noves.” Por exemplo, quatro noves indicariam que um SLI atinge a meta em 99,99% das vezes. Muito importante, os SLOs nunca devem ser “100%”. Este número não é realista em quase nenhuma situação. É melhor deixar uma margem de erro, alguma margem de manobra para interrupções planejadas e não planejadas.

Na verdade, uma equipe do Google descobriu que poderia aumentar a confiabilidade geral do sistema causando tempo de inatividade de seu serviço de forma artificial para evitar que outras equipes esperem que ele seja 100% confiável.³ Não é recomendado replicar isto na sua organização.



Os acordos de nível de serviço ou SLAs podem parecer semelhantes aos SLOs, mas são usados para finalidades muito diferentes. O SLA faz parte de um contrato de negócios e especifica o que acontece se a meta for violada, geralmente algum tipo de penalidade financeira.

Os SLAs não são interessantes para os desenvolvedores até que sejam violados. Os SLOs são extremamente úteis para os desenvolvedores porque ajudam a entender a confiabilidade geral dos aplicativos e a determinar se é seguro investir em novos recursos.



Além do código aberto



Com as ferramentas de código aberto do capítulo anterior, você pode reunir muitos dados de telemetria de serviços e começar a reuni-los de maneira útil. No entanto, é difícil definir que você realmente alcançou a observabilidade.

A *observabilidade* exige que seja possível responder a questões sobre situações desconhecidas. Sua telemetria deve ser capaz de se adaptar e mudar conforme seus serviços. Até agora, nenhuma ferramenta de código aberto pode oferecer esse nível de automação. Mas uma solução como o IBM® Instana consegue.

As soluções de observabilidade são adaptáveis e dinâmicas

Seus aplicativos provavelmente são compostos por dezenas, centenas ou até milhares de serviços que utilizam diferentes linguagens e tecnologias. Manter a instrumentação manual consistente em toda a área de superfície do aplicativo seria praticamente impossível.

Com o uso da automação, como a descoberta dinâmica de serviços e instrumentação automática, você pode ter um entendimento muito mais abrangente de seus sistemas antes que ocorra algum incidente. Isso ocorre porque, durante a triagem de uma indisponibilidade de produção, não é o momento adequado para descobrir que está faltando uma peça-chave do quebra-cabeça.



O benefício da correlação automática

Um grande benefício da solução de observabilidade corporativa é a capacidade de correlacionar máquinas, infraestrutura e métricas e rastreios de serviços e aplicativos. Os rastreios distribuídos proporcionam um entendimento do fluxo da solicitação, enquanto as métricas oferecem os pontos de desempenho necessários. Correlacionar tudo isso manualmente, no entanto, é bastante complicado. A principal razão que faz com que as pessoas não gostem de ter vários dashboards de serviços diferentes é que pode ser quase impossível combinar quaisquer dados e chegar ao contexto geral do problema.

O maior benefício da correlação automática é o insight imediato. Ao analisar um problema ou incidente, a solução faz todo o trabalho de investigação. Isso traz informações importantes como evidências contextuais e leva você diretamente à área de interesse.

O IBM Instana é adequado para você?



O IBM® Instana é uma [plataforma de observabilidade corporativa](#) que conta com recursos de [monitoramento de desempenho dos aplicativos](#). Ele foi criado visando empresas que operam aplicativos complexos e modernos nativos da nuvem, independente de onde residam: no local, na nuvem pública e privada, em dispositivos móveis ou em um ambiente IBM Z®.

O IBM® Instana ajuda você a controlar aplicativos híbridos modernos com a descoberta impulsionada por IA de dependências contextuais profundas dentro de aplicativos híbridos. O Instana oferece também visibilidade sobre os pipelines de desenvolvimento para viabilizar a automação com DevOps de circuito fechado.

Esses recursos oferecem o feedback prático necessário para os clientes, pois otimizam o desempenho dos aplicativos, viabilizam a inovação e mitigam os riscos. Esses recursos ajudam o DevOps a aumentar a eficiência e agregar valor aos pipelines de entrega do software para que eles possam atender aos seus objetivos de serviço e de negócios.

Veja você mesmo o poder do IBM® Instana. Inscreva-se hoje para fazer uma avaliação sem custo de 14 dias da versão integral do produto. Não é necessário informar o cartão de crédito.

[Avaliação sem custo do IBM Instana](#) →

[Conheça o IBM Instana](#) →



1. R. E. Kalman, “On the general theory of control systems,” Agosto de 1960.
2. Blog do Twitter, “Observability at Twitter,” 9 de setembro de 2013, acessado em julho de 2023.
3. Livro online do Google sobre SRE, “Objetivos de nível de serviço,” acessado em julho de 2023.

© Copyright IBM Corporation 2023

IBM Brasil Ltda
Rua Tutóia, 1157
CEP 04007-900
São Paulo, SP
IBM Corporation
New Orchard Road
Armonk, NY 10504

Produzido nos Estados Unidos da América em agosto de 2023

IBM, o logotipo IBM. IBM Z e Instana são marcas comerciais ou marcas registradas da International Business Machines Corporation nos Estados Unidos e/ou em outros países. Outros nomes de produtos e serviços podem ser marcas comerciais da IBM ou de outras empresas. Há uma lista atualizada de marcas registradas da IBM disponível em [ibm.com/br-pt/trademark](https://www.ibm.com/br-pt/trademark).

Este documento é atual na data de sua publicação inicial, podendo ser alterado pela IBM a qualquer momento. Nem todas as ofertas estão disponíveis em todos os países nos quais a IBM opera.

É responsabilidade do usuário avaliar e verificar a operação de qualquer outro produto ou programa com produtos e programas IBM. AS INFORMAÇÕES CONTIDAS NESTE DOCUMENTO SÃO FORNECIDAS NO ESTADO EM QUE SE ENCONTRAM, SEM QUALQUER GARANTIA, EXPRESSA OU IMPLÍCITA, SEM NENHUMA GARANTIA DE COMERCIALIZAÇÃO, ADEQUAÇÃO A DETERMINADO FIM E QUALQUER GARANTIA OU CONDIÇÃO DE NÃO INFRAÇÃO. Os produtos IBM têm garantia de acordo com os termos e condições dos contratos sob os quais são fornecidos.