

在云原生应用中 获得可观察性



目录

01 →

简介

02 →

基于云的技术与云原生
技术之对比

03 →

云原生环境中的可观察性

04 →

可观察性挑战

05 →

适用于云原生环境的可观察性所需具备的关键特征

06 →

不在云原生环境中采用
可观察性将面对的风险

07 →

可观察性为云原生应用
带来的优点

08 →

企业可观察性简介

09 →

关于 IBM Instana



简介

随着各类云原生技术的广泛采用,应用开发、交付和运行也因此而发生重大变化,出现新的竞争格局。行动更迅速的组织充分把握这一转变带来的契机,一举超越规模较大但行动迟缓的同行。

应用技术栈的变化令人眼花缭乱。各种云环境(无论是公有云还是私有云)不断扩展,虚拟服务器发展演变为容器,而 Kubernetes 则成为编排标准。容器的采用呈现爆炸式增长,无服务器环境中的无服务器工作负载日益普及,Dev 团队也变得更为敏捷。随着软件与基础架构之间的界线变得日益模糊,两个相互矛盾的观点随之出现:

- 应用性能监控 (APM) 成为企业成功的重要因素。
- 而传统 APM 方法很难提供管理现代应用所需的可视性。

组织既需要加速开发与交付流程,同时还要确保 IT 运维团队从 APM 工具获得期望的可视性。因此,可观察性这一概念应运而生,并以多种形式呈现在大家面前。不过,可观察性自身面临一定的挑战,包括传统监控工具同样面临的那些难题。快速灵活的应用交付组织(例如 Dev、Ops、DevOps 和 SRE 团队)希望使用整套云原生技术。为此,他们必须采用全面的可观察性方法,充分利用开发人员推动的可视性以及现代 APM 解决方案所带来的各种最佳实践。



基于云的技术与云原生技术之对比

就应用开发而言,云技术已成为真正的分水岭。云和多云环境提供了更强的功能和更高的冗余度,但它们也要求应用实现现代化。基于云的应用经过改造,适合在云端运行,而云原生应用则是专为云部署而开发。但我们指的是属于谁的云呢?随着越来越多的企业选择多云环境,复杂性也与日俱增。混合云由私有云和公有云混合组成,进一步加剧了这种复杂性。当我们在本地托管环境中添加一层,形成混合环境时,现代技术栈可能会给开发团队带来额

外的压力,导致他们难以确保及时交付应用。

事实证明,监控云原生应用的性能也并非易事,因为传统的 APM 和可观察性工具可能难以像洞悉其他应用栈那样,轻松揭示云原生要素(例如容器)的内部情况。相比之下,云原生可观察性平台可监控整个技术栈中的所有指标、事件跟踪和日志,从而提供更广泛的可视性。



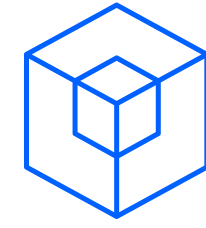
基于云的应用已经过改造,适合在云端运行。



云原生应用专为云环境而开发。

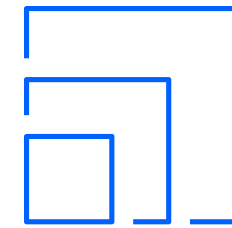
云原生环境中的可观察性

在构成现代云原生技术栈的技术面前,以下障碍使得传统监控工具力有不逮:



缺乏向后兼容性

容器经过专门构建,出于保持轻便的考虑,仅包含必需的服务,这意味着老旧的检测引擎可能无法访问完成工作所需的服务。



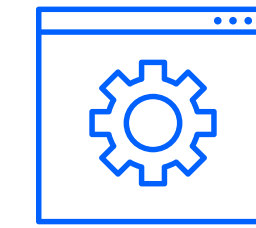
无服务器的消除服务理想

无服务器架构通过将服务器从本地控制迁移至云平台,将消除服务的理想提升到更高的水平,但这同样也会导致服务可用性问题的。



编排信息关联

Kubernetes 模糊了软件与基础架构之间的界线:现有工具可能需要使用多个代理程序才能提取基本数据,导致无法直接关联各种信息。



加快开发和部署速度

由于容器的短暂性特质,应用环境会不断发生变化,因此传统监控工具很难达到所需的可视性水平。

由于传统 APM 工具无法满足云原生环境的需求, 替代性的应用可观察性解决方案随之兴起, 而这也得益于以下其他因素:

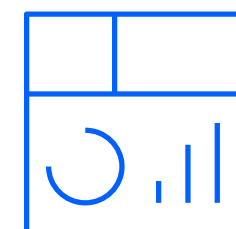
- 开发人员更大程度地参与到运行生产工作负载和确保应用性能的工作中。
- 云原生技术栈加快了开发和应用管道的速度, 这意味着尽管现有工具存在明显的可视性差距, 但更多组件仍可照常发挥作用。

可观察性从性能角度, 提供云原生要素 (例如容器、微服务、Kubernetes 之类编排工具以及无服务器环境) 中应用的运行状况和状态的外部指标。



可观察性挑战

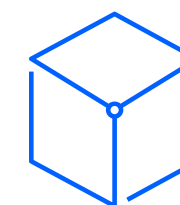
无论是“黑盒”（隐藏产品或程序的内部工作方式），还是大量的短期部署，云原生技术栈给可观察性和 APM 工具带来了巨大的挑战：



传统监控工具面对分布式微服务环境显得无能为力

基于容器的应用所处的环境往往持续动态变化；此外，在同样动态变化的服务器集群中部署大量的分布式微服务。一个完整的应用可能涉及多种不同的基础架构、平台、语言以及技术。

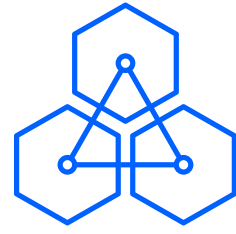
传统的基础架构和应用监控工具是针对映射到单个静态服务器的单体式应用而设计的，它们往往专注于提供语言层面的可视性。而如今，微服务分布在庞大的服务器集群之中，采用多种语言、中间件组件以及数据库系统，导致这些传统的监控方法无法发挥作用。



容器托管着各种各样的工作负载

容器可以托管各种类型的工作负载，既包括 Apache HTTP 服务器之类的单体式应用，也覆盖 Java 虚拟机和数据库，因此带来了复杂的监控挑战。混合环境中永久服务器上外包的持久数据存储使问题进一步复杂化。由于容器化环境中工作负载的多变性质，监控工具必须支持范围广泛的技术、架构和配置，这无疑是一项艰巨的任务。

由于每个容器都各不相同，因此传统的监控工具难以针对容器进行配置，也很难确定该提前为每个容器设定怎样的阈值。此外，监控工具必须实现自动化，才能轻松应对不可预测的工作负载类型，因为人工配置并实施监控策略完全不可行。



容器需要更高水平的监控

容器平台通常只包含基本的监控功能,而统计工具无力监控大规模的生产环境。这是容器与其他应用基础架构技术的主要区别。

同样,操作系统提供有关系统性能的海量数据,但不包括应用级别的数据。即便是像包含相对成熟监控工具的 VMware 这样的高级虚拟机平台,也无法提供应用层数据或分布式跟踪能力。



Kubernetes 并非性能监控工具

诸如 Kubernetes 之类的容器编排器非常适合组织、配置和管理各种生产容器环境和应用的部署。

但编排器并非用于监控目的,虽然它们高度关注容器和主机资源,但编排管理并不涉及用户体验或应用性能方面的问题。

此外,容器编排器仅管理它们了解的内容,也就是在容器中运行的基础架构和服务。根据定义,在混合环境中,容器编排器并不监控任何其他资源的使用情况。这样,环境的绝大部分都面临着性能问题的风险。



适用于云原生环境的可观察性所需具备的关键特征

可视性是云原生可观察性平台的核心要素。对于基于动态微服务的架构而言,跟踪和依赖关系映射都是非常艰巨的任务,因此从内部监控容器就成为一项至关重要的功能。

尽管容器的性质决定了其存在时间短暂,有时还不完整,但云原生可观察性仍可提供切实可行的洞察,帮助团队从被动的观察者转变为解决甚至预防问题的主动参与者。熟悉容器化微服务的代理可在 IT 架构中映射资源,即使这些资源几乎毫无关联而且不断变化,也不在话下。

其他重要特征包括:

所有数据

DevOps 团队必须能够使用完整的分析功能,以便确切了解问题是在何时、何地以及如何发生的。做到这一点的唯一途径是捕获每一项请求的分布式跟踪,不留任何缺口、不进行抽样也不使用任何代理。此外,部署到容器化环境中的所有基础架构、平台和代码行都必须支持跟踪功能。

云原生部署

云原生可观察性工具可无缝集成到云原生应用环境、完全自动化的部署和检测流程中。

根本原因分析

无论环境多么复杂,DevOps 团队必须能够尽快发现并解决性能问题。可观察性解决方案必须能够自动确定分布式应用故障的发生位置及原因。

不在云原生环境中采用可观察性会面临的风险

对于某些组织而言,为云原生应用部署可观察性可能并非优先任务,这也许是因为他们的 APM 解决方案目前还没有报告数量惊人的错误,或者是因为他们的团队还需要了解新平台。

但是,当传统监控解决方案导致错误漏报、造成事务延迟以及错失优化机会时,您可能并未意识到需要企业可观察性平台。延误意味着风险:

- 未检测到的错误可能导致面向客户的服务性能下降或发生中断。
- 从检测到事件到人员发现问题之间的时间间隔有多长,决定了组织是防患于未然,还是被迫进行重大事件管理。
- 监控工具可能会找到问题,但无法提供足够的背景信息以帮助您做出反应。

所有这些场景都有可能导致企业声誉受损、宕机而造成收入损失,并且影响客户体验。



可观察性为云原生应用带来的优点

当然,更为全面的监控解决方案的主要优点在于可以发现问题并预防事件发生。借助可观察性,透明而且可观察的系统可将工程师和 SRE 团队从以前那种耗时耗力的持续监控工作中解放出来,使他们能够全身心地投入具有更高附加值的任务中,例如为客户开发新的产品功能。

持续集成/持续部署 (CI/CD) 流程的执行速度不断提高,无用功明显减少,错误率大幅下降,客户满意度显著提升,从而为企业节省大量时间和成本。但 IBM Instana™ 的优点远不止于此:

缩短平均修复时间 (MTTR)

事件关联能力的高低,决定了组织是花费数小时进行事件分类,还是轻松化解潜在问题,防患于未然。

左移效应

工程师与开发人员越早发现问题,就越能主动地采取措施,而不是被动应对,这样每个问题造成的影响就越小。

减少 CI/CD 影响

可观察性可帮助组织更轻松地跟踪进行中的金丝雀式发布或蓝/绿部署成功与否。如果有不成功的节点发布至生产环境,势必会造成损失。

系统运行状况更理想

云原生环境中可观察性工具的自然输出与 DevOps 相结合,有助于减少系统错误,提高流程效率。

客户满意度更高

作为客户,如果所使用的应用或服务频频出错、响应缓慢、出现宕机甚至对业务造成威胁,您还乐意继续买单吗?

企业可观察性简介



尽管可观察性可提供跟踪应用性能所需的信号,但企业可观察性才是真正全面了解整个系统的更佳选择。

借助企业可观察性,您不仅可以监控各个单独的系统。还可将有关这些系统的数据与背景信息相结合,将整个企业 IT 环境中应用与系统之间的各种互动联系起来。企业可观察性的基础包括自动化、背景信息、智能措施、修复以及使用简便性。

更具体地说,企业可观察性以下列几项关键实践和原则为基础:

- 系统性优化

企业可观察性的重点不是管理各个单独的应用或系统的运行状况和性能,而是优化整个 IT 环境。这意味着对架构中的所有资源进行映射并提供背景信息,即便这些资源处于持续变化或松散耦合的状态,也是如此。

- 完整的背景信息

每个可观察性数据单元在交付时必须具有完整的背景信息。团队不能依靠抽样和猜测做出反应;他们必须进行端到端的跟踪,获得完整的背景信息。

– 云原生部署

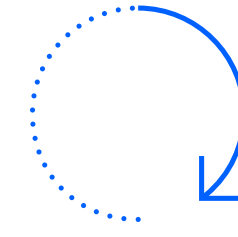
企业可观察性工具必须能够无缝集成到它所支持的云原生应用环境中。部署和检测流程必须完全实现自动化。

– 为数据采集提供全面支持

现代企业应用环境以各种方式公开数据：以标准输出或传统日志形式，亦或是通过开源监控 API。企业可观测性工具必须支持所有数据格式，从每个数据源采集数据和背景信息。

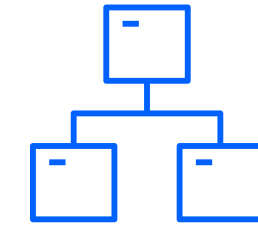
– 贯穿整个管道的可观察性

团队必须从 CI/CD 管道的起点就开始监控应用行为并获取背景信息，直至部署完成。不应等到新应用发布到生产环境中之后，才能够了解该应用与其他系统的互动方式，以及该如何优化其行为。



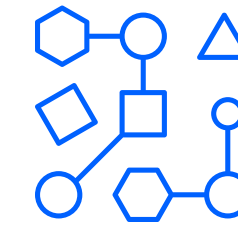
自动化

当部署了新代码或系统发生变更时，企业可观察性会自动、持续地发现变化，并提供即时反馈。



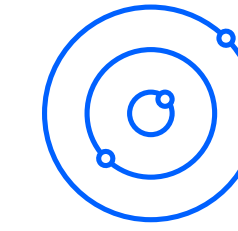
背景信息

企业可观察性会解释每个应用组件和服务与其他组件和服务之间如何相互关联，从而优化资源的性能和可用性。



智能措施

发生变化时，企业可观察性会主动提供深入分析以及背景信息，并建议系统优化步骤。



易于使用

随着组织不断推广敏捷开发流程，应用更新频率不断提高，CI/CD 管道日益繁忙，更多的利益相关方需要掌握切实可行的信息。

关于 IBM Instana



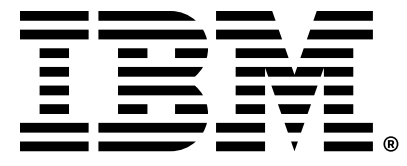
IBM Instana 为运行各种复杂的现代云原生应用的企业提供[企业可观察性平台](#)，该平台具备[自动化的应用性能监控](#)功能，无论这些应用是部署在本地还是公有云或私有云（包括移动设备或 IBM zSystems™），该平台均适用。

使用 IBM Instana 基于 AI 的发现功能，洞悉混合应用内部深层背景中的依赖关系，组织就能轻松掌控现代混合应用。IBM Instana 还提供针对开发管道的可视性，以帮助实现闭环的 DevOps 自动化。

这些功能提供客户所需的切实可行的反馈，帮助优化应用性能，支持创新并降低风险，从而推动 DevOps 团队提高效率，为软件交付管道增添价值，同时满足服务和业务级别的目标。

[IBM Instana 一探究竟](#) →

[IBM Instana 免费试用](#) →



© Copyright IBM Corporation 2023

国际商业机器(中国)有限公司
了解更多信息, 欢迎访问我们的中文官网:
<https://www.ibm.com/cn-zh>

美国出品
2023 年 5 月

IBM、IBM 徽标、IBM Instana 和 zSystems 是 International Business Machines Corporation 在美国和/或其他国家/地区的商标或注册商标。其他产品和服务名称可能是 IBM 或其他公司的商标。Web 地址 ibm.com/trademark 上提供了 IBM 商标的最新列表。

本文档为自最初公布日期起的最新版本, IBM 可随时对其进行修改。IBM 并不一定在开展业务的所有国家或地区提供所有产品或服务。

用户负责评估并验证与 IBM 产品和程序配合使用的任何其他产品或程序的运行。本文档内的信息"按现状"提供, 不附有任何种类的(无论是明示的还是默示的)保证, 包括不附有关于适销性、适用于某种特定用途的任何保证以及非侵权的任何保证或条件。IBM 产品根据其提供时所依据的协议条款和条件获得保证。