通过部署自动化方案，实现应用程序驱动的容器弹性

适用于希望在确保应用程序性能的同时加快上市速度的平台和 DevOps 工程师

目录

03
执行摘要

03
对速度、敏捷性、弹性和规模的承诺

05
平台和基础架构

07
应用程序驱动方法

08
疫情期间加速数字化转型

执行摘要

您的竞争优势取决于创意转化为业务交易成果的速度，以及创意如何为客户带来显著效益。技术就是推动力。

容器提供了速度、敏捷性、弹性和规模，从根本上改变了我们构建、部署和运行这些应用程序的方式。容器开创了应用程序可以真正随处运行的世界；更新和新功能可以每天多次部署到生产中，动态波动的工作负载需求可以随时随地通过弹性基础架构供应进行管理。Kubernetes 平台能帮助企业变得敏捷和富有弹性，但并不能在保证高效的同时确保性能。

尽管容器化提供了所有的简便性和敏捷性，但统筹编排的平台只提供了一种管理这些服务生命周期的方式，就是按照您描述的方式部署和维护您的服务。

容器平台本身无法确保服务满足 SLO 要求，也无法动态管理资源
基于阈值的策略无法解决持续性能问题；这种方法从未奏效，而且就容器平台的变化速度而言，不相关的触发式自动扩展最终可能会导致问题。具有弹性的基础架构是实现性能的关键，但需要自动分析来持续管理需求、供应和限制，以实现预期服务水平目标 (SLO) 要求。

本白皮书讨论了采用容器平台作为业务运行方式时需要考虑的关键概念，以及如何通过自动化来保护投资，以确保性能，同时最大限度地降低成本并保持合规。它概述了您需要自上而下的驱动型分析，以自我管理的 Kubernetes 平台来运行您的服务的原因。在早期阶段构建多云规模，为您的 IT 部门提供运营“肌肉记忆”，从根本上改变您实现更多创新的方式和时间。

部署频率	每周–每月	每小时–每天
变更准备时间	超过六个月	少于一小时
变更失败率	16%–30%	0%–15%

对速度、敏捷性、弹性和规模的承诺

Kubernetes 可助您实现弹性，但不会自动确保您满足并保证应用程序 SLO 要求。

采用容器化的成功与否取决于您如何为开发人员提供他们所需的敏捷性、规模化适应波动频繁的需求所需的弹性，以及保证应用程序以所需的速度运行。

采用云原生方法并将应用程序分解为不同的服务集，可以推动提高应用程序开发和部署敏捷性。容器提供打包功能，使您的服务具有可移植性和可扩展性。Kubernetes 提供了运行数字应用程序和服务的框架和控制点。但是，要为您的业务提供高性能的企业级平台，您仍然需要添加功能，以释放平台所带来的弹性，从而满足并确保应用程序 SLO 要求。

利用 CICD 和生产反馈加快部署速度
基于自动化的正确持续集成和持续部署 (CICD) 方法是加快上市速度的关键。在 Google Cloud《2021 年 DevOps 现状报告》中，[1] 受访者表示实施 CICD 后取得了显著改善：

随着速度加快，需要有一种方法来管理生产中的不断变化，并获得有关服务执行情况的反馈循环，以及如何预测基础架构的需求。我们的目标是提供一种方法来定义 SLO 要求，
并让平台就如何配置容器和基础架构提供反馈，以降低出现性能问题的风险。

- 谁来决定如何为服务分配资源?他们如何决定?根据设定的 SLO 等指标进行压力测试和基准测试。
- 如何衡量性能?CICD 流水线中是否有反馈循环，以确保容器和各类 pod 配置正确?
- 如何确保新部署始终具备足够的容量?

查找 IBM Turbonomic 答案

选项	限制	IBM Turbonomic 答案
手动分析 容器和 pod 使用数据 从而确定 各类资源规范。	– 数据收集设置 – 分析所需的人工	– 自上而下的 应用程序驱动型分析， 决定如何调整容器大小 – 反馈到 CICD – 在不需要时 减少请求的机会
手动分析堆栈中 所有点的资源数据， 从而确定生产能力。	– 收集多来源数据 所需的人工 – 分析所需的人工	根据使用率进行分析， 从而确定整个全栈的资源需求

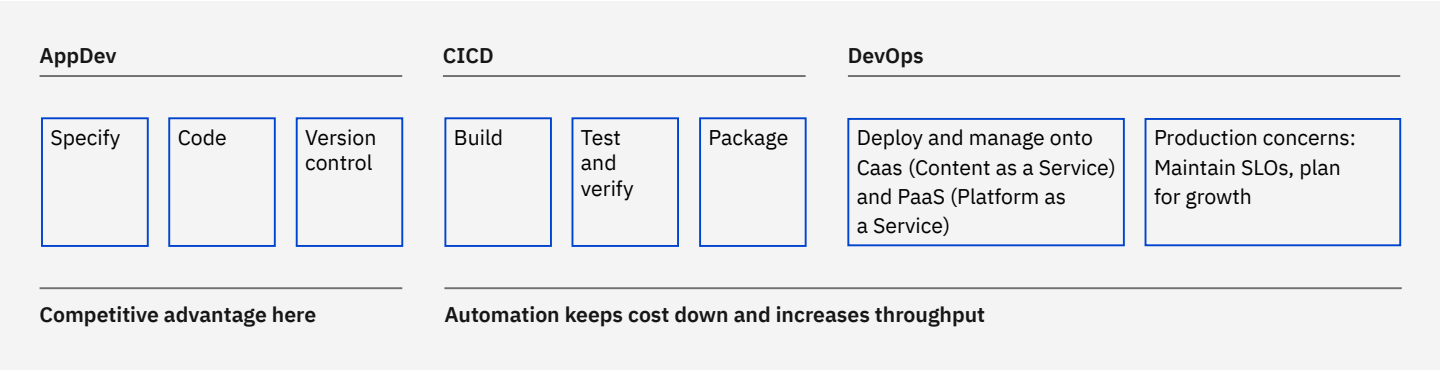


图 1：实现应用程序敏捷性的流程

平台和基础架构

为什么需要应用程序驱动型全栈管理
无论您选择何种容器平台或底层基础架构（私有云、公有云、混合云、多云甚至裸机），平台即服务 (PaaS) 的运营挑战都是一样的：

- 如何确定是否有足够的容量来满足当前和扩展需求？
- 如何决定何时启动更多应用程序节点？
- 如何决定何时暂停？
- 如何处理高峰需求？
- 如何利用公共云资源管理云爆发？
- 如何确保整个堆栈的高可用性 (HA) 和弹性？
- 如何执行业务约束？

容器平台所带来的弹性为您提供了机会，可以根据应用程序的平均需求总和而非峰值需求总和进行配置。要利用这种能力，提供可随需求波动而持续向上和向下扩展的平台，需要软件持续做出资源配置决策，从而确保应用程序在需要时获得所需的计算、存储和网络。

查找 IBM Turbonomic 答案

选项	限制	IBM Turbonomic 答案
在提供自动扩展组（如附属服务组 (ASG)、可用性集等）的服务提供商上运行。	– 基于阈值的策略 – 无法扩展特定节点：所有节点的约束、节点标签等必须相同	– 自上而下的应用程序驱动型 SLO 要求 – 持续调整基础架构资源以满足应用程序需求 – 持续向上、向下、纵向和横向扩展合适的容器、pod 和节点 – 持续将 pod 部署到合适的节点
分析堆栈中所有点的资源数据，从而确定生产能力。	– 收集多来源数据所需的人工 – 分析所需的人工	– 根据使用率进行分析，从而确定整个全栈的资源需求 – 持续向上、向下、纵向和横向扩展合适的容器、pod 和节点 – 持续触发操作，防止出现瓶颈问题

规模化运行 SLO 要求

构建容器平台宗旨是以完成业务所必需的服务水平运行各类应用程序。随着应用程序数量增加，您需要始终确保性能。通常情况下，我们看到客户需要花费 12 个月以上的时间来运行最初的 1-3 个应用程序。对于后续应用程序，由于需要进一步熟悉这些技能和最佳实践所产生的益处，可能还需要 6-12 个月的时间。当业务线了解到可能的情况后，需要管理的单个服务的数量规模已经超出了人力管理的范围。即使您利用容器的短暂性构建了无状态服务，您对最终用户体验性能下降的容忍度有多高?如何做到不仅可顺利管理需求，而且还能成功管控持续增长的变更率?答案就在于自动化，通过分析确保 SLO 要求所需的服务实例数量、配置工作负载的大小和位置，以及基础架构提供符合要求的资源等方面的平衡关系，执行相应的操作。

阈值不能解决问题

容器平台将确保您拥有最低数量的可用服务；如果其中一个崩溃，容器平台将尝试重新启动。但如果要确保良好的用户体验，就需要系统在性能下降和崩溃发生之前做出响应。您可以设置本机水平自动扩展来满足需求，但需要决定哪些指标最能体现所需的资源，配置阈值和上下限，测试并推断其是否能在生产需求下正常运行，然后对部署的每个服务重复上述过程。试想一下，如果一个应用程序有 100 多个服务项目呢?这些策略互不关联。如何确保增加某个服务的 pod，却不会造成其他区域的拥塞?您是否在克隆一个配置不当的 pod，此 pod 是否需要先完成垂直扩展?如何管理节点拥塞、考虑业务相互干扰的情况，以及识别可以释放出来满足这一需求的未使用的已分配资源?

此外，配置容器、pod 和 pod 水平自动扩缩 (HPA) 或集群自动扩展策略并不是一劳永逸的做法。必须对最优的猜测方案持续监控，并重新定义最优的猜测方案。如果您的团队不必手动设置和重置这些阈值，他们可以利用节省下来的时间做些什么?

正确配置的重要性直接影响到数字化转型战略的成功实施。一些错误的部署会大大减缓您正在构建的平台和系统的采用速度。手动配置这些控制点所耗费的时间和人工过多，会严重影响企业实现平台优先的能力。您的业务能承受这种延误吗?您需要的是一种控制系统，可以权衡管理所有资源，定义容器垂直扩展限制和请求、所需 pod 的数量，以及使用单一分析引擎重新分配 pod 和管理集群资源的部署决策。

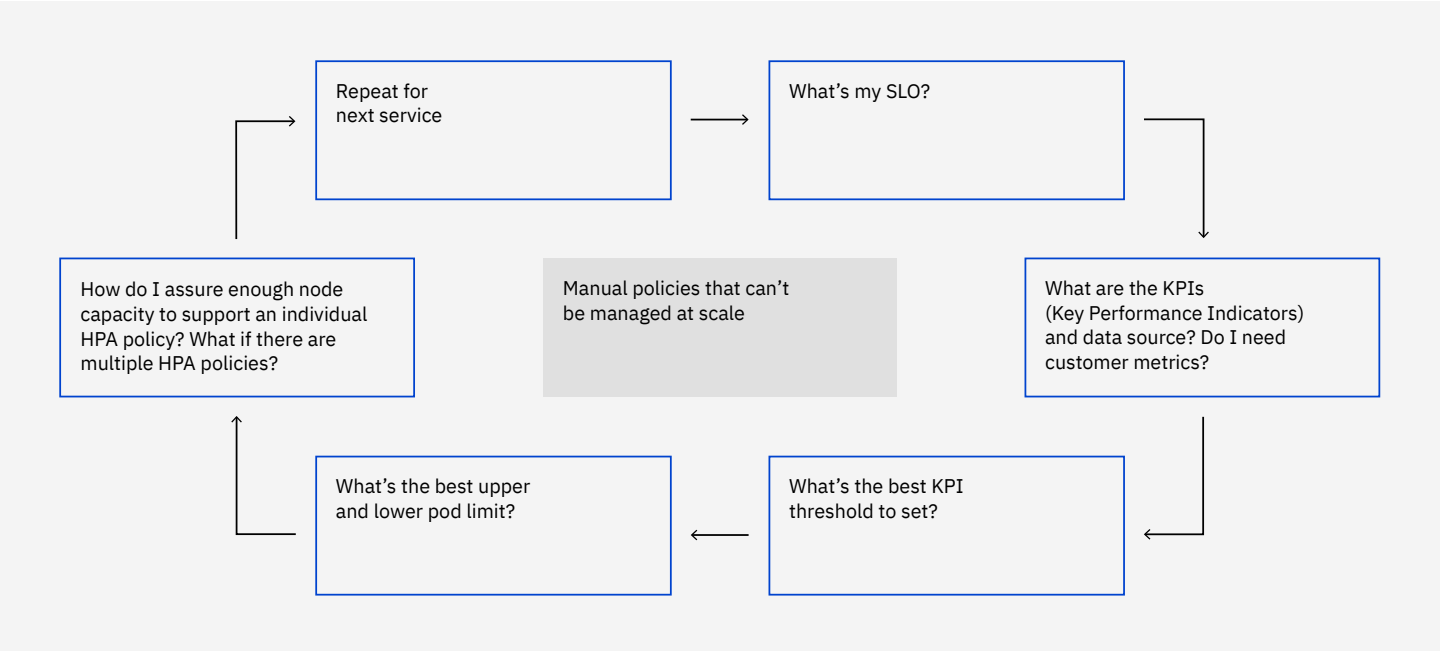


图 2：无法规模化管理的手动策略

查找 IBM Turbonomic 答案

选项	限制	IBM Turbonomic 答案
基于 HPA 阈值的策略，何时触发扩展 pod 的进出设定	– 按服务配置 – 基于服务所有 pod 的平均值 – 手动定义 KPI 和阈值以及 pod 上限和下限	– 自上而下的应用程序驱动型 SLO 要求 – 使用响应时间数据驱动服务横向扩展，以满足 SLO 要求 – 持续向上、向下、纵向和横向扩展合适的容器、pod 和节点 – 持续将 pod 部署到合适的节点 – 持续调整基础架构资源以满足应用程序需求
pod 垂直自动扩缩 (VPA) 基于阈值的策略，用于垂直扩展容器	– 必须为每项服务定义 – Beta 项目：请自行承担使用风险 – 无法访问节点容量以执行操作	
让 pod 崩溃，以便将其重新部署到更优的节点上	在即将崩溃的 pod 上处理事务时，用户体验不佳	
Prometheus 可观察性解决方案可收集和合并数据	– 不提供数据分析 – 不提供操作	

应用程序驱动方法

应用程序 SLO 要求应驱动基础架构
任务关键型应用程序的容器化是优势颇多的投资。但是要充分获得这些速度、弹性和可移植性方面的优势，就需要软件全天候在正确的时间做出正确的资源配置决策。否则，复杂性将拖慢您的速度。

IBM® Turbonomic® 应用程序资源管理将您的任务关键型应用程序与 Kubernetes 平台和底层基础架构相连接，无论您的应用程序运行在哪里。基于实时应用程序需求，并考虑到堆栈各层（从逻辑层到物理层）的约束和相互依赖关系，软件可在正确的时间确定正确的操作，从而帮助确保应用程序始终准确地获得所需的执行内容。实时执行、计划执行或作为 DevOps 流水线的一部分执行。

智能调整大小：如何确定容器的大小？
- 自动部署：作为流水线的一部分执行和持续调整大小，例如 YAML、Jenkins 等。
- 实时自动化：
  通过 Kubernetes 动态执行。

持续部署：何时需要移动 pod？
移动到哪个节点？
- 通过 Kubernetes 实时动态执行。
  仅适用于无中断无状态服务。

动态扩展：何时需要扩展或缩减集群？
扩展或缩减多少？
- 通过基础架构即代码或 Kubernetes 群集 API 实时动态执行集群扩展。

SLO 驱动型扩展：何时需要扩展或缩减 pod 以满足应用程序响应时间 SLO 要求？
扩展或缩减多少？

SLO 驱动型扩展的前提条件：
- 应用程序专为水平无状态微服务而设计。
- 这些应用程序拥有 Kubernetes 不提供的 SLO 数据定义和来源。

这种智能自动化对您、您的团队和您的业务意味着什么？
以下是 IBM Turbonomic 提供的独特优势，无论您是在本地、云端、裸机服务器或任何组合上运行 Kubernetes。

应用程序“巡航控制”：您的团队设定了响应时间 SLO 要求；人工智能软件有助于确保平台和底层基础架构始终提供满足这些 SLO 要求所需的资源，无论应用程序在哪里运行。

最大限度减少手动工作：开发人员、DevOps 和站点可靠性工程师 (SRE) 无需设置阈值、约束或自动扩展策略。软件会为您做出正确的资源决策，提供实际上可以自动执行的操作。

避免容量超支：无需依赖开发人员做出资源配置决策。为了安全起见，他们经常会超额配置资源，对吗？我们的软件可以准确地确定需要哪些资源服务，一切以应用程序需求为基础。

自信地加速 DevOps：安全地提高部署频率和规模。我们的分析与您的 DevOps 工作流程相集成，有助于确保新部署的服务和现有服务始终正常运行。

更加轻松地为增长做规划：使用我们的软件模拟新服务上线。准确确定支持新增长所需的节点数量。

客户亮点

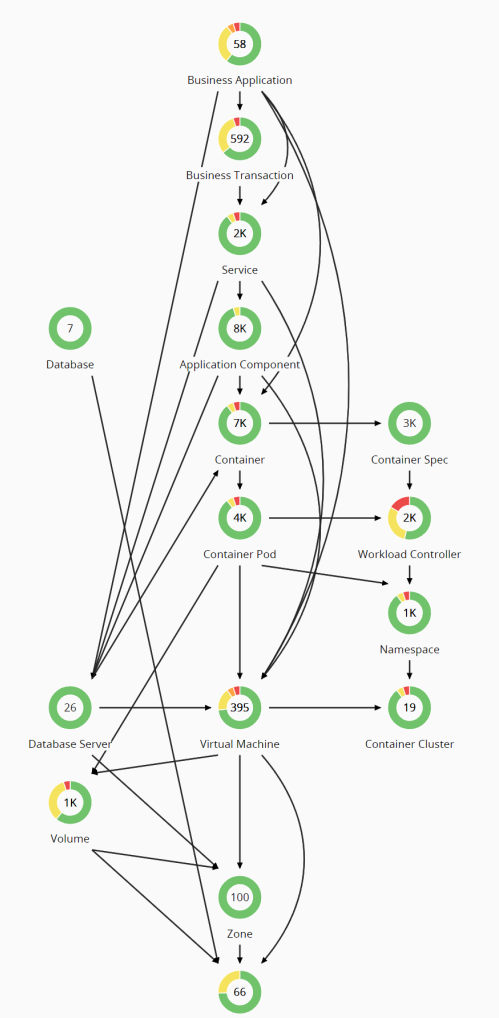
疫情期间
加速数字化转型

IBM Turbonomic 在 Kubernetes 平台和底层基础架构中的动态资源配置保持了较快的响应时间。

该客户是南美最大的保险公司之一，拥有 600 多万客户。其管理现有和下一代环境资源配置的行业标准方法正在减缓数字化转型和公司应对疫情的速度。

IBM Turbonomic 自动化系统在节假日需求激增期间为客户提供了较快的响应时间

该客户的业务应用程序与当地最大的廉价航空公司之一相集成。旅行保险通过该应用预订，因此我们在图 3 中看到的高峰与复活节小长假有关。虽然针对应用程序需求增加了，但 IBM Turbonomic 在 Kubernetes 平台和底层基础架构中的动态资源配置能够为客户保障较短的响应时间。



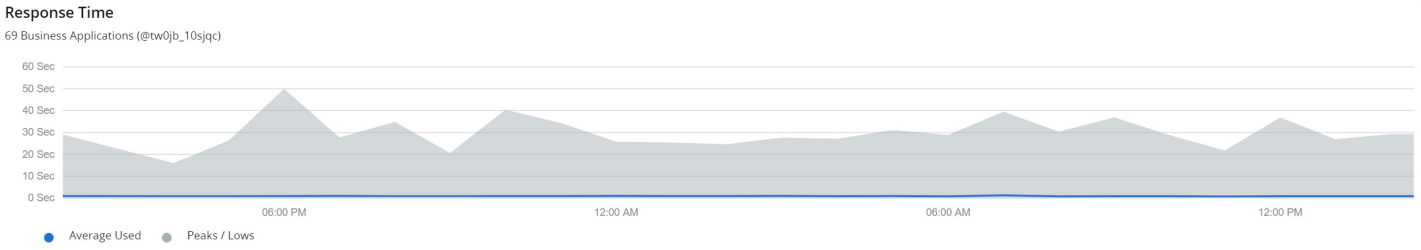


图 3：单个业务应用程序及其响应时间的全栈视图，即使在需求高峰期，自动化响应时间只需维持在较低的水平

57 个任务关键型应用程序
- 例如，车载 GPS：报告车辆失窃、新保单报价等
- 约 3,000 个 pod（约由 7,000 个容器组成）
- 链接到 Dynatrace

自动化
- 调整容器大小（暂存）
- 连续部署（全部）

凭单减少约 70%

关于 IBM 旗下公司 IBM Turbonomic

IBM® Turbonomic® Application Resource Management 提供客户使用的应用程序资源管理 (ARM) 软件，通过在混合和多云环境中动态配置应用程序，帮助确保应用程序性能和治理。IBM Turbonomic 网络性能管理 (NPM) 提供现代化的监控和分析解决方案，帮助企业、运营商和托管服务提供商在多厂商网络中确保持续的网络性能。

如需了解更多有关 IBM Turbonomic 智能自动化的信息，请访问 ibm.com/cn-zh/cloud/turbonomic 或与 IBM 代表联系。

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
IBM Corporation
New Orchard Road
Armonk, NY 10504

美国出品
2022 年 3 月

[1]《2021 年 DevOps 现状报告》，Google Cloud，2021 年