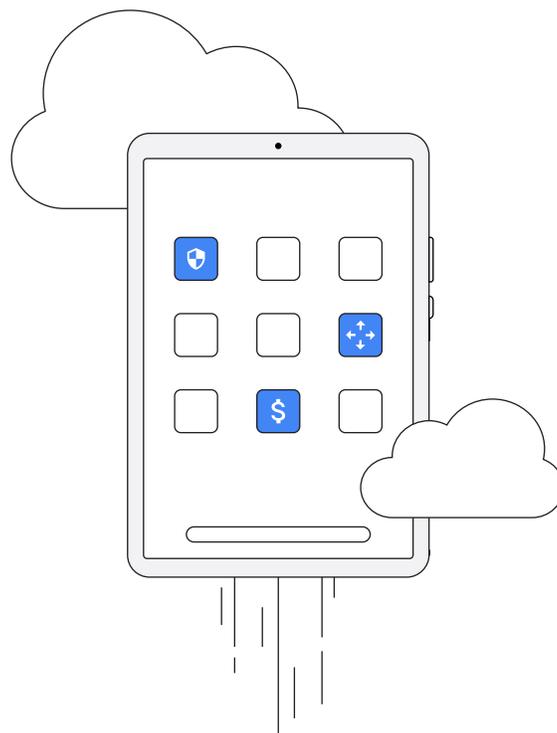


基础架构的未来是 容器化

技术公司和初创公司正在快速采用代管式容器平台,以减少在维护基础架构方面花费的工程资源,并将更多资源投入可产生业务成果(例如增长、竞争优势和盈利能力)的路线图优先任务。



如何简化基础架构管理以加快上市速度

第 02 页

为何考虑基于开放标准的代管式服务

第 05 页

如何让您的迁移支出发挥最大作用

第 9 页

第 1 章

如何简化基础架构管理以加快上市速度

虽然当今大多数技术公司和初创公司都在云上运行,但许多公司没有发挥云的全部优势。如果您在云上运行但未使用 [Kubernetes](#),那么您很可能采用的是专有解决方案,同时构建和维护自己的定制化辅助工具。由于运行工作负载的[虚拟机\(VM\)](#)利用率过低,您的效率会大受影响。此外,您还可能会锁定于特定平台。

您目前使用多少个工具来管理和修补虚拟机?您如何升级应用?虚拟机利用率是多少?您目前的部署可能不够高效。虚拟机架构中的弱点导致故障(服务中断、可伸缩性问题等),或者费用暴涨、失去控制,或者基础架构无法支撑您的许多业务需求,包括:

- 将 MVP 重构/重新设计为可伸缩的解决方案
- 扩展到其他云服务商以满足监管要求或客户期望
- 扩展到更多地理位置以缩短延迟时间,为分布在各处的广大客户群提供更好的体验
- 改善端到端安全状况
- 改善客户体验(例如服务可用性)

旧版架构和技术债务可能会拖累发展速度。这就是大量企业转向 Kubernetes 的原因。由托管式容器组成的现代化架构可为您提供运行安全可靠的基础架构的成熟模式。您可以加快上市速度并提高生产效率,同时兼顾稳定性和安全性,还能吸引优秀的技术人才投入创新工作。

您还应该注意自己是否会因为受到平台束缚而无法加入 Kubernetes 社区及相关生态系统。该社区及生态系统有着稳定的创新步伐,定义了如今的业界标准和最佳实践。最后,您必须决定希望工程师投入主要精力的工作:是维护基础架构以及构建和维护定制化工具,还是完成您的高优先级任务以促进业务发展。考虑到当下的招聘挑战,这一点甚至更为重要。您当前的部署可能运转得不错,但路线图中可能包含您不希望看到的工作,例如偿还技术债务和弥补平台在以下方面的不足:

- 端到端加密
- 可观测性(日志、指标、自动日志记录)
- 政策管理和执行
- 高可用性和自动故障切换
- 成本削减

Kubernetes 是一种不局限于特定平台的开源技术,提供各种开箱即用的常用工具,以保护并加速构建与部署生命周期中的每个阶段。它整合了大多数系统管理员长期积累得到的所有 bash 脚本和最佳实践,以一个系统的形式提供,可通过一组声明式 API 访问。Kubernetes 会自动执行所有操作,不显示细节,且随时可供使用。在将平台转换为基础架构即数据的同时,Kubernetes 可以消除大多数的基础架构即代码。您无需编写或维护代码,只需告诉 Kubernetes 您想要什么(而不是要做什么)。这可以为您减少庞大的管理开销,节省大量时间。

容器是运用互联计算的最佳方式。



如果要运行传统工作负载,您可以将应用与虚拟机分隔开来,并将其放在容器中,然后借助 Kubernetes 在现代平台上运行。使用容器映像来打包软件可以使虚拟机升级更加轻松。您现在可以使虚拟机的生命周期管理与应用的生命周期管理分离开来,通过移除 Ruby、Python 和 Java 等所有内容来简化虚拟机。而且,通过将其移到应该在的位置(靠近开发者的应用),您可以在一个地方进行控制并让您的机器保持裸金属状态。



托管式计算平台可将云服务转换为平台即服务,为您提供容器的强大功能和灵活性,以及无服务器的便利性。没有服务器、无需配置集群、无需进行维护,这意味着在保持控制的同时,您可以获得巨大的优势。



对于不怎么需要控制集群配置的工作负载,您可以让服务预配和管理集群的底层基础架构(包括节点和节点池),而您只需为工作负载(而非集群)付费。通过这种方式,您可以免除集群管理,同时优化安全性并节省大量资金。



对于更具云原生性的应用,无服务器可以事半功倍地实现相同的效果,免除底层基础架构并充当应用、数据甚至分析的端到端主机。借助无服务器平台,您可以开始在全托管式环境中以极低的复杂性运行容器,同时获得安全性、性能、可伸缩性和内置的最佳实践。

我们来看看对您业务的潜在影响,以及如何让您的组织在效率方面领先一步。

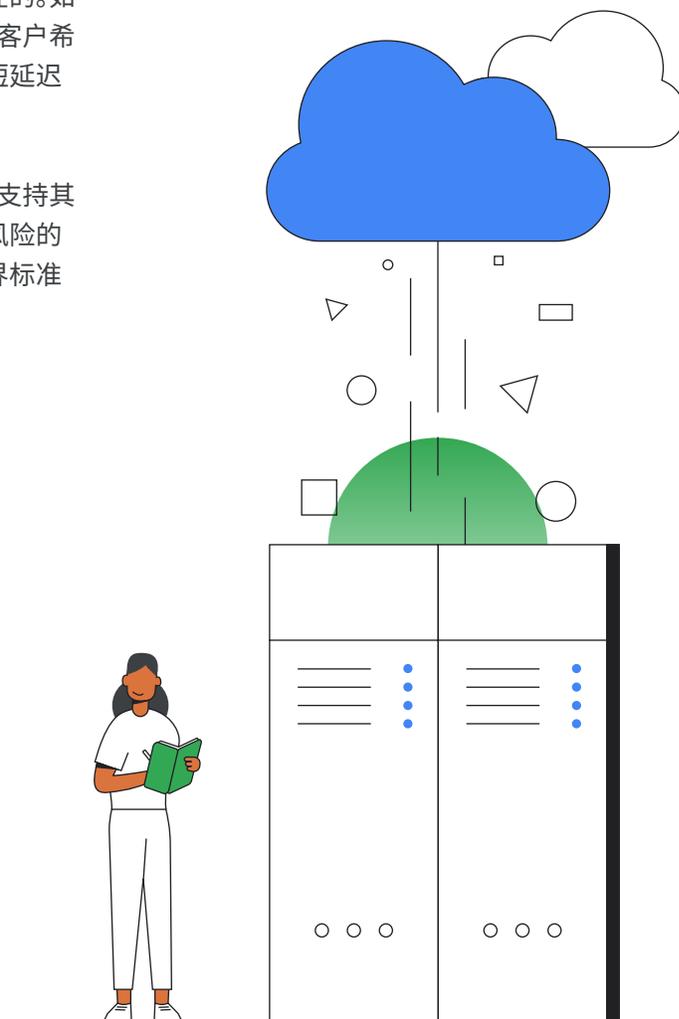
第 2 章

为何考虑基于开放标准的代管式服务

如果您有多云需求, 就应该运用可简化相关操作的技术

一些企业发现有必要在多个云中运营。数据引力是真实存在的。如果您的客户群体遍布全球, 那么您一定会发现: 您所服务的客户希望在实际存放数据的位置附近运行计算, 从而尽可能地缩短延迟时间并降低网络费用。

在这种情况下, 多云可以拓展您的潜在市场; 您最终将需要支持其他服务商的代管式服务和数据。其他公司将多云视为缓解风险的良策。无论是哪种情况, 关键在于正确采用多云技术, 而业界标准解决方案可以助您一臂之力。



01

首先：**确保您可以重复利用尽可能多的工作流程来完成各项工作。** workflow是指通过任务自动化在数据库和计算之间创建数据流架构。在这一点上，开源很重要；如果您选择的数据库不是开源的，则很难实现数据流架构和自动化。无论采用哪个云服务商的服务，都最好使用 Postgres 协议（例如 [Cloud Spanner](#)）或代管式 Postgres 数据库（例如 [Cloud SQL](#)）。

02

第二：在计算方面，Kubernetes 可节省大量在部署和自动化上花费的时间，因此在挑选一组技术时，**请确保它们能够在您的服务商边界之外正常工作。** 边界可以是不同的区域或可用区、不同的项目或帐号，甚至是本地或云端。不要浪费时间为每个云服务商单独构建和维护基础架构（例如，一个在 AWS 上、一个在 Google 上、一个在 Azure 上）；为了使它们保持一致，您的工程师会精疲力尽。如果您构建一次并部署在多个云上，则可以在需要时以集中且一致的方式进行更新。对于那些真正想高效地利用多云，并且不希望每次使用新的云服务商时都从事重复工作的客户，Kubernetes 这样的计算栈提供了巨大的优势。

03

第三：**风险管理。**能够在其他环境中运行技术栈将有助于降低云服务商服务中断或开始与您的企业竞争的风险。为了遵守相关法规，许多组织将会选择能帮助确保业务连续性的服务商。比方说，如果您在某个区域的运维出现问题，备用服务商可让您避免遭遇停机情况。

利用开放标准的多云迁移通常表现良好。以 Kubernetes 为例，它提供了一个服务商中立的 API，可用于运行、配置和部署应用，以及用于集成安全政策、网络等。您可以将 Kubernetes 视为多云操作系统；将它作为抽象层时，您通常可以忽略大多数主要云服务商之间的差异。

采用全托管式服务,降低运营开销

使用 Kubernetes 时,您拥有充分的选择权。您当然可以自己运行;Kubernetes 是一个开源项目,因此您可以下载它并花费数年时间将其集成到您的云服务商环境或自己的首选环境中。

但是如果您觉得这样效率太低,则可以使用托管式 Kubernetes 产品。如果您在 AWS 上,就选择 EKS。如果在 Azure 上,则选择 AKS。如果您在 Google Cloud 上,则应该选择 [Google Kubernetes Engine \(GKE\)](#)。所有这些方案都会为您提供一个通用的 Kubernetes API,当您的团队构建其工具和工作流时,您可以在不同的服务商处重复使用它们。

不过,并非所有托管式服务都一样。Kubernetes 的能力上限取决于其运行的基础架构,而 GKE 是成熟的全托管式 Kubernetes 编排服务,可以弥补能力的不足。它提供完全集成的 IaaS,包括 [Tau VM](#) 的虚拟机预配(与同类通用产品相比性价比高 42%¹)、跨多个可用区和升级版本的自动扩缩,以及按需创建和管理 GPU、TPU 以用于机器学习、存储卷和安全凭据。您只需将应用放入容器中,然后根据需要选择系统即可。

如果已经选择 AWS 作为虚拟机的云服务商,该怎么办?需要继续使用 EKS 吗?从整体上看,Kubernetes 对所有云服务商而言并无差异;您最终都会使用 Kubernetes API。但该 API 的背后是包括集群、工作器节点、安全政策在内的所有一切。在这方面,GKE 具有明显优势。

例如,如果您仍然需要其他集群,则可以将它们连接到 [GKE Connect](#),从而在一个位置集中管理、查看和调试集群并对其进行问题排查,同时集中管理凭据等。GKE 是出色的 Kubernetes 产品,这不仅是因为其控制平面、多区域或多可用区的高可用性,更是因为它的端到端易管理性。GKE 还可以跨多个集群和多个区域使用集中管理的 [多集群 Ingress](#),从而利用全球负载均衡器。

如果您希望使用 Kubernetes API,但又不希望自己来预配、扩缩和升级集群,该怎么办?对于大多数工作负载,[GKE Autopilot](#) 会将集群的底层基础架构(包括节点和节点池)抽象化,您只需为工作负载付费。GKE Autopilot 可为您提供具备所有强大安全默认设置的标准 Kubernetes API,因此您可以专注于工作负载,而不是集群。

¹结果基于 SPECrate@2017_int_base 估算值。测试中使用的是其他两家领先的云供应商的生产环境虚拟机以及采用了供应商推荐的编译器的测试环境 Google Cloud Tau VM。SPECrate 是 Standard Performance Evaluation Corporation 的商标。有关详情,请访问 www.spec.org。

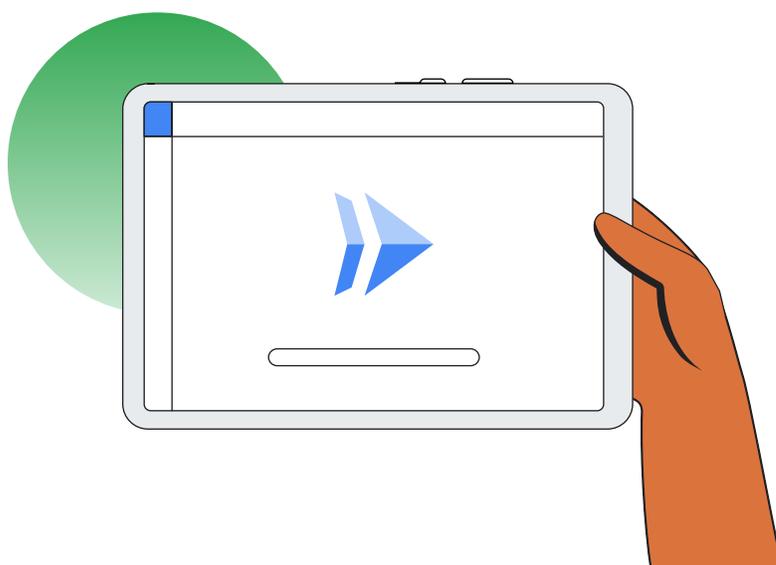
利用无服务器技术简化应用交付

Kubernetes 将组织从虚拟机迁移到一组新的抽象,这些抽象可让您实现运营自动化并专注于您的应用。但是,对于更具体的工作负载(例如 Web 和移动应用、REST API 后端、数据处理、工作流自动化),您可以利用无服务器模型进一步简化和优化部署。

您可能正在使用 AWS Lambda,这是一个热门的无服务器平台,可让您编写函数即服务,并将其连接到各种事件。但由于您最终会连接到数据库并处理安全问题,这些函数往往会变得越来越复杂,有些函数实际上比普通应用还要大。因此,如果您的某个应用的发展使之不再适合函数即服务这种简单的模式,或者您想要以无服务器方式运行现有应用,会发生什么情况?

与需要重写应用的传统无服务器平台不同, [Cloud Run](#) 提供了一种方法,可帮助您重复使用现有的容器化应用投资。虽然 GKE 是一项托管式服务,但您仍然需要做出一些关键决策:在哪些可用区运行,在哪里存储日志,如何管理不同应用版本之间的流量,如何注册域名,如何管理 SSL 证书。

使用 Cloud Run 则不必做出上述决策,因此您能够运行更多传统工作负载,通过停用“缩减至零”来完全避免冷启动。如果您的应用必须始终运行,Cloud Run 也支持始终运行,另外还支持 NFS、WebSocket 和 VPC 集成等其他传统要求。但与大多数传统的无服务器平台一样,Cloud Run 具有固有特性,并提供内置流量管理和自动扩缩等功能。



第 3 章

如何让您的迁移支出发挥最大作用

理清迁移逻辑

假设您根本没有使用容器，并且不知道从何处入手。以下是采用容器化技术的实用方法。

01

采用容器的第一个原因是**解决打包问题**。如今，大量的工作用于生成可重现的制品和了解软件内的实际结构，也就是业界所说的“[安全软件供应链](#)”。我们认为，实现此目的的理想方法是利用包含应用代码和运行时的容器映像及其依赖项。容器的一个好处是它们可以部署到虚拟机，从而降低跨机器部署和维护软件的复杂性。

02

采用容器的第二个原因是**编排**。管理虚拟机需要大量管理和维护开销。如果您与大多数公司一样，您的团队需要执行数十个或数百个步骤来管理基础架构，那么即使这些步骤是自动化的，这些自动化工具仍然需要持续维护。这是假设您使用 Terraform、Puppet、Chef 和 Ansible 等业界标准自动化工具的情况。如果使用自行开发的工具或定制化工具，维护开销会更大。

03

采用容器的第三个原因是**效率**。除了维护负担外，大多数组织的 CPU 利用率仅为 5-10%，更不用说内存和存储空间。这会浪费大量资源。许多团队会构建更多定制化工具，通过实现自动扩缩和装箱等功能来改善这一问题，继而浪费了更多运营时间。这会导致超支和高昂的云费用。

很少有组织能够成功构建自己的编排平台,因此利用 Kubernetes、Mesos 或 Nomad 等开源平台是一种常见的选择。但是,如果您希望有一个平台帮助您减少维护工作、采用业界标准最佳实践、与云服务商的其余功能进行深度集成,并为您带来其他种种好处,GKE 这样的代管式服务是很好的选择,可帮助您极大地提高容器的潜在价值。



成功迁移的基础知识

此时您可能会问:迁移至容器真的需要长时间停机,且完全没有价值吗?您最不希望看到的就是需要暂停未来的开发,对吗?

为解答此问题,我们来看看如何充分利用您一路收获的云经验。

将应用从虚拟机迁移到容器似乎是一项令人望而生畏的工作,尤其是在需要将所有计算都迁移到新的云服务商处时。但实际情况是,您无需一次完成。您可以一次迁移一个应用,从虚拟机开始,然后将一两个应用迁移到 Kubernetes,这些应用可以位于同一网络上并与虚拟机通信。您不需要一次彻底完成迁移,而是可以慢慢地从一个平台迁移到另一个平台。

对于少数应用,继续留在虚拟机上可能更适合,例如很难从 Kubernetes 受益的大型数据库。这也是完全可行的,您可以混搭。但我们看到,大多数客户都通过将应用和开源项目迁移到 Kubernetes 环境获得了很多好处。优先处理迁移到 GKE 可以带来最多回报的应用;而且您无需一次就迁移所有应用。

Kubernetes 仍在您当前可能正在使用的 Linux 虚拟机上运行。您真正获得的是更简洁、更一致的工作流,其中包含应该在您的基础架构路线图上的功能,而不是一堆自己开发的脚本和自动化。使用 Kubernetes 等业界标准工具后,新团队成员的入职培训也变得容易得多,因为不必再教他们贵公司所采用的所有定制化方法。

走到这一步,您就有了一条可以逐步进行但切实可行的容器采用路径,从而节省团队在管理开销上花费的时间,并节省公司的计算费用。

准备好踏出 下一步了吗？

我们在这里讨论了很多关于容器和云迁移的问题, 无论您是希望全面采用 Kubernetes 并在寻找更好的方案, 或者是希望寻找适合现有应用和路线图的其他方法。无论您的应用开发路径是什么, 托管式容器方案都可以极大地降低基础架构费用, 同时极大地提高团队专注于构建优质产品的能力。基础架构的未来是容器化。现在, 您需要确保您的工程师和企业做好一切准备去取得最大成功。



如需详细了解业界领先的 Google Cloud 托管式服务如何帮助您充分利用现代应用, 请与我们联系。

咨询专家



补充阅读材料

- [从 Kubernetes 到云原生: 开启您在 GKE 上的开发之旅](#)
- [云中的容器 - 一种更优越的开发和部署应用的方法](#)
- [与 Eric Brewer 一起探讨 Kubernetes 的过去、现在和未来](#)
- [根据云原生的特点重新设计架构: 一种大规模提高开发者工作效率的演进式方法](#)

