白皮书 - **2018 年 6 月**

在虚拟机还是裸机上运行容器?

大规模部署和安全管理容器化应用

vmware®

目录

引言 3

安全性 3

- 容器本身是不够安全的边界 3
- 在物理主机上配置错误的风险 4
- 保护编排系统 4
- 充分利用各种先进趋势 5
- 使用虚拟机保护微服务 5

可用性 5

资源管理 6

数据持久性 6

性能 6

可扩展性 7

网络连接 8

基础架构管理和 IT 运维 9

在 VMware SDDC 中的虚拟机上运行容器 9

VMware Pivotal Container Service 10

- 体系结构 10
- 使用 BOSH 最大限度减少开销并简化管理 11
- 使用 Harbor 保护容器镜像 11
- 使用 NSX-T Data Center 为容器工作负载实现网络连接和安全保护 11

结束语：多云灵活性、管理和安全性 12

“尽管有时人们认为容器超越硬件虚拟化而成为了虚拟化的下一个阶段，但事实上，大多数企业更需要循序渐进，而不是剧烈变革。容器和硬件虚拟化不仅能够很好地共存，而且实际上能够增强彼此的功能。虚拟机具有诸多优势，例如强大的隔离功能、操作系统自动化以及广泛而深入的解决方案生态系统。企业不需要在容器和虚拟机之间进行选择，而是可以继续使用虚拟机执行硬件部署、分区和管理，同时使用容器打包应用，更高效地利用每台虚拟机。”

APPLICATION CONTAINER SECURITY GUIDE，NIST 特别出版物 800-190

引言

有人说，容器让硬件虚拟化变得多余了：既然可以在物理硬件上运行容器，为什么还需要虚拟机 (VM) 呢?

在生产环境中运行应用需要遵守一系列既定的运维要求：安全性、合规性、性能、资源管理、可扩展性、可用性、数据持久性、网络连接和监控。运行容器化应用也不例外。但是，它们有一个额外的要求：编排。

您可以冒着极大的风险花费高额成本在物理硬件上构建自定义体系，以尝试满足容器化应用的要求，或者，您可以将经验证、经济高效的低风险虚拟化解决方案用作容器及其编排系统的底层基础架构。

而后一种方法还有一个好处：将容器和虚拟机结合起来可以利用两种技术的优势，创造一个一加一大于二的有组织整体，这也是 Google 和 Amazon 等各大主流云服务提供商使用虚拟机来运行容器的原因之一。借助虚拟机，您能够在可轻松管理、监控、扩展和优化的软件定义的基础架构上，安全高效地在生产环境中运行容器化应用。同时，借助容器，您能够提高开发人员的敏捷性，增加应用的移动性，并提升部署的自动化程度。这两者相结合，可以简化企业级应用的开发、部署和管理。

本文通过技术解释和证据援引，回应了人们对于在虚拟机上运行容器的质疑。本文认为，将容器与虚拟机结合在一起能够为可靠、稳健地大规模部署和运维容器化应用提供完美的催化剂。VMware® Pivotal Container Service 使用 Kubernetes 在 VMware Software-Defined Data Center 中的虚拟机上编排容器，它是这个组合的核心要素。

安全性

2017 年 9 月，美国国家标准与技术研究院 (NIST) 发布了“Application Container Security Guide”，也称为 NIST 特别出版物 800-190。它介绍了容器的安全性问题，并提供了建议的解决方法。该指南揭示了人们在容器方面关注的几个基本领域：

- 隔离程度
- 操作系统管理和配置
- 保护措施不足的编排系统

容器本身是不够安全的边界

容器不是微型虚拟机，容器也不会像虚拟机那样建立安全边界。“Application Container Security Guide”的一个重要暗示就是要在虚拟机上运行容器化应用：该指南表明，容器“不会提供像虚拟机一样清晰和具体的安全边界。多个容器共享同一内核，并且可以在一台主机上

“Docker 容器与虚拟化技术相得益彰，虚拟化技术除了保护虚拟机自身，还可以为容器主机提供深度防御。”

DOCKER 安全白皮书

以截然不同的功能和权限运行，因此，它们之间的隔离程度远低于 hypervisor 为虚拟机提供的隔离程度。”[1]

随虚拟机部署容器可以提供两个隔离层来封装应用，这种方法非常适合具有多租户和多个工作负载的云环境。一份 Docker 安全性白皮书表明，“Docker 容器与虚拟化技术相得益彰，虚拟化技术除了保护虚拟机自身，还可以为容器主机提供深度防御”。[2]

要在运行 Linux 的物理主机上正确隔离租户的容器，您需要在不同的物理机上运行不同的租户。可能的结果有两种：一种是碎片化导致资源利用率低下；一种是由于利用率过高，需要等待很长时间才能使用新硬件。Google 和 Amazon Web Services (AWS) 等主流云服务提供商通过独立的虚拟机来隔离租户的容器工作负载。容器是不够安全的边界，因此，只有可信度极高的代码才能在同一虚拟机或物理主机上的容器中运行。

这一点同样适用于 Kubernetes 中的单元 (pod)。“最终，对于同时在虚拟机和容器中运行的应用，虚拟机可以提供最终的安全屏障。就像您不会在同一虚拟机上运行具有不同安全级别的程序一样，您也不应该在同一节点上运行具有不同安全级别的单元 (pod)，因为这些单元 (pod) 之间缺乏有保证的安全边界。”Jianing Guo 在 Google Cloud Platform 博客上写道。[3]

在物理主机上配置错误的风险

NIST 的“Application Container Security Guide”表明，容器或物理主机的操作系统很容易被错误配置，从而扩大受攻击面并增加风险等级。“如果环境配置不仔细，与同一主机上的多个虚拟机相比，容器能够更容易、更直接地相互交互并与主机相互交互。”

相比之下，对 hypervisor 中的虚拟机上运行的操作系统实现抽象化、自动化和隔离可减小受攻击面，并降低出现安全漏洞的风险。

保护编排系统

“Application Container Security Guide”的另一个关注点是提供建议的对策，以保护用来管理容器的编排系统。NIST 指南中建议的对策包括以下几条：

- 使用带强大凭证和目录服务功能的企业级身份验证服务
- 基于主机、容器和镜像对管理操作实施精细访问控制
- 根据容器中运行的应用的敏感度级别将容器隔离到不同的主机

1 NIST 特别出版物 800-190，Application Container Security Guide，作者：Murugiah Souppaya，NIST 信息技术实验室计算机安全部；John Morello，Twistlock，美国路易斯安那州巴吞鲁日；Karen Scarfone，Scarfone Cybersecurity，美国弗吉尼亚州克利夫顿。2017 年 9 月，https://doi.org/10.6028/NIST.SP.800-190

2 Introduction to Container Security，Docker 白皮书，Docker.com。

3 “Demystifying container vs VM-based security: Security in plaintext”，Google Cloud Platform 博客，作者：Jianing Guo，2017 年 8 月 9 日，https://cloudplatform.googleblog.com/2017/08/demystifying-container-vs-VM-based-security-security-in-plaintext.html。

NIST 的另一份文件“Security Assurance Requirements for Linux Application Container Deployments”阐述了在生产环境中部署容器化应用时的一些安全要求和对策，有助于满足“Application Container Security Guide”中的各项建议。编排系统或其组件和工具应具备以下功能：

- 对容器的资源使用情况进行日志记录和监控，以确保关键资源的可用性
- 编排系统必须与多个容器主机（而不仅仅是一个）配合使用，才能为所有正在运行的容器提供资源使用情况的全局汇总

如果要使用编排系统在物理硬件上运行容器并管理容器，您需要将每台物理机连接到一个身份验证和访问控制系统。

要按敏感度级别隔离容器，必须使用大量物理机，这是非常低效的。因此，资源利用率将受到影响，同时管理开销将增加，更糟糕的是，NIST 还要求使用多种类型的容器主机。[1]

充分利用各种先进趋势

通过在虚拟机上运行容器，您可以充分利用虚拟化技术中的各种安全创新。AMD SEV-ES 就是一个例子。安全加密虚拟化 (SEV) 技术将内存加密与 AMD-V 虚拟化集成在一起，能够支持适合多租户环境的加密虚拟机。

具有加密状态的 SEV (SEV-ES) 基于 SEV 构建，即使 hypervisor 遭到破坏，它也可以通过 hypervisor 为客户虚拟机缩小受攻击面并提供额外的保护。当虚拟机停止运行时，SEV-ES 通过加密和保护所有 CPU 寄存器内容来阻止攻击，以防止 CPU 寄存器中的信息泄露到 hypervisor。SEV-ES 可以检测并防止对 CPU 寄存器状态的恶意修改。[4]

使用虚拟机保护微服务

微服务为容器安全性增加了另一个维度。根据一份有关安全性的 Docker 白皮书，“共同部署虚拟机和 Docker 容器可以让一整组服务相互隔离，并在虚拟机主机内完成分组”。[2]

可用性

使用微服务完美构建的容器化应用可以依赖像 Kubernetes 这样的容器编排系统来管理可用性。但是，大多数容器化应用的体系结构并不完美：它们可能是经过重建且分成几个宏组件的整体式应用，也可能只进行了部分重构，即，使用了一些微服务，而其余部分仍采用 n 层模式。此类应用仍然依赖（至少是部分依赖）底层基础架构来保证可用性。VMware Software-Defined Data Center 提供 VMware vSphere® vMotion®、VMware vSphere High Availability 和 VMware vSphere Distributed Resource Scheduler™ (DRS) 等诸多经验证的技术，它们对于保持可用性至关重要。

4 Protecting VM Register State with SEV-ES，David Kaplan，AMD，2017 年 2 月，https://support.amd.com/TechDocs/Protecting%20VM%20Register%20State%20with%20SEV-ES.pdf。

“应用可以受益于虚拟机提供的安全性和性能隔离，并且仍然充分利用容器在调配和部署方面的优势。在公有云环境中运行容器时，隔离和安全性是主要关切，这种方法非常流行。”

CONTAINERS AND VIRTUAL MACHINES AT SCALE: A COMPARATIVE STUDY

即使是使用微服务完美构建，可以处理自身可用性的容器化应用，也可以从软件定义的基础架构中获益。例如，Redis 是常驻内存的复制数据库，它可以通过以下方式容许基础架构故障：启动一个新的 Redis 实例，替换故障实例。但是，在新实例启动时，它会从其他 Redis 节点复制数据，直到其状态完全还原为止。不过，为复制操作传输数据需要付出代价：它会降低 Redis 集群的整体性能。一种更高效的方法是使用 vMotion 移动原始 Redis 节点，以免出现性能受损的情况。

资源管理

Kubernetes 提供功能强大的服务质量 (QoS) 机制，可在运行不同工作负载的团队之间共享集群。在 vSphere 上运行 Kubernetes 集群可以对 Kubernetes 的 QoS 机制形成补充，特别是当您需要强大的工作负载隔离功能时。vSphere 的高级调度和动态资源管理有助于在团队之间或跨 Kubernetes 集群回收并共享未使用的资源。

vSphere 的资源管理功能包括智能初始安置、动态重新均衡、资源池、份额、预留、限制和安全超额分配。借助所有这些功能，您能够在通用基础架构上运行传统及容器化工作负载，同时确保最佳性能并防止工作负载之间出现干扰。

对于在 vSphere 上运行的 Kubernetes 集群，vMotion 和 DRS 等 VMware 技术可在不中断工作负载的情况下动态地重新均衡集群，从而最大限度提高硬件利用率。

数据持久性

虽然有很多容器化应用是无状态的，但移植应用到容器也需要容器能够支持有状态应用，使它们能够利用主机本地存储、非共享存储来实现跨主机的数据持久性。

然而，管理物理存储设备是一个棘手的手动过程，通常会因特定于应用的工作流而雪上加霜。通过新增固态硬盘来扩展容量的方法十分低效。跨各种孤立的基础架构小环境部署不同的技术和流程会使情况变得复杂，此外，将物理硬件专用于个别应用从经济上来说也很浪费。

如果有一个适用于应用（无论是否容器化）的软件驱动模式，则可以从根本上简化存储管理、运维、故障排除、容量扩展，以及备份和灾难恢复等存储操作。通过提供分布式的无共享存储抽象化，VMware vSAN™ 可简化存储操作，并整合同一存储基础架构上的传统和云原生工作负载。

性能

借助 VMware ESXi™ 的 CPU 调度程序，hypervisor 能够为容器提供与物理硬件上运行的 Linux 系统等效或较之更好的总体工作负载性能。

“容器可以保证裸机性能，但正如我们所展示的，在多租户场景中，它们可能会受到性能干扰。容器共享底层操作系统内核，而这会导致缺乏隔离。与具有严格资源限制的虚拟机不同，容器也允许软性限制，这在超额分配场景中很有帮助，因为它们可以使用分配给其他容器但没有被充分利用的资源。由于缺乏隔离，外加软性限制促成了更高效的资源共享，在虚拟机内部运行容器成为了可行的体系结构。”

CONTAINERS AND VIRTUAL MACHINES AT SCALE: A COMPARATIVE STUDY

VMware 的一项对比研究表明，与在裸机上的 Docker 容器中运行相比，在 vSphere 6.5 上的 Docker 容器中运行的企业级 Web 应用性能更高，这主要是因为 vSphere CPU 调度程序针对非一致内存访问 (NUMA) 体系结构进行了优化，推翻了“在虚拟机上运行容器会降低性能”这种观点。[5] vSphere 做得好的一点是，它会在内存驻留的 NUMA 节点上调度虚拟机。而 Linux 则试图最大限度地提高处理器利用率，这意味着它可能在未驻留内存的 NUMA 节点上调度进程，从而减缓内存访问速度并降低性能。一项针对 vSphere 上大数据工作负载的性能分析显示了相同的结果。

与在 Linux 中运行容器相比，虚拟化可以提供更好的性能隔离，特别是在邻位干扰情况下。一项针对规模化容器和虚拟机的学术对比研究结果表明，“位于同一位置的应用可能导致性能干扰，对于某些类型的工作负载，容器的干扰程度更高。”[6] 因 Linux 内核工作方式的原因，您还会受到共享相同内核资源或组件的诸多容器的跨容器干扰。“内核会按容器完全隔离每个底层资源”的假设是错误的。[7]

如果在 vSphere 上运行 Kubernetes 集群，也将大有裨益。如果在裸机上运行 Kubernetes，其性能不太可能胜过在 vSphere 上运行 Kubernetes，因为 vSphere 使用高级调度算法优化所有工作负载，包括使用容器的工作负载。在物理硬件上运行 Kubernetes 的企业会发现自己难以扩展和高效运维基础架构。与物理硬件相比，在 vSphere 上运行 Kubernetes 集群的优势在于：多年的微调使其非常擅长优化大型集群和混合工作负载的性能。

可扩展性

Hypervisor 最初是为解决使用物理硬件的难题而开发的；这些难题从耗时的管理问题和耗钱的利用率低下问题，到难以针对开发人员环境或应用不断扩展的工作负载扩展硬件的问题，不一而足。通过优化利用率，虚拟化使您可以在提高可扩展性的同时降低物理硬件成本。

如果 IT 运维部署了一个带容器运行时的裸机服务器，开发人员可以将容器推送到其中，那么该系统的扩展工作会既困难又耗时：您必须添加另一个裸机服务器，在其上安装一个容器运行时，手动将服务器连接到网络，并将运行时连接到容器编排引擎。

相比之下，借助 hypervisor，您可以在几分钟内将一个新的裸机服务器连接到容器域。虚拟化带来的易扩展性是主流公有云提供商使用 hypervisor 来运行容器服务的原因之一。例如，当您在 Google Container Engine 或 Amazon Elastic Container Service 上创建一个 Kubernetes 集群后，相应的云服务提供商将启动一个或多个虚拟机。与裸机相比，虚拟机的周转速度要快得多。

5 《在 VMware vSphere 6.5 上的 Docker 容器中运行的企业级 Web 应用的性能》(Performance of Enterprise Web Applications in Docker Containers on VMware vSphere 6.5)，VMware，2017 年 9 月，https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/performance/docker-vsphere65-weathervane-perf.pdf。

6 “Containers and Virtual Machines at Scale: A Comparative Study”，Prateek Sharma，Lucas Chaufournier，Prashant Shenoy，Y.C.Tay；Middleware'16，2016 年 12 月 12-16 日，http://dx.doi.org/10.1145/2988336.2988337

7 “Container isolation gone wrong”，作者：Gianluca Borello，Sysdig，2017 年 5 月 22 日，https://sysdig.com/blog/container-isolation-gone-wrong/

容器微分段

微分段使用网络虚拟化将数据中心及其工作负载划分为多个逻辑分段，每个逻辑分段包含一个工作负载。然后，您可以对每个分段应用安全控制措施，限制攻击者移动到另一分段或工作负载的能力。

这一点在全球范围内也适用。如果您要在某数据中心的裸机上构建 Kubernetes，您如何将其扩展到全球的 1,000 个站点，同时维持安全性、网络连接、监控级别和集中管理？

对于大型集群来说，在裸机上使用 Kubernetes 也会对规模利用率产生不利影响。从 Kubernetes 版本 1.10 开始，Kubernetes 文档中已发布的受支持的集群配置如下：[8]

- 节点数不超过 5,000
- 单元 (pod) 总数不超过 150,000
- 容器总数不超过 300,000
- 每个节点上的单元 (pod) 数不超过 100

如果每个单元 (pod) 只有一个容器（这并不少见），那么，若有任何一个容器功能低下，则裸机上就会有未使用的资源。

但是，有了 vSphere，您可以在每个物理主机上运行超过 100 个单元 (pod)，从而提高底层硬件的利用率。

网络连接

容器依赖于网络连接中的三个抽象化层：

- 底层网络
- 叠加网络
- 服务网格

底层网络使用基于硬件的传统方式或组合使用硬件和软件的方式将虚拟机或物理机连接起来。叠加网络位于底层网络之上，在容器和主机的生命周期中提供网络连接能力，例如 IP 地址和端口。服务网格在 IP 地址和端口之上移动，专注于为容器化应用连接服务。

在这种情况下，如何安全高效地连接大量运行 Linux 的裸机容器主机？如果您使用将所有内容都推送到叠加层的扁平 L2 网络，那么，您将不得不按多个物理主机组对容器化应用进行分组（因为容器本身是不够安全的边界）。

但是，如果容器的网络连接能力未进行隔离，并且与容器的生命周期无关，那么，攻击应用可以连接到环境中的所有其他物理主机。为了确保网络安全性，需要借助 VLAN、东西向防火墙或其他技术在环境中进行硬件级网络隔离，而所有这些技术都需要手动、耗时且与容器化应用的生命周期无关的管理。当生命周期发生变化时，需要执行更多手动且容易出错的网络更改。

VMware NSX® Data Center 也在虚拟机上实现了单一底层网络，可为容器和传统应用提供端到端连接和管理。单一底层网络具有如下几个优势：

8 “Building Large Clusters”，Kubernetes，https://kubernetes.io/docs/admin/cluster-large/

- 将容器化应用轻松连接到传统的非容器化组件（如数据库）。
- 使用集中式策略和高级安全功能（如微分段）从根本上简化网络管理。[9]
- 选择最适合您的容器化应用的叠加网络和服务网格。

NSX Data Center 与容器网络接口集成，以提供叠加网络。当新的容器化应用部署完毕后，NSX Data Center 可以自动新建一个虚拟网络，将该应用与环境中的其他所有应用完全隔离开来。

基础架构管理和 IT 运维

在物理硬件上运行容器会再次带来基础架构管理和运维方面的难题。Kubernetes 管理的是容器化应用，而不是运行它们的底层基础架构。如果您要选择使用物理硬件作为底层基础架构，则必须满足大量需求，同时避免形成难以管理的孤立小环境：

- 基础架构部署和配置
- 修补、更新和升级
- 备份和灾难恢复
- 日志记录和监控

多个孤立的基础架构小环境可能需要多个重复的团队、工具和流程。IT 最终会不断重复执行相同的任务，而不是在单一平台上专注于推动创新。

如果在物理主机上运行容器，虚拟化已经解决的许多旧问题会再次困扰 IT，而与此同时，IT 面临巨大的压力，他们需要在不增加复杂性和风险的情况下提高敏捷性、帮助应用缩短投产时间、快速采用新服务并管理成本。这些都是当前的核心 IT 需求，或很快将成为新的需求。但是，随着异构云计算服务进入企业，IT 发现自己越来越难满足这些需求。使用 Pivotal Container Service (PKS) 的 VMware Software-Defined Data Center (SDDC) 可通过全面而灵活的解决方案来解决这些问题；该解决方案使用 BOSH 的强大功能来部署和管理多个 Kubernetes 集群，并修补和升级容器主机操作系统。

在 VMware SDDC 中的虚拟机上运行容器

vSphere 与其他多种 VMware 技术相结合，可提供全体系 SDDC，在虚拟机上安全高效地运行容器，同时轻松管理底层基础架构。vSAN 提供分布式且可扩展的软件定义的存储。NSX Data Center 为容器提供软件定义的网络虚拟化。此外，PKS 将 Kubernetes 和 BOSH 与此 SDDC 集成在一起，以编排容器化应用，将它们扩展到云端，并管理它们的底层基础架构。

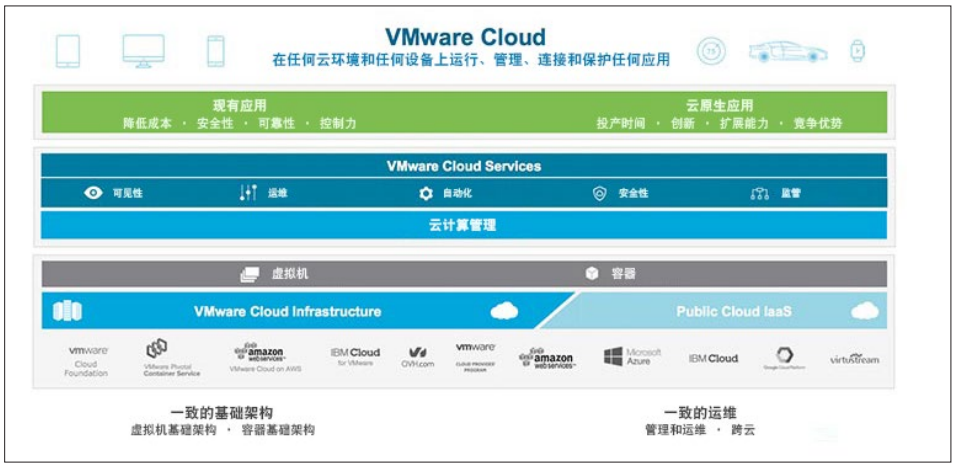
9 Micro-segmentation for Dummies，作者：Lawrence Miller 和 Joshua Soto，John Wiley & Sons, Inc.，2015 年。

VMWARE PKS 概览

VMware PKS 提供高度可用且基于 Kubernetes 的生产级容器服务，具有来自 VMware NSX Data Center 的高级网络连接、名为 Harbor 的安全镜像仓库以及使用了 BOSH 的生命周期管理等功能。该解决方案可以从根本上简化 Kubernetes 集群的部署和运维，以便您可以在 VMware vSphere 上大规模运行、管理、保护和维护容器。

VMWARE PKS 的主要优势

- 快速按需调配 Kubernetes 集群
- 通过滚动升级、运行状况检查和自动修复，为 Kubernetes 组件提供高可用性。
- 使用包括微分段、负载均衡和安全策略在内的高级容器网络连接功能
- 通过漏洞扫描和镜像签名来保护容器镜像
- 通过监控、日志记录和分析提高运维效率



在使用 VMware Pivotal Container Service 的 VMware SDDC 中的虚拟机上运行容器，可以在一致的基础架构上实现一致的运维。

最终将产生一个可与现有工作负载互操作的通用平台（规范数据集通常仍然驻留在其中），以及公有云（新应用将植根于此，且可以使用机器学习和分析等关键服务）。

在虚拟机（而不是物理硬件）上运行容器时，这个通用平台可最大限度降低运维复杂性并简化管理，同时推动实现最高的硬件利用率和最大的规模经济。

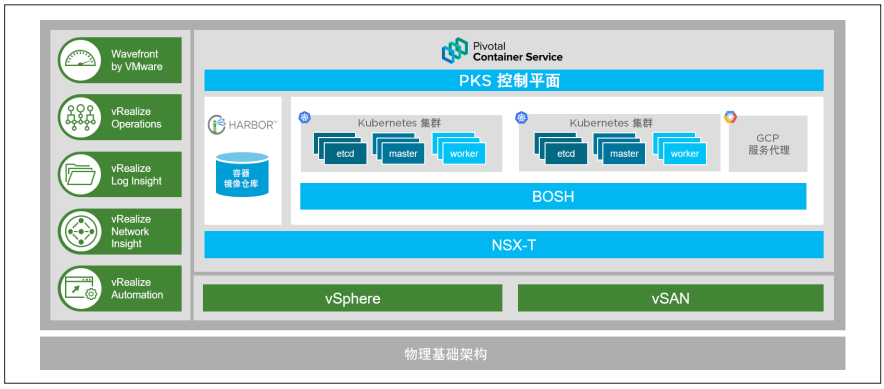
VMware Pivotal Container Service

尽管容器本身并非新事物，但诸多障碍让它们无法用于构建和部署企业级应用。容器之前缺乏用于实现企业级部署、管理、运维、安全性和可扩展性的工具和生态系统，这种状况直到最近才有了改观。此外，IT 管理员的要求之前也常常得不到满足，因为运行容器的基础架构忽视了网络连接、存储、监控、日志记录、备份、灾难恢复、维护和高可用性。

而 PKS 提供了基于 Kubernetes 的生产级容器服务，具有高级网络连接、专有容器镜像仓库和全面的生命周期管理等功能。此解决方案从根本上简化了 Kubernetes 集群的部署和运维，以便您可以在 vSphere 和公有云上大规模运行和管理容器。

体系结构

PKS 将 Kubernetes、BOSH、VMware NSX-T™ Data Center 和 Harbor 结合，形成一种高度可用的容器服务。PKS 将这些开源商用模块连接在一起，以高效部署和管理 Kubernetes 及在其上运行的容器。



VMware Pivotal Container Service 的体系结构。

使用 BOSH 最大限度减少开销并简化管理

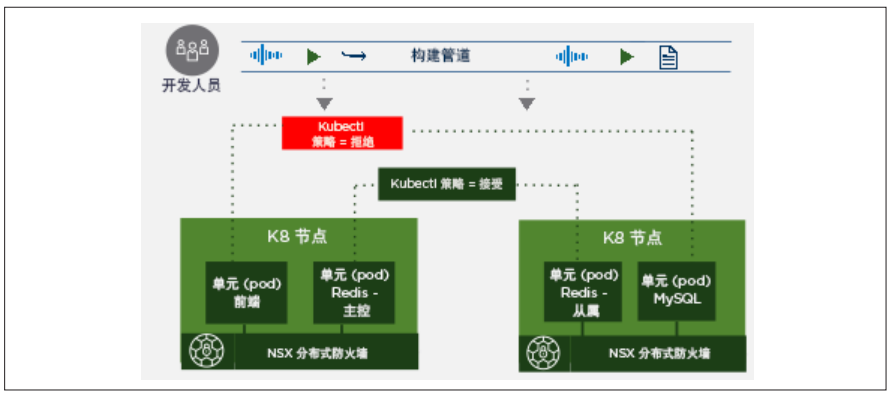
BOSH 是一款用于发布工程的开源工具，可简化大型分布式系统的部署和生命周期管理。PKS 使用 BOSH，以一致且可重现的方式对 Kubernetes 进行版本管理、打包和部署。通过使用 BOSH，PKS 支持多种不同基础架构即服务 (IaaS) 提供商的部署，如 vSphere、Google Compute Platform。

使用 Harbor 保护容器镜像

Harbor 是 VMware 的一个开源的企业级容器镜像仓库，用于在防火墙后的专有镜像仓库中存储和分发 Docker 镜像。Harbor 包括基于角色的访问控制、容器镜像的漏洞扫描、基于策略的镜像复制、与 LDAP 或 Microsoft Active Directory 的集成、公证和审核服务。

使用 NSX-T Data Center 为容器工作负载实现网络连接和安全保护

NSX-T Data Center 为 Kubernetes 集群提供高级容器网络连接、安全策略和微分段功能。它提供 Kubernetes 中单元 (pod) 级网络连接所需的第 2 层到第 7 层的全套网络连接服务。您可以快速部署网络，为容器和单元 (pod) 提供微分段和按需网络虚拟化，包括负载均衡和传入服务。



NSX 可在 Kubernetes 上提供单元 (pod) 级网络连接和安全策略。

了解有关 VMWARE PIVOTAL CONTAINER SERVICE 的更多信息

要详细了解 VMware 如何帮助您构建、运行和管理云原生应用，请访问 https://cloud.vmware.com/pivotal-container-service。

NSX-T Data Center 与 PKS 的集成对虚拟机上容器的网络运维产生了直接而深远的影响：

- 对 NSX-T Data Center 负载均衡器的原生支持可以为在 Kubernetes 集群上运行的应用提供高度可靠且高性能的流量分布。
- 微分段策略的优先级高于 Kubernetes 的标准安全策略。
- 网络策略有助于保护不同 Kubernetes 命名空间之间以及同一命名空间中不同单元（pod）之间的流量。
- 运维工具和故障排除实用程序可以调试单元 (pod) 之间的通信。

结束语：多云灵活性、管理和安全性

借助使用 PKS 的 VMware Software-Defined Data Center 在虚拟机（而不是物理硬件）上运行和编排容器，可以满足容器化应用在运维、管理和安全方面的一整套要求，同时将其移动性延展到云端。使用 PKS 在 vSphere 上运行和管理容器的多云灵活性具有一系列优势：

- 使用隔离、强安全边界、身份验证、访问控制、镜像漏洞扫描、微分段和其他措施保护容器及编排系统。
- 使用经验证的 VMware 技术（如 vMotion 和 DRS）保持高可用性。
- 使用 vSphere 的资源管理功能最大限度提高 Kubernetes 集群的硬件利用率。
- 使用 vSAN 和分布式的无共享存储简化存储操作并整合工作负载，从而为容器化应用提供数据持久性。
- 优化大型集群和混合工作负载的性能。
- 无需费力添加和配置物理硬件，即可扩展容器化应用。
- 使用 NSX Data Center 简化网络管理并提高网络安全性。
- 最大限度降低运维复杂性并简化管理，同时推动实现最高的硬件利用率和最大的规模经济。

使用 PKS 在 VMware SDDC 中的虚拟机上运行容器，这可以将经验证的虚拟化技术的优势与容器和 Kubernetes 编排的新优势强强融合。这种组合产生了一种可持续的多云解决方案，它具有强大的功能和灵活性，可推动您的云原生战略取得成果。

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com/cn
威睿信息技术（中国）有限公司
中国北京办公室 北京市朝阳区新源南路 8 号启皓北京东塔 8 层 801 邮编：100027 电话：86-10-5976-6300 传真：86-10-5976-6302
中国上海办公室 上海市淮海中路 333 号瑞安大厦 805B-809 室 邮编：200021 电话：+86-21-8024-9200
中国广州办公室 广州市天河路 385 号太古汇一座 3502 室 邮编：510610 电话：+86-20-87146110
中国香港公司 香港港岛东太古城太古湾道 12 号太古城中心 4 期 4 楼 电话：852-3696 6100 传真 852-3696 6101 www.vmware.com/cn