



白皮书

追本溯源：为何代码推动基础架构投资

借助云架构基础设施建立竞争优势并实现增值

云架构基础设施：为持续创新注入动力

有关当今企业变革局势的新闻频繁见诸报端。新兴的创新型高科技企业往往处于变革的风口浪尖。这些企业成绩斐然，惯于发布网络级软件和更新。这种网络级发布能力往往取决于整个公司范围内的 DevOps 框架。DevOps 在推动企业持续创新、快速部署软件以及频繁更新软件功能和产品方面，提供了无限先机。这些企业的成功，一部分要归功于企业底层数据中心基础架构和流程对 DevOps 的支持。

传统企业与新兴企业可以复制这条成功之路吗？从上至下，直至数据中心基础架构级别，这条成功之路需要哪些支持？如今，大多数企业已经认识到，软件产品是一项核心差异化优势，可以帮助企业吸引新客户、维护客户忠诚度、抵御竞争对手的进攻。正如一名评论员所说：“我们所说的企业不必是技术企业……实际上，目前所有企业都可以称为技术企业，无论他们接受与否。”¹

此外，大多数公司也认识到，传统的应用程序开发与部署框架已不足以胜任它们的工作。他们将能够快速进行微服务部署的全新 DevOps 模式视为发展重点。因此，许多公司开始尝试在公司各个层面开展 DevOps 实践，从 HR 到数据中心，无一例外。因此，他们调整流程和数据中心基础架构，以推动技术和新型软件产品的发展。

从宏观上讲，这样固然很好。但从微观上看，这种转型究竟意味着什么？这种创新型 DevOps 理念将促使企业在数据中心基础架构方面展开怎样的实际行动和投资呢？

本白皮书将介绍企业在过渡到下一代数据中心的过程中有可能采取的一些实际行动以及可能经历的一些必要阶段。一些趋势对这种灵活、具有高能力和高创新水平的企业的形成起到了辅助作用，本文也将对其中一些辅助趋势予以介绍。

数据中心新兴创新引擎简述

这种新型的云架构基础架构是什么样子的？这是一种能够帮助企业快速无缝地创建、测试、为不断增长的客户群交付和部署新创意和高质量软件的基础架构，一种可以为企业赢得竞争优势的基础架构。这种新型组织不可或缺的一部分是拥有一个成功的 DevOps 流程和底层数据中心基础架构，帮助协调开发人员需求与运营和测试需求。结果如何呢？结果是最终获得一个有助于持续开发、测试、集成和交付高质量软件和新功能的顺畅的工作流。

从基础架构角度来讲，该部门可以根据需要快速地自动完成生产级开发和测试环境的搭建或拆除。这与一种通过本机 API 和无缝集成顺畅地配置、更改或监控底层基础架构组件的可扩展基础架构紧密相关。这意味着开发和测试团队可以毫不费力地通过代码使用这些基础架构组件，比如使用基本定义文件、API 调用，或者使用与他们自己的、熟悉的 DevOps 工具、框架和平台（例如 Kubernetes、Ansible、Jenkins、JFrog、VMware、OpenStack 和 Splunk）集成的随时可用的基础架构插件。



数据中心新兴创新“引擎”之瞥

这还意味着运营团队再也不需要处理冗长的资源配置与手动存储管理工作。实现自动化后，运营团队可以更专注于监控生态系统，确保合理部署基础架构工具、集成和自动化策略，全力推动企业创新。

您的企业与这种全新数据中心基础架构有多大差距呢？如何消除这个差距呢？

DevOps 的兴起

Damon Edwards 在《Short History of DevOps》(DevOps 简史) 中谈到：“DevOps 可以说最早是从 Twitter 的一个话题标签 (#devops) 引发的一场技术变革，至今也仍然是焦点话题。”³ DevOps 对于不同的人意义也不同，它可以是一场草根运动，也可以是一个哲学理念、一场革命、一场文化改革、一个软件框架，或是一组流程和工具。

“可扩展性衡量的是一项技术向现有结构添加额外元素和功能的能力。例如，如果某个软件程序可通过附加产品和插件进行扩充增强，即视为具备可扩展性。”²

来源：Techopedia.com

与软件开发生命周期 (Software Development LifeCycle, SDLC) 通常采用的“瀑布式”冗长的传统软件开发和部署方法相比，DevOps 具有完全不同的特点。虽然 DevOps 仍然支持 SDLC 的主要原则（开发、测试、部署和分析），但可以大幅缩短整个循环周期，并消除开发、测试和运营团队之间经常存在的壁垒以及手动操作带来的延迟。同时，DevOps 实践还可以促进企业内跨所有组织实施项目和创新举措。这一系列的业务和技术优势使得 DevOps 转型备受青睐（见图 1）。

如今的企业组织日益强烈地感受到 DevOps 的影响。一个典型例子就是沃尔玛 (Walmart)。沃尔玛公开表示欣然接纳 DevOps 实践和使用 OpenStack 和 Puppet 等开源软件。这家零售巨头通过这些实践和框架来管理基础架构，例如自动执行服务器构建流程、简化应用程序云迁移。⁴

无论是开源还是传统企业级软件框架，DevOps 实践都能与之和谐相容，因此已经开始在创新型软件即服务 (SaaS) 提供商以及其他优秀的 IT 企业中大放异彩。Puppet《2018 年 DevOps 状况报告》⁵ 中对 30,000 名专业技术人员的一项调查显示，高绩效的组织通过 DevOps 原则可以将部署代码的频率提高 46 倍。从提交到部署，它们还能将提前期缩短为原来的 1/2555，将故障率降低为原来的 1/7。

DevOps 如何推动“通过代码使用基础架构”这种新消费需求

无论您选择如何定义 DevOps，都不能忽视它的影响。现在，它已经开始影响当前和未来的基础架构决策。

对于企业组织来说，底层基础架构如何无缝支持 DevOps 工作、甚至在多大程度上实现隐形支持，变得日益重要。这对于公司能否实施创新并快速应对新的市场需求同样至关重要。

Robert Stroud 是一名 Forrester 分析师，专门研究与 IT 基础架构和运营相关的趋势，他在一篇博文中提到了“通过代码使用基础架构”这种模式。在这篇博文中，他对许多企业的云计划趋势这样描述：“要过渡到云优先策略，则需要转变基础架构交付、管理和维护方式，这样才能将其作为一种可重复使用的软件组件进行交付和使用。这种基础架构可以是虚拟的，也可以是物理的，可以按需消费，没有冗长的构建和部署周期。”⁷

DevOps 的兴起让我们进入了一个新时代：“通过代码使用基础架构”的消费需求成为一种新常态，随之而来的是使用基于代码的框架来自动部署、管理、更改或支持底层基础架构组件。



技术优势

软件持续交付

降低问题复杂性

可见性提高

问题解决速度加快

返工和计划外任务减少



业务优势

加速交付功能

运营环境稳定性提高

资源利用效率提高

有更多时间创造附加价值

修复和维护用时减少

系统运行结果可见性提高

图 1) DevOps 的优势。

“能够动态地进行迭代和调整，这一价值不可估量。我们不会再深陷于三个月前的决策而难以自救。”⁶

— Polaris Alpha 公司信息系统部门高级副总裁 David Coker

这意味着，开发人员将使用代码来执行各种操作，例如为自己的虚拟机或容器环境创建和挂载存储卷等。同样，测试人员也可以通过代码使用最新的生产级数据，无缝、快速地配置或构建自己的测试台环境，所有这些都不需要再额外添加存储。能否实时按需提供资源是满足 IT 新需求的关键，通过代码无缝调用和配置基础架构组件的能力也同等重要。

为了帮助企业组织完成这一过渡，运营和基础架构团队以及基础架构供应商需要考虑以下问题：

- 底层基础架构的可扩展性如何？
- 我们的基础架构如何才能不仅限于支持应用程序或存储数据？它能如何为 DevOps 和整个企业创造价值？
- 基础架构与企业的自选工具和框架的集成能力如何？
- 使用基于软件代码的方法（API、插件、本机集成等）调用或访问基础架构组件有多简单可行？
- 基础架构能够以多快的速度适应不断变化的需求和状况？

对这些问题的回答或许代表了不同方法之间的差异：资源配置需要几分钟还是几周；采用直观的自动化工作流还是费力的手动流程；以及采用快速而简化的部署还是麻烦重重的交接和延迟。此外，答案的不同还可能意味着，仅仅满足应用程序的需求还是真正为企业创造更多价值，这二者之间的差异。

南非提供商 Internet Solutions 通过 NetApp® SolidFire® 存储基础架构轻松地与 OpenStack 和 VMware 实现了集成。该基础架构可以使 Internet Solutions 的持续集成/持续交付 (Continuous Integration/Continuous Delivery, CI/CD) 工作流实现全面自动化。同时，与传统架构相比，DevOps 服务以及带容器的虚拟机部署速度提高 15 倍之多。

“SolidFire 与 OpenStack 的集成简直不可思议。无论从集成还是从可扩展性角度看，它都为这一领域带来了了不可估量的价值。”⁸

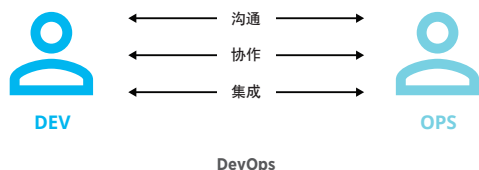
— Internet Solutions 技术总监 Kervin Pillay

DevOps：为现有基础架构理念与功能注入新的活力

需要管理数据中心基础架构环境的企业常常要竭尽所能确保运行不会中断。在管理为数据中心应用程序提供支持的计算、网络或存储资源方面，这往往意味着，不仅需要避免风险，还要确保系统可用性、性能和可靠性。就存储系统而言，这一问题可以进一步细化，即通过一些功能来最大限度地减少数据丢失和停机，同时缩短恢复时间。有关这方面的典型例子是为数据集快速创建快照（即时间点 (Point in Time, PiT) 副本）。

快照能够快速恢复，因此经常用作企业 IT 灾难恢复计划中的一个重要组成部分。

但是，如今随着 DevOps 的兴起，以及随之而来的通过代码创建基础架构方法的诞生，我们需要以新的视角来看待基础架构功能（例如快照和克隆）。在下一代数据中心中，这些技术和功能还可以为企业带来哪些价值？



例如，如果开发人员或测试和 QA 人员可以直接调用快照和克隆，会给 DevOps 带来什么价值？如果开发人员或测试/QA 人员可以使用这些功能来快速访问或刷新生产级数据或按需自行添加新卷，又会如何？

在现有 DevOps 工具和框架下，这些功能可以通过与基础架构集成的方式来直接使用，无需多个团队花费几个月的时间来单独构建。有了这些功能，不再需要处理因配置交付周期很长而产生的过时数据集，因为按需实时交付数据集更容易在流程的更早期即引入代码质量保证流程。唯一需要做的就是，配备合适的可扩展基础架构类型，以及掌握部署该基础架构的专业知识。

如果基础架构可以在需要时无缝提供，那么在开发、测试和部署阶段可以节省多少时间，避免多少令人烦躁的手动交接工作呢？

以上这些仅仅是新型数据中心优势的冰山一角。

“我们可以随时在 10 分钟内向任何人提供一份银行副本。只需要单击一个按钮，开发人员即可得到所需的所有数据，包括与数据相关的所有服务器、应用程序、配置、测试数据等等，并且是经过全面验证、随时可投入使用的现成品。然后，他们便能运行想要进行的任何测试。这太神奇了！”⁹

— ING DIRECT Australia IT 战略主管 Ben Issa

打造成熟的 DevOps 方法和基础架构实践

在本白皮书开头部分，我们描述了 DevOps 实践正处于巅峰期，正在全面加快创新速度的新形势。在这种情况下，基础架构可以无缝地移动到一个面向持续开发和持续交付的快速工作流程中。

但现实情况是，无论是 DevOps 成功率，还是探索新兴的 DevOps 工具或动态自动使用基础架构等方面，许多企业组织尚未达到这个理想的境界。此外，在实际工作中，大多数企业组织还发现在其发展进程的不同时期需要关注不同的领域。

就 DevOps 而言，每个企业关注的领域可能根据其自身的业务优先级或自己独特的优势和劣势不断改变。有些企业可能会侧重改进代码质量或加快测试速度，而有些企业可能会努力提高系统可见性，深入了解其环境的运行状况。另外，还有一些企业可能会关注如何更快地找到错误并根据需要进行返工。有些企业可能会因部署频率降低（至少会在前期部署）而感到满意，而有些企业则可能会每天执行多次功能部署。企业的优先事项也可能会随着限制因素和需求的变化而不断改变。但是，为了支持 DevOps 框架，企业必须找到一种方法来支持以下六个核心功能：容器、配置管理、代码和二进制文件管理、持续集成/持续交付、云计算和平台即服务 (PaaS) 以及分析。

不管各家企业组织有什么独特的关注领域和目标，基础架构和运营团队都可以不断发展，打破现状，朝着未来的愿景进军。在此发展进程中，许多企业的基础架构都会经历若干关键成熟度阶段（见图 2）。

通过对众多企业级环境以及行业领先 SaaS 企业进行观察，我们发现，企业在过渡到下一代数据中心过程中，可能会经历四个常见的基础架构成熟度阶段。

大多数基础架构都已经过早期的融合、虚拟化等阶段，甚至已经开始利用脚本化任务来自动执行配置和报告等基本的基础架构任务。

但是，随着企业进一步发展到自动化阶段，他们会希望基础架构实现哪些目标呢？概括来说分为四步。

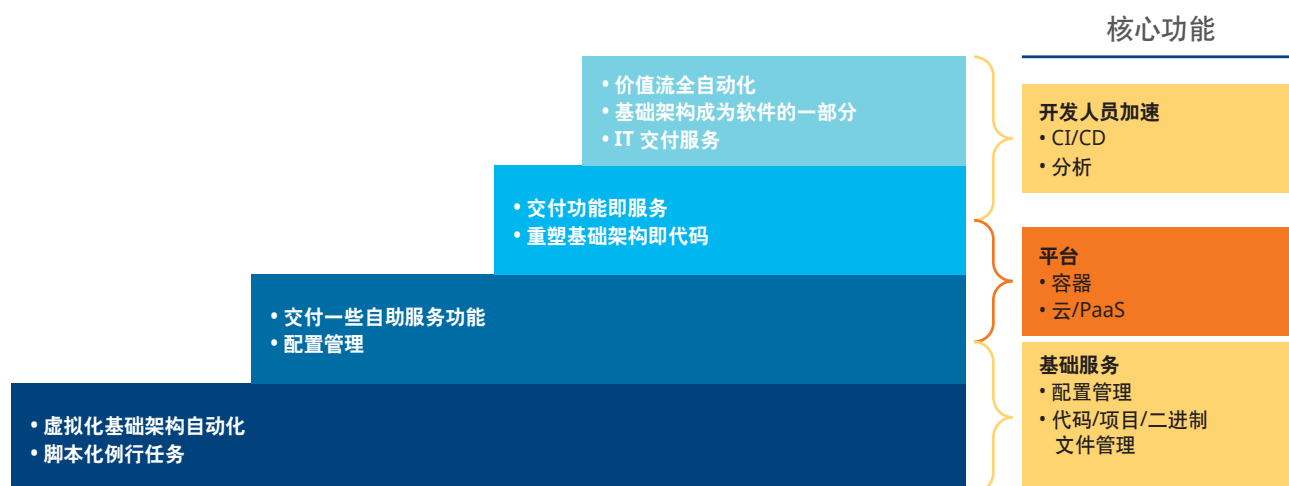


图 2) 成熟之路。

第 1 步：向基于基础架构的 API 集成迈进

这一步明确通过代码提供基础架构功能的目的。在此阶段，专攻基础架构集成的 DevOps 团队成员应着眼于识别可能对其环境和优先事项有用的可扩展基础架构特性、功能以及可能的 API 集成。他们应咨询基础架构供应商，了解可为开发人员提供的首选工具集的集成水平。然后应该开始考虑如何合理部署这些组件，推进 DevOps 进程向前发展。

这一步工作的示例包括通过代码还是通过基础架构控制台或图形用户界面访问基础架构功能。理想情况下，这项工作应包括充分利用与通用工具集的本机集成。使用此类本机集成有助于最大限度地减少所需代码量。除此之外，可能还会涉及到使用 REST 协议访问基础架构功能。如果无法实现本机集成，则可以将这些功能集成到您自选的 DevOps 工具或框架中。

现在，Premier Eye Care 不仅能在数分钟内刷新开发环境，而且还可以利用 NetApp SolidFire 阵列上的最新数据（而不是前一晚的备份数据）执行刷新。为了实现这一功能，在 NetApp 牵线下，Premier 与服务合作伙伴 Wirestorm 携手，创建了一个 Windows PowerShell 脚本。通过它，Premier 可以将 SolidFire 阵列的 NetApp Snapshot™ 副本自动用于开发环境的备份。

“这彻底颠覆了我们的工作方式。当我按下 SolidFire 的“刷新”按钮时，用户可以得到当前最新的数据。现在，如果我在星期五晚上五点接到电话，通知我内部用户需要刷新开发环境，我在几分钟内即可搞定，然后回家。用户拿到手的再也不是前一天晚上的旧数据了。”¹¹

— Premier Eye Care 数据库管理员 Rob Connock

第 2 步：消除交接工作，推动深度自动化

尽管此时可能已经实现了一定程度的自动化，但第二个成熟度步骤要求开发人员和基础架构运营团队从软件开发/测试/部署周期中进一步消除人员和手动交接工作。企业内各组织应尝试自动执行通常可能涉及向其他团队提交服务单或请求的工作流。枯燥的手动任务也应自动执行，例如通过自动化策略准则或为开发人员工具提供基础架构插件等。

这一步涉及的示例包括让应用程序所有者或开发人员使用 Docker 卷插件通过预定义的操作准则创建存储卷，并将其挂载到操作系统中。

“SolidFire 的精简性既简化了存储管理又为 DevOps 团队提供了支持，从此存储工程师可以自由执行意义重大的计划。我们正在努力通过基础架构保持我们的市场优势。”¹²

— FICO 平台运营与工程部门高级经理 Donald Talton

第 3 步：通过基础架构打造自助服务

在这一步中，运营团队的重点任务是，如何通过进一步降低参与度来简化向用户交付自助式基础架构务。在这一阶段，开发和测试团队不需要与运营或基础架构团队沟通，便可以在需要时获得所需的工具。这可能涉及到构建自助式基础架构系统，在系统中将相关技术与开发人员常用工具完美地集成在一起，不管这些工具涉及 Ansible、Jenkins 还是 Kubernetes，或者是需要通过私有云系统获得。

这可能意味着扩大配置管理工具（例如 Ansible）的使用范围，以便可以直接从配置管理工具集内部连接或更换基础架构组件（例如存储）。

在此阶段，管理员、运营团队和存储团队的角色发生了变化。他们不必再亲自执行所有基础架构任务；相反，任务是通过由使用者直接启动的代码自动执行的。运营人员可以集中精力全面了解、监控和管理基础架构，不必等待其他团队提供信息或亲自执行手动任务。在此过程中，安全性、可靠性和可用性丝毫不会受到影响，并且可以通过基于策略的管理以及使用代码将基础架构集成到常见 DevOps 工具中，无缝实现自助服务。

这一阶段的示例有：使用 Jenkins 定义测试框架的测试团队。在该框架中，测试团队可能需要对一个基础架构运行多个测试台环境。使用可扩展的基础架构来实现这种自助服务，有助于快速配置或刷新测试环境，从而更快地完成对更多比较数据的测试。测试团队可以完成所有这些测试，不必等待交接给基础架构团队。

其他自助式基础架构还包括，例如通过 Kubernetes 卷插件使开发人员直接配置和管理与特定应用程序关联的永久性存储，或者是开发人员通过使用 VMware vSphere 虚拟卷或 Ansible 等其他技术来直接更改底层基础架构环境，而无需人工干预。

“我的团队可以将部署系统的数量减少到原来的 1/10，并将总部署时间从 20 个小时减少到不到一分钟。”¹³

— Wirestorm 工程部门副总裁 Jeremy Goodrum

第 4 步：采用软件定义的基础架构 (SDI) 框架

如果开发和测试团队不需要与运营或基础架构团队沟通，便可以在需要时获得所需的工具，那么您其实已经开始过渡到真正意义上的软件定义的基础架构了。

这一步不只是能够在任意硬件上安装软件，还应该能够设置策略和框架，以便根据应用程序不断变化的需求按需更改基础架构，同时始终保持基础架构的稳定性、可靠性、可用性和可见性。

此阶段可能涉及到需要开发人员更改底层基础架构环境。这一变化可能需要通过策略来改善某些基础架构行为和性能特征，例如与特定应用程序卷相关的行为和特征。借助 VMware vSphere 虚拟卷或 Ansible，团队也可以通过此类策略实施任何必要的更改。基于策略的框架可以使基础架构变得灵活机变，并且更易于管理和使用。

助您启航

向 DevOps 与下一代数据中心迈进并非朝夕之功，也不可能一帆风顺，企业需要从不断的尝试、失败和学习中获得必要的技能、知识和经验。

本白皮书总结了一些趋势、不同阶段和建议的里程碑，为企业提供了一盏指路明灯，可帮助企业顺利抵达梦想的彼岸。在旅程的初期阶段，企业应认识到可扩展基础架构在实现 DevOps 的关键阶段性目标方面所发挥的重要作用。

为了顺利采用 DevOps 模式并将 IT 变为创新引擎和利润中心，企业需要利用这种底层基础架构流动性进行扩展和缩减，并在需要时做出调整，满足开发人员和测试/QA 团队不断增长的需求。

基础架构和运营团队对于顺利完成这一过渡至关重要。虽然角色会改变，但仍然需要利用他们的专业知识，来为相应任务合理地评估、甄选、实施、监控和管理适当的可扩展基础架构组件。

对特定可扩展基础架构功能（例如插件、API、SDK、其他集成以及自动化策略功能等）的可用性加强关注后，这些可以获得充足的自信。同时，他们还应更认真地审视相关的合作伙伴关系，通过其自身、基础架构供应商以及其他各方之间的通力合作，帮助他们更顺畅地过渡到可通过代码调用或使用的自助服务和软件定义的基础架构。

其他资源

[详细了解](#)软件开发生命周期及其与 NetApp 的集成。

参考

- 1.《进化而非变化：野蛮生长的 DevOps》，作者：Boyd E.Hemphill，2015 年 4 月 2 日，DevOps.com，<https://devops.com/evolution-not-revolution-devops-wild/>。最后访问日期：2017 年 5 月 3 日。
- 2.《“可扩展”定义》，Techopedia，<https://www.techopedia.com/definition/7107/extensible>。最后访问日期：2017 年 4 月 19 日。
- 3.《DevOps 简史》，作者：Damon Edwards，<http://itrevolution.com/the-history-of-devops/>。最后访问日期：2017 年 4 月 19 日。
- 4.《Walmart 因 OpenStack 上的 213,000 个核心而引以为豪》，作者：Linda Hardesty，2017 年 2 月 24 日，SDxCentral，<https://www.sdxcentral.com/articles/news/walmart-boasts-213000-cores-openstack/2017/02>。最后访问日期：2017 年 4 月 19 日。
- 5.《2018 年 DevOps 状况报告》，Puppet，<https://puppet.com/resources/whitepaper/state-of-devops-report>。最后访问日期：2018 年 12 月 7 日。
- 6.《DevOps 方法以更出色的敏捷性帮助 Polaris Alpha 加快上市速度》，NetApp，<http://www.netapp.com/cn/media/cs-polaris-alpha.pdf>。最后访问日期：2018 年 12 月 21 日。
- 7.《基础架构即代码：基础架构和运营议程中缺失的一环》，作者：Robert Stroud，2016 年 2 月 9 日，Forrester，<http://blogs.forrester.com/robert-stroud/16-02-09-infrastructure-as-code-the-missing-element-in-the-io-agenda-0>。最后访问日期：2017 年 4 月 19 日。
- 8.《我们不再需要叉车式升级》，NetApp，<http://www.netapp.com/us/media/cs-internet-solutions.pdf>。
- 9.《ING DIRECT 借助 NetApp、Cisco 和 Microsoft 私有云加速创新》，NetApp，<https://www.netapp.com/cn/media/ing-direct.pdf>。最后访问日期：2018 年 12 月 21 日。
- 10.《DARZ Docker 和容器即服务通过 DevOps 促进数字化转型》，客户成功案例：服务提供商 DARZ，NetApp，<http://www.netapp.com/us/media/cs-internet-solutions.pdf>。
- 11.《医疗保健公司利用 SolidFire 消除低效存储》，客户成功案例：Premier Eye Care，NetApp，<http://www.netapp.com/cn/media/cs-premier-eye-care.pdf>。
- 12.《通过闪存驱动型云服务加快上市速度》，客户成功案例：软件服务/FICO，<http://www.netapp.com/cn/media/cs-premier-eye-care.pdf>。
- 13.《速度快成为 Wirestorm 的 DevOps 新标准》，客户成功案例：服务提供商/WireStorm，NetApp，<http://www.netapp.com/cn/media/cs-fico.pdf>。

要验证您的特定环境是否支持本文档所述的确切产品和功能版本，请参见 NetApp 支持站点上的 [互操作性表工具 \(Interoperability Matrix Tool, IMT\)](#)。NetApp IMT 中定义的产品组件和版本可用于构建 NetApp 所支持的配置。具体的配置结果取决于每个客户如何依照所发布规格进行安装。

版权信息

版权所有 © 2018 NetApp, Inc. 保留所有权利。中国印刷。未经版权所有者优先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

本文档中所含数据与商用项目（按照 FAR 2.101 中的定义）相关，属于 NetApp, Inc. 的专有信息。美国政府对这些数据拥有非排他性、不可转让、无转授权、全球性、受限废除的许可，但仅限于与交付数据所依据的美国政府合同有关和支持此类合同的情况。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b) 条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。

WP-7292-0219-zhCN

全国销售热线：4008-1818-11