



白皮书  
轻松地为容器配置  
持久存储

加快应用程序开发速度，同时提高  
DevOps 效率



挑战：打造 DevOps 成熟度	3
采用大容器的障碍：为有状态应用程序管理永久性存储	3
容器、数据永久性以及无状态与有状态应用程序的简短概述	4
良好开端：容器永久性存储的基本配置	4
结论：取得了一些进展，但仍有太多等待	4
存储类、存储池和 Trident：按需提供永久性存储	5
构建存储类目录有那么容易吗？	5
DevOps 展望	6
转变您的工作方式	6
帮助您及您的企业取得成功	7
其他资源	7

### 挑战：打造 DevOps 成熟度

持续不断地创新；多快好省地推出产品；更大程度地简化运营；越来越多的客户感到满意。

这些是众多企业普遍怀有的目标。也是实现更出色业务成果的里程碑：更高的收入增长和盈利能力。

企业在取得这些成就的同时，往往会经历更多的转型阵痛和挫折。应用程序开发人员急于加快开发工作来解决客户需求，因此他们可能会对服务单系统以及长时间等待存储和计算资源感到失望。相比之下，基础架构和运营 (I&O) 团队则难以处理不断传入的服务单，而且很难控制基础架构的使用方式。

这种目标和成长的痛苦混合在一起，推动着企业追求敏捷与精益的方法，以提高产品开发的速度和效率。在敏捷与精益方法发挥出巨大优势的同时，如果 IT 基础架构和流程能够协调一致地提供企业所需的效率和速度，则可以实现更大的优势。

这正是 DevOps 的用武之地。DevOps 方法旨在更快地实现应用程序开发和 IT 基础架构运营两方面的转型。

企业能否成功实现 DevOps 成熟度，通常取决于对六种关键能力的掌握程度：

- **代码、项目和二进制文件管理。**用于保留和管理软件组件的存储库
- **配置管理。**通过已知方式配置和维护基础架构和软件系统
- **云/PaaS。**使用公有云、私有云和混合云基础架构支持软件开发
- **容器。**高度可扩展的轻型应用程序运行时环境
- **分析。**自动化监控和管理基础架构
- **持续集成/持续部署 (CI/CD)。**支持开发人员编写并自动部署代码的端到端自动化流程

上述六个方面中，无论哪一个方面——特别是容器——对于希望加快速度并提高效率的应用程序开发和基础架构运营团队来说，都具有越来越重要的意义。本白皮书将从容器的角度介绍向 DevOps 成熟度迈进的过程，同时重点分析必须考虑的关键问题之一：为有状态应用程序管理永久性存储。

### 采用大容器的障碍：为有状态应用程序管理永久性存储

当 DevOps 团队开始考虑更为实际的容器化应用程序生产部署时，各种难题便纷至沓来。其中最棘手的是管理容器的永久性数据存储。

如果您需要更好地了解诸如容器数据永久性或者有状态与无状态应用程序等概念，请参见第 4 页的简短插页。

早期在移动应用程序容器时，处理数据存储就已经成为一大挑战。云原生计算基金会 (Cloud Native Computing Foundation, CNCF) 所做的一项调查<sup>1</sup>显示，近一半受访者 (42%) 表示，存储和资源管理是采用容器的主要难题 (图 1)。其中许多人提到了一直存在的与存储永久性相关的问题。而其他受访者则表示希望更方便地访问网络存储。

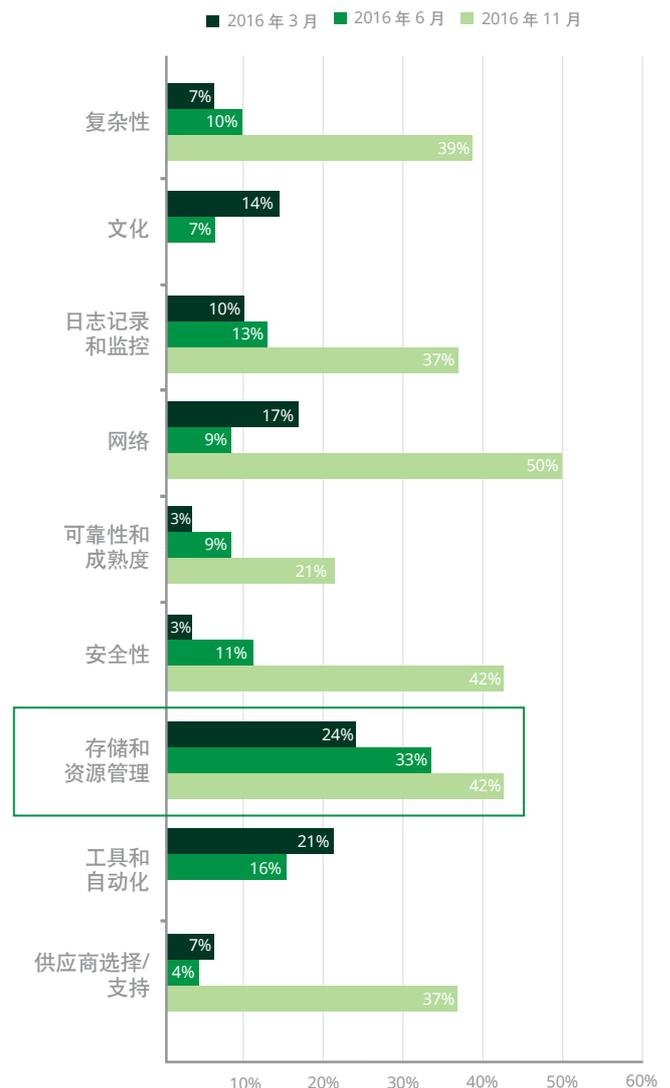


图 1) 采用容器面临的难题 (允许采取多种应对措施)。来源：云原生计算基金会。

因此，容器平台开始着力解决永久性存储难题。早期的永久性存储配置工作取得了良好的开端，但仍然有些缺乏灵活性，而且过于依赖手动操作。加快 DevOps 管道运行速度也似乎不可行，尤其是在需要对应用程序进行编码、测试并部署到数百个或数千个容器时。

从可扩展性角度来看，必须手动为数千个容器配置存储的情况开始变得让人无法忍受：过于繁琐、容易出错而且难以维护。因此必须采取更好的方式。

在容器出现之前，许多企业级应用程序均通过集中连接到共享企业级存储成功满足了永久性数据存储需求。在容器环境中，如果这些有状态的应用程序能够很容易地访问类似的共享存储功能，它们就不能正常运行吗？不过仍然存在这样的问题：它能在多大程度上轻松访问此类存储？

### 良好开端：容器永久性存储的基本配置

Docker 和 Kubernetes 等容器环境最初采用有用的半自动机制应对存储永久性难题。这种机制支持用户创建和“请求”特定的永久性存储卷，供一个或多个容器进程使用。这就要求存储管理员首先基于各种底层网络存储资源创建永久性卷。

### Kubernetes：实际的基本配置操作

对于 Kubernetes，该进程采用了与永久性卷 (PersistentVolume, PV) 和匹配的永久性卷请求 (PersistentVolumeClaim, PVC) 关联的基于代码的机制。

管理员必须先创建一个静态永久性卷（例如，一个具有 8 GB 全闪存存储容量的卷）。然后，应用程序用户（或开发人员）可以使用几行代码请求或申请该特定永久性卷。

请求 PV 并将其与 PVC 绑定对容器环境具有吸引力，因为它能让容器化应用程序开始通过代码使用永久性网络存储。

虽然这是启用网络存储的良好开端，但是这种基本配置方法仍然面临许多挑战。其中最大的难题莫过于需要实现自动化。

### 结论：虽然有所进步，但等待时间仍然太长

通过使用预先配置的 PV 及其 PVC 对，在解决容器永久性存储问题方面成功迈出了第一步。然而，对于开发和运营来说，大部分存储配置过程仍然主要是手动、静态和重复的工作。这给应用程序开发和 DevOps 团队带来了一些效率低下的问题，这些团队希望减少交接并自动化执行 CI/CD 任务：

### 容器、数据永久性以及无状态与有状态应用程序的简短概述

从本质上讲，容器是无状态的。其内容存在的时间很短暂。这意味着，在给定会话期间，可以快速启动、停止和重新启动容器的相关应用程序或进程。默认情况下，这意味着在容器处于活动状态时创建的数据将在容器终止或销毁时立即销毁。

遗憾的是，当 DevOps 团队希望在容器平台上开发更多应用程序并将其部署到生产环境中时，在任何单个容器生命周期之后保留或保持数据的需求纷纷浮现。

许多人很快发现，几乎每个应用程序至少都有一个进程或微服务需要使用一个永久性卷 (PV)，用于存储在给定容器的生命周期之后需要维护的状态数据。

下面是一些需要数据永久性的有状态应用程序的示例：

- **数据库环境。**数据库容器需要让其数据存储库使用的存储永久化。然而，考虑到容器的短暂性，如果没有帮助，这是不可行的。本地存储也不是一个合适的选择。如果容器移动或销毁，它将失去对这些数据的访问权。
- **环境数据或会话数据。**有状态应用程序往往还会收集和保存应用程序环境属性或客户会话数据（状态）。这些数据作为历史数据可增强客户体验。在这种情况下，当客户端下次与应用程序交互时，它可以显示相关数据，或者更好地操作在前一个会话中创建的数据。

与需要保存状态数据同等重要的是需要共享数据。构建或迁移到容器化应用程序的企业很快发现，参与开发、测试和运营的每个人都需要访问来自同一集中化网络存储资源的相同数据集。存储在网络存储上的永久性卷还可以确保容器化应用程序得到企业级存储功能的更多保护，从而提高应用程序的可用性、可靠性和安全性，增强数据保护。

- **手动存储请求。**正在处理代码的应用程序开发人员或测试 Sprint 的 QA 人员，可能需要停止他们正在做的工作，要求存储管理员创建 PV。当扩展更大的容器应用程序时，这可能会涉及数百个或数千个持续的存储请求。
- **令人沮丧的存储等待时间。**开发团队或 QA 团队接下来可能需要等待请求获批，或者等待管理员安排时间手动创建一个或多个 PV。
- **存储使用效率低下。**使用预先为 PV 配置的存储支持开发和 QA，存储管理员可能容易配置过少（配置不足）或配置过多（过度配置）存储容量（图 2）。而且，他们为每个卷保留的 IOPS 也有可能过少或过多：
  - **配置不足的风险。**当为 PV 提前配置的存储太少时，可能会导致瓶颈，限制开发人员需要比现有可用容量更多的容量。
  - **过度配置的风险。**过度配置的存储资源现在锁定在一个或多个 PV 中，一旦绑定到对容量或性能要求小得多的 PVC 上，以后可能会被浪费。（还记得过去那些过度配置、利用率低下的存储孤岛吗？）
- **管理员资源的无效使用。**在进行多项应用程序开发工作时，参与存储或集群管理的人员可能发现自己在一个或多个 DevOps 管道中面临新的瓶颈。如果最后一小时出现请求多个新卷的情况，管理员在设置和创建每个静态 PV 时可能仍需要执行多个手动步骤。这种做法对于负责进一步自动化基础架构并监控其总体使用情况的管理人员也毫无帮助。

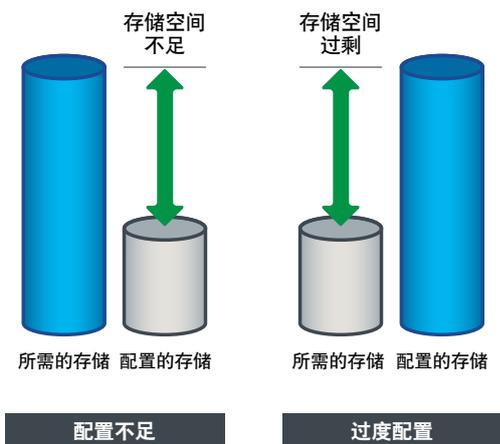


图 2) 配置不足或过度配置的风险。

如果您觉得听起来与过去的存储配置很像，那么有这种想法的并不只是您一个人。NetApp 技术营销工程师 Andrew Sullivan 在 2017 年的 Tech Field Day 演讲“**通过期望的状态进行 DevOps**”的最后，表示自己非常理解 DevOps 团队希望摆脱这种传统配置方法的心情。他表示：“**存储并不是我非得忍气吞声，在存储团队办公桌前乞求给予更多容量的一种资源。我不希望配置存储。我不希望在 2017 年还按 1989 年那样的方式使用存储。**”<sup>2</sup>

值得庆幸的是，NetApp 及其他支持容器环境的厂商知道必须采取更好的方式让 DevOps 团队按需使用存储，并在需要时随时随地动态地配置存储。

### 存储类、存储池和 Trident：按需提供永久性存储

最后，Andrew Sullivan 指出需要采取更好的方式配置存储资源。这种方式体现了容器最初的承诺：具有按需动态创建和部署功能，很少甚至不需要人工干预。

对 Kubernetes 而言，这意味着通过另一个概念“存储类” (StorageClass) 引入动态存储配置。它同时还意味着通过适用于 Kubernetes 的 Trident 这样的存储配置程序增强自动化：

- **按需提供永久性存储。**由 NetApp 开发的开源 Trident 项目是一个动态存储配置程序，它支持容器化应用程序在需要时按需访问 NetApp® 存储的永久性卷。再也不需要等待。
- **充分发挥强大底层存储的潜能。**使用 Trident 的容器环境能够充分发挥来自底层 NetApp 数据管理平台（例如 NetApp HCI、ONTAP®、NetApp SolidFire® Element® OS、SANtricity®）的强大存储功能。
- **并非仅仅适用于 Kubernetes。**通过 Trident 还可以为 Docker 和 OpenShift 提供类似的永久性卷功能。

### 构建存储类目录有那么容易吗？

使用 Trident，Kubernetes 环境中的开发人员现在只需通过从底层存储的虚拟池请求一个存储类，就可以动态地配置永久性卷。

下面我们来了解一下其工作原理。利用存储类，即可使用代码按需自动配置永久性卷。用户只需发出永久性卷请求，并将特定的 Trident 存储类命名为 Gold（金牌）、Silver（银牌）或 Bronze（铜牌）等。

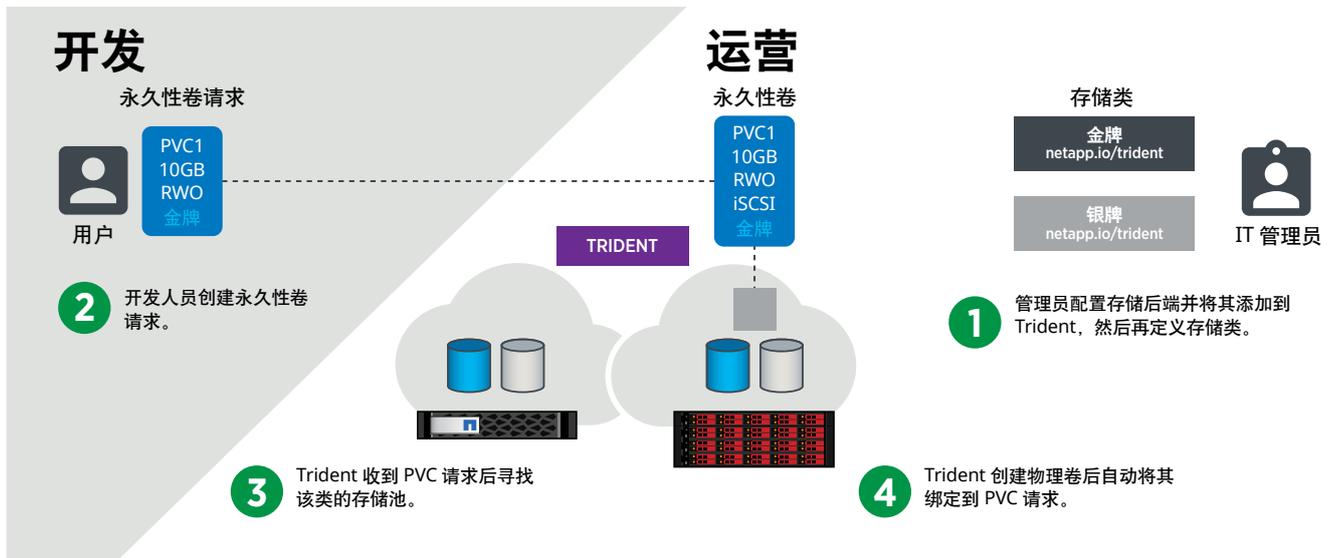


图 3) 使用 Trident 动态配置永久性卷。

注：底层存储类属性的配置和命名约定是一项后端功能，通常由管理员在构建永久性存储类的初始目录时执行。一个企业的存储类可以简单地命名为 Dev（开发）、Staging（试运行）和 Production（生产），另外也可以命名为 Fast（快速）或 Slow（慢速）。有关此类配置功能的更多详细信息，请参见 Trident [文档](#)。<sup>3</sup>

然后，从底层 NetApp 存储池中的存储类创建一个永久性卷。接着将创建的永久性卷绑定到用户的永久性卷请求 (PVC)。用户再也不需要了解底层存储的相关情况。Trident 会处理这些细节。这就是它的工作原理。

#### 观看 Trident 的实际操作

了解 Trident 价值的最佳方式是观看其如何进行动态配置。请花点时间看看下面的在线演示：

- [简短演示](#)（时长 3 分 20 秒）：[使用 Trident 为 OpenShift 动态配置存储](#)<sup>4</sup>
- [长版演示](#)（时长 23 分 53 秒）：[管理 Kubernetes 中的永久性数据](#)<sup>5</sup>

这种类型的配置功能对于 IT 和 DevOps 的不同成员意味着什么？

- **对于开发团队或 QA 团队而言。**再也不用等待服务单和存储请求审批的漫长过程。再也不用需要进行移交。使用具有 SLA 承诺的存储，无论您何时何地需要，都能使用熟悉的代码接口动态地配置。开发人员现在拥有了一个动态灵活的自动配置系统，在让他们获得解放的同时仍能保持对操作的控制。

- **对于 IT 或存储管理员而言。**不用再手忙脚乱地应对最后一刻的存储配置请求和无休止的服务单。存储管理任务减少。自动化程度更高的基础架构扩展。对存储使用进行更具预测性的控制，更轻松地监控资源。
- **对于 IT 主管而言。**更快速地交付产品，更加合理的流程并且显著节省资源。

#### DevOps 展望

DARZ 是一家全方位的 IT 服务提供商，通过 Docker 和容器即服务产品实现了 DevOps 敏捷性。此产品基于 NetApp 全闪存存储和适用于 Docker 的 Trident。客户不需要全能型的操作系统即可快速启用或停用应用程序容器，从而将计算需求削减为原来的 1/4。

采用精益灵活的容器化环境，客户可以缩短测试周期，加快开发速度，并且更快地部署新产品。通过 Docker 卷命令集简化容器-存储交互，Trident 使管理 Docker 环境中的永久性数据变得非常容易。<sup>6</sup>

#### 转变您的工作方式

除了能够动态配置永久性卷之外，Trident 等工具还支持 DevOps 团队执行更多任务。

其中包括通过代码按需访问 Snapshot™ 副本和克隆等 NetApp 的存储效率功能。对于需要以最少资源做到事半功倍的开发团队或 QA 团队来说，此类功能可改变游戏规则。

通过 NetApp 和 Trident 使用 Snapshot 副本和克隆可实现以下目标：

- **快速为完整生产数据集创建实时克隆。**这意味着只需几行代码就可以在几秒钟内动态创建新的开发人员或测试工作区。参见图 4。
- **便捷恢复数据。**借助 Snapshot 副本，开发人员可以快速将数据集回滚到先前版本。这对于测试代码很方便；开发人员可以快速进行迭代，而不用担心需要重新创建测试数据集。
- **通过 Snapshot 副本或克隆节省存储容量。**如果企业为生产数据创建多个克隆用于开发或测试，通常可以节省 40% 至 90%<sup>7</sup> 的存储容量。

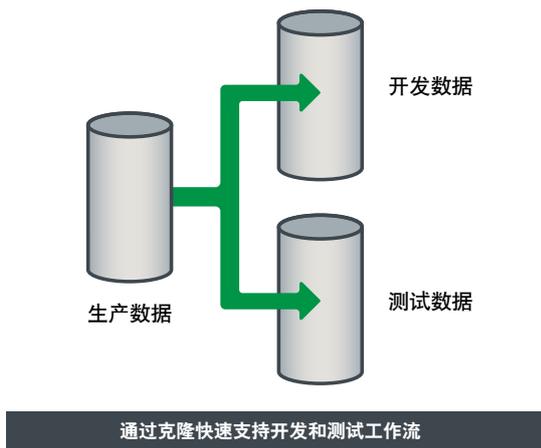


图 4) NetApp 克隆加速 DevOps 工作流的能力。

详细了解如何将 NetApp 克隆和 Snapshot 与 Trident 相结合使用：

- **对 Kubernetes 进行自助式卷克隆<sup>8</sup>**
- **借助 Trident 进行 Snapshot 副本和自助式卷恢复<sup>9</sup>**
- 本期 Tech ONTAP 播客介绍了 NetApp 的 Oracle 专家 Jeff Steiner 如何结合使用 Docker 和 Trident，在 22 秒内克隆了一个大型 Oracle 数据库<sup>10</sup>

### 帮助您及您的企业取得成功

NetApp 已经惯于根据开发人员和工程师的需求提供解决对策。事实上，这正是 NetApp 的传统。作为一家公司，NetApp 很早就知道如何使用存储基础架构支持和加速开发人员及工程师的重要工作，并帮助他们的公司实现更广泛的目标。在 NetApp，为企业构建更好的方式管理数据和使用存储是我们的职责所在。我们将容器等开放式生态系统视为一个朝气蓬勃、有待进一步创新的新领域。这种创新是由社区成员推动的。NetApp 是这个社区的一份子，积极参与

推动其创新。为了实现这一目标，NetApp 正在研究通过多种方法，让社区成员能够更容易地在需要的地方和时间使用存储。如今，NetApp 非常自豪地在各种开放式生态系统之间支持无缝地使用存储。我们将一如既往地支持 Docker、Kubernetes、OpenShift、OpenStack、Ansible、Chef、Puppet 等环境开发一套业界最全面的 API 和集成。

Trident 只是这些努力结果的一部分。推荐您在您企业的容器环境中试用 Trident，并期待听到它为您的 DevOps 管道带来惊人效率和节省的好消息。

### 其他资源

要了解有关 Trident 和任何其他 NetApp DevOps 集成的更多信息，建议您查阅以下资源：

### 有关 NetApp 和 Trident 的更多信息

[NetApp 容器解决方案](#)



[Trident 功能概述：  
Trident 简介](#)



[采用 Trident 的  
Kubernetes 永久性存储简介](#)



[克隆：采用适用于 Kubernetes 的  
Trident 的卷克隆简介](#)



[Trident 文档](#)



[Trident GitHub 下载](#)



### 有关 NetApp DevOps 的更多信息

[NetApp DevOps 解决方案](#)



[thePub \(netapp.io\)](#)



[NetApp Slack 频道  
\(netapp.io/slack\)](#)



[@NetAppPub](#)



## 尾注

- <sup>1</sup> 《使用和部署容器时遇到的难题》，作者：Sarah Conway，2017年4月27日，云原生计算基金会，<https://www.cncf.io/blog/2017/04/27/meeting-challenges-using-deploying-containers/>。依据 [Creative Commons CC-BY 4.0 许可证](https://creativecommons.org/licenses/by/4.0/) 重新出版并添加红色圆圈标记。
- <sup>2</sup> 《通过期望的状态进行 DevOps》，演示者：NetApp 公司的 Andrew Sullivan，Tech Field Day，第 14 天，2017 年 5 月 11 日，<https://www.youtube.com/watch?v=btLZl7M6gnY&list=PLInuRwpnsHacYmunO7zyES6SyrSrfFu5O&index=4>。
- <sup>3</sup> 获取最新 Trident 文档请访问：<https://netapp-trident.readthedocs.io/>。
- <sup>4</sup> “使用 Trident 为 OpenShift 动态配置存储”，在线演示，时长：3 分 20 秒，演示者：The Pub @ NetApp，2017 年 2 月 17 日，<https://www.youtube.com/watch?v=97VZWyssl2E>。
- <sup>5</sup> “管理 Kubernetes 中的永久性数据”，在线演示，时长：23 分 53 秒，演示者：The Pub @ NetApp，2017 年 5 月 15 日，<https://www.youtube.com/watch?v=XIuN91vG2wM>。
- <sup>6</sup> 《DARZ Docker 和容器即服务通过 DevOps 推动数字化转型》，NetApp 客户成功案例，2017 年，<https://www.netapp.com/cn/media/cs-darz-devops.pdf>。
- <sup>7</sup> 《NetApp IT 如何使用 FlexClone 缩短开发周期》，NetApp 社区博客，作者：Gopal Parthasarathy，2015 年 10 月 8 日，<https://community.netapp.com/t5/Technology/How-NetApp-IT-Shortened-Development-Cycles-Using-FlexClone/ba-p/110581>。
- <sup>8</sup> 《Trident 18.01 测试版 1：将卷克隆引入 Kubernetes》，作者：NetApp 公司的 Garrett Mueller，2017 年 12 月 14 日，<https://netapp.io/2017/12/14/trident-18-01-beta-1-introducing-volume-cloning-kubernetes/>。（另请参见：《Trident 18.01 闪亮登场》，作者：NetApp 公司的 Andrew Sullivan，2018 年 1 月 25 日，<https://netapp.io/2018/01/25/trident-18-01/>）。
- <sup>9</sup> 《使用 Trident 和 NFS 自助恢复数据》，作者：NetApp 公司的 Andrew Sullivan，2018 年 4 月 3 日，<https://netapp.io/2018/04/03/self-service-data-recovery-using-trident-nfs/>。
- <sup>10</sup> “第 99 期 — 数据库即服务：容器”，Tech ONTAP 播客，2017 年，[https://soundcloud.com/techontap\\_podcast/episode-99-databases-as-a-service-containers](https://soundcloud.com/techontap_podcast/episode-99-databases-as-a-service-containers)。

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。

有限权利说明：美国政府使用、复制或公开本档受 DFARS 252.277-7103 (1988 年 10 月) 和 FAR 52-227-19 (1987 年 6 月) 中“技术数据和计算机软件权利”条款第 (c)(1)(ii) 条规定的限制条件的约束。

## 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-7270-071-zhCN

**全国销售热线：4008-1818-11**