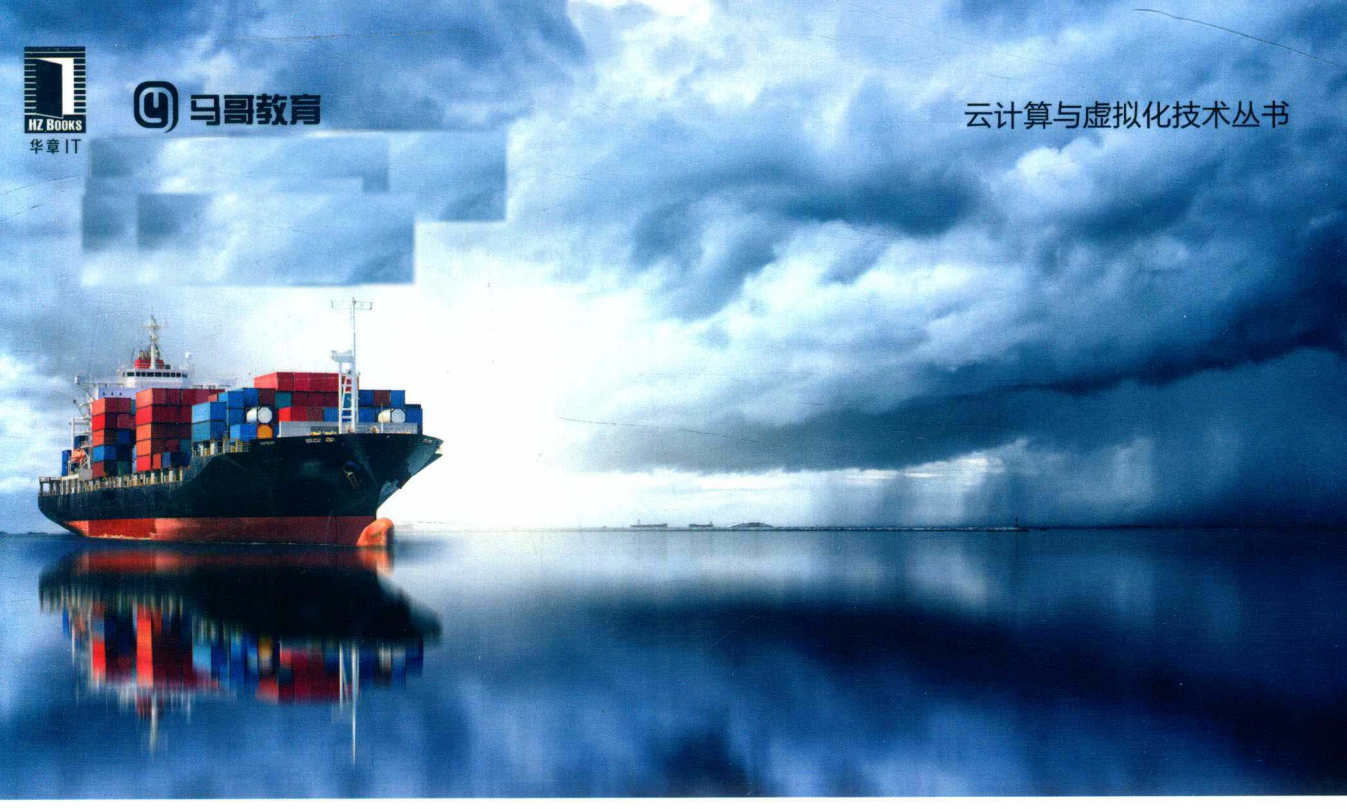


马哥教育

云计算与虚拟化技术丛书

Kubernetes Advanced Practical

Kubernetes 进阶实战

马永亮 著

马哥教育CEO马哥（马永亮）撰写，实用性和专业性毋庸置疑

涵盖Kubernetes架构、部署、核心组件、扩缩容、存储与网络策略、安全、系统扩展等话题

Kubernetes主流知识点全覆盖，渐进式讲解，手把手示范，大量实操案例，随时动手验证

机械工业出版社
China Machine Press

云计算与虚拟化技术丛书

Kubernetes Advanced Practical

Kubernetes 进阶实战

马永亮 著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Kubernetes 进阶实战 / 马永亮著 . —北京：机械工业出版社，2019.1
（云计算与虚拟化技术丛书）

ISBN 978-7-111-61445-6

I. K… II. 马… III. Linux 操作系统 – 程序设计 IV. TP316.85

中国版本图书馆 CIP 数据核字（2018）第 275406 号

Kubernetes 进阶实战

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：张志铭　　责任校对：殷　虹
印　　刷：北京市荣盛彩色印刷有限公司　　版　　次：2019 年 1 月第 1 版第 1 次印刷
开　　本：186mm × 240mm　1/16　　印　　张：28.75
书　　号：ISBN 978-7-111-61445-6　　定　　价：109.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88379426　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzit@hzbook.com

华章IT
HZBOOKS | Information Technology

Preface 前　言

为什么要写这本书

作为置身于IT技术领域多年的实践者和教育者，我们一直盼望着行业迎来这样一个时刻：异构的IT基础设施环境所造成的开发和部署系统应用纷繁复杂的局面终于迎来了终结者，开发人员无须再考虑复杂多样的运行环境下软件程序的移植问题，运维人员不用再手动解决运行环境中组件间的依赖关系等，从而让各自的核心职责都回归到开发和保证系统稳定运行本身。终于，以Docker为首的容器技术为此带来了基础保障，并在容器编排技术的支撑下尘埃落定，甚至连IT管理者心心念念多年的DevOps文化运动也借此找到了易于落地的实现方案。于是，系统运行割据多年的局面终于将走向天下一统。

尽管距Kubernetes 1.0的发布不过三四年的光景，但其如今的影响力在IT技术领域完全算得上空前绝后，目前，一众大小公司都在使用或正筹划使用这一IT技术发展史上可能最为成功的开源项目。Linux软件基金会的常务董事Jim Zemlin在Google Cloud Next 17大会上曾表示，Kubernetes是“云时代的Linux”。的确，Kubernetes应该是开源世界有史以来迭代最快的项目，而且在几乎所有需要采用容器技术的场景里成为占统治地位的解决方案，其发展速度恐怕也仅有Linux内核项目可堪匹敌。2018年3月，Kubernetes成为CNCF旗下“毕业”的第一个项目，并荣获2018年OSCON最具影响力奖项。

目前，Kubernetes保持着每年发布四个重要版本的节奏，版本的每次更新都会引入数个新特性。这种快速迭代的机制在为用户不断带来惊喜的同时，也给他们在学习和使用上造成了一些困扰：相关领域的可参考书籍仍不丰富，互联网上可以得到的众多文档并非源于同一个版本，以及厘清脉络拼凑成完整的知识框架所需的时间成本较大。因此，我在课程以及直接或间接参与生产或测试环境的交付之余便萌生了撰写一本Kubernetes入门、进阶与实战的书籍的想法，将自己学习和使用的经验总结、沉淀并分享给更多有此需求的技术同行，帮助大家快速找到入门路径，降低时间成本，并迅速投入测试和生产之用。

的确，在写作过程中，Kubernetes 这种快速迭代的机制，以及每每引入的新特性，在小惊喜之余带给笔者更多的却是真真切切的梦魇般的恐惧感：在一年多的写作时间里，许多章节几易其稿，却也依然无法确保能够涵盖即将成为核心功能的特性，于是沮丧感几度如影随形，直到自我安慰着“基础的核心特性基本不会发生大的变动，只要能帮助读者弄清楚 Kubernetes 系统的基础架构及核心工作逻辑就算工夫没有白费”之后方才释然。于是便有了这本力图尽量多地包罗 Kubernetes 系统目前主流特性及实践路径的入门和进阶之书、工具之书。

本书特色

本书致力于帮助容器编排技术的初级和中级用户循序渐进地理解与使用 Kubernetes 系统，因此本书的编写充分考虑到初学者进入新知识领域时的茫然，采用由浅入深、提纲挈领、再由点到面的方式讲解每一个知识细节。对于每个知识点，不仅介绍了其概念和用法，还分析了为什么要有这个概念，实现的方式是什么，背后的逻辑为何，等等，使读者不仅能知其然，还能知其所以然。

本书不仅要带领读者入门，更是一本可以随时动手加以验证的实践手册，而且对于部分重要的内容还会专门一步步地给出具体的实操案例，帮助读者在实践中升华对概念的理解。本书几乎涵盖了应用 Kubernetes 系统的所有主流知识点，它甚至可以作为计划考取 CKA 认证的读者的配套参考图书。

读者对象

- ❑ 云计算工程师
- ❑ 运维工程师
- ❑ 系统开发工程师
- ❑ 程序架构师
- ❑ 计划考取 CKA 认证的人员
- ❑ 其他对容器编排感兴趣的人员

如何阅读本书

阅读使用本书之前，读者需要具备 Docker 容器技术的基础使用能力。本书逻辑上共分为五大部分，15 章。

第一部分（第 1 ～ 2 章），介绍 Kubernetes 系统的基础概念及其基本应用。

第 1 章介绍容器编排系统出现的背景，以及 Kubernetes 系统的功能、特性、核心概念、

系统组件及应用模型。

第 2 章讲解 Kubernetes 的核心对象，以及直接使用命令管理资源对象的快速入门技巧。

第二部分（第 3 ～ 6 章），介绍核心资源类型及其应用。

第 3 章介绍资源管理模型、陈述式与声明式资源管理接口，并通过命令对比说明两种操作方式的不同之处。

第 4 章介绍 Pod 资源的常用配置、生命周期、存储状态和就绪状态检测，以及计算资源的需求及限制等。

第 5 章介绍 Pod 控制器资源类型，重点讲解了控制无状态应用的 ReplicaSet、Deployment、DaemonSet 控制器，并介绍了 Job 和 CronJob 控制器。

第 6 章介绍 Service 和 Ingress 资源类型，涵盖 Service 类型、功用及其实现，以及 Ingress 控制器、Ingress 资源的种类及其实现，并通过案例详细说明了 Ingress 资源的具体使用方式。

第三部分（第 7 ～ 9 章），介绍存储卷及 StatefulSet 控制器。

第 7 章主要介绍存储卷类型及常见存储卷的使用方式、PV 和 PVC 出现的原因及应用，以及存储类资源的应用和存储卷的动态供给。

第 8 章介绍使用一等资源类型 ConfigMap 和 Secret 为容器应用提供配置及敏感信息的方式。

第 9 章主要介绍有状态应用的 Pod 控制器资源 StatefulSet，包括基础应用、动态扩缩容及更新机制等。

第四部分（第 10 ～ 11 章），介绍安全相关的话题，主要涉及认证、授权、准入控制、网络模型与网络策略。

第 10 章重点讲解认证方式、Service Account 和 TLS 认证、授权插件类型及 RBAC，并于章节的最后介绍 LimitRanger、ResourceQuota 和 PodSecurityPolicy 三种类型的准入控制器及相关的资源类型。

第 11 章主要介绍网络插件基础及 flannel 的三种后端实现与应用、借助 Canal 插件实现网络策略的方式，以及 Calico 网络插件的基础使用。

第五部分（第 12 ～ 15 章），介绍 Kubernetes 系统的高级话题。

第 12 章介绍 Pod 资源的调度策略及高级调度方式的应用，包括节点亲和、Pod 资源亲和以及基于污点和容忍度的调度。

第 13 章介绍系统资源的扩展方式，包括自定义资源类型、自定义资源对象、自定义 API 及控制器、Master 节点的高可用、基于 Kubernetes 的 PaaS 系统等话题。

第 14 章介绍资源指标、自定义指标、监控系统及 HPA 控制器的应用。

第 15 章介绍简化应用管理的工具 Helm，并基于 Helm 介绍如何为 Kubernetes 系统提供统一的日志收集与管理工具栈 EFK。

有一定 Kubernetes 使用经验的读者可以挑选感兴趣的章节阅读。而对于初学者，建议从基础部分逐章阅读，但构建在 Kubernetes 系统之上的应用多数都要求读者能熟练使用相关领域的知识和技能，如果对某些内容的理解比较困难，那么可能是由于相关知识欠缺，建议读者通过其他资料补充学习相关知识后再阅读本书。编撰本书的主要意图是为初学者提供一个循序渐进的实操手册，不过，任何读者也都可以将它作为一本案头的工具书随时进行查阅。

排版约定

本书中所有的命令都附带了或长格式或短格式的命令提示符，以便读者区分文中正常使用的“#”和“$”，命令提示格式为“~]#”或“~]$”，较长的命令使用了“\”为续行符，且命令及其输出使用了有别于正文的字体。

更多内容和附带代码

本书相关的配置清单等都放置于 https://github.com/ikubernetes/ 的相关仓库中，在实践中需要时可直接克隆至本地实验环境中使用。

勘误和支持

尽管进行授课及技术写作已十数年，但动笔著书尚属首次，考虑到排版印刷后的表述无可更改，整个写作过程战战兢兢、如履薄冰。进行每一个关键话题的表述之前都查阅了大量资料，并且反复斟酌，既期望能够将知识点清晰、准确地加以描述，也试图避免因自己的理解偏差而误导读者。尽管如此，由于笔者水平有限，加之编写时间仓促，书中难免存在不妥之处，恳请读者批评指正。如果读者有更多的宝贵意见，请通过邮箱 mage@magedu.com 联系我，期待能够得到你们的真挚反馈，在技术之路上互勉共进。另外，本书的勘误将会发布在笔者的博客（http://www.ilinux.io）或本书专用的 GitHub 主页（https://github.com/ikubernetes）上，欢迎读者朋友们关注并留言讨论。

参考资料

对于具有不同知识基础和结构的读者来说，仅凭一本书的内容根本不足以获取所需的全部信息，大家还可以通过以下信息获取关于 Kubernetes 系统的更多资料，本书在写作期间也从这些参考资料中获得了很大的帮助。

- Kubernetes Documentation 和 Kubernetes API Reference，这是提供 Kubernetes 领域相关知识最全面、最深入和最准确的参考材料。
- 《Kubernetes in Action》，本书的谋篇布局及写作理念与此书不谋而合，因此笔者在本书中对许多概念的理解和验证也以此书为素材，并且在写作之时有多处概念的描述也借鉴了此书的内容。
- Red Hat Openshift Documentation，Red Hat 公司的产品文档规范、权威、细致且条理清晰，是不可多得的参考材料。
- The New Stack 的技术文章及调研报告是了解 Kubernetes 系统技术细节和行业应用现状及趋势的不可多得的优秀资源。
- Bitnami 及 Heptio 站点上的博客文章提供了深入了解和学习 Kubernetes 系统某个特定技术细节的可靠资料。

另外，本书还大量借鉴了通过搜索引擎所获取到的不少技术文章和参考文档，在这里一并向这些书籍和文章的作者表示深深的谢意！

致谢

感谢 Kubernetes 社区创造性的劳动成果和辛苦付出，我们因此有了学习和使用如此优秀的开源系统的可能性，这也是本书得以编撰的基石。

感谢我的同事们在我写作期间给予的支持和理解，他们的努力让我拥有了得以放心写作的时间和精力。感谢提供了相关行业信息并促使我下定决心开始编写此书的张常安和张士杰先生，本书的写作过程中也得到了他们支援的宝贵素材。

感谢参加了我的课程（马哥教育）的学员朋友们，大家的学习热情及工作中源源不断反馈而来的信息与需求在不同程度上帮助我一直保持着对技术的追求和热忱，教学相长在此得到了充分的体现。

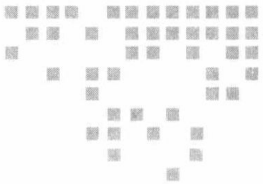
感谢机械工业出版社华章公司高婧雅女士对本书写作的悉心指导，以及对我本人的包容和理解。

最后要特别感谢我的家人，为写作这本书，我牺牲了很多陪伴他们的时间，也正因为他们在生活中的关怀和鼓励才使我能够踏踏实实地完成本书内容的编写。

马永亮

2018 年 10 月

目　录 *Contents*



第 1 章 Chapter 1

Kubernetes 系统基础

近十几年来，IT 领域新技术、新概念层出不穷，例如 DevOps、微服务（Microservice）、容器（Container）、云计算（Cloud Computing）和区块链（Blockchain）等，直有“乱花渐欲迷人眼”之势。另外，出于业务的需要，IT 应用模型也在不断地变革，例如，开发模式从瀑布式（Waterfall）到敏捷（Agile）再到精益（Lean），甚至是与 QA 和 Operations 融合的 DevOps，应用程序架构从单体（monolithic）模型到分层模型再到微服务，部署及打包方式从面向物理机到虚拟机再到容器，应用程序的基础架构从自建机房到托管再到云计算，等等，这些变革使得 IT 技术应用的效率大大提升，同时却以更低的成本交付更高质量的产品。

尤其是以 Docker 为代表的容器技术的出现，终结了 DevOps 中交付和部署环节因环境、配置及程序本身的不同而造成的动辄几种甚至十几种部署配置的困境，将它们统一在容器镜像（image）之上。如今，越来越多的企业或组织开始选择以镜像文件作为交付载体。容器镜像之内直接包含了应用程序及其依赖的系统环境、库、基础程序等，从而能够在容器引擎上直接运行。于是，IT 运维工程师（operator）无须关注开发应用程序的编程语言、环境配置等，甚至连业务逻辑本身也不必过多关注，而只需要掌握容器管理的单一工具链即可。

部署的复杂度虽然降低了，但以容器格式运行的应用程序间的协同却成了一个新的亟待解决的问题，这种需求在微服务架构中表现得尤为明显。结果，以 Kubernetes 为代表的容器编排系统应需而生。

1.1 容器技术概述

容器是一种轻量级、可移植、自包含的软件打包技术，它使得应用程序可以在几乎任