第一本Docker书

THE DOCKER BOOK

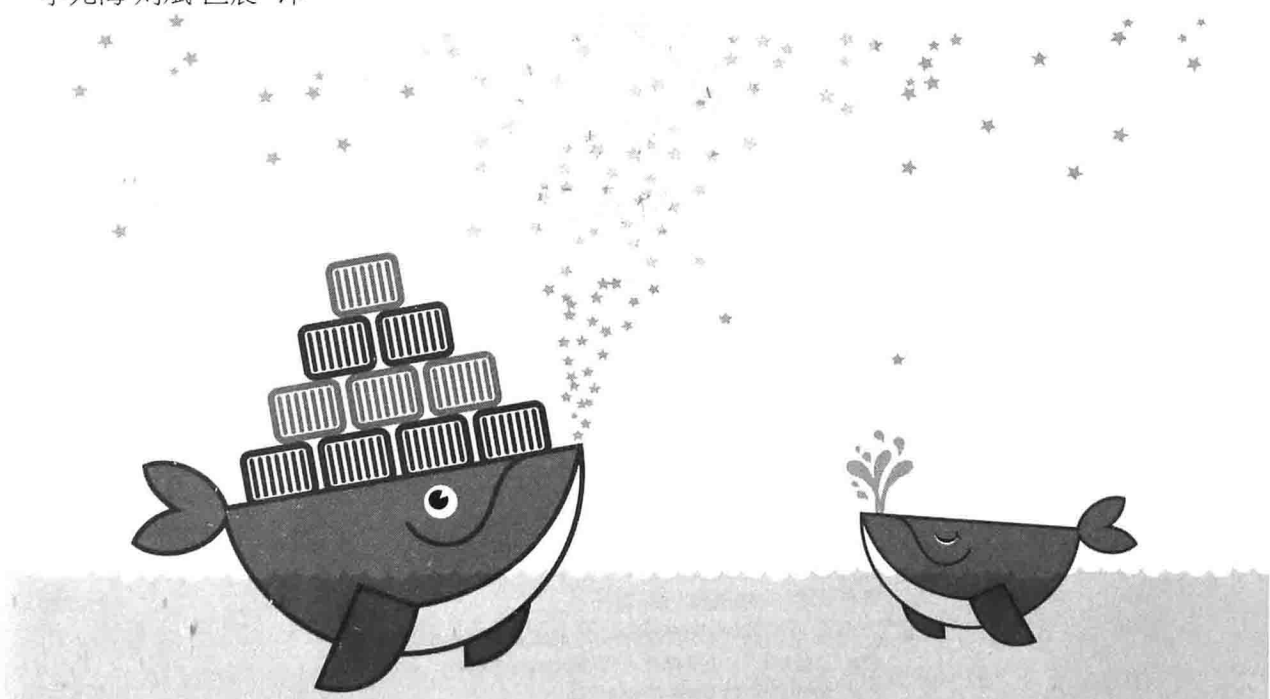
[澳] James Turnbull 著
李兆海 刘斌 巨震 译

人民邮电出版社
POSTS & TELECOM PRESS

第一本Docker书

THE DOCKER BOOK

[澳] James Turnbull 著
李兆海 刘斌 巨震 译



人民邮电出版社
北京

图书在版编目（CIP）数据

第一本Docker书 / （澳）特恩布尔（Turnbull, J.）著 ; 李兆海, 刘斌, 巨震译. -- 北京 : 人民邮电出版社, 2015.1（2015.2重印）
书名原文: The Docker book
ISBN 978-7-115-37733-3

Ⅰ. ①第… Ⅱ. ①特… ②李… ③刘… ④巨… Ⅲ. ①Linux操作系统一程序设计 Ⅳ. ①TP316.89

中国版本图书馆CIP数据核字(2014)第288134号

版 权 声 明

内 容 提 要

Docker 是一个开源的应用容器引擎，开发者可以利用 Docker 打包自己的应用以及依赖包到一个可移植的容器中，然后发布到任何流行的 Linux 机器上，也可以实现虚拟化。

本书由 Docker 公司前服务与支持副总裁 James Turnbull 编写，是权威的 Docker 开发指南。本书会指导读者完成 Docker 的安装、部署、管理和扩展，带领读者经历从测试到生产的整个开发生命周期，让读者了解 Docker 适用于什么场景。书中先介绍 Docker 及其组件的基础知识，然后用 Docker 构建容器和服务来完成各种任务：利用 Docker 为新项目建立测试环境，演示如何使用持续集成的工作流集成 Docker，如何构建应用程序服务和平台，如何使用 Docker 的 API，如何扩展 Docker。

本书适合对 Docker 或容器开发感兴趣的系统管理员、运维人员和开发人员阅读。

◆ 著 [澳] James Turnbull
译 李兆海 刘 斌 巨 震
责任编辑 杨海玲
责任印制 张佳莹 彭志环

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：16
字数：328 千字 2015 年 1 月第 1 版
印数：4001-6500 册 2015 年 2 月北京第 2 次印刷
著作权合同登记号 图字：01-2014-6456 号

定价：59.00 元

读者服务热线：(010)81055410 印装质量热线：(010)81055316
反盗版热线：(010)81055315

对本书的赞誉

Docker 来了！突然发现曾经顶礼膜拜的 Hypervisor 虚拟化已处于被整个颠覆的悬崖边缘，Docker 容器技术的直接虚拟化不仅在技术方面使 CPU 利用率得到显著提升，还因 80:20 法则可在业务上更大程度发挥 CPU 利用率，而恰恰后者才真正体现了虚拟化之精髓！这本书用了大量简短可操作的程序实例介绍 Docker 的工作原理，几乎页页都是满满的代码干货，程序员读者可跟着这些例子自己动手玩转 Docker，这真是一部专为程序员写的好书！IT 行业如此“昨嫌紫莽长，今怜破袄寒”，我们码农们呢？还等什么，赶紧开始知识更新吧，别让你的知识技能这一看家本领也被颠覆了！

——毛文波，道里云 CEO，
曾创建 EMC 中国实验室并担任首席科学家，曾参与创建 HP 中国实验室

Docker 是什么？它有什么用？为什么身边的程序员都在谈论这个新兴的开源项目？

Docker 是轻量级容器管理引擎，它的出现为软件开发和云计算平台之间建立了桥梁。Docker 将成为互联网应用开发领域最重要的平台级技术和标准。这本书由曾任职于 Docker 公司的资深工程师编写，由国内社区以最快的速度完成翻译，是学习 Docker 的最佳入门书籍。如果你是一位希望让自己的代码运行在云端的程序员，现在就开始学习 Docker 吧！

——喻勇，Cloud Foundry 社区创始人

Docker 的核心价值在于，它很有可能改变传统的软件“交付”方式和“运行”方式。传统的交付源码或交付软件包的方式的最大问题在于，软件运行期间所“依赖的环境”是无法控制、不能标准化的，IT 人员常常需要耗费很多精力来解决因为“依赖的环境”而导致软件运行出现的各种问题。而 Docker 将软件与其“依赖的环境”打包在一起，以镜像的方式交付，让软件运行在“标准的环境”中，这非常符合云计算的要求。这种变革一旦为 IT 人员接受，可能会对产业链带来很大的冲击。我们熟悉的 `apt-get` 和 `yum` 是否会逐渐被 `docker pull` 取代？

有了标准化的运行环境，加上对 CPU、内存、网络等动态资源的限制，Docker 构造了一个“轻量虚拟环境”，传统虚拟机的绝大多数使用场景可以被 Docker 取代，这将给 IT 基

础实施带来一次更大的冲击；KVM、ZEN、VMWare 将会何去何从？此外，Docker 秒级创建/删除虚拟机及动态调整资源的能力，也非常契合云计算“实例水平扩展，资源动态调整”的需求，Docker 很有可能成为云计算的基石。

正是因为 Docker 将对传统 IT 技术带来上述两种“革命性”的冲击，所以我们看到围绕 Docker 的创业项目如火如荼。IT 从业人员应该及早拥抱 Docker，拥抱变化。阅读本书就是最佳入门途径。

——陈轶飞，原百度 PaaS 平台负责人，国内最早大规模应用 Docker 的实践者

Docker 今天已经算是明星技术了，各种技术大会都会有人谈论它，越来越多的人像我一样对这门技术着迷。我开始关注 Docker 是因为当时主要关注 PaaS 的一些技术进展，当时 PaaS 受到云计算三层模型的禁锢，诞生很多非常重型的解决方案。我并不喜欢这些方案，不仅是这些方案复杂，而且充满了重复造轮子的自以为是。但是 Docker 的诞生给我眼前一亮的感觉，Docker 诞生时从技术上貌似没有任何亮点：隔离和资源限定都是 LXC 做的，安全是用 GRSec（那时候还没有 SELinux 支持），镜像文件依赖于 AUFS。但是，就是因为 Docker 什么有技术含量的活儿都没做，才显得它非常干净，非常优雅。Docker 提供了一种优雅的方式去使用上述技术，而不是自顾自地去实现这些已有的很好的技术，这样会使构建环境的成本极大降低，从而非常有效地减少运维的复杂性，这不就是 PaaS 的本质吗？

短短一年的时间过后，在中国都有了专门的容器大会，甚至出现了一票难求的情况，可见这一技术不但在国外快速地被社区认同，在国内也得到广泛的应用。我坚信 PaaS 的构建应该是多种组件灵活搭配组合的，而不应该是包罗万象的全套方案，Docker 的设计符合这种原则，这是我最为欣赏的。Docker 的发展异常迅猛，整个社区生态蓬勃向上一片繁荣。希望阅读本书的读者也尽快加入充满乐趣的 Docker 大家庭中来。

——程显峰，MongoDB 中文社区创始人，蓝海讯通 COO

以 Docker 为代表的容器技术是目前非常流行的一类技术，对虚拟化、云计算乃至软件开发流程都有革命性的影响。本书系统而又深入浅出地介绍了与 Docker 部署和应用相关的各个方面，体现了 Docker 的最新进展，并附有大量详尽的实例。无论系统架构师、IT 决策者，还是云端开发人员、系统管理员和运维人员，都能在本书中找到所需的关于 Docker 的内容。本书非常适合作为进入 Docker 领域的第一本书。

——商之狄，微软开放技术（中国）首席项目经理

我很高兴能看到第一本引进国内的 Docker 技术书籍。这本书对于迫切想了解 Docker 技术以及相关工具使用的技术爱好者来说，是一本值得阅读的入门书籍。

——肖德时，InfoQ《深入浅出 Docker》专栏作者

阅读本书，就像参加一个 Docker 专家的面授课程，书中包含了很多非常实用的小型案例，让你能够循序渐进地照着学习，加深理解。好多示例代码都可以拿来直接在开发中使用。James Turnbull 是个写书的高手，章节安排合理紧凑，由浅入深地慢慢引领你理解 Docker 的奥秘。Docker 不仅仅是技术，更是一个生态系统，技术和新项目层出不穷，每一章最后都有介绍本章相关的互联网项目（都可能是下一个 Google），这是最能体现作者技术功力的。无论你是哪个行业的程序员，这本 Docker 的书绝对会让你受益匪浅。

——蔡煜，爱立信软件开发高级专家

相比 OpenStack 这种厂商主导的开源项目，Docker 的社区更加极客，更加活跃，也更具颠覆潜力。对 Docker 本身，已经不用我再多说，只希望大家都看看这本书，并能积极尝试 Docker。纵观 IT 行业历史，大的技术变革从来不是诞生于大厂商口中的金蛋，而是一小搓儿爱好者的小玩意儿，而 Docker 正是这个路子。

——赵鹏，VisualOps 创始人

Go 语言是近年来 IT 技术发展历程中最伟大的事情，而 Docker 的出现则是云计算发展的重要里程碑。作为 Go 语言的杀手级应用，Docker 推动了 Go 语言社区的发展。技术的全球同步化在加速，但非英语母语一定程度制约了中国 IT 技术的发展。这是一本 Docker 团队成员撰写的书，是一份难得的学习 Docker 技术的权威教材。我很高兴见到中文翻译能够如此迅速地跟进，这是一件了不起的事情。通过它，我很期待更多人能够了解 Docker，参与到 Docker 的生态中，共同推进中国 IT 产业的进步。

——许式伟，七牛云存储 CEO，《Go 语言编程》作者

我非常喜欢这本书，它弥补了开源项目通常缺失的文档部分。书中涉及从安装到入门到业务场景下的各种应用及开发。本书作者的权威性以及译者的专业态度也保证了这本书的严谨性。这本书非常适合广大的 Docker 爱好者阅读。

——杜玉杰，OpenStack 基金会董事

DevOps 未死，ContainerOps 已到

发现 Docker 项目还是 2013 年中，我正在为构架一个 Micro Service 的游戏云而测试各种 PaaS 平台和产品。研究 CloudFoundry 的过程中，被 Warden 子项目吸引，转而在 GitHub 中寻找类似的、更容易使用和部署的容器虚拟化解决方案，最终一个 Linux Container 的框架 Docker 成为我的首选。2013 年底在深圳举行的 ECUG Con（实效云时效用户组大会）是我第一次在大型的技术会议上宣讲 Docker 开源技术，此时它已经被 Golang 社区评为 2013 年的十大杀手级应用，也是这次会议我开始了 Docker 技术布道之旅。从 LXC 的框架到 Container 引擎，再到如今的 SaaS 平台，Docker 在开源社区的强大推动下快速向前演进，ContainerOps 平台或是 Docker 的下一个里程碑。

对 Docker 研究得越多，就越容易被它在网络、安全方面的各种问题所困扰，忘却了 Docker 使用 Union FileSystem 技术带来的巨大技术变革的机会。当超越容器虚拟化引擎的标签去看 Docker 时，发现它是实现应用版本管理的最佳技术选择。比起从源代码的某个分支或标签起构建应用的版本，Union FileSystem 更适于实现从开发到运维的版本管理。随着 OverlayFS 被 Linux 内核 3.18 合并到主干，Docker 也会在最新的版本中支持它（也许在你读这本书的时候就已经支持了）。不管是 AUFS 还是 OverlayFS，将摆脱被认为是嵌入式设备的文件格式，成为应用版本管理的技术基石。

在一次技术布道之后，有听众和我交流如何使用 Gnome Desktop 的 Docker 容器为团队提供标准的 Android 开发环境。正值 Docker 在刚刚发布的 1.2 版本中加入了 Device 特性，我建议他可以使用这个特性为 Desktop 加入真机的调试功能。此时我才意识到开发环境甚至是桌面环境是可以通过 Docker 容器来实现统一的。当微软公司和 Docker 深度合作的新闻震惊所有人时，才发现微软早在多年前就布局容器虚拟化的技术。Windows 成为最后一个（FreeBSD 有容器引擎 Jails，Solaris 有容器引擎 Zones）能运行容器的主流操作系统。Windows 操作系统可以通过容器化技术运行多个 Windows 的容器，Docker 引擎也终于有了打通所有平台的机会。不管是 Linux 还是 Windows，开发环境最终都可以被容器管理起来，开发配置管理将会变得非常简单。

当软件的开发环境、版本管理、交付和运行都以 Docker 为工具 Container 为基础进行流转时，就构成了以 Container 为核心的开发和运维流程，软件的构架也因此发生改变（Micro Service 的构架方式可能会因此流行）。但持续集成、持续部署和自动化运维等生产理念没有改变，只是增加了 Container 的解决方案，未来必定会有基于 Docker 的平台来管理整个开发和生产的流程。

DevOps 未死，ContainerOps 已到。

在此感谢三位译者李兆海、刘斌和巨震的辛苦工作，把第一本 Docker 技术书籍带入中国。这不仅是一本 Docker 技术的入门书籍，也介绍了很多 Docker 的最佳实践，是学习 Docker 的绝佳选择。尽管没有参与此书的翻译，甚为遗憾，但我会继续努力在国内推广 Docker 开源技术。

——马全一

Docker 中文社区和 docker.cn 项目创始人，Docker 开源技术布道师，资深构架师

我们走在容器化的大道上

如果你是一位技术爱好者，时刻关心业界最新动态，那么最近一定没少听说 Docker 吧，这绝对是在技术圈线上线下都在谈论的一个热门话题。

说实话，Docker 算不上是什么全新的技术，它基于 LXC（LinuX Containers），使用 AUFS，而这些都是已经存在很长时间并被广泛应用了的技术。但运营 PaaS 服务的 dotCloud 公司将这些技术整合到一起，提供了简单易用的跨平台、可移植的容器解决方案。Docker 最初由 dotCloud 公司在 2013 年发布。自发布以来，其发展速度之快超乎了很多人的想象，一路高歌猛进，2014 年 6 月终于发布了 1.0 稳定版，而 dotCloud 在 2013 年 10 月干脆连公司名字也改为了 Docker, Inc.。

Docker 也可以被称为轻量级虚拟化技术。与传统的 VM 相比，它更轻量，启动速度更快，单台硬件上可以同时跑成百上千个容器，所以非常适合在业务高峰期通过启动大量容器进行横向扩展。现在的云计算可能更多地是在使用类似 EC2 的云主机，以后也许应该更多地关注容器了。

Docker 是可移植（或者说跨平台）的，可以在各种主流 Linux 发布版或者 OS X 以及 Windows 上（需要使用 boot2docker 或者虚拟机）使用。Java 可以做到“一次编译，到处运行”，而 Docker 则可以称为“构建一次，在各平台上运行”（Build once，run anywhere）。

从这一点可以豪不夸张地说，Docker 是革命性的，它重新定义了软件开发、测试、交付和部署的流程。我们交付的东西不再只是代码、配置文件、数据库定义等，而是整个应用程序运行环境：“OS+各种中间件、类库+应用程序代码”。

无论你是开发人员、测试人员还是运维人员，随着对 Docker 越来越深入的了解，你都会爱上它。我们只需要运行几条 `docker run` 就可以配置好开发环境，通过 Dockerfile 或者 Docker Hub 与他人分享我们的镜像，与其他服务集成，进行开发流程的自动化。

- 开发工程师开发、提交代码到代码服务器（GitHub、BitBucket、Gitlab 等）。

- 代码服务器通过 webhook 调用 CI/CD 服务，如 Codeship（没错，就是 2014 年 11 月刚融资 800 万美元的那家初创公司）、Shippable、CircleCI 或者自建 Jenkins 等。
- CI 服务器下载最新代码，构建 Docker 镜像，并进行测试。
- 自动集成测试通过之后，就可以将之前构建的镜像推送到私有 Registry。
- 运维使用新版的 Docker 镜像进行部署。

试想一下这种开发流程是不是很酷？除了工作流程的自动化之外，还能消除线上线下环境不一致导致的问题。以后“在我的机器上运行得好好的……”这种托词应该再也没人信了吧。

Docker 是为 Infrastructure as code 而生的，通过 Dockerfile，镜像创建过程变得自动且可重复，还能进行版本管理。

Docker 是为不可变基础设施（Immutable Infrastructure）而生的，对无状态服务的升级、部署将会更轻便更简单：我们无需再对它们的配置进行修改，只需要销毁这个服务并重建一个就好了。

Docker 也是为云计算而生的，Docker 的出现离不开云计算的兴起，反过来更多的云计算服务提供商也都开始把 Docker 纳入自己的服务体系之中，比如最近一个大事件就是 Google 刚刚发布了 Google Container Engine（alpha）服务，一个基于其开源 Docker 编配工具 Kubernetes 的“Cluster-as-a-Service”。容器技术在云计算时代的重要程度由此可见一斑。

这是一本带领读者进入 Docker 世界的入门书。阅读本书除了能帮助读者理解 Docker 的基本原理，熟练掌握 Docker 的各种常见的基本操作之外，还能帮助读者了解 Docker 的实际应用场景以及如何利用 Docker 进行开发等话题，比如，如何使用 Docker 和 Jenkins 进行测试，如何对应用程序进行 Docker 化，以及如何构建由 Node.js 和 Redis 组成的多容器应用栈。当然，书中也不会忘了最近比较火的 Fig——一个 Docker 编配工具，开发此工具的公司是位于英国伦敦的 Orchard Laboratories，前段时间该公司刚刚被 Docker 收购，继续 Fig 的开发。现代应用程序都离不开 API，Docker 当然也不例外。在第 8 章中，读者将学到如何使用 API 而不是 Docker 命令来对 Docker 镜像和容器进行管理。如果你也想为 Docker 贡献自己的力量，那么一定不能错过第 9 章的内容，这一章将会主要介绍如何给 Docker 提 issue，如何完成 Docker 文档，以及如何构建 Docker 开发环境和提交 Pull Request。

最后，我谨代表合译者李兆海和巨震，向在本书翻译过程中给与了很大帮助的一些人表

示最诚挚的感谢。Fiona（冯钊）在本书编写过程中做了很多沟通和协调工作，也对很多术语翻译提出了建议。马全一是 Docker 中文社区和 https://docker.cn 的创始人，也是本书中文版翻译工作的发起人。此外还有人民邮电出版社的杨海玲等编辑老师，没有她们的认真校对和排版工作，这本书也不会以完美的形式展现在各位读者面前。

——刘斌

前言

本书面向的读者

本书适合希望实施 Docker 或基于容器的虚拟化技术的开发者、系统管理员和有意 DevOps 的人员阅读。

要阅读本书，读者需要具备一定的 Linux/Unix 技能，并熟悉命令行、文件编辑、软件包安装、服务管理和基本的网络知识。

注意

本书基于 1.0.0 或更高版本的 Docker，该版本不向下兼容。而且，在生产环境中，我们也推荐使用 1.0.0 或更高版本。

致谢

- 感谢我的合伙人及好友 Ruth Brown，感谢你迁就我进行本书的写作。
- 感谢 Docker 公司的团队，感谢你们开发出 Docker，并在本书写作期间提供无私的帮助。
- 感谢#docker 频道和 Docker 邮件列表里的朋友们，感谢你们在我遇到问题时为我答疑解惑。
- 感谢 Royce Gilbert，感谢你不仅提供超赞的技术插图，还为本书英文版设计了封面。
- 感谢 Abhinav Ajgaonkar，感谢你提供自己的 Node.js 和 Express 示例应用程序。
- 感谢本书的技术审校团队，你们让我时刻保持头脑清醒，并指出了书中的愚蠢错误。

本书中有 3 张配图是由 Docker 公司提供的。

技术审稿人团队

Scott Collier

Scott Collier 是一位高级主任系统工程师，就职于 Red Hat 的系统设计及工程团队。该团队根据从销售、市场以及工程团队收集到的数据，甄别并提供高价值的解决方案，并为内外部用户开发参考架构。Scott 是 Red Hat 认证构架师（RHCA），具有超过 15 年的 IT 从业经验，他现在专注于 Docker、OpenShift 以及 Red Hat 系列产品。

除思考分布式构架之外，Scott 喜欢跑步、登山、露营，还喜欢陪妻子和三个孩子在得克萨斯州的奥斯汀享受烧烤。他的技术文章以及相关信息都发表在他的个人网站（http://colliernotes.com）上。

John Ferlito

John 是一位连续创业者，同时也是高可用性、可扩展性基础设备专家。John 现在在自己创建的 Bulletproof 公司担任 CTO，这是一家提供关键任务的云服务商，同时，John 还兼任提供综合视频服务的 Vquence 公司的 CTO。

在空闲时间，John 投身自由及开源软件（Free and Open Source Soft，FOSS）社区。他是 linux.conf.au 2007 会议的联合发起人，也是 2007 年悉尼 Linux 用户委员会（Sydney Linux User Group，SLUG）的委员。他做过大量的开源项目，如 Debian、Ubuntu、Puppet 以及 Annodex 套件。读者可以在他的个人博客（http://inodes.org/blog）上查看他的文章。John 拥有新南威尔士大学的工程学士荣誉学位（计算机科学类）。

Paul Nasrat

Paul Nasrat 就职于 Google 公司，是一位网站可靠性工程师，同时也是 Docker 的贡献者。他在系统工程领域做了大量的开源工具，包括启动加载器、包管理以及配置管理等。

Paul 做过各种系统管理和软件开发的工作。他曾在 Red Hat 担任软件工程师，还在 ThoughtWorks 公司担任过基础设备专家级顾问。Paul 在各种大会上做过演讲，既有 DevOps 活动早期在 2009 年敏捷大会上关于敏捷基础设备的演讲，也有在小型聚会和会议上的演讲。

技术插图作家

Royce Gilbert（ksuroyce@yahoo.com）是本书技术插图的作者，在他超过 30 年的从业经验中，他做过 CAD 设计、计算机支持、网络技术、项目管理，还曾为多家世界 500 强企业提供商务系统分析，包括安然（Enron）、康柏（Compaq）、科氏（Koch）和阿莫科（Amoco）集团。Royce 在位于堪萨斯州曼哈顿的堪萨斯州立大学担任系统/业务分析员。他业余时间在自己的 Royce 艺术工作室进行创作，是一位独立艺术家和技术插画家。他和 38 岁的妻子在堪萨斯州的弗林特山上修复了一间 127 年历史的石头老屋，并以此为居所，过着平静的生活。

校对者

Q 女士在纽约地区长大，是一位高中教师、纸杯蛋糕冷冻师、业余科学家、法医人类学家，还是一名灾难应急专家。她现居旧金山，制作音乐，研究表演，整理 ng-newsletter[①]，并负责照顾 Stripe 公司的名流。

排版约定

这是行内代码语句：`inline code statement`。

下面是代码块：

代码清单 0-1　示例代码块

```
This is a code block
```

过长的代码行会换行。

代码及示例

读者可以在网站上获取本书的代码和示例程序[②]，也可以获取代码和示例的 Git 库[③]。

① http://www.ng-newsletter.com/
② http://www.dockerbook.com/code/index.html
③ https://github.com/jamtur01/dockerbook-code

说明

本书英文原版是用 Markdown 格式写的，同时也使用了大量的 LaTeX 格式的标记符号，然后用 PanDoc 转成 PDF 和其他格式（还使用了 Backbone.js on Rails [①]那帮好兄弟写的脚本）。

勘误

如果读者发现任何错误，请用电子邮件与我联系，我的邮箱是 james+errata@lovedthanlost.net。

版本

本书是《The Docker Book》一书 v1.3.1 版的中文版。

① https://learn.thoughtbot.com/products/1-backbone-js-on-rails

目录