

Broadview®  
www.broadview.com.cn

MANNING

# Docker

## 实战

[美] Jeff Nickoloff 著  
胡震 杨润青 黄帅 译

Docker in Action

Ahmet Alp Balkan  
为本书作序



中国工信出版集团

电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
http://www.phei.com.cn

# Docker 实战



## Docker in Action

[美] Jeff Nickoloff 著  
胡震 杨润青 黄帅 译

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

自从 2013 年 3 月 Docker 0.1 版本发布以来,以其为代表的容器技术也走上了快速发展之路, Docker 容器在很大程度上改变了软件的架构设计、开发和运维部署方式,也给早些年就提出微服务的架构模式插上了快速起飞的翅膀。本书由 Docker 社区第一人 Jeff Nickoloff 编写,共分为 3 部分,第 1 部分(第 1~6 章)重点介绍了 Docker 容器的资源隔离和权限控制及基础原理,第 2 部分(第 7~10 章)详细解释了如何打包构建镜像以及各种镜像分发基础设施的建设,第 3 部分(第 11~12 章)聚焦于 Docker 容器的组合操作,也就是多容器和多主机环境的管理。本书图文并茂,结合基本原理和具体案例给大家提供了多个不错的实战机会。

作为目前热门的容器技术类图书,本书适用于互联网,云计算,企业级软件开发、架构、测试,以及运维人员快速上手的 Docker 容器;同样适用于搭建以 Docker 为核心的基础设施,并在生产环境中快速部署应用以及管理容器集群。

Original English Language edition published by Manning Publications, USA. Copyright © 2016 by Manning Publications. Simplified Chinese-language edition copyright © 2017 by Publishing House of Electronics Industry. All rights reserved.

本书简体中文版专有出版权由 Manning Publications 授予电子工业出版社。未经许可,不得以任何方式复制或抄袭本书的任何部分。专有出版权受法律保护。

版权贸易合同登记号 图字:01-2016-2997

### 图书在版编目(CIP)数据

Docker 实战 / (美)杰夫·尼克洛夫(Jeff Nickoloff)著;胡震,杨润青,黄帅译. —北京:电子工业出版社,2017.1

书名原文: Docker in Action

ISBN 978-7-121-30306-7

I. ①D… II. ①杰… ②胡… ③杨… ④黄… III. ①Linux 操作系统—程序设计 IV. ①TP316.85

中国版本图书馆 CIP 数据核字(2016)第 270351 号

策划编辑:张春雨

责任编辑:徐津平

印 刷:三河市鑫金马印装有限公司

装 订:三河市鑫金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编:100036

开 本:787×980 1/16 印张:18.25 字数:352 千字

版 次:2017 年 1 月第 1 版

印 次:2017 年 1 月第 1 次印刷

定 价:79.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888, 88258888。

质量投诉请发邮件至 [zlt@phei.com.cn](mailto:zlt@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

本书咨询联系方式:(010) 51260888-819, [faq@phei.com.cn](mailto:faq@phei.com.cn)。

# 译 序

---

2016年4月中旬 DockOne.io 社区的发起人李颖杰找到了杨润青、黄帅和我三位社区译者，说要配合电子工业出版社翻译 Docker 大神 Jeff Nickoloff 的新书——《Docker 实战》，我们仨欣然答应。这一方面是出于对大神的景仰，另一方面也是我们 DockOne.io 社区以及我们每一位会员想为国内 Docker 容器技术的发展添砖加瓦。紧张的翻译工作从今年4月底开始，到6月底交稿，之后我们又配合出版社完成了本书的审读工作，今天我们终于看到本书中文版《Docker 实战》的出版。此时此刻的心情我们也是非常激动，感谢我们 DockOne.io 社区以及出版社的小伙伴，大家都是好样的。

本书的内容，顾名思义，就是为广大技术人员如何创建、部署和管理 Docker 容器以及基于 Docker 的应用程序进行理论及实战练习准备的。对于 Docker 容器的工作原理以及每一个具体功能都面面俱到，特别地，对于如何在生产环境中部署管理 Docker 容器和集群，本书也花了很多的笔墨在探讨。

以 Docker 为代表的容器适合运行所有的无桌面的服务型应用，从2013年3月 Docker 0.1 版本发布以来，在短短的四年时间里，Docker 已经风靡全球的互联网技术圈。回顾过去的四年，Docker 容器技术可以说是每年都上一个新台阶，2013年的小试牛刀，2014年年中的 Docker 1.0 版本的发布以及年底 Docker Machine、Swarm 和 Compose 三剑客的推出，2015年 Docker 开始进入互联网公司的生产环境部署，以及2016年开始发力进入企业级应用，CaaS 市场也日渐成熟，与此相呼应的就是整个容器生态圈包括 K8S、Mesos 等集群调度系统在内的蓬勃发展以及微服务架构模式的落地开花。特别地，微服务可以说与 Docker 容器是绝配，目前国内有不少互联网公司包括一些创业公司正在打算或者已经实施基于 Docker 容器部署实现各自的微服务架构模式了。

本书在翻译出版阶段，适逢 Docker 1.12 版本的发布，该版本的内置 Swam 编排机制可以说是划时代性的新特性。由于时间的关系，该部分最新的内容还无法及时写入到本书中，

不过没有关系，大家在阅读本书时，也可以看到作者对于容器集群编排系统是如何做出一些前瞻性的预判的介绍。相信大家通过阅读本书，都会受益匪浅！

胡震

上海凡用信息科技有限公司 CTO

# 序

---

我第一次听说 Docker，是通过从 2013 年 PyCon 会议的 YouTube 视频，它被第一时间发布到了 Hacker News。Docker 的创始人 Solomon Hykes 在他题为“Linux 容器未来”的五分钟谈话中，揭开了未来我们如何对外交付和运行软件，不仅仅是在 Linux 中，而是在几乎所有的平台和架构之上。虽然他在第五分钟突然语塞，但毫不影响我清晰地认识这项技术在沙箱环境中运行 Linux 应用程序，配以用户友好的命令行工具和镜像分层的独特概念，都将改变很多东西。

Docker 大大改变了多数软件的开发和运维模式。架构设计、开发，以及运维方式，在 Docker 出现前后有很大的不同。尽管 Docker 没有规定具体的方案，但它迫使人们思考微服务和不可变基础设施的方法。

一旦 Docker 被更广泛地采用，人们便开始调查 Docker 所使用的低层技术，更明确了 Docker 的成功并不是技术本身，而是围绕这个项目的人性化接口、API 和生态系统。

许多大公司如谷歌、微软和 IBM 都在 Docker 项目聚首，并携手合作，使它变得更好，而不是去构建另一个竞争对手。事实上，像微软、Joyent、英特尔和 VMware 等公司，虽在 Linux 容器实现中都换掉了 Docker，却为他们自己的容器产品保留了 Docker 的命令行接口。在短短两年时间里，很多新公司都如雨后春笋般成立，为的是增强开发者的体验和填补 Docker 生态系统的空白，形成一个健康且热心的 Docker 社区。

就我个人而言，我帮助微软发布了其第一款支持跨平台 ASP.NET 的正式 Docker 镜像。我的下一个贡献，会将 Docker 命令行接口移植到 Windows。这个项目帮助了许多 Windows 开发人员熟悉 Docker，奠定了微软对 Docker 项目进行长期贡献的基础。Windows 移植项目也让我以少数 Docker 贡献者的视角工作了两个多月。后来，我们还有许多其他的贡献，以确保 Docker 成为微软 Azure 云服务的重要工具。我们下一个大的目标就是 Windows 容器，以及将 Windows Server 2016 的新功能与 Docker 完全集成在一起。

令人兴奋的是，我们还在容器革命的初期。一切快得令人难以置信，随着每天出现的

新技术和开源工具，如今，我们认为这一切将理所当然地会在未来数月内继续发生改变。就在这个领域，创新者和我们行业最伟大的头脑正在合作，为软件行业的其他领域，打造大量的创新工具，使发布和运行软件不再受到规模变化的影响。

通过许多发表在网上一有关 Docker 和微服务的文章，Jeff Nickoloff 将自己定位为新兴的 Docker 社区的第一人。他良好的写作功底加上对技术性话题的详细解答，能帮助开发人员快速学习和使用 Docker 生态系统，并从中受益。而且，同样重要的是，他还指出了 Docker 的缺点。本书从零开始，展示了在生产环境中 Docker 的部署，描述了 Docker 拥有的复杂功能，并对相同任务的多种实现方案进行了比较。

在阅读本书时，你不仅要学会如何有效地使用 Docker，也一定要弄明白它的工作原理，Docker 每一个详细功能如何使用，以及在生产环境中使用 Docker 的最佳方法。我个人在读本书时，有很多“哦，这个功能是做这个的呀”的时刻。写这本书，特别是针对一项总是以惊人速度在变革的技术，就像是在一辆行驶速度为 60 英里/小时的车子上画画一样困难。Jeff 的出色工作，既覆盖了 Docker 的前沿功能，又为本书奠定了一个坚实的理念基础，即无论 Docker 在未来的几个月甚至几年中有什么样的变化，本书关于容器和微服务的理念，不会改变。

希望你和我一样觉得能从这本书受益匪浅。

Ahmet Alp Balkan  
微软开放源码软件工程师  
Docker 贡献者

# 前 言

---

2011 年，我开始在 Amazon.com 工作。第一周我的生活就被改变了，因为我学会了如何使用他们的内部版本，对组件依赖建模和制作部署工具。这种自动化的管理方法，我一直都知道是可能的，但从来没有见过。我过去的团队，每一季度部署一次，需要 10 小时完成。在亚马逊公司，我看着滚动部署工具，将当天早些时候我做出的改动，推送到数百台遍布全球的计算机上。和其他公司的前景相比，这就是大科技公司的工程优势。

早在 2013 年，我想使用 Graphite（监控数据的收集和绘图套件）工作。有一天，我要安装该软件，并开始整合个人项目。应付这些事情，我有几年开源工具的使用经验，但很少依赖于 Python 这么大的生态系统。安装说明很长且模糊，在接下来的几个小时里，我发现许多安装步骤没有文档说明。这对一个有较为深入的 Python 生态系统知识的人来说，可能显而易见。但对我来说，在尝试了几个安装指南，阅读了相关的配置文件，并和代码库的依赖关系进行了史诗般地战斗后，我认输了。

那是我一生中最沮丧的几个小时。我不想和这个项目再有任何关系。更糟糕的是，因为它，我改变了当前的系统环境，经常使用的那些软件现在变得无法兼容。而要还原这些变化，则需要很长一段时间，让我尴尬不已。

我清楚地记得那年 5 月 1 日，我在办公室，当我决定看看 Hacker News 上面有没有新的方法来提高我的能力时，并于那个被称为 Docker 的技术文章，一整个星期上了几次的头版。那天晚上，我决定去试试。我找到了那个网站，在几分钟内安装了软件。我在计算机桌面上运行着 Ubuntu，而 Docker 只有两个依赖：LXC 和 Linux 内核本身。

像其他人一样，我尝试了“Hello, World”的示例，但什么都不会。接下来，我安装了 Memcached，一分钟内下载并运行。然后我启用了 WordPress，它带着捆绑好的 MySQL 数据库。我还拉了几个不同的 Java、Python 镜像。此时我回想起安装 Graphite 那可怕的一天。我打开 Docker Index（在 Docker Hub 之前），并做了一个快速搜索。



结果出来了，它就在那里——某些用户做好了 Graphite 镜像。我抓取下来，并创建了一个新容器，一个简单且配置好的 Graphite 服务器就在我的机器上运行了。我只用了不到一分钟的下载时间就搞定了它，而几个月前同样的安装花费了几个小时，结果还是失败。Docker 能够用最简单的示例，以及最小的努力来展现它的价值，我臣服了。

接下来的一周，我通过和好朋友之间关于 Docker 和容器的直接对话，测试了他的耐心。我解释了包管理的好处，以及文件系统默认强制隔离可解决的一些管理上的问题；我大谈资源利用效率和初始化延迟问题。我向其他几个同事反复地讲摸索着使用容器的故事。每个人听后都有类似的问题，“哦，这就像虚拟化吗？”“如果我有虚拟机，为什么我还需要这个？”问的问题越多，我就想了解得更多。随着项目的普及，我将这个故事分享给了许许多多的人。

当我有机会公开发言时，我开始谈论有关 Docker 的内容。在 2013 年和 2014 年，只有少数人听说过 Docker，甚至更少的人在实际中试过该软件。大多数情况下，总有一些怀疑的系统管理员和兴奋的开发人员，而且他们数量相当。人们的反应都大不相同。有些是纯粹的反对者，显然倾向于维持现状。看得出来，他们经历过某些困难，曾经遇到过很类似的日常问题。而另一些人的兴奋反应和我很相似。

2014 年的夏天，Manning 的副发行人打电话给我，让我谈谈 Docker。在手机上谈了一个多小时后，他问我是否有足够的内容构成一本书。我的看法是这些内容足够几本书。他问我，是否有兴趣写写，这让我变得更加兴奋，虽然那时我用 Docker 已经有一段时间了。那年秋天，我离开 Amazon.com，然后开始了《Docker 实战》这本书的创作。

今天，我坐在完成的手稿面前。写这本书的目的是让更多的人能尽快学会使用 Docker，在本书中，他们也将了解基本的机制和原理。我希望读者有了这些知识后，可以理解 Docker 是如何被应用到实际问题中，以及如何更好地应用在自己的实例中。

# 感 谢

---

与写书不同，过去我的人生中大部分时间在处理简单的事情。在开始写作本书之前我就知道，写作需要高度的纪律性和积极性，但我没有让自己失望。

首先，我想感谢 Manning 出版社给我这个机会来完成这项工作、Ahmet Alp Baken 写了序，以及 Nick Palm 给整个书稿做了技术校对。许多人审阅书稿，并在成书的各个阶段提供了很多意见，包括 Robert Wenner, Jean-Pol Landrain, John Guthrie, Benoît Benedetti, Thomas Peclak, Jeremy Gailor, Fernando Fraga Rodrigues, Gregor Zurowski, Peter Sellars, Mike Shepard, Peter Krey, Fernando Kobayashi, 和 Edward Kuns。

在这个最困难的时刻，成功依赖于一个集体的贡献和支持。如果没有他们的贡献，我不会站在这里。

- Portia Dean，她在过去一年给了我合作和支持。Portia，你是我的伙伴，我的正义和坚持的核心。没有你，我将在某处迷失一年。我爱冒险，对下一次会发生什么总是缺乏等待的耐心。
- 我的父母，Kathy 以及 Jeff Nickoloff，从年轻时就支持我的技术好奇心，培养我坚强的意志。
- Neil Fritz，超过 15 年和我一起做项目，为了得到切片比萨，我们彼此心灵相通。
- Andy Will 以及 PHX2 的优秀工程师们，他们欢迎我到亚马逊，并始终在提高我们的技术门槛，与他们合作本身就是一种学习。
- Nick Ciubotariu，打了漂亮的一仗，提高了技术领导力的门槛。
- 卡特尔咖啡工作室，今年我在你们的总部花的时间要比我在家多很多。你那有世界上最好的烤肉之一。旧金山人都会想念它。

最后，我要感谢全世界志同道合的朋友，他们在这段旅程与我共同学习、分享、挑战或倾听。

# 关于本书

---

《Docker 实战》的宗旨是向开发人员、系统管理员和混合技能的其他计算机用户，介绍 Docker 项目和 Linux 容器的概念。Docker 和 Linux 都是开源的项目，有丰富的在线文档，但无论如何，入门仍是一项艰巨的任务。

Docker 是有史以来增长最快的开源项目之一，在其周围的生态系统也是以类似的速度不断发展。由于这些原因，本书的重点完全在于 Docker 的工具集。这一范围限定不仅能使内容选材更精准，帮助读者了解在他们的特有实例中如何应用 Docker 功能；一旦读者们牢牢把握住了本书涉及的基础知识，他们也能应对更大的问题，进而探索整个生态系统。

## 路线图

本书被分成 3 个部分。

第 1 部分介绍 Docker 和容器的特点。帮助你理解如何安装和卸载 Docker 中发布的软件。你将学习如何运行、管理，并在不同的容器结构连接不同类型的软件。第一部分介绍每一个 Docker 用户需要的基本技能。

第 2 部分介绍 Docker 的封装和软件的分发，涵盖了不同大小 Docker 镜像的底层机制，以及对不同的封装和分发方法所进行的调查。这一部分还包括对 Docker Distribution 项目的深入分析。

第 3 部分介绍多容器项目和多主机环境，覆盖了 Docker Compose、Machine 和 Swarm 项目。这部分内容会指导构建和部署多个真实的实例，规模接近于大型的服务器软件。

## 代码约定和下载

本书是关于一个多用途的工具，所以很少有“代码”列入书中。取而代之的是数以百

计的 shell 命令和配置文件。它们通常来用 POSIX 兼容的语法。针对 Docker 提供的一些特定于 Windows 的功能，用户需要注意那些为了提高可读性或澄清注解而分行的命令。代码托管在 GitHub 上（<https://github.com/dockerinaction>），引用的镜像库可在 Docker Hub（<https://hub.docker.com/u/dockerinaction/>）中找到。运行这些示例并不需要 Docker Hub 或 GitHub 的经验。

本书使用了几个开源项目，既展示了 Docker 的各种功能，又帮助读者转变了软件管理的范式。没有一个单独的软件“堆栈”或系列比 Docker 本身更突出。通过这些实例，读者将会使用如 WordPress、Elasticsearch、Postgres、shell 脚本、Netcat、Flask、JavaScript、NGINX 和 Java 等工具。唯一的依赖就是 Linux 内核。

## 关于作者

Jeff Nickoloff 会建立大规模的服务站，写关于技术的文字，并帮助人们实现他们的产品目标。他曾在 Amazon.com、Limelight 网络和亚利桑那州立大学做这些事情。2014 年离开亚马逊，他创办了一家咨询公司，专注于为财富 100 强的企业和创业公司提供工具、培训和最佳实践。如果你想与他聊天或者一起工作，可以在 <http://allingeek.com> 找到他，或者在 Twitter 上找 @allingeek。

## 关于封面插图

---

《Docker 实战》的封面，标题为“渔父”的插图，是从 19 世纪的许多艺术家珍藏中挑选出来的，由 Louis Curmer 编辑，于 1841 年在巴黎出版。藏品的标题意为他们自己所画的法国人。每个插图，都是手绘，藏品丰富的变化生动地勾勒出 200 年前在世界各地区、城镇、村庄和邻里文化有多么不同。由于相互隔离，说着不同的方言或语言，在街道上或农村，从人们的着装中很容易辨认他们住在哪里，以及交易着什么，生活的状态如何。

自那时以来，着装规范改变了，那时丰富的地区多样性，也已经消失了。现在很难分辨来自不同大陆的居民，更别说不同的城市或地区。也许我们已经为更多样化的个人生活，当然包括更多样化和快节奏的技术化生活，牺牲了文化的多样性。

而此时也很难区分计算机类的书籍，Manning 出版社用该书封面来纪念计算机业务的开创性和创新性，以及 200 年前丰富的地域和多样性的生活——就像这个藏品的那些图片，把我们带回到了生活本身。

# 目 录

---

## 第 1 部分 保持一台整洁的机器

第 1 章 欢迎来到 Docker 世界 .....	2
1.1 什么是 Docker .....	3
1.1.1 容器 .....	3
1.1.2 容器不是虚拟化 .....	4
1.1.3 在隔离的容器中运行软件 .....	4
1.1.4 分发容器 .....	6
1.2 Docker 解决了什么问题 .....	6
1.2.1 组织有序 .....	7
1.2.2 提高可移植性 .....	8
1.2.3 保护你的机器 .....	9
1.3 为什么 Docker 如此重要 .....	10
1.4 何时何处使用 Docker .....	11
1.5 案例：“Hello World” .....	11
1.6 小结 .....	13
第 2 章 在容器中运行软件 .....	14
2.1 从 Docker 命令行工具获得帮助 .....	14
2.2 控制容器：建立一个网站的监控器 .....	15
2.2.1 创建和启动一个新的容器 .....	16
2.2.2 运行交互式容器 .....	17

2.2.3 列举、停止、重新启动和查看容器输出	18
2.3 已解决的问题和 PID 命名空间	20
2.4 消除元数据冲突：构建一个网站农场	23
2.4.1 灵活的容器标识	24
2.4.2 容器的状态和依赖	26
2.5 构建与环境无关的系统	28
2.5.1 只读文件系统	29
2.5.2 环境变量的注入	31
2.6 建立持久化的容器	34
2.6.1 自动重启容器	35
2.6.2 使用 init 和 supervisor 进程维持容器的运行状态	36
2.7 清理	38
2.8 小结	39
<b>第 3 章 软件安装的简化</b>	<b>40</b>
3.1 选择所需的软件	41
3.1.1 什么是仓库	41
3.1.2 使用标签	42
3.2 查找和安装软件	43
3.2.1 命令行使用 Docker Hub	43
3.2.2 通过网站访问 Docker Hub	45
3.2.3 使用替代注册服务器	47
3.2.4 镜像文件	47
3.2.5 从 Dockerfile 安装	49
3.3 安装文件和隔离	49
3.3.1 镜像层实战	50
3.3.2 分层关系	51
3.3.3 容器文件系统抽象和隔离	52
3.3.4 分层文件系统及其工具的优点	53
3.3.5 Union 文件系统的不足	53
3.4 小结	54

---

第 4 章 持久化存储和卷间状态共享	55
4.1 存储卷的简介	56
4.1.1 存储卷提供容器无关的数据管理方式	56
4.1.2 NoSQL 数据库使用存储卷	57
4.2 存储卷的类型	60
4.2.1 绑定挂载卷	60
4.2.2 Docker 管理卷	63
4.3 共享存储卷	65
4.3.1 主机依赖的共享	65
4.3.2 共享和 volumes-from 标志	66
4.4 管理卷的生命周期	68
4.4.1 管理卷的权限	68
4.4.2 存储卷的清理	69
4.5 存储卷的高级容器模式	70
4.5.1 卷容器模式	70
4.5.2 数据打包的存储卷容器	72
4.5.3 多态容器模式	73
4.6 小结	74
第 5 章 网络访问	75
5.1 网络相关的背景知识	76
5.1.1 基础：协议，接口和端口	76
5.1.2 高级：网络，NAT 和端口转发	77
5.2 Docker 的网络	79
5.2.1 本地 Docker 网络的拓扑结构	79
5.2.2 四种网络容器原型	80
5.3 Closed 容器	81
5.4 Bridged 容器	83
5.4.1 访问外部网络	84
5.4.2 自定义命名解析	85
5.4.3 开放对容器的访问	88



5.4.4	跨容器通信	91
5.4.5	修改网桥接口的配置	92
5.5	Joined 容器	93
5.6	Open 容器	95
5.7	跨容器依赖	96
5.7.1	链接——本地服务发现	97
5.7.2	链接别名	98
5.7.3	环境变量的改动	99
5.7.4	链接的本质和缺点	101
5.8	小结	102
<b>第 6 章</b>	<b>隔离——限制危险</b>	<b>103</b>
6.1	资源分配	104
6.1.1	内存限制	104
6.1.2	CPU	105
6.1.3	设备的访问权	108
6.2	共享内存	108
6.2.1	跨容器的进程间通信	109
6.2.2	开放内存容器	110
6.3	理解用户	111
6.3.1	Linux 用户命令空间	111
6.3.2	run-as 用户	111
6.3.3	用户和卷	114
6.4	能力——操作系统功能的授权	116
6.5	运行特权容器	117
6.6	使用加强工具创建更健壮的容器	118
6.6.1	指定额外的安全选项	119
6.6.2	微调 LXC	120
6.7	因地制宜地构建容器	121
6.7.1	应用	121
6.7.2	高层的系统服务	122