

Docker Containers Build and Deploy with Kubernetes,
Flannel, Cockpit, and Atomic

Docker 容器

利用Kubernetes、Flannel、Cockpit和
Atomic构建和部署

[美] Christopher Negus 著 任发科 王东 译



Docker 容器 利用Kubernetes、Flannel、Cockpit和Atomic构建和部署

Linux系统或云环境中运行Docker的实用指南!

无论是在笔记本上还是在远程云上，Docker 都能够改变创建、测试、部署和运行应用程序的方式。本书中，作者 Christopher Negus 帮助读者从头开始掌握 Docker 容器化技术。

开始的时候读者能够运行一些 Ubuntu、Fedora、RHEL、CoreOS 或 Project Atomic。读完本书之后，读者就可以在现代 Linux 和云环境中部署企业级质量、多容器的 Kubernetes。

本书是写给系统管理员、软件开发人员和技术狂热者的，书中触及了 Docker 使用的方方面面：设置容器化应用，使用单个和多个容器，在云环境中运行容器，以及开发容器。

书中通过桌面应用、系统服务和游戏等实际例子进行教学，指导读者构建和部署自己的 Docker 化应用。随着经验的充实，读者还会学到不可或缺的 Docker 最佳实践，这些实践用于构建和集成容器、进行日常的 Docker 管理，以及以下几个方面：

- 理解Docker是什么以及能用它来做什么；
- 在标准Linux上安装Docker，或者在Atomic Host和CoreOS这样的容器专用操作系统上安装Docker；
- 搭建容器运行环境和私有Docker registry；
- 创建、运行、探究Docker镜像和容器；
- 查找、拉取、保存、加载和标记容器镜像；
- 在本地系统和Docker registry之间拉取和推送容器；
- 将Docker容器与宿主机的网络和存储集成起来；
- 使用docker build命令和Dockerfile文件构建容器；
- 减少空间消耗并删除不需要的容器；
- 从容器内访问特定的宿主机权限；
- 通过Kubernetes将多个容器编排为复杂的应用；
- 在云环境中使用超级特权容器；
- 使用Cockpit管理云上的容器；
- 开始Docker容器开发；
- 通过共享的Dockerfile学习容器构建技巧。

Christopher Negus写了十多本关于Linux的书，包括畅销书《Red Hat Linux Bible》和《Linux Bible》。他是Red Hat认证架构师（RHCA）以及RHCE认证教练和考官。作为Red Hat的员工，他获得了RHCE和更高认证方面的Linux专业训练。目前，他是多个项目的主要作者，这些项目涉及Docker、Project Atomic、Kubernetes和OpenStack。



Pearson
www.pearson.com

异步社区
人民邮电出版社
www.epubit.com.cn



异步社区 www.epubit.com.cn
新浪微博 @人邮异步社区
投稿/反馈邮箱 contact@epubit.com.cn

美术编辑：董志桢

分类建议：计算机 / 软件开发
人民邮电出版社网址：www.ptpress.com.cn



ISBN 978-7-115-45105-7



9 787115 451057 >

ISBN 978-7-115-45105-7

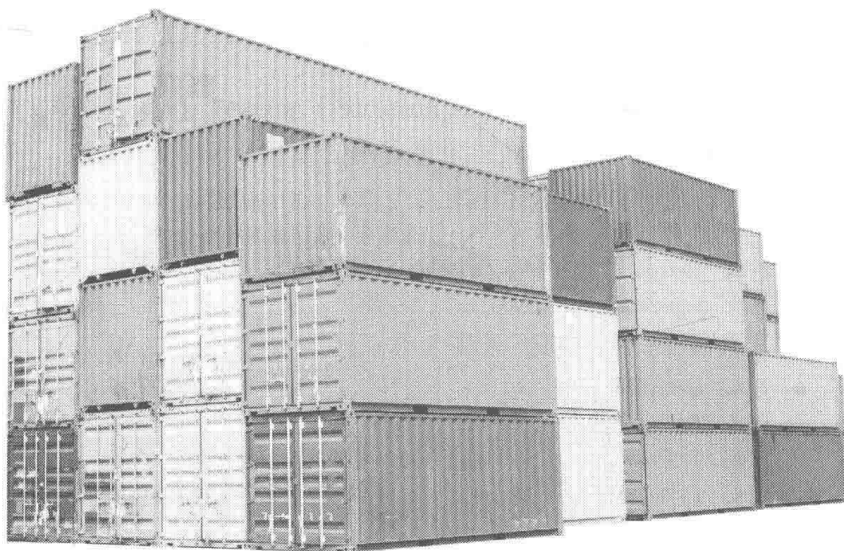
定价：59.00 元

Docker Containers Build and Deploy with Kubernetes,
Flannel, Cockpit, and Atomic

Docker 容器

利用Kubernetes、Flannel、Cockpit和
Atomic构建和部署

[美] Christopher Negus 著 任发科 王东 译



人民邮电出版社
北京

图书在版编目 (CIP) 数据

Docker容器 : 利用Kubernetes、Flannel、Cockpit
和Atomic构建和部署 / (美) 克里斯托弗·尼格斯
(Christopher Negus) 著 ; 任发科, 王东译. — 北京 :
人民邮电出版社, 2017.5

书名原文: Docker Containers: Build and Deploy
with Kubernetes, Flannel, Cockpit, and Atomic
ISBN 978-7-115-45105-7

I. ①D… II. ①克… ②任… ③王… 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(2017)第052959号

内 容 提 要

本书是 Linux 系统及云环境中运行 Docker 的实用指南。书中全面讲解创建、运行、停止、启动、保存和管理容器的具体方法,同时也涉及了多容器管理的主题。本书内容分成 5 个部分。第一部分开启容器之旅,介绍开始使用 Docker 容器所需了解的知识;第二部分关注单个容器,主要介绍如何通过 docker 命令直接使用单个容器;第三部分主题是在云环境中运行容器,描述如何运行超级特权容器(SPC),以及如何使用 Cockpit 在云环境或者本地环境下跨多个宿主机管理容器;第四部分重点是管理多容器,探究容器的编排(利用 Kubernetes);第五部分专注于开发容器,描述一些开发 Docker 容器的建议和技巧,最后通过展示一些 Dockerfile 文件阐述如何构建容器。

本书是写给系统管理员、软件开发人员、运维人员和技术狂热者的,书中触及 Docker 使用的方方面面。阅读本书不需要读者了解 Docker 或者容器化方面的知识,读者可以将本书作为 Docker 的入门书。当然,本书也探讨了更高级的 Docker 话题,如编排和容器的开发。

-
- ◆ 著 [美] Christopher Negus
 - 译 任发科 王东
 - 责任编辑 杨海玲
 - 责任印制 焦志炜
 - ◆ 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>
 - 北京市昌平百善印刷厂印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 14.75
 - 字数: 315 千字 2017 年 5 月第 1 版
 - 印数: 1-3 000 册 2017 年 5 月北京第 1 次印刷
 - 著作权合同登记号 图字: 01-2016-3946 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

广告经营许可证: 京东工商广字第 8052 号

版权声明

Authorized translation from the English language edition, entitled DOCKER CONTAINERS (INCLUDES CONTENT UPDATE PROGRAM): BUILD AND DEPLOY WITH KUBERNETES, FLANNEL, COCKPIT, AND ATOMIC, 1E, by NEGUS, CHRISTOPHER, published by Pearson Education, Inc., Copyright © 2016 by Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD. and POSTS & TELECOM PRESS Copyright © 2017.

This edition is manufactured in the People's Republic of China, and is authorized for sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macau SAR).

本书中文简体字版由 Pearson Education Asia Ltd. 授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

本书封面贴有 Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。版权所有，侵权必究。

一如既往，谨以此书献给我的爱妻 Sheree。

——Christopher Negus

译者序

自互联网诞生以来，其简单、快速的产品开发理念和实践渐成主流，并迅速突破互联网行业向软件行业的各个方面波及开来。然而，这种表面的高歌猛进的背后却是龙鱼混杂、泥沙俱下。一方面，有些人将手拉肩扛的“996”当成文化，追求通过简单粗暴的方式维持较高的执行力；另一方面，有些人致力于寻找更为优秀的流程及工具，希冀通过它们来高效地应对越来越复杂的问题。无涉对错，追求进步的努力从未稍作停歇。

回溯软件开发短短几十年的历史，自第一次软件危机以来，寻找高效而优雅的开发方法是一个一路陪伴软件行业成长的问题。从早期的软件工程，到以 CMMI 为代表的过程改进，再到以精益为宗的敏捷系列方法，在面向交付的领域，我们已经摸索出一套研发对接业务的有效方法。而后互联网的兴起，交付不再是软件开发生命周期的终点，部署和运维成为软件开发所需关注的新阶段。正是在此前提下，DevOps 应时而起，其意在给出一个将代码变更高效地推送到生产环境的方式。但有别于过程优化，技术在部署和运维领域始终是最为关键的因素。而这其中，对软件运行环境的抽象和管理一直是整个部署和运维高效化的难点。

在 Docker 之前，环境管理与应用管理通常被分割开来，交给不同团队负责并通过变更流程进行协调。虽然 Ansible、Puppet 这类配置管理工具能够将环境信息抽象出来并以描述性的方式进行配置，从而大大降低环境管理的难度。但从持续部署的角度看，部署流水线需要将环境部署和应用部署统一起来，以便实现一键式的高效自动化部署。那些早期互联网企业，已在其成长过程中逐步构建起自己的整个部署生态。例如，亚马逊通过 Brazil Build 和 Apollo 系统可以完成每日 25 000 次以上的部署工作，而 Google 和 Facebook 也通过各自的工具可以达到每日 5 000 次左右的部署。但对于中小型互联网公司而言，从头自己研发一套这样的系统投入依然巨大。

Docker 这种轻量级容器技术，不但可以将环境与应用打包分发，而且可以通过描述性的方式进行构建，这无疑使部署流水线中最复杂的问题变得简单。加上 Kubernetes 这样的容器集群管理技术，中小型互联网公司得以以较低的成本完成自己的部署工具链，从而在部署和运维方面可以和互联网巨头一样高效。

毋庸多言，现代 IT 企业的发展走到了一个加速分化的时代。将优秀产品推向用户的成本

和速度将成为决定企业生存与否的关键因素之一。优秀的互联网企业已装备精良并持续优化，你的企业是要日以继夜地赶工，还是要周期性地吐旧纳新？

工具已在手中，时不我待，行动起来……

任发科（常新居士）

2017年2月28日于柔克学院

译者简介



任发科 网名常新居士，曾任职于亚马逊中国，从事 SDE 和 SDM 的工作。具有十余年的企业软件架构、开发和管理经验，了解软件研发与运营中各环节的工作，目前从事大数据分析工具以及 DevOps 工具链的研发，同时关注 IT 研发管理和团队建设。



王东 毕业于天津大学，毕业后一直从事软件相关研发和架构设计工作，曾在普元软件任资深架构师、在 IBM GBS 任咨询经理、在亚马逊任架构师等，后加入创业公司，从事研发和管理工作；热爱编程，喜欢钻研新技术，对微服务、企业架构、大数据以及 DevOps 有浓厚的兴趣。

前言

Docker 这一容器化技术目前正处于新浪潮的中心，这一浪潮波及了应用的构建、打包和部署。它有可能影响计算机技术的方方面面，从应用程序的开发流程到应用程序如何部署以及跨大规模数据中心进行垂直和水平扩展。

尽管 Docker 非常流行，但它依然是一个非常新的项目，许多人并没有真正理解什么是 Docker。如果你也是其中一员，那么本书会帮你迈出第一步，并让你见识到容器化所承诺的巨大潜力。我的目标是通过本书引领你进入容器化的世界，这些目标可以概括为以下几种方式。

- **通过实践来学习：**我常在自己的书中提及这一点，我相信学习技术最好的方式就是获取并使用它。为了做到这一点，我让你从几种流行的 Linux 系统中进行选择，并介绍如何在所选的系统上安装 Docker，而且还给出使用 Docker 做各种事情的实际例子——从运行简单的容器到构建和管理自己的容器镜像。之后学习会延伸至编排和管理容器的工具和技术。
- **如何通过 Docker 受益：**我描述了在容器中创建和运行应用程序的好处，而不是安装软件包（如 RPM 或者 Deb 格式）并直接从硬盘上运行非容器化的应用程序。除了运行应用程序，我还描述了容器化如何为软件开发人员和系统管理人员带来益处。
- **Docker 的本质特性：**我描述了 Docker 如何利用诸如 Linux 容器（LXC）技术保持容器与宿主机上运行的其他应用程序隔离或者如何有选择性地介入宿主机系统。这些特性包括 Docker 如何使用命名空间、元数据以及独立的文件系统来管理容器化应用并保证其安全。

阅读本书并不需要读者了解 Docker 或者容器化方面的知识，读者可以将本书作为 Docker 的入门书。当然，本书也探讨了更高级的 Docker 话题，如编排和容器的开发。

随着阅读的深入，读者将看到运行、调查、停止和启动、保存以及管理容器的具体方法。开始创建容器时，我讨论了一些技巧，这些技巧将有助于读者创建高效地构建和运行的容器镜像。我还将带读者逐步研究其他人为了生成自己的容器而创建的构建文件（其被称为 Dockerfile）。

阅读本书大体上并不需要 Linux 容器，或者 Docker 容器的知识。不过，话虽如此，有些

技术在 Docker 容器内部和外部都会用到，了解其中的一些会有助于更加得心应手地使用 Docker。

对使用 Docker 有帮助的知识

了解一些 Docker 运行环境的知识有助于更好地使用 Docker 容器。Docker 构建在 Linux 技术之上，并与特定的高级特性相结合，包括用于管理 Linux 命名空间的 Linux 容器 (LXC)，以及用于管理容器对系统资源（如 CPU 和内存）的访问的 Cgroups。

即使是与 Docker 容器最基本的交互，也依赖于底层的 Linux 技术。你可能听说过 Docker 可以运行在 Windows 或者 Mac 系统之上，但在这些系统上运行 Docker 始终依赖 Linux 虚拟机。换句话说，没有 Linux 就没有 Docker 容器。同样，每个容器自身通常都是从基础镜像构建的，而这个基础镜像又是从特定的 Linux 发行版创建的。

因此，如果没有 Linux 系统的使用经验，你会发现，了解 Linux 的如下方面及相关技术会很有用。

- **命令 shell:** 虽然有用来使用 Docker 的图形界面，但本书中的大多数 Docker 的例子都是通过 Linux 命令行 shell 完成的。了解如何使用 Linux shell 可以更高效地使用 Docker。
- **软件包:** Docker 本身是一种软件的交付机制，软件会被打包并作为一个套装应用一同交付。然而，为了构建容器镜像，多数 Docker 基础镜像都允许从其所基于的特定 Linux 发行版安装软件包。

因此，对于 Ubuntu 基础镜像，需要了解如何通过 apt-get 之类的工具安装 Deb 包。对于 Fedora、Red Hat Enterprise Linux 或者 CentOS 等的 Docker 镜像，yum、dnf 和 rpm 命令非常有用。当你使用这些基础镜像构建自己的 Docker 容器时，这些基础镜像通常能从在线的软件 repository 中自动获取你请求的软件包。了解在你选择的 Linux 发行版上如何获取和安装软件包对于成功使用 Docker 是至关重要的。

- **文件所有权和权限:** Linux 系统中以及容器中的每一个文件都由特定的用户和组所拥有，并设定了某种权限来访问这些文件。有时你想授权在容器中访问来自宿主机的文件和目录。它们中的一些可能是特殊文件，如设备或套接字，这些都是运行应用程序所需要的。进程也需要作为特定的用户运行。了解这些权限如何工作对于容器的正确工作至关重要。

我只提及了有效使用 Docker 容器所需了解的一部分比较显而易见的特性。随着不断探索如何最好地利用自己使用和创建的 Docker 容器，你会接触到很多其他 Linux 相关的特性。

如果不熟悉 Linux，强烈建议你上一门 Linux 基础知识的课程或者读一本讲解 Linux 基础知识的书，以便使用 Docker 容器。我谨建议你阅读本人所著的《Linux Bible》（第 9 版）(<http://www.wiley.com/WileyCDA/WileyTitle/productCd-1118999878.html>)。这本书不但在构建 Docker 容器所需的技术方面能对你有具体帮助，而且能在开发 Docker 容器镜像时帮你应对

Linux 环境下的工作。

本书涵盖的内容

对于刚开始使用 Docker 容器的人来说，本书要从头至尾地阅读。之后，它可以当作参考资料，提示你与 Docker 容器相关的不同选项和特性。本书内容分成 5 个部分。

第一部分 开启容器之旅

在第一部分中，将学习开始使用 Docker 容器所需了解的知识。第 1 章将描述什么是容器，以及容器与非容器化应用的差别。在第 2 章中，将学习如何在通用 Linux 系统（如 Fedora 和 Ubuntu）以及面向容器的特定 Linux 系统（如 CoreOS 和 Project Atomic）上安装 Docker。在第 3 章中，我们将展示如何通过配置私有 Docker registry 来保存自己的 Docker 镜像，以此来完成一个基本的容器设置。

第二部分 玩转单个容器

这部分主要涉及通过 docker 命令直接使用单个容器。第 4 章中将展示如何运行你的第一个容器镜像。为了帮你查找并获取容器镜像，第 5 章会描述如何从 Docker registry 搜索容器镜像，然后拉取想要的镜像，将它保存到文件，并将该镜像加载到其他 Docker 系统中。

第 6 章中将学习如何为镜像添加标签，从而更好地识别镜像所包含的内容，并利用这些信息将镜像推送到 registry。第 7 章中将展示如何探查容器或容器镜像的内部，看一下容器或镜像工作方式的细节。第 8 章中将学习如何停止、启动和重启容器。

第 9 章中将学习如何通过将宿主机的目录挂载到容器中来配置存储。为了学习如何配置容器的网络，第 10 章将描述如何配置 Docker 服务通常使用的默认网络（或不使用网络），以及运行容器的人为单个容器配置网络接口的方法。

为了可能的重用，Docker 缓存了大量数据。第 11 章将展示如何清理创建或者运行容器镜像时遗留下来的缓存数据。第 12 章将学习如何构建 Docker 容器，包括高效构建并运行的容器是如何创建的。

第三部分 在云环境上运行容器

第 13 章将描述如何运行所谓的超级特权容器（super privileged container, SPC）。为了阐述超级特权容器如何工作，我会展示怎样获取那些可以在 RHEL Atomic 系统上完成不同管理任务的镜像。第 14 章将描述如何使用 Cockpit（基于 Web 的容器管理工具）在云环境或者本地环境下跨多个宿主机管理容器。

第四部分 管理多容器

在这一部分，我们将探究容器的编排。第 15 章将描述如何在一个系统中使用 Kubernetes 的 master 和 node 服务，以便能够尝试 Kubernetes。第 16 章我们将超越一体化 Kubernetes 系统，描述如何搭建 Kubernetes 集群。在 Kubernetes 集群就位后，可以通过 master 计算机将容器 pod 中的应用部署到不同的 node 计算机上进行管理。

第五部分 开发容器

在 Docker 出现的很短的一段时间里，能够更加高效地构建容器的技术就已经被开发出来了。第 17 章将描述一些开发 Docker 容器的建议和技巧。最后，第 18 章通过展示我接触的一些 Dockerfile 文件阐述不同的人是如何克服障碍来构建他们自己的容器的。

如果已经准备好了，马上开始阅读第 1 章吧。希望你喜欢这本书！

致谢

我在创作本书的过程中获得了非常多的帮助。日常工作中，我有幸直接与 Red Hat 的人员一同工作，他们在 Docker、Kubernetes 和 Atomic 这样的项目上完成了卓越的工作，并将这些项目扩展和集成到可以用于最严格的企业环境的操作系统中。总之，我想感谢 Red Hat Enterprise Linux Atomic、OpenShift 以及 Linux 容器团队的开发人员、测试人员和其他作者，感谢他们帮助我每天了解有哪些东西使 Linux 容器适用于企业环境。

我想单独提一下来自 Red Hat 的一些人，他们对本书有着直接的影响。首先，William Henry 写了本书关于存储和网络的这两章。能让他来写这些关键章节，我真是幸运之极。除了这些工作，William 还为 Docker 相关的项目做出了卓越的贡献。实际上，许多 docker 命令的联机帮助页都是 William 写的，这些帮助页随 Docker 软件本身一同交付。再者，有 William 一同参与确定本书的内容，这帮助也是无价的。

本书另一个来自 Red Hat 的重要贡献者是 Scott Collier。Scott 对 Docker 一般知识的公开贡献包括关于 Docker 和 Kubernetes 设置的博客，以及通过 Fedora 云计划分享许多 Dockerfile 示例。就本书而言，Scott 十分慷慨地贡献了他的时间，帮我整理了整本书插图说明的技术和示例。

因为是用业余时间写这本书（这就是花费的时间比我预期要长的原因），所以我常常依赖晚上和周末的时间与出版商（Pearson）进行互动。因此，感谢编辑 Chris Zahn 和 Elaine Wiley 审查本书内容，偶尔要在周日晚上给出回复，并且压缩他们的日程安排以便满足我的日程安排。还有来自 Pearson 的我亲爱的朋友 Debra Williams Cauley，他与我一起开发了这个项目，当我想方设法要将满满当当的日程与我想花时间去写其他书籍进行平衡时，他表现出了极大的耐心。

最后，我想感谢我的家人。当一个人写书时，他无论如何会有一段时间忽视他的家庭。我真的为你们所有人感到自豪。尽管我迷迷糊糊地去写作，我儿子 Seth 却作为校园剧的主演在高中音乐节上出色地模仿了 Zac Efron，他还为自己的高中足球队进行了表演。我儿子 Caleb 找到了自己称心的工作，把家安在了缅因州的自己的小有机农场。而我的妻子 Sheree，继续以健身和斯巴达式运动激励着年轻一代。你们的爱和支持是让我持续前进的动力。

作者介绍

Christopher Negus 是 Linux 的畅销书作者、认证的 Linux 教练和考官、Red Hat 认证架构师和 Red Hat 首席技术作家。目前，Christopher 从事的项目包括 Red Hat OpenStack Platform High Availability、Red Hat Enterprise Linux Atomic Enterprise、Kubernetes 及 Docker 格式的 Linux 容器。

作为一个作者，Christopher 已经写了十多本有关 Linux 和开源软件的书。他的《Linux Bible》（第 9 版，2015 年出版）如今一直位列最畅销的 Linux 书。在互联网爆发的时候，Christopher 的《Red Hat Linux Bible》（第 8 版）卖出超过 25 万册，并两次被选为年度最佳 Linux 图书。Christopher 撰写或合著的其他著作还包括《Linux Toolbox》系列、《Linux Toys》系列、《Fedora and Red Hat Enterprise Linux Bible》系列以及 Wiley 出版的《Linux Troubleshooting Bible》。

对于 Prentice Hall 出版社，Christopher 帮助出版了 Negus 软件解决方案丛书。该丛书中，Christopher 撰写了《Live Linux CDs》并合著了《The Official Damn Small Linux Book》。该丛书还包括关于软件开发、Google Apps 和虚拟化方面的书。

Christopher 在 2008 年作为 RHCE 教练加入 Red Hat。针对这个角色，他成为 Red Hat 认证教练（RHCI）和 Red Hat 认证考官（RHCX）。2014 年，Christopher 成为 Red Hat 认证架构师（RHCA），认证涵盖虚拟化管理、部署和系统管理、集群和存储管理，以及服务器硬化（Server Hardening）。2011 年，Christopher 从 Linux 教练的角色转换回 Red Hat 全职作者，从那时起一直做到现在。

目录

第一部分 开启容器之旅

第 1 章 使用 Docker 对应用进行容器化..... 2	Docker 配置..... 17
1.1 了解容器化应用的优缺点..... 2	2.1.4 其他操作系统的 Docker 配置..... 19
1.1.1 容器化应用不是直接在 宿主机上运行的应用..... 3	2.2 容器式 Linux 系统的 Docker 配置..... 20
1.1.2 容器化应用不是直接在 虚拟机上运行的应用..... 3	2.2.1 Atomic Host 的 Docker 配置..... 21
1.1.3 了解容器的优点..... 3	2.2.2 CoreOS 的 Docker 配置..... 23
1.1.4 了解容器化应用面临的 挑战..... 4	2.3 小结..... 25
1.2 了解容器的组成..... 5	第 3 章 搭建私有 Docker Registry..... 26
1.2.1 Docker 项目..... 5	3.1 获取和启动私有 Docker registry..... 27
1.2.2 Docker Hub Registry..... 6	3.1.1 在 Fedora 中搭建 Docker registry..... 27
1.2.3 Docker 镜像和容器..... 7	3.1.2 在 Ubuntu 中搭建 Docker registry..... 30
1.2.4 docker 命令..... 7	3.2 配置私有 Docker registry..... 32
1.3 探究容器..... 9	3.2.1 配置 docker-registry 包..... 32
1.4 小结..... 10	3.2.2 配置 registry 容器..... 35
第 2 章 搭建容器运行时环境..... 11	3.3 了解 Docker 镜像命名空间..... 35
2.1 标准 Linux 系统的 Docker 配置..... 12	3.4 小结..... 37
2.1.1 Ubuntu 的 Docker 配置..... 12	
2.1.2 Fedora 的 Docker 配置..... 14	
2.1.3 Red Hat Enterprise Linux 的	

第二部分 玩转单个容器

第 4 章 运行容器镜像..... 40	4.1.1 启动交互 Bash shell..... 42
4.1 以交互方式运行容器镜像..... 42	4.1.2 玩些基于字符的游戏..... 44

4.1.3 在容器内运行管理命令	45	7.3.3 检视单个容器属性	78
4.2 运行容器化服务	46	7.3.4 检视运行终端会话的容器	79
4.2.1 运行容器化 Web 服务器	46	7.3.5 检视容器的内存和 CPU 限制	80
4.2.2 在容器中运行服务时进行资源限制	48	7.3.6 检视容器的 SELinux 上下文	81
4.3 运行特权容器	49	7.4 寻找探查容器的更多方法	82
4.4 小结	50	7.4.1 用 docker top 查看容器进程	82
第 5 章 查找、拉取、保存和装载容器镜像	51	7.4.2 用 docker attach 与容器内服务进行交互	83
5.1 搜索镜像	51	7.4.3 用 docker exec 在正在运行的容器中启动新进程	83
5.1.1 用 docker 命令搜索镜像	52	7.4.4 用 docker logs 查看容器进程的输出	84
5.1.2 在 Docker Hub 上检索镜像	54	7.4.5 用 docker diff 查看容器的变化	84
5.2 其他 repository	55	7.4.6 用 docker cp 从容器复制文件	85
5.3 从 Registry 拉取镜像	57	7.5 小结	85
5.4 保存和装载镜像	60	第 8 章 容器的启动、停止与重启	86
5.5 小结	61	8.1 停止和启动容器	86
第 6 章 为镜像添加标签	62	8.1.1 停止和启动分离的容器	87
6.1 为镜像分配名称	63	8.1.2 启动和停止交互式容器	88
6.2 为镜像分配标签	64	8.2 重启容器	89
6.3 为镜像分配 repository 名	65	8.3 向容器发送信号	90
6.3.1 为镜像附加用户名	65	8.4 暂停容器与取消暂停	91
6.3.2 为镜像附加 repository 名	67	8.5 等待容器的退出代码	91
6.4 小结	68	8.6 重命名容器	92
第 7 章 探究容器	69	8.7 创建容器	92
7.1 检视镜像和容器	69	8.8 小结	93
7.2 检视镜像	70	第 9 章 配置容器的存储	95
7.2.1 用 docker inspect 检视基础镜像	70	9.1 管理容器的存储	95
7.2.2 用 docker inspect 检视应用程序镜像	72	9.1.1 使用宿主机的卷	95
7.2.3 查看镜像的历史	73	9.1.2 数据卷容器	97
7.3 检视正在运行的容器	73		
7.3.1 启动要检视的容器	73		
7.3.2 检视整个容器配置	74		