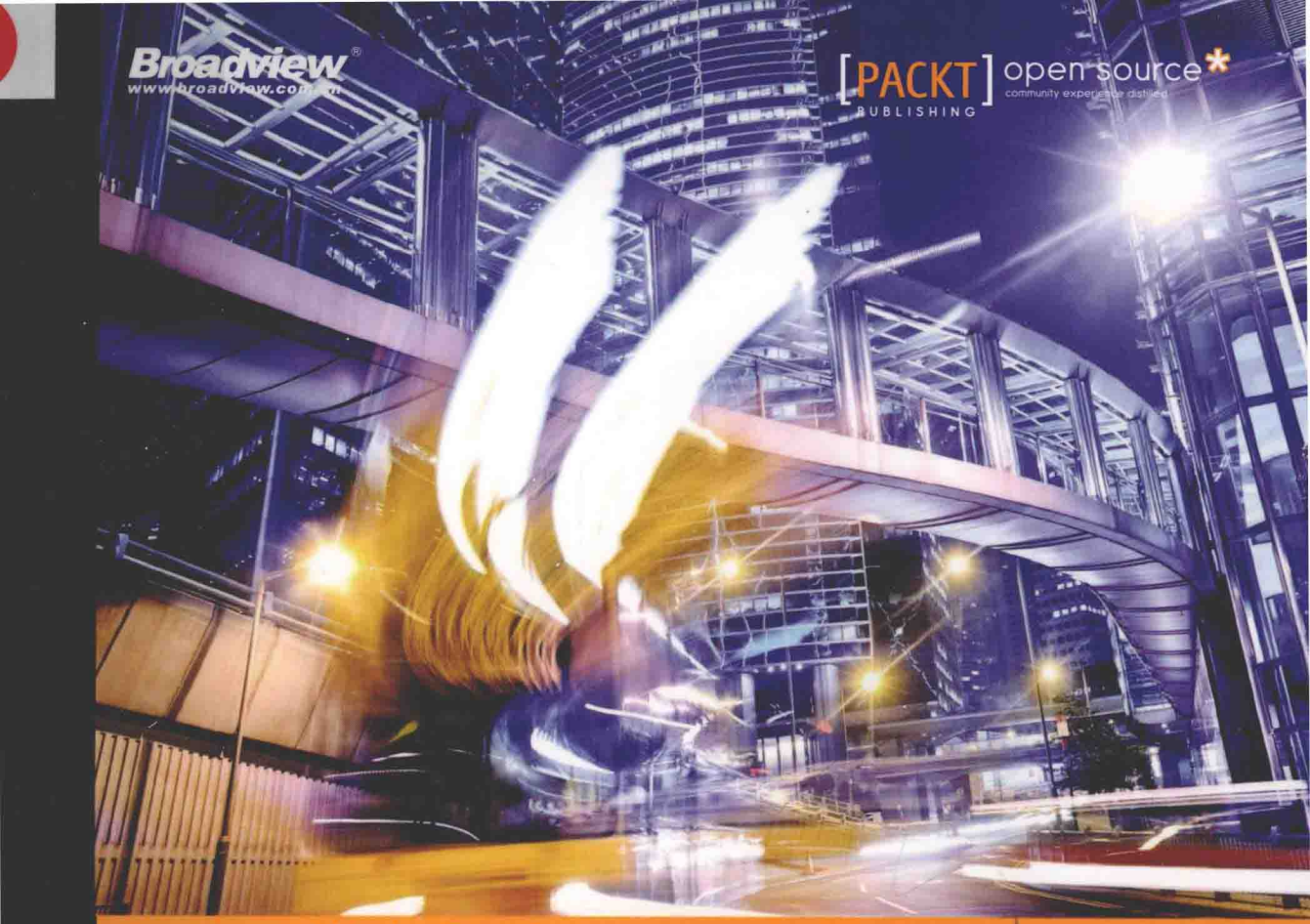

高性能Docker

掌握Docker性能优化实践，更快更高效地部署容器，改善开发工作流

[美] Allan Espinosa 著
DockOne社区：陈杰 杨峰 夏彬 译

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Docker High Performance

高性能Docker

[美] Allan Espinosa 著
DockOne社区：陈杰 杨峰 夏彬 译

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书共分 8 章，旨在帮助读者改善其 Docker 工作流，并保证应用在生产环境中顺利进行。

书中简单回顾了 Docker 是如何工作的。除了 Docker 的基础知识外，读者还会学到如何优化 Docker 基础架构和大规模应用。本书最后讲解的如何在基础架构中部署监控和故障排除系统，更是可以让读者更好地将学到的 Docker 的特性、概念等运用到实践中。

如果你对于管理 Docker 服务和 Linux 文件系统有充分的理解，并希望优化你的 Docker 容器，那本书将非常适合你。

版权贸易合同登记号　图字：01-2016-3365

图书在版编目（CIP）数据

高性能 Docker/（美）艾伦·埃斯皮诺萨（Allan Espinosa）著；陈杰，杨峰，夏彬译. —北京：电子工业出版社，2016.9

书名原文：Docker High Performance

ISBN 978-7-121-28963-7

Ⅰ. ①高… Ⅱ. ①艾… ②陈… ③杨… ④夏… Ⅲ. ①Linux 操作系统－程序设计 Ⅳ. ①TP316.89

中国版本图书馆 CIP 数据核字（2016）第 123021 号

策划编辑：张春雨　刘　芸

责任编辑：刘　舫

印　　刷：北京中新伟业印刷有限公司

装　　订：北京中新伟业印刷有限公司

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：9　字数：186 千字

版　　次：2016 年 9 月第 1 版

印　　次：2016 年 9 月第 1 次印刷

定　　价：69.00 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 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819　faq@phei.com.cn。

译者序

Docker 容器技术从 LXC 技术发展而来，是一个程序运行、测试和交付的开放平台。其中，可以将不同功能的虚拟主机以一个应用程序的方式进行管理，从而帮助用户实现快速测试、编码和交付。Docker 从出现，到开源，到被新一代技术架构整合，到各种初创公司，无疑一直带有传奇色彩，为 IT 技术在中国“互联网+”创业浪潮中增添了一抹亮色。

围绕 Docker 的生态系统，目前在大厂商如 Google，以及本地创业公司的共同推动下，在图像化管理、作业调度、作业编排等领域都有了长足发展。另外，目前最新的发展趋势是，直接在 Mac 和 Windows 中使用的 Docker 已经推出了 beta 版本，以一个后台运行的应用形式表现出来。

Docker 更多与 IaaS、PaaS 的生态系统对接起来，围绕客户管理、维护、排错和作业调度等需求，出现各种管理方式，例如 Kubernetes 和 Swarm 等编排工具。其趋势也是将底层 Docker 的具体启动、运行、暂停和停止进行包装，以便客户更好地专注于 Docker 平台之上的应用，简化底层资源层的管理和维护。

但是对于万千技术爱好者而言，如果不能深入 Docker 内部进行运行机制研究、不能对 Docker 内部性能进行调优，那么就不能很好地应用这一新技术带来的益处。因为从深入研究技术的角度来看，不仅需要使用集成化工具带来的简便，更需要深入了解 Docker 底层的机制，以便真正需要时，找到面临问题的解决方案。

电子工业出版社一直紧跟国外技术热点，适时引进了 *Docker High Performance* 原版书，它是一本面向有一定 Linux 基础、想深入了解 Docker 内部机理以及如何监控和提高容器性能的 Docker 初学者的书。

全书共分 8 章，每一章都有详细步骤讲解，从裸机开始，到最终运行一个模拟系统为止，读者如能全部操作下来，必将从中获得很大收获。

本书译者都是 Docker 社区很有经验的志愿者，在百忙之中抽出宝贵时间进行翻译、校对，希望能对容器技术在国内的推广和使用做出相应的贡献。因为容器技术发展很快，各位译者竭尽所能展现最新技术，但是水平有限，错误在所难免，希望读者能够批评指正。

杨峰

2016 年 7 月

关于作者

Allan Espinosa 是一名生活在东京的 DevOps 从业者，他是很多分布式系统工具的活跃的开源贡献者，比如 Docker 和 Chef。Allan 维护了若干个流行的开源软件的 Docker 镜像，这些镜像甚至比开源团体的官方发布版还要流行。

在他的职业生涯中，Allan 还管理过一些大型分布式系统，包含生产环境中的数百到数千台服务器。他在不同的平台上构建了很多大规模应用，从美国的大型超级计算中心到日本的生产环境企业系统。

你可以通过 Allan 的 Twitter 账号`@AllanEspinosa` 联系到他。他的个人网站是 `http://aespinosa.github.io`，其中有很多关于 Docker 和分布式系统的博客文章。

我要感谢我的妻子 Kana，她一如既往地支持我，使我能够花费大量的时间来写作。

关于审校者

Shashikant Bangera 是一名 DevOps 架构师，具有 16 年的 IT 领域经验。他对于开源 DevOps 工具具有丰富的专业知识。Shashikant 曾经参与大量的价值数百万英镑的项目。从传统的开发实践到敏捷工具和流程的使用的转变这方面，他具有丰富的实践经验，这可以提高发布频率和软件质量。此外，Shashikant 已经使用开源工具设计了一个自动化的、按需的（on-demand）环境。对于大量的 DevOps 工具，他也具有丰富的实践经验。

Packt Publishing 的另一本书 *Learning Docker* 也是由 Shashikant 审校的。

前言

Docker 是一款很好的用于构建和部署应用的工具。可移植的容器格式使我们可以在任何地方运行代码，从开发者工作站到著名的云计算提供商。Docker 的工作流使开发、测试和部署更加容易和快速。然而，Docker 核心和最佳实践的持续改善是很重要的，可帮助你实现 Docker 最大的潜在价值。

本书的主要内容

凡是对 Docker 有基本理解的工程师都可以按顺序一章一章地阅读本书。对 Docker 具有深入理解或者在生产环境中部署过应用的技术领导者们，可以直接阅读第 8 章的内容，了解 Docker 是如何适应已有应用的。以下是本书介绍的一系列主题。

第 1 章，简单介绍了如何搭建和运行 Docker，介绍了贯穿本书都会用到的搭建步骤。

第 2 章，介绍了为什么调优 Docker 镜像是很重要的，介绍了多个调优小窍门，从而改善可部署性和 Docker 容器的性能。

第 3 章，介绍了如何自动化搭建 Docker 宿主机，并讨论了自动化的重要性以及它是如何促进 Docker 容器的大规模部署的。

第 4 章，介绍了如何使用 Graphite 搭建监控系统和使用 ELK 搭建日志系统。

第 5 章，介绍了如何使用 Apache JMeter 来创建负载，并测试 Docker 容器的性能。本章回顾了第 4 章中搭建的监控系统，并分析了若干 Docker 应用的性能基准结果，例如响应时间和吞吐量。

第 6 章，介绍了如何配置和部署基于 Nginx 的负载均衡容器。同时，也介绍了如何使用负载均衡器来水平扩展 Docker 应用的性能和可部署性。

第 7 章，介绍了典型 Linux 系统中的通用调试工具是如何调试 Docker 容器的。并介绍

了每种工具是如何工作的以及是如何读取运行中的 Docker 容器的诊断信息的。

第 8 章，综合了前面几章中的性能优化方法，并介绍了如何在生产环境中使用 Docker 部署任何一个 Web 应用。

你需要做什么准备

你需要一台安装了最新版本内核的 Linux 工作站作为 Docker 1.10.0 的宿主机。本书使用 Debian Jessie 8.2 作为基础操作系统来安装和搭建 Docker。

本书的目标读者

本书是为想要在生产环境中部署 Docker 应用和架构的开发者和运维者而写。如果你已经了解了 Docker 的基本知识，并且想进一步学习 Docker 的话，那么这本书就是适合你的。

惯例

在本书中，我们使用了不同的文本格式，用以区分不同类型的信息。以下是这些格式的例子以及它们的具体含义。

文本格式的代码、数据库表名、目录名、文件名、文件扩展名、路径名、URL、用户输入、Twitter 用户名都是用以下格式书写的："我们会使用`--link <source>:<alias>`来创建源容器 source 到另一个容器 webapp 的连接"。

代码块的格式如下：

```
FROM ubuntu:14.04
MAINTAINER Docker Education Team <education@docker.com>
RUN apt-get update
RUN DEBIAN_FRONTEND=noninteractive apt-get \
        install -y -q python-all python-pip
ADD ./webapp/requirements.txt /tmp/requirements.txt
RUN pip install -qr /tmp/requirements.txt
ADD ./webapp /opt/webapp/
WORKDIR /opt/webapp
```

```
EXPOSE 5000
CMD ["python", "app.py"]
```

当我们希望着重表示代码块中的某一部分时，这一部分就会被设置为粗体。

```
import os
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello():
    provider = str(os.environ.get('PROVIDER', 'world'))
    return 'Hello '+provider+'!'
if __name____== '__main__':
    # Bind to PORT if defined, otherwise default to 5000.
    port = int(os.environ.get('PORT', 5000))
    app.run(host='0.0.0.0', port=port)
```

命令行输入或输出的格式如下：

```
dockerhost$ docker inspect -f "{{ .NetworkSettings.IPAddress }}" \
                           source
172.17.0.15
dockerhost$ docker inspect -f "{{ .NetworkSettings.IPAddress }}" \
                           destination
172.17.0.28
dockerhost$ iptables -L DOCKER
Chain DOCKER (1 references)
target     prot opt source               destination
ACCEPT     tcp  --  172.17.0.28         172.17.0.15          tcp dpt:5000
ACCEPT     tcp  --  172.17.0.15         172.17.0.28          tcp spt:5000
```

警告或者重要注解会出现在这样的图标之后。

提示和技巧将会出现在这样的图标之后。

下载示例代码

你可以从 http://www.broadview.com.cn 下载所有已购买的博文视点书籍的示例代码文件。

勘误表

虽然我们已经尽力谨慎地确保内容的准确性，但错误仍然存在。如果你发现了书中的错误，包括正文和代码中的错误，请告诉我们，我们会非常感激。这样，你不仅帮助了其他读者，也帮助我们改进后续的出版。如发现任何勘误，可以在博文视点网站相应图书的页面提交勘误信息。一旦你找到的错误被证实，你提交的信息就会被接受，我们的网站也会发布这些勘误信息。你可以随时浏览图书页面，查看已发布的勘误信息。

DockOne.io
Community of Container

DockOne.io 成立于 2014 年，是国内最大的容器社区。社区主要关注 Docker、Mesos、CoreOS、Kubernetes、Ceph、OpenStack 等容器生态圈相关软件，致力于为广大容器爱好者提供一个分享、学习和交流的平台，目前已有活跃会员逾 50000 精品文章 1000 余篇。

目录

1

准备 Docker 宿主机

Docker 使应用交付到客户更加便捷。它通过简单地创建和运行容器这种形式，简化了代码从开发环境到生产环境部署过程中的工作流。本章将快速学习如何准备一个基于 Docker 的开发环境，具体的操作步骤如下：

- 准备一个 Docker 宿主机
- Docker 镜像的基本操作
- 运行 Docker 容器

本章的大部分内容是我们已经熟悉的知识，并且可以在 Docker 官方文档网站上查阅到。这里将介绍的是与 Docker 宿主机有关的部分命令和在后续几章中将使用到的交互操作。

准备一个 Docker 宿主机

假定读者熟悉如何创建一个 Docker 宿主机。对于本书的大部分章节，除非有特别说明的情况，我们都将在如下环境中运行示例：

- 操作系统：Debian 8.2 Jessie
- Docker 版本：1.10.0

下面的命令显示了操作系统和 Docker 的版本:

```
$ ssh dockerhost
dockerhost$ lsb_release -a
No LSB modules are available.
```