

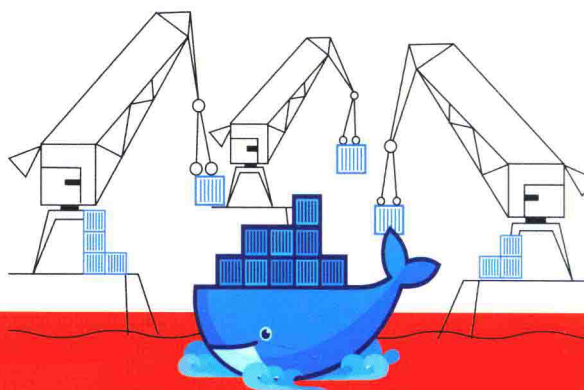
深度剖析Docker的核心概念、实现原理、应用技巧和生态系统
结合实际生产环境，通过实战案例提供有价值的参考

Docker

从入门到实战

Introduction and Advanced Usage of Docker

黄靖钧◎编著



- 涵盖Docker四大管理工具的基本知识，并深入分析
- 从三大组件入门应用到集群编排进阶实战，条理清晰
- 结合实际生产环境介绍上百个案例，内容都是有价值的干货
- 以Docker当前的流行版本为例讲解Swarm集群管理



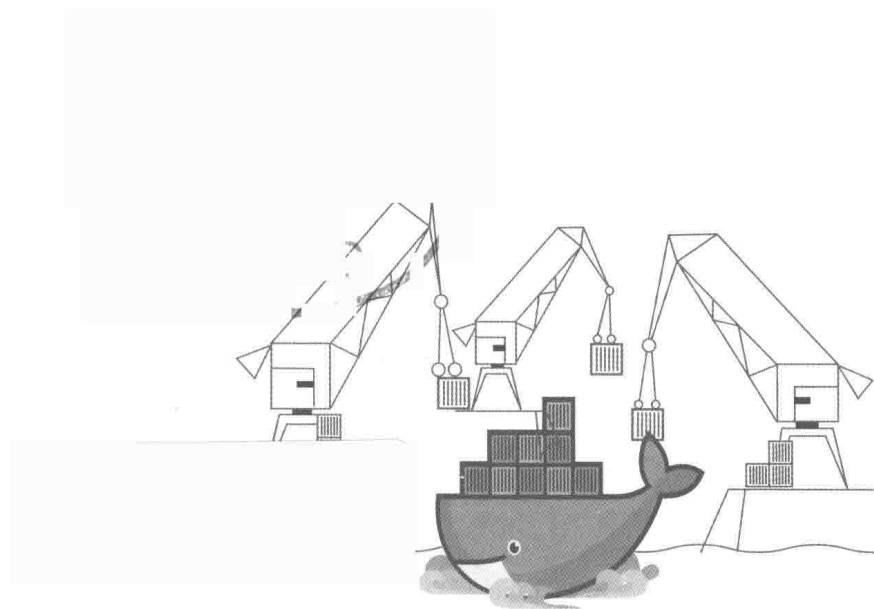
机械工业出版社
China Machine Press

Docker

从入门到实战

Introduction and Advanced Usage of Docker

黄靖钧◎编著



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Docker 从入门到实战 / 黄靖钧编著. —北京：机械工业出版社，2017.6

ISBN 978-7-111-57328-9

I. D… II. 黄… 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（2017）第 166174 号

Docker 从入门到实战

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：欧振旭

责任校对：姚志娟

印 刷：中国电影出版社印刷厂

版 次：2017 年 8 月第 1 版第 1 次印刷

开 本：185mm×260mm 1/16

印 张：22.75

书 号：ISBN 978-7-111-57328-9

定 价：69.00 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

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问：北京大成律师事务所 韩光/邹晓东

前言

Docker 作为一个 2013 年才诞生的开源项目，其发展的速度和火爆程度却令人惊叹。容器技术本不是什么新鲜事物，但是在 Docker 的整合下，一切变得清晰、易用起来，并且随着各大云计算厂商的进场，使 Docker 得到了极大的推广。

如今，Docker 已经成为容器技术领域当仁不让的领头羊。国内外以 Docker 技术起家的创业公司如雨后春笋般涌现出来，体现了容器市场的巨大需求。越来越多的企业开始逐步把传统的应用开发流程迁移到 Docker 容器中作为开发部署流程的一环。伴随而来的是各种复杂的需求与 Docker 尚不算完善的功能所产生的矛盾，这些问题制约着企业容器化的脚步。

另一方面，Docker 以其友好的使用体验使广大开发者对其“一见倾心”，越来越多的开发者使用 Docker 作为应用分发部署的一个重要阵地。尽管如此，Docker 对于大部分开发者而言还是尚未开拓的疆土。特别是对于国内环境而言，Docker 的推广基本上靠国内几家与 Docker 相关的初创公司。本书以一位普通的全栈开发者的身份，详细介绍了 Docker 的基础知识，分享了企业级容器云的实战经验。

为什么学习 Docker

如果您是一名开发者，想必遇到过“这个程序只有在我的机器上才可以运行”的情况。随着用户需求变得多样，软件愈发复杂，所依赖环境愈发庞大，使得软件在其他机器上运行需要做大量的迁移工作。更糟糕的是，这些琐事完成后软件还不一定能正常运行。

为了解决这些问题，虚拟化技术开始普及。人们可以通过各种虚拟化技术来实现软件的迁移和分发。最常见的就是虚拟机或 KVM 技术。在虚拟机里完成开发再迁移到线上不会出现环境问题，解决了迁移过程中的诸多难题，但是仍然存在性能低下、分发流程麻烦、耗时和成本昂贵等问题。在云计算时代这些问题更加突出。

随着容器技术的普及，人们意识到容器技术可以极大地降低成本。容器技术具有启动快、体积小和分发迅速等诸多特点，这简直就是开发人员梦寐以求的工具。

而“欣喜若狂”的不止是开发人员，还有运维人员。如果在十年前，普通企业要管理上百台服务器，最可能使用的方法是通过 Shell 脚本的方式使用 SSH 连接到所有服务器然后执行相同的指令，并把日志保存起来归档。这种方式我们称之为第一代运维。那时维护服务器是一项繁重的工作，工程师不得不把大量的时间耗费在服务器管理上。

随着技术的发展，虚拟化技术的普及和云计算的出现，企业需要管理的服务器数量大幅增长。过去我们只要管理企业内部数据中心的物理服务器，而现在则要管理遍布全球的服务器，运维成本愈发昂贵。于是开发者开始针对云计算时代服务器运维方式做出改变，涌现出了诸如 Ansible、Puppet、SaltStack 和 Chef 等出色的运维工具。人们可以通过这些

工具快速地完成对上百台甚至上千台服务器的管理操作。这被称之为第二代运维。它极大地解决了管理庞大服务器集群的难题，使人们可以在屏幕面前通过一个界面管理所有服务器。但本质上这些工具都是通过 SSH 或者类似于 SSH 的方式连接到服务器来管理服务器集群，这意味着其实第二代运维和第一代运维并没有发生根本性的改变。

上面那些运维工具在云计算普及的大势下很快暴露了它们的问题——速度。因为大部分运维工具依靠的是 SSH 连接来交换信息，这使得整个过程十分耗时，更不用说其他复杂的管理操作。而随着容器技术的爆发，以 Docker 为代表的容器技术开始发力，并随着 DevOps 概念的普及，使运维发生了根本性的改变。容器集群管理不再是通过低效的 SSH 来连接服务器，甚至不需要登录服务器就可以完成对服务器的管理。人们发现，通过容器管理集群可以抛弃传统的“SSH+秘钥”的连接方式来连接服务器，这对大规模集群来说是一个极大的变革。而且在速度上，容器技术在上百台服务器上启动应用只需要一眨眼的时间，这使得运维的工作大大减轻。

运维和开发在容器时代逐步“融为一体”，形成一个流水线车间的工作环境。这对于软件行业来说无疑是一次巨大的变革。

如果您也对传统的软件开发和运维的烦琐流程感到吃力，又对容器技术感兴趣，那么本书将是很好的入门书籍。

如果您不是职业的开发和运维人员，对 Linux 也不算熟悉，但属于一个对 Docker 感兴趣的极客，想通过 Docker 部署一些复杂的应用，本书也一样适合您。本书虽以 Linux 为平台介绍 Docker 的使用，但是与在 Windows 和 Mac OS 平台上的操作基本一致，普通用户完全可以把 Docker 当做一个“好玩的工具”来使用，体验 Docker 带来的便捷。

本书特色

- **适合新手入门。**本书在基础方面内容非常详尽，包括镜像的构建、容器的运行监控、网络的管理、仓库的应用、集群的部署等内容，全面、细致地介绍了 Docker 的基本使用方法与实现原理，适合新手入门。
- **应用结合实际。**本书在实战应用部分结合实际情况，从不同的角度分析问题并提出对应的解决办法，扩展了很多实用的实战技巧。实战部分根据不同类型的开发环境构建基础开发环境镜像，使读者可以直接使用 Docker 进入测试开发，并根据不同类型的部署做了详细介绍。
- **范例丰富。**在实战章节中的范例皆由浅入深，全面、实用且不缺乏趣味性，有助于读者了解其内部原理，进而应用到其他项目的思考与开发中。全书的代码均有指明出处以及链接，读者可以通过文中链接找到源代码。
- **版本最新。**本书使用目前流行的 Docker 1.12 版本，紧跟 Docker 更新步伐，介绍了新的 Docker Swarm 集群管理方式。

本书内容体系

第1篇 容器技术与Docker概念（第1~3章）

本篇主要介绍了容器技术的发展历史与容器技术的原理，并解释了 Docker 与其他容器

技术的区别，对比了 Docker 与虚拟机的异同，客观地评价了两者的优缺点。另外，本篇还介绍了 Docker 分别在 Linux、Windows 和 Mac OS 系统下的安装方法，以及二进制安装方法。

第2篇 Docker基础知识（第4~10章）

本篇主要介绍了 Docker 的基本操作及简单应用，包括 Docker 基础、Docker 镜像、Docker file 文件、Docker 仓库、Docker 容器、数据卷的使用方法与原理及网络管理等内容。通过对本篇内容的学习，读者可以掌握最常用的 Docker 知识。

第3篇 Docker进阶实战（第11~19章）

本篇包含了许多 Docker 在实际开发中的应用实例，包括操作系统、编排工具 Compose、Web 服务器与应用、数据库、编程语言、Docker API、私有仓库、集群网络、Docker 安全等内容，详细讲解了 Docker 在容器云环境中的应用。读者通过这部分内容的学习已经完全可以在实际生产环境中应用 Docker 了。

本书读者对象

- Dock 开发入门人员；
- 容器技术爱好者；
- 各类运维人员；
- 基于 Docker 构建云计算平台的技术人员；
- 大、中专院校的学生；
- 相关培训学校的学员。

本书配套资源及获取方式

本书涉及的源代码文件等配套学习资源需要读者自行下载。请读者登录机械工业出版社华章公司的网站 www.hzbook.com，然后搜索到本书页面，按照页面上的说明进行下载即可。

本书作者

本书由黄靖钧主笔编写。其他参与编写的人员有张昆、张友、赵桂芹、晁楠、高彩琴、郭现杰、刘琳、王凯迪、王晓燕、吴金艳、尹继平、张宏霞、张晶晶、陈冠军、魏春、张燕、范陈琼、孟春燕、王晓玲、项宇峰、肖磊鑫、薛楠、杨丽娜、闫利娜、王韶、李杨坡、刘春华、黄艳娇、刘雁。

本书的顺利出版，要感谢机械工业出版社华章公司各位编辑的辛勤劳动和付出，另外对网络上提供有益资料的众多作者也在此表示感谢。

虽然我们对本书中所述内容都尽量核实，并多次进行文字校对，但因时间所限，加之水平所限，书中疏漏和错误在所难免，敬请广大读者批评指正。

目录

前言

第 1 篇 容器技术与 Docker 概念

第 1 章 容器技术	2
1.1 什么是容器	2
1.1.1 关于虚拟化	2
1.1.2 容器的定义	3
1.1.3 为什么使用容器	3
1.2 容器技术的前世今生	4
1.2.1 容器技术的起源	4
1.2.2 容器技术的发展	5
1.3 容器的原理	7
1.3.1 从 namespace 说起	7
1.3.2 认识 Cgroups	9
1.3.3 容器的创建	11
1.4 容器云	12
1.5 容器与 Docker	13
1.6 本章小结	13
第 2 章 Docker 简介	14
2.1 什么是 Docker	14
2.1.1 Docker 的历史	14
2.1.2 Docker 的现状	16
2.1.3 Docker 的未来	17
2.2 Docker 的功能及优缺点	18
2.2.1 Docker 在解决什么	18
2.2.2 为什么选择 Docker	19
2.2.3 Docker 的缺点	19
2.3 Docker 和虚拟机	19
2.3.1 Docker 与虚拟机的区别	20
2.3.2 Docker 与虚拟机的优缺点	20
2.4 Docker 与 runC	21
2.4.1 libcontainer 与 runC	21
2.4.2 runC 的使用	22

2.4.3	runC 原理	22
2.5	Docker 基本架构	24
2.5.1	Docker Client 介绍	24
2.5.2	Docker daemon 介绍	25
2.5.3	Docker 镜像	25
2.5.4	Docker 容器	26
2.5.5	Docker 仓库	26
2.6	本章小结	26
第 3 章	安装 Docker	27
3.1	Linux 系统	27
3.1.1	一键安装脚本	27
3.1.2	Debian 发行版	28
3.1.3	Ubuntu 发行版	30
3.1.4	Centos/Fedora 发行版	33
3.1.5	Arch Linux 发行版	37
3.1.6	Suse/openSUSE 发行版	38
3.2	Windows 与 Mac OS 系统	38
3.2.1	在 Windows 上安装原生 Docker	39
3.2.2	在 Mac OS 上安装原生 Docker	41
3.3	二进制安装	43
3.3.1	获取 Linux 二进制包	44
3.3.2	获取 Mac OS X 二进制包	44
3.3.3	获取 Windows 的二进制包	45
3.3.4	树莓派安装 Docker	45
3.4	本章小结	46

第 2 篇 Docker 基础知识

第 4 章	Docker 基础	48
4.1	Docker 基本操作	48
4.1.1	依附容器的 docker attach 命令	49
4.1.2	构建镜像的 docker build 命令	51
4.1.3	提交容器的 docker commit 命令	52
4.1.4	复制文件到宿主机的 docker cp 命令	52
4.1.5	创建容器的 docker create 命令	53
4.1.6	查看容器变化的 docker diff 命令	54
4.1.7	查看事件的 docker events 命令	54
4.1.8	进入容器的 docker exec 命令	55
4.1.9	导出容器的 docker export 命令	56
4.1.10	查看镜像历史的 docker history 命令	56
4.1.11	查看本地镜像的 docker images 命令	57
4.1.12	导入容器的 docker import 命令	58

4.1.13	查看 Docker 信息的 docker info 命令	58
4.1.14	查看各项详细信息的 docker inspect 命令	59
4.1.15	杀死容器的 docker kill 命令	60
4.1.16	导入镜像的 docker load 命令	60
4.1.17	登录仓库的 docker login 命令	61
4.1.18	登出仓库的 docker logout 命令	61
4.1.19	查看容器日志的 docker logs 命令	62
4.1.20	管理网络的 docker network 命令	62
4.1.21	管理节点的 docker node 命令	63
4.1.22	暂停容器的 docker pause 命令	64
4.1.23	查看容器端口的 docker port 命令	64
4.1.24	查看本地容器信息的 docker ps 命令	65
4.1.25	拉取镜像的 docker pull 命令	65
4.1.26	推送镜像的 docker push 命令	66
4.1.27	重命名容器的 docker rename 命令	66
4.1.28	重启容器的 docker restart 命令	66
4.1.29	删除容器的 docker rm 命令	67
4.1.30	删除镜像的 docker rmi 命令	67
4.1.31	运行容器的 docker run 命令	68
4.1.32	导出镜像的 docker save 命令	72
4.1.33	搜索镜像的 docker search 命令	73
4.1.34	管理服务的 docker service 命令	74
4.1.35	启动容器的 docker start 命令	74
4.1.36	查看容器状态的 docker stats 命令	75
4.1.37	停止容器的 docker stop 命令	75
4.1.38	管理集群的 docker swarm 命令	76
4.1.39	设置镜像标签的 docker tag 命令	76
4.1.40	查看容器进程的 docker top 命令	77
4.1.41	恢复暂停容器的 docker unpause 命令	77
4.1.42	更新容器的 docker update 命令	77
4.1.43	查看 Docker 版本的 docker version 命令	78
4.1.44	管理数据卷的 docker volume 命令	78
4.1.45	设置等待的 docker wait 命令	79
4.2	启动第一个 Docker 容器	79
4.3	构建第一个 Docker 镜像	80
4.4	本章小结	81
第 5 章	Docker 镜像	82
5.1	认识镜像	82
5.1.1	使用 docker pull 拉取镜像	82
5.1.2	搜索镜像	83
5.1.3	查看镜像信息	84
5.2	创建镜像	86
5.2.1	剖析 Hello World 镜像	86
5.2.2	从 Dockerfile 构建镜像	86

5.2.3	自动构建镜像	87
5.2.4	提交容器为镜像	90
5.3	导出和导入镜像	91
5.3.1	导出镜像到本地文件系统	91
5.3.2	从本地文件系统导入镜像	91
5.4	发布镜像	91
5.4.1	发布镜像到 Docker Hub	92
5.4.2	给镜像打上标签	92
5.4.3	发布到第三方镜像仓库	92
5.5	删除镜像	93
5.5.1	删除本地镜像	93
5.5.2	删除仓库镜像	93
5.6	Docker 镜像扩展	94
5.6.1	Docker 镜像里有什么	94
5.6.2	Docker 镜像的存储方式	95
5.6.3	联合挂载	95
5.6.4	Git 式管理	96
5.7	本章小结	96
第 6 章	Dockerfile 文件	97
6.1	Dockerfile 基本结构	97
6.1.1	Dockerfile 基础	97
6.1.2	Dockerfile 的书写规则	98
6.1.3	基础镜像信息和维护者信息	99
6.2	Dockerfile 指令	99
6.2.1	指定基础镜像的 FROM 指令	99
6.2.2	设置维护者信息的 MAINTAINER 指令	99
6.2.3	执行构建命令的 RUN 指令	99
6.2.4	设置镜像环境变量的 ENV 指令	100
6.2.5	复制文件的 COPY 指令	100
6.2.6	添加文件的 ADD 指令	100
6.2.7	指定端口暴露的 EXPOSE 指令	100
6.2.8	设置镜像启动命令的 CMD 指令	101
6.2.9	设置接入点的 ENTRYPOINT 指令	102
6.2.10	设置数据卷的 VOLUME 指令	102
6.2.11	设置构建用户的 USER 指令	103
6.2.12	设置工作目录的 WORKDIR 指令	103
6.2.13	设置二次构建指令的 ONBUILD 指令	104
6.2.14	设置元数据的 LABEL 指令	105
6.2.15	设置构建变量的 ARG 指令	105
6.2.16	设置停止信号的 STOPSIGNAL 指令	105
6.2.17	检查镜像状态的 HEALTHCHECK 指令	105
6.2.18	设置命令执行环境的 SHELL 指令	106
6.3	镜像构建实战	106
6.3.1	收集应用信息	106

6.3.2	编写 Dockerfile	106
6.3.3	设置自动构建	107
6.4	本章小结	108
第 7 章	Docker 仓库	109
7.1	官方仓库 Docker Hub	109
7.1.1	Docker Hub 登录与使用	109
7.1.2	Docker Hub 与 Docker Cloud	110
7.2	国内镜像仓库	111
7.2.1	中国科学技术大学镜像仓库	111
7.2.2	DaoCloud 镜像加速器	112
7.2.3	阿里云镜像加速器	113
7.2.4	灵雀云镜像加速器	114
7.2.5	时速云镜像加速器	115
7.2.6	网易蜂巢	116
7.2.7	自建镜像加速器	116
7.3	私有仓库	117
7.3.1	搭建私有仓库	117
7.3.2	私有仓库的使用	117
7.3.3	私有仓库安全性	118
7.4	Registry 原理	118
7.4.1	Registry 组成	118
7.4.2	Registry 工作流程	119
7.5	本章小结	120
第 8 章	Docker 容器	121
8.1	容器基本操作	121
8.1.1	创建容器	122
8.1.2	启动容器	122
8.1.3	后台运行容器	123
8.1.4	自动重启容器	123
8.1.5	停止与杀死容器	124
8.1.6	删除容器	125
8.1.7	查看容器信息	125
8.2	进入容器内部	128
8.2.1	使用 attach 进入容器	128
8.2.2	使用 exec 进入容器	129
8.2.3	使用 nsenter 进入容器	129
8.3	导出和导入容器	130
8.3.1	导出容器	130
8.3.2	导入容器	130
8.4	容器结构	131
8.4.1	容器格式是什么	131
8.4.2	容器内部结构	132
8.5	本章小结	133

第 9 章 数据卷 134

9.1 数据卷是什么 134

9.1.1 数据卷介绍 134

9.1.2 数据卷容器介绍 135

9.2 为容器挂载数据卷 135

9.2.1 挂载数据卷 135

9.2.2 挂载数据卷容器 136

9.2.3 数据卷挂载小结 137

9.3 备份、恢复、迁移数据卷 139

9.3.1 备份数据卷 139

9.3.2 迁移、恢复数据卷 140

9.4 容器数据卷扩展 140

9.4.1 卷插件介绍 140

9.4.2 Convoy 的使用 140

9.4.3 Flocker 的使用 141

9.5 本章小结 142

第 10 章 网络管理 143

10.1 Docker 网络基础 143

10.1.1 端口映射 143

10.1.2 端口暴露 145

10.1.3 容器互联 146

10.2 Docker 网络模式 147

10.2.1 none 模式 147

10.2.2 container 模式 149

10.2.3 host 模式 150

10.2.4 bridge 模式 151

10.2.5 overlay 模式 152

10.3 Docker 网络配置 152

10.3.1 Daemon 网络参数 152

10.3.2 配置 DNS 153

10.3.3 network 命令 154

10.4 本章小结 154

第 3 篇 Docker 进阶实战

第 11 章 操作系统 156

11.1 Alpine 发行版 156

11.1.1 官方镜像 156

11.1.2 运行 Alpine Linux 157

11.1.3 构建基于 Alpine Linux 的镜像 157

11.1.4 Alpine Linux 软件仓库 158

11.2 Busybox 发行版 159

11.2.1	官方镜像	159
11.2.2	运行 Busybox	159
11.2.3	构建基于 Busybox 的镜像	159
11.3	Debian/Ubuntu 发行版	160
11.3.1	官方镜像	160
11.3.2	运行 Debian/Ubuntu	160
11.3.3	构建基于 Debian/Ubuntu 的镜像	161
11.4	CentOS/Fedora 发行版	162
11.4.1	官方镜像	162
11.4.2	运行 CentOS/Fedora	162
11.5	CoreOS 发行版	163
11.5.1	为什么使用 CoreOS	163
11.5.2	用 Vagrant 安装 CoreOS	163
11.6	RancherOS 发行版	165
11.6.1	为什么使用 RancherOS	165
11.6.2	在服务器安装 RancherOS	166
11.6.3	基于 RancherOS 的 Docker 管理	167
11.7	本章小结	167
第 12 章	编排工具 Compose	169
12.1	安装 Docker Compose	169
12.1.1	二进制安装	169
12.1.2	使用 Python pip 安装	169
12.2	Compose 命令基础	170
12.2.1	指定配置文件	171
12.2.2	指定项目名称	171
12.2.3	Compose 环境变量	171
12.2.4	构建服务镜像的 build 命令	172
12.2.5	生成 DAB 包的 bundle 命令	173
12.2.6	检查配置语法的 config 命令	173
12.2.7	创建服务容器的 create 命令	174
12.2.8	清理项目的 down 命令	174
12.2.9	查看事件的 events 命令	175
12.2.10	进入服务的 exec 命令	176
12.2.11	杀死服务容器的 kill 命令	176
12.2.12	查看服务容器日志的 logs 命令	176
12.2.13	暂停服务容器的 pause 命令	177
12.2.14	查看服务容器端口状态的 port 命令	177
12.2.15	查看项目容器信息 ps 命令	177
12.2.16	拉取项目镜像的 pull 命令	178
12.2.17	推送项目镜像的 push 命令	179
12.2.18	重启服务容器的 restart 命令	179
12.2.19	删除项目容器的 rm 命令	179
12.2.20	执行一次性命令的 run 命令	180
12.2.21	设置服务容器数量的 scale 命令	182

12.2.22	启动服务容器的 start 命令	184
12.2.23	停止服务容器的 stop 命令	184
12.2.24	取消暂停的 unpause 命令	185
12.2.25	启动项目的 up 命令	185
12.3	Compose 配置文件	186
12.3.1	配置文件基础	186
12.3.2	基本配置	187
12.3.3	网络配置	196
12.3.4	配置扩展	197
12.4	Compose 实战	200
12.4.1	部署 Django	200
12.4.2	部署 Rails	202
12.4.3	部署 WordPress	203
12.5	本章小结	205
第 13 章	Web 服务器与应用	206
13.1	Apache 服务器	206
13.1.1	官方镜像	206
13.1.2	运行官方镜像	208
13.1.3	基于 Ubuntu 构建 Apache 镜像	209
13.1.4	基于 Alpine 构建 Apache 镜像	210
13.1.5	第三方优质镜像	211
13.2	Nginx 服务器	212
13.2.1	官方镜像介绍	212
13.2.2	运行官方镜像	212
13.2.3	构建 Nginx 镜像	214
13.2.4	第三方镜像推荐	216
13.3	Tomcat 服务器	216
13.3.1	官方镜像介绍	217
13.3.2	运行官方镜像	217
13.3.3	构建 Tomcat 镜像	218
13.4	其他 Web 服务器	220
13.4.1	Caddy 服务器	220
13.4.2	WebLogic 服务器	221
13.5	本章小结	223
第 14 章	数据库	224
14.1	MySQL 数据库	224
14.1.1	官方镜像的剖析与使用	224
14.1.2	构建自己的 MySQL 镜像	226
14.2	PostgreSQL 数据库	228
14.2.1	官方镜像的使用	228
14.2.2	官方镜像的剖析	230
14.2.3	构建 PostgreSQL 镜像	232
14.2.4	数据备份与导入	234
14.3	Redis 数据库	235

14.3.1	官方镜像的使用	235
14.3.2	构建自己的 Redis 镜像	236
14.4	MongoDB 数据库	238
14.4.1	官方镜像的使用	238
14.4.2	构建自己的 MongoDB 镜像	239
14.5	其他	240
14.5.1	在容器中使用 SQLite	240
14.5.2	构建自己的 MariaDB 镜像	241
14.5.3	使用 Docker 部署 Oracle XE 数据库	243
14.6	本章小结	244
第 15 章	编程语言	245
15.1	C/C++ 语言	245
15.1.1	官方镜像 library/gcc	245
15.1.2	基于 Alpine 构建 C/C++ 镜像	246
15.2	Golang 语言	247
15.2.1	官方镜像 library/golang	248
15.2.2	Beego 框架	250
15.2.3	自助 Git 服务——Gogs	250
15.2.4	基于 Alpine 构建 Golang 镜像	252
15.3	Java 语言	253
15.3.1	官方镜像 library/openjdk	253
15.3.2	基于 Alpine 构建 Java 镜像	254
15.3.3	Tomcat 服务器	255
15.3.4	下一代集成开发环境——Eclipse Che	256
15.4	JavaScript (Node.js) 语言	258
15.4.1	官方镜像 library/node	258
15.4.2	vue.js 框架	260
15.4.3	Express 框架	261
15.4.4	浏览器里的 IDE——Cloud9-IDE	262
15.4.5	基于 Alpine 构建 Node.js 镜像	264
15.5	PHP 语言	265
15.5.1	官方镜像 library/php	265
15.5.2	快速安装扩展	267
15.5.3	LNMP 环境组合	268
15.5.4	基于 Alpine 构建 PHP 镜像	270
15.5.5	自建私有云存储——ownCloud	271
15.5.6	Typecho 博客系统	273
15.6	Python 语言	275
15.6.1	官方镜像 library/python	275
15.6.2	Flask 框架	275
15.6.3	基于 Alpine 构建 Python 镜像	277
15.7	Swift 语言	277
15.7.1	构建 Swift 镜像	277
15.7.2	Kitura 框架	278

15.8 本章小结	280
第 16 章 Docker API 介绍	281
16.1 认识 Docker API	281
16.1.1 RESTful 介绍	281
16.1.2 开启 socket	282
16.1.3 使用 curl	283
16.1.4 使用 Postman	284
16.2 Docker Remote API 介绍	286
16.2.1 容器 API	286
16.2.2 镜像 API	293
16.3 其他 API	299
16.3.1 常用 API	300
16.3.2 Trusted Registry API 介绍	302
16.4 本章小结	304
第 17 章 私有仓库	305
17.1 Docker Registry 介绍	305
17.1.1 部署 Docker Registry	305
17.1.2 私有仓库的 push 与 pull	305
17.1.3 配置 Registry	306
17.1.4 添加 Docker Hub Mirror 功能	311
17.2 认证与前端	312
17.2.1 设置反代理	312
17.2.2 为私有仓库添加认证服务	313
17.2.3 为私有仓库添加可视化界面	315
17.3 企业级私有仓库 Harbor	316
17.3.1 Harbor 配置详解	316
17.3.2 配置 HTTPS	318
17.3.3 使用 Compose 启动 Harbor	320
17.4 私有仓库前端授权工具 Portus	321
17.4.1 一键部署 Portus	321
17.4.2 手动配置	321
17.4.3 启动 Portus	322
17.5 本章小结	323
第 18 章 集群网络	324
18.1 Swarm 集群	324
18.1.1 认识 Swarm	324
18.1.2 建立跨主机网络	328
18.1.3 在跨主机网络上部署应用	328
18.1.4 Swarm 集群负载	331
18.2 第三方网络管理工具	333
18.2.1 Weave 介绍	333
18.2.2 Flannel 介绍	335
18.2.3 Pipwork 介绍	335

18.3	第三方服务发现	336
18.3.1	Etd介绍	336
18.3.2	Consul介绍	337
18.4	第三方集群管理	337
18.4.1	Kubernetes介绍	337
18.4.2	Mesos Shipyard介绍	339
18.5	本章小结	339
第 19 章	Docker 安全	340
19.1	Docker 安全机制	340
19.1.1	Daemon 安全	340
19.1.2	容器与镜像安全	341
19.2	Docker 资源控制	342
19.2.1	限制 CPU	342
19.2.2	限制内存	343
19.2.3	限制 I/O	343
19.2.4	文件系统防护	344
19.2.5	镜像瘦身神器 Docker Slim	344
19.2.6	强制访问控制工具 SELinux 或 AppArmor	344
19.3	本章小结	345