

# 《Docker进阶与实战》

## 图书基本信息

书名：《Docker进阶与实战》

13位ISBN编号：9787111523393

出版时间：2016-1-5

作者：华为Docker实践小组

页数：264

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu000.com](http://www.tushu000.com)

# 《Docker进阶与实战》

## 内容概要

本书由一个真正钻研容器技术的团队写作，他们不仅仅是在使用Docker，更多的是在探索容器的未来之路，希望把“代码与产品，理论与实践”完美结合。本书内容从Docker的来源、镜像、仓库、安全、网络、卷存储，到生态、测试及社区贡献都有涉猎。无论你是入门级，还是已经有了较深的功底，这本书都会带你踏上新的台阶——正所谓“进阶”。

# 《Docker进阶与实战》

## 作者简介

华为Docker实践小组，致力于容器技术探索，结合华为CT/IT/Mobile等应用场景，重点在容器引擎、仓库、网络和卷存储等多领域攻关；同时该团队遵循开源精神，将开发过程中的诸多价值特性同步贡献到Docker社区中，目前在Docker社区贡献中全球排名稳居前三，国内排名第一。

## 书籍目录

序

前言

第1章 Docker简介 1

1.1 引言 1

1.1.1 Docker的历史和发展 1

1.1.2 Docker的架构介绍 2

1.2 功能和组件 3

1.2.1 Docker客户端 3

1.2.2 Docker daemon 3

1.2.3 Docker容器 3

1.2.4 Docker镜像 4

1.2.5 Registry 4

1.3 安装和使用 5

1.3.1 Docker的安装 5

1.3.2 Docker的使用 6

1.4 概念澄清 8

1.4.1 Docker在LXC基础上做了什么工作 8

1.4.2 Docker容器和虚拟机之间有什么不同 9

1.5 本章小结 10

第2章 关于容器技术 11

2.1 容器技术的前世今生 11

2.1.1 关于容器技术 11

2.1.2 容器技术的历史 12

2.2 一分钟理解容器 14

2.2.1 容器的组成 14

2.2.2 容器的创建原理 15

2.3 Cgroup介绍 16

2.3.1 Cgroup是什么 16

2.3.2 Cgroup的接口和使用 17

2.3.3 Cgroup子系统介绍 18

2.4 Namespace介绍 20

2.4.1 Namespace是什么 20

2.4.2 Namespace的接口和使用 21

2.4.3 各个Namespace介绍 22

2.5 容器造就Docker 26

2.6 本章小结 27

第3章 理解Docker镜像 28

3.1 Docker image概念介绍 28

3.2 使用Docker image 29

3.2.1 列出本机的镜像 29

3.2.2 Build：创建一个镜像 31

3.2.3 Ship：传输一个镜像 32

3.2.4 Run：以image为模板启动一个容器 32

3.3 Docker image的组织结构 33

3.3.1 数据的内容 33

3.3.2 数据的组织 35

3.4 Docker image扩展知识 37

3.4.1	联合挂载	37
3.4.2	写时复制	37
3.4.3	Git式管理	40
3.5	本章小结	40
第4章	仓库进阶	41
4.1	什么是仓库	41
4.1.1	仓库的组成	41
4.1.2	仓库镜像	42
4.2	再看Docker Hub	43
4.2.1	Docker Hub的优点	43
4.2.2	网页分布	44
4.2.3	账户管理系统	46
4.3	仓库服务	49
4.3.1	Registry功能和架构	49
4.3.2	Registry API	50
4.3.3	Registry API传输过程分析	53
4.3.4	鉴权机制	57
4.4	部署私有仓库	61
4.4.1	运行私有服务	61
4.4.2	构建反向代理	61
4.5	Index及仓库高级功能	64
4.5.1	Index的作用和组成	64
4.5.2	控制单元	65
4.5.3	鉴权模块	66
4.5.4	数据库	67
4.5.5	高级功能	68
4.5.6	Index客户端界面	69
4.6	本章小结	69
第5章	Docker网络	71
5.1	Docker网络现状	71
5.2	基本网络配置	73
5.2.1	Docker网络初探	73
5.2.2	Docker网络相关参数	80
5.3	高级网络配置	85
5.3.1	容器跨主机多子网方案	85
5.3.2	容器跨主机多子网配置方法	86
5.4	网络解决方案进阶	90
5.4.1	Weave	90
5.4.2	Flannel	91
5.4.3	SocketPlane	94
5.5	本章小结	98
第6章	容器卷管理	99
6.1	Docker卷管理基础	99
6.1.1	增加新数据卷	99
6.1.2	将主机目录挂载为数据卷	100
6.1.3	创建数据卷容器	100
6.1.4	数据卷的备份、转储和迁移	101
6.1.5	Docker卷管理的问题	101
6.2	使用卷插件	102

6.2.1	卷插件简介	102
6.2.2	卷插件的使用	102
6.3	卷插件剖析	103
6.3.1	卷插件工作原理	104
6.3.2	卷插件API接口	105
6.3.3	插件发现机制	105
6.4	已有的卷插件	106
6.5	本章小结	107
第7章	Docker API	108
7.1	关于 Docker API	108
7.1.1	REST 简介	108
7.1.2	Docker API 初探	109
7.1.3	Docker API 种类	110
7.2	RESTful API应用示例	110
7.2.1	前期准备	111
7.2.2	Docker API的基本示例	116
7.3	API的高级应用	123
7.3.1	场景概述	123
7.3.2	场景实现	124
7.4	本章小结	131
第8章	Docker安全	132
8.1	深入理解Docker的安全	132
8.1.1	Docker的安全性	132
8.1.2	Docker容器的安全性	132
8.2	安全策略	133
8.2.1	Cgroup	133
8.2.2	ulimit	135
8.2.3	容器组网	135
8.2.4	容器+全虚拟化	136
8.2.5	镜像签名	136
8.2.6	日志审计	136
8.2.7	监控	137
8.2.8	文件系统级防护	137
8.2.9	capability	137
8.2.10	SELinux	138
8.2.11	AppArmor	142
8.2.12	Seccomp	144
8.2.13	grsecurity	145
8.2.14	几个与Docker安全相关的项目	146
8.3	安全加固	146
8.3.1	主机逃逸	147
8.3.2	安全加固之capability	150
8.3.3	安全加固之SELinux	151
8.3.4	安全加固之AppArmor	152
8.4	Docker安全遗留问题	153
8.4.1	User Namespace	153
8.4.2	非root运行Docker daemon	153
8.4.3	Docker热升级	153
8.4.4	磁盘限额	154

8.4.5	网络I/O	154
8.5	本章小结	154
第9章	Libcontainer简介	155
9.1	引擎的引擎	155
9.1.1	关于容器的引擎	155
9.1.2	对引擎的理解	156
9.2	Libcontainer的技术原理	157
9.2.1	为容器创建新的命名空间	158
9.2.2	为容器创建新的Cgroup	159
9.2.3	创建一个新的容器	160
9.2.4	Libcontainer的功能	164
9.3	关于runC	166
9.3.1	runC和Libcontainer的关系	166
9.3.2	runC的工作原理	167
9.3.3	runC的未来	168
9.4	本章小结	169
第10章	Docker实战	170
10.1	Dockerfile简介	170
10.1.1	一个简单的例子	171
10.1.2	Dockerfile指令	171
10.1.3	再谈Docker镜像制作	173
10.2	基于Docker的Web应用和发布	174
10.2.1	选择基础镜像	174
10.2.2	制作HTTPS服务器镜像	175
10.2.3	将Web源码导入Tomcat镜像中	178
10.2.4	部署与验证	179
10.3	为Web站点添加后台服务	180
10.3.1	代码组织结构	180
10.3.2	组件镜像制作过程	183
10.3.3	整体部署服务	183
10.4	本章小结	184
第11章	Docker集群管理	185
11.1	Compose	185
11.1.1	Compose概述	185
11.1.2	Compose配置简介	186
11.2	Machine	187
11.2.1	Machine概述	187
11.2.2	Machine的基本概念及运行流程	188
11.3	Swarm	188
11.3.1	Swarm概述	188
11.3.2	Swarm内部架构	189
11.4	Docker在OpenStack上的集群实战	190
11.5	本章小结	196
第12章	Docker生态圈	197
12.1	Docker生态圈介绍	197
12.2	重点项目介绍	198
12.2.1	编排	198
12.2.2	容器操作系统	203
12.2.3	PaaS平台	206

12.3	生态圈的未来发展	208
12.3.1	Docker公司的发展和完善方向	208
12.3.2	OCI组织	209
12.3.3	生态圈格局的分化和发展	210
12.4	本章小章	211
第13章	Docker测试	212
13.1	Docker自身测试	212
13.1.1	Docker自身的测试框架	212
13.1.2	运行Docker测试	213
13.1.3	在容器中手动运行测试用例	215
13.1.4	运行集成测试中单个或多个测试用例	215
13.1.5	Docker测试用例集介绍	216
13.1.6	Docker测试需要改进的方面	217
13.1.7	构建和测试文档	217
13.1.8	其他Docker测试套	218
13.2	Docker技术在测试中的应用	220
13.2.1	Docker对测试的革命性影响	221
13.2.2	Docker技术适用范围	222
13.2.3	Jenkins+Docker自动化环境配置	223
13.3	本章小结	229
第14章	参与Docker开发	230
14.1	改进Docker	230
14.1.1	报告问题	230
14.1.2	提交补丁	231
14.2	编译自己的Docker	235
14.2.1	使用make工具编译	235
14.2.2	手动启动容器编译	235
14.2.3	编译动态链接的可执行文件	237
14.2.4	跑测试用例及小结	237
14.3	开源的沟通和交流	238
14.3.1	Docker沟通和交流的途径	238
14.3.2	开源沟通和交流的建议	238
14.4	Docker项目的组织架构	239
14.4.1	管理模型	239
14.4.2	组织架构	240
14.5	本章小章	242
附录A	FAQ	243
附录B	常用Dockerfile	247
附录C	Docker信息获取渠道	250



# 《Docker进阶与实战》

## 精彩短评

- 1、面相中级用户的书，虽不全面，但聊胜于无吧
- 2、“授之以鱼不如授之以渔”，这本书对docker的介绍还是比较通透的，既知其然又知所以然~~~
- 3、内容偏向中高级读者，讲解的很细致，想深入研究下Docker的原理的话这本书还是很值得一读的。
- 4、Docker的书现在已经不少，但内容往往过于初级，这本不错。
- 5、对docker的深入讲解，有好多神奇的命令
- 6、看的几本docker书里，相对最好的一本了。更多介绍了原理与设计，而不是泛泛而谈，贴代码也很节制。Docker本身没有太多可讲，毕竟都是使用内核提供的虚拟化技术，但是其庞大的生态系统错综复杂。不可能全部介绍，并且很多工具，只有用到时才有必要去了解。总体来说，这是一本优秀的“第二本Docker书”。
- 7、泛泛而谈，可以作为了解Docker相关技术的Index
- 8、3.5\*
- 9、基本上没有什么用。。还卖的那么贵。。最tmd让人不爽的是大篇幅的终端输出copy，最后参与docker开发也写了一章。。。。

## 章节试读

### 1、《Docker进阶与实战》的笔记-第80页

实现环境需要三台机器consul-server, app-server, db-server。因此选择在Ubuntu机器上通过docker-machine创建虚拟机来完成。

首先创建host1：docker-machine create -d virtualbox consul-server

执行下面的代码在consul-server上启动consul server：docker \$(docker-machine config consul-server) run -d -p "8500:8500" -h "consul" progrium/consul -server -bootstrap

执行下面的命令创建app-server：docker-machine create -d virtualbox \

--engine-opt="cluster-store=consul://\$(docker-machine ip consul-server):8500" \

--engine-opt="cluster-advertise=eth1:0" \

app-server 执行下面的命令创建db-server：docker-machine create -d virtualbox \

--engine-opt="cluster-store=consul://\$(docker-machine ip consul-server):8500" \

--engine-opt="cluster-advertise=eth1:0" \

db-server overlay网络要管理app-server db-server两台主机，所以，首先在app-server上建立overlay

network "dev"：docker \$(docker-machine config app-server) network create --driver overlay

--subnet=10.0.9.0/24 dev用下面的命令启动两个容器，他们都运行于app-server，不过连接到dev网络中

了:docker \$(docker-machine config app-server) run --net dev -h app-worker-1 --name app-worker-1 -itd

ubuntu:14.04 bash

docker \$(docker-machine config app-server) run --net dev -h app-worker-2 --name app-worker-2 -itd

ubuntu:14.04 bash 用下面的命令启动两个容器，他们都运行于app-server，不过连接到dev网络中

了:docker \$(docker-machine config db-server) run --net dev -h db-worker-1 --name db-worker-1 -itd

ubuntu:14.04 bash

docker \$(docker-machine config db-server) run --net dev -h db-worker-2 --name db-worker-2 -itd ubuntu:14.04

bash 实验环境创建完毕。

用下面命令进入到app-worker-1容器中：docker \$(docker-machine config app-server) exec -it app-worker-1

bash

然后可以尝试这个容器到app-worker-2, db-worker-1, db-worker-2的连通性。当然了，连通性肯定没有问题。

接下来，可以看一下网络的详细情况了。首先ssh登陆app-server，docker-machine ssh app-server

在这个主机中，Docker的沙箱（Network Namespace）在/var/run/docker/netns/目录下，可以通过符号连接

把它加到/var/run/netns目录下，这样就可以利用ip命令查看具体信息了。In -s

/var/run/docker/netns/1-164825fbdf /var/run/netns/1-164825fbdf

ip netns exec 1-164825fbdf ip addr show

可以看到如下的输入：

1: lo: xxxxxxxxxxxxxx

2: br0: xxxxxxxxxxxxxx

10: vxlan1: xxxxxxxxxxxxxx

12: veth1@if11: xxxxxxxxxxxxxx

17: veth3@if16: xxxxxxxxxxxxxx

对照CNM模型，可以知道vxlan1是垮主机通讯的关键，veth1和veth2分别是app-worker-1 app-worker-2接入docker bridge的虚拟网络接口。

### 2、《Docker进阶与实战》的笔记-第127页

调用与Registry相关的API时需要一个头域，X-Registry-Auth，包含username，password，email\_address，server\_address。

```
> XRA=$(echo {"username": $USERNAME, "password": $PASSWORD, "email": $EMAIL_ADDRESS, "serveraddress": $SERVER_ADDRESS} | base64)
(以上字符串传给echo前需要讲双引号转义)
```

```
> echo $XRA
```

```
> curl -XPOST -H "X-Registry-Auth: $XRA"
http://DOCKER_ENGINE/images/REGISTRY_SERVER/IMAGE_TAG
```

### 3、《Docker进阶与实战》的笔记-第72页

CNM模型

## 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:[www.tushu000.com](http://www.tushu000.com)