

Docker 源码分析

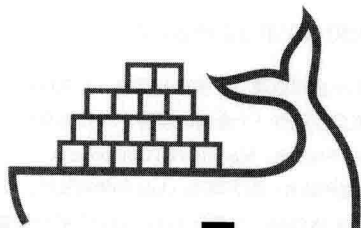
THE SOURCE CODE ANALYSIS OF DOCKER

孙宏亮 著

- 国内首部Docker源码分析著作
- 从源码角度全面解析Docker设计与实现
- 填补Docker理论与实践之间的鸿沟



机械工业出版社
China Machine Press



Docker 源码分析

THE SOURCE CODE ANALYSIS OF DOCKER

孙宏亮 著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Docker 源码分析 / 孙宏亮著. —北京: 机械工业出版社, 2015.8 (2015.9 重印)
(容器技术系列)

ISBN 978-7-111-51072-7

I. D… II. 孙… III. Linux 操作系统—程序设计 IV. TP316.89

中国版本图书馆 CIP 数据核字 (2015) 第 174926 号

本书是一本引导读者深入了解 Docker 实现原理的技术普及读物, 主要目标是通过对 Docker 架构和源代码的详细讲解和解剖, 帮助读者对 Docker 的底层实现有一个全面的理解。本书的内容主要集中于三部分: Docker 的架构、Docker 的模块, 以及 Docker 的三驾马车 Swarm、Machine 和 Compose。

第一部分 (第 1 章) 主要从宏观的角度和读者一起领略 Docker 的架构设计, 并初步介绍架构中各模块的职责。

第二部分 (第 2 ~ 14 章) 是本书的主体部分, 主要针对 Docker 中多个重要的模块进行具体深入分析, 包括 DockerClient、DockerDaemon、DockerServer、Docker 网络、Docker 镜像、Docker 容器等。读者可以发现, Docker 的模块之间耦合度非常低, 很适合循序渐进, 层层深入。第 2 ~ 8 章主要从 Docker 软件的架构入手, 勾勒骨架; 第 9 ~ 11 章集中于 Docker 镜像技术, 夯实基础; 第 12 ~ 14 章则进一步分析 Docker 容器的始末, 阐述本质。

第三部分 (第 15 ~ 17 章) 介绍 Docker 生态三驾马车——Swarm、Machine 和 Compose。Docker 拥有强大的单机能力, 三驾马车可以很好地补充 Docker 的跨主机能力以及部署能力。读者可以通过这几章感受 Docker 生态中其他功能强大的软件。

Docker 源码分析

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: 陈佳媛 谢晓芳

责任校对: 董纪丽

印刷: 北京市荣盛彩色印刷有限公司

版次: 2015 年 9 月第 1 版第 2 次印刷

开本: 186mm × 240mm 1/16

印张: 16.5

书号: ISBN 978-7-111-51072-7

定价: 59.00 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

像谷歌一样部署自己的应用，这是很多软件工程师的梦想。Docker 的目标是圆很多人的梦。自从 InfoQ 推出 Docker 系列文章，作为操作系统课程教师的我一直在学习并关注 Docker 的茁壮成长。

当我发现这上面刊登的“Docker 源码分析”系列文章的作者居然是我们课程组的研究生助教孙宏亮时，惊喜之情溢于言表。宏亮对 Docker 的理解十分深刻，他本人是 Docker 的积极拥护者、倡导者和贡献者。他在研究生毕业以后投身到了创业公司 DaoCloud，去为 Docker 的梦想开创美好的未来。

最近，我又欣喜地发现，这系列文章以及后续章节即将正式出版成书，有机会同更多的 Docker 用户、开发者、学习者见面。本书通过分析解读 Docker 源码，让读者了解 Docker 的内部结构和实现，以便更好地使用 Docker。该书的内容组织深入浅出，表述准确到位，有大量流程图和代码片段帮助读者理解 Docker 各个功能模块的流程，是学习 Docker 开源系统的良师益友。

——寿黎旦，浙江大学计算机学院教授

近年来，Docker 迅速风靡了云计算世界，但是专门针对 Docker 的技术实现进行深入分析的文章却相对较少。这一方面由于 Docker 技术变化很快，源码分析很快会跟不上版本发展；另一方面，对源代码的解析，需要对整个 Docker 设计具备全局的视角，才能深入浅出地找到源码中的脉络。

宏亮的这本《Docker 源码分析》，恰如其时的出现，弥补了这个空白，对于希望参与到 Docker 社区、参与代码贡献或构建自己的 Docker 应用环境的读者来说，应是一本案头必备书籍。

——王兴宇，《Linux 中国》创始人

在崇尚源码至上的工程师文化里，文档介绍、发布会材料都是苍白的，唯有研读源码，才能深刻理解软件背后的原理。与其他所有软件一样，读源码并不是学习 Docker 最快的途

径，但是如果有人通读源码后给出了详细分析，你就可以轻松地站在巨人的肩膀上。

很高兴看到国内这么快就出版源码分析的书籍。对于所有想在 Docker 方面进阶和想晋升为高端用户的读者，都值得阅读本书，也希望通过《Docker 源码分析》一书，可以诞生更多的社区贡献者，共同推动 Docker 的发展。

——黄强，华为 Docker Committer

值得自豪的是，就在 Docker 蓬勃发展之际，第一本详尽剖析 Docker 源码的著作出自国人之手。

本书在每章宏观的流程梳理背后都伴随有更加细致深入的源码分析。无论读者是只想了解使用 Docker，还是抱着深入理解 Docker 并参与社区开发、二次开发的心态，本书都值得一读。

——胡科平，华为 Docker Committer

这本书从源码的角度对 Docker 的实现原理进行了深入的探讨和细腻的讲解，将当前热门的容器技术的背后机理讲解得深入浅出明白透彻。无论是 Docker 的用户还是开发者，通过阅读本书都可以对 Docker 有更深刻的理解，能够更好地使用或者开发 Docker。

——雷继棠，华为 Docker Committer

Docker 已经是一个成长 2 年时间的云计算技术，它正在以惊人的速度在全球范围内扩张自己的“疆土”。我作为 Docker 中国区的开发者，非常希望能看到有一本书详细地告诉我，Docker 的每一个细节是如何实现的。所以当我在 InfoQ 上看到宏亮的“Docker 源码分析”专栏时眼前一亮。今天，它终于汇编成书摆在你我的眼前。我希望你能在这本书上学到更多 Docker 技术的精髓思想，在实战 Docker 技术时可以运用自如！

——肖德时，数人科技 CTO

我家里的书柜中至今仍然保留着一本《Linux 内核完全注释》，它伴随我很长一段时间，使我受益匪浅。10 年之后，当我拿到宏亮的《Docker 源代码分析》草稿，昨日又重现。这本书无论是对学习 Docker，还是掌握 Go 语言，都是非常好的一手资源。雷锋不常有，大家要支持！

——赵鹏，VisualOps 创始人

Foreword 序

三年前，我在 VMware 负责 Cloud Foundry 这一款开源 PaaS 产品在中国的开发者社区和生态系统建设工作。当时关于 Cloud Foundry 的中文文档非常少，更不用说有深度的技术干货了，所以当 CSDN 上出现了一个专门从底层模块和源码的角度，对 Cloud Foundry 做深度剖析的系列博客文章时，一下子就引起了我的注意。

经过一番“人肉搜索”，我非常惊奇的发现，这一系列干货的作者孙宏亮，竟然是浙江大学计算系的一年级研究生。当时，VMware 跟浙江大学在 Cloud Foundry 方面有比较深入的合作，浙大计算机系的 SEL 实验室，投入了精锐的师资力量，从事分布式系统和新一代 PaaS 的研究工作。宏亮初入浙江大学，就在这样的氛围下开始了他的研究生求学生涯，应当说是非常幸运的。

宏亮在 CSDN 上的系列文章，主打“源码分析”，这正是当时开源社区内比较缺少的内容。提笔写“源码分析”，需要一定的勇气，阅读源码需要耗费大量的精力，需要从数十万行代码中整理出清晰的逻辑，从中抽丝拨茧、概括精华，这是一份非常辛苦的工作。而且，分析源码也需要一些“挑战权威”的精神，不仅仅是简单的代码解析，更需要提炼出自己的观点，甚至敢于发现和纠正一些已有的问题。

在源码分析方面，宏亮充分体现了“初生牛犊不怕虎”的精神，非常详细地剖析了当时 Cloud Foundry 的几个主要模块，条理清晰，技术分析准确到位。宏亮这一系列文章帮助了包括一线互联网公司在内的许多企业了解、认识和最终使用 Cloud Foundry，宏亮也借此奠定了他在 PaaS 社区的“江湖地位”。

从去年开始，Docker 的热潮开始波及中国开发者社区。我有幸跟宏亮一起在 CSDN 主办的第一届 Container 技术大会上发表联合主题演讲，向来宾介绍 Cloud Foundry 内部的容器技术实现，以及对 Docker 的一些展望。那次大会是一个很重要的里程碑，在那之后，宏亮开始深入研究 Docker 的底层实现，并在 InfoQ 连载“Docker 源码分析”系列文章。

对 PaaS 平台的研究越深入，越能够发现和体会 Docker 对开发和运维的价值。如果说当初的 Cloud Foundry 模块和源码分析文章，是读研期间的学习笔记，那这次宏亮的《Docker 源码分析》，则是一个经过了深思熟虑、系统性、结构化的工作。Docker 开源项

目发展速度非常快，这次在文章连载内容的基础上出书，为了保证内容的准确性和时效性，宏亮补充了大量 Docker 最新项目的内容，特别是对 Swarm、Machine 和 Compose 这三个模块的开发进展做了紧密的追踪。

这是一本从架构和代码角度讲解 Docker 底层实现的技术图书，我从连载第一篇开始就对这个系列的文章保持了紧密的关注，也目睹了宏亮在后期整理加工成书过程中的辛勤努力。在《Docker 源码分析》成书付梓出版之际，非常幸运，能够为宏亮写着一篇推荐序。这本书非常适合以下三类读者学习和阅读。

□ 希望以 Docker 容器交付软件的程序员。

Docker 是未来互联网软件的交付件，这件事随着 OCP 标准的制定，正在逐渐成为事实。程序员和运维工程师都需要了解 Docker 的工作方式，尤其是 Docker 镜像的结构，软件通过 Dockerfile 打包时的优化方式等，这些内容在本书中都有非常详细的阐述。

□ Docker 化云计算平台的建设者和维护者。

Docker 公司在今年的全球开发者大会上提出了“Production Ready”的口号，有越来越多的互联网公司和传统企业采用 Docker 来构建开发、测试和运维平台。Docker 在网络、存储、安全等领域的细节，是平台建设者和维护者必须深入了解的部分，这些领域还在不断变化，新的项目也层出不穷，但本书对网络、存储和安全的基本知识和概念，做了非常清晰的介绍，也深入到了底层实现的代码。

□ Go 语言程序员。

即使不在项目中使用 Docker，本书也能够为 Go 语言程序员带来帮助。Docker 项目中大量采用了 Go 语言，尤其是在处理并发场景时，Docker 对 Go 语言的运用可谓出神入化。本书可以帮助 Go 语言程序员亲身体验特大型项目中 Go 语言的威力，以及实战场景中 Golang 模式和功能的用法。

最后，预祝宏亮在 Docker 的学习和工作中再创佳绩，也希望读者可以从本书收获知识，开阔眼界。

喻 勇

2015 年 7 月 13 日

Docker 是什么

Docker 从 2013 年诞生，短短两年时间就在全球 IT 技术圈内迅速走红，实乃技术圈内不可忽视的一阵飓风。然而，Docker 是什么，Docker 带来了什么？

Docker 官方如此描述 Docker：“Build, Ship, Run. An open platform for distributed applications for developers and sysadmins”。换言之，Docker 为开发者与系统管理者提供了分布式应用的开放平台，从而可以便捷地构建、迁移与运行分布式应用。

多年来，IT 行业中开发与运维一直是两个界限清晰的词。开发工程师专门从事软件的开发工作，最终交付软件代码；运维工程师则部署前者交付的软件，并接管软件的运行与管理。然而，在长时间的实践过程中，开发与运维分离的方式难免存在弊病，两者职责的过分清晰导致软件效率的降低。随着分布式系统的流行，系统规模越来越大，软件越来越复杂，系统环境配置暴露的问题层出不穷。究其缘由，还是因为开发人员缺少软件运行环境的认知，而运维人员对软件逻辑所知甚少。在这样的背景下，DevOps 横空出世，提倡开发与运维不可分割，协调并进。

Docker 无疑是 DevOps 大潮中最具实践价值的**不二法宝**。Docker 从 Linux 内核的角度出发，属于轻量级虚拟化技术，有能力秒级提供应用隔离环境，完成云计算时代分布式应用的第一需求“隔离”。另外，Docker 的镜像技术利用联合文件系统的优势，自下至上打包系统软件、系统环境以及软件程序，将运行环境与应用程序灵活地结合，快速运行 Docker 化的应用程序。同时，可读性极强的 Dockerfile，极大地简化镜像的复杂性，并为镜像的转移与重新构建提供了可能性。

Docker 提供轻便的资源分配方式，解决应用运行与系统环境的依赖，弥合应用跨节点迁移的鸿沟，种种特性都表明 Docker 几乎就是为“云计算”而生的。如今，Docker 社区不断扩大并健康发展，多家国际 IT 巨头也纷纷宣布支持 Docker，这一切更是让全球 IT 人士对 Docker 的未来充满信心。

本书的内容

本书是一本引导读者了解 Docker 实现原理的技术普及书。笔者一直坚信，了解软件或者系统最直接、最透彻的方式就是研读它们的源码。“源码即文档”也是鼓励开发者能更多地从源码的角度去学习软件或系统的本质。

本书的内容主要集中于 3 个部分：Docker 的架构，Docker 的模块，Docker 的三驾马车 Swarm、Machine 以及 Compose。

第一部分，主要从宏观的角度和读者一起领略 Docker 的架构设计，并初步介绍架构中各模块的职责。

第二部分是本书的主体部分，主要针对 Docker 中多个重要的模块进行具体深入分析，包括：Docker Client、Docker Daemon、Docker Server、Docker 网络、Docker 镜像、Docker 容器等。读者可以发现，Docker 的模块之间耦合度非常低，很适合循序渐进，层层深入。第 2 章至第 8 章主要从 Docker 软件的架构入手，勾勒骨架；第 9 章至第 11 章重点讨论 Docker 镜像技术，夯实基础；第 12 章至第 14 章则进一步分析 Docker 容器的始末，阐述本质。

第三部分介绍 Docker 生态三驾马车 Swarm、Machine、Compose。Docker 拥有强大的单机能力，三驾马车可以很好地补充 Docker 的跨主机能力以及部署能力。读者可以通过第 15 章至第 17 章感受 Docker 生态圈中其他功能强大的软件。

希望本书能够让读者最大化地感受 Docker 的神奇与魅力。

关于勘误

由于时间与水平都比较有限，因此本书难免会存在一些纰漏和错误。如果读者发现了问题，请及时与我联系。我也会在本书后续的版本中加以改正。我的邮箱是：allen.sun@daocloud.io。我非常希望和大家一起学习与讨论 Docker，并共同推动 Docker 在社区的发展。

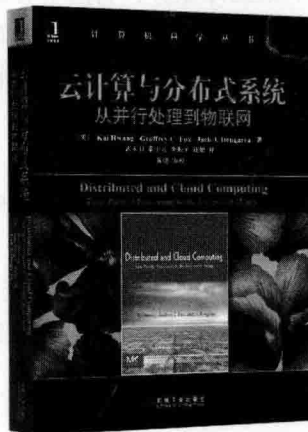
致谢

最后，向本书编写过程中给予我巨大帮助的人们表示最诚挚的感谢。感谢我的父母，没有他们的鼓励和支持，此书不可能在如此短的时间内完成。感谢我的母校浙江大学以及 SEL 实验室的老师与同学们，是他们在求学过程中给予无私的指引与帮助。感谢我的同事熊中祥，是他在本书编写过程中提出了很多宝贵的建议，尤其在 Machine 和 Compose 部分。感谢我的同事喻勇和冯钊，他们为本书的编写做了很多沟通与协调工作。最后，还要感谢华章公司的编辑们，她们认真细致的工作，使本书以完美的形式展现给各位读者。

孙宏亮

2015 年 6 月

推荐阅读



云计算：概念、技术与架构

作者：Thomas Erl 等 译者：龚奕利 等 ISBN：978-7-111-46134-0 定价：69.00元

“我读过Thomas Erl写的每一本书，云计算这本书是他的又一部杰作，再次证明了Thomas Erl选择最复杂的主题却以一种符合逻辑而且易懂的方式提供关键核心概念和技术信息的罕见能力。”

——Melanie A. Allison, Integrated Consulting Services

在本书中，世界上最畅销的IT书籍作者之一Thomas Erl联合云计算专家和研究者，详细分析了业已证明的、成熟的云计算技术和实践，并将其组织成一系列定义准确的概念、模型、技术机制和技术架构，所有这些都是以工业为中心但是与厂商无关的。

本书理论与实践并重，重点放在主流云计算平台和解决方案的结构和基础上。除了以技术为中心的内容以外，还包括以商业为中心的模型和标准，以便读者对基于云的IT资源进行经济评估，把它们与传统企业内部的IT资源进行比较。此外，本书提供了一些用来计算与SLA相关的服务质量的模板和公式，还给出了大量的SaaS、PaaS和IaaS交付模型。

本书包括超过260幅图、29个架构模型和20种机制，是一本不可或缺的指导书，是对云计算技术的详细解读。

云计算与分布式系统：从并行处理到物联网

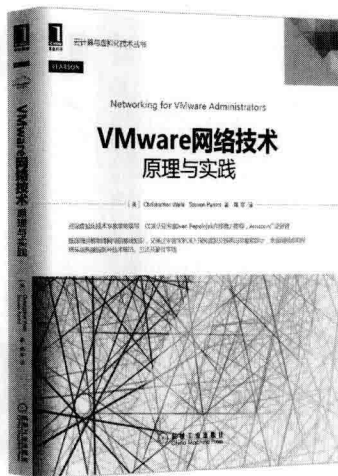
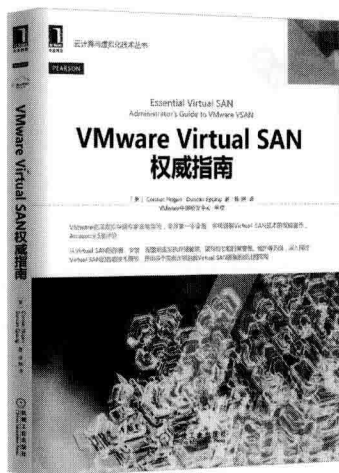
作者：Kai Hwang 等 译者：武永卫 等 ISBN：978-7-111-41065-2 定价：85.00元

“本书是一本全面而新颖的教材，内容覆盖高性能计算、分布式与云计算、虚拟化和网格计算。作者将应用与技术趋势相结合，揭示了计算的未来发展。无论是对在校学生还是经验丰富的实践者，本书都是一本优秀的读物。”

——Thomas J. Hacker, 普度大学

本书是一本完整讲述云计算与分布式系统基本理论及其应用的教材。书中从现代分布式模型概述开始，介绍了并行、分布式与云计算系统的设计原理、系统体系结构和创新应用，并通过开源应用和商业应用例子，阐述了如何为科研、电子商务、社会网络和超级计算等创建高性能、可扩展的、可靠的系统。

推荐阅读



VMware Virtual SAN权威指南

作者：(美) Cormac Hogan 等 ISBN: 978-7-111-48023-5 定价: 59.00元

不论您是虚拟化新手，还是存储专家，这本书是有关VMware Virtual SAN最权威的解读，是实现软件定义存储最有效的指南。

——任道远，VMware中国研发中心总经理

VMware资深虚拟存储专家亲笔撰写，全球第一本全面、系统讲解Virtual SAN技术的权威著作，Amazon全5星评价。从Virtual SAN的部署、安装、配置到虚拟机存储管理、架构细节和日常管理、维护等方面，深入探讨Virtual SAN的各项技术细节，并用多个实例详细讲解Virtual SAN群集的设计和实现。

本书专为管理员、咨询师和架构师所著，在书中Cormac Hogan和Duncan Epping既介绍了Virtual SAN如何实现基于对象的存储和策略平台，这些功能简化了虚拟机存储的放置，还介绍了Virtual SAN如何与vSphere协同工作，大幅提高系统弹性、存储横向扩展和QoS控制的能力。

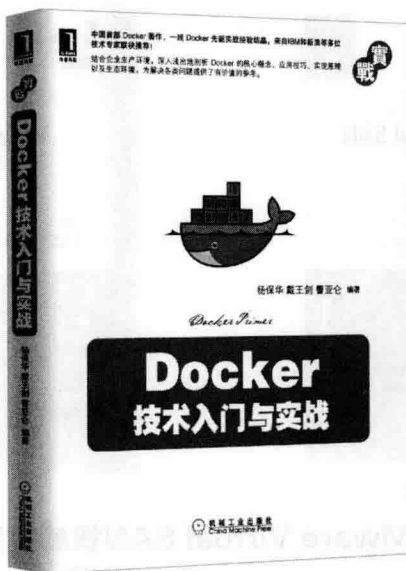
VMware网络技术：原理与实践

作者：(美) Christopher Wahl 等 ISBN: 978-7-111-47987-1 定价: 59.00元

资深虚拟化技术专家亲笔撰写，CCIE认证专家Ivan Pepelnjak作序鼎力推荐，Amazon广泛好评
既详细讲解物理网络的基础知识，又通过丰富实例深入探究虚拟交换机的功能和设计，
全面阐释虚拟网络环境构建的各种技术细节、方法及最佳实践

本书针对VMware专业人员，阐述了现代网络的核心概念，并介绍了如何在虚拟网络环境设计、配置和故障检修中应用这些概念。作者凭借其在虚拟化项目实施方面的丰富经验，从网络模型、常见网络层次的介绍开始，由浅入深地介绍了现代网络的基本概念，并自然地过渡到虚拟交换等虚拟化环境中与物理网络最为关联的部分，最后扩展到实际的设计用例，详细介绍了不同实用场景、不同的硬件配置下，虚拟化环境构建的考虑因素和具体实施方案。

推荐阅读



Docker 技术入门与实战

作者：杨保华 等 书号：978-7-111-48852-1 定价：79.00元

**中国首部 Docker 著作，一线 Docker 先驱实战经验结晶，
来自 IBM 和新浪等多位技术专家联袂推荐！
结合企业生产环境，深入浅出地剖析 Docker 的核心概念、应用技巧、
实现原理以及生态环境，为解决各类问题提供了有价值的参考。**

本书作者之一杨保华博士在加入 IBM 之后，一直从事云计算与软件定义网络领域的相关解决方案和核心技术的研发，热心关注 OpenStack、Docker 等开源社区，热衷使用开源技术，并积极参与开源社区的讨论、积极提交代码。这使得他既能从宏观上准确把握 Docker 技术在整个云计算产业中的定位，又能从微观上清晰理解技术人员所渴望获知的核心之处。

—— 刘天成，IBM 中国研究院云计算运维技术研究组经理

好的 IT 技术总是迅速“火爆”，Docker 就是这样。好像忽然之间，在企业一线工作的毕业生们都在谈论 Docker。在 IT 云化的今天，系统的规模和复杂性，呼唤着标准化的构件和自动化的管理，Docker 正是这种强烈需求的产物之一。这本书很及时，相信会成为 IT 工程师的宝典。

—— 李军，清华大学信息技术研究院院长

赞誉
序
前言

第 1 章 Docker 架构 1

- 1.1 引言 1
- 1.2 Docker 总架构图 2
- 1.3 Docker 各模块功能与实现分析 3
 - 1.3.1 Docker Client 4
 - 1.3.2 Docker Daemon 4
 - 1.3.3 Docker Registry 6
 - 1.3.4 Graph 7
 - 1.3.5 Driver 7
 - 1.3.6 libcontainer 10
 - 1.3.7 Docker Container 10
- 1.4 Docker 运行案例分析 11
 - 1.4.1 docker pull 11
 - 1.4.2 docker run 12
- 1.5 总结 14

第 2 章 Docker Client 创建与命令执行 15

- 2.1 引言 15
- 2.2 创建 Docker Client 16
 - 2.2.1 Docker 命令的 flag 参数解析 17

- 2.2.2 处理 flag 信息并收集 Docker Client 的配置信息 20
- 2.2.3 如何创建 Docker Client 22
- 2.3 Docker 命令执行 24
 - 2.3.1 Docker Client 解析请求命令 24
 - 2.3.2 Docker Client 执行请求命令 25
- 2.4 总结 27

第 3 章 启动 Docker Daemon 28

- 3.1 引言 28
- 3.2 Docker Daemon 的启动流程 29
- 3.3 mainDaemon() 的具体实现 30
 - 3.3.1 配置初始化 30
 - 3.3.2 flag 参数检查 32
 - 3.3.3 创建 engine 对象 33
 - 3.3.4 设置 engine 的信号捕获 34
 - 3.3.5 加载 builtins 35
 - 3.3.6 使用 goroutine 加载 daemon 对象并运行 38
 - 3.3.7 打印 Docker 版本及驱动信息 41
 - 3.3.8 serveapi 的创建与运行 42
- 3.4 总结 42

第 4 章 Docker Daemon 之 NewDaemon 实现 43

- 4.1 引言 43

4.2	NewDaemon 具体实现	44	4.15	总结	62
4.3	应用配置信息	45	第 5 章 Docker Server 的创建 63		
4.3.1	配置 Docker 容器的 MTU	45	5.1	引言	63
4.3.2	检测网桥配置信息	46	5.2	Docker Server 创建流程	63
4.3.3	查验容器间的通信配置	46	5.2.1	创建名为“serveapi”的 Job	64
4.3.4	处理网络功能配置	47	5.2.2	配置 Job 环境变量	64
4.3.5	处理 PID 文件配置	47	5.2.3	运行 Job	65
4.4	检测系统支持及用户权限	48	5.3	ServeApi 运行流程	65
4.5	配置工作路径	49	5.4	ListenAndServe 实现	68
4.6	加载并配置 graphdriver	49	5.4.1	创建 router 路由实例	69
4.6.1	创建 graphdriver	49	5.4.2	创建 listener 监听实例	74
4.6.2	验证 btrfs 与 SELinux 的兼容性	51	5.4.3	创建 http.Server	74
4.6.3	创建容器仓库目录	51	5.4.4	启动 API 服务	75
4.6.4	迁移容器至 aufs 类型	51	5.5	总结	75
4.6.5	创建镜像 graph	52	第 6 章 Docker Daemon 网络 76		
4.6.6	创建 volumesdriver 以及 volumes graph	53	6.1	引言	76
4.6.7	创建 TagStore	53	6.2	Docker Daemon 网络介绍	77
4.7	配置 Docker Daemon 网络环境	54	6.3	Docker Daemon 网络配置接口	78
4.7.1	创建 Docker 网络设备	55	6.4	Docker Daemon 网络初始化	79
4.7.2	启用 iptables 功能	55	6.4.1	启动 Docker Daemon 传递 flag 参数	80
4.7.3	启用系统数据包转发功能	56	6.4.2	解析网络 flag 参数	80
4.7.4	创建 DOCKER 链	56	6.4.3	预处理 flag 参数	80
4.7.5	注册处理方法至 Engine	57	6.4.4	确定 Docker 网络模式	81
4.8	创建 graphdb 并初始化	57	6.5	创建 Docker 网桥	82
4.9	创建 execdriver	58	6.5.1	提取环境变量	83
4.10	创建 daemon 实例	59	6.5.2	确定 Docker 网桥设备名	83
4.11	检测 DNS 配置	60	6.5.3	查找 bridgeIface 网桥设备	83
4.12	启动时加载已有 Docker 容器	61	6.5.4	bridgeIface 已创建	84
4.13	设置 shutdown 的处理方法	61	6.5.5	bridgeIface 未创建	85
4.14	返回 daemon 对象实例	62			

6.5.6	获取网桥设备的网络地址	88	网络环境	120
6.5.7	配置 Docker Daemon 的 iptables	88	7.7 总结	122
6.5.8	配置网络设备间数据报转发功能	88	第 8 章 Docker 镜像	123
6.5.9	注册网络 Handler	89	8.1 引言	123
6.6	总结	89	8.2 Docker 镜像介绍	124
第 7 章 Docker 容器网络		90	8.3 rootfs	124
7.1	引言	90	8.4 Union Mount	125
7.2	Docker 容器网络模式	93	8.5 image	127
7.2.1	bridge 桥接模式	93	8.6 layer	128
7.2.2	host 模式	95	8.7 总结	129
7.2.3	other container 模式	96	第 9 章 Docker 镜像下载	130
7.2.4	none 模式	97	9.1 引言	130
7.3	Docker Client 配置容器网络模式	97	9.2 Docker 镜像下载流程	131
7.3.1	使用 Docker Client	98	9.3 Docker Client	131
7.3.2	runconfig 包解析	98	9.3.1 解析镜像参数	132
7.3.3	CmdRun 执行	102	9.3.2 配置认证信息	136
7.4	Docker Daemon 创建容器网络流程	103	9.3.3 发送 API 请求	137
7.4.1	创建容器之网络配置	103	9.4 Docker Server	138
7.4.2	启动容器之网络配置	105	9.4.1 解析请求参数	138
7.5	execdriver 网络执行流程	111	9.4.2 创建并配置 Job	139
7.5.1	创建 libcontainer 的 Config 对象	112	9.4.3 触发执行 Job	139
7.5.2	调用 libcontainer 的 namespaces 启动容器	116	9.5 Docker Daemon	140
7.6	libcontainer 实现内核态网络配置	117	9.5.1 解析 Job 参数	140
7.6.1	创建 exec.Cmd	118	9.5.2 创建 session 对象	141
7.6.2	启动 exec.Cmd 创建进程	119	9.5.3 执行镜像下载	142
7.6.3	为容器进程初始化		9.6 总结	147
			第 10 章 Docker 镜像存储	149
			10.1 引言	149
			10.2 镜像注册	150

10.3	验证镜像 ID	151	12.3	Docker Daemon 创建容器对象	185
10.4	创建镜像路径	152	12.3.1	LookupImage	186
10.4.1	创建 mnt、diff 和 layers 子目录	153	12.3.2	CheckDepth	188
10.4.2	挂载祖先镜像并返回根 目录	155	12.3.3	mergeAndVerifyConfig	188
10.5	存储镜像内容	157	12.3.4	newContainer	189
10.5.1	解压镜像内容	158	12.3.5	createRootfs	190
10.5.2	收集镜像大小并记录	160	12.3.6	ToDisk	190
10.5.3	存储 jsonData 信息	161	12.3.7	Register	191
10.6	注册镜像 ID	162	12.4	Docker Daemon 启动容器	191
10.7	总结	163	12.4.1	setupContainerDns	192
第 11 章	docker build 实现	164	12.4.2	Mount	193
11.1	引言	164	12.4.3	initializeNetworking	194
11.2	docker build 执行流程	165	12.4.4	verifyDaemonSetting	194
11.2.1	Docker Client 与 docker build	166	12.4.5	prepareVolumesForContainer	195
11.2.2	Docker Server 与 docker build	170	12.4.6	setupLinkedContainers	196
11.2.3	Docker Daemon 与 docker build	171	12.4.7	setupWorkingDirectory	199
11.3	Dockerfile 命令解析流程	174	12.4.8	createDaemonEnvironment	199
11.4	Dockerfile 命令分析	177	12.4.9	populateCommand	200
11.4.1	FROM 命令	177	12.4.10	setupMountsForContainer	200
11.4.2	RUN 命令	178	12.4.11	waitForStart	201
11.4.3	ENV 命令	182	12.5	总结	202
11.5	总结	182	第 13 章	dockerinit 启动	203
第 12 章	Docker 容器创建	183	13.1	引言	203
12.1	引言	183	13.2	dockerinit 介绍	204
12.2	Docker 容器运行流程	184	13.2.1	dockerinit 初始化内容	204
			13.2.2	dockerinit 与 Docker Daemon	204
			13.3	dockerinit 执行入口	205
			13.3.1	createCommand 分析	205
			13.3.2	namespace.exec	207
			13.4	dockerinit 运行	208
			13.4.1	reexec.Init() 的分析	208

13.4.2	dockerinit 的执行流程	210	15.2.1	Swarm Node	231		
13.5	libcontainer 的运行	211	15.2.2	Docker Node	231		
13.5.1	Docker Daemon 设置 cgroups 参数	213	15.2.3	node discovery	231		
13.5.2	Docker Daemon 创建网络 栈资源	213	15.2.4	scheduler	232		
13.5.3	dockerinit 配置网络栈	213	15.3	Swarm 命令	232		
13.5.4	dockerinit 初始化 mount namespace	215	15.3.1	swarm create	232		
13.5.5	dockerinit 完成 namespace 配置	215	15.3.2	swarm manage	232		
13.5.6	dockerinit 执行用户命令 Entrypoint	217	15.3.3	swarm join	233		
13.6	总结	218	15.3.4	swarm list	234		
第 14 章 libcontainer 介绍			218	15.4	总结	234	
14.1	引言	219	第 16 章 Machine 架构设计与实现			235	
14.2	Docker、libcontainer 以及 LXC 的关系	220	16.1	引言	235		
14.3	libcontainer 模块分析	221	16.2	Machine 架构	236		
14.3.1	namespace	221	16.2.1	Machine	237		
14.3.2	cgroup	224	16.2.2	Store	237		
14.3.3	网络	225	16.2.3	Host	237		
14.3.4	挂载	226	16.2.4	Driver	238		
14.3.5	设备	227	16.2.5	Provisioner	238		
14.3.6	nsinit	227	16.2.6	Machine 运行流程	239		
14.3.7	其他模块	227	16.3	Machine 与 Swarm 的结合	240		
14.4	总结	228	16.4	总结	241		
第 15 章 Swarm 架构设计与实现			241	第 17 章 Compose 架构设计与实现			242
15.1	引言	229	17.1	引言	242		
15.2	Swarm 架构	230	17.2	Compose 介绍	242		
			17.3	Compose 架构	243		
			17.4	Compose 评价	246		
			17.4.1	Compose 单机能力	246		
			17.4.2	Compose 跨节点能力	247		
			17.4.3	Compose 与 Swarm	247		
			17.5	总结	247		