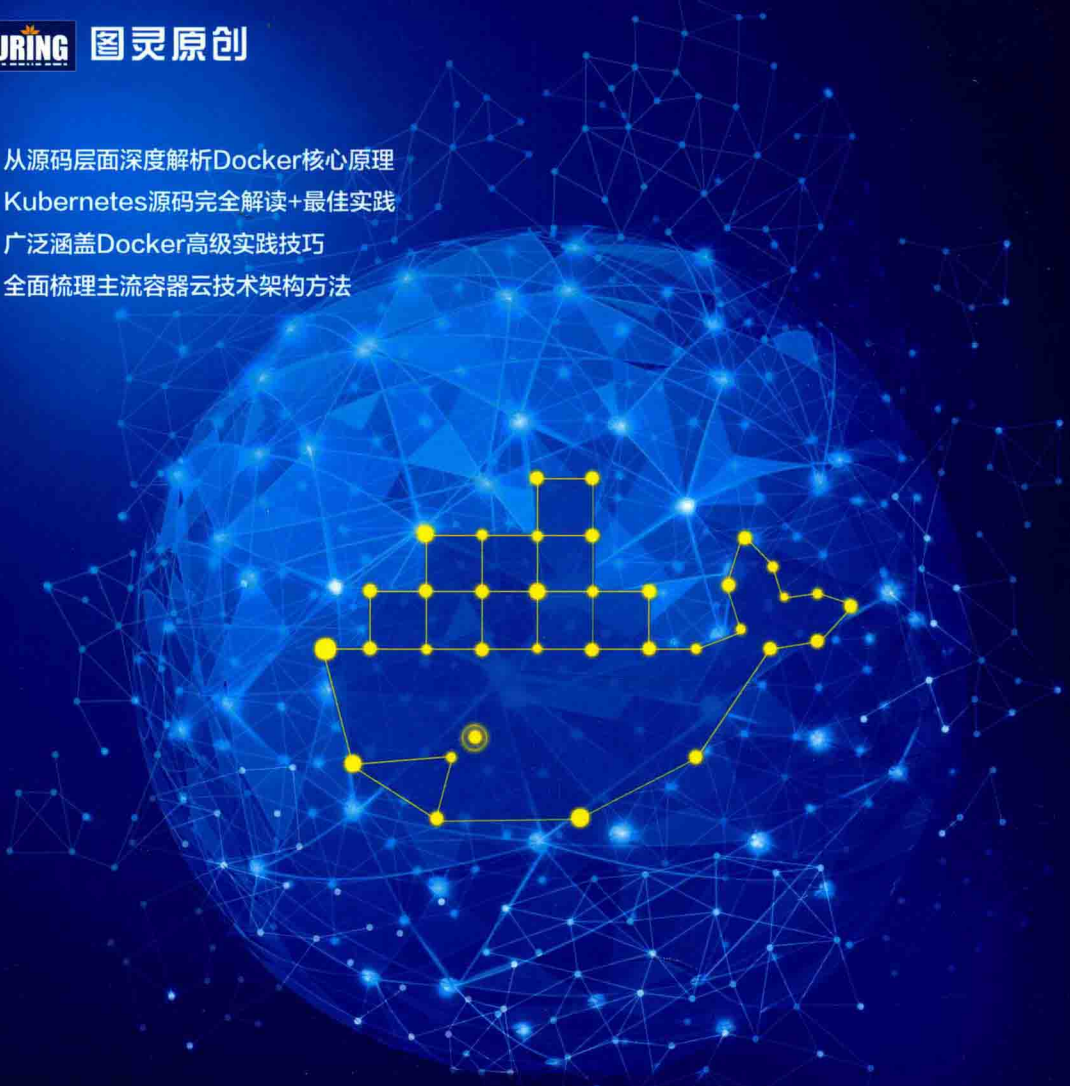
TURING 图灵原创

- 从源码层面深度解析Docker核心原理
- Kubernetes源码完全解读+最佳实践
- 广泛涵盖Docker高级实践技巧
- 全面梳理主流容器云技术架构方法



Docker 容器与容器云

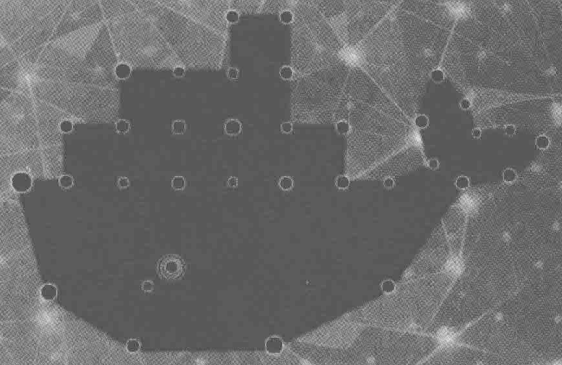
浙江大学SEL实验室◎著

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵原创

Docker 容器与容器云

浙江大学SEL实验室◎著



人民邮电出版社
北京

图书在版编目（CIP）数据

Docker : 容器与容器云 / 浙江大学SEL实验室著.
-- 北京 : 人民邮电出版社, 2015.9
（图灵原创）
ISBN 978-7-115-40110-6

Ⅰ. ①D… Ⅱ. ①浙… 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(2015)第170576号

内 容 提 要

本书从实践者的角度出发，基于Docker和Kubernetes最新源码，系统梳理了Docker容器技术和Kubernetes项目的实现原理和设计思路，有助于读者在实际场景中利用Docker容器和容器云解决问题并启发新的思考。全书包括两部分，第一部分深入解读Docker容器技术，包括Docker架构与设计、核心源码解读和高级实践技巧；第二部分归纳和比较了三类基于Docker的主流容器云项目，包括专注Docker容器编排与部署的容器云、专注应用支撑的容器云以及一切皆容器的Kubernetes，进而详细解读了Kubernetes核心源码的设计与实现，最后介绍了几种典型场景下的Kubernetes最佳实践。

本书适用于有一定Docker基础的开发者、架构师、IT专业学生以及探索基于Docker构建云计算平台的技术人员，也非常适合作为高校教材或培训资料。

◆ 著　　浙江大学SEL实验室
责任编辑　王军花
策划编辑　张　霞
责任印制　杨林杰
◆ 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
北京天宇星印刷厂印刷
◆ 开本：800×1000　1/16
印张：26
字数：614千字　　2015年9月第1版
印数：1－3 500册　　2015年9月北京第1次印刷

定价：89.00元

读者服务热线：(010)51095186转600　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京崇工商广字第0021号

推荐语

“虽然在此之前已经有了由Docker团队出的第一本Docker书，但是这是国内第一本深入解读Docker与Kubernetes原理的原创图书，这一点意义重大。本书比较完整地介绍了Docker与Kubernetes的工作原理和生态，非常有借鉴意义。”

——许式伟，七牛云存储 CEO

“Docker容器技术已经在国内如火如荼地流行起来，浙江大学SEL实验室目前是国内掌握Docker技术最熟练的技术团队之一，他们在国内Docker技术界一直产生着重要影响。这次他们把Docker的实战经验汇编成书，可以帮助更多的Docker爱好者学习到一手的实战经验。”

——肖德时，数人科技 CTO

“本书非常细致地讲解了Docker技术的来龙去脉和技术细节，更为难得的是还加入了Docker生态当中的其他技术。Docker这项技术本身就是将多种思想和技术融合的产物，从生态的视角去解读技术的来龙去脉将极大地促进读者对云计算和容器技术的重新思考。”

——程显峰，OpenAPM 首席运营官

“本书宏观上描绘了容器和容器云技术发展的浪潮和生态系统，微观上以Docker和Kubernetes为典型进行了深度分析。无论是Docker技术爱好者，还是系统架构师、云端开发者、系统管理和运维人员，都能在本书中找到适合自己阅读的要点。浙江大学SEL实验室云计算团队是一支非常优秀的云计算研究团队，很多 85 后、90 后人才活跃在顶级社区前沿，感谢他们能将多年的知识和智慧积累分享出来!”

——刘俊，百度运维部高级架构师，百度最高奖获得者

“本书是浙江大学 SEL 实验室云计算团队多年深耕 Docker 及背后的容器技术的结晶。最大的特点就是深入，并且有各种实用案例和细致讲解。另外，这本书在怎样真正地把 Docker 及周边

产品落地以构建灵活多变的云平台方面也进行了生动的阐释。”

——郝林，微影时代架构师，《Go 并发编程实战》作者

“Docker 颠覆了容器技术，也将容器技术带到了新的高度。InfoQ 从 2014 年初就开始密切关注容器技术，见证并切身参与了容器技术的发展。作为我们的优秀作者，浙江大学 SEL 实验室在 InfoQ 撰写了很多与 Docker、Kubernetes 相关的技术文章，得到了广大读者的肯定。希望这本书能推动容器技术在中国的落地。”

——郭蕾，InfoQ 主编

“浙江大学 SEL 实验室属于国内较早接触并研究开源 PaaS 技术的团队之一，从传统 PaaS 的开源代表 CloudFoundry、OpenShift，到新一代基于 Docker 的 PaaS 平台如 DEIS、Flynn 等，他们均有深入的研究和实践经验。更为难得的是，他们不仅参与开源贡献，而且笔耕不辍，通过博客、论坛等方式积极分享有深度、有内涵的技术文章，并广泛参与国内 PaaS 届各种技术交流会议。华为 PaaS 团队也在与之交流中汲取了不少营养。此次，他们将近年来对 Docker 容器和 Kubernetes、DEIS、Flynn 等 PaaS 开源平台的研究成果结集成册，内容详尽且深入浅出。我相信，无论是入门者还是老手，都能够从中获益。”

——刘赫伟，华为中央软件院高级软件架构师

“容器技术在大型互联网企业中已广泛应用，而 Docker 是容器技术中的杰出代表。本书不仅介绍了 Docker 基础知识，而且进行了代码级的深入分析，并通过对 Kubernetes 等技术的讲解延伸至集群操作系统以及对 Docker 生态领域的思考，同时结合了大量实践，内容丰富，值得拥有。”

——王炜煜，百度运维部高级架构师，JPaaS 项目负责人

“Docker 作为操作系统层面轻量级的虚拟化技术，凭借简易的使用、快速的部署以及灵活敏捷的集成支持等优势，奠定了 Docker 如今在 PaaS 领域的江湖地位。浙江大学 SEL 实验室在云计算和 PaaS 领域耕耘多年，积累了丰富的经验。本书既有对 Docker 源代码层面的深度解读，也有实战经验的分享，希望这本书能够帮助 Docker 开发者在技术上更上一层楼。”

——李三红，蚂蚁金服基础技术部 JVM Architect

序

我已从事软件工程研究工作二十余年，在这期间，软件开发方式发生了巨大的变化。瞬息万变是这个时代的特征，固守经典、一成不变已无法应对，当代的软件工程拥有快速迭代的生命周期，越来越多的开发组织投入巨大精力关注软件开发的敏捷性。

云计算有了明确定义，浙江大学就组织研究力量投入这滚滚浪潮之中。云计算定义了一种按需索取、实时供应的特性，它是敏捷的。云平台提供的资源是计算能力，人们获得计算能力资源一如获取自来水和管道煤气一样方便。这为软件工程注入了新的活力，如果软件开发者可以快速、自由地获取开发过程中所需的各种资源，那么软件开发必将迎来一次飞跃式的发展。

然而，我们似乎并没有获得想要的飞跃。

众所周知，云计算拥有一个圣经般的三层模型，界限明确，职责分明。当下，依照圣经“戒律”，众多业内巨头率先建立起一批重型云平台，然而问题却慢慢浮现——按照传统定义设计的“云”对应用不够友好要么做得不够，要么管得太死。

是时候打破“戒律”了吗？我认为是。

Docker 让所有人眼前一亮，它模糊了 IaaS 与 PaaS 之间的界限，为云计算的服务形式带来了无限的可能，Docker 带着它的容器理念破而后立，是云计算运动中一项了不起的创举。

丁铁群老师带领他的团队写作的这本《Docker——容器与容器云》，在很大程度上填补了国内容器与容器云技术领域深度分析的空白。本书浓缩了浙大 SEL 实验室 4 年来在 PaaS 以及容器技术领域的研究成果与开发实践经验，深入浅出地分析了云计算领域容器应用现状，是一部值得业内人士和容器技术爱好者长置案头的好书。

杨小虎

浙江大学软件学院院长

前　言

本书的写作目的不仅是在技术层面深入分析 Docker 背后的技术原理和设计思想，更在于从我们团队几年来在云计算方面的积累出发，理清当前基于 Docker 的云计算平台（容器云）的发展脉络，以期对 IT 企业的开发运维人员、容器云服务提供商以及 Docker 技术爱好者在技术选型、技术路线规划上有所帮助。

2013 年是 Docker 正式开源发布的年份，也是我们团队开始使用 Docker 的时间。当时 Docker 作为一个单机版轻量级虚拟化工具，并没有如当前发达的生态系统。我们使用 Docker 处理 Cloud Foundry 这类复杂分布式系统的快速部署和迁移。结果是我们体验到了惊喜还有遗憾。确实，那时候 Docker 1.0 尚未发布，作为最先吃螃蟹的人之一，我们除了能感受到 Docker 相比虚拟机的在资源利用率和性能上的巨大优势以及在使用方式上的高效便捷，还不得不忍受当时的 Docker 与一个完整的数据中心运维系统之间的差距。比如网络，跨宿主机间的通信在很长一段时间都困扰着我们；比如容器内部不能单独配置内核参数，一旦应用对性能有特殊要求的时候，就无法单独进行优化定制；再比如维护，时常需要手动清理僵尸容器、镜像等。

在随后的一整年里，我们真真切切地感受到了 Docker 是如何从一个开发运维人员略有耳闻的工具成为一个技术圈里家喻户晓的名词。基于 Docker 的公有云、私有云项目也如雨后春笋般涌现；各大知名技术社区都为 Docker 开辟专栏，甚至出现了专为讨论 Docker 而生的技术社区。基于 Docker 的中国本土化也开始萌芽，各类国内镜像托管和加速服务层出不穷。Docker 官方也没有闲着，前不久 Docker 的各类邮件列表中都出现了招聘中国区执行官的消息。Docker 生态系统的建立已经是不争的事实，在这一年的时间里，我们团队也从 Docker 的使用者，成为了 Docker、Kubernetes、libcontainer 等开源项目的特性维护者（maintainer）和代码贡献者（contributor）。

当前 Docker 已决不仅仅是一项轻量级虚拟化技术，官方的 Docker 运维三件套、来自第三方的 Kubernetes、OpenShift v3、Flynn、Deis 等项目已经基于 Docker 这种容器技术构建出各种各样的容器云服务平台，关于 Docker 等容器技术的讨论重心也已经从“容器”转变为“容器云”。Docker 对于 IT 行业的价值也从节省资源这一方面扩展到对整个软件开发运维生命周期的改造。

作为软件行业多年的实践者和教育者，我们一直试图探索这样一些问题：云计算除了当前被广为接受的基础设施云平台（IaaS）的形态，是否还有更加贴近开发人员和运维人员的形态？云计算如何以更好的形态服务于互联网、互联网+这样一个以软件连接人与人，以软件连接人与企业，以软件连接企业与企业的时代？正是 Docker 这样的容器技术的出现使得这样的探索成为可能。

本书结构

本书共分两部分，沿着从容器到容器云的发展脉络，介绍 Docker 以及围绕 Docker 构建的各类容器云平台技术，深入分析了 Kubernetes 背后的技术原理和设计思想。

第一部分讲解了 Docker 容器的核心原理和实践技巧。其中第 1 章和第 2 章能够让读者在短时间内体验这场 IT 界的风暴，并且初步了解 Docker 的使用方法，为后续的源码解析做铺垫。第 3 章是本书第一部分的核心，这一章以最新的 Docker 1.6.2 版本源码为基础，结合最新的社区主干 1.7.0 代码进展，深入地分析了容器的 namespace 和 cgroups 原理，紧接着我们以 docker run 命令为线索，一路贯穿 Docker 的容器创建、镜像组织、联合文件系统以及容器网络初始化的源码，深入透彻地向读者展示了从一句指令到最终 Linux 容器生成的整个过程中，Docker 源码背后的执行路线和设计原理。第 4 章则介绍了 Docker 相关的几类实践技巧，包括网络、监控、服务发现等。值得一提的是，在上述代码走读的过程中，本书几乎没有贴出任何一部分 Docker 源码或者函数，而是力图使用平实的语言和生动的图示来展示代码背后的执行逻辑和设计思想。Docker 的源码字字珠玑，我们希望能够使用这样的解读方式使读者真正理解 Docker 和容器背后的设计方法和技术本质，而不是变成一本单纯的技术手册。

第二部分深入分析基于 Docker 的各类容器云平台的架构细节和背后的设计理念，这些容器云虽然在底层技术上都基于 Docker 这样的容器技术，但在背后的设计思想上却存在很大的差异。我们将看到一个颠覆原有的 IaaS、PaaS 云计算生硬的分类方式而精彩纷呈的容器云世界，其中第 5 章介绍了一个最简单的容器云解决方案作为引子；第 6 章和第 7 章分析和比较了几类典型的容器云开源项目，这包括了 Docke 官方的三剑客项目、Fleet 以及更类似经典 PaaS 的 Flynn 和 Deis；第 8 章是本书第二部分的重点章节，我们以 Kubernetes 0.16 版本源码为基础，从核心概念到架构梳理，再到深入到组件级别的 Kubernetes 源码解析，从多个维度详细讲解了 Kubernetes 容器云平台的各种技术细节，这在国内尚属首次。希望通过这部分容器云平台源码解读，我们能够带领读者从纷繁复杂的容器云项目中梳理出一个细致的脉络，让读者在选型和二次开发的过程中减少迷茫和试错。而作为 Kubernetes 项目的贡献者和特性维护者，我们希望有更多的技术人员能够从源码层面对 Kubernetes 有更深刻的理解和认识，并且同我们一起来推动这个优秀的开源项目在国内的进步和落地。在第二部分的结尾，我们试图回答之前的提问，即容器云应该以何种形态来更好地支撑当今时代。

致谢

对于能够编写国内第一本在源代码层面深度解析 Docker 和 Kubernetes，并揭秘基于 Docker 容器的云计算生态圈底层技术的图书，我们感到非常荣幸。浙江大学 SEL 实验室云计算团队在此向所有支持帮助我们的朋友表达最诚挚的谢意，没有大家的支持，我们很可能无法顺利地完成这项工作。

感谢浙江大学软件学院杨小虎院长对云计算团队一直以来的关怀和支持，杨院长的远见卓识和诲人不倦令人钦佩。

感谢多年来与我一同奋战在云计算前线，并参与本书大量编写工作的张磊和孙健波。他们自加入云计算团队以来一直战斗在云计算领域最前沿，是写作本书的中坚力量。

感谢以极大热情参与到本书写作中的浙江大学计算机学院、软件学院的各位研究生，他们的热情是我们团队活力的源泉，他们使那些分散在各个领域的技术得以整合。在本书编写过程中，他们不计个人得失地精诚合作，这是本书得以成书的基石。

特别要感谢不辞辛劳为本书出谋划策、日以继夜不断审阅修改的图灵公司的编辑们。在整个写作过程中，我们团队得到了出版方的大力支持，他们认真负责的态度是本书顺利出版的保证。

感谢 InfoQ 主编郭蕾一直以来对浙大团队技术分享工作所做出的支持和推广，他和 InfoQ 的同事们的努力是推动本书发起的最大动力。

感谢《第一本 Docker 书》的译者刘斌对本书进行的细致的审读，并为我们提出了宝贵的修订建议。

感谢浙江大学 SEL 实验室云计算团队的其他所有人，他们认真负责的工作态度和令人满意的工作成果是本书不可或缺的支持力量。

感谢大家的共同努力，让我们的成果得以面世，在 Docker 布道之路上贡献一份光热，传播惠及当下的云计算前沿技术。

丁轶群

浙江大学 SEL 实验室云计算团队

目　录

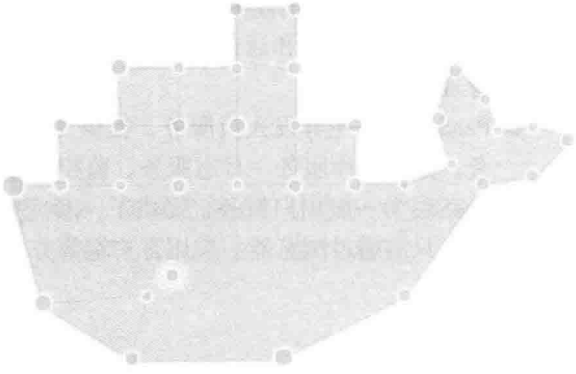
第一部分　Docker深入解读

第三部分　附录

第一部分

Docker 深入解读

Docker 作为时下流行的容器技术，已经在云计算领域掀起了一股狂潮。本书的第一部分从 Docker 容器技术的出现背景谈起，阐述这门技术的方方面面。第 1 章中说明了 Docker 的发展脉络，帮助读者了解到底是什么赋予了 Docker 如此大的魅力。第 2 章用 Docker 的一个实际使用案例展示 Docker 的基本使用方法。第 3 章是本书的一大核心，从源码的层面，对构成 Docker 的各个模块进行原理上的解读。第一部分的最后一章介绍使用 Docker 的高级实践技巧。相信当你阅读完本部分的内容后，会对 Docker 容器技术有全面而深刻的理解。



第1章 从容器到容器云

2013年初，一个名字从云计算领域横空出世，并在整个IT行业激起千层浪。这就是Docker——一个孕育着新思想的“容器”。Docker选择容器作为核心和基础，依靠容器技术支撑的Docker迅速成为国内外各大云计算厂商以及开发者手中的至宝。在一片热火朝天之中，新的革命已经悄然来临。

1.1 云计算平台

回首历史，云计算时代蕴育出了众多的云计算平台，虽然在服务类型或平台功能上有所差异，但它们的本质上如出一辙，都与NIST[①]对于云计算平台的定义有着密切的关系。

> 云计算是一种资源的服务模式，该模式可以实现随时随地、便捷按需地从可配置计算资源共享池中获取所需的资源（如网络、服务器、存储、应用及服务），资源能够快速供应并释放，大大减少了资源管理工作开销，你甚至可以再也不用理会那些令人头痛的传统服务供应商了。

经典云计算架构包括了IaaS（Infrastructure as a Service，基础设施即服务）、PaaS（Platform as a Service，平台即服务）、SaaS（Software as a Service，软件即服务）三层服务，如图1-1所示。

- IaaS层为基础设施运维人员服务，提供计算、存储、网络及其他基础资源，云平台使用者可以在上面部署和运行包括操作系统和应用程序在内的任意软件，无需再为基础设施的管理而分心。
- PaaS层为应用开发人员服务，提供支撑应用运行所需的软件运行时环境、相关工具与服务，如数据库服务、日志服务、监控服务等，让应用开发者可以专注于核心业务的开发。
- SaaS层为一般用户服务，提供了一套完整可用的软件系统，让一般用户无需关注技术细节，只需通过浏览器、应用客户端等方式就能使用部署在云上的应用服务。

① National Institute of Standards and Technology，“The NIST Definition of Cloud Computing”论文定义了人们认可的云计算三层服务模型。

图1-1 云平台经典架构

同时，随着计算机技术推陈出新，应用的规模愈发庞大，逻辑愈发复杂，迭代更新愈发频繁，应用开发所需的统一规范和原有开发模式杂乱无章成了追求进步的主要障碍。在尖锐的矛盾中，云时代应用生命周期管理机制（Application Lifecycle Management，ALM）和十二要素应用规范（The Twelve-Factor App）[①]应运而生。

所有理论设计和预想一定是美好的，只是对于一个新的理论而言，如何经受住实践的考验，并将美好的愿景转化为生产力才是重中之重。IaaS的发展主要以虚拟机为最小粒度的资源调度单位，出现了资源利用率低、调度分发缓慢、软件堆栈环境不统一等一堆问题。PaaS在IaaS基础上发展而来，众多PaaS已经意识到可以利用容器技术解决资源利用率问题，但是PaaS通常在应用架构选择、支持的软件环境服务方面有较大的限制，这带来了应用与平台无法解耦、应用运行时环境局限性强、运维人员控制力下降的问题。

可见不论IaaS还是PaaS都有各自适用的场景，但依旧存在诸多缺陷，人们亟需一个真正可用的解决方案。

1.2 容器，新的革命

每一场革命背后都有着深刻的历史背景和矛盾冲突，新陈代谢是历史的必然结果，新生取代陈旧得益于理念的飞跃和对时代发展需求的契合，很显然Docker抓住了这个契机。

Docker是什么？

① 参考自The Twelve-Factor App英文原版（http://12factor.net/）。

根据官方的定义，Docker是以Docker容器为资源分割和调度的基本单位，封装整个软件运行时环境，为开发者和系统管理员设计的，用于构建、发布和运行分布式应用的平台。它是一个跨平台、可移植并且简单易用的容器解决方案。Docker的源代码托管在GitHub上，基于Go语言开发并遵从Apache 2.0协议。Docker可在容器内部快速自动化地部署应用，并通过操作系统内核技术（namespaces、cgroups等）为容器提供资源隔离与安全保障。

我们应该看看Docker的发展历程。

每一个传奇都需要一个这样的开头，很久很久以前：Docker项目由Solomon Hykes所带领的团队发起，在Docker公司的前身dotCloud内部启动孕育，代码托管于GitHub。

2013年3月：Docker正式发布开源版本，GitHub中Docker代码提交盛况空前，风头之劲一时无两。见图1-2。

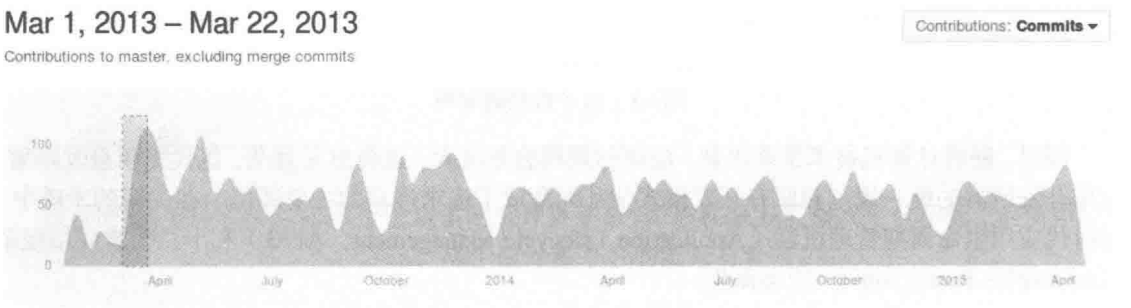


图1-2　GitHub上Docker项目提交数统计图

2013年11月：REHL 6.5正式版发布，集成了对Docker的支持，拉开了业界各大厂商竞相支持Docker的序幕。

2014年4月到6月：云技术市场上的三大巨头Amazon、Google及MicroSoft Azure相继宣布支持Docker，并着手开发基于容器的全新产品。

2014年6月：DockerCon 2014大会召开，会上来自Google、IBM、Amazon、Red Hat、Facebook及Twitter等全球领先企业的演讲嘉宾组成了豪华的阵容。此时，Docker自开源版本后又经历了15个月左右的飞速发展，Docker 1.0版本正式发布。

2014年8月：VMware宣布与Docker建立合作关系，标志了虚拟化市场形成了新的格局。

2014年10月：微软宣布将整合Docker进入下一代的Windows Server中。

2014年10月15日：Azure和Docker共同举办了Docker全球开发者大会，并宣布双方建立战略合作伙伴关系。

2014年底：Google率先发布容器引擎Google Container Engine（GCE），整合了Docker、Google自有容器技术和在DockerCon 2014大会上发布的Kubernetes，致力于为用户提供面向Docker化应

用的云计算平台；Amazon发布EC2 Container Service（ECS），它是一项高度可扩展、高性能、免费的容器管理服务，并能够在托管的Amazon EC2实例集群上轻松地发布、管理和扩展Docker容器，使得Amazon Web Services（AWS）用户能够使用AWS上的容器轻松地运行和管理分布式应用。

2015年4月：Docker公司宣布完成了9500万美元的D轮融资。此前，他们已完成三轮融资，包括1500万美元的B轮融资及4000万美元的C轮融资。

截至2015年5月：GitHub中Docker的贡献者多达近千人，被关注和喜爱（Star）多达两万一千余次（相比之下，此时Linux源码多年来积累的被关注次数为两万两千余次），并有近五千多个开发分支（Fork），Docker成为了GitHub上排名前20的明星项目。

Docker官方存储应用镜像的镜像仓库也获得了大量开发者支持，其镜像仓库里已有四万五千余个不同应用功能的公共镜像。最受欢迎的Ubuntu、MySQL、nginx、WordPress镜像，下载量已达到三四百万次。这些数字还在不断地增长！

在国内一线城市，几乎每一两周就有一场关于Docker的讨论大会，Docker永远不会让你孤独。

从此以后的未来，Docker已经给云计算带来了深远的影响，这是一次真正的计算机技术革命，来吧，拥抱变化！

一个软件项目的成功常常需要依托其衍生的生态系统，围绕或基于核心技术而构建的相关项目日臻丰富和完善，软件本身的功能和易用性也随之增加，Docker的迅猛发展与其强大的生态系统息息相关。首先我们从图1-3中整体上来审视一下它[①]。

如图1-3所示，围绕Docker的生态系统自下而上分别覆盖了IaaS层和PaaS层所涉及的各类问题，包括资源调度、编排、部署、配置管理、网络管理、应用开发和部署平台、应用开发工具、应用服务供应以及大数据分析等云计算相关的服务。除了基于Docker技术解决构建分布式平台无法回避的经典问题，Docker及其生态系统主要带来了以下几点好处。

- **持续部署与测试。**Docker消除了线上线下的环境差异，保证了应用生命周期的环境一致性和标准化。开发人员使用镜像实现标准开发环境的构建，开发完成后通过封装着完整环境和应用的镜像进行迁移，由此，测试和运维人员可以直接部署软件镜像来进行测试和发布，大大简化了持续集成、测试和发布的过程。
- **跨云平台支持。**Docker带来的最大好处之一就是其适配性，越来越多的云平台都支持Docker，用户再也无需担心受到云平台的捆绑，同时也让应用多平台混合部署成为可能。目前支持Docker的IaaS云平台包括但不限于亚马逊云平台（AWS）、Google云平台（GCP）、微软云平台（Azure）、OpenStack等，还包括如Chef、Puppet、Ansible等配置管理工具。

① 参考自“Docker Ecosystem-MindMeister Mind Map”（https://www.mindmeister.com/389671722/docker-ecosystem）。