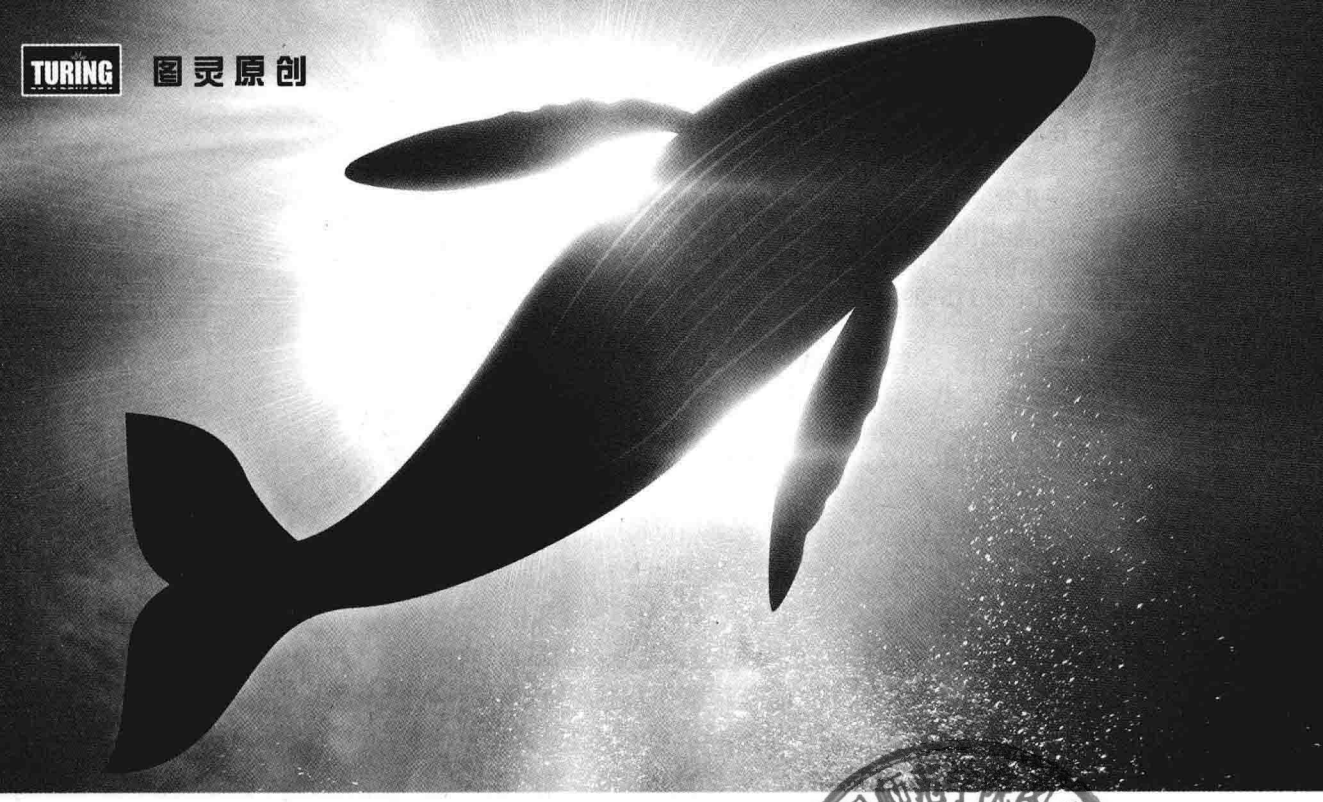
TURING 图灵原创

Docker 开发实践

曾金龙 肖新华 刘清 编著

中国工信出版集团

人民邮电出版社
POSTS & TELECOM PRESS



Docker 开发实践

曾金龙 肖新华 刘清 编著

人民邮电出版社
北 京

图书在版编目（CIP）数据

Docker开发实践 / 曾金龙，肖新华，刘清编著. --
北京 : 人民邮电出版社，2015.7
（图灵原创）
ISBN 978-7-115-39519-1

Ⅰ. ①D… Ⅱ. ①曾… ②肖… ③刘… 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122053号

内 容 提 要

本书由浅入深地介绍了 Docker 的实践之道，首先讲解 Docker 的概念、容器和镜像的相关操作、容器的数据管理等内容，接着通过不同类型的应用说明 Docker 的实际应用，然后介绍了网络、安全、API、管理工具 Fig、Kubernetes、shipyard 以及 Docker 三件套（Machine+Swarm+Compose）等，最后列举了常见镜像、Docker API 等内容。

本书适合 Docker 开发人员阅读。

◆ 编　　著　曾金龙　肖新华　刘　清
责任编辑　王军花
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　100061　　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：18.25
字数：431千字　　2015年7月第1版
印数：1-4 000册　　2015年7月北京第1次印刷

定价：59.00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

序

在互联网这个领域，每年都会涌现出非常多的新技术，其中总有那么一两门是引领潮流的，例如前些年的 Hadoop 和如今的 Docker。热门的技术之所以能够得到业内的推崇，主要是因为它解决了行业痛点。Hadoop 解决的是分布式计算的问题，它提供了一套分布式存储和计算的解决方案，而且这个方案很廉价但很高效；Docker 解决的是服务器应用快速构建、部署和分享的问题，它能够把服务器应用像 APP 一样简单地安装到各种平台环境中，而不用受真实环境的影响。细说起来，它有 3 个核心概念：容器、镜像和 Docker Hub。容器提供一个隔离的安全运行环境，使得不同应用之间不会相互干扰。和容器经常对比的是传统的虚拟机技术，这二者侧重点不一样，但在资源节省方面，容器占极大优势。镜像是容器的静态存在形式，就好比程序是进程的静态文件形式一样。Docker 的镜像采用分层机制，每次改变就增加一层，用来记录这些改变，这样在传输的时候只需要传输改变的层即可。Docker Hub 是一个公共的镜像平台，为镜像分享提供便利。用户可以根据自己的需要，在已有的镜像上定制自己的镜像。除了这 3 大核心组件外，由于越来越多开发者加入到 Docker 的生态圈中，出现了越来越多的工具，例如 Fig、Kubernetes、shipyard 等。这些都是非常优秀的辅助工具，而 Fig 也被 Docker 官方收购，成为了官方的 Compose 服务，对外提供 Machine+Swarm+Compose 整套服务。

Docker 发展很快，网络上的资料也比较少，有的也只是一些初级操作，此时适时宜的一本好书对一门技术的布道也极为重要。而我看完本书之后，觉得它可以堪此大任。本书把内容分为了三篇：基础篇、案例篇和高级篇。基础篇向我们介绍了 Docker、容器、镜像等核心内容和操作，这部分的亮点在于体系性，对命令详细的解释以及对命令的体系编排，这是其他书所没有的。案例篇为我们提供了一些综合案例，每一个案例都是精挑细选的，所有的案例都非常具备代表性。在高级篇，本书的深度就体现出来了，它不是简单地给读者罗列几个概念和工具，而是在介绍完概念后进行实实在在的操作，这是目前任何关于 Docker 的书都没有的。所以，Docker 的高级使用者应该会对这部分内容非常感兴趣。例如，Docker API 编程、Fig、Kubernetes、shipyard 等，本书都给出了很完整的介绍和操作。总的来说，这本书在体系编排、广度和深度上都做得非常好。作者的文笔也很干练，没有过多的兜兜转转，这也是技术类图书该有的风格。这是一本关于 Docker 难得的宝典，无论你是初级用户还是高级用户，本书都能够带你走进 Docker 的世界，登堂入室。

甘南南

迅雷看看副总裁

2015 年 3 月

推 荐 语

好的技术需要一本好书去布道，让更多的开发者从中受益。本书通过基础、案例到高级话题三篇，带领着读者由浅至深，登堂入室。

——罗笑南，中山大学信息科学与技术学院教授

Docker是当之无愧的Go语言杀手级应用，并且现如今Docker这个词的含义越来越丰富了，以至于它已经代表了容器技术的生态圈。本书奉行实践出真知，其中的案例都非常棒。更为关键的是，其中的高级篇对 Docker 生态圈的各个新成员也做了非常详实的介绍和实践，这真的很难能可贵。

——郝林，Go语言北京用户组发起人，《Go并发编程实战》作者

本人是在互联网行业摸爬滚打了十几年的老兵，我们开发的视频搜索服务，曾经占据了日本九成以上的市场，并被日本雅虎、乐天等主流公司采用。在此过程中，我接触到了日本大部分的主流互联网公司，深深地感觉到，应用的灵活部署、方便迁移、灵活扩充等是很大的难题，这方面的能力也是一个公司的核心竞争力。我们自己以及日本的合作伙伴，都为此做了许多的探索。

Docker是解决这些问题的很好的方案，是互联网行业的利器，今后必然会得到广泛的应用。

本书既有高屋建瓴的技术理念，也有实际操作的详尽说明，深入浅出了说明了 Docker 的相关知识，是一本不错的好书，值得每一位对Docker有兴趣的人收藏。

——颜文远，技术大牛

互联网技术每天都在快速更新，而Docker无疑是2014年最热门的互联网技术。这本书是我见过讲解Docker最为全面而又深入的图书，各个方面都有涉及，而且以实战为主。

——赵伟，北京精益智慧教育科技CTO

这是一本关于 Docker 的好书，值得所有想了解 Docker 的人放在键盘左边。

——李毅秋，人人网技术总监

腾讯的互娱的开发节奏，只有 Docker 跟得上！如果你想你的团队加快开发速度，那么我推荐你使用 Docker，而本书从基础、案例到高级话题，都有很全面的覆盖。

——易剑，腾讯互动娱乐事业群高级架构师

后台工程师（特别是运维工程师）都应该了解和熟悉 Docker 这一利器，使用它能极大地提高开发和部署效率。这本书通俗易懂，对于学习 Docker 的人非常有帮助！

——罗海江，大疆创新科技有限公司高级技术经理

云计算的初级是数据的云化，下一步是程序的云化，而 Docker 则是程序云化当前最好的工具。让你的程序一次配置，全网增量迁移、运行。本书出自一线互联网研发人员之手，它是实战的结晶，所涉案例都是互联网公司的真实应用，对 Docker 的应用都不是浅尝辄止，而是带你登堂入室。

——潘向荣，迅雷看看高级技术经理

前　言

Docker从提出到现在已经走过了两年的时间，在这两年里，它一直都是云计算领域的热点。可以说，它是2014年互联网最热门的技术。Docker得到Google、微软、IBM、Red Hat的声援，而它也不负众望，在这短短的两年里快速迭代，一步步变得更加完善。在Docker之前，开发者都深陷软件环境的配置之苦，虽然说并不是所有的软件配置都很难，但不同环境下的配置问题却层出不穷，相信很多读者和我有类似的经历，就是想用一款开源软件，结果配置了许久都跑不起来，然后不得不放弃用它。而网上总是有很多的答疑，可是跟着照做，发现在自己的机器上就是跑不起来。环境差异，可能会让原本简单的问题复杂化，迟滞我们的开发进程。拿来主义对于懒惰的程序员来说是件好事，我们都希望拿来就用，这样就可以专注于我们本该干的活。对于测试人员和运维人员来说，也是如此，没人喜欢处理这些本不该重点关注、处理不好却会让人寸步难行的问题。Docker就像一个打包器，可以把你的应用及其环境整体打包，然后很方便地迁移到不同的平台，到处运行。在用户看来就如同运行在原来的机器上一样。或许该有人说我用虚拟机也可以实现同样的效果，为什么要选择Docker。当两样东西都能够做同一件事情时，我们比的是效率。你可以在一台服务器上部署几个或者十几个虚拟机实例，但我相信没人会在一台服务器上部署上百个虚拟机实例，这是因为资源的限制。而在一台服务器上部署上百个Docker容器却并不是什么难事。在镜像的传输和共享方面，Docker也做得非常好，它能够只传输那些改变了的数据，而不用像传输虚拟机镜像那样，动辄至少几百兆。在共享方面，Docker建立了Docker Hub，你可以根据已有的镜像定制自己的镜像，而无需每次都再造轮子。如此接地气的技术，怪不得业内都惊呼Docker是下一个Hadoop。

本书的起源

虽然Docker人气旺盛，但关于Docker的书却少之又少，更别说汗牛充栋了，这主要是Docker出现的时间尚短。对于已有的书，基础内容不够体系，大多只停留寥寥几个基础命令的展示，并没有很好地归纳整理；而高级篇又只停留于粗浅的概念介绍，毫无实践价值，特别是对Docker具有很大作用的管理工具，例如Fig、Kubernetes、shipyard等内容，没有一本书去系统讲解它们。我们觉得这么好的技术，应该有更为系统的书去让更多的人了解它，理解它，这正是本书存在的价值。

如果不是遇到了王军花编辑，或许就没有本书，而正是在她的鼓励下，才让我们有勇气去将一些原本零碎的知识归纳整理为一本完整的书。

本书内容

本书主要介绍了 Docker 的实践之道。我们按照由浅入深的编排将本书分为三篇。在基础篇，主要是让读者认识 Docker 的概念和基础操作，对比介绍了 Docker 和虚拟机等技术，从容器、镜像、数据卷以及容器的连接等方面说明 Docker 的操作。通过对基础篇的学习，读者不仅对 Docker 有了全局的认识，而且能够对 Docker 的基础操作得心应手。该篇包含第 1 章至第 4 章的内容。

- 第 1 章从概念上介绍了 Docker，让读者对它的概念、背景、组件以及相关技术有了全局的认识。
- 第 2 章和第 3 章分别介绍了容器和镜像的相关操作，二者是 Docker 操作的核心对象。
- 第 4 章介绍了容器的网络基础、数据卷的配置以及多个容器之间的互联。

第二篇是案例篇。在这一篇中，我们通过不同类型的应用来说明 Docker 的实际应用。我们不做案例的简单堆砌，而是通过不同类型的案例来说明各个知识点的应用，它包含第 5 章至第 11 章。

- 第 5 章介绍了如何创建 SSH 服务镜像，这满足了日常 SSH 远程登录的需求。
- 第 6 章构建了一个采用 Apache 作为 Web 服务器、PHP 作为 Web 开发语言、MySQL 作为数据库的 Web 应用案例。
- 第 7 章构建了一个采用 Node.js 作为开发语言、MongoDB 作为数据库的 Web 案例，该案例着重说明了跨主机的多容器代理互联。
- 第 8 章和第 9 章说明了如何在公共云平台——阿里云上部署 Docker 应用，这里以 WordPress 为例进行介绍。
- 第 10 章介绍了如何使用私有仓库。
- 第 11 章将云计算的两大热点联合，说明了如何通过 Docker 来构建 Hadoop 镜像及集群。

第三篇是高级篇。在这一篇中，对 Docker 的 API 以及管理工具 Fig、Kubernetes、shipyard 以及 Docker 三件套（Machine+Swarm+Compose）都有实践操作，该篇包含第 12 章至第 18 章。

- 第 12 章介绍的是容器的高级网络知识。
- 第 13 章从命名空间、cgroups、Linux 能力机制以及服务端防护等方面入手介绍了安全方面的知识。
- 第 14 章则是通过 curl 工具来学习 Docker 的 API 接口，并给出 docker-py 库的编程实例。
- 在第 15 章至第 18 章中，我们分别介绍了 Fig、Kubernetes、shipyard 以及 Machine+Swarm+Compose 三件套，这些都是为了更好地管理和使用 Docker 的工具。

第四篇为附录。在附录 A 中，我们按照系统镜像、数据库镜像、Web 镜像、语言镜像的类别来编排，列举了常见的镜像，以供读者查阅。附录 B 是 Docker API 列表的归纳整理，分为容

器相关和镜像相关，亦是为了方便读者查阅。附录 C 是我们在写作过程中所用到的资料引用。

阅读须知

阅读本书时，最好能够从前往后依序阅读。特别是基础篇，是理解案例篇和高级篇的基石，读者最好能够读完基础篇，再阅读案例篇和高级篇。在案例篇和高级篇中，章和章之间的联系并没有那么紧密，例如第 6 章的案例和第 7 章的案例并没有关联，它们属于侧重点不同的案例，所以无需依序阅读。高级篇的工具也是如此，读者可以根据需求的迫切程度而选择性阅读。

目标读者

一切想了解及深入理解 Docker 技术的人，都是本书的目标读者。对于初级读者，通过本书的基础篇，你就可以成为一个能够灵活应用 Docker 的人。当你对 Docker 有一定了解后，通过学习案例篇和高级篇，你也就可以登堂入室了。

致谢

本书是很多人共同劳动的成果。三位作者要感谢一些人，感谢他们为本书所作出的巨大贡献和支持。

感谢在迅雷的同事：特别感谢甘南南和潘向荣两位的工作支持和悉心指导，感谢他们的慷慨大方，提供给我们充足的时间来写作本书。此外，还需感谢的同事有涂海涛、李明良、吴小强、易萌萌、何赞裕、吴建国和何锐，感谢他们在成书过程中的宝贵意见。

特别感谢本书的编辑王军花，没有她，本书就只能停留在笔者的脑海里。她不仅仅是一位资深的计算机类图书编辑，更像一名计算机专家，本书很多内容都是在她的指点下得以完善。

本书还需致谢的人有：盛建强博士、高怀恩博士、广发证券信息部的刘润佳和周英贵、美团网的蒋朋、百度的徐则水和罗剑波、腾讯的杨晓颖和郑克松、阿里巴巴的田晓娇以及李海龙、邱俊凌、杨文武、刘汇洋、冯学汉等人。感谢他们让各自公司相关平台成为本书部分案例的尝鲜者，并提出了诸多宝贵意见。

最后，感谢我们的家人。没有你们的支持，就没有这一切。

目　录

第一篇　基础篇：Docker基础

第二篇 案例篇：综合案例

第三篇 高级篇：高级话题、API、工具及集群管理

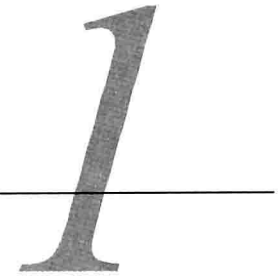
第四篇 附录

Part 1 第一篇

基础篇：Docker 基础

本篇内容

第1章 Docker简介



从虚拟机到容器，再到现在Docker的出现，虚拟化技术越来越受到互联网业界的关注和看好。在本章中，我们简要介绍一下Docker，其中主要包括以下内容。

Docker简介，其中包括Docker的概念、产生的背景、两个比Docker更早的虚拟化技术（虚拟机和容器）及其异同、Docker相对于一般容器的改进和优点以及Docker的应用场景。

- Docker的架构和组件。Docker是一种C/S架构的容器引擎，包含镜像、容器和库这3个重要概念。
- Docker的相关技术，主要从隔离性、可度量性、移植性和安全性这4个方面讨论。
- Docker的安装，其中包含各种Linux变种系统上的安装以及通过虚拟工具在Windows和OS X上的安装。

1.1 Docker 简介

在互联网初期，几乎所有的应用都以协议栈堆叠的形式进行开发，并且部署到单一的专有服务器上。如图1-1所示，15年前的应用又笨又重，而当时的终端设备也非常笨重，应用是基于一系列良好定义的协议栈进行开发的，它们包含中间件、运行时环境和操作系统；硬件所在的硬件环境也完全一致，即为一个服务配置单一的专有服务器，也就是说脱离了它原本的环境，整套系统或许就不能正常工作。随着互联网的发展，这种模式越来越不能满足日益复杂的互联网环境和产品需求。今天，应用开发者可以通过组合不同的服务来构建和装配应用，并使得应用能够跨越不同的硬件环境，如公共的、私有的以及虚拟的云服务器。

做到既能够组合当前最佳服务又跨越多种运行环境并非容易的事情。图1-2展示了当前一个网络应用可能涉及的方方面面，在软件层面，它的前台可能采用Nginx 1.5+ModSecurity+OpenSSL+Bootstrap 2来构建，后台采用Python 3.0等来构建，而在API端可能采用Python 2.7，数据库方面可能有多种数据库存在，每一项都是拿现存已有的服务，进而装配出应用。在硬件层面，它面对的环境错综复杂，可能是在虚拟机上部署，也可能在公共云、开发者的个人电脑、测试服务器以及产品集群等上部署。当一个应用拥有复杂的软件依赖关系和多样的硬件运行环境时，有以下几个问题必须面对。

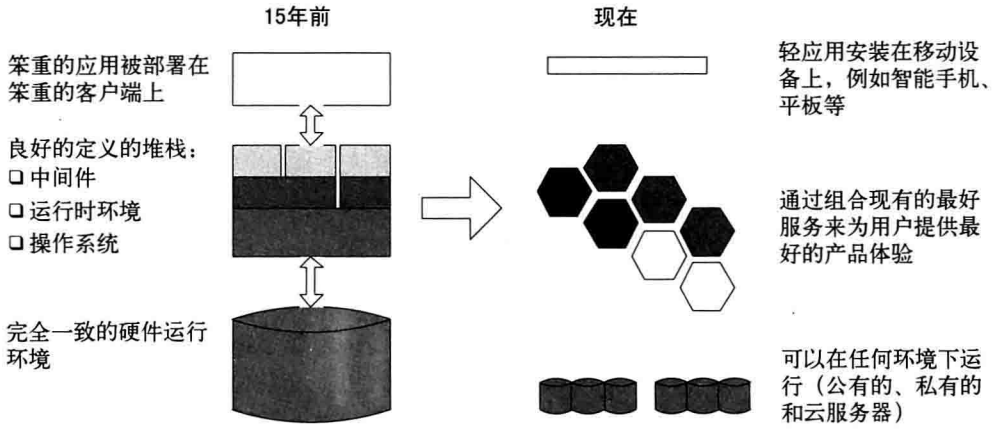


图1-1 互联网应用的演变

- ❑ 是否能够处理应用依赖的多样性和依赖库之间的不良反应？
- ❑ 是否能够适应硬件环境的多样性？
- ❑ 服务和应用之间的交互是否合理？
- ❑ 是否可以在多个平台之间快捷移动？

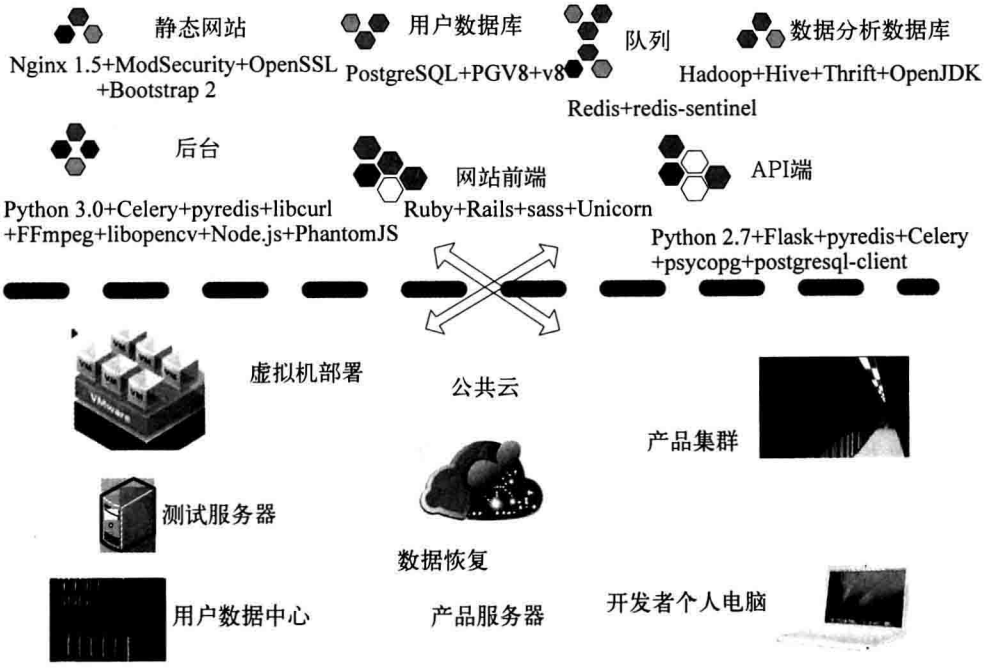


图1-2 应用的软件依赖和硬件运行环境的复杂性

想象一下将图1-2所示的各项应用和服务部署到各种硬件环境中的组合，每一个服务都有可能被部署在硬件环境中的一种甚至是多种。这些服务和运行环境的组合是一个可怕的矩阵，每一

个服务都需要考虑它将来可能会在多种运行环境下运行。而不同环境对服务的配置也不尽相同，一个环节出错都将给工程进度带来不可预知的迟滞。这给应用的开发人员带来非常大的麻烦，他们可能因此而陷入这些琐碎的事物之中。

有没有一种方法或是一样东西，能够让开发者一次性解决上面的所有问题呢？有，当然存在！那它是什么呢？在揭开谜底之前，我们先卖个关子。

如图1-3所示，20世纪60年代以前的海运，大多数散货通过船进行托运，托运人和承运商都担心货物放置在一起会发生挤压、受损等不良现象，例如一批咖啡豆和钢琴放在了一起，或是一批钢材和香蕉压在了一起等。而且，不同的运输方式之间转运也非常麻烦，港口的码头需要装卸各式各样的货物，其效率低下，而且转运到火车汽车的时候也得面临这个问题。不同货物和不同交通工具之间的组合也是一个巨大的二维矩阵，你会发现这个传统行业的问题和我们刚才遇到的问题是如此的一样！海运界最后在美国海陆运输公司的推动下，制定了国际标准集装箱来解决这个棘手的问题。

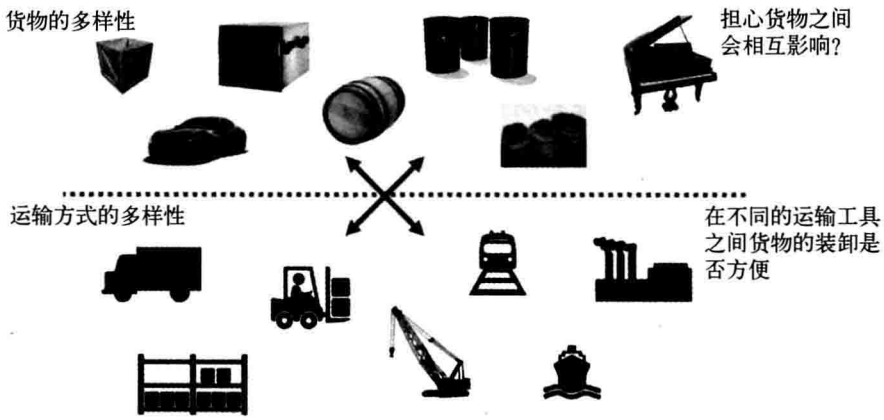


图1-3　1960年前的货运

如图1-4所示，所有的货物都可以通过集装箱指定的方式打包进集装箱内部，货物之间的相互影响被集装箱隔绝。集装箱是一个货物集对外的标准接口，无论是海运码头还是汽运，处理的都是标准统一的集装箱，一个标准让货物在多种运输方式下畅通无阻，这极大地加快了货物的装卸、堆积和运输速度，提高了运输的安全性，降低了运输成本。集装箱出现后，得到了大规模的推广，也进一步促进了大宗贸易的全球化。现在，集装箱的货运量已经占到了全球贸易的90%以上，每年大约有5000多艘货轮运送着2亿集装箱。

回到主题，在软件应用开发中，我们期待有一种东西，它能够像集装箱一样方便地打包应用程序，隔离它们之间的不良影响，使应用能够在各种运行环境下运行并且在平台之间易于移植。Docker，正是这个集装箱。在Docker出现之前，类似的技术已经存在，例如虚拟机和容器，然而它们能够解决前面4个问题里面的其中某些，却不能解决所有，这也正是它与众不同和受到热捧的重要原因。接下来，我们进入Docker的主题。

图1-4 标准集装箱的出现解决了运输方面的难题

1.1.1 Docker的概念

什么是Docker?

Docker是一个开源平台，它包含容器引擎和Docker Hub注册服务器。

- **Docker容器引擎**：该引擎可以让开发者打包他们的应用和依赖包到一个可移植的容器中，然后将其发布到任何流行的Linux机器上。
- **Docker Hub注册服务器**：用户可以在该服务器上创建自己的镜像库来存储、管理和分享镜像。利用Docker，可实现软件的一次配置、处处运行。

Docker是PaaS提供商dotCloud开源的一个基于LXC的高级容器引擎，其源代码托管在GitHub上，它基于Go语言并遵从Apache 2.0协议。Docker自2013年诞生以来，就受到业内的高度关注，从RedHat在RHEL 6.5中集成对Docker的支持，到Google的计算引擎支持Docker在其之上运行，再到国内百度的App引擎也是基于Docker部署的，以及越来越多的软件采用Docker部署。可以说，在云计算的背景下，Docker正带来一场软件开发的革命。由于Docker的影响越来越大，dotCloud公司也更名为Docker公司。随着Docker越来越成熟，它后面采用了自己的libContainer来替换LXC。

1.1.2 Docker的背景

前面提到，软件开发者急需一种像集装箱一样的容器来装配运输软件应用。而在云计算兴起后，服务和运行平台越来越多样，这种需求变得更加迫切。云计算兴起后，软件开发更趋向于组件组装，选取最合适的服务集合装配到一起构建出最终的产品，并将应用部署到各类云平台之上。云计算平台对硬件进行抽象和虚拟，按量提供给开发者，使得开发者从硬件管理问题中解脱出来，典型的云计算平台有亚马逊的AWS、国内的阿里云等。在云计算时代，硬件的部署和管理问题已经得到了很好的解决，然而软件的依赖、部署和管理问题依然存在，这主要体现在以下几个方面。