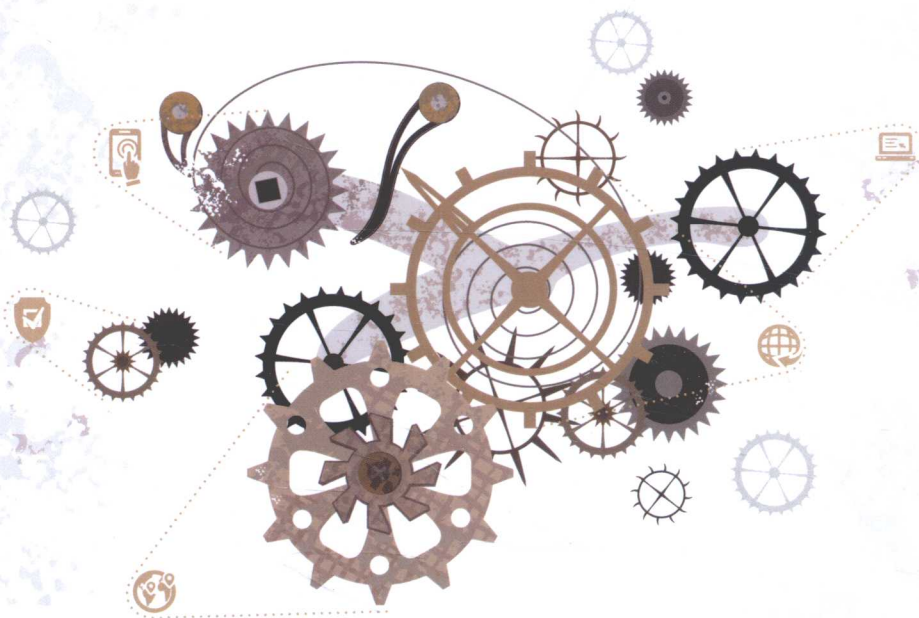


追随Google | Facebook | Amazon | Netflix
化大而复杂为小而简单，用快速交付支撑持续创新

微服务 架构与实践

王磊◎著



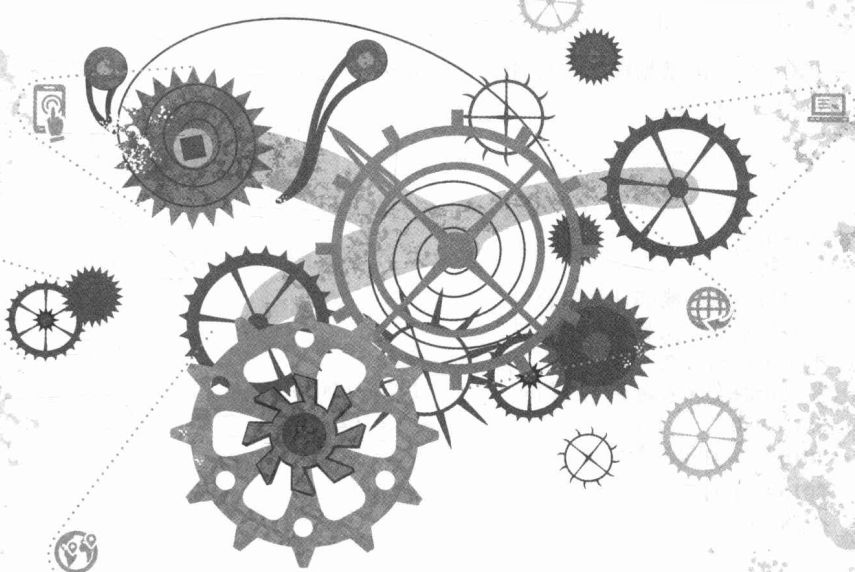
中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

微服务 架构与实践

王磊◎著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

随着 RESTful、云计算、DevOps、持续交付等概念的深入人心，微服务架构逐渐成为系统架构的一个代名词。本书首先从理论出发，介绍了微服务架构的概念、诞生背景、本质特征以及优缺点；然后基于实践，探讨了如何从零开始构建第一个微服务，包括 Hello World API、Docker 映像构建与部署、日志聚合、监控告警、持续交付流水线等；最后，在进阶部分讨论了微服务的轻量级通信、消费者驱动的契约测试，并通过一个真实的案例描述了如何使用微服务架构改造遗留系统。全书内容丰富，条理清晰，通俗易懂，是一本理论结合实践的微服务架构的实用书籍。

本书不仅适合架构师、开发人员、测试人员以及运维人员阅读，也适合正在尝试使用微服务架构解耦历史遗留系统的团队或者个人参考，希望本书能在实际工作中对读者有所帮助。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目 (CIP) 数据

微服务架构与实践/王磊著. —北京：电子工业出版社，2016.1
ISBN 978-7-121-27591-3

I. ①微… II. ①王… III. ①互联网络—网络服务器 IV. ①TP368.5

中国版本图书馆 CIP 数据核字 (2015) 第 273639 号

策划编辑：张春雨

责任编辑：刘 舫

印 刷：三河市双峰印刷装订有限公司

装 订：三河市双峰印刷装订有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：787×980 1/16

印张：14.75 字数：312 千字

版 次：2016 年 1 月第 1 版

印 次：2016 年 1 月第 1 次印刷

定 价：65.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件到 dbqq@phei.com.cn。

服务热线：(010) 88258888。

推荐语

天下武功，唯“快”不破。在当下互联网环境下，相信每一个 IT 从业者都能够深切地体会到“快”这个字对应用开发的影响。互联网产品的需求来得快，变得快，你的产品必须持续创新，不断给用户带来新的价值，否则用户会毫不犹豫弃你而去。用户期望的交付周期也极大缩短了，导致传统以月为单位的交付周期不得不被压缩到天甚至小时，这就要求互联网产品必须小步快跑，快速迭代，总之，就是要“快”。我拜访过很多互联网公司，知道做互联网要“快”，但是现实情况是他们中的大部分都面临着产品迭代速度越来越慢的问题。分析原因可以发现一个共同点，就是随着产品功能的累积，应用实现越来越复杂，代码规模越来越大，开发团队工作在一个逻辑复杂、模块耦合的单块架构应用之上，从而导致应用难于维护和更新，发布过程很长，而且随时面临发布失败的风险。微服务架构就能够很好地解决这个问题。微服务架构自 2010 年开始逐渐被大家熟知，通过对传统单块应用进行服务化切分，把一个大而复杂的问题化解为多个小而简单的问题，服务之间通过契约来约定依赖，做到服务独立发布和演进。今天，微服务架构已经被广泛运用在像 Google、Facebook 这样的大型互联网公司，为他们的快速交付和持续创新提供软件架构支撑。本书

中有大量微服务架构实战经验的总结，不仅仅有应用架构设计的内容，还涵盖了微服务大背景下应用测试、发布、日志、监控等方面，让读者可以全面应对微服务架构需求。

有人把微服务比作一把双刃剑，一方面它把单个问题域的复杂性降低了，服务可以独立更新、快速交付；但另一方面，面对一个由不同技术栈支撑的整体系统，运维和交付的难度增大了。在本书中，作者通过把微服务架构和当下热门的 Docker 容器技术、AWS 自动化部署相结合，向读者介绍了具有前瞻性的微服务自动化运维最佳实践，同时详细阐述了微服务化应用的持续交付流程和设计要领，不乏独到见解和技术细节，相信企业 CIO、软件设计师、架构师们读完这本书一定会受益匪浅。

郭峰

DaoCloud 联合创始人

微服务的出现，为运维又打开了一扇窗。微服务将整个业务系统拆分为相对独立的业务模块，并强调各个微服务都可以独立测试、独立部署、独立运行；微服务之间是一种真正的低耦合，就像汽车的各个零部件，哪个坏了，拆掉换个新的就能组装上；微服务面向产品而不是项目，这样，开发、测试、运维（系统、DBA 等）可形成更稳定的“小”团队，而不是项目周期一到，各个职能解散，各回各家；微服务配以 Docker，更可谓珠联璧合。这些都对运维提出了新的机遇和挑战，熟悉 DevOps、懂 Docker、沟通能力强的综合型运维人员，市场需求和价值更加突显。

纵览全书，说理清楚，用清晰明了的文字，帮助大家理清了很多似是而非的概念；图文并茂，图片既清晰又贴切，语言朴实、平易近人，没有从国外翻译过来的书籍那种生硬、别扭的感觉；理论结合实际，更多融合了作者实施微服务的一线经验。是一本非常用心、又可以实际落地的好书。

萧田国

开放运维联盟联合主席，高效运维社区创始人

当我开始阅读本书时，我并不知道微服务是否适合我。然而在读完前几章后，我发现这正是我在寻求的。微服务其实已经无处不在，在全球各地的技术会议上每个人都在讨论微服务。但它不是新东西，亚马逊和 Netflix 等大公司已经使用微服务架构建造分布式系统多年。在本书中，作者跟随领先者在本领域深入浅出地解释了一些重要的概念，并给出许多有价值的实践指导。最后，本书蒸馏出很多它处寻不到的信息，提供给大家一个通用的、坚固的开始。

龚勇

辉门（中国）有限公司，亚太区信息技术总监

微服务架构作为 SOA 在众多互联网公司中的成功新实践，是广大企业在互联网化进程中必须理解的概念。本书不仅讲述了微服务的基础理论，而且通过实例，深入浅出地涵盖了微服务构建过程中持续集成、构建、部署、持续交付以及日志聚合和运维的过程，体现了作者深厚的理论功底与扎实的实践经验，推荐阅读。

徐唤春

上海商派软件有限公司技术副总裁

微服务的概念初看简单清晰、容易理解，但在企业中的实际实施其实是一件很困难的事情。尤其很多计划实施微服务的公司在服务划分、DevOps 和相应的组织结构变化方面毫无经验，付出了实施的代价，却很难真正享受到微服务带来的好处。这本书总结了作者两年多在真实大型软件系统上实施微服务的经验和心得，具体指导了微服务实施在技术方面的实践，非常值得参考。

杨云

ThoughtWorks 首席咨询师，前支付宝资深架构师

随着应用系统的不断发展演进，单体应用变得越来越大，越来越复杂，导致扩展性差，资源优化难，维护成本高等问题。为了应对这一挑战，一种更加灵活、轻便、松耦合的设计架构——微服务架构，正受到越来越多应用系统开发者的青睐，它的敏捷开发、灵活部署、易扩展等特性，使它成为解决复杂应用的一把利器。微服务架构在具体实践中是怎样实施的？它在实施过程中存在怎样的困难和挑战？作者在本书中通过理论结合实践的方式，深入浅出地阐述了微服务的本质以及如何有效地、持续地交付微服务，并给出了许多有价值的实践指导，全书内容丰富，理论结合实际，推荐阅读。

薛正华博士

中国计算机学会高级会员，大数据专委会委员

前言

一直以来，系统的架构设计就是 IT 领域经久不衰的话题之一，是每个系统构建过程中极其关键的一部分，它决定了系统是否能够被正确、有效地构建。系统架构设计描述了在应用系统内部，如何根据业务、技术、组织、灵活性、可扩展性以及可维护性等多种因素，将应用系统划分成不同的部分，并使这些部分之间相互分工、相互协作，从而为用户提供某种特定的价值。多年来，我们一直在技术的浪潮中乘风破浪，扬帆奋进，寻找更优秀的系统架构设计方式来构建系统。

由来

随着 RESTful、云计算、DevOps、持续交付等概念的深入人心，微服务架构逐渐成为系统架构的一个代名词。那么微服务是否是业界期待已久的企业架构解决方案呢？在微服务架构的实施过程中存在着怎样的困难和挑战呢？

在过去两年多的时间里，笔者一直在探索和实践，并助力国外某房地产互联网门户，将其复杂的业务支撑系统逐渐演进为基于微服务架构的系统。

这期间也经历了从微服务的理论认识，到小范围实践、迭代，再到多个基于微服务构建的项目已经成功上线的过程。在感受微服务为开发实践、测试策略、部署、运维等带来改变的同时，也切身体会到使用微服务架构，对系统灵活性、可伸缩性方面的提升，以及对团队应对变化能力的提升。

结构

鉴于此，本书从笔者实践的角度出发，首先阐述了单块架构存在的弊端以及微服务的理论基础。接着通过实践部分，让读者能够体验从零开始搭建第一个微服务的过程，包括代码静态检查、AWS 基础设施构建、Docker 映像构建及部署、持续交付流水线、服务的日志聚合以及监控和告警。随后，探讨了笔者在微服务的实践过程中所积累的经验，包括基于 HAL 的通信机制、消费者驱动的测试，并通过一个真实的案例，帮助读者更好地理解微服务架构所带来的灵活性、易扩展性和独立性。

全书分成 3 部分，共 14 章。

第 1 部分为基础部分。包括第 1 章和第 2 章，概述了三层应用架构以及微服务架构。

第 2 部分为实践部分。包括第 3 章至第 10 章，通过一个具体的实例，从头到尾介绍了一个服务从需求到实现，再到构建、部署以及运维的整个过程。

第 3 部分为进阶部分。包括第 11 章至第 14 章，讨论了微服务的持续交付、测试策略、通信机制，并描述了一个使用微服务改造遗留系统的真实案例。

部分和部分之间几乎是相互独立的，没有必然的前后依赖关系，因此，读者可以从任何一个感兴趣的部分开始阅读。但是，每部分中的各章节之间的内容是相互关联的，建议按照章节的顺序阅读。

结合作者本人的工作经验和使用习惯，书中的大部分案例代码均采用 Ruby 编写，并且运行在 Mac OS 环境中。因此，读者最好对 Ruby 语言有一定了解，并且熟悉 Mac OS 或者 Linux 环境下的基本操作，以便能够更加顺利地阅读本书。

感悟

2014 年 10 月，我在 InfoQ 上发表了一篇题为《使用微服务架构改造遗留系统》的文章，

收到了很多朋友的反馈与建议，并有一些朋友和我积极探讨微服务架构的实践与心得。2015年初，基于在项目中积累的经验，我开始在博客上连载《解析微服务架构》系列的文章。机缘巧合，认识了电子工业出版社计算机出版分社的张春雨编辑，并在他的建议下，开始构思如何将自己在微服务实践中积累的经验以书的方式展现出来。但由于工作项目进度一直很紧，直到2015年4月，参加完北京QCon会议后，才开始真正动笔。

当时，自己曾信心满满地认为，应该能够比较顺利地完成这本书，因为大部分内容都比较熟悉，而且平时工作中也有笔记和积累。但当真正动笔之后才发现，理解领悟和用文字表述清楚是截然不同的两件事。有时候看似很容易的知识点，用文字解释清楚却并非易事。有时候工作中天天遵循的实践，总结清楚却需要花些工夫。当然，整个写作的过程，也是自己将微服务的相关知识点以及积累的经验从头到尾梳理的过程。通过编写本书，使得自己对这些知识的认识和理解更加深入和全面，受益匪浅。每次重新回过头来审阅书稿时，总会觉得某些知识点讲述得不够透彻，需要进行补充，或者应该用更好的方式将其展示出来。

微服务作为当下热门的话题之一，其涉及的部分已经不仅仅局限在技术层面，而是关注整个产品或者组织的价值。因此，它不仅涉及技术选择、服务划分、服务注册、服务安全、服务测试、服务运维，也包括微服务下团队组织架构的变化，全功能团队的构建，可靠的持续交付流水线，DevOps文化等。作为一本微服务架构的书，很难做到面面俱到。另外，由于时间仓促以及作者自身水平有限，书中难免有疏漏之处，在此敬请广大读者批评指正。在阅读本书的过程中，如有任何问题，可通过微信号：5109343 或邮箱：wldandan@gmail.com 与我联系。

最后，希望读者能享受微服务架构的实践之路。

王磊

2015年10月7日于西安

致谢

首先，要感谢我的家人，特别是我的妻子。在我占用大量周末、休假的时间进行写作的时候，她给予了极大的宽容、支持和理解，并对我悉心照顾且承担了全部的家务，让我能够全身心地投入到写作之中，而无须操心家庭琐事。没有她的支持和鼓励，这本书是无法完成的。

同时，要感谢 ThoughtWorks 提供了优秀的环境和平台，使我的技能能够得以施展，并且身处一群技术牛人中间，也得到了很多学习和成长的机会。

另外，感谢我的同事赵国庆和陈熙，他们为本书“Pact 实现契约测试”小节提供了示例代码。也要感谢我所在的团队对我一直以来的鼓励与支持，并在日常的工作中给予了我很多帮助。同时，感谢张凯峰、吕健、崔福罡、周星、邱俊涛（排名不分先后）等同事的帮助，在我写书的过程中，他们提出了很多宝贵的建议。谢谢各位！

最后，还要感谢电子工业出版社计算机出版分社的张春雨编辑和刘舫编辑，在我写作期间给予了很多指导和帮助。根据他们的建议，我对本书的内容做了很多修正，使内容更加充实，也更加易懂，本书能够出版，离不开他们的敬业精神和一丝不苟的工作态度。

目录

第 1 部分 基础篇

第 1 章 单块架构及其面临的挑战	3
1.1 三层应用架构	4
1.1.1 三层应用架构的发展	4
1.1.2 什么是三层架构	5
1.1.3 三层架构的优势	6
1.2 单块架构	6
1.2.1 什么是单块架构	6
1.2.2 单块架构的优势	7
1.2.3 单块架构面临的挑战	8
1.3 小结	12
第 2 章 微服务架构综述	13
2.1 什么是微服务架构	13
2.1.1 多微才够微	14
2.1.2 单一职责	17
2.1.3 轻量级通信	17
2.1.4 独立性	19
2.1.5 进程隔离	20

2.2	微服务的诞生背景	22
2.2.1	互联网行业的快速发展	23
2.2.2	敏捷、精益方法论的深入人心	23
2.2.3	单块架构系统面临的挑战	23
2.2.4	容器虚拟化技术	23
2.3	微服务架构与 SOA	24
2.3.1	SOA 概述	24
2.3.2	微服务与 SOA	25
2.4	微服务的本质	26
2.4.1	服务作为组件	27
2.4.2	围绕业务组织团队	28
2.4.3	关注产品而非项目	29
2.4.4	技术多样性	31
2.4.5	业务数据独立	32
2.4.6	基础设施自动化	33
2.4.7	演进式架构	33
2.5	微服务不是银弹	34
2.5.1	分布式系统的复杂度	35
2.5.2	运维成本	36
2.5.3	部署自动化	36
2.5.4	DevOps 与组织架构	37
2.5.5	服务间的依赖测试	37
2.5.6	服务间的依赖管理	37
2.6	小结	38

第 2 部分 实践篇

第 3 章	构建第一个服务	41
3.1	场景分析	41
3.2	任务拆分	43
第 4 章	Hello World API	45
4.1	API 实现	45
4.1.1	开发语言——Ruby	45
4.1.2	Web 框架——Grape	46
4.1.3	API 的具体实现	47

4.2	代码测试与静态检查	50
4.2.1	代码测试	50
4.2.2	测试覆盖率统计	53
4.2.3	静态检查	54
4.2.4	代码复杂度检查	57
第 5 章	构建 Docker 映像	61
5.1	定义 Dockerfile	61
5.2	配置 Docker 主机	63
5.3	构建 Docker 映像	64
5.4	运行 Docker 容器	64
5.5	发布 Docker 映像	65
5.6	小结	69
第 6 章	部署 Docker 映像	71
6.1	基础设施 AWS	71
6.2	基础设施自动化	73
6.3	部署 Docker 映像	80
6.4	自动化部署	81
6.5	小结	84
第 7 章	持续交付流水线	85
7.1	持续集成环境	85
7.2	提交阶段	87
7.3	验证阶段	91
7.4	构建阶段	91
7.5	发布阶段	94
7.6	小结	96
第 8 章	日志聚合	97
8.1	日志聚合工具简介	97
8.2	Splunk 的核心	99
8.3	安装 Splunk 索引器	100
8.4	安装 Splunk 转发器	101
8.5	日志查找	102

8.6	告警设置	103
8.7	小结	104
第 9 章	监控与告警	105
9.1	Nagios 简介	105
9.2	Nagios 的工作原理	107
9.3	Nagios 安装	108
9.4	Nagios 的配置	109
9.5	监控 products-service	111
9.6	告警	113
9.7	小结	114
第 10 章	功能迭代	115
10.1	定义模型	116
10.2	持久化模型	117
10.3	定义表现形式	119
10.4	实现 API	122
10.5	服务描述文件	125
10.6	小结	127

第 3 部分 进阶篇

第 11 章	微服务与持续交付	131
11.1	持续交付的核心	132
11.2	微服务架构与持续交付	133
11.2.1	开发	133
11.2.2	测试	137
11.2.3	持续集成	139
11.2.4	构建	139
11.2.5	部署	140
11.2.6	运维	143
11.3	小结	144
第 12 章	微服务与轻量级通信机制	145
12.1	同步通信与异步通信	145
12.1.1	概述	145

12.1.2	同步通信与异步通信的选择.....	146
12.2	远程调用 RPC.....	147
12.2.1	远程过程调用的核心.....	147
12.2.2	远程方法调用.....	148
12.2.3	远程过程调用的弊端.....	148
12.3	REST.....	149
12.3.1	概述.....	149
12.3.2	REST 的核心.....	150
12.3.3	REST 的优势.....	152
12.3.4	REST 的不足.....	152
12.3.5	本节小结.....	155
12.4	HAL.....	155
12.4.1	概述.....	155
12.4.2	HAL 的核心.....	156
12.4.3	HAL 浏览器.....	160
12.5	消息队列.....	161
12.5.1	核心部分.....	162
12.5.2	访问方式.....	163
12.5.3	消息队列的优缺点.....	164
12.6	后台任务处理系统.....	165
12.6.1	核心部分.....	165
12.6.2	服务回调.....	166
12.6.3	一个例子.....	167
12.6.4	后台任务与微服务.....	169
12.7	小结.....	170
第 13 章	微服务与测试.....	171
13.1	微服务的结构.....	171
13.2	微服务的测试策略.....	173
13.3	微服务的单元测试.....	175
13.3.1	单元测试综述.....	175
13.3.2	单元测试的内容.....	176
13.4	微服务的集成测试.....	179
13.4.1	集成测试综述.....	179
13.4.2	集成测试的实施方法.....	179

13.4.3	集成测试的内容	180
13.5	基于消费者驱动的契约测试.....	181
13.5.1	集成测试存在的弊端.....	181
13.5.2	什么是契约	183
13.5.3	什么是契约测试	184
13.5.4	契约测试的方法	185
13.5.5	Pact 实现契约测试.....	187
13.5.6	一个例子	192
13.5.7	本节小结	205
13.6	微服务的组件测试	205
13.6.1	组件测试概述	205
13.6.2	组件测试的方法	206
13.6.3	本节小结	207
13.7	微服务的端到端测试	208
13.7.1	端到端测试概述	208
13.7.2	端到端测试的内容	208
13.7.3	本节小结	209
13.8	小结	210
第 14 章	使用微服务架构改造遗留系统	211
14.1	背景与挑战	211
14.2	改造策略	212
14.2.1	最小修改	212
14.2.2	功能剥离	212
14.2.3	数据解耦	213
14.2.4	数据同步	213
14.2.5	迭代替换	214
14.3	快速开发实践	215
14.3.1	快速开发模板	215
14.3.2	代码生成工具	217
14.3.3	持续集成模板	217
14.3.4	一键部署工具	217
14.4	微服务架构下的新系统	218
14.5	小结	220