



架构探险

轻量级微服务架构 上册

黄勇 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>

架构探险

轻量级微服务架构 上册

黄勇 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本系列从开发与运维两方面分别对微服务架构的实践过程进行描述,全套分为上下两册,上册偏重于开发,下册偏重于运维。在上册中读者会学习到微服务架构所需的开发技能,包括使用 Spring Boot 搭建微服务开发框架,使用 Node.js 搭建微服务网关,使用 ZooKeeper 实现微服务注册与发现,使用 Docker 封装微服务,使用 Jenkins 部署微服务。通过阅读上册,读者可轻松搭建一款轻量级微服务架构。

本书适合对微服务实践感兴趣,以及想成为微服务架构师的人员阅读。

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

轻量级微服务架构.上册 / 黄勇著. —北京:电子工业出版社,2016.9
ISBN 978-7-121-29804-2

I. ①轻… II. ①黄… 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(2016)第203295号

责任编辑:陈晓猛

印 刷:三河市鑫金马印装有限公司

装 订:三河市鑫金马印装有限公司

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:13.5

字数:259.2千字

版 次:2016年9月第1版

印 次:2016年9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, 88258888

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

本书咨询联系方式:010-51260888-819 faq@phei.com.cn。

序一

微服务，应用开发的新起点

研究现在的软件体系，不难发现：现在的软件专家们仍需要与大量的需求、设计、代码的细节打交道。出于项目实施时间、投入资源等方面的限制，软件往往以实现若干具体的用户功能需求为目标。专家们没有时间，也没有精力去追求软件的美学目标。日复一日，随着用户需求的变化，软件项目成为大量代码的随机而无序的堆积，奇丑无比。许多功能成一旦完成项目，就恐避之不及，不愿再去碰自己几个月来夜以继日的劳动成果。

黄勇的《架构探险：轻量级微服务架构》一书，融合了软件设计的最新理念，系统性介绍了微服务的设计、开发、运维等各方面，书中不仅仅是技术的描述和讲解。看到黄勇在技术方面这么多年的不断积累和提炼，我很欣慰。

微服务的兴起和移动应用的快速发展相对应。移动应用的基本框架是事件和响应，用户在碎片化的时间和地点，按自己的节奏完成综合起来是一个复杂的事情。这不同于传统软件，往往是流程和复杂业务驱动的过程和算法。移动计算所需要的跨界沟通和协作，在传统应用架构中则很难实现，而这恰恰是微服务的优势所在。微服务从技术的视角，使用各种协议和框架，便于不同开发者软件碎片之间的协同工作。但是各种软件交互协议并不稀缺，总是不断地出现各种协议的标准。微服务的成功使用，需要注意微服务在软件重用方面的能力，正是这种能力，使得微服务的使用更加具有普遍的意义。不同于传统的构件或服务，微服务的调用参数接口具有更大的融合性和灵活性。微服务的调用，不需要拘泥于严格的数据类型，而是遵循更高层次的语法结构。特别是应用软件走向人工智能的时代，微服务将更深的演化带来更智能的微服务对接。微服务对于传统的过程式软件，是一个破坏性的改变。这一特征既给了微服务无限的想象空间，也给实施带来了许多挑战。并不是每个应用，特别是成熟领域的软件应用都适合微服务的改造。但是对于移动应用领域和跨应用跨企业的对接，是一个很必要的选择。

我早年写了一些关于 SOA 和“面向构件”方面的东西，有人问我：“SOA 和微服务有何差异？”我认为：SOA 的核心还是企业级应用。最大的差异，是微服务对于调用参数的宏定义，

语义的适应性，使得微服务的复用性大大提升。比较有意思的是，新的微服务调用参数体系，和普元 EOS 非常类同，15 年前我们就是这样设计的。微服务是 SOA 后的一个突破性的东西，不是简单的落地，SOA 本身也有落地，比如普元的 EOS 就是 SOA 落地后的产品。SOA 到微服务一方面是网络协议的提升，更加适应跨应用跨企业的服务调用。还有人问我：“构件和微服务到底有什么区别？”我认为：构件是装配、开发的视角，一台机器由一个个构件装配而成；服务是运行、传动的视角，能量从活塞到轮胎传播。微服务用代码来开发，但微服务可以当成一个构件装配到应用。两边视角不同，但是微服务给了软件模块更多生命力。构件是静态的，服务是动态的。

这本书对于微服务架构的介绍非常完整，如果你和你们的企业正在开发移动应用，或者对已有的应用正在规划架构性的重构，这本书很值得一读。

——黄柳青

序二

微服务，我们如何与你相处

微服务来了，有了“服务”这两个字，这注定又是个一说就明白、一举例就糊涂、一讨论就吵架的概念。微服务的出现有其必然的商业背景和架构哲学，如何更好地认识微服务的内涵、如臂使指地应用微服务架构，还是有着很多挑战的，这也许就是本书被命名为“架构探险”的原因。

企业数字化转型驱动架构升级

互联网经济深刻改变了我们身边的商业环境，消费者的生活方式日益数字化，人们可以在任何时间、任何地点利用线上、线下渠道体验无缝购物，运用社交媒体表达自我，企业也在运用多种技术手段，发挥数字化潜力，改善客户联系，促进企业业务模式的转型。Gartner 认为，数字化就是把人、事、物和商业联系起来，建立新的商业模式。未来的企业都将是 IT 企业，IT 将从后台走向前台，从 ERP、CRM 等内部流程优化为主的业务，逐步转向内外兼修的模式，从而实现商业创新。

这一变化要求 IT 架构更加灵活地与上下游企业协作，更加快速地响应客户的个性化需求，更加弹性地应对无时不在的客户请求并提供良好的客户体验，同时云计算、大数据等技术的出现也为上述改变提供了新的技术选择，我们正面临 B/S 多层架构出现后新的一次架构升级，而微服务架构就在这个架构升级过程中应运而生。

分而治之的哲学是微服务的理论基础

把大的问题分解为容易解决的小问题，找到小问题的解决办法，再来解决大问题，这就是分而治之的哲学。正如万事万物由分子、原子组成一样，软件也可以分解为基本单元，以这样的基本单元进行开发、测试、维护，是解决大规模系统建设的思路。分而治之首先要解决如何分的问题，企业软件的分法应该是以业务驱动的，而不是以技术驱动的，也就是分解为独立的

业务逻辑，而这样的不可再分的业务逻辑就是微服务。

凡事有一利必有一弊，细分为微服务后，势必带来部署、测试、信息集成难度的提高，分而治之除了“分”，还需要“治”。传统恐龙型 ERP 是一个面向组织的软件，完备、复杂、响应变化慢，适合业务稳定的情况，而在数字化时代，客户个性化的要求让我们从这种面向组织的软件逐渐演变为面向个体的软件。例如，从前的 EHR 软件是为人力资源部门服务的，整体开发、整体实施，而现在我们会从个体的角度规划软件，可以先从招聘专员开始做一个面试管理的流程，逐步推出新的流程，完善现有的流程。这些面向个体的流程就是微应用，企业应用将由无数个微应用组成。微服务则是一个技术概念，能更好地解决微应用的技术实现问题，是一个事物的不同侧面，所谓“横看成岭侧成峰，远近高低各不同”，微服务和微应用是事物的一体两面。正因为微服务实际就是一个业务逻辑，因此做好微服务需要从微应用的维度考虑，将分解开的逻辑形成一个整体，要从多渠道接入、客户体验、数据管理、应用交付、运维全方位的视角考虑，这就是分而治之中实现“治”的体验，也是微服务架构需要解决的问题。

站在 SOA 的肩膀上践行微服务

微服务是一个新概念，但这绝不是一个全新架构，更不是一个包治百病的架构。由于有服务二字，很容易让人联想到面向服务架构(SOA)，其实微服务架构属于应用技术架构，和以 B/S 为代表的三层架构相对应，强调将巨石型应用拆分为由微服务组成的应用，在数据上也视情况从集中的存储拆解为更小的存储单元。而 SOA 属于企业架构的范畴，从企业架构出发把业务分解为不同领域的服务，不同物理系统提供不同服务，注重系统之间通过服务互联互通的规范，对服务如何实现并不关注。因此，面向服务架构的服务应该是一个业务意义的服务，而微服务是系统中的技术服务，更关注服务的实现，虽然提供了业务意义的服务，但是不能混为一谈。微服务使用也不是无限度的，事实上由于数据一致性等问题的限制，不能无限度拆分微服务，可以把微服务分为系统对外提供的远程服务、系统内部的远程服务和系统内部的本地服务，显式声明、明确职责。事实上，在企业架构上使用 SOA 支撑业务，而在应用技术架构上使用微服务架构，是一个合适的选择。

黄柳青博士是我和黄勇共同的导师，他在 2004 年所著的《软件的涅槃》一书中指出：“互联网时代的企业应用定义，正发生革命性的变化...横向的部门互动、实时的企业间互动、多样的交互渠道、灵活的业务规则，使得原有意义上的独立应用不复存在...对软件设计者来说，能直观地分割并具有最小内部耦合的软件结构是简约之美...美的软件是软件企业与软件开发者的终极目标”，那时候他把这种全新的软件生产模式称为“面向构件”。回头看来，微服务正是“面向构件”在数字化时代的解读，用微服务架构实现软件之美，加速企业数字化转型。

专家推荐

(排名不分先后)

在几年前我们还在大谈 SOA 架构，而随着 Docker 的普及，微服务逐渐成为近年来备受关注的话题，为企业的架构治理带来了新的思路，本书从微服务的理念、开发框架，到微服务网关、注册与发现、微服务的封装与部署几个角度，较为系统地介绍了微服务的实践过程，非常值得大家参考。

——陈康贤，淘宝技术部技术专家，《大型分布式网站架构设计与实践》作者

书中围绕着如何构建微服务逐渐展开，详细介绍了 Spring Boot、Node.js，以及如何使用 ZooKeeper 进行服务治理，在 Docker 上部署微服务，等等。通过这本书读者能够从零基础学习如何构建微服务应用，技术涵盖了开发、测试、运维等环节，可见作者技术功力之深厚。我将此书推荐给对微服务感兴趣的朋友们，相信你们一定能从书中获益良多，快速掌握微服务架构！

——黄哲铿，1 药网技术副总裁，《技术管理之巅》作者

黄勇是 InfoQ 非常知名的作者，他创作了很多优秀的内容，深得社区喜欢。本书以实践为主，内容涵盖了微服务的整个生态，推荐想转型微服务架构的同学阅读！

——郭蕾，InfoQ 主编

非常有幸结识了黄勇，并拜读了他在微服务方面的沉淀总结。从这本书的字里行间，能感受到作者对技术的热爱和厚积薄发的功力。微服务是当下技术架构的演化方向，但并非选用了一种框架就有了微服务，微服务更多地是工程化的底蕴和架构上的落地。黄勇以严谨、认真的笔触，井井有条地将微服务的每个细节讲述清楚并加以落地，实属难得。希望阅读本书，能给读者带来对微服务全方位的提高。

——韩陆，《Java RESTful Web Service 实战》作者

买书分三种，一种是需要好好浏览内容才决定是否购买；第二种是看作者，只要是某人写的就可以买；黄勇的书就是第三种，兼顾了前两种，且内容接地气，结构安排合理，所以一定要买！

——红薯，开源中国创始人

软件架构的核心是管理复杂度，微服务带来的模块化、隔离性无疑是解决这一问题的一剂良药。但是一提起服务化，我们之前的印象通常是这样的：开发成本没有降低，运维成本增加了很多，需要部署很多应用，还要引入一系列重量级的中间件。实际上，时至今日，Spring Boot 和 Docker 等技术的兴起，已经使得微服务的实施变得更加容易。可惜国内并没有成体系的资料，讲解如何运用这些新技术，来搭建自己的微服务架构。作者勇哥结合了时下热门的技术，提出了一套行之有效的架构。不但简单易落地，而且全面覆盖微服务的各个方面，对于想要实施微服务的企业具有很大参考价值。勇哥是一个资深的 blogger，讲技术有趣而不失深度。虽然书中内容跨度较大，但是仍然可以在轻松愉快的氛围中完成阅读。

——黄亿华，票牛网架构师，开源爬虫框架 WebMagic 作者

It is no surprise that smart developers who have experience building systems at scale are using Spring Boot. Spring Boot makes building production-worthy systems quick and easy. I'm happy to see Leo Huang's book giving a quick look not just at Spring Boot itself but at some of the production-ready features in Spring Boot. Leo has experience building large systems at scale in Alibaba and can appreciate how important it is to build production-ready systems.

——Josh Long, Spring Developer Advocate

本书以微服务的生命周期为主线，系统地介绍了微服务技术架构的选型，微服务的开发和测试，基于 Docker 容器的部署，以及基础设施自动化和持续交付等。围绕各个环节，给出了技术选型和详尽的使用说明。对于微服务初学者，是本难得的入门好书。

——李林锋，华为软件平台开放实验室资深架构师，
《分布式服务框架原理与实践》和《Netty 权威指南》作者

低耦合、分而治之的思想贯穿人类软件开发的全部历史，在目前阶段，代表这种思想最热门的架构方法非微服务莫属。本书从实践角度，带你领略目前构建微服务的几种主要工具，一

窥微服务的个中奥秘。

——李智慧，宅米 CTO，《大型网站技术架构：核心原理与案例分析》作者

黄勇老师曾出品了《程序员》之架构技术与实践的封面专题，对架构和新技术有着深入的理解和浓厚的兴趣。在一年前的面访中，就巨细谈及了贵司的微服务实践，本书必是一年多来宝贵经验的总结。同时，在社区和技术大会里上，微服务话题往往受到热捧，聚焦实践的本书，有有助于将概念化的技术落地，是一本不可多得且适合国内开发者学习的好书。

——钱曙光，CSDN 资深编辑/记者，多年关注互联网架构领域

SOA 从企业级应用到互联网领域火了很多年，曾经是我招聘架构师的必考题目之一，但 SOA 在大型系统的落地从来都是高难度动作，令许多架构师欲仙欲死。如今又兴起了微服务架构，要把 SOA 进行到底，实现彻底的服务化，从此世间再无系统切分，只有微服务小而美好。那么到底如何实现微服务呢？黄老师这本书教我们轻松上手，一步步把理想变成现实，体现出多年实战派的底蕴，是一本不可多得的武功秘籍，期待下半部早日面世！

——史海峰，当当网架构部总监

近年来，微服务俨然成为行业内广受关注的热点。不论是微服务的价值，还是微服务的阻碍，都是行业在架构技术选型中最为关心的前提。除此之外，技术的践行流程，对现有组织架构、软件模式的影响，都是决策者不敢忽视的要素。我很庆幸看到，国内能诞生这本微服务领域的巨著。本书从架构发展史的角度，阐述了微服务兴起的客观性与必然性；从技术的角度，深入分析了践行微服务的种种要点；更从实践的角度，通过案例事无巨细地帮助读者去体会、理解、掌握微服务。实属呕心沥血之作，极力推荐大家阅读。

——孙宏亮，DaoCloud 技术合伙人，《Docker 源码分析》作者

黄勇的这本书从微服务实操的角度，通过在微服务架构体系的不同关注点，选择多样而务实的技术栈，为大家全方位地阐述了微服务架构体系的各种最佳实践，对微服务感兴趣的同学不容错过。

——王福强，《Spring 揭秘》和《Spring Boot 揭秘》作者

微服务架构，虽然诞生时间不长，却已成为软件架构领域讨论的热点。微服务的概念看似简单，但涉及诸多方法论和实践积累，这就是为什么有人说它非常好，但就是“玩不起”。随着微服务生态系统的日趋完善，微服务架构的讨论也从 API 接口、服务间通信、接口测试、基础设施自动化等，逐渐扩展到了 API 网关、微服务的注册与发现、Docker 封装与部署、持续交付以及运维体系的优化等多方面。本书结合作者过去多年的实战经验，深入浅出地梳理了微服务构建过程中遇到的诸多挑战，并给出了切实可行的解决方案（如何使用 Spring Boot 构建服务、使用 ZooKeeper 注册服务，如何结合 Docker 封装服务和发布服务等），是一本能帮助读者立刻动手、落地微服务的好书。同时，作者从开发和运维两个角度入手，详细地剖析了微服务实施过程中，如何有效解决“最后一公里”的部署以及运维难题。纵览全书，说理清楚，图文并茂，理论结合实际，是一本非常用心，又注重实操的好书，对企业的微服务架构实施，具有很大的参考意义，相信企业的架构师、软件开发人员、运维人员读完这本书一定会受益匪浅。

——王磊，尚度元科技 CTO，《微服务架构与实践》作者

微服务是近几年的一大热点，其模块化、跨语言和自治隔离等思想，有望大幅降低研发和运维成本。微服务架构，无论对传统企业，还是互联网公司，都会有很大影响。黄勇老师结合了 Spring Boot、Jenkins 和 Docker 等热点技术，对微服务的整个生命周期做了全面介绍，通俗易懂、深入浅出，致力于打造微服务领域最佳实践，不失为一本好书。

——吴其敏，携程框架研发部高级总监，开源分布式实时监控系統 CAT 作者

当今，微服务已经不是概念，而是势不可挡的潮流，它在大型互联网电商类企业，已有丰富的实践，效果很好。但对于其他有志于向微服务架构转型的技术爱好者，微服务如何落地还存在很多不清楚的地方，本文从细节入手，结合具体实例，娓娓道来，为大家提供一个很好的微服务实践参考，带领大家走进微服务之门。

——王庆友，1号店首席架构师，现独立架构顾问，《架构的本质》作者

软件开发从来没有银弹，微服务也不是。我认为微服务本质上是要解决一个可伸缩性的问题，以应对访问的增加、业务复杂度的增加和开发团队人员的增加。黄勇在本书中详细解释了实践微服务必须要面对的架构模式，包括服务注册与发现、API 网关、以及简单部署系统的搭建，并辅以样例代码，对于正面临可伸缩性问题的开发人员有很大的参考价值。

——许晓斌，阿里巴巴高级技术专家，《Maven 实战》作者

近年来，软件开发领域的新思想、新方法、新工具、新实践层出不穷。简直有令人应接不暇、目眩神迷的感觉。要想走出这团迷雾，微服务是纲，容器化、自动化运维、自动化部署、服务监控与治理等等，都是目。通过阅读本书，纲举目张，则一切将尽在掌握！

——庄表伟，华为内源平台架构师，《开源思索集》作者

随着移动互联网的崛起，Web 网关越来越重要，本书从 Web 网关的视角带领大家学习微服务架构。通过本书可以学习到如何使用 Spring Boot 与 Docker 等技术构建 Web 型微服务架构，值得 Web 开发人员学习。

——张开涛，“开涛的博客”博主

微服务是最近几年在架构方面比较热的一个话题，本书从概念到具体的落地，比较系统地介绍了微服务从构建到部署等环节的知识和具体方案，是了解和学习微服务相关技能的一本好书。

——曾宪杰，美丽联合集团副总裁，《大型网站系统与 Java 中间件实践》作者

读者推荐

分合：“凡用兵之法，三军之众，必有分合之变”——语出《六韬之犬韬》，“分合”乃御变之道。当下盛行的微服务技术无非是在“分”与“合”层面的实践与细化。由上层来俯视下层为“分”，由下层来仰视上层为“合”；由外而内为“分”，由内而外为“合”；“分”即是“合”，“合”即是“分”；当“分”则“分”，当“合”则“合”，合乎自然。极力推荐对微服务架构感兴趣的朋友们来品读一下黄勇老师的这部新作，以领略“分合”之妙。

——杨新伦，饿了么资深工程师

上善若水，水善利万物而不争，故几于道！黄勇老师之前的架构探险一书可谓善利万人，一时洛阳纸贵。而这本微服务架构，更是从开章的循循善诱，到后来的深入浅出，无不体现了黄勇老师的丹青妙笔与渊博学识，极力推荐对微服务感兴趣的朋友们，值得一观。

——流浪狗，架构探险读者俱乐部资深群友

第一次阅读黄勇老师的书籍是《架构探险——从零开始写 Java Web 框架》，该书一切都是从零开始基于 Servlet 搭建一款轻量级 Java Web 框架。作为一名 Java Web 程序员，平时都只是使用 Java Web 框架，知其然；一口气读完本书，Step by Step 让框架跑起来，从而对架构有了更深入的理解，知其所以然，并将其架构思想成功地应用于实际工作中，卓有成效，受益匪浅。随着 RESTful、DevOps、服务解耦等概念的推广流行，微服务架构逐渐成为大型系统架构的一种趋势。黄老师推出的新书通过揭秘 Spring Boot 快速构建微服务体系，着实详解了服务的注册与发布、封装部署、日志监控、服务治理等微服务的方方面面，教授大家手把手的微服务搭手入门。读完样章，觉得依然不错，推荐给大家。希望能早日拿到实体书，继续在微服务架构中探险。

——mickey，沉迷于十余年开发的南航集团软件工程师

在伟人的实践论中，有这样一句话，“真理的标准只能是社会的实践”。微服务的概念早就已经火遍大江南北了，可是纸上得来终觉浅，绝知此事要躬行。到底什么是微服务？到底怎么做微服务？到底选哪本书？走过路过不要错过，看黄勇老师为你庖丁解牛，轻量级微服务架构，有这一本足矣。

——齐鑫，寺库网开发主管

Java 服务端开发最大的障碍是资源太多，变化太快，很多开发者一上来就会被各种名词的海洋弄得晕头转向，往往只能照猫画虎，而不能真正掌握其中的奥义，造成 Java 服务端开发这个领域初级程序员多，能够成长为独担一面的架构师的寥寥无几。黄勇的《架构探险》系列则是深入本质，从实际的业务场景出发，一步一步地用代码和框架去解决问题，十分接地气。我认为国内少有的优秀技术图书，读来收获满满，推荐给所有想成为架构师的程序员。

——何小敏，广联达科技股份有限公司架构师

黄勇老师这本书，不仅从宏观角度给我们介绍了什么是微服务架构，更重要的是从微观角度教会我们如何搭建一款轻量级微服务架构，全书深入浅出，给我们带来了淋漓快感，读完样章后仍旧意犹未尽，值得大家反复阅读和实践。

——快枪手，伪全栈工程师

前言

微服务是近年来备受关注的话题，它的出现让我们想起了十年前的 SOA (Service-Oriented Architecture, 面向服务架构)，但它比传统的 SOA 更容易理解，也更容易实践，它将“面向服务”的思想做得更加彻底。

当国外一些知名技术公司成功实践了微服务以后，这股热潮就吹遍了国内的大街小巷，大家街头巷尾都在聊微服务，对它众说纷纭且褒贬不一。有人说它非常好，但就是“玩不起”，为何会这样呢？

我们不妨带着这个问题来简单介绍一下，究竟什么才是微服务。

微服务是一种分布式系统架构，它建议我们将业务切分为更加细粒度的服务，并使每个服务的责任单一且可独立部署，服务内部高内聚，隐含内部细节，服务之间低耦合，彼此相互隔离。此外，我们根据面向服务的业务领域来建模，对外提供统一的 API 接口。微服务的思想不只是停留在开发阶段，它贯穿于设计、开发、测试、部署、运维等软件生命周期阶段。

可见，我们提到的微服务，实际上是一种架构思想，我们不妨称它为“微服务架构”。

微服务架构看起来如此之好，我们真的就需要它吗？

微服务架构建议我们按照业务来切分服务，我们完全可以选择最合适的技术来实现具体的服务，只需确保对外提供的 API 接口保持一致即可，也就是说，微服务架构使我们技术选型的自由度更加宽广了。既然系统可拆分为多个服务，这样非常有利于我们对每个服务进行监控，可不断收集每个服务的性能指标数据，当某个服务出现性能瓶颈时，会发出预警，我们可随时水平地扩展该服务，以支撑更大的流量，而不至于复制整个系统。由于服务之间彼此隔离，相互之间不会产生影响，因此我们可借助技术的手段来实现自动化部署，这会使我们的部署过程变得更加高效。

其实微服务架构的优点数不胜数，但是大家可能还是不敢用，因为它对我们的技术要求具有一定的挑战。比如，我们需要一个自动化部署系统，也需要解决分布式系统带来的一系列问题，还需要服务之间能做到彼此隔离且互不影响，同时还不能影响通信过程中所带来的性能开销。因此很多人认为，只有大公司或强悍的技术团队才能玩得起微服务架构，自己只能“远观”

却不能“近玩”。甚至还有人认为，微服务架构实际上就是以前谈论多年而难以落地的 SOA。

实际上，我们认为微服务架构的本质仍然符合 SOA 思想，只不过它比 SOA 更容易落地。为了证明这件事情，我们经过了大量的实践，借助了许多优秀的开源技术，搭建了一款“轻量级微服务架构”。实践证明，该架构不仅可以适应大中型公司的业务变化，还能满足中小型公司的快速增长。我们真心地希望这款轻量级微服务架构能够帮助更多的技术爱好者以及更多的技术团队，顺利地走出技术困境，以全新的视角去迎接新的挑战。

不得不提醒大家的是：微服务并不是万灵丹！它不能包治百病，我们更不要为了微服务而去微服务。而是需要根据自身的情况，灵活地选择最合适的技术，通过技术的手段实现更高的目标。

为什么写这本书？

我一直很关注服务化方面的技术，记得在 2014 年，我偶然发现国外技术博客中有人在写关于“微服务”方面的概念与技术。坦白地讲，当我第一次看到微服务时，并没有对它产生浓厚的兴趣，我当时认为微服务是笨重的，它和传统的 SOA 没有太大的区别，同样都是服务化，只不过微服务更加细粒度而已。直到 Docker 容器技术逐步成熟起来，越来越多的人开始使用 Docker 来封装应用程序，并借助 Docker 技术让软件交付变得更加灵活而高效，这让我不由地对微服务的未来产生了强烈的期待，我坚信微服务将伴随着 Docker 技术成为未来软件开发与运维的核心武器。

我开始疯狂地学习微服务，研究它的各种架构模式与应用领域，开始自己动手做一些练习，并在公司内部大力推广这种新架构。在实践中，我获得了一点收获，曾在一些技术大会与企业内训中讲过微服务的原理与实践。我发现一个问题，虽然大家对微服务都非常关注，但往往却不知应该如何开始，应该使用哪些技术来搭建微服务架构，以及在实践中应该如何避免掉进坑里。甚至有些人还认为微服务只是大公司才玩得起的东西，因为它需要借助像 SOA 那样重量级的基础设施，需要付出大量的成本，但收益却不一定。其实，我想告诉大家的是，微服务架构并非这样，它应该是轻量级的，它应该是很容易上手的，它应该是任何公司想用就敢用的。虽然国内外已经有几本关于微服务方面的好书了，但我仍然希望这本书能为大家在微服务实践方面带来微薄的价值。

我们可以把微服务架构想象成海面上的一座冰山，看得见的部分是开发，看不见的部分是运维，一个好的微服务架构需要同时关注开发和运维两个方面。本系列书分上下两册，上册偏开发，下册偏运维。

您适合看这本书吗?

如果您还没听说过微服务，或者您听说了但不知道它究竟是什么，或者您正在尝试微服务的实践，那么这本书就非常适合您。不管您是一名开发人员还是一名运维人员，如果您向往成为一名优秀的微服务架构师，那么这本书更加值得您反复阅读和实践。

本书是如何组织的?

第 1 章：微服务架构设计概述。

从为什么需要微服务架构开始讲起，接着描述微服务架构是什么，以及微服务架构有哪些特点，最后以如何搭建微服务架构来结束本章。本章是全书的概述，从一个宏观的视角来讲解微服务，为后续章节搭建了一个骨架。

第 2 章：微服务开发框架。

本章我们将使用流行的 Spring Boot 来搭建微服务开发框架，对 Spring Boot 是什么，以及如何使用 Spring Boot 都做了描述，此外还对 Spring Boot 的重要产品级特性做了相关介绍。通过学习本章，大家可掌握 Spring Boot 的基本使用方法，并具备开发微服务接口的技能。

第 3 章：微服务网关。

本章我们将学习 Node.js 技术，描述 Node.js 是什么，以及如何使用 Node.js，此外还对 Node.js 的重要高级特性做了补充。最后我们将使用 Node.js 搭建一个微服务网关基础框架，后续章节会对此框架进行扩展。

第 4 章：微服务注册与发现。

本章我们将学习 ZooKeeper 框架，从认识 ZooKeeper 开始，到如何使用 ZooKeeper。最后我们将使用 ZooKeeper 实现一个简单的服务注册组件，并结合第 3 章中介绍的微服务网关框架，使用 Node.js 实现一个服务发现组件。

第 5 章：微服务封装。

本章我们将学习 Docker 技术，从了解 Docker 是什么开始，到如何使用 Docker，并通过手工和 Dockerfile 的方式构建 Docker 镜像，此外还会介绍 Docker Registry 的使用方法，最后将以 Spring Boot 与 Docker 做一个整合来结束本章。通过学习本章，大家可熟练使用 Docker，为后续自动化运维提供基础。