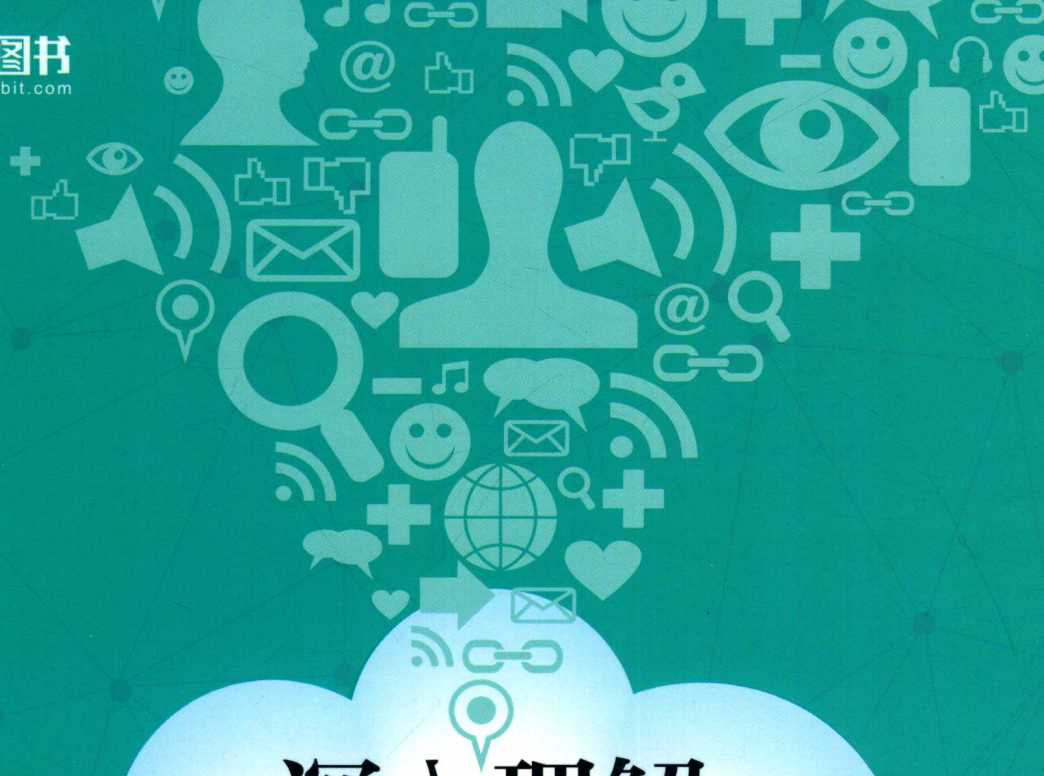
异步图书
www.epubit.com



深入理解 Spring Cloud 与微服务构建

（第2版）

方志朋◎著

基于 Greenwich 版本，全面讲解 Spring Cloud 原生组件。
深入原理，辅以图解，生动串联整个 Spring Cloud 生态。
总结提升，利用综合案例展现构建微服务系统的全过程。
附带全书源码供读者下载，方便学习和使用。

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

深入理解 Spring Cloud 与微服务构建

（第2版）

方志朋◎著

人民邮电出版社

北京

图书在版编目（CIP）数据

深入理解Spring Cloud与微服务构建 / 方志朋著
. -- 2版. -- 北京 : 人民邮电出版社, 2019.9
ISBN 978-7-115-51488-2

Ⅰ. ①深… Ⅱ. ①方… 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(2019)第117816号

内 容 提 要

本书共分为 18 章，全面涵盖了通过 Spring Cloud 构建微服务的相关知识点。第 1、2 章详细介绍了微服务架构和 Spring Cloud。第 3、4 章讲解了通过 Spring Cloud 构建微服务的准备工作。第 5～14 章以案例为切入点，讲解了通过 Spring Cloud 构建微服务的基础组件，包括 Eureka、Ribbon、Feign、Hystrix、Zuul、Gateway、Consul、Config、Sleuth、Admin 等组件。第 15～17 章讲述了使用 Spring Cloud OAuth2 来保护微服务系统的相关知识。第 18 章用一个综合案例全面讲解了如何使用 Spring Cloud 构建微服务，可用于实际开发中。

本书既适合 Spring Cloud 初学者使用，也适合正在做微服务实践的架构师或将要实施微服务的团队参考，同时也可作为高等院校计算机相关专业的师生用书和培训学校的教材。

◆ 著　　方志朋
责任编辑　张　爽
责任印制　焦志炜

◆ 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
涿州市京南印刷厂印刷

◆ 开本：800×1000　1/16
印张：19.75
字数：431 千字　　2019 年 9 月第 2 版
印数：13 701 – 17 200 册　　2019 年 9 月河北第 1 次印刷

定价：79.00 元

读者服务热线：(010)81055410　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

序

在业务驱动的时代，企业大多急于解决效率、成本和质量的问题，开发团队的不稳定，导致业务设计、技术架构和代码脉络的不连续性。企业领导对技术储备的不重视，导致企业内部重业务、轻技术，很多研发人员缺乏设计意识。各个行业对软件系统的诉求越来越多，导致软件系统规模不断增大、复杂度不断增加，最终的结果是系统难以扩展、维护、重构、跟踪、评估工期。软件在发展过程中会面临遗留系统的种种问题，但“外科手术”治不好遗留系统复杂的病症，这是架构需要演进的根源之一。

复杂系统的各个组成部分自然趋向于混乱无序，越混乱，越稳定。如果要打破混乱、建立秩序，根据熵增定律，就要赋予系统一定的力量或者能量。混乱是事物最自然的状态，秩序却不是，但是人们希望有秩序，所以要付出代价建立制度，并维护秩序。

在这个背景下，软件系统的架构一步步演进和发展，经历了单体架构、分布式应用架构、微服务架构、服务网格架构、Serverless 架构……其中，单体架构经历了简单单体时期（例如经典的 JSP）、MVC 分层时期（各种 MVC 框架受到追捧）、前后端分离时期。从整体上看，这一次次的演进是软件垂直和水平方向上的拆分，屏蔽了底层与重新定位。在演进过程中，软件开发人员的关注点越来越远离底层的部分，更多地关注上层简单的架构，技术团队的职能划分也越来越清晰。这使得软件的研发过程更高效，质量更可控，工期也更易评估。从这个角度来看，作为技术人员，我们都需要用历史的眼光去看技术的发展，拥抱变化。

微服务架构不是银弹，就像最近不断被提起的中台不能解决所有企业的问题。我们有时会存在某种认知的误区，对成功案例方法本身的关注甚于对问题本身的关注。企业的健康发展在于发现、分析和解决自身的问题，而不是盲目模仿成功的企业。

基于以上的警示，是否要在团队内部进行微服务实践？微服务落地该如何选型呢？作为一个技术人员，选择需要慎重。希望 Spring Cloud 能够帮你解决当前的问题，而不只是做一个简单的门户或者网站，因为没有必要为一两个简单的管理系统来维护 Spring Cloud 的一整套组件。在为新系统和遗留系统选择使用 Spring Cloud 前，我们需要分析当前面临的问题。

我见过一些团队对选型的背景、原因和目的并不是十分了解，就在新项目中直接使用 Spring Cloud，只是为了不让自己与当前服务化阶段受追捧的微服务架构潮流脱节。很少有人在选型前对自己软件系统的规模、所在企业的底层资源自动化的程度、技术团队的组织形式、当前业务所面临的问题、团队的技术栈，以及引入 Spring Cloud 的成本等进行关注和研究。

我也见过一些团队在遗留系统复杂度高、业务耦合严重、模块划分不清晰，甚至模块拆分在垂直和水平层面都不彻底时，整个系统臃肿不堪，团队迫不得已用 Spring Cloud 将遗留系统强势进行拆分，在将复杂系统迁移到 Spring Cloud 的过程中遇到了很多问题。

《深入理解 Spring Cloud 与微服务构建（第 2 版）》深入浅出地讲解了 Spring Cloud 生态组件，包括服务注册发现组件 Eureka、配置中心 Spring Cloud Config、容错组件 Hystrix、接入赋能组件 Zuul、路由负载均衡组件（高可用性和稳定性）Ribbon 等，使读者能够熟悉 Spring Cloud 各组件的作用和使用方法。

此外，本书还对一些技术点举一反三，例如在讲解 RestTemplate 作为网络请求时，提到其他 Spring Template，包括 JdbcTemplate 和 JmsTemplate 等。本书实用性强，代码示例全面，能够使读者在技术学习方法与认知上有一定的转变和提升。

我相信无论是正在学习 Spring Cloud 的朋友，还是正在推进或选型 Spring Cloud 落地的团队，都能从本书中有所收获。

中国的近现代史是一段“师夷长技以制夷”的历史，在如今信息技术和互联网技术快速发展的时代，我们不能只停留于学习和模仿，更要发现、耕耘、创新。在此与大家共勉！

徐凌云 高级架构师

2019 年夏 于湖北

前　言

作为 Java 语言的落地微服务框架，Spring Cloud 已经在各大企业普遍应用，各大云厂商也支持 Spring Cloud 微服务框架的云产品。可以说，Spring Cloud 微服务框架已经应用到了各大行业之中，并成为 Java 开发者的必备技能之一，熟练掌握 Spring Cloud 是面试者的加分项。

Spring Cloud 由 Spring Cloud 社区维护，并且在 Pivatol 和 Netflix 两大公司的推动下飞速发展。随着 Eureka 的闭源，虽然 Netflix OSS 等组件进入维护期，不再提供新功能，但 Spring Cloud 微服务框架并没有受到显著影响，而是被越来越多的企业和开发者所接受。阿里巴巴推出的 Nacos 和 Sentinel 等组件已经加入 Spring Cloud 孵化器项目，未来极有可能替代 Netflix OSS，因此 Spring Cloud 是一个极具生命力的微服务框架。

在本书第 1 版出版后不到一年的时间，我便开始着手准备第 2 版，在短时间内更新第 1 版的原因有以下几点。

第一，为了快速跟进 Spring Cloud 新版本。本书使用的 Spring Cloud 版本为 Greenwich，Spring Boot 版本为 2.1.0。众所周知，Spring Cloud 最大的特色是开源，它是由众多优秀的开源组件封装、集成的。比如，它集成了 Netflix OSS 组件和 Nacos 组件等；Spring Cloud 社区十分活跃，吸引了众多优秀的开发者加入其项目开发中，因此 Spring Cloud 的版本迭代非常快。本书第 1 版使用的 Spring Cloud 版本为 Dalston，Spring Boot 版本为 1.5.3。在短短一年多的时间内，Spring Cloud 已经迭代了 Edgware、Finchley 和 Greenwich 三大版本，对应的 Spring Boot 版本分别为 1.5.x、2.0.x 和 2.1.x。2.0.x 版本的 Spring Boot 更新幅度较大，支持 Webflux 响应式编程和 Http2 等新特性。Spring Cloud 基于 Spring Boot，所以 Spring Cloud 的 Finchley 版本是一个变动较大的版本。本书使用的 Greenwich 版本是基于 Finchley 的一个小迭代。

第二，使用 Spring Cloud 作为微服务框架的大多数企业都使用 Consul 作为服务注册组件。Consul 为微服务的元数据提供了强一致性的保障，支持多个数据中心，这是它相对于 Eureka 1.0 的优点。此外，Eureka 的闭源使得作为注册中心的 Consul 越来越重要，所以本书第 2 版详细介绍了 Consul。

第三，Spring Cloud 的第一代网关 Zuul 是一个阻塞式网关，在性能方面有诸多不足。Spring Cloud 在 Finchley 版本中推出了新一代网关 Spring Cloud Gateway。Spring Cloud Gateway 是非阻塞式网关，与 Zuul 相比，性能有较大提升。

第四，很多组件在新版本中的变动较大，如 Sleuth、Admin 和 Security 等。部分读者反馈，第 1 版中的源码在更新版本后出现了无法运行的现象，因此内容更新和升级迫在眉睫。

本书内容

本书共分为 18 章，各章主要内容如下。

第 1 章介绍了什么是微服务、为什么需要微服务、微服务的优缺点和面临的挑战，并且将单体架构的系统和微服务架构的系统进行了比较。

第 2 章主要介绍微服务应该具备的功能以及 Spring Cloud 的基本组件，最后介绍了 Spring Cloud 与 Dubbo、Kubernetes 之间的差异。

第 3、4 章介绍了构建微服务的准备工作：开发环境的构建和 Spring Boot 的使用。其中，第 3 章介绍了开发环境的构建，包括 JDK 的安装、IDEA 和 Maven 的使用等；第 4 章介绍了 Spring Boot 的基本使用方法，包括 Spring Boot 的特点、用 IDEA 创建一个 Spring Boot 项目、Spring Boot 配置文件详情、Spring Boot 的 Actuator 模块，以及 Spring Boot 集成 JPA、Redis 和 Swagger2 等。

第 5～9 章介绍了 Spring Cloud 框架的基础模块——Spring Cloud Netflix 模块，涵盖了 Spring Cloud 构建微服务的基础组件。诸如 Eureka、Ribbon、Feign、Hystrix 和 Zuul 等组件为微服务系统提供了基本的服务治理能力。这些章以案例为切入点，由浅入深介绍这些组件，并从源码的角度分析这些组件的工作原理。

第 10 章介绍了 Spring Cloud 的第二代网关 Gateway。Gateway 在性能上比 Zuul 要优异很多，是 Spring Cloud 的新一代网关。

第 11 章介绍了服务注册中心 Consul，详细讲解了如何使用 Consul 进行服务注册和发现，以及如何使用 Consul 作为分布式配置中心。

第 12 章介绍了分布式配置中心 Spring Cloud Config，详细讲解了 Config Server 如何从本地仓库和远程 Git 仓库读取配置文件，以及如何构建高可用的分布式配置中心和使用消息总线刷新配置文件。

第 13 章介绍了链路追踪组件 Spring Cloud Sleuth，包括微服务系统为什么需要链路追踪组件，并以案例的形式详细介绍了如何在 Spring Cloud 微服务系统中使用链路追踪，以及如何传输、存储和展示链路数据。

第 14 章以案例的形式介绍了 Spring Boot Admin，包括 Spring Boot Admin 在微服务系统中的应用、在 Spring Boot Admin 中集成安全组件。

第 15～17 章介绍了 Spring Cloud 微服务系统的安全验证模块，包括 Spring Boot Security 组件和 Spring Cloud OAuth2 模块。第 15 章详细介绍了如何在 Spring Boot 应用中使用 Spring Boot Security；第 16 章介绍了如何在 Spring Cloud 微服务系统中使用 Spring Cloud OAuth2 来保障微服务系统的安全；第 17 章介绍了如何在 Spring Cloud 微服务系统中使用 Spring Cloud OAuth2 和 JWT 来保护微服务的系统安全。

第 18 章以一个综合案例介绍了使用 Spring Cloud 构建微服务系统的全过程，该案例是对全书内容的总结和提炼。

本书特色

❑ 案例丰富，通俗易懂

本书的写作目标之一就是将复杂问题简单化，从而让读者轻松地学习到技术。本书用丰富的案例循序渐进地讲解了如何使用 Spring Cloud 构建微服务。

❑ 深入浅出，透析本质

以案例为切入点，基于代码对 Spring Cloud 关键组件进行解读，深入讲解原理，并在案例中使用大量图片（包括展示图和架构图等），帮助读者深入理解。最后以一个综合案例完整讲解如何使用 Spring Cloud 构建微服务，达到学以致用的目的。

❑ 网络资源，技术支持

本书中所有的源码按章节划分，每章都有独立的源码，便于读者使用和理解。读者可以到异步社区或扫描下方二维码到我的微信公众号（walkingstory）中下载源码。打开源码即可轻松运行。为了快速学习和掌握 Spring Cloud，建议对照源码阅读本书。

致谢

感谢我的家人在我写作本书过程中给予我的支持和鼓励。
感谢我的大学导师王为民教授对我的指导和培养。
感谢编辑张爽在本书写作和出版过程中所做的工作。
感谢各位读者和朋友的厚爱！

方志朋
2019 年夏

资源与支持

本书由异步社区出品，社区（https://www.epubit.com/）为您提供相关资源和后续服务。

配套资源

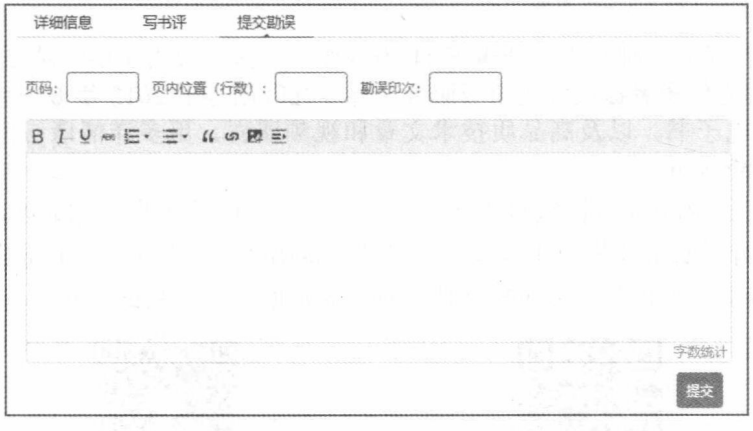
本书提供源代码文件，请在异步社区本书页面中点击 配套资源，跳转到下载界面，按提示进行操作即可。注意：为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

如果您是教师，希望获得教学配套资源，请在社区本书页面中直接联系本书的责任编辑。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，点击“提交勘误”，输入勘误信息，点击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。



扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。

与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，并请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 www.epubit.com/selfpublish/submission 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“**异步社区**”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 https://www.epubit.com。

“**异步图书**”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术等。

异步社区

微信服务号

目　录