异步图书
www.epubit.com.cn

深入理解 Spring Cloud 与微服务构建

方志朋◎著

非外借

由浅入深，全面讲解 Spring Cloud 基础组件。

一站式了解用 Spring Cloud 构建微服务，**实战案例，快速上手**。

详解 Spring Security OAuth2，为微服务系统的**安全保驾护航**。

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

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
. -- 北京 : 人民邮电出版社, 2018.3 (2018.4重印)
ISBN 978-7-115-47522-0

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(2017)第315454号

内 容 提 要

本书共分 16 章，全面涵盖了 Spring Cloud 构建微服务相关的知识点。第 1、2 章详细介绍了微服务架构和 Spring Cloud。第 3、4 章讲解了用 Spring Cloud 构建微服务的准备工作。第 5～12 章以案例为切入点，讲解了 Spring Cloud 构建微服务的基础组件，包括 Eureka、Ribbon、Feign、Hystrix、Zuul、Config、Sleuth、Admint 等组件。第 13～15 章讲述了使用 Spring Cloud OAuth2 来保护微服务系统的相关知识。第 16 章用一个综合案例，全面讲解了如何使用 Spring Cloud 构建微服务，可以作为实际开发的样例工程。

本书既适合 Spring Cloud 初学者入门使用，又适合正在做微服务实践的架构师或打算实施微服务的团队作为参考用书，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　涛
执行编辑　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
三河市君旺印务有限公司印刷

◆ 开本：800×1000　1/16
印张：17.5
字数：412 千字　　2018 年 3 月第 1 版
印数：3501-5500册　　2018 年 4 月河北第 2 次印刷

定价：69.00 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　一

一直以来，系统的架构设计是 IT 领域经久不衰的话题，也是构建每一个系统最核心且重要的部分之一。它决定了系统能否满足业务、技术、组织、灵活、可扩展性等多种要求，同时肩负起了解放程序员生产力的作用。

2016 年底，由于业务的不断发展，我所在公司维护的项目也越来越“臃肿”。随着无数个版本的迭代，以及开发人员的不断增加，开发效率越来越低，每次投产的人力成本和时间成本都逐渐增加，我们一直在思索如何能破局。评估了各种方案后，最终微服务进入了我们的视野。

谈到微服务，大家众说纷纭，但却很难有一个清晰的概念来描述。微服务不是“银弹”，我理解的微服务是一种文化，而我们要做的就是将微服务的理念运用到实际开发中。经过一系列的技术选型，最终 Spring Cloud 凭借其成熟的组件、完善的一站式解决方案，最终成为了我们落地微服务的选择。

此时的 Spring Cloud 相关资料在国内还是凤毛麟角，没有完整的中文书籍和教程可以参考，只有官方的英文文档以及网上零零散散的教程可以阅读。就是在这种情况下，本书的作者方志朋在公司技术选型以及后续的微服务落地过程中，逐渐有了自己的积累和理解，同时在博客中连载了“史上最简单的 Spring Cloud 教程”。此教程一出，就受到广大程序员的欢迎，因此最终整理为此书。

纵览全书，文字清晰明了，通过理论结合实践的方式介绍了 Spring Cloud 的每一个组件的实践，并解读了部分源代码。图文并茂，语言朴实，不愧为“简单”之名。本书融合了作者实施微服务的一线经验和心得，具体指导了 Spring Cloud 在落地方面的实践，非常值得参考。

深圳小安时代互联网金融服务有限公司，研发中心总经理
于际予

序　二

2015 年，Spring Cloud 项目最初在 GitHub 上出现时，我就一直在关注其项目发展。到 2016 年，国内关注 Spring Cloud 的人越来越多，但是相应学习交流的平台和材料却比较分散，不利于学习交流，机缘巧合之下 Spring Cloud 中国社区诞生并发展至今。

Spring Cloud 中国社区是国内首个 Spring Cloud 构建微服务架构的交流社区，也是致力于为 Spring Boot 和 Spring Cloud 技术人员提供分享和交流的平台，推动 Spring Cloud 在中国的普及和应用。因为技术交流，我和志朋相识。志朋所写的博客“史上最简单的 Spring Cloud 教程”系列，浏览量已经达到数百万，并获得了广大读者的一致好评，为 Spring Cloud 学习者和开发者答疑解惑。

《深入理解 Spring Cloud 与微服务构建》一书总结并延伸了博客中的精华内容，结合社区中的常见问题进行了方案梳理，抛砖引玉，通俗易懂，涵盖了 Spring Cloud 中常用的组件和项目案例实战。希望本书能够帮助开发者快速使用 Spring Cloud 对企业 IT 架构微服务进行开发和改造！

Spring Cloud 中国社区创始人
许进

序　三

好像一夜之间，全世界都在讨论微服务，讨论如何使用 Spring Cloud 来构建自己的微服务体系。但是一切都像是“徒手抓泥鳅”，无处下手，很是难受，方兄在前期也遇到过这种情况。这在一般人看来就是“算了，还是换一种技术框架吧”的结果，但是方兄反其道而行之，不仅把这些东西一一梳理起来，编写了一系列的教程，而且作为 Spring Cloud 中国社区联合发起人，他还向 Spring Cloud 开源社区贡献了大量的代码。

从 Spring Cloud 的造诣上来说，方兄很是值得我们学习。此书循循善诱，从字里行间就能感受到方兄在理论和实践方面“两手抓、两手强”，造诣之深，着实难得。更难得的是方兄的分享精神，他毫无保留地把自己的项目、思考、总结和方法集结成书，与广大读者共同成长，可谓无私。

好了，闲言少叙，赶紧翻开这本书，你会喜欢上它的！

前平安普惠数据应用架构师，微信公众号“一名叫大蕉的程序员”出品人
杨钊

序　四

本书对于想了解 Spring Cloud 但又无从下手的读者来说简直就是福音，看完前几章就可以对 Spring Cloud 有大致的了解。

本书对于那些已经了解了 Spring Cloud 的专业人士来说也是福音，因为这里可能有你没见过的技术点。

作为见证了本书从无到有的旁观者来说，本书不仅解决了我在技术上的疑惑，还让我明白了该如何面对挫折。

大概在开始写作一个多月的时间后，志朋感到很迷茫，害怕技术点写得不够深入，对购书的读者不负责任。大概有近一周的时间，他惶惶不可终日。所幸的是，他的博客中有很多的读者和网友发信息鼓励他。其中一个网友说“只要投入了最大的热情，专注于技术就会感到很满足”，这句话让志朋重新燃起了斗志，有了坚持写下去的动力和勇气。

为了保证本书的质量，将一手资料及时准确地传送给读者，他翻阅了大量的中英文资料，反复论证内容的可行性，举一反三。为了让读者能看的懂，并且有看下去的欲望，他反复打磨每一句话，尽可能使其简单化，通俗易懂。

世间万物，都有其精神，本书之精神乃艰苦奋斗、专注专业的工匠精神！

深圳捷顺科技架构师

鄂超

序　五

当今世界，互联网飞速发展，并融入生活的方方面面，人们对互联网产品提出了更严格的要求，重体验、重响应速度、关联性强、业务场景越来越复杂等是互联网产品的新特点。这对软件架构师提出了新的要求，如何使得系统架构能够轻松持续改进、快速部署、业务之间低耦合，成为软件架构师的新的思考方向。

微服务作为系统解决这些问题的一种思路应运而生，方兴未艾。本书作者对微服务的见解独到，实战经验丰富，为我们介绍了什么是微服务，微服务与传统软件架构的对比，为什么要使用微服务，微服务的设计原则和具体特点，微服务应该具备的功能，微服务的应用场景，需要使用的组件，以及如何综合运用这些组件构建整个微服务框架。

本书主要针对 Java 开发者构建微服务框架，作者比较青睐于 Java 语言的 Spring Cloud 微服务框架，究其原因是 Spring Cloud 有快速开发、持续交付和易于部署等特点，且开源社区比较活跃，同时有国际巨头公司的推动。本书在 Spring Cloud 框架范围内，介绍了服务注册和发现的 Eureka 组件、负载均衡 Ribbon 组件、熔断器 Hystrix 组件、路由网关 Zuul 组件、Spring Cloud 配置中心、服务链路追踪等内容，同时也与其他微服务框架做了对比，拓展了微服务知识的深度和广度。本书结构清晰，行文优美，每一个例子都经过作者斟酌再三，力求使用最简单的例子，将复杂的逻辑原理阐述清楚，让读者印象深刻。

本书是一本用于微服务入门和提高的好书，相信你读完本书后定能感受到作者的严谨与智慧，同时对微服务有一定的理解，并能够灵活运用。

平安集团
胡益雄

前　言

近年来随着互联网的飞速发展，各行各业都在拥抱互联网。互联网给人类生活带来了翻天覆地的变化，人们在享受互联网给生活带来便捷的同时，业务需求的发展也对互联网技术提出了更高的要求，传统的单体架构对越来越复杂的业务需求显得力不从心。此外，随着大数据、云计算和人工智能的飞速发展，软件的架构显得越来越重要。近几年来，“微服务”这一名词在各大网站、论坛、演讲中出现的频率足以让人们感觉到它对软件架构带来的影响。目前，各大公司都在纷纷采用微服务架构。

Spring Cloud 作为 Java 语言的微服务落地框架，在 Spring 开源社区和 Pivatol、Netflix 两大公司的推动下飞速发展，得到了众多开发者的认可，Spring Cloud 在未来很可能成为微服务框架的领导者和规范。和众多 Spring Cloud 开发者一样，我在工作和学习中对 Spring Cloud 系列框架、组件非常痴迷。我利用业余时间在 CSDN 博客上发表了一系列关于 Spring Cloud 的文章，受到广大开发人员的欢迎，在短短半年的时间里，Spring Cloud 系列文章的阅读量就超过 200 万。另外，作为 Spring Cloud 中国社区的联合发起人，我持续为社区贡献文章，得到了社区朋友们的认可。因为 Spring Cloud 是一个新技术，很多人对此还不是很了解，所以希望我的文章可以作为大家学习的资料，也欢迎读者关注我的博客 http://blog.csdn.net/forezp。

沿袭了博客的写作风格，我花费了半年的时间和大量的心血来完成本书。从 Spring Cloud 的基础组件开始讲解，对关键组件做了源码分析，力求帮助读者深入理解原理。同时也重点讲解了如何在 Spring Cloud 微服务系统中进行身份认证和权限安全的验证。在本书的最后，以一个综合案例来全面讲解 Spring Cloud 是如何构建微服务的，这个案例是我在学习和工作过程中使用 Spring Cloud 的提炼和总结，具有非常高的参考价值。

本书内容

本书共分为 16 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 之间的比较及优缺点。

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 的特点和优点、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、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。例如 Eureka、Ribbon、Feign、Hystrix 和 Zuul 等，这些组件为微服务系统提供了基本的服务治理能力。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 11 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 12 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 中集成安全登录组件。

第 13～15 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 13 章详细介绍了如何在 Spring Boot 应用中使用 Spring Boot Security；第 14 章介绍了如何在 Spring Cloud 微服务系统中使用 Spring Cloud OAuth2 来保护微服务的系统安全；第 15 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 16 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

1. 案例丰富，通俗易懂

我的写作目标之一就是将复杂的事情简单化，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。

2. 深入浅出，透析本质

以案例为切入点，对 Spring Cloud 关键组件进行源码解读，深入讲解原理，并在案例中使用大量的图解，包括展示图、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了如何使用 Spring Cloud 构建微服务，达到学以致用的目的。

3. 网络资源，技术支持

本书中所有的源码按章节划分，每一章节都有独立的源码，方便读者使用和理解。读者可以扫描下方二维码，到我的微信公众号（微信号 walkingstory）中下载源码。源码打开即用，

可以轻松运行。读者可以一边看书，一边看源码，易于快速学习和理解。

阅读建议

本书的读者对象既可以是 Spring Cloud 的初学者，也可以是经验丰富的架构师。建议循序渐进、从前往后对照源码通读本书。本书采用的 Spring Cloud 版本为 Dalston，Spring Boot 版本为 1.5.3。

建议和反馈

由于作者能力有限，虽然对书稿做了多次认真的检查和修改，但错漏之处在所难免，敬请读者批评指正。读者可以到我的微信公众号或者博客中留言反馈，也可以将意见或建议发至本书编辑的邮箱 zhangshuang@ptpress.com.cn，我会及时给出解答。

致谢

感谢我的家人在我写作本书过程中所给予的支持和鼓励。感谢大学时的导师王为民教授在精神上对我的鼓舞，使我如沐春风，终身受益。感谢我的同事提出的宝贵意见，和你们一起工作非常荣幸，也非常开心。感谢所有的技术朋友给我的帮助和建议，包括 Spring Cloud 中国社区的小伙伴们，以及各大社区的技术朋友们。感谢编辑张爽在本书写作和出版过程中所做的工作。感谢这么多良师益友……

方志朋

2017 年秋

目　录

第1章 微服务简介

随着互联网技术的飞速发展，目前全球超过一半的人口在使用互联网，人们的生活随着互联网的发展，发生了翻天覆地的变化。各行各业都在应用互联网，国家政策也在大力支持互联网的发展。随着越来越多的用户参与，业务场景越来越复杂，传统的单体架构已经很难满足互联网技术的发展要求。这主要体现在两方面，一是随着业务复杂度的提高，代码的可维护性、扩展性和可读性在降低；二是维护系统的成本、修改系统的成本在提高。所以，改变单体应用架构已经势在必行。另外，随着云计算、大数据、人工智能的飞速发展，对系统架构也提出了越来越高的要求。

微服务，是著名的 OO（面向对象，Object Oriented）专家 Martin Fowler 提出来的，它是用来描述将软件应用程序设计为独立部署的服务的一种特殊方式。最近两年，微服务在各大技术会议、文章、书籍上出现的频率已经让人们意识到它对于软件领域所带来的影响力。微服务架构的系统是一个分布式系统，按业务领域划分为独立的服务单元，有自动化运维、容错、快速演进的特点，它能够解决传统单体架构系统的痛点，同时也能满足越来越复杂的业务需求。

1.1 单体架构及其存在的不足

1.1.1 单体架构简介

在软件设计中，经常提及和使用经典的 3 层模型，即表示层、业务逻辑层和数据访问层。

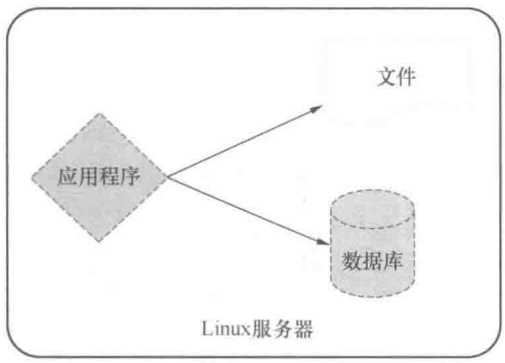
- 表示层：用于直接和用户交互，也称为交互层，通常是网页、UI 等。
- 业务逻辑层：即业务逻辑处理层，例如用户输入的信息要经过业务逻辑层的处理后，才能展现给用户。
- 数据访问层：用于操作数据库，用户在表示层会产生大量的数据，通过数据访问层对数据库进行读写操作。

虽然在软件设计中划分了经典的 3 层模型，但是对业务场景没有划分。一个典型的单体应用就是将所有的业务场景的表示层、业务逻辑层和数据访问层放在一个工程中，最终经过编译、

打包，部署在一台服务器上。例如典型的 J2EE 工程，它是将表示层的 JSP、业务逻辑层的 Service、Controller 和数据访问层的 Dao，打成 war 包，部署在 Tomcat、Jetty 或者其他 Servlet 容器中运行。经典的单体应用如图 1-1 所示。

在一个小型应用的初始阶段，访问量较小，应用只需要一台服务器就能够部署所有的资源，例如将应用程序、数据库、文件资源等部署在同一台服务器上。最典型的就是 LAMP 系统，即服务器采用 Linux 系统，开发应用程序的语言为 PHP，部署在 Apache 服务器上，采用 MySQL 数据库。在应用程序的初始阶段，采用这种架构的性价比是非常高的，开发速度快，开发成本低，只需要一台廉价的服务器。此时的服务器架构如图 1-2 所示。

▲图 1-1　经典的单体应用架构



▲图 1-2　LAMP 应用服务器示意图

1.1.2　单体架构存在的不足

在应用的初始阶段，单体架构无论是在开发速度、运维难度上，还是服务器的成本上都有着显著的优势。在一个产品的前景不明确的初始阶段，用单体架构是非常明智的选择。随着应用业务的发展和业务复杂度的提高，这种架构明显存在很多的不足，主要体现在以下 3 个方面。

- 业务越来越复杂，单体应用的代码量越来越大，代码的可读性、可维护性和可扩展性下降，新人接手代码所需的时间成倍增加，业务扩展带来的代价越来越大。
- 随着用户越来越多，程序承受的并发越来越高，单体应用的并发能力有限。
- 测试的难度越来越大，单体应用的业务都在同一个程序中，随着业务的扩张、复杂度的增加，单体应用修改业务或者增加业务或许会给其他业务带来一定的影响，导致测试难度增加。

1.1.3　单体架构使用服务器集群及存在的不足

随着业务的发展，大多数公司会将单体应用进行集群部署，并增加负载均衡服务器（例如 Nginx 等）。另外，还需要增加集群部署的缓存服务器和文件服务器，并将数据库读写分离，以应对用户量的增加而带来的高并发访问量。此时的系统架构如图 1-3 所示。

用负载均衡服务器分发高并发的网络请求，用户的访问被分派到不同的应用服务器，应用服务器的负载不再成为瓶颈，用户量增加时，添加应用服务器即可。通过添加缓存服务器来缓