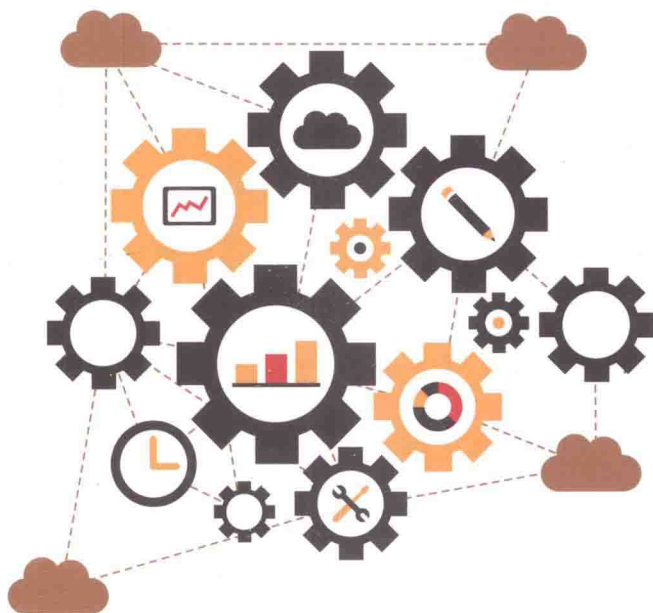


Spring Cloud与Docker

微服务架构实战

周立 著



中国工信出版集团

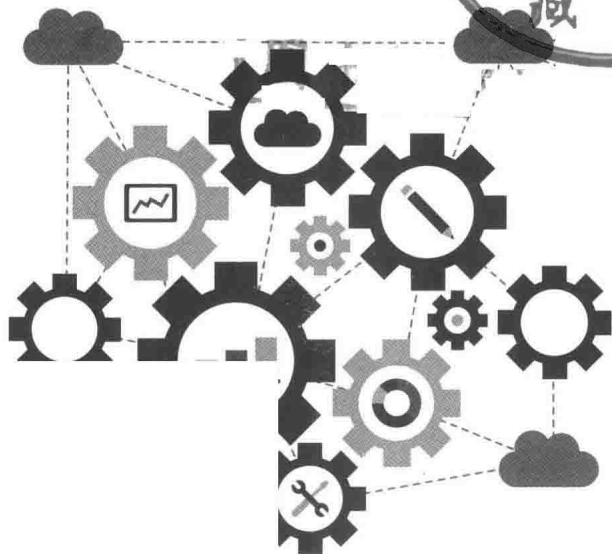


电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Spring Cloud与Docker

微服务架构实战

周立 著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内容简介

作为一部帮助大家实现微服务架构落地的作品，本书基于 Spring Cloud Camden SR4 Docker 1.13.0，覆盖了微服务理论、微服务开发框架（Spring Cloud）以及运行平台（Docker）三大主题。全书可分为三部分，第 1 章对微服务架构进行了系统的介绍；第 2~11 章使用 Spring Cloud 开发框架编写了一个“电影售票系统”；第 12~14 章则讲解了如何将微服务应用运行在 Docker 之上。全书 Demo 驱动学习，以连贯的场景、具体的代码示例来引导读者学习相关知识，最终使用特定的技术栈实现微服务架构的落地。

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

Spring Cloud 与 Docker 微服务架构实战 / 周立著. —北京：电子工业出版社，2017.5

ISBN 978-7-121-31271-7

I. ①S…II. ①周…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 070057 号

策划编辑：张春雨

责任编辑：徐津平

印刷：北京天宇星印刷厂

装订：北京天宇星印刷厂

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：17

字数：342.18 千字

版次：2017 年 5 月第 1 版

印次：2017 年 7 月第 4 次印刷

定价：69.00 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 zltz@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 1

2016年国庆假期之后，我所在的公司因为业务需要，想搭建一个API网关来综合治理已有业务调用服务（我司之前采用的是当当的Dubbo扩展框架Dubbox）。前期，我和同事们在技术选型环节，讨论了诸多目前比较红火的技术框架和工具。最后达成一致，采用微服务来重构和调整原先这些Dubbox服务，并决定使用Spring Cloud（以下简称sc）来实现API网关，争取在2017年能顺利平滑地从Dubbox过度到sc。而具体的API网关demo研发工作就落实到了我这里。

在开始研发工作之前，我参阅了包括官网在内很多sc研发资料，也去全球最大的同性技术交友网站GitHub上找了很多代码来仔细研读。但感觉老外的这些Guide（指南）总是讲得不是很通透。也许是有些概念他们觉得太基础了，就直接略过不表。因此我也感到很迷茫，老是问自己，到底应该如何去实现这个API网关，完成公司指派给我的研发任务呢？

幸好，某一天我看到《Spring Cloud与Docker实战微服务》这本开源书。根据书中例子，我几乎没有费什么大工夫就搭建了一个API网关的demo。甚至其中某些讲解点，看了之后能让我一下子恍然大悟，回头再看那些老外的Guide，我终于明白了其中的“奥秘”。我真的非常感激这位开源书作者，他深入浅出地将sc所涉及的各种知识点和工具的使用做了完整和详细的叙述。从此，我也记住了此书作者的网名itmuch。

几天后，我将demo做了细化和扩展，并在oschina的码云网站上开源分享了出去（具体网址见<http://git.oschina.net/darkranger/spring-cloud-books>）。而无巧不巧，itmuch居然在我项目下的评论区留言了。经过加QQ、加微信等一系列同性技术交友过程（你们懂的）取得了联系，也终于知道了这位itmuch的真名，那就是此书作者周立同学。在闲聊过程中，他透露了自己正在以那本《Spring Cloud与Docker实战微服务》开源书为基础，继续扩展和具体深入sc这套微服务开发体系所包含的所有技术点，准备出版成册，让更多的朋友和企业能学习和借鉴sc这套东西开发符合自己业务场景的微服务框架。并邀请我为新书做校对和修正工作。正巧，我也越来越喜欢钻研sc，也希望对自己碰到的一些问题向他指教，所以就答应了下来。这其中的过程真的一言难叙，总算最后我也不辱使命地完成了这本书的校对工作。

我可以很负责地说，本书是周立同学本人在工作和学习 sc 后总结出的精华，书中每段代码、每个字都是他自己写的，绝无任何抄袭之举。完全可以说是一个努力勤奋、能独立思考、认真做事的同学的良心之作。这样的“业界良心”在如今这个充满浮躁的社会中已不多见。希望有更多的读者能珍惜此书，感谢周立同学给我们的帮助。除此之外，我也希望读者能在阅读完此书后，可以自己写点代码，亲身去实践一把，感受 sc 的精妙之处。

最后，在仓促完成本文之前，值此新春佳节之际，我也祝大家新年快乐，家庭安康，财源滚滚，爱情事业双丰收。

2017 年 1 月 25 日

农历丙申年腊月二十八

吴峻申 青客机器人有限公司架构师

序 2

2013年，我在 EMC 听了一个关于 Docker 与测试的分享，才第一次近距离认识 Docker。在 2014 年底时，在项目上开始接触 Docker。2015 年上半年，我读了两本书：*The Phoenix Project* 和 *Migrating to Cloud-Native Application Architectures*。这两本书让我对 DevOps、微服务和云原生架构有了初步的认识。

2015 年 9 月，我以首席架构师的身份加入麻袋理财，当时第一件事情就是借助 DaoCloud 在公司内部推行基于 Docker 的基础落地的方案。花了三个月，一个简易的方案就已经可以正常运作。但是在这个过程中，却发现和应用的契合度不是太高，需要对应用的架构做改造。

2016 年年初，当时正好有一个项目要做 2.0，之前是一个典型的单体应用（使用 Spring MVC），这次准备做微服务改造，以满足业务对技术快速迭代、横向扩展的要求。我当时对 Spring Boot 和 Spring Cloud 已经有所耳闻，但是还停留于 Demo 的地步。正好借着这个机会，准备推广 Spring Boot。之后有个全新的项目，我们完全按照微服务架构，使用 Spring Boot 和 Cloud 进行开发，并采用 CI/CD 自动化流程和容器化部署。

2016 年 10 月份时，一次偶然的计划，Spring Cloud 中国社区的许进找到了我，让我把团队在实践过程中的经验总结在社区做了分享，从而认识了本书的作者周立。当时周立正好在写一本书，他希望我能够帮他进行 review，我就欣然答应了。

看到了书的标题《Spring Cloud 与 Docker 微服务实战》，这不就是我一直在做的工作吗？于是我连夜把这本书读了一遍，感觉相见恨晚，如果一年前有这本书，那我就可以少走很多弯路了。

本书用一个例子贯穿始终，讲解了 Spring Cloud 的经典组件、微服务架构，以及与 Docker 的集成。书中提供了详细的代码，可以让读者在了解基础概念的同时，可以马上脚踏实地地撸起袖子写代码。

王天青 DaoCloud 首席架构师

2017 年 3 月

序 3

最近几年，微服务的概念非常火爆，由于它确实能解决传统单体应用所带来的种种问题（比如代码可维护性低、部署不灵活、不够稳定、不易扩展，等等），所以大家对“如何成功实施微服务架构”越来越感兴趣。在 Java 技术栈中，Spring Cloud 独树一帜，提供了一整套微服务解决方案，它基于 Spring Boot 而构建，延续了 Spring 体系一贯的“简单可依赖”，但是由于微服务本身涉及的技术或概念比较广，所以在正式“入坑”之前，最好能有一本实战性强的书籍作为参考。但是很遗憾，Spring Cloud 太新了，国内几乎没有一本完整讲解其用法的新书。

今年年初，我偶然得知周立兄在编写 Spring Cloud 相关的书籍，感到非常惊喜，在和他交流的过程中，我能感觉到他对技术的把控力以及对知识分享的热情！阅读这本书的过程是非常愉悦的，不仅仅是因为它结构之清晰，文风之流畅，更重要的是实战性极强，相信大家能在本书的指导下顺利地基于 Spring Cloud & Docker 打造出自己的微服务应用。

杜云飞 上海小虫数据技术合伙人，风控大数据负责人

序 4

在 Spring 尚未出现的“蛮荒”时代，Java 程序员们还在迷茫地创造着各种“语法糖”来试图提高生产效率。然而无论怎么努力，Java 语言仍被许多人冠以“裹脚布”的名号——毕竟你一不小心就会把它写得又臭又长。

随着 Spring 体系的出现与逐步完善，似乎有一种经历着 Java 工业革命的感觉。的确，任何事物都各有利弊，但我仍然想说，Spring 团队给 Java 程序员们带来了春天（就像它的名字一样），它神奇地把“裹脚布”变成了“丝绸”，因为它最大的特质可以用两个字来形容——优雅。相信你如果使用过 Spring Framework、Spring MVC、Spring Data、Spring Boot 或 Spring Cloud 等一系列框架，并研读过它们的源代码，就一定能够体会到“优雅”二字的含义。

尽管 Spring 家族拥有如此多而美好的大块“语法糖”，但它们过去在国内的传播似乎都不怎么顺利。我经常说，国内对新技术的广泛应用一般比国外要晚三到五年，无论后端、前端还是架构理念。这是许多因素导致的，比如信息闭塞、语言不通，甚至固步自封。我相信随着国内互联网人才越来越多，新技术应用的延时一定会越来越短。或许很多人为了旧系统的稳定而不愿升级，这可以理解，但我希望人们可以拥抱新的事物，而不是排斥。现如今微服务架构理念兴起，人们急需一个快捷、稳定、一站式的分布式微服务解决方案，Spring Cloud 正是为此而诞生。可国内熟知 Spring Cloud 的人目前仍寥寥无几，大部分人从未听说过，想要学习的人不知从何开始，对官方的英文文档也一知半解。人们需要一本能把他们领进 Spring Cloud 这扇门的“红宝书”，这便是本书的目的，也是本书作者周立的初衷——希望能够为减少国内新技术的延时而出一份力。

我与周立在 2016 年相识，在短暂的交流后我们都产生了相见恨晚的感觉。遇见志同道合的人并不容易，我们的技术理念很相似。他有着对技术的热忱、灵活的头脑，以及开源分享技术的无私精神，正是这股精神促使他做了许多分享技术的事情，并且编写了这本书（相信我，写书并不赚钱）。我十分欣赏周立身上的这些特质，因此当他跟我提到想出书并找我帮忙时，我毫不犹豫地答应了他。我相信他未来一定能够成为某一技术领域的专家，这是他的目标，他也具备这样的潜质。

本书的切入点非常好，它并不纠结于冗长的源码解读或原理解释，而是更多地注重实战，这在如今互联网爆炸式发展的时代相当重要。现在人们更倾向于使用敏捷开发尽快做出产品来进行试错，并在后续版本中快速迭代。因此本书的实战经验在软件工程层面上会给予读者很大提升，它可以让你更快地搭建分布式微服务架构，然后把精力留在编写业务逻辑上，提高你的生产力，并最终做出更好的产品——这也是 Spring 团队一直希望达到的效果。

现在，让我们随本书进入 Spring Cloud 的世界，一起感受它的优雅吧！

张英磊 云账房 CTO

2017年3月29日

前言

随着业务的发展，笔者当时所在公司的项目越来越臃肿。随着代码的堆砌，项目变得越来越复杂、开发效率越来越低、越来越难以维护，小伙伴们苦不堪言，毫无幸福感可言。我们迫切需要能够解放生产力、放飞小伙伴的“良药”，于是，微服务进入视野。然而，微服务究竟是什么，众说纷纭，没有人能说清楚什么是微服务。不仅如此，大家对微服务的态度也是泾渭分明，吹捧者、贬低者比比皆是，在笔者的 QQ 群、微信群中硝烟四起。笔者参加了不少交流会，感觉许多分享常常停留在理论阶段。一场会下来，觉得似乎懂了，却苦于没有对应的技术栈去实现这些理论。

Docker、Jenkins 等工具笔者均有涉猎，然而使用什么技术栈去实践微服务架构，在很长时间内都是笔者心中的疑问。

2015 年中，笔者偶然在 GitHub 上看到一个名为 Spring Cloud 的框架，它基于 Spring Boot，配置简单、设计优雅，并且大多组件经过了生产环境的考验。笔者花 1 个月左右的时间详细研究了 Spring Cloud 的相关组件后，体会更深。然而，技术选型必须要进行客观、多维度、全方位的分析，而不应由我个人的主观意见作为决定因素。文档丰富程度、社区活跃度、技术栈生态、开发效率、运行效率、成功案例等，都是我们选型的重要因素。经过调研，其他几点都很 OK，只缺成功案例——在当时，国内几乎没什么成功案例，甚至连中文的博客、相关资料都没有。

这让笔者陷入两难，在这一过程中，公司一边继续使用阿里巴巴开源的 Dubbo（Dubbo 虽然在国内非常流行，但毕竟有段时间没有维护了，开源生态也不是很好），一边在笔者的组织下进行一些 Spring Cloud 相关的技术分享。一方面是希望借此开拓小伙伴们的视野，另一方面也希望能将两者相互印证，看能否在现有平台上借鉴 Spring Cloud 的设计或使用其部分组件。

2016 年 8 月，笔者有幸代表公司参加了全球微服务架构高峰论坛。会上，Josh Long 对 Spring Cloud 的讲解在现场引起了不小的轰动，也让笔者眼前一亮。会后笔者咨询 Josh，Spring Cloud 能否用于生产、是否大规模使用、国内是否已有成功案例，对方一一给出了肯定的回答。这一回答消除了笔者最后的一点疑虑，开始考虑从 Dubbo 逐步迁移至

Spring Cloud 的规划与方案。会后，笔者心想，不妨将 Spring Cloud 相关知识总结成一个“系列博客”，一来是加深自己的理解，二来也算是丰富 Spring Cloud 的中文资料。于是，笔者创建了自己的博客（<http://www.itmuch.com>），并开始了系列博客的编写。写了两篇后，笔者将博客链接分享到微信群中，没成想，恰好被 Josh Long 看到，并引用到 Spring 官方博客中去了。这让笔者感到无比荣幸的同时，也让自己贡献开源社区的欲望空前强烈，于是乎，一口气又写了两篇。

再后来，笔者成立了微服务/Spring Cloud/Docker 相关的 QQ 群（157525002），在 QQ 群小伙伴的鼓励下，笔者决定写一本 Spring Cloud 开源书（<https://github.com/eacdy/spring-cloud-book>），没想到竟然获得开源中国的推荐。再然后，笔者在许进的邀请下，联合创办了 Spring Cloud 中国社区。最后，在群管理员冯靖的引荐下，认识了网红级的大牛张开涛，开涛帮忙引荐了电子工业出版社编辑侠少。从此，笔者正式撰写实体书。

本以为，有了开源书的撰写经验，实体书应该是较为轻松的一件事。然而，样稿发出后，却被侠少鄙视……主要是语文是体育老师教的，病句满天飞，况且，理论不是我的专长。期间一度想要放弃，多亏了侠少的鼓励，总算坚持写了下去……

仓促完稿之际，感慨万千，激动与感激交织，于是，本段不可免俗，进入老生常谈的“鸣谢”环节——感谢我的家人，写书是个费时费力的活儿，在近半年的时间，我的父母和妻子给予了我极大的支持；感谢电子工业出版社小伙伴们的辛苦工作，没有刘佳禾、孙奇俏、侠少等可爱的朋友们，我的书不可能面世；衷心感谢丁露、冯靖、张英磊、王天青、吴峻申（N 本书的作者）在百忙之中帮忙校对；衷心感谢 QQ 群、微信群的朋友们，你们给了笔者最大的帮助和支持！（注：排名不分先后。）

特别鸣谢：感谢吴峻申给笔者提出很多中肯实用的建议和意见；感谢张英磊帮忙重绘、美化书中绝大部分架构图。

谨以此书献给想要学习微服务、Spring Cloud、Docker 又不知从何开始的读者朋友们。希望本书能切切实实地帮助你使用特定技术栈实现微服务架构的落地，也希望本书不会令你失望。本书很多理论性的内容并未展开，例如 Cloud Native、12-factor App、DDD 等，但笔者都在文中以 TIPS、拓展阅读或 WARNING 的形式进行了标记，这部分内容希望读者能够自行拓展阅读。

周立 千米网

排版约定

在阅读本书时，你会遇到不同类型的提示信息，这里展示一些例子及其含义。



推荐阅读按此格式展示。



Tips 按此格式展示。



Warning 按此格式展示。



测试

测试按此格式展示。

读者服务

轻松注册成为博文视点社区用户（www.broadview.com.cn），您即可享受以下服务。

- **下载资源：**本书所提供的示例代码及资源文件均可在【下载资源】处下载。
- **提交勘误：**您对书中内容的修改意见可在【提交勘误】处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **与作者交流：**在页面下方【读者评论】处留下您的疑问或观点，与作者和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/31271>

二维码：



配套代码下载地址

- 1-11 章配套代码:

<https://github.com/itmuch/spring-cloud-docker-microservice-book-code>

- 12-14 章配套代码:

<https://github.com/itmuch/spring-cloud-docker-microservice-book-code-docker>

目录

1 微服务架构概述	1
1.1 单体应用架构存在的问题	1
1.2 如何解决单体应用架构存在的问题	3
1.3 什么是微服务	3
1.4 微服务架构的优点与挑战	4
1.4.1 微服务架构的优点	5
1.4.2 微服务架构面临的挑战	5
1.5 微服务设计原则	6
1.6 如何实现微服务架构	7
1.6.1 技术选型	7
1.6.2 架构图及常用组件	8
2 微服务开发框架——Spring Cloud	9
2.1 Spring Cloud 简介	9
2.2 Spring Cloud 特点	10
2.3 Spring Cloud 版本	10
2.3.1 版本简介	10
2.3.2 子项目一览	11
2.3.3 Spring Cloud/Spring Boot 版本兼容性	12
3 开始使用 Spring Cloud 实战微服务	13
3.1 Spring Cloud 实战前提	13
3.1.1 技术储备	13
3.1.2 工具及软件版本	14
3.2 服务提供者与服务消费者	15

3.3	编写服务提供者	15
3.3.1	手动编写项目	15
3.3.2	使用 Spring Initializr 快速创建 Spring Boot 项目	20
3.4	编写服务消费者	22
3.5	为项目整合 Spring Boot Actuator	23
3.6	硬编码有哪些问题	26
4	微服务注册与发现	27
4.1	服务发现简介	27
4.2	Eureka 简介	29
4.3	Eureka 原理	29
4.4	编写 Eureka Server	31
4.5	将微服务注册到 Eureka Server 上	33
4.6	Eureka Server 的高可用	34
4.7	为 Eureka Server 添加用户认证	37
4.8	Eureka 的元数据	39
4.8.1	改造用户微服务	39
4.8.2	改造电影微服务	39
4.9	Eureka Server 的 REST 端点	41
4.10	Eureka 的自我保护模式	49
4.11	多网卡环境下的 IP 选择	50
4.12	Eureka 的健康检查	51
5	使用 Ribbon 实现客户端侧负载均衡	53
5.1	Ribbon 简介	53
5.2	为服务消费者整合 Ribbon	54
5.3	使用 Java 代码自定义 Ribbon 配置	57
5.4	使用属性自定义 Ribbon 配置	60
5.5	脱离 Eureka 使用 Ribbon	61
6	使用 Feign 实现声明式 REST 调用	63
6.1	Feign 简介	64

6.2	为服务消费者整合 Feign	64
6.3	自定义 Feign 配置.....	66
6.4	手动创建 Feign.....	69
6.4.1	修改用户微服务.....	70
6.4.2	修改电影微服务.....	73
6.5	Feign 对继承的支持	75
6.6	Feign 对压缩的支持	76
6.7	Feign 的日志.....	77
6.8	使用 Feign 构造多参数请求.....	79
6.8.1	GET 请求多参数的 URL	79
6.8.2	POST 请求包含多个参数	81
7	使用 Hystrix 实现微服务的容错处理.....	82
7.1	实现容错的手段	82
7.1.1	雪崩效应.....	83
7.1.2	如何容错.....	83
7.2	使用 Hystrix 实现容错.....	85
7.2.1	Hystrix 简介.....	85
7.2.2	通用方式整合 Hystrix.....	86
7.2.3	Hystrix 断路器的状态监控与深入理解.....	89
7.2.4	Hystrix 线程隔离策略与传播上下文.....	90
7.2.5	Feign 使用 Hystrix	93
7.3	Hystrix 的监控	98
7.4	使用 Hystrix Dashboard 可视化监控数据	100
7.5	使用 Turbine 聚合监控数据.....	102
7.5.1	Turbine 简介	102
7.5.2	使用 Turbine 监控多个微服务.....	103
7.5.3	使用消息中间件收集数据.....	105
8	使用 Zuul 构建微服务网关	110
8.1	为什么要使用微服务网关.....	110
8.2	Zuul 简介.....	112

8.3	编写 Zuul 微服务网关.....	112
8.4	Zuul 的路由端点.....	115
8.5	路由配置详解.....	116
8.6	Zuul 的安全与 Header.....	119
8.6.1	敏感 Header 的设置.....	119
8.6.2	忽略 Header.....	120
8.7	使用 Zuul 上传文件.....	121
8.8	Zuul 的过滤器.....	124
8.8.1	过滤器类型与请求生命周期.....	124
8.8.2	编写 Zuul 过滤器.....	125
8.8.3	禁用 Zuul 过滤器.....	127
8.9	Zuul 的容错与回退.....	127
8.10	Zuul 的高可用.....	130
8.10.1	Zuul 客户端也注册到了 Eureka Server 上.....	130
8.10.2	Zuul 客户端未注册到 Eureka Server 上.....	131
8.11	使用 Sidecar 整合非 JVM 微服务.....	132
8.11.1	编写 Node.js 微服务.....	133
8.11.2	编写 Sidecar.....	134
8.11.3	Sidecar 的端点.....	136
8.11.4	Sidecar 与 Node.js 微服务分离部署.....	136
8.11.5	Sidecar 原理分析.....	137
8.12	使用 Zuul 聚合微服务.....	139
9	使用 Spring Cloud Config 统一管理微服务配置.....	144
9.1	为什么要统一管理微服务配置.....	144
9.2	Spring Cloud Config 简介.....	145
9.3	编写 Config Server.....	146
9.4	编写 Config Client.....	149
9.5	Config Server 的 Git 仓库配置详解.....	151
9.6	Config Server 的健康状况指示器.....	154
9.7	配置内容的加解密.....	155
9.7.1	安装 JCE.....	155