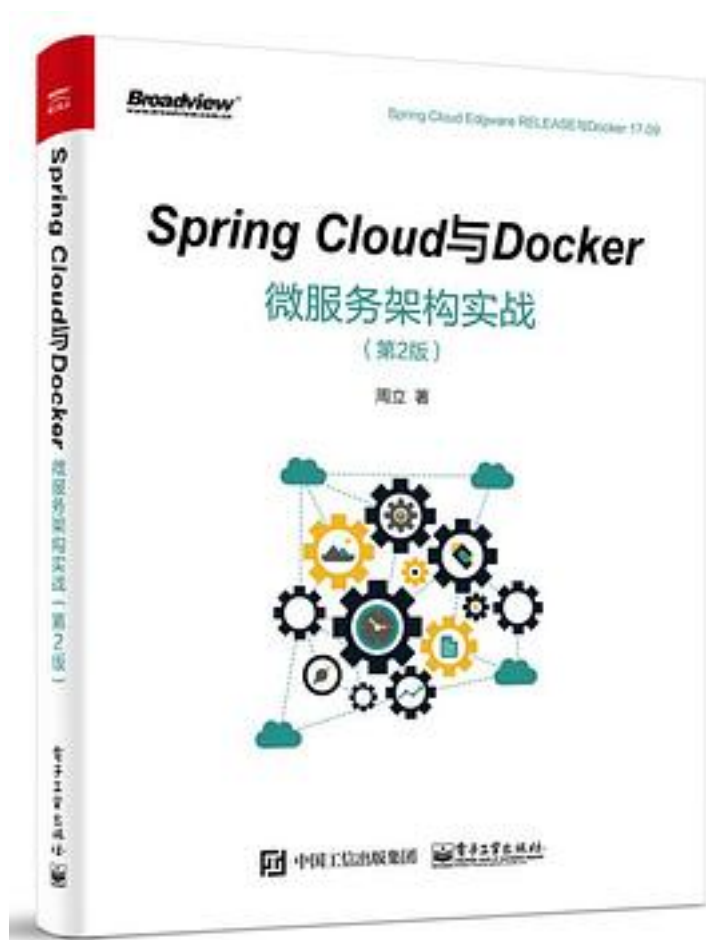


# Spring Cloud与Docker微服务架构实战（第2版）



[Spring Cloud与Docker微服务架构实战（第2版）\\_下载链接1](#)

著者:周立

出版者:电子工业出版社

出版时间:2018-7

装帧:平装

isbn:9787121340154

《Spring Cloud与Docker微服务架构实战（第2版）》基于Spring Cloud Edgware RELEASE 与Docker 17.09，以指导技术团队实现微服务架构落地为宗旨，覆盖微服务理论、微服务开发框

架（Spring Cloud）及运行平台（Docker）三大主题。全书分为3部分，第1部分对微服务架构进行系统介绍；第2部分使用Spring Cloud开发框架编写一个“电影售票系统”；第3部分讲解如何将微服务应用运行在Docker之上。全书采用精心设计的Demo驱动，以连贯的场景、真实的代码示例来引导读者学习用强大的微服务技术实现架构设计。作为第2版，《Spring Cloud与Docker微服务架构实战（第2版）》增补大量新版本框架独有的技术特性及其技巧、用例，以帮助读者避开升级中可能遇到的困难与陷阱。

《Spring Cloud与Docker微服务架构实战（第2版）》适合Java工程师、架构师及运维人员及技术爱好者阅读。

作者介绍:

周立，Spring Cloud中国社区联合发起人。拥有近7年的软件系统开发经验，多年系统架构经验。对Spring Cloud、微服务、持续集成、持续交付有一定见地。

热爱技术交流，曾代表公司参加全球微服务架构高峰论坛、QCon等技术沙龙。拥抱开源，在GitHub与Git@OSC上开源多个项目，并获得了开源中国的推荐，例如开源电子书《使用Spring Cloud与Docker实战微服务》等。

笔者博客：<http://itmuch.com>，定期分享Spring Cloud相关博客。读者可扫码关注Spring Cloud 中国社区公众号以及作者公众号。

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	12
2.3.3 Spring Cloud/Spring Boot 版本兼容性	13
3 开始使用Spring Cloud 实战微服务	14
3.1 Spring Cloud 实战前提	14
3.1.1 技术储备	14
3.1.2 工具及软件版本	15
3.2 服务提供者与服务消费者	16
3.3 编写服务提供者	16
3.3.1 手动编写项目	17
3.3.2 使用Spring Initializr 快速创建Spring Boot 项目	21
3.4 编写服务消费者	23

- 3.5 为项目整合Spring Boot Actuator . 25
- 3.6 硬编码有哪些问题 28
- 4 微服务注册与发现 30
  - 4.1 服务发现简介 . 30
  - 4.2 Eureka 简介 . 32
  - 4.3 Eureka 原理 . 32
  - 4.4 编写Eureka Server 34
  - 4.5 将微服务注册到Eureka Server 上 36
  - 4.6 Eureka Server 的高可用 38
    - 4.6.1 编写高可用Eureka Server 38
    - 4.6.2 将应用注册到Eureka Server 集群上 41
  - 4.7 用户认证 . 41
    - 4.7.1 为Eureka Server 添加用户认证 . 41
    - 4.7.2 将微服务注册到需认证的Eureka Server 43
  - 4.8 Eureka 的元数据 43
    - 4.8.1 改造用户微服务 . 43
    - 4.8.2 改造电影微服务 . 44
  - 4.9 Eureka Server 的REST 端点 46
    - 4.9.1 示例 47
    - 4.9.2 注销微服务实例 . 52
  - 4.10 Eureka 的自我保护模式 . 53
  - 4.11 多网卡环境下的IP 选择 . 54
  - 4.12 Eureka 的健康检查 56
  - 4.13 排除Jersey 依赖 . 57
- 5 使用Ribbon 实现客户端侧负载均衡 . 59
  - 5.1 Ribbon 简介 59
  - 5.2 为服务消费者整合Ribbon . 60
  - 5.3 Ribbon 配置自定义 63
    - 5.3.1 使用Java 代码自定义Ribbon 配置 63
    - 5.3.2 使用属性自定义Ribbon 配置 67
  - 5.4 脱离Eureka 使用Ribbon 68
  - 5.5 饥饿加载 . 70
- 6 使用Feign 实现声明式REST 调用 . 72
  - 6.1 Feign 简介 73
  - 6.2 为服务消费者整合Feign 73
  - 6.3 自定义Feign 配置 . 75
    - 6.3.1 使用Java 代码自定义Feign 配置 . 75
    - 6.3.2 使用属性自定义Feign 配置 79
  - 6.4 手动创建Feign 80
    - 6.4.1 修改用户微服务 . 80
    - 6.4.2 修改电影微服务 . 84
  - 6.5 Feign 对继承的支持 . 86
  - 6.6 Feign 对压缩的支持 . 87
  - 6.7 Feign 的日志 87
    - 6.7.1 编码方式设置日志级别 88
    - 6.7.2 使用属性配置日志级别 89
  - 6.8 使用Feign 构造多参数请求 90
    - 6.8.1 GET 请求多参数的URL . 90
    - 6.8.2 POST 请求包含多个参数 91
  - 6.9 使用Feign 上传文件 92
- 7 使用Hystrix 实现微服务的容错处理 . 94
  - 7.1 实现容错的手段 94
    - 7.1.1 雪崩效应 95
    - 7.1.2 如何容错 95

- 7.2 使用Hystrix 实现容错 . 97
  - 7.2.1 Hystrix 简介 . 97
  - 7.2.2 通用方式整合Hystrix 98
  - 7.2.3 Hystrix 断路器的状态监控与深入理解 102
  - 7.2.4 Hystrix 线程隔离策略与传播上下文 103
  - 7.2.5 Feign 使用Hystrix . 106
- 7.3 Hystrix 的监控 112
- 7.4 使用Hystrix Dashboard 可视化监控数据 . 113
- 7.5 使用Turbine 聚合监控数据 116
  - 7.5.1 Turbine 简介 116
  - 7.5.2 使用Turbine 监控多个微服务 117
  - 7.5.3 使用消息中间件收集数据 119
- 8 使用Zuul 构建微服务网关 124
  - 8.1 为什么要使用微服务网关 . 124
  - 8.2 Zuul 简介 . 126
  - 8.3 编写Zuul 微服务网关 . 126
  - 8.4 管理端点 . 129
    - 8.4.1 routes 端点 129
    - 8.4.2 filters 端点 131
  - 8.5 路由配置详解 . 132
  - 8.6 Zuul 的安全与Header . 136
    - 8.6.1 敏感Header 的设置 . 136
    - 8.6.2 忽略Header . 136
  - 8.7 使用Zuul 上传文件 . 137
  - 8.8 Zuul 的过滤器 140
    - 8.8.1 过滤器类型与请求生命周期 . 140
    - 8.8.2 内置过滤器详解 . 141
    - 8.8.3 编写Zuul 过滤器 144
    - 8.8.4 禁用Zuul 过滤器 146
  - 8.9 Zuul 的容错与回退 146
  - 8.10 饥饿加载 . 149
  - 8.11 Query String 编码 . 150
  - 8.12 Hystrix 隔离策略与线程池 . 150
    - 8.12.1 隔离策略 150
    - 8.12.2 线程池配置 . 151
  - 8.13 Zuul 的高可用 152
    - 8.13.1 Zuul 客户端也注册到了Eureka Server 上 . 152
    - 8.13.2 Zuul 客户端未注册到Eureka Server 上 152
  - 8.14 使用Sidecar 整合非JVM 微服务 . 154
    - 8.14.1 编写Node.js 微服务 . 154
    - 8.14.2 编写Sidecar . 155
    - 8.14.3 Sidecar 的端点 . 157
    - 8.14.4 Sidecar 与Node.js 微服务分离部署 . 158
    - 8.14.5 Sidecar 原理分析 158
  - 8.15 使用Zuul 聚合微服务 . 160
- 9 使用Spring Cloud Config 统一管理微服务配置 166
  - 9.1 为什么要统一管理微服务配置 . 166
  - 9.2 Spring Cloud Config 简介 167
  - 9.3 编写Config Server . 168
  - 9.4 编写Config Client . 171
  - 9.5 Config Server 的Git 仓库配置详解 . 173
    - 9.5.1 占位符支持 . 173
    - 9.5.2 模式匹配 174
    - 9.5.3 搜索目录 174

- 9.5.4 启动时加载配置文件 175
- 9.6 Config Server 的健康状况指示器 . 176
- 9.7 配置内容的加解密 177
  - 9.7.1 安装JCE 177
  - 9.7.2 Config Server 的加解密端点 177
  - 9.7.3 对称加密 177
  - 9.7.4 存储加密的内容 . 178
  - 9.7.5 非对称加密 . 179
- 9.8 使用/refresh 端点手动刷新配置 180
- 9.9 使用Spring Cloud Bus 自动刷新配置 . 181
  - 9.9.1 Spring Cloud Bus 简介 181
  - 9.9.2 实现自动刷新 . 182
  - 9.9.3 局部刷新 183
  - 9.9.4 架构改进 184
  - 9.9.5 跟踪总线事件 . 184
- 9.10 Spring Cloud Config 与Eureka 配合使用 186
- 9.11 Spring Cloud Config 的用户认证 . 187
- 9.12 Config Server 的高可用 188
  - 9.12.1 Git 仓库的高可用 188
  - 9.12.2 RabbitMQ 的高可用 . 189
  - 9.12.3 Config Server 自身的高可用 189
- 10 使用Spring Cloud Sleuth 实现微服务跟踪 . 191
  - 10.1 为什么要实现微服务跟踪 . 191
  - 10.2 Spring Cloud Sleuth 简介 . 192
  - 10.3 整合Spring Cloud Sleuth . 194
  - 10.4 Spring Cloud Sleuth 与ELK 配合使用 . 196
  - 10.5 Spring Cloud Sleuth 与Zipkin 配合使用 . 200
    - 10.5.1 Zipkin 简介 . 200
    - 10.5.2 编写Zipkin Server . 200
    - 10.5.3 微服务整合Zipkin . 202
    - 10.5.4 Zipkin 与Eureka 配合使用 . 205
    - 10.5.5 使用消息中间件收集数据 206
    - 10.5.6 使用Elasticsearch 存储跟踪数据 209
    - 10.5.7 依赖关系图 . 211
- 11 Spring Cloud 常见问题与总结 . 213
  - 11.1 Eureka 常见问题 213
    - 11.1.1 Eureka 注册服务慢 213
    - 11.1.2 已停止的微服务节点注销慢或不注销 214
    - 11.1.3 如何自定义微服务的Instance ID . 215
    - 11.1.4 Eureka 的UNKNOWN 问题总结与解决 217
  - 11.2 整合Hystrix 后首次请求失败 218
    - 11.2.1 原因分析 218
    - 11.2.2 解决方案 218
  - 11.3 Turbine 聚合的数据不完整 219
  - 11.4 Spring Cloud 各组件超时 220
    - 11.4.1 RestTemplate 的超时 . 221
    - 11.4.2 Ribbon 的超时 . 221
    - 11.4.3 Feign 的超时 221
    - 11.4.4 Hystrix 的超时 222
    - 11.4.5 Zuul 的超时 . 222
  - 11.5 Spring Cloud 各组件重试 223
    - 11.5.1 重试步骤 224
    - 11.5.2 基于HTTP 响应码重试 224
    - 11.5.3 关闭重试 224

- 11.5.4 注意点 224
- 11.6 Spring Cloud 各组件调优 225
  - 11.6.1 Tomcat 参数 . 225
  - 11.6.2 Hystrix 参数 . 225
  - 11.6.3 Feign 参数 226
  - 11.6.4 Zuul 参数 . 226
- 11.7 Spring Cloud 各组件配置属性 228
  - 11.7.1 Spring Boot 的配置 228
  - 11.7.2 Spring Cloud 的配置 . 228
  - 11.7.3 原生配置 229
- 11.8 Spring Cloud 定位问题思路总结 . 229
- 12 Docker 入门 232
  - 12.1 Docker 简介 232
  - 12.2 版本与迭代计划 232
    - 12.2.1 版本区别 233
    - 12.2.2 迭代计划 233
  - 12.3 Docker 的架构 234
  - 12.4 安装Docker 236
    - 12.4.1 CentOS 7 安装Docker . 236
    - 12.4.2 Ubuntu 安装Docker . 239
    - 12.4.3 Windows 安装Docker 244
    - 12.4.4 Mac OS 安装Docker . 245
  - 12.5 配置镜像加速器 245
  - 12.6 Docker 常用命令 246
    - 12.6.1 Docker 镜像常用命令 246
    - 12.6.2 Docker 容器常用命令 252
- 13 将微服务运行在Docker 上 260
  - 13.1 使用Dockerfile 构建Docker 镜像 260
    - 13.1.1 Dockerfile 常用指令 . 261
    - 13.1.2 使用Dockerfile 构建镜像 266
  - 13.2 使用Docker Registry 管理Docker 镜像 . 268
    - 13.2.1 使用Docker Hub 管理镜像 . 268
    - 13.2.2 使用私有仓库管理镜像 270
  - 13.3 使用Maven 插件构建Docker 镜像 . 271
    - 13.3.1 快速入门 272
    - 13.3.2 插件读取Dockerfile 进行构建 273
    - 13.3.3 将插件绑定在某个phase 执行 274
    - 13.3.4 推送镜像 275
  - 13.4 常见问题与总结 277
- 14 使用Docker Compose 编排微服务 . 278
  - 14.1 Docker Compose 简介 . 278
  - 14.2 安装Docker Compose . 279
    - 14.2.1 安装Compose . 279
    - 14.2.2 安装Compose 命令补全工具 . 279
  - 14.3 Docker Compose 快速入门 . 280
    - 14.3.1 基本步骤 280
    - 14.3.2 入门示例 280
    - 14.3.3 工程、服务、容器 281
  - 14.4 docker-compose.yml 常用命令 . 281
  - 14.5 docker-compose 常用命令 . 285
  - 14.6 Docker Compose 网络设置 . 286
    - 14.6.1 基本概念 286
    - 14.6.2 更新容器 287
    - 14.6.3 links 287

14.6.4 指定自定义网络 . 288  
14.6.5 配置默认网络 . 289  
14.6.6 使用已存在的网络 289  
14.7 综合实战：使用Docker Compose 编排Spring Cloud 微服务 289  
14.7.1 编排Spring Cloud 微服务 289  
14.7.2 编排高可用的Eureka Server 293  
14.7.3 编排高可用Spring Cloud 微服务集群及动态伸缩 . 295  
14.8 常见问题与总结 297  
后记 298  
附录A：本书配套代码 . 299  
附录B：Spring Cloud YES——快速开发脚手架 300  
附录C：使用Docker 快速安装本书中的组件 301  
· · · · · (收起)

[Spring Cloud与Docker微服务架构实战（第2版）](#) [下载链接1](#)

## 标签

SpringCloud

Docker

技术-SpringCloud

Java

软件开发

微服务

IT

架构

## 评论

比较后悔，半天翻完，没什么技术含量。springcloud可以带你入个门，docker有点凑篇幅（介绍一堆安装，为什么要用docker一点没提）。这些知识点网上一大堆，书应该体现某一个技术的精华和思想，或者说是作者对这个技术的思考抉择。而不是写一堆傻瓜式的操作教程，这样的知识流传不下去的，不是浪费纸吗

-----  
[Spring Cloud与Docker微服务架构实战（第2版）\\_下载链接1\\_](#)

## 书评

这一本书也写微服务架构，不过与挑着写的《未来架构》，扩着写的《持续演进的Cloud Native》，分着写的《云原生应用架构实践》和实着写的《大型企业微服务架构实践与运营》相比，这本书的鲜明特点是小着写和教着写。前面提到的几本书基本都是大场面，无论下笔所触及到的是否是...

-----  
[Spring Cloud与Docker微服务架构实战（第2版）\\_下载链接1\\_](#)